Numerical Linear Algebra and Algorithms

MATLAB Edition

数值线性代数与算法

（MATLAB版）

◎马昌凤 柯艺芬 唐 嘉 陈宝国 编著

国防工业出版社
National Defense Industry Press

责任编辑：丁福志　ding@ndip.cn
责任校对：苏向颖
封面设计：蒋秀芹

本书各章源程序可到国防工业出版社“资源下载”栏目下载，
或发邮件到896369667@qq.com索取

上架建议：数值代数
http://www.ndip.cn
ISBN 978-7-118-11320-4
9 787118 113204
定价：59.00 元

数值线性代数与算法

(MATLAB 版)

马昌凤　柯艺芬
唐　嘉　陈宝国　编著

国防工业出版社
·北京·

内 容 简 介

本书较为系统地介绍了数值线性代数的基本理论、方法及其主要算法的 MATLAB 程序实现. 全书共分为 7 章, 内容包括矩阵代数基础、正交变换和投影方法、线性方程组的矩阵分裂迭代法、线性方程组的 Krylov 子空间迭代法、线性最小二乘问题的数值解法、解线性方程组的直接法和矩阵特征值问题的数值方法. 书中配有丰富的例题和习题, 可供学习者使用. 本书既注意保持理论分析的严谨性, 又注重计算方法的实用性, 强调算法的 MATLAB 程序在计算机上的实现.

本书内容新颖, 叙述流畅, 可作为高等学校数学与应用数学和信息与计算科学专业高年级本科生教材, 特别适用于计算数学专业研究生“数值线性代数”课程的教材或参考书, 也可供理工科其他有关专业的研究生和对数值代数与算法感兴趣的工程技术人员参考使用.

图书在版编目（CIP）数据

数值线性代数与算法：MATLAB 版 / 马昌凤等编著. —北京：国防工业出版社，2017.6
ISBN 978-7-118-11320-4

Ⅰ. ①数… Ⅱ. ①马… Ⅲ. ①Matlab 软件－应用－线性代数计算法 Ⅳ. ①O241.6-39

中国版本图书馆 CIP 数据核字（2017）第 079784 号

※

国防工业出版社出版发行
（北京市海淀区紫竹院南路 23 号 邮政编码 100048）
腾飞印务有限公司印刷
新华书店经售
*

开本 787×1092 1/16 **印张** 26 **字数** 629 千字
2017 年 6 月第 1 版第 1 次印刷 **印数** 1—3000 册 **定价** 59.00 元

国防书店：(010) 88540777 发行邮购：(010) 88540776
发行传真：(010) 88540755 发行业务：(010) 88540717

前　言

数值线性代数是研究代数问题 (线性代数方程组和矩阵特征值问题) 的数值计算方法及其有关理论的一门学科. 可以说科学与工程计算领域的相当一部分应用问题最终都会归结为线性代数方程组的数值求解问题, 或矩阵的特征值与特征向量的计算问题. 数值代数是一门理论性和实际应用性都很强的学科, 并且随着计算机技术的发展, 能够进行数值计算的实际问题的规模不断增大, 相应的数值方法也在不断地改进和创新. 因此, 对于从事科学计算的大学生、研究生甚至工程技术人员来说, 系统地了解和掌握数值代数的基本理论和方法, 特别是新近发展起来且较为成熟的新型算法是非常重要的.

本书系统地介绍了数值线性代数的三大分支, 即线性代数方程组的解法、线性最小二乘法和矩阵特征值问题, 内容包括: 数值线性代数理论基础、正交变换和投影方法、线性方程组的矩阵分裂迭代法、线性方程组的 Krylov 子空间迭代法、线性最小二乘问题的数值解法、解线性方程组的直接法、矩阵特征值问题的数值方法等. 对所讨论的方法, 除了对其收敛性及计算过程的稳定性有较详尽的论述外, 还特别注重这些算法的 MATLAB 程序在计算机上的实现. 本书可作为高等学校数学与应用数学和信息与计算科学专业高年级本科生教材, 特别适用于计算数学专业研究生“数值线性代数”课程的教材或参考书, 也可供理工科其他有关专业的研究生和对数值代数与算法感兴趣的工程技术人员参考使用. 读者只需具备微积分、线性代数和 MATLAB 程序设计方面的初步知识即可顺利阅读.

本书各章节的主要算法都给出了 MATLAB 程序及相应的计算实例. 为了更好地配合教学或自学, 作者编制了与本书配套的电子课件 (PDF 格式的 PPT) 和全部算法的 MATLAB 程序, 需要的读者可到国防工业出版社网站“资源下载”栏目下载 (网址: http://www.ndip.cn), 或发邮件至 896369667@qq.com 索取.

由于作者水平有限, 加之时间仓促, 书中的缺点和错误在所难免, 恳请读者不吝赐教. 来信请发至 macf88@163.com 或 keyifen2017@163.com.

作　者

2017 年 3 月

目 录

第 1 章　数值线性代数理论基础

本章介绍后面章节中需要用到的一些矩阵代数基础知识: 首先介绍一些基本概念和几种常用的矩阵分解; 其次介绍作为数值误差分析度量工具的向量和矩阵范数及其性质; 接着介绍矩阵的广义逆以及几种特殊的矩阵类型; 最后介绍数值代数中常用的模型问题.

1.1　一些概念和记号

本书引用下列记号. 用 $\mathbb{C}^n$ ($\mathbb{R}^n$) 表示 n 维复 (实) 向量空间, $\mathbb{C}^{m\times n}$ ($\mathbb{R}^{m\times n}$) 表示 $m\times n$ 阶复 (实) 矩阵空间, $\mathbb{C}_r^{m\times n}$ ($\mathbb{R}_r^{m\times n}$) 表示秩为 r 的 $m\times n$ 阶复 (实) 矩阵集合. 对于任意的矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, $\boldsymbol{A}^{\mathrm{T}},\boldsymbol{A}^{\mathrm{H}},\boldsymbol{A}^{-1}$ 分别表示矩阵 $\boldsymbol{A}$ 的转置矩阵、共轭转置矩阵和逆矩阵. 对于任意的向量 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{C}^n$, 用 $(\boldsymbol{x},\boldsymbol{y}):=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}$ 表示 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的欧几里得内积.

定义 1.1　设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$. 若存在数 $\lambda\in\mathbb{C}$ 和非零向量 $\boldsymbol{x}\in\mathbb{C}^n$ 使得 $\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}$, 则称 λ 为 $\boldsymbol{A}$ 的特征值, $\boldsymbol{x}$ 为 $\boldsymbol{A}$ 属于 λ 的特征向量.

由定义 1.1 可知, λ 是 $\boldsymbol{A}$ 的特征值当且仅当 $\det(\lambda\boldsymbol{I}-\boldsymbol{A})=0$. 称 $p(\lambda)=\det(\lambda\boldsymbol{I}-\boldsymbol{A})$ 为 $\boldsymbol{A}$ 的特征多项式.

注意到 $\det(\lambda\boldsymbol{I}-\boldsymbol{A}^{\mathrm{T}})=\det(\lambda\boldsymbol{I}-\boldsymbol{A})$, 此即 $\boldsymbol{A}$ 和 $\boldsymbol{A}^{\mathrm{T}}$ 具有相同的特征值. 故存在非零向量 $\boldsymbol{y}\in\mathbb{C}^n$ 满足 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}=\lambda\boldsymbol{y}$, 即 $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}=\lambda\boldsymbol{y}^{\mathrm{T}}$, 称 $\boldsymbol{y}$ 为 $\boldsymbol{A}$ 属于 λ 的左特征向量. 相应地, 将 $\boldsymbol{x}$ 称为 $\boldsymbol{A}$ 属于 λ 的右特征向量. 一般来说, 左右特征向量不相等.

如果 $\boldsymbol{A}$ 有 r 个互不相同的特征值 $\lambda_1,\lambda_2,\cdots,\lambda_r$, 则 $\boldsymbol{A}$ 的特征多项式可表示为

$$\det(\lambda\boldsymbol{I}-\boldsymbol{A})=(\lambda-\lambda_1)^{n_1}(\lambda-\lambda_2)^{n_2}\cdots(\lambda-\lambda_r)^{n_r},$$

称 n_i 为 λ_i 的代数重数. 记 $\gamma_i=n-\operatorname{rank}(\lambda_i\boldsymbol{I}-\boldsymbol{A})$, 称 γ_i 为 λ_i 的几何重数, 它表示属于 λ_i 的线性无关特征向量的个数, 满足 $1\leqslant\gamma_i\leqslant n_i$. 若 $n_i=1$, 则称 λ_i 为 $\boldsymbol{A}$ 的简单特征值. 若 $\gamma_i=n_i$, 则称 λ_i 为 $\boldsymbol{A}$ 的半简单特征值. 显然, 简单特征值必为半简单特征值. 满足 $p(\boldsymbol{A})=\boldsymbol{O}$ 的首项系数为 1 且次数最低的多项式 p 称为 $\boldsymbol{A}$ 的最小多项式. 可以证明, $\boldsymbol{A}$ 的最小多项式具有下列形式, 即

$$p(\lambda)=(\lambda-\lambda_1)^{l_1}(\lambda-\lambda_2)^{l_2}\cdots(\lambda-\lambda_r)^{l_r},\ 1\leqslant l_i\leqslant n_i,\ i=1,2,\cdots,r.$$

定义 1.2　设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$. 若 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}=\boldsymbol{I}$, 则称 $\boldsymbol{A}$ 为酉矩阵. 实的酉矩阵称为正交矩阵, 即 $\boldsymbol{A}$ 满足 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{I}$.

由定义 1.2 显然有, 酉矩阵 $\boldsymbol{A}$ 的逆矩阵为 $\boldsymbol{A}^{-1}=\boldsymbol{A}^{\mathrm{H}}$, 正交矩阵 $\boldsymbol{A}$ 的逆矩阵为 $\boldsymbol{A}^{-1}=\boldsymbol{A}^{\mathrm{T}}$.

由酉矩阵的定义, 容易证明酉矩阵还具有下列性质.

性质 1.1　设 $\boldsymbol{U},\boldsymbol{V}\in\mathbb{C}^{n\times n}$ 是酉矩阵, 则 $|\det(\boldsymbol{U})|=1$, 且 $\boldsymbol{U}^{-1}$ 和 $\boldsymbol{U}\boldsymbol{V}$ 也是酉矩阵.

性质 1.2 $\boldsymbol{U}\in\mathbb{C}^{n\times n}$ 是酉矩阵的充分必要条件是 $\boldsymbol{U}$ 的列向量是两两正交的单位向量.

定义 1.3 设 $\boldsymbol{A},\boldsymbol{B}\in\mathbb{C}^{n\times n}$. 若存在非奇异矩阵 $\boldsymbol{P}$ 使得 $\boldsymbol{A}=\boldsymbol{P}^{-1}\boldsymbol{B}\boldsymbol{P}$, 则称 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 相似. 若 $\boldsymbol{P}$ 为酉矩阵, 则称为酉相似. 若 $\boldsymbol{P}$ 为正交矩阵, 则称为正交相似.

显然, 相似矩阵有相同的特征多项式, 因而有相同的特征值.

定义 1.4 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$. 若 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}=\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$, 则称 $\boldsymbol{A}$ 为正规矩阵. 若 $\boldsymbol{A}^{\mathrm{H}}=\boldsymbol{A}$, 则称 $\boldsymbol{A}$ 为 Hermite 矩阵. 实的 Hermite 矩阵称为实对称矩阵, 即 $\boldsymbol{A}$ 满足 $\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{A}$. 若 $\boldsymbol{A}$ 满足 $\boldsymbol{A}^{\mathrm{H}}=-\boldsymbol{A}$, 则称 $\boldsymbol{A}$ 为反 Hermite 矩阵. 实的反 Hermite 矩阵称为反对称矩阵, 即 $\boldsymbol{A}$ 满足 $\boldsymbol{A}^{\mathrm{T}}=-\boldsymbol{A}$.

容易验证, 酉矩阵、正交矩阵、Hermite 矩阵、反 Hermite 矩阵、实对称矩阵、反对称矩阵都是正规矩阵.

根据反对称矩阵的定义容易证明反对称矩阵的下列性质.

性质 1.3 设 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$ 是反对称矩阵, 即满足 $\boldsymbol{A}^{\mathrm{T}}=-\boldsymbol{A}$. 则

(1) $a_{ii}=0, i=1,2,\cdots,n$.

(2) 不存在奇数阶非奇异反对称矩阵.

(3) $\boldsymbol{A}$ 的特征值只能是 0 或纯虚数.

注 1.1 若 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 为 Hermite 矩阵, 则其对角元为 0 或纯虚数, 其特征值也只能是 0 或纯虚数.

定义 1.5 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$. 若对任意的非零向量 $\boldsymbol{x}\in\mathbb{R}^{n}$ 有 $(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x})>0$ $(\geqslant 0)$, 则称 $\boldsymbol{A}$ 为 (实) 正定矩阵 (半正定矩阵). 若 $\boldsymbol{A}$ 还是对称的, 则称为 (实) 对称正定矩阵 (对称半正定矩阵).

由定义 1.5 不难推得, 对称矩阵的特征值均为实数. 对称正定矩阵 (对称半正定矩阵) 的特征值均为正数 (非负数). 此外, 若记

$$\boldsymbol{H}=\frac{1}{2}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}),\quad \boldsymbol{S}=\frac{1}{2}(\boldsymbol{A}-\boldsymbol{A}^{\mathrm{T}}),$$

分别称为 $\boldsymbol{A}$ 的对称部分和反对称部分, 则显然对任意的矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 都可唯一地分裂为

$$\boldsymbol{A}=\boldsymbol{H}+\boldsymbol{S},$$

称为矩阵 $\boldsymbol{A}$ 的对称–反对称分裂.

性质 1.4 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 正定 (或半正定) 的充分必要条件是其对称部分 $\boldsymbol{H}$ 对称正定 (或对称半正定).

定义 1.6 设 $i_1,i_2,\cdots,i_n$ 是 $1,2,\cdots,n$ 的一个排列, 以 n 阶单位矩阵 $\boldsymbol{I}_n$ 的 n 个列向量

$$\boldsymbol{e}_1=(1,0,\cdots,0)^{\mathrm{T}},\boldsymbol{e}_2=(0,1,\cdots,0)^{\mathrm{T}},\cdots,\boldsymbol{e}_n=(0,0,\cdots,1)^{\mathrm{T}}$$

为列构成的 n 阶矩阵 $\boldsymbol{P}=[\boldsymbol{e}_{i_1},\boldsymbol{e}_{i_2},\cdots,\boldsymbol{e}_{i_n}]$, 称为置换矩阵或排列矩阵.

例如

$$\boldsymbol{P}=[\boldsymbol{e}_3,\boldsymbol{e}_1,\boldsymbol{e}_2]=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}$$

是一个 3 阶置换矩阵.

性质 1.5 置换矩阵具有如下性质:

(1) 置换矩阵的转置仍是置换矩阵.

(2) 置换矩阵是正交矩阵.

(3) 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, $\boldsymbol{P}=[\boldsymbol{e}_{i_1},\boldsymbol{e}_{i_2},\cdots,\boldsymbol{e}_{i_n}]$, 则 $\boldsymbol{P}^{\mathrm{T}}\boldsymbol{A}$ 是将 $\boldsymbol{A}$ 按 $i_1,i_2,\cdots,i_n$ 行重新排列所得到的矩阵, $\boldsymbol{AP}$ 是将 $\boldsymbol{A}$ 按 $i_1,i_2,\cdots,i_n$ 列重新排列得到的矩阵.

定义 1.7 设 $\boldsymbol{A},\boldsymbol{B}\in\mathbb{C}^{n\times n}$. 如果存在 n 阶非奇异矩阵 $\boldsymbol{C}$, 使得

(1) $\boldsymbol{B}=\boldsymbol{C}^{\mathrm{T}}\boldsymbol{AC}$, 则称 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 为 T–合同.

(2) $\boldsymbol{B}=\boldsymbol{C}^{\mathrm{H}}\boldsymbol{AC}$, 则称 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 为 H–合同.

显然, 这两个合同概念具有密切的联系. 当 $\boldsymbol{C}$ 是实矩阵时, T–合同和 H–合同是一致的. 此外, 容易证明, T–合同和 H–合同都是等价关系.

利用合同的定义还可以得到:

(1) 如果 $\boldsymbol{A}$ 是 Hermite 矩阵, 则 $\boldsymbol{C}^{\mathrm{H}}\boldsymbol{AC}$ 也是 Hermite 矩阵 (即使 $\boldsymbol{C}$ 是奇异矩阵).

(2) 如果 $\boldsymbol{A}$ 是对称矩阵 (不一定是实矩阵), 则 $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{AC}$ 也是对称矩阵.

(3) 如果 $\boldsymbol{A}$ 是 Hermite 正定 (半正定) 矩阵, 则 $\boldsymbol{C}^{\mathrm{H}}\boldsymbol{AC}$ 也是 Hermite 正定 (半正定) 矩阵.

(4) 如果 $\boldsymbol{A}$ 是对称正定 (半正定) 矩阵, 则 $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{AC}$ 也是对称正定 (半正定) 矩阵.

定义 1.8 对于 2×2 分块矩阵

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{A}_{11}&\boldsymbol{A}_{12}\\\boldsymbol{A}_{21}&\boldsymbol{A}_{22}\end{bmatrix},\tag{1.1}$$

当 $\boldsymbol{A}_{11}$ 可逆时, 称 $\boldsymbol{A}_{22}-\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}$ 为 $\boldsymbol{A}$ 关于 $\boldsymbol{A}_{11}$ 的 Schur 补, 记为 $\boldsymbol{A}/\boldsymbol{A}_{11}$. 当 $\boldsymbol{A}_{22}$ 可逆时, 称 $\boldsymbol{A}_{11}-\boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}$ 为 $\boldsymbol{A}$ 关于 $\boldsymbol{A}_{22}$ 的 Schur 补, 记为 $\boldsymbol{A}/\boldsymbol{A}_{22}$.

定理 1.1 设 $\boldsymbol{A}$ 具有式 (1.1) 的分块形式, 且 $\boldsymbol{A}_{11}$ 可逆. 则

(1) $\det(\boldsymbol{A})=\det(\boldsymbol{A}_{11})\det(\boldsymbol{A}/\boldsymbol{A}_{11})$.

(2) $\operatorname{rank}(\boldsymbol{A})=\operatorname{rank}(\boldsymbol{A}_{11})+\operatorname{rank}(\boldsymbol{A}/\boldsymbol{A}_{11})$.

证明 注意到

$$\boldsymbol{B}=\begin{bmatrix}\boldsymbol{I}&\boldsymbol{O}\\-\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}&\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_{11}&\boldsymbol{A}_{12}\\\boldsymbol{A}_{21}&\boldsymbol{A}_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{I}&-\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}\\\boldsymbol{O}&\boldsymbol{I}\end{bmatrix}$$

$$=\begin{bmatrix}\boldsymbol{A}_{11}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{A}_{22}-\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}_{11}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{A}/\boldsymbol{A}_{11}\end{bmatrix}.$$

由此立即可得定理的结论. 证毕. □

由定理 1.1 立即可得下面的结论.

推论 1.1 设 $\boldsymbol{A}$ 具有式 (1.1) 的分块形式, 且 $\boldsymbol{A}_{22}$ 可逆. 则

(1) $\det(\boldsymbol{A}) = \det(\boldsymbol{A}_{22})\det(\boldsymbol{A}/\boldsymbol{A}_{22})$.

(2) $\operatorname{rank}(\boldsymbol{A}) = \operatorname{rank}(\boldsymbol{A}_{22}) + \operatorname{rank}(\boldsymbol{A}/\boldsymbol{A}_{22})$.

推论 1.2 设

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}$$

是 Hermite 矩阵, $\boldsymbol{A}$ 关于 $\boldsymbol{A}_{11}$ 的 Schur 补 $\boldsymbol{A}/\boldsymbol{A}_{11} = \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}$. 则

(1) $\boldsymbol{A}$ 正定当且仅当 $\boldsymbol{A}_{11}$ 及 $\boldsymbol{A}/\boldsymbol{A}_{11}$ 均正定.

(2) 若 $\boldsymbol{A}_{11}$ 正定, 则 $\boldsymbol{A}$ 正定当且仅当 $\boldsymbol{A}/\boldsymbol{A}_{11}$ 正定.

1.2 几种常用的矩阵分解

所谓矩阵分解, 就是将一个矩阵分解为 (从某种意义上讲) 比较简单或对其性质比较熟悉的若干个矩阵的乘积. 下面介绍几种常用的矩阵分解.

1.2.1 矩阵的特征分解

特征分解, 又称谱分解, 是将矩阵分解为由其特征值和特征向量表示的矩阵之积的方法. 注意: 只有对可对角化矩阵才可以实施特征分解.

设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 有 n 个线性无关的特征向量 $\boldsymbol{q}_i\,(i = 1, 2, \cdots, n)$. 则 $\boldsymbol{A}$ 可以被分解为

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}, \tag{1.2}$$

式中: $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$, $\lambda_i\ (i = 1, 2, \cdots, n)$ 为 $\boldsymbol{A}$ 的特征值; $\boldsymbol{Q}$ 为 n 阶方阵, 且其第 i 列为 $\boldsymbol{A}$ 的特征向量 $\boldsymbol{q}_i$.

一般来说, 特征向量 $\boldsymbol{q}_i\,(i = 1, 2, \cdots, n)$ 通常被正交单位化 (但这不是必须的). 未被正交单位化的特征向量组 $\boldsymbol{v}_i\,(i = 1, 2, \cdots, n)$ 也可以作为 $\boldsymbol{Q}$ 的列向量.

通常可以通过特征分解来求矩阵的逆. 若矩阵 $\boldsymbol{A}$ 可被特征分解并特征值中不含零, 则矩阵 $\boldsymbol{A}$ 为非奇异矩阵, 且其逆矩阵可以由下式给出:

$$\boldsymbol{A}^{-1} = \boldsymbol{Q}\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}^{-1} = \boldsymbol{Q}\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}^{\mathrm{T}}, \tag{1.3}$$

此处假设 $\boldsymbol{Q}$ 的列向量被正交单位化. 因为 $\boldsymbol{\Lambda}$ 为对角矩阵, 其逆矩阵容易计算出:

$$\boldsymbol{\Lambda}^{-1} = \operatorname{diag}(\lambda_1^{-1}, \lambda_2^{-1}, \cdots, \lambda_n^{-1}).$$

由于任意的 n 阶实对称矩阵 $\boldsymbol{A}$ 都有 n 个线性无关的特征向量, 并且这些特征向量都可以正交单位化而得到一组正交且模为 1 的向量. 故实对称矩阵 $\boldsymbol{A}$ 可被分解成

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}},$$

式中: $\boldsymbol{Q}$ 为正交矩阵; $\boldsymbol{\Lambda}$ 为实对角矩阵.

类似地, 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 为正规矩阵, 由于正规矩阵是可对角化的 (见定理 1.5), 故其具有一组标准正交特征向量基, 因此可被分解成

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{H}},$$

式中: $\boldsymbol{U}$ 为酉矩阵. 进一步地, 若 $\boldsymbol{A}$ 是 Hermite 矩阵, 那么对角矩阵 $\boldsymbol{\Lambda}$ 的对角元全为实数. 若 $\boldsymbol{A}$ 是酉矩阵, 则 $\boldsymbol{\Lambda}$ 的所有对角元在复平面的单位圆上取得.

下面再给出矩阵特征分解的一个应用.

定理 1.2 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 为 Hermite 正定 (半正定) 矩阵, 则存在唯一的 Hermite 正定 (半正定) 矩阵 $\boldsymbol{B}$ 使得

$$\boldsymbol{A} = \boldsymbol{B}^2. \tag{1.4}$$

称这样定义的矩阵 $\boldsymbol{B}$ 为矩阵 $\boldsymbol{A}$ 的平方根矩阵, 常记为 $\boldsymbol{A}^{1/2}$.

证明 只证明正定的情形. 存在性. 由于 $\boldsymbol{A}$ 是 Hermite 正定矩阵, 故存在酉矩阵 $\boldsymbol{P}$ 使得

$$\boldsymbol{A} = \boldsymbol{P}\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{P}^{\mathrm{H}},$$

式中: $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n > 0$ 为 $\boldsymbol{A}$ 的特征值. 令

$$\boldsymbol{B} = \boldsymbol{P}\mathrm{diag}(\lambda_1^{1/2}, \lambda_2^{1/2}, \cdots, \lambda_n^{1/2})\boldsymbol{P}^{\mathrm{H}},$$

则显然 $\boldsymbol{B}$ 为 Hermite 正定矩阵, 且 $\boldsymbol{B}^2 = \boldsymbol{A}$.

唯一性. 假设另有 Hermite 正定矩阵 $\boldsymbol{C}$ 满足 $\boldsymbol{A} = \boldsymbol{C}^2$. 则存在酉矩阵 $\boldsymbol{Q}$ 使得

$$\boldsymbol{C} = \boldsymbol{Q}\mathrm{diag}(\mu_1, \mu_2, \cdots, \mu_n)\boldsymbol{Q}^{\mathrm{H}},$$

式中: $\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_n > 0$ 为 $\boldsymbol{C}$ 的特征值. 由于

$$\boldsymbol{C}^2 = \boldsymbol{Q}\mathrm{diag}(\mu_1^2, \mu_2^2, \cdots, \mu_n^2)\boldsymbol{Q}^{\mathrm{H}} = \boldsymbol{A},$$

故 $\mu_i^2 = \lambda_i\,(i = 1, 2, \cdots, n)$, 即 $\mu_i = \lambda_i^{1/2}$. 于是

$$\boldsymbol{C} = \boldsymbol{Q}\mathrm{diag}(\lambda_1^{1/2}, \lambda_2^{1/2}, \cdots, \lambda_n^{1/2})\boldsymbol{Q}^{\mathrm{H}}.$$

再由 $\boldsymbol{A} = \boldsymbol{B}^2 = \boldsymbol{C}^2$, 得

$$\boldsymbol{P}\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{P}^{\mathrm{H}} = \boldsymbol{Q}\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{Q}^{\mathrm{H}},$$

或等价地, 有

$$\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{P}\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n) = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{P}.$$

记 $\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{P} = (t_{ij})$, 则有

$$t_{ij}\lambda_j = \lambda_i t_{ij},\ i, j = 1, 2, \cdots, n.$$

因此, 当 $\lambda_i \neq \lambda_j$ 时, 必有 $t_{ij} = 0$. 故无论 λ_i 与 λ_j 是否相等, 都有

$$t_{ij}\lambda_j^{1/2} = t_{ij}\lambda_i^{1/2},\ i, j = 1, 2, \cdots, n.$$

即

$$\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{P}\mathrm{diag}(\lambda_1^{1/2}, \lambda_2^{1/2}, \cdots, \lambda_n^{1/2}) = \mathrm{diag}(\lambda_1^{1/2}, \lambda_2^{1/2}, \cdots, \lambda_n^{1/2})\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{P},$$

或者等价地, 有

$$\boldsymbol{P}\mathrm{diag}(\lambda_1^{1/2}, \lambda_2^{1/2}, \cdots, \lambda_n^{1/2})\boldsymbol{P}^{\mathrm{H}} = \boldsymbol{Q}\mathrm{diag}(\lambda_1^{1/2}, \lambda_2^{1/2}, \cdots, \lambda_n^{1/2})\boldsymbol{Q}^{\mathrm{H}},$$

即 $\boldsymbol{B} = \boldsymbol{C}$. 证毕. □

1.2.2 矩阵的 Schur 分解

矩阵的 Schur 分解在理论上十分重要, 它是许多重要定理证明的出发点. 如矩阵论中极为重要的 Hamilton–Cayley (哈密顿–凯莱) 定理, 就可以利用矩阵的 Schur 分解定理进行简洁而优美的证明.

定理 1.3 (Schur 分解定理) 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, 则存在酉矩阵 $\boldsymbol{P} \in \mathbb{C}^{n\times n}$ 使得

$$\boldsymbol{P}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{P} = \boldsymbol{T},$$

式中: $\boldsymbol{T}$ 为上三角矩阵, 其对角元素是 $\boldsymbol{A}$ 的特征值, 而且可以选取 $\boldsymbol{P}$ 使得 $\boldsymbol{T}$ 的对角元可以任意排列.

证明 对矩阵的阶数 n 用数学归纳法. 当 $n = 1$ 时, 结论显然成立. 假设结论对 $n-1$ 阶矩阵成立, 下证对 n 阶矩阵成立.

设 $\boldsymbol{A}$ 的特征值为 $\lambda_1, \lambda_2, \cdots, \lambda_n$, $\boldsymbol{x}_1$ 为对应于特征值 λ_1 的单位特征向量. 将 $\boldsymbol{x}_1$ 扩充为向量空间 $\mathbb{C}^n$ 的一组标准正交基 $\boldsymbol{x}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n$. 定义酉矩阵 $\boldsymbol{P}_0 = [\boldsymbol{x}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n]$, 则

$$\boldsymbol{P}_0^{\mathrm{H}}\boldsymbol{A}\boldsymbol{P}_0 = \begin{bmatrix} \lambda_1 & * \\ \boldsymbol{0} & \boldsymbol{A}_1 \end{bmatrix},$$

式中: $\boldsymbol{A}_1$ 为 $n-1$ 阶方阵, 其特征值为 $\lambda_2, \cdots, \lambda_n$.

由归纳法假设, 存在 $n-1$ 阶酉矩阵 $\boldsymbol{P}_1$ 使得

$$\boldsymbol{P}_1^{\mathrm{H}}\boldsymbol{A}_1\boldsymbol{P}_1 = \boldsymbol{T}_1,$$

式中: $\boldsymbol{T}_1$ 为 $n-1$ 阶上三角矩阵, 其对角元为 $\boldsymbol{A}_1$ 的特征值 $\lambda_2, \cdots, \lambda_n$. 令

$$\boldsymbol{P} = \boldsymbol{P}_0\begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_1 \end{bmatrix},\quad \boldsymbol{T} = \begin{bmatrix} \lambda_1 & * \\ \boldsymbol{0} & \boldsymbol{T}_1 \end{bmatrix}.$$

则有

$$\boldsymbol{P}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{P} = \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_1^{\mathrm{H}} \end{bmatrix}\boldsymbol{P}_0^{\mathrm{H}}\boldsymbol{A}\boldsymbol{P}_0\begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_1 \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_1^{\mathrm{H}} \end{bmatrix}\begin{bmatrix} \lambda_1 & * \\ \boldsymbol{0} & \boldsymbol{A}_1 \end{bmatrix}\begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_1 \end{bmatrix}$$

$$= \begin{bmatrix} \lambda_1 & * \\ \mathbf{0} & \boldsymbol{P}_1^{\mathrm{H}}\boldsymbol{A}_1\boldsymbol{P}_1 \end{bmatrix} = \begin{bmatrix} \lambda_1 & * \\ \mathbf{0} & \boldsymbol{T}_1 \end{bmatrix} = \boldsymbol{T}.$$

由归纳法假设, $\boldsymbol{T}_1$ 的对角元可以任意排列, 故 $\boldsymbol{T}$ 的对角元也可以任意排列. 证毕. □

推论 1.3 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 的特征值都是实数, 则 $\boldsymbol{A}$ 正交相似于上三角矩阵.

设多项式

$$f(\lambda) = b_0\lambda^m + b_1\lambda^{m-1} + \cdots + b_{m-1}\lambda + b_m,$$

对于 n 阶矩阵 $\boldsymbol{A}$, 定义矩阵多项式

$$f(\boldsymbol{A}) = b_0\boldsymbol{A}^m + b_1\boldsymbol{A}^{m-1} + \cdots + b_{m-1}\boldsymbol{A} + b_m\boldsymbol{I},$$

式中: $\boldsymbol{I}$ 为单位矩阵. 如果 $f(\boldsymbol{A}) = \boldsymbol{O}$, 那么称 $\boldsymbol{A}$ 为 $f(\lambda)$ 的矩阵根.

下面利用矩阵的 Schur 分解来证明著名的 Hamilton–Cayley 定理.

定理 1.4 (Hamilton–Cayley 定理) 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $p(\lambda) = \det(\lambda\boldsymbol{I} - \boldsymbol{A})$, 则 $p(\boldsymbol{A}) = \boldsymbol{O}$. 即矩阵 $\boldsymbol{A}$ 的特征多项式是它的零化多项式.

证明 设 n 阶矩阵 $\boldsymbol{A}$ 的特征多项式为

$$p(\lambda) = \det(\lambda\boldsymbol{I} - \boldsymbol{A}) = \lambda^n + a_1\lambda^{n-1} + \cdots + a_{n-1}\lambda + a_n,$$

记 $\boldsymbol{A}$ 的 n 个特征值为 $\lambda_1, \lambda_2, \cdots, \lambda_n$, 则有

$$p(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n),$$

由定理 1.3 可知, 存在酉矩阵 $\boldsymbol{U}$, 使得

$$\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U} = \begin{bmatrix} \lambda_1 & * & \cdots & * \\ & \lambda_2 & \ddots & \vdots \\ & & \ddots & * \\ & & & \lambda_n \end{bmatrix},$$

于是,

$$p(\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U}) = (\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U} - \lambda_1\boldsymbol{I})(\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U} - \lambda_2\boldsymbol{I})\cdots(\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U} - \lambda_n\boldsymbol{I})$$

$$= \begin{bmatrix} 0 & * & \cdots & * \\ & \lambda_2 - \lambda_1 & \ddots & \vdots \\ & & \ddots & * \\ & & & \lambda_n - \lambda_1 \end{bmatrix} \times \begin{bmatrix} \lambda_1 - \lambda_2 & * & \cdots & * \\ & 0 & \ddots & \vdots \\ & & \ddots & * \\ & & & \lambda_n - \lambda_2 \end{bmatrix}$$

$$
\times \begin{bmatrix} \lambda_1-\lambda_3 & * & * & \cdots & * \\ & \lambda_2-\lambda_3 & * & \cdots & * \\ & & 0 & \ddots & \vdots \\ & & & \ddots & * \\ & & & & \lambda_n-\lambda_3 \end{bmatrix} \times \cdots \times \begin{bmatrix} \lambda_1-\lambda_n & * & * & \cdots & * \\ & \lambda_2-\lambda_n & * & \cdots & * \\ & & \ddots & \ddots & \vdots \\ & & & \lambda_{n-1}-\lambda_n & * \\ & & & & 0 \end{bmatrix}
$$

$$
= \begin{bmatrix} 0 & 0 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ 0 & 0 & & \ddots & \vdots \\ 0 & 0 & & & * \end{bmatrix} \times \begin{bmatrix} \lambda_1-\lambda_3 & * & * & \cdots & * \\ & \lambda_2-\lambda_3 & * & \cdots & * \\ & & 0 & \ddots & \vdots \\ & & & \ddots & * \\ & & & & \lambda_n-\lambda_3 \end{bmatrix}
$$

$$
\times \cdots \times \begin{bmatrix} \lambda_1-\lambda_n & * & * & \cdots & * \\ & \lambda_2-\lambda_n & * & \cdots & * \\ & & \ddots & \ddots & \vdots \\ & & & \lambda_{n-1}-\lambda_n & * \\ & & & & 0 \end{bmatrix} = \boldsymbol{O},
$$

即 $\boldsymbol{U}^{\mathrm{H}} p(\boldsymbol{A}) \boldsymbol{U} = \boldsymbol{O}$, 也就是 $p(\boldsymbol{A}) = \boldsymbol{O}$. 证毕. □

在计算数学领域, 人们颇为关心的是, 通过相似变换可以把一个矩阵变换成何种最简单的形状. 利用定理 1.3 和推论 1.3, 可以导出矩阵酉 (正交) 相似于对角矩阵的充要条件.

定理 1.5 矩阵 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 酉相似于对角矩阵当且仅当 $\boldsymbol{A}$ 是正规矩阵, 即 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$. 换言之, 正规矩阵一定可对角化.

证明 必要性. 设酉矩阵 $\boldsymbol{U}$, 使得 $\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U} = \boldsymbol{\Lambda}$, 其中 $\boldsymbol{\Lambda}$ 为对角矩阵, 则有

$$
\begin{aligned}
&\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{H}}, \quad \boldsymbol{A}^{\mathrm{H}} = \boldsymbol{U}\bar{\boldsymbol{\Lambda}}\boldsymbol{U}^{\mathrm{H}}, \\
&\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A} = \boldsymbol{U}\bar{\boldsymbol{\Lambda}}\boldsymbol{U}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{H}} = \boldsymbol{U}\bar{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{H}} = \boldsymbol{U}\boldsymbol{\Lambda}\bar{\boldsymbol{\Lambda}}\boldsymbol{U}^{\mathrm{H}} \\
&\qquad = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{H}}\boldsymbol{U}\bar{\boldsymbol{\Lambda}}\boldsymbol{U}^{\mathrm{H}} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}.
\end{aligned}
$$

充分性. 由 Schur 分解定理可知, 存在酉矩阵 $\boldsymbol{U}$, 使得

$$
\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ & t_{22} & \cdots & t_{2n} \\ & & \ddots & \vdots \\ & & & t_{nn} \end{bmatrix} := \boldsymbol{T}.
$$

利用条件 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}=\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$, 得

$$\begin{aligned}\boldsymbol{T}^{\mathrm{H}}\boldsymbol{T}&=\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U}=\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U}\\&=\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{U}=\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U}\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{U}=\boldsymbol{T}\boldsymbol{T}^{\mathrm{H}}.\end{aligned}$$

比较上式两端矩阵的对应元素, 可得 $t_{ij}=0\ (i<j)$, 也就是 $\boldsymbol{T}=\mathrm{diag}(t_{11},t_{22},\cdots,t_{nn})$, 即 $\boldsymbol{A}$ 酉相似于对角矩阵. 证毕. □

推论 1.4 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 的 n 个特征值都是实数, 则 $\boldsymbol{A}$ 正交相似于对角矩阵的充要条件是 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$.

推论 1.5 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 为正规矩阵, λ 是 $\boldsymbol{A}$ 的特征值, $\boldsymbol{x}$ 是对应的特征向量. 则 $\overline{\lambda}$ 是 $\boldsymbol{A}^{\mathrm{H}}$ 的特征值, $\overline{\lambda}$ 对应的特征向量仍然是 $\boldsymbol{x}$.

证明 由定理 1.5, 存在 n 阶酉矩阵 $\boldsymbol{U}$ 使得 $\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$. 于是有 $\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{U}=\mathrm{diag}(\overline{\lambda}_1,\overline{\lambda}_2,\cdots,\overline{\lambda}_n)$. 设 $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n]$, 则上面两式可写为

$$\boldsymbol{A}\boldsymbol{u}_i=\lambda_i\boldsymbol{u}_i,\quad \boldsymbol{A}^{\mathrm{H}}\boldsymbol{u}_i=\overline{\lambda}_i\boldsymbol{u}_i,\quad i=1,2,\cdots,n.$$

由此可见, 若 λ_i 是 $\boldsymbol{A}$ 的特征值且 $\boldsymbol{u}_i$ 是相应的特征向量时, $\overline{\lambda}_i$ 是 $\boldsymbol{A}^{\mathrm{H}}$ 的特征值且相应的特征向量仍是 $\boldsymbol{u}_i$. 证毕. □

推论 1.6 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 为正规矩阵, λ,μ 是 $\boldsymbol{A}$ 的特征值, $\boldsymbol{x},\boldsymbol{y}$ 是对应的特征向量. 如果 $\lambda\neq\mu$, 则 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 正交, 即 $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}=0$.

证明 因为 $\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}$, $\boldsymbol{A}\boldsymbol{y}=\mu\boldsymbol{y}$. 由推论 1.5 得 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{x}=\overline{\lambda}\boldsymbol{x}$. 故

$$\overline{\mu}\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}=(\mu\boldsymbol{y})^{\mathrm{H}}\boldsymbol{x}=(\boldsymbol{A}\boldsymbol{y})^{\mathrm{H}}\boldsymbol{x}=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{x}=\overline{\lambda}\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x},$$

即 $\overline{(\lambda-\mu)}\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}=0$. 由题设 $\lambda\neq\mu$, 故必有 $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}=0$. 证毕. □

推论 1.7 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, 则 $\boldsymbol{A}$ 酉相似于实对角矩阵的充要条件是 $\boldsymbol{A}$ 为 Hermite 矩阵.

证明 必要性. 设存在 n 阶酉矩阵 $\boldsymbol{U}$ 使得

$$\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)=\boldsymbol{\Lambda},$$

式中: $\lambda_i\,(i=1,2,\cdots,n)$ 为实数, 则有

$$\boldsymbol{A}^{\mathrm{H}}=(\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{H}})^{\mathrm{H}}=\boldsymbol{U}\boldsymbol{\Lambda}^{\mathrm{H}}\boldsymbol{U}^{\mathrm{H}}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{H}}=\boldsymbol{A},$$

即 $\boldsymbol{A}$ 是 Hermite 矩阵.

充分性. 若 $\boldsymbol{A}$ 是 Hermite 矩阵, 则 $\boldsymbol{A}$ 为正规矩阵, 于是存在酉矩阵 $\boldsymbol{U}$ 使得

$$\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)=\boldsymbol{\Lambda},$$

故

$$\boldsymbol{\Lambda}^{\mathrm{H}}=(\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U})^{\mathrm{H}}=\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{U}=\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U}=\boldsymbol{\Lambda}.$$

从而 $\overline{\lambda}_i=\lambda_i\,(i=1,2,\cdots,n)$, 即 λ_i 都是实数. 证毕. □

推论 1.8 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 是反 Hermite 矩阵, 则存在 n 阶酉矩阵 $\boldsymbol{U}$ 使得

$$\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{U}=\mathrm{diag}(\mathrm{i}b_1,\mathrm{i}b_2,\cdots,\mathrm{i}b_n),$$

式中: $b_i\,(i=1,2,\cdots,n)$ 为实数.

证明 由于反 Hermite 矩阵是正规矩阵且其特征值为 0 或纯虚数, 由定理 1.5 即得结论. 证毕. □

下面介绍两个 n 阶实对称矩阵“同时”相似于对角矩阵的问题.

定理 1.6 设 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 都是 n 阶实对称矩阵, 则存在正交矩阵 $\boldsymbol{P}$, 使得 $\boldsymbol{P}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P}$ 和 $\boldsymbol{P}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{P}$ 都是对角矩阵的充要条件是 $\boldsymbol{A}\boldsymbol{B}=\boldsymbol{B}\boldsymbol{A}$.

证明 必要性. 设 n 阶正交矩阵 $\boldsymbol{P}$ 使得

$$\boldsymbol{P}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P}=\boldsymbol{\Lambda},\quad \boldsymbol{P}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{P}=\boldsymbol{\Sigma},$$

式中: $\boldsymbol{\Lambda}$ 和 $\boldsymbol{\Sigma}$ 为对角矩阵. 则有

$$\begin{aligned}\boldsymbol{A}\boldsymbol{B}&=\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{P}^{\mathrm{T}}=\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{P}^{\mathrm{T}}\\&=\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{\Lambda}\boldsymbol{P}^{\mathrm{T}}=\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{\mathrm{T}}=\boldsymbol{B}\boldsymbol{A}.\end{aligned}$$

充分性. 设 $\boldsymbol{A}$ 的全体互异特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_r$, 相应的重数为 $n_1,n_2,\cdots,n_r$ $(n_1+n_2+\cdots+n_r=n)$. 由 $\boldsymbol{A}$ 实对称知, 存在正交矩阵 $\boldsymbol{Q}$ 使得

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\begin{bmatrix}\lambda_1\boldsymbol{I}_{n_1}&&&\\&\lambda_2\boldsymbol{I}_{n_2}&&\\&&\ddots&\\&&&\lambda_r\boldsymbol{I}_{n_r}\end{bmatrix},$$

将矩阵 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{Q}$ 划分为如下分块形式

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{B}_{11}&\boldsymbol{B}_{12}&\cdots&\boldsymbol{B}_{1r}\\\boldsymbol{B}_{21}&\boldsymbol{B}_{22}&\cdots&\boldsymbol{B}_{2r}\\\vdots&\vdots&\ddots&\vdots\\\boldsymbol{B}_{r1}&\boldsymbol{B}_{r2}&\cdots&\boldsymbol{B}_{rr}\end{bmatrix},$$

式中: $\boldsymbol{B}_{ij}$ 为 $n_i\times n_j$ 阶矩阵 $(i,j=1,2,\cdots,r)$.

由 $\boldsymbol{A}\boldsymbol{B}=\boldsymbol{B}\boldsymbol{A}$ 及 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}=\boldsymbol{I}$, 得

$$(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q})(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{Q})=(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{Q})(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}),$$

即

$$\begin{bmatrix} \lambda_1\boldsymbol{B}_{11} & \lambda_1\boldsymbol{B}_{12} & \cdots & \lambda_1\boldsymbol{B}_{1r} \\ \lambda_2\boldsymbol{B}_{21} & \lambda_2\boldsymbol{B}_{22} & \cdots & \lambda_2\boldsymbol{B}_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_r\boldsymbol{B}_{r1} & \lambda_r\boldsymbol{B}_{r2} & \cdots & \lambda_r\boldsymbol{B}_{rr} \end{bmatrix} = \begin{bmatrix} \lambda_1\boldsymbol{B}_{11} & \lambda_2\boldsymbol{B}_{12} & \cdots & \lambda_r\boldsymbol{B}_{1r} \\ \lambda_1\boldsymbol{B}_{21} & \lambda_2\boldsymbol{B}_{22} & \cdots & \lambda_r\boldsymbol{B}_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1\boldsymbol{B}_{r1} & \lambda_2\boldsymbol{B}_{r2} & \cdots & \lambda_r\boldsymbol{B}_{rr} \end{bmatrix},$$

也就是

$$\lambda_i\boldsymbol{B}_{ij} = \lambda_j\boldsymbol{B}_{ij} \quad (i,j = 1,2,\cdots,r).$$

当 $i \neq j$ 时, 由 $\lambda_i \neq \lambda_j$ 可得 $\boldsymbol{B}_{ij} = \boldsymbol{O}$ $(i \neq j;\ i,j = 1,2,\cdots,r)$. 于是

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{B}_{11},\boldsymbol{B}_{22},\cdots,\boldsymbol{B}_{rr}),$$

因为 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{Q}$ 实对称, 所以 $\boldsymbol{B}_{ii}\,(i = 1,2,\cdots,r)$ 实对称, 那么存在正交矩阵 $\boldsymbol{U}_i$, 使得 $\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{B}_{ii}\boldsymbol{U}_i = \boldsymbol{\Sigma}_i\,(i = 1,2,\cdots,r)$, 其中 $\boldsymbol{\Sigma}_i$ 为对角矩阵. 令 $\boldsymbol{U} = \mathrm{diag}(\boldsymbol{U}_1,\boldsymbol{U}_2,\cdots,\boldsymbol{U}_r)$, 则 $\boldsymbol{U}$ 为正交矩阵, 从而 $\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{U}$ 也为正交矩阵, 且有

$$\begin{aligned} \boldsymbol{P}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{P} &= \boldsymbol{U}^{\mathrm{T}}(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{Q})\boldsymbol{U} = \mathrm{diag}(\boldsymbol{\Sigma}_1,\boldsymbol{\Sigma}_2,\cdots,\boldsymbol{\Sigma}_r), \\ \boldsymbol{P}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P} &= \boldsymbol{U}^{\mathrm{T}}(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q})\boldsymbol{U} = \mathrm{diag}(\lambda_1\boldsymbol{I}_{n_1},\lambda_2\boldsymbol{I}_{n_2},\cdots,\lambda_r\boldsymbol{I}_{n_r}). \end{aligned}$$

证毕. □

定理 1.6 给出了两个实对称矩阵“同时”正交相似于对角矩阵的充要条件, 下面再给出两个 Hermite 矩阵“同时”合同于对角矩阵的充分条件.

定理 1.7 *设 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 都是 n 阶 Hermite 矩阵, 且 $\boldsymbol{B}$ 为正定矩阵, 则存在可逆矩阵 $\boldsymbol{P}$, 使得 $\boldsymbol{P}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{P}$ 和 $\boldsymbol{P}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{P}$ 都是对角矩阵.*

证明 由 $\boldsymbol{B}$ 正定知, 存在酉矩阵 $\boldsymbol{Q}_1$, 使得

$$\boldsymbol{Q}_1^{\mathrm{H}}\boldsymbol{B}\boldsymbol{Q}_1 = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} = \boldsymbol{\Lambda}_1, \quad (\lambda_i > 0),$$

令

$$\boldsymbol{Q}_2 = \boldsymbol{Q}_1 \begin{bmatrix} 1/\sqrt{\lambda_1} & & \\ & \ddots & \\ & & 1/\sqrt{\lambda_n} \end{bmatrix},$$

则 $\boldsymbol{Q}_2$ 可逆, 且有 $\boldsymbol{Q}_2^{\mathrm{H}}\boldsymbol{B}\boldsymbol{Q}_2 = \boldsymbol{I}$. 对于 Hermite 矩阵 $\boldsymbol{Q}_2^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_2$, 存在酉矩阵 $\boldsymbol{Q}_3$, 使得

$$\boldsymbol{Q}_3^{\mathrm{H}}(\boldsymbol{Q}_2^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_2)\boldsymbol{Q}_3 = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_n \end{bmatrix} = \boldsymbol{\Lambda}_2.$$

令 $\boldsymbol{P}=\boldsymbol{Q}_2\boldsymbol{Q}_3$, 则 $\boldsymbol{P}$ 可逆, 且有

$$\boldsymbol{P}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{P}=\boldsymbol{\Lambda}_2,\quad \boldsymbol{P}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{P}=\boldsymbol{I},$$

式中: $\boldsymbol{\Lambda}_2$ 和 $\boldsymbol{I}$ 为对角矩阵. 证毕. □

1.2.3 矩阵的奇异值分解

矩阵的奇异值分解在理论上和实际应用中都十分重要. 特别地, 它已成为信息处理、多元统计分析等工程技术领域中不可缺少的工具. 由 1.2.2 节的 Schur 分解定理可知, 任一方阵用酉相似变换只能约化为上三角矩阵, 不能约化为对角矩阵. 而奇异值分解定理表明: 用两个酉矩阵乘到一个矩阵 $\boldsymbol{A}$ 的两边就可以变为对角矩阵. 奇异值分解常用于奇异的或数值上非常接近奇异的矩阵计算. 它不仅能判断矩阵是否接近奇异, 而且也用于数值求解.

对于任意的 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$, 容易验证:

(1) $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 是 Hermite (半) 正定矩阵.

(2) 齐次线性方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0}$ 与 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0}$ 同解.

(3) $\mathrm{rank}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})=\mathrm{rank}(\boldsymbol{A})$.

(4) $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}=\boldsymbol{O}\Longleftrightarrow\boldsymbol{A}=\boldsymbol{O}$.

定义 1.9 设 $\boldsymbol{A}\in\mathbb{C}_r^{m\times n}(r\geqslant 1)$, 记 Hermite 矩阵 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 的 n 个特征值为 $\lambda_1\geqslant\lambda_2\geqslant\cdots\lambda_r>\lambda_{r+1}=\cdots=\lambda_n=0$, 则称

$$\sigma_i=\sqrt{\lambda_i},\quad i=1,2,\cdots,n$$

为 $\boldsymbol{A}$ 的奇异值.

容易看出, $\boldsymbol{A}$ 的奇异值的个数与 $\boldsymbol{A}$ 的列数相同, $\boldsymbol{A}$ 的正奇异值的个数与 $\boldsymbol{A}$ 的秩相同.

定理 1.8 (奇异值分解定理) 设 $\boldsymbol{A}\in\mathbb{C}_r^{m\times n}$ 的正奇异值为 $\sigma_1,\sigma_2,\cdots,\sigma_r$. 则存在 m 阶酉矩阵 $\boldsymbol{U}$ 和 n 阶酉矩阵 $\boldsymbol{V}$, 使得

$$\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}},\tag{1.5}$$

式中: $\boldsymbol{\Sigma}$ 为 $m\times n$ 阶对角矩阵, 且

$$\boldsymbol{\Sigma}=\begin{array}{c}\left[\begin{array}{cc}\boldsymbol{\Sigma}_r & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O}\end{array}\right]\begin{array}{c}r\\ m-r\end{array}\\ \begin{array}{cc}r & n-r\end{array}\end{array},\tag{1.6}$$

这里 $\boldsymbol{\Sigma}_r=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_r)$, $\sigma_i>0$, $1\leqslant i\leqslant r$, $r\leqslant\min\{m,n\}$.

证明 对任意的 $\boldsymbol{A}\in\mathbb{C}_r^{m\times n}$, $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 是 n 阶 Hermite 半正定矩阵, 它的 n 个非负特征值记为 $\sigma_1^2,\sigma_2^2,\cdots,\sigma_n^2$. 不妨设 $\sigma_1^2\geqslant\sigma_2^2\geqslant\cdots\geqslant\sigma_r^2>0$, $\sigma_{r+1}^2=\sigma_{r+2}^2=\cdots=\sigma_n^2=0$, 这里 $r\leqslant\min\{m,n\}$. $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 的特征分解为

$$\boldsymbol{V}^{\mathrm{H}}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})\boldsymbol{V}=\mathrm{diag}(\sigma_1^2,\sigma_2^2,\cdots,\sigma_n^2),\tag{1.7}$$

式中: $\boldsymbol{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n]$ 为 n 阶酉矩阵.

划分 $\boldsymbol{V}=[\boldsymbol{V}_1,\boldsymbol{V}_2]$, 其中 $\boldsymbol{V}_1=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_r]$, $\boldsymbol{V}_2=[\boldsymbol{v}_{r+1},\boldsymbol{v}_{r+2},\cdots,\boldsymbol{v}_n]$, 令 $\boldsymbol{\Sigma}_r=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_r)$, 则有

$$\begin{aligned}\boldsymbol{V}^{\mathrm{H}}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})\boldsymbol{V}&=\begin{bmatrix}\boldsymbol{V}_1^{\mathrm{H}}\\ \boldsymbol{V}_2^{\mathrm{H}}\end{bmatrix}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})[\boldsymbol{V}_1,\boldsymbol{V}_2]\\&=\begin{bmatrix}\boldsymbol{V}_1^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_1 & \boldsymbol{V}_1^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_2\\ \boldsymbol{V}_2^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_1 & \boldsymbol{V}_2^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_2\end{bmatrix}.\end{aligned}\tag{1.8}$$

对比式 (1.7) 和式 (1.8), 得

$$\begin{aligned}&\boldsymbol{V}_1^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_1=\boldsymbol{\Sigma}_r^2,\quad \boldsymbol{V}_2^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_2=\boldsymbol{O},\\&\boldsymbol{V}_1^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_2=\boldsymbol{O},\quad \boldsymbol{V}_2^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_1=\boldsymbol{O}.\end{aligned}\tag{1.9}$$

由式 (1.9) 的第 1, 2 个等式可知 $\boldsymbol{A}\boldsymbol{V}_1$ 的列互相正交, $\boldsymbol{A}\boldsymbol{V}_2$ 的列为零向量. 取 $m\times r$ 阶矩阵

$$\boldsymbol{U}_1=\boldsymbol{A}\boldsymbol{V}_1\boldsymbol{\Sigma}_r^{-1},\tag{1.10}$$

则 $\boldsymbol{U}_1^{\mathrm{H}}\boldsymbol{U}_1=\boldsymbol{I}_r$. 再取 $\boldsymbol{U}_2\in\mathbb{C}^{m\times(n-r)}$, 使得 $\boldsymbol{U}=[\boldsymbol{U}_1,\boldsymbol{U}_2]$ 为酉矩阵. 则由 $\boldsymbol{A}\boldsymbol{V}_2=\boldsymbol{O}$ 及 $\boldsymbol{U}_2^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_1=\boldsymbol{U}_2^{\mathrm{H}}(\boldsymbol{U}_1\boldsymbol{\Sigma}_r)=(\boldsymbol{U}_2^{\mathrm{H}}\boldsymbol{U}_1)\boldsymbol{\Sigma}_r=\boldsymbol{O}$, 立即推得

$$\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}=\begin{bmatrix}\boldsymbol{U}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_1 & \boldsymbol{U}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_2\\ \boldsymbol{U}_2^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_1 & \boldsymbol{U}_2^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_2\end{bmatrix}=\begin{bmatrix}\boldsymbol{\Sigma}_r & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O}\end{bmatrix}=\boldsymbol{\Sigma}.$$

证毕. □

在奇异值分解式 (1.5) 中, $\boldsymbol{U}$ 的第 i 列是 $\boldsymbol{A}$ 的对应于奇异值 σ_i 的左奇异向量, $\boldsymbol{V}$ 的第 i 列是 $\boldsymbol{A}$ 的对应于奇异值 σ_i 的右奇异向量. 从定理的证明过程不难看出, $\boldsymbol{A}$ 的奇异值由 $\boldsymbol{A}$ 唯一确定, 但对应于每个奇异值的奇异向量一般不是唯一的.

注 1.2 若 $\boldsymbol{A}\in\mathbb{R}_r^{m\times n}$, 则定理 1.8 中的 $\boldsymbol{U},\boldsymbol{V}$ 都可以取为实矩阵, 即 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 分别为 m 阶和 n 阶的正交矩阵.

两个与矩阵 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$ 有关的重要子空间是其列空间和零空间.

定义 1.10 设矩阵 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$, 分别称

$$\mathcal{R}(\boldsymbol{A})=\{\boldsymbol{y}:\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x},\boldsymbol{x}\in\mathbb{C}^n\}\subset\mathbb{C}^m,\quad \mathcal{N}(\boldsymbol{A})=\{\boldsymbol{x}:\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0},\boldsymbol{x}\in\mathbb{C}^n\}\subset\mathbb{C}^n$$

为 $\boldsymbol{A}$ 的列空间 (值域) 和零空间 (核).

容易验证: 若将 $\boldsymbol{A}$ 按列划分为 $\boldsymbol{A}=[\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n]$, 则

$$\mathcal{R}(\boldsymbol{A})=\mathrm{span}\{\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n\}.$$

给定两个向量子空间 $\mathcal{S}_1$ 和 $\mathcal{S}_2$, 它们的和 $\mathcal{S}$ 是一个子空间, 其每一个向量都是 $\mathcal{S}_1$ 的一个向量与 $\mathcal{S}_2$ 的一个向量之和. 两个子空间的交也是一个子空间. 如果 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 的交退化为 $\{\boldsymbol{0}\}$, 则 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 之和称为直和, 表示为 $\mathcal{S}=\mathcal{S}_1\oplus\mathcal{S}_2$. 当 $\mathcal{S}$ 等于 $\mathbb{C}^n$ 时, 则 $\mathbb{C}^n$ 的每一个向量 $\boldsymbol{x}$ 可被唯一地写成 $\boldsymbol{x}=\boldsymbol{x}_1+\boldsymbol{x}_2$, 其中 $\boldsymbol{x}_1\in\mathcal{S}_1, \boldsymbol{x}_2\in\mathcal{S}_2$.

关于矩阵 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$ 的列空间和零空间有下面的正交分解:

$$\mathcal{R}(\boldsymbol{A})\oplus\mathcal{N}(\boldsymbol{A}^{\mathrm{H}})=\mathbb{C}^m,\quad \mathcal{N}(\boldsymbol{A})\oplus\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})=\mathbb{C}^n.$$

此外, 当 $\boldsymbol{A}\mathcal{S}\subset\mathcal{S}$ 时, 称子空间 $\mathcal{S}$ 在矩阵 $\boldsymbol{A}$ 之下是不变的. 易见, 矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的列空间 $\mathcal{R}(\boldsymbol{A})$ 和零空间 $\mathcal{N}(\boldsymbol{A})$ 都是 $\boldsymbol{A}$ 的不变子空间. 特别地, 对 $\boldsymbol{A}$ 的任意特征值 λ, 子空间 $\mathcal{N}(\boldsymbol{A}-\lambda\boldsymbol{I})$ 在 $\boldsymbol{A}$ 下不变. 子空间 $\mathcal{N}(\boldsymbol{A}-\lambda\boldsymbol{I})$ 称为相应于 λ 的特征子空间, 它包含零向量及 $\boldsymbol{A}$ 的所有相应于 λ 的特征向量.

定理 1.9 在 $\boldsymbol{A}$ 的奇异值分解式 (1.5) 中, 记 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 的列向量分别为 $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m$ 和 $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n$, 则

(1) $\boldsymbol{A}$ 的秩等于其非零奇异值的个数.

(2) $\|\boldsymbol{A}\|_2=\max\limits_{1\leqslant i\leqslant n}\sigma_i$. 设 σ_1 为最大奇异值, 其对应的右奇异向量为 $\boldsymbol{v}_1$, 则 $\|\boldsymbol{A}\|_2=\|\boldsymbol{A}\boldsymbol{v}_1\|_2$.

(3) $\mathcal{N}(\boldsymbol{A})=\operatorname{span}\{\boldsymbol{v}_{r+1}, \boldsymbol{v}_{r+2}, \cdots, \boldsymbol{v}_n\}$.

(4) $\mathcal{R}(\boldsymbol{A})=\operatorname{span}\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r\}$.

(5) $\boldsymbol{A}=\sum\limits_{i=1}^{r}\sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^{\mathrm{H}}$.

证明 沿用定理 1.8 中的记号, 式 (1.5) 可写为

$$\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}=[\boldsymbol{U}_1, \boldsymbol{U}_2]\begin{bmatrix}\boldsymbol{\Sigma}_r & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_1^{\mathrm{H}}\\ \boldsymbol{V}_2^{\mathrm{H}}\end{bmatrix}=\boldsymbol{U}_1\boldsymbol{\Sigma}_r\boldsymbol{V}_1^{\mathrm{H}}. \tag{1.11}$$

(1) 由于 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 都是非奇异矩阵, 故立即有 $\operatorname{rank}(\boldsymbol{A})=\operatorname{rank}(\boldsymbol{\Sigma})=r$.

(2) 注意到 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 都是酉矩阵, 故有

$$\begin{aligned}\|\boldsymbol{A}\|_2&=\max_{\|\boldsymbol{x}\|_2=1}\|\boldsymbol{A}\boldsymbol{x}\|_2=\max_{\|\boldsymbol{x}\|_2=1}\|\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}\boldsymbol{x}\|_2=\max_{\|\boldsymbol{y}\|_2=1}\|\boldsymbol{\Sigma}\boldsymbol{y}\|_2\\&=\max\left\{\left[(\sigma_1y_1)^2+(\sigma_2y_2)^2+\cdots+(\sigma_ny_n)^2\right]^{1/2}:\|\boldsymbol{y}\|_2=1\right\}\\&=\max_{1\leqslant i\leqslant n}\sigma_i.\end{aligned}$$

设 σ_1 为最大奇异值, 其对应的右奇异向量为 $\boldsymbol{v}_1$, 则

$$\|\boldsymbol{A}\boldsymbol{v}_1\|_2=\|\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}\boldsymbol{v}_1\|_2=\|\boldsymbol{\Sigma}\boldsymbol{e}_1\|_2=\sigma_1=\|\boldsymbol{A}\|_2.$$

(3) 有

$$\mathcal{N}(\boldsymbol{A})=\{\boldsymbol{x}:\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0}\}=\{\boldsymbol{x}:\boldsymbol{U}_1\boldsymbol{\Sigma}_r\boldsymbol{V}_1^{\mathrm{H}}\boldsymbol{x}=\boldsymbol{0}\}=\{\boldsymbol{x}:\boldsymbol{V}_1^{\mathrm{H}}\boldsymbol{x}=\boldsymbol{0}\}$$

$$= \{\boldsymbol{x} : \boldsymbol{x} = k_{r+1}\boldsymbol{v}_{r+1} + \cdots + k_n\boldsymbol{v}_n, k_i \in \mathbb{C}\}$$
$$= \operatorname{span}\{\boldsymbol{v}_{r+1}, \boldsymbol{v}_{r+2}, \cdots, \boldsymbol{v}_n\}.$$

(4) 由于

$$\mathcal{R}(\boldsymbol{A}) = \{\boldsymbol{y} : \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}\} = \{\boldsymbol{y} : \boldsymbol{y} = \boldsymbol{U}_1(\boldsymbol{\Sigma}_r\boldsymbol{V}_1^{\mathrm{H}}\boldsymbol{x})\} \subset \mathcal{R}(\boldsymbol{U}_1),$$
$$\mathcal{R}(\boldsymbol{U}_1) = \{\boldsymbol{y} : \boldsymbol{y} = \boldsymbol{U}_1\boldsymbol{z}\} = \{\boldsymbol{y} : \boldsymbol{y} = (\boldsymbol{U}_1\boldsymbol{\Sigma}_r\boldsymbol{V}_1^{\mathrm{T}})(\boldsymbol{V}_1\boldsymbol{\Sigma}_r^{-1}\boldsymbol{z})\}$$
$$= \{\boldsymbol{y} : \boldsymbol{y} = \boldsymbol{A}(\boldsymbol{V}_1\boldsymbol{\Sigma}_r^{-1}\boldsymbol{z})\} \subset \mathcal{R}(\boldsymbol{A}),$$

故 $\mathcal{R}(\boldsymbol{A}) = \mathcal{R}(\boldsymbol{U}_1) = \operatorname{span}\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r\}$.

(5) 直接计算式 (1.11), 得

$$\boldsymbol{A} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1^{\mathrm{H}} \\ \vdots \\ \boldsymbol{v}_r^{\mathrm{H}} \end{bmatrix}$$
$$= [\sigma_1\boldsymbol{u}_1, \sigma_2\boldsymbol{u}_2, \cdots, \sigma_r\boldsymbol{u}_r] \begin{bmatrix} \boldsymbol{v}_1^{\mathrm{H}} \\ \vdots \\ \boldsymbol{v}_r^{\mathrm{H}} \end{bmatrix}$$
$$= \sum_{i=1}^{r} \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^{\mathrm{H}}.$$

证毕. □

下面的定理表明奇异值可以刻画一个矩阵与更低秩的矩阵之间的距离.

定理 1.10 设 $\boldsymbol{A} \in \mathbb{C}_r^{m\times n}$, 它的奇异值分解由定理 1.8 给出, 其中 $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0$ 为其非零奇异值, 则有

$$\min_{\operatorname{rank}(\boldsymbol{B})=k} \|\boldsymbol{A} - \boldsymbol{B}\|_2 = \|\boldsymbol{A} - \boldsymbol{A}_k\|_2 = \sigma_{k+1}, \quad 0 \leqslant k \leqslant r-1, \tag{1.12}$$

式中:

$$\boldsymbol{A}_k = \sum_{i=1}^{k} \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^{\mathrm{H}}, \quad k > 0, \ \boldsymbol{A}_0 = \boldsymbol{O}. \tag{1.13}$$

证明 当 $k = 0$ 时, $\|\boldsymbol{A}\|_2 = \sigma_1$, 结论显然成立. 设 $k > 0$, 由 $\boldsymbol{A}_k$ 的定义及 $\boldsymbol{A}$ 的奇异值分解, 有

$$\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}_k\boldsymbol{V} = \operatorname{diag}(\sigma_1, \cdots, \sigma_k, 0, \cdots, 0),$$
$$\boldsymbol{U}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V} = \operatorname{diag}(\sigma_1, \cdots, \sigma_k, \sigma_{k+1}, \cdots, \sigma_n).$$

于是有

$$\boldsymbol{U}^{\mathrm{H}}(\boldsymbol{A} - \boldsymbol{A}_k)\boldsymbol{V} = \operatorname{diag}(0, \cdots, 0, \sigma_{k+1}, \cdots, \sigma_n).$$

注意到 σ_{k+1} 是 $\boldsymbol{A}-\boldsymbol{A}_k$ 的最大奇异值, 立即有 $\|\boldsymbol{A}-\boldsymbol{A}_k\|_2=\sigma_{k+1}$. 由于 $\operatorname{rank}(\boldsymbol{A}_k)=k$, 故

$$\min_{\operatorname{rank}(\boldsymbol{B})=k}\|\boldsymbol{A}-\boldsymbol{B}\|_2\leqslant\|\boldsymbol{A}-\boldsymbol{A}_k\|_2=\sigma_{k+1}.$$

下证任意秩为 k 的 $m\times n$ 阶矩阵 $\boldsymbol{B}$, 均有 $\|\boldsymbol{A}-\boldsymbol{B}\|_2\geqslant\sigma_{k+1}$. 由于 $\operatorname{rank}(\boldsymbol{B})=k$, 故 $\boldsymbol{B}$ 的零空间 $\mathcal{N}(\boldsymbol{B})=\{\boldsymbol{x}:\boldsymbol{B}\boldsymbol{x}=\boldsymbol{0}\}$ 的维数为 $n-k$, 故存在单位正交向量 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_{n-k}$ 使得 $\mathcal{N}(\boldsymbol{B})=\operatorname{span}\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_{n-k}\}$. 利用维数定理可知

$$\mathcal{S}:=\operatorname{span}\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_{n-k}\}\cap\operatorname{span}\{\boldsymbol{v}_1,\cdots,\boldsymbol{v}_{k+1}\}\neq\{\boldsymbol{0}\}.$$

在 $\mathcal{S}$ 中取一非零单位向量 $\boldsymbol{z}$, 即 $\boldsymbol{z}$ 满足 $\|\boldsymbol{z}\|_2=1$, $\boldsymbol{B}\boldsymbol{z}=\boldsymbol{0}$ 及

$$\boldsymbol{z}=\sum_{i=1}^{k+1}\alpha_i\boldsymbol{v}_i,\quad \boldsymbol{A}\boldsymbol{z}=\sum_{i=1}^{k+1}\alpha_i\boldsymbol{A}\boldsymbol{v}_i=\sum_{i=1}^{k+1}\sigma_i\alpha_i\boldsymbol{u}_i.$$

注意到 $\|\boldsymbol{z}\|_2^2=\sum\limits_{i=1}^{k+1}\alpha_i^2=1$, 立即有

$$\|\boldsymbol{A}-\boldsymbol{B}\|_2^2\geqslant\|(\boldsymbol{A}-\boldsymbol{B})\boldsymbol{z}\|_2^2=\|\boldsymbol{A}\boldsymbol{z}\|_2^2=\sum_{i=1}^{k+1}\sigma_i^2\alpha_i^2\geqslant\sigma_{k+1}^2.$$

证毕. □

1.2.4 矩阵的极分解和满秩分解

本节讨论矩阵的极分解和满秩分解. 极分解是指将一个复方阵分解为一个酉矩阵和 Hermite 半正定矩阵的乘积. 满秩分解是指将一个非零矩阵分解成列满秩矩阵和行满秩矩阵的乘积, 它是研究广义逆矩阵的重要工具之一. 对于极分解有下面的定理.

定理 1.11 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$. 则存在 n 阶酉矩阵 $\boldsymbol{U}$ 和 n 阶 Hermite 半正定矩阵 $\boldsymbol{H}$, 使得

$$\boldsymbol{A}=\boldsymbol{H}\boldsymbol{U}. \tag{1.14}$$

若 $\boldsymbol{A}$ 是非奇异矩阵, 则上述分解是唯一的, 此时, $\boldsymbol{H}$ 是 Hermite 正定矩阵.

证明 由奇异值分解定理可知, 存在酉矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 使得

$$\boldsymbol{A}=\boldsymbol{P}\begin{bmatrix}\boldsymbol{\Sigma}_r & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O}\end{bmatrix}\boldsymbol{Q}^{\mathrm{H}},$$

式中: $\boldsymbol{\Sigma}_r=\operatorname{diag}(\sigma_1,\sigma_2,\cdots,\sigma_r)$, $\sigma_i>0$ 为 $\boldsymbol{A}$ 的奇异值; $r=\operatorname{rank}(\boldsymbol{A})$. 将上式改写为

$$\boldsymbol{A}=\boldsymbol{P}\begin{bmatrix}\boldsymbol{\Sigma}_r & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O}\end{bmatrix}\boldsymbol{P}^{\mathrm{H}}\boldsymbol{P}\boldsymbol{Q}^{\mathrm{H}},$$

令

$$\boldsymbol{H}=\boldsymbol{P}\begin{bmatrix}\boldsymbol{\Sigma}_r & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O}\end{bmatrix}\boldsymbol{P}^{\mathrm{H}},\quad \boldsymbol{U}=\boldsymbol{P}\boldsymbol{Q}^{\mathrm{H}},$$

容易验证 $\boldsymbol{H}$ 为 Hermite 半正定矩阵, 而 $\boldsymbol{U}$ 为酉矩阵.

下面证明当 $\boldsymbol{A}$ 非奇异时分解是唯一的. 事实上, 若 $\boldsymbol{A}$ 还有极分解 $\boldsymbol{A}=\boldsymbol{H}_1\boldsymbol{U}_1$, 即

$$\boldsymbol{A}=\boldsymbol{H}\boldsymbol{U}=\boldsymbol{H}_1\boldsymbol{U}_1,$$

式中: $\boldsymbol{U},\boldsymbol{U}_1$ 为酉矩阵; $\boldsymbol{H},\boldsymbol{H}_1$ 为 Hermite 正定矩阵. 注意到

$$\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}=\boldsymbol{H}^2=\boldsymbol{H}_1^2,$$

则由定理 1.2 可知 $\boldsymbol{H}_1=\boldsymbol{H}$. 于是, $\boldsymbol{U}_1=\boldsymbol{H}_1^{-1}\boldsymbol{A}=\boldsymbol{H}^{-1}\boldsymbol{A}=\boldsymbol{U}$. 证毕. □

注 1.3 若定理 1.11 中的矩阵 $\boldsymbol{A}$ 是实方阵, 则分解式中的 $\boldsymbol{U}$ 和 $\boldsymbol{H}$ 分别为正交矩阵和对称半正定矩阵.

下面讨论满秩分解.

定义 1.11 设 $\boldsymbol{A}\in\mathbb{C}_r^{m\times n}(r>0)$, 若存在列满秩矩阵 $\boldsymbol{F}\in\mathbb{C}_r^{m\times r}$ 和行满秩矩阵 $\boldsymbol{G}\in\mathbb{C}_r^{r\times n}$, 使得 $\boldsymbol{A}=\boldsymbol{F}\boldsymbol{G}$, 则称 $\boldsymbol{F}\boldsymbol{G}$ 为 $\boldsymbol{A}$ 的一个满秩分解.

定理 1.12 设 $\boldsymbol{A}\in\mathbb{C}_r^{m\times n}(r>0)$, 则存在 $\boldsymbol{F}\in\mathbb{C}_r^{m\times r}$ 和 $\boldsymbol{G}\in\mathbb{C}_r^{r\times n}$, 使得 $\boldsymbol{A}=\boldsymbol{F}\boldsymbol{G}$.

证明 利用初等变换, 可将 $\boldsymbol{A}$ 化为阶梯形矩阵

$$\boldsymbol{B}=\begin{bmatrix}\boldsymbol{G}\\ \boldsymbol{O}\end{bmatrix},\quad (\boldsymbol{G}\in\mathbb{C}_r^{r\times n}),$$

即存在有限个初等矩阵的乘积 $\boldsymbol{P}\in\mathbb{C}^{m\times m}$, 使得 $\boldsymbol{P}\boldsymbol{A}=\boldsymbol{B}$. 由于 $\boldsymbol{P}$ 可逆, 划分

$$\boldsymbol{P}^{-1}=[\boldsymbol{F},\boldsymbol{S}],\quad (\boldsymbol{F}\in\mathbb{C}_r^{m\times r},\boldsymbol{S}\in\mathbb{C}^{m\times(m-r)}).$$

于是有

$$\boldsymbol{A}=\boldsymbol{P}^{-1}\boldsymbol{B}=[\boldsymbol{F},\boldsymbol{S}]\begin{bmatrix}\boldsymbol{G}\\ \boldsymbol{O}\end{bmatrix}=\boldsymbol{F}\boldsymbol{G}.$$

证毕. □

例 1.1 求下列矩阵的满秩分解:

$$\boldsymbol{A}=\begin{bmatrix}-1 & 0 & 1 & 2\\ 1 & 2 & -1 & 1\\ 2 & 2 & -2 & -1\end{bmatrix}.$$

解 由

$$[\boldsymbol{A},\ \boldsymbol{I}]=\left[\begin{array}{cccc:ccc}-1&0&1&2&1&0&0\\1&2&-1&1&0&1&0\\2&2&-2&-1&0&0&1\end{array}\right]\xrightarrow{\text{行}}\left[\begin{array}{cccc:ccc}-1&0&1&2&1&0&0\\0&2&0&3&1&1&0\\0&0&0&0&1&-1&1\end{array}\right],$$

可知

$$\boldsymbol{P}=\begin{bmatrix}1&0&0\\1&1&0\\1&-1&1\end{bmatrix},\quad \boldsymbol{P}^{-1}=\begin{bmatrix}1&0&0\\-1&1&0\\-2&1&1\end{bmatrix},$$

于是有

$$\boldsymbol{F}=\begin{bmatrix}1&0\\-1&1\\-2&1\end{bmatrix},\quad \boldsymbol{G}=\begin{bmatrix}-1&0&1&2\\0&2&0&3\end{bmatrix},\quad \boldsymbol{A}=\boldsymbol{F}\boldsymbol{G}.$$

定理 1.12 提供的算法求矩阵的满秩分解时需要计算矩阵 $\boldsymbol{P}$ 及 $\boldsymbol{P}^{-1}$ 的前 r 列, 这在高维情形计算量是巨大的. 下面介绍一种避免求逆矩阵的方法.

定义 1.12 设 $\boldsymbol{B}\in\mathbb{C}_r^{m\times n}\,(r>0)$ 满足: ① $\boldsymbol{B}$ 的后 $m-r$ 行上的元素都是零; ② $\boldsymbol{B}$ 中有 r 个列 (c_1 列, $\cdots$, c_r 列) 构成 $\boldsymbol{I}_m$ 的前 r 个列. 则称 $\boldsymbol{B}$ 为拟 Hermite 标准形.

定理 1.13 设 $\boldsymbol{A}\in\mathbb{C}_r^{m\times n}\,(r>0)$ 经过初等行变换化为拟 Hermite 标准形 $\boldsymbol{B}$, 那么 $\boldsymbol{A}$ 有满秩分解 $\boldsymbol{A}=\boldsymbol{F}\boldsymbol{G}$, 其中 $\boldsymbol{F}$ 是由"$\boldsymbol{A}$ 的 c_1 列, $\cdots$, c_r 列"构成的矩阵, $\boldsymbol{G}$ 是由"$\boldsymbol{B}$ 的前 r 行"构成的矩阵.

证明 经过初等行变换将 $\boldsymbol{A}$ 化为拟 Hermite 标准形 $\boldsymbol{B}$, 等价于存在可逆矩阵 $\boldsymbol{P}\in\mathbb{C}^{m\times m}$, 使得 $\boldsymbol{P}\boldsymbol{A}=\boldsymbol{B}$, 或者 $\boldsymbol{A}=\boldsymbol{P}^{-1}\boldsymbol{B}$.

根据 $\boldsymbol{B}$ 中的列标 $c_1,\cdots,c_r$ 构造 n 阶置换矩阵

$$\boldsymbol{P}_1=[\boldsymbol{e}_{c_1},\cdots,\boldsymbol{e}_{c_r},\boldsymbol{e}_{c_{r+1}},\cdots,\boldsymbol{e}_{c_n}],$$

并对矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 按列分块:

$$\boldsymbol{A}=[\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n],\quad \boldsymbol{B}=[\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n].$$

于是可得

$$\boldsymbol{A}\boldsymbol{P}_1=[\boldsymbol{a}_{c_1},\cdots,\boldsymbol{a}_{c_r},\boldsymbol{a}_{c_{r+1}},\cdots,\boldsymbol{a}_{c_n}],$$

$$\boldsymbol{B}\boldsymbol{P}_1=[\boldsymbol{b}_{c_1},\cdots,\boldsymbol{b}_{c_r},\boldsymbol{b}_{c_{r+1}},\cdots,\boldsymbol{b}_{c_n}]=\begin{bmatrix}\boldsymbol{I}_r&\boldsymbol{B}_{12}\\\boldsymbol{O}&\boldsymbol{O}\end{bmatrix}.$$

划分 $\boldsymbol{P}^{-1}=[\boldsymbol{F},\boldsymbol{S}]$, 其中 $\boldsymbol{F}\in\mathbb{C}_r^{m\times r}$, $\boldsymbol{S}\in\mathbb{C}^{m\times(m-r)}$. 根据定理 1.12 可得满秩分解 $\boldsymbol{A}=\boldsymbol{F}\boldsymbol{G}$, 且 $\boldsymbol{G}$ 是由“$\boldsymbol{B}$ 的前 r 行”构成的矩阵. 由于

$$\boldsymbol{A}\boldsymbol{P}_1=\boldsymbol{P}^{-1}(\boldsymbol{B}\boldsymbol{P}_1)=[\boldsymbol{F},\boldsymbol{S}]\begin{bmatrix}\boldsymbol{I}_r & \boldsymbol{B}_{12}\\ \boldsymbol{O} & \boldsymbol{O}\end{bmatrix}=[\boldsymbol{F},\boldsymbol{F}\boldsymbol{B}_{12}].$$

所以 $\boldsymbol{F}$ 是由“$\boldsymbol{A}\boldsymbol{P}_1$ 的前 r 列”构成的矩阵, 也就是由“$\boldsymbol{A}$ 的 c_1 列, $\cdots$, c_r 列”构成的矩阵. 证毕. □

例 1.2 求下列矩阵的满秩分解:

$$\boldsymbol{A}=\begin{bmatrix}2&1&0&2\\0&0&1&2\\2&1&1&4\end{bmatrix}.$$

解 注意到

$$\boldsymbol{A}\xrightarrow{\text{行}}\begin{bmatrix}2&1&0&2\\0&0&1&2\\0&0&0&0\end{bmatrix}$$

为拟 Hermite 标准形, 且 $c_1=2$, $c_2=3$, 故

$$\boldsymbol{F}=\begin{bmatrix}1&0\\0&1\\1&1\end{bmatrix},\quad \boldsymbol{G}=\begin{bmatrix}2&1&0&2\\0&0&1&2\end{bmatrix},\quad \boldsymbol{A}=\boldsymbol{F}\boldsymbol{G}.$$

1.3 向量和矩阵的范数

在许多实际问题中, 常需对同一线性空间中的向量 (或矩阵) 引入作为它们“大小”的一种度量, 进而比较两个向量 (或矩阵) 的“接近”程度. 引入这种体现其“大小”的量就是范数, 它们在理论与实际应用中都占有重要的地位.

1.3.1 向量内积与向量范数

定义 1.13 设 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in\mathbb{C}^n$, $\boldsymbol{y}=(y_1,y_2,\cdots,y_n)^{\mathrm{T}}\in\mathbb{C}^n$, 称复数

$$(\boldsymbol{x},\boldsymbol{y}):=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}=\sum_{i=1}^{n}x_i\bar{y}_i$$

为向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的 (欧几里得) 内积或数量积. 而称

$$\|\boldsymbol{x}\|_2=(\boldsymbol{x},\boldsymbol{x})^{\frac{1}{2}}=\left(\sum_{i=1}^{n}|x_i|^2\right)^{\frac{1}{2}}\tag{1.15}$$

为向量 $\boldsymbol{x}$ 的 (欧几里得) 范数或 2-范数.

内积的一个重要性质是

$$(\boldsymbol{Ax}, \boldsymbol{y}) = (\boldsymbol{x}, \boldsymbol{A}^{\mathrm{H}}\boldsymbol{y}),\ \boldsymbol{A} \in \mathbb{C}^{m\times n},\ \boldsymbol{x} \in \mathbb{C}^n,\ \boldsymbol{y} \in \mathbb{C}^m. \tag{1.16}$$

$\mathbb{C}^n$ 中的欧几里得内积与欧几里得范数还具有下列性质.

性质 1.6 设 $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$, 则

(1) $(\boldsymbol{x}, \boldsymbol{x}) = 0$ 当且仅当 $\boldsymbol{x} = \boldsymbol{0}$.

(2) $(\lambda\boldsymbol{x}, \boldsymbol{y}) = \lambda(\boldsymbol{x}, \boldsymbol{y})$, $(\boldsymbol{x}, \lambda\boldsymbol{y}) = \bar{\lambda}(\boldsymbol{x}, \boldsymbol{y})$, $\forall \lambda \in \mathbb{C}$.

(3) $(\boldsymbol{x}, \boldsymbol{y}) = \overline{(\boldsymbol{y}, \boldsymbol{x})}$.

(4) $(\boldsymbol{x}_1 + \boldsymbol{x}_2, \boldsymbol{y}) = (\boldsymbol{x}_1, \boldsymbol{y}) + (\boldsymbol{x}_2, \boldsymbol{y})$.

(5) Cauchy–Schwarz 不等式: $|(\boldsymbol{x}, \boldsymbol{y})| \leqslant \|\boldsymbol{x}\|_2 \cdot \|\boldsymbol{y}\|_2$.

(6) 三角不等式: $\|\boldsymbol{x} + \boldsymbol{y}\|_2 \leqslant \|\boldsymbol{x}\|_2 + \|\boldsymbol{y}\|_2$.

由 Schwarz 不等式容易证明

$$\|\boldsymbol{x}\|_2 = \max_{\|\boldsymbol{y}\|_2=1} |\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}|,\quad \boldsymbol{x} \in \mathbb{C}^n. \tag{1.17}$$

下面给出向量范数的一般定义.

定义 1.14 给定 $\mathbb{C}^n$ 中的某个实值函数 $\mathfrak{N}(\boldsymbol{x}) = \|\boldsymbol{x}\|$. 若对任意的 $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$ 有

(1) $\|\boldsymbol{x}\| \geqslant 0$, 且 $\|\boldsymbol{x}\| = 0$ 当且仅当 $\boldsymbol{x} = \boldsymbol{0}$.

(2) $\|\lambda\boldsymbol{x}\| = |\lambda| \cdot \|\boldsymbol{x}\|$, $\forall \lambda \in \mathbb{C}$.

(3) $\|\boldsymbol{x} + \boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$.

则称 $\|\boldsymbol{x}\|$ 为向量 $\boldsymbol{x}$ 的范数. 定义了范数的线性空间称为赋范线性空间.

常用的向量范数有

(1) $\|\boldsymbol{x}\|_\infty = \max\limits_{1\leqslant i\leqslant n} |x_i|$ (∞–范数).

(2) $\|\boldsymbol{x}\|_1 = \sum\limits_{i=1}^{n} |x_i|$ (1–范数).

(3) $\|\boldsymbol{x}\|_2 = \Big(\sum\limits_{i=1}^{n} |x_i|^2\Big)^{1/2}$ (2–范数).

(4) $\|\boldsymbol{x}\|_p = \Big(\sum\limits_{i=1}^{n} |x_i|^p\Big)^{1/p}$ (p–范数).

不难证明, 当 $p \to \infty$ 时, 对于任意的 $\boldsymbol{x} \in \mathbb{C}^n$, 都有 $\|\boldsymbol{x}\|_p \to \|\boldsymbol{x}\|_\infty$. 此外, 由于 $\big|\|\boldsymbol{x}\| - \|\boldsymbol{y}\|\big| \leqslant \|\boldsymbol{x} - \boldsymbol{y}\|$, 故向量范数是 $\mathbb{C}^n$ 中的连续函数.

性质 1.7 (向量范数的连续性) $\mathbb{C}^n$ 中的向量范数 $\|\boldsymbol{x}\|$ 是 $\mathbb{C}^n$ 中的连续函数.

性质 1.8 (向量范数的等价性) 设 $\|\boldsymbol{x}\|$ 和 $\|\boldsymbol{x}\|'$ 是 $\mathbb{C}^n$ 中的任意两种范数, 则成立

$$c_1\|\boldsymbol{x}\| \leqslant \|\boldsymbol{x}\|' \leqslant c_2\|\boldsymbol{x}\|,\quad \boldsymbol{x} \in \mathbb{C}^n, \tag{1.18}$$

式中: $c_1, c_2 > 0$ 为与向量 $\boldsymbol{x}$ 无关的常数.

定义 1.15 设 $\{\boldsymbol{x}^{(k)}=(x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)})^{\mathrm{T}}\}_{k=0}^{\infty}$ 是 $\mathbb{C}^n$ 中的向量序列, $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in\mathbb{C}^n$. 若

$$\lim_{k\to\infty}x_i^{(k)}=x_i,\quad i=1,2,\cdots,n,$$

则称序列 $\{\boldsymbol{x}^{(k)}\}$ 收敛于 $\boldsymbol{x}$, 记为 $\lim\limits_{k\to\infty}\boldsymbol{x}^{(k)}=\boldsymbol{x}$.

利用范数的等价性可以得到向量序列收敛的充分必要条件.

定理 1.14 $\lim\limits_{k\to\infty}\boldsymbol{x}^{(k)}=\boldsymbol{x}$ 当且仅当 $\lim\limits_{k\to\infty}\|\boldsymbol{x}^{(k)}-\boldsymbol{x}\|=0$, 其中 $\|\cdot\|$ 是 $\mathbb{C}^n$ 中任意的向量范数.

1.3.2 矩阵范数与内积

将 $m\times n$ 阶矩阵 $\boldsymbol{A}$ 看作线性空间 $\mathbb{C}^{m\times n}$ 中的元素, 则完全可以按照定义 1.14 的方式引入矩阵的范数. 其中最常用的是与向量 2–范数相对应的范数

$$\|\boldsymbol{A}\|_{\mathrm{F}}=\Big(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\Big)^{1/2}\tag{1.19}$$

称为矩阵 $\boldsymbol{A}$ 的 Frobenius 范数, 简称 F–范数.

利用奇异值分解定理, 容易证明

$$\|\boldsymbol{A}\|_{\mathrm{F}}=\Big(\sum_{i=1}^{n}\sigma_i^2\Big)^{1/2},$$

式中: $\sigma_1,\sigma_2,\cdots,\sigma_n$ 为矩阵 $\boldsymbol{A}$ 奇异值.

类似于向量范数的定义, 可以给出矩阵范数的一般定义.

定义 1.16 给定 $\mathbb{C}^{m\times n}$ 中的某个实值函数 $\mathfrak{N}(\boldsymbol{A})=\|\boldsymbol{A}\|$. 若对任意的 $\boldsymbol{A},\boldsymbol{B}\in\mathbb{C}^{m\times n}$ 有

(1) $\|\boldsymbol{A}\|\geqslant 0$, 且 $\|\boldsymbol{A}\|=0$ 当且仅当 $\boldsymbol{A}=\boldsymbol{O}$.

(2) $\|\lambda\boldsymbol{A}\|=|\lambda|\cdot\|\boldsymbol{A}\|$, $\forall\lambda\in\mathbb{C}$.

(3) $\|\boldsymbol{A}+\boldsymbol{B}\|\leqslant\|\boldsymbol{A}\|+\|\boldsymbol{B}\|$.

则称 $\|\boldsymbol{A}\|$ 为矩阵 $\boldsymbol{A}$ 的范数. 若矩阵范数满足

$$\|\boldsymbol{A}\boldsymbol{x}\|\leqslant\|\boldsymbol{A}\|\cdot\|\boldsymbol{x}\|,\quad\forall\,\boldsymbol{x}\in\mathbb{C}^n,\ \boldsymbol{A}\in\mathbb{C}^{m\times n},$$

则称矩阵范数 $\|\cdot\|$ 和向量范数 $\|\cdot\|$ 是相容的, 其中 $\|\boldsymbol{A}\boldsymbol{x}\|$ 和 $\|\boldsymbol{x}\|$ 分别是 $\mathbb{C}^m$ 和 $\mathbb{C}^n$ 中的向量范数.

跟向量范数的等价性一样, 可以证明 $\mathbb{C}^{m\times n}$ 中的任意两个矩阵范数也等价, 即存在与矩阵 $\boldsymbol{A}$ 无关的常数 $c_1,c_2>0$ 使得

$$c_1\|\boldsymbol{A}\|\leqslant\|\boldsymbol{A}\|'\leqslant c_2\|\boldsymbol{A}\|.$$

定义 1.17 设 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$, 给定一种向量范数 $\|\cdot\|$, 定义矩阵范数

$$\|\boldsymbol{A}\|=\max_{\|\boldsymbol{x}\|\neq 0}\frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}=\max_{\|\boldsymbol{x}\|=1}\|\boldsymbol{A}\boldsymbol{x}\|. \tag{1.20}$$

称 $\|\boldsymbol{A}\|$ 为 $\mathbb{C}^{m\times n}$ 中矩阵 $\boldsymbol{A}$ 的算子范数, 或称由该向量范数诱导出来的矩阵范数.

容易证明, 矩阵的算子范数与相应的向量范数是相容的, 且

$$\|\boldsymbol{A}\boldsymbol{B}\|\leqslant\|\boldsymbol{A}\|\cdot\|\boldsymbol{B}\|,\quad \boldsymbol{A}\in\mathbb{C}^{m\times n},\ \boldsymbol{B}\in\mathbb{C}^{n\times p},$$

其中矩阵范数都是由某个向量范数诱导出来的算子范数.

给定 $\mathbb{C}^n$ 中的向量 p–范数, 可以诱导出相应的矩阵 p–算子范数, 记为 $\|\cdot\|_p$. 下面是三种常用的矩阵算子范数.

定理 1.15 设 $\boldsymbol{A}=(a_{ij})\in\mathbb{C}^{m\times n}$, 则

(1) $\|\boldsymbol{A}\|_\infty=\max\limits_{1\leqslant i\leqslant m}\sum\limits_{j=1}^{n}|a_{ij}|$, 称为行和范数或 ∞–范数.

(2) $\|\boldsymbol{A}\|_1=\max\limits_{1\leqslant j\leqslant n}\sum\limits_{i=1}^{m}|a_{ij}|$, 称为列和范数或 1–范数.

(3) $\|\boldsymbol{A}\|_2=\sigma_1=\sqrt{\lambda_{\max}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})}$, 称为谱范数或 2–范数, 其中 σ_1 是矩阵 $\boldsymbol{A}$ 的最大奇异值, $\lambda_{\max}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})$ 表示矩阵 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 的最大特征值.

证明 先证明 (1). 记 $\sigma=\max\limits_{1\leqslant i\leqslant m}\sum\limits_{j=1}^{n}|a_{ij}|$. 则对任意满足 $\|\boldsymbol{x}\|_\infty=1$ 的向量 $\boldsymbol{x}\in\mathbb{C}^n$, 有

$$\begin{aligned}\|\boldsymbol{A}\boldsymbol{x}\|_\infty&=\max_{1\leqslant i\leqslant m}\Big|\sum_{j=1}^{n}a_{ij}x_j\Big|\leqslant\max_{1\leqslant i\leqslant m}\sum_{j=1}^{n}|a_{ij}|\cdot|x_j|\\&\leqslant\Big(\max_{1\leqslant i\leqslant m}\sum_{j=1}^{n}|a_{ij}|\Big)\|\boldsymbol{x}\|_\infty=\sigma.\end{aligned} \tag{1.21}$$

另外, 不妨设 $\sigma=\sum\limits_{j=1}^{n}|a_{i_0j}|$. 取

$$\widetilde{\boldsymbol{x}}=\big(\mathrm{sign}(a_{i_01}),\mathrm{sign}(a_{i_02}),\cdots,\mathrm{sign}(a_{i_0n})\big)^{\mathrm{T}},$$

则 $\boldsymbol{A}\neq\boldsymbol{O}$ 蕴含着 $\|\widetilde{\boldsymbol{x}}\|_\infty=1$. 由 $|a|=\mathrm{sign}(a)a$, 有

$$\|\boldsymbol{A}\widetilde{\boldsymbol{x}}\|_\infty=\max_{1\leqslant i\leqslant m}\Big|\sum_{j=1}^{n}a_{ij}\widetilde{x}_j\Big|\geqslant\Big|\sum_{j=1}^{n}a_{i_0j}\widetilde{x}_j\Big|=\sum_{j=1}^{n}|a_{i_0j}|=\sigma. \tag{1.22}$$

结论 (2) 的证明与结论 (1) 类似. 最后给出结论 (3) 的证明. 注意到 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 是 Hermite 半正定矩阵, 故其特征值为 $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n\geqslant 0$, 相应的单位正交特征向量为 $\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\cdots,\boldsymbol{\xi}_n\in\mathbb{C}^n$. 因此, 对任意满足 $\|\boldsymbol{x}\|_2=1$ 的向量 $\boldsymbol{x}\in\mathbb{C}^n$, 有

$$\boldsymbol{x}=\sum_{i=1}^{n}c_i\boldsymbol{\xi}_i,\quad \sum_{i=1}^{n}c_i^2=1.$$

于是有

$$\begin{aligned}\boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x} &= \Big(\sum_{i=1}^{n} c_i\boldsymbol{\xi}_i\Big)^{\mathrm{H}} \sum_{j=1}^{n} c_j\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{\xi}_j \\ &= \Big(\sum_{i=1}^{n} c_i\boldsymbol{\xi}_i\Big)^{\mathrm{H}} \sum_{j=1}^{n} c_j\lambda_j\boldsymbol{\xi}_j \\ &= \sum_{i=1}^{n} c_i^2\lambda_i\boldsymbol{\xi}_i^{\mathrm{H}}\boldsymbol{\xi}_i = \sum_{i=1}^{n} c_i^2\lambda_i \leqslant \lambda_1.\end{aligned}$$

注意到

$$\|\boldsymbol{A}\|_2 = \max_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{A}\boldsymbol{x}\|_2 = \max_{\|\boldsymbol{x}\|_2=1} \left(\boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}\right)^{1/2},$$

故 $\|\boldsymbol{A}\|_2 \leqslant \sqrt{\lambda_1}$. 取 $\boldsymbol{x} = \boldsymbol{\xi}_1$, 则有

$$\boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{\xi}_1^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{\xi}_1 = \lambda_1\boldsymbol{\xi}_1^{\mathrm{H}}\boldsymbol{\xi}_1 = \lambda_1,$$

故

$$\begin{aligned}\|\boldsymbol{A}\|_2 &= \max_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{A}\boldsymbol{x}\|_2 = \max_{\|\boldsymbol{x}\|_2=1} \left(\boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}\right)^{1/2} \\ &\geqslant \left(\boldsymbol{\xi}_1^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{\xi}_1\right)^{1/2} = \left(\lambda_1\boldsymbol{\xi}_1^{\mathrm{H}}\boldsymbol{\xi}_1\right)^{1/2} = \sqrt{\lambda_1}.\end{aligned}$$

因此 $\|\boldsymbol{A}\|_2 = \sqrt{\lambda_1} = \sqrt{\lambda_{\max}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})}$. 证毕. □

与矩阵的行和范数及列和范数相比, 矩阵的谱范数显得不易计算, 但它有良好的分析性质.

定理 1.16 设 $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, 则

(1) $\|\boldsymbol{A}\|_2 = \max\limits_{\|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=1} |\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}|$.

(2) $\|\boldsymbol{A}^{\mathrm{H}}\|_2 = \|\boldsymbol{A}^{\mathrm{T}}\|_2 = \|\boldsymbol{A}\|_2$.

(3) $\|\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\|_2 = \|\boldsymbol{A}\|_2^2$.

(4) $\|\boldsymbol{A}\|_2^2 \leqslant \|\boldsymbol{A}\|_1 \cdot \|\boldsymbol{A}\|_\infty$.

(5) $\|\boldsymbol{A}\|_2 \leqslant \|\boldsymbol{A}\|_{\mathrm{F}}$.

(6) 设 $\boldsymbol{U} \in \mathbb{C}^{m\times m}$ 和 $\boldsymbol{V} \in \mathbb{C}^{n\times n}$ 满足 $\boldsymbol{U}^{\mathrm{H}}\boldsymbol{U} = \boldsymbol{I}_m$ 和 $\boldsymbol{V}^{\mathrm{H}}\boldsymbol{V} = \boldsymbol{I}_n$, 则 $\|\boldsymbol{U}\boldsymbol{A}\boldsymbol{V}\|_2 = \|\boldsymbol{A}\|_2$.

证明 由算子范数的定义, 存在 $\boldsymbol{x}_0 \in \mathbb{C}^n$ 满足 $\|\boldsymbol{x}_0\|_2 = 1$ 和 $\|\boldsymbol{A}\|_2 = \|\boldsymbol{A}\boldsymbol{x}_0\|_2$. 由式 (1.17) 知,

$$\|\boldsymbol{A}\boldsymbol{x}_0\|_2 = \max_{\|\boldsymbol{y}\|_2=1} |\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}_0|,$$

这意味着成立

$$\|\boldsymbol{A}\|_2 = \|\boldsymbol{A}\boldsymbol{x}_0\|_2 \leqslant \max_{\|\boldsymbol{x}\|_2=1,\,\|\boldsymbol{y}\|_2=1} |\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}|.$$

另外, 由 Schwarz 不等式及范数的相容性, 有

$$|\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}| \leqslant \|\boldsymbol{y}\|_2 \cdot \|\boldsymbol{A}\boldsymbol{x}\|_2 \leqslant \|\boldsymbol{y}\|_2 \cdot \|\boldsymbol{A}\|_2 \cdot \|\boldsymbol{x}\|_2,$$

故

$$\max_{\|\boldsymbol{x}\|_2=1,\,\|\boldsymbol{y}\|_2=1} |\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}| \leqslant \|\boldsymbol{A}\|_2,$$

这就证明了结论 (1) .

由结论 (1) 有

$$\|\boldsymbol{A}^{\mathrm{T}}\|_2 = \max_{\|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=1} |\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}| = \max_{\|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=1} |\bar{\boldsymbol{x}}^{\mathrm{H}}\boldsymbol{A}\bar{\boldsymbol{y}}|$$

$$= \max_{\|\bar{\boldsymbol{x}}\|_2=\|\bar{\boldsymbol{y}}\|_2=1} |\bar{\boldsymbol{x}}^{\mathrm{H}}\boldsymbol{A}\bar{\boldsymbol{y}}| = \|\boldsymbol{A}\|_2,$$

式中: $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$ 为 $\boldsymbol{x}, \boldsymbol{y}$ 的共轭向量, 这里用到了一个标量与其转置相等.

同理, 有

$$\|\boldsymbol{A}^{\mathrm{H}}\|_2 = \max_{\|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=1} |\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{x}| = \max_{\|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=1} |\boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{y}| = \|\boldsymbol{A}\|_2,$$

这里用到了一个复数的模与其共轭转置的模相等. 至此证明了结论 (2).

注意到 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 是 Hermite 矩阵, 故由谱范数的定义, 有

$$\|\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\|_2 = \sqrt{\lambda_{\max}\big((\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})^{\mathrm{H}}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})\big)} = \sqrt{\lambda_{\max}\big((\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})^2\big)}$$

$$= \lambda_{\max}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}) = \|\boldsymbol{A}\|_2^2,$$

结论 (3) 得证.

由于 $\|\boldsymbol{A}\|_2^2$ 是 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 的最大特征值, 设其相应的特征向量为 $\boldsymbol{x} \neq \boldsymbol{0}$, 即 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x} = \|\boldsymbol{A}\|_2^2\boldsymbol{x}$, 两边取 1–范数, 得

$$\|\boldsymbol{A}\|_2^2\|\boldsymbol{x}\|_1 = \big\|\|\boldsymbol{A}\|_2^2\boldsymbol{x}\big\|_1 = \|\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}\|_1$$

$$\leqslant \|\boldsymbol{A}^{\mathrm{H}}\|_1 \cdot \|\boldsymbol{A}\|_1 \cdot \|\boldsymbol{x}\|_1 = \|\boldsymbol{A}\|_\infty \cdot \|\boldsymbol{A}\|_1 \cdot \|\boldsymbol{x}\|_1,$$

两边除以 $\|\boldsymbol{x}\|_1$ 即得结论 (4).

由于 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 是 Hermite 半正定矩阵, 故其特征值 $\lambda_s(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})$, $s = 1, 2, \cdots, n$ 均非负, 且满足

$$\sum_{s=1}^{n} \lambda_s(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}) = \mathrm{tr}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}) = \sum_{s=1}^{n} \big(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\big)_{ss},$$

从而有

$$\|\boldsymbol{A}\|_2^2 = \lambda_{\max}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}) \leqslant \sum_{s=1}^{n} \lambda_s(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})$$

$$= \sum_{s=1}^{n} (\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})_{ss} = \sum_{s=1}^{n} \Big(\sum_{i=1}^{m} |a_{is}|^2 \Big) = \|\boldsymbol{A}\|_{\mathrm{F}}^2,$$

两边开平方即得结论 (5).

最后证明结论 (6). 注意到对任意的 n 阶酉矩阵 $\boldsymbol{U}$ 和 n 维列向量 $\boldsymbol{x}$, 有

$$\|\boldsymbol{U}\boldsymbol{x}\|_2 = \sqrt{(\boldsymbol{U}\boldsymbol{x})^{\mathrm{H}}(\boldsymbol{U}\boldsymbol{x})} = \sqrt{\boldsymbol{x}^{\mathrm{H}}(\boldsymbol{U}^{\mathrm{H}}\boldsymbol{U})\boldsymbol{x}} = \sqrt{\boldsymbol{x}^{\mathrm{H}}\boldsymbol{x}} = \|\boldsymbol{x}\|_2,$$

即酉矩阵保持列向量的 2–范数不变. 由此, 有

$$\begin{aligned} \|\boldsymbol{U}\boldsymbol{A}\boldsymbol{V}\|_2 &= \max_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{U}\boldsymbol{A}\boldsymbol{V}\boldsymbol{x}\|_2 = \max_{\|\boldsymbol{y}\|_2=1} \|\boldsymbol{U}\boldsymbol{A}\boldsymbol{y}\|_2 \\ &= \max_{\|\boldsymbol{y}\|_2=1} \|\boldsymbol{A}\boldsymbol{y}\|_2 = \|\boldsymbol{A}\|_2. \end{aligned} \tag{1.23}$$

证毕. □

定义 1.18 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, 其特征值为 $\lambda_1, \lambda_2, \cdots, \lambda_n$, 则称

$$\rho(\boldsymbol{A}) = \max_{1\leqslant i\leqslant n} |\lambda_i|$$

为矩阵 $\boldsymbol{A}$ 的谱半径.

由上述定义, $\|\boldsymbol{A}\|_2$ 可定义为

$$\|\boldsymbol{A}\|_2 = \sqrt{\rho(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})}.$$

特别地, 当 $\boldsymbol{A}$ 为 Hermite 矩阵时, 有

$$\|\boldsymbol{A}\|_2 = \rho(\boldsymbol{A}).$$

对于一般情况, 有如下定理.

定理 1.17 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, 则

(1) $\boldsymbol{A}$ 的谱半径不超过 $\boldsymbol{A}$ 的任何范数, 即

$$\rho(\boldsymbol{A}) \leqslant \|\boldsymbol{A}\|.$$

(2) 对任意正数 ε, 存在 $\mathbb{C}^{n\times n}$ 上的某种算子范数 $\|\cdot\|_\varepsilon$, 使得

$$\|\boldsymbol{A}\|_\varepsilon \leqslant \rho(\boldsymbol{A}) + \varepsilon.$$

证明 (1) 设 λ 是 $\boldsymbol{A}$ 的一个特征值, 且 $\boldsymbol{A}$ 的谱半径 $\rho(\boldsymbol{A}) = |\lambda|$, $\boldsymbol{x}_0$ 是对应于 λ 的特征向量, 即 $\boldsymbol{A}\boldsymbol{x}_0 = \lambda\boldsymbol{x}_0$, 所以对任何一种向量范数, 有 $\|\boldsymbol{A}\boldsymbol{x}_0\| = |\lambda| \cdot \|\boldsymbol{x}_0\|$, 故有

$$\rho(\boldsymbol{A}) = |\lambda| = \frac{\|\boldsymbol{A}\boldsymbol{x}_0\|}{\|\boldsymbol{x}_0\|} \leqslant \max_{\|\boldsymbol{x}\|\neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} = \|\boldsymbol{A}\|.$$

(2) 由 Jordan 分解定理知, 存在非奇异矩阵 $\boldsymbol{P}$ 使得

$$\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}=\begin{bmatrix}\lambda_1 & \delta_1 & & & \\ & \lambda_2 & \delta_2 & & \\ & & \ddots & \ddots & \\ & & & \lambda_{n-1} & \delta_{n-1} \\ & & & & \lambda_n\end{bmatrix},$$

式中: δ_i 等于 1 或 0. 对于任意的正数 ε, 令

$$\boldsymbol{D}_\varepsilon=\begin{bmatrix}1 & & & & \\ & \varepsilon & & & \\ & & \ddots & & \\ & & & \varepsilon^{n-1}\end{bmatrix},$$

则

$$\boldsymbol{D}_\varepsilon^{-1}\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}\boldsymbol{D}_\varepsilon=\begin{bmatrix}\lambda_1 & \varepsilon\delta_1 & & & \\ & \lambda_2 & \varepsilon\delta_2 & & \\ & & \ddots & \ddots & \\ & & & \lambda_{n-1} & \varepsilon\delta_{n-1} \\ & & & & \lambda_n\end{bmatrix}.$$

现定义一种新的矩阵范数

$$\|\boldsymbol{A}\|_\varepsilon=\|\boldsymbol{D}_\varepsilon^{-1}\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}\boldsymbol{D}_\varepsilon\|_\infty,\quad \boldsymbol{A}\in\mathbb{C}^{n\times n},$$

则容易验证这样定义的矩阵范数 $\|\cdot\|_\varepsilon$ 是由如下定义的向量范数诱导出来的算子范数:

$$\|\boldsymbol{x}\|_{\boldsymbol{P}\boldsymbol{D}_\varepsilon}=\|(\boldsymbol{P}\boldsymbol{D}_\varepsilon)^{-1}\boldsymbol{x}\|_\infty,\quad \boldsymbol{x}\in\mathbb{C}^n.$$

从而有

$$\|\boldsymbol{A}\|_\varepsilon=\|\boldsymbol{D}_\varepsilon^{-1}\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}\boldsymbol{D}_\varepsilon\|_\infty=\max_{1\leqslant i\leqslant n}\left(|\lambda_i|+\varepsilon\delta_i\right)\leqslant\rho(\boldsymbol{A})+\varepsilon,$$

这里假设 $\delta_n=0$. 证毕. □

类似于向量序列的极限定义, 有以下定义.

定义 1.19 设矩阵序列 $\left\{\boldsymbol{A}^{(k)}=(a_{ij}^{(k)})\in\mathbb{C}^{m\times n}\right\}_{k=0}^{\infty}$, $\boldsymbol{A}=(a_{ij})\in\mathbb{C}^{m\times n}$. 若

$$\lim_{k\to\infty}a_{ij}^{(k)}=a_{ij},\quad i=1,2,\cdots,m;\ j=1,2,\cdots,n,$$

则称序列 $\{\boldsymbol{A}^{(k)}\}$ 收敛于 $\boldsymbol{A}$, 记为 $\lim\limits_{k\to\infty}\boldsymbol{A}^{(k)}=\boldsymbol{A}$.

利用矩阵范数的等价性, 可得以下定理.

定理 1.18 $\lim\limits_{k\to\infty}\boldsymbol{A}^{(k)}=\boldsymbol{A}$ 当且仅当 $\lim\limits_{k\to\infty}\|\boldsymbol{A}^{(k)}-\boldsymbol{A}\|=0$, 其中 $\|\cdot\|$ 是 $\mathbb{C}^{m\times n}$ 中任意的矩阵范数.

关于矩阵范数, 还有下述定理.

定理 1.19 若 $\|\boldsymbol{A}\| < 1$, 则矩阵 $\boldsymbol{I} - \boldsymbol{A}$ 非奇异, 且满足

$$\|(\boldsymbol{I} - \boldsymbol{A})^{-1}\| \leqslant \frac{1}{1 - \|\boldsymbol{A}\|}.$$

证明 用反证法. 设 $\det(\boldsymbol{I} - \boldsymbol{A}) = 0$, 则方程 $(\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} = \boldsymbol{0}$ 有非零解, 即存在 $\boldsymbol{x}_0 \in \mathbb{C}^n$, $\boldsymbol{x}_0 \neq \boldsymbol{0}$, 使得 $(\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x}_0 = \boldsymbol{0}$. 故

$$\|\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\| \neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \geqslant \frac{\|\boldsymbol{A}\boldsymbol{x}_0\|}{\|\boldsymbol{x}_0\|} = 1.$$

这与 $\|\boldsymbol{A}\| < 1$ 矛盾.

进一步, 由于 $(\boldsymbol{I} - \boldsymbol{A})(\boldsymbol{I} - \boldsymbol{A})^{-1} = \boldsymbol{I}$, 则 $(\boldsymbol{I} - \boldsymbol{A})^{-1} = \boldsymbol{I} + \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{A})^{-1}$, 从而

$$\|(\boldsymbol{I} - \boldsymbol{A})^{-1}\| \leqslant \|\boldsymbol{I}\| + \|\boldsymbol{A}\| \cdot \|(\boldsymbol{I} - \boldsymbol{A})^{-1}\|.$$

将上式整理即得要证的结论. 证毕. □

定理 1.20 设 $\boldsymbol{A} \in \mathbb{C}^{n \times n}$, 则

(1) $\lim\limits_{k \to \infty} \boldsymbol{A}^k = \boldsymbol{O} \Longleftrightarrow \rho(\boldsymbol{A}) < 1$.

(2) $\lim\limits_{k \to \infty} \|\boldsymbol{A}^k\|^{\frac{1}{k}} = \rho(\boldsymbol{A})$.

证明 (1) 充分性. 设 $\rho(\boldsymbol{A}) < 1$, 则存在 $\varepsilon > 0$ 满足 $\rho(\boldsymbol{A}) + 2\varepsilon < 1$. 于是由定理 1.17 知存在算子范数 $\|\cdot\|$ 使得 $\|\boldsymbol{A}\| \leqslant \rho(\boldsymbol{A}) + \varepsilon$. 因此

$$\|\boldsymbol{A}^k\| \leqslant \|\boldsymbol{A}\|^k \leqslant \big[\rho(\boldsymbol{A}) + \varepsilon\big]^k < (1 - \varepsilon)^k \to 0.$$

必要性. 设 $\lim\limits_{k \to \infty} \boldsymbol{A}^k = \boldsymbol{O}$, 并假定 $\lambda \in \lambda(\boldsymbol{A})$ 满足 $\rho(\boldsymbol{A}) = |\lambda|$. 由于 $\lambda \in \lambda(\boldsymbol{A})$ 蕴含 $\lambda^k \in \lambda(\boldsymbol{A}^k)$, 故由定理 1.17 知对一切的 $k = 1, 2, \cdots$ 成立

$$\big[\rho(\boldsymbol{A})\big]^k = |\lambda|^k = |\lambda^k| \leqslant \rho(\boldsymbol{A}^k) \leqslant \|\boldsymbol{A}^k\|,$$

故必有 $\rho(\boldsymbol{A}) < 1$.

(2) 由结论 (1) 的证明可知, 对任意的 $\varepsilon > 0$, 存在算子范数 $\|\cdot\|$, 满足

$$\big[\rho(\boldsymbol{A})\big]^k \leqslant \|\boldsymbol{A}^k\| \leqslant \big[\rho(\boldsymbol{A}) + \varepsilon\big]^k,$$

即

$$\rho(\boldsymbol{A}) \leqslant \|\boldsymbol{A}^k\|^{\frac{1}{k}} \leqslant \rho(\boldsymbol{A}) + \varepsilon,$$

由 $\varepsilon > 0$ 的任意性, 上式令 $k \to \infty$ 得

$$\lim_{k \to \infty} \|\boldsymbol{A}^k\|^{\frac{1}{k}} = \rho(\boldsymbol{A}).$$

再根据有限维空间范数的等价性, 对 $\mathbb{C}^{n \times n}$ 中的任意范数, 上述极限成立. 证毕. □

利用定理 1.20 不难证明下面的重要结论.

定理 1.21 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, 则

(1) 矩阵级数 $\sum\limits_{k=0}^{\infty}\boldsymbol{A}^k$ 收敛的充分必要条件是 $\rho(\boldsymbol{A})<1$.

(2) 若矩阵级数 $\sum\limits_{k=0}^{\infty}\boldsymbol{A}^k$ 收敛, 则

$$\sum_{k=0}^{\infty}\boldsymbol{A}^k=(\boldsymbol{I}-\boldsymbol{A})^{-1},$$

且对于 $\mathbb{C}^{n\times n}$ 中的任意矩阵范数都有

$$\left\|(\boldsymbol{I}-\boldsymbol{A})^{-1}-\sum_{k=0}^{m}\boldsymbol{A}^k\right\|\leqslant\frac{\|\boldsymbol{A}\|^{m+1}}{1-\|\boldsymbol{A}\|}.$$

1.4 矩阵的广义逆

矩阵的广义逆是研究最小二乘问题及矩阵方程的一个重要工具. 首先给出矩阵广义逆的定义.

定义 1.20 设 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$, 若有 $\boldsymbol{X}\in\mathbb{C}^{n\times m}$ 满足 Penrose (彭罗斯) 方程, 即

(1) $\boldsymbol{AXA}=\boldsymbol{A}$.

(2) $\boldsymbol{XAX}=\boldsymbol{X}$.

(3) $(\boldsymbol{AX})^{\mathrm{H}}=\boldsymbol{AX}$.

(4) $(\boldsymbol{XA})^{\mathrm{H}}=\boldsymbol{XA}$.

则称 $\boldsymbol{X}$ 为 $\boldsymbol{A}$ 的 Moore–Penrose 逆, 记为 $\boldsymbol{A}^{\dagger}$.

注 1.4 若 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$, 则 $\boldsymbol{X}=\boldsymbol{A}^{\dagger}\in\mathbb{R}^{n\times m}$ 定义为满足下列四个方程的 $\boldsymbol{X}$:

(1) $\boldsymbol{AXA}=\boldsymbol{A}$.

(2) $\boldsymbol{XAX}=\boldsymbol{X}$.

(3) $(\boldsymbol{AX})^{\mathrm{T}}=\boldsymbol{AX}$.

(4) $(\boldsymbol{XA})^{\mathrm{T}}=\boldsymbol{XA}$.

由定义 1.20 容易得到:

(1) $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 可逆时, $\boldsymbol{X}=\boldsymbol{A}^{-1}$ 满足 Penrose 方程, 故 $\boldsymbol{A}^{\dagger}=\boldsymbol{A}^{-1}$.

(2) $\boldsymbol{A}=\boldsymbol{O}_{m\times n}$ 时, $\boldsymbol{X}=\boldsymbol{O}_{n\times m}$ 满足 Penrose 方程, 故 $\boldsymbol{O}_{m\times n}^{\dagger}=\boldsymbol{O}_{n\times m}$.

(3) $\boldsymbol{A}=\boldsymbol{x}\in\mathbb{C}^{n\times 1}$ 时, $\boldsymbol{X}=\dfrac{1}{\|\boldsymbol{x}\|_2^2}\boldsymbol{x}^{\mathrm{H}}$ 满足 Penrose 方程, 故 $\boldsymbol{x}^{\dagger}=\dfrac{1}{\|\boldsymbol{x}\|_2^2}\boldsymbol{x}^{\mathrm{H}}$. 例如:

$$\begin{bmatrix}1\\1+2\mathrm{i}\end{bmatrix}^{\dagger}=\frac{1}{6}(1,1-2\mathrm{i}).$$

定理 1.22 设 $\boldsymbol{F}\in\mathbb{C}_r^{m\times r}$, $\boldsymbol{G}\in\mathbb{C}_r^{r\times n}$ $(r\geqslant 1)$, 则有

(1) $\boldsymbol{F}^{\dagger}=(\boldsymbol{F}^{\mathrm{H}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathrm{H}}$, $\boldsymbol{F}^{\dagger}\boldsymbol{F}=\boldsymbol{I}_r$.

(2) $\boldsymbol{G}^{\dagger}=\boldsymbol{G}^{\mathrm{H}}(\boldsymbol{G}\boldsymbol{G}^{\mathrm{H}})^{-1}$, $\boldsymbol{G}\boldsymbol{G}^{\dagger}=\boldsymbol{I}_r$.

证明 先证结论 (1). 对于 $\boldsymbol{F}$, 记 $\boldsymbol{X}=(\boldsymbol{F}^{\mathrm{H}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathrm{H}}$, 则有

$$\boldsymbol{F}\boldsymbol{X}\boldsymbol{F}=\boldsymbol{F}(\boldsymbol{F}^{\mathrm{H}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathrm{H}}\boldsymbol{F}=\boldsymbol{F},\quad \boldsymbol{X}\boldsymbol{F}\boldsymbol{X}=(\boldsymbol{F}^{\mathrm{H}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathrm{H}}\boldsymbol{F}\boldsymbol{X}=\boldsymbol{X},$$

$$(\boldsymbol{F}\boldsymbol{X})^{\mathrm{H}}=\boldsymbol{X}^{\mathrm{H}}\boldsymbol{F}^{\mathrm{H}}=\boldsymbol{F}(\boldsymbol{F}^{\mathrm{H}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathrm{H}}=\boldsymbol{F}\boldsymbol{X},\quad (\boldsymbol{X}\boldsymbol{F})^{\mathrm{H}}=\boldsymbol{I}_r^{\mathrm{H}}=\boldsymbol{I}_r=\boldsymbol{X}\boldsymbol{F}.$$

所以 $\boldsymbol{F}^{\dagger}=(\boldsymbol{F}^{\mathrm{H}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathrm{H}}$, 并且 $\boldsymbol{F}^{\dagger}\boldsymbol{F}=\boldsymbol{I}_r$.

同理, 可证结论 (2). 证毕. □

定理 1.23 设 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$, 则 $\boldsymbol{A}^{\dagger}$ 存在且唯一.

证明 存在性. 当 $\boldsymbol{A}=\boldsymbol{O}$ 时, $\boldsymbol{A}^{\dagger}=\boldsymbol{O}$; 当 $\boldsymbol{A}\neq\boldsymbol{O}$ 时, $\mathrm{rank}(\boldsymbol{A})=r\geqslant 1$, 从而 $\boldsymbol{A}$ 有满秩分解 $\boldsymbol{A}=\boldsymbol{F}\boldsymbol{G}$ $(\boldsymbol{F}\in\mathbb{C}_r^{m\times r}, \boldsymbol{G}\in\mathbb{C}_r^{r\times n})$. 记 $\boldsymbol{X}=\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger}$, 则有

$$\begin{aligned}
&\boldsymbol{A}\boldsymbol{X}\boldsymbol{A}=\boldsymbol{F}\boldsymbol{G}\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger}\boldsymbol{F}\boldsymbol{G}=\boldsymbol{F}(\boldsymbol{G}\boldsymbol{G}^{\dagger})(\boldsymbol{F}^{\dagger}\boldsymbol{F})\boldsymbol{G}=\boldsymbol{F}\boldsymbol{G}=\boldsymbol{A},\\
&\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}=\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger}\boldsymbol{F}\boldsymbol{G}\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger}=\boldsymbol{G}^{\dagger}(\boldsymbol{F}^{\dagger}\boldsymbol{F})(\boldsymbol{G}\boldsymbol{G}^{\dagger})\boldsymbol{F}^{\dagger}=\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger}=\boldsymbol{X},\\
&(\boldsymbol{A}\boldsymbol{X})^{\mathrm{H}}=(\boldsymbol{F}\boldsymbol{G}\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger})^{\mathrm{H}}=(\boldsymbol{F}\boldsymbol{F}^{\dagger})^{\mathrm{H}}=\boldsymbol{F}\boldsymbol{F}^{\dagger}=\boldsymbol{F}\boldsymbol{G}\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger}=\boldsymbol{A}\boldsymbol{X},\\
&(\boldsymbol{X}\boldsymbol{A})^{\mathrm{H}}=(\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger}\boldsymbol{F}\boldsymbol{G})^{\mathrm{H}}=(\boldsymbol{G}^{\dagger}\boldsymbol{G})^{\mathrm{H}}=\boldsymbol{G}^{\dagger}\boldsymbol{G}=\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger}\boldsymbol{F}\boldsymbol{G}=\boldsymbol{X}\boldsymbol{A},
\end{aligned}$$

由定义可得

$$\boldsymbol{A}^{\dagger}=\boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger}=\boldsymbol{G}^{\mathrm{H}}(\boldsymbol{F}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{G}^{\mathrm{H}})^{-1}\boldsymbol{F}^{\mathrm{H}}. \tag{1.24}$$

唯一性. 对于 $\boldsymbol{A}$, 如果 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 都满足 Penrose 方程, 则有

$$\begin{aligned}
\boldsymbol{X}&=\boldsymbol{X}\boldsymbol{A}\boldsymbol{X}=\boldsymbol{X}\cdot\boldsymbol{A}\boldsymbol{Y}\boldsymbol{A}\cdot\boldsymbol{X}=\boldsymbol{X}\cdot(\boldsymbol{A}\boldsymbol{Y})(\boldsymbol{A}\boldsymbol{X})\\
&=\boldsymbol{X}\cdot(\boldsymbol{A}\boldsymbol{Y})^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{X})^{\mathrm{H}}=\boldsymbol{X}\cdot(\boldsymbol{A}\boldsymbol{X}\boldsymbol{A}\boldsymbol{Y})^{\mathrm{H}}=\boldsymbol{X}\cdot(\boldsymbol{A}\boldsymbol{Y})^{\mathrm{H}}\\
&=\boldsymbol{X}\boldsymbol{A}\boldsymbol{Y}=\boldsymbol{X}\cdot\boldsymbol{A}\boldsymbol{Y}\boldsymbol{A}\cdot\boldsymbol{Y}=(\boldsymbol{X}\boldsymbol{A})(\boldsymbol{Y}\boldsymbol{A})\cdot\boldsymbol{Y}\\
&=(\boldsymbol{X}\boldsymbol{A})^{\mathrm{H}}(\boldsymbol{Y}\boldsymbol{A})^{\mathrm{H}}\cdot\boldsymbol{Y}=(\boldsymbol{Y}\boldsymbol{A}\boldsymbol{X}\boldsymbol{A})^{\mathrm{H}}\cdot\boldsymbol{Y}\\
&=(\boldsymbol{Y}\boldsymbol{A})^{\mathrm{H}}\cdot\boldsymbol{Y}=\boldsymbol{Y}\boldsymbol{A}\boldsymbol{Y}=\boldsymbol{Y},
\end{aligned}$$

所以 $\boldsymbol{A}^{\dagger}$ 唯一. 证毕. □

定理 1.23 的存在性证明实际上给出了使用矩阵满秩分解的方法来计算非零矩阵广义逆. 下面给出一个例子.

例 1.3 已知

$$\boldsymbol{A}=\begin{bmatrix}1&2&1&2\\0&1&0&1\\1&0&1&0\\2&1&2&1\end{bmatrix},$$

求 $\boldsymbol{A}$ 的满秩分解和 $\boldsymbol{A}^{\dagger}$.

解 $\boldsymbol{A} \xrightarrow{\text{初等行变换}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$: $c_1 = 1, c_2 = 2$,

$$\boldsymbol{A} = \boldsymbol{F}\boldsymbol{G}: \quad \boldsymbol{F} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

因此, 有

$$\boldsymbol{F}^{\dagger} = (\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathrm{T}} = \begin{bmatrix} 6 & 4 \\ 4 & 6 \end{bmatrix}^{-1} \boldsymbol{F}^{\mathrm{T}} = \frac{1}{10}\begin{bmatrix} -1 & -2 & 3 & 4 \\ 4 & 3 & -2 & -1 \end{bmatrix},$$

$$\boldsymbol{G}^{\dagger} = \boldsymbol{G}^{\mathrm{T}}(\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}})^{-1} = \boldsymbol{G}^{\mathrm{T}}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\boldsymbol{A}^{\dagger} = \boldsymbol{G}^{\dagger}\boldsymbol{F}^{\dagger} = \frac{1}{20}\begin{bmatrix} -1 & -2 & 3 & 4 \\ 4 & 3 & -2 & -1 \\ -1 & -2 & 3 & 4 \\ 4 & 3 & -2 & -1 \end{bmatrix}.$$

下面的定理说明使用矩阵奇异值分解的方法来计算广义逆更为便捷.

定理 1.24 设 $\boldsymbol{A} \in \mathbb{C}_r^{m\times n}\,(r \geqslant 1)$ 的奇异值分解为

$$\boldsymbol{A} = \boldsymbol{U}\begin{bmatrix} \boldsymbol{\Sigma}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}\boldsymbol{V}^{\mathrm{H}},$$

式中: $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 及 $\boldsymbol{\Sigma}_r$ 的意义同式 (1.5), 则

$$\boldsymbol{A}^{\dagger} = \boldsymbol{V}\begin{bmatrix} \boldsymbol{\Sigma}_r^{-1} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}\boldsymbol{U}^{\mathrm{H}}. \tag{1.25}$$

证明 根据 Moore–Penrose 广义逆的定义, 直接验证即可证明.

注 1.5 需要指出, 可逆矩阵的逆 $\boldsymbol{A}^{-1}$ 具有的性质, 对于一般矩阵的广义逆 $\boldsymbol{A}^{\dagger}$ 不一定具有, 例如:

(1) $(\boldsymbol{A}\boldsymbol{B})^{\dagger} \neq \boldsymbol{B}^{\dagger}\boldsymbol{A}^{\dagger}$: 取 $\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $\boldsymbol{B} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, 则

$$\boldsymbol{A}\boldsymbol{B} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad (\boldsymbol{A}\boldsymbol{B})^{\dagger} = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix},$$

$$\boldsymbol{A}^{\dagger}=\begin{bmatrix}1&0\\0&0\end{bmatrix},\quad \boldsymbol{B}^{\dagger}=\boldsymbol{B}^{-1}=\begin{bmatrix}1&-1\\0&1\end{bmatrix},\quad \boldsymbol{B}^{\dagger}\boldsymbol{A}^{\dagger}=\begin{bmatrix}1&0\\0&0\end{bmatrix}.$$

(2) $\boldsymbol{A}^{\dagger}\boldsymbol{A}\neq\boldsymbol{A}\boldsymbol{A}^{\dagger}$: $\boldsymbol{A}$ 不是方阵时, 这是明显的. $\boldsymbol{A}$ 是方阵但不可逆时, 取

$$\boldsymbol{A}=\begin{bmatrix}1&1\\0&0\end{bmatrix},\quad \boldsymbol{A}^{\dagger}=\frac{1}{2}\begin{bmatrix}1&0\\1&0\end{bmatrix},$$

则

$$\boldsymbol{A}^{\dagger}\boldsymbol{A}=\frac{1}{2}\begin{bmatrix}1&1\\1&1\end{bmatrix},\quad \boldsymbol{A}\boldsymbol{A}^{\dagger}=\begin{bmatrix}1&0\\0&0\end{bmatrix}.$$

1.5 几种特殊的矩阵类型

1. 不可约矩阵和对角占优矩阵

定义 1.21 设 $\boldsymbol{A}$ 是一个 $n(n\geqslant 2)$ 阶矩阵, 如果存在 n 阶置换矩阵 $\boldsymbol{P}$, 使得

$$\boldsymbol{P}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P}=\begin{bmatrix}\boldsymbol{A}_{11}&\boldsymbol{A}_{12}\\\boldsymbol{O}&\boldsymbol{A}_{22}\end{bmatrix},$$

式中: $\boldsymbol{A}_{11}$ 和 $\boldsymbol{A}_{22}$ 分别为 r 阶和 $n-r$ 阶方阵 $(1\leqslant r\leqslant n-1)$. 则称 $\boldsymbol{A}$ 为可约矩阵. 如果不存在这样的置换矩阵, 则称 $\boldsymbol{A}$ 为不可约矩阵.

可约矩阵的一个等价定义如下.

定义 1.21′ 设 $\boldsymbol{A}=(a_{ij})\in\mathbb{C}^{n\times n}(n\geqslant 2)$. 又设 $\mathcal{I}=\{1,2,\cdots,n\}$. 若存在 $\mathcal{I}$ 的一个非空真子集 $\mathcal{K}$, 使得 $a_{ij}=0(i\notin\mathcal{K},j\in\mathcal{K})$, 则称 $\boldsymbol{A}$ 为可约矩阵. 否则, 称 $\boldsymbol{A}$ 为不可约矩阵.

用定义 1.21′ 判定一个矩阵是否可约更为方便.

例 1.4 设 $\boldsymbol{A}=(a_{ij})$ 为三对角矩阵, 即满足 $a_{ij}=0(|i-j|>1)$. 证明: 若

$$a_{i+1,i}\neq 0,\quad a_{i,i+1}\neq 0,\quad i=1,2,\cdots,n-1,$$

则 $\boldsymbol{A}$ 是不可约矩阵.

证明 用反证法. 若 $\boldsymbol{A}$ 可约, 则由定义 1.21′ 知, 存在 $\mathcal{I}$ 的非空真子集 $\mathcal{K}$, 使得 $a_{ij}=0(i\notin\mathcal{K},j\in\mathcal{K})$. 由于 $\mathcal{K}$ 非空, 则 $\mathcal{K}$ 至少有一个数, 设为 r. 由 $a_{r,r+1}\neq 0$ 知, $r+1\in\mathcal{K}$. 类似可得 $r+2\in\mathcal{K},\cdots,n\in\mathcal{K}$. 另外, 又由 $a_{r-1,r}\neq 0$ 知, $r-1\in\mathcal{K}$. 类似可得 $r-2\in\mathcal{K},\cdots,1\in\mathcal{K}$. 即有 $\mathcal{K}=\mathcal{I}$, 这与 $\mathcal{K}$ 是 $\mathcal{I}$ 的真子集矛盾. 故 $\boldsymbol{A}$ 为不可约矩阵.

定义 1.22 设 $\boldsymbol{A}$ 是一个 n 阶矩阵.

(1) 若 $\boldsymbol{A}$ 满足

$$|a_{ii}| \geqslant \sum_{j\neq i} |a_{ij}|,\ i = 1, 2, \cdots, n,$$

且其中至少有个严格不等式成立, 则称 $\boldsymbol{A}$ 为 (行) 弱对角占优.

(2) 若 $\boldsymbol{A}$ 满足

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}|,\ i = 1, 2, \cdots, n,$$

则称 $\boldsymbol{A}$ 为 (行) 严格对角占优.

类似可定义 (列) 弱对角占优和 (列) 严格对角占优.

以后往往不严格区分按行 (严格) 对角占优或按列 (严格) 对角占优而模糊地统称为 (严格) 对角占优. 弱对角占优且不可约的矩阵简称为不可约对角占优.

引理 1.1 若 n 阶矩阵 $\boldsymbol{A}$ 满足下列条件之一:

(1) 严格对角占优.

(2) 不可约对角占优.

则 $\boldsymbol{A}$ 非奇异, 即其行列式 $\det(\boldsymbol{A}) \neq 0$.

证明 首先考虑 $\boldsymbol{A}$ 为严格对角占优的情形. 用反证法. 假设 $\boldsymbol{A}$ 是奇异的, 则存在非零向量 $\boldsymbol{x}$ 使得 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$. 不妨设 $|x_i| = \|\boldsymbol{x}\|_\infty = 1$. 则有

$$|a_{ii}| = |a_{ii}x_i| = \left|\sum_{j\neq i} a_{ij}x_j\right| \leqslant \sum_{j\neq i} |a_{ij}|,$$

这与 $\boldsymbol{A}$ 严格对角占优矛盾.

再考虑 $\boldsymbol{A}$ 为不可约对角占优时结论成立. 仍用反证法. 设非零向量 $\boldsymbol{x}$ 满足 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, 并定义

$$\mathcal{K} = \{k : |x_k| < 1\}.$$

易知 $\mathcal{K}$ 非空. 否则 $\boldsymbol{x}$ 的所有分量的模均为 1, 则对于所有的 $i \in \mathcal{K}$, 均有

$$|a_{ii}| \leqslant \sum_{j\neq i} |a_{ij}|,$$

这与 $\boldsymbol{A}$ 对角占优矛盾. 此外, 因 $\boldsymbol{A}$ 不可约, 必存在 i, k 使得

$$a_{ik} \neq 0,\ i \notin \mathcal{K},\ k \in \mathcal{K}.$$

于是 $|a_{ik}x_k| < |a_{ik}|$, 且

$$|a_{ii}| \leqslant \sum_{j\notin\mathcal{K}, j\neq i} |a_{ij}|\cdot|x_j| + \sum_{j\in\mathcal{K}} |a_{ij}|\cdot|x_j| < \sum_{j\notin\mathcal{K}, j\neq i} |a_{ij}| + \sum_{j\in\mathcal{K}} |a_{ij}| = \sum_{j\neq i} |a_{ij}|,$$

这又与 $\boldsymbol{A}$ 对角占优矛盾. 证毕. □

2. 非负矩阵和 M 矩阵

非负矩阵在迭代法中起着非常重要的作用. 一个非负矩阵就是其元素全为非负数的矩阵. 对两个 $m \times n$ 阶矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$, 如果其所有相应的元素满足 $a_{ij} \geqslant b_{ij}$, 则称 $\boldsymbol{A} \geqslant \boldsymbol{B}$, 这样"$\geqslant$"就定义了矩阵集合上的一个偏序关系. 利用这种偏序关系, 非负矩阵 $\boldsymbol{A}$ 可以表示为 $\boldsymbol{A} \geqslant \boldsymbol{O}$. $\boldsymbol{A} \geqslant \boldsymbol{B}$ 也可表示为 $\boldsymbol{B} \leqslant \boldsymbol{A}$, 即"$\leqslant$"也定义了矩阵集合上的一个偏序关系.

性质 1.9 关于非负矩阵, 下列性质成立:

(1) 若 $\boldsymbol{A} \geqslant \boldsymbol{O}, \boldsymbol{B} \geqslant \boldsymbol{O}$, 则 $\boldsymbol{AB} \geqslant \boldsymbol{O}, \boldsymbol{A}+\boldsymbol{B} \geqslant \boldsymbol{O}$ 且 $\boldsymbol{A}^k \geqslant \boldsymbol{O}$.

(2) 若 $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ 为非负矩阵且 $\boldsymbol{A} \leqslant \boldsymbol{B}$, 则 $\boldsymbol{AC} \leqslant \boldsymbol{BC}, \boldsymbol{CA} \leqslant \boldsymbol{CB}$.

(3) 若 $\boldsymbol{O} \leqslant \boldsymbol{A} \leqslant \boldsymbol{B}$, 则 $\boldsymbol{A}^{\mathrm{T}} \leqslant \boldsymbol{B}^{\mathrm{T}}$ 且 $\boldsymbol{A}^k \leqslant \boldsymbol{B}^k$ 对任意的 k 均成立.

(4) 若 $\boldsymbol{O} \leqslant \boldsymbol{A} \leqslant \boldsymbol{B}$, 则 $\|\boldsymbol{A}\|_1 \leqslant \|\boldsymbol{B}\|_1$, 且类似地 $\|\boldsymbol{A}\|_\infty \leqslant \|\boldsymbol{B}\|_\infty$.

定理 1.25 (Perron–Frobenius 定理) 设 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, 则有下列结论成立:

(1) 若 $\boldsymbol{A} \geqslant \boldsymbol{O}$, 则 $\boldsymbol{A}$ 有一个非负特征值等于 $\rho(\boldsymbol{A})$; 对 $\rho(\boldsymbol{A}) > 0$, 相应的特征向量 $\boldsymbol{x} \geqslant \boldsymbol{0}$, 且当 $\boldsymbol{A}$ 的任意元素增加时, $\rho(\boldsymbol{A})$ 不减少.

(2) 若 $\boldsymbol{A} \geqslant \boldsymbol{O}$ 不可约, 则 $\boldsymbol{A}$ 有一个正特征值等于 $\rho(\boldsymbol{A})$; 对 $\rho(\boldsymbol{A}) > 0$, 其对应的特征向量 $\boldsymbol{x} > \boldsymbol{0}$; 当 $\boldsymbol{A}$ 的任意元素增加时, $\rho(\boldsymbol{A})$ 增加; 且 $\rho(\boldsymbol{A})$ 是 $\boldsymbol{A}$ 的一个简单特征值.

性质 1.10 设 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, 则有下列性质成立:

(1) 若 $\boldsymbol{O} \leqslant \boldsymbol{A} \leqslant \boldsymbol{B}$, 则 $\rho(\boldsymbol{A}) \leqslant \rho(\boldsymbol{B})$.

(2) 若 $|\boldsymbol{A}| \leqslant \boldsymbol{B}$ ($|\boldsymbol{A}|$ 表示 $\boldsymbol{A}$ 的元素取绝对值后得到的矩阵), 则 $\rho(\boldsymbol{A}) \leqslant \rho(\boldsymbol{B})$.

(3) 设 $\boldsymbol{A} \geqslant \boldsymbol{O}$, 则 $\rho(\boldsymbol{A}) < 1$ 当且仅当 $\boldsymbol{I}-\boldsymbol{A}$ 非奇异且 $(\boldsymbol{I}-\boldsymbol{A})^{-1} \geqslant \boldsymbol{O}$.

(4) 设 $\boldsymbol{A} \geqslant \boldsymbol{O}, \boldsymbol{x} \geqslant \boldsymbol{0}$ 为非零向量, $\alpha > 0$. 若 $\boldsymbol{Ax} > (\geqslant) \alpha \boldsymbol{x}$, 则 $\rho(\boldsymbol{A}) > (\geqslant) \alpha$; 若 $\boldsymbol{Ax} < \alpha \boldsymbol{x}$, 则 $\rho(\boldsymbol{A}) < \alpha$.

定义 1.23 (1) 设 $\boldsymbol{A}$ 是一个 n 阶矩阵. 若 $\boldsymbol{A}$ 的主对角元是正的, 即 $a_{ii} > 0, i = 1, 2, \cdots, n$, 而其非主对角元是非正的, 即 $a_{ij} \leqslant 0, i \neq j, i, j = 1, 2, \cdots, n$, 则称 $\boldsymbol{A}$ 为 Z 矩阵. 若 Z 矩阵 $\boldsymbol{A}$ 为非奇异的, 且 $\boldsymbol{A}^{-1} \geqslant \boldsymbol{O}$, 则称 $\boldsymbol{A}$ 为 M 矩阵.

(2) 记矩阵 $\langle \boldsymbol{A} \rangle = (\alpha_{ij})$, 其中主对角元 $\alpha_{ii} = |a_{ii}|$, 而非主对角元 $\alpha_{ij} = -|a_{ij}|$, 则称 $\langle \boldsymbol{A} \rangle$ 为 $\boldsymbol{A}$ 的比较矩阵. 若 $\langle \boldsymbol{A} \rangle$ 为非奇异的 M 矩阵, 则称 $\boldsymbol{A}$ 为 H 矩阵.

关于 M 矩阵和 H 矩阵, 有下面一些常用的性质.

性质 1.11 设 $\boldsymbol{A}$ 是 Z 矩阵, 则 $\boldsymbol{A}$ 是 M 矩阵当且仅当 $\rho(\boldsymbol{B}) < 1$, 其中 $\boldsymbol{B} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A}$ 且 $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{A})$.

性质 1.12 设 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, 则有下列性质成立:

(1) 严格对角占优或不可约对角占优矩阵是 H 矩阵.

(2) 严格对角占优或不可约对角占优矩阵的 Z 矩阵是 M 矩阵.

(3) 若 $\boldsymbol{A}$ 为 H 矩阵, 则 $|\boldsymbol{A}|^{-1} \leqslant \langle \boldsymbol{A} \rangle^{-1}$.

(4) 若 $\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{B}$ 为 H 矩阵, $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{A})$, 则 $|\boldsymbol{D}|$ 非奇异, 且 $\rho(|\boldsymbol{D}|^{-1}|\boldsymbol{B}|) < 1$.

性质 1.13 设 $\boldsymbol{A},\boldsymbol{B}\in\mathbb{R}^{n\times n}$ 是两个非奇异的 M 矩阵, 且 $\boldsymbol{A}\leqslant\boldsymbol{B}$, 则

(1) $\boldsymbol{A}^{-1}\geqslant\boldsymbol{B}^{-1}$.

(2) $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的每个主子矩阵也是 M 矩阵.

(3) 满足 $\boldsymbol{A}\leqslant\boldsymbol{C}\leqslant\boldsymbol{B}$ 的 $\boldsymbol{C}$ 都是 M 矩阵. 特别地, 若 $\boldsymbol{A}\leqslant\boldsymbol{C}\leqslant\operatorname{diag}(\boldsymbol{A})$, 则 $\boldsymbol{C}$ 是 M 矩阵.

性质 1.14 若 $\boldsymbol{A}$ 是 M 矩阵, $\boldsymbol{B}$ 是 Z 矩阵, 且 $A\leqslant B$, 则 $\boldsymbol{B}$ 也是 M 矩阵.

性质 1.15 (1) 若分块矩阵 $\boldsymbol{A}=\left(\boldsymbol{A}_{ij}\right)_{i,j=1}^{m}$ 是 M 矩阵, 且其主对角块 $\boldsymbol{A}_{ii}$ 为方阵, 则 $\boldsymbol{A}_{ii}$ 也是 M 矩阵.

(2) 若 $\boldsymbol{A}=\begin{bmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12}\\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22}\end{bmatrix}$ 是 M 矩阵, 则 Schur 补 $\boldsymbol{S}=\boldsymbol{A}_{22}-\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}$ 也是 M 矩阵.

3. 正定矩阵

关于正定矩阵, 在实矩阵的情形, 前面已作了介绍. 下面考虑复矩阵的情形.

定义 1.24 (1) 设矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, 若对任意非零向量 $\boldsymbol{x}\in\mathbb{C}^{n}$ 都有 $\operatorname{Re}(\boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x})>0$, 则称 $\boldsymbol{A}$ 为正定的. 若 $\boldsymbol{A}$ 还是 Hermite 的, 则称为 Hermite 正定的, 记为 HPD.

(2) 设矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, 若对任意非零向量 $\boldsymbol{x}\in\mathbb{R}^{n}$ 都有 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}>0$, 则称 $\boldsymbol{A}$ 为正实的.

任意 (实或复) 方阵 $\boldsymbol{A}$ 均可分裂为

$$\boldsymbol{A}=\boldsymbol{H}+\mathrm{i}\boldsymbol{S},\tag{1.26}$$

式中:

$$\boldsymbol{H}=\frac{1}{2}\left(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{H}}\right),\quad \boldsymbol{S}=\frac{1}{2\mathrm{i}}\left(\boldsymbol{A}-\boldsymbol{A}^{\mathrm{H}}\right).$$

注意到 $\boldsymbol{H}$ 和 $\boldsymbol{S}$ 都是 Hermite 的, 而 $\mathrm{i}\boldsymbol{S}$ 是反 Hermite 的. 矩阵 $\boldsymbol{H}$ 和 $\mathrm{i}\boldsymbol{S}$ 分别称为 $\boldsymbol{A}$ 的 Hermite 部分和反 Hermite 部分. 当 $\boldsymbol{A}$ 是实矩阵且 $\boldsymbol{u}$ 是实向量时, 有 $(\boldsymbol{A}\boldsymbol{u},\boldsymbol{u})=(\boldsymbol{H}\boldsymbol{u},\boldsymbol{u})$. 分裂 (1.26) 称为矩阵 $\boldsymbol{A}$ 的一个 Hermite–反 Hermite 分裂, 简称 HS 分裂.

正定矩阵具有下面一些常用的性质.

性质 1.16 关于矩阵 $\boldsymbol{A}$, 下列性质成立:

(1) 设 $\boldsymbol{A}$ 为实正定矩阵, 则 $\boldsymbol{A}$ 是非奇异的, 且对任意的实向量 $\boldsymbol{x}$, 存在标量 $\alpha>0$ 满足 $(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x})\geqslant\alpha\|\boldsymbol{x}\|^{2}$.

(2) 正定矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的所有特征特均有正的实部.

(3) 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 为正定矩阵, $\boldsymbol{P}\in\mathbb{C}^{n\times n}$ 为 n 阶非奇异矩阵, 则 $\boldsymbol{P}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{P}$ 亦为正定矩阵.

(4) 正定矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的所有主子阵均为正定矩阵. 特别地, $\boldsymbol{A}$ 的所有对角元素均有正的实部.

(5) 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, 则 $\boldsymbol{A}$ 是正定矩阵当且仅当 $\boldsymbol{A}+\boldsymbol{A}^{\mathrm{H}}$ 为 Hermite 正定的.

定理 1.26 对于矩阵 $\boldsymbol{A}$ 的 HS 分裂 (1.26), 其特征值 λ_i 满足

$$\lambda_{\min}(\boldsymbol{H}) \leqslant \mathrm{Re}(\lambda_i) \leqslant \lambda_{\max}(\boldsymbol{H}), \quad \lambda_{\min}(\boldsymbol{S}) \leqslant \mathrm{Im}(\lambda_i) \leqslant \lambda_{\max}(\boldsymbol{S}).$$

当 $\boldsymbol{B}$ 为对称正定矩阵时, $\mathbb{C}^n \times \mathbb{C}^n$ 到 $\mathbb{C}$ 上的映射 $\boldsymbol{x}, \boldsymbol{y} \to (\boldsymbol{x}, \boldsymbol{y})_{\boldsymbol{B}} \equiv (\boldsymbol{B}\boldsymbol{x}, \boldsymbol{y})$ 是 $\mathbb{C}^n$ 上的一个内积, 称此内积为能量内积. 而称其诱导范数为能量范数. 有时, 可以找到一个 HPD 矩阵 $\boldsymbol{B}$, 使得一个给定的在欧几里得内积下的非 Hermite 矩阵 $\boldsymbol{A}$ 在能量内积下成为 Hermite 的, 即:

$$(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{y})_{\boldsymbol{B}} = (\boldsymbol{x}, \boldsymbol{A}\boldsymbol{y})_{\boldsymbol{B}}, \quad \forall\, \boldsymbol{x}, \boldsymbol{y}.$$

最简单的例子是 $\boldsymbol{A} = \boldsymbol{B}^{-1}\boldsymbol{C}$ 和 $\boldsymbol{A} = \boldsymbol{C}\boldsymbol{B}$, 其中 $\boldsymbol{C}$ 是 Hermite 的且 $\boldsymbol{B}$ 是 Hermite 正定的.

1.6 模型问题: Poisson 问题

考虑二维规则区域上的 Poisson 问题

$$\begin{cases} -\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right) + c\dfrac{\partial u}{\partial x} + d\dfrac{\partial u}{\partial y} + eu = f(x,y), \quad (x,y) \in \Omega, \\ u(x,y)|_{\partial\Omega} = 0, \end{cases} \tag{1.27}$$

式中: $\Omega = (0,1) \times (0,1)$; f 为给定的函数; c, d, e 为非负常数.

设 x, y 方向均取 $n+1$ 个等距网格, 则内网格点共有 n^2 个, 步长为 $h = \dfrac{1}{n+1}$, 如图 1.1 所示.

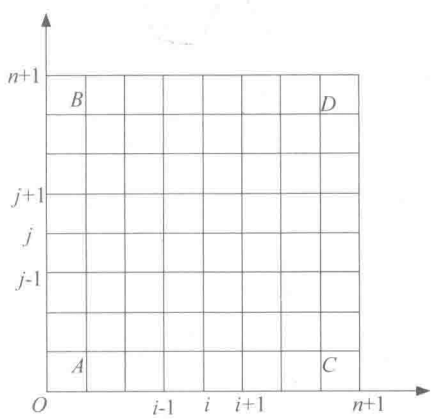

图 1.1 区域离散和节点标号

将

$$u_{xx}(x_i, y_j) \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2}, \quad u_{yy}(x_i, y_j) \approx \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h^2},$$

$$u_x(x_i, y_j) \approx \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \quad u_y(x_i, y_j) \approx \frac{u_{i,j+1} - u_{i,j-1}}{2h},$$

$$u(x_i, y_j) \approx u_{ij}, \quad f(x_i, y_j) \approx f_{ij}.$$

代入式 (1.27) 的第 1 式并整理, 得

$$(4+h^2e)u_{ij} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) + \frac{hc}{2}(u_{i+1,j} - u_{i-1,j})$$

$$+\frac{hd}{2}(u_{i,j+1} - u_{i,j-1}) = h^2 f_{ij}, \quad i,j = 1,2,\cdots,n.$$

该方程组包含了 n^2 个未知量, 它的矩阵形式为

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}, \tag{1.28}$$

式中: $\boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2 + \boldsymbol{A}_3$ 为 n^2 阶的块三对角矩阵, 其具体形式为

$$\boldsymbol{A}_1 = \begin{bmatrix} \boldsymbol{B} & -\boldsymbol{I} & & & \\ -\boldsymbol{I} & \boldsymbol{B} & -\boldsymbol{I} & & \\ & \ddots & \ddots & \ddots & \\ & & -\boldsymbol{I} & \boldsymbol{B} & -\boldsymbol{I} \\ & & & -\boldsymbol{I} & \boldsymbol{B} \end{bmatrix}, \tag{1.29}$$

$$\boldsymbol{A}_2 = \frac{hc}{2}\begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} & & & \\ -\boldsymbol{I} & \boldsymbol{O} & \boldsymbol{I} & & \\ & \ddots & \ddots & \ddots & \\ & & -\boldsymbol{I} & \boldsymbol{O} & \boldsymbol{I} \\ & & & -\boldsymbol{I} & \boldsymbol{O} \end{bmatrix}, \tag{1.30}$$

$$\boldsymbol{A}_3 = \frac{hd}{2}\begin{bmatrix} \boldsymbol{O} & -\boldsymbol{I} & & & \\ \boldsymbol{I} & \boldsymbol{O} & -\boldsymbol{I} & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{I} & \boldsymbol{O} & -\boldsymbol{I} \\ & & & \boldsymbol{I} & \boldsymbol{O} \end{bmatrix}, \tag{1.31}$$

这里

$$\boldsymbol{B} = \begin{bmatrix} 4+h^2e & -1 & & & \\ -1 & 4+h^2e & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4+h^2e & -1 \\ & & & -1 & 4+h^2e \end{bmatrix} \tag{1.32}$$

为对称三对角矩阵, $\boldsymbol{I}$ 为 n 阶单位矩阵. n^2 维列向量 $\boldsymbol{u}$ 定义为

$$\boldsymbol{u} = (u_{11}, u_{12}, \cdots, u_{1n}, u_{21}, u_{22}, \cdots, u_{2n}, \cdots, u_{n1}, u_{n2}, \cdots, u_{nn})^{\mathrm{T}}, \tag{1.33}$$

网格中的每个内部节点都对应一个分量, u_{ij} 表示 $u(ih, jh)$ 的近似, 这些分量按沿 y 方向变化最快的方式排列. 图 1.1 中节点 A, B, C, D 的标号分别为 $(1,1), (1,n), (n,1), (n,n)$, 节点 (i,j) 对应的标号为 $i*n+j$, 它的上下左右的四个相邻节点为

$$i*n+(j+1),\quad i*n+(j-1),\quad (i-1)*n+j,\quad (i+1)*n+j.$$

右端向量 $\boldsymbol{f}$ 为

$$\boldsymbol{f} = h^2(f_{11}, f_{12}, \cdots, f_{1n}, f_{21}, f_{22}, \cdots, f_{2n}, \cdots, f_{n1}, f_{n2}, \cdots, f_{nn})^{\mathrm{T}}. \tag{1.34}$$

以后常常用此模型做数值实验, 故称它为模型问题.

习题 1

1.1 设 $\boldsymbol{A}$ 为正定矩阵, $\boldsymbol{B}$ 是与 $\boldsymbol{A}$ 同阶的 Hermite 矩阵. 试证明: $\boldsymbol{A}+\boldsymbol{B}$ 为正定矩阵的充分必要条件是 $\boldsymbol{A}^{-1}\boldsymbol{B}$ 的特征值都大于 1.

1.2 设矩阵 $\boldsymbol{A}$ 是可对角化的. 试证明: $\boldsymbol{A}$ 的特征值都是实数的充分必要条件是存在正定矩阵 $\boldsymbol{B}$ 使得 $\boldsymbol{BA}$ 为 Hermite 矩阵.

1.3 设 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 为同阶半正定矩阵. 证明: $\det(\boldsymbol{A}+\boldsymbol{B}) \geqslant \det(\boldsymbol{A}) + \det(\boldsymbol{B})$.

1.4 设 $\boldsymbol{A}$ 为反对称矩阵. 证明:

(1) 矩阵 $\boldsymbol{I}-\boldsymbol{A}$ 是非奇异的;

(2) 矩阵 $\boldsymbol{C} = (\boldsymbol{I}-\boldsymbol{A})^{-1}(\boldsymbol{I}+\boldsymbol{A})$ 是正交矩阵.

1.5 设 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 为同阶幂等矩阵, 并且满足 $\mathcal{R}(\boldsymbol{A}) = \mathcal{R}(\boldsymbol{B})$ 和 $\mathcal{N}(\boldsymbol{A}) = \mathcal{N}(\boldsymbol{B})$. 证明: $\boldsymbol{A} = \boldsymbol{B}$.

1.6 设

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

求矩阵 $\boldsymbol{A}$ 的奇异值分解.

1.7 设矩阵 $\boldsymbol{A} \in \mathbb{C}_r^{m\times n}$ 的一个满秩分解为 $\boldsymbol{A} = \boldsymbol{FG}$, 求矩阵 $\begin{bmatrix} \boldsymbol{A} & \boldsymbol{A} \\ \boldsymbol{A} & \boldsymbol{A} \end{bmatrix}$ 的一个满秩分解.

1.8 设非零向量 $\boldsymbol{x} \in \mathbb{C}^n$. 证明:

(1) $\|\boldsymbol{x}\|_\infty \leqslant \|\boldsymbol{x}\|_1 \leqslant n\|\boldsymbol{x}\|_\infty$; (2) $\frac{1}{\sqrt{n}}\|\boldsymbol{x}\|_1 \leqslant \|\boldsymbol{x}\|_2 \leqslant \|\boldsymbol{x}\|_1$;

(3) $\|\boldsymbol{x}\|_\infty \leqslant \|\boldsymbol{x}\|_2 \leqslant \sqrt{n}\|\boldsymbol{x}\|_\infty$.

1.9 设 $\boldsymbol{A}$ 为 n 阶矩阵. 证明:

(1) $\frac{1}{n}\|\boldsymbol{A}\|_\infty \leqslant \|\boldsymbol{A}\|_1 \leqslant n\|\boldsymbol{A}\|_\infty$; (2) $\frac{1}{\sqrt{n}}\|\boldsymbol{A}\|_2 \leqslant \|\boldsymbol{A}\|_1 \leqslant \sqrt{n}\|\boldsymbol{A}\|_2$;

(3) $\frac{1}{\sqrt{n}}\|\boldsymbol{A}\|_2 \leqslant \|\boldsymbol{A}\|_\infty \leqslant \sqrt{n}\|\boldsymbol{A}\|_2$; (4) $\frac{1}{\sqrt{n}}\|\boldsymbol{A}\|_{\mathrm{F}} \leqslant \|\boldsymbol{A}\|_2 \leqslant \|\boldsymbol{A}\|_{\mathrm{F}}$.

1.10 设 $\mathbf{0} \neq \boldsymbol{v} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. 证明:

$$\|\boldsymbol{A}(\boldsymbol{I} - (\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v})^{-1}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}})\|_{\mathrm{F}}^2 = \|\boldsymbol{A}\|_{\mathrm{F}}^2 - \frac{\|\boldsymbol{A}\boldsymbol{v}\|_2^2}{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}}.$$

1.11 证明: $\rho(\boldsymbol{A}) < 1$ 当且仅当存在正定矩阵 $\boldsymbol{B}$, 使得 $\boldsymbol{B} - \boldsymbol{A}\boldsymbol{B}\boldsymbol{A}^{\mathrm{H}}$ 也为正定矩阵.

1.12 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 且 $\boldsymbol{A} \geqslant \boldsymbol{O}$.

(1) 若 $\boldsymbol{A}$ 的各行元素之和相等, 则 $\rho(\boldsymbol{A}) = \|\boldsymbol{A}\|_\infty$;

(2) 若 $\boldsymbol{A}$ 的各列元素之和相等, 则 $\rho(\boldsymbol{A}) = \|\boldsymbol{A}\|_1$.

1.13 设 $\boldsymbol{A}$ 为 Z 矩阵. 证明: $\boldsymbol{A}$ 是非奇异的 M 矩阵当且仅当存在 $\boldsymbol{x} > \mathbf{0}$ 使得 $\boldsymbol{A}\boldsymbol{x} > \mathbf{0}$.

1.14 设 $\boldsymbol{A}$ 是 Hermite 矩阵, 且 $\boldsymbol{X}$ 是 $\boldsymbol{A}$ 的广义逆. 证明: $\boldsymbol{X}^2$ 是 $\boldsymbol{A}^2$ 的广义逆.

1.15 证明: $\boldsymbol{B}^\dagger\boldsymbol{A}^\dagger = (\boldsymbol{A}\boldsymbol{B})^\dagger$ 当且仅当 $\mathcal{R}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{B}) \subseteq \mathcal{R}(\boldsymbol{B})$, 且 $\mathcal{R}(\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}) \subseteq \mathcal{R}(\boldsymbol{A}^{\mathrm{H}})$.

1.16 试证明下述结论:

(1) $\boldsymbol{A}^\dagger = (\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})^\dagger\boldsymbol{A}^{\mathrm{H}} = \boldsymbol{A}^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}})^\dagger$;

(2) 设 $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{C}^n$, 则 $(\boldsymbol{a}\boldsymbol{b}^{\mathrm{H}})^\dagger = (\boldsymbol{a}^{\mathrm{H}}\boldsymbol{a})^\dagger(\boldsymbol{b}^{\mathrm{H}}\boldsymbol{b})^\dagger\boldsymbol{b}\boldsymbol{a}^{\mathrm{H}}$.

1.17 设 $\boldsymbol{A}$ 为正规矩阵. 证明: $\boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{A}^\dagger\boldsymbol{A}$, 且对任一自然数 k, 有 $(\boldsymbol{A}^k)^\dagger = (\boldsymbol{A}^\dagger)^k$.

1.18 设 $\boldsymbol{A}$ 为 Z 矩阵. 证明: $\boldsymbol{A}$ 是 M 矩阵当且仅当 $\rho(\boldsymbol{B}) < 1$, 其中 $\boldsymbol{B} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A}$ 且 $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{A})$.

第 2 章　正交变换和投影方法

向量序列的正交化和矩阵的正交 (相似) 变换在数值代数中占有独特的地位. QR 分解可用于线性方程组特别是最小二乘问题的求解. 而基于 QR 分解的 QR 方法可用于计算矩阵的特征值和特征向量. 此外, Krylov 子空间的正交化可用于构造线性方程组的迭代法. 投影方法的思想则是构造线性方程组迭代法和矩阵特征值计算的基础和出发点. 本章将主要介绍两种常用的正交变换、QR 分解、线性无关向量组的正交化以及 Krylov 子空间的性质及其正交化, 最后介绍投影方法的一般框架.

2.1　两种常用的正交变换

2.1.1　Householder 变换

定义 2.1　设 $\boldsymbol{u}\in\mathbb{R}^n$ 满足 $\|\boldsymbol{u}\|_2=1$, 称 n 阶矩阵

$$\boldsymbol{H}=\boldsymbol{I}-2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} \tag{2.1}$$

为 Householder 矩阵 (初等反射矩阵), $\boldsymbol{u}$ 称为 Householder 向量.

下面的定理给出了 Householder 矩阵的性质.

定理 2.1　设 $\boldsymbol{H}$ 为式 (2.1) 定义的 Householder 矩阵, 则

(1) $\det(\boldsymbol{H})=-1$.

(2) $\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{H}$, $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}=\boldsymbol{I}$, $\boldsymbol{H}^{-1}=\boldsymbol{H}$, $\boldsymbol{H}^2=\boldsymbol{I}$.

(3) $\boldsymbol{H}$ 仅有两个互不相同的特征值 -1 和 1, 且 -1 是单重的, 相应的特征向量为 $\boldsymbol{u}$. 而 1 是 $n-1$ 重的, 相应的特征向量为所有与 $\boldsymbol{u}$ 正交的非零向量.

(4) 设 $\boldsymbol{x},\boldsymbol{u}\in\mathbb{R}^n$ 满足 $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}=0$, $\alpha\in\mathbb{R}$. 则

$$\boldsymbol{H}(\boldsymbol{x}+\alpha\boldsymbol{u})=\boldsymbol{x}-\alpha\boldsymbol{u}.$$

证明　(1) 利用分块矩阵的乘法运算可以方便地验证 $\det(\boldsymbol{H})=-1$. 事实上, 对

$$\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ -\boldsymbol{u}^{\mathrm{T}} & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & 2\boldsymbol{u}\\ \boldsymbol{0} & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{I}-2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} & \boldsymbol{0}\\ \boldsymbol{u}^{\mathrm{T}} & 1\end{bmatrix}$$

$$=\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ -\boldsymbol{u}^{\mathrm{T}} & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & 2\boldsymbol{u}\\ \boldsymbol{u}^{\mathrm{T}} & 1\end{bmatrix}=\begin{bmatrix}\boldsymbol{I} & 2\boldsymbol{u}\\ \boldsymbol{0} & 1-2\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}\end{bmatrix}$$

两端取行列式, 得 $\det(\boldsymbol{H})=1-2\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}=-1$.

(2) 利用 $\boldsymbol{H}$ 的定义, 容易验证这四个等式.

(3) 由于 $\boldsymbol{H}\boldsymbol{u}=(\boldsymbol{I}-2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})\boldsymbol{u}=-\boldsymbol{u}$, 故 -1 是 $\boldsymbol{H}$ 的一个特征值, 且几何重数至少为 1, $\boldsymbol{u}$ 为相应的特征向量. 另外, 对于任意与 $\boldsymbol{u}$ 正交的 n 维向量 $\boldsymbol{x}$, 有 $\boldsymbol{H}\boldsymbol{x}=\boldsymbol{x}$, 即 1 为 $\boldsymbol{H}$ 的一个特征值, 且几何重数至少为 $n-1$, 与 $\boldsymbol{u}$ 正交的任一非零向量作为相应

的特征向量. 由于特征值的几何重数不超过代数重数, 故特征值 -1 与 1 的代数重数分别至少为 1 与 $n-1$, 其和至少为 n. 注意到矩阵的特征值代数重数之和不会超过 n, 故 -1 与 1 的代数重数分别刚好为 1 与 $n-1$.

(4) 直接计算

$$\boldsymbol{H}(\boldsymbol{x}+\alpha\boldsymbol{u}) = (\boldsymbol{I}-2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})(\boldsymbol{x}+\alpha\boldsymbol{u}) = \boldsymbol{x}-\alpha\boldsymbol{u}.$$

证毕. □

下面的定理是 Householder 矩阵的另一个性质.

定理 2.2 设 $\boldsymbol{H}_{n-m}$ 是 $n-m$ 阶 Householder 矩阵, 则

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{H}_{n-m} \end{bmatrix}$$

是 n 阶 Householder 矩阵.

证明 设 $\boldsymbol{u}_{n-m}$ 是 $n-m$ 维单位列向量, 则有

$$\boldsymbol{H}_{n-m} = \boldsymbol{I}_{n-m} - 2\boldsymbol{u}_{n-m}\boldsymbol{u}_{n-m}^{\mathrm{T}},$$

$$\begin{aligned}
\boldsymbol{H} &= \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{n-m} - 2\boldsymbol{u}_{n-m}\boldsymbol{u}_{n-m}^{\mathrm{T}} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{n-m} \end{bmatrix} - 2\begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{u}_{n-m}\boldsymbol{u}_{n-m}^{\mathrm{T}} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{n-m} \end{bmatrix} - 2\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{n-m} \end{bmatrix}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{u}_{n-m}^{\mathrm{T}} \end{bmatrix} \\
&= \boldsymbol{I}_n - 2\boldsymbol{u}_n\boldsymbol{u}_n^{\mathrm{T}},
\end{aligned}$$

式中: $\boldsymbol{u}_n = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{n-m} \end{bmatrix} \in \mathbb{R}^n$.

由于 $\boldsymbol{u}_n^{\mathrm{T}}\boldsymbol{u}_n = \boldsymbol{u}_{n-m}^{\mathrm{T}}\boldsymbol{u}_{n-m} = 1$, 所以 $\boldsymbol{H}$ 是 n 阶 Householder 矩阵. 证毕. □

定理 2.3 设 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}} \in \mathbb{R}^n\,(n > 1)$ 为任意的非零向量, 则存在 $\boldsymbol{u} \in \mathbb{R}^n$ 满足 $\|\boldsymbol{u}\|_2 = 1$, 使得式 (2.1) 定义的 Householder 矩阵满足

$$\boldsymbol{H}\boldsymbol{x} = \alpha\boldsymbol{e}_1, \tag{2.2}$$

式中: $\alpha = \pm\|\boldsymbol{x}\|_2$. 且使得式 (2.2) 成立的 $\boldsymbol{u}$ 在相差一个符号的意义下是唯一确定的.

证明 由定义 2.1, $\boldsymbol{H}=\boldsymbol{I}-2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$, 于是

$$\boldsymbol{H}\boldsymbol{x}=\boldsymbol{x}-2(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x})\boldsymbol{u}.$$

欲使式 (2.2) 成立, 必须满足

$$2(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x})\boldsymbol{u}=\boldsymbol{x}-\alpha\boldsymbol{e}_1.$$

由于 $\|\boldsymbol{u}\|_2=1$, 可取

$$\boldsymbol{u}=\frac{\boldsymbol{x}-\alpha\boldsymbol{e}_1}{\|\boldsymbol{x}-\alpha\boldsymbol{e}_1\|_2}.$$

又因 $\boldsymbol{H}$ 是正交矩阵, 为使式 (2.2) 成立, 必须有

$$\|\boldsymbol{x}\|_2=\|\boldsymbol{H}\boldsymbol{x}\|_2=\|\alpha\boldsymbol{e}_1\|_2=|\alpha|\cdot\|\boldsymbol{e}_1\|_2=|\alpha|,$$

即 $\alpha=\pm\|\boldsymbol{x}\|_2$. 容易验证, 如上选取的 $\boldsymbol{H}$ 满足式 (2.2). 事实上, 有

$$\begin{aligned}\boldsymbol{H}\boldsymbol{x}&=\boldsymbol{x}-2\frac{(\boldsymbol{x}-\alpha\boldsymbol{e}_1)(\boldsymbol{x}-\alpha\boldsymbol{e}_1)^{\mathrm{T}}}{\|\boldsymbol{x}-\alpha\boldsymbol{e}_1\|_2^2}\boldsymbol{x}\\&=\boldsymbol{x}-\frac{2(\boldsymbol{x}-\alpha\boldsymbol{e}_1)^{\mathrm{T}}\boldsymbol{x}}{\|\boldsymbol{x}-\alpha\boldsymbol{e}_1\|_2^2}(\boldsymbol{x}-\alpha\boldsymbol{e}_1)\\&=\boldsymbol{x}-\frac{2\|\boldsymbol{x}\|_2^2-2\alpha\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}}{\|\boldsymbol{x}\|_2^2-2\alpha\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}+\alpha^2}(\boldsymbol{x}-\alpha\boldsymbol{e}_1)\\&=\boldsymbol{x}-\frac{2\alpha^2-2\alpha\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}}{\alpha^2-2\alpha\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}+\alpha^2}(\boldsymbol{x}-\alpha\boldsymbol{e}_1)\\&=\boldsymbol{x}-(\boldsymbol{x}-\alpha\boldsymbol{e}_1)=\alpha\boldsymbol{e}_1.\end{aligned}$$

此外, 由 $\boldsymbol{u}$ 的选取过程知, 这样的 $\boldsymbol{u}$ 在相差一个符号的意义下是唯一确定的. 证毕. □

注 2.1 由定理 2.3, 可按如下步骤来构造确定 $\boldsymbol{H}$ 的单位向量 $\boldsymbol{u}$:

(1) 计算 $\boldsymbol{v}=\boldsymbol{x}\pm\|\boldsymbol{x}\|_2\boldsymbol{e}_1$.

(2) 计算 $\boldsymbol{u}=\boldsymbol{v}/\|\boldsymbol{v}\|_2$.

上述计算涉及 $\|\boldsymbol{x}\|_2$ 前的符号选取问题. 如果选取

$$\boldsymbol{v}=\boldsymbol{x}-\|\boldsymbol{x}\|_2\boldsymbol{e}_1,$$

就会出现计算

$$v_1=x_1-\|\boldsymbol{x}\|_2$$

的问题, 其中 v_1,x_1 分别表示 $\boldsymbol{v},\boldsymbol{x}$ 的第 1 个分量. 在 $x_1>0$ 时, 按上式计算 v_1 可能会导致有效数字的丢失. 在这种情形可改用下面的计算方式, 即

$$v_1=x_1-\|\boldsymbol{x}\|_2=\frac{x_1^2-\|\boldsymbol{x}\|_2^2}{x_1+\|\boldsymbol{x}\|_2}=\frac{-(x_2^2+\cdots+x_n^2)}{x_1+\|\boldsymbol{x}\|_2},$$

并且只要在 $x_1>0$ 时使用这一公式计算, 就可以避免两个相近的数相减的情形.

注意到

$$\boldsymbol{H}=\boldsymbol{I}-2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}=\boldsymbol{I}-\frac{2}{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}=\boldsymbol{I}-\beta\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}},$$

式中: $\beta=2/(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v})$. 这样就没有必要显式地求出向量 $\boldsymbol{u}$, 只需求出 β 和 $\boldsymbol{v}$ 即可. 在实际计算中, 往往将 $\boldsymbol{v}$ 归化为第 1 个分量为 1 的向量 (只需作变换 $\boldsymbol{v}:=\boldsymbol{v}/v_1$ 即可). 这样做的优点是可以把 $\boldsymbol{v}$ 的后 $n-1$ 个分量保存在 $\boldsymbol{x}$ 的后 $n-1$ 个化为 0 的分量位置上, 而 $\boldsymbol{v}$ 的第一个分量 1 就无需保存了.

在计算时, 溢出现象也是必须要考虑的问题. 在上述计算中, 如果 $\boldsymbol{x}$ 某分量过大, 其平方运算可能会出现上溢的现象. 为了解决这个问题, 可以用 $\boldsymbol{x}/\|\boldsymbol{x}\|_\infty$ 代替 $\boldsymbol{x}$ 来构造 $\boldsymbol{v}$, 相当于在原来的 $\boldsymbol{v}$ 之前乘了一个常数 $\alpha=1/\|\boldsymbol{x}\|_\infty$, 而 $\alpha\boldsymbol{v}$ 与 $\boldsymbol{v}$ 的单位化向量是相同的.

基于上述讨论, 可得如下基本算法.

算法 2.1 本算法计算实 Householder 矩阵 $\boldsymbol{H}=\boldsymbol{I}-\beta\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}$ 中满足 $v_1=1$ 的 $\boldsymbol{v}$ 和 β.

步 1, 输入 n 维实向量 $\boldsymbol{x}$. 计算 $\eta=\|\boldsymbol{x}\|_\infty$, 置 $\boldsymbol{x}:=\boldsymbol{x}/\eta$.

步 2, $\boldsymbol{v}:=[1,\boldsymbol{x}(2:n)]^{\mathrm{T}}$, 计算 $\sigma=x_2^2+\cdots+x_n^2$.

步 3, 对于 $\sigma=0$, 若 $x_1\geqslant 0$, $\beta:=0$, 否则 $\beta:=2$, 终止计算.

步 4, 对于 $\sigma>0$, 计算 $\alpha:=\sqrt{x_1^2+\sigma}$. 若 $x_1\leqslant 0$, $v_1=x_1-\alpha$, 否则 $v_1=-\sigma/(x_1+\alpha)$.

步 5, 计算 $\beta:=2v_1^2/(\sigma+v_1^2)$, $\boldsymbol{v}:=\boldsymbol{v}/v_1$.

执行算法 2.1 的运算量约为 $4n$. 进一步, 根据上述算法可编制 MATLAB 程序如下:

```
%实向量Householder变换程序-r_house.m
function [v,beta]=r_house(x)
%本函数计算Householder矩阵H=I-beta*v*v'中满足v(1)=1的v和beta.
n=length(x);
eta=norm(x,inf); x=x/eta;
sigma=x(2:n)'*x(2:n);
v=[1; x(2:n)];
if sigma==0
     if x(1)>=0
         beta=0;
     else
         beta=2;
     end
else
    alpha=(x(1)^2+sigma)^0.5;
    if x(1)<=0
        v(1)=x(1)-alpha;
```

```
    else
        v(1)=-sigma/(x(1)+alpha);
    end
    beta=2*v(1)^2/(sigma+v(1)^2);
    v=v/v(1);
end
```

例 2.1 已知向量 $\boldsymbol{x}=(2,5,7,1)^{\mathrm{T}}$, 构造 Householder 矩阵 $\boldsymbol{H}$ 使得 $\boldsymbol{H}\boldsymbol{x}=\|\boldsymbol{x}\|_2\boldsymbol{e}_1$.

解 在 MATLAB 命令窗口依次运行如下代码:

```
>> x=[2,5,7,1]'; [v,beta]=r_house(x);
>> H=eye(length(x))-beta*v*v';
>> y=H*x
```

即得所需求的结果.

注 2.2 在应用 Householder 变换约化一个给定的矩阵为某一需要的形式时, 利用其特殊结构是非常重要的. 当计算 Householder 矩阵 $\boldsymbol{H}=\boldsymbol{I}-\beta\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\in\mathbb{R}^{m\times m}$ 与一个已知矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ 的乘积时, 实际计算中 $\boldsymbol{H}$ 并不需要以显式的方式给出, 而是根据如下的公式来计算:

$$\boldsymbol{H}\boldsymbol{A}=(\boldsymbol{I}-\beta\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}})\boldsymbol{A}=\boldsymbol{A}-\beta\boldsymbol{v}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{v})^{\mathrm{T}}=\boldsymbol{A}-\boldsymbol{v}\boldsymbol{w}^{\mathrm{T}},$$

式中: $\boldsymbol{w}=\beta\boldsymbol{A}^{\mathrm{T}}\boldsymbol{v}$. 换言之, 可以按下列两个步骤计算 $\boldsymbol{H}\boldsymbol{A}$:

(1) 计算 $\boldsymbol{w}=\beta\boldsymbol{A}^{\mathrm{T}}\boldsymbol{v}$.

(2) 计算 $\boldsymbol{B}=\boldsymbol{A}-\boldsymbol{v}\boldsymbol{w}^{\mathrm{T}}$.

矩阵 $\boldsymbol{B}$ 即为所求的乘积 $\boldsymbol{H}\boldsymbol{A}$. 完成这一计算任务所需要的运算量为 $4mn$.

注 2.3 在复数域情形, Householder 变换是指如下形式的矩阵

$$\boldsymbol{H}(\boldsymbol{w})=\boldsymbol{I}-2\boldsymbol{w}\boldsymbol{w}^{\mathrm{H}}, \tag{2.3}$$

式中: $\boldsymbol{w}\in\mathbb{C}^n$ 满足 $\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}=1$. 相应于实 Householder 矩阵是正交矩阵, 复 Householder 矩阵是一个酉矩阵. 事实上, 复 Householder 变换除具备实 Householder 变换相应的 (1) ~ (4) 条性质外, 还有如下性质:

(5) 设 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{C}^n$, $\boldsymbol{x}\neq\boldsymbol{y}$. 则存在 Householder 变换 $\boldsymbol{H}(\boldsymbol{w})$ 使得 $\boldsymbol{H}(\boldsymbol{w})\boldsymbol{x}=\boldsymbol{y}$ 的充分必要条件是

$$\boldsymbol{x}^{\mathrm{H}}\boldsymbol{x}=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{y},\quad \boldsymbol{x}^{\mathrm{H}}\boldsymbol{y}=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}.$$

并且在上述条件成立时, 所需的向量 $\boldsymbol{w}$ 可取为

$$\boldsymbol{w}=\mathrm{e}^{\mathrm{i}\theta}\frac{\boldsymbol{x}-\boldsymbol{y}}{\|\boldsymbol{x}-\boldsymbol{y}\|_2},$$

式中: θ 为任意实数.

由注 2.3, 若取 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\neq\boldsymbol{0}$, $\boldsymbol{y}=-\overline{\tau}\|\boldsymbol{x}\|_2\boldsymbol{e}_1$, 其中

$$\tau=\begin{cases}1, & \text{当 } x_1=0 \text{ 时},\\ \dfrac{\overline{x}_1}{|x_1|}, & \text{当 } x_1\neq 0 \text{ 时},\end{cases}\tag{2.4}$$

则满足 $\boldsymbol{x}^{\mathrm{H}}\boldsymbol{x}=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{y}$, $\boldsymbol{x}^{\mathrm{H}}\boldsymbol{y}=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}$. 此时成立

$$\boldsymbol{H}(\boldsymbol{w})\boldsymbol{x}=-\overline{\tau}\|\boldsymbol{x}\|_2\boldsymbol{e}_1,$$

式中: $\boldsymbol{w}$ 可取为

$$\boldsymbol{w}=\frac{\tau\boldsymbol{x}+\gamma\boldsymbol{e}_1}{\|\tau\boldsymbol{x}+\gamma\boldsymbol{e}_1\|_2},\quad \gamma=\|\boldsymbol{x}\|_2.$$

而且易见当 $\boldsymbol{x}$ 为实向量时, 这样得到的 $\boldsymbol{H}(\boldsymbol{w})$ 是实对称正交矩阵.

基于以上讨论, 有下面的算法.

算法 2.2 给定一个向量 $\boldsymbol{x}\in\mathbb{C}^n$, 本算法计算一个向量 $\boldsymbol{w}$ 和一个数 γ, 使得 $\boldsymbol{H}\boldsymbol{x}=\gamma\boldsymbol{e}_1$, 其中 $\boldsymbol{H}=\boldsymbol{I}-\boldsymbol{w}\boldsymbol{w}^{\mathrm{H}}$, $\|\boldsymbol{w}\|_2=\sqrt{2}$.

function $[\boldsymbol{w},\gamma]=$ **c_house**$(\boldsymbol{x})$

$\boldsymbol{w}=\boldsymbol{x}$; $\gamma=\|\boldsymbol{x}\|_2$;

if $\gamma=0$

  $w(1)=\sqrt{2}$; 结束

end

if $w(1)\neq 0$

  $\tau=\overline{w(1)}/|w(1)|$;

else

  $\tau=1$;

end

$\boldsymbol{w}=(\tau/\gamma)\boldsymbol{w}$; $w(1)=w(1)+1$;

$\boldsymbol{w}=\boldsymbol{w}/\sqrt{w(1)}$; $\gamma=-\overline{\tau}\gamma$;

注 2.4 算法 2.2 是用类似于 MATLAB 语句给出的, 后文的许多算法也将采用这种形式给出. 因此, 熟悉 MATLAB 语言是很重要的. 可以说, 用类似于 MATLAB 语言的语句来描述算法往往要比用文字和公式描述算法精到得多.

注 2.5 算法 2.1 中的 Householder 变换只是针对实向量的, 而算法 2.2 既能用于实向量又能用于复向量. 这一点是很重要的, 因为许多实际问题, 如时谐涡流场的计算, 其离散形式是一个大型稀疏且具有特殊块结构的复线性方程组. 此时使用 Krylov 子空间方法 (如 GMRES 方法) 求解时, 需要用到针对复向量的 Householder 变换.

算法 2.2 的 MATLAB 程序如下:

```
%复向量的Householder变换程序-c_house.m
function [w,gamma]=c_house(x)
%给定复向量x，本函数计算一个向量w和一个数gamma
%满足H*x=gamma*e1，其中H=I-w*w'，||w||=sqrt(2).
w=x; gamma=norm(x);
if gamma==0
    w(1)=sqrt(2); return;
end
if w(1)==0
    tau=1;
else
    tau=conj(w(1))/abs(w(1));
end
w=(tau/gamma)*w; w(1)=w(1)+1;
w=w/sqrt(w(1)); gamma=-conj(tau)*gamma;
```

例 2.2 已知复向量 $\boldsymbol{x}=(2+\mathrm{i},5-3\mathrm{i},7,1+2\mathrm{i})^{\mathrm{T}}$, 构造 Householder 矩阵 $\boldsymbol{H}$ 使得 $\boldsymbol{H}\boldsymbol{x}=\|\boldsymbol{x}\|_2\boldsymbol{e}_1$.

解 在 MATLAB 命令窗口依次运行如下代码:

```
>> x=[2+i,5-3*i,7,1+2*i]'; [w,gama]=c_house(x);
>> H=eye(length(x))-w*w'; y=H*x
```

即得所需求的结果.

2.1.2 Givens 变换

Householder 变换可以将一个向量中若干相邻分量约化为 0. 但如果将向量中的某一个分量化为 0, 则采用 Givens 变换更为有效. Givens 变换 (矩阵) 定义如下.

定义 2.2 设实数 c 和 s 满足 $c^2+s^2=1$, 称 n 阶矩阵

$$\boldsymbol{G}_{ik}(c,s)=\begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ \cdots & \cdots & c & \cdots & s & \cdots & \cdots \\ & & \vdots & \ddots & \vdots & & \\ \cdots & \cdots & -s & \cdots & c & \cdots & \cdots \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}\begin{matrix} \\ \\ (i) \\ \\ (k) \\ \\ \\ \end{matrix} \qquad (i\neq k) \tag{2.5}$$

为 Givens 变换 (矩阵), 或初等旋转矩阵.

容易证明, Givens 矩阵具有如下性质.

定理 2.4 设 $\boldsymbol{G}$ 是由式 (2.5) 定义的 Givens 矩阵. 则

(1) $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}=\boldsymbol{I}$, 即 $\boldsymbol{G}_{ik}(c,s)$ 为正交矩阵.

(2) $\det(\boldsymbol{G})=1$.

(3) $\boldsymbol{G}_{ik}(c,s)^{-1}=\boldsymbol{G}_{ik}(c,s)^{\mathrm{T}}=\boldsymbol{G}_{ik}(c,-s)$.

(4) 对于任意的 $\boldsymbol{x}\in\mathbb{R}^n$, Givens 变换 $\boldsymbol{y}=\boldsymbol{G}_{ik}(c,s)\boldsymbol{x}$ 只改变 $\boldsymbol{x}$ 的第 i,k 个分量.

证明 结论 (1) $\sim$ (3) 容易验证. 只证明结论 (4). 设 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$, $\boldsymbol{y}=(y_1,y_2,\cdots,y_n)^{\mathrm{T}}$, 则由 $\boldsymbol{y}=\boldsymbol{G}_{ik}(c,s)\boldsymbol{x}$, 得

$$y_i=cx_i+sx_k,\quad y_k=-sx_i+cx_k,\quad y_j=x_j,\quad (j\neq i,k).$$

证毕. □

由定理 2.4 的结论 (4) 不难发现, 当 $x_i^2+x_k^2\neq 0$ 时, 选取

$$c=\frac{x_i}{\sqrt{x_i^2+x_k^2}},\quad s=\frac{x_k}{\sqrt{x_i^2+x_k^2}},\tag{2.6}$$

可使 $y_i=\sqrt{x_i^2+x_k^2}>0$, $y_k=0$. 由此, 给定 $\boldsymbol{x}$ 第 i 个和第 k 个分量, 可按下列步骤计算 c 和 s 使得 $\boldsymbol{G}_{ik}(c,s)\boldsymbol{x}$ 的第 k 个分量为 0.

算法 2.3 (计算实 Givens 变换) 给定两个实数 a 和 b, 本算法计算三个数 c,s,η, 使得

$$\begin{bmatrix} c & s\\ -s & c\end{bmatrix}\begin{bmatrix} a\\ b\end{bmatrix}=\begin{bmatrix}\eta\\ 0\end{bmatrix}.$$

function $[c,s,\eta]=$ **r_givens**(a,b)
if $b=0$
  $c=1;\ s=0;\ \eta=a;$ 结束
end
if $a=0$
  $c=0;\ s=1;\ \eta=b;$ 结束
end
if $|b|\geqslant|a|$
  $t=a/b;\ s=1/\sqrt{1+t^2};\ c=st;\ \eta=|b|/s;$
else
  $t=b/a;\ c=1/\sqrt{1+t^2};\ s=ct;\ \eta=|a|/c;$
end

执行算法 2.3 只需 5 次乘除法、1 次加法和 1 次开平方运算. 如果不计算 η, 则可以减少一次除法运算. 进一步, 根据上述算法可编制 MATLAB 程序如下:

```
%Given变换程序-r_givens.m
function [c,s,eta]=r_givens(a,b)
%计算c,s满足[c s;-s c]*[a;b]=[eta;0]
```

```
if b==0, c=1; s=0; eta=a; end
if a==0, c=0; s=1; eta=b; end
if abs(b)>abs(a)
    t=a/b; s=1/sqrt(1+t^2); c=s*t;eta=abs(b)/s;
else
    t=b/a; c=1/sqrt(1+t^2); s=c*t;eta=abs(a)/c;
end
```

下面定理的结论表明 Givens 变换具有与 Householder 变换相同的功能.

定理 2.5 设 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\neq\boldsymbol{0}$, 则存在有限个 Givens 矩阵的乘积 $\boldsymbol{G}$, 使得 $\boldsymbol{G}\boldsymbol{x}=\|\boldsymbol{x}\|_2\boldsymbol{e}_1$.

证明 (1) 设 $x_1\neq 0$, 依次构造

$$\boldsymbol{G}_{12}:\ c=\frac{x_1}{\sqrt{x_1^2+x_2^2}},\quad s=\frac{x_2}{\sqrt{x_1^2+x_2^2}},$$

$$\Longrightarrow\quad \boldsymbol{G}_{12}\boldsymbol{x}=\left(\sqrt{x_1^2+x_2^2},0,x_3,\cdots,x_n\right)^{\mathrm{T}}.$$

$$\boldsymbol{G}_{13}:\ c=\frac{\sqrt{x_1^2+x_2^2}}{\sqrt{x_1^2+x_2^2+x_3^2}},\quad s=\frac{x_3}{\sqrt{x_1^2+x_2^2+x_3^2}},$$

$$\Longrightarrow\quad \boldsymbol{G}_{13}(\boldsymbol{G}_{12}\boldsymbol{x})=\left(\sqrt{x_1^2+x_2^2+x_3^2},0,0,x_4,\cdots,x_n\right)^{\mathrm{T}}.$$

$$\vdots$$

$$\boldsymbol{G}_{1n}:\ c=\frac{\sqrt{x_1^2+\cdots+x_{n-1}^2}}{\sqrt{x_1^2+\cdots+x_n^2}},\quad s=\frac{x_n}{\sqrt{x_1^2+\cdots+x_n^2}},$$

$$\Longrightarrow\quad \boldsymbol{G}_{1n}(\boldsymbol{G}_{1,n-1}\cdots\boldsymbol{G}_{13}\boldsymbol{G}_{12}\boldsymbol{x})=\left(\sqrt{x_1^2+\cdots+x_n^2},0,\cdots,0\right)^{\mathrm{T}}.$$

令 $\boldsymbol{G}=\boldsymbol{G}_{1n}\boldsymbol{G}_{1,n-1}\cdots\boldsymbol{G}_{13}\boldsymbol{G}_{12}$, 则有 $\boldsymbol{G}\boldsymbol{x}=\|\boldsymbol{x}\|_2\boldsymbol{e}_1$.

(2) 设 $x_1=\cdots=x_{i-1}=0$, $x_i\neq 0$ $(1<i\leqslant n)$, 则由 $\boldsymbol{G}_{1i}$ 开始即可. 证毕. □

与 Householder 矩阵类似, 用 Givens 矩阵左乘或右乘一个已知矩阵 $\boldsymbol{A}$, 利用其特殊结构也是极为有利的. 假定 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$. 如果 $\boldsymbol{G}_{ik}(c,s)\in\mathbb{R}^{m\times m}$, 则用 $\boldsymbol{G}_{ik}(c,s)\boldsymbol{A}$ 修正 $\boldsymbol{A}$ 仅影响 $\boldsymbol{A}$ 的 i,k 两行, 可以用

$$\begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in}\\ a_{k1} & a_{k2} & \cdots & a_{kn}\end{bmatrix}:=\begin{bmatrix} c & s\\ -s & c\end{bmatrix}\begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in}\\ a_{k1} & a_{k2} & \cdots & a_{kn}\end{bmatrix}$$

实现, 这仅需 $6n$ 个 flop (浮点运算). 实现上述运算的 MATLAB 程序段如下:

```
for j=1:n
```

```
    t1=A(i,j); t2=A(k,j);
    A(i,j)=c*t1+s*t2;
    A(k,j)=-s*t1+c*t2;
end
```

同样, 如果 $\boldsymbol{G}_{ik}(c,s)\in\mathbb{R}^{n\times n}$, 则用 $\boldsymbol{A}\boldsymbol{G}_{ik}(c,s)$ 修正 $\boldsymbol{A}$ 仅影响 $\boldsymbol{A}$ 的 i,k 两列, 可以用

$$\begin{bmatrix} a_{1i} & a_{1k} \\ a_{2i} & a_{2k} \\ \vdots & \vdots \\ a_{mi} & a_{mk} \end{bmatrix} := \begin{bmatrix} a_{1i} & a_{1k} \\ a_{2i} & a_{2k} \\ \vdots & \vdots \\ a_{mi} & a_{mk} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

实现, 这只需 $6m$ 个 flop. 实现上述运算的 MATLAB 程序段如下:

```
for j=1:m
    t1=A(j,i); t2=A(j,k);
    A(j,i)=c*t1-s*t2;
    A(j,k)=s*t1+c*t2;
end
```

注 2.6 复数情形的 Givens 矩阵是指定义 2.2 中的 c 和 s 为复数且满足 $|c|^2+|s|^2=1$. 此时的 Givens 矩阵 $\boldsymbol{G}_{ik}(c,s)$ 是一个酉矩阵. 设 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in\mathbb{C}^n$, $\boldsymbol{y}=\boldsymbol{G}_{ik}(c,s)\boldsymbol{x}=(y_1,y_2,\cdots,y_n)^{\mathrm{T}}$. 容易验证, 当 $|x_i|^2+|x_k|^2\neq 0$ 时, 若取

$$c=\frac{|x_i|}{\sqrt{|x_i|^2+|x_k|^2}},\quad s=\frac{\overline{x}_k}{\sqrt{|x_i|^2+|x_k|^2}}\frac{x_i}{|x_i|}, \tag{2.7}$$

则有

$$y_s=x_s,\ s\neq i,k,\quad y_i=\frac{x_i}{|x_i|}\sqrt{|x_i|^2+|x_k|^2},\quad y_k=0, \tag{2.8}$$

即可以通过复 Givens 变换将复向量 $\boldsymbol{x}$ 的第 k 个分量化为零.

根据式 (2.7), 可设计算法如下.

算法 2.4 (计算复 Givens 变换) 给定两个复数 a 和 b, 本算法计算三个数 c,s,η, 使得

$$\begin{bmatrix} c & s \\ -\bar{s} & \bar{c} \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix}=\begin{bmatrix} \eta \\ 0 \end{bmatrix}.$$

function $[c,s,\eta]=$ **c_givens**(a,b)
if $b=0$
  $c=1;\ s=0;\ \eta=a;$ 结束
end
if $a=0$
  $c=0;\ s=1;\ \eta=b;$ 结束

end

$u=a/|a|;\ t=\sqrt{|a|^2+|b|^2};$

$c=|a|/t;\ s=u\bar{b}/t;\ \eta=ut;$

算法 2.4 的 MATLAB 程序如下:

```
%复Given变换程序-c_givens.m
function [c,s,eta]=c_givens(a,b)
%计算c,s满足[c,s;-s',c']*[a;b]=[eta;0]
if b==0, c=1; s=0; eta=a; end
if a==0, c=0; s=1; eta=b; end
u=a/abs(a); t=sqrt(abs(a)^2+abs(b)^2);
c=abs(a)/t; s=u*conj(b)/t; eta=u*t;
```

2.2 QR 分解

实现矩阵 QR 分解最常用的方法有三种, 分别是 Householder 正交化方法、Givens 正交化方法和 Gram–Schmidt 正交化方法. 本节先给出前两种方法的具体实现.

定义 2.3 设 $\boldsymbol{A}\in\mathbb{R}^{m\times n}\,(m\geqslant n)$ 为列满秩矩阵, 若有正交矩阵 $\boldsymbol{Q}\in\mathbb{R}^{m\times m}$ 与上三角矩阵 $\boldsymbol{R}\in\mathbb{R}^{m\times n}$ 使得 $\boldsymbol{A}=\boldsymbol{QR}$, 则称 $\boldsymbol{QR}$ 为 $\boldsymbol{A}$ 的 QR 分解.

2.2.1 Householder 变换 QR 分解

本节介绍如何使用 Householder 变换求矩阵的 QR 分解.

定理 2.6 设 $\boldsymbol{A}\in\mathbb{R}^{m\times n}\,(m\geqslant n)$ 为列满秩矩阵, 则存在有限个 Householder 矩阵的乘积 $\boldsymbol{Q}$, 使得

$$\boldsymbol{A}=\boldsymbol{Q}\widetilde{\boldsymbol{R}}=\boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix},\tag{2.9}$$

式中: $\boldsymbol{R}\in\mathbb{R}^{n\times n}$ 为非奇异的上三角矩阵.

证明 取 $\boldsymbol{A}_1:=\boldsymbol{A}$. 因为 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{m\times n}$ 列满秩, 故 $\boldsymbol{A}$ 的第 1 列 $\boldsymbol{a}_1\neq\boldsymbol{0}$. 于是可以构造 Householder 矩阵 $\boldsymbol{H}_1\in\mathbb{R}^{m\times m}$, 使得 $\boldsymbol{H}_1\boldsymbol{a}_1=\alpha_1\boldsymbol{e}_1$, 这里 $\alpha_1=\|\boldsymbol{a}_1\|_2$, $\boldsymbol{e}_1\in\mathbb{R}^m$. 则有

$$\boldsymbol{A}_2=\boldsymbol{H}_1\boldsymbol{A}_1=\begin{bmatrix}\alpha_1 & *\\ \boldsymbol{0} & \boldsymbol{A}_{22}^{(2)}\end{bmatrix}.$$

易见 $\boldsymbol{A}_{22}^{(2)}\in\mathbb{R}^{(m-1)\times(n-1)}$ 列满秩. 对 $\boldsymbol{A}_{22}^{(2)}$ 第 1 列的 $m-1$ 维非零向量 $\boldsymbol{a}_2$ 构造 $m-1$ 阶 Householder 矩阵 $\widetilde{\boldsymbol{H}}_2$, 使得 $\widetilde{\boldsymbol{H}}_2\boldsymbol{a}_2=\alpha_2\boldsymbol{e}_1$, 这里 $\alpha_2=\|\boldsymbol{a}_2\|_2$, $\boldsymbol{e}_1\in\mathbb{R}^{m-1}$. 令

$\boldsymbol{H}_2 = \mathrm{diag}(1, \widetilde{\boldsymbol{H}}_2)$, 则有

$$\boldsymbol{A}_3 = \boldsymbol{H}_2\boldsymbol{A}_2 = \begin{bmatrix} \alpha_1 & * & * & \cdots & * \\ 0 & \alpha_2 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11}^{(3)} & \boldsymbol{A}_{12}^{(3)} \\ \boldsymbol{O} & \boldsymbol{A}_{22}^{(3)} \end{bmatrix} \begin{matrix} 2 \\ m-2 \end{matrix}, \quad \begin{matrix} 2 & n-2 \end{matrix}$$

式中: $\boldsymbol{A}_{11}^{(3)}$ 为上三角矩阵.

重复上述过程, 假定已经进行了 $k-1$ 步, 得到了 Householder 变换 $\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_{k-1}$, 使得

$$\boldsymbol{A}_k = \boldsymbol{H}_{k-1}\cdots\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A}_1 = \begin{bmatrix} \boldsymbol{A}_{11}^{(k)} & \boldsymbol{A}_{12}^{(k)} \\ \boldsymbol{O} & \boldsymbol{A}_{22}^{(k)} \end{bmatrix} \begin{matrix} k-1 \\ m-k+1 \end{matrix}, \quad \begin{matrix} k-1 & n-k+1 \end{matrix}$$

式中: $\boldsymbol{A}_{11}^{(k)}$ 为上三角矩阵.

假定

$$\boldsymbol{A}_{22}^{(k)} = \left[\boldsymbol{a}_k, \boldsymbol{a}_{k+1}, \cdots, \boldsymbol{a}_n\right].$$

第 k 步是先对非零向量 $\boldsymbol{a}_k$, 构造

$$\widetilde{\boldsymbol{H}}_k = \boldsymbol{I}_{m-k+1} - \beta_k \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{T}} \in \mathbb{R}^{(m-k+1)\times(m-k+1)},$$

使得 $\widetilde{\boldsymbol{H}}_k\boldsymbol{a}_k = \alpha_k\boldsymbol{e}_1$, 这里 $\alpha_k = \|\boldsymbol{a}_k\|_2$, $\boldsymbol{e}_1 \in \mathbb{R}^{m-k+1}$. 令

$$\boldsymbol{H}_k = \mathrm{diag}(\boldsymbol{I}_{k-1}, \widetilde{\boldsymbol{H}}_k),$$

则

$$\boldsymbol{A}_{k+1} = \boldsymbol{H}_k\boldsymbol{A}_k = \begin{bmatrix} \boldsymbol{A}_{11}^{(k)} & \boldsymbol{A}_{12}^{(k)} \\ \boldsymbol{O} & \widetilde{\boldsymbol{H}}_k\boldsymbol{A}_{22}^{(k)} \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{A}_{11}^{(k+1)} & \boldsymbol{A}_{12}^{(k+1)} \\ \boldsymbol{O} & \boldsymbol{A}_{22}^{(k+1)} \end{bmatrix} \begin{matrix} k \\ m-k \end{matrix}, \quad \begin{matrix} k & n-k \end{matrix}$$

式中: $\boldsymbol{A}_{11}^{(k+1)}$ 为上三角矩阵.

这样, 从 $k=1$ 出发, 对 $\boldsymbol{A}$ 依次进行 n 次 Householder 变换, 就可将 $\boldsymbol{A}$ 约化为上三角矩阵 $\widetilde{\boldsymbol{R}}$, 即

$$\widetilde{\boldsymbol{R}} = \boldsymbol{A}_{n+1} = \boldsymbol{H}_n\boldsymbol{A}_n = \begin{bmatrix} \boldsymbol{A}_{11}^{(n+1)} \\ \boldsymbol{O} \end{bmatrix} \begin{matrix} n \\ m-n \end{matrix},$$

式中: $\boldsymbol{A}_{11}^{(n+1)}$ 为上三角矩阵.

现记

$$\boldsymbol{R}=\boldsymbol{A}_{11}^{(n+1)},\quad \boldsymbol{Q}^{\mathrm{T}}=\boldsymbol{H}_n\boldsymbol{H}_{n-1}\cdots\boldsymbol{H}_1,$$

则

$$\boldsymbol{A}=\boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ \\ \boldsymbol{O}\end{bmatrix},$$

式中: $\boldsymbol{R}\in\mathbb{R}^{n\times n}$ 为上三角矩阵.

若令 $\boldsymbol{Q}=[\boldsymbol{Q}_1,\boldsymbol{Q}_2]$, 这里 $\boldsymbol{Q}_1\in\mathbb{R}^{m\times n}$, 则有

$$\boldsymbol{A}=\boldsymbol{Q}_1\boldsymbol{R}.$$

证毕. □

注 2.7 若 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 为非奇异矩阵, 则由定理 5.4, $\boldsymbol{A}$ 有 QR 分解

$$\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R},$$

式中: $\boldsymbol{Q}\in\mathbb{R}^{n\times n}$ 为正交矩阵; $\boldsymbol{R}\in\mathbb{R}^{n\times n}$ 为非奇异的上三角矩阵.

下面考虑计算 $\boldsymbol{A}$ 的 QR 分解的存储问题. 一般来说, 在完成 QR 分解之后 $\boldsymbol{A}$ 就不再需要, 可用它来存放 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$. 通常不必将 $\boldsymbol{Q}$ 显式地算出, 而只存放构成它的 n 个 Householder 矩阵 $\boldsymbol{H}_k(k=1,2,\cdots,n)$, 而对每个 $\boldsymbol{H}_k$, 只需保存 $\boldsymbol{v}_k$ 和 β_k 即可. 注意到 $\boldsymbol{v}_k$ 具有如下形式:

$$\boldsymbol{v}_k=\left(1,v_{k+1}^{(k)},\cdots,v_n^{(k)}\right)^{\mathrm{T}}\in\mathbb{R}^{m-k+1},$$

正好可以将 $\boldsymbol{v}_k$ 的第 2 到 $m-k+1$ 个分量, 即 $\boldsymbol{v}_k(2:m-k+1)$ 存放在 $\boldsymbol{A}$ 的第 k 列对角元以下的位置上, 而 $\boldsymbol{A}$ 的上三角部分用来存放 $\boldsymbol{R}$ 的上三角部分.

综合上述讨论, 可得如下算法.

算法 2.5 (Householder 变换 QR 分解)

for $k=1:n$

  if $k<m$

    $[v,\beta]=\mathbf{r\_house}(\boldsymbol{A}(k:m,k))$;

    $\boldsymbol{A}(k:m,k:n)=(\boldsymbol{I}_{m-k+1}-\beta\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}})\boldsymbol{A}(k:m,k:n)$;

    $d(k)=\beta$;

    $\boldsymbol{A}(k+1:m,k)=\boldsymbol{v}(2:m-k+1)$;

  end

end

注 2.8 算法 2.5 是指对于给定的矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times n}(m\geqslant n)$, 计算 Householder 矩阵 $\boldsymbol{H}_1,\boldsymbol{H}_2,\cdots,\boldsymbol{H}_n$ 满足: 如果 $\boldsymbol{Q}=\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_n$, 则 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{R}$ 是上三角形矩阵, 且 $\boldsymbol{A}$ 的上三角部分被 $\boldsymbol{R}$ 的上三角部分所覆盖, 第 k 个 Householder 向量的第 $k+1$ 到第 m 个分量存放于 $\boldsymbol{A}(k+1:m,k)$, $k<m$ 的位置. 容易计算出, 该算法的运算量为 $2n^2(m-n/3)$ 个 flop.

根据上述算法可编制 MATLAB 程序如下:

```
function [A,d]=house_qr(A)
%Householder变换QR分解, 不显式计算和存储正交矩阵Q
[m,n]=size(A);
for k=1:n
    if k<m
        [v,beta]=r_house(A(k:m,k));
        I=eye(m-k+1);
        A(k:m,k:n)=(I-beta*v*v')*A(k:m,k:n);
        d(k)=beta;
        A(k+1:m,k)=v(2:m-k+1);
    end
end
```

例 2.3 利用算法 2.5 对矩阵 $\boldsymbol{A}$ 进行 QR 分解, 其中

$$\boldsymbol{A}=\begin{bmatrix} 76 & 96 & 85 & 35 \\ 26 & 55 & 26 & 20 \\ 51 & 14 & 82 & 26 \\ 70 & 15 & 25 & 62 \\ 90 & 26 & 93 & 48 \end{bmatrix}.$$

解 由于算法 2.5 中没有显式计算和存储正交矩阵 $\boldsymbol{Q}$, 故需对程序 house_qr.m 稍作修改:

```
function [Q,R]=Qhouse_qr(A)
%Householder QR分解, 显式计算并存储正交矩阵Q
[m,n]=size(A); Q=eye(m);
for k=1:n
    if k<m
        [v,beta]=r_house(A(k:m,k));
        H=eye(m-k+1)-beta*v*v';
        A(k:m,k:n)=H*A(k:m,k:n);
        Q=Q*blkdiag(eye(k-1),H);
    end
end
R=triu(A);
```

然后再在命令窗口依次运行如下代码:

```
>> A=[76,96,85,35;26,55,26,20; ...
>>    51,14,82,26;70,15,25,62;90,26,93,48];
>> [Q,R]=Qhouse_qr(A)
```

即得矩阵 $\boldsymbol{A}$ 的 QR 分解, 且满足 $\|\boldsymbol{A}-\boldsymbol{QR}\|_2 = 4.3169\times 10^{-14}$.

2.2.2 Givens 变换 QR 分解

Givens 变换也可以实现矩阵 $\boldsymbol{A}$ 的 QR 分解. 先考虑 $\boldsymbol{A}$ 为非奇异实方阵的情形, 有下面的定理.

定理 2.7 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 为非奇异矩阵, 则存在有限个 Givens 矩阵的乘积 $\boldsymbol{Q}$, 使得 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}$ 为可逆上三角矩阵 $\boldsymbol{R}$, 即 $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$.

证明 因为 $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$ 可逆, 故 $\boldsymbol{A}$ 的第 1 列 $\boldsymbol{a}^{(0)} \neq \boldsymbol{0}$. 于是可以构造有限个 Givens 矩阵的乘积 $\boldsymbol{G}_0 \in \mathbb{R}^{n\times n}$, 使得 $\boldsymbol{G}_0\boldsymbol{a}^{(0)} = \|\boldsymbol{a}^{(0)}\|_2\boldsymbol{e}_1$, 这里 $\boldsymbol{e}_1 \in \mathbb{R}^n$. 记 $a_{11}^{(1)} = \|\boldsymbol{a}^{(0)}\|_2$, 则有

$$\boldsymbol{G}_0\boldsymbol{A} = \left[\begin{array}{c|ccc} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ \hline 0 & & & \\ \vdots & & \boldsymbol{A}^{(1)} & \\ 0 & & & \end{array}\right].$$

易见 $\boldsymbol{A}^{(1)}$ 可逆. 于是 $\boldsymbol{A}^{(1)}$ 的第 1 列 $\boldsymbol{a}^{(1)} \neq \boldsymbol{0}$. 故存在有限个 Givens 矩阵的乘积 $\boldsymbol{G}_1 \in \mathbb{R}^{(n-1)\times(n-1)}$, 使得 $\boldsymbol{G}_1\boldsymbol{a}^{(1)} = \|\boldsymbol{a}^{(1)}\|_2\boldsymbol{e}_1$, 这里 $\boldsymbol{e}_1 \in \mathbb{R}^{n-1}$. 记 $a_{22}^{(2)} = \|\boldsymbol{a}^{(1)}\|_2$, 则有

$$\boldsymbol{G}_1\boldsymbol{A}^{(1)} = \left[\begin{array}{c|ccc} a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ \hline 0 & & & \\ \vdots & & \boldsymbol{A}^{(2)} & \\ 0 & & & \end{array}\right].$$

依此类推, 最后得到 $\boldsymbol{A}^{(n-2)} \in \mathbb{R}^{2\times 2}$ 可逆, $\boldsymbol{A}^{(n-2)}$ 的第 1 列 $\boldsymbol{a}^{(n-2)} \neq \boldsymbol{0}$. 那么可构造 Givens 矩阵 $\boldsymbol{G}_{n-2} \in \mathbb{R}^{2\times 2}$, 使得 $\boldsymbol{G}_{n-2}\boldsymbol{a}^{(n-2)} = \|\boldsymbol{a}^{(n-2)}\|_2\boldsymbol{e}_1$, 这里 $\boldsymbol{e}_1 \in \mathbb{R}^2$. 记 $a_{n-1,n-1}^{(n-1)} = \|\boldsymbol{a}^{(n-2)}\|_2$, 则有

$$\boldsymbol{G}_{n-2}\boldsymbol{A}^{(n-2)} = \begin{bmatrix} a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ 0 & a_{n,n}^{(n-1)} \end{bmatrix}.$$

令

$$\boldsymbol{Q}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{I}_{n-2} & \\ & \boldsymbol{G}_{n-2} \end{bmatrix} \cdots \begin{bmatrix} \boldsymbol{I}_2 & \\ & \boldsymbol{G}_2 \end{bmatrix} \begin{bmatrix} 1 & \\ & \boldsymbol{G}_1 \end{bmatrix} \boldsymbol{G}_0,$$

则 $\boldsymbol{Q}$ 仍为有限个 Givens 矩阵之积, 且有

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,n-1}^{(1)} & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2,n-1}^{(2)} & a_{2n}^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & & a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ & & & & a_{nn}^{(n-1)} \end{bmatrix} := \boldsymbol{R},$$

即 $\boldsymbol{A}=\boldsymbol{QR}$. 证毕. □

定理 2.7 的结论可以推广到 $\boldsymbol{A}$ 是长方阵的情形.

定理 2.8 设 $\boldsymbol{A}\in\mathbb{R}^{m\times n}\,(m\geqslant n)$ 是列满秩的, 则有 m 阶正交矩阵 $\boldsymbol{Q}$ 及 n 阶上三角矩阵 $\boldsymbol{R}$, 使得

$$\boldsymbol{A}=\boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix}=\boldsymbol{Q}_1\boldsymbol{R},$$

式中: $\boldsymbol{Q}_1$ 为由 $\boldsymbol{Q}$ 的前 n 列构成的矩阵.

证明 已知 $\boldsymbol{A}$ 是列满秩的, 即它的 n 个列向量线性无关, 使用与证明定理 2.7 相同的方法, 可找到有限个 m 阶 Givens 矩阵的乘积 $\boldsymbol{G}$, 使得 $\boldsymbol{GA}=\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix}$, 再令 $\boldsymbol{Q}=\boldsymbol{G}^{\mathrm{T}}$ 即得所求. 证毕. □

下面考虑 Givens 变换 QR 分解的实现过程. 可以用一个 5×3 阶矩阵的例子来表明其一般思想.

$$\begin{bmatrix}\times&\times&\times\\ \times&\times&\times\\ \times&\times&\times\\ \times&\times&\times\\ \times&\times&\times\end{bmatrix}\xrightarrow{(4,5)}\begin{bmatrix}\times&\times&\times\\ \times&\times&\times\\ \times&\times&\times\\ \times&\times&\times\\ 0&\times&\times\end{bmatrix}\xrightarrow{(3,4)}\begin{bmatrix}\times&\times&\times\\ \times&\times&\times\\ \times&\times&\times\\ 0&\times&\times\\ 0&\times&\times\end{bmatrix}\xrightarrow{(2,3)}\begin{bmatrix}\times&\times&\times\\ \times&\times&\times\\ 0&\times&\times\\ 0&\times&\times\\ 0&\times&\times\end{bmatrix}\xrightarrow{(1,2)}$$

$$\begin{bmatrix}\times&\times&\times\\ 0&\times&\times\\ 0&\times&\times\\ 0&\times&\times\\ 0&\times&\times\end{bmatrix}\xrightarrow{(4,5)}\begin{bmatrix}\times&\times&\times\\ 0&\times&\times\\ 0&\times&\times\\ 0&\times&\times\\ 0&0&\times\end{bmatrix}\xrightarrow{(3,4)}\begin{bmatrix}\times&\times&\times\\ 0&\times&\times\\ 0&\times&\times\\ 0&0&\times\\ 0&0&\times\end{bmatrix}\xrightarrow{(2,3)}\begin{bmatrix}\times&\times&\times\\ 0&\times&\times\\ 0&0&\times\\ 0&0&\times\\ 0&0&\times\end{bmatrix}\xrightarrow{(4,5)}$$

$$\begin{bmatrix}\times&\times&\times\\ 0&\times&\times\\ 0&0&\times\\ 0&0&\times\\ 0&0&0\end{bmatrix}\xrightarrow{(3,4)}\begin{bmatrix}\times&\times&\times\\ 0&\times&\times\\ 0&0&\times\\ 0&0.&0\\ 0&0&0\end{bmatrix}:=\begin{bmatrix}\boldsymbol{R}_1\\ \boldsymbol{O}\end{bmatrix}:=\boldsymbol{R},$$

此处已标记定义所对应 Givens 变换的 2 维向量. 显然, 若 $\boldsymbol{G}_i$ 表示在约化过程中的第 i 次 Givens 变换, 则 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{R}$ 是上三角形矩阵, 其中 $\boldsymbol{Q}=\boldsymbol{G}_1\boldsymbol{G}_2\cdots\boldsymbol{G}_t$, 且 t 是 Givens 变换的总次数. 对于一般的 $\boldsymbol{A}\in\mathbb{R}^{m\times n}\,(m\geqslant n)$, 有如下算法.

算法 2.6 (Givens 变换 QR 分解)

$[m,n]=\mathbf{size}(\boldsymbol{A})$;

for $k=1:n$

  for $i=m:-1:k+1$

$$[c,s]=\mathbf{r\_givens}(\boldsymbol{A}(i-1,k),\boldsymbol{A}(i,k));$$

$$\boldsymbol{A}(i-1:i,k:n)=\begin{bmatrix} c & s \\ -s & c \end{bmatrix}\boldsymbol{A}(i-1:i,k:n);$$

end

end

注 2.9 算法 2.6 是指对于给定的矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times n}(m\geqslant n)$, 计算 Givens 矩阵 $\boldsymbol{G}_1,\boldsymbol{G}_2,\cdots,\boldsymbol{G}_t\,(t=\frac{1}{2}n(2m-n-1))$ 满足: 如果 $\boldsymbol{Q}=\boldsymbol{G}_1\boldsymbol{G}_2\cdots\boldsymbol{G}_t$, 则 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{R}$ 是上三角形矩阵, 且 $\boldsymbol{A}$ 被 $\boldsymbol{R}$ 所覆盖. 容易计算出, 该算法的运算量为 $3n^2(m-n/3)$ 个 flop.

根据上述算法可编制 MATLAB 程序如下:

```
function [A]=givens_qr(A)
%Givens变换QR分解
[m,n]=size(A);
for k=1:n
    for i=m:-1:k+1
        [c,s]=r_givens(A(i-1,k), A(i,k));
        A(i-1:i, k:n)=[c, s; -s, c]*A(i-1:i, k:n);
    end
end
```

例 2.4 利用算法 2.6 对例 2.3 中的矩阵 $\boldsymbol{A}$ 进行 QR 分解.

解 由于算法 2.6 中没有显式计算和存储正交矩阵 $\boldsymbol{Q}$, 故不能直接调用程序 givens_qr.m, 需对其稍作修改:

```
function [Q,R]=Qgivens_qr(A)
%Givens变换QR分解, 显式计算并存储正交矩阵Q
[m,n]=size(A); Q=eye(m);
for k=1:n
    for i=m:-1:k+1
        [c,s]=r_givens(A(i-1,k),A(i,k));
        A(i-1:i,k:n)=[c,s;-s,c]*A(i-1:i,k:n);
        Q(:,i-1:i)=Q(:,i-1:i)*[c,s;-s,c]';
    end
end
R=triu(A);
```

然后再在命令窗口依次运行如下代码:

```
>> A=[76,96,85,35;26,55,26,20; ...
>>    51,14,82,26;70,15,25,62;90,26,93,48];
>> [Q,R]=Qgivens_qr(A)
```

即得矩阵 $\boldsymbol{A}$ 的 QR 分解, 且满足 $\|\boldsymbol{A}-\boldsymbol{QR}\|_2 = 7.6458 \times 10^{-14}$.

作为 Givens 变换 QR 分解的一个应用, 下面讨论上 Hessenberg 矩阵的 QR 分解问题. 从前述讨论可知, Givens 变换 QR 分解的运算量约为 Householder 变换的 1.5 倍. 尽管如此, 用 Givens 变换对上 Hessenberg 矩阵进行 QR 分解却具有独特的优势. 上 Hessenberg 矩阵是一类具有重要应用的矩阵类, 例如第 4 章即将介绍的广义极小残量法 (GMRES) 需要求解一个上 Hessenberg 矩阵的最小二乘问题. 下面介绍上 Hessenberg 矩阵的 Givens 变换 QR 分解的具体实现.

定义 2.4 *如果 n 阶矩阵的下次对角线以下的元素都为零, 则称该矩阵为上 Hessenberg 矩阵. 上 Hessenberg 矩阵的转置称为下 Hessenberg 矩阵.*

根据上 Hessenberg 矩阵的特殊结构 (矩阵的每一列对角线以下的元素中只有第 1 个元素可能非零), 因而只需对每一列作一次 Givens 变换即可, 具体来说有下面的算法.

算法 2.7 (上 Hessenberg 矩阵 QR 分解)

for $k = 1 : n-1$

$[c_k, s_k] = \mathbf{r\_givens}(\boldsymbol{A}(k,k), \boldsymbol{A}(k+1,k))$;

$$\boldsymbol{A}(k:k+1, k:n) = \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix} \boldsymbol{A}(k:k+1, k:n);$$

end

注 2.10 *算法 2.7 是指对于给定的上 Hessenberg 矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 计算 Givens 矩阵 $\boldsymbol{G}_1, \boldsymbol{G}_2, \cdots, \boldsymbol{G}_{n-1}$ 满足: 如果 $\boldsymbol{Q} = \boldsymbol{G}_1\boldsymbol{G}_2\cdots\boldsymbol{G}_{n-1}$, 则 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{R}$ 是上三角形矩阵, 且 $\boldsymbol{A}$ 被 $\boldsymbol{R}$ 所覆盖, 其中 $\boldsymbol{G}_k$ 形如 $\boldsymbol{G}_{k,k+1}(c_k, s_k)$. 容易计算出, 该算法的运算量只有 $3n^2$ 个 flop.*

根据上述算法可编制 MATLAB 程序如下:

```
function [A]=hessenberg_qr(A)
%上Hessenberg矩阵的QR分解
[n]=size(A,2);
for k=1:n-1
    [c,s]=r_givens(A(k,k),A(k+1,k));
    A(k:k+1,k:n)=[c,s;-s,c]*A(k:k+1,k:n);
end
```

作为 Givens 变换 (或 Householder 变换) 的另一个应用, 下面介绍 n 阶实矩阵正交相似于上 Hessenberg 矩阵的计算问题.

定理 2.9 *设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 则存在有限个 Givens 矩阵 (或 Householder 矩阵) 的乘积 $\boldsymbol{Q}$, 使得 $\boldsymbol{QAQ}^{\mathrm{T}}$ 为上 Hessenberg 矩阵.*

证明 仅讨论使用 Givens 矩阵的情形.

第 1 步, 设 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$, 记 $\boldsymbol{\alpha}^{(0)}=(a_{21},\cdots,a_{n1})^{\mathrm{T}}\in\mathbb{R}^{n-1}$, 当 $\boldsymbol{\alpha}^{(0)}=\boldsymbol{0}$ 时转入第 2 步; 当 $\boldsymbol{\alpha}^{(0)}\neq\boldsymbol{0}$ 时, 构造有限个 Givens 矩阵的乘积 $\boldsymbol{G}_0$, 使得

$$\boldsymbol{G}_0\boldsymbol{\alpha}^{(0)}=\|\boldsymbol{\alpha}^{(0)}\|_2\boldsymbol{e}_1,\quad(\boldsymbol{e}_1\in\mathbb{R}^{n-1}).$$

记 $a_{21}^{(1)}=\|\boldsymbol{\alpha}^{(0)}\|_2$, 则有

$$\begin{bmatrix}1&\\&\boldsymbol{G}_0\end{bmatrix}\boldsymbol{A}\begin{bmatrix}1&\\&\boldsymbol{G}_0\end{bmatrix}^{\mathrm{T}}=\left[\begin{array}{c|ccc}a_{11}&a_{12}^{(1)}&a_{13}^{(1)}&\cdots&a_{1n}^{(1)}\\\hline a_{21}^{(1)}&&&&\\0&&&&\\\vdots&&\multicolumn{3}{c}{\boldsymbol{A}^{(1)}}\\0&&&&\end{array}\right].$$

第 2 步, 设 $\boldsymbol{A}^{(1)}\in\mathbb{R}^{(n-1)\times(n-1)}$, 记 $\boldsymbol{\alpha}^{(1)}=(a_{32}^{(1)},\cdots,a_{n2}^{(1)})^{\mathrm{T}}\in\mathbb{R}^{n-2}$, 当 $\boldsymbol{\alpha}^{(1)}=\boldsymbol{0}$ 时转入第 3 步; 当 $\boldsymbol{\alpha}^{(1)}\neq\boldsymbol{0}$ 时, 构造有限个 Givens 矩阵的乘积 $\boldsymbol{G}_1$, 使得

$$\boldsymbol{G}_1\boldsymbol{\alpha}^{(1)}=\|\boldsymbol{\alpha}^{(1)}\|_2\boldsymbol{e}_1,\quad(\boldsymbol{e}_1\in\mathbb{R}^{n-2}).$$

记 $a_{32}^{(2)}=\|\boldsymbol{\alpha}^{(1)}\|_2$, 则有

$$\begin{bmatrix}1&\\&\boldsymbol{G}_1\end{bmatrix}\boldsymbol{A}^{(1)}\begin{bmatrix}1&\\&\boldsymbol{G}_1\end{bmatrix}^{\mathrm{T}}=\left[\begin{array}{c|ccc}a_{22}^{(1)}&a_{23}^{(2)}&a_{24}^{(2)}&\cdots&a_{2n}^{(2)}\\\hline a_{32}^{(2)}&&&&\\0&&&&\\\vdots&&\multicolumn{3}{c}{\boldsymbol{A}^{(2)}}\\0&&&&\end{array}\right].$$

继续上述过程, 直到第 $n-2$ 步:

$$\boldsymbol{A}^{(n-3)}=\begin{bmatrix}a_{n-2,n-2}^{(n-3)}&a_{n-2,n-1}^{(n-3)}&a_{n-2,n}^{(n-3)}\\a_{n-1,n-2}^{(n-3)}&a_{n-1,n-1}^{(n-3)}&a_{n-1,n}^{(n-3)}\\a_{n,n-2}^{(n-3)}&a_{n,n-1}^{(n-3)}&a_{n,n}^{(n-3)}\end{bmatrix}\in\mathbb{R}^{3\times3},$$

记 $\boldsymbol{\alpha}^{(n-3)}=\begin{bmatrix}a_{n-1,n-2}^{(n-3)}\\a_{n,n-2}^{(n-3)}\end{bmatrix}$. 当 $\boldsymbol{\alpha}^{(n-3)}=\boldsymbol{0}$ 时结束; 当 $\boldsymbol{\alpha}^{(n-3)}\neq\boldsymbol{0}$ 时, 构造 Givens 矩阵 $\boldsymbol{G}_{n-3}$, 使得

$$\boldsymbol{G}_{n-3}\boldsymbol{\alpha}^{(n-3)}=\|\boldsymbol{\alpha}^{(n-3)}\|_2\boldsymbol{e}_1,\quad(\boldsymbol{e}_1\in\mathbb{R}^2).$$

记 $a_{n-1,n-2}^{(n-2)} = \|\boldsymbol{\alpha}^{(n-3)}\|_2$, 则有

$$\begin{bmatrix} 1 & \\ & \boldsymbol{G}_{n-3} \end{bmatrix} \boldsymbol{A}^{(n-3)} \begin{bmatrix} 1 & \\ & \boldsymbol{G}_{n-3} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} a_{n-2,n-2}^{(n-3)} & a_{n-2,n-1}^{(n-3)} & a_{n-2,n}^{(n-3)} \\ a_{n-1,n-2}^{(n-2)} & a_{n-1,n-1}^{(n-2)} & a_{n-1,n}^{(n-2)} \\ 0 & a_{n,n-1}^{(n-2)} & a_{n,n}^{(n-2)} \end{bmatrix}.$$

最后, 构造正交矩阵

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{I}_{n-2} & \\ & \boldsymbol{G}_{n-3} \end{bmatrix} \cdots \begin{bmatrix} \boldsymbol{I}_2 & \\ & \boldsymbol{G}_1 \end{bmatrix} \begin{bmatrix} 1 & \\ & \boldsymbol{G}_0 \end{bmatrix},$$

可使 $\boldsymbol{QAQ}^{\mathrm{T}}$ 为上 Hessenberg 矩阵. 证毕. □

推论 2.1 设 $\boldsymbol{A}$ 是 n 阶实对称矩阵, 则存在有限个 Givens 矩阵 (或 Householder 矩阵) 的乘积 $\boldsymbol{Q}$, 使得 $\boldsymbol{QAQ}^{\mathrm{T}}$ 为实对称三对角矩阵.

证明 对 $\boldsymbol{A}$ 应用定理 2.9, 则存在正交矩阵 $\boldsymbol{Q}$, 使得 $\boldsymbol{QAQ}^{\mathrm{T}}$ 为上 Hessenberg 矩阵. 由于 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, 又有 $\boldsymbol{QAQ}^{\mathrm{T}} = (\boldsymbol{QAQ}^{\mathrm{T}})^{\mathrm{T}}$ 为下 Hessenberg 矩阵, 故 $\boldsymbol{QAQ}^{\mathrm{T}}$ 是三对角矩阵, 从而 $\boldsymbol{QAQ}^{\mathrm{T}}$ 是实对称的三对角矩阵. 证毕. □

2.3 线性无关向量组的正交化

本节讨论线性无关向量组的正交化. 对以这些向量为列的矩阵施以 Householder 变换或 Givens 变换作正交化, 就可以得到该线性无关向量组的正交化. 首先介绍经典的 Gram–Schmidt 正交化过程.

2.3.1 Gram–Schmidt 正交化

考虑 m 个线性无关的 n 维向量组 $\{\boldsymbol{x}_i\}_{i=1}^m (m \leqslant n)$. 记列满秩 $n \times m$ 阶矩阵 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]$. 对向量组 $\{\boldsymbol{x}_i\}_{i=1}^m$ 用 Gram–Schmidt 正交化产生相互正交的单位向量组 $\{\boldsymbol{q}_i\}_{i=1}^m$, 使得

$$\mathrm{span}\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k\} = \mathrm{span}\{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_k\},\ k = 1, 2, \cdots, m. \tag{2.10}$$

按如下方式构造 $\{\boldsymbol{q}_i\}_{i=1}^m$. 令 $\boldsymbol{q}_1 = \boldsymbol{x}_1/\|\boldsymbol{x}_1\|_2$. 设已经构造好相互正交的单位向量组 $\{\boldsymbol{q}_i\}_{i=1}^{k-1}$ 使得式 (2.10) 对 $k-1$ 成立. 利用正交性, 得

$$\widetilde{\boldsymbol{q}}_k = \boldsymbol{x}_k - \sum_{i=1}^{k-1} r_{ik}\boldsymbol{q}_i,\ r_{is} = (\boldsymbol{x}_k, \boldsymbol{q}_i),\ i = 1, 2, \cdots, k-1;\ \boldsymbol{q}_k = \frac{\widetilde{\boldsymbol{q}}_k}{\|\widetilde{\boldsymbol{q}}_k\|_2}. \tag{2.11}$$

从而 $\boldsymbol{q}_k$ 与 $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{k-1}$ 都正交且为单位向量. 在 $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{k-1}$ 都已经单位正交化的前提下, 式 (2.11) 表明 $\boldsymbol{x}_k$ 去掉它在 $\boldsymbol{q}_i$ 上的分量 (投影) $r_{ik}\boldsymbol{q}_i\,(i = 1, 2, \cdots, k-1)$ 后剩下的部分与 $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{k-1}$ 都正交. 据此, 可以写出 Gram–Schmidt 正交化的算法如下:

算法 2.8 (标准 Gram–Schmidt 正交化) 给定 $n \times m (m \leqslant n)$ 阶列满秩矩阵 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]$, 本算法产生 $n \times m$ 阶矩阵 $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_m]$(其列是单位正交的) 和 $m \times m$ 阶非奇异上三角矩阵 $\boldsymbol{R} = (r_{ik})$.

$r_{11} = \|\boldsymbol{x}_1\|_2$; $\boldsymbol{q}_1 = \boldsymbol{x}_1 / r_{11}$;

for $k = 2 : m$

for $(i = 1 : k-1)$, $r_{ik} = (\boldsymbol{x}_k, \boldsymbol{q}_i)$; **end**

$$\widetilde{\boldsymbol{q}} = \boldsymbol{x}_k - \sum_{i=1}^{k-1} r_{ik} \boldsymbol{q}_i;$$

$r_{kk} = \|\widetilde{\boldsymbol{q}}\|_2$; $\boldsymbol{q}_k = \widetilde{\boldsymbol{q}} / r_{kk}$;

end

由算法 2.8 可以得到关系式

$$\boldsymbol{x}_k = \sum_{i=1}^{k} r_{ik} \boldsymbol{q}_i, \quad k = 1, 2, \cdots, m. \tag{2.12}$$

写成矩阵形式即为

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R}, \tag{2.13}$$

式中: $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_m] \in \mathbb{R}^{n \times m}$, 其列是相互正交的单位向量; $\boldsymbol{R}$ 为 $m \times m$ 阶上三角矩阵, $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$ 的线性无关性保证了所有的 $r_{ii} (i = 1, 2, \cdots, m)$ 都不为零, 因而 $\boldsymbol{R}$ 是非奇异的.

算法 2.8 表明, 通过对矩阵 $\boldsymbol{X}$ 列向量的 Gram–Schmidt 正交化, 实现了矩阵 $\boldsymbol{X}$ 的 QR 分解.

标准 Gram–Schmidt 正交化过程的 MATLAB 程序如下:

```
%标准Gram-Schmidt正交化程序-G_Schmidt1.m
function [Q,R]=G_Schmidt1(X)
[m]=size(X,2); %矩阵的列
R=zeros(m); R(1,1)=norm(X(:,1));
Q(:,1)=X(:,1)/R(1,1); %单位化
for k=2:m
    for (i=1:k-1), R(i,k)=Q(:,i)'*X(:,k); end
    qt=X(:,k);
    for (i=1:k-1), qt=qt-R(i,k)*Q(:,i); end
    R(k,k)=norm(qt); Q(:,k)=qt/R(k,k); %单位化
end
```

从算法 2.8 可以看出, 当 $\boldsymbol{x}_k$ 与前面的 $\boldsymbol{x}_i$ 接近线性相关时, r_{kk} 将接近于 0, 用它作分母会导致很大的舍入误差, $\boldsymbol{q}_i$ 之间很快失去了正交性. 一种改进就是修正的 Gram–Schmidt 正交化. 它把算法 2.8 的第 3 行和第 4 行修改为下面的循环:

$$\widetilde{\boldsymbol{q}} = \boldsymbol{x}_k, \quad r_{ik} = (\widetilde{\boldsymbol{q}}, \boldsymbol{q}_i), \quad \widetilde{\boldsymbol{q}} := \widetilde{\boldsymbol{q}} - r_{ik} \boldsymbol{q}_i, \quad i = 1, 2, \cdots, k-1. \tag{2.14}$$

在不考虑舍入误差时, 上述计算结果 $\widetilde{\boldsymbol{q}}$ 与算法 2.8 中第 4 行的 $\widetilde{\boldsymbol{q}}$ 是相同的. 实际上, 修正的 Gram–Schmidt 正交化利用了最新的 $\widetilde{\boldsymbol{q}}$, 有助于减少舍入误差影响.

为了保证正交性, 更可靠的方法是进行重正交化. 设已用式 (2.14) 计算了 $\widetilde{\boldsymbol{q}}$. 若 $\|\widetilde{\boldsymbol{q}}\|_2$ 很小 (与 $\|\boldsymbol{x}_k\|_2$ 相比), 用它作分母计算 $\boldsymbol{q}_k$ 会有很大的舍入误差. 此时, 对已计算的 $\widetilde{\boldsymbol{q}}$ 再正交化, 得

$$\widetilde{r}_{ik}=(\widetilde{\boldsymbol{q}},\boldsymbol{q}_i),\quad \widetilde{\boldsymbol{q}}:=\widetilde{\boldsymbol{q}}-\widetilde{r}_{ik}\boldsymbol{q}_i,\quad i=1,2,\cdots,k-1. \tag{2.15}$$

由于 $\widetilde{\boldsymbol{q}}$ 已与 $\boldsymbol{q}_i$ 接近正交, 故 $\widetilde{r}_{ik}$ 接近 0. 把它累加到 r_{ik} 上, 得

$$r_{ik}:=r_{ik}+\widetilde{r}_{ik},\quad i=1,2,\cdots,k-1. \tag{2.16}$$

可以写出如下的修正 Gram-Schmidt 正交化算法.

算法 2.9 (修正 Gram–Schmidt 正交化) 给定 $n\times m(m\leqslant n)$ 阶列满秩矩阵 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m]$, 本算法产生 $n\times m$ 阶矩阵 $\boldsymbol{Q}=[\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_m]$(其列是单位正交的) 和 $m\times m$ 阶非奇异上三角矩阵 $\boldsymbol{R}=(r_{ik})$.

$r_{11}=\|\boldsymbol{x}_1\|_2$; $\boldsymbol{q}_1=\boldsymbol{x}_1/r_{11}$;
for $k=2:m$
  $\widetilde{\boldsymbol{q}}=\boldsymbol{x}_k$;
  for $(i=1:k-1)$
    $r_{ik}=(\widetilde{\boldsymbol{q}},\boldsymbol{q}_i)$; $\widetilde{\boldsymbol{q}}=\widetilde{\boldsymbol{q}}-r_{ik}\boldsymbol{q}_i$;
  end
  for $(i=1:k-1)$ %重正交化
    $\widetilde{r}_{ik}=(\widetilde{\boldsymbol{q}},\boldsymbol{q}_i)$; $\widetilde{\boldsymbol{q}}=\widetilde{\boldsymbol{q}}-\widetilde{r}_{ik}\boldsymbol{q}_i$;
    $r_{ik}=r_{ik}+\widetilde{r}_{ik}$;
  end
  $r_{kk}=\|\widetilde{\boldsymbol{q}}\|_2$; $\boldsymbol{q}_k=\widetilde{\boldsymbol{q}}/r_{kk}$;
end

再给出修正 Gram–Schmidt 正交化过程的 MATLAB 程序如下:

```
%修正Gram-Schmidt正交化程序-G_Schmidt2.m
function [Q,R]=G_Schmidt2(X)
[n,m]=size(X); %矩阵的列
R=zeros(m); Q=zeros(n,m);
R(1,1)=norm(X(:,1));
Q(:,1)=X(:,1)/R(1,1); %单位化
for k=2:m
    qt=X(:,k);
    for i=1:k-1
```

```
        R(i,k)=qt'*Q(:,i);
        qt=qt-R(i,k)*Q(:,i); %对剩余向量进行修正
    end
    for i=1:k-1  %重正交化
        rt=qt'*Q(:,i);
        qt=qt-rt*Q(:,i);
        R(i,k)=R(i,k)+rt;
    end
    R(k,k)=norm(qt);
    Q(:,k)=qt/R(k,k); %对本次得到的向量单位化
end
```

例 2.5 用 Gram–Schmidt 方法对矩阵 $\boldsymbol{A}$ 的列向量进行正交化, 其中

$$\boldsymbol{A}=\begin{bmatrix} 9 & 1 & 4 & 5 \\ 6 & 2 & 1 & 3 \\ 4 & 1 & 9 & 9 \\ 5 & 2 & 9 & 4 \\ 4 & 2 & 5 & 1 \end{bmatrix}.$$

解 分别用标准 Gram–Schmidt 方法和修正 Gram–Schmidt 方法对矩阵 $\boldsymbol{A}$ 列向量进行正交化. 在命令窗口输入:

```
>> A=[9,1,4,5;6,2,1,3;4,1,9,9;5,2,9,4;4,2,5,1];
>> [Q1,R1]=G_Schmidt1(A);err1=norm(A-Q1*R1)
>> [Q2,R2]=G_Schmidt2(A);err2=norm(A-Q2*R2)
```

运行后即得结果, 且分别满足 $\|\boldsymbol{A}-\boldsymbol{Q}_1\boldsymbol{R}_1\|_2 = 1.0175\times 10^{-15}$ 和 $\|\boldsymbol{A}-\boldsymbol{Q}_2\boldsymbol{R}_2\|_2 = 2.2204\times 10^{-16}$.

2.3.2 Householder 正交化

2.2 节中介绍了 Householder 变换可以对矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times n}\,(m\geqslant n)$ 进行 QR 分解. 这一过程也可以对任意 m 个线性无关的 n 维向量组进行. 与 Gram–Schmidt 正交化相比, 它具有更好的数值稳定性. 给定列满秩矩阵 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m]$, 这里假设 $m\leqslant n$. 令 $\boldsymbol{X}_1=\boldsymbol{X}$, 构造 n 阶 Householder 矩阵 $\boldsymbol{H}_1$ 使得 $\boldsymbol{X}_2=\boldsymbol{H}_1\boldsymbol{X}_1$ 的第 1 列的后 $n-1$ 个分量全部为 0. 一般地, 假设构造了 $k-1$ 个 Householder 矩阵 $\{\boldsymbol{H}_i\}_{i=1}^{k-1}$, 使得

$$\boldsymbol{X}_k=\boldsymbol{H}_{k-1}\boldsymbol{H}_{k-2}\cdots\boldsymbol{H}_1\boldsymbol{X}_1=\begin{bmatrix} * & * & * & * & \cdots & * \\ & \ddots & * & * & \cdots & * \\ & & * & * & \cdots & * \\ & & & x_{kk} & \cdots & x_{km} \\ & & & \vdots & \ddots & \vdots \\ & & & x_{nk} & \cdots & x_{nm} \end{bmatrix}. \tag{2.17}$$

由于 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m$ 线性无关, 故 $\boldsymbol{X}_k$ 的前 k 列线性无关. 故 $\widetilde{\boldsymbol{x}}_k=(x_{kk},x_{k+1,k},\cdots,x_{nk})^{\mathrm{T}}$ 不是零向量. 构造 $n-k+1$ 阶 Householder 矩阵 $\widetilde{\boldsymbol{H}}_k=\boldsymbol{I}_{n-k+1}-\beta_k\boldsymbol{v}_k\boldsymbol{v}_k^{\mathrm{T}}$, 使得 $\widetilde{\boldsymbol{H}}_k\widetilde{\boldsymbol{x}}_k$ 的后 $n-k$ 个分量为 0. 把 n 阶 Householder 矩阵 $\boldsymbol{H}_k=\mathrm{diag}(\boldsymbol{I}_{k-1},\widetilde{\boldsymbol{H}}_k)$ 作用到 $\boldsymbol{X}_k$ 后得到

$$\boldsymbol{X}_{k+1}=\boldsymbol{H}_k\boldsymbol{X}_k=\boldsymbol{H}_k\boldsymbol{H}_{k-1}\cdots\boldsymbol{H}_1\boldsymbol{X}_1=\begin{bmatrix}*&*&*&*&*&\cdots&*\\&\ddots&*&*&*&\cdots&*\\&&*&*&*&\cdots&*\\&&&\otimes&\otimes&\cdots&\otimes\\&&&0&\otimes&\cdots&\otimes\\&&&\vdots&\vdots&\ddots&\vdots\\&&&0&\otimes&\cdots&\otimes\end{bmatrix}. \tag{2.18}$$

$\boldsymbol{X}_{k+1}$ 与 $\boldsymbol{X}_k$ 相比, 改变的仅仅是 $\boldsymbol{X}_k$ 右下角的 $n-k+1$ 阶子矩阵, 并产生如式 (2.18) 右端矩阵中位置的零元素. 上述过程进行 m 次之后, 就产生了 $n\times m$ 阶矩阵

$$\boldsymbol{X}_{m+1}=\boldsymbol{H}_m\boldsymbol{H}_{m-1}\cdots\boldsymbol{H}_1\boldsymbol{X}_1=\begin{bmatrix}\boldsymbol{R}\\\boldsymbol{O}\end{bmatrix},$$

式中: $\boldsymbol{R}$ 为 m 阶上三角矩阵. 令

$$\boldsymbol{H}=\boldsymbol{H}_m\boldsymbol{H}_{m-1}\cdots\boldsymbol{H}_1,\quad \boldsymbol{Q}=\boldsymbol{H}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{I}_m\\\boldsymbol{O}\end{bmatrix}=[\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_m]\in\mathbb{R}^{n\times m},$$

则

$$\boldsymbol{X}=\boldsymbol{H}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{R}\\\boldsymbol{O}\end{bmatrix}=\boldsymbol{H}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{I}_m\\\boldsymbol{O}\end{bmatrix}\boldsymbol{R}=\boldsymbol{Q}\boldsymbol{R},$$

这就对一般矩阵实现了 QR 分解.

下面考虑这一分解算法的程序实现. 注意到

$$\boldsymbol{H}_k\boldsymbol{e}_i^{(n)}=\boldsymbol{e}_i^{(n)},\quad k>i,$$

式中: $\boldsymbol{e}_i^{(n)}$ 为 n 维坐标向量, 故 $\boldsymbol{Q}=[\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_m]$ 满足

$$\boldsymbol{q}_i=\boldsymbol{Q}\boldsymbol{e}_i^{(n)}=\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_m\boldsymbol{e}_i^{(n)}=\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_i\boldsymbol{e}_i^{(n)}.$$

为了描述整个算法, 再引入 $\boldsymbol{X}_{m+1}$ 的第 k 列为

$$\begin{aligned}\boldsymbol{r}_k&=\boldsymbol{H}_m\boldsymbol{H}_{m-1}\cdots\boldsymbol{H}_1\boldsymbol{X}_1\boldsymbol{e}_k=\boldsymbol{H}_m\boldsymbol{H}_{m-1}\cdots\boldsymbol{H}_1\boldsymbol{x}_k\\&=\boldsymbol{H}_k\boldsymbol{H}_{k-1}\cdots\boldsymbol{H}_1\boldsymbol{x}_k,\ 1\leqslant k\leqslant m,\end{aligned}\tag{2.19}$$

它就是 $\boldsymbol{X}_{k+1}$ 的第 k 列. 显然, 在计算出 $\boldsymbol{r}_k$ 之后, 在以后的各次正交化过程中, 它不再改变. 下面的算法给出了这一正交化过程的详细步骤.

算法 2.10 (Householder 正交化) 给定 $n\times m$ 阶列满秩矩阵 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m]$, 本算法产生 $n\times m$ 阶矩阵 $\boldsymbol{X}_{m+1}=[\boldsymbol{r}_1,\boldsymbol{r}_2,\cdots,\boldsymbol{r}_m]$ 和 $\boldsymbol{Q}=[\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_m]$(其列是单位正交的, 如果不需要 $\boldsymbol{Q}$ 则不需计算). $\boldsymbol{X}_{m+1}$ 的前 m 行构成 m 阶上三角矩阵 $\boldsymbol{R}$.

① $\boldsymbol{r}_1=\boldsymbol{x}_1$;

for $k=1:m$

② **if** $(k>1)$, 计算 $\boldsymbol{r}_k=\boldsymbol{H}_{k-1}\boldsymbol{H}_{k-2}\cdots\boldsymbol{H}_1\boldsymbol{x}_k$; **end**

③ 根据 $\boldsymbol{r}_k$ 的后 $n-k+1$ 个分量构成的向量 $\widetilde{\boldsymbol{x}}$, 计算 $\widetilde{\boldsymbol{H}}_k$ 满足
$\widetilde{\boldsymbol{H}}_k\widetilde{\boldsymbol{x}}=\eta_k\boldsymbol{e}_1^{(n-k+1)}$; 令 $\boldsymbol{H}_k=\mathrm{diag}(\boldsymbol{I}_{k-1},\widetilde{\boldsymbol{H}}_k)$;

④ 计算 $\boldsymbol{r}_k=\boldsymbol{H}_k\boldsymbol{r}_k$;

⑤ 计算 $\boldsymbol{q}_k=\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_k\boldsymbol{e}_k$;

end

注 2.11 算法 2.10 的第 ② 步可采取如下方式实现:

$$\begin{aligned}&\boldsymbol{z}=\boldsymbol{x}_k;\\&\textbf{for } (i=1:k-1),\ \boldsymbol{z}=\boldsymbol{H}_i\boldsymbol{z};\ \textbf{end}\\&\boldsymbol{r}_k=\boldsymbol{z};\end{aligned}\tag{2.20}$$

类似地, 第 ⑤ 步可采取如下方式实现:

$$\begin{aligned}&\boldsymbol{z}=\boldsymbol{e}_k;\\&\textbf{for } (i=k:-1:1),\ \boldsymbol{z}=\boldsymbol{H}_i\boldsymbol{z};\ \textbf{end}\\&\boldsymbol{q}_k=\boldsymbol{z};\end{aligned}$$

现在来看一下算法 2.10 的程序执行. 当 $k=1$ 时, 根据 $\boldsymbol{r}_1=\boldsymbol{x}_1$ 的后 n 个分量构成 $\widetilde{\boldsymbol{x}}_1$ (就是 n 维列向量 $\boldsymbol{x}_1$), 构造 $\widetilde{\boldsymbol{H}}_1=\boldsymbol{H}_1$, 计算 $\boldsymbol{X}_{m+1}$ 的第 1 列 $\boldsymbol{r}_1:=\boldsymbol{H}_1\boldsymbol{x}_1$, $k=1$ 迭代结束. 当 $k=2$ 时, 计算 $\boldsymbol{r}_2:=\boldsymbol{H}_1\boldsymbol{x}_2$, 根据 $\boldsymbol{r}_2$ 的后 $n-1$ 个分量构成 $\widetilde{\boldsymbol{x}}_2$, 构造 $\widetilde{\boldsymbol{H}}_2$ 和 $\boldsymbol{H}_2=\mathrm{diag}(\boldsymbol{I}_1,\widetilde{\boldsymbol{x}}_2)$. 计算 $\boldsymbol{X}_{m+1}$ 的第 2 列 $\boldsymbol{r}_2:=\boldsymbol{H}_2\boldsymbol{r}_2=\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{x}_1$. 至此, 确实 $\boldsymbol{r}_2$ 已经计算了. 注意算法 2.10 的第 ② 步计算的 $\boldsymbol{r}_k$ 是 $\boldsymbol{X}_k$ 的第 k 列, 第 ④ 步计算的 $\boldsymbol{r}_k$ 才是 $\boldsymbol{X}_{k+1}$ 的第 k 列, 亦即 $\boldsymbol{X}_{m+1}$ 的第 k 列.

此外, 该算法要保存 $\widetilde{\boldsymbol{H}}_k=\boldsymbol{I}_{n-k+1}-\beta_k\boldsymbol{v}_k\boldsymbol{v}_k^{\mathrm{T}}$ 中的数 β_k 和 $n-k+1$ 维列向量 $\boldsymbol{v}_k$. 在式 (2.20) $\boldsymbol{H}_i\boldsymbol{z}$ 的计算中, 由于 $\boldsymbol{H}_i=\mathrm{diag}(\boldsymbol{I}_{i-1},\widetilde{\boldsymbol{H}}_i)$, 所以 $\boldsymbol{H}_i\boldsymbol{z}$ 的前 $i-1$ 个分量与 $\boldsymbol{z}$ 的前 $s-1$ 个分量相同, 而 $\boldsymbol{H}_i\boldsymbol{z}$ 的后 $n-i+1$ 个分量构成的列向量就是 $\widetilde{\boldsymbol{H}}_k$ 与 $\boldsymbol{z}$ 后 $n-i+1$ 个分量构成的列向量的乘积.

算法 2.10 的 MATLAB 程序如下:

```
%Householder正交化程序-House_orth.m
function [Q,R]=House_orth(X)
[n,m]=size(X); %矩阵的行和列
R(:,1)=X(:,1); E=eye(n); Y=cell(m,1);
```

```
for k=1:m
    if k>1
        z=X(:,k);
        for i=1:k-1
            v=Y{i};  %取出来
            z=blkdiag(eye(i-1),eye(n-i+1)-b(i)*v*v')*z;
        end
        R(:,k)=z;
    end
    [v,beta]=r_house(R(k:n,k));
    H=blkdiag(eye(k-1),eye(n-k+1)-beta*v*v');
    R(:,k)=H*R(:,k);
    b(k)=beta;  Y{k}=v;  %存起来
    z=E(:,k);
    for i=k:-1:1
        v=Y{i};  %取出来
        z=blkdiag(eye(i-1),eye(n-i+1)-b(i)*v*v')*z;
    end
    Q(:,k)=z;
end
R=R(1:m,:);
```

例 2.6 用 Householder 正交化方法对例 2.5 中矩阵 $\boldsymbol{A}$ 的列向量进行正交化.

解 利用程序 House_orth.m, 在命令窗口输入:

```
A=[9,1,4,5;6,2,1,3;4,1,9,9;5,2,9,4;4,2,5,1];
[Q,R]=House_orth(A); err=norm(A-Q*R)
```

运行后即得结果, 且满足 $\|\boldsymbol{A}-\boldsymbol{QR}\|_2 = 4.3565 \times 10^{-15}$.

2.4 Krylov 子空间及其正交化

本节阐述与 Krylov 子空间有关的一些基本概念和重要性质, 并且给出计算其正交基的几个实用的算法.

2.4.1 Krylov 子空间

首先来介绍一下 Krylov 子空间的有关概念和性质. 考虑这样一个问题: 假如并不明确地知道矩阵 $\boldsymbol{A}$, 而只知道对任意给定向量 $\boldsymbol{x}$ 可以产生向量 $\boldsymbol{y}=\boldsymbol{Ax}$ 的机制, 那么可以从中得到 $\boldsymbol{A}$ 的一些什么信息呢?

当然, 在这种情况下, 若选定一个向量 $\boldsymbol{v}$, 便可得到一个序列

$$\boldsymbol{v}_0=\boldsymbol{v},\ \boldsymbol{v}_1=\boldsymbol{A}\boldsymbol{v}_0=\boldsymbol{A}\boldsymbol{v},\ \boldsymbol{v}_2=\boldsymbol{A}\boldsymbol{v}_1=\boldsymbol{A}^2\boldsymbol{v},\ \cdots,\ \boldsymbol{v}_{k+1}=\boldsymbol{A}\boldsymbol{v}_k=\boldsymbol{A}^{k+1}\boldsymbol{v},\ \cdots. \tag{2.21}$$

假定 $\boldsymbol{A}$ 的特征多项式为

$$p_A(\lambda)=\det(\lambda\boldsymbol{I}-\boldsymbol{A})=\lambda^n+\alpha_{n-1}\lambda^{n-1}+\cdots+\alpha_1\lambda+\alpha_0,$$

则由 Cayley–Hamilton 定理可知

$$\boldsymbol{v}_n+\alpha_{n-1}\boldsymbol{v}_{n-1}+\cdots+\alpha_1\boldsymbol{v}_1+\alpha_0\boldsymbol{v}_0=\boldsymbol{0},$$

从而有

$$\boldsymbol{V}\boldsymbol{p}=-\boldsymbol{v}_n, \tag{2.22}$$

式中:

$$\boldsymbol{V}=[\boldsymbol{v}_0,\boldsymbol{v}_1,\cdots,\boldsymbol{v}_{n-1}],\quad \boldsymbol{p}=(\alpha_0,\alpha_1,\cdots,\alpha_{n-1})^{\mathrm{T}}.$$

若 $\boldsymbol{V}$ 非奇异, 则由方程组 (2.22) 可唯一地确定特征多项式的系数构成的向量 $\boldsymbol{p}$, 从而可求得特征多项式, 进而得到 $\boldsymbol{A}$ 的所有特征值. 这就是说, 在所给定的条件下, 能得到 $\boldsymbol{A}$ 所有的特征值的信息.

上述求 $\boldsymbol{A}$ 的特征多项式的方法, 是在 1931 年首先由 Krylov 提出的. 因此, 后人就将序列式 (2.21) 称为 Krylov 序列, 而将子空间

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})=\mathrm{span}\{\boldsymbol{v},\boldsymbol{A}\boldsymbol{v},\cdots,\boldsymbol{A}^{k-1}\boldsymbol{v}\} \tag{2.23}$$

称为 Krylov 子空间, 将矩阵

$$\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})=[\boldsymbol{v},\boldsymbol{A}\boldsymbol{v},\cdots,\boldsymbol{A}^{k-1}\boldsymbol{v}] \tag{2.24}$$

称为 Krylov 矩阵.

由 Krylov 子空间的定义, 容易验证其有如下的性质.

定理 2.10 假定 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 和 $\boldsymbol{v}\in\mathbb{R}^n$ 已经给定, 其中 $\boldsymbol{v}\neq\boldsymbol{0}$, 则 Krylov 子空间有如下性质:

(1) Krylov 子空间序列满足

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})\subset\mathcal{K}_{k+1}(\boldsymbol{A},\boldsymbol{v}),\quad \boldsymbol{A}\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})\subset\mathcal{K}_{k+1}(\boldsymbol{A},\boldsymbol{v}).$$

(2) 对任意的非零实数 α, 有

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})=\mathcal{K}_k(\alpha\boldsymbol{A},\boldsymbol{v})=\mathcal{K}_k(\boldsymbol{A},\alpha\boldsymbol{v})\quad(\text{伸缩不变性}).$$

(3) 对任意的实数 μ, 有

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})=\mathcal{K}_k(\boldsymbol{A}-\mu\boldsymbol{I},\boldsymbol{v})\quad(\text{位移不变性}).$$

(4) 对任意的非奇异矩阵 $\boldsymbol{W}\in\mathbb{R}^{n\times n}$, 有

$$\mathcal{K}_k(\boldsymbol{W}^{-1}\boldsymbol{A}\boldsymbol{W},\boldsymbol{W}^{-1}\boldsymbol{v})=\boldsymbol{W}^{-1}\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v}).$$

(5) 若记次数不超过 $k-1$ 的实系数多项式的全体为 $\mathcal{P}_{k-1}$, 则 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})$ 有如下表示

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})=\{p(\boldsymbol{A})\boldsymbol{v}:p\in\mathcal{P}_{k-1}\}.$$

注 2.12 定理 2.10 的第 2 条是说, 对矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{v}$ 乘以任意非零实数, 并不改变其生成的 Krylov 子空间, 称 Krylov 子空间的这一性质为伸缩不变性. 第 3 条是说, 对矩阵进行位移, 并不改变其生成的 Krylov 子空间, 称这一性质为位移不变性. 第 4 条性质给出了在矩阵的相似变换下相应的 Krylov 子空间之间的关系, 这对处理与 Krylov 子空间有关的理论时十分有用. 第 5 条给出了 Krylov 子空间一个重要性质, 即其每个元都可看作是某个矩阵多项式作用到初始向量 $\boldsymbol{v}$ 上而得到的.

假设 $(\lambda, \boldsymbol{v})$ 是 $\boldsymbol{A}$ 的特征对, 则有 $\boldsymbol{A}^i\boldsymbol{v} = \lambda^i\boldsymbol{v}$, 从而有

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_1(\boldsymbol{A}, \boldsymbol{v}), \quad k = 1, 2, \cdots.$$

换句话说, 此时在 Krylov 序列中, 从第 1 项之后就没有任何新的向量产生, 即它终止于第一项. 一般来讲, Krylov 序列在第 ℓ 项终止, 是指 ℓ 为满足 $\mathcal{K}_{\ell+1}(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v})$ 的最小整数.

定理 2.11 若 Krylov 序列在第 ℓ 项终止, 则 $\mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v})$ 是 $\boldsymbol{A}$ 的一个 ℓ 维不变子空间. 反过来, 若 $\boldsymbol{v}$ 属于 $\boldsymbol{A}$ 的一个 m 维不变子空间, 则必存在一个 $\ell \leqslant m$, 使得对应的 Krylov 序列在第 ℓ 项终止.

证明 若 Krylov 序列在第 ℓ 项终止, 则对任意 $\boldsymbol{x} \in \mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v})$, 有

$$\boldsymbol{A}\boldsymbol{x} \in \mathcal{K}_{\ell+1}(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v}),$$

即 $\mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v})$ 是 $\boldsymbol{A}$ 的不变子空间. 此外, 由于 ℓ 是使得 $\mathcal{K}_{\ell+1}(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v})$ 的最小整数, 故必有 $\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{v}$ 这 ℓ 个向量是线性无关的, 从而就有 $\dim \mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v}) = \ell$.

反过来, 若 $\boldsymbol{v} \in \mathcal{X}_m$, 这里 $\mathcal{X}_m$ 是 $\boldsymbol{A}$ 的 m 维不变子空间, 则对任意的整数 k, 必有 $\boldsymbol{A}^k\boldsymbol{v} \in \mathcal{X}_m$. 故必存在一个 $\ell \leqslant m$, 使得 $\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{v}$ 线性无关, 但 $\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{v}, \boldsymbol{A}^\ell\boldsymbol{v}$ 线性相关, 从而有 $\mathcal{K}_{\ell+1}(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v})$. 当然这样的 ℓ 必须是使得上述等式成立的最小整数. 因此, 对应的 Krylov 序列必在第 ℓ 项终止. 证毕. □

由定理 2.11 可知, 当 Krylov 序列在第 ℓ 项终止时, 则 $\mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v})$ 中就包含了 $\boldsymbol{A}$ 的一些特征向量. 这对完成诸多的计算任务是十分有利的, 但也有不利的一面. 例如, 如果 ℓ 很小, 那么 $\mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v})$ 中可能并不含有所需要的信息, 这就会给计算带来困难, 必须重新开始.

2.4.2 Arnoldi 正交分解

在实际应用 Krylov 子空间时, 需要它的一些基向量来描述它. 当然, 定义它的向量组应该是它的一组天然的基. 然而这组基在实际计算时是没有什么用处的, 因为随着 k 的增加, Krylov 矩阵会变得越来越病态. 因此, 为了实际计算的需要, 必须寻找其他对扰动并不敏感的基向量.

设 $\boldsymbol{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}) = [\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^k\boldsymbol{v}]$ 是列满秩的, 并假定它的 QR 分解为

$$\boldsymbol{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}) = \boldsymbol{V}_{k+1}\boldsymbol{R}_{k+1}, \tag{2.25}$$

式中: $\boldsymbol{V}_{k+1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{k+1}] \in \mathbb{R}^{n\times(k+1)}$ 满足 $\boldsymbol{V}_{k+1}^{\mathrm{T}}\boldsymbol{V}_{k+1} = \boldsymbol{I}_{k+1}$, $\boldsymbol{R}_{k+1}$ 为非奇异的上三角矩阵.

通常称 $\boldsymbol{V}_{k+1}$ 为 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})$ 的 QR 分解的 Q 因子, 简称为 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})$ 的 Q 因子. 将 $\boldsymbol{V}_{k+1}$ 和 $\boldsymbol{R}_{k+1}$ 作如下分块:

$$\boldsymbol{V}_{k+1}=[\boldsymbol{V}_k,\boldsymbol{v}_{k+1}],\quad \boldsymbol{R}_{k+1}=\begin{bmatrix}\boldsymbol{R}_k & \boldsymbol{r}_{k+1}\\ \boldsymbol{0} & \rho_{k+1,k+1}\end{bmatrix}.$$

再注意到 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})=[\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v}),\boldsymbol{A}^k\boldsymbol{v}]$, 由式 (2.25), 得

$$\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})=\boldsymbol{V}_k\boldsymbol{R}_k, \tag{2.26}$$

即 $\boldsymbol{V}_k$ 就是 $\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的 QR 分解的 Q 因子. 从而有

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})=\mathcal{R}(\boldsymbol{V}_k)=\operatorname{span}\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k\},$$

即 $\boldsymbol{V}_k$ 的列向量就构成了 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的一组标准正交基.

这样自然想到利用 $\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的 QR 分解来产生 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的一组正交基. 然而这一方法是不可取的, 因为 $\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$ 是十分病态的, 在计算机上形成 $\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$ 就会引起较大的误差. 因此, 需要寻求其他方法来计算 $\boldsymbol{V}_k$.

事实上, 由式 (2.25) 和式 (2.26) 可以导出如下结果.

定理 2.12 给定矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 和向量 $\boldsymbol{v}\in\mathbb{R}^n$, 并假定 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})$ 是列满秩的, 而且其 QR 分解的 Q 因子为 $\boldsymbol{V}_{k+1}$, 则必有一个 $(k+1)\times k$ 阶不可约上 Hessenberg 矩阵 $\widetilde{\boldsymbol{H}}_k$, 使得

$$\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_{k+1}\widetilde{\boldsymbol{H}}_k, \tag{2.27}$$

式中: $\boldsymbol{V}_k$ 为由 $\boldsymbol{V}_{k+1}$ 的前 k 列构成的矩阵. 反过来, 若有

$$\boldsymbol{V}_{k+1}=[\boldsymbol{V}_k,\boldsymbol{v}_{k+1}]\in\mathbb{R}^{n\times(k+1)},\quad \boldsymbol{V}_{k+1}^{\mathrm{T}}\boldsymbol{V}_{k+1}=\boldsymbol{I}_{k+1},$$

以及不可约的上 Hessenberg 矩阵 $\widetilde{\boldsymbol{H}}_k$ 使得式 (2.27) 成立, 则 $\boldsymbol{V}_{k+1}$ 必是 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v}_1)$ 的 QR 分解的 Q 因子, 这里 $\boldsymbol{v}_1=\boldsymbol{V}_{k+1}\boldsymbol{e}_1$ 是 $\boldsymbol{V}_{k+1}$ 的第 1 列.

证明 设 $\boldsymbol{V}_{k+1}=[\boldsymbol{V}_k,\boldsymbol{v}_{k+1}]$ 是 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})$ 的 QR 分解的 Q 因子, 则由式 (2.26) 得

$$[\boldsymbol{A}\boldsymbol{v},\cdots,\boldsymbol{A}^k\boldsymbol{v}]=\boldsymbol{A}\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})=\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{R}_k. \tag{2.28}$$

再由式 (2.25) 得

$$[\boldsymbol{v},\boldsymbol{A}\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})]=\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})=\boldsymbol{V}_{k+1}\boldsymbol{R}_{k+1}.$$

比较上式两边的后 k 列, 得

$$\boldsymbol{A}\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})=\boldsymbol{V}_{k+1}\widehat{\boldsymbol{H}}_k, \tag{2.29}$$

式中: $\widehat{\boldsymbol{H}}_k$ 为由 $\boldsymbol{R}_{k+1}$ 的后 k 列构成的 $(k+1)\times k$ 阶上 Hessenberg 矩阵.

注意: $\boldsymbol{R}_{k+1}$ 非奇异蕴含着 $\widehat{\boldsymbol{H}}_k$ 是不可约的. 将式 (2.28) 与式 (2.29) 相结合, 得

$$\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{R}_k=\boldsymbol{V}_{k+1}\widehat{\boldsymbol{H}}_k\Longrightarrow\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_{k+1}\widetilde{\boldsymbol{H}}_k,$$

式中: $\widetilde{\boldsymbol{H}}_k = \widehat{\boldsymbol{H}}_k \boldsymbol{R}_k^{-1}$.

由于 $\widehat{\boldsymbol{H}}_k$ 是不可约的上 Hessenberg 矩阵, 而 $\boldsymbol{R}_k$ 是非奇异的上三角矩阵, 因此 $\widetilde{\boldsymbol{H}}_k$ 必为不可约的上 Hessenberg 矩阵.

反过来, 若有式 (2.27) 成立, 比较式 (2.27) 两边的各列, 得

$$\boldsymbol{A}\boldsymbol{v}_j = \sum_{i=1}^{j+1} h_{ij}\boldsymbol{v}_i, \quad j = 1, 2, \cdots, k.$$

于是有

$$\begin{aligned}
\boldsymbol{v}_2 &= \frac{1}{h_{21}}(\boldsymbol{A}\boldsymbol{v}_1 - h_{11}\boldsymbol{v}_1) = \rho_{12}\boldsymbol{v}_1 + \rho_{22}\boldsymbol{A}\boldsymbol{v}_1, \quad \rho_{22} = \frac{1}{h_{21}} \neq 0; \\
\boldsymbol{v}_3 &= \frac{1}{h_{32}}(\boldsymbol{A}\boldsymbol{v}_2 - h_{12}\boldsymbol{v}_1 - h_{22}\boldsymbol{v}_2) \\
&= \frac{1}{h_{32}}(\boldsymbol{A} - h_{22}\boldsymbol{I})(\rho_{12}\boldsymbol{v}_1 + \rho_{22}\boldsymbol{A}\boldsymbol{v}_1) - \frac{h_{12}}{h_{32}}\boldsymbol{v}_1 \\
&= \rho_{13}\boldsymbol{v}_1 + \rho_{23}\boldsymbol{A}\boldsymbol{v}_1 + \rho_{33}\boldsymbol{A}^2\boldsymbol{v}_1, \quad \rho_{33} = \frac{\rho_{22}}{h_{32}} \neq 0.
\end{aligned}$$

如此下去, 得

$$\boldsymbol{v}_j = \rho_{1j}\boldsymbol{v}_1 + \rho_{2j}\boldsymbol{A}\boldsymbol{v}_1 + \cdots + \rho_{jj}\boldsymbol{A}^{j-1}\boldsymbol{v}_1, \ \rho_{jj} = \frac{\rho_{j-1,j-1}}{h_{j,j-1}} \neq 0, \ 2 \leqslant j \leqslant k+1. \tag{2.30}$$

令

$$\boldsymbol{R}^{-1} = \begin{bmatrix}
1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1,k+1} \\
 & \rho_{22} & \rho_{23} & \cdots & \rho_{2,k+1} \\
 & & \rho_{33} & \cdots & \rho_{3,k+1} \\
 & & & \ddots & \vdots \\
 & & & & \rho_{k+1,k+1}
\end{bmatrix},$$

则将式 (2.30) 写成矩阵形式, 并且再补充第 1 列 $\boldsymbol{v}_1$, 得

$$[\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{k+1}] = [\boldsymbol{v}_1, \boldsymbol{A}\boldsymbol{v}_1, \cdots, \boldsymbol{A}^k\boldsymbol{v}_1]\boldsymbol{R}^{-1},$$

即

$$\boldsymbol{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}_1) = \boldsymbol{V}_{k+1}\boldsymbol{R},$$

这表明 $\boldsymbol{V}_{k+1}$ 是 $\boldsymbol{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}_1)$ 的 QR 分解的 Q 因子. 证毕. □

基于这一结果, 称形如式 (2.27) 的公式为一个长度为 k 的 Arnoldi 分解; 若其中的 $\widetilde{\boldsymbol{H}}_k$ 是不可约的, 则称这一分解是不可约的, 否则称为可约的.

由定理 2.12 可知, 只需求得分解式 (2.27), 即 $\boldsymbol{K}_k(\boldsymbol{A}, \boldsymbol{v})$ 的 QR 分解的 Q 因子, 就能得到 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ 的一组标准正交基. 下面计算分解式 (2.27).

将 $\widetilde{\boldsymbol{H}}_k$ 分块为

$$\widetilde{\boldsymbol{H}}_k = \begin{bmatrix} \boldsymbol{H}_k \\ \beta_k \boldsymbol{e}_k^{\mathrm{T}} \end{bmatrix},$$

则 Arnoldi 分解式 (2.27) 可改写为

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}. \tag{2.31}$$

这种形式有时使用起来更方便. 在式 (2.31) 两边左乘 $\boldsymbol{V}_k^{\mathrm{T}}$, 并且注意到 $\boldsymbol{V}_k$ 与 $\boldsymbol{v}_{k+1}$ 的正交性, 即有 $\boldsymbol{H}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k$. 通常称矩阵 $\boldsymbol{H}_k$ 为 $\boldsymbol{A}$ 关于 $\boldsymbol{V}_k$ 的 Rayleigh 商. 事实上, 式 (2.31) 提供了一种由 $\boldsymbol{V}_k$ 来计算 $\boldsymbol{v}_{k+1}$ 的方法. 比较式 (2.31) 两边矩阵的最后一列, 得

$$\boldsymbol{A}\boldsymbol{v}_k = \boldsymbol{V}_k\boldsymbol{h}_k + \beta_k\boldsymbol{v}_{k+1}, \tag{2.32}$$

式中: $\boldsymbol{h}_k$ 为 $\boldsymbol{H}_k$ 的最后一列.

由于 $\boldsymbol{V}_{k+1}$ 是列正交的, 故在式 (2.32) 两边左乘 $\boldsymbol{V}_k^{\mathrm{T}}$, 得

$$\boldsymbol{h}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_k. \tag{2.33}$$

又由式 (2.32), 得

$$\beta_k\boldsymbol{v}_{k+1} = \boldsymbol{A}\boldsymbol{v}_k - \boldsymbol{V}_k\boldsymbol{h}_k. \tag{2.34}$$

于是, 便有

$$\beta_k = \|\boldsymbol{A}\boldsymbol{v}_k - \boldsymbol{V}_k\boldsymbol{h}_k\|_2, \quad \boldsymbol{v}_{k+1} = (\boldsymbol{A}\boldsymbol{v}_k - \boldsymbol{V}_k\boldsymbol{h}_k)/\beta_k. \tag{2.35}$$

这本质上是一个求向量 $\boldsymbol{A}\boldsymbol{v}_k$ 在 $\mathcal{R}(\boldsymbol{V}_k)^{\perp}$ 上的正交投影的 Gram–Schmidt 的正交化过程. 从对 Gram–Schmidt 正交化过程的研究可知, 为了保证其数值稳定性, 应该使用修正的 Gram–Schmidt 正交化过程. 记

$$\boldsymbol{h}_k = (h_{1k}, h_{2k}, \cdots, h_{kk})^{\mathrm{T}}, \quad h_{k+1,k} = \beta_k, \quad \boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}_k, \quad \boldsymbol{r} = \boldsymbol{A}\boldsymbol{v}_k - \boldsymbol{V}_k\boldsymbol{h}_k.$$

利用这些记号, 将式 (2.33) 和式 (2.34) 换一种表达方式即为

$$h_{ik} = \boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{w}, \quad i = 1, 2, \cdots, k, \quad \boldsymbol{r} = \boldsymbol{w} - \sum_{i=1}^{k} h_{ik}\boldsymbol{v}_i.$$

上式是说, 经典的 Gram–Schmidt 正交化过程, 是将 $\boldsymbol{w}$ 在每个 $\boldsymbol{v}_i$ 上的投影 h_{ik} 都算好之后, 再从 $\boldsymbol{w}$ 中一并减去 $\boldsymbol{w}$ 在 $\mathcal{R}(\boldsymbol{V}_k)^{\perp}$ 上的正交投影. 而修正的 Gram–Schmidt 正交化过程是, 当算好 $\boldsymbol{w}$ 在 $\boldsymbol{v}_1$ 上的正交投影 $h_{1k} = \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{w}$ 后, 马上从 $\boldsymbol{w}$ 中减去 $h_{1k}\boldsymbol{v}_1$, 即计算 $\boldsymbol{w}_1 = \boldsymbol{w} - h_{1k}\boldsymbol{v}_1$. 然后, 再用 $h_{2k} = \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{w}_1$ 确定 h_{2k}, 而不是用 $h_{2k} = \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{w}$ 确定. 当然, 从理论上来讲, 二者是等价的, 即 $h_{2k} = \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{w} = \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{w}_1$. 然后再从 $\boldsymbol{w}_1$ 中减去 $h_{2k}\boldsymbol{v}_2$, 即计算 $\boldsymbol{w}_2 = \boldsymbol{w}_1 - h_{2k}\boldsymbol{v}_2$, 依此类推. 数值实验证明, 修正的 Gram–Schmidt 正交化过程要比经典的 Gram–Schmidt 过程数值性态好得多.

综合上面的讨论, 可得如下算法.

算法 2.11 (Arnoldi 正交分解) 给定矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 向量 $\boldsymbol{v}_1 \in \mathbb{R}^n$ ($\|\boldsymbol{v}_1\|_2 = 1$) 和正整数 k, 本算法计算一个长度为 k 的 Arnoldi 分解: $\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + h_{k+1,k}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}$.

for $j = 1 : k$

$\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}_j$;

```
    for i = 1 : j
        h_ij = v_i^T w;  w = w − h_ij v_i;
    end
    h_{j+1,j} = ‖w‖_2;
    if h_{j+1,j} = 0
        stop
    else
        v_{j+1} = w/h_{j+1,j};
    end
end
```

注 2.13 算法 2.11 只在计算 $\boldsymbol{w}$ 时用到 $\boldsymbol{A}$ 与一个向量作乘积, 这使得该算法可充分利用 $\boldsymbol{A}$ 的稀疏性和其所具有的特殊结构. 此外, 若 Arnoldi 过程中途中断, 即计算过程中出现了 $h_{j+1,j}=0$, 则这表明 $\mathcal{K}_j(\boldsymbol{A},\boldsymbol{v}_1)$ 已经是 $\boldsymbol{A}$ 的不变子空间. 在很多情况下, 这是有利的.

这里需要指出的是, 数值实验表明, 算法 2.11 在实际使用时, $\boldsymbol{v}_j$ 之间的正交性很快就会损失掉. 解决这一问题的一种方法就是在计算过程中使用重正交化技术 (即在算法中再重复执行一次 Gram–Schmidt 正交化过程). 这就得到如下的算法.

算法 2.12 (重正交化 Arnoldi 分解) 给定矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 向量 $\boldsymbol{v}_1\in\mathbb{R}^n$ 满足 $\|\boldsymbol{v}_1\|_2=1$ 以及正整数 k, 本算法计算一个长度为 k 的 Arnoldi 分解: $\boldsymbol{AV}_k=\boldsymbol{V}_k\boldsymbol{H}_k+h_{k+1,k}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}$, 并且在计算过程中使用重正交化技术.

```
for j = 1 : k
    w = A v_j;
    for i = 1 : j
        h_ij = v_i^T w;  w = w − h_ij v_i;
    end
    for i = 1 : j  (重正交化)
        s = v_i^T w;  h_ij = h_ij + s;  w = w − s v_i;
    end
    h_{j+1,j} = ‖w‖_2;
    if h_{j+1,j} = 0
        stop
    else
        v_{j+1} = w/h_{j+1,j};
    end
end
```

2.4.3 Lanczos 正交分解

当 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 是对称矩阵时, 在 Arnoldi 分解中, 其关于 $\boldsymbol{V}_k$ 的 Rayleigh 商 $\boldsymbol{H}_k=\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k$ 就是一个对称三对角矩阵. 这样对应的 Arnoldi 分解就变成

$$\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_k\boldsymbol{T}_k+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}, \tag{2.36}$$

式中:

$$\boldsymbol{T}_k=\begin{bmatrix}\alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{k-1}\\ & & \beta_{k-1} & \alpha_k\end{bmatrix}.$$

此时, 称式 (2.36) 为一个长度为 k 的 Lanczos 分解.

比较式 (2.36) 两边的各列, 得

$$\boldsymbol{A}\boldsymbol{v}_1=\alpha_1\boldsymbol{v}_1+\beta_1\boldsymbol{v}_2,$$

$$\boldsymbol{A}\boldsymbol{v}_j=\beta_{j-1}\boldsymbol{v}_{j-1}+\alpha_j\boldsymbol{v}_j+\beta_j\boldsymbol{v}_{j+1},\quad j=2,\cdots,k.$$

于是有

$$\alpha_1=\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_1,$$

$$\beta_1=\|\boldsymbol{A}\boldsymbol{v}_1-\alpha_1\boldsymbol{v}_1\|_2,\quad \boldsymbol{v}_2=(\boldsymbol{A}\boldsymbol{v}_1-\alpha_1\boldsymbol{v}_1)/\beta_1,$$

$$\alpha_j=\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_j,\quad \boldsymbol{r}_j=\boldsymbol{A}\boldsymbol{v}_j-\alpha_j\boldsymbol{v}_j-\beta_{j-1}\boldsymbol{v}_{j-1},$$

$$\beta_j=\|\boldsymbol{r}_j\|_2,\quad \boldsymbol{v}_{j+1}=\boldsymbol{r}_j/\beta_j,\quad j=2,\cdots,k.$$

这样, 就得到了如下的算法.

算法 2.13 (Lanczos 方法) 给定矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 满足 $\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{A}$, 向量 $\boldsymbol{v}_1\in\mathbb{R}^n$ 满足 $\|\boldsymbol{v}_1\|_2=1$ 以及正整数 k, 本算法计算一个长度为 k 的 Lanczos 分解: $\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_k\boldsymbol{T}_k+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}$.

$\beta_0=0;\ \boldsymbol{v}_0=\boldsymbol{0};$
for $j=1:k$
  $\boldsymbol{w}=\boldsymbol{A}\boldsymbol{v}_j;\quad \alpha_j=\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{w};$
  $\boldsymbol{w}=\boldsymbol{w}-\alpha_j\boldsymbol{v}_j-\beta_{j-1}\boldsymbol{v}_{j-1};$
  $\beta_j=\|\boldsymbol{w}\|_2;$
  if $\beta_j=0$
    stop
  else
    $\boldsymbol{v}_{j+1}=\boldsymbol{w}/\beta_j;$
  end
end

当然在实际应用时, 该算法所产生的 $\boldsymbol{v}_j$ 也将很快的失去它们之间的正交性. 弥补这一损失的方法仍然是重正交化方法. 其实重正交化就是在算法中的 $\boldsymbol{w} = \boldsymbol{w} - \alpha_j \boldsymbol{v}_j - \beta_{j-1}\boldsymbol{v}_{j-1}$ 之后, 再加入一句

$$. \boldsymbol{w} = \boldsymbol{w} - \sum_{i=1}^{j-1} (\boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{w}) \boldsymbol{v}_i.$$

具体算法如下.

算法 2.14 (重正交化的 Lanczos 方法) 给定矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 满足 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, 向量 $\boldsymbol{v}_1 \in \mathbb{R}^n$ 满足 $\|\boldsymbol{v}_1\|_2 = 1$ 以及正整数 k, 本算法计算一个长度为 k 的 Lanczos 分解: $\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k \boldsymbol{T}_k + \beta_k \boldsymbol{v}_{k+1} \boldsymbol{e}_k^{\mathrm{T}}$.

$\beta_0 = 0;\ \boldsymbol{v}_0 = \boldsymbol{0};$
for $j = 1 : k$
  $\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}_j;\quad \alpha_j = \boldsymbol{v}_j^{\mathrm{T}} \boldsymbol{w};$
  $\boldsymbol{w} = \boldsymbol{w} - \alpha_j \boldsymbol{v}_j - \beta_{j-1}\boldsymbol{v}_{j-1};$
  $\boldsymbol{w} = \boldsymbol{w} - \sum_{i=1}^{j-1} (\boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{w}) \boldsymbol{v}_i;$
  $\beta_j = \|\boldsymbol{w}\|_2;$
  if $\beta_j = 0$
    stop
  else
    $\boldsymbol{v}_{j+1} = \boldsymbol{w}/\beta_j;$
  end
end

当然, 在具体计算时, 也可以用修正的 Gram–Schmidt 正交化方法来实现上式的计算. 有时为了使算法所产生的 $\boldsymbol{v}_j$ 之间有更好的正交性, 也可以连续进行两次重正交化, 即所谓的完全重正交化.

下面再给出一个具体的例子, 观察一下重正交化的 Lanczos 过程的效果.

例 2.7 考虑矩阵

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \cdots & n \end{bmatrix}. \tag{2.37}$$

随机地选取一个初始向量 $\boldsymbol{v}_1$, 并且取 $n = 1000$, $k = 50$. 若用算法 2.13, 则计算得到的矩阵 $\boldsymbol{V}_{50}$ 满足

$$\|\boldsymbol{V}_{50}^{\mathrm{T}} \boldsymbol{V}_{50} - \boldsymbol{I}_{50}\|_{\mathrm{F}} = 9.5938.$$

而若用重正交化的 Lanczos 算法, 则计算得到的矩阵 $\boldsymbol{V}_{50}$ 满足

$$\|\boldsymbol{V}_{50}^{\mathrm{T}}\boldsymbol{V}_{50}-\boldsymbol{I}_{50}\|_{\mathrm{F}}=2.0073\times10^{-13}.$$

由此观察到, 重正交化的 Lanczos 过程的效果是明显的.

2.5 投影方法

大多数已有的求解线性方程组迭代方法都按某种方式使用一种投影过程. 投影过程表达了从一个子空间推断出线性方程组近似解的一种规范方式. 本节在一般的框架下描述这种规范方式, 并给出一些理论. 进一步, 对一维情形的投影方法给出了较为详细的描述, 这给第 4 章中更为复杂的子空间投影方法提供了一个预览.

2.5.1 投影算子及其性质

投影算子 $\boldsymbol{P}$ 是从 $\mathbb{C}^n$ 到其自身的一个幂等线性映射, 即满足

$$\boldsymbol{P}^2=\boldsymbol{P}.$$

由此定义可得下述一些简单的性质. 首先, 如果 $\boldsymbol{P}$ 是一个投影, 则 $\boldsymbol{I}-\boldsymbol{P}$ 也是投影, 且下述关系成立,

$$\mathcal{N}(\boldsymbol{P})=\mathcal{R}(\boldsymbol{I}-\boldsymbol{P}).$$

另外, 两个子空间 $\mathcal{N}(\boldsymbol{P})$ 和 $\mathcal{R}(\boldsymbol{P})$ 的交只有元素 $\boldsymbol{0}$, 即 $\mathcal{N}(\boldsymbol{P})\cap\mathcal{R}(\boldsymbol{P})=\{\boldsymbol{0}\}$. 事实上, 如果向量 $\boldsymbol{x}$ 属于 $\mathcal{R}(\boldsymbol{P})$, 则由幂等性有 $\boldsymbol{P}\boldsymbol{x}=\boldsymbol{x}$. 如果它也属于 $\mathcal{N}(\boldsymbol{P})$, 则 $\boldsymbol{P}\boldsymbol{x}=\boldsymbol{0}$, 因此 $\boldsymbol{x}=\boldsymbol{P}\boldsymbol{x}=\boldsymbol{0}$. 此外, $\mathbb{C}^n$ 的每个元素可被写成 $\boldsymbol{x}=\boldsymbol{P}\boldsymbol{x}+(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}$, 因此, 空间 $\mathbb{C}^n$ 可被分解成直和

$$\mathbb{C}^n=\mathcal{N}(\boldsymbol{P})\oplus\mathcal{R}(\boldsymbol{P}).$$

反之, 每个形成 $\mathbb{C}^n$ 的直和的子空间对 $\mathcal{K}$ 和 $\mathcal{S}$ 唯一定义一个投影算子, 使得 $\mathcal{R}(\boldsymbol{P})=\mathcal{K}$ 和 $\mathcal{N}(\boldsymbol{P})=\mathcal{S}$. 相应的投影算子 $\boldsymbol{P}$ 映 $\mathbb{C}^n$ 的一个元素 $\boldsymbol{x}$ 为分量 $\boldsymbol{x}_1$, 其中 $\boldsymbol{x}_1$ 是相应直和唯一分解 $\boldsymbol{x}=\boldsymbol{x}_1+\boldsymbol{x}_2$ 中的 $\mathcal{K}$ 分量.

事实上, 这种对应是唯一的, 即任意投影算子 $\boldsymbol{P}$ 可由给定的两个子空间完全确定: $\boldsymbol{P}$ 的值域 $\mathcal{K}$ 和它的核 $\mathcal{S}$ (亦即 $\boldsymbol{I}-\boldsymbol{P}$ 的值域). 若对任意的 $\boldsymbol{x}$, 向量 $\boldsymbol{P}\boldsymbol{x}$ 满足

$$\boldsymbol{P}\boldsymbol{x}\in\mathcal{K},\quad \boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}\in\mathcal{S},$$

则称线性映射 $\boldsymbol{P}$ 沿 (或平行于) 子空间 $\mathcal{S}$ 将 $\boldsymbol{x}$ 投影到 $\mathcal{K}$ 上. 如果 $\boldsymbol{P}$ 的秩为 m, 则 $\boldsymbol{I}-\boldsymbol{P}$ 的值域是 $n-m$ 维的. 因此, 通过其 m 维的正交补 $\mathcal{L}=\mathcal{S}^{\perp}$ 来定义 $\mathcal{S}$ 是自然的. 对任意的 $\boldsymbol{x}$, 上述定义 $\boldsymbol{u}=\boldsymbol{P}\boldsymbol{x}$ 的条件变为

$$\boldsymbol{u}\in\mathcal{K},\tag{2.38}$$

$$\boldsymbol{x}-\boldsymbol{u}\perp\mathcal{L}.\tag{2.39}$$

这定义了一个正交于子空间 $\mathcal{L}$ 的到 $\mathcal{K}$ 上的投影算子 $\boldsymbol{P}$. 式 (2.38) 建立了 m 个自由度, 式 (2.39) 给出了从这些自由度来定义 $\boldsymbol{P}\boldsymbol{x}$ 的 m 个约束.

现在的问题是: 任给两个都是 m 维的子空间 $\mathcal{K}$ 和 $\mathcal{L}$, 是否总能通过式 (2.38) 和式 (2.39) 定义一个正交于 $\mathcal{L}$ 的到 $\mathcal{K}$ 上的投影算子 $\boldsymbol{P}$? 下面的引理回答这个问题.

引理 2.1 *给定两个 m 维子空间 $\mathcal{K}$ 和 $\mathcal{L}$, 下面两个条件是等价的:*

(1) *$\mathcal{K}$ 中没有非零向量正交于 $\mathcal{L}$.*

(2) *对任意的 $\boldsymbol{x} \in \mathbb{C}^n$, 存在唯一向量 $\boldsymbol{u}$ 满足式 (2.38) 和 (2.39).*

引理 2.1 说明给定两个满足条件 $\mathcal{K} \cap \mathcal{L}^{\perp} = \{\boldsymbol{0}\}$ 的子空间 $\mathcal{K}$ 和 $\mathcal{L}$, 存在一个正交于 $\mathcal{L}$ 的到 $\mathcal{K}$ 上的投影算子 $\boldsymbol{P}$, 它用式 (2.38) 和 (2.39) 定义任意向量 $\boldsymbol{x}$ 的投影向量 $\boldsymbol{u}$. 此投影算子满足

$$\mathcal{R}(\boldsymbol{P}) = \mathcal{K}, \quad \mathcal{N}(\boldsymbol{P}) = \mathcal{L}^{\perp}.$$

特别地, 条件 $\boldsymbol{P}\boldsymbol{x} = \boldsymbol{0}$ 转化为 $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{P})$, 而这意味着 $\boldsymbol{x} \in \mathcal{L}^{\perp}$. 反之亦真. 这可得下面的性质:

$$\boldsymbol{P}\boldsymbol{x} = \boldsymbol{0} \iff \boldsymbol{x} \in \mathcal{L}^{\perp}.$$

为了得到一般投影算子的矩阵表示, 需要两组基: 子空间 $\mathcal{K} = \mathcal{R}(\boldsymbol{P})$ 的一组基 $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m]$ 和子空间 $\mathcal{L}$ 的一组基 $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]$. 当

$$(\boldsymbol{v}_i, \boldsymbol{w}_j) = \delta_{ij} = \begin{cases} 1, & \text{若 } i = j \\ 0, & \text{若 } i \neq j \end{cases},$$

时, 称这两组基是双正交的, 写成矩阵形式为 $\boldsymbol{W}^{\mathrm{H}}\boldsymbol{V} = \boldsymbol{I}$. 由于 $\boldsymbol{P}\boldsymbol{x}$ 属于 $\mathcal{K}$, 故可表示为 $\boldsymbol{P}\boldsymbol{x} = \boldsymbol{V}\boldsymbol{y}$. 约束 $\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x} \perp \mathcal{L}$ 等价于条件

$$(\boldsymbol{x} - \boldsymbol{V}\boldsymbol{y}, \boldsymbol{w}_i) = 0, \quad i = 1, \cdots, m.$$

写成矩阵形式, 即

$$\boldsymbol{W}^{\mathrm{H}}(\boldsymbol{x} - \boldsymbol{V}\boldsymbol{y}) = \boldsymbol{0}. \tag{2.40}$$

如果两组基是双正交的, 则得 $\boldsymbol{y} = \boldsymbol{W}^{\mathrm{H}}\boldsymbol{x}$. 因此, 此时 $\boldsymbol{P}\boldsymbol{x} = \boldsymbol{V}\boldsymbol{W}^{\mathrm{H}}\boldsymbol{x}$, 这导致 $\boldsymbol{P}$ 的矩阵表示

$$\boldsymbol{P} = \boldsymbol{V}\boldsymbol{W}^{\mathrm{H}}. \tag{2.41}$$

对两组基 $\boldsymbol{V}$ 和 $\boldsymbol{W}$ 不是双正交的情形, 由式 (2.40) 易见

$$\boldsymbol{P} = \boldsymbol{V}(\boldsymbol{W}^{\mathrm{H}}\boldsymbol{V})^{-1}\boldsymbol{W}^{\mathrm{H}}.$$

如果假设 $\mathcal{K}$ 中没有向量正交于 $\mathcal{L}$, 则可以证明 $m \times m$ 阶矩阵 $\boldsymbol{W}^{\mathrm{H}}\boldsymbol{V}$ 是非奇异的.

当子空间 $\mathcal{L} = \mathcal{K}$ 时, 称投影算子 $\boldsymbol{P}$ 为到 $\mathcal{K}$ 上的正交投影算子. 非正交投影算子称为斜投影算子. 对于正交投影算子, 有

$$\mathcal{N}(\boldsymbol{P}) = \mathcal{R}(\boldsymbol{P})^{\perp}.$$

因此, 可由下面对任意向量 $\boldsymbol{x}$ 都要满足的要求来定义一个正交投影算子:

$$\boldsymbol{P}\boldsymbol{x} \in \mathcal{K} \text{ 且 } (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x} \perp \mathcal{K},$$

或等价地, 有

$$\boldsymbol{P}\boldsymbol{x} \in \mathcal{K} \text{ 且 } \big((\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}, \boldsymbol{y}\big) = 0, \quad \forall\, \boldsymbol{y} \in \mathcal{K}.$$

考虑映射 $\boldsymbol{P}$ 的伴随映射 $\boldsymbol{P}^{\mathrm{H}}$, 它定义为

$$(\boldsymbol{P}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}, \boldsymbol{P}\boldsymbol{y}), \quad \forall\, \boldsymbol{x}, \boldsymbol{y}. \tag{2.42}$$

首先注意到 $\boldsymbol{P}^{\mathrm{H}}$ 也是一个投影算子, 因为对所有的 $\boldsymbol{x}$ 和 $\boldsymbol{y}$,

$$((\boldsymbol{P}^{\mathrm{H}})^2\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{P}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{P}\boldsymbol{y}) = (\boldsymbol{x}, \boldsymbol{P}^2\boldsymbol{y}) = (\boldsymbol{x}, \boldsymbol{P}\boldsymbol{y}) = (\boldsymbol{P}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{y}).$$

由式 (2.42), 得

$$\mathcal{N}(\boldsymbol{P}^{\mathrm{H}}) = \mathcal{R}(\boldsymbol{P})^{\perp}, \tag{2.43}$$

$$\mathcal{N}(\boldsymbol{P}) = \mathcal{R}(\boldsymbol{P}^{\mathrm{H}})^{\perp}. \tag{2.44}$$

上述关系导致下面的性质.

性质 2.1 *投影算子是正交的当且仅当它是 Hermite 的.*

任给一个矩阵 $\boldsymbol{V} \in \mathbb{C}^{n\times m}$, 其列满足 $\boldsymbol{v}_i^{\mathrm{H}}\boldsymbol{v}_i = 1 (i = 1, 2, \cdots, m)$. 则其列形成 $\mathcal{K} = \mathcal{R}(\boldsymbol{P})$ 的一组标准正交基, 可用矩阵 $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^{\mathrm{H}}$ 表示 $\boldsymbol{P}$. 这是投影算子矩阵表示 (2.41) 的一种特殊情况. 除了是幂等的, 与此矩阵相应的线性映射满足上述给出的特征, 即

$$\boldsymbol{V}\boldsymbol{V}^{\mathrm{H}}\boldsymbol{x} \in \mathcal{K} \text{ 且 } (\boldsymbol{I}-\boldsymbol{V}\boldsymbol{V}^{\mathrm{H}})\boldsymbol{x} \in \mathcal{K}^{\perp}.$$

重要的是注意到正交投影算子 $\boldsymbol{P}$ 的这种表示不是唯一的. 事实上, 任意标准正交基 $\boldsymbol{V}$ 将给出 $\boldsymbol{P}$ 的一种不同的形式的表示. 因此, 对 $\mathcal{K}$ 的任意两组正交基 $\boldsymbol{V}_1$ 和 $\boldsymbol{V}_2$, 必须有 $\boldsymbol{V}_1\boldsymbol{V}_1^{\mathrm{H}} = \boldsymbol{V}_2\boldsymbol{V}_2^{\mathrm{H}}$, 这一等式可独立地证明.

下面来看看正交投影的性质. 当 $\boldsymbol{P}$ 是一个正交投影算子时, 分解 $\boldsymbol{x} = \boldsymbol{P}\boldsymbol{x}+(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}$ 中的两个向量是正交的, 得到下述关系:

$$\|\boldsymbol{x}\|_2^2 = \|\boldsymbol{P}\boldsymbol{x}\|_2^2 + \|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}\|_2^2.$$

因此, 对任意 $\boldsymbol{x}$, 有

$$\|\boldsymbol{P}\boldsymbol{x}\|_2 \leqslant \|\boldsymbol{x}\|_2.$$

故对所有 $\boldsymbol{x} \in \mathbb{C}^n$, $\|\boldsymbol{P}\boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2$ 的最大值不超过 1, 另外, 对 $\mathcal{R}(\boldsymbol{P})$ 中任意元素达到 1. 所以对任意正交投影算子 $\boldsymbol{P}$, 有

$$\|\boldsymbol{P}\|_2 = 1.$$

正交投影算子只有两个特征值: 0 和 1. $\boldsymbol{P}$ 的值域中的任意向量都是相应于特征值 1 的特征向量, 核中的任意向量都是相应于特征值 0 的特征向量.

下面建立关于正交投影算子的一个重要的最优性性质.

定理 2.13 *设 $\boldsymbol{P}$ 是到子空间 $\mathcal{K}$ 上的正交投影算子, 则对任意给定的向量 $\boldsymbol{x} \in \mathbb{C}^n$, 下式成立:*

$$\min_{\boldsymbol{y}\in\mathcal{K}} \|\boldsymbol{x}-\boldsymbol{y}\|_2 = \|\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}\|_2. \tag{2.45}$$

证明 设 $\boldsymbol{y}$ 为 $\mathcal{K}$ 中任意向量, 考虑它到 $\boldsymbol{x}$ 的距离的平方. 由于 $\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}$ 正交于 $\mathcal{K}$, 而 $\boldsymbol{P}\boldsymbol{x}-\boldsymbol{y}$ 属于 $\mathcal{K}$, 则

$$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2=\|\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}+(\boldsymbol{P}\boldsymbol{x}-\boldsymbol{y})\|_2^2=\|\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}\|_2^2+\|\boldsymbol{P}\boldsymbol{x}-\boldsymbol{y}\|_2^2.$$

因此, 对任意 $\boldsymbol{y}\in\mathcal{K}$, $\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\geqslant\|\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}\|_2^2$. 注意到对 $\boldsymbol{y}=\boldsymbol{P}\boldsymbol{x}$ 达到极小, 这就证明了式 (2.45). 证毕. □

通过对子空间 $\mathcal{K}$ 上的正交投影算子 $\boldsymbol{P}$ 定义 $\boldsymbol{y}^*\equiv\boldsymbol{P}\boldsymbol{x}$, 可将上述结果重新表示成一个充分必要条件的形式, 这个条件可以确定一个向量 $\boldsymbol{x}$ 在最小二乘意义下的最佳逼近.

推论 2.2 设给定一个子空间 $\mathcal{K}$ 和一个向量 $\boldsymbol{x}\in\mathbb{C}^n$, 则

$$\min_{\boldsymbol{y}\in\mathcal{K}}\|\boldsymbol{x}-\boldsymbol{y}\|_2=\|\boldsymbol{x}-\boldsymbol{y}^*\|_2,$$

当且仅当满足下述条件:

$$\begin{cases}\boldsymbol{y}^*\in\mathcal{K},\\ \boldsymbol{x}-\boldsymbol{y}^*\perp\mathcal{K}.\end{cases}$$

2.5.2 投影方法的基本框架

考虑线性方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 其中 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$. 这里, 符号 $\boldsymbol{A}$ 既用来表示矩阵, 也用来表示它所代表的 $\mathbb{R}^n$ 中的线性映射. 投影方法的思想是从 $\mathbb{R}^n$ 的一个子空间推断出问题的一个近似解. 如果 $\mathcal{K}$ 是这种候选近似子空间, 或称“搜索子空间”, 并且维数为 m, 则一般地为推断出这种近似必须加 m 个约束. 描述这些约束的一个典型的方式是强加 m 个正交性条件. 特别地, 残差向量 $\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$ 限制为与 m 个线性无关的向量正交. 这就定义了另一个 m 维子空间 $\mathcal{L}$, 称其为“约束子空间”. 这种简单的框架称为 Petrov–Galerkin 条件, 它在许多不同的数学方法中是很普遍的.

投影方法有两大类: 正交投影方法和斜投影方法. 在正交投影技术中, 子空间 $\mathcal{L}$ 与 $\mathcal{K}$ 是相同的. 在斜投影方法中 $\mathcal{L}$ 与 $\mathcal{K}$ 不相同, 甚至彼此完全无关. 这种差别非常重要并由此给出不同类型的算法.

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 且 $\mathcal{K}$ 和 $\mathcal{L}$ 为 $\mathbb{R}^n$ 的两个 m 维子空间. 正交于 $\mathcal{L}$ 的到 $\mathcal{K}$ 上的投影技术是一个过程: 通过条件 $\widetilde{\boldsymbol{x}}\in\mathcal{K}$ 并且新的残差向量正交于 $\mathcal{L}$ 来求 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的一个近似解 $\widetilde{\boldsymbol{x}}$, 即

$$\text{求 }\widetilde{\boldsymbol{x}}\in\mathcal{K},\ \text{使得 }\boldsymbol{b}-\boldsymbol{A}\widetilde{\boldsymbol{x}}\perp\mathcal{L}.$$

如果希望利用解的初始值 $\boldsymbol{x}^{(0)}$ 的信息, 则必须在仿射空间 $\boldsymbol{x}^{(0)}+\mathcal{K}$ 中而不是在齐次的向量空间 $\mathcal{K}$ 中寻求近似解. 这需要对上述公式进行稍微的修改. 近似解问题需要重新定义为

$$\widetilde{\boldsymbol{x}}\in\boldsymbol{x}^{(0)}+\mathcal{K},\ \text{使得 }\boldsymbol{b}-\boldsymbol{A}\widetilde{\boldsymbol{x}}\perp\mathcal{L}.$$

注意到, 如果将 $\widetilde{\boldsymbol{x}}$ 写成形式 $\widetilde{\boldsymbol{x}}=\boldsymbol{x}^{(0)}+\boldsymbol{z}$, 且初始残差向量 $\boldsymbol{r}^{(0)}$ 定义为 $\boldsymbol{r}^{(0)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(0)}$, 则上述方程变为

$$\boldsymbol{b}-\boldsymbol{A}(\boldsymbol{x}^{(0)}+\boldsymbol{z})\perp\mathcal{L}\ \text{ 或 }\ \boldsymbol{r}^{(0)}-\boldsymbol{A}\boldsymbol{z}\perp\mathcal{L}.$$

换言之, 可定义近似解为

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x}^{(0)} + \boldsymbol{z}, \quad \boldsymbol{z} \in \mathcal{K}, \tag{2.46}$$

$$(\boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{z}, \boldsymbol{w}) = 0, \quad \forall\, \boldsymbol{w} \in \mathcal{L}. \tag{2.47}$$

这是最一般形式的一个基本投影步. 大多数标准技术都连续地使用这种投影. 投影方法为科学计算中许多著名的方法提供了统一的框架. 事实上, 几乎所有的基本迭代法都可视为投影技术. 只要通过 m 个自由度 (子空间 $\mathcal{K}$) 和 m 个约束 (子空间 $\mathcal{L}$) 来定义一个近似, 就可得到一个投影过程.

正交投影方法相应于两个子空间 $\mathcal{K}$ 和 $\mathcal{L}$ 相等的特殊情形. 这种差别在 Heimite 的情形下特别重要, 因为在此情形下, 需要保证被投影的问题是 Heimite 的. 当 $\mathcal{K} = \mathcal{L}$ 时, Petrov–Galerkin 条件也简称为 Galerkin 条件.

设 $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m]$ 是一个 $n \times m$ 阶矩阵, 其列向量形成 $\mathcal{K}$ 的一组基. 类似地, $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]$ 也是一个 $n \times m$ 阶矩阵, 其列向量形成 $\mathcal{L}$ 的一组基. 如果将近似解写成

$$\boldsymbol{x} = \boldsymbol{x}^{(0)} + \boldsymbol{V}\boldsymbol{y},$$

则正交性条件立即导致下面关于向量 $\boldsymbol{y}$ 的方程组:

$$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}\boldsymbol{y} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{r}^{(0)}.$$

如果假设 $m \times m$ 阶矩阵 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 是非奇异的, 则有下述关于近似解 $\boldsymbol{x}$ 的表达式:

$$\boldsymbol{x} = \boldsymbol{x}^{(0)} + \boldsymbol{V}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V})^{-1}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{r}^{(0)}. \tag{2.48}$$

在许多算法中, 没有必要形成矩阵 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$, 因此它可作为对算法的一个"副产品"供使用. 典型投影方法由下述算法表示.

算法 2.15 (投影方法的基本框架)

步 1, 选取一对子空间 $\mathcal{K}$ 和 $\mathcal{L}$.

步 2, 选取 $\mathcal{K}$ 和 $\mathcal{L}$ 的基 $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m]$ 和 $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]$.

步 3, 计算残差 $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$.

步 4, 解方程组 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}\boldsymbol{y} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{r}$ 得到 $\boldsymbol{y}$.

步 5, 计算近似解 $\boldsymbol{x} := \boldsymbol{x} + \boldsymbol{V}\boldsymbol{y}$.

只有当矩阵 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 是非奇异的, 近似解才是有意义的. 即使当 $\boldsymbol{A}$ 是非奇异的, 也不能保证 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 是非奇异的. 看下面的例子.

例 2.8 考虑矩阵

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{I} \end{bmatrix},$$

式中: $\boldsymbol{I}$ 为 m 阶单位矩阵且 $\boldsymbol{O}$ 为 m 阶零矩阵. 设 $\boldsymbol{V} = \boldsymbol{W} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_m]$, 其中 $\boldsymbol{e}_i$ 是第 i 个分量为 1 而其余分量均为 0 的 $2m$ 维列向量. 尽管 $\boldsymbol{A}$ 是非奇异的, 而矩阵 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V} = \boldsymbol{O}$, 因此是奇异的.

下面的命题表明, 有两种重要的特殊情形可以保证 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 的非奇异性.

命题 2.1 设 $\boldsymbol{A}$, $\mathcal{K}$ 和 $\mathcal{L}$ 满足下面两个条件之一:

(1) $\boldsymbol{A}$ 是正定的且 $\mathcal{L}=\mathcal{K}$.

(2) $\boldsymbol{A}$ 是非奇异的且 $\mathcal{L}=\boldsymbol{A}\mathcal{K}$.

则矩阵 $\boldsymbol{B}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 对 $\mathcal{K}$ 和 $\mathcal{L}$ 的任意基 $\boldsymbol{V}$ 和 $\boldsymbol{W}$ 均是非奇异的.

证明 (1) 设 $\boldsymbol{V}$ 和 $\boldsymbol{W}$ 分别为 $\mathcal{K}$ 和 $\mathcal{L}$ 的任意一组基. 事实上, 由于 $\mathcal{L}=\mathcal{K}$, $\boldsymbol{W}$ 总可以表示为 $\boldsymbol{W}=\boldsymbol{V}\boldsymbol{G}$, 其中 $\boldsymbol{G}$ 是一个非奇异 m 阶矩阵, 则

$$\boldsymbol{B}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}=\boldsymbol{G}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}.$$

由于 $\boldsymbol{A}$ 是正定的, 所以 $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 正定, 从而证明了 $\boldsymbol{B}$ 是非奇异的.

(2) 设 $\boldsymbol{V}$ 和 $\boldsymbol{W}$ 分别为 $\mathcal{K}$ 和 $\mathcal{L}$ 的任意一组基. 由于 $\mathcal{L}=\boldsymbol{A}\mathcal{K}$, 此时 $\boldsymbol{W}$ 总可表示为 $\boldsymbol{W}=\boldsymbol{A}\boldsymbol{V}\boldsymbol{G}$, 其中 $\boldsymbol{G}$ 是一个非奇异 m 阶矩阵, 则

$$\boldsymbol{B}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}=\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{V})^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}.$$

由于 $\boldsymbol{A}$ 是非奇异的, $n\times m$ 阶矩阵 $\boldsymbol{A}\boldsymbol{V}$ 是满秩的, 因此 $(\boldsymbol{A}\boldsymbol{V})^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 是非奇异的. 由此及上式证明了 $\boldsymbol{B}$ 是非奇异的. 证毕. □

注 2.14 考虑特殊情形: $\boldsymbol{A}$ 是实对称的且使用正交投影, 此时, 对 $\mathcal{L}$ 和 $\mathcal{K}$ 使用相同的基, 因为它们有相同的子空间, 则投影矩阵 $\boldsymbol{B}=\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 是对称的. 另外, 如果矩阵 $\boldsymbol{A}$ 是对称正定的, 则 $\boldsymbol{B}$ 也是对称正定的.

进一步, 分别考察命题 2.1 中的两个条件所对应的最优性结果.

命题 2.2 假设 $\boldsymbol{A}$ 是对称正定的且 $\mathcal{L}=\mathcal{K}$, 初始向量为 $\boldsymbol{x}^{(0)}$. 则向量 $\widetilde{\boldsymbol{x}}$ 为到 $\mathcal{K}$ 上的一个正交投影方法的结果, 当且仅当 $\widetilde{\boldsymbol{x}}$ 在 $\boldsymbol{x}^{(0)}+\mathcal{K}$ 上极小化误差的 $\boldsymbol{A}$ 范数 (能量范数), 即当且仅当

$$\mathcal{E}(\widetilde{\boldsymbol{x}})=\min_{\boldsymbol{x}\in\boldsymbol{x}^{(0)}+\mathcal{K}}\mathcal{E}(\boldsymbol{x}),$$

式中: $\mathcal{E}(\boldsymbol{x})\equiv\|\boldsymbol{x}^*-\boldsymbol{x}\|_{\boldsymbol{A}}=\sqrt{(\boldsymbol{A}(\boldsymbol{x}^*-\boldsymbol{x}),\boldsymbol{x}^*-\boldsymbol{x})}$.

证明 由推论 2.2, 为使 $\widetilde{\boldsymbol{x}}$ 极小化 $\mathcal{E}(\boldsymbol{x})$, 当且仅当 $\boldsymbol{x}^*-\widetilde{\boldsymbol{x}}$ 与 $\mathcal{K}$ 的所有向量均 $\boldsymbol{A}$ 正交. 由此得

$$(\boldsymbol{A}(\boldsymbol{x}^*-\widetilde{\boldsymbol{x}}),\boldsymbol{v})=0,\quad\forall\,\boldsymbol{v}\in\mathcal{K},$$

或等价地, 有

$$(\boldsymbol{b}-\boldsymbol{A}\widetilde{\boldsymbol{x}},\boldsymbol{v})=0,\quad\forall\,\boldsymbol{v}\in\mathcal{K},$$

这正是对近似解 $\widetilde{\boldsymbol{x}}$ 定义一个正交投影的 Galerkin 条件. 证毕. □

命题 2.3 设 $\boldsymbol{A}$ 是任意方阵且 $\mathcal{L}=\boldsymbol{A}\mathcal{K}$, 初始向量为 $\boldsymbol{x}^{(0)}$. 则向量 $\widetilde{\boldsymbol{x}}$ 为正交于 $\mathcal{L}$ 的到 $\mathcal{K}$ 上一个斜投影方法的结果, 当且仅当 $\widetilde{\boldsymbol{x}}$ 在 $\boldsymbol{x}\in\boldsymbol{x}^{(0)}+\mathcal{K}$ 上极小化残差向量 $\boldsymbol{r}(\boldsymbol{x})=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$ 的 2-范数, 即当且仅当

$$\|\boldsymbol{r}(\widetilde{\boldsymbol{x}})\|_2=\min_{\boldsymbol{x}\in\boldsymbol{x}^{(0)}+\mathcal{K}}\|\boldsymbol{r}(\boldsymbol{x})\|_2.$$

证明 如前所见, 为使 $\widetilde{\boldsymbol{x}}$ 极小化 $\|\boldsymbol{r}(\boldsymbol{x})\|_2$, 其充分必要条件是 $\boldsymbol{b}-\boldsymbol{A}\widetilde{\boldsymbol{x}}$ 与所有形如 $\boldsymbol{v}=\boldsymbol{A}\boldsymbol{y}$ 的向量均正交, 其中 $\boldsymbol{y}\in\mathcal{K}$. 亦即

$$(\boldsymbol{b}-\boldsymbol{A}\widetilde{\boldsymbol{x}},\boldsymbol{v})=0,\quad \forall\,\boldsymbol{v}\in\boldsymbol{A}\mathcal{K},$$

这正是定义近似解 $\widetilde{\boldsymbol{x}}$ 的 Petrov–Galerkin 条件. 证毕. □

值得指出的是, 上述命题中 $\boldsymbol{A}$ 不需要是非奇异的. 当 $\boldsymbol{A}$ 为奇异时, 可能有无穷多个向量 $\widetilde{\boldsymbol{x}}$ 满足最优性条件.

下面给出用投影算子术语的解释. 对 $\mathcal{L}=\mathcal{K}$ 和 $\mathcal{L}=\boldsymbol{A}\mathcal{K}$ 的情形, 可容易地用在初始残差或初始误差上正交投影算子的作用来解释投影方法的结果. 先考虑 $\mathcal{L}=\boldsymbol{A}\mathcal{K}$ 的情形, 设初始残差为 $\boldsymbol{r}^{(0)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(0)}$, 且 $\widetilde{\boldsymbol{r}}=\boldsymbol{b}-\boldsymbol{A}\widetilde{\boldsymbol{x}}$ 为投影过程之后所得的残差, 则

$$\widetilde{\boldsymbol{r}}=\boldsymbol{b}-\boldsymbol{A}(\boldsymbol{x}^{(0)}+\boldsymbol{z})=\boldsymbol{r}^{(0)}-\boldsymbol{A}\boldsymbol{z}.$$

其中 $\boldsymbol{z}$ 由强加条件 $\boldsymbol{r}^{(0)}-\boldsymbol{A}\boldsymbol{z}\perp\boldsymbol{A}\mathcal{K}$ 而得到. 因此, 向量 $\boldsymbol{A}\boldsymbol{z}$ 是向量 $\boldsymbol{r}^{(0)}$ 到子空间 $\boldsymbol{A}\mathcal{K}$ 上的正交投影. 从而有下面的命题.

命题 2.4 *设 $\widetilde{\boldsymbol{x}}$ 为由正交于 $\mathcal{L}=\boldsymbol{A}\mathcal{K}$ 的到 $\mathcal{K}$ 上的一个投影过程所得的近似解, 且设 $\widetilde{\boldsymbol{r}}=\boldsymbol{b}-\boldsymbol{A}\widetilde{\boldsymbol{x}}$ 为相应的残差, 则*

$$\widetilde{\boldsymbol{r}}=(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{r}^{(0)},\tag{2.49}$$

式中: $\boldsymbol{P}$ 为子空间 $\boldsymbol{A}\mathcal{K}$ 上的正交投影算子.

命题 2.4 说明一个投影步后所得残差的 2–范数将不超过初始残差的 2–范数, 即

$$\|\widetilde{\boldsymbol{r}}\|_2\leqslant\|\boldsymbol{r}^{(0)}\|_2,$$

这是一个已经建立了的结果. 这类方法称为*残差投影方法*.

现在考虑 $\mathcal{L}=\mathcal{K}$ 且 $\boldsymbol{A}$ 是对称正定的情形. 设 $\boldsymbol{e}^{(0)}=\boldsymbol{x}^*-\boldsymbol{x}^{(0)}$ 为初始误差, 类似地, 设 $\widetilde{\boldsymbol{e}}=\boldsymbol{x}^*-\widetilde{\boldsymbol{x}}$, 其中 $\widetilde{\boldsymbol{x}}=\boldsymbol{x}^{(0)}+\boldsymbol{z}$ 为由投影步所得的近似解. 由式 (2.49), 得

$$\boldsymbol{A}\widetilde{\boldsymbol{e}}=\boldsymbol{A}(\boldsymbol{x}^*-\widetilde{\boldsymbol{x}})=\widetilde{\boldsymbol{r}}=\boldsymbol{A}(\boldsymbol{e}^{(0)}-\boldsymbol{z})=\boldsymbol{r}^{(0)}-\boldsymbol{A}\boldsymbol{z},$$

式中: $\boldsymbol{z}$ 是由限制残差向量 $\boldsymbol{r}^{(0)}-\boldsymbol{A}\boldsymbol{z}\perp\mathcal{K}$ 而得到, 即

$$(\boldsymbol{r}^{(0)}-\boldsymbol{A}\boldsymbol{z},\boldsymbol{w})=0,\quad\forall\,\boldsymbol{w}\in\mathcal{K}\iff(\boldsymbol{A}(\boldsymbol{e}^{(0)}-\boldsymbol{z}),\boldsymbol{w})=0,\quad\forall\,\boldsymbol{w}\in\mathcal{K}.$$

由于 $\boldsymbol{A}$ 是对称正定的, 它定义了一种通常表示为 $(\cdot,\cdot)_{\boldsymbol{A}}$ 的内积, 则条件变为

$$(\boldsymbol{e}^{(0)}-\boldsymbol{z},\boldsymbol{w})_{\boldsymbol{A}}=0,\quad\forall\,\boldsymbol{w}\in\mathcal{K}.$$

此条件可解释为: 向量 $\boldsymbol{z}$ 是初始误差向量 $\boldsymbol{e}^{(0)}$ 到子空间 $\mathcal{K}$ 上的 $\boldsymbol{A}$ 正交投影.

命题 2.5 *设 $\widetilde{\boldsymbol{x}}$ 为到 $\mathcal{K}$ 上的一个正交投影过程所得的近似解, 且设 $\widetilde{\boldsymbol{e}}=\boldsymbol{x}^*-\widetilde{\boldsymbol{x}}$ 为相应的误差, 则*

$$\widetilde{\boldsymbol{e}}=(\boldsymbol{I}-\boldsymbol{P}_{\boldsymbol{A}})\boldsymbol{e}^{(0)},$$

式中: $\boldsymbol{P}_{\boldsymbol{A}}$ 为子空间 $\mathcal{K}$ 上的投影算子, 它关于 $\boldsymbol{A}$ 内积是正交的.

命题 2.5 表明一个投影步后所得的误差向量的 $\boldsymbol{A}$ 范数不超过初始误差的 $\boldsymbol{A}$ 范数, 即

$$\|\widetilde{\boldsymbol{e}}\|_{\boldsymbol{A}} \leqslant \|\boldsymbol{e}^{(0)}\|_{\boldsymbol{A}},$$

这正是所期望的, 因为已知误差的 $\boldsymbol{A}$ 范数在 $\boldsymbol{x}^{(0)}+\mathcal{K}$ 中被极小化. 这类方法称为*误差投影方法*.

2.5.3 一维投影方法

这里给出几种基于一维投影过程的投影方法. 以下用 $\boldsymbol{r}$ 表示当前近似解 $\boldsymbol{x}$ 的残差向量 $\boldsymbol{r}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$. 为避免下标, 用 ":=" 表示向量校正, 即 $\boldsymbol{x}:=\boldsymbol{x}+\alpha\boldsymbol{r}$ 意味着 "计算 $\boldsymbol{x}+\alpha\boldsymbol{r}$ 并用结果覆盖当前的 $\boldsymbol{x}$".

一维投影过程可由下面定义, 即

$$\mathcal{K}=\operatorname{span}\{\boldsymbol{v}\} \quad 且 \quad \mathcal{L}=\operatorname{span}\{\boldsymbol{w}\},$$

式中: $\boldsymbol{v}$ 和 $\boldsymbol{w}$ 为两个向量.

此时, 新的近似取为 $\boldsymbol{x}:=\boldsymbol{x}+\alpha\boldsymbol{v}$ 且 Petrov–Galerkin 条件 $\boldsymbol{r}-\boldsymbol{A}\boldsymbol{z}\perp\boldsymbol{w}$ 导致

$$\alpha=\frac{(\boldsymbol{r},\boldsymbol{w})}{(\boldsymbol{A}\boldsymbol{v},\boldsymbol{w})}.$$

下面考虑三种流行的选择.

1) *最速下降法*

最速下降法对矩阵 $\boldsymbol{A}$ 为对称正定的情形来定义. 它由每一步取 $\boldsymbol{v}=\boldsymbol{r}$ 和 $\boldsymbol{w}=\boldsymbol{r}$ 组成 (即一维正交投影). 这导致下面描述的一个迭代.

算法 2.16 (最速下降法)

步 1, $\boldsymbol{r}:=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$.

步 2, $\alpha\leftarrow(\boldsymbol{r},\boldsymbol{r})/(\boldsymbol{A}\boldsymbol{r},\boldsymbol{r})$.

步 3, $\boldsymbol{x}:=\boldsymbol{x}+\alpha\boldsymbol{r}$.

此迭代的每一步在所有形式为 $\boldsymbol{x}+\alpha\boldsymbol{r}$ 的向量上极小化

$$f(\boldsymbol{x})=\|\boldsymbol{x}-\boldsymbol{x}^*\|_{\boldsymbol{A}}^2=(\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^*),\boldsymbol{x}-\boldsymbol{x}^*),$$

式中: $\boldsymbol{r}$ 为负梯度方向 $-\nabla f(\boldsymbol{x})$.

负梯度方向是使 $f(\boldsymbol{x})$ 局部最快速下降的方向. 其次, 证明当 $\boldsymbol{A}$ 是对称正定的时, 收敛性是有保证的. 它是下面著名的 Kantorovich (康托洛维奇) 不等式的一个结论.

引理 2.2 (Kantorovich 不等式) *设 $\boldsymbol{B}$ 为任意对称正定实矩阵且 $\lambda_{\max}$, $\lambda_{\min}$ 为其最大、最小特征值, 则*

$$\frac{(\boldsymbol{B}\boldsymbol{x},\boldsymbol{x})(\boldsymbol{B}^{-1}\boldsymbol{x},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})^2}\leqslant\frac{(\lambda_{\max}+\lambda_{\min})^2}{4\lambda_{\max}\lambda_{\min}},\quad\forall\,\boldsymbol{x}\neq\boldsymbol{0}. \tag{2.50}$$

定理 2.14 设 $\boldsymbol{A}$ 是对称正定的. 则由算法 2.16 生成的误差向量 $\boldsymbol{e}^{(k)}=\boldsymbol{x}^*-\boldsymbol{x}^{(k)}$ 的 $\boldsymbol{A}$ 范数满足关系

$$\|\boldsymbol{e}^{(k+1)}\|_{\boldsymbol{A}}\leqslant\frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}\|\boldsymbol{e}^{(k)}\|_{\boldsymbol{A}},\tag{2.51}$$

且算法 2.16 对任意初始向量 $\boldsymbol{x}^{(0)}$ 都是收敛的.

证明 首先注意到 $\|\boldsymbol{e}^{(k+1)}\|_{\boldsymbol{A}}^2=(\boldsymbol{A}\boldsymbol{e}^{(k+1)},\boldsymbol{e}^{(k+1)})=(\boldsymbol{r}^{(k+1)},\boldsymbol{e}^{(k+1)})$, 则由简单的代入有

$$\|\boldsymbol{e}^{(k+1)}\|_{\boldsymbol{A}}^2=(\boldsymbol{r}^{(k+1)},\boldsymbol{e}^{(k)}-\alpha_k\boldsymbol{r}^{(k)}).$$

由于新残差向量 $\boldsymbol{r}^{(k+1)}$ 必须正交于搜索方向 $\boldsymbol{r}^{(k)}$, 上式右端项中第 2 项为零. 因此

$$\begin{aligned}\|\boldsymbol{e}^{(k+1)}\|_{\boldsymbol{A}}^2&=(\boldsymbol{r}^{(k)}-\alpha_k\boldsymbol{A}\boldsymbol{r}^{(k)},\boldsymbol{e}^{(k)})\\&=(\boldsymbol{r}^{(k)},\boldsymbol{A}^{-1}\boldsymbol{r}^{(k)})-\alpha_k(\boldsymbol{r}^{(k)},\boldsymbol{r}^{(k)})\\&=\|\boldsymbol{e}^{(k)}\|_{\boldsymbol{A}}^2\left(1-\frac{(\boldsymbol{r}^{(k)},\boldsymbol{r}^{(k)})}{(\boldsymbol{r}^{(k)},\boldsymbol{A}\boldsymbol{r}^{(k)})}\cdot\frac{(\boldsymbol{r}^{(k)},\boldsymbol{r}^{(k)})}{(\boldsymbol{r}^{(k)},\boldsymbol{A}^{-1}\boldsymbol{r}^{(k)})}\right).\end{aligned}$$

应用 Kantorovich 不等式 (2.50) 即得定理的结论. 证毕. □

2) 极小残差 (MR) 法

现在假设 $\boldsymbol{A}$ 不必为对称的而只是正定的, 即其对称部分 $\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}$ 是对称正定的. 在每一步取 $\boldsymbol{v}=\boldsymbol{r}$ 和 $\boldsymbol{w}=\boldsymbol{A}\boldsymbol{r}$, 可得下述迭代过程.

算法 2.17 (极小残差法)

步 1, $\boldsymbol{r}:=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$.

步 2, $\alpha:=(\boldsymbol{A}\boldsymbol{r},\boldsymbol{r})/(\boldsymbol{A}\boldsymbol{r},\boldsymbol{A}\boldsymbol{r})$.

步 3, $\boldsymbol{x}:=\boldsymbol{x}+\alpha\boldsymbol{r}$.

这里, 每一步在方向 $\boldsymbol{r}$ 上极小化 $\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2^2$. 如下定理所述, 迭代在 $\boldsymbol{A}$ 为正定时收敛.

定理 2.15 设 $\boldsymbol{A}$ 为实正定矩阵, 且设

$$\mu=\lambda_{\min}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})/2,\quad\sigma=\|\boldsymbol{A}\|_2.$$

则由算法 2.17 产生的残差向量满足关系

$$\|\boldsymbol{r}^{(k+1)}\|_2\leqslant\left(1-\frac{\mu^2}{\sigma^2}\right)^{1/2}\|\boldsymbol{r}^{(k)}\|_2,\tag{2.52}$$

且算法 2.17 对任意初始向量 $\boldsymbol{x}^{(0)}$ 均收敛.

证明 类似于最速下降法来处理, 有关系式

$$\|\boldsymbol{r}^{(k+1)}\|_2^2=(\boldsymbol{r}^{(k)}-\alpha_k\boldsymbol{A}\boldsymbol{r}^{(k)},\boldsymbol{r}^{(k)}-\alpha_k\boldsymbol{A}\boldsymbol{r}^{(k)})$$

$$= (\boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)}) - \alpha_k (\boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{A}\boldsymbol{r}^{(k)}).$$

由构造, 新的残差向量 $\boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{A}\boldsymbol{r}^{(k)}$ 必须正交于搜索方向 $\boldsymbol{A}\boldsymbol{r}^{(k)}$, 因此, 上式右端项中第 2 项为零, 且得

$$\begin{aligned}
\|\boldsymbol{r}^{(k+1)}\|_2^2 &= (\boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)}) = (\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)}) - \alpha_k (\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)}) \\
&= \|\boldsymbol{r}^{(k)}\|_2^2 \left(1 - \frac{(\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})}{(\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})} \cdot \frac{(\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})}{(\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{A}\boldsymbol{r}^{(k)})}\right) \\
&= \|\boldsymbol{r}^{(k)}\|_2^2 \left(1 - \frac{(\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})^2}{(\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})^2} \cdot \frac{\|\boldsymbol{r}^{(k)}\|_2^2}{\|\boldsymbol{A}\boldsymbol{r}^{(k)}\|_2^2}\right). \qquad (2.53)
\end{aligned}$$

因为 $\boldsymbol{A}$ 为实正定矩阵, 从而有

$$\frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} = \frac{1}{2} \frac{((\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \geqslant \mu > 0, \qquad (2.54)$$

式中: $\mu = \lambda_{\min}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})/2$.

由不等式 $\|\boldsymbol{A}\boldsymbol{r}^{(k)}\|_2 \leqslant \|\boldsymbol{A}\|_2 \|\boldsymbol{r}^{(k)}\|_2$ 立即可得定理的结论. 证毕. □

3) 残差范数最速下降法

在残差范数最速下降法中, 只需 $\boldsymbol{A}$ 为非奇异矩阵. 在每一步, 算法使用 $\boldsymbol{v} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}$ 和 $\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}$, 给出下面运算序列:

$$\begin{aligned}
&\boldsymbol{r} := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{v} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}, \\
&\alpha := \|\boldsymbol{v}\|_2^2 / \|\boldsymbol{A}\boldsymbol{v}\|_2^2, \\
&\boldsymbol{x} := \boldsymbol{x} + \alpha \boldsymbol{v}.
\end{aligned}$$

注意到基于上述运算序列的算法需要三个矩阵向量乘积, 这是本节其他算法的 3 倍. 通过不同的方式计算残差, 每步中矩阵向量乘积的个数可降为两个. 这种变形如下:

算法 2.18 (残差范数最速下降)

步 1, 计算 $\boldsymbol{r} := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$.

步 2, $\boldsymbol{v} := \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}$.

步 3, 计算 $\boldsymbol{A}\boldsymbol{v}$ 和 $\alpha := \|\boldsymbol{v}\|_2^2 / \|\boldsymbol{A}\boldsymbol{v}\|_2^2$.

步 4, $\boldsymbol{x} := \boldsymbol{x} + \alpha \boldsymbol{r}$.

步 5, $\boldsymbol{r} := \boldsymbol{r} - \alpha \boldsymbol{A}\boldsymbol{v}$.

这里, 每一步在方向 $-\nabla f(\boldsymbol{x})$ 上极小化 $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2$. 由此得出, 这等价于最速下降算法应用于法方程. 因为当 $\boldsymbol{A}$ 是非奇异时, $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ 是正定的, 则根据定理 2.15, 只要 $\boldsymbol{A}$ 是非奇异的, 方法将收敛.

习题 2

2.1 设向量 $\boldsymbol{x}=(2,4,0,5,1,3)^{\mathrm{T}}$. 求一个 Householder 变换 $\boldsymbol{H}$ 和一个正常数 α, 使得 $\boldsymbol{H}\boldsymbol{x}=(2,4,\alpha,5,0,0)^{\mathrm{T}}$.

2.2 假定 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 是 $\mathbb{R}^n$ 中满足 $\|\boldsymbol{x}\|_2=\|\boldsymbol{y}\|_2=1$ 的两个向量. 给出一种使用 Givens 变换的算法, 构造一个正交矩阵 $\boldsymbol{Q}$, 使得 $\boldsymbol{Q}\boldsymbol{x}=\boldsymbol{y}$.

2.3 假定 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 是 $\mathbb{R}^n$ 中的两个非零向量. 给出一种算法来构造一个 Householder 矩阵 $\boldsymbol{H}$, 使得 $\boldsymbol{H}\boldsymbol{x}=\alpha\boldsymbol{y}$, 其中 $\alpha\in\mathbb{R}$.

2.4 设矩阵

$$\boldsymbol{A}=\begin{bmatrix}0&4&1\\1&1&1\\0&3&2\end{bmatrix},$$

利用 Householder 变换求 $\boldsymbol{A}$ 的 QR 分解.

2.5 设 $\boldsymbol{H}\in\mathbb{R}^{n\times n}$ 是上 Hessenberg 矩阵, $\boldsymbol{R}\in\mathbb{R}^{n\times n}$ 是非奇异的上三角矩阵. 证明: $\boldsymbol{R}\boldsymbol{H}\boldsymbol{R}^{-1}$ 仍是上 Hessenberg 矩阵.

2.6 设

$$\boldsymbol{A}=\begin{bmatrix}2&2&1\\0&2&2\\2&1&2\end{bmatrix},$$

利用 Givens 变换求矩阵 $\boldsymbol{A}$ 的 QR 分解.

2.7 设矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 可以分解为 $\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R}$, 其中 $\boldsymbol{Q}$ 为正交矩阵, $\boldsymbol{R}$ 是对角元全为正的上三角矩阵. 记 $\boldsymbol{A}$ 的第 i 列为 $\boldsymbol{a}_i$. 证明:

$$|\det(\boldsymbol{A})|\leqslant\prod_{i=1}^{n}\|\boldsymbol{a}_i\|_2.$$

2.8 设 $\boldsymbol{A}$ 为 $n\times n$ 非奇异实矩阵, 其 QR 分解为 $\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R}$. 记 $\boldsymbol{B}=\boldsymbol{R}\boldsymbol{Q}$, 证明:

(1) 若 $\boldsymbol{A}$ 是对称的, 则 $\boldsymbol{B}$ 也是对称的;

(2) 若 $\boldsymbol{A}$ 是上 Hessenberg 矩阵, 则 $\boldsymbol{B}$ 也是上 Hessenberg 矩阵.

2.9 假设 $\boldsymbol{v}=p(\boldsymbol{A})\boldsymbol{u}$, 其中 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, $\boldsymbol{u},\boldsymbol{v}\in\mathbb{R}^n$, p 是多项式. 证明: 对每个 $k=1,2,\cdots,n$, 都有 $p(\boldsymbol{A})\mathcal{K}_k(\boldsymbol{A},\boldsymbol{u})=\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})$.

2.10 设 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k$ 是矩阵 $\boldsymbol{A}$ 的 k 个线性无关的特征向量, 令

$$\boldsymbol{v}=\alpha_1\boldsymbol{x}_1+\alpha_2\boldsymbol{x}_2+\cdots+\alpha_k\boldsymbol{x}_k,$$

其中系数 $\alpha_1,\alpha_2,\cdots,\alpha_k$ 都是非零的. 证明:

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})=\operatorname{span}\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k\}.$$

2.11 证明: 任一正交投影算子 $\boldsymbol{P}$ 均可表示为 $\boldsymbol{P}=\boldsymbol{V}\boldsymbol{V}^{\mathrm{H}}$, 其中 $\boldsymbol{V}$ 满足 $\boldsymbol{V}^{\mathrm{H}}\boldsymbol{V}=\boldsymbol{I}$.

2.12 设 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 是从 $\mathbb{C}^n$ 到其自身的两个投影算子. 证明:

(1) $\boldsymbol{A}+\boldsymbol{B}$ 是投影算子当且仅当 $\boldsymbol{BA}=\boldsymbol{AB}=\boldsymbol{O}$;

(2) 若 $\boldsymbol{AB}=\boldsymbol{BA}$, 则 $\boldsymbol{A}+\boldsymbol{B}-\boldsymbol{AB}$ 也是投影算子.

2.13 考虑如下形式的矩阵

$$\boldsymbol{A}=\boldsymbol{I}-\alpha\boldsymbol{B},$$

式中: α 为实数; $\boldsymbol{B}\in\mathbb{R}^{n\times n}$ 为反对称矩阵, 即 $\boldsymbol{B}^{\mathrm{T}}=-\boldsymbol{B}$. 证明:

(1) 对任意的非零向量 $\boldsymbol{x}$, 有 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Ax}/\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}=1$;

(2) 在 Arnoldi 分解中, 关于 $\boldsymbol{V}_k$ 的 Rayleigh 商 $\boldsymbol{H}_k=\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{AV}_k$ 的形式为

$$\boldsymbol{H}_k=\begin{bmatrix}1 & -\beta_1 & & \\ \beta_1 & 1 & \ddots & \\ & \ddots & \ddots & -\beta_{k-1}\\ & & \beta_{k-1} & 1\end{bmatrix}.$$

2.14 证明 Kantorovich 不等式 (2.50).

第 3 章　线性方程组的矩阵分裂迭代法

在科学与工程计算中, 很多问题往往最终归结为求解一个线性方程组问题, 如结构分析、网络分析、大地测量、数据分析, 以及用有限差分法或有限元法求解微分方程边值问题等. 因此, 研究求解大规模线性方程组快速、稳定的数值算法已成为当前科学与工程计算的核心问题之一. 本章主要讨论求解线性方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \tag{3.1}$$

的迭代方法, 其中 $\boldsymbol{A}$ 是 n 阶可逆矩阵, $\boldsymbol{b}$ 是 n 维列向量. 在实际应用中, 矩阵 $\boldsymbol{A}$ 的阶数 n 一般很大, 通常采用迭代法来求解. 迭代法一般只涉及矩阵与向量的乘法运算. 本章介绍迭代法的一些基本理论, 以及基于矩阵分裂的几种经典迭代法, 如 Jacobi 迭代法、Gauss–Seidel 迭代法以及松弛型迭代法、HSS 分裂迭代法等. 当迭代矩阵的谱半径接近 1 时, 迭代的收敛速度变得很缓慢, 此时可用加速技巧对这些迭代法进行加速.

3.1　迭代法的一般理论

3.1.1　迭代法的定义与分类

迭代法本质上是一个递推公式, 或者说是基于算子的重复利用. 一般地, 求解式 (3.1) 的迭代法定义如下.

定义 3.1　求解式 (3.1) 的迭代法就是寻找一个式 (3.1) 的近似解序列 $\left\{\boldsymbol{x}^{(k)}\right\}_{k=0}^{\infty}$, 使得

$$\begin{cases}\boldsymbol{x}^{(0)}=\phi_0(\boldsymbol{A},\boldsymbol{b}),\\ \boldsymbol{x}^{(k)}=\phi_k(\boldsymbol{x}^{(0)},\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(k-1)};\boldsymbol{A},\boldsymbol{b}),\quad k=1,2,\cdots,\end{cases} \tag{3.2}$$

式中: $\left\{\boldsymbol{x}^{(k)}\right\}_{k=0}^{\infty}$ 称为迭代序列; $\left\{\phi_k\right\}_{k=0}^{\infty}$ 称为迭代算子序列; k 为迭代指标或迭代步数.

定义 3.2　若对某个整数 $s>0$, $k\geqslant s$ 时, ϕ_k 与 k 无关, 则称此迭代法为定常的, 否则称为非定常的. 若对任意的 k, ϕ_k 是 $\boldsymbol{x}^{(0)},\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(k-1)}$ 的线性函数, 则称迭代法是线性的, 否则称为非线性的.

按照定义 3.2, 可以将迭代法分为线性定常迭代法、线性非定常迭代法、非线性定常迭代法、非线性非定常迭代法四类.

构造迭代法的一个基本思路是将式 (3.1) 变成如下形式的同解不动点方程组:

$$\boldsymbol{x}=\boldsymbol{B}\boldsymbol{x}+\boldsymbol{f}, \tag{3.3}$$

然后任取一个初始点 $\boldsymbol{x}^{(0)}$, 由迭代公式

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}\boldsymbol{x}^{(k)}+\boldsymbol{f},\quad k=0,1,\cdots \tag{3.4}$$

产生迭代序列 $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$, 其中 $\boldsymbol{B}$ 称为迭代矩阵. 当然, 若 $\boldsymbol{B}$ 与迭代指标 k 有关, 则迭代为非定常迭代, 否则为定常迭代. 可以构造各种可能的迭代矩阵 $\boldsymbol{B}$, 但必须保证迭代序列收敛 (收敛性问题), 且收敛到方程组 (3.1) 的解 $\boldsymbol{x}^*$ (相容性问题), 并由此确定出有效的迭代法. 一般地, 迭代序列的收敛过程是无限的, 而实际计算中只能且只需得到满足精度要求的某个近似解, 因此还要求适当地选取收敛准则, 这样, 一个迭代法才算完整.

3.1.2 收敛性与收敛速度

设 $\{\boldsymbol{x}^{(k)}\}$ 是 $\mathbb{C}^n$ 中的某个向量序列, $\boldsymbol{x}^* \in \mathbb{C}^n$ 是某个向量, 有如下定义:

定义 3.3 若按某种范数有 $\lim\limits_{k\to\infty} \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\| = 0$, 则称向量序列收敛于 $\boldsymbol{x}^*$, 并记为 $\lim\limits_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}^*$.

事实上, $\lim\limits_{k\to\infty} \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\| = 0$ 当且仅当对 $i = 1, 2, \cdots, n$, $\lim\limits_{k\to\infty} x_i^{(k)} = x_i^*$.

记线性方程组 (3.1) 的精确解为 $\boldsymbol{x}^*$, 第 k 步迭代的误差向量记为 $\boldsymbol{e}^{(k)} = \boldsymbol{x}^* - \boldsymbol{x}^{(k)}$, 相应的残差向量记为 $\boldsymbol{r}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}$.

定义 3.4 设 $\{\boldsymbol{x}^{(k)}\}$ 是某个迭代法产生的迭代序列, 若对任意的初始点 $\boldsymbol{x}^{(0)}$, 序列 $\{\boldsymbol{x}^{(k)}\}$ 均收敛到一个与 $\boldsymbol{x}^{(0)}$ 无关的极限, 则称此迭代法是收敛的. 若在精确运算下, 存在某个正整数 l, 使得 $\boldsymbol{x}^{(l)} = \boldsymbol{x}^*$, 则称此迭代法是精确收敛的或有限终止的.

下面给出收敛速度的概念. 注意到迭代法 (3.4) 的误差传播方程为

$$\boldsymbol{e}^{(k)} = \boldsymbol{B}\boldsymbol{e}^{(k-1)} = \cdots = \boldsymbol{B}^k\boldsymbol{e}^{(0)}, \tag{3.5}$$

这里, $\boldsymbol{e}^{(0)} = \boldsymbol{x}^{(0)} - \boldsymbol{x}^*$ 是解的初始近似 $\boldsymbol{x}^{(0)}$ 与精确解的误差. 注意到迭代法 (3.4) 的误差向量 $\boldsymbol{e}^{(k)}$ 与右端项 $\boldsymbol{f}$ 无关, 故其收敛性和收敛速度完全由迭代矩阵 $\boldsymbol{B}$ 确定.

引进误差向量后, 迭代的收敛问题就等价于误差向量序列收敛于 $\boldsymbol{0}$ 的问题.

欲使迭代法 (3.4) 对任意的初始向量 $\boldsymbol{x}^{(0)}$ 都收敛, 误差向量 $\boldsymbol{e}^{(k)}$ 应对任意的初始误差 $\boldsymbol{e}^{(0)}$ 都收敛于零向量. 于是迭代法 (3.4) 对任意的初始向量都收敛的充分必要条件是

$$\lim_{k\to\infty} \boldsymbol{B}^k = \boldsymbol{O}. \tag{3.6}$$

定理 3.1 迭代法 (3.4) 对任意的初始向量 $\boldsymbol{x}^{(0)}$ 都收敛的充分必要条件是 $\rho(\boldsymbol{B}) < 1$, 这里 $\rho(\boldsymbol{B})$ 表示 $\boldsymbol{B}$ 的谱半径.

证明 必要性. 设对初始向量 $\boldsymbol{x}^{(0)}$, 迭代法 (3.4) 是收敛的, 那么式 (3.6) 成立. 由定理 1.17(1), 对于任意的矩阵范数, 有

$$\rho(\boldsymbol{B}) \leqslant \|\boldsymbol{B}\|.$$

若 $\rho(\boldsymbol{B}) < 1$ 不成立, 即 $\rho(\boldsymbol{B}) \geqslant 1$, 则

$$\|\boldsymbol{B}^k\| \geqslant \rho(\boldsymbol{B}^k) = [\rho(\boldsymbol{B})]^k \geqslant 1,$$

这与式 (3.6) 矛盾.

充分性. 若 $\rho(\boldsymbol{B})<1$, 则存在一个正数 ε, 使得

$$\rho(\boldsymbol{B})+2\varepsilon<1.$$

根据定理 1.17 (2), 存在一种矩阵范数 $\|\boldsymbol{B}\|_\varepsilon$, 使

$$\|\boldsymbol{B}\|_\varepsilon<\rho(\boldsymbol{B})+\varepsilon<1-\varepsilon.$$

故得

$$\|\boldsymbol{B}^k\|_\varepsilon\leqslant\|\boldsymbol{B}\|_\varepsilon^k<(1-\varepsilon)^k,$$

从而当 $k\to\infty$ 时, $\|\boldsymbol{B}^k\|_\varepsilon\to0$, 即 $\boldsymbol{B}^k\to0$. 证毕. □

由此可见, 迭代是否收敛仅与迭代矩阵的谱半径有关, 即仅与方程组的系数矩阵和迭代格式的构造有关, 而与方程组的右端向量 $\boldsymbol{b}$ 及初始向量 $\boldsymbol{x}^{(0)}$ 无关.

推论 3.1 *若迭代矩阵 $\boldsymbol{B}$ 的某种算子范数 $\|\boldsymbol{B}\|_p<1$, 则迭代法 (3.4) 收敛.*

设 $\boldsymbol{B}$ 有 n 个线性无关的特征向量 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n$ (可作为 n 维线性空间的一组基), 相应的特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$. 那么在这组基下, 初始误差向量 $\boldsymbol{e}^{(0)}$ 可以线性表示为

$$\boldsymbol{e}^{(0)}=\sum_{i=1}^n\alpha_i\boldsymbol{u}_i,$$

将其代入式 (3.5), 得

$$\boldsymbol{e}^{(k)}=\boldsymbol{B}^k\boldsymbol{e}^{(0)}=\sum_{i=1}^n\alpha_i\boldsymbol{B}^k\boldsymbol{u}_i=\sum_{i=1}^n\alpha_i\lambda_i^k\boldsymbol{u}_i.\tag{3.7}$$

故当 $\rho(\boldsymbol{B})<1$ 时, 其值越小, 迭代法收敛越快. 因此, 用 $\rho(\boldsymbol{B})$ 来刻画迭代法 (3.4) 的收敛速度是合适的.

另外, 对任何向量范数及相应的矩阵范数, 根据式 (3.5), 利用范数的性质, 有

$$\|\boldsymbol{e}^{(k)}\|_p\leqslant\|\boldsymbol{B}^k\|_p\|\boldsymbol{e}^{(0)}\|_p.$$

可见, $\|\boldsymbol{B}^k\|_p\,(<1)$ 也给出了一种尺度, 它表示经 k 次迭代后误差向量范数减少的程度. 由此可定义*平均收敛速度*为

$$R_k(\boldsymbol{B})=-\frac{1}{k}\log_{10}\|\boldsymbol{B}^k\|_p.$$

可以证明, 若 $\rho(\boldsymbol{B})<1$, 则有

$$\lim_{k\to\infty}\left(\|\boldsymbol{B}^k\|_p\right)^{\frac{1}{k}}=\rho(\boldsymbol{B}).$$

由于这一性质, 可定义*渐近收敛速度*为

$$R(\boldsymbol{B})=\lim_{k\to\infty}R_k(\boldsymbol{B})=-\log_{10}\rho(\boldsymbol{B}).$$

根据这个定义, $\rho(\boldsymbol{B})$ 越小, $R(\boldsymbol{B})$ 越大, 迭代法收敛越快. 值得注意的是, 尽管 $R_k(\boldsymbol{B})$ 依赖于所用的范数, 但 $R(\boldsymbol{B})$ 却与的无关. 因此, 通常将 $R(\boldsymbol{B})$ 称为收敛速度.

如果要使误差向量的范数减少一个因子 $10^{-k_1}(k_1>0)$, 需迭代 k 次, 则可粗略求得 k 是满足不等式

$$k \geqslant k_1/R(\boldsymbol{B})$$

的最小正整数, 即所要迭代次数与收敛速度成反比.

迭代法 (3.4) 是收敛的, 还可以给出近似解与准确解的误差估计.

定理 3.2 设迭代法 (3.4) 的迭代矩阵 $\boldsymbol{B}$ 满足 $\|\boldsymbol{B}\|=q<1$, 则迭代法是收敛的, 且有误差估计式

$$\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\| \leqslant \frac{q^k}{1-q}\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\|, \tag{3.8}$$

$$\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\| \leqslant \frac{q}{1-q}\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|. \tag{3.9}$$

证明 由式 (3.5), 有

$$\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\| = \|\boldsymbol{e}^{(k)}\| \leqslant \|\boldsymbol{B}^k\|\cdot\|\boldsymbol{e}^{(0)}\| \leqslant q^k\|\boldsymbol{e}^{(0)}\|.$$

注意到 $\boldsymbol{x}^*=(\boldsymbol{I}-\boldsymbol{B})^{-1}\boldsymbol{f}$, 于是

$$\begin{aligned}\|\boldsymbol{e}^{(0)}\| &= \|\boldsymbol{x}^{(0)}-\boldsymbol{x}^*\| = \|\boldsymbol{x}^{(0)}-(\boldsymbol{I}-\boldsymbol{B})^{-1}\boldsymbol{f}\| \\ &= \|(\boldsymbol{I}-\boldsymbol{B})^{-1}[(\boldsymbol{I}-\boldsymbol{B})\boldsymbol{x}^{(0)}-\boldsymbol{f}]\| \\ &= \|(\boldsymbol{I}-\boldsymbol{B})^{-1}(\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)})\| \\ &\leqslant \|(\boldsymbol{I}-\boldsymbol{B})^{-1}\|\cdot\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\|.\end{aligned}$$

因为 $\|\boldsymbol{B}\|<1$, 根据定理 1.19, 有

$$\|(\boldsymbol{I}-\boldsymbol{B})^{-1}\| \leqslant \frac{1}{1-\|\boldsymbol{B}\|} = \frac{1}{1-q},$$

于是

$$\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\| \leqslant \frac{q^k}{1-q}\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\|.$$

下证式 (3.9). 由于

$$\begin{aligned}\boldsymbol{e}^{(k)} &= \boldsymbol{x}^{(k)}-\boldsymbol{x}^* = (\boldsymbol{B}\boldsymbol{x}^{(k-1)}+\boldsymbol{f})-(\boldsymbol{B}\boldsymbol{x}^*+\boldsymbol{f}) \\ &= \boldsymbol{B}\boldsymbol{x}^{(k-1)}-\boldsymbol{B}\boldsymbol{x}^* = \boldsymbol{B}\boldsymbol{x}^{(k-1)}-\boldsymbol{B}(\boldsymbol{I}-\boldsymbol{B})^{-1}\boldsymbol{f} \\ &= \boldsymbol{B}(\boldsymbol{I}-\boldsymbol{B})^{-1}[(\boldsymbol{I}-\boldsymbol{B})\boldsymbol{x}^{(k-1)}-\boldsymbol{f}] \\ &= \boldsymbol{B}(\boldsymbol{I}-\boldsymbol{B})^{-1}(\boldsymbol{x}^{(k-1)}-\boldsymbol{x}^{(k)}),\end{aligned}$$

利用定理 1.19, 对上式两边取范数即得式 (3.9). 证毕. □

在理论上, 可用式 (3.8) 估计近似解达到某一精度所需要的迭代次数 (但由于 q 不易计算, 故计算实践中很少使用). 而式 (3.9) 则表明, 只要 $\|\boldsymbol{B}\|$ 不很接近于 1, 即可用 $\{\boldsymbol{x}^{(k)}\}$ 的相邻两项之差的范数 $\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|$ 来估计 $\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{*}\|$ 的大小, 这为用 $\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|$ 作为算法的终止准则值提供了理论上的依据.

下面简要介绍一下矩阵分裂迭代法的基本知识. 设矩阵 $\boldsymbol{A}$ 非奇异且具有分裂

$$\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}, \tag{3.10}$$

其中 $\boldsymbol{M}$ 是非奇异的, 则方程组 (3.1) 等价于

$$\boldsymbol{M}\boldsymbol{x}=\boldsymbol{N}\boldsymbol{x}+\boldsymbol{b}. \tag{3.11}$$

构造迭代法

$$\boldsymbol{M}\boldsymbol{x}^{(k+1)}=\boldsymbol{N}\boldsymbol{x}^{(k)}+\boldsymbol{b},\ k=0,1,\cdots \tag{3.12}$$

求解方程组 (3.1), 这种方法统称为*矩阵分裂迭代法*, 许多迭代法都可以写成这样的形式, 如后面要介绍的 Richardson 迭代法、Jacobi 迭代法、Gauss–Seidel 迭代法、SOR 迭代法等. 在迭代法 (3.12) 中, 由于每一次迭代需要解一个以 $\boldsymbol{M}$ 为系数矩阵的方程组, 故一般要求非奇异矩阵 $\boldsymbol{M}$ 的形式比较简单, 如对角矩阵、三对角矩阵、上 (下) 三角形矩阵等. 从后面的讨论知道, 当 $\boldsymbol{M}$ 是 $\boldsymbol{A}$ 的一个很好的近似时, 迭代解将很快收敛到方程组 (3.1) 的真解.

显然, 迭代法 (3.12) 的迭代矩阵为 $\boldsymbol{B}=\boldsymbol{M}^{-1}\boldsymbol{N}$, 并且它是线性定常迭代法. 根据定理 3.1 可知线性定常迭代法收敛的充要条件是

$$\rho(\boldsymbol{B})<1. \tag{3.13}$$

3.1.3 相容性和敏感性分析

假设由一个迭代法所得的迭代序列是收敛的, 问题是它是否收敛到待求解方程组的解. 对于线性定常迭代法, 只要适当构造迭代矩阵, 这不是个太大的问题. 但对非定常迭代法, 就必须回答这问题. 需要指出, 对于非定常迭代法, 每一步迭代矩阵的谱半径 $\rho(\boldsymbol{B}_k)<1$ 既不是收敛的必要条件, 也不是收敛的充分条件.

例 3.1 *考虑方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 其中*

$$\boldsymbol{A}=\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}0\\ 1\end{bmatrix},$$

其精确解为

$$\boldsymbol{x}^{*}=\begin{bmatrix}-1/3\\ 2/3\end{bmatrix}.$$

非定常迭代法

$$\begin{cases}\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_1\boldsymbol{x}^{(k)}+\boldsymbol{f}_1, & k\ \text{为偶数},\\ \boldsymbol{x}^{(k+1)}=\boldsymbol{B}_2\boldsymbol{x}^{(k)}+\boldsymbol{f}_2, & k\ \text{为奇数},\end{cases} \tag{3.14}$$

式中:

$$\boldsymbol{B}_1=\begin{bmatrix}0&0\\2&0\end{bmatrix},\quad \boldsymbol{f}_1=\begin{bmatrix}-1/3\\4/3\end{bmatrix},\quad \boldsymbol{B}_2=\begin{bmatrix}0&2\\0&0\end{bmatrix},\quad \boldsymbol{f}_2=\begin{bmatrix}-5/3\\2/3\end{bmatrix}.$$

显然, 每个迭代法均收敛, 因为 $\rho(\boldsymbol{B}_1)=\rho(\boldsymbol{B}_2)=0$. 然而, 有

$$\boldsymbol{B}_k=\begin{cases}\boldsymbol{B}_1(\boldsymbol{B}_2\boldsymbol{B}_1)^{k/2}=\begin{bmatrix}0&0\\2^{k+1}&0\end{bmatrix}, & k\text{ 为偶数},\\[2ex] (\boldsymbol{B}_2\boldsymbol{B}_1)^{(k+1)/2}=\begin{bmatrix}2^{k+1}&0\\0&0\end{bmatrix}, & k\text{ 为奇数},\end{cases}$$

从而式 (3.13) 不成立.

设

$$\boldsymbol{B}_1=\begin{bmatrix}2&0\\0&2\end{bmatrix},\quad \boldsymbol{f}_1=\begin{bmatrix}1/3\\-2/3\end{bmatrix},\quad \boldsymbol{B}_2=\begin{bmatrix}1/4&0\\0&1/4\end{bmatrix},\quad \boldsymbol{f}_2=\begin{bmatrix}-1/4\\1/2\end{bmatrix}.$$

显然前一方法不收敛. 然而, 有

$$\boldsymbol{B}_2\boldsymbol{B}_1=\begin{bmatrix}1/2&0\\0&1/2\end{bmatrix},$$

$$\boldsymbol{B}_k=\begin{cases}\boldsymbol{B}_1(\boldsymbol{B}_2\boldsymbol{B}_1)^{k/2}=\begin{bmatrix}(1/2)^{(k-2)/2}&0\\0&(1/2)^{(k-2)/2}\end{bmatrix}, & k\text{ 为偶数},\\[2ex] (\boldsymbol{B}_2\boldsymbol{B}_1)^{(k+1)/2}=\begin{bmatrix}(1/2)^{(k+1)/2}&0\\0&(1/2)^{(k+1)/2}\end{bmatrix}, & k\text{ 为奇数},\end{cases}$$

从而 $\lim\limits_{k\to\infty}\boldsymbol{B}_k=\boldsymbol{O}$, 且非定常迭代法 (3.14) 收敛.

下面考虑一种非定常迭代法, 其每一步使用公式

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_{k+1}\boldsymbol{x}^{(k)}+\boldsymbol{f}_{k+1},\quad k\geqslant 0. \tag{3.15}$$

显然, $\boldsymbol{x}^{(k)}$ 可表示为

$$\boldsymbol{x}^{(k)}=\widetilde{\boldsymbol{B}}_k\boldsymbol{x}^{(0)}+\widetilde{\boldsymbol{f}}_k,\quad k\geqslant 1. \tag{3.16}$$

其中

$$\widetilde{\boldsymbol{B}}_k=\boldsymbol{B}_k\boldsymbol{B}_{k-1}\cdots\boldsymbol{B}_1, \tag{3.17}$$

且

$$\widetilde{\boldsymbol{f}}_k=\boldsymbol{f}_k+\boldsymbol{B}_k\boldsymbol{f}_{k-1}+\boldsymbol{B}_k\boldsymbol{B}_{k-1}\boldsymbol{f}_{k-2}+\cdots+\boldsymbol{B}_k\boldsymbol{B}_{k-1}\cdots\boldsymbol{B}_2\boldsymbol{f}_1. \tag{3.18}$$

定义 3.5 (相容性) (1) 若 $\boldsymbol{x}^{(k)}=\boldsymbol{x}^*$, 且对所有的 $k'\geqslant k$ 有 $\boldsymbol{x}^{(k')}=\boldsymbol{x}^{(k)}=\boldsymbol{x}^*$, 则称非定常迭代法 (3.16) 与式 (3.1) 是相容的.

(2) 若由非定常迭代法 (3.16) 定义的序列 $\boldsymbol{x}^{(0)},\boldsymbol{x}^{(1)},\cdots$ 收敛, 且收敛到式 (3.1) 的解 $\boldsymbol{x}^*$, 则称非定常迭代法 (3.16) 反相容于式 (3.1).

(3) 若非定常迭代法 (3.16) 与式 (3.1) 既是相容的又是反相容的, 则称其与式 (3.1) 是完全相容的.

定理 3.3 假设非定常迭代法 (3.16) 由式 (3.15) 而得, 若对每个 $s=1,2,\cdots$, 线性定常迭代

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_s\boldsymbol{x}^{(k)}+\boldsymbol{f}_s,\ k\geqslant 0 \tag{3.19}$$

是相容的, 则非定常迭代法 (3.16) 是相容的. 反之, 若非定常迭代法 (3.16) 是相容的, 则对每个 s, 线性定常迭代法 (3.19) 是相容的.

定理 3.4 若 $\boldsymbol{A}$ 是非奇异的且非定常迭代法 (3.16) 是相容的, 则对每个 k, 有

$$\widetilde{\boldsymbol{f}}_k=(\boldsymbol{I}-\widetilde{\boldsymbol{B}}_k)\boldsymbol{A}^{-1}\boldsymbol{b}.$$

定理 3.5 非定常迭代法 (3.16) 收敛的充要条件是: $\{\widetilde{\boldsymbol{f}}_k\}$ 收敛, 且 $\lim\limits_{k\to\infty}\widetilde{\boldsymbol{B}}_k=\boldsymbol{O}$.

3.1.4 几种常见的矩阵分裂

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}(\mathbb{C}^{n\times n})$ 非奇异, 若存在矩阵 $\boldsymbol{M}$ 和 $\boldsymbol{N}$ 满足 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$, 且 $\boldsymbol{M}$ 是可逆的, 则称 $\boldsymbol{M}$ 和 $\boldsymbol{N}$ 为 $\boldsymbol{A}$ 的一个分裂.

定理 3.6 设 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$ 是非奇异矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的一个分裂, 则

$$\boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{A}^{-1}=\boldsymbol{A}^{-1}\boldsymbol{N}\boldsymbol{M}^{-1}, \tag{3.20}$$

矩阵 $\boldsymbol{M}^{-1}\boldsymbol{N}$ 与 $\boldsymbol{A}^{-1}\boldsymbol{N}$ 可交换, 且 $\boldsymbol{N}\boldsymbol{M}^{-1}$ 与 $\boldsymbol{N}\boldsymbol{A}^{-1}$ 也可交换. 从而, 矩阵 $\boldsymbol{M}^{-1}\boldsymbol{N}$ 与 $\boldsymbol{A}^{-1}\boldsymbol{N}$ (或 $\boldsymbol{N}\boldsymbol{M}^{-1}$ 与 $\boldsymbol{N}\boldsymbol{A}^{-1}$) 具有相同的特征向量.

证明 直接验证式 (3.20), 得

$$\begin{aligned}\boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{A}^{-1}&=\boldsymbol{M}^{-1}(\boldsymbol{M}\boldsymbol{A}^{-1}-\boldsymbol{I})=\boldsymbol{A}^{-1}-\boldsymbol{M}^{-1}\\&=\boldsymbol{A}^{-1}(\boldsymbol{M}-\boldsymbol{A})\boldsymbol{M}^{-1}=\boldsymbol{A}^{-1}\boldsymbol{N}\boldsymbol{M}^{-1}.\end{aligned}$$

由式 (3.20) 立即可得矩阵 $\boldsymbol{M}^{-1}\boldsymbol{N}$ 与 $\boldsymbol{A}^{-1}\boldsymbol{N}$ 及 $\boldsymbol{N}\boldsymbol{M}^{-1}$ 与 $\boldsymbol{N}\boldsymbol{A}^{-1}$ 可交换.

设 $\boldsymbol{x}$ 为 $\boldsymbol{A}^{-1}\boldsymbol{N}$ 对应于特征值 λ 的特征向量, 则有

$$\boldsymbol{A}^{-1}\boldsymbol{N}\boldsymbol{x}=\lambda\boldsymbol{x}\Longrightarrow\boldsymbol{N}\boldsymbol{x}=\lambda\boldsymbol{A}\boldsymbol{x}=\lambda(\boldsymbol{M}-\boldsymbol{N})\boldsymbol{x}.$$

由此可得

$$\boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{x}=\frac{\lambda}{1+\lambda}\boldsymbol{x},\ \lambda\neq-1.$$

上式表明 $\boldsymbol{x}$ 也是矩阵 $\boldsymbol{M}^{-1}\boldsymbol{N}$ 对应于特征值 $\dfrac{\lambda}{1+\lambda}$ 的特征向量. 证毕. □

定理 3.7 设 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$ 是非奇异矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的一个分裂, 则有

$$\lambda(\boldsymbol{M}^{-1}\boldsymbol{N})=\frac{\lambda(\boldsymbol{A}^{-1}\boldsymbol{N})}{1+\lambda(\boldsymbol{A}^{-1}\boldsymbol{N})}. \tag{3.21}$$

因此, 若 $\lambda(\boldsymbol{A}^{-1}\boldsymbol{N})\in\mathbb{R}$, 则相应的特征值 $\lambda(\boldsymbol{M}^{-1}\boldsymbol{N})\in\mathbb{R}$, 反之亦然. 若 $\lambda(\boldsymbol{A}^{-1}\boldsymbol{N})\in\mathbb{C}$, 则相应的特征值 $\lambda(\boldsymbol{M}^{-1}\boldsymbol{N})\in\mathbb{C}$, 反之亦然.

下面给出文献中常见的一些矩阵分裂.

定义 3.6 设 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$ 是非奇异矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的一个分裂.

(1) 若 $\boldsymbol{M}^{-1}\geqslant\boldsymbol{O}$ 且 $\boldsymbol{N}\geqslant\boldsymbol{O}$, 则称为正规分裂.

(2) 若 $\boldsymbol{M}^{-1}\geqslant\boldsymbol{O}$, $\boldsymbol{M}^{-1}\boldsymbol{N}\geqslant\boldsymbol{O}$ 且 $\boldsymbol{N}\boldsymbol{M}^{-1}\geqslant\boldsymbol{O}$, 则称为非负分裂.

(3) 若 $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})=\rho(\boldsymbol{N}\boldsymbol{M}^{-1})<1$, 则称为收敛的分裂.

定理 3.8 设 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$ 是非奇异矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的一个分裂, 如果 $\boldsymbol{M}^{-1}\boldsymbol{N}$ 和 $\boldsymbol{A}^{-1}\boldsymbol{N}$ 均非负, 则分裂是收敛的且有

$$\rho(\boldsymbol{M}^{-1}\boldsymbol{N})=\frac{\rho(\boldsymbol{A}^{-1}\boldsymbol{N})}{1+\rho(\boldsymbol{A}^{-1}\boldsymbol{N})}. \tag{3.22}$$

定义 3.7 设 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$ 是非奇异矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的一个分裂.

(1) 若 $\boldsymbol{M}^{\mathrm{H}}+\boldsymbol{N}$ 是 Hermite 正定的, 且 $\boldsymbol{N}\geqslant\boldsymbol{O}$, 则称为 P 正规分裂.

(2) 若 $\langle\boldsymbol{M}\rangle-|\boldsymbol{N}|$ 是单调的 $\big($即 $(\langle\boldsymbol{M}\rangle-|\boldsymbol{N}|)^{-1}\geqslant\boldsymbol{O}\big)$, 则称为 H 分裂.

(3) 若 $\langle\boldsymbol{A}\rangle=\langle\boldsymbol{M}\rangle-|\boldsymbol{N}|$, 则称为 H 相容分裂.

(4) 若 $\boldsymbol{M}$ 是一个 M 矩阵且 $\boldsymbol{N}\geqslant\boldsymbol{O}$, 则称为 M 分裂.

下面给出使几种常见分裂收敛的一些条件及一些比较定理, 这些知识在证明相应的分裂迭代法的收敛性方面是很有用的.

定理 3.9 (1) 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 且 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$ 为正规分裂, 如果 $\boldsymbol{A}^{-1}\geqslant\boldsymbol{O}$, 则

$$\rho(\boldsymbol{M}^{-1}\boldsymbol{N})=\frac{\rho(\boldsymbol{A}^{-1}\boldsymbol{N})}{1+\rho(\boldsymbol{A}^{-1}\boldsymbol{N})}<1. \tag{3.23}$$

反之, 若 $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})<1$, 则 $\boldsymbol{A}^{-1}\geqslant\boldsymbol{O}$.

(2) 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 是 Hermite 的且 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$ 为 P 正规分裂, 则 $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})<1$ 当且仅当 $\boldsymbol{A}$ 是 Hermite 正定的.

定理 3.9 说明, 在满足定理的条件下, 以 $\boldsymbol{M}^{-1}\boldsymbol{N}$ 为迭代矩阵的分裂迭代法是收敛的.

定理 3.10 设 $\boldsymbol{A}^{-1}\geqslant\boldsymbol{O}$ 且 $\boldsymbol{A}=\boldsymbol{M}_1-\boldsymbol{N}_1=\boldsymbol{M}_2-\boldsymbol{N}_2$ 为正规分裂, 如果下面条件之一成立:

(1) $\boldsymbol{N}_1\leqslant\boldsymbol{N}_2$.

(2) $\boldsymbol{M}_1^{-1} \geqslant \boldsymbol{M}_2^{-1}$.

(3) $\boldsymbol{M}_2^{-1}\boldsymbol{N}_2 \geqslant \boldsymbol{N}_1\boldsymbol{M}_1^{-1} \geqslant \boldsymbol{O}$.

(4) $\boldsymbol{A}^{-1}\boldsymbol{N}_2 \geqslant \boldsymbol{N}_1\boldsymbol{A}^{-1} \geqslant \boldsymbol{O}$.

则

$$\rho(\boldsymbol{M}_1^{-1}\boldsymbol{N}_1) \leqslant \rho(\boldsymbol{M}_2^{-1}\boldsymbol{N}_2). \tag{3.24}$$

特别地, 若 $\boldsymbol{A}^{-1} \geqslant \boldsymbol{O}$ 且 $\boldsymbol{N}_2 \geqslant \boldsymbol{N}_1$ 而 $\boldsymbol{N}_2 \neq \boldsymbol{N}_1$ (或 $\boldsymbol{M}_1^{-1} > \boldsymbol{M}_2^1$), 则

$$\rho(\boldsymbol{M}_1^{-1}\boldsymbol{N}_1) \leqslant \rho(\boldsymbol{M}_2^{-1}\boldsymbol{N}_2). \tag{3.25}$$

定理 3.10 说明, 在满足定理的条件下, 以 $\boldsymbol{A} = \boldsymbol{M}_1 - \boldsymbol{N}_1$ 构造的分裂迭代法比以 $\boldsymbol{A} = \boldsymbol{M}_2 - \boldsymbol{N}_2$ 构造的分裂迭代法收敛得更快.

定理 3.11 若 $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ 为 $\boldsymbol{A}$ 的一个 H 分裂, 则 $\boldsymbol{A}$ 和 $\boldsymbol{M}$ 都是 H 矩阵, 且 $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) \leqslant \rho(\langle\boldsymbol{M}\rangle^{-1}|\boldsymbol{N}|) < 1$. 若分裂是一个 H 相容分裂, 则它一定是 H 分裂, 从而是收敛的分裂.

定理 3.12 设 $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ 为 $\boldsymbol{A}$ 的一个 M 分裂, 则

(1) 若 $\boldsymbol{A}$ 是不可约的, 则存在一个正向量 $\boldsymbol{x} > \boldsymbol{0}$ 使得 $(\boldsymbol{M}^{-1}\boldsymbol{N})\boldsymbol{x} = \rho(\boldsymbol{M}^{-1}\boldsymbol{N})\boldsymbol{x}$.

(2) $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$ 当且仅当 $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ 是一个非奇异 M 分裂.

3.2 几种经典迭代法

本节给出求解 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 的几个经典迭代法及相应的收敛性定理.

3.2.1 Richardson 迭代法

在式 (3.10) 中取 $\boldsymbol{M} = \boldsymbol{I}$, $\boldsymbol{N} = \boldsymbol{I} - \boldsymbol{A}$, 则相应于式 (3.12) 的迭代法为

$$\boldsymbol{x}^{(k)} = (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x}^{(k-1)} + \boldsymbol{b} = \boldsymbol{x}^{(k-1)} + \boldsymbol{r}^{(k-1)}, \tag{3.26}$$

式中: $\boldsymbol{r}^{(k-1)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k-1)}$ 为前一步的残差.

这就是著名的 Richardson 迭代法. 式 (3.26) 基本上不具有实用性, 因为它要求迭代矩阵 $\boldsymbol{I} - \boldsymbol{A}$ 的谱半径 $\rho(\boldsymbol{I} - \boldsymbol{A}) < 1$, 这等价于 $\rho(\boldsymbol{A}) < 2$. 但它对于加速技术、Krylov 子空间方法及多重网格法都具有理论意义.

用 $-\boldsymbol{A}$ 乘以式 (3.26) 两边并加上 $\boldsymbol{b}$, 有

$$\boldsymbol{r}^{(k)} = (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{r}^{(k-1)} = \cdots = (\boldsymbol{I} - \boldsymbol{A})^k\boldsymbol{r}^{(0)} = p_k(\boldsymbol{A})\boldsymbol{r}^{(0)}, \tag{3.27}$$

式中: $p_k(\boldsymbol{A})$ 为 $\boldsymbol{A}$ 的 k 次多项式.

式 (3.27) 表明, 如果 $\|\boldsymbol{I} - \boldsymbol{A}\|$ 比 1 小得多, 则可期望其具有快速的收敛性.

注意到式 (3.27) 将残差表示成 $\boldsymbol{A}$ 的一个多项式与初始残差的乘积, 对于标准的 Richardson 迭代法, 此多项式非常简单, 即 $p_k(\boldsymbol{A}) = (\boldsymbol{I} - \boldsymbol{A})^k$. 以后将会看到, 许多流行的迭代法都具有类似的性质: 第 k 步的残差 $\boldsymbol{r}^{(k)}$ 可以表示为 $\boldsymbol{A}$ 的某个 k 次多项式

$p_k(\boldsymbol{A})$ 与初始残差 $\boldsymbol{r}^{(0)}$ 的乘积, 并称此多项式为*残差多项式*. 这种性质是得到迭代法收敛界的一个有力工具.

还可以引进参数化的 Richardson 迭代法. 考虑 $\alpha > 0$ 并将 $\boldsymbol{A}$ 分裂为

$$\boldsymbol{A} = \frac{1}{\alpha}\boldsymbol{I} - \Big(\frac{1}{\alpha}\boldsymbol{I} - \boldsymbol{A}\Big),$$

则导致一种定常的参数化 Richardson 迭代:

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + \alpha \boldsymbol{r}^{(k-1)}. \tag{3.28}$$

若在每一步迭代取不同的参数 α_k, 则得到非定常的参数化 Richardson 迭代:

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + \alpha_k \boldsymbol{r}^{(k-1)}. \tag{3.29}$$

上式中的 α_k 可有多种选取方式, 如通过极小化 $\|\boldsymbol{r}^{(k)}\|$ 来选取. 注意到

$$\boldsymbol{r}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k-1)} - \alpha_k \boldsymbol{A}\boldsymbol{r}^{(k-1)} = (\boldsymbol{I} - \alpha_k \boldsymbol{A})\boldsymbol{r}^{(k-1)},$$

容易推得

$$\boldsymbol{r}^{(k)} = p_k(\boldsymbol{A})\boldsymbol{r}^{(0)}, \quad \text{其中} \quad p_k(\boldsymbol{A}) = \prod_{i=1}^{k}(\boldsymbol{I} - \alpha_i \boldsymbol{A}).$$

注 3.1 从式 (3.26)、式 (3.28) 和式 (3.29) 可知, Richardson 迭代法新的近似解 $\boldsymbol{x}^{(k)}$ 是前一步近似解 $\boldsymbol{x}^{(k-1)}$ 的一个校正, 校正量或者是前一步的残差 $\boldsymbol{r}^{(k-1)}$, 或者是沿残差方向前进某个步长 α_k (即 $\alpha_k \boldsymbol{r}^{(k-1)}$). 这一思想在现代变分迭代法中得到了很好的继承和发展. 最具代表性的是共轭梯度 (CG) 法和多重网格 (MG) 法.

由定理 3.1 不难得到定常的参数化 Richardson 迭代法 (3.28) 的收敛性定理.

定理 3.13 (1) 设 $\boldsymbol{A}$ 为对称正定矩阵且其特征值 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n > 0$, 则定常的参数化 Richardson 迭代法 (3.28) 收敛的充要条件是 $\alpha < \dfrac{2}{\lambda_1}$.

(2) 对于定常的参数化 Richardson 迭代法 (3.28), 参数 α 的最优值为 $\alpha_{\text{opt}} = \dfrac{2}{\lambda_1 + \lambda_n}$, 此时,

$$\rho(\boldsymbol{I} - \alpha_{\text{opt}}\boldsymbol{A}) = \frac{\kappa(\boldsymbol{A}) - 1}{\kappa(\boldsymbol{A}) + 1},$$

式中: $\kappa(\boldsymbol{A}) = \dfrac{\lambda_1}{\lambda_n}$.

3.2.2 Jacobi 迭代法

将系数矩阵 $\boldsymbol{A} = (a_{ij})$ 分裂为

$$\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{L} - \boldsymbol{U}, \tag{3.30}$$

式中: $\boldsymbol{D}=\mathrm{diag}(a_{11},a_{22},\cdots,a_{nn})$,

$$\boldsymbol{L}=-\begin{bmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & 0 \end{bmatrix},\quad \boldsymbol{U}=-\begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ & 0 & a_{23} & \cdots & a_{2n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & a_{n-1,n} \\ & & & & 0 \end{bmatrix}.$$

在式 (3.10) 中, 取

$$\boldsymbol{M}=\boldsymbol{D},\quad \boldsymbol{N}=\boldsymbol{L}+\boldsymbol{U}, \tag{3.31}$$

则迭代法 (3.4) 中的迭代矩阵和右端项分别为

$$\boldsymbol{B}_J=\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U}),\quad \boldsymbol{f}_J=\boldsymbol{D}^{-1}\boldsymbol{b}.$$

相应的迭代法为

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_J\boldsymbol{x}^{(k)}+\boldsymbol{f}_J,\quad k=0,1,2,\cdots, \tag{3.32}$$

式 (3.32) 称为 Jacobi 迭代法. 其分量形式为

$$x_i^{(k+1)}=\frac{1}{a_{ii}}\Big(b_i-\sum_{j\neq i}a_{ij}x_j^{(k)}\Big),\quad i=1,2,\cdots,n;\ k=0,1,\cdots. \tag{3.33}$$

该迭代法具有与 $\boldsymbol{x}^{(k+1)}$ 中分量的计算次序无关, 因此容易实现并行计算等优点.

为了便于编程, 下面给出 Jacobi 迭代法的具体算法步骤如下:

算法 3.1 (Jacobi 迭代法)

步 1, 取初始点 $\boldsymbol{x}^{(0)}$, 精度要求 ε, 最大迭代次数 N, 置 $k:=0$.

步 2, 由式 (3.32) 或式 (3.33) 计算 $\boldsymbol{x}^{(k+1)}$.

步 3, 若 $\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k+1)}\|/\|\boldsymbol{b}\|\leqslant\varepsilon$, 则停算, 输出 $\boldsymbol{x}^{(k+1)}$ 作为方程组的近似解.

步 4, 置 $\boldsymbol{x}^{(k)}:=\boldsymbol{x}^{(k+1)}$, $k:=k+1$, 转步 2.

根据算法 3.1, 可编制 MATLAB 程序如下:

```
%Jacobi迭代法程序-mjacobi.m
function [x,k,err,time]=mjacobi(A,b,x,tol,max_it)
if nargin<5, max_it=1000; end
if nargin<4, tol=1.e-5; end
if nargin<3, x=zeros(size(b)); end
tic;  bnrm2 = norm(b);
r=b-A*x;  %计算初始残差r0=b-Ax
err=norm(r)/bnrm2;
if (err<tol), return; end
D=diag(diag(A));
```

```
for k=1:max_it,   % 迭代开始
    x=D\((D-A)*x+b);
    r=b-A*x;   %计算残差r=b-Ax
    err=norm(r)/bnrm2;
    if(err<=tol), break; end
end
time=toc;
```

下面的定理给出了 Jacobi 迭代法的一个充分必要条件.

定理 3.14 设 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$ 是对称矩阵, 且 $a_{ii}>0$ $(i=1,2,\cdots,n)$, 则 Jacobi 迭代法收敛的充要条件是 $\boldsymbol{A}$ 和 $2\boldsymbol{D}-\boldsymbol{A}$ 都是正定矩阵.

证明 记 $\boldsymbol{D}^{\frac{1}{2}}=\mathrm{diag}(\sqrt{a_{11}},\sqrt{a_{22}},\cdots,\sqrt{a_{nn}})$, $\boldsymbol{D}^{-\frac{1}{2}}=(\boldsymbol{D}^{\frac{1}{2}})^{-1}$, 则

$$\boldsymbol{B}_J=\boldsymbol{D}^{-1}(\boldsymbol{D}-\boldsymbol{A})=\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{D}^{\frac{1}{2}},$$

即 $\boldsymbol{B}_J$ 与 $\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 相似, 从而它们有相同的特征值. 再由 $\boldsymbol{A}$ 的对称性可得

$$(\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})^{\mathrm{T}}=\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$$

即 $\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 的特征值都是实数, 从而 $\boldsymbol{B}_J$ 的特征值都是实数.

必要性. 设 Jacobi 迭代法收敛, 即 $\rho(\boldsymbol{B}_J)<1$, 则 $\rho(\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})<1$. 设实数 λ 是实对称矩阵 $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 的任一特征值, 则有

$$0=\det(\lambda\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})=(-1)^n\det[(1-\lambda)\boldsymbol{I}-(\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})],$$

即实数 $1-\lambda$ 是实对称矩阵 $\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 的特征值, 从而有

$$|1-\lambda|<1 \quad \text{或者} \quad 0<\lambda<2.$$

因此 $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 是正定矩阵. 由于 $\boldsymbol{A}$ 与 $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 合同, 所以 $\boldsymbol{A}$ 也是正定矩阵.

再设 $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 的全体特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$, 则存在正交矩阵 $\boldsymbol{P}$, 使得

$$\boldsymbol{P}^{\mathrm{T}}(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{P}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n).$$

于是有

$$\boldsymbol{P}^{\mathrm{T}}(2\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{P}=\mathrm{diag}(2-\lambda_1,2-\lambda_2,\cdots,2-\lambda_n).$$

由于 $2-\lambda_i>0$ $(i=1,2,\cdots,n)$, 故 $2\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 正定. 注意到 $2\boldsymbol{D}-\boldsymbol{A}=\boldsymbol{D}^{\frac{1}{2}}(2\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})\boldsymbol{D}^{\frac{1}{2}}$ 合同于 $2\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$, 可得 $2\boldsymbol{D}-\boldsymbol{A}$ 正定.

充分性. 设 $\boldsymbol{A}$ 和 $2\boldsymbol{D}-\boldsymbol{A}$ 都是正定矩阵, 则有

$$\boldsymbol{A}\text{ 正定}\Longrightarrow\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\text{ 正定}\Longrightarrow\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\text{ 的特征值大于零},$$

$$2\boldsymbol{D}-\boldsymbol{A}\text{ 正定}\Longrightarrow2\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}=\boldsymbol{D}^{-\frac{1}{2}}(2\boldsymbol{D}-\boldsymbol{A})\boldsymbol{D}^{-\frac{1}{2}}\text{ 正定}$$

$$\Longrightarrow2\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\text{ 的特征值大于零}.$$

设 μ 是 $\boldsymbol{B}_J$ 的任一特征值, 从而也是 $\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 的特征值, 且为实数, 则有

$$0=\det(\mu\boldsymbol{I}-\boldsymbol{B}_J)=\det[\mu\boldsymbol{I}-(\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})]$$

$$=\begin{cases}(-1)^n\det[(1-\mu)\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}]\\ \det[(\mu+1)\boldsymbol{I}-(2\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}})]\end{cases},$$

即 $1-\mu$ 是 $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 的特征值, $\mu+1$ 是 $2\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 的特征值, 所以

$$1-\mu>0,\mu+1>0\Longrightarrow|\mu|<1\Longrightarrow\rho(\boldsymbol{B}_J)<1.$$

由定理 3.1 知 Jacobi 迭代法收敛. 证毕. □

下面的定理是 Jacobi 迭代法的另一个收敛性定理.

定理 3.15 *若 $\boldsymbol{A}$ 为严格对角占优, 或不可约对角占优, 或 $\boldsymbol{A}$ 为 H 矩阵, 则 Jacobi 迭代法收敛.*

证明 只证明 $\boldsymbol{A}$ 是严格对角占优或不可约对角占优的情形. 注意到 Jacobi 迭代法的迭代矩阵为

$$\boldsymbol{B}_J=\boldsymbol{D}^{-1}(\boldsymbol{D}-\boldsymbol{A})=\boldsymbol{I}-\boldsymbol{D}^{-1}\boldsymbol{A}.$$

只需证明 $\rho(\boldsymbol{B}_J)<1$. 事实上, 设 λ 为 $\boldsymbol{B}_J$ 的特征值, 对于行的情形, 当 $\boldsymbol{A}$ 为严格对角占优或不可约对角占优时, 矩阵 $\boldsymbol{I}-\boldsymbol{B}_J=\boldsymbol{D}^{-1}\boldsymbol{A}$ 也为严格对角占优或不可约对角占优. 因此, 当 $|\lambda|\geqslant1$ 时, $\boldsymbol{I}-\dfrac{1}{\lambda}\boldsymbol{B}_J$ 也是严格对角占优或不可约对角占优的 (相当于 $\boldsymbol{I}-\boldsymbol{B}_J$ 的对角元不变, 非对角元模变小). 根据引理 1.1, 当 $|\lambda|\geqslant1$ 时, 有

$$\det\left(\boldsymbol{I}-\frac{1}{\lambda}\boldsymbol{B}_J\right)\neq0\Longrightarrow\det(\lambda\boldsymbol{I}-\boldsymbol{B}_J)\neq0,$$

故 $\boldsymbol{B}_J$ 的特征值 λ 不满足 $|\lambda|\geqslant1$, 即 $\rho(\boldsymbol{B}_J)<1$, 从而 Jacobi 迭代法收敛.

对于列的情形, 当 $\boldsymbol{A}$ 为严格对角占优或不可约对角占优时, 矩阵

$$\boldsymbol{I}-\boldsymbol{D}\boldsymbol{B}_J\boldsymbol{D}^{-1}=\boldsymbol{A}\boldsymbol{D}^{-1}$$

也为严格对角占优或不可约对角占优. 因此, 当 $|\lambda|\geqslant1$ 时,

$$\boldsymbol{I}-\frac{1}{\lambda}\boldsymbol{D}\boldsymbol{B}_J\boldsymbol{D}^{-1}$$

也是严格对角占优或不可约对角占优的. 根据引理 1.1, 当 $|\lambda|\geqslant1$ 时, 有

$$\det\left(\boldsymbol{I}-\frac{1}{\lambda}\boldsymbol{D}\boldsymbol{B}_J\boldsymbol{D}^{-1}\right)\neq0,$$

从而

$$\det\left(\boldsymbol{I}-\frac{1}{\lambda}\boldsymbol{B}_J\right)\neq0\Rightarrow\det(\lambda\boldsymbol{I}-\boldsymbol{B}_J)\neq0,$$

故 $\boldsymbol{B}_J$ 的特征值 λ 不满足 $|\lambda| \geqslant 1$, 即 $\rho(\boldsymbol{B}_J) < 1$, 从而 Jacobi 迭代法收敛. 证毕. □

Jacobi 迭代法的一种推广是将其用于分块矩阵

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \cdots & \boldsymbol{A}_{1m} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \cdots & \boldsymbol{A}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{m1} & \boldsymbol{A}_{m2} & \cdots & \boldsymbol{A}_{mm} \end{bmatrix}, \quad \begin{array}{l} \boldsymbol{A}_{ir} \in \mathbb{R}^{n_i \times n_r},\ 1 \leqslant i, r \leqslant m, \\ n_1 + n_2 + \cdots + n_m = n, \end{array} \tag{3.34}$$

式中: 每个主对角块 $\boldsymbol{A}_{ii} (i = 1, 2, \cdots, m)$ 为非奇异的方阵. 将 $\boldsymbol{x}^{(k+1)}$, $\boldsymbol{x}^{(k)}$ 和 $\boldsymbol{b}$ 作相应的分块, 有

$$\boldsymbol{A}_{ii}\boldsymbol{x}_i^{(k+1)} = \boldsymbol{b}_i - \sum_{r=1}^{i-1} \boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k)} - \sum_{r=i+1}^{m} \boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k)}, \quad i = 1, 2, \cdots, m, \ k = 0, 1, \cdots. \tag{3.35}$$

称上述迭代法为块 Jacobi (BJ) 迭代法, 其中用 $\boldsymbol{x}_i$ 表示第 i 个块分量以示区别. 它也有上面类似的收敛性定理.

另一种推广是在 Jacobi 迭代法中引入外推法技术, 即进行如下迭代

$$\begin{cases} \boldsymbol{D}\widetilde{\boldsymbol{x}}^{(k+1)} = (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x}^{(k)} + \boldsymbol{b}, \\ \boldsymbol{x}^{(k+1)} = \gamma\widetilde{\boldsymbol{x}}^{(k+1)} + (1 - \gamma)\boldsymbol{x}^{(k)}, \end{cases} \quad k = 0, 1, \cdots, \tag{3.36}$$

式中: γ 为外推因子. 或等价地, 有

$$\frac{1}{\gamma}\boldsymbol{D}\boldsymbol{x}^{(k+1)} = \frac{1 - \gamma}{\gamma}\boldsymbol{D}\boldsymbol{x}^{(k)} + (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x}^{(k)} + \boldsymbol{b}, \tag{3.37}$$

该方法称为外推 Jacobi 方法, 其中, 若 $\gamma > 1$ 也称为 JOR 方法. 它相应于将 $\boldsymbol{A}$ 分裂为

$$\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N} = \frac{1}{\gamma}\boldsymbol{D} - \Big[\frac{1 - \gamma}{\gamma}\boldsymbol{D} + (\boldsymbol{L} + \boldsymbol{U})\Big].$$

关于外推 Jacobi 迭代法, 有下面的收敛性定理.

定理 3.16 (1) 设 $\boldsymbol{A}$ 为对称正定的, 若 $\dfrac{2}{\gamma}\boldsymbol{D} - \boldsymbol{A}$ 是对称正定的, 则迭代法 (3.37) 对任意 $\boldsymbol{x}^{(0)}$ 都收敛.

(2) 设 $\boldsymbol{A}$ 对称且 $\boldsymbol{D}$ 为正定矩阵, 则 $\dfrac{2}{\gamma}\boldsymbol{D} - \boldsymbol{A}$ 是对称正定的充要条件是

$$0 < \gamma < \frac{2}{1 - \lambda_{\min}\big(\boldsymbol{D}^{-1}(\boldsymbol{D} - \boldsymbol{A})\big)}.$$

3.2.3 Gauss–Seidel (GS) 迭代法

从前面的讨论可以看到, Jacobi 迭代法的主要优点是方法简单, 在一定条件下具有实用性, 比如它可用作建立其他一些迭代过程的辅助方法. 然而, Jacobi 迭代法并不总是收敛的, 收敛时通常速度也很缓慢. 下面以 Jacobi 迭代法为基础, 考虑关于方程组

(3.1) 的其他迭代法. 仍假设 $\boldsymbol{A}$ 的对角元素不为零. 从串行计算的角度来看, Jacobi 迭代法很自然地先计算分量 x_1 的新迭代值

$$x_1^{(k+1)} = \frac{1}{a_{11}}\left(b_1 - \sum_{r=2}^{n} a_{1r}x_r^{(k)}\right). \tag{3.38}$$

得到 $x_1^{(k+1)}$ 后, 在后面 $x_i^{(k+1)}$ 的计算中就有理由用 $x_1^{(k+1)}$ 取代 $x_1^{(k)}$. 依次对 $x_2^{(k+1)}$, $x_3^{(k+1)}$ 等也用得到的新值立即取代旧值. Gauss–Seidel 原理就是“一旦获得新信息便立即利用”. 按此原理, 有

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{r=1}^{i-1} a_{ir}x_r^{(k+1)} - \sum_{r=i+1}^{n} a_{ir}x_r^{(k)}\right), \quad i = 1, 2, \cdots, n;\ k = 0, 1, \cdots. \tag{3.39}$$

此方法称为 Gauss–Seidel 迭代法 (GS 迭代法). 沿用前面的记号, 取

$$\boldsymbol{M} = \boldsymbol{D} - \boldsymbol{L}, \quad \boldsymbol{N} = \boldsymbol{U}, \tag{3.40}$$

则式 (3.39) 可以写成

$$\boldsymbol{x}^{(k+1)} = (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)} + (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{b}, \quad k = 0, 1, 2, \cdots. \tag{3.41}$$

因此, Gauss–Seidel 迭代法的迭代矩阵 $\boldsymbol{B}_S = (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{U}$, 常向量 $\boldsymbol{f}_S = (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{b}$. $\boldsymbol{A}$ 的对角元素非零的假设确保了 $(\boldsymbol{D} - \boldsymbol{L})^{-1}$ 的存在性.

与 Jacobi 迭代法相比, Gauss–Seidel 迭代法使用了最新已经计算的分量. 因此, 一般情况下 Gauss–Seidel 迭代法要比 Jacobi 迭代法有效, 且在编程时 Gauss–Seidel 迭代法只要一个数组就够了, 将新计算出来的分量及时覆盖旧的分量.

为了便于计算机编程实现, 给出 Gauss-Seidel 迭代法的具体算法步骤如下.

算法 3.2 (GS 迭代法)

步 1, 输入矩阵 $\boldsymbol{A}$, 右端向量 $\boldsymbol{b}$, 初始点 $\boldsymbol{x}^{(0)}$, 精度要求 ε, 最大迭代次数 N, 置 $k := 0$.

步 2, 由式 (3.41) 或式 (3.39) 计算 $\boldsymbol{x}^{(k+1)}$.

步 3, 若 $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k+1)}\|/\|\boldsymbol{b}\| \leqslant \varepsilon$, 则停算, 输出 $\boldsymbol{x}^{(k+1)}$ 作为方程组的近似解.

步 4, 置 $\boldsymbol{x}^{(k)} := \boldsymbol{x}^{(k+1)}$, $k := k + 1$, 转步 2.

根据算法 3.2, 编制 MATLAB 程序如下:

```
%GS迭代法程序-mseidel.m
function [x,k,err,time]=mseidel(A,b,x,tol,max_it)
if nargin<5, max_it=1000; end
if nargin<4, tol=1.e-5; end
if nargin<3, x=zeros(size(b)); end
tic;  bnrm2=norm(b);
r=b-A*x;   %计算初始残差r0=b-Ax
```

```
err=norm(r)/bnrm2;
if (err<tol), return; end
DL=tril(A); U=-triu(A,1);
for k=1:max_it,    % 迭代开始
    x=DL\(U*x+b);
    r=b-A*x;   %计算残差r=b-Ax
    err=norm(r)/bnrm2;
    if (err<=tol), break; end
end
time=toc;
```

下面考虑 Gauss–Seidel 迭代法的收敛性.

定理 3.17 设 $\boldsymbol{A}$ 为严格对角占优, 或不可约对角占优, 或 $\boldsymbol{A}$ 是 H 矩阵, 则 Gauss-Seidel 迭代法收敛.

证明 只证明 $\boldsymbol{A}$ 为严格对角占优, 或不可约对角占优时的结论. 只需证 Gauss–Seidel 迭代法的迭代矩阵 $\boldsymbol{B}_S=(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}$ 满足 $\rho(\boldsymbol{B}_S)<1$ 即可. 事实上, 设 μ 为 $\boldsymbol{B}_S$ 的特征值, 对于行的情形, 注意到 $(\boldsymbol{D}-\boldsymbol{L})-\boldsymbol{U}=\boldsymbol{A}$, 若 $\boldsymbol{A}$ 为严格对角占优或不可约对角占优, 当 $|\mu|\geqslant 1$ 时,

$$(\boldsymbol{D}-\boldsymbol{L})-\frac{1}{\mu}\boldsymbol{U}$$

也是严格对角占优或不可约对角占优. 根据引理 1.1, 当 $|\mu|\geqslant 1$ 时, 有

$$\det\left[(\boldsymbol{D}-\boldsymbol{L})-\frac{1}{\mu}\boldsymbol{U}\right]\neq 0\Longrightarrow \det\left[\mu\boldsymbol{I}-(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}\right]\neq 0,$$

故 $\boldsymbol{B}_S$ 特征值 μ 不满足 $|\mu|\geqslant 1$, 即 $\rho(\boldsymbol{B}_S)<1$, 从而 Gauss–Seidel 迭代法收敛.

对于列的情形, 注意到矩阵

$$(\boldsymbol{I}-\boldsymbol{L}\boldsymbol{D}^{-1})-\boldsymbol{U}\boldsymbol{D}^{-1}=(\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U})\boldsymbol{D}^{-1}=\boldsymbol{A}\boldsymbol{D}^{-1},$$

因此, 若 $\boldsymbol{A}$ 为严格对角占优或不可约对角占优, 当 $|\mu|\geqslant 1$ 时, 有

$$(\boldsymbol{I}-\boldsymbol{L}\boldsymbol{D}^{-1})-\frac{1}{\mu}\boldsymbol{U}\boldsymbol{D}^{-1}$$

也为严格对角占优或不可约对角占优. 根据引理 1.1, 当 $|\mu|\geqslant 1$ 时, 有

$$\det\left[(\boldsymbol{I}-\boldsymbol{L}\boldsymbol{D}^{-1})-\frac{1}{\mu}\boldsymbol{U}\boldsymbol{D}^{-1}\right]=\det\left[(\boldsymbol{D}-\boldsymbol{L})-\frac{1}{\mu}\boldsymbol{U}\right]\det(\boldsymbol{D}^{-1})\neq 0,$$

故

$$\det\left[(\boldsymbol{D}-\boldsymbol{L})-\frac{1}{\mu}\boldsymbol{U}\right]\neq 0\Longrightarrow \det[\mu\boldsymbol{I}-(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}]\neq 0,$$

即 $\boldsymbol{B}_S$ 特征值 μ 不满足 $|\mu|\geqslant 1$, 即 $\rho(\boldsymbol{B}_S)<1$, 从而 Gauss–Seidel 迭代法收敛. 证毕. □

可见, 定理 3.15 中 Jacobi 迭代法收敛的充分条件也是 Gauss–Seidel 迭代法收敛的充分条件. 然而, 由于 Gauss–Seidel 迭代法利用了最新迭代信息, 在两种方法都收敛的情形下, Gauss-Seidel 迭代法往往比 Jacobi 迭代法收敛更快, 这是因为 $\boldsymbol{D}-\boldsymbol{L}$ 用到了 $\boldsymbol{A}$ 的更多的信息, 或 $(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{A}$ 比 $\boldsymbol{D}^{-1}\boldsymbol{A}$ 更接近单位矩阵.

定理 3.18 设 $\boldsymbol{A}$ 是 Hermite 正定矩阵, 则 Gauss–Seidel 迭代法收敛.

证明 设 λ 是 $\boldsymbol{B}_S$ 的任一特征值, 对应的特征向量为 $\boldsymbol{y}$. 由 $\boldsymbol{B}_S\boldsymbol{y}=\lambda\boldsymbol{y}$, 得

$$(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}\boldsymbol{y}=\lambda\boldsymbol{y} \implies \boldsymbol{U}\boldsymbol{y}=\lambda(\boldsymbol{D}-\boldsymbol{L})\boldsymbol{y}.$$

两端同时左乘 $\boldsymbol{y}^{\mathrm{H}}$, 得

$$\boldsymbol{y}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{y}=\lambda(\boldsymbol{y}^{\mathrm{H}}\boldsymbol{D}\boldsymbol{y}-\boldsymbol{y}^{\mathrm{H}}\boldsymbol{L}\boldsymbol{y}).$$

因为 $\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}$ 是 Hermite 矩阵, 故 $\boldsymbol{U}=\boldsymbol{L}^{\mathrm{H}}$. 记

$$\boldsymbol{y}^{\mathrm{H}}\boldsymbol{D}\boldsymbol{y}=p,\ \ \boldsymbol{y}^{\mathrm{H}}\boldsymbol{L}\boldsymbol{y}=c+\mathrm{i}\,d\ (c,d\ \text{为实数}),$$

则 $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{y}=c-\mathrm{i}\,d$. 再由 $\boldsymbol{A}$ 正定, 得

$$\begin{aligned}&\boldsymbol{D}\ \text{正定} \implies p=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{D}\boldsymbol{y}>0,\\&\boldsymbol{y}^{\mathrm{H}}(\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U})\boldsymbol{y}=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{y}>0 \implies p-2c>0.\end{aligned}$$

于是有

$$\lambda=\frac{c-\mathrm{i}\,d}{p-(c+\mathrm{i}\,d)},\quad |\lambda|^2=\frac{c^2+d^2}{(p-c)^2+d^2}.$$

由于

$$c^2-(p-c)^2=-p(p-2c)<0,$$

所以 $|\lambda|^2<1$, 从而 $\rho(\boldsymbol{B}_S)<1$, 故 Gauss–Seidel 迭代法收敛. 证毕. □

对具有如式 (3.34) 结构的分块系数矩阵及相应的向量划分, 有 Gauss–Seidel 迭代法的一种推广:

$$\boldsymbol{A}_{ii}\boldsymbol{x}_i^{(k+1)}=\boldsymbol{b}_i-\sum_{r=1}^{i-1}\boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k+1)}-\sum_{r=i+1}^{m}\boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k)},\ i=1,2,\cdots,m;\ k=0,1,2,\cdots,\tag{3.42}$$

此即块 Gauss-Seidel (BGS) 迭代法.

另一种重要推广是双步法. 在第 $k+1$ 步, 作 Gauss–Seidel 迭代并将结果记为 $\boldsymbol{x}^{(k+\frac{1}{2})}$:

$$\boldsymbol{x}^{(k+\frac{1}{2})}=(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)}+(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{b},\ k=0,1,2,\cdots,\tag{3.43}$$

然后各方程按相反顺序利用 $\boldsymbol{x}^{(k+\frac{1}{2})}$ 进行计算:

$$x_i^{(k+1)}=\Big(b_i-\sum_{r=1}^{i-1}a_{ir}x_r^{(k+\frac{1}{2})}-\sum_{r=i+1}^{n}a_{ir}x_r^{(k+1)}\Big)\Big/a_{ii},\ i=n,n-1,\cdots,1;k=0,1,2,\cdots.\tag{3.44}$$

上式可写成矩阵形式

$$\boldsymbol{x}^{(k+1)} = (\boldsymbol{D}-\boldsymbol{U})^{-1}\boldsymbol{L}\boldsymbol{x}^{(k+\frac{1}{2})} + (\boldsymbol{D}-\boldsymbol{U})^{-1}\boldsymbol{b},\ k = 0, 1, 2, \cdots. \tag{3.45}$$

$\boldsymbol{L}$ 与 $\boldsymbol{U}$ 在式 (3.45) 中的作用正好与式 (3.43) 相反. 将式 (3.43) 代入式 (3.45), 得

$$\boldsymbol{x}^{(k+1)} = (\boldsymbol{D}-\boldsymbol{U})^{-1}\boldsymbol{L}(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{U}\boldsymbol{x}^{(k)} + \boldsymbol{f},\ k = 0, 1, 2, \cdots, \tag{3.46}$$

式中:

$$\boldsymbol{f} = (\boldsymbol{D}-\boldsymbol{U})^{-1}\boldsymbol{L}(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{b} + (\boldsymbol{D}-\boldsymbol{U})^{-1}\boldsymbol{b} = (\boldsymbol{D}-\boldsymbol{U})^{-1}\boldsymbol{D}(\boldsymbol{D}-\boldsymbol{L})^{-1}\boldsymbol{b}.$$

式 (3.46) 称为*对称 Gauss-Seidel (SGS) 迭代法*.

3.3 松弛型迭代法

3.3.1 SOR 迭代法

如前面对 Jacobi 迭代法每个迭代步引入外推参数 (松弛因子) 那样, 可在 Gauss–Seidel 迭代法每个迭代步引入松弛因子, 从而得到所谓的 SOR 迭代法.

对 Gauss–Seidel 迭代法的一种重要改进如下:

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega\widetilde{x}_i^{(k+1)},\ i = 1, 2, \cdots;\ k = 0, 1, 2, \cdots, \tag{3.47}$$

式中: $\omega > 0$ 为松弛因子, 是一个可以适当选取的参数, 用来加快收敛速度; $\widetilde{x}_i^{(k+1)}$ 为式 (3.39) 的右端, 将其代入式 (3.47), 得

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\Big(b_i - \sum_{r=1}^{i-1} a_{ir}x_r^{(k+1)} - \sum_{r=i+1}^{n} a_{ir}x_r^{(k)}\Big),$$

$$i = 1, 2, \cdots, n;\ k = 0, 1, \cdots. \tag{3.48}$$

也可以由矩阵分裂导出迭代法的矩阵向量形式. 取

$$\boldsymbol{M} = \frac{1}{\omega}\boldsymbol{D} - \boldsymbol{L},\quad \boldsymbol{N} = \frac{1-\omega}{\omega}\boldsymbol{D} + \boldsymbol{U}, \tag{3.49}$$

则有

$$\Big(\frac{1}{\omega}\boldsymbol{D} - \boldsymbol{L}\Big)\boldsymbol{x}^{(k+1)} = \Big(\frac{1-\omega}{\omega}\boldsymbol{D} + \boldsymbol{U}\Big)\boldsymbol{x}^{(k)} + \boldsymbol{b}, \tag{3.50}$$

或等价地, 有

$$\boldsymbol{x}^{(k+1)} = (\boldsymbol{D}-\omega\boldsymbol{L})^{-1}\big[(1-\omega)\boldsymbol{D} + \omega\boldsymbol{U}\big]\boldsymbol{x}^{(k)} + \omega(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}\boldsymbol{b}, \tag{3.51}$$

这样定义的迭代法称为*逐次超松弛迭代法* (Successive Over Relaxation, SOR 迭代法), 参数 ω 称为*松弛因子*. 适当选取 ω, 可使 SOR 迭代法的收敛速度优于 Gauss–Seidel 迭代法. 尽管 $0 < \omega < 1$ 时应该称为*低松弛*, 但为了方便, 对于任意的 $\omega \in (0, 2)$ 均使用

超松弛这一术语. 注意到, 在实际计算中, 松弛因子 ω 的选取通常是十分困难的. 没有一个通用的选取规则, 只能就某些具有特殊结构的矩阵讨论最佳松弛因子的计算.

SOR 迭代法的迭代矩阵和右端向量分别为

$$\boldsymbol{B}_\omega = (\boldsymbol{D} - \omega\boldsymbol{L})^{-1}\big[(1-\omega)\boldsymbol{D} + \omega\boldsymbol{U}\big], \quad \boldsymbol{f}_\omega = \omega(\boldsymbol{D} - \omega\boldsymbol{L})^{-1}\boldsymbol{b}. \tag{3.52}$$

在 $\boldsymbol{A}$ 的对角元素均非零的条件下 $(\boldsymbol{D} - \omega\boldsymbol{L})^{-1}$ 是存在的. 显然, 当 $\omega = 1$ 时, SOR 迭代法退化为 Gauss–Seidel 迭代法.

为了便于计算机编程实现, 下面给出 SOR 迭代法的具体算法步骤.

算法 3.3 (SOR 迭代法)

步 1, 输入矩阵 $\boldsymbol{A}$, 右端向量 $\boldsymbol{b}$, 初始点 $\boldsymbol{x}^{(0)}$, 精度要求 ε, 最大迭代次数 N, 置 $k := 0$.

步 2, 由式 (3.48) 或式 (3.51) 计算 $\boldsymbol{x}^{(k+1)}$.

步 3, 若 $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k+1)}\|/\|\boldsymbol{b}\| \leqslant \varepsilon$, 则停算, 输出 $\boldsymbol{x}^{(k+1)}$ 作为方程组的近似解.

步 4, 置 $\boldsymbol{x}^{(k)} := \boldsymbol{x}^{(k+1)}$, $k := k+1$, 转步 2.

根据算法 3.3, 编制 MATLAB 程序如下:

```
%SOR迭代法程序-msor.m
function [x,k,err,time]=msor(A,b,w,x,tol,max_it)
if nargin<6, max_it=1000; end
if nargin<5, tol=1.e-5; end
if nargin<4, x=zeros(size(b)); end
tic; bnrm2=norm(b);
r=b-A*x;  %计算初始残差r0=b-Ax
err=norm(r)/bnrm2;
if (err<tol), return; end
D=diag(diag(A)); L=-tril(A,-1); U=-triu(A,1);
for k=1:max_it  % 迭代开始
    x=(D-w*L)\(((1-w)*D+w*U)*x+w*b);
    r=b-A*x;  %计算残差r=b-Ax
    err= norm(r)/bnrm2;
    if (err<=tol), break; end
end
time=toc;
```

关于 SOR 迭代法的收敛性, 有如下的定理.

定理 3.19 对于任何参数 ω, SOR 迭代法的迭代矩阵 $\boldsymbol{B}_\omega$ 满足

$$\rho(\boldsymbol{B}_\omega) \geqslant |1-\omega|. \tag{3.53}$$

证明 在

$$\begin{aligned}\boldsymbol{B}_\omega &= (\boldsymbol{D}-\omega\boldsymbol{L})^{-1}\big[(1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}\big]\\ &= \big(\boldsymbol{I}-\omega\boldsymbol{D}^{-1}\boldsymbol{L}\big)^{-1}\big[(1-\omega)\boldsymbol{I}+\omega\boldsymbol{D}^{-1}\boldsymbol{U}\big]\end{aligned}$$

中, $\boldsymbol{D}^{-1}\boldsymbol{L}$ 和 $\boldsymbol{D}^{-1}\boldsymbol{U}$ 分别是严格下三角和严格上三角矩阵, 所以

$$\det(\boldsymbol{B}_\omega) = \det(\boldsymbol{I}-\omega\boldsymbol{D}^{-1}\boldsymbol{L})^{-1}\det[(1-\omega)\boldsymbol{I}+\omega\boldsymbol{D}^{-1}\boldsymbol{U}] = (1-\omega)^n.$$

再由 $\boldsymbol{B}_\omega$ 的 n 个特征值之积等于 $\det(\boldsymbol{B}_\omega)$, 得

$$\rho(\boldsymbol{B}_\omega) \geqslant (|\det(\boldsymbol{B}_\omega)|)^{\frac{1}{n}} = |1-\omega|.$$

证毕. □

定理 3.20 设 ω 为实参数, 若 SOR 迭代法收敛, 则 $\omega\in(0,2)$.

证明 根据定理 3.1, SOR 迭代法收敛时, $\rho(\boldsymbol{B}_\omega)<1$. 再由定理 3.19, 得 $|1-\omega|<1$, 即 $0<\omega<2$. 证毕. □

定理 3.20 表明, 如果松弛参数 $\omega\notin(0,2)$, 则 SOR 迭代法不收敛. 但反之结论一般不成立, 即 $\omega\in(0,2)$ 不能保证 SOR 迭代法收敛. 然而, 对 Hermite 正定矩阵 (对称正定矩阵), 这一条件是充分必要的.

定理 3.21 设 $\boldsymbol{A}$ 是 Hermite 正定矩阵, 则 SOR 迭代法收敛的充分必要条件是松弛因子 $\omega\in(0,2)$.

证明 仅证明充分性即可. 设 λ 是 $\boldsymbol{B}_\omega$ 的任一特征值, 对应的特征向量为 $\boldsymbol{y}$. 由 $\boldsymbol{B}_\omega\boldsymbol{y}=\lambda\boldsymbol{y}$, 得

$$[(1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}]\boldsymbol{y} = \lambda(\boldsymbol{D}-\omega\boldsymbol{L})\boldsymbol{y}.$$

两端同时左乘 $\boldsymbol{y}^{\mathrm{H}}$, 得

$$(1-\omega)\boldsymbol{y}^{\mathrm{H}}\boldsymbol{D}\boldsymbol{y}+\omega\boldsymbol{y}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{y} = \lambda(\boldsymbol{y}^{\mathrm{H}}\boldsymbol{D}\boldsymbol{y}-\omega\boldsymbol{y}^{\mathrm{H}}\boldsymbol{L}\boldsymbol{y}).$$

因为 $\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}$ 是 Hermite 矩阵, 故 $\boldsymbol{U}=\boldsymbol{L}^{\mathrm{H}}$. 记

$$\boldsymbol{y}^{\mathrm{H}}\boldsymbol{D}\boldsymbol{y}=p,\quad \boldsymbol{y}^{\mathrm{H}}\boldsymbol{L}\boldsymbol{y}=c+\mathrm{i}\,d\quad (c,d\text{ 为实数}),$$

则 $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{U}\boldsymbol{y}=c-\mathrm{i}\,d$. 再由 $\boldsymbol{A}$ 正定, 得

$$\begin{aligned}&\boldsymbol{D}\text{ 正定}\Longrightarrow p=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{D}\boldsymbol{y}>0,\\ &\boldsymbol{y}^{\mathrm{H}}(\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U})\boldsymbol{y}=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{y}>0\Longrightarrow p-2c>0.\end{aligned}$$

于是, 当 $0<\omega<2$ 时, $p-\omega(c+\mathrm{i}\,d)\neq0$, 且有

$$\lambda=\frac{(1-\omega)p+\omega(c-\mathrm{i}\,d)}{p-\omega(c+\mathrm{i}\,d)},\quad |\lambda|^2=\frac{[(1-\omega)p+\omega c]^2+\omega^2d^2}{(p-\omega c)^2+\omega^2d^2}.$$

由于

$$[(1-\omega)p+\omega c]^2-(p-\omega c)^2=-p\omega(2-\omega)(p-2c)<0,$$

所以 $|\lambda|^2<1$, 从而 $\rho(\boldsymbol{B}_\omega)<1$, 故 SOR 迭代法收敛. 证毕. □

定理 3.22 设 $\boldsymbol{A}$ 严格对角占优或不可约对角占优, 则当实参数 ω 满足 $0<\omega\leqslant 1$ 时, SOR 迭代法收敛.

证明 设 $\boldsymbol{A}$ 按行不可约对角占优, 且 λ 为 $\boldsymbol{B}_\omega$ 的特征值. 如果 λ 满足 $|\lambda|\geqslant 1$, 则由 $0<\omega\leqslant 1$, 得 $1-\omega-\lambda\neq 0$. 再由

$$\lambda=(\lambda+\omega-1)+(1-\omega),$$

得

$$|\lambda+\omega-1|\geqslant|\lambda|-(1-\omega)$$

及

$$\Big|\frac{\omega}{\lambda+\omega-1}\Big|\leqslant\Big|\frac{\omega\lambda}{\lambda+\omega-1}\Big|\leqslant\frac{|\omega\lambda|}{|\lambda|-(1-\omega)}\leqslant\frac{\omega|\lambda|}{|\lambda|-(1-\omega)|\lambda|}=1.$$

因为 $\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{U}$ 按行不可约对角占优, 从而

$$\boldsymbol{D}-\frac{\omega\lambda}{\lambda+\omega-1}\boldsymbol{L}-\frac{\omega}{\lambda+\omega-1}\boldsymbol{U}$$

也按行不可约对角占优, 根据引理 1.1, 得

$$\det\Big(\boldsymbol{D}-\frac{\omega\lambda}{\lambda+\omega-1}\boldsymbol{L}-\frac{\omega}{\lambda+\omega-1}\boldsymbol{U}\Big)\neq 0,$$

即

$$\det\big[\lambda(\boldsymbol{D}-\omega\boldsymbol{L})-(1-\omega)\boldsymbol{D}-\omega\boldsymbol{U}\big]\neq 0,$$

或

$$\det\big(\lambda\boldsymbol{I}-(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}[(1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}]\big)\neq 0.$$

故 $\boldsymbol{B}_\omega$ 的特征值 λ 不满足 $|\lambda|\geqslant 1$, 即 $\rho(\boldsymbol{B}_\omega)<1$, 从而 SOR 迭代法收敛.

当 $\boldsymbol{A}$ 为另外三种情形 (按列不可约对角占优、按行 (列) 严格对角占优) 时, 采用和上面类似的方法可以推得 SOR 迭代法收敛. 注意: 列的情形需要考虑矩阵 $\boldsymbol{I}-\boldsymbol{L}\boldsymbol{D}^{-1}-\boldsymbol{U}\boldsymbol{D}^{-1}=\boldsymbol{A}\boldsymbol{D}^{-1}$. 证毕. □

定理 3.23 设 $\boldsymbol{A}$ 为 H 矩阵, 若松弛因子 ω 满足

$$0<\omega<\frac{2}{1+\rho(\boldsymbol{B}_J)},$$

式中: $\boldsymbol{B}_J$ 为 Jacobi 迭代法的迭代矩阵. 则 SOR 迭代法对任意的初始点 $\boldsymbol{x}^{(0)}$ 均收敛.

SOR 迭代法也可推广到分块矩阵的情形. 对具有如式 (3.34) 结构的分块系数矩阵及相应的向量划分, 有 SOR 迭代法的一种推广:

$$\boldsymbol{x}_i^{(k+1)} = (1-\omega)\boldsymbol{x}_i^{(k)} + \omega \boldsymbol{A}_{ii}^{-1}\Big(\boldsymbol{b}_i - \sum_{r=1}^{i-1} \boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k+1)} - \sum_{r=i+1}^{n} \boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k)}\Big),$$

$$i = 1, 2, \cdots, n;\ k = 0, 1, \cdots. \tag{3.54}$$

此即块 SOR (BSOR) 迭代法.

特别地, 对于 2×2 分块的经典鞍点方程组

$$\mathcal{A}\boldsymbol{u} := \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^{\mathrm{T}} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix} := \boldsymbol{b}, \tag{3.55}$$

式中: $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ 为对称正定阵; $\boldsymbol{B} \in \mathbb{R}^{m\times n}$ 为列满秩阵.

注意: 此时系数矩阵 $\mathcal{A}$ 非奇异, 且 Schur 补 $\boldsymbol{S} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{B}$ 对称正定. 给出分裂 $\mathcal{A} = \mathcal{D} - \mathcal{L} - \boldsymbol{U}$, 其中

$$\mathcal{D} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{S} \end{bmatrix},\ \mathcal{L} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} \\ -\boldsymbol{B}^{\mathrm{T}} & \boldsymbol{O} \end{bmatrix},\ \mathcal{U} = \begin{bmatrix} \boldsymbol{O} & -\boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{S} \end{bmatrix},$$

由此得到迭代公式

$$(\mathcal{D} - \omega\mathcal{L})\boldsymbol{u}^{(k+1)} = \big[(1-\omega)\mathcal{D} + \omega\mathcal{U}\big]\boldsymbol{u}^{(k)} + \omega\boldsymbol{b},$$

即有

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \omega\boldsymbol{B}^{\mathrm{T}} & \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{(k+1)} \\ \boldsymbol{y}^{(k+1)} \end{bmatrix} = \begin{bmatrix} (1-\omega)\boldsymbol{A} & -\omega\boldsymbol{B} \\ \boldsymbol{O} & (1-\omega)\boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{(k)} \\ \boldsymbol{y}^{(k)} \end{bmatrix} + \omega \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix}.$$

整理可得

$$\begin{cases} \boldsymbol{x}^{(k+1)} = (1-\omega)\boldsymbol{x}^{(k)} + \omega\boldsymbol{A}^{-1}\big(\boldsymbol{f} - \boldsymbol{B}\boldsymbol{y}^{(k)}\big), \\ \boldsymbol{y}^{(k+1)} = (1-\omega)\boldsymbol{y}^{(k)} + \omega\boldsymbol{S}^{-1}\big(\boldsymbol{g} - \boldsymbol{B}^{\mathrm{T}}\boldsymbol{x}^{(k+1)}\big), \end{cases} \quad k = 0, 1, \cdots.$$

在实际计算中, 可选取对称正定矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 为 $\boldsymbol{A}$ 和 Schur 补 $\boldsymbol{S}$ 的近似矩阵, 即得到所谓的非精确类算法.

3.3.2 SSOR 迭代法

SOR 迭代法的计算公式为

$$(\boldsymbol{D} - \omega\boldsymbol{L})\boldsymbol{x}^{(k+1)} = \big((1-\omega)\boldsymbol{D} + \omega\boldsymbol{U}\big)\boldsymbol{x}^{(k)} + \omega\boldsymbol{b},\ \ k = 0, 1, 2, \cdots. \tag{3.56}$$

它的计算是依赖顺序的, 即它是按从第 1 个到第 n 个分量依次计算的. 一个自然的想法是改变计算顺序, 即按从第 n 个到第 1 个分量逆序计算, 只需将式 (3.56) 中的 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 互换位置即得

$$(\boldsymbol{D} - \omega\boldsymbol{U})\boldsymbol{x}^{(k+1)} = \big((1-\omega)\boldsymbol{D} + \omega\boldsymbol{L}\big)\boldsymbol{x}^{(k)} + \omega\boldsymbol{b},\ \ k = 0, 1, 2, \cdots. \tag{3.57}$$

这种技巧导致一个对称方案, 称其为对称 SOR 迭代法 (Symmetric Successive Over Relaxation, SSOR 迭代法). 具体表示为

$$\begin{cases} (\boldsymbol{D}-\omega\boldsymbol{L})\boldsymbol{x}^{(k+\frac{1}{2})} = \big((1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}\big)\boldsymbol{x}^{(k)}+\omega\boldsymbol{b}, \\ (\boldsymbol{D}-\omega\boldsymbol{U})\boldsymbol{x}^{(k+1)} = \big((1-\omega)\boldsymbol{D}+\omega\boldsymbol{L}\big)\boldsymbol{x}^{(k+\frac{1}{2})}+\omega\boldsymbol{b}, \end{cases} \quad k=0,1,2,\cdots. \tag{3.58}$$

若消去 $\boldsymbol{x}^{(k+\frac{1}{2})}$, 即得到形如式 (3.4) 的迭代法

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{B}_{\mathrm{SSOR}}\boldsymbol{x}^{(k)} + \boldsymbol{f}_{\mathrm{SSOR}}, \tag{3.59}$$

式中:

$$\begin{aligned} \boldsymbol{B}_{\mathrm{SSOR}} &= (\boldsymbol{D}-\omega\boldsymbol{U})^{-1}[(1-\omega)\boldsymbol{D}+\omega\boldsymbol{L}](\boldsymbol{D}-\omega\boldsymbol{L})^{-1}[(1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}], \\ \boldsymbol{f}_{\mathrm{SSOR}} &= \omega(2-\omega)(\boldsymbol{D}-\omega\boldsymbol{U})^{-1}\boldsymbol{D}(\boldsymbol{D}-\omega\boldsymbol{L})^{-1}\boldsymbol{b}. \end{aligned} \tag{3.60}$$

从矩阵分裂的观点, SSOR 迭代法对应于分裂 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$, 其中

$$\boldsymbol{M} = \frac{1}{\omega(2-\omega)}(\boldsymbol{D}-\omega\boldsymbol{L})\boldsymbol{D}^{-1}(\boldsymbol{D}-\omega\boldsymbol{U}), \tag{3.61}$$

$$\boldsymbol{N} = \frac{1}{\omega(2-\omega)}\big((1-\omega)\boldsymbol{D}+\omega\boldsymbol{L}\big)\boldsymbol{D}^{-1}\big((1-\omega)\boldsymbol{D}+\omega\boldsymbol{U}\big). \tag{3.62}$$

此外, 若记 $\boldsymbol{B}_\omega$ 是式 (3.52) 中定义的 SOR 迭代矩阵, $\bar{\boldsymbol{B}}_\omega$ 是由 $\boldsymbol{B}_\omega$ 中的 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 互换得到的. 则迭代矩阵 $\boldsymbol{B}_{\mathrm{SSOR}}$ 可表示为

$$\boldsymbol{B}_{\mathrm{SSOR}} = \boldsymbol{M}^{-1}\boldsymbol{N} = \bar{\boldsymbol{B}}_\omega\boldsymbol{B}_\omega, \tag{3.63}$$

为了便于计算机编程实现, 下面给出 SSOR 迭代法的具体算法步骤.

算法 3.4 (SSOR 迭代法)

步 1, 输入矩阵 $\boldsymbol{A}$, 右端向量 $\boldsymbol{b}$, 初始点 $\boldsymbol{x}^{(0)}$, 精度要求 ε, 最大迭代次数 N, 置 $k:=0$.

步 2, 由式 (3.58) 计算 $\boldsymbol{x}^{(k+1)}$.

步 3, 若 $\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k+1)}\|/\|\boldsymbol{b}\| \leqslant \varepsilon$, 则停算, 输出 $\boldsymbol{x}^{(k+1)}$ 作为方程组的近似解.

步 4, 置 $\boldsymbol{x}^{(k)}:=\boldsymbol{x}^{(k+1)}$, $k:=k+1$, 转步 2.

根据算法 3.4, 编制 MATLAB 程序如下:

```
%SSOR迭代法程序-mssor.m
[x,k,err,time]=mssor(A,b,w,x,tol,max_it)
if nargin<6, max_it=1000; end
if nargin<5, tol=1.e-6; end
if nargin<4, x=zeros(size(b)); end
tic; bnrm2=norm(b);
```

```
r=b-A*x;  %计算初始残差
err=norm(r)/bnrm2;
if (err<tol), return; end
D=diag(diag(A)); L=-tril(A,-1); U=-triu(A,1);
for k=1:max_it  %迭代开始
    x=(D-w*L)\(((1-w)*D+w*U)*x+w*b);
    x=(D-w*U)\(((1-w)*D+w*L)*x+w*b);
    r=b-A*x;  %计算残差
    err=norm(r)/bnrm2;
    if (err<= tol), break; end
end
time=toc;
```

关于 SSOR 迭代法, 有下面的收敛性定理.

定理 3.24 SSOR 迭代法收敛的一个必要条件是 $|\omega-1|<1$. 对于 $\omega\in\mathbb{R}$, 条件变为 $\omega\in(0,2)$.

定理 3.25 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 为具有正对角元的 Hermite 矩阵. 则对任意的 $\omega\in(0,2)$, SSOR 迭代矩阵 $\boldsymbol{B}_{\text{SSOR}}$ 具有实非负特征值. 此外, 若 $\boldsymbol{A}$ 是对称正定的, 则 SSOR 迭代法收敛. 反之, 若 SSOR 迭代法收敛且 $\omega\in\mathbb{R}$, 则 $\omega\in(0,2)$ 且 $\boldsymbol{A}$ 是对称正定的.

注 3.2 SSOR 迭代法实质上就是将 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 等同看待连续两次使用 SOR 迭代法. 这样做好处有两个:

(1) 某些特殊问题, 用 SOR 迭代法不收敛, 但依然可构造出收敛的 SSOR 迭代法.

(2) 一般来说, SOR 迭代法的渐近收敛速度对松弛因子 ω 的选择是非常敏感的, 而 SSOR 却不敏感.

SSOR 迭代法的一种推广是它的两个半步迭代格式中使用不同的松弛因子:

$$\begin{cases}(\boldsymbol{D}-\omega_1\boldsymbol{L})\boldsymbol{x}^{(k+\frac{1}{2})}=\big((1-\omega_1)\boldsymbol{D}+\omega_1\boldsymbol{U}\big)\boldsymbol{x}^{(k)}+\omega_1\boldsymbol{b},\\ (\boldsymbol{D}-\omega_2\boldsymbol{U})\boldsymbol{x}^{(k+1)}=\big((1-\omega_2)\boldsymbol{D}+\omega_2\boldsymbol{L}\big)\boldsymbol{x}^{(k+\frac{1}{2})}+\omega_2\boldsymbol{b},\end{cases}\quad k=0,1,2,\cdots. \tag{3.64}$$

若消去 $\boldsymbol{x}^{(k+\frac{1}{2})}$, 得

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}_{\omega_1,\omega_2}\boldsymbol{x}^{(k)}+\boldsymbol{f}_{\omega_1,\omega_2}, \tag{3.65}$$

式中:

$$\begin{aligned}\boldsymbol{B}_{\omega_1,\omega_2}&=(\boldsymbol{D}-\omega_2\boldsymbol{U})^{-1}[(1-\omega_2)\boldsymbol{D}+\omega_2\boldsymbol{L}](\boldsymbol{D}-\omega_1\boldsymbol{L})^{-1}[(1-\omega_1)\boldsymbol{D}+\omega_1\boldsymbol{D})],\\ \boldsymbol{f}_{\omega_1,\omega_2}&=(\omega_1+\omega_2-\omega_1\omega_2)(\boldsymbol{D}-\omega_2\boldsymbol{U})^{-1}\boldsymbol{D}(\boldsymbol{D}-\omega_1\boldsymbol{L})^{-1}\boldsymbol{b}.\end{aligned} \tag{3.66}$$

这种推广得到的迭代法通常称为不对称 SOR 迭代法, 简记为 USSOR 迭代法.

进一步, SSOR 迭代法也可推广到分块矩阵的情形. 对具有如式 (3.34) 结构的分块系数矩阵及相应的向量划分, 有 SSOR 迭代法的另一种推广:

$$\boldsymbol{x}_i^{(k+\frac{1}{2})} = (1-\omega)\boldsymbol{x}_i^{(k)} + \omega \boldsymbol{A}_{ii}^{-1}\Big(\boldsymbol{b}_i - \sum_{r=1}^{i-1} \boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k+1)} - \sum_{r=i+1}^{n} \boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k)}\Big),\ i = 1, 2, \cdots, n,$$

$$\boldsymbol{x}_i^{(k+1)} = (1-\omega)\boldsymbol{x}_i^{(k+\frac{1}{2})} + \omega \boldsymbol{A}_{ii}^{-1}\Big(\boldsymbol{b}_i - \sum_{r=1}^{i-1} \boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k+\frac{1}{2})} - \sum_{r=i+1}^{n} \boldsymbol{A}_{ir}\boldsymbol{x}_r^{(k+1)}\Big),$$

$$i = n, n-1, \cdots, 1;\quad k = 0, 1, 2, \cdots. \tag{3.67}$$

此即块 SSOR (BSSOR) 迭代法.

例 3.2 用不同的迭代法求解 n 元线性方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, 其中

$$\boldsymbol{A} = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{bmatrix},\quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 2 \\ \vdots \\ 2 \\ 3 \end{bmatrix}. \tag{3.68}$$

方程的精确解为 $\boldsymbol{x}^* = (1, 1, \cdots, 1)^{\mathrm{T}}$. 取 $n = 2^{12} - 1 = 4095$, 初始向量为零向量, 容许误差为 10^{-10}.

解 用 Jacobi, GS, SOR 和 SSOR 四种迭代法进行测试, 取松弛因子 $\omega = 1.1$, 得到计算结果如表 3.1 所示.

表 3.1 Jacobi, GS, SOR 和 SSOR 四种迭代法的数值比较

迭代法	迭代次数(k)	CPU 时间	相对残差 ($\|\boldsymbol{r}^{(k)}\|_2/\|\boldsymbol{b}\|_2$)
Jacobi	34	0.0043	5.8104e-11
GS	21	0.0030	9.5383e-11
SOR	17	0.0049	3.4644e-11
SSOR	9	0.0051	8.0601e-12

3.3.3 AOR 迭代法

有一种技术可以对收敛的迭代格式进行“加速”, 或者使得不收敛的迭代格式变得收敛. 这种“加速”技术通常是通过引进一个“加速”或“松弛”参数 $\gamma \in \mathbb{C}\backslash\{0\}$ 来实现的. 基于矩阵分裂 $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$, 考虑一个新的矩阵 $\boldsymbol{M}_\gamma = \dfrac{1}{\gamma}\boldsymbol{M}$. 设原迭代法的迭代格式为 $\boldsymbol{x}^{(k+1)} = \boldsymbol{B}\boldsymbol{x}^{(k)} + \boldsymbol{f}$, 则新迭代法的迭代格式为

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{B}_\gamma \boldsymbol{x}^{(k)} + \boldsymbol{f}_\gamma,\quad \boldsymbol{B}_\gamma = (1-\gamma)\boldsymbol{I} + \gamma\boldsymbol{B},\quad \boldsymbol{f}_\gamma = \gamma\boldsymbol{f}.$$

参数 $\gamma \in \mathbb{C}\backslash\{0\}$ 称为外推参数, 且相应的格式称为原格式的外推. 最优外推参数的确定一般需要较强的假设条件并涉及原迭代矩阵 $\boldsymbol{B}$ 的谱 $\sigma(\boldsymbol{B})$ 的有关信息.

利用外推法的思想, Hadjidimas (哈吉迪马斯) 于 1978 年引进一种具有两个松弛参数 r 和 ω 的 SOR 类迭代法, 称为*加速超松弛方法* (简记为 AOR 方法), 其定义如下:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{B}_{r,\omega}\boldsymbol{x}^{(k)} + \boldsymbol{f}_{r,\omega}, \tag{3.69}$$

式中:

$$\boldsymbol{B}_{r,\omega} = (\boldsymbol{D} - r\boldsymbol{L})^{-1}\big[(1-\omega)\boldsymbol{D} + (\omega - r)\boldsymbol{L} + \omega\boldsymbol{U}\big], \quad \boldsymbol{f}_{r,\omega} = \omega(\boldsymbol{D} - r\boldsymbol{L})^{-1}\boldsymbol{b}. \tag{3.70}$$

容易证明: 当 $r = 0$ 时, AOR 方法就是具有外推参数 ω 的外推 Jacobi 方法; 当 $r \neq 0$ 时, AOR 方法就是具有外推参数 $s = \omega/r$ 的外推 SOR 方法, 其中 r 是原 SOR 方法的松弛因子. 此外, 显然前面的 Jacobi, Gauss–Seidel, SOR 方法等均可认为是 AOR 方法的特殊情形:

(1) 当 $r = 0, \omega = 1$ 时, AOR 方法即为 Jacobi 方法.

(2) 当 $r = \omega = 1$ 时, AOR 方法即为 Gauss–Seidel 方法.

(3) 当 $r = \omega \neq 0$ 时, AOR 方法即为 SOR 方法.

关于 AOR 方法, 有下面的收敛性定理.

定理 3.26 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 为 Hermite 矩阵, 且 $\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{L} - \boldsymbol{L}^{\mathrm{H}}$, $\boldsymbol{D}$ 是 Hermite 正定的, $\det(\boldsymbol{D} - r\boldsymbol{L}) \neq 0$, $\forall\, \omega \in (0,2)$ 且 $r \in \big(\omega + (2-\omega)/\mu_{\min}, \omega + (2-\omega)/\mu_{\max}\big)$, 其中 $\mu_{\min} < 0 < \mu_{\max}$ 为矩阵 $\boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{L}^{\mathrm{H}})$ 的最小和最大特征值. 则 $\rho(\boldsymbol{B}_{r,\omega}) < 1$ 当且仅当 $\boldsymbol{A}$ 是正定的.

类似于 SSOR 方法, 可以推广 AOR 方法得到对称 AOR方法 (简记为 SAOR 方法). 具体形式为

$$\begin{cases} (\boldsymbol{D} - r\boldsymbol{L})\boldsymbol{x}^{(k+\frac{1}{2})} = \big[(1-\omega)\boldsymbol{D} + (\omega - r)\boldsymbol{L} + \omega\boldsymbol{U}\big]\boldsymbol{x}^{(k)} + \omega\boldsymbol{b}, \\ (\boldsymbol{D} - r\boldsymbol{U})\boldsymbol{x}^{(k+1)} = \big[(1-\omega)\boldsymbol{D} + (\omega - r)\boldsymbol{U} + \omega\boldsymbol{L}\big]\boldsymbol{x}^{(k+\frac{1}{2})} + \omega\boldsymbol{b}, \end{cases} \quad k = 0, 1, 2, \cdots.$$

上式可以化简为

$$\begin{cases} (\boldsymbol{D} - r\boldsymbol{L})\boldsymbol{x}^{(k+\frac{1}{2})} = (\boldsymbol{D} - r\boldsymbol{L} - \omega\boldsymbol{A})\boldsymbol{x}^{(k)} + \omega\boldsymbol{b}, \\ (\boldsymbol{D} - r\boldsymbol{U})\boldsymbol{x}^{(k+1)} = (\boldsymbol{D} - r\boldsymbol{U} - \omega\boldsymbol{A})\boldsymbol{x}^{(k+\frac{1}{2})} + \omega\boldsymbol{b}, \end{cases} \quad k = 0, 1, 2, \cdots. \tag{3.71}$$

消去 $\boldsymbol{x}^{(k+\frac{1}{2})}$, 得

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{B}_{\mathrm{SAOR}}\boldsymbol{x}^{(k)} + \boldsymbol{f}_{\mathrm{SAOR}}, \tag{3.72}$$

式中:

$$\begin{aligned} \boldsymbol{B}_{\mathrm{SAOR}} &= (\boldsymbol{D} - r\boldsymbol{L})^{-1}(\boldsymbol{D} - r\boldsymbol{U} - \omega\boldsymbol{A})(\boldsymbol{D} - r\boldsymbol{U})^{-1}(\boldsymbol{D} - r\boldsymbol{L} - \omega\boldsymbol{A}), \\ \boldsymbol{f}_{\mathrm{SAOR}} &= \omega(\boldsymbol{D} - \omega\boldsymbol{U})^{-1}(2\boldsymbol{D} - r\boldsymbol{L} - \omega\boldsymbol{A})(\boldsymbol{D} - \omega\boldsymbol{L})^{-1}\boldsymbol{b}. \end{aligned}$$

同样可以将 AOR 方法推广到非对称 AOR 方法 (UAOR 方法) 和分块 AOR 方法 (BAOR 方法) 的情形, 这里不再详述.

3.4 HSS 迭代法

本节内容主要取材于文献 [21, 22]. HSS 迭代法是中国学者白中治等提出的用于求解大型稀疏非 Hermite 正定方程组的一种有效的数值方法. 它是基于系数矩阵的 Hermite 和反 Hermite 分裂 (Hermitian / skew-Hermitian splitting, HSS), 其中包括 HSS 迭代和非精确 HSS (IHSS) 迭代, IHSS 在 HSS 外迭代的每一步用某种 Krylov 子空间方法 (见第 4 章) 求解子问题作为其内迭代过程. 理论分析显示 HSS 方法无条件地收敛到线性方程组的唯一解. 目前, 基于 HSS 的各种迭代方法及与不同预处理方式的结合成为一个研究热点.

3.4.1 HSS 和 IHSS 方法

许多科学计算问题需要求解 (复) 线性方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{3.73}$$

式中: $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 为非奇异的大型稀疏非 Hermite 正定矩阵; $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{C}^n$.

将矩阵 $\boldsymbol{A}$ 进行 Hermite–反 Hermite 分裂 (HS) 分裂

$$\boldsymbol{A} = \boldsymbol{H} + \boldsymbol{S}, \tag{3.74}$$

式中:

$$\boldsymbol{H} = \frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{H}}), \quad \boldsymbol{S} = \frac{1}{2}(\boldsymbol{A} - \boldsymbol{A}^{\mathrm{H}}). \tag{3.75}$$

下面将研究基于此特殊矩阵分裂的求解线性方程组 (3.73) 的有效迭代方法.

基于 HS 分裂式 (3.74), 文献 [21] 提出了求解线性方程组 (3.73) 的 Hermite–反 Hermite 分裂迭代法 (简称 HSS 迭代法), 其形式如下:

算法 3.5 (HSS 迭代法)

步 1, 给定初始向量 $\boldsymbol{x}^{(0)}$, 选取参数 $\alpha > 0$, 容许误差 $\varepsilon > 0$, 置 $k := 0$.

步 2, 计算

$$\begin{cases} (\alpha\boldsymbol{I} + \boldsymbol{H})\boldsymbol{x}^{(k+\frac{1}{2})} = (\alpha\boldsymbol{I} - \boldsymbol{S})\boldsymbol{x}^{(k)} + \boldsymbol{b}, \\ (\alpha\boldsymbol{I} + \boldsymbol{S})\boldsymbol{x}^{(k+1)} = (\alpha\boldsymbol{I} - \boldsymbol{H})\boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{b}, \end{cases}$$

步 3, 计算残差 $\boldsymbol{r}^{(k+1)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k+1)}$. 若 $\|\boldsymbol{r}^{(k+1)}\|_2 \leqslant \varepsilon$, 停算. 否则, 置 $k := k+1$, 转步 2.

明显地, HSS 迭代法的每一步都在矩阵 $\boldsymbol{A}$ 的 Hermite 部分和反 Hermite 部分之间进行交替, 这类似于求解偏微分方程的交替方向隐式 (ADI) 迭代法.

注意到可以调换上述 HSS 迭代法中矩阵 $\boldsymbol{H}$ 和 $\boldsymbol{S}$ 的角色, 即先解关于 $\alpha\boldsymbol{I} + \boldsymbol{S}$ 的方程组, 然后再解关于 $\alpha\boldsymbol{I} + \boldsymbol{H}$ 的方程组. 每个 HSS 迭代的两个半步需要精确求解具有 n 阶系数矩阵 $\alpha\boldsymbol{I} + \boldsymbol{H}$ 和 $\alpha\boldsymbol{I} + \boldsymbol{S}$ 的方程组. 然而, 这在实际实现中是耗时和不切实际的. 为了改进 HSS 迭代法的计算效率, 例如, 在 HSS 迭代法的每一步利用共轭梯度法 (CG 法) 求解关于 $\alpha\boldsymbol{I} + \boldsymbol{H}$ 的方程组 (因为 $\alpha\boldsymbol{I} + \boldsymbol{H}$ 是 Hermite 正定的), 并用某种 Krylov 子空间方法 (如 GMRES 方法, 见第 4 章) 解关于 $\alpha\boldsymbol{I} + \boldsymbol{S}$ 的方程组, 直到预

先设定的精度. 这导致一种非精确的 Hermite–反 Hermite 分裂迭代法, 简称 IHSS 迭代法. 内迭代的阈值 (或内迭代步数) 可随外迭代方案的不同而变化. 因此, IHSS 迭代实际上是求解式 (3.73) 的一种非定常迭代法. 对给定正常数 α, IHSS 迭代法形式如下.

算法 3.6 (IHSS 迭代法)

步 1, 给定初始向量 $\boldsymbol{x}^{(0)}$, 选取参数 $\alpha > 0$, 容许误差 $\varepsilon > 0$, 置 $k := 0$.

步 2, 以 $\boldsymbol{x}^{(k)}$ 为初值, 利用一种内迭代 (如 CG 法) 近似求解 $\boldsymbol{x}^{(k+\frac{1}{2})}$:

$$(\alpha \boldsymbol{I} + \boldsymbol{H})\boldsymbol{x}^{(k+\frac{1}{2})} \approx (\alpha \boldsymbol{I} - \boldsymbol{S})\boldsymbol{x}^{(k)} + \boldsymbol{b}.$$

步 3, 以 $\boldsymbol{x}^{(k+\frac{1}{2})}$ 为初值, 利用一种内迭代 (如某种 krylov 子空间方法) 近似求解 $\boldsymbol{x}^{(k+1)}$:

$$(\alpha \boldsymbol{I} + \boldsymbol{S})\boldsymbol{x}^{(k+1)} \approx (\alpha \boldsymbol{I} - \boldsymbol{H})\boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{b}.$$

步 4, 计算残差 $\boldsymbol{r}^{(k+1)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k+1)}$. 若 $\|\boldsymbol{r}^{(k+1)}\|_2 \leqslant \varepsilon$, 停算. 否则, 置 $k := k+1$, 转步 2.

为了便于数值实现和理论分析, 可将上面的 IHSS 迭代法重写为如下等价形式: 其中 $\|\cdot\|$ 是某种向量范数, 并记原方程组的残差为 $\boldsymbol{r}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}$, $\boldsymbol{r}^{(k+\frac{1}{2})} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k+\frac{1}{2})}$, 内迭代的残差分别记为

$$\boldsymbol{p}^{(k)} = \boldsymbol{r}^{(k)} - (\alpha \boldsymbol{I} + \boldsymbol{H})\boldsymbol{z}^{(k)}; \tag{3.76}$$

$$\boldsymbol{q}^{(k+\frac{1}{2})} = \boldsymbol{r}^{(k+\frac{1}{2})} - (\alpha \boldsymbol{I} + \boldsymbol{S})\boldsymbol{z}^{(k+\frac{1}{2})}. \tag{3.77}$$

则可以改写算法 3.6 为如下形式.

算法 3.7 (IHSS 迭代法)

步 1, 给定初始向量 $\boldsymbol{x}^{(0)}$, 选取参数 $\alpha > 0$, 容许误差 $\varepsilon > 0$, 置 $k := 0$.

步 2, 以 $\boldsymbol{x}^{(k)}$ 为初值, 迭代求解 $(\alpha \boldsymbol{I} + \boldsymbol{H})\boldsymbol{z}^{(k)} = \boldsymbol{r}^{(k)}$, 直到 $\|\boldsymbol{p}^{(k)}\| \leqslant \varepsilon_k \|\boldsymbol{r}^{(k)}\|$.

步 3, 计算 $\boldsymbol{x}^{(k+\frac{1}{2})} = \boldsymbol{x}^{(k)} + \boldsymbol{z}^{(k)}$.

步 4, 以 $\boldsymbol{x}^{(k+\frac{1}{2})}$ 为初值, 迭代求解 $(\alpha \boldsymbol{I} + \boldsymbol{S})\boldsymbol{z}^{(k+\frac{1}{2})} = \boldsymbol{r}^{(k+\frac{1}{2})}$, 直到 $\|\boldsymbol{q}^{(k+\frac{1}{2})}\| \leqslant \eta_k \|\boldsymbol{r}^{(k+\frac{1}{2})}\|$.

步 5, 计算 $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{z}^{(k+\frac{1}{2})}$.

步 6, 计算残差 $\boldsymbol{r}^{(k+1)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k+1)}$. 若 $\|\boldsymbol{r}^{(k+1)}\|_2 \leqslant \varepsilon$, 停算. 否则, 置 $k := k+1$, 转步 2.

事实上, 这里构造了求解非 Hermite 正定方程组的一个一般的迭代法框架. 在此迭代框架下有一些组合. 可精确或非精确地求解 Hermite 部分, 也可精确或非精确地求解反 Hermite 部分. 最佳选择依赖于 Hermite 和反 Hermite 矩阵的结构. 所得的相应精确或非精确 Hermite–反 Hermite 分裂迭代的收敛性理论可类似于下面的分析来建立, 只要做少许的修改即可.

下面考虑 HSS 和 IHSS 迭代法的收敛性和收敛速度方面的一些理论结果. 首先注意到 HSS 迭代法可推广到两步分裂迭代框架, 下面的引理先给出两步分裂迭代的一个一般性收敛条件.

引理 3.1 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, $\boldsymbol{A}=\boldsymbol{M}_i-\boldsymbol{N}_i\,(i=1,2)$ 为矩阵 $\boldsymbol{A}$ 的两个分裂. $\boldsymbol{x}^{(0)}$ 为给定的初始向量. 如果 $\{\boldsymbol{x}^{(k)}\}$ 是由下面两步分裂迭代法所定义的序列

$$\begin{cases}\boldsymbol{M}_1\boldsymbol{x}^{(k+\frac{1}{2})}=\boldsymbol{N}_1\boldsymbol{x}^{(k)}+\boldsymbol{b},\\ \boldsymbol{M}_2\boldsymbol{x}^{(k+1)}=\boldsymbol{N}_2\boldsymbol{x}^{(k+\frac{1}{2})}+\boldsymbol{b},\end{cases}\qquad k=0,1,2,\cdots.$$

则

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{M}_2^{-1}\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{x}^{(k)}+\boldsymbol{M}_2^{-1}(\boldsymbol{M}_1+\boldsymbol{N}_2)\boldsymbol{M}_1^{-1}\boldsymbol{b},\quad k=0,1,2,\cdots.$$

进一步, 如果迭代矩阵 $\boldsymbol{M}_2^{-1}\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1$ 的谱半径 $\rho(\boldsymbol{M}_2^{-1}\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1)$ 小于 1, 则迭代序列 $\{\boldsymbol{x}^{(k)}\}$ 对所有初始向量 $\boldsymbol{x}^{(0)}\in\mathbb{C}^n$ 都收敛于线性方程组 (3.73) 的唯一解 $\boldsymbol{x}^*\in\mathbb{C}^n$.

对于 HSS 迭代法, 应用引理 3.1 及矩阵谱的相似不变性易得如下定理.

定理 3.27 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 为正定矩阵, $\boldsymbol{H}=\dfrac{1}{2}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{H}})$ 和 $\boldsymbol{S}=\dfrac{1}{2}(\boldsymbol{A}-\boldsymbol{A}^{\mathrm{H}})$ 为其 Hermite 和反 Hermite 部分, 且 α 是一个正常数. 则 HSS 迭代矩阵 $\boldsymbol{M}(\alpha)$ 为

$$\boldsymbol{M}(\alpha)=(\alpha\boldsymbol{I}+\boldsymbol{S})^{-1}(\alpha\boldsymbol{I}-\boldsymbol{H})(\alpha\boldsymbol{I}+\boldsymbol{H})^{-1}(\alpha\boldsymbol{I}-\boldsymbol{S}),\tag{3.78}$$

且其谱半径 $\rho(\boldsymbol{M}(\alpha))$ 有上界

$$\sigma(\alpha)\equiv\max_{\lambda_i\in\lambda(\boldsymbol{H})}\left|\frac{\alpha-\lambda_i}{\alpha+\lambda_i}\right|,$$

式中: $\lambda(\boldsymbol{H})$ 为矩阵 $\boldsymbol{H}$ 的谱集, 因此成立

$$\rho(\boldsymbol{M}(\alpha))\leqslant\sigma(\alpha)<1,\quad\forall\,\alpha>0,$$

即 HSS 迭代收敛到线性方程组 (3.73) 的唯一解 $\boldsymbol{x}^*\in\mathbb{C}^n$.

证明 在引理 3.1 中, 记

$$\boldsymbol{M}_1=\alpha\boldsymbol{I}+\boldsymbol{H},\boldsymbol{N}_1=\alpha\boldsymbol{I}-\boldsymbol{S},\ \boldsymbol{M}_2=\alpha\boldsymbol{I}+\boldsymbol{S},\ \boldsymbol{N}_2=\alpha\boldsymbol{I}-\boldsymbol{H},$$

注意到对任意的 $\alpha>0$, $\alpha\boldsymbol{I}+\boldsymbol{H}$ 和 $\alpha\boldsymbol{I}+\boldsymbol{S}$ 是非奇异的, 则由引理 3.1 立即可得式 (3.78).

进一步, 利用矩阵谱的相似不变性, 得

$$\begin{aligned}\rho(\boldsymbol{M}(\alpha))&=\rho((\alpha\boldsymbol{I}+\boldsymbol{S})^{-1}(\alpha\boldsymbol{I}-\boldsymbol{H})(\alpha\boldsymbol{I}+\boldsymbol{H})^{-1}(\alpha\boldsymbol{I}-\boldsymbol{S}))\\&=\rho((\alpha\boldsymbol{I}-\boldsymbol{H})(\alpha\boldsymbol{I}+\boldsymbol{H})^{-1}(\alpha\boldsymbol{I}-\boldsymbol{S})(\alpha\boldsymbol{I}+\boldsymbol{S})^{-1})\\&\leqslant\|(\alpha\boldsymbol{I}-\boldsymbol{H})(\alpha\boldsymbol{I}+\boldsymbol{H})^{-1}\|_2\cdot\|(\alpha\boldsymbol{I}-\boldsymbol{S})(\alpha\boldsymbol{I}+\boldsymbol{S})^{-1}\|_2.\end{aligned}$$

记 $\boldsymbol{Q}(\alpha)=(\alpha\boldsymbol{I}-\boldsymbol{S})(\alpha\boldsymbol{I}+\boldsymbol{S})^{-1}$, 注意到 $\boldsymbol{S}^{\mathrm{H}}=-\boldsymbol{S}$, 有

$$\begin{aligned}\boldsymbol{Q}(\alpha)^{\mathrm{H}}\boldsymbol{Q}(\alpha)&=(\alpha\boldsymbol{I}-\boldsymbol{S})^{-1}(\alpha\boldsymbol{I}+\boldsymbol{S})(\alpha\boldsymbol{I}-\boldsymbol{S})(\alpha\boldsymbol{I}+\boldsymbol{S})^{-1}\\&=(\alpha\boldsymbol{I}-\boldsymbol{S})^{-1}(\alpha\boldsymbol{I}-\boldsymbol{S})(\alpha\boldsymbol{I}+\boldsymbol{S})(\alpha\boldsymbol{I}+\boldsymbol{S})^{-1}=\boldsymbol{I},\end{aligned}$$

即 $\boldsymbol{Q}(\alpha)$ 是酉矩阵, 故有 $\|\boldsymbol{Q}(\alpha)\|_2 = 1$. 从而, 有

$$\rho(\boldsymbol{M}(\alpha)) \leqslant \|(\alpha\boldsymbol{I} - \boldsymbol{H})(\alpha\boldsymbol{I} + \boldsymbol{H})^{-1}\|_2 = \max_{\lambda_i \in \lambda(\boldsymbol{H})} \left|\frac{\alpha - \lambda_i}{\alpha + \lambda_i}\right| \equiv \sigma(\alpha).$$

由于 $\lambda_i > 0\,(i = 1, 2, \cdots, n)$ 及 $\alpha > 0$, 容易推得 $\rho(\boldsymbol{M}(\alpha)) \leqslant \sigma(\alpha) < 1$. 证毕. □

定理 3.27 表明 HSS 迭代法的收敛速度以 $\sigma(\alpha)$ 为上界, 而 $\sigma(\alpha)$ 仅依赖于 $\boldsymbol{A}$ 的 Hermite 部分 $\boldsymbol{H}$ 的谱 (而不依赖于反 Hermite 部分 $\boldsymbol{S}$ 的谱或系数矩阵 $\boldsymbol{A}$ 的谱).

现引入向量范数 $\|\boldsymbol{x}\|_* = \|(\alpha\boldsymbol{I} + \boldsymbol{S})\boldsymbol{x}\|_2$ $(\forall\, \boldsymbol{x} \in \mathbb{C}^n)$ 且其诱导矩阵范数表示为

$$\|\boldsymbol{A}\|_* = \|(\alpha\boldsymbol{I} + \boldsymbol{S})\boldsymbol{A}(\alpha\boldsymbol{I} + \boldsymbol{S})^{-1}\|_2 \ (\forall\, \boldsymbol{A} \in \mathbb{C}^{n\times n}),$$

则由定理 3.27 的证明可见

$$\|\boldsymbol{M}(\alpha)\|_* = \|(\alpha\boldsymbol{I} - \boldsymbol{H})(\alpha\boldsymbol{I} + \boldsymbol{H})^{-1}(\alpha\boldsymbol{I} - \boldsymbol{S})(\alpha\boldsymbol{I} + \boldsymbol{S})^{-1}\|_2 \leqslant \sigma(\alpha),$$

从而

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|_* \leqslant \sigma(\alpha)\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_*, \quad k = 0, 1, 2, \cdots.$$

因此, $\sigma(\alpha)$ 也是 HSS 迭代压缩因子在 $\|\cdot\|_*$ 范数意义下的一个上界. 如果 Hermite 部分 $\boldsymbol{H}$ 的特征值的上界和下界是已知的, 则可得 $\sigma(\alpha)$ (或 $\rho(\boldsymbol{M}(\alpha))$ 和 $\|\boldsymbol{M}(\alpha)\|_*$ 之上界) 的最优参数. 此事实在下面的推论中陈述.

推论 3.2 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 为正定矩阵, $\boldsymbol{H} = \frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{H}})$ 和 $\boldsymbol{S} = \frac{1}{2}(\boldsymbol{A} - \boldsymbol{A}^{\mathrm{H}})$ 为其 Hermite 和反 Hermite 部分, 且 $\lambda_{\min}$ 和 $\lambda_{\max}$ 分别为矩阵 $\boldsymbol{H}$ 的最小和最大特征值, α 是一个正常数. 则

$$\alpha^* \equiv \arg\min_{\alpha} \left\{ \max_{\lambda_{\min} \leqslant \lambda \leqslant \lambda_{\max}} \left|\frac{\alpha - \lambda}{\alpha + \lambda}\right| \right\} = \sqrt{\lambda_{\min}\lambda_{\max}},$$

且

$$\sigma(\alpha^*) = \frac{\sqrt{\lambda_{\max}} - \sqrt{\lambda_{\min}}}{\sqrt{\lambda_{\max}} + \sqrt{\lambda_{\min}}} = \frac{\sqrt{\kappa(\boldsymbol{H})} - 1}{\sqrt{\kappa(\boldsymbol{H})} + 1},$$

式中: $\kappa(\boldsymbol{H}) = \lambda_{\max}/\lambda_{\min}$ 为 $\boldsymbol{H}$ 的谱条件数.

证明 注意到对任意的 $\alpha > 0$, 函数 $f(\lambda) = \dfrac{\alpha - \lambda}{\alpha + \lambda}$ 关于 λ 是单调递减的 ($f'(\lambda) = -2\alpha/(\alpha + \alpha) < 0$), 故有

$$\sigma(\alpha) = \max\left\{ \left|\frac{\alpha - \lambda_{\min}}{\alpha + \lambda_{\min}}\right|, \ \left|\frac{\alpha - \lambda_{\max}}{\alpha + \lambda_{\max}}\right| \right\}. \tag{3.79}$$

若 α^* 是 $\sigma(\alpha)$ 的极小点, 则必有 $\alpha^* = \lambda_{\min} > 0$, $\alpha^* - \lambda_{\max} < 0$,

$$\frac{\alpha^* - \lambda_{\min}}{\alpha^* + \lambda_{\min}} = -\frac{\alpha^* - \lambda_{\max}}{\alpha^* + \lambda_{\max}}.$$

从上式解得

$$\alpha^* = \sqrt{\lambda_{\min}\lambda_{\max}},$$

从而推论的结论成立. 证毕. □

这里强调在推论 3.2 中, 最优参数 α^* 极小化迭代矩阵谱半径的上界 $\sigma(\alpha)$, 而不是极小化迭代矩阵的谱半径. 推论表明当使用最优参数 α^* 时, HSS 迭代法的收敛速度上界与共轭梯度方法的大致相同, 且当 $\boldsymbol{A}$ 为 Hermite 时, 它们是相同的. 应该指出, 当系数矩阵 $\boldsymbol{A}$ 为正规矩阵时, 有 $\boldsymbol{HS} = \boldsymbol{SH}$, 因此 $\rho(\boldsymbol{M}(\alpha)) = \|\boldsymbol{M}(\alpha)\|_* = \sigma(\alpha)$, 此时最优参数 α^* 极小化所有这三个量.

下面的定理在更一般的情况下分析 IHSS 迭代法. 特别地, 考虑对两步分裂技术的非精确迭代 (比较引理 3.1). 为此, 将 $\|\cdot\|_*$ 推广到 $\|\cdot\|_{\boldsymbol{M}_2}$, 现定义为 $\|\boldsymbol{x}\|_{\boldsymbol{M}_2} = \|\boldsymbol{M}_2\boldsymbol{x}\|_2\,(\forall\,\boldsymbol{x}\in\mathbb{C}^n)$ 且它诱导矩阵范数为 $\|\boldsymbol{A}\|_{\boldsymbol{M}_2} = \|\boldsymbol{M}_2\boldsymbol{A}\boldsymbol{M}_2^{-1}\|_2\,(\forall\,\boldsymbol{A}\in\mathbb{C}^{n\times n})$.

定理 3.28 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 且 $\boldsymbol{A} = \boldsymbol{M}_i - \boldsymbol{N}_i\ (i=1,2)$ 为 $\boldsymbol{A}$ 的两个分裂. 如果 $\{\boldsymbol{x}^{(k)}\}$ 是一个如下定义的迭代序列

$$\boldsymbol{x}^{(k+\frac{1}{2})} = \boldsymbol{x}^{(k)} + \boldsymbol{z}^{(k)},\quad \boldsymbol{M}_1\boldsymbol{z}^{(k)} = \boldsymbol{r}^{(k)} - \boldsymbol{p}^{(k)} \tag{3.80}$$

满足 $\|\boldsymbol{p}^{(k)}\|_2 \leqslant \varepsilon_k\|\boldsymbol{r}^{(k)}\|_2$, 其中 $\boldsymbol{r}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}$, 且

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{z}^{(k+\frac{1}{2})},\quad \boldsymbol{M}_2\boldsymbol{z}^{(k+\frac{1}{2})} = \boldsymbol{r}^{(k+\frac{1}{2})} - \boldsymbol{q}^{(k+\frac{1}{2})} \tag{3.81}$$

满足 $\|\boldsymbol{q}^{(k+\frac{1}{2})}\|_2 \leqslant \eta_k\|\boldsymbol{r}^{(k+\frac{1}{2})}\|_2$, 其中 $\boldsymbol{r}^{(k+\frac{1}{2})} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k+\frac{1}{2})}$, 则 $\{\boldsymbol{x}^{(k)}\}$ 形如

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{M}_2^{-1}\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{x}^{(k)} + \boldsymbol{M}_2^{-1}(\boldsymbol{I} + \boldsymbol{N}_2\boldsymbol{M}_1^{-1})\boldsymbol{b} - \boldsymbol{M}_2^{-1}(\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)} + \boldsymbol{q}^{(k+\frac{1}{2})}). \tag{3.82}$$

进一步, 如果 $\boldsymbol{x}^*\in\mathbb{C}^n$ 是方程组 (3.73) 的精确解, 则有

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} \leqslant \big(\sigma + \mu\theta\varepsilon_k + \theta(\rho + \theta\nu\varepsilon_k)\eta_k\big)\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2},\quad k = 0,1,2,\cdots, \tag{3.83}$$

式中:

$$\sigma = \|\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{M}_2^{-1}\|_2,\quad \rho = \|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{M}_2^{-1}\|_2,$$
$$\mu = \|\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\|_2,\quad \theta = \|\boldsymbol{A}\boldsymbol{M}_2^{-1}\|_2,\quad \nu = \|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\|_2.$$

特别地, 如果

$$\sigma + \mu\theta\varepsilon_{\max} + \theta(\rho + \theta\nu\varepsilon_{\max})\eta_{\max} < 1,$$

则迭代序列 $\{\boldsymbol{x}^{(k)}\}$ 收敛到 $\boldsymbol{x}^*\in\mathbb{C}^n$, 其中 $\varepsilon_{\max} = \max\limits_k\{\varepsilon_k\}$ 且 $\eta_{\max} = \max\limits_k\{\eta_k\}$.

证明 由式 (3.80) 并将 $\boldsymbol{r}^{(k)}$ 代入其中, 得

$$\boldsymbol{x}^{(k+\frac{1}{2})} = \boldsymbol{x}^{(k)} + \boldsymbol{M}_1^{-1}(\boldsymbol{r}^{(k)} - \boldsymbol{p}^{(k)}) = \boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{x}^{(k)} + \boldsymbol{M}_1^{-1}\boldsymbol{b} - \boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)}. \tag{3.84}$$

类似地, 由式 (3.81) 并将 $\boldsymbol{r}^{(k+\frac{1}{2})}$ 代入其中, 得

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{M}_2^{-1}(\boldsymbol{r}^{(k+\frac{1}{2})} - \boldsymbol{q}^{(k+\frac{1}{2})}) = \boldsymbol{M}_2^{-1}\boldsymbol{N}_2\boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{M}_2^{-1}\boldsymbol{b} - \boldsymbol{M}_2^{-1}\boldsymbol{q}^{(k+\frac{1}{2})}.$$

因此有

$$\begin{aligned}\boldsymbol{x}^{(k+1)} &= \boldsymbol{M}_2^{-1}\boldsymbol{N}_2(\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{x}^{(k)} + \boldsymbol{M}_1^{-1}\boldsymbol{b} - \boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)}) + \boldsymbol{M}_2^{-1}\boldsymbol{b} - \boldsymbol{M}_2^{-1}\boldsymbol{q}^{(k+\frac{1}{2})} \\ &= \boldsymbol{M}_2^{-1}\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{x}^{(k)} + \boldsymbol{M}_2^{-1}(\boldsymbol{I} + \boldsymbol{N}_2\boldsymbol{M}_1^{-1})\boldsymbol{b} - \boldsymbol{M}_2^{-1}(\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)} + \boldsymbol{q}^{(k+\frac{1}{2})}).\end{aligned} \tag{3.85}$$

由于 $\boldsymbol{x}^* \in \mathbb{C}^n$ 是方程组 (3.73) 的精确解, 必满足

$$\boldsymbol{x}^* = \boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{x}^* + \boldsymbol{M}_1^{-1}\boldsymbol{b}, \tag{3.86}$$

$$\boldsymbol{x}^* = \boldsymbol{M}_2^{-1}\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{x}^* + \boldsymbol{M}_2^{-1}(\boldsymbol{I} + \boldsymbol{N}_2\boldsymbol{M}_1^{-1})\boldsymbol{b}. \tag{3.87}$$

分别由式 (3.84) 减去式 (3.86), 由式 (3.85) 减去式 (3.87), 有

$$\boldsymbol{x}^{(k+\frac{1}{2})} - \boldsymbol{x}^* = \boldsymbol{M}_1^{-1}\boldsymbol{N}_1(\boldsymbol{x}^{(k)} - \boldsymbol{x}^*) - \boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)}, \tag{3.88}$$

$$\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^* = \boldsymbol{M}_2^{-1}\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1(\boldsymbol{x}^{(k)} - \boldsymbol{x}^*) - \boldsymbol{M}_2^{-1}(\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)} + \boldsymbol{q}^{(k+\frac{1}{2})}). \tag{3.89}$$

式 (3.88) 和式 (3.89) 两边取范数, 得

$$\begin{aligned}&\|\boldsymbol{x}^{(k+\frac{1}{2})} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} \\ &\leqslant \|\boldsymbol{M}_1^{-1}\boldsymbol{N}_1(\boldsymbol{x}^{(k)} - \boldsymbol{x}^*)\|_{\boldsymbol{M}_2} + \|\boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)}\|_{\boldsymbol{M}_2} \\ &\leqslant \|\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\|_{\boldsymbol{M}_2} \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} + \|\boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)}\|_{\boldsymbol{M}_2} \\ &\leqslant \|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{M}_2^{-1}\|_2 \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} + \|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\|_2 \cdot \|\boldsymbol{p}^{(k)}\|_2,\end{aligned} \tag{3.90}$$

$$\begin{aligned}&\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} \\ &\leqslant \|\boldsymbol{M}_2^{-1}\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\|_{\boldsymbol{M}_2} \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} + \|\boldsymbol{M}_2^{-1}(\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)} + \boldsymbol{q}^{(k+\frac{1}{2})})\|_{\boldsymbol{M}_2} \\ &= \|\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{M}_2^{-1}\|_2 \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} + \|\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{p}^{(k)} + \boldsymbol{q}^{(k+\frac{1}{2})}\|_2 \\ &\leqslant \|\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{M}_2^{-1}\|_2 \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} + \|\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\|_2 \cdot \|\boldsymbol{p}^{(k)}\|_2 + \|\boldsymbol{q}^{(k+\frac{1}{2})}\|_2.\end{aligned} \tag{3.91}$$

注意到

$$\|\boldsymbol{r}^{(k)}\|_2 = \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}\|_2 = \|\boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x}^{(k)})\|_2 \leqslant \|\boldsymbol{A}\boldsymbol{M}_2^{-1}\|_2 \cdot \|\boldsymbol{x}^* - \boldsymbol{x}^{(k)}\|_{\boldsymbol{M}_2},$$

$$\|\boldsymbol{r}^{(k+\frac{1}{2})}\|_2 = \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k+\frac{1}{2})}\|_2 = \|\boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x}^{(k+\frac{1}{2})})\|_2 \leqslant \|\boldsymbol{A}\boldsymbol{M}_2^{-1}\|_2 \cdot \|\boldsymbol{x}^* - \boldsymbol{x}^{(k+\frac{1}{2})}\|_{\boldsymbol{M}_2},$$

由式 (3.88) 和序列 $\{\boldsymbol{p}^{(k)}\}$ 和 $\{\boldsymbol{q}^{(k+\frac{1}{2})}\}$ 的定义, 有

$$\|\boldsymbol{p}^{(k)}\|_2 \leqslant \varepsilon_k\|\boldsymbol{r}^{(k)}\|_2 \leqslant \varepsilon_k\|\boldsymbol{A}\boldsymbol{M}_2^{-1}\|_2 \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2}, \tag{3.92}$$

$$\begin{aligned}&\|\boldsymbol{q}^{(k+\frac{1}{2})}\|_2 \leqslant \eta_k\|\boldsymbol{r}^{(k+\frac{1}{2})}\|_2 \\ &\leqslant \eta_k\|\boldsymbol{A}\boldsymbol{M}_2^{-1}\|_2\big(\|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{M}_2^{-1}\|_2 \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} + \|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\| \cdot \|\boldsymbol{p}^{(k)}\|_2\big) \\ &\leqslant \eta_k\|\boldsymbol{A}\boldsymbol{M}_2^{-1}\|_2\big(\|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{M}_2^{-1}\|_2 + \varepsilon_k\|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\|_2\|\boldsymbol{A}\boldsymbol{M}_2^{-1}\|_2\big)\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2}.\end{aligned} \tag{3.93}$$

将式 (3.92) 和式 (3.93) 代入式 (3.91), 得

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} \leqslant \big[\|\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{M}_2^{-1}\| + \varepsilon_k\|\boldsymbol{N}_2\boldsymbol{M}_1^{-1}\| \cdot \|\boldsymbol{A}\boldsymbol{M}_2^{-1}\| + \eta_k\|\boldsymbol{A}\boldsymbol{M}_2^{-1}\|$$

$$\times \left(\|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\boldsymbol{N}_1\boldsymbol{M}_2^{-1}\| + \varepsilon_k\|\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\| \cdot \|\boldsymbol{A}\boldsymbol{M}_2^{-1}\|\right)\big] \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2}$$
$$\leqslant \big[\sigma + \mu\theta\varepsilon_k + \theta(\rho + \theta\nu\varepsilon_k)\eta_k\big] \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2}.$$

证毕. □

注 3.3 如果在某些应用中可精确求解内方程组, 相应的量 $\{\varepsilon_k\}$ 和 $\{\eta_k\}$ 及 $\varepsilon_{\max}$ 和 $\eta_{\max}$ 等于零. 从而得到 IHSS 迭代法的收敛速度与 HSS 迭代法的相同.

将定理 3.28 应用到 Hermite 和反 Hermite 分裂

$$\begin{aligned}\boldsymbol{A} &= \boldsymbol{M}_1 - \boldsymbol{N}_1 \equiv (\alpha\boldsymbol{I} + \boldsymbol{H}) - (\alpha\boldsymbol{I} - \boldsymbol{S})\\ &= \boldsymbol{M}_2 - \boldsymbol{N}_2 \equiv (\alpha\boldsymbol{I} + \boldsymbol{S}) - (\alpha\boldsymbol{I} - \boldsymbol{H}),\end{aligned}$$

直接得到下面关于 IHSS 迭代法的收敛性定理.

定理 3.29 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 为正定矩阵, $\boldsymbol{H} = \frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{H}})$ 和 $\boldsymbol{S} = \frac{1}{2}(\boldsymbol{A} - \boldsymbol{A}^{\mathrm{H}})$ 为其 Hermite 和反 Hermite 部分, 且 α 是一个正常数. 若 $\{\boldsymbol{x}^{(k)}\}$ 是由 IHSS 迭代法产生的迭代序列且方程组 (3.73) 的精确解是 $\boldsymbol{x}^* \in \mathbb{C}^n$, 则成立

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|_* \leqslant (\sigma(\alpha) + \theta\rho\eta_k)(1 + \theta\varepsilon_k)\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_*, \quad k = 0, 1, 2, \cdots,$$

式中: 范数 $\|\cdot\|_*$ 的定义如前,

$$\rho = \|(\alpha\boldsymbol{I} + \boldsymbol{S})(\alpha\boldsymbol{I} + \boldsymbol{H})^{-1}\|_2, \quad \theta = \|\boldsymbol{A}(\alpha\boldsymbol{I} + \boldsymbol{S})^{-1}\|_2. \tag{3.94}$$

特别地, 若 $(\sigma(\alpha) + \theta\rho\eta_{\max})(1 + \theta\varepsilon_{\max}) < 1$, 则迭代序列 $\{\boldsymbol{x}^{(k)}\}$ 收敛到 $\boldsymbol{x}^* \in \mathbb{C}^n$, 其中 $\varepsilon_{\max} = \max\limits_k\{\varepsilon_k\}$ 且 $\eta_{\max} = \max\limits_k\{\eta_k\}$.

依据定理 3.28, 可选择阈值来极小化两步分裂迭代法的计算工作量. 注意到不需要当 k 增大时阈值 $\{\varepsilon_k\}$ 和 $\{\eta_k\}$ 趋向于零来得到 IHSS 迭代法的收敛性. 下面的定理给出选择阈值 $\{\varepsilon_k\}$ 和 $\{\eta_k\}$ 的一种方式, 使得两步分裂迭代法的原收敛速度 (比较引理 3.1) 可被渐近地恢复.

定理 3.30 设定理 3.28 的条件成立, $\{\tau_1(k)\}$ 和 $\{\tau_2(k)\}$ 是满足 $\tau_1(k) \geqslant 1$, $\tau_2(k) \geqslant 1$ 的非降正数序列, 且 $\limsup\limits_{k\to\infty}\tau_1(k) = \limsup\limits_{k\to\infty}\tau_2(k) = +\infty$, δ_1 和 δ_2 为属于 $(0,1)$ 区间的实常数, 满足

$$\varepsilon_k \leqslant c_1\delta_1^{\tau_1(k)}, \quad \eta_k \leqslant c_2\delta_2^{\tau_2(k)}, \quad k = 0, 1, 2, \cdots, \tag{3.95}$$

式中: c_1 和 c_2 为非负常数, 则有

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2} \leqslant \left(\sqrt{\sigma} + \omega\theta\delta^{\tau(k)}\right)^2\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_{\boldsymbol{M}_2}, \quad k = 0, 1, 2, \cdots,$$

式中:

$$\tau(k) = \min\{\tau_1(k),\ \tau_2(k)\}, \quad \delta = \max\{\delta_1,\ \delta_2\}, \quad \omega = \max\left\{\sqrt{c_1c_2\nu},\ \frac{1}{2\sqrt{\sigma}}(c_1\mu + c_2\rho)\right\}.$$

特别地, 有

$$\limsup_{k\to\infty}\frac{\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^*\|_{M_2}}{\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_{M_2}}=\sigma,$$

即非精确两步分裂迭代法的收敛速度渐近地与精确两步分裂迭代法的相同.

证明 由式 (3.83) 和式 (3.95), 得

$$\begin{aligned}
\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^*\|_{M_2} &\leqslant \big(\sigma+\mu\theta\varepsilon_k+\theta(\rho+\theta\nu\varepsilon_k)\eta_k\big)\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_{M_2}\\
&\leqslant \big[\sigma+\mu\theta c_1\delta_1^{\tau_1(k)}+\theta(\rho+\theta\nu c_1\delta_1^{\tau_1(k)})c_2\delta_2^{\tau_2(k)}\big]\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_{M_2}\\
&\leqslant \big[\sigma+\mu\theta c_1\delta^{\tau(k)}+\theta(\rho+\theta\nu c_1\delta^{\tau(k)})c_2\delta^{\tau(k)}\big]\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_{M_2}\\
&= \big[\sigma+(c_1\mu+c_2\rho)\theta\delta^{\tau(k)}+c_1c_2\nu\theta^2\delta^{2\tau(k)}\big]\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_{M_2}\\
&\leqslant \big(\sigma+2\omega\sqrt{\sigma}\theta\delta^{\tau(k)}+\omega^2\theta^2\delta^{2\tau(k)}\big)\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_{M_2}\\
&= \big(\sqrt{\sigma}+\omega\theta\delta^{\tau(k)}\big)^2\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_{M_2}.
\end{aligned}$$

由此即得定理的结论. 证毕. □

由定理 3.29 和定理 3.30 立即可导出下面 IHSS 迭代法的收敛性结果.

定理 3.31 设定理 3.29 的条件成立, $\{\tau_1(k)\}$ 和 $\{\tau_2(k)\}$ 是满足 $\tau_1(k)\geqslant 1$, $\tau_2(k)\geqslant 1$ 的非降正数序列, 且 $\limsup\limits_{k\to\infty}\tau_1(k)=\limsup\limits_{k\to\infty}\tau_2(k)=+\infty$, δ_1 和 δ_2 为属于 $(0,1)$ 区间的实常数, 满足式 (3.95). 则成立

$$\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^*\|_*\leqslant(\sqrt{\sigma(\alpha)}+\omega\theta\delta^{\tau(k)})^2\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_*,\quad k=0,1,2,\cdots,$$

式中: ρ 和 θ 由式 (3.94) 定义; $\tau(k)$ 和 δ 由式 (3.30) 定义, 且

$$\omega=\max\left\{\sqrt{c_1c_2\rho},\ \frac{1}{2\sqrt{\sigma(\alpha)}}(c_1\sigma(\alpha)+c_2\rho)\right\}.$$

特别地, 有

$$\limsup_{k\to\infty}\frac{\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^*\|_*}{\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^*\|_*}=\sigma(\alpha),$$

即 IHSS 迭代法的收敛速度渐近地与 HSS 迭代法的相同. 此处的范数 $\|\cdot\|_*$ 定义如前.

定理 3.31 表明如果阈值 $\{\varepsilon_k\}$ 和 $\{\eta_k\}$ 如式 (3.95) 中所选择, 则 IHSS 迭代法收敛到方程组 (3.73) 的唯一解 $\boldsymbol{x}^*\in\mathbb{C}^n$, 且 IHSS 迭代渐近收敛因子的上界趋向于 HSS 迭代的 $\sigma(\alpha)$ (见定理 3.27). 进一步, 式 (3.95) 也可用 $\{\varepsilon_k\}$ 和 $\{\eta_k\}$ 趋向于零来替代.

下面给出 IHSS 迭代法的运算量和存储量估计.

用 ν 表示一次矩阵向量乘积 $\boldsymbol{Ax}$ 所需要的运算量, $\chi_k(\boldsymbol{H})$ 和 $\chi_k(\boldsymbol{S})$ 分别是用阈值 $\{\varepsilon_k\}$ 和 $\{\eta_k\}$ 来非精确求解涉及 $\boldsymbol{H}$ 和 $\boldsymbol{S}$ 的内方程组所需的运算量, 则一次 IHSS 迭代的计算工作量可由表 ?? 来估计. 直接计算表明计算 IHSS 迭代每一步的总工作量是 $O(4n-2\nu+\chi_k(\boldsymbol{H})+\chi_k(\boldsymbol{S}))$.

另外, 简单的计算表明所需要的存储量是储存 $\boldsymbol{x}^{(k)}$, $\boldsymbol{b}$, $\boldsymbol{r}^{(k)}$, $\boldsymbol{z}^{(k)}$. 对于内方程组的非精确解法器, 只需要另外一些辅助向量, 例如, CG 方法需要大约五个向量 (见第 4 章). 另外, 不需要显式地存储 $\boldsymbol{H}$ 和 $\boldsymbol{S}$, 所需要的仅是执行关于这两个矩阵的矩阵向量乘积子程序. 因此, 所需的存储总量是 $O(n)$, 即与未知量个数的量同阶.

表 3.2 一次 IHSS 迭代所需的工作量

运算	工作量	运算	工作量
$\boldsymbol{r}^{(k)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k)}$	$n+\nu$	$(\alpha\boldsymbol{I}+\boldsymbol{H})\boldsymbol{z}^{(k+\frac{1}{2})}=\boldsymbol{r}^{(k)}$	$\chi_k(\boldsymbol{H})$
$\boldsymbol{x}^{(k+\frac{1}{2})}=\boldsymbol{x}^{(k)}+\boldsymbol{z}^{(k+\frac{1}{2})}$	n	$\boldsymbol{r}^{(k+\frac{1}{2})}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k+\frac{1}{2})}$	$n+\nu$
$(\alpha\boldsymbol{I}+\boldsymbol{S})\boldsymbol{z}^{(k+\frac{1}{2})}=\boldsymbol{r}^{(k+\frac{1}{2})}$	$\chi_k(\boldsymbol{S})$	$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k+\frac{1}{2})}+\boldsymbol{z}^{(k+1)}$	n

3.4.2 PHSS 迭代法

预处理技术可以大大加快迭代法的收敛速度. 利用 HSS 迭代技术, 文献 [22] 对 2×2 块结构的正半定线性方程组, 建立了一类预处理 HSS 迭代法, 简称 PHSS 迭代法. 理论分析表明方法无条件地收敛到线性方程组的唯一解. 此外, 推导出其收缩因子的上界并给出所涉及迭代参数的最佳选择.

考虑具有块系数矩阵的线性方程组

$$\mathcal{A}\boldsymbol{x}:=\begin{bmatrix}\boldsymbol{A} & \boldsymbol{B}\\ -\boldsymbol{B}^{\mathrm{H}} & \boldsymbol{O}\end{bmatrix}\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{z}\end{bmatrix}=\begin{bmatrix}\boldsymbol{f}\\ \boldsymbol{g}\end{bmatrix}:=\boldsymbol{b}, \tag{3.96}$$

式中: 子矩阵 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$ 是 Hermite 正定的; $\boldsymbol{B}\in\mathbb{C}^{m\times n}\,(m\geqslant n)$ 是列满秩的. 因此, 矩阵 $\mathcal{A}\in\mathbb{C}^{(m+n)\times(m+n)}$ 是非奇异、非 Hermite 半正定矩阵.

块线性方程组 (3.96) 相应于线性约束二次规划问题或鞍点问题的 Kuhn–Tucker 条件. 这些方程组典型地来自于二阶椭圆方程、弹性问题或 Stokes 方程的混合有限元近似. 下面首先给出预处理块线性方程组 (3.96) 一个等价的形式, 然后应用 HSS 方法于预处理块线性方程组, 从而对非 Hermite 半正定方程组 (3.96) 建立一类 PHSS 迭代法.

引入预处理矩阵

$$\mathcal{P}=\begin{bmatrix}\boldsymbol{A} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{S}\end{bmatrix}\in\mathbb{C}^{(m+n)\times(m+n)}\ \text{和}\ \widehat{\boldsymbol{B}}=\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{S}^{-\frac{1}{2}}\in\mathbb{C}^{m\times n}, \tag{3.97}$$

式中: $\boldsymbol{S}\in\mathbb{C}^{n\times n}$ 为可自由选择的 Hermite 正定子矩阵, 其最佳选择在后面讨论. 定义

$$\widehat{\mathcal{A}}=\mathcal{P}^{-\frac{1}{2}}\mathcal{A}\mathcal{P}^{-\frac{1}{2}}=\begin{bmatrix}\boldsymbol{I} & \widehat{\boldsymbol{B}}\\ -\widehat{\boldsymbol{B}}^{\mathrm{H}} & \boldsymbol{O}\end{bmatrix},\ \begin{bmatrix}\widehat{\boldsymbol{y}}\\ \widehat{\boldsymbol{z}}\end{bmatrix}=\mathcal{P}^{\frac{1}{2}}\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{z}\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{y}\\ \boldsymbol{S}^{\frac{1}{2}}\boldsymbol{z}\end{bmatrix},$$

$$\widehat{\boldsymbol{b}}=\begin{bmatrix}\widehat{\boldsymbol{f}}\\ \widehat{\boldsymbol{g}}\end{bmatrix}=\mathcal{P}^{-\frac{1}{2}}\boldsymbol{b}=\begin{bmatrix}\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{f}\\ \boldsymbol{S}^{-\frac{1}{2}}\boldsymbol{g}\end{bmatrix}.$$

则线性方程组 (3.96) 可转换成下面等价的预处理形式:

$$\widehat{\mathcal{A}}\begin{bmatrix}\widehat{\boldsymbol{y}}\\ \widehat{\boldsymbol{z}}\end{bmatrix}=\widehat{\boldsymbol{b}}. \tag{3.98}$$

显然, 矩阵 $\widehat{\mathcal{A}}\in\mathbb{C}^{(m+n)\times(m+n)}$ 的 Hermite 和反 Hermite 部分分别为

$$\widehat{\boldsymbol{H}}=\frac{1}{2}(\widehat{\mathcal{A}}+\widehat{\mathcal{A}}^{\mathrm{H}})=\begin{bmatrix}\boldsymbol{I} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O}\end{bmatrix},\ \widehat{\boldsymbol{S}}=\frac{1}{2}(\widehat{\mathcal{A}}-\widehat{\mathcal{A}}^{\mathrm{H}})=\begin{bmatrix}\boldsymbol{O} & \widehat{\boldsymbol{B}}\\ -\widehat{\boldsymbol{B}}^{\mathrm{H}} & \boldsymbol{O}\end{bmatrix}.$$

对式 (3.98) 直接应用 HSS 迭代格式, 得

$$\begin{cases} (\alpha \boldsymbol{I} + \widehat{\boldsymbol{H}}) \begin{bmatrix} \widehat{\boldsymbol{y}}^{(k+\frac{1}{2})} \\ \widehat{\boldsymbol{z}}^{(k+\frac{1}{2})} \end{bmatrix} = (\alpha \boldsymbol{I} - \widehat{\boldsymbol{S}}) \begin{bmatrix} \widehat{\boldsymbol{y}}^{(k)} \\ \widehat{\boldsymbol{z}}^{(k)} \end{bmatrix} + \widehat{\boldsymbol{b}}, \\ (\alpha \boldsymbol{I} + \widehat{\boldsymbol{S}}) \begin{bmatrix} \widehat{\boldsymbol{y}}^{(k+1)} \\ \widehat{\boldsymbol{z}}^{(k+1)} \end{bmatrix} = (\alpha \boldsymbol{I} - \widehat{\boldsymbol{H}}) \begin{bmatrix} \widehat{\boldsymbol{y}}^{(k+\frac{1}{2})} \\ \widehat{\boldsymbol{z}}^{(k+\frac{1}{2})} \end{bmatrix} + \widehat{\boldsymbol{b}}. \end{cases}$$

或等价地, 有

$$\begin{bmatrix} \alpha \boldsymbol{I} & \widehat{\boldsymbol{B}} \\ -\widehat{\boldsymbol{B}}^{\mathrm{H}} & \alpha \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{y}}^{(k+1)} \\ \widehat{\boldsymbol{z}}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \frac{\alpha(\alpha-1)}{\alpha+1} \boldsymbol{I} & -\frac{\alpha-1}{\alpha+1} \widehat{\boldsymbol{B}} \\ \widehat{\boldsymbol{B}}^{\mathrm{H}} & \alpha \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{y}}^{(k)} \\ \widehat{\boldsymbol{z}}^{(k)} \end{bmatrix} + \begin{bmatrix} \frac{2\alpha}{\alpha+1} \widehat{\boldsymbol{f}} \\ 2\widehat{\boldsymbol{g}} \end{bmatrix}.$$

从而按原变量, 得

$$\begin{bmatrix} \alpha \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^{\mathrm{H}} & \alpha \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}^{(k+1)} \\ \boldsymbol{z}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \frac{\alpha(\alpha-1)}{\alpha+1} \boldsymbol{A} & -\frac{\alpha-1}{\alpha+1} \boldsymbol{B} \\ \boldsymbol{B}^{\mathrm{H}} & \alpha \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}^{(k)} \\ \boldsymbol{z}^{(k)} \end{bmatrix} + \begin{bmatrix} \frac{2\alpha}{\alpha+1} \boldsymbol{f} \\ 2\boldsymbol{g} \end{bmatrix}, \tag{3.99}$$

这导致下面求解块线性方程组 (3.96) 的 PHSS 迭代法.

算法 3.8 (PHSS 迭代法)

步 1, 给定一个初始向量 $\boldsymbol{x}^{(0)} = \begin{bmatrix} \boldsymbol{y}^{(0)} \\ \boldsymbol{z}^{(0)} \end{bmatrix} \in \mathbb{C}^{m+n}$, 参数 $\alpha > 0$, 容许误差 $\varepsilon > 0$. 置 $k := 0$.

步 2, 用 HSS 迭代法求解式 (3.99) 计算 $\boldsymbol{x}^{(k+1)} = \begin{bmatrix} \boldsymbol{y}^{(k+1)} \\ \boldsymbol{z}^{(k+1)} \end{bmatrix} \in \mathbb{C}^{m+n}$.

步 3, 计算残差 $\boldsymbol{r}^{(k+1)} = \boldsymbol{b} - \mathcal{A}\boldsymbol{x}^{(k+1)}$. 若 $\|\boldsymbol{r}^{(k+1)}\|_2 \leqslant \varepsilon$, 停算. 否则, 置 $k := k+1$, 转步 2.

明显地, PHSS 迭代法可等价地重写为

$$\begin{bmatrix} \boldsymbol{y}^{(k+1)} \\ \boldsymbol{z}^{(k+1)} \end{bmatrix} = \mathcal{T}(\alpha) \begin{bmatrix} \boldsymbol{y}^{(k)} \\ \boldsymbol{z}^{(k)} \end{bmatrix} + \mathcal{M}(\alpha)^{-1} \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix}, \tag{3.100}$$

式中:

$$\begin{cases} \mathcal{T}(\alpha) = \begin{bmatrix} \alpha \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^{\mathrm{H}} & \alpha \boldsymbol{S} \end{bmatrix}^{-1} \begin{bmatrix} \dfrac{\alpha(\alpha-1)}{\alpha+1} \boldsymbol{A} & -\dfrac{\alpha-1}{\alpha+1} \boldsymbol{B} \\ \boldsymbol{B}^{\mathrm{H}} & \alpha \boldsymbol{S} \end{bmatrix}, \\ \mathcal{M}(\alpha)^{-1} = \begin{bmatrix} \alpha \boldsymbol{A} & \boldsymbol{B} \\ -\boldsymbol{B}^{\mathrm{H}} & \alpha \boldsymbol{S} \end{bmatrix}^{-1} \begin{bmatrix} \dfrac{2\alpha}{\alpha+1} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & 2\boldsymbol{I} \end{bmatrix}. \end{cases} \tag{3.101}$$

这里 $\mathcal{T}(\alpha)$ 是 PHSS 迭代法的迭代矩阵. 事实上, 式 (3.100) 也可来自于系数矩阵 $\mathcal{A}$ 的分裂

$$\mathcal{A} = \mathcal{M}(\alpha) - \mathcal{N}(\alpha) = \begin{bmatrix} \dfrac{\alpha+1}{2} \boldsymbol{A} & \dfrac{\alpha+1}{2\alpha} \boldsymbol{B} \\ -\dfrac{1}{2} \boldsymbol{B}^{\mathrm{H}} & \dfrac{\alpha}{2} \boldsymbol{S} \end{bmatrix} - \begin{bmatrix} \dfrac{\alpha-1}{2} \boldsymbol{A} & -\dfrac{\alpha-1}{2\alpha} \boldsymbol{B} \\ \dfrac{1}{2} \boldsymbol{B}^{\mathrm{H}} & \dfrac{\alpha}{2} \boldsymbol{S} \end{bmatrix}. \tag{3.102}$$

在实际计算中, PHSS 迭代法的每次迭代需要求解具有如下系数矩阵的线性方程组

$$\widetilde{\mathcal{M}}(\alpha)=\begin{bmatrix}\alpha\boldsymbol{A} & \boldsymbol{B}\\ -\boldsymbol{B}^{\mathrm{H}} & \alpha\boldsymbol{S}\end{bmatrix}, \tag{3.103}$$

或等价地, 有

$$\mathcal{M}(\alpha)=\begin{bmatrix}\dfrac{\alpha+1}{2}\boldsymbol{A} & \dfrac{\alpha+1}{2\alpha}\boldsymbol{B}\\ -\dfrac{1}{2}\boldsymbol{B}^{\mathrm{H}} & \dfrac{\alpha}{2}\boldsymbol{S}\end{bmatrix}. \tag{3.104}$$

由于这些矩阵是正定的, 可用另一个迭代过程非精确地求解前述线性方程组, 如 HSS 迭代. 这导致非 Hermite 正半定线性方程组 (3.96) 的一个非精确预处理 Hermite–反 Hermite 分裂迭代法 (简记为 IPHSS), 并已经在前面有所讨论.

为了对 PHSS 迭代法进行收敛性分析, 通过简单推导、奇异值分解等可得式 (3.101) 中矩阵 $\mathcal{T}(\alpha)$ 的显式表达式、特征值结构以及谱半径, 这由下面的三个引理描述, 其证明详见文献 [22].

引理 3.2 考虑线性方程组 (3.96). 设 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$ 是 Hermite 正定的, $\boldsymbol{B}\in\mathbb{C}^{m\times n}$ 列满秩, $\alpha>0$ 是一个给定常数. 假设 $\boldsymbol{S}\in\mathbb{C}^{n\times n}$ 是 Hermite 正定矩阵. 则划分式 (3.101) 中的迭代矩阵 $\mathcal{T}(\alpha)$ 为

$$\mathcal{T}(\alpha)=\begin{bmatrix}\mathcal{T}_{11}(\alpha) & \mathcal{T}_{12}(\alpha)\\ \mathcal{T}_{21}(\alpha) & \mathcal{T}_{22}(\alpha)\end{bmatrix},$$

式中:

$$\mathcal{T}_{11}(\alpha)=\frac{\alpha-1}{\alpha+1}\boldsymbol{I}-\frac{2}{\alpha+1}\boldsymbol{A}^{-1}\boldsymbol{B}\widetilde{\boldsymbol{S}}(\alpha)^{-1}\boldsymbol{B}^{\mathrm{H}},\quad \mathcal{T}_{12}(\alpha)=-\frac{2\alpha}{\alpha+1}\boldsymbol{A}^{-1}\boldsymbol{B}\widetilde{\boldsymbol{S}}(\alpha)^{-1}\boldsymbol{S},$$
$$\mathcal{T}_{21}(\alpha)=\frac{2\alpha}{\alpha+1}\boldsymbol{S}\widetilde{\boldsymbol{S}}(\alpha)^{-1}\boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}^{-1},\qquad \mathcal{T}_{22}(\alpha)=-\frac{\alpha-1}{\alpha+1}\boldsymbol{I}+\frac{2\alpha^2}{\alpha+1}\widetilde{\boldsymbol{S}}(\alpha)^{-1}\boldsymbol{S},$$

且

$$\widetilde{\boldsymbol{S}}(\alpha)=\alpha\boldsymbol{S}+\frac{1}{\alpha}\boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}^{-1}\boldsymbol{B}$$

为式 (3.103) 中矩阵 $\widetilde{\mathcal{M}}(\alpha)$ 的 Schur 补.

引理 3.3 设引理 3.2 中的条件满足. 如果 $\sigma_k\,(k=1,2,\cdots,n)$ 是式 (3.97) 中矩阵 $\widehat{\boldsymbol{B}}\in\mathbb{C}^{m\times n}$ 的正奇异值, 则 PHSS 方法迭代矩阵 $\mathcal{T}(\alpha)$ 的特征值是具有重数 $m-n$ 的

$$\frac{\alpha-1}{\alpha+1}$$

和

$$\frac{1}{(\alpha+1)(\alpha^2+\sigma_k^2)}\Big(\alpha(\alpha^2-\sigma_k^2)\pm\sqrt{(\alpha^2+\sigma_k^2)^2-4\alpha^4\sigma_k^2}\Big),\quad k=1,2,\cdots,n.$$

引理 3.4 设引理 3.2 中的条件满足. 如果 $\sigma_k\,(k=1,2,\cdots,n)$ 是式 (3.97) 中矩阵 $\widehat{\boldsymbol{B}}\in\mathbb{C}^{m\times n}$ 的正奇异值, 且 λ 是 PHSS 方法迭代矩阵 $\mathcal{T}(\alpha)$ 的主特征值, 即 $\mathcal{T}(\alpha)$ 的谱半径可由 $|\lambda|$ 达到, 则对 $k=1,2,\cdots,n$, 成立

$$|\lambda|=\begin{cases}\dfrac{|\alpha-1|}{\alpha+1};\\ \dfrac{\alpha}{\alpha+1}\left(\dfrac{|\alpha^2-\sigma_k^2|}{\alpha^2+\sigma_k^2}+\sqrt{\dfrac{1}{\alpha^2}-\dfrac{4\alpha^2\sigma_k^2}{(\alpha^2+\sigma_k^2)^2}}\right), & \text{对}\quad \alpha^2+\sigma_k^2>2\alpha^2\sigma_k;\\ \sqrt{\dfrac{\alpha-1}{\alpha+1}}, & \text{对}\quad \alpha^2+\sigma_k^2\leqslant 2\alpha^2\sigma_k.\end{cases}$$

基于引理 3.4, 下面证明求解线性方程组 (3.96) 的 PHSS 迭代法的收敛性.

定理 3.32 考虑线性方程组 (3.96). 设 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$ 是 Hermite 正定的, $\boldsymbol{B}\in\mathbb{C}^{m\times n}$ 列满秩, $\alpha>0$ 是一个给定常数. 假设 $\boldsymbol{S}\in\mathbb{C}^{n\times n}$ 是 Hermite 正定矩阵, 则

$$\rho(\mathcal{T}(\alpha))<1,\quad\forall\,\alpha>0,$$

即 PHSS 迭代序列收敛到线性方程组 (3.96) 的精确解.

证明 显然有

$$\frac{|\alpha-1|}{\alpha+1}<1\ \ (\forall\,\alpha>0)\ \ 和\ \ \sqrt{\frac{\alpha-1}{\alpha+1}}<1\ \ (\forall\,\alpha>1).$$

因为对 $k=1,2,\cdots,n$, 当 $\alpha^2+\sigma_k^2>2\alpha^2\sigma_k$ 时, 成立

$$\begin{aligned}&\frac{\alpha}{\alpha+1}\left(\frac{|\alpha^2-\sigma_k^2|}{\alpha^2+\sigma_k^2}+\sqrt{\frac{1}{\alpha^2}-\frac{4\alpha^2\sigma_k^2}{(\alpha^2+\sigma_k^2)^2}}\right)\\ &<\frac{\alpha}{\alpha+1}\left(\frac{|\alpha^2-\sigma_k^2|}{\alpha^2+\sigma_k^2}+\frac{1}{\alpha}\right)<\frac{\alpha}{\alpha+1}\left(1+\frac{1}{\alpha}\right)=1,\end{aligned}$$

利用引理 3.4 易见对 $\forall\,\alpha>0$ 均成立 $\rho(\mathcal{T}(\alpha))<1$. 证毕. □

下面的定理描述了 PHSS 迭代法的最佳迭代参数和相应的渐近收敛因子.

定理 3.33 考虑线性方程组 (3.96). 设 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$ 是 Hermite 正定的, $\boldsymbol{B}\in\mathbb{C}^{m\times n}$ 列满秩, $\alpha>0$ 是一个给定常数. 假设 $\boldsymbol{S}\in\mathbb{C}^{n\times n}$ 是 Hermite 正定矩阵. 如果 $\sigma_k\,(k=1,2,\cdots,n)$ 是矩阵 $\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{S}^{-\frac{1}{2}}\in\mathbb{C}^{m\times n}$ 的正奇异值, 且 $\sigma_{\min}=\min\limits_{1\leqslant k\leqslant n}\sigma_k$ 和 $\sigma_{\max}=\max\limits_{1\leqslant k\leqslant n}\sigma_k$, 则对线性方程组 (3.96) 的 PHSS 迭代法, 迭代参数 α 的最佳值由下式给出,

$$\alpha^*=\arg\min_{\alpha}\rho(\mathcal{T}(\alpha))=\sqrt{\sigma_{\min}\sigma_{\max}},$$

且相应地, 有

$$\rho(\mathcal{T}(\alpha^*))=\frac{\sigma_{\max}-\sigma_{\min}}{\sigma_{\max}+\sigma_{\min}}.$$

证明 首先注意到下面两个事实成立:

(1) 当 $\alpha \leqslant 1$ 时, $\alpha^2+\sigma_k^2 > 2\alpha^2\sigma_k,\ k=1,2,\cdots,n.$

(2) 当 $\alpha > 1$ 时, 记 $\alpha_- = \alpha^2-\alpha\sqrt{\alpha^2-1}$ 和 $\alpha_+ = \alpha^2+\alpha\sqrt{\alpha^2-1}$, 则

① $\alpha^2+\sigma_k^2 > 2\alpha^2\sigma_k$ 当且仅当 $\sigma_k \in (0,\alpha_-)\cup(\alpha_+,+\infty),\ k\in\{1,2,\cdots,n\}$;

② $\alpha^2+\sigma_k^2 \leqslant 2\alpha^2\sigma_k$ 当且仅当 $\sigma_k \in [\alpha_-,\alpha_+],\ k\in\{1,2,\cdots,n\}$;

③ $\dfrac{\alpha-1}{\alpha+1} < \sqrt{\dfrac{\alpha-1}{\alpha+1}}.$

设

$$\theta(\alpha,\sigma) = \frac{\alpha}{\alpha+1}\left(\frac{|\alpha^2-\sigma^2|}{\alpha^2+\sigma^2}+\sqrt{\frac{1}{\alpha^2}-\frac{4\alpha^2\sigma^2}{(\alpha^2+\sigma^2)^2}}\right).$$

则基于事实 (1) 和 (2), 根据引理 3.4, 易知

$$\rho(\mathcal{T}(\alpha)) = \begin{cases} \max\left\{\dfrac{1-\alpha}{1+\alpha},\ \max\limits_{1\leqslant k\leqslant n}\theta(\alpha,\sigma_k)\right\}, & \alpha\leqslant 1, \\ \max\left\{\sqrt{\dfrac{\alpha-1}{\alpha+1}},\ \max\limits_{\substack{\sigma_k<\alpha_- \text{ 或 } \sigma_k>\alpha_+ \\ k\in\{1,2,\cdots,n\}}}\theta(\alpha,\sigma_k)\right\}, & \alpha>1. \end{cases} \tag{3.105}$$

对任意固定的 $\beta>0$, 定义两个函数 $\theta_1,\theta_2:\ (0,+\infty)\to\ (0,+\infty)$, 即

$$\theta_1(t) = \frac{\beta-t}{\beta+t},\quad \theta_2(t) = \frac{1}{\beta}-\frac{4\beta t}{(\beta+t)^2}.$$

直接计算, 得

$$\frac{\mathrm{d}\theta_1(t)}{\mathrm{d}t} = \frac{-2\beta}{(\beta+t)^2},\quad \frac{\mathrm{d}\theta_2(t)}{\mathrm{d}t} = \frac{4\beta(t-\beta)}{(\beta+t)^3}.$$

则

(1) 对 $\alpha\leqslant 1$ 且 $\sigma_{\min}\leqslant\alpha\leqslant\sigma_{\max}$, 有

$$\max_{1\leqslant k\leqslant n}\theta(\alpha,\sigma_k) = \max\{\theta(\alpha,\sigma_{\min}),\ \theta(\alpha,\sigma_{\max})\}.$$

(2) 对 $\alpha>1$ 且 $\sigma_{\min}<\alpha_-$ 或 $\sigma_{\max}>\alpha_+$, 有

$$\max_{\substack{\sigma_k<\alpha_- \text{ 或 } \sigma_k>\alpha_+ \\ k\in\{1,2,\cdots,n\}}}\theta(\alpha,\sigma_k) = \max\{\theta(\alpha,\sigma_{\min}),\ \theta(\alpha,\sigma_{\max})\}.$$

因此, 当 $\alpha\leqslant 1$ 时, 最佳参数 α^* 必须满足 $\sigma_{\min}\leqslant\alpha^*\leqslant\sigma_{\max}$ 和下面三个条件之一:

$$\frac{1-\alpha^*}{1+\alpha^*} = \theta(\alpha^*,\sigma_{\min}) \geqslant \theta(\alpha^*,\sigma_{\max});$$

$$\frac{1-\alpha^*}{1+\alpha^*} = \theta(\alpha^*,\sigma_{\max}) \geqslant \theta(\alpha^*,\sigma_{\min});$$

$$\theta(\alpha^*,\sigma_{\min})=\theta(\alpha^*,\sigma_{\max})\geqslant\frac{1-\alpha^*}{1+\alpha^*}.$$

且当 $\alpha>1$ 时, 最佳参数 α^* 必须满足 $\sigma_{\min}<\alpha_-^*$ 或 $\sigma_{\max}>\alpha_+^*$ 及下面三个条件之一:

$$\sqrt{\frac{\alpha^*-1}{\alpha^*+1}}=\theta(\alpha^*,\sigma_{\min})\geqslant\theta(\alpha^*,\sigma_{\max});$$

$$\sqrt{\frac{\alpha^*-1}{\alpha^*+1}}=\theta(\alpha^*,\sigma_{\max})\geqslant\theta(\alpha^*,\sigma_{\min});$$

$$\theta(\alpha^*,\sigma_{\min})=\theta(\alpha^*,\sigma_{\max})\geqslant\sqrt{\frac{\alpha^*-1}{\alpha^*+1}}.$$

其中 $\alpha_-^*=(\alpha^*)^2-\alpha^*\sqrt{(\alpha^*)^2-1}$ 和 $\alpha_+^*=(\alpha^*)^2+\alpha^*\sqrt{(\alpha^*)^2-1}$. 直接求解当 $a\leqslant 1$ 的三个条件和 $a>1$ 的三个条件, 可得 α^* 的具体表达式. 然后将 α^* 代入式 (3.105), 可得 $\rho(\mathcal{T}(\alpha^*))$. 证毕. □

由定理 3.33 可见应该选择 Hermite 正定矩阵 $\boldsymbol{S}\in\mathbb{C}^{n\times n}$, 使得具有系数矩阵 $\boldsymbol{S}$ 的线性方程组是易于求解的且矩阵 $\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{S}^{-\frac{1}{2}}\in\mathbb{C}^{m\times n}$ 的奇异值是紧密聚集的, 换言之, $\boldsymbol{S}$ 应该是 Schur 补 $\boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}^{-1}\boldsymbol{B}\in\mathbb{C}^{n\times n}$ 的一个好的预处理子. 寻求好的矩阵 $\boldsymbol{S}$ 对加快 PHSS 迭代法的收敛速度是至关重要的. 至于最佳参数 $\alpha^*=\sqrt{\sigma_{\min}\sigma_{\max}}$ 在实际应用中的计算, 由于矩阵 $\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{S}^{-\frac{1}{2}}$ 的最小和最大奇异值 $\sigma_{\min}$ 和 $\sigma_{\max}$ 分别等于矩阵 $\boldsymbol{S}^{-1}\boldsymbol{B}^{\mathrm{H}}\boldsymbol{A}^{-1}\boldsymbol{B}$ 的最小和最大特征值 $\lambda_{\min}$ 和 $\lambda_{\max}$ 的平方根, 即 $\sigma_{\min}=\sqrt{\lambda_{\min}}$ 和 $\sigma_{\max}=\sqrt{\lambda_{\max}}$, 从而通过计算 $\lambda_{\min}$ 和 $\lambda_{\max}$, 可得 $\alpha^*=\sqrt[4]{\lambda_{\min}\lambda_{\max}}$.

3.5 迭代法的加速方法

使用一阶线性定常迭代法 (3.4), 或者其他的迭代格式求解方程组 (3.1) 时, 迭代过程可能收敛, 也可能不收敛. 即便收敛, 也可能收敛得很慢. 无论是哪种情况, 都希望找到一种改进方法, 使得不收敛的格式变得收敛, 收敛慢的格式变得收敛快.

给定第 k 次迭代后的估计解 $\boldsymbol{x}^{(k)}$, 由迭代格式 (3.4) 得到了一个新的估计解 $\boldsymbol{B}\boldsymbol{x}^{(k)}+\boldsymbol{f}$, 再将它与 $\boldsymbol{x}^{(k)},\boldsymbol{x}^{(k-1)}$ 等进行适当的组合, 就可能得到更快的迭代收敛速度. 这就是迭代法加速的基本思想.

3.5.1 外推方法

在迭代格式 (3.4) 中引入一个参数 $\gamma\neq 0$, 构造新的迭代格式

$$\boldsymbol{x}^{(k+1)}=(1-\gamma)\boldsymbol{x}^{(k)}+\gamma(\boldsymbol{B}\boldsymbol{x}^{(k)}+\boldsymbol{f}):=\boldsymbol{B}_\gamma\boldsymbol{x}^{(k)}+\gamma\boldsymbol{f},\quad k=0,1,\cdots,\tag{3.106}$$

式中:

$$\boldsymbol{B}_\gamma=(1-\gamma)\boldsymbol{I}+\gamma\boldsymbol{B}.\tag{3.107}$$

当 $\gamma=1$ 时, 就退化为原迭代格式 (3.4). 若迭代格式 (3.106) 收敛到某个 $\boldsymbol{x}^*$, 则有

$$\boldsymbol{x}^*=(1-\gamma)\boldsymbol{x}^*+\gamma(\boldsymbol{B}\boldsymbol{x}^*+\boldsymbol{f})\Longrightarrow\boldsymbol{x}^*=\boldsymbol{B}\boldsymbol{x}^*+\boldsymbol{f}.$$

所以, 不管 $\gamma \neq 0$ 如何选取, 当迭代格式 (3.106) 收敛时, 它必定收敛到原来方程组 $\boldsymbol{x} = \boldsymbol{B}\boldsymbol{x} + \boldsymbol{f}$ 的解. 选取适当的 $\boldsymbol{B}$, 根据式 (3.3) 中的定义可知, 这样的解也是 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 的解. 因此, 可望选取一个比较好的参数 γ, 使得迭代格式 (3.106) 收敛得尽可能快. 这就是外推法, 它依赖于原迭代矩阵 $\boldsymbol{B}$ 的选择. 所以分别有 Jacobi 迭代、Gauss–Seidel 迭代和 SOR 迭代的外推格式.

注 3.4 设 $\boldsymbol{B}$ 是 Gauss–Seidel 迭代矩阵, 即使 γ 取为 SOR 迭代法的松弛因子 ω, 式 (3.106) 与 SOR 迭代法也是截然不同的. 在式 (3.106) 中, 当 $\boldsymbol{B}\boldsymbol{x}^{(k)} + \boldsymbol{f}$ 的每个分量计算完之后, 才能再与 $\boldsymbol{x}^{(k)}$ 作加权平均得到 $\boldsymbol{x}^{(k+1)}$.

为了使得外推格式 (3.106) 收敛得尽可能快, 需要选择参数 γ 使得 $\rho(\boldsymbol{B}_\gamma)$ 尽可能小. 如果 $\boldsymbol{B}$ 的所有特征值均为实数, 且分布在 $[a, b]$ 之间, 则有

$$\begin{aligned}\rho(\boldsymbol{B}_\gamma) &= \max_{\lambda \in \lambda(\boldsymbol{B})} |(1-\gamma) + \gamma\lambda| \leqslant \max_{\lambda \in [a,b]} |(1-\gamma) + \gamma\lambda| \\ &= \max\{|(1-\gamma) + \gamma a|,\ |(1-\gamma) + \gamma b|\} \\ &= \max\{|1 + (a-1)\gamma|,\ |1 + (b-1)\gamma|\}.\end{aligned}$$

上式第 2 个等式成立是因为 $(1-\gamma) + \gamma\lambda$ 是 λ 的线性函数, 其在 $[a, b]$ 上的最大绝对值必在端点取得. 为使 $\rho(\boldsymbol{B}_\gamma)$ 尽可能小, 只要右端项尽可能小即可. 下面的定理给出了最佳 γ 的取法.

定理 3.34 设 $\boldsymbol{B}$ 的所有特征值均为实数, 且分布在 $[a, b]$ 之间, 又假定 $1 \notin [a, b]$, 则函数

$$f(\gamma) = \max\{|1 + (a-1)\gamma|,\ |1 + (b-1)\gamma|\}$$

在极小点

$$\gamma_{\min} = \frac{2}{2 - a - b} \tag{3.108}$$

处取得极小值

$$f_{\min} = \begin{cases} 1 - (a-1)|\gamma_{\min}|, & \text{当 } 1 < a \leqslant b, \\ 1 - (1-b)|\gamma_{\min}|, & \text{当 } a \leqslant b < 1. \end{cases}$$

根据定理 3.34, 得

$$\rho(\boldsymbol{B}_\gamma) = f_{\min} = 1 - d|\gamma_{\min}| < 1, \tag{3.109}$$

式中: d 为 1 到 $[a, b]$ 的距离, 即

$$d = \begin{cases} a - 1, & 1 < a \leqslant b, \\ 1 - b, & a \leqslant b < 1. \end{cases}$$

所以, 取 $\gamma = \gamma_{\min}$ 时, 式 (3.106) 收敛. 由于 $\gamma_{\min} \neq 1$, 故取 $\gamma = \gamma_{\min}$ 时的式 (3.106) 要比式 (3.4) (即取 $\gamma = 1$ 时的迭代法) 收敛得更快. 此外, 为了使得 $f_{\min}$ 尽可能小, 可取 a, b 分别为迭代矩阵 $\boldsymbol{B}$ 的最小和最大特征值.

3.5.2 整体校正方法

设 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\,(m>1)$ 是方程组 (3.1) 按照某种方式获取的互异近似解, 并设 $\boldsymbol{A}\boldsymbol{x}_i \neq \boldsymbol{b}\,(i=1,2,\cdots,m)$. 利用诸 $\boldsymbol{x}_i$ 提供的信息, 构造向量 $\boldsymbol{x}$, 使得它比 $\boldsymbol{x}_i$ 都更接近于方程组 (3.1) 的精确解 $\boldsymbol{x}^*$, 即要求

$$\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2 < \min_{1\leqslant i\leqslant m} \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_i\|_2, \tag{3.110}$$

称满足式 (3.110) 的向量 $\boldsymbol{x}$ 为方程组 (3.1) 关于 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$ 的校正解, 确定 $\boldsymbol{x}$ 的过程称为校正过程. 建立整体校正模型

$$\begin{cases} \boldsymbol{x} = \sum\limits_{i=1}^{m} \alpha_i \boldsymbol{x}_i, \quad \sum\limits_{i=1}^{m} \alpha_i = 1, \\ \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2 = \min. \end{cases} \tag{3.111}$$

所谓整体校正, 是指 $\boldsymbol{x}_i$ 的每一个分量对 $\boldsymbol{x}$ 的对应分量的贡献比例相同. 下面求解模型 (3.111).

任意选定 $s \in \{1,2,\cdots,m\}$, 则有

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{x}_s + \sum_{i\neq s} \alpha_i(\boldsymbol{x}_i - \boldsymbol{x}_s), \\ \boldsymbol{r}(\boldsymbol{x}) &= \boldsymbol{b}-\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_s) - \sum_{i\neq s} \alpha_i(\boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{x}_s) \\ &= \boldsymbol{r}_s + \sum_{i\neq s} \alpha_i(\boldsymbol{r}_i - \boldsymbol{r}_s). \end{aligned} \tag{3.112}$$

记

$$\begin{aligned} &\boldsymbol{\beta}_i = \boldsymbol{r}_i - \boldsymbol{r}_s, \quad \boldsymbol{Q}_s = [\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_{s-1}, \boldsymbol{\beta}_{s+1}, \cdots, \boldsymbol{\beta}_m], \\ &\boldsymbol{y}_s = (\alpha_1, \cdots, \alpha_{s-1}, \alpha_{s+1}, \cdots, \alpha_m)^{\mathrm{T}}. \end{aligned}$$

那么式 (3.112) 可写为 $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{r}_s + \boldsymbol{Q}_s\boldsymbol{y}_s$, 从而求

$$\boldsymbol{x} = \sum_{i=1}^{m} \alpha_i \boldsymbol{x}_i = \alpha_1\boldsymbol{x}_1 + \alpha_2\boldsymbol{x}_2 + \cdots + \alpha_m\boldsymbol{x}_m$$

使得 $\|\boldsymbol{r}(\boldsymbol{x})\|_2 = \min$, 等价于求 $\boldsymbol{y}_s \in \mathbb{C}^{m-1}$ 使得 $\|\boldsymbol{r}_s + \boldsymbol{Q}_s\boldsymbol{y}_s\|_2 = \min$, 其极小范数解为 (见定理 5.2 和定理 5.3)

$$\boldsymbol{y}_s = -\boldsymbol{Q}_s^{\dagger}\boldsymbol{r}_s. \tag{3.113}$$

再由 $\alpha_s = 1 - \sum\limits_{i\neq s} \alpha_i$, 即可确定数组 $\alpha_1, \cdots, \alpha_m$, 从而可确定校正解 $\boldsymbol{x}$.

特别地, 对于 $m=2$, 取 $s=2$ 时, 有

$$\begin{cases} \alpha_1 = y_2 = -\boldsymbol{Q}_2{}^{\dagger}\boldsymbol{r}_2 = -\boldsymbol{\beta}_1^{\dagger}\boldsymbol{r}_2 = -\dfrac{(\boldsymbol{r}_1-\boldsymbol{r}_2)^{\mathrm{H}}\boldsymbol{r}_2}{\|\boldsymbol{r}_1-\boldsymbol{r}_2\|_2^2}, \\ \alpha_2 = 1-\alpha_1 = \dfrac{(\boldsymbol{r}_1-\boldsymbol{r}_2)^{\mathrm{H}}\boldsymbol{r}_1}{\|\boldsymbol{r}_1-\boldsymbol{r}_2\|_2^2}. \end{cases} \tag{3.114}$$

取 $s=1$ 时,

$$\begin{cases} \alpha_2 = y_1 = -\boldsymbol{Q}_1^{\dagger}\boldsymbol{r}_1 = -\boldsymbol{\beta}_2^{\dagger}\boldsymbol{r}_1 = -\dfrac{(\boldsymbol{r}_2-\boldsymbol{r}_1)^{\mathrm{H}}\boldsymbol{r}_1}{\|\boldsymbol{r}_2-\boldsymbol{r}_1\|_2^2}, \\ \alpha_1 = 1-\alpha_2 = \dfrac{(\boldsymbol{r}_2-\boldsymbol{r}_1)^{\mathrm{H}}\boldsymbol{r}_2}{\|\boldsymbol{r}_2-\boldsymbol{r}_1\|_2^2}. \end{cases} \tag{3.115}$$

易见, 式 (3.114) 与式 (3.115) 的结果一致.

下面来看模型 (3.111) 的几何意义 (图3.1). 把 $\boldsymbol{r}(\boldsymbol{x})$ 和 $\boldsymbol{r}_i$ 分别看作以坐标原点为

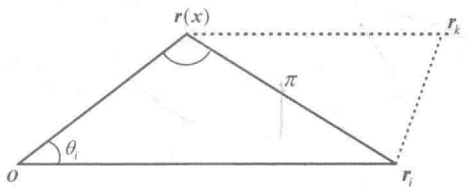

图 3.1 整体校正模型

起点, 以 $\boldsymbol{r}(\boldsymbol{x})$ 和 $\boldsymbol{r}_i$ 为终点的空间向量, 那么由 $\boldsymbol{r}(\boldsymbol{x}) = \alpha_1\boldsymbol{r}_1 + \alpha_2\boldsymbol{r}_2 + \cdots + \alpha_m\boldsymbol{r}_m$ 及 $\alpha_1 + \alpha_2 + \cdots + \alpha_m = 1$ 可得: 点 $\boldsymbol{r}(\boldsymbol{x})$ 在由点 $\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_m$ 确定的超平面 π 上, 且当 $i \neq s$ 时, 有

$$\begin{aligned} &(\boldsymbol{r}_i - \boldsymbol{r}_s)^{\mathrm{H}}\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{\beta}_i^{\mathrm{H}}(\boldsymbol{r}_s + \boldsymbol{Q}_s\boldsymbol{y}_s) \\ &= \boldsymbol{\beta}_i^{\mathrm{H}}(\boldsymbol{r}_s - \boldsymbol{Q}_s\boldsymbol{Q}_s^{\dagger}\boldsymbol{r}_s) = \boldsymbol{\beta}_i^{\mathrm{H}}[\boldsymbol{r}_s - (\boldsymbol{Q}_s\boldsymbol{Q}_s^{\dagger})^{\mathrm{H}}\boldsymbol{r}_s] \\ &= [\boldsymbol{\beta}_i^{\mathrm{H}} - (\boldsymbol{Q}_s\boldsymbol{Q}_s^{\dagger}\boldsymbol{\beta}_i)^{\mathrm{H}}]\boldsymbol{r}_s = (\boldsymbol{\beta}_i^{\mathrm{H}} - \boldsymbol{\beta}_i^{\mathrm{H}})\boldsymbol{r}_s = 0. \end{aligned} \tag{3.116}$$

上述推导过程用到了广义逆矩阵的性质

$$(\boldsymbol{Q}_s\boldsymbol{Q}_s^{\dagger})^{\mathrm{H}} = \boldsymbol{Q}_s\boldsymbol{Q}_s^{\dagger}, \quad \boldsymbol{Q}_s\boldsymbol{Q}_s^{\dagger}\boldsymbol{Q}_s = \boldsymbol{Q}_s.$$

式 (3.116) 表明 $\boldsymbol{r}(\boldsymbol{x}) \perp \pi$. 由于"垂足"为 $\boldsymbol{r}(\boldsymbol{x})$, 所以

$$\|\boldsymbol{r}(\boldsymbol{x})\|_2 \leqslant \|\boldsymbol{r}_i\|_2, \quad i = 1, 2, \cdots, m. \tag{3.117}$$

定理 3.35 模型 (3.111) 的校正解 $\boldsymbol{x}$ 满足 $\|\boldsymbol{r}(\boldsymbol{x})\|_2 < \min\limits_{1\leqslant i\leqslant m}\|\boldsymbol{r}_i\|_2$ 的充分必要条件是 $\boldsymbol{x} \neq \boldsymbol{x}_i\,(i = 1, 2, \cdots, m)$.

证明 必要性. 由 $\|\boldsymbol{r}(\boldsymbol{x})\|_2 < \|\boldsymbol{r}_i\|_2$, 得 $\boldsymbol{r}(\boldsymbol{x}) \neq \boldsymbol{r}_i$, 即 $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} \neq \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_i$, 从而有

$$\boldsymbol{x} \neq \boldsymbol{x}_i, \quad (i = 1, 2, \cdots, m).$$

充分性. 由 $\boldsymbol{x} \neq \boldsymbol{x}_i$, 得 $\boldsymbol{A}\boldsymbol{x} \neq \boldsymbol{A}\boldsymbol{x}_i$, 即 $\boldsymbol{r}(\boldsymbol{x}) \neq \boldsymbol{r}_i$, 从而有

$$\|\boldsymbol{r}(\boldsymbol{x})\|_2 \neq \|\boldsymbol{r}_i\|_2 \quad \text{(垂足唯一性)}.$$

再由式 (3.117), 得 $\|\boldsymbol{r}(\boldsymbol{x})\|_2 < \|\boldsymbol{r}_i\|_2,\ i = 1, 2, \cdots, m.$ 证毕. □

如果 $m = n+1$, 且 $\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_{s-1}, \boldsymbol{\beta}_{s+1}, \cdots, \boldsymbol{\beta}_m$ 线性无关, 则由 $\boldsymbol{r}(\boldsymbol{x}) \perp \boldsymbol{\beta}_i\ (i \neq s)$ 可得 $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{0}$, 从而校正解 $\boldsymbol{x} = \boldsymbol{x}^*$.

为了衡量校正过程的效果, 引进整体缩减系数

$$\alpha = \frac{\|\boldsymbol{r}(\boldsymbol{x})\|_2}{\min\limits_{1 \leqslant i \leqslant m} \|\boldsymbol{r}_i\|_2}. \tag{3.118}$$

由式 (3.117) 知 $0 \leqslant \alpha \leqslant 1$. 当 $\alpha = 1$ 时, 称校正过程失败; 当 $0 \leqslant \alpha < 1$ 时, 称校正过程成功. 特别地, 当 $\alpha = 0$ 时, 称校正过程完成, 此时校正解 $\boldsymbol{x} = \boldsymbol{x}^*$. 若用 θ_i 表示 $\boldsymbol{r}(\boldsymbol{x})$ 与 $\boldsymbol{r}_i$ 之间的夹角 (图 3.1), 则由 $\boldsymbol{r}(\boldsymbol{x}) \perp \pi$ 及 $\boldsymbol{r}(\boldsymbol{x}) \in \pi$ 可得 $\boldsymbol{r}(\boldsymbol{x}) \perp (\boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{r}_i)$. 于是有

$$\begin{aligned}
\sin\theta_i &= \frac{\|\boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{r}_i\|_2}{\|\boldsymbol{r}_i\|_2}, \\
\cos\theta_i &= \frac{\|\boldsymbol{r}(\boldsymbol{x})\|_2}{\|\boldsymbol{r}_i\|_2} = \left(1 - \frac{\|\boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{r}_i\|_2^2}{\|\boldsymbol{r}_i\|_2^2}\right)^{\frac{1}{2}} \\
&= \left(1 - \frac{\|\boldsymbol{Q}_i \boldsymbol{y}_i\|_2^2}{\|\boldsymbol{r}_i\|_2^2}\right)^{\frac{1}{2}} = \left(1 - \frac{\|Q_i Q_i^{\dagger} \boldsymbol{r}_i\|_2^2}{\|\boldsymbol{r}_i\|_2^2}\right)^{\frac{1}{2}}.
\end{aligned}$$

选取 s_0 满足$\|\boldsymbol{r}_{s_0}\|_2 < \min\limits_{1 \leqslant i \leqslant m} \|\boldsymbol{r}_i\|_2$, 则有

$$\alpha = \frac{\|\boldsymbol{r}(\boldsymbol{x})\|_2}{\|\boldsymbol{r}_{s_0}\|_2} = \left(1 - \frac{\|Q_{s_0} Q_{s_0}^{\dagger} \boldsymbol{r}_{s_0}\|_2^2}{\|\boldsymbol{r}_{s_0}\|_2^2}\right)^{\frac{1}{2}}. \tag{3.119}$$

应用式 (3.119) 能够事先估计校正过程的效果.

将整体校正过程施加于任何一种求解方程组 (3.1) 的迭代格式, 都可以改善原始式的收敛性态. 设求解方程组 (3.1) 的单步迭代格式为

$$\boldsymbol{x}^{(i)} = \varphi(\boldsymbol{x}^{(i-1)}) \quad (i = 1, 2, 3, \cdots). \tag{3.120}$$

改造格式 (3.120) 为以下算法.

算法 3.9 (整体校正法)

步 1, 输入矩阵 $\boldsymbol{A}$, 右端向量 $\boldsymbol{b}$, 初始点 $\boldsymbol{x}^{(0)}$, 精度要求 ε, 最大迭代次数 N, 取定整数 m 及 $s \in [1, m]$. 置 $k := 1$.

步 2, 置 $\boldsymbol{x}_1^{(k)} = \boldsymbol{x}^{(k-1)}$, 计算

$$\boldsymbol{x}_i^{(k)} = \varphi(\boldsymbol{x}_{i-1}^{(k)}), \quad i = 2, 3, \cdots, m.$$

步 3, 由式 (3.113) 计算 $\boldsymbol{y}_s^{(k)} = (\alpha_1^{(k)}, \cdots, \alpha_{s-1}^{(k)}, \alpha_{s+1}^{(k)}, \cdots, \alpha_m^{(k)})^{\mathrm{T}}$ 及

$$\alpha_s^{(k)} = 1 - \sum_{i=1,\, i \neq s}^{m} \alpha_i^{(k)}.$$

步 4, 计算 $\boldsymbol{x}^{(k)}=\alpha_1^{(k)}\boldsymbol{x}_1^{(k)}+\alpha_2^{(k)}\boldsymbol{x}_2^{(k)}+\cdots+\alpha_m^{(k)}\boldsymbol{x}_m^{(k)}$.

步 5, 若 $\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k)}\|/\|\boldsymbol{b}\|\leqslant\varepsilon$, 则停算, 输出 $\boldsymbol{x}^{(k)}$ 作为方程组的近似解.

步 6, 置 $k:=k+1$, 转步 2.

注 3.5 记 $\boldsymbol{r}_i^{(k)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_i^{(k)}$, 由式 (3.117) 可得: $\|\boldsymbol{r}^{(k)}\|_2$ 是坐标原点与"由点 $\boldsymbol{r}^{(k-1)},\boldsymbol{r}_2^{(k)},\cdots,\boldsymbol{r}_m^{(k)}$"确定的超平面 π_k 之间的最短距离. 因此, $\|\boldsymbol{r}^{(k)}\|_2<\|\boldsymbol{r}^{(k-1)}\|_2$ 等价于 π_k 不与"以 $\boldsymbol{r}^{(k-1)}$ 为法向量"的超平面

$$S_k=\{\boldsymbol{r}(\boldsymbol{z}):\boldsymbol{r}^{(k-1)}\perp(\boldsymbol{r}(\boldsymbol{z})-\boldsymbol{r}^{(k-1)}),\ \boldsymbol{z}\in\mathbb{C}^n\}$$

重合, 即存在 $i_0\in\{2,3,\cdots,m\}$, 使 $\boldsymbol{r}_{i_0}^{(k)}\notin S_k$, 也就是

$$\boldsymbol{x}_{i_0}^{(k)}\notin T_k=\{\boldsymbol{z}:\boldsymbol{r}^{(k-1)}\perp(\boldsymbol{r}(\boldsymbol{z})-\boldsymbol{r}^{(k-1)}),\ \boldsymbol{z}\in\mathbb{C}^n\}.$$

若用空间 $\mathbb{C}^n$ 中的"体积"定义集合的测度, 那么超平面集合 T_k 的测度值为零. 因此, $\boldsymbol{x}_i^{(k)}\ (i=2,3,\cdots,m)$ 几乎不同时落入集合 T_k, 从而可得: $\|\boldsymbol{r}^{(k)}\|_2<\|\boldsymbol{r}^{(k-1)}\|_2$ 几乎对所有的 k 成立.

一方面, 当式 (3.120) 收敛时, 算法 3.9 必定收敛. 进一步, 若第 k 次校正过程的整体缩减系数

$$\alpha^{(k)}=\frac{\|\boldsymbol{r}^{(k)}\|_2}{\min\limits_{1\leqslant i\leqslant m}\|\boldsymbol{r}_i^{(k)}\|_2}<1,$$

则算法 3.9 比式 (3.120) 收敛得快; 另一方面, 当式 (3.120) 不收敛时, 由于 $\|\boldsymbol{r}^{(i)}\|_2<\|\boldsymbol{r}^{(i-1)}\|_2$ 几乎对所有的 i 成立, 所以算法 3.9 也能收敛.

例 3.3 设

$$\boldsymbol{A}=\begin{bmatrix}1.2&0.3&0.4\\0.4&1.2&0.3\\0.3&0.4&1.2\end{bmatrix},\ \boldsymbol{b}=\begin{bmatrix}1\\1\\1\end{bmatrix}.$$

取 $\boldsymbol{x}^{(0)}=(1,2,0)^{\mathrm{T}}$, 式 (3.120) 采用 Richardson 迭代格式 (3.26), 其中

$$\boldsymbol{B}=\boldsymbol{I}-\boldsymbol{A}=-\begin{bmatrix}0.2&0.3&0.4\\0.4&0.2&0.3\\0.3&0.4&0.2\end{bmatrix}.$$

取终止准则值为

$$\frac{\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k)}\|_2}{\|\boldsymbol{b}\|_2}\leqslant10^{-5}.$$

整体校正法中的 $m=2$, $s=2$. 由 $\rho(\boldsymbol{B})\leqslant\|\boldsymbol{B}\|_1=0.9$ 知格式 (3.4) 收敛, 而算法 3.9 收敛较快. 计算结果如表 3.3 所示.

方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的精确解为 $\boldsymbol{x}^*=\left(\frac{1}{1.9},\frac{1}{1.9},\frac{1}{1.9}\right)^{\mathrm{T}}$.

表 3.3 整体校正取 $m=2, k=2$

迭代法	迭代次数	CPU 时间	相对残差
Richardson 迭代	109	0.0004	9.2614e-06
算法 3.9	11	0.0004	2.9541e-06

例 3.4 设

$$\boldsymbol{A}=\begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{bmatrix}\in\mathbb{R}^{1023\times 1023},\quad \boldsymbol{b}=\begin{bmatrix} 3 \\ 2 \\ \vdots \\ 2 \\ 3 \end{bmatrix}.$$

取 $\boldsymbol{x}^{(0)}=\boldsymbol{0}$, 式 (3.120) 仍采用 Richardson 迭代格式 (3.26), 其中

$$\boldsymbol{B}=\boldsymbol{I}-\boldsymbol{A}=\begin{bmatrix} -3 & 1 & & & \\ 1 & -3 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -3 & 1 \\ & & & 1 & -3 \end{bmatrix}.$$

取终止准则值为

$$\frac{\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k)}\|_2}{\|\boldsymbol{b}\|_2}\leqslant 10^{-10}.$$

整体校正法中的 $m=10$, $s=10$. 由 $\rho(\boldsymbol{B})>1$ 知格式 (3.4) 发散, 而算法 3.9 收敛. 计算结果如表 3.4 所示.

表 3.4 整体校正取 $m=10, k=10$

迭代法	迭代次数	CPU 时间	相对残差
Richardson 迭代	1000	0.4950	NaN
算法 3.9	5	0.1458	2.7583e-12

3.5.3 基于矩阵特征值的外推方法

考虑迭代格式 (3.4), 当 $\rho(\boldsymbol{B})<1$, 但 $\rho(\boldsymbol{B})\approx 1$ 时, 格式收敛缓慢. 下面使用外推技术, 改善格式 (3.4) 的收敛性态.

设迭代矩阵 $\boldsymbol{B}$ 的 n 个实特征值满足 $1>|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$, 对应的特征向量分别为 $\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_n$, 且它们线性无关. 分解

$$\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}=c_1\boldsymbol{z}_1+c_2\boldsymbol{z}_2+\cdots+c_n\boldsymbol{z}_n,$$

$$\boldsymbol{x}^{(i+1)}-\boldsymbol{x}^{(i)}=\boldsymbol{B}^i(\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)})=c_1\lambda_1^i\boldsymbol{z}_1+c_2\lambda_2^i\boldsymbol{z}_2+\cdots+c_n\lambda_n^i\boldsymbol{z}_n,$$

则有

$$\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}=c_1\lambda_1^k\boldsymbol{z}_1+O(\lambda_2^k).$$

由于 $\rho(\boldsymbol{B})<1$, 所以 $\lim\limits_{m\to\infty}\boldsymbol{x}^{(m)}=\boldsymbol{x}^*$, 且有

$$\begin{aligned}\boldsymbol{x}^*-\boldsymbol{x}^{(k)}&=\lim_{m\to\infty}(\boldsymbol{x}^{(m)}-\boldsymbol{x}^{(k)})=\lim_{m\to\infty}\sum_{i=k}^{m-1}(\boldsymbol{x}^{(i+1)}-\boldsymbol{x}^{(i)})\\&=\lim_{m\to\infty}\sum_{i=k}^{m-1}(c_1\lambda_1^i\boldsymbol{z}_1+c_2\lambda_2^i\boldsymbol{z}_2+\cdots+c_n\lambda_n^i\boldsymbol{z}_n)\\&=\frac{\lambda_1^k}{1-\lambda_1}c_1\boldsymbol{z}_1+\frac{\lambda_2^k}{1-\lambda_2}c_2\boldsymbol{z}_2+\cdots+\frac{\lambda_n^k}{1-\lambda_n}c_n\boldsymbol{z}_n.\end{aligned}\tag{3.121}$$

从而可得

$$\boldsymbol{x}^*=\boldsymbol{x}^{(k)}+\frac{1}{1-\lambda_1}(\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)})+O(\lambda_2^k).$$

记

$$\widetilde{\boldsymbol{x}}^{(k+1)}=\boldsymbol{x}^{(k)}+\frac{1}{1-\lambda_1}(\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}),$$

则有 $\widetilde{\boldsymbol{x}}^{(k+1)}-\boldsymbol{x}^*=O(\lambda_2^k)$. 再由式 (3.121), 得 $\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^*=O(\lambda_1^{k+1})$.

当 $|\lambda_2|\ll|\lambda_1|<1$ 且 $|\lambda_1|\approx1$ 时, 有

$$\frac{\lambda_2^k}{\lambda_1^{k+1}}=\left(\frac{\lambda_2}{\lambda_1}\right)^k\frac{1}{\lambda_1}\to0\quad(k\to\infty).$$

因此, $\widetilde{\boldsymbol{x}}^{(k+1)}$ 比 $\boldsymbol{x}^{(k+1)}$ 更接近于 $\boldsymbol{x}^*$.

将外推技术施加于迭代格式 (3.4), 即可得到基于迭代矩阵特征值的外推格式.

算法 3.10 (基于特征值的外推法)

给定整数 $m\geqslant1$, 初始值 $\boldsymbol{x}_1^{(0)}$ 及最大迭代次数 N.

while $i\leqslant N$

  for $k=1:m$

    $\boldsymbol{x}_i^{(k)}=\boldsymbol{B}\boldsymbol{x}_i^{(k-1)}+\boldsymbol{f}$;

  end

  $\boldsymbol{x}_{i+1}^{(0)}=\boldsymbol{x}_i^{(m-1)}+\dfrac{1}{1-\lambda_1}(\boldsymbol{x}_i^{(m)}-\boldsymbol{x}_i^{(m-1)})$;

  $i=i+1$;

end

易知, 当 $|\lambda_2|\ll|\lambda_1|<1$ 且 $|\lambda_1|\approx1$ 时, 算法 3.10 比迭代格式 (3.4) 收敛得快.

例 3.5 在例 3.2 中, 用 Jacobi 迭代法收敛的速度相当缓慢. 现在对 Jacobi 迭代法进行加速. 先计算 Jacobi 迭代矩阵 $\boldsymbol{B}_J$ 的特征值. 注意到 $\boldsymbol{B}_J$ 可以表示为

$$\boldsymbol{B}_J = \boldsymbol{D}^{-1}(\boldsymbol{D}-\boldsymbol{A}) = \frac{1}{4}(2\boldsymbol{I}-\boldsymbol{S}), \tag{3.122}$$

式中: $\boldsymbol{I}$ 为 n 阶单位矩阵.

$$\boldsymbol{S} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}. \tag{3.123}$$

S 的第 k 个特征值为 $4\sin^2\left(\frac{k\pi}{2(n+1)}\right)$, $k=1,2,\cdots,n$. 故 $\boldsymbol{B}_J$ 的第 k 个特征值为

$$\lambda_k = \frac{1}{4}\left[2-4\sin^2\left(\frac{k\pi}{2(n+1)}\right)\right] = \frac{1}{2}\cos\frac{k\pi}{n+1},\ k=1,2,\cdots,n. \tag{3.124}$$

由式 (3.124) 可知 $\boldsymbol{B}_J$ 模最大特征值为 $\lambda_1 = \frac{1}{2}\cos\frac{\pi}{n+1}$. 取 $n = 2^{14}-1 = 16383$, 初始向量为零向量, 容许误差为 10^{-10}. 利用算法 3.10, 得到计算结果如表 3.5 所示.

表 3.5 Jacobi 迭代法和特征值外推法的数值比较

迭代格式	迭代次数 (k)	CPU 时间	相对残差
Jacobi 迭代法	34	0.0158	5.8182e-11
特征值外推法	4	0.0076	1.1501e-12

3.5.4 Chebyshev 加速方法

考虑迭代格式 (3.4). 设 $\rho(\boldsymbol{B})<1$, 由初始向量 $\boldsymbol{x}^{(0)}$ 出发, 经过 k 次迭代得到 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(k)}$ $(k\geqslant 1)$. 选择数组 $\alpha_0^{(k)},\alpha_1^{(k)},\cdots,\alpha_k^{(k)}$, 使得

$$\boldsymbol{u}^{(k)} = \sum_{i=0}^{k}\alpha_i^{(k)}\boldsymbol{x}^{(i)},\quad \sum_{i=0}^{k}\alpha_i^{(k)} = 1. \tag{3.125}$$

设 $\boldsymbol{x}^* = \boldsymbol{B}\boldsymbol{x}^* + \boldsymbol{f}$, 则

$$\boldsymbol{u}^{(k)} - \boldsymbol{x}^* = \sum_{i=0}^{k}\alpha_i^{(k)}(\boldsymbol{x}^{(i)}-\boldsymbol{x}^*) = \sum_{i=0}^{k}\alpha_i^{(k)}\boldsymbol{B}^i(\boldsymbol{x}^{(0)}-\boldsymbol{x}^*).$$

定义 k 次多项式

$$p_k(t) = \sum_{i=0}^{k}\alpha_i^{(k)}t^i,\ \ p_k(1)=1,$$

即有

$$\boldsymbol{u}^{(k)} - \boldsymbol{x}^* = p_k(\boldsymbol{B})(\boldsymbol{x}^{(0)}-\boldsymbol{x}^*).$$

根据上式, 有

$$\|\boldsymbol{u}^{(k)}-\boldsymbol{x}^*\|_2 \leqslant \|p_k(\boldsymbol{B})\|_2 \cdot \|\boldsymbol{x}^{(0)}-\boldsymbol{x}^*\|_2.$$

对所有的 $\|p_k(\boldsymbol{B})\|_2$ 取下确界就是 $\rho(p_k(\boldsymbol{B}))$. 很自然地, 要求选取多项式 p_k 使得 $\rho(p_k(\boldsymbol{B}))$ 达到最小. 设 S 是复平面上包含 $\boldsymbol{B}$ 的所有特征值的集合, 则

$$\rho(p_k(\boldsymbol{B})) \leqslant \max_{\lambda\in\lambda(\boldsymbol{B})}|p_k(\lambda)| \leqslant \max_{\lambda\in S}|p_k(\lambda)|.$$

设迭代矩阵 $\boldsymbol{B}$ 的 n 个特征值 $\lambda_1,\lambda_2,\cdots,\lambda_n$ 均为实数, 都在区间 $[a,b]$ 上, 并且 $1\notin[a,b]$. 这里只考虑 $a<b<1$ 的情形. 为了使 $\rho(p_k(\boldsymbol{B}))$ 达到极小, 只要考虑极小极大问题

$$\min_{p_k\in\mathcal{P}_k^{(1)}}\left\{\max_{\lambda\in[a,b]}|p_k(\lambda)|\right\}, \tag{3.126}$$

式中: $\mathcal{P}_k^{(1)}$ 为满足 $p_k(1)=1$ 所有次数不超过 k 的实系数多项式的集合. 这是一个在多项式空间有归一化条件的最佳一致逼近问题, 它与 Chebyshev 多项式有关. k 次 Chebyshev 多项式 $C_k(t)$ 是极小极大问题

$$\min_{p_k\in\mathcal{P}_k^{(1)}}\left\{\max_{t\in[-1,1]}|p_k(t)|\right\} \tag{3.127}$$

的解. k 次 Chebyshev 多项式的定义为

$$C_k(t)=\begin{cases}\cos(k\cdot\arccos t), & |t|\leqslant 1,\\ \cosh(k\cdot\operatorname{arccosh} t), & |t|>1,\end{cases} \tag{3.128}$$

它满足递推关系

$$\begin{cases}C_0(t)=1,\quad C_1(t)=t,\\ C_{k+1}(t)=2tC_k(t)-C_{k-1}(t),\quad k\geqslant 1.\end{cases} \tag{3.129}$$

下面的引理给出了在归一化条件下极小极大问题 (3.127) 的最优解.

引理 3.5 设 $\beta\notin[-1,1]$. 若 $p_k\in\mathcal{P}_k$, $p_k(\beta)=1$, 则

$$\|p_k\|\equiv\max_{-1\leqslant t\leqslant 1}|p_k(t)|\geqslant\frac{1}{C_k(\beta)}.$$

特别地, 当取 $p_k(t)=\dfrac{C_k(t)}{C_k(\beta)}$ 时, $\|p_k\|=\dfrac{1}{C_k(\beta)}$.

证明 令 $\alpha=\dfrac{1}{C_k(\beta)}$. 不难求得 $C_k(t)$ 的极值点为 $t_i=\cos\left(\dfrac{i\pi}{k}\right)$, 满足 $C_k(t_i)=\cos(i\pi)=(-1)^i$, $i=0,1,\cdots,k$. 用反证法. 若 $\|p_k\|<\alpha$, 则有

$$\operatorname{sign}(\alpha)(-1)^i\big(\alpha C_k(t_i)-p_k(t_i)\big)\geqslant|\alpha|-\|p_k\|>0,$$

这表明多项式 $\alpha C_k(t)-p_k(t)$ 在 $t_i\,(i=0,1,\cdots,k)$ 处有交替的正负号, 所以它在 $(-1,1)$ 中有 k 个零点. 但 $\beta\notin[-1,1]$ 也是 $\alpha C_k(t)-p_k(t)$ 的零点, 故 k 次多项式 $\alpha C_k(t)-p_k(t)$ 有 $k+1$ 个零点, 矛盾. 因此, 必有 $\|p_k\|\geqslant\alpha$. 另外, 显然当 $p_k(t)=C_k(t)/C_k(\beta)$ 时, $\|p_k\|=\alpha$. 证毕. □

下面的引理给出了极小极大问题 (3.126) 的最优解.

引理 3.6 设 $a<b<1$. 若 $p_k\in\mathcal{P}_k$ 且满足 $p_k(1)=1$, 则

$$\|p_k\|\equiv\max_{a\leqslant t\leqslant b}|p_k(t)|\geqslant\frac{1}{C_k(w(1))},$$

式中:

$$w(t)=\frac{2t-a-b}{b-a}.$$

特别地, 当取 $p_k(t)=\dfrac{C_k(w(t))}{C_k(w(1))}$ 时, $\|p_k\|=\dfrac{1}{C_k(w(1))}$.

证明 记 $\beta=w(1)=\dfrac{2-a-b}{b-a}>1$. 注意到 $w(t):[a,b]\to[-1,1]$, 取 $q_k\in\mathcal{P}_k$, 使得 $p_k(t)=q_k(w(t))$. 显然有 $1=p_k(1)=q_k(w(1))=q_k(\beta)$. 根据引理 3.5, 有

$$\|q_k\|=\max_{-1\leqslant w(t)\leqslant1}|q_k(w(t))|\geqslant\frac{1}{C_k(\beta)},$$

即

$$\|p_k\|=\max_{a\leqslant t\leqslant b}|p_k(t)|\geqslant\frac{1}{C_k(\beta)}=\frac{1}{C_k(w(1))}.$$

此外, 显然当 $p_k(t)=\dfrac{C_k(w(t))}{C_k(w(1))}$ 时, $\|p_k\|=\dfrac{1}{C_k(w(1))}$. 证毕. □

注 3.6 若 $1<a<b$, 只需作变换 $w(t):[a,b]\to[-1,1]$,

$$w(t)=\frac{2t-a-b}{a-b},$$

则有

$$\beta=w(1)=\frac{2-a-b}{a-b}=1+2\frac{1-a}{a-b}>1,$$

从而亦可推导出, 当 $p_k(t)=\dfrac{C_k(w(t))}{C_k(w(1))}$ 时, $\|p_k\|=\dfrac{1}{C_k(w(1))}$.

下面的引理给出了最优解 p_k 的递推关系.

引理 3.7 $p_k(t)=C_k(w(t))/C_k(w(1))$ 满足如下的递推关系:

$$\begin{cases}p_0(t)=1,\quad p_1(t)=w(t)/w(1)=(2t-a-b)/(2-a-b),\\ p_k(t)=\rho_kp_1(t)p_{k-1}(t)+(1-\rho_k)p_{k-2}(t),\quad k\geqslant2,\end{cases}\tag{3.130}$$

式中: ρ_k 满足的递推关系为

$$\rho_1=2,\quad \rho_k=(1-\alpha\rho_{k-1})^{-1},\quad \alpha=(2w(1))^{-2},\quad k\geqslant2.\tag{3.131}$$

证明 令 $\beta_k = C_k(w(1))$. 根据 C_k 的递推关系, 得

$$\begin{aligned} p_k(t) &= \beta_k^{-1} C_k(w(t)) = \beta_k^{-1}\big[2w(t)C_{k-1}(w(t)) - C_{k-2}(w(t))\big] \\ &= 2\beta_k^{-1}\beta_{k-1}w(1)p_1(t)p_{k-1}(t) - \beta_k^{-1}\beta_{k-2}p_{k-2}(t). \end{aligned} \tag{3.132}$$

定义

$$\rho_k = 2\beta_k^{-1}\beta_{k-1}w(1) = \alpha^{-1/2}\beta_k^{-1}\beta_{k-1}. \tag{3.133}$$

则由 C_k 的递推过程可以推出 β_k 的递推关系为

$$\beta_k = C_k(w(1)) = 2w(1)C_{k-1}(w(1)) - C_{k-2}(w(1)) = \alpha^{-1/2}\beta_{k-1} - \beta_{k-2}.$$

从而有

$$\beta_k^{-1}\beta_{k-2} = \beta_k^{-1}(\alpha^{-1/2}\beta_{k-1} - \beta_k) = \rho_k - 1. \tag{3.134}$$

将式 (3.133) 和式 (3.134) 代入式 (3.132) 即可得到 p_k 的递推公式 (3.130). ρ_k 的递推关系亦可从 β_k 的递推过程中得到, 即由式 (3.133), 得

$$\begin{aligned} \rho_k &= \alpha^{-1/2}\beta_{k-1}\beta_k^{-1} = \alpha^{-1/2}\beta_{k-1}\big(\alpha^{-1/2}\beta_{k-1} - \beta_{k-2}\big)^{-1} \\ &= \big(1 - \alpha^{1/2}\beta_{k-1}^{-1}\beta_{k-2}\big)^{-1} = \big(1 - \alpha\rho_{k-1}\big)^{-1}. \end{aligned}$$

证毕. □

引理 3.7 的意义在于表明了由 Chebyshev 加速方法得到的序列 $\{\boldsymbol{u}^{(k)}\}$ 也有递推公式, 从而在加速过程中没有必要真正地引入 Chebyshev 多项式. 有下面的定理.

定理 3.36 *设 $\{\boldsymbol{u}^{(k)}\}$ 是由 Chebyshev 加速方法得到的序列, 则对于任意的 $\boldsymbol{u}^{(0)}$, 有如下的递推公式*

$$\begin{cases} \boldsymbol{u}^{(1)} = (1-\gamma)\boldsymbol{u}^{(0)} + \gamma(\boldsymbol{B}\boldsymbol{u}^{(0)} + \boldsymbol{f}), \quad \gamma = 2/(2-a-b), \\ \boldsymbol{u}^{(k)} = (1-\rho_k)\boldsymbol{u}^{(k-2)} + \rho_k\big[(1-\gamma)\boldsymbol{u}^{(k-1)} + \gamma(\boldsymbol{B}\boldsymbol{u}^{(k-1)} + \boldsymbol{f})\big], \ k \geqslant 2, \end{cases} \tag{3.135}$$

式中: 序列 $\{\rho_k\}$ 由式 (3.131) 所定义.

证明 由于

$$\boldsymbol{u}^{(k)} = \sum_{i=0}^{k} \alpha_i^{(k)}\boldsymbol{x}^{(i)}, \quad p_k(t) = \sum_{i=0}^{k}\alpha_i^{(k)}t^i, \quad \sum_{i=0}^{k}\alpha_i^{(k)} = 1,$$

故 $\boldsymbol{u}^{(0)} = \boldsymbol{x}^{(0)}$, $\boldsymbol{u}^{(1)}$ 由式 (3.135) 的第 1 式给出. 设 $\boldsymbol{u}^* = \boldsymbol{B}\boldsymbol{u}^* + \boldsymbol{f}$, 则

$$\begin{aligned} \boldsymbol{u}^{(k)} - \boldsymbol{u}^* &= p_k(\boldsymbol{B})(\boldsymbol{u}^{(0)} - \boldsymbol{u}^*) \\ &= \big[\rho_k p_1(\boldsymbol{B})p_{k-1}(\boldsymbol{B}) + (1-\rho_k)p_{k-2}(\boldsymbol{B})\big](\boldsymbol{u}^{(0)} - \boldsymbol{u}^*) \\ &= \rho_k p_1(\boldsymbol{B})(\boldsymbol{u}^{(k-1)} - \boldsymbol{u}^*) + (1-\rho_k)(\boldsymbol{u}^{(k-2)} - \boldsymbol{u}^*), \end{aligned}$$

即

$$\boldsymbol{u}^{(k)} = \rho_k p_1(\boldsymbol{B})\boldsymbol{u}^{(k-1)} + (1-\rho_k)\boldsymbol{u}^{(k-2)} + \rho_k\big(\boldsymbol{I} - p_1(\boldsymbol{B})\big)\boldsymbol{x}^*,\ k \geqslant 2. \tag{3.136}$$

注意到

$$p_1(\boldsymbol{B}) = (1-\gamma)\boldsymbol{I} + \gamma\boldsymbol{B},\ \ (\boldsymbol{I}-\boldsymbol{B})\boldsymbol{x}^* = \boldsymbol{f},$$

代入式 (3.136) 即得要证的递推式. 证毕. □

注 3.7 式 (3.135) 中的迭代格式表明, 对于 $\boldsymbol{u}^{(k-1)}$, 用原来的迭代格式迭代一次产生 $\boldsymbol{B}\boldsymbol{u}^{(k-1)} + \boldsymbol{f}$, 它与 $\boldsymbol{u}^{(k-1)}$ 组合产生 $(1-\gamma)\boldsymbol{u}^{(k-1)} + \gamma(\boldsymbol{B}\boldsymbol{u}^{(k-1)} + \boldsymbol{f})$, 其中组合系数 γ 由 a,b 决定, 这种组合就是外推格式. 这样外推之后的结果再与前一步迭代向量 $\boldsymbol{u}^{(k-2)}$ 做组合, 组合系数 ρ_k 由式 (3.131) 的递推公式确定. 可以期望, Chebyshev 加速应该比一般的外推加速具有更快的收敛速度.

Chebyshev 加速迭代格式的速度由它的迭代矩阵的谱半径确定. 有

$$\rho(p_k(\boldsymbol{B})) \leqslant \max_{a\leqslant t\leqslant b}|p_k(\lambda)| = \frac{1}{C_k(w(1))}.$$

经过简单的计算 (定理 4.2 (2)) 可知

$$\frac{1}{C_k(w(1))} = \frac{2}{r^k + r^{-k}},\ r = t + \sqrt{t^2-1},\ t = w(1) = \frac{2-a-b}{b-a} > 1.$$

外推格式和 Chebyshev 加速迭代格式都依赖于某个简单的迭代格式 (3.4). 当迭代格式 (3.4) 容易并行实现时, 相应的外推格式和 Chebyshev 加速迭代格式也都可以并行实现.

例 3.6 考虑用 Chebyshev 加速方法对例 3.2 中的 Jacobi 迭代法进行加速. 现在对 Jacobi 迭代法进行加速. 根据例 3.5, 可知 Jacobi 迭代矩阵 $\boldsymbol{B}_J$ 的 n 个特征值为

$$\lambda_k = \frac{1}{2}\cos\frac{k\pi}{n+1},\ k = 1,2,\cdots,n.$$

因此, $\boldsymbol{B}_J$ 所有特征值都在 $a = \lambda_n$ 和 $b = \lambda_1$ 之间. 取 $n = 2^{14} - 1 = 16383$, 初始向量为零向量, 容许误差为 10^{-10}. 由于 $\lambda_n = -\lambda_1$, 故外推参数 $\gamma = 2/(2-a-b) = 1$, 因此外推法 (3.108) 对此例不起作用. 得到计算结果如表 3.6 所示.

表 3.6 Jacobi 迭代法和 Chebyshev 加速迭代的数值比较

迭代格式	迭代次数 (k)	CPU 时间	相对残差
Jacobi 迭代法	34	0.0130	5.8182e-11
Chebyshev 加速	21	0.0089	6.0411e-11

3.6 块三对角方程组的迭代解法

本节讨论 $\boldsymbol{A}$ 为块三对角矩阵时, 方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的迭代解法. 设

$$\boldsymbol{A}=\begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{C}_1 & & & \\ \boldsymbol{A}_2 & \boldsymbol{B}_2 & \boldsymbol{C}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{A}_{m-1} & \boldsymbol{B}_{m-1} & \boldsymbol{C}_{m-1} \\ & & & \boldsymbol{A}_m & \boldsymbol{B}_m \end{bmatrix}, \tag{3.137}$$

式中: $\boldsymbol{B}_i$ 为 n_i 阶矩阵 $(n_1+n_2+\cdots+n_m=n)$.

3.6.1 PE(α) 方法

对矩阵 $\boldsymbol{A}$ 进行块三角分解 (假定这种分解存在), 即

$$\boldsymbol{A}=\begin{bmatrix} \boldsymbol{D}_1 & & & \\ \boldsymbol{A}_2 & \boldsymbol{D}_2 & & \\ & \ddots & \ddots & \\ & & \boldsymbol{A}_m & \boldsymbol{D}_m \end{bmatrix}\begin{bmatrix} \boldsymbol{I}_1 & \boldsymbol{U}_1 & & \\ & \ddots & \ddots & \\ & & \boldsymbol{I}_{m-1} & \boldsymbol{U}_{m-1} \\ & & & \boldsymbol{I}_m \end{bmatrix}, \tag{3.138}$$

式中: $\boldsymbol{I}_i$ 为 n_i 阶单位矩阵.

比较式 (3.137) 和式 (3.138), 得

$$\begin{aligned} &\boldsymbol{D}_1=\boldsymbol{B}_1, \\ &\boldsymbol{U}_i=\boldsymbol{D}_i^{-1}\boldsymbol{C}_i \quad (i=1,2,\cdots,m-1), \\ &\boldsymbol{D}_i=\boldsymbol{B}_i-\boldsymbol{A}_i\boldsymbol{U}_{i-1}=\boldsymbol{B}_i-\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1} \quad (i=2,3,\cdots,m). \end{aligned}$$

由此, 在一定条件下, 有

$$\begin{aligned} \boldsymbol{D}_i^{-1}&=(\boldsymbol{B}_i-\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1}=(\boldsymbol{I}_i-\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1}\boldsymbol{B}_i^{-1} \\ &=\left[\boldsymbol{I}_i+\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1}+(\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1})^2+\cdots\right]\boldsymbol{B}_i^{-1} \\ &\approx(\boldsymbol{I}_i+\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1})\boldsymbol{B}_i^{-1}. \end{aligned}$$

若将右端的 $\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{D}_{i-1}^{-1}\boldsymbol{C}_{i-1}$ 也省略, 则有 $\boldsymbol{D}_i^{-1}\approx\boldsymbol{B}_i^{-1}$. 于是, 用 $\boldsymbol{B}_{i-1}^{-1}$ 代替 $\boldsymbol{D}_{i-1}^{-1}$ 时, 上式可写为 $\boldsymbol{D}_i^{-1}\approx(\boldsymbol{I}_i+\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})\boldsymbol{B}_i^{-1}$, 或

$$\boldsymbol{D}_i\approx\boldsymbol{B}_i(\boldsymbol{I}_i+\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1}=\boldsymbol{S}_i, \tag{3.139}$$

这样得到 $\boldsymbol{A}$ 的一个近似块三角分解

$$\boldsymbol{A}\approx\begin{bmatrix} \boldsymbol{S}_1 & & & \\ \boldsymbol{A}_2 & \boldsymbol{S}_2 & & \\ & \ddots & \ddots & \\ & & \boldsymbol{A}_m & \boldsymbol{S}_m \end{bmatrix}\begin{bmatrix} \boldsymbol{I}_1 & \boldsymbol{T}_1 & & \\ & \ddots & \ddots & \\ & & \boldsymbol{I}_{m-1} & \boldsymbol{T}_{k-1} \\ & & & \boldsymbol{I}_m \end{bmatrix}, \tag{3.140}$$

式中:

$$\begin{cases} \boldsymbol{S}_1 = \boldsymbol{B}_1, \\ \boldsymbol{T}_i = \boldsymbol{S}_i^{-1}\boldsymbol{C}_i \quad (i = 1, 2, \cdots, m-1), \\ \boldsymbol{S}_i = \boldsymbol{B}_i(\boldsymbol{I}_i + \boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1} \quad (i = 2, 3, \cdots, m). \end{cases} \tag{3.141}$$

若用 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 分别表示式 (3.140) 右端的第 1 个和第 2 个矩阵, 则有 $\boldsymbol{A} \approx \boldsymbol{LU}$. 令

$$\boldsymbol{M} = \boldsymbol{LU} = \begin{bmatrix} \boldsymbol{S}_1 & \boldsymbol{C}_1 & & & \\ \boldsymbol{A}_2 & \boldsymbol{A}_2\boldsymbol{T}_1 + \boldsymbol{S}_2 & \boldsymbol{C}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{A}_{m-1} & \boldsymbol{A}_{m-1}\boldsymbol{T}_{m-2} + \boldsymbol{S}_{m-1} & \boldsymbol{C}_{m-1} \\ & & & \boldsymbol{A}_m & \boldsymbol{A}_m\boldsymbol{T}_{m-1} + \boldsymbol{S}_m \end{bmatrix},$$

$$\boldsymbol{N} = \boldsymbol{M} - \boldsymbol{A} = \operatorname{diag}\big(\boldsymbol{O}, \boldsymbol{N}_2, \cdots, \boldsymbol{N}_m\big), \tag{3.142}$$

式中: $\boldsymbol{N}_i = \boldsymbol{A}_i\boldsymbol{T}_{i-1} + \boldsymbol{S}_i - \boldsymbol{B}_i\,(i = 2, 3, \cdots, m)$. 那么 $\boldsymbol{Ax} = \boldsymbol{b}$ 等价于 $\boldsymbol{Mx} = \boldsymbol{Nx} + \boldsymbol{b}$. 建立迭代格式

$$\boldsymbol{Mx}^{(k+1)} = \boldsymbol{Nx}^{(k)} + \boldsymbol{b}, \quad k = 0, 1, 2, \cdots, \tag{3.143}$$

称式 (3.143) 为拟消去 (Pseudo-Elimination, PE) 迭代方法. 使用式 (3.143) 时, 对于每个 k, 求 $\boldsymbol{x}^{(k+1)}$ 可转化为下面两个块三角方程组的求解问题:

$$\boldsymbol{Lz}^{(k+1)} = \boldsymbol{Nx}^{(k)} + \boldsymbol{b}, \quad \boldsymbol{Ux}^{(k+1)} = \boldsymbol{z}^{(k+1)}.$$

若用 $\boldsymbol{b}_i, \boldsymbol{x}_i^{(k)}, \boldsymbol{x}_i^{(k+1)}, \boldsymbol{z}_i^{(k+1)}$ 分别表示向量 $\boldsymbol{b}, \boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k+1)}, \boldsymbol{z}^{(k+1)}$ 的第 i 个子向量, 则分组计算的递推公式为

$$\boldsymbol{z}_1^{(k+1)} = \boldsymbol{S}_1^{-1}\boldsymbol{b}_1, \quad \boldsymbol{z}_i^{(k+1)} = \boldsymbol{S}_i^{-1}(\boldsymbol{N}_i\boldsymbol{x}_i^{(k)} + \boldsymbol{b}_i - \boldsymbol{A}_i\boldsymbol{z}_{i-1}^{(k+1)}), \quad i = 2, 3, \cdots, m,$$

$$\boldsymbol{x}_m^{(k+1)} = \boldsymbol{z}_m^{(k+1)}, \quad \boldsymbol{x}_i^{(k+1)} = \boldsymbol{z}_i^{(k+1)} - \boldsymbol{T}_i\boldsymbol{x}_{i+1}^{(k+1)}, \quad i = m-1, \cdots, 2, 1.$$

下面讨论 PE 方法的收敛性.

定理 3.37 设 $\boldsymbol{A}$ 为 Hermite 矩阵, 则 PE 方法中的矩阵 $\boldsymbol{S}_i$, $\boldsymbol{M}$ 和 $\boldsymbol{N}$ 均为 Hermite 矩阵.

证明 由 $\boldsymbol{A}^{\mathrm{H}} = \boldsymbol{A}$ 知 $\boldsymbol{B}_i^{\mathrm{H}} = \boldsymbol{B}_i$, $\boldsymbol{A}_i^{\mathrm{H}} = \boldsymbol{C}_{i-1}$, 故有

$$\begin{aligned} \boldsymbol{S}_i^{\mathrm{H}} &= [(\boldsymbol{I}_i + \boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1}]^{\mathrm{H}}\boldsymbol{B}_i^{\mathrm{H}} = (\boldsymbol{I}_i + \boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}\boldsymbol{B}_i^{-1})^{-1}\boldsymbol{B}_i \\ &= [\boldsymbol{B}_i(\boldsymbol{I}_i + \boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})\boldsymbol{B}_i^{-1}]^{-1}\boldsymbol{B}_i = \boldsymbol{B}_i(\boldsymbol{I}_i + \boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1} \\ &= \boldsymbol{S}_i \end{aligned}$$

及

$$(\boldsymbol{A}_i\boldsymbol{T}_{i-1})^{\mathrm{H}} = (\boldsymbol{A}_i\boldsymbol{S}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{\mathrm{H}} = \boldsymbol{C}_{i-1}^{\mathrm{H}}\boldsymbol{S}_{i-1}^{-1}\boldsymbol{A}_i^{\mathrm{H}} = \boldsymbol{A}_i\boldsymbol{S}_{i-1}^{-1}\boldsymbol{C}_{i-1} = \boldsymbol{A}_i\boldsymbol{T}_{i-1}.$$

因此 $\boldsymbol{M}^{\mathrm{H}} = \boldsymbol{M}$, 从而 $\boldsymbol{N}^{\mathrm{H}} = (\boldsymbol{M} - \boldsymbol{A})^{\mathrm{H}} = \boldsymbol{M}^{\mathrm{H}} - \boldsymbol{A}^{\mathrm{H}} = \boldsymbol{N}$. 证毕. □

引理 3.8 设 $\boldsymbol{A}$ 的特征值 $\lambda(\boldsymbol{A})$ 为非负实数, $\boldsymbol{B}$ 为 Hermite 正定矩阵, $\boldsymbol{C}=\boldsymbol{AB}$ (或 $\boldsymbol{C}=\boldsymbol{BA}$) 为 Hermite 矩阵, 则 $\boldsymbol{C}$ 为半正定矩阵. 进一步, 若 $\lambda(\boldsymbol{A})$ 均为正数, 则 $\boldsymbol{C}$ 为正定矩阵.

证明 设 $\lambda(\boldsymbol{A})\geqslant 0$, 由于 $\boldsymbol{A}$ 相似于

$$\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{\frac{1}{2}}=\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^{-\frac{1}{2}}=\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{C}\boldsymbol{B}^{-\frac{1}{2}},$$

所以 $\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{C}\boldsymbol{B}^{-\frac{1}{2}}$ 的特征值也是非负实数, 从而 $\boldsymbol{C}$ 的特征值是非负实数. 故 $\boldsymbol{C}$ 为半正定矩阵. 同理, 可证 $\boldsymbol{C}=\boldsymbol{BA}$ 及 $\lambda(\boldsymbol{A})>0$ 的情形. 证毕. □

引理 3.9 设 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$, 且 $\boldsymbol{A}$ 和 $\boldsymbol{M}$ 都可逆, 则对 $\boldsymbol{M}^{-1}\boldsymbol{N}$ 的任意特征值 μ, 存在 $\boldsymbol{A}^{-1}\boldsymbol{N}$ 的某个特征值 λ, 使得

$$\mu=\frac{\lambda}{1+\lambda}. \tag{3.144}$$

证明 设 $\boldsymbol{M}^{-1}\boldsymbol{N}$ 的一个特征值为 μ, 对应的特征向量为 $\boldsymbol{z}\neq\boldsymbol{0}$, 则有 $(\boldsymbol{M}^{-1}\boldsymbol{N})\boldsymbol{z}=\mu\boldsymbol{z}$. 因为

$$\boldsymbol{M}^{-1}\boldsymbol{N}=(\boldsymbol{A}+\boldsymbol{N})^{-1}\boldsymbol{N}=(\boldsymbol{I}+\boldsymbol{A}^{-1}\boldsymbol{N})^{-1}(\boldsymbol{A}^{-1}\boldsymbol{N}),$$

所以

$$(\boldsymbol{I}+\boldsymbol{A}^{-1}\boldsymbol{N})^{-1}(\boldsymbol{A}^{-1}\boldsymbol{N})\boldsymbol{z}=\mu\boldsymbol{z},$$

即

$$(\boldsymbol{A}^{-1}\boldsymbol{N})\boldsymbol{z}=\mu(\boldsymbol{I}+\boldsymbol{A}^{-1}\boldsymbol{N})\boldsymbol{z}.$$

若 $\mu=1$, 则上式成为 $\boldsymbol{z}=\boldsymbol{0}$, 这与 $\boldsymbol{z}$ 是特征向量矛盾. 故 $\mu\neq 1$, 从而有

$$(\boldsymbol{A}^{-1}\boldsymbol{N})\boldsymbol{z}=\frac{\mu}{1-\mu}\boldsymbol{z},$$

这表明 $\dfrac{\mu}{1-\mu}$ 是 $\boldsymbol{A}^{-1}\boldsymbol{N}$ 的一个特征值, 记作 λ, 即 $\lambda=\dfrac{\mu}{1-\mu}$. 易见 $\lambda\neq -1$, 从而有 $\mu=\dfrac{\lambda}{1+\lambda}$. 证毕. □

定理 3.38 设 $\boldsymbol{A}$ 为 Hermite 正定矩阵, 则 PE 方法收敛.

证明 (1) 先证 PE 方法是适定的, 即算法中的矩阵 $\boldsymbol{S}_i\,(i=1,2,\cdots,m)$ 均可逆. 由 $\boldsymbol{A}$ 正定知 $\boldsymbol{B}_i$ 均为正定矩阵, 从而 $\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}=\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_i^{\mathrm{H}}$ 半正定, 这是因为 $\boldsymbol{z}\neq\boldsymbol{0}$ 时, 有

$$\boldsymbol{z}^{\mathrm{H}}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{z}=(\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{z})^{\mathrm{H}}\boldsymbol{B}_{i-1}^{-1}(\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{z})\geqslant 0.$$

又由

$$\boldsymbol{B}_i^{\frac{1}{2}}(\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})\boldsymbol{B}_i^{-\frac{1}{2}}=\boldsymbol{B}_i^{-\frac{1}{2}}(\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})\boldsymbol{B}_i^{-\frac{1}{2}},$$

知 $\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}$ 的特征值为非负实数, 从而 $\boldsymbol{I}_i+\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}$ 的特征值不为零 (实际上不小于 1). 因此 $\boldsymbol{S}_i$ 均可逆.

(2) 再证 PE 方法是收敛的, 即迭代矩阵 $\boldsymbol{M}^{-1}\boldsymbol{N}$ 的谱半径 $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})<1$. 由于

$$\begin{aligned}\boldsymbol{N}_i &= \boldsymbol{A}_i\boldsymbol{T}_{i-1}+\boldsymbol{S}_i-\boldsymbol{B}_i=\boldsymbol{A}_i\boldsymbol{S}_{i-1}^{-1}\boldsymbol{C}_{i-1}+\boldsymbol{S}_i-\boldsymbol{B}_i\\ &=\boldsymbol{A}_i(\boldsymbol{I}_{i-1}+\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_{i-1}\boldsymbol{B}_{i-2}^{-1}\boldsymbol{C}_{i-2})\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}\\ &\quad+\boldsymbol{B}_i(\boldsymbol{I}_i+\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1}-\boldsymbol{B}_i\\ &\quad(i=2,3,\cdots,m;\ \boldsymbol{A}_1=\boldsymbol{O},\ \boldsymbol{C}_0=\boldsymbol{O}).\end{aligned}$$

利用恒等式

$$(\boldsymbol{I}+\boldsymbol{G})^{-1}=\boldsymbol{I}-\boldsymbol{G}+\boldsymbol{G}^2(\boldsymbol{I}+\boldsymbol{G})^{-1},$$

改写上式第 2 项中的逆矩阵, 整理, 得

$$\begin{aligned}\boldsymbol{N}_i &= \boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_{i-1}\boldsymbol{B}_{i-2}^{-1}\boldsymbol{C}_{i-2}\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}\\ &\quad+\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}(\boldsymbol{I}+\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1}.\end{aligned}\tag{3.145}$$

由 $\boldsymbol{A}^{\mathrm{H}}=\boldsymbol{C}_{i-1}$ 知, 上式等号右端第 1 项为 Hermite 矩阵. 由定理 3.37 知 $\boldsymbol{N}_i$ 均为 Hermite 矩阵, 从而上式等号右端第 2 项也是 Hermite 矩阵. 由于

$$\begin{aligned}\boldsymbol{V}_i &= \boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_{i-1}\boldsymbol{B}_{i-2}^{-1}\boldsymbol{C}_{i-2}\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}\\ &=(\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_{i-1})\boldsymbol{B}_{i-2}^{-1}(\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_{i-1})^{\mathrm{H}},\\ \boldsymbol{W}_i &= \boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}\\ &=(\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_i^{\mathrm{H}})\boldsymbol{B}_i^{-1}(\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_i^{\mathrm{H}})^{\mathrm{H}},\\ \boldsymbol{Z}_i &= \boldsymbol{B}_i+\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}=\boldsymbol{B}_i+\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_i^{\mathrm{H}}.\end{aligned}$$

所以 $\boldsymbol{V}_i$ 和 $\boldsymbol{W}_i$ 为半正定矩阵, $\boldsymbol{Z}_i$ 为正定矩阵. 再由

$$\boldsymbol{W}_i\boldsymbol{Z}_i^{-1}=\boldsymbol{Z}_i^{\frac{1}{2}}(\boldsymbol{Z}_i^{-\frac{1}{2}}\boldsymbol{W}_i\boldsymbol{Z}_i^{-\frac{1}{2}})\boldsymbol{Z}_i^{-\frac{1}{2}},$$

知 $\boldsymbol{W}_i\boldsymbol{Z}_i^{-1}$ 的特征值为非负实数. 式 (3.145) 等号右端的第 2 项是 Hermite 矩阵, 且可写为 $(\boldsymbol{W}_i\boldsymbol{Z}_i^{-1})\boldsymbol{B}_i$, 而 $\boldsymbol{B}_i$ 正定, 根据引理 3.8 可得 $(\boldsymbol{W}_i\boldsymbol{Z}_i^{-1})\boldsymbol{B}_i$ 为半正定矩阵. 于是有

$$\begin{aligned}&\boldsymbol{N}_i=\boldsymbol{V}_i+(\boldsymbol{W}_i\boldsymbol{Z}_i^{-1})\boldsymbol{B}_i \text{ 半正定}\\ &\Longrightarrow \boldsymbol{N}=\mathrm{diag}(\boldsymbol{O},\boldsymbol{N}_2,\cdots,\boldsymbol{N}_m) \text{ 半正定}\\ &\Longrightarrow \boldsymbol{A}^{-1}\boldsymbol{N}=\boldsymbol{A}^{-\frac{1}{2}}(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{N}\boldsymbol{A}^{-\frac{1}{2}})\boldsymbol{A}^{\frac{1}{2}} \text{ 的特征值为非负实数}\\ &\Longrightarrow 0\leqslant\frac{\lambda(\boldsymbol{A}^{-1}\boldsymbol{N})}{1+\lambda(\boldsymbol{A}^{-1}\boldsymbol{N})}<1.\end{aligned}$$

由式 (3.140) 知 $\boldsymbol{M}$ 可逆, 根据引理 3.9, 得 $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})<1$. 证毕. □

在式 (3.139) 中, 改写 $\boldsymbol{S}_i$ 为

$$\boldsymbol{S}_i=\boldsymbol{B}_i(\boldsymbol{I}_i+\alpha\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})^{-1}\quad(i=2,3,\cdots,m),\tag{3.146}$$

式中: α 为参数, 相应的迭代格式 (3.143) 称为 PE(α) 方法. 可以验证: PE(0) 方法为对称块 Gauss–Seidel 方法 (简记为 SBGS 方法).

定理 3.39 设 $\boldsymbol{A}$ 为 Hermite 正定矩阵, 令 $\boldsymbol{G}_i = \boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}$, $\delta = \lambda_{\min}(\boldsymbol{G}_i)$, 若参数 α 满足

$$\begin{cases} \alpha \geqslant 0, & \text{当 } \delta \geqslant 1 \text{ 时}, \\ 0 \leqslant \alpha \leqslant \dfrac{1}{1-\delta}, & \text{当 } 0 \leqslant \delta < 1 \text{ 时}, \end{cases} \tag{3.147}$$

则 PE(α) 方法收敛.

在定理 3.39 中, $\boldsymbol{G}_i$ 的特征值为非负实数. 因此, 当 $\alpha \geqslant 0$ 时, PE(α) 方法收敛. 特别地, 当 $\alpha = 1$ 时, PE 方法收敛; 当 $\alpha = 0$ 时, SBGS 方法收敛.

例 3.7 设 $\boldsymbol{A}$ 形如式 (3.137), 其中

$$\boldsymbol{B}_i = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, \qquad \boldsymbol{A}_i = \boldsymbol{C}_i = \begin{bmatrix} -1 & & \\ & -1 & \\ & & -1 \end{bmatrix}.$$

可以验证 $\boldsymbol{A}$ 为对称正定矩阵. 取 $\boldsymbol{b} = (1, 2, \cdots, n)^{\mathrm{T}}$, 用 PE($\alpha$) 方法求解线性方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, 计算结果如表 3.7 所示 (终止准则为 10^{-12}, 时间单位为秒).

表 3.7 PE(α) 方法的数值结果

参数 α	$n = 6000$			$n = 12000$		
	迭代次数	计算时间	相对残差	迭代次数	计算时间	相对残差
0.5	22	0.8709	6.2931e-13	23	1.8432	3.3271e-13
1.0	15	0.6299	4.1516e-13	15	1.2642	7.6464e-13
1.4	8	0.3770	7.6054e-13	9	0.8406	4.6594e-14
1.5	7	0.3410	1.5256e-13	7	0.6851	2.8009e-13
1.6	9	0.4082	4.6582e-14	9	0.8277	8.5260e-14
2.0	17	0.6899	2.5846e-13	17	1.4063	4.7639e-13
SBGS	29	1.1470	8.4202e-13	30	2.4169	9.4989e-13

表 3.7 中最后一行是用 SBGS 方法求解该方程组时的迭代次数、计算时间和相对残差. 易见, 当参数 $\alpha = 1.5$ 时, PE(α) 方法的收敛速度较快.

3.6.2 二次 PE(α) 方法

在式 (3.139) 中, 改写 $\boldsymbol{S}_i$ 为

$$\boldsymbol{S}_i = \boldsymbol{B}_i(\boldsymbol{I}_i + \boldsymbol{G}_i + \boldsymbol{G}_i^2)^{-1} \quad (i = 2, 3, \cdots, m), \tag{3.148}$$

式中: $\boldsymbol{G}_i = \boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}$. 相应的迭代格式 (3.143) 称为二次 PE 方法.

定理 3.40 设 $\boldsymbol{A}$ 为 Hermite 矩阵, 则二次 PE 方法收敛.

证明 (1) 先证算法是适定的, 即所有的 $\boldsymbol{S}_i$ 是可逆的. 事实上, 由 $\boldsymbol{A}$ 正定知 $\boldsymbol{B}_i\,(i=1,2,\cdots,m)$ 正定, 故 $\boldsymbol{S}_1=\boldsymbol{B}_1$ 可逆. 而由 $\boldsymbol{A}_i^{\mathrm{H}}=\boldsymbol{C}_{i-1}$ 知 $\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}=\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_i^{\mathrm{H}}$ 半正定. 再由

$$\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}=\boldsymbol{B}_i^{-\frac{1}{2}}\cdot\boldsymbol{B}_i^{-\frac{1}{2}}(\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})\boldsymbol{B}_i^{-\frac{1}{2}}\boldsymbol{B}_i^{\frac{1}{2}},\tag{3.149}$$

得 $\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}$ 的特征值为非负实数, 从而

$$\boldsymbol{I}_i+(\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})+(\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})^2$$

的特征值为正数, 故该矩阵可逆. 根据式 (3.148), 得 $\boldsymbol{S}_i\,(i=1,2,\cdots,m)$ 可逆.

(2) 再证算法是收敛的, 即迭代矩阵的谱半径 $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})<1$. 由于

$$\begin{aligned}\boldsymbol{N}_i&=\boldsymbol{A}_i\boldsymbol{T}_{i-1}+\boldsymbol{S}_i-\boldsymbol{B}_i=\boldsymbol{A}_i\boldsymbol{S}_{i-1}^{-1}\boldsymbol{C}_{i-1}+\boldsymbol{S}_i-\boldsymbol{B}_i\\&=\boldsymbol{A}_i[\boldsymbol{I}_{i-1}+(\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_{i-1}\boldsymbol{B}_{i-2}^{-1}\boldsymbol{C}_{i-2})+(\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_{i-1}\boldsymbol{B}_{i-2}^{-1}\boldsymbol{C}_{i-2})^2]\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}\\&\quad+\boldsymbol{B}_i[\boldsymbol{I}_i+(\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})+(\boldsymbol{B}_i^{-1}\boldsymbol{A}_i\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1})^2]^{-1}-\boldsymbol{B}_i.\end{aligned}$$

利用恒等式 $(\boldsymbol{I}_i-\boldsymbol{G}_i)(\boldsymbol{I}_i+\boldsymbol{G}_i+\boldsymbol{G}_i^2)=\boldsymbol{I}_i-\boldsymbol{G}_i^3$, 得

$$(\boldsymbol{I}_i+\boldsymbol{G}_i+\boldsymbol{G}_i^2)^{-1}=\boldsymbol{I}_i-\boldsymbol{G}_i+\boldsymbol{G}_i^3(\boldsymbol{I}_i+\boldsymbol{G}_i+\boldsymbol{G}_i^2)^{-1}.$$

代入上式并整理, 得

$$\begin{aligned}\boldsymbol{N}_i&=\boldsymbol{A}_i(\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_{i-1}\boldsymbol{B}_{i-2}^{-1}\boldsymbol{C}_{i-2})\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}\\&\quad+\boldsymbol{A}_i(\boldsymbol{B}_{i-1}^{-1}\boldsymbol{A}_{i-1}\boldsymbol{B}_{i-2}^{-1}\boldsymbol{C}_{i-2})^2\boldsymbol{B}_{i-1}^{-1}\boldsymbol{C}_{i-1}+\boldsymbol{B}_i\boldsymbol{G}_i^3(\boldsymbol{I}_i+\boldsymbol{G}_i+\boldsymbol{G}_i^2)^{-1}.\end{aligned}\tag{3.150}$$

可以验证, 式 (3.150) 右端的前两项均为 Hermite 半正定矩阵. 由式 (3.149) 知, 矩阵 $\boldsymbol{G}_i^3(\boldsymbol{I}_i+\boldsymbol{G}_i+\boldsymbol{G}_i^2)^{-1}$ 的特征值为非负实数, 而 $\boldsymbol{B}_i$ 为 Hermite 正定矩阵, 且

$$\begin{aligned}\boldsymbol{B}_i\boldsymbol{G}_i^3(\boldsymbol{I}_i+\boldsymbol{G}_i+\boldsymbol{G}_i^2)^{-1}&=\boldsymbol{B}_i[(\boldsymbol{I}_i+\boldsymbol{G}_i+\boldsymbol{G}_i^2)^{-1}+\boldsymbol{G}_i-\boldsymbol{I}_i]\\&=(\boldsymbol{B}_i^{-1}+\boldsymbol{G}_i\boldsymbol{B}_i^{-1}+\boldsymbol{G}_i^2\boldsymbol{B}_i^{-1})^{-1}+\boldsymbol{B}_i\boldsymbol{G}_i-\boldsymbol{B}_i\end{aligned}$$

为 Hermite 矩阵, 由引理 3.8 知式 (3.150) 右端的第 3 项为半正定矩阵. 因此, $\boldsymbol{N}_i$ 为半正定矩阵, 从而 $\boldsymbol{N}$ 亦为半正定矩阵. 再由 $\boldsymbol{A}^{-1}\boldsymbol{N}=\boldsymbol{A}^{-\frac{1}{2}}(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{N}\boldsymbol{A}^{-\frac{1}{2}})\boldsymbol{A}^{\frac{1}{2}}$ 可知 $\lambda(\boldsymbol{A}^{-1}\boldsymbol{N})$ 为非负实数, 根据结论 (1) 可得 $\boldsymbol{M}$ 为可逆矩阵, 于是有

$$\boldsymbol{M}^{-1}\boldsymbol{N}=(\boldsymbol{N}+A)^{-1}\boldsymbol{N}=(\boldsymbol{A}^{-1}\boldsymbol{N}+\boldsymbol{I})^{-1}(\boldsymbol{A}^{-1}\boldsymbol{N}).$$

由此, 得

$$\lambda(\boldsymbol{M}^{-1}\boldsymbol{N})=\frac{\lambda(\boldsymbol{A}^{-1}\boldsymbol{N})}{1+\lambda(\boldsymbol{A}^{-1}\boldsymbol{N})}.$$

因此 $0\leqslant\lambda(\boldsymbol{M}^{-1}\boldsymbol{N})<1$, 故迭代格式 (3.143) 收敛. 证毕. □

在式 (3.148) 中, 改写 $\boldsymbol{S}_i$ 为

$$\boldsymbol{S}_i = \boldsymbol{B}_i(\boldsymbol{I}_i + \boldsymbol{G}_i + \alpha\boldsymbol{G}_i^2)^{-1} \quad (i = 2, 3, \cdots, m), \tag{3.151}$$

式中: α, β 为参数. 相应的迭代格式 (3.143) 称为二次 PE(α) 方法. 易见, PE(α) 方法和二次 PE 方法都是二次 PE(α) 方法的特殊情形. 特别地, 二次 PE(0) 方法即为 PE 方法.

定理 3.41 设 $\boldsymbol{A}$ 为 Hermite 正定矩阵, 若参数 α 满足式 (3.147), 则二次 PE(α) 方法收敛.

例 3.8 设 $\boldsymbol{A}$ 形如式 (3.137), 其中

$$\boldsymbol{B}_i = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, \qquad \boldsymbol{A}_i = \boldsymbol{C}_i = \begin{bmatrix} -1 & & \\ & -1 & \\ & & -1 \end{bmatrix}.$$

可以验证 $\boldsymbol{A}$ 为对称正定矩阵. 取 $\boldsymbol{b} = (1, 2, \cdots, n)^{\mathrm{T}}$, 用二次 PE($\alpha$) 方法求解线性方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, 计算结果如表 3.8 所示 (终止准则为 10^{-12}, 时间单位为秒).

表 3.8 二次 PE(α) 方法的数值结果

参数 α	$n = 6000$			$n = 12000$		
	迭代次数	计算时间	相对残差	迭代次数	计算时间	相对残差
0.0	15	0.6491	2.6966e-13	15	1.2793	7.6464e-13
1.0	13	0.5943	1.5076e-13	13	1.1402	4.2746e-13
2.0	10	0.4680	5.2057e-13	11	0.9823	9.7413e-14
3.0	7	0.3516	1.3442e-13	7	0.6955	3.8024e-13
4.0	8	0.3927	7.4033e-13	9	0.8440	7.2741e-14
5.0	12	0.5265	8.9187e-14	12	1.0605	2.5292e-13
SBGS	29	1.1483	8.4202e-13	30	2.3955	9.4989e-13

表 3.8 中最后一行是用 SBGS 方法求解该方程组时的迭代次数、计算时间和相对残差. 易见, 当参数 $\alpha = 3.0$ 时, 二次 PE(α) 方法的收敛速度较快.

习题 3

3.1 若将方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 的每个方程两边除以 $\boldsymbol{A}$ 的相应对角元, 再对新得到的方程组构造 Richardson 迭代格式. 试证明它就是 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 的 Jacobi 迭代格式.

3.2 设 $\boldsymbol{A} \in \mathbf{R}^{n\times n}$ 对称正定, 其最小特征值和最大特征值分别为 λ_1 和 λ_n. 证明: 迭代法

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \theta(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)})$$

收敛的充分必要条件是 $0<\theta<2/\lambda_n$.

3.3 设矩阵 $\boldsymbol{B}\in\mathbb{R}^{n\times n}$ 满足 $\rho(\boldsymbol{B})=0$. 证明: 对于任意的向量 $\boldsymbol{f}$, $\boldsymbol{x}^{(0)}\in\mathbb{R}^n$, 定常迭代格式

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}\boldsymbol{x}^{(k)}+\boldsymbol{f},\quad k=0,1,\cdots$$

最多迭代 n 次即可得到不动点方程 $\boldsymbol{x}=\boldsymbol{B}\boldsymbol{x}+\boldsymbol{f}$ 的精确解.

3.4 设对称矩阵 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$ 是非奇异的, 且 $a_{ii}>0\,(i=1,2,\cdots,n)$. 证明: 若求解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的 Gauss–Seidel 迭代法对任意初始向量 $\boldsymbol{x}^{(0)}$ 都收敛, 则 $\boldsymbol{A}$ 必是对称正定矩阵.

3.5 证明: 若方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 是严格对角占优或不可约对角占优的, 且松弛参数 $0<\omega<2$, 则 SOR 迭代法是收敛的.

3.6 若存在对称正定矩阵 $\boldsymbol{P}$, 使得

$$\boldsymbol{B}=\boldsymbol{P}-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{H}$$

为对称正定矩阵. 证明: 迭代格式

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{H}\boldsymbol{x}^{(k)}+\boldsymbol{f},\quad k=0,1,2,\cdots$$

收敛.

3.7 若存在可逆矩阵 $\boldsymbol{P}$ 使得 $\boldsymbol{P}^{-1}(\boldsymbol{I}-\boldsymbol{B})\boldsymbol{P}$ 为对称正定矩阵, 则称单步定常迭代法

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}\boldsymbol{x}^{(k)}+\boldsymbol{f},\quad k=0,1,\cdots$$

是可对角化的. 试证明: 对于可对角化的单步定常迭代格式,

(1) 其迭代矩阵 $\boldsymbol{B}$ 的所有特征值都是小于 1 的实数;

(2) 存在实数 γ, 使得其外推格式

$$\boldsymbol{x}^{(k+1)}=(1-\gamma)\boldsymbol{x}^{(k)}+\gamma(\boldsymbol{B}\boldsymbol{x}^{(k)}+\boldsymbol{f}),\quad k=0,1,\cdots$$

收敛.

3.8 对于单步定常迭代格式

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}\boldsymbol{x}^{(k)}+\boldsymbol{f},\quad k=0,1,\cdots.$$

若其迭代矩阵 $\boldsymbol{B}\in\mathbb{R}^{n\times n}$ 的特征值 $\lambda_1,\lambda_2,\lambda_3$ 满足 $|\lambda_1|<1$, $|\lambda_2|=1$, $|\lambda_3|>1$. 试问: 如何选取初始向量 $\boldsymbol{x}^{(0)}$ 使得该迭代格式是收敛的或发散的?

3.9 证明: 严格对角占优矩阵不一定是不可约对角占优矩阵, 反之亦然. 试举例加以说明.

3.10 设 Hermite 矩阵 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 有分裂 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$, 其中 $\boldsymbol{M}$ 非奇异. 试证明:

(1) $\boldsymbol{M}^{\mathrm{H}}+\boldsymbol{N}$ 也是 Hermite 矩阵;

(2) 对于任意的 $\boldsymbol{x}\in\mathbb{C}^n$, 都有

$$\boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}-\widetilde{\boldsymbol{x}}^{\mathrm{H}}\boldsymbol{A}\widetilde{\boldsymbol{x}}=(\boldsymbol{x}-\widetilde{\boldsymbol{x}})^{\mathrm{H}}(\boldsymbol{M}^{\mathrm{H}}+\boldsymbol{N})(\boldsymbol{x}-\widetilde{\boldsymbol{x}}),$$

式中: $\widetilde{\boldsymbol{x}}=\boldsymbol{M}^{-1}\boldsymbol{N}\boldsymbol{x}$;

(3) 若 $\boldsymbol{A}$ 和 $\boldsymbol{M}^{\mathrm{H}}+\boldsymbol{N}$ 均正定, 则 $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})<1$;

(4) 分别对 Jacobi 迭代法和 SOR 迭代法, 写出 $\boldsymbol{M}^{\mathrm{H}}+\boldsymbol{N}$ 的表达式.

第 4 章　线性方程组的 Krylov 子空间迭代法

本章着重介绍求解大型稀疏线性方程组的 Krylov 子空间方法. 考虑线性方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}, \tag{4.1}$$

式中: 矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 和向量 $\boldsymbol{b}\in\mathbb{R}^n$ 是已经给定的, 而 $\boldsymbol{x}\in\mathbb{R}^n$ 是待求的未知向量. 取初始向量 $\boldsymbol{x}_0\in\mathbb{R}^n$, 则解方程组 (4.1) 等价于求解

$$\boldsymbol{A}\boldsymbol{z}=\boldsymbol{r}_0,\quad \boldsymbol{x}=\boldsymbol{x}_0+\boldsymbol{z},\quad \boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0. \tag{4.2}$$

所谓求解线性方程组 (4.2) 的 Krylov 子空间方法, 就是先构造一个 Krylov 子空间

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\mathrm{span}\{\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{r}_0,\cdots,\boldsymbol{A}^{k-1}\boldsymbol{r}_0\}; \tag{4.3}$$

然后设法求一个 $\boldsymbol{z}_k\in\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 使其在某种意义下是方程组 (4.2) 解的最佳逼近, 然后得到原方程组 (4.1) 的近似解 $\boldsymbol{x}_k=\boldsymbol{x}_0+\boldsymbol{z}_k$.

因此, 具体实现这一思想的关键是如何定量地刻画“最佳逼近”. 目前最常用的最佳性标准主要有两类: 一类是残量极小化标准; 另一类是残量正交化标准.

残量极小化标准是指求 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 使得

$$\|\boldsymbol{r}_k\|_2=\min\{\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2:\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)\}, \tag{4.4}$$

式中: $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$ 为 $\boldsymbol{x}_k$ 的残差向量 (简称残量).

本章主要介绍由这一标准导出的七种方法: 求解对称正定线性方程组的共轭梯度法 (CG 方法), 求解对称不定线性方程组的极小残量法 (MINRES 方法) 和 SYMMLQ 方法, 求解非对称线性方程组的广义极小残量法 (GMRES 方法)、拟极小残量法 (QMR 方法)、LSQR 方法和广义共轭残量法等.

残量正交化标准是指求 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 使得

$$\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k\perp\mathcal{L}_k, \tag{4.5}$$

式中: $\mathcal{L}_k$ 是另一选定的 k 维子空间, 称为约束空间.

$\mathcal{L}_k$ 通常有三种典型取法: ① $\mathcal{L}_k=\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$; ② $\mathcal{L}_k=\boldsymbol{A}\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$; ③ $\mathcal{L}_k=\mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}},\boldsymbol{r}_0)$. 式 (4.5) 常称为 Galerkin 条件. 这里主要介绍这方面的三类方法: 双共轭梯度法 (BiCG 方法)、共轭梯度平方法 (CGS 方法) 和稳定化双共轭梯度法 (BICGSTAB 方法).

Krylov 子空间方法的一个最显著的特点就是整个计算过程只涉及矩阵 $\boldsymbol{A}$ 与某些向量的乘积, 这使得在算法执行过程中可以充分利用 $\boldsymbol{A}$ 的特殊性. 因此, 这种类型的线性方程组的求解方法就特别适用于大型稀疏线性方程组的求解. 当然, 对某些具有某种特殊结构的大型稠密线性方程组也是适用的.

4.1　共轭梯度法

本节介绍求解对称正定线性方程组的一类最著名的 Krylov 子空间方法— 共轭梯度法.

4.1.1 基本 CG 方法

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 和 $\boldsymbol{b}\in\mathbb{R}^n$ 已经给定, 其中 $\boldsymbol{A}$ 是对称正定的. 对于给定初始向量 $\boldsymbol{x}_0\in\mathbb{R}^n$, 求一个 $\boldsymbol{z}\in\mathbb{R}^n$ 满足式 (4.2), 便可得到方程组 (4.1) 的解 $\boldsymbol{x}=\boldsymbol{x}_0+\boldsymbol{z}$.

由于 $\boldsymbol{A}$ 是对称正定的, 所以在 $\mathbb{R}^n$ 上可以定义两种向量范数:

$$\|\boldsymbol{v}\|_{\boldsymbol{A}}=\sqrt{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}}\quad\text{和}\quad\|\boldsymbol{v}\|_{\boldsymbol{A}^{-1}}=\sqrt{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{v}},\quad\boldsymbol{v}\in\mathbb{R}^n.\tag{4.6}$$

考虑由式 (4.2) 的系数矩阵 $\boldsymbol{A}$ 和右端项 $\boldsymbol{r}_0$ 所产生的 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$. 并记式 (4.1) 的真解为 $\boldsymbol{x}^*$ 即 $\boldsymbol{x}^*=\boldsymbol{A}^{-1}\boldsymbol{b}$. 考虑求一个 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 使其在某种条件下是 $\boldsymbol{x}^*$ 的最佳逼近. 下面的定理是关于这种最佳性的几种等价表述.

定理 4.1 符号如上所述, 则下面三条等价:

(1) 向量 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 满足

$$\|\boldsymbol{x}_k-\boldsymbol{x}^*\|_{\boldsymbol{A}}=\min\{\|\boldsymbol{x}-\boldsymbol{x}^*\|_{\boldsymbol{A}}:\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)\}\ \text{(误差极小)}.\tag{4.7}$$

(2) 向量 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 满足

$$\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k\|_{\boldsymbol{A}^{-1}}=\min\{\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{A}^{-1}}:\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)\}\ \text{(残量极小)}.\tag{4.8}$$

(3) 向量 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 满足

$$\boldsymbol{z}^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k)=0\ \ (\text{即 }\boldsymbol{r}_k\perp\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)),\ \ \forall\,\boldsymbol{z}\in\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)\ \ \text{(残量正交)}.\tag{4.9}$$

注 4.1 定理 4.1 说明, 求 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 使其在 $\|\cdot\|_{\boldsymbol{A}}$ 范数下与真解 $\boldsymbol{x}^*$ 的距离最小就等价于其残量 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$ 在 $\|\cdot\|_{\boldsymbol{A}^{-1}}$ 范数下达到最小, 也等价于其残量 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$ 与 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 是正交的.

共轭梯度法基本迭代格式的推导需要利用对称 Lanczos 正交化过程. 假设已求得一个以 $\boldsymbol{v}_1=\boldsymbol{r}_0/\|\boldsymbol{r}_0\|_2$ 为初始向量的长度 k 的 Lanczos 分解 (见算法 2.13):

$$\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_k\boldsymbol{T}_k+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}},\tag{4.10}$$

式中: $\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k=\boldsymbol{I}_k$, $\boldsymbol{T}_k=\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k$ 为如下形式的对称三对角矩阵

$$\boldsymbol{T}_k=\begin{bmatrix}\alpha_1&\beta_2&&&\\\beta_2&\alpha_2&\beta_3&&\\&\ddots&\ddots&\ddots&\\&&\beta_{k-1}&\alpha_{k-1}&\beta_k\\&&&\beta_k&\alpha_k\end{bmatrix},$$

且 $\boldsymbol{v}_{k+1}^{\mathrm{T}}\boldsymbol{V}_k=\boldsymbol{0}$, $\|\boldsymbol{v}_{k+1}\|_2=1$. 这里假定初始向量 $\boldsymbol{v}_1=\boldsymbol{r}_0/\|\boldsymbol{r}_0\|_2$, $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$. 经过 Lanzcos 正交化后, 得到的 $\boldsymbol{V}_k=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k]$, 其列向量已构成了 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 的一组标准正交基, 即有

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\mathrm{span}\{\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{r}_0,\cdots,\boldsymbol{A}^{k-1}\boldsymbol{r}_0\}=\mathcal{R}(\boldsymbol{V}_k).\tag{4.11}$$

现在希望求一个 $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$ 使得式 (4.9) 成立. 由式 (4.11) 可知, 这就相当于求 $\boldsymbol{z}_k \in \mathbb{R}^k$, 使得

$$\boldsymbol{V}_k^{\mathrm{T}}(\boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{V}_k\boldsymbol{z}_k) = \boldsymbol{V}_k^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) = \boldsymbol{0}, \tag{4.12}$$

式中: $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k\boldsymbol{z}_k$.

注意到 $\boldsymbol{T}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k$ 和 $\boldsymbol{r}_0 = \|\boldsymbol{r}_0\|_2\boldsymbol{v}_1$, 由式 (4.12), 得

$$\boldsymbol{T}_k\boldsymbol{z}_k = \beta\boldsymbol{e}_1^{(k)}, \tag{4.13}$$

式中: $\beta = \|\boldsymbol{r}_0\|_2$, $\boldsymbol{e}_1^{(k)}$ 表示第 1 个分量为 1、其余分量为 0 的 k 维列向量.

注意到 $\boldsymbol{A}$ 对称正定蕴涵着 $\boldsymbol{T}_k$ 也是正定的, 便知 $\boldsymbol{T}_k$ 的 $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$ 分解存在, 设其为

$$\boldsymbol{T}_k = \boldsymbol{L}_k\boldsymbol{D}_k\boldsymbol{L}_k^{\mathrm{T}}, \tag{4.14}$$

式中: $\boldsymbol{L}_k$ 为单位下三角矩阵; $\boldsymbol{D}_k = \mathrm{diag}(\delta_1, \delta_2, \cdots, \delta_k)$, $\delta_i > 0$, $i = 1, 2, \cdots, k$. $\boldsymbol{T}_k$ 的三对角结构蕴涵着 $\boldsymbol{L}_k$ 必有如下形状:

$$\boldsymbol{L}_k = \begin{bmatrix} 1 & & & \\ \gamma_1 & 1 & & \\ & \ddots & \ddots & \\ & & \gamma_{k-1} & 1 \end{bmatrix}. \tag{4.15}$$

利用分解 (4.14), 线性方程组 (4.13) 的解可以表示为

$$\boldsymbol{z}_k = \boldsymbol{L}_k^{-\mathrm{T}}\boldsymbol{D}_k^{-1}\boldsymbol{L}_k^{-1}(\beta\boldsymbol{e}_1^{(k)}),$$

从而有

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k\boldsymbol{L}_k^{-\mathrm{T}}\boldsymbol{D}_k^{-1}\boldsymbol{L}_k^{-1}(\beta\boldsymbol{e}_1^{(k)}). \tag{4.16}$$

再令

$$\widetilde{\boldsymbol{P}}_k = [\tilde{\boldsymbol{p}}_1, \tilde{\boldsymbol{p}}_2, \cdots, \tilde{\boldsymbol{p}}_k] = \boldsymbol{V}_k\boldsymbol{L}_k^{-\mathrm{T}},$$

$$\boldsymbol{z}_k = (\zeta_1, \zeta_2, \cdots, \zeta_k)^{\mathrm{T}} = \boldsymbol{D}_k^{-1}\boldsymbol{L}_k^{-1}(\beta\boldsymbol{e}_1^{(k)}),$$

则式 (4.16) 又可写为

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \widetilde{\boldsymbol{P}}_k\boldsymbol{z}_k. \tag{4.17}$$

注意到

$$\left[\begin{array}{c|c} \boldsymbol{T}_k & \beta_k\boldsymbol{e}_k \\ \hline \beta_k\boldsymbol{e}_k^{\mathrm{T}} & \alpha_{k+1} \end{array}\right] = \boldsymbol{T}_{k+1} := \boldsymbol{L}_{k+1}\boldsymbol{D}_{k+1}\boldsymbol{L}_{k+1}^{\mathrm{T}}$$

$$= \left[\begin{array}{c|c} \boldsymbol{L}_k & \boldsymbol{0} \\ \hline \gamma_k\boldsymbol{e}_k^{\mathrm{T}} & 1 \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{D}_k & \boldsymbol{0} \\ \hline \boldsymbol{0} & \delta_{k+1} \end{array}\right] \left[\begin{array}{c|c} \boldsymbol{L}_k & \boldsymbol{0} \\ \hline \gamma_k\boldsymbol{e}_k^{\mathrm{T}} & 1 \end{array}\right]^{\mathrm{T}},$$

式中: $\boldsymbol{e}_k$ 表示第 k 个分量为 1、其余分量为 0 的 k 维列向量. 于是有

$$\begin{aligned}
\widetilde{\boldsymbol{P}}_{k+1} &= \boldsymbol{V}_{k+1}\boldsymbol{L}_{k+1}^{-\mathrm{T}} = [\boldsymbol{V}_k, \boldsymbol{v}_{k+1}] \left[\begin{array}{c:c} \boldsymbol{L}_k & \boldsymbol{0} \\ \hdashline \gamma_k \boldsymbol{e}_k^{\mathrm{T}} & 1 \end{array}\right]^{-\mathrm{T}} \\
&= [\boldsymbol{V}_k, \boldsymbol{v}_{k+1}] \left[\begin{array}{c:c} \boldsymbol{L}_k^{-\mathrm{T}} & -\gamma_k \boldsymbol{L}_k^{-\mathrm{T}} \boldsymbol{e}_k^{(k)} \\ \hdashline \boldsymbol{0} & 1 \end{array}\right] \\
&= [\boldsymbol{V}_k \boldsymbol{L}_k^{-\mathrm{T}}, \boldsymbol{v}_{k+1} - \gamma_k \boldsymbol{V}_k \boldsymbol{L}_k^{-\mathrm{T}} \boldsymbol{e}_k^{(k)}] \\
&= [\widetilde{\boldsymbol{P}}_k, \boldsymbol{v}_{k+1} - \gamma_k \widetilde{\boldsymbol{P}}_k \boldsymbol{e}_k^{(k)}] := [\widetilde{\boldsymbol{P}}_k, \widetilde{\boldsymbol{p}}_{k+1}].
\end{aligned}$$

由此可得

$$\widetilde{\boldsymbol{p}}_{k+1} = \boldsymbol{v}_{k+1} - \gamma_k \widetilde{\boldsymbol{P}}_k \boldsymbol{e}_k^{(k)} = \boldsymbol{v}_{k+1} - \gamma_k \widetilde{\boldsymbol{p}}_k, \tag{4.18}$$

$$\begin{aligned}
\boldsymbol{z}_{k+1} &= \boldsymbol{D}_{k+1}^{-1} \boldsymbol{L}_{k+1}^{-1} (\beta \boldsymbol{e}_1^{(k+1)}) \\
&= \left[\begin{array}{c:c} \boldsymbol{D}_k^{-1} & \boldsymbol{0} \\ \hdashline \boldsymbol{0} & \delta_{k+1}^{-1} \end{array}\right] \left[\begin{array}{c:c} \boldsymbol{L}_k^{-1} & \boldsymbol{0} \\ \hdashline -\gamma_k \boldsymbol{e}_k^{\mathrm{T}} \boldsymbol{L}_k^{-1} & 1 \end{array}\right] (\beta \boldsymbol{e}_1^{(k+1)}) \\
&= \left[\begin{array}{c} \boldsymbol{D}_k^{-1} \boldsymbol{L}_k^{-1} (\beta \boldsymbol{e}_1^{(k)}) \\ \hdashline \zeta_{k+1} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{z}_k \\ \hdashline \zeta_{k+1} \end{array}\right],
\end{aligned}$$

式中: $\zeta_{k+1} = -\beta \gamma_k \delta_{k+1}^{-1} \boldsymbol{e}_k^{\mathrm{T}} \boldsymbol{L}_k^{-1} \boldsymbol{e}_1^{(k)}$. 从而有

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_0 + \widetilde{\boldsymbol{P}}_{k+1} \boldsymbol{z}_{k+1} = \boldsymbol{x}_0 + [\widetilde{\boldsymbol{P}}_k, \widetilde{\boldsymbol{p}}_{k+1}] \left[\begin{array}{c} \boldsymbol{z}_k \\ \hdashline \zeta_{k+1} \end{array}\right] = \boldsymbol{x}_k + \zeta_{k+1} \widetilde{\boldsymbol{p}}_{k+1}. \tag{4.19}$$

由式 (4.19), 得

$$\begin{aligned}
\boldsymbol{r}_{k+1} &= \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k - \zeta_{k+1} \boldsymbol{A} \widetilde{\boldsymbol{p}}_{k+1} \\
&= \boldsymbol{r}_k - \zeta_{k+1} \boldsymbol{A} \widetilde{\boldsymbol{p}}_{k+1}.
\end{aligned} \tag{4.20}$$

这样, 综合上面的推导, 得如下三个基本的迭代公式:

$$\left\{\begin{array}{l}
\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \zeta_{k+1} \widetilde{\boldsymbol{p}}_{k+1}, \\
\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \zeta_{k+1} \boldsymbol{A} \widetilde{\boldsymbol{p}}_{k+1}, \\
\widetilde{\boldsymbol{p}}_{k+1} = \boldsymbol{v}_{k+1} - \gamma_k \widetilde{\boldsymbol{p}}_k.
\end{array}\right. \tag{4.21}$$

下面设法消去式 (4.21) 第 3 式中的 $\boldsymbol{v}_{k+1}$. 由 $\boldsymbol{r}_0 \in \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$, $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$ 可知, $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k \in \mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{r}_0)$, 从而 $\boldsymbol{r}_k$ 可以由 $\boldsymbol{V}_{k+1}$ 的列向量线性表出, 即

$$\boldsymbol{r}_k = \widetilde{\gamma}_1 \boldsymbol{v}_1 + \widetilde{\gamma}_2 \boldsymbol{v}_2 + \cdots + \widetilde{\gamma}_k \boldsymbol{v}_k + \widetilde{\gamma}_{k+1} \boldsymbol{v}_{k+1}.$$

又由于 $\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{r}_k=\boldsymbol{0}$, 故 $\widetilde{\gamma}_i=0,\ i=1,2,\cdots,k$, 从而

$$\boldsymbol{r}_k=\widetilde{\gamma}_{k+1}\boldsymbol{v}_{k+1}. \tag{4.22}$$

由式 (4.22) 可知 $|\widetilde{\gamma}_{k+1}|=\|\boldsymbol{r}_k\|_2$, 不妨设 $\widetilde{\gamma}_{k+1}=\|\boldsymbol{r}_k\|_2$, 并定义

$$\boldsymbol{p}_{k+1}=\|\boldsymbol{r}_k\|_2\widetilde{\boldsymbol{p}}_{k+1}, \tag{4.23}$$

则由式 (4.21) 和式 (4.22), 得

$$\begin{aligned}
\boldsymbol{x}_{k+1}&=\boldsymbol{x}_k+\frac{\zeta_{k+1}}{\|\boldsymbol{r}_k\|_2}\boldsymbol{p}_{k+1},\\
\boldsymbol{r}_{k+1}&=\boldsymbol{r}_k-\frac{\zeta_{k+1}}{\|\boldsymbol{r}_k\|_2}\boldsymbol{A}\boldsymbol{p}_{k+1},\\
\boldsymbol{p}_{k+1}&=\boldsymbol{r}_k-\frac{\|\boldsymbol{r}_k\|_2\gamma_k}{\|\boldsymbol{r}_{k-1}\|_2}\boldsymbol{p}_k.
\end{aligned}$$

令

$$\alpha_{k+1}=\frac{\zeta_{k+1}}{\|\boldsymbol{r}_k\|_2},\quad \beta_k=-\frac{\|\boldsymbol{r}_k\|_2\gamma_k}{\|\boldsymbol{r}_{k-1}\|_2},$$

即得

$$\begin{cases}
\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\alpha_{k+1}\boldsymbol{p}_{k+1},\\
\boldsymbol{r}_{k+1}=\boldsymbol{r}_k-\alpha_{k+1}\boldsymbol{A}\boldsymbol{p}_{k+1},\\
\boldsymbol{p}_{k+1}=\boldsymbol{r}_k+\beta_k\boldsymbol{p}_k.
\end{cases} \tag{4.24}$$

下面导出不涉及分解式 (4.14) 的信息计算 α_{k+1} 和 β_k 的公式. 为此, 先导出向量组 $\{\boldsymbol{p}_i\}$ 和 $\{\boldsymbol{r}_i\}$ 所具有的基本性质.

由 $\widetilde{\boldsymbol{P}}_k$ 的定义, 可以立即导出

$$\widetilde{\boldsymbol{P}}_k^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{P}}_k=\boldsymbol{L}_k^{-1}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{L}_k^{-\mathrm{T}}=\boldsymbol{D}_k.$$

这表明 $\widetilde{\boldsymbol{p}}_i$ 之间是 $\boldsymbol{A}$ 正交的 (或称为 $\boldsymbol{A}$ 共轭的), 即有

$$\widetilde{\boldsymbol{p}}_i^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{p}}_j=0,\quad i\neq j.$$

于是有

$$\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_j=0,\quad i\neq j. \tag{4.25}$$

此外, 式 (4.1.1) 的第 1 式和式 (4.23) 蕴涵着

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\mathcal{R}(\boldsymbol{V}_k)=\mathcal{R}(\widetilde{\boldsymbol{P}}_k)=\mathrm{span}\{\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_k\}, \tag{4.26}$$

即 $\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_k$ 是 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 的一组 $\boldsymbol{A}$ 正交基, 再由式 (4.22) 和 $\boldsymbol{v}_i$ 的相互正交性, 又可以导出

$$\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_j=0,\quad i\neq j. \tag{4.27}$$

即残量是相互正交的. 这样, 再由 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\mathcal{R}(\boldsymbol{V}_k)$ 便有

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\mathrm{span}\{\boldsymbol{r}_0,\boldsymbol{r}_1,\cdots,\boldsymbol{r}_{k-1}\}, \tag{4.28}$$

即 $\boldsymbol{r}_0,\boldsymbol{r}_1,\cdots,\boldsymbol{r}_{k-1}$ 构成了 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 的一组正交基.

在式 (4.24) 的第 3 式两边左乘 $\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}$ 并利用式 (4.25), 得

$$0=\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_{k+1}=\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r}_k+\beta_k\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k.$$

于是有

$$\beta_k=-\frac{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r}_k}{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k}. \tag{4.29}$$

再在式 (4.24) 的第 2 式两边左乘 $\boldsymbol{r}_k^{\mathrm{T}}$ 并利用式 (4.27), 得

$$0=\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_{k+1}=\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k-\alpha_{k+1}\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_{k+1}.$$

因此又有

$$\alpha_{k+1}=\frac{\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k}{\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_{k+1}}. \tag{4.30}$$

这样就得到了不需要分解式 (4.14) 的信息就可计算式 (4.24) 中的系数 β_k 和 α_{k+1} 的公式.

事实上, 利用式 (4.25) 和式 (4.27) 还可导出更有效的计算公式. 首先在式 (4.24) 的第 3 式两边左乘 $\boldsymbol{p}_{k+1}^{\mathrm{T}}\boldsymbol{A}$, 并利用式 (4.25), 得

$$\boldsymbol{p}_{k+1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_{k+1}=\boldsymbol{p}_{k+1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r}_k=\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_{k+1}. \tag{4.31}$$

将式 (4.24) 第 2 式中的 k 用 $k-1$ 替换, 有

$$\boldsymbol{r}_k=\boldsymbol{r}_{k-1}-\alpha_k\boldsymbol{A}\boldsymbol{p}_k.$$

将上式两边分别左乘 $\boldsymbol{r}_k^{\mathrm{T}}$ 和 $\boldsymbol{r}_{k-1}^{\mathrm{T}}$, 并利用式 (4.27), 得

$$\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k=-\alpha_k\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k, \tag{4.32}$$

$$\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1}=\alpha_k\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k=\alpha_k\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k, \tag{4.33}$$

这里式 (4.33) 的第 2 个等号利用了式 (4.31).

现将式 (4.31) 代入式 (4.30), 得

$$\alpha_{k+1}=\frac{\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k}{\boldsymbol{p}_{k+1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_{k+1}}. \tag{4.34}$$

再将式 (4.32)、式 (4.33) 与式 (4.29) 相结合, 有

$$\beta_k=\frac{\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k}{\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1}}. \tag{4.35}$$

显然, 这两个计算公式比原来的计算公式减少了内积和矩阵乘向量的次数, 因此更有效.

综合上述讨论, 可得共轭梯度法 (简称 CG 方法) 求解对称正定线性方程组的基本迭代格式如下.

选取初值 $\boldsymbol{x}_0$, 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{p}_1 = \boldsymbol{r}_0$; $k = 0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\| > \varepsilon)$

$$k = k + 1;$$

$$\alpha_k = \frac{(\boldsymbol{r}_{k-1}, \boldsymbol{r}_{k-1})}{(\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_k)};\ \ \boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \alpha_k \boldsymbol{p}_k;$$

$$\boldsymbol{r}_k = \boldsymbol{r}_{k-1} - \alpha_k \boldsymbol{A}\boldsymbol{p}_k;$$

$$\beta_k = \frac{(\boldsymbol{r}_k, \boldsymbol{r}_k)}{(\boldsymbol{r}_{k-1}, \boldsymbol{r}_{k-1})};\ \ \boldsymbol{p}_{k+1} = \boldsymbol{r}_k + \beta_k \boldsymbol{p}_k;$$

end

为了优化算法程序 (考虑计算量和存储量), 通常写成如下形式.

算法 4.1 (CG 方法) 给定对称正定方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 和 $\varepsilon > 0$. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{p}_1 = \boldsymbol{r}_0$; $\rho_0 = \boldsymbol{r}_0^{\mathrm{T}}\boldsymbol{r}_0$; $k = 0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 > \varepsilon)$

$$k = k + 1;$$

$$\boldsymbol{z}_k = \boldsymbol{A}\boldsymbol{p}_k;\ \ \alpha_k = \rho_{k-1}/\boldsymbol{z}_k^{\mathrm{T}}\boldsymbol{p}_k;$$

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \alpha_k \boldsymbol{p}_k;\ \ \boldsymbol{r}_k = \boldsymbol{r}_{k-1} - \alpha_k \boldsymbol{z}_k;$$

$$\rho_k = \boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k;\ \ \beta_k = \rho_k/\rho_{k-1};$$

$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_k + \beta_k \boldsymbol{p}_k;$$

end

算法 4.1 的迭代终止准则也可以用 $\rho_k \leqslant \varepsilon\rho_0$, 其中 $\varepsilon > 0$ 是给定的允许误差. CG 方法的 MATLAB 程序如下:

```
%共轭梯度法程序-mcg.m
function [x,iter,time,res,resvec]=mcg(A,b,x,max_it,tol)
%输入:系数阵A,右端量b,初始值x,容许误差tol,最大迭代数max_it
%输出:解向量x,迭代次数iter,CPU时间time,
%     终止时相对残差模res,相对残差模向量resvec
tic;r=b-A*x;p=r;rho=r'*r;
mr=sqrt(rho);iter=0;
while (iter<max_it)
    iter=iter+1;
    z=A*p; alpha=rho/(z'*p);
```

```
    x=x+alpha*p; r=r-alpha*z;
    rho1=r'*r; beta=rho1/rho;
    p=r+beta*p;
    res=sqrt(rho1)/mr;
    resvec(iter)=res;
    if (res<tol), break; end
    rho=rho1;
end
time=toc;
```

例 4.1 假定线性方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 和右端项 $\boldsymbol{b}$ 分别为

$$\boldsymbol{A}=\begin{bmatrix} 1 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & n \end{bmatrix}, \quad \boldsymbol{b}=\boldsymbol{A}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

显然, 此方程组的真解为 $\boldsymbol{x}^* = (1,1,\cdots,1)^{\mathrm{T}}$. 应用共轭梯度法求解该线性方程组, 取 $n=1000$, 迭代 193 步后得到的近似解 $\widehat{\boldsymbol{x}}=\boldsymbol{x}_{193}$ 满足

$$\|\widehat{\boldsymbol{x}}-\boldsymbol{x}^*\|_2 = 3.7417\times 10^{-8},$$

迭代过程的收敛轨迹如图 4.1 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差.

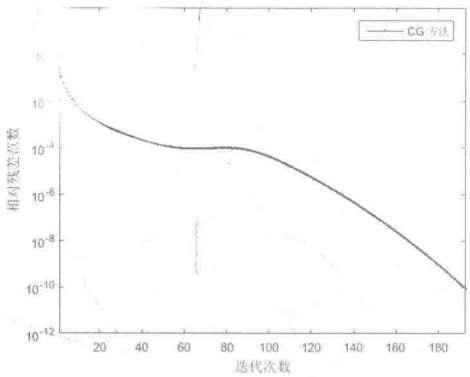

图 4.1 CG 算法的收敛特性

4.1.2 收敛性分析

在没有误差的情况下, 由式 (4.27) 可知, 必存在一个 $\ell \leqslant n$, 使得 $\boldsymbol{r}_\ell = \boldsymbol{0}$, 即有 $\boldsymbol{x}^* = \boldsymbol{x}_\ell$. 换言之, 算法在有限步可得到方程组 (4.1) 的精确解. 从这个意义上而言, 共

轭梯度法是一种直接法. 但实际使用时, 一般是将其作为一种迭代法来用的, 其原因有二: 一是在有误差的计算机上, 有限步终止的结论已经不再成立; 二是即使出现了 $\boldsymbol{r}_\ell = \boldsymbol{0}$, 但如果 ℓ 非常大, 也是难以忍受的, 因为实际应用中所考虑的问题其阶数 n 是十分巨大的.

下面介绍共轭梯度法 (作为一种迭代法) 的收敛性和收敛速度. 由于收敛性分析需要利用 Chebyshev 多项式的性质. 在第 3 章已给出 k 次 Chebyshev 多项式 $C_k(t)$ 的定义:

$$C_k(t) = \begin{cases} \cos(k \arccos t), & |t| \leqslant 1, \\ \cosh(k \operatorname{arccosh} t), & |t| > 1. \end{cases} \tag{4.36}$$

下面再回顾一下 Chebyshev 多项式的一些最常用的性质.

定理 4.2 设 $C_k(t)$ 是由式 (4.36) 定义的 Chebyshev 多项式, 则它具有如下性质:

(1) $C_k(t)$ 具有递推式

$$C_0(t) = 1;\ C_1(t) = t;\ C_{k+1}(t) = 2tC_k(t) - C_{k-1}(t),\ k = 1, 2, \cdots.$$

(2) 对于 $|t| > 1$, $C_k(t)$ 有表达式

$$C_k(t) = \frac{1}{2}\left[\left(t + \sqrt{t^2 - 1}\right)^k + \left(t + \sqrt{t^2 - 1}\right)^{-k}\right]. \tag{4.37}$$

(3) 对任意的 $t \in [-1, 1]$, 有 $|C_k(t)| \leqslant 1$, 且

$$C_k(t_i^{(k)}) = (-1)^i,\ \ t_i^{(k)} = \cos\frac{i\pi}{k},\ \ i = 0, 1, \cdots, k,$$

即 $C_k(t)$ 在 $[-1, 1]$ 上刚好有 $k+1$ 个极值点, 在这些点上交错地取 1 和 -1.

(4) 对任意的 $s > 1$, 有

$$\min_{p \in \mathcal{P}_k,\, p(s)=1} \max_{t \in [-1,1]} |p(t)| = \frac{1}{C_k(s)}, \tag{4.38}$$

式中:

$$\mathcal{P}_k = \{p : p \text{ 是次数不超过 } k \text{ 的实系数多项式}\},$$

而且上述极小极大问题有唯一解

$$p(t) = \frac{C_k(t)}{C_k(s)}. \tag{4.39}$$

证明 (1) 对于 $|t| \leqslant 1$, 令 $\theta = \arccos t$, 利用余弦函数的和差化积公式

$$\cos\alpha + \cos\beta = 2\cos\frac{\alpha+\beta}{2}\cos\frac{\alpha-\beta}{2},$$

有

$$C_{k+1}(t) + C_{k-1}(t) = \cos(k+1)\theta + \cos(k-1)\theta = 2\cos k\theta\cos\theta = 2tC_k(t).$$

对于 $|t|>1$, 令 $\theta=\operatorname{arccosh}(t)$, 利用双曲余弦函数的定义

$$\cosh(\theta)=\frac{\mathrm{e}^{\theta}+\mathrm{e}^{-\theta}}{2},$$

有

$$\begin{aligned}C_{k+1}(t)+C_{k-1}(t)&=\cosh(k+1)\theta+\cosh(k-1)\theta\\&=\frac{\mathrm{e}^{(k+1)\theta}+\mathrm{e}^{-(k+1)\theta}}{2}+\frac{\mathrm{e}^{(k-1)\theta}+\mathrm{e}^{-(k-1)\theta}}{2}\\&=\mathrm{e}^{k\theta}\cdot\frac{\mathrm{e}^{\theta}+\mathrm{e}^{-\theta}}{2}+\mathrm{e}^{-k\theta}\cdot\frac{\mathrm{e}^{\theta}+\mathrm{e}^{-\theta}}{2}\\&=2\cdot\frac{\mathrm{e}^{\theta}+\mathrm{e}^{-\theta}}{2}\cdot\frac{\mathrm{e}^{k\theta}+\mathrm{e}^{-k\theta}}{2}\\&=2\cosh\theta\cosh(k\theta)=2tC_k(t).\end{aligned}$$

(2) 对于 $|t|>1$, 令 $\theta=\operatorname{arccosh}(t)$, 有 $t=\cosh(\theta)=\mathrm{e}^{\theta}+\mathrm{e}^{-\theta}/2$, 即

$$\left(\mathrm{e}^{\theta}\right)^2-2t\mathrm{e}^{\theta}+1=0.$$

得

$$\mathrm{e}^{\theta}=\frac{2t+\sqrt{4t^2-4}}{2}=t+\sqrt{t^2-1}.$$

于是, 有

$$\begin{aligned}C_k(t)=\cosh(k\theta)&=\frac{1}{2}\left(\mathrm{e}^{k\theta}+\mathrm{e}^{-k\theta}\right)=\frac{1}{2}\left[\left(\mathrm{e}^{\theta}\right)^k+\left(\mathrm{e}^{\theta}\right)^{-k}\right]\\&=\frac{1}{2}\left[\left(t+\sqrt{t^2-1}\right)^k+\left(t+\sqrt{t^2-1}\right)^{-k}\right].\end{aligned}$$

(3) $\forall t\in[-1,1]$, $|C_k(t)|\leqslant 1$ 成立是显然的. 直接验证, 得

$$C_k(t_i^{(k)})=\cos(k\arccos t_i^{(k)})=\cos(i\pi)=(-1)^i.$$

(4) 注意到 $s>1$, $p(t)$ 是 k 次多项式且 $p(s)=1$. 记

$$\alpha=1/C_k(s)>0,\quad p_{\max}=\max_{-1\leqslant t\leqslant 1}|p(t)|.$$

若 $p_{\max}<\alpha$, 则

$$\begin{aligned}(-1)^i\left(\alpha C_k(t_i^{(k)})-p(t_i^{(k)})\right)&=(-1)^i\alpha(-1)^i-(-1)^ip(t_i^{(k)})\\&=\alpha-(-1)^ip(t_i^{(k)})\geqslant\alpha-p_{\max}>0.\end{aligned}$$

这表明多项式 $\psi=\alpha C_k(t)-p(t)$ 在 $t_i^{(k)}$ $(i=0,1,\cdots,k)$ 处有交替的正负号. 所以它在 $[-1,1]$ 中有 k 个零点. 但 $s\notin[-1,1]$ 也是 k 次多项式 $\alpha C_k(t)-p(t)$ 的零点, 从而 k 次多项式 $\alpha C_k(t)-p(t)$ 有 $k+1$ 个零点, 矛盾. 故必有

$$\max_{-1\leqslant t\leqslant 1}|p(t)|\geqslant\alpha=\frac{1}{C_k(s)}.$$

若取 $p(t)=\dfrac{C_k(t)}{C_k(s)}$, 则

$$\max_{-1\leqslant t\leqslant 1}|p(t)|=\max_{-1\leqslant t\leqslant 1}\left|\frac{C_k(t)}{C_k(s)}\right|=\frac{\max\limits_{-1\leqslant t\leqslant 1}|C_k(t)|}{C_k(s)}=\frac{1}{C_k(s)}.$$

由此可得

$$\min_{p\in\mathcal{P}_k,\,p(s)=1}\ \max_{t\in[-1,1]}|p(t)|=\frac{1}{C_k(s)}.$$

证毕. □

推论 4.1 设 $a<b<1$(或 $1<a<b$). 若 $p\in\mathcal{P}_k$ 且 $p(1)=1$. 则

$$\min_{p\in\mathcal{P}_k,\,p(1)=1}\ \max_{t\in[a,b]}|p(t)|=\frac{1}{C_k(w(1))},\tag{4.40}$$

式中:

$$w(t)=\begin{cases}\dfrac{2t-a-b}{b-a}, & a<b<1,\\[2ex] \dfrac{2t-a-b}{a-b}, & 1<a<b.\end{cases}$$

且上述极小极大问题的唯一解为

$$p(t)=\frac{C_k(w(t))}{C_k(w(1))}.$$

现在考虑 CG 方法的收敛性分析. 由定理 4.1 和前面的推导过程可知, 共轭梯度法第 k 步得到的近似解 $\boldsymbol{x}_k$ 满足

$$\|\boldsymbol{x}_k-\boldsymbol{x}^*\|_{\boldsymbol{A}}=\min\left\{\|\boldsymbol{x}-\boldsymbol{x}^*\|_{\boldsymbol{A}}:\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)\right\}.$$

由 Krylov 子空间的性质可知, $\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 的充分必要条件是存在 $\varphi\in\mathcal{P}_{k-1}$ 使得 $\boldsymbol{x}=\boldsymbol{x}_0+\varphi(\boldsymbol{A})\boldsymbol{r}_0$, 这里 $\mathcal{P}_{k-1}$ 表示次数不超过 $k-1$ 的多项式的全体. 于是, 对任意的 $\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 有

$$\boldsymbol{x}^*-\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{b}-\boldsymbol{x}_0-\varphi(\boldsymbol{A})\boldsymbol{r}_0=\boldsymbol{A}^{-1}\psi(\boldsymbol{A})\boldsymbol{r}_0,$$

式中: $\psi(t)=1-t\varphi(t)$.

设 $\boldsymbol{A}$ 的谱分解为 $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}$, 其中 $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n]$ 是正交矩阵, $\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$, $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n>0$. 现将 $\boldsymbol{r}_0$ 按 $\boldsymbol{A}$ 的特征向量展开, 有

$$\boldsymbol{r}_0=\eta_1\boldsymbol{u}_1+\eta_2\boldsymbol{u}_2+\cdots+\eta_n\boldsymbol{u}_n=\sum_{i=1}^{n}\eta_i\boldsymbol{u}_i.$$

这样, 便有

$$\|\boldsymbol{x}^*-\boldsymbol{x}\|_{\boldsymbol{A}}^2=\|\boldsymbol{A}^{-1}\psi(\boldsymbol{A})\boldsymbol{r}_0\|_{\boldsymbol{A}}^2=\left(\boldsymbol{A}^{-1}\psi(\boldsymbol{A})\boldsymbol{r}_0,\psi(\boldsymbol{A})\boldsymbol{r}_0\right)$$

$$
\begin{aligned}
&= \left(\sum_{i=1}^{n} \eta_i \boldsymbol{A}^{-1} \psi(\boldsymbol{A}) \boldsymbol{u}_i, \sum_{j=1}^{n} \eta_j \psi(\boldsymbol{A}) \boldsymbol{u}_j \right) \\
&= \left(\sum_{i=1}^{n} \eta_i \psi(\lambda_i) \lambda_i^{-1} \boldsymbol{u}_i, \sum_{j=1}^{n} \eta_j \psi(\lambda_j) \boldsymbol{u}_j \right) \\
&= \sum_{i=1}^{n} \eta_i^2 \psi^2(\lambda_i) \lambda_i^{-1} \leqslant \max_{1 \leqslant i \leqslant n} \psi^2(\lambda_i) \sum_{i=1}^{n} \lambda_i^{-1} \eta_i^2 \\
&= \max_{1 \leqslant i \leqslant n} \psi^2(\lambda_i) \|\boldsymbol{x}^* - \boldsymbol{x}_0\|_{\boldsymbol{A}}^2,
\end{aligned}
$$

从而有

$$
\frac{\|\boldsymbol{x}^* - \boldsymbol{x}\|_{\boldsymbol{A}}}{\|\boldsymbol{x}^* - \boldsymbol{x}_0\|_{\boldsymbol{A}}} \leqslant \min_{\psi \in \mathcal{P}_k^0} \max_{1 \leqslant i \leqslant n} |\psi(\lambda_i)|, \tag{4.41}
$$

式中: $\mathcal{P}_k^0 = \{\psi \in \mathcal{P}_k : \psi(0) = 1\}$. 令

$$
a = \min_{1 \leqslant i \leqslant n} \lambda_i = \lambda_n, \quad b = \max_{1 \leqslant i \leqslant n} \lambda_i = \lambda_1,
$$

则由推论 4.1, 知极小极大问题

$$
\min_{\psi \in \mathcal{P}_k^0} \max_{\lambda \in [a,b]} |\psi(\lambda)|
$$

有唯一的解

$$
\widehat{\psi}(\lambda) = \frac{C_k\Big(\dfrac{b+a-2\lambda}{b-a}\Big)}{C_k\Big(\dfrac{b+a}{b-a}\Big)},
$$

而且有

$$
\min_{\psi \in \mathcal{P}_k^0} \max_{\lambda \in [a,b]} |\psi(\lambda)| = \frac{1}{C_k\Big(\dfrac{b+a}{b-a}\Big)}, \tag{4.42}
$$

这里假设了 $1 < a < b$. 将式 (4.41) 和式 (4.42) 相结合, 可证明如下结果.

定理 4.3 共轭梯度法产生的近似解 $\boldsymbol{x}_k$ 满足

$$
\frac{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{A}}}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_{\boldsymbol{A}}} \leqslant \frac{1}{C_k\Big(1 + \dfrac{2}{\kappa - 1}\Big)}, \tag{4.43}
$$

式中: C_k 为 k 阶 Chebyshev 多项式, $\kappa = \kappa_2(\boldsymbol{A}) = \|\boldsymbol{A}\|_2 \cdot \|\boldsymbol{A}^{-1}\|_2 = b/a$.

注 4.2 由定理 4.2 的结论 (2), 得

$$
\left[C_k\left(1 + \frac{2}{\kappa - 1}\right) \right]^{-1} = \left[C_k\left(\frac{b+a}{b-a}\right) \right]^{-1} = \frac{2\sigma^k}{1 + \sigma^{2k}}, \tag{4.44}
$$

式中: $\sigma = (\sqrt{b} - \sqrt{a})/(\sqrt{b} + \sqrt{a})$. 由式 (4.44) 可证

$$
\left[C_k\left(1 + \frac{2}{\kappa - 1}\right) \right]^{-1} \leqslant 2\left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k. \tag{4.45}
$$

由此可知, 系数矩阵的条件数 κ 越小, 共轭梯度法收敛就越快.

4.1.3 预处理 CG 方法

由式 (4.43), 在实际使用共轭梯度法时, 通常需对方程组 (4.1) 作预处理, 使其系数矩阵的谱相对集中, 这就是所谓的预处理共轭梯度法 (PCG 方法). 简单来说, 预处理共轭梯度法就是预先选择一个适当的对称正定矩阵 $\boldsymbol{M}$, 使得矩阵 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 的谱相对集中. 设 $\boldsymbol{M}$ 有分解 $\boldsymbol{M}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$, 则方程组 (4.1) 经过预处理后变为

$$(\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}})\widetilde{\boldsymbol{x}}=\boldsymbol{L}^{-1}\boldsymbol{b},\quad \boldsymbol{x}=\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{x}}. \tag{4.46}$$

该方程组的系数矩阵 $\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}$ 仍为对称正定矩阵. 通过 $\boldsymbol{M}$ 和 $\boldsymbol{L}$ 的选取, 可以使得 $\widetilde{\boldsymbol{A}}=\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}$ 比 $\boldsymbol{A}$ 有更好的条件数. 所以预处理后的方程组 (4.46) 比原方程组更容易计算. 现在对方程组 (4.46) 使用共轭梯度法:

取 $\widetilde{\boldsymbol{x}}_0$, 令 $\widetilde{\boldsymbol{r}}_0=\boldsymbol{L}^{-1}\boldsymbol{b}-(\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}})\widetilde{\boldsymbol{x}}_0$, $\widetilde{\boldsymbol{p}}_1=\widetilde{\boldsymbol{r}}_0$, 对 $k=1,2,\cdots$, 计算出

$$\widetilde{\alpha}_k=\frac{\widetilde{\boldsymbol{r}}_{k-1}^{\mathrm{T}}\widetilde{\boldsymbol{r}}_{k-1}}{\widetilde{\boldsymbol{p}}_k^{\mathrm{T}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{p}}_k},\quad \widetilde{\boldsymbol{x}}_k=\widetilde{\boldsymbol{x}}_{k-1}+\widetilde{\alpha}_k\widetilde{\boldsymbol{p}}_k,\quad \widetilde{\boldsymbol{r}}_k=\widetilde{\boldsymbol{r}}_{k-1}-\widetilde{\alpha}_k\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{p}}_k,$$

$$\widetilde{\beta}_k=\frac{\widetilde{\boldsymbol{r}}_k^{\mathrm{T}}\widetilde{\boldsymbol{r}}_k}{\widetilde{\boldsymbol{r}}_{k-1}^{\mathrm{T}}\widetilde{\boldsymbol{r}}_{k-1}},\quad \widetilde{\boldsymbol{p}}_{k+1}=\widetilde{\boldsymbol{r}}_k+\widetilde{\beta}_k\widetilde{\boldsymbol{p}}_k.$$

为了在公式中使用原来变量, 定义 $\boldsymbol{x}_k=\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{x}}_k$, 则 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k=\boldsymbol{L}\widetilde{\boldsymbol{r}}_k$. 再令 $\boldsymbol{p}_k=\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{p}}_k$, $\boldsymbol{z}_k=\boldsymbol{M}^{-1}\boldsymbol{r}_k$, 就得到了预处理的共轭梯度法, 具体步骤如下:

步 1, 给定 $\boldsymbol{x}_0\in\mathbb{R}^n$. 计算 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$, $\boldsymbol{z}_0=\boldsymbol{M}^{-1}\boldsymbol{r}_0$, $\boldsymbol{p}_1=\boldsymbol{z}_0$.

步 2, 对 $k=1,2,\cdots$, 计算

$$\alpha_k=\frac{\boldsymbol{z}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1}}{\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k},\ \boldsymbol{x}_k=\boldsymbol{x}_{k-1}+\alpha_k\boldsymbol{p}_k,$$

$$\boldsymbol{r}_k=\boldsymbol{r}_{k-1}-\alpha_k\boldsymbol{A}\boldsymbol{p}_k,\ \boldsymbol{z}_k=\boldsymbol{M}^{-1}\boldsymbol{r}_k, \tag{4.47}$$

$$\beta_k=\frac{\boldsymbol{z}_k^{\mathrm{T}}\boldsymbol{r}_k}{\boldsymbol{z}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1}},\ \boldsymbol{p}_{k+1}=\boldsymbol{z}_k+\beta_k\boldsymbol{p}_k. \tag{4.48}$$

为了优化算法程序, 写成如下形式.

算法 4.2 (PCG 方法) 给定对称正定方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 预处理子 $\boldsymbol{M}$ 和 $\varepsilon>0$. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2\leqslant\varepsilon$, 其中 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{z}_0=\boldsymbol{M}^{-1}\boldsymbol{r}_0$;

$\boldsymbol{p}_1=\boldsymbol{z}_0$; $\rho_0=\boldsymbol{z}_0^{\mathrm{T}}\boldsymbol{r}_0$; $k=0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2>\varepsilon)$

$k=k+1$;

$\boldsymbol{u}_k=\boldsymbol{A}\boldsymbol{p}_k$; $\alpha_k=\rho_{k-1}/\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{p}_k$;

$\boldsymbol{x}_k=\boldsymbol{x}_{k-1}+\alpha_k\boldsymbol{p}_k$; $\boldsymbol{r}_k=\boldsymbol{r}_{k-1}-\alpha_k\boldsymbol{u}_k$;

$\boldsymbol{z}_k = \boldsymbol{M}^{-1}\boldsymbol{r}_k;\ \rho_k = \boldsymbol{z}_k^{\mathrm{T}}\boldsymbol{r}_k;\ \beta_k = \rho_k/\rho_{k-1};$

$\boldsymbol{p}_{k+1} = \boldsymbol{z}_k + \beta_k\boldsymbol{p}_k;$

end

算法 4.2 仅与预处理矩阵 $\boldsymbol{M}$ 有关, 而与 $\boldsymbol{L}$ 无关. 当 $\boldsymbol{M}$ 为单位阵时, 它就是没有经过预处理的共轭梯度法. 与共轭梯度法相比较, 预处理共轭梯度法仅增加了式 (4.47) 的工作量, 即每一步需要求一个以预处理矩阵 $\boldsymbol{M}$ 为系数矩阵的方程组得到 $\boldsymbol{z}_k$. 算法 4.2 每一步的主要工作量是作一次矩阵 $\boldsymbol{A}$ 与向量的乘法操作和一次 $\boldsymbol{M}^{-1}$ 与向量的乘法操作.

下面讨论预处理矩阵 $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 的选取. $\boldsymbol{M}$ 取法之一是 $\boldsymbol{A}$ 的不完全 Cholesky 分解, 即 $\boldsymbol{A} = \widehat{\boldsymbol{L}}\widehat{\boldsymbol{L}}^{\mathrm{T}} + \boldsymbol{R}$, 然后取 $\boldsymbol{M} := \boldsymbol{L}\widehat{\boldsymbol{L}}^{\mathrm{T}}$, 其中 $\boldsymbol{R} = \boldsymbol{A} - \boldsymbol{M}$ 满足某种稀疏模式, 使得它比较小.

如果对 $\boldsymbol{A}$ 进行分裂, 即 $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$, 则需要判断古典迭代格式中的 $\boldsymbol{M}$ 是否满足要求, 即是否是对称正定阵. 对于 Jacobi 迭代, $\boldsymbol{M}$ 为 $\boldsymbol{A}$ 的对角元所组成的对角矩阵, 它显然是对称正定的, 因此可以作为预处理矩阵. 这样的预处理称为 Jacobi 预处理. 对于 Gauss–Seidel 和 SOR 迭代, 其分裂格式中的 $\boldsymbol{M}$ 都是下三角矩阵, 不是对称正定阵, 因此不能作为预处理矩阵. 而对于 SSOR 迭代, 其分裂格式中的 $\boldsymbol{M}$ 是对称正定的, 因此可以作为预处理矩阵, 称其为 SSOR 预处理矩阵.

给出预处理共轭梯度法的 MATLAB 程序如下:

```
%预处理共轭梯度法程序-pcg.m
function [x,iter,time,res,resvec]=pcg(A,b,x,M,max_it,tol)
%输入:系数矩阵A,右端向量b,初始向量x,预处理子M,
%     容许误差tol,最大迭代次数max_it
%输出:解向量x,迭代次数iter,CPU时间time,
%     终止时相对残差模res,相对残差模向量resvec
tic; r=b-A*x; z=M\r; p=z;
rho=z'*r; mr=norm(r); iter=0;
while (iter<max_it)
    iter=iter+1;
    u=A*p; alpha=rho/(p'*u);
    x=x+alpha*p; r=r-alpha*u;
    z=M\r; rho1=z'*r;
    beta=rho1/rho; p=z+beta*p;
    res=norm(r)/mr;
    resvec(iter)=res;
    if (res<tol), break; end
    rho=rho1;
end
time=toc;
```

为了对预处理的效果有一个更直观更深刻的印象, 下面再给出一个数值算例.

例 4.2 考虑例 4.1 中的线性方程组, 如果取预处理矩阵 $\boldsymbol{M}$ 为 $\boldsymbol{A}$ 的对角元构成的对角矩阵, 即 $\boldsymbol{M}=\mathrm{diag}(1,2,\cdots,n)$. 实验中取 $n=1000$, 则预处理共轭梯度法在 12 步后得到的近似解 $\widehat{\boldsymbol{x}}=\boldsymbol{x}_{12}$ 就满足 $\|\widehat{\boldsymbol{x}}-\boldsymbol{x}^*\|_2=3.7305\times10^{-9}$. 迭代的收敛轨迹如图 4.2 所示, 其中横坐标表示迭代步数 k, 纵坐标表示相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差.

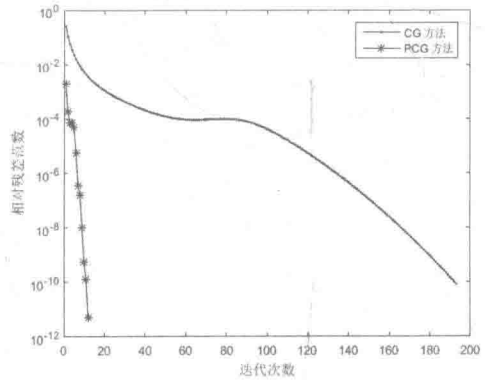

图 4.2 PCG 法和 CG 法的收敛特性

与例 4.1 的数值结果相比较可知, 虽然这里的预处理矩阵 $\boldsymbol{M}$ 选得非常简单, 但其加速效果是显著的. 为什么会这样呢? 因为预处理后矩阵 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 的特征值分布很集中, 故其数值效果非常好.

4.1.4 CGNR 方法和 CGNE 方法

1. CGNR 方法

任何对称不定或非对称非奇异的方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 都可转化为一个对称正定的方程组 (即法方程)

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ax}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}. \tag{4.49}$$

对方程组 (4.49) 应用 CG 法, 就导出了下面的关于法方程残差的共轭梯度法 (CGNR).

算法 4.3 (CGNR 方法) 给定系数矩阵 $\boldsymbol{A}$, 右端向量 $\boldsymbol{b}$, 初始向量 $\boldsymbol{x}_0$ 和容许误差限 $\varepsilon>0$. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2\leqslant\varepsilon$, 其中 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{Ax}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{Ax}_0$; $\boldsymbol{p}_0=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0$; $k=0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|>\varepsilon)$

$$\alpha_k=\frac{(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_k,\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_k)}{(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_k\boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_k)};$$

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\alpha_k\boldsymbol{p}_k;\quad \boldsymbol{r}_{k+1}=\boldsymbol{r}_k-\alpha_k\boldsymbol{Ap}_k;$$

$$\beta_k = \frac{(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_{k+1}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_{k+1})}{(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_k, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_k)};$$

$$\boldsymbol{p}_{k+1} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k;$$

$$k = k + 1;$$

end

算法 4.4 每次迭代需要计算三次矩阵与向量的乘法: $\boldsymbol{A}\boldsymbol{p}_k$, $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_k$ 和 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_{k+1}$. 注意到, CGNR 方法是极小化误差

$$\|\boldsymbol{e}_k\|_{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}} = \|\boldsymbol{x}_k - \boldsymbol{x}^*\|_{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}},$$

式中: $\boldsymbol{x}^*$ 为精确解, 满足 $\boldsymbol{x}^* = \boldsymbol{A}^{-1}\boldsymbol{b}$; $\|\boldsymbol{e}_k\|_{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}}$ 为仿射空间

$$\boldsymbol{x}_0 + \mathrm{span}\{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0, (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0, \cdots, (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{k-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0\} := \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0)$$

上关于残量 $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ 的 2–范数. 因此, 可以比照 CG 方法的收敛性分析技巧对 CGNR 方法进行相应的收敛性分析.

2. CGNE 方法

对称不定或非对称非奇异的方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 还可转化为如下形式的对称正定的方程组, 即

$$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{b},\ \boldsymbol{x} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}. \tag{4.50}$$

在 "$\boldsymbol{y}$" 空间中, 对方程组 (4.50) 应用 CG 法, 得到如下形式:

选取初值 $\boldsymbol{y}_0$, 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}_0$; $\boldsymbol{p}_0 = \boldsymbol{r}_0$; $k = 0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\| > \varepsilon)$

$$\alpha_k = \frac{(\boldsymbol{r}_k, \boldsymbol{r}_k)}{(\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_k)} = \frac{(\boldsymbol{r}_k, \boldsymbol{r}_k)}{(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_k, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_k)};$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \alpha_k \boldsymbol{p}_k;\ \ \boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_k;$$

$$\beta_k = \frac{(\boldsymbol{r}_{k+1}, \boldsymbol{r}_{k+1})}{(\boldsymbol{r}_k, \boldsymbol{r}_k)};\ \ \boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k;$$

$$k = k + 1;$$

end

作变量替换 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}_k \to \boldsymbol{x}_k$, $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_k \to \boldsymbol{p}_k$, 再化简上述有关式子则得到下面的关于法方程误差的共轭梯度法 (CGNE).

算法 4.4 (CGNE 方法) *给定初始向量 $\boldsymbol{x}_0$ 和容许误差限 $\varepsilon > 0$. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.*

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{p}_0 = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0$; $k = 0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\| > \varepsilon)$

$$\alpha_k = \frac{(\boldsymbol{r}_k, \boldsymbol{r}_k)}{(\boldsymbol{p}_k, \boldsymbol{p}_k)};\ \ \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k;$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k;\ \ \beta_k = \frac{(\boldsymbol{r}_{k+1}, \boldsymbol{r}_{k+1})}{(\boldsymbol{r}_k, \boldsymbol{r}_k)};$$

$$\boldsymbol{p}_{k+1} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k;$$

$$k = k + 1;$$

end

算法 4.4 每次迭代需要计算两次矩阵与向量的乘法: $\boldsymbol{A}\boldsymbol{p}_k$ 和 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_{k+1}$. 此外, CGNE 方法是极小化误差

$$\|\widetilde{\boldsymbol{e}}_k\|_{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}} = \|\boldsymbol{y}_k - \boldsymbol{y}^*\|_{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}},$$

式中: $\boldsymbol{y}^*$ 满足 $\boldsymbol{y}^* = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{b}$; $\|\widetilde{\boldsymbol{e}}_k\|_{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}}$ 为仿射空间

$$\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathrm{span}\{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0, (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0, \cdots, (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{k-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0\} \equiv \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0)$$

上的 2–范数 $\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_2$.

一般来说, CGNR 和 CGNE 这两种方法的困难或者说缺点在于矩阵的条件数变成了原来的平方, 然而, 在有些情况下它们是很有效的. 可参看文献 [25] 中的论述.

例 4.3 考虑方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, 其中

$$\boldsymbol{A} = \begin{bmatrix} 12 & 3 & 2 & & & \\ -3 & 12 & 3 & 2 & & \\ -2 & -3 & 12 & 3 & 2 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & -2 & -3 & 12 & 3 \\ & & & -2 & -3 & 12 \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad \boldsymbol{b} = \boldsymbol{A}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ \\ 1 \\ 1 \end{bmatrix} \in \mathbb{R}^n.$$

取 $n = 1000$, 分别将 CGNR 和 CGNE 方法应用到该线性方程组上, 两个迭代法均在 10 步后收敛 ($\varepsilon = 10^{-10}$), 计算得到的近似解 $\widehat{\boldsymbol{x}}$ 和 $\widetilde{\boldsymbol{x}}$ 与真解 $\boldsymbol{x}^*$ 之间的绝对值误差分别为

$$\|\widehat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 3.4705 \times 10^{-10}, \quad \|\widetilde{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 3.4515 \times 10^{-10}.$$

残量分别为

$$\|\boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}}\|_2 = 4.5764 \times 10^{-9}, \quad \|\boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}\|_2 = 4.6018 \times 10^{-9}.$$

迭代过程的收敛轨迹如图 4.3 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

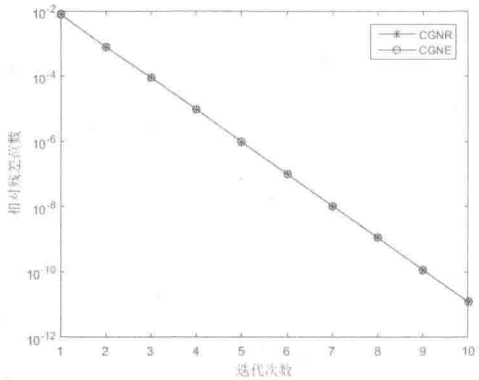

图 4.3 CGNR 和 CGNE 方法的收敛特性

4.2 广义极小残量法

广义极小残量法是目前求解大型稀疏非对称线性方程组的最常用方法之一, 在文献中常称为 GMRES 方法 (Generalized Minimal RESidual Method). 本节主要介绍这一算法的详细计算步骤及其收敛性理论.

4.2.1 GMRES 方法

考虑如下的线性方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{4.51}$$

式中: 矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 和向量 $\boldsymbol{b} \in \mathbb{R}^n$ 是给定的, 而 $\boldsymbol{x} \in \mathbb{R}^n$ 是待求的未知向量. 这里假定系数矩阵 $\boldsymbol{A}$ 是非奇异的大型稀疏矩阵, 而且 $\boldsymbol{A}^{\mathrm{T}} \neq \boldsymbol{A}$. 广义极小残量法是求 $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$, 使得

$$\|\boldsymbol{r}_k\|_2 = \min\{\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2 : \boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)\}, \tag{4.52}$$

式中: $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$, 即求 $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$, 使得残量 $\boldsymbol{r}_k$ 的 2–范数达到最小.

由于现在 $\boldsymbol{A}$ 是非对称的, 所以只能借助 $\boldsymbol{A}$ 的 Arnoldi 正交化过程极小化问题 (4.52). 回顾一下 Arnoldi 正交化过程 (见算法 2.11):

选取初始向量 $\boldsymbol{v}_1$, 使得 $\|\boldsymbol{v}_1\|_2 = 1$;

for $j = 1 : k$

for $i = 1 : j$

$$h_{ij} = (\boldsymbol{A}\boldsymbol{v}_j, \boldsymbol{v}_i);$$

$$\widetilde{\boldsymbol{v}}_{j+1} = \boldsymbol{A}\boldsymbol{v}_j - \sum_{i=1}^{j} h_{ij}\boldsymbol{v}_i; \tag{4.53}$$

$$h_{j+1,j} = \|\widetilde{\boldsymbol{v}}_{j+1}\|_2; \ \ \boldsymbol{v}_{j+1} = \widetilde{\boldsymbol{v}}_{j+1}/h_{j+1,j};$$

end

end

不难发现, 式 (4.53) 可以写成

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}} = \boldsymbol{V}_{k+1}\widetilde{\boldsymbol{H}}_{k+1,k}, \tag{4.54}$$

式中: $\boldsymbol{V}_{k+1} = [\boldsymbol{V}_k, \boldsymbol{v}_{k+1}] \in \mathbb{R}^{n\times(k+1)}$ 满足 $\boldsymbol{V}_{k+1}^{\mathrm{T}}\boldsymbol{V}_{k+1} = \boldsymbol{I}_{k+1}$; 矩阵

$$\widetilde{\boldsymbol{H}}_{k+1,k} = \begin{bmatrix} \boldsymbol{H}_k \\ \beta_k\boldsymbol{e}_k^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k}$$

为上 Hessenberg 矩阵, 其中 $\beta_k = h_{k+1,k}$,

$$\boldsymbol{H}_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2k} \\ & h_{32} & \cdots & h_{3k} \\ & \ddots & & \vdots \\ & & h_{k,k-1} & h_{k,k} \end{bmatrix}.$$

注意到 $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0) = \mathcal{R}(\boldsymbol{V}_k)$, 而且对任意的 $\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{V}_k\boldsymbol{z} \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$, 有

$$\begin{aligned} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2 &= \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{A}\boldsymbol{V}_k\boldsymbol{z}\|_2 = \|\boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{V}_k\boldsymbol{z}\|_2 \\ &= \|\beta\boldsymbol{V}_{k+1}\boldsymbol{e}_1^{(k+1)} - \boldsymbol{V}_{k+1}\widetilde{\boldsymbol{H}}_{k+1,k}\boldsymbol{z}\|_2 \\ &= \|\beta\boldsymbol{e}_1^{(k+1)} - \widetilde{\boldsymbol{H}}_{k+1,k}\boldsymbol{z}\|_2, \end{aligned}$$

式中: $\beta = \|\boldsymbol{r}_0\|_2$; $\boldsymbol{e}_1^{(k+1)}$ 表示第 1 个分量为 1、其余分量均为 0 的 $k+1$ 维列向量. 由此可知, 极小化问题 (4.52) 等价于求 $\boldsymbol{z}_k \in \mathbb{R}^k$, 使得

$$\|\beta\boldsymbol{e}_1^{(k+1)} - \widetilde{\boldsymbol{H}}_{k+1,k}\boldsymbol{z}_k\|_2 = \min\{\|\beta\boldsymbol{e}_1^{(k+1)} - \widetilde{\boldsymbol{H}}_{k+1,k}\boldsymbol{z}\|_2 : \boldsymbol{z} \in \mathbb{R}^k\}. \tag{4.55}$$

一旦这样的 $\boldsymbol{z}_k$ 已经求得, 则所需的 $\boldsymbol{x}_k$ 就为 $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k\boldsymbol{z}_k$.

最小二乘问题 (4.55) 也可以用 $\widetilde{\boldsymbol{H}}_{k+1,k}$的 QR 分解来求解. 由于 $\widetilde{\boldsymbol{H}}_{k+1,k}$ 是上 Hessenberg 矩阵, 可以计算 k 个 Givens 变换

$$\boldsymbol{G}_i = \mathrm{diag}\left(\boldsymbol{I}_{i-1}, \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix}, \boldsymbol{I}_{k-i}\right), \quad c_i^2 + s_i^2 = 1,$$

使得

$$(\boldsymbol{G}_k\boldsymbol{G}_{k-1}\cdots\boldsymbol{G}_2\boldsymbol{G}_1)\widetilde{\boldsymbol{H}}_{k+1,k} = \begin{bmatrix} \boldsymbol{R}_k \\ \boldsymbol{0} \end{bmatrix}, \tag{4.56}$$

式中: $\boldsymbol{R}_k$ 为非奇异的上三角矩阵 (因为 $\widetilde{\boldsymbol{H}}_{k+1,k}$ 的次对角元均不为零). 令

$$\boldsymbol{G} = \boldsymbol{G}_k\boldsymbol{G}_{k-1}\cdots\boldsymbol{G}_2\boldsymbol{G}_1, \quad \begin{bmatrix} \boldsymbol{t}_k \\ \rho_k \end{bmatrix} = \boldsymbol{G}(\beta\boldsymbol{e}_1), \quad \boldsymbol{t}_k = (\tau_1, \tau_2, \cdots, \tau_k)^{\mathrm{T}}, \tag{4.57}$$

则 $\boldsymbol{G}$ 是 $k+1$ 阶正交矩阵, 而且直接计算, 有

$$\begin{cases} \tau_1 = \beta c_1, \\ \tau_i = (-1)^{i-1}\beta s_1 s_2 \cdots s_{i-1} c_i, \quad i = 2, 3, \cdots, k, \\ \rho_k = (-1)^k \beta s_1 s_2 \cdots s_k. \end{cases} \tag{4.58}$$

由此立即可得最小二乘问题 (4.55) 的解为

$$\boldsymbol{z}_k = \boldsymbol{R}_k^{-1} \boldsymbol{t}_k. \tag{4.59}$$

此外, 此时的残量范数为

$$\|\boldsymbol{r}_k\|_2 = \|\beta \boldsymbol{e}_1^{(k+1)} - \widetilde{\boldsymbol{H}}_{k+1,k} \boldsymbol{z}_k\|_2 = |\rho_k|. \tag{4.60}$$

综合上面的讨论, 可以将广义极小残量法 (GMRES) 的步骤总结如下:

(1) 令 $\boldsymbol{v}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|_2$, 产生一个长度为 k 的 Arnoldi 分解 (4.54).

(2) 利用 Givens 变换求 $\widetilde{\boldsymbol{H}}_{k+1,k}$ 的 QR 分解 (4.56), 并按式 (4.58) 求得向量 $\boldsymbol{t}_k$ 和数 ρ_k.

(3) 用回代法求解上三角方程组 $\boldsymbol{R}_k \boldsymbol{z}_k = \boldsymbol{t}_k$, 得 $\boldsymbol{z}_k$.

(4) 计算 $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{z}_k$.

(5) 若 $|\rho_k|/\beta < \varepsilon$ (事先给定的误差界), 则终止; 否则增加 k 的值, 重复上面的过程.

上述 GMRES 方法可写成如下便于编程的算法形式.

算法 4.5 (GMRES 方法) 给定矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 向量 $\boldsymbol{b} \in \mathbb{R}^n$ 和允许误差 $\varepsilon > 0$, 取初始向量 $\boldsymbol{x}_0$. 本算法计算 $\boldsymbol{x}_k \in \mathbb{R}^n$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

步 1, 计算初始残量 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$, $\beta = \|\boldsymbol{r}_0\|_2$ 和初始正交化向量 $\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$, $\boldsymbol{\xi} = \boldsymbol{e}_1 = (1, 0, \cdots, 0)^{\mathrm{T}}$. 置 $k := 1$.

步 2, 用 Arnoldi 过程计算 $\boldsymbol{v}_{k+1}$ 和 $h_{i,k}\,(i = 1, 2, \cdots, k+1)$. 将 Givens 变换 $\boldsymbol{G}_i$ 作用于矩阵 $\widetilde{\boldsymbol{H}}_{i+1,i}$ 的最后一列:

$$\begin{bmatrix} h_{i,k} \\ h_{i+1,k} \end{bmatrix} := \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} \begin{bmatrix} h_{i,k} \\ h_{i+1,k} \end{bmatrix}, \quad i = 1, \cdots, k-1.$$

步 3, 计算第 k 次 Givens 变换 $\boldsymbol{G}_k$ 中的 c_k 和 s_k:

$$c_k = \frac{h_{k,k}}{\sqrt{h_{k,k}^2 + h_{k+1,k}^2}}, \quad s_k = \frac{h_{k+1,k}}{\sqrt{h_{k,k}^2 + h_{k+1,k}^2}}, \tag{4.61}$$

$$\left(\tau_k = \frac{h_{k+1,k}}{h_{k,k}}, \quad c_k = \frac{1}{\sqrt{1+\tau_k^2}}, \quad s_k = c_k \tau_k\right).$$

步 4, 对 $k+1$ 维向量 $\boldsymbol{\xi}$ 的最后两个元素和矩阵 $\widetilde{\boldsymbol{H}}_{k+1,k}$ 分别作用第 k 次 Givens 变换 $\boldsymbol{G}_k$, 有

$$\begin{bmatrix} \xi_k \\ \xi_{k+1} \end{bmatrix} := \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix} \begin{bmatrix} \xi_k \\ 0 \end{bmatrix} = \begin{bmatrix} c_k\xi_k \\ -s_k\xi_k \end{bmatrix},$$

$$\begin{bmatrix} h_{k,k} \\ h_{k+1,k} \end{bmatrix} := \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix} \begin{bmatrix} h_{k,k} \\ h_{k+1,k} \end{bmatrix} = \begin{bmatrix} c_k h_{k,k} + s_k h_{k+1,k} \\ 0 \end{bmatrix}. \tag{4.62}$$

步 5, 若 $|\rho_k|/\beta = |\xi_{k+1}|/\beta \leqslant \varepsilon$, 求解关于 $\boldsymbol{z}_k$ 的上三角方程组 $\boldsymbol{H}_{k,k}\boldsymbol{z}_k = [\beta\boldsymbol{\xi}]_{k\times 1}$, 计算近似解向量 $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k\boldsymbol{z}_k$, 停算; 否则, 置 $k := k+1$, 转步 2.

算法 4.5 的 MATLAB 程序如下:

```
%GMRES方法程序-mgmres.m
function [x,k,time,res,resvec,flag]=mgmres(A,b,x,max_it,tol)
tic; flag=0; r=b-A*x; beta=norm(r);  %计算残差
n=length(b); e1=zeros(n,1); e1(1)=1.0;
res=norm(r)/beta; resvec(1)=res;
V(:,1)=r/beta; xi=beta*e1; k=0;
while (k<=max_it)
    k=k+1; w=A*V(:,k);
    for i=1:k  %修正Arnoldi过程
        H(i,k)=w'*V(:,i);
        w=w-H(i,k)*V(:,i);
    end
    H(k+1,k)=norm(w);
    if abs(H(k+1,k))/beta<tol,
        return;
    else
        V(:,k+1)=w/H(k+1,k);
    end
    for i=1:k-1,
        temp=c(i)*H(i,k)+s(i)*H(i+1,k);
        H(i+1,k)=-s(i)*H(i,k)+c(i)*H(i+1,k);
        H(i,k)=temp;
    end
    [c(k),s(k),H(k,k)]=givens(H(k,k), H(k+1,k));%第k次Givens变换
    xi(k+1)=-s(k)*xi(k);
    xi(k)=c(k)*xi(k); H(k+1,k)=0.0;
    res=abs(xi(k+1))/beta;
    resvec(k+1)=res;
```

```
    if (res<=tol ),
        y=H(1:k,1:k)\xi(1:k); x=x+V(:,1:k)*y;
        break;  %跳出循环
    end
end
if (res>tol), flag=1; end;  %不收敛
time=toc;
```

例 4.4 考虑系数矩阵 $\boldsymbol{A}$ 和右端项 $\boldsymbol{b}$ 分别为

$$\boldsymbol{A}=\begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & n \\ 1 & 2 & -1 & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & 1 & n-1 & -1 \\ -n & 0 & \cdots & 0 & 1 & n \end{bmatrix}, \quad \boldsymbol{b}=\boldsymbol{A}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

的线性方程组 (4.51). 显然, 该方程组的真解为 $\boldsymbol{x}^*=(1,1,\cdots,1)^{\mathrm{T}}$. 取 $n=10^3$, 应用 GMRES 算法到该方程组上, 迭代 172 步后得到的近似解 $\widetilde{\boldsymbol{x}}=\boldsymbol{x}_{172}$ 满足

$$\|\widetilde{\boldsymbol{x}}-\boldsymbol{x}_*\|_2=1.1427\times 10^{-7}.$$

迭代过程的收敛轨迹如图 4.4 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

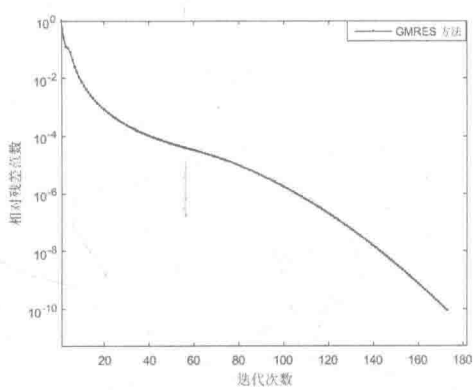

图 4.4 GMRES 方法的收敛特性

实际使用 GMRES 方法时的主要问题是 k 不能太大, 这是因为这一方法的内存需求量为 $O(kn)$, 而计算机的内存又是有限的, 这样就会出现近似解 $\boldsymbol{x}_k$ 还不满足精

度要求, 而 k 已经不能再增加的情形. 解决这一问题的一个简单而行之有效的办法就是重开始技术. 它的基本思想是, 先选定一个不太大的正整数 m, 用 GMRES 方法产生 $\boldsymbol{x}_m$, 然后再以 $\boldsymbol{x}_m$ 作为初始向量重新开始. 这就是重开始 GMRES 方法, 通常记为 GMRES(m) 方法, 具体计算过程可简述如下.

算法 4.6 (GMRES(m) 方法) 给定矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 向量 $\boldsymbol{b}\in\mathbb{R}^n$, 正整数 m, 初始向量 $\boldsymbol{x}_0$ 和允许误差 $\varepsilon>0$. 本算法计算 $\boldsymbol{x}_m\in\mathbb{R}^n$, 使得 $\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_m\|_2/\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0\|_2\leqslant\varepsilon$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$; $\beta=\|\boldsymbol{r}_0\|_2$; $\rho_m:=\beta$; $\boldsymbol{x}_m:=\boldsymbol{x}_0$;

while $(\rho_m/\beta>\varepsilon)$

$\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_m$; $\beta_0=\|\boldsymbol{r}_0\|_2$; $\boldsymbol{v}_1=\boldsymbol{r}_0/\beta_0$;

以 $\boldsymbol{v}_1$ 为初始向量产生一个长度为 m 的 Arnoldi 分解

$$\boldsymbol{A}\boldsymbol{V}_m=\boldsymbol{V}_{m+1}\widetilde{\boldsymbol{H}}_{m+1,m};$$

计算 $\widetilde{\boldsymbol{H}}_{m+1,m}$ 的 QR 分解: $\widetilde{\boldsymbol{H}}_{m+1,m}=\boldsymbol{G}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{R}_m\\ \boldsymbol{0}\end{bmatrix}$;

按式 (4.58) 计算 $\boldsymbol{t}_m$ 和 ρ_m;

求解 $\boldsymbol{R}_m\boldsymbol{z}_m=\boldsymbol{t}_m$ 得到 $\boldsymbol{z}_m$;

计算 $\boldsymbol{x}_m=\boldsymbol{x}_0+\boldsymbol{V}_m\boldsymbol{z}_m$;

end

这一算法是目前求解大型稀疏非对称线性方程组的常用方法之一. 至于算法 4.6 中的 m 取多大为好, 现在还没有理论上的结果. 在理论上仅可以保证, 在系数矩阵 $\boldsymbol{A}$ 有正定的对称部分时, 对任意的 m, 算法 4.6 总是收敛的 (见后面的收敛性定理). 但对于一般情形, 并非对任意的 m 总能保证其收敛. 例如, 对线性方程组

$$\boldsymbol{A}\boldsymbol{x}:=\begin{bmatrix}0&1\\-1&0\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}1\\1\end{bmatrix}:=\boldsymbol{b},$$

利用 GMRES(1) 求解, 不论循环多少次, 总有 $\boldsymbol{x}_k=\boldsymbol{0}$, 而 $\|\boldsymbol{A}\boldsymbol{x}_k-\boldsymbol{b}\|_2=\sqrt{2}$, 永远不会得到原方程组之真解满足要求的近似解. 但如果用 GMRES 求解, 则只需两步就可以得到该方程组的精确解.

GMRES(m) 的 MATLAB 程序如下:

```
%重开始GMRES方法-GMRES(m)-gmresm.m
function [x,out,int,time,res,resvec,flag]=gmresm(A,b,x,m,max_it,tol)
tic; flag=0; int=0; r= b-A*x;  %计算残差
beta=norm(r); res=norm(r)/beta; resvec(1)=res;
n=length(b); %m=restrt;
V(1:n,1:m+1)=zeros(n,m+1);
H(1:m+1,1:m)=zeros(m+1,m);
e1=zeros(n,1); e1(1)=1.0;
```

```
c(1:m)=zeros(m,1); s(1:m)=zeros(m,1);
for k=1:max_it
    V(:,1)=r/norm(r);
    xi=norm(r)*e1;
    for j=1:m  %用Arnoldi方法构造正交基
        w=A*V(:,j);
        for i=1:j
            H(i,j)=w'*V(:,i);
            w=w-H(i,j)*V(:,i);
        end
        H(j+1,j)=norm(w);
        if abs(H(j+1,j))/beta<tol,
            return;
        else
            V(:,j+1)=w/H(j+1,j);
        end
        for i=1:j-1  %第i次Givens变换
            temp=c(i)*H(i,j)+s(i)*H(i+1,j);
            H(i+1,j)=-s(i)*H(i,j)+c(i)*H(i+1,j);
            H(i,j)=temp;
        end
        [c(j),s(j),H(j,j)]=givens(H(j,j),H(j+1,j));%第j次Givens变换
        xi(j+1)=-s(j)*xi(j);xi(j)=c(j)*xi(j);
        H(j+1,j)=0.0;res=abs(xi(j+1))/beta;
        resvec((k-1)*m+j+1)=res;
        if(res<=tol)
            y=H(1:j,1:j)\xi(1:j);
            x=x+V(:,1:j)*y;
            break;  %跳出内循环
        end
    end
    if (res<tol )
        out=k; int=j; break; %跳出外循环
    end
    y=H(1:m,1:m)\xi(1:m);
    x=x+V(:,1:m)*y;
    r=b-A*x ;
end
if (res>tol), flag=1; end;  %不收敛
time=toc;
```

例 4.5 用 GMRES(m) 方法求解例 4.4 的线性方程组. 表 4.1 列出了对于不同的 m, 达到收敛 (容许误差为 $\varepsilon = 10^{-10}$) 所需要的外迭代次数、内迭代次数、总迭代次数和 CPU 时间 (s). 从表 4.1 中的数据可看出, 对于本例而言, 与 GMRES 方法相比, 对

表 4.1 GMRES(m) 方法对 m 的依赖性

m	外迭代次数	内迭代次数	总迭代次数	CPU 时间	相对残差
10	47	3	463	0.2621	9.8273e-11
20	14	12	272	0.1488	9.1166e-11
30	9	8	248	0.1327	9.3534e-11
40	6	27	227	0.1267	9.4923e-11
50	5	19	219	0.1244	9.9472e-11
60	4	26	206	0.1203	9.9062e-11
GMRES	1	172	172	0.1651	8.8473e-11

于比较小的 m 值, GMRES(m) 方法并不占优势. 事实上, 前面已经指出, 只有在无法使用 GMRES 方法时 (如超出计算机的内存), 才考虑使用 GMRES(m) 方法, 并且 m 不宜取得太小.

此外, 对于不同的 m 值, 迭代过程的收敛轨迹如图 4.5 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

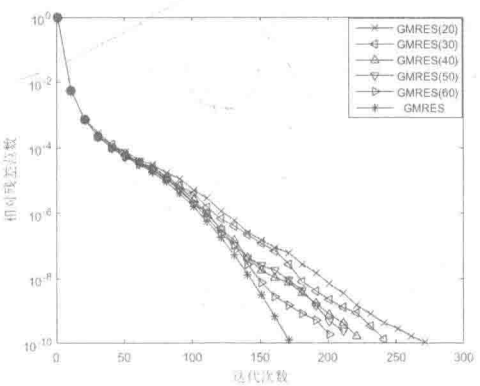

图 4.5 GMRES 方法的收敛特性

4.2.2 预处理 GMRES 方法

回顾共轭梯度法求解方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 时预处理技术的重要性, 考虑预处理广义极小残量法 (简记为 PGMRES 方法). 因为共轭梯度法适用于对称正定矩阵 $\boldsymbol{A}$, 所以预处理矩阵 $\boldsymbol{M}$ 需要保持对称. 对于非对称矩阵 $\boldsymbol{A}$, 预处理矩阵 $\boldsymbol{M}$ 就没有必要是对称的. 此时方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 被预处理为

$$\boldsymbol{M}^{-1}\boldsymbol{Ax} = \boldsymbol{M}^{-1}\boldsymbol{b}. \tag{4.63}$$

这里 $\boldsymbol{M}$ 应该取为 $\boldsymbol{A}$ 的一个近似矩阵, 以保证用 GMRES 方法求解预处理方程组 (4.63) 比求解原方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 有更快的收敛速度.

由于 GMRES 方法只涉及到矩阵与向量的乘法, 所以要保证 $\boldsymbol{M}^{-1}$ 与某一个向量 $\boldsymbol{r}$ 的乘法 $\boldsymbol{z}=\boldsymbol{M}^{-1}\boldsymbol{r}$ 容易计算, 即 $\boldsymbol{Mz}=\boldsymbol{r}$ 容易求解. 这启发选取 $\boldsymbol{M}$ 为 (块) 对角矩阵或 (块) 三角形矩阵, 如 Jacobi 预处理矩阵

$$\boldsymbol{M}=\boldsymbol{D}=\operatorname{diag}\{a_{11},\cdots,a_{nn}\},$$

Gauss–Seidel 预处理矩阵

$$\boldsymbol{M}=\boldsymbol{D}-\boldsymbol{L}$$

以及 SOR 预处理矩阵

$$\boldsymbol{M}=\frac{1}{\omega}(\boldsymbol{D}-\omega\boldsymbol{L}),$$

式中:

$$\boldsymbol{L}=-\begin{bmatrix} 0 & & & \\ a_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix}.$$

对方程组 (4.63) 用 GMRES 方法, 得到如下 PGMRES 方法.

算法 4.7 (PGMRES 方法) 给定矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 向量 $\boldsymbol{b}\in\mathbb{R}^n$, 预处理子 $\boldsymbol{M}$, 初始向量 $\boldsymbol{x}_0$ 和允许误差 $\varepsilon>0$. 本算法计算 $\boldsymbol{x}_k\in\mathbb{R}^n$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2\leqslant\varepsilon$, 其中 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{Ax}_k$.

$\boldsymbol{r}_0=\boldsymbol{M}^{-1}(\boldsymbol{b}-\boldsymbol{Ax}_0)$; $\beta=\|\boldsymbol{r}_0\|_2$;

for $k=1,2,\cdots$

对 $\mathcal{K}_k(\boldsymbol{M}^{-1}\boldsymbol{A},\boldsymbol{v}_1)$ 用 Arnoldi 分解计算 $\boldsymbol{V}_k=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k]$ 以及 $(k+1)\times k$ 阶矩阵 $\widetilde{\boldsymbol{H}}_{k+1,k}$.

计算 $\widetilde{\boldsymbol{H}}_{k+1,k}$ 的 QR 分解: $\widetilde{\boldsymbol{H}}_{k+1,k}=\boldsymbol{G}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{R}_k\\ \boldsymbol{0}\end{bmatrix}$;

按式 (4.58) 计算 $\boldsymbol{t}_k$ 和 ρ_k;

if $|\rho_k|/\beta\leqslant\varepsilon$

求解 $\boldsymbol{R}_k\boldsymbol{z}_k=\boldsymbol{t}_k$ 得到 $\boldsymbol{z}_k$;

计算 $\boldsymbol{x}_k=\boldsymbol{x}_0+\boldsymbol{V}_k\boldsymbol{z}_k$; 停算.

end

end

对方程组 (4.63) 用 GMRES(m) 得到如下算法:

算法 4.8 (PGMRES(m) 方法) 给定矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 向量 $\boldsymbol{b}\in\mathbb{R}^n$, 正整数 m, 预处理子 $\boldsymbol{M}$ 和允许误差 $\varepsilon>0$, 选取初始向量 $\boldsymbol{x}_0\in\mathbb{R}^n$. 本算法计算 $\boldsymbol{x}_k\in\mathbb{R}^n$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2\leqslant\varepsilon$, 其中 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$.

步 1, 计算残量 $\boldsymbol{r}_0=\boldsymbol{M}^{-1}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0)$, $\beta=\|\boldsymbol{r}_0\|_2$ 和初始正交化向量 $\boldsymbol{v}_1=\boldsymbol{r}_0/\beta$.

步 2, 对 $\mathcal{K}_m(\boldsymbol{M}^{-1}\boldsymbol{A},\boldsymbol{v}_1)$ 用 Arnoldi 分解计算 $\boldsymbol{V}_m=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m]$ 以及 $(m+1)\times m$ 阶矩阵 $\widetilde{\boldsymbol{H}}_{m+1,m}$.

步 3, 计算满足极小化问题

$$\min\left\{\|\beta\boldsymbol{e}_1^{(m+1)}-\widetilde{\boldsymbol{H}}_{m+1,m}\boldsymbol{z}\|_2:\boldsymbol{z}\in\mathbb{R}^m\right\}$$

的最小二乘解 $\boldsymbol{z}_m$, 令 $\boldsymbol{x}_m=\boldsymbol{x}_0+\boldsymbol{V}_m\boldsymbol{z}_m$. 若 $\boldsymbol{x}_m$ 达到精度要求, 停算.

步 4, 置 $\boldsymbol{x}_0:=\boldsymbol{x}_m$, 转步 1.

下面通过数值例子, 观察 PGMRES 方法的效果.

例 4.6 仍考虑例 4.4 的线性方程组. 取预处理矩阵 $\boldsymbol{M}=\mathrm{diag}(\boldsymbol{A})$, 应用 PGMRES 方法到该方程组上, 迭代 12 步后得到的近似解 $\widetilde{\boldsymbol{x}}=\boldsymbol{x}_{12}$ 满足

$$\|\widetilde{\boldsymbol{x}}-\boldsymbol{x}_*\|_2=1.7407\times10^{-8}.$$

迭代过程的收敛轨迹如图 4.6 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

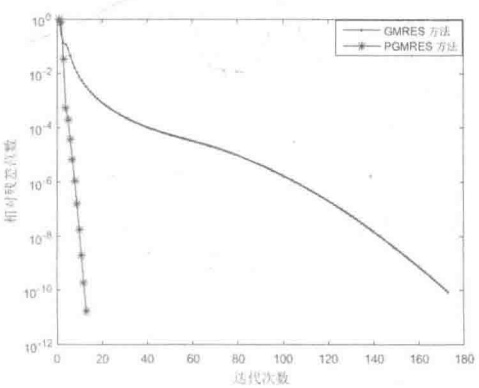

图 4.6 GMRES 和 PGMRES 的收敛特性

例 4.7 用 PGMRES(m) 方法求解例 4.4 的线性方程组, 取预处理矩阵 $\boldsymbol{M}=\mathrm{diag}(\boldsymbol{A})$. 表 4.2 列出了对于不同的 m, 达到收敛 (容许误差为 $\varepsilon=10^{-10}$) 所需要的外迭代次数、内迭代次数、总迭代次数和 CPU 时间 (s).

此外, 对于不同的 m 值, 迭代过程的收敛轨迹如图 4.7 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

表 4.2 PGMRES(m) 方法对 m 的依赖性

m	外迭代次数	内迭代次数	总迭代次数	CPU 时间	相对残差
3	21	3	63	0.1692	7.3458e-11
6	4	3	21	0.0449	3.8001e-11
9	2	5	14	0.0290	8.0750e-11
12	1	12	12	0.0238	1.7551e-11
PGMRES	1	12	12	0.0394	1.7551e-11

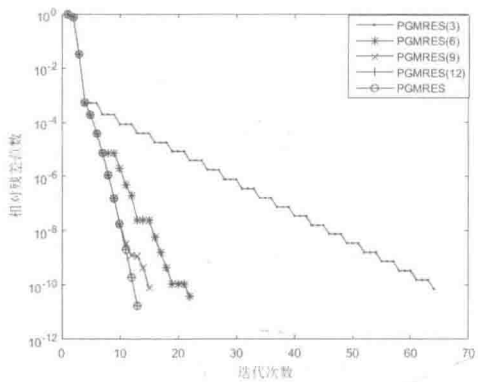

图 4.7 PGMRES(m) 方法对 m 的依赖性

4.2.3 收敛性分析

前一节 CG 方法的收敛性都是关于对称矩阵的. 当系数矩阵非对称时, 对 Krylov 子空间方法 (比如 GMRES 方法) 的收敛性进行分析就会遇到很大的困难. 首先来看 GMRES 方法的一个重要性质.

定理 4.4 GMRES 方法不会发生中断.

证明 若 $h_{k+1,k} \neq 0$, 则计算过程直至第 k 步都不会中断. 事实上, 当 $h_{k+1,k} \neq 0$ 时, 由式 (4.61) 和式 (4.62), $\boldsymbol{R}_k$ 的对角元满足

$$r_{k,k} = h_{k,k} := c_k h_{k,k} + s_k h_{k+1,k} = \sqrt{h_{k,k}^2 + h_{k+1,k}^2} > 0.$$

故正交化可进行, 极小化问题可解.

由此可知, 只有当 $h_{k+1,k} = 0$ 时, 计算过程才在第 k 步中断, 此时向量 $\boldsymbol{v}_{k+1}$ 不能构造. 但此时成立

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k \boldsymbol{H}_k,$$

故 $\sigma(\boldsymbol{H}_k) \subset \sigma(\boldsymbol{A})$. 由于 $\boldsymbol{A}$ 非奇异, 故 $\boldsymbol{H}_k$ 也非奇异. 在第 k 步, 极小化问题 (4.55) 变为

$$\|\beta \boldsymbol{v}_1 - \boldsymbol{A}\boldsymbol{V}_k \boldsymbol{z}\|_2 = \|\beta \boldsymbol{v}_1 - \boldsymbol{V}_k \boldsymbol{H}_k \boldsymbol{z}\|_2 = \left\|\boldsymbol{V}_k\left(\beta \boldsymbol{e}_1^{(k)} - \boldsymbol{H}_k \boldsymbol{z}\right)\right\|_2$$

$$= \|\beta \boldsymbol{e}_1^{(k)} - \boldsymbol{H}_k \boldsymbol{z}\|_2.$$

而 $\boldsymbol{H}_k$ 非奇异, 当 $\boldsymbol{z}_k = \boldsymbol{H}_k^{-1}(\beta \boldsymbol{e}_1^{(k)})$ 时, $\|\beta \boldsymbol{v}_1 - \boldsymbol{A}\boldsymbol{V}_k \boldsymbol{z}\|_2$ 达到极小值 0, 即 $\|\boldsymbol{r}_k\|_2 = 0$, 故

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{z}_k$$

是精确解, 换言之, GMRES 方法若在第 k 步中断, 则在这一步已得到精确解. 证毕. □

下面的定理是 GMRES 方法的另一个性质.

定理 4.5 *设 $\{\boldsymbol{x}_i\}$ 是 GMRES 方法产生的迭代序列. 若 $\boldsymbol{x}_k$ 是精确解, 而 $\boldsymbol{x}_i\,(i < k)$ 不是精确解, 则算法在第 k 步中断.*

证明 由假设, 有

$$\boldsymbol{r}_i \neq \boldsymbol{0}, \quad i = 1, 2, \cdots, k-1, \quad \text{而 } \boldsymbol{r}_k = \boldsymbol{0}.$$

但 $\|\boldsymbol{r}_k\|_2$ 是 $\widetilde{\boldsymbol{\xi}}_k = \beta \boldsymbol{V}_k \boldsymbol{e}_1^{(k+1)}$ 最后一个分量的绝对值, 即它是

$$\begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & c_k & s_k \\ & & & -s_k & c_k \end{bmatrix} \begin{bmatrix} & \\ & \widetilde{\boldsymbol{\xi}}_{k-1} \\ & \\ & \\ 0 & \end{bmatrix}$$

的第 $k+1$ 个分量的绝对值, 即

$$\|\boldsymbol{r}_k\|_2 = |s_k \boldsymbol{e}_k^{\mathrm{T}} \widetilde{\boldsymbol{\xi}}_{k-1}|. \tag{4.64}$$

但

$$|\boldsymbol{e}_k^{\mathrm{T}} \widetilde{\boldsymbol{\xi}}_{k-1}| = \|\boldsymbol{r}_{k-1}\|_2 \neq 0, \tag{4.65}$$

故 $s_k = 0$. 由于

$$s_k = \frac{h_{k+1,k}}{\sqrt{h_{k,k}^2 + h_{k+1,k}^2}},$$

故有 $h_{k+1,k} = 0$, 这样 $\widetilde{\boldsymbol{v}}_{k+1} = \boldsymbol{0}$ (由式 (4.53) 所定义), 算法中断. 证毕. □

由式 (4.64) 和式 (4.65) 容易得到下面的推论.

推论 4.2 *GMRES 方法的残量范数 $\|\boldsymbol{r}_k\|_2$ 有表达式*

$$\|\boldsymbol{r}_k\|_2 = \left(\prod_{i=1}^{k} |s_i| \right) \|\boldsymbol{r}_0\|_2. \tag{4.66}$$

定理 4.6 *初始残量 $\boldsymbol{r}_0$ 的最小多项式次数是 k 的充分必要条件是 $\widetilde{\boldsymbol{v}}_{k+1} = \boldsymbol{0}$ 且 $\widetilde{\boldsymbol{v}}_i \neq \boldsymbol{0}\,(i = 1, 2, \cdots, k)$, 其中 $\widetilde{\boldsymbol{v}}_{i+1}\,(i = 1, 2, \cdots, k)$ 由式 (4.53) 所定义.*

证明 若 $\boldsymbol{r}_0 = \|\boldsymbol{r}_0\|_2 \boldsymbol{v}_1$ 的最小多项式次数为 k, 则存在一个 k 次多项式 ϕ_k, 使得 $\phi_k(\boldsymbol{A})\boldsymbol{v}_1 = \boldsymbol{0}$, 且 ϕ_k 是次数最低者. 因此

$$\mathcal{K}_{k+1} = \mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{A}\boldsymbol{v}_1, \cdots, \boldsymbol{A}^k\boldsymbol{v}_1\} = \mathcal{K}_k.$$

由于 $\widetilde{\boldsymbol{v}}_{k+1} \in \mathcal{K}_{k+1} = \mathcal{K}_k$, 且 $\widetilde{\boldsymbol{v}}_{k+1} \perp \mathcal{K}_k$, 故 $\widetilde{\boldsymbol{v}}_{k+1} = \boldsymbol{0}$. 另外, 若存在某个 $i(1 \leqslant i \leqslant k)$, 使 $\widetilde{\boldsymbol{v}}_i = \boldsymbol{0}$, 则存在一个 $i-1$ 次多项式 ϕ_{i-1} 使 $\phi_{i-1}(\boldsymbol{A})\boldsymbol{v}_1 = \boldsymbol{0}$, 这与 $\boldsymbol{v}_1$ 的最小多项式次数为 k 矛盾, 故 $\widetilde{\boldsymbol{v}}_i \neq \boldsymbol{0}$, $1 \leqslant i \leqslant k$.

反之, 设 $\widetilde{\boldsymbol{v}}_{k+1} = \boldsymbol{0}$ 且 $\widetilde{\boldsymbol{v}}_i \neq \boldsymbol{0}$, $i = 1, 2, \cdots, k$. 由 $\widetilde{\boldsymbol{v}}_{k+1} = \boldsymbol{0}$ 可知, 存在一个 k 次多项式 ϕ_k, 使 $\phi_k(\boldsymbol{A})\boldsymbol{v}_1 = \boldsymbol{0}$, 而且 k 是次数最小者. 否则, 当存在 $\phi_i\,(i < k)$ 使 $\phi_i(\boldsymbol{A})\boldsymbol{v}_1 = \boldsymbol{0}$ 时, 就有 $\mathcal{K}_{i+1} = \mathcal{K}_i$, 因此 $\widetilde{\boldsymbol{v}}_{i+1} = \boldsymbol{0}$, 这与 $\widetilde{\boldsymbol{v}}_i \neq \boldsymbol{0}\,(1 \leqslant i \leqslant k)$ 矛盾. 证毕. □

进一步, 由上述三个定理可以推出:

推论 4.3 *GMRES 方法在第 k 步产生的解 $\boldsymbol{x}_k$ 是精确解的充分必要条件为下列诸等价条件:*

(1) *算法在第 k 步中断.*

(2) $h_{k+1,k} = 0$.

(3) $\widetilde{\boldsymbol{v}}_{k+1} = \boldsymbol{0}$.

(4) $\boldsymbol{r}_0$ *的最小多项式次数为 k.*

推论 4.4 *GMRES 方法得到的残量模序列 $\{\|\boldsymbol{r}_k\|_2\}$ 是单调下降的, 对于 n 阶方程组至多迭代 n 步即可得到精确解.*

现在考虑 GMRES(m) 方法, 虽然它总能进行下去, 但可能不收敛. 当然, 当 m 充分大时是收敛的. 特别地, 当 $m = n$ 时, 一个重开始迭代即可得到精确解.

1. GMRES(m) 的收敛性定理

现在考虑 GMRES(m) 方法的误差估计. 利用 Krylov 子空间的性质, 注意到任意的 $\boldsymbol{x} = \boldsymbol{x}_0 + \varphi_{k-1}(\boldsymbol{A})\boldsymbol{r}_0$ (即 $\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$), 可导出

$$\begin{aligned}\|\boldsymbol{r}_k\|_2 = \min\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2 &= \min_{\varphi_{k-1}\in\mathcal{P}_{k-1}} \|\boldsymbol{b} - \boldsymbol{A}(\boldsymbol{x}_0 + \varphi_{k-1}(\boldsymbol{A})\boldsymbol{r}_0)\|_2 \\ &= \min_{\varphi_{k-1}\in\mathcal{P}_{k-1}} \|(\boldsymbol{I} - \boldsymbol{A}\varphi_{k-1}(\boldsymbol{A}))\boldsymbol{r}_0\|_2 = \min_{\psi\in\mathcal{P}_k^0} \|\psi(\boldsymbol{A})\boldsymbol{r}_0\|_2, \end{aligned} \tag{4.67}$$

式中: $\mathcal{P}_k^0$ 如式 (4.41) 中所定义.

定理 4.7 *设 $\boldsymbol{A}$ 是正定的 (即 $\boldsymbol{A}$ 的对称部分是对称正定的), $m > 0$ 是任意给定的整数, 并假定在 GMRES(m) 中重新开始了 ℓ 次产生了近似解 $x_m^{(\ell)}$, 则有*

$$\frac{\|\boldsymbol{r}_m^{(\ell)}\|_2}{\|\boldsymbol{r}_0\|_2} \leqslant \left[\sqrt{1 - \frac{\lambda_{\min}^2(\boldsymbol{M})}{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}}\right]^{\ell m}, \tag{4.68}$$

式中: $\boldsymbol{M} = \dfrac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})$, 其他的记号如前所述.

证明 因为 $\boldsymbol{A}$ 是正定的, 即其对称部分

$$\boldsymbol{M}=\frac{1}{2}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}) \tag{4.69}$$

是对称正定的, 定义

$$\psi_\alpha(t)=(1+\alpha t)^m,\quad \alpha\in\mathbb{R}. \tag{4.70}$$

显然有 $\psi_\alpha(t)\in\mathcal{P}_m^0$. 下面先给出 $\min\limits_{\alpha}\|\psi_\alpha(\boldsymbol{A})\|_2$ 的上界估计.

现在任取一个 $\boldsymbol{u}\in\mathbb{R}^n, \|\boldsymbol{u}\|_2=1$ (即 $\boldsymbol{u}$ 是单位向量), 并记 $\widehat{\psi}_\alpha(t)=1+\alpha t$, 则有

$$\begin{aligned}\|\widehat{\psi}_\alpha(\boldsymbol{A})\boldsymbol{u}\|_2^2&=\boldsymbol{u}^{\mathrm{T}}(\boldsymbol{I}+\alpha\boldsymbol{A})^{\mathrm{T}}(\boldsymbol{I}+\alpha\boldsymbol{A})\boldsymbol{u}\\&=1+2\alpha\boldsymbol{u}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{u}+\alpha^2\boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u},\end{aligned}$$

从而, 当 $\alpha\geqslant 0$ 时, 有

$$\|\widehat{\psi}_\alpha(\boldsymbol{A})\|_2\geqslant\|\widehat{\psi}_\alpha(\boldsymbol{A})\boldsymbol{u}\|_2\geqslant 1, \tag{4.71}$$

而当 $\alpha\leqslant 0$ 时, 有

$$\|\widehat{\psi}_\alpha(\boldsymbol{A})\boldsymbol{u}\|_2^2\leqslant 1+2\alpha\lambda_{\min}(\boldsymbol{M})+\alpha^2\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}), \tag{4.72}$$

式中: $\lambda_{\min}(\boldsymbol{M})$ 和 $\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$ 分别为 $\boldsymbol{M}$ 的最小特征值和 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的最大特征值.

注意到单位向量 $\boldsymbol{u}$ 的任意性, 式 (4.72) 蕴涵着当 $\alpha\leqslant 0$ 时, 有

$$\|\widehat{\psi}_\alpha(\boldsymbol{A})\|_2^2\leqslant 1+2\alpha\lambda_{\min}(\boldsymbol{M})+\alpha^2\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}). \tag{4.73}$$

不等式 (4.73) 的右边是关于 α 的二次函数, 在

$$\alpha=-\frac{\lambda_{\min}(\boldsymbol{M})}{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}\leqslant 0$$

处达到最小值 $1-\lambda_{\min}^2(\boldsymbol{M})/\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$, 从而有

$$\min_{\alpha<0}\|\widehat{\psi}_\alpha(\boldsymbol{A})\|_2^2\leqslant 1-\frac{\lambda_{\min}^2(\boldsymbol{M})}{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}<1.$$

再注意到式 (4.71) , 有

$$\min_{\alpha\in\mathbb{R}}\|\widehat{\psi}_\alpha(\boldsymbol{A})\|_2\leqslant\sqrt{1-\frac{\lambda_{\min}^2(\boldsymbol{M})}{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}}\equiv\kappa. \tag{4.74}$$

如果从 $\boldsymbol{x}_0$ 出发, 应用 GMRES 迭代 m 步得到 $\boldsymbol{x}_m$, 则对 $k=m$ 应用式 (4.67), 有

$$\begin{aligned}\|\boldsymbol{r}_m\|_2&\leqslant\min_{\alpha}\|\psi_\alpha(\boldsymbol{A})\boldsymbol{r}_0\|_2\leqslant\min_{\alpha}\|\psi_\alpha(\boldsymbol{A})\|_2\|\boldsymbol{r}_0\|_2\\&\leqslant\min_{\alpha}\|\widehat{\psi}_\alpha(\boldsymbol{A})\|_2^m\|\boldsymbol{r}_0\|_2\leqslant\kappa^m\|\boldsymbol{r}_0\|_2,\end{aligned} \tag{4.75}$$

其中最后一个不等式用到了式 (4.74).

假定以 $\boldsymbol{x}_m^{(1)}=\boldsymbol{x}_m$ 再重新开始, 用 GMRES 迭代产生 $\boldsymbol{x}_m^{(2)}$, 则由式 (4.75) 又有

$$\|\boldsymbol{r}_m^{(2)}\|_2 \leqslant \kappa^m \|\boldsymbol{r}_m^{(1)}\|_2 \leqslant \kappa^{2m}\|\boldsymbol{r}_0\|_2,$$

式中:

$$\boldsymbol{r}_m^{(2)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_m^{(2)};\ \boldsymbol{r}_m^{(1)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_m^{(1)}=\boldsymbol{r}_m.$$

如此可证, 若重新开始了 ℓ 次, 产生了 $\boldsymbol{x}_m^{(\ell)}$, 则有

$$\|\boldsymbol{r}_m^{(\ell)}\|_2 \leqslant \kappa^{\ell m}\|\boldsymbol{r}_0\|_2,$$

式中: $\boldsymbol{r}_m^{(\ell)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_m^{(\ell)}$. 将 κ 的表达式代入上式即得定理的结论. 证毕. □

注 4.3 定理 4.7 表明, 如果系数矩阵 $\boldsymbol{A}$ 是正定的, 则对任意给定的正数 $m>0$, GMRES(m) 总是收敛的.

2. 儒科夫斯基映射

要研究 GMRES 方法对更广一类线性方程组的收敛性, 需要借助复变函数中的儒科夫斯基 (Joukowski) 映射来导出复变元的 Chebyshew 多项式的一种易于使用的定义.

儒科夫斯基映射是指如下定义的从复平面到复平面的映射:

$$z=\frac{1}{2}(w+w^{-1})\equiv J(w). \tag{4.76}$$

对任意的 $w=r\mathrm{e}^{\mathrm{i}\theta}$, $r>1$, 有

$$z=J(w)=x+\mathrm{i}y\equiv a_r\cos\theta+\mathrm{i}b_r\sin\theta,$$

式中:

$$a_r=\frac{1}{2}(r+r^{-1}),\quad b_r=\frac{1}{2}(r-r^{-1}). \tag{4.77}$$

从几何上看, J 将 w 平面内的圆

$$C_r=\{w=r\mathrm{e}^{\mathrm{i}\theta}:0\leqslant\theta\leqslant 2\pi\},$$

映射到 z 平面的一个椭圆

$$E_r=\left\{z=x+\mathrm{i}y:\frac{x^2}{a_r^2}+\frac{y^2}{b_r^2}=1\right\}.$$

显然, 这一映射也将 w 平面内的圆

$$C_{r^{-1}}=\{w=r^{-1}\mathrm{e}^{-\mathrm{i}\theta}:0\leqslant\theta\leqslant 2\pi\},$$

映射到了 E_r.

当 $r=1$ 时, 有 $b_r=0$. 此时 $z=J(w)=\cos\theta$, 即 J 把单位圆周映射到 z 平面内的线段 $[-1,1]$. 当 θ 从 0 变到 π 时, z 从 1 变到 -1; 而当 θ 从 π 变到 2π 时, z 从 -1 变到 1.

当 C_r 连续地收缩到单位圆周 C_1 时, E_r 将连续地收缩到线段 $[-1,1]$. 因此, 由 E_r 所包围的区域

$$E_r=\left\{z=x+\mathrm{i}\,y:\frac{x^2}{a_r^2}+\frac{y^2}{b_r^2}\leqslant 1\right\}$$

中的每一个点 z, 在圆环

$$C_{1,r}=\left\{w:1\leqslant|w|\leqslant r\right\}$$

中都存在一个点 w, 使得 $z=J(w)$, 即 $J(C_{1,r})=E_r$.

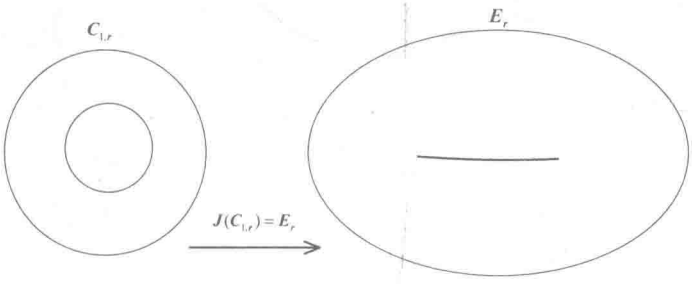

图 4.8 儒可夫斯基映射 $z=J(w)$

现在将 z 平面沿线段 $[-1,1]$ 切开, 取定 $\sqrt{z^2-1}$ 的一个解析分支, 使得

$$w=z+\sqrt{z^2-1}\equiv J^{-1}(z). \tag{4.78}$$

刚好在单位圆外, 即 $|w|\geqslant 1$. 显然, J^{-1} 刚好将 z 平面内的椭圆 E_r 映射到 w 平面内的圆环 $C_{1,r}$, 而且有 $J^{-1}(E_r)=C_{1,r}$.

下面考察 z 平面内的一个中心位于 $z_c=x_c+\mathrm{i}\,y_c$ 的椭圆

$$E(a,b;z_c)=\left\{z=x+\mathrm{i}\,y:\frac{(x-x_c)^2}{a^2}+\frac{(y-y_c)^2}{b^2}=1\right\},$$

这里 $0<b<a$. 令 $d=\sqrt{a^2-b^2}$, 即 d 为椭圆的半焦距. 现在作平移伸缩变换

$$\widetilde{z}=\frac{z-z_c}{d}\quad\left(\text{即}\ \ \widetilde{x}=\frac{x-x_c}{d},\ \ \widetilde{y}=\frac{y-y_c}{d}\right),$$

则该变换将椭圆 $E(a,b;z_c)$ 变为中心在原点的椭圆

$$E_{\widetilde{r}}=\left\{\widetilde{z}=\widetilde{x}+\mathrm{i}\,\widetilde{y}:\frac{\widetilde{x}^2}{a_{\widetilde{r}}^2}+\frac{\widetilde{y}^2}{b_{\widetilde{r}}^2}=1\right\},$$

式中: $\widetilde{r}$ 为

$$\frac{a}{d}=\frac{1}{2}(r+r^{-1})$$

的最大根

$$\widetilde{r}=\frac{a}{d}+\sqrt{\left(\frac{a}{d}\right)^2-1}\,, \tag{4.79}$$

$a_{\widetilde{r}}$ 和 $b_{\widetilde{r}}$ 是在式 (4.77) 中将 r 换作 $\widetilde{r}$ 而得到的.

再由儒科夫斯基映射的性质, 映射

$$\frac{z-z_c}{d}=\frac{w+w^{-1}}{2} \tag{4.80}$$

实现了区域

$$E(a,b;z_c)=\left\{z=x+\mathrm{i}\,y:\frac{(x-x_c)^2}{a^2}+\frac{(y-y_c)^2}{b^2}\leqslant 1\right\} \tag{4.81}$$

与圆环

$$C_{1,\widetilde{r}}=\left\{w:1\leqslant|w|\leqslant\widetilde{r}\right\} \tag{4.82}$$

之间的点与点之间的一一对应关系.

3. 复变元的 Chebyshev 多项式

借助儒科夫斯基映射, 定义函数

$$C_k(z)=\frac{1}{2}(w^k+w^{-k}), \tag{4.83}$$

式中:

$$z=\frac{1}{2}(w+w^{-1}).$$

由定义显然有

$$C_0(z)=\frac{1}{2}(w^0+w^0)=1,$$
$$C_1(z)=\frac{1}{2}(w+w^{-1})=z.$$

而且容易导出

$$\begin{aligned}zC_k(z)&=\frac{1}{2}(w+w^{-1})\cdot\frac{1}{2}(w^k+w^{-k})\\&=\frac{1}{4}\left[(w^{k+1}+w^{-(k+1)})+(w^{k-1}+w^{-(k-1)})\right]\\&=\frac{1}{2}\left[C_{k+1}(z)+C_{k-1}(z)\right],\end{aligned}$$

从而有

$$C_{k+1}(z)=2zC_k(z)-C_{k-1}(z).$$

这表明由式 (4.83) 所定义的复变函数就是第一类 Chebyshev 多项式. 换句话说, 式 (4.83) 是复变元的 Chebyshev 多项式的又一种表达方式.

再由式 (4.78), 即知式 (4.83) 又可写为

$$C_k(z)=\frac{1}{2}\left[\left(z+\sqrt{z^2-1}\right)^k+\left(z+\sqrt{z^2-1}\right)^{-k}\right].$$

4. GMRES 方法的收敛性定理

现在考虑 GMRES 方法的收敛性和误差估计. 设 $\boldsymbol{A}$ 是可对角化的, 即存在一个非奇异矩阵 $\boldsymbol{X}\in\mathbb{R}^{n\times n}$, 使得 $\boldsymbol{A}=\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$, 其中 $\boldsymbol{\Lambda}=\mathrm{diag}\{\lambda_1,\lambda_2,\cdots,\lambda_n\}$. 注意到

$$\psi(\boldsymbol{A})=\boldsymbol{X}\psi(\boldsymbol{\Lambda})\boldsymbol{X}^{-1},\quad \psi(\boldsymbol{\Lambda})=\mathrm{diag}\{\psi(\lambda_1),\psi(\lambda_2),\cdots,\psi(\lambda_n)\},$$

由式 (4.67), 得

$$\|\boldsymbol{r}_k\|_2\leqslant\|\boldsymbol{X}\|_2\cdot\|\boldsymbol{X}^{-1}\|_2\min_{\psi\in\mathcal{P}_k^0}\max_{1\leqslant i\leqslant n}|\psi(\lambda_i)|\cdot\|\boldsymbol{r}_0\|_2. \tag{4.84}$$

再定义

$$\nu(\boldsymbol{A})=\inf\{\|\boldsymbol{X}\|_2\|\boldsymbol{X}^{-1}\|_2:\boldsymbol{A}=\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}\}, \tag{4.85}$$

则有

$$\|\boldsymbol{r}_k\|_2\leqslant\nu(\boldsymbol{A})\min_{\psi\in\mathcal{P}_k^0}\max_{1\leqslant i\leqslant n}|\psi(\lambda_i)|\cdot\|\boldsymbol{r}_0\|_2, \tag{4.86}$$

其中由式 (4.85) 所定义的数 $\nu(\boldsymbol{A})$ 被称为 $\boldsymbol{A}$ 的*谱条件数*.

下面通过选择特殊的多项式 $\psi\in\mathcal{P}_k^0$ 给出

$$\min_{\psi\in\mathcal{P}_k^0}\max_{1\leqslant i\leqslant n}|\psi(\lambda_i)|$$

的尽可能小的上界估计. 由于这里的 λ_i 可能是复数, 因此比对称矩阵时的相应估计要困难得多, 需要用到复变元 Chebyshev 多项式的有关性质.

定理 4.8 *设式 (4.51) 的系数矩阵 $\boldsymbol{A}$ 是可对角化的, 并满足*

$$\lambda(\boldsymbol{A})\subset E(a,b;z_c),$$

其中 $z_c=c$, 而且 $0<b<a<c$, 并假设 GMRES 方法已经进行了 k 步得到了 $\boldsymbol{x}_k$, 则残量 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$ 满足

$$\frac{\|\boldsymbol{r}_k\|_2}{\|\boldsymbol{r}_0\|_2}\leqslant 2\nu(\boldsymbol{A})\left(\frac{a+b}{c+\sqrt{c^2+b^2-a^2}}\right)^k, \tag{4.87}$$

这里的 $\nu(\boldsymbol{A})$ 由式 (4.85) 所定义.

证明 令 $d=\sqrt{a^2-b^2}$, 并定义

$$\widehat{\psi}(z)=C_k\Big(\frac{z-z_c}{d}\Big)/C_k\Big(\frac{-z_c}{d}\Big),$$

则有 $\widehat{\psi}\in\mathcal{P}_k^0$. 这样, 由式 (4.86), 有

$$\frac{\|\boldsymbol{r}_k\|_2}{\|\boldsymbol{r}_0\|_2}\leqslant\nu(\boldsymbol{A})\max_{\lambda\in\lambda(\boldsymbol{A})}|\widehat{\psi}(\lambda)|\leqslant\nu(\boldsymbol{A})\max_{z\in E(a,b,z_c)}|\widehat{\psi}(z)|. \tag{4.88}$$

由式 (4.83) 和式 (4.88), 有

$$C_k\Big(\frac{z-z_c}{d}\Big)=\frac{1}{2}(w^k+w^{-k}). \tag{4.89}$$

式中:

$$\frac{z-z_c}{d}=\frac{1}{2}(w+w^{-1}). \tag{4.90}$$

注意: 式 (4.90) 所定义的映射实现了区域 $E(a,b,z_c)$ 与圆环 $C_{1,\widetilde{r}}$ 之间的点与点之间的一一对应关系, 有

$$\max_{z\in E(a,b,z_c)}|\widehat{\psi}(z)|=\max_{w\in C_{1,\widetilde{r}}}\left|\frac{w^k+w^{-k}}{2C_k(-z_c/d)}\right|. \tag{4.91}$$

在式 (4.90) 中令 $z=0$, 得

$$-\frac{z_c}{d}=-\frac{c}{d}=\frac{1}{2}(w_0+w_0^{-1}). \tag{4.92}$$

解此方程, 并选择其模最大者, 即

$$w_0=-\frac{c}{d}-\sqrt{\left(\frac{c}{d}\right)^2-1}. \tag{4.93}$$

此外, 对任意的 $w=r\mathrm{e}^{\mathrm{i}\theta}$, 其中 $1\leqslant r\leqslant\widetilde{r}$, 有

$$w^k+w^{-k}=(r^k+r^{-k})\cos k\theta+\mathrm{i}\,(r^k-r^{-k})\sin k\theta,$$

从而有

$$\begin{aligned}|w^k+w^{-k}|^2&=(r^k+r^{-k})^2\cos^2 k\theta+(r^k-r^{-k})^2\sin^2 k\theta\\&=(r^k-r^{-k})^2+4\cos^2 k\theta.\end{aligned}$$

显然, 该函数在圆周 $w=r\mathrm{e}^{\mathrm{i}\theta}$ 上的最大值为

$$(r^k-r^{-k})^2+4=(r^k+r^{-k})^2,$$

从而

$$|w^k+w^{-k}|\leqslant r^k+r^{-k},\quad w=r\mathrm{e}^{\mathrm{i}\theta}. \tag{4.94}$$

这样, 由式 (4.89), 式 (4.91), 式 (4.92) 和式 (4.94), 有

$$\max_{z\in E(a,b,z_c)}|\widehat{\psi}(z)|=\frac{\widetilde{r}^k+\widetilde{r}^{-k}}{|w_0^k+w_0^{-k}|}, \tag{4.95}$$

这里 $\widetilde{r}$ 由 (4.79) 定义.

此外, 由 $\widetilde{r}$ 的定义式 (4.79), w_0 的定义 (4.93) 以及 $d^2=a^2-b^2$, 得

$$\widetilde{r}=\frac{a+b}{d},\quad w_0=-\frac{c+\sqrt{c^2+b^2-a^2}}{d},$$

从而

$$\frac{\widetilde{r}^k+\widetilde{r}^{-k}}{|w_0^k+w_0^{-k}|}=\frac{[(a+b)/d]^k+[(a+b)/d]^{-k}}{\left[(c+\sqrt{c^2+b^2-a^2})/d\right]^k+\left[(c+\sqrt{c^2+b^2-a^2})/d\right]^{-k}}$$

$$
\begin{aligned}
&= \left(\frac{a+b}{c+\sqrt{c^2+b^2-a^2}}\right)^k \frac{1+[d/(a+b)]^{2k}}{1+\left[(c+\sqrt{c^2+b^2-a^2})/d\right]^{-2k}} \\
&\leqslant \left(\frac{a+b}{c+\sqrt{c^2+b^2-a^2}}\right)^k \left[1+\left(\frac{a-b}{a+b}\right)^k\right] \\
&\leqslant 2\left(\frac{a+b}{c+\sqrt{c^2+b^2-a^2}}\right)^k . \qquad (4.96)
\end{aligned}
$$

将式 (4.96), 式 (4.95) 和式 (4.88) 结合起来即得定理的结论. 证毕. □

4.3 极小残量法

本节介绍求解对称不定线性方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \qquad (4.97)$$

的极小残量法, 其中系数矩阵 $\boldsymbol{A}$ 是给定的对称不定矩阵 (其特征值有正有负), $\boldsymbol{b} \in \mathbb{R}^n$ 是给定的列向量, 而 $\boldsymbol{x} \in \mathbb{R}^n$ 是待求的未知向量.

若将 4.1 节所介绍的共轭梯度法应用到这类线性方程组上, 其投影方程组 (4.13) 的系数矩阵 $\boldsymbol{T}_k$ 就可能是奇异的, 从而导致算法失败 (即在没有求得式 (4.97) 的很好的近似解之前就发生中断). 基于这种情况, Paige 和 Saunders 于 1975 年在文献 [26] 中提出了一种克服这一困难的方法, 导出了求解对称不定线性方程组的 SYMMLQ 方法. 此外, 在该文献中还给出了求解这类方程组的极小残量法 (MINRES). 本节着重介绍极小残量法, SYMMLQ 方法将在后面再做介绍.

4.3.1 MINRES 方法

极小残量法的出发点是寻找一个 $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$, 使得

$$\|\boldsymbol{r}_k\|_2 = \min\{\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2 : \boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)\}, \qquad (4.98)$$

这里 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ 是残量, 而 $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$ 是由矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{r}_0$ 所生成的 Krylov 子空间.

假设已经求得一个以 $\boldsymbol{v}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|_2$ 为初始向量的长度为 k 的 Lanczos 分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_{k+1}\widetilde{\boldsymbol{T}}_k, \qquad (4.99)$$

式中:

$$\boldsymbol{V}_{k+1} = [\boldsymbol{V}_k, \boldsymbol{v}_{k+1}] = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k, \boldsymbol{v}_{k+1}] \in \mathbb{R}^{n\times(k+1)}$$

满足 $\boldsymbol{V}_{k+1}^{\mathrm{T}}\boldsymbol{V}_{k+1}=\boldsymbol{I}_{k+1}$, 而 $\widetilde{\boldsymbol{T}}_k$ 是 $(k+1)\times k$ 阶的三对角矩阵, 即

$$\widetilde{\boldsymbol{T}}_k=\begin{bmatrix}\alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-2} & \alpha_{k-1} & \beta_{k-1} \\ & & & \beta_{k-1} & \alpha_k \\ & & & 0 & \beta_k\end{bmatrix},\ \beta_i\neq 0. \tag{4.100}$$

注意到 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\mathcal{R}(\boldsymbol{V}_k)$, 则对任意的 $\boldsymbol{x}=\boldsymbol{x}_0+\boldsymbol{V}_k\boldsymbol{z}\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 有

$$\begin{aligned}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2&=\|\boldsymbol{r}_0-\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{z}\|_2=\|\beta\boldsymbol{V}_{k+1}\boldsymbol{e}_1-\boldsymbol{V}_{k+1}\widetilde{\boldsymbol{T}}_k\boldsymbol{z}\|_2\\&=\|\boldsymbol{V}_{k+1}(\beta\boldsymbol{e}_1-\widetilde{\boldsymbol{T}}_k\boldsymbol{z})\|_2=\|\beta\boldsymbol{e}_1-\widetilde{\boldsymbol{T}}_k\boldsymbol{z}\|_2,\end{aligned} \tag{4.101}$$

式中: $\beta=\|\boldsymbol{r}_0\|_2$. 这样求 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 满足式 (4.98) 就等价于求 $\boldsymbol{z}_k\in\mathbb{R}^k$, 使得

$$\|\beta\boldsymbol{e}_1-\widetilde{\boldsymbol{T}}_k\boldsymbol{z}_k\|_2=\min\{\|\beta\boldsymbol{e}_1-\widetilde{\boldsymbol{T}}_k\boldsymbol{z}\|_2:\boldsymbol{z}\in\mathbb{R}^k\}. \tag{4.102}$$

一旦这样的 $\boldsymbol{z}_k$ 求得, 则所寻求的 $\boldsymbol{x}_k$ 就是 $\boldsymbol{x}_k=\boldsymbol{x}_0+\boldsymbol{V}_k\boldsymbol{z}_k$.

极小化问题 (4.102) 是一个系数矩阵为三对角矩阵的最小二乘问题, 现在已有很多十分成熟的数值方法来求它的解. 这里将用 QR 分解来求解式 (4.102).

由于 $\widetilde{\boldsymbol{T}}_k$ 具有式 (4.100) 所示形状, 故可以计算 k 个 Givens 变换 $\boldsymbol{G}_1,\boldsymbol{G}_2,\cdots,\boldsymbol{G}_k$, 使得

$$\boldsymbol{G}_k\boldsymbol{G}_{k-1}\cdots\boldsymbol{G}_2\boldsymbol{G}_1\widetilde{\boldsymbol{T}}_k=\begin{bmatrix}\boldsymbol{R}_k\\ \boldsymbol{0}\end{bmatrix}, \tag{4.103}$$

式中:

$$\boldsymbol{G}_i=\mathrm{diag}\left(\boldsymbol{I}_{i-1},\begin{bmatrix}c_i & s_i\\ -s_i & c_i\end{bmatrix},\boldsymbol{I}_{k-i}\right)\in\mathbb{R}^{(k+1)\times(k+1)},\quad c_i^2+s_i^2=1,$$

$$\boldsymbol{R}_k=\begin{bmatrix}\gamma_1 & \delta_1 & \varepsilon_1 & & \\ & \gamma_2 & \delta_2 & \ddots & \\ & & \ddots & \ddots & \varepsilon_{k-2}\\ & & & \gamma_{k-1} & \delta_{k-1}\\ & & & & \gamma_k\end{bmatrix}, \tag{4.104}$$

而且 $\beta_i\neq 0$ 蕴涵着 $\gamma_i\neq 0$, $i=1,2,\cdots,k$, 从而 $\boldsymbol{R}_k$ 是非奇异的. 令

$$\boldsymbol{G}=\boldsymbol{G}_k\boldsymbol{G}_{k-1}\cdots\boldsymbol{G}_2\boldsymbol{G}_1,\quad\begin{bmatrix}\boldsymbol{t}_k\\ \rho_k\end{bmatrix}=\boldsymbol{G}(\beta\boldsymbol{e}_1),\quad\boldsymbol{t}_k=(\tau_1,\tau_2,\cdots,\tau_k)^{\mathrm{T}}, \tag{4.105}$$

则 $\boldsymbol{G}$ 是 $k+1$ 阶正交矩阵, 且 τ_k, ρ_k 如式 (4.58) 所定义, 即

$$\begin{cases} \tau_1 = \beta c_1, \\ \tau_i = (-1)^{i-1}\beta s_1 s_2 \cdots s_{i-1} c_i, \quad i = 2, 3, \cdots, k, \\ \rho_k = (-1)^k \beta s_1 s_2 \cdots s_k. \end{cases} \tag{4.106}$$

这样, 利用分解式 (4.103) 和记号 (4.105), 有

$$\begin{aligned} \|\widetilde{\boldsymbol{T}}_k \boldsymbol{z} - \beta \boldsymbol{e}_1\|_2^2 = \|\boldsymbol{G}(\widetilde{\boldsymbol{T}}_k \boldsymbol{z} - \beta \boldsymbol{e}_1)\|_2^2 &= \left\| \begin{bmatrix} \boldsymbol{R}_k \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{z} - \begin{bmatrix} \boldsymbol{t}_k \\ \rho_k \end{bmatrix} \right\|_2^2 \\ &= \|\boldsymbol{R}_k \boldsymbol{z} - \boldsymbol{t}_k\|_2^2 + \rho_k^2. \end{aligned}$$

对任意的 $\boldsymbol{z} \in \mathbb{R}^k$ 成立. 由此立即知道, 最小二乘问题 (4.102) 有唯一解, 即

$$\boldsymbol{z}_k = \boldsymbol{R}_k^{-1} \boldsymbol{t}_k, \tag{4.107}$$

而且有

$$\|\widetilde{\boldsymbol{T}}_k \boldsymbol{z}_k - \beta \boldsymbol{e}_1\|_2 = |\rho_k|. \tag{4.108}$$

由式 (4.107) 求得 $\boldsymbol{z}_k$ 之后, 就可算出所求的 $\boldsymbol{x}_k$ 为 $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{z}_k$.

由式 (4.101) 和式 (4.108), 得

$$\|\boldsymbol{r}_k\|_2 = \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k\|_2 = \|\widetilde{\boldsymbol{T}}_k \boldsymbol{z}_k - \beta \boldsymbol{e}_1\|_2 = |\rho_k|,$$

故在实际计算时, 可以用

$$|\rho_k| / \beta \leqslant \varepsilon \tag{4.109}$$

作为迭代终止的准则, 其中 $\varepsilon > 0$ 是给定的误差要求.

由式 (4.106) 可知, ρ_k 的值并不需要 $\boldsymbol{z}_k$ 和 $\boldsymbol{x}_k$ 的信息. 因此, 只需在 ρ_k 满足式 (4.109) 之后, 再去计算 $\boldsymbol{z}_k$ 和 $\boldsymbol{x}_k$ 即可.

至此, 已经给出了求解极小化问题 (4.98) 的具体方法, 然而这样做的一个最大缺点就是需要保存 $\boldsymbol{V}_k$ 的所有列向量, 随着 k 的增加, 存储量的需求会越来越大. 幸运的是, 在文献 [26] 中还给出了一种巧妙的方法来避免这种情形的出现. 这一方法的具体做法是不将 $\boldsymbol{z}_k$ 明确求出, 而是由 $\boldsymbol{z}_k$ 直接导出计算 $\boldsymbol{x}_k$ 的递推公式.

将式 (4.107) 代入 $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{z}_k$, 得

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{R}_k^{-1} \boldsymbol{t}_k = \boldsymbol{x}_0 + \boldsymbol{P}_k \boldsymbol{t}_k, \tag{4.110}$$

式中: $\boldsymbol{P}_k = \boldsymbol{V}_k \boldsymbol{R}_k^{-1}$. 这样, 只要将 $\boldsymbol{P}_k$ 算出, 就可以通过式 (4.110) 计算 $\boldsymbol{x}_k$. 令 $\boldsymbol{P}_k = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_k]$, 比较 $\boldsymbol{P}_k \boldsymbol{R}_k = \boldsymbol{V}_k$ 两边的每一列, 得

$$\begin{aligned} &\gamma_1 \boldsymbol{p}_1 = \boldsymbol{v}_1, \\ &\delta_1 \boldsymbol{p}_1 + \gamma_2 \boldsymbol{p}_2 = \boldsymbol{v}_2, \\ &\varepsilon_{i-2} \boldsymbol{p}_{i-2} + \delta_{i-1} \boldsymbol{p}_{i-1} + \gamma_i \boldsymbol{p}_i = \boldsymbol{v}_i, \ i = 3, 4, \cdots, k. \end{aligned}$$

由此可求得 $\boldsymbol{P}_k$ 的列向量为

$$\begin{cases} \boldsymbol{p}_1 = \boldsymbol{v}_1/\gamma_1, \\ \boldsymbol{p}_2 = (\boldsymbol{v}_2 - \delta_1\boldsymbol{p}_1)/\gamma_2, \\ \boldsymbol{p}_i = (\boldsymbol{v}_i - \varepsilon_{i-2}\boldsymbol{p}_{i-2} - \delta_{i-1}\boldsymbol{p}_{i-1})/\gamma_i, \quad i = 3, 4, \cdots, k. \end{cases} \tag{4.111}$$

下面借助式 (4.106), 式 (4.110) 和式 (4.111) 导出计算 $\boldsymbol{x}_k$ 的递推公式.

首先注意, 若 Lanczos 分解的长度由 k 增加到 $k+1$, 则有

$$\widetilde{\boldsymbol{T}}_{k+1} = \left[\begin{array}{c|c} \widetilde{\boldsymbol{T}}_k & \widetilde{\boldsymbol{t}}_{k+1} \\ \hline \boldsymbol{0} & \beta_{k+1} \end{array}\right], \quad \widetilde{\boldsymbol{t}}_{k+1} = (0, \cdots, 0, \beta_k, \alpha_{k+1})^{\mathrm{T}},$$

于是, 有

$$\boldsymbol{R}_{k+1} = \left[\begin{array}{c|c} \boldsymbol{R}_k & \widetilde{\boldsymbol{r}}_{k+1} \\ \hline \boldsymbol{0} & \gamma_{k+1} \end{array}\right], \quad \widetilde{\boldsymbol{r}}_{k+1} = (0, \cdots, 0, \varepsilon_{k-1}, \delta_k)^{\mathrm{T}}, \tag{4.112}$$

式中:

$$\begin{aligned} &\varepsilon_{k-1} = s_{k-1}\beta_k, \quad \widetilde{\beta}_k = c_{k-1}\beta_k, \\ &\delta_k = c_k\widetilde{\beta}_k + s_k\alpha_{k+1}, \quad \widetilde{\alpha}_{k+1} = -s_k\widetilde{\beta}_k + c_k\alpha_{k+1}, \end{aligned} \tag{4.113}$$

上面的四个等式由如下 Givens 变换得到:

$$\widetilde{\boldsymbol{T}}_{k+1} = \left[\begin{array}{ccccccc:c} \alpha_1 & \beta_1 & & & & & & \\ \beta_1 & \alpha_2 & \beta_2 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & \beta_{k-3} & \alpha_{k-2} & \beta_{k-2} & & & \\ & & & \beta_{k-2} & \alpha_{k-1} & \beta_{k-1} & & \\ & & & & \beta_{k-1} & \alpha_k & \beta_k & \\ & & & & & \beta_k & \alpha_{k+1} & \\ \hdashline & & & & & & & \beta_{k+1} \end{array}\right] \xrightarrow{\boldsymbol{G}_1} \left[\begin{array}{ccccccc:c} \gamma_1 & \delta_1 & \varepsilon_1 & & & & & \\ 0 & \widetilde{\alpha}_2 & \widetilde{\beta}_2 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & \beta_{k-3} & \alpha_{k-2} & \beta_{k-2} & & & \\ & & & \beta_{k-2} & \alpha_{k-1} & \beta_{k-1} & & \\ & & & & \beta_{k-1} & \alpha_k & \beta_k & \\ & & & & & \beta_k & \alpha_{k+1} & \\ \hdashline & & & & & & & \beta_{k+1} \end{array}\right]$$

$$\xrightarrow{\boldsymbol{G}_2} \cdots \xrightarrow{\boldsymbol{G}_{k-2}} \left[\begin{array}{ccccccc:c} \gamma_1 & \delta_1 & \varepsilon_1 & & & & & \\ & \gamma_2 & \delta_2 & \varepsilon_2 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & & \gamma_{k-2} & \delta_{k-2} & \varepsilon_{k-2} & \\ & & & & & \widetilde{\alpha}_{k-1} & \widetilde{\beta}_{k-1} & \\ & & & & & \beta_{k-1} & \alpha_k & \beta_k \\ & & & & & & \beta_k & \alpha_{k+1} \\ \hdashline & & & & & & & \beta_{k+1} \end{array}\right]$$

$$\xrightarrow{\boldsymbol{G}_{k-1}}\left[\begin{array}{ccccc:c}
\gamma_1 & \delta_1 & \varepsilon_1 & & & \\
 & \gamma_2 & \delta_2 & \varepsilon_2 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & \gamma_{k-2} & \delta_{k-2} & \varepsilon_{k-2} \\
 & & & & \gamma_{k-1} & \delta_{k-1} & \varepsilon_{k-1} \\
 & & & & & \widetilde{\alpha}_k & \widetilde{\beta}_k \\
 & & & & & \beta_k & \alpha_{k+1} \\
\hdashline
 & & & & & & \beta_{k+1}
\end{array}\right]\xrightarrow{\boldsymbol{G}_k}\left[\begin{array}{ccccc:c}
\gamma_1 & \delta_1 & \varepsilon_1 & & & \\
 & \gamma_2 & \delta_2 & \varepsilon_2 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & \gamma_{k-2} & \delta_{k-2} & \varepsilon_{k-2} \\
 & & & & \gamma_{k-1} & \delta_{k-1} & \varepsilon_{k-1} \\
 & & & & & \gamma_k & \delta_k \\
 & & & & & & \widetilde{\alpha}_{k+1} \\
\hdashline
 & & & & & & \beta_{k+1}
\end{array}\right],$$

即由

$$\begin{bmatrix} c_{k-1} & s_{k-1} \\ -s_{k-1} & c_{k-1} \end{bmatrix}\begin{bmatrix} 0 \\ \beta_k \end{bmatrix}=\begin{bmatrix} \varepsilon_{k-1} \\ \widetilde{\beta}_k \end{bmatrix},\quad \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix}\begin{bmatrix} \widetilde{\beta}_k \\ \alpha_{k+1} \end{bmatrix}=\begin{bmatrix} \delta_k \\ \widetilde{\alpha}_{k+1} \end{bmatrix}$$

得到. 而 γ_{k+1} 是在确定第 $k+1$ 个 Givens 变换 $\boldsymbol{G}_{k+1}$ 时得到的, 即计算 c_{k+1} 和 s_{k+1}, 使得

$$\begin{bmatrix} c_{k+1} & s_{k+1} \\ -s_{k+1} & c_{k+1} \end{bmatrix}\begin{bmatrix} \widetilde{\alpha}_{k+1} \\ \beta_{k+1} \end{bmatrix}=\begin{bmatrix} \gamma_{k+1} \\ 0 \end{bmatrix}. \tag{4.114}$$

由式 (4.106) 可知 $\boldsymbol{t}_{k+1}=(\boldsymbol{t}_k^{\mathrm{T}},\tau_{k+1})^{\mathrm{T}}$, 其中

$$\tau_{k+1}=(-1)^k\beta s_1s_2\cdots s_kc_{k+1}=\rho_kc_{k+1}, \tag{4.115}$$

而

$$\rho_{k+1}=(-1)^{k+1}\beta s_1s_2\cdots s_ks_{k+1}=-\rho_ks_{k+1}. \tag{4.116}$$

由式 (4.111) 和式 (4.112), 有

$$\boldsymbol{P}_{k+1}=\boldsymbol{V}_{k+1}\boldsymbol{R}_{k+1}^{-1}=[\boldsymbol{V}_k,\boldsymbol{v}_{k+1}]\left[\begin{array}{c|c} \boldsymbol{R}_k^{-1} & \boldsymbol{s}_{k+1} \\ \hline \boldsymbol{0} & \gamma_{k+1}^{-1} \end{array}\right]=\left[\boldsymbol{P}_k,\boldsymbol{p}_{k+1}\right], \tag{4.117}$$

其中

$$\boldsymbol{p}_{k+1}=(\boldsymbol{v}_{k+1}-\varepsilon_{k-1}\boldsymbol{p}_{k-1}-\delta_k\boldsymbol{p}_k)/\gamma_{k+1}. \tag{4.118}$$

这是由于

$$\begin{aligned}\boldsymbol{I}_{k+1}=\boldsymbol{R}_{k+1}\boldsymbol{R}_{k+1}^{-1}&=\left[\begin{array}{c|c} \boldsymbol{R}_k & \widetilde{\boldsymbol{r}}_{k+1} \\ \hline \boldsymbol{0} & \gamma_{k+1} \end{array}\right]\left[\begin{array}{c|c} \boldsymbol{R}_k^{-1} & \boldsymbol{s}_{k+1} \\ \hline \boldsymbol{0} & \gamma_{k+1}^{-1} \end{array}\right] \\ &=\left[\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{R}_k\boldsymbol{s}_{k+1}+\gamma_{k+1}^{-1}\widetilde{\boldsymbol{r}}_{k+1} \\ \hline \boldsymbol{0} & 1 \end{array}\right],\end{aligned}$$

由此可得

$$\boldsymbol{R}_k\boldsymbol{s}_{k+1}+\gamma_{k+1}^{-1}\widetilde{\boldsymbol{r}}_{k+1}=\boldsymbol{0}\Longrightarrow\boldsymbol{s}_{k+1}=-\gamma_{k+1}^{-1}\boldsymbol{R}_k^{-1}\widetilde{\boldsymbol{r}}_{k+1}.$$

于是

$$\begin{aligned}\boldsymbol{p}_{k+1} &= \boldsymbol{V}_k\boldsymbol{s}_{k+1} + \gamma_{k+1}^{-1}\boldsymbol{v}_{k+1} = -\gamma_{k+1}^{-1}\boldsymbol{V}_k\boldsymbol{R}_k^{-1}\widetilde{\boldsymbol{r}}_{k+1} + \gamma_{k+1}^{-1}\boldsymbol{v}_{k+1}\\ &= \big(\boldsymbol{v}_{k+1} - \boldsymbol{P}_k\widetilde{\boldsymbol{r}}_{k+1}\big)/\gamma_{k+1} = (\boldsymbol{v}_{k+1} - \varepsilon_{k-1}\boldsymbol{p}_{k-1} - \delta_k\boldsymbol{p}_k)/\gamma_{k+1}.\end{aligned}$$

从而有

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_0 + \boldsymbol{P}_{k+1}\boldsymbol{t}_{k+1} = \boldsymbol{x}_0 + [\boldsymbol{P}_k, \boldsymbol{p}_{k+1}]\begin{bmatrix}\boldsymbol{t}_k\\ \tau_{k+1}\end{bmatrix} = \boldsymbol{x}_k + \tau_{k+1}\boldsymbol{p}_{k+1}. \tag{4.119}$$

这就有人们希望得到的 $\boldsymbol{x}_k$ 的递推公式. 这样, 每次迭代只需保存 $\boldsymbol{x}_k, \boldsymbol{p}_{k-1}, \boldsymbol{p}_k, \boldsymbol{v}_k, \boldsymbol{v}_{k+1}$ 这五个向量即可.

综述上面的讨论, 就得到了如下的极小残量方法.

算法 4.9 (MINRES 方法) 给定 n 阶非奇异的实对称矩阵 $\boldsymbol{A}$, n 维向量 $\boldsymbol{b}$ 和允许误差 $\varepsilon > 0$. 本算法计算近似解向量 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\beta = \|\boldsymbol{r}_0\|_2$; $\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$;

$\alpha_1 = \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_1$; $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}_1 - \alpha_1\boldsymbol{v}_1$; $\beta_1 = \|\boldsymbol{u}\|_2$;

if $\beta_1 = 0$

  $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{r}_0/\alpha_1$; 结束

else

  $\boldsymbol{v}_2 = \boldsymbol{u}/\beta_1$;

end

$c_0 = 1$; $s_0 = 0$; $\boldsymbol{p}_0 = \boldsymbol{0}$;

确定 $c_1 = \cos\theta_1$ 和 $s_1 = \sin\theta_1$, 使得

$$\begin{bmatrix} c_1 & s_1\\ -s_1 & c_1\end{bmatrix}\begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix} = \begin{bmatrix}\gamma_1\\ 0\end{bmatrix};$$

$\boldsymbol{p}_1 = \boldsymbol{v}_1/\gamma_1$; $\rho_1 = -\beta s_1$; $\tau_1 = \beta c_1$;

$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \tau_1\boldsymbol{p}_1$; $k = 1$;

while $(|\rho_k|/\beta > \varepsilon)$

  $\alpha_{k+1} = \boldsymbol{v}_{k+1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_{k+1}$;

  $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}_{k+1} - \alpha_{k+1}\boldsymbol{v}_{k+1} - \beta_k\boldsymbol{v}_k$;

  $\beta_{k+1} = \|\boldsymbol{u}\|_2$;

  if $\beta_{k+1} \neq 0$

    $\boldsymbol{v}_{k+2} = \boldsymbol{u}/\beta_{k+1}$;

  end

  $\varepsilon_{k-1} = s_{k-1}\beta_k$; $\widetilde{\beta}_k = c_{k-1}\beta_k$;

  $\delta_k = c_k\widetilde{\beta}_k + s_k\alpha_{k+1}$; $\widetilde{\alpha}_{k+1} = -s_k\widetilde{\beta}_k + c_k\alpha_{k+1}$;

  确定 $c_{k+1} = \cos\theta_{k+1}$和 $s_{k+1} = \sin\theta_{k+1}$, 使得

$$\begin{bmatrix} c_{k+1} & s_{k+1}\\ -s_{k+1} & c_{k+1}\end{bmatrix}\begin{bmatrix}\widetilde{\alpha}_{k+1}\\ \beta_{k+1}\end{bmatrix} = \begin{bmatrix}\gamma_{k+1}\\ 0\end{bmatrix};$$

$\tau_{k+1} = \rho_k c_{k+1}$; $\rho_{k+1} = -\rho_k s_{k+1}$;

$\boldsymbol{p}_{k+1} = (\boldsymbol{v}_{k+1} - \varepsilon_{k-1}\boldsymbol{p}_{k-1} - \delta_k \boldsymbol{p}_k)/\gamma_{k+1}$;

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \tau_{k+1}\boldsymbol{p}_{k+1}$;

$k = k + 1$;

end

MINRES 方法的 MATLAB 程序如下:

```
%MINRES方法的程序-mminres.m
function [x,iter,time,res,resvec]=mminres(A,b,x,max_it,tol)
%极小残量法求解对称不定方程组Ax=b
tic; r=b-A*x; b=norm(r); v=r/b; bt=b;
z=A*v; a=v'*z; v=z-a*v; b1=norm(v);
if (b1~=0), v1=v/b1; end
c0=1; s0=0; p0=zeros(length(b),1);
[c,s,gama]=givens(a,b1);  %Givens变换
p=v/gama; rho=-b*s; tau=b*c;
x=x+tau*p; iter=1;
while (iter<max_it)
    res=abs(rho)/bt; resvec(iter)=res;
    if (res<tol), break; end
    z=A*v1; a=v1'*z; v=z-a*v1-b1*v; b2=norm(v);
    if (b2~=0), v2=v/b2; end
    epsi=s0*b1; bh1=c0*b1;
    dta=c*bh1+s*a; ah=-s*bh1+c*a;
    [c1,s1,gama]=givens(ah,b2);  %Givens变换
    tau=rho*c1; rho=-rho*s1;
    p1=(v1-epsi*p0-dta*p)/gama;
    x=x+tau*p1; b1=b2; iter=iter+1;
    v=v1; v1=v2; p0=p; p=p1;
    c0=c; c=c1; s0=s; s=s1;
end
time=toc;
```

例 4.8 假设线性方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, 其中矩阵 $\boldsymbol{A}$ 来自 http://www.cise.ufl.edu./research/sparse/matrices/PARSEC/SiNa.html, 它是一个 5743 阶的实对称不定稀疏矩阵, 右端项 $\boldsymbol{b}$ 为

$$\boldsymbol{b} = \boldsymbol{A}\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n.$$

显然, 该方程组的真解为 $\boldsymbol{x}^* = (1, 1, \cdots, 1)^{\mathrm{T}}$. 应用算法 4.9 到该线性方程组上, 第 224 步得到的 $\widehat{\boldsymbol{x}} = \boldsymbol{x}_{224}$ 满足

$$\|\widehat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 5.9139 \times 10^{-8},$$

迭代过程的收敛轨迹如图 4.9 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

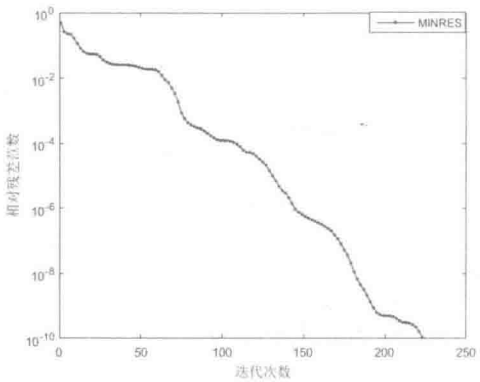

图 4.9 MINRES 方法的收敛特性

4.3.2 PMINRES 方法

MINRES 方法 (算法 4.9) 与 4.1 节的共轭梯度法在形式上很不相同, 且不便于导出相应的预处理算法. 下面从另一个途径推导出与共轭梯度法形式类似的 MINRES 方法, 并由此导出预处理极小残量法 (简记为 PMINRES 方法) 的递推算法. 由前面的论述可知, 极小残量法就是求向量 $\boldsymbol{z}_k$ 及 $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k\boldsymbol{z}_k$ 使得 $\|\boldsymbol{r}_k\|_2^2$ 达到极小, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$. 由于 $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0) = \mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$, 因此, 若记 $\boldsymbol{S}_k = \big[\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\big]$, 则 $\boldsymbol{x}_k$ 也可表示成

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{S}_k\boldsymbol{z}_k.$$

于是问题等价于求 $\boldsymbol{z}_k$ 使 $\|\boldsymbol{r}_k\|_2^2$ 达到极小. 注意到

$$\begin{aligned}\|\boldsymbol{r}_k\|_2^2 &= \big(\boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x}_k), \boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x}_k)\big)\\ &= \big(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x}_0 - \boldsymbol{S}_k\boldsymbol{z}_k), \boldsymbol{x}^* - \boldsymbol{x}_0 - \boldsymbol{S}_k\boldsymbol{z}_k\big).\end{aligned} \tag{4.120}$$

因为 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{A}^2$ 是对称正定矩阵, 因此可以定义一种新内积

$$(\boldsymbol{x}, \boldsymbol{z})_{\boldsymbol{A}^2} = \big(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}, \boldsymbol{z}\big),$$

因此

$$\|\boldsymbol{r}_k\|_2^2 = (\boldsymbol{x}^* - \boldsymbol{x}_0 - \boldsymbol{S}_k\boldsymbol{z}_k, \boldsymbol{x}^* - \boldsymbol{x}_0 - \boldsymbol{S}_k\boldsymbol{z}_k)_{\boldsymbol{A}^2}.$$

按照极小与正交关系定理, 可得 $\boldsymbol{z}_k$ 当且仅当满足条件 (即 $\boldsymbol{r}_k \perp \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$)

$$\big(\boldsymbol{x}^* - \boldsymbol{x}_0 - \boldsymbol{S}_k\boldsymbol{z}_k, \boldsymbol{A}^i\boldsymbol{r}_0\big)_{\boldsymbol{A}^2} = 0, \quad i = 0, 1, \cdots, k-1.$$

若将 Krylov 子空间 $\mathrm{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\}$ 按内积 $(\cdot,\cdot)_{\boldsymbol{A}^2}$ 标准正交化得 $\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_{k-1}$, 则有

$$\boldsymbol{P}_k = [\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_{k-1}] = \boldsymbol{S}_k \boldsymbol{R}_k,$$

式中: $\boldsymbol{R}_k$ 为上三角矩阵, 它在 $\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0$ 线性无关的假设下是非奇异的, 且在对角元确定后是唯一确定的. 于是有

$$\boldsymbol{S}_k \boldsymbol{z}_k = \boldsymbol{P}_k \boldsymbol{R}_k^{-1} \boldsymbol{z}_k.$$

记 $\boldsymbol{\alpha}_k := \boldsymbol{R}_k^{-1}\boldsymbol{z}_k$, 则

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{P}_k \boldsymbol{\alpha}_k.$$

注意到

$$\boldsymbol{P}_k \boldsymbol{\alpha}_k = \sum_{j=0}^{k-1} \alpha_j^{(k)} \boldsymbol{p}_j.$$

由

$$\left(\boldsymbol{x}^* - \boldsymbol{x}_0 - \boldsymbol{P}_k \boldsymbol{\alpha}_k, \boldsymbol{p}_i\right)_{\boldsymbol{A}^2} = 0, \ i = 0, 1, \cdots, k-1$$

及 $\boldsymbol{p}_i$ 的 $(\cdot,\cdot)_{\boldsymbol{A}^2}$ 正交性, 得

$$\begin{aligned} \alpha_i^{(k)} &= \frac{(\boldsymbol{x}^* - \boldsymbol{x}_0, \boldsymbol{p}_i)_{\boldsymbol{A}^2}}{(\boldsymbol{p}_i, \boldsymbol{p}_i)_{\boldsymbol{A}^2}} = \frac{(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{A}\boldsymbol{x}_0, \boldsymbol{A}\boldsymbol{p}_i)}{(\boldsymbol{A}\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_i)} \\ &= \frac{(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0, \boldsymbol{A}\boldsymbol{p}_i)}{(\boldsymbol{A}\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_i)} = \frac{(\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{p}_i)}{(\boldsymbol{A}\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_i)}. \end{aligned}$$

注意到 $\alpha_i^{(k)}$ 计算公式与 k 无关, 于是可以得到简单的递推公式, 即

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{x}_0 + \boldsymbol{P}_k \boldsymbol{\alpha}_k = \boldsymbol{x}_0 + \sum_{i=0}^{k-1} \alpha_i \boldsymbol{p}_i \\ &= \boldsymbol{x}_0 + \sum_{i=0}^{k-2} \alpha_i \boldsymbol{p}_i + \alpha_{k-1}\boldsymbol{p}_{k-1} \\ &= \boldsymbol{x}_{k-1} + \alpha_{k-1}\boldsymbol{p}_{k-1}. \end{aligned}$$

现在剩下的问题是如何求得标准正交向量组 $\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_{k-1}$. 按照共轭梯度法的思想, 取

$$\boldsymbol{p}_0 = \boldsymbol{r}_0, \quad \boldsymbol{p}_1 = \boldsymbol{A}\boldsymbol{p}_0 + \gamma_0^{(1)}\boldsymbol{p}_0.$$

要使 $(\boldsymbol{p}_0, \boldsymbol{p}_1)_{\boldsymbol{A}^2} = 0$, 必须

$$\gamma_0^{(1)} = -\frac{(\boldsymbol{A}\boldsymbol{p}_0, \boldsymbol{p}_0)_{\boldsymbol{A}^2}}{(\boldsymbol{p}_0, \boldsymbol{p}_0)_{\boldsymbol{A}^2}} = -\frac{(\boldsymbol{A}^2\boldsymbol{p}_0, \boldsymbol{A}\boldsymbol{p}_0)}{(\boldsymbol{A}\boldsymbol{p}_0, \boldsymbol{A}\boldsymbol{p}_0)}.$$

然后选取

$$\boldsymbol{p}_2 = \boldsymbol{A}\boldsymbol{p}_1 + \gamma_1^{(2)}\boldsymbol{p}_1 + \gamma_0^{(2)}\boldsymbol{p}_0.$$

利用 $\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2$ 的正交性, 分别用 $\boldsymbol{p}_1, \boldsymbol{p}_0$ 与上式两边作 $(\cdot,\cdot)_{\boldsymbol{A}^2}$ 内积, 得

$$\begin{aligned}\gamma_1^{(2)} &= -\frac{(\boldsymbol{A}\boldsymbol{p}_1, \boldsymbol{p}_1)_{\boldsymbol{A}^2}}{(\boldsymbol{p}_1, \boldsymbol{p}_1)_{\boldsymbol{A}^2}} = -\frac{(\boldsymbol{A}^2\boldsymbol{p}_1, \boldsymbol{A}\boldsymbol{p}_1)}{(\boldsymbol{A}\boldsymbol{p}_1, \boldsymbol{A}\boldsymbol{p}_1)},\\ \gamma_0^{(2)} &= -\frac{(\boldsymbol{A}\boldsymbol{p}_1, \boldsymbol{p}_0)_{\boldsymbol{A}^2}}{(\boldsymbol{p}_0, \boldsymbol{p}_0)_{\boldsymbol{A}^2}} = -\frac{(\boldsymbol{A}^2\boldsymbol{p}_1, \boldsymbol{A}\boldsymbol{p}_0)}{(\boldsymbol{A}\boldsymbol{p}_0, \boldsymbol{A}\boldsymbol{p}_0)}.\end{aligned}$$

同理, 令

$$\boldsymbol{p}_3 = \boldsymbol{A}\boldsymbol{p}_2 + \gamma_2^{(3)}\boldsymbol{p}_2 + \gamma_1^{(3)}\boldsymbol{p}_1 + \gamma_0^{(3)}\boldsymbol{p}_0.$$

分别用 $\boldsymbol{p}_2, \boldsymbol{p}_1, \boldsymbol{p}_0$ 与上式两边作 $(\cdot,\cdot)_{\boldsymbol{A}^2}$ 内积得

$$\begin{aligned}\gamma_2^{(3)} &= -\frac{(\boldsymbol{A}\boldsymbol{p}_2, \boldsymbol{p}_2)_{\boldsymbol{A}^2}}{(\boldsymbol{p}_2, \boldsymbol{p}_2)_{\boldsymbol{A}^2}} = -\frac{(\boldsymbol{A}^2\boldsymbol{p}_2, \boldsymbol{A}\boldsymbol{p}_2)}{(\boldsymbol{A}\boldsymbol{p}_2, \boldsymbol{A}\boldsymbol{p}_2)},\\ \gamma_1^{(3)} &= -\frac{(\boldsymbol{A}\boldsymbol{p}_2, \boldsymbol{p}_1)_{\boldsymbol{A}^2}}{(\boldsymbol{p}_1, \boldsymbol{p}_1)_{\boldsymbol{A}^2}} = -\frac{(\boldsymbol{A}^2\boldsymbol{p}_2, \boldsymbol{A}\boldsymbol{p}_1)}{(\boldsymbol{A}\boldsymbol{p}_1, \boldsymbol{A}\boldsymbol{p}_1)},\\ \gamma_0^{(3)} &= -\frac{(\boldsymbol{A}\boldsymbol{p}_2, \boldsymbol{p}_0)_{\boldsymbol{A}^2}}{(\boldsymbol{p}_0, \boldsymbol{p}_0)_{\boldsymbol{A}^2}} = -\frac{(\boldsymbol{p}_2, \boldsymbol{A}\boldsymbol{p}_0)_{\boldsymbol{A}^2}}{(\boldsymbol{p}_0, \boldsymbol{p}_0)_{\boldsymbol{A}^2}}\\ &= -\frac{(\boldsymbol{p}_2, \boldsymbol{p}_1 - \gamma_0^{(1)}\boldsymbol{p}_0)_{\boldsymbol{A}^2}}{(\boldsymbol{p}_0, \boldsymbol{p}_0)_{\boldsymbol{A}^2}} = 0.\end{aligned}$$

依此类推, 有

$$\begin{aligned}&\boldsymbol{p}_{i+1} = \boldsymbol{A}\boldsymbol{p}_i + \lambda_i\boldsymbol{p}_i + \mu_i\boldsymbol{p}_{i-1},\\ &\lambda_i = -\frac{(\boldsymbol{A}^2\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_i)}{(\boldsymbol{A}\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_i)}, \quad \mu_i = -\frac{(\boldsymbol{A}^2\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_{i-1})}{(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1})},\\ &\mu_0 = 0, \ i = 0, 1, 2, \cdots.\end{aligned}$$

综合上述, 即可得到形式与共轭梯度法类似的极小残量递推算法:

取 $\boldsymbol{p}_0 = \boldsymbol{r}_0$, $\mu_0 = 0$, 对 $k = 0, 1, 2, \cdots$ 进行下列计算直到满足终止条件:

$$\alpha_k = \frac{(\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{p}_k)}{(\boldsymbol{A}\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_k)}, \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k; \tag{4.121}$$

$$\lambda_k = -\frac{(\boldsymbol{A}^2\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_k)}{(\boldsymbol{A}\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_k)}, \quad \mu_k = -\frac{(\boldsymbol{A}^2\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_{k-1})}{(\boldsymbol{A}\boldsymbol{p}_{k-1}, \boldsymbol{A}\boldsymbol{p}_{k-1})},$$

$$\boldsymbol{p}_{k+1} = \boldsymbol{A}\boldsymbol{p}_k + \lambda_k\boldsymbol{p}_k + \mu_k\boldsymbol{p}_{k-1}. \tag{4.122}$$

上述算法每次迭代需要作两次矩阵与向量乘法 $(\boldsymbol{A}\boldsymbol{p}_k, \boldsymbol{A}(\boldsymbol{A}\boldsymbol{p}_k))$, 要存储四个向量 $\boldsymbol{x}_k, \boldsymbol{p}_k, \boldsymbol{p}_{k-1}, \boldsymbol{A}\boldsymbol{p}_k$, 计算量和存储量均为共轭梯度法的两倍.

回顾共轭梯度法中, 在求 $\boldsymbol{p}_{k+1}$ 时利用了 $\boldsymbol{r}_{k+1}$ 代替 $\boldsymbol{A}\boldsymbol{p}_k$ 来减少计算量和存储量. 在此, 也来分析一下 $\boldsymbol{r}_{k+1}$. 由

$$\left(\boldsymbol{x}^* - \boldsymbol{x}_0 - \boldsymbol{P}_k\boldsymbol{\alpha}_k, \boldsymbol{p}_i\right)_{\boldsymbol{A}^2} = 0, \quad i = 0, 1, \cdots, k-1,$$

可知

$$\left(\boldsymbol{x}^* - \boldsymbol{x}_k, \boldsymbol{p}_i\right)_{\boldsymbol{A}^2} = 0, \quad i = 0, 1, \cdots, k-1.$$

上式即

$$(\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_i) = (\boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x}_k), \boldsymbol{A}\boldsymbol{p}_i) = 0, \quad i = 0, 1, \cdots, k-1. \tag{4.123}$$

但

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k, \tag{4.124}$$

即

$$\boldsymbol{r}_k = \boldsymbol{r}_{k+1} + \alpha_k \boldsymbol{A}\boldsymbol{p}_k,$$

故有

$$\|\boldsymbol{r}_k\|_2^2 = \|\boldsymbol{r}_{k+1}\|_2^2 + \alpha_k^2 \|\boldsymbol{A}\boldsymbol{p}_k\|_2^2.$$

因此, 只要 $\alpha_k \|\boldsymbol{A}\boldsymbol{p}_k\|_2 \neq 0$, 就必有

$$\|\boldsymbol{r}_{k+1}\|_2 < \|\boldsymbol{r}_k\|_2.$$

由式 (4.122), 有

$$\begin{aligned}
\boldsymbol{p}_i &= \boldsymbol{A}\boldsymbol{p}_{i-1} + \lambda_{i-1}\boldsymbol{p}_{i-1} + \mu_{i-1}\boldsymbol{p}_{i-2} = \cdots \\
&= \boldsymbol{A}^i \boldsymbol{p}_0 + \sum_{s=0}^{i-1} \delta_s^{(i)} \boldsymbol{A}^s \boldsymbol{p}_0 = \boldsymbol{A}^i \boldsymbol{r}_0 + \sum_{s=0}^{i-1} \delta_s^{(i)} \boldsymbol{A}^s \boldsymbol{r}_0,
\end{aligned}$$

得

$$\begin{aligned}
\boldsymbol{r}_{i+1} &= \boldsymbol{r}_i - \alpha_i \boldsymbol{A}\boldsymbol{p}_i = \boldsymbol{r}_i - \alpha_i \boldsymbol{A}\Big(\boldsymbol{A}^i \boldsymbol{r}_0 + \sum_{s=0}^{i-1} \delta_s^{(i)} \boldsymbol{A}^s \boldsymbol{r}_0\Big) \\
&= \boldsymbol{r}_i - \alpha_i \sum_{s=0}^{i-1} \delta_s^{(i)} \boldsymbol{A}^{s+1} \boldsymbol{r}_0 - \alpha_i \boldsymbol{A}^{i+1} \boldsymbol{r}_0 \\
&= \cdots = \mathcal{P}_i(\boldsymbol{A}) \boldsymbol{r}_0 - \alpha_i \boldsymbol{A}^{i+1} \boldsymbol{r}_0,
\end{aligned}$$

式中: $\mathcal{P}_i(\boldsymbol{A})$ 为关于矩阵 $\boldsymbol{A}$ 的 i 次多项式. 由上式, 得

$$\left[\boldsymbol{r}_0, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_{k-1}\right] = \left[\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\right] \widetilde{\boldsymbol{R}}_k = \boldsymbol{S}_k \widetilde{\boldsymbol{R}}_k, \tag{4.125}$$

式中: $\widetilde{\boldsymbol{R}}_k$ 为 k 阶上三角矩阵, 且其对角元为 $1, -\alpha_0, -\alpha_1, \cdots, -\alpha_{k-2}$.

若构造关于内积 $(\cdot,\cdot)_{\boldsymbol{A}^2}$ 的正交化向量组 $\widetilde{\boldsymbol{p}}_0, \widetilde{\boldsymbol{p}}_1, \cdots, \widetilde{\boldsymbol{p}}_{k-1}$:

$$\begin{aligned}
&\widetilde{\boldsymbol{p}}_0 = \boldsymbol{r}_0, \\
&\widetilde{\boldsymbol{p}}_1 = \boldsymbol{r}_1 + \gamma_0^{(1)} \widetilde{\boldsymbol{p}}_0, \\
&\widetilde{\boldsymbol{p}}_2 = \boldsymbol{r}_2 + \gamma_1^{(2)} \widetilde{\boldsymbol{p}}_1 + \gamma_0^{(2)} \widetilde{\boldsymbol{p}}_0, \\
&\quad \vdots
\end{aligned}$$

$$\widetilde{\boldsymbol{p}}_i = \boldsymbol{r}_i + \sum_{s=0}^{i-1} \gamma_s^{(i)} \widetilde{\boldsymbol{p}}_s = \boldsymbol{r}_i + \gamma_{i-1}^{(i)} \widetilde{\boldsymbol{p}}_{i-1} + \cdots + \gamma_1^{(i)} \widetilde{\boldsymbol{p}}_1 + \gamma_0^{(i)} \widetilde{\boldsymbol{p}}_0,$$

式中:

$$\gamma_s^{(i)} = -\frac{\left(\boldsymbol{r}_i, \widetilde{\boldsymbol{p}}_s\right)_{\boldsymbol{A}^2}}{\left(\widetilde{\boldsymbol{p}}_s, \widetilde{\boldsymbol{p}}_s\right)_{\boldsymbol{A}^2}}, \quad i = 0, 1, \cdots, k-1; \ s = 0, 1, \cdots, i-1.$$

不失一般性, 设 $\widetilde{\boldsymbol{p}}_i = \omega_i \boldsymbol{p}_i$, 利用 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, 有

$$\begin{aligned}
\left(\boldsymbol{r}_i, \widetilde{\boldsymbol{p}}_s\right)_{\boldsymbol{A}^2} &= \left(\boldsymbol{A}\boldsymbol{r}_i, \boldsymbol{A}\widetilde{\boldsymbol{p}}_s\right) = \left(\boldsymbol{A}\boldsymbol{r}_i, \omega_s \boldsymbol{A}\boldsymbol{p}_s\right) \\
&= \left(\boldsymbol{A}\boldsymbol{r}_i, \omega_s(\boldsymbol{p}_{s+1} - \lambda_s \boldsymbol{p}_s - \mu_s \boldsymbol{p}_{s-1}\right) \\
&= \left(\boldsymbol{r}_i, \omega_s \boldsymbol{A}\boldsymbol{p}_{s+1} - \omega_s \lambda_s \boldsymbol{A}\boldsymbol{p}_s - \omega_s \mu_s \boldsymbol{A}\boldsymbol{p}_{s-1}\right).
\end{aligned}$$

由上式及式 (4.123), 不难发现, 当 $s+1 \leqslant i-1$, 即 $s \leqslant i-2$ 时, 有

$$\left(\boldsymbol{r}_i, \widetilde{\boldsymbol{p}}_s\right)_{\boldsymbol{A}^2} = 0,$$

即

$$\gamma_s^{(i)} = 0, \ s \leqslant i-2, \ i = 0, 1, \cdots, k-1.$$

于是有

$$\widetilde{\boldsymbol{p}}_i = \boldsymbol{r}_i + \gamma_{i-1}^{(i)} \widetilde{\boldsymbol{p}}_{i-1}, \ i = 0, 1, \cdots, k-1, \ \widetilde{\boldsymbol{p}}_{-1} = \boldsymbol{0}.$$

回顾式 (4.122), $\boldsymbol{p}_i$ 与 $\boldsymbol{A}\boldsymbol{p}_{i-1}$, $\boldsymbol{p}_{i-1}$, $\boldsymbol{p}_{i-2}$ 有关, 可以看到, $\widetilde{\boldsymbol{p}}_i$ 只与 $\boldsymbol{r}_i$ 和 $\widetilde{\boldsymbol{p}}_{i-1}$ 有关. 现记 $\beta_{i-1} := \gamma_{i-1}^{(i)}$, 在不混淆的情况下, 再记 $\boldsymbol{p}_i := \widetilde{\boldsymbol{p}}_i$, 则有

$$\begin{aligned}
&\boldsymbol{p}_i = \boldsymbol{r}_i + \beta_{i-1} \boldsymbol{p}_{i-1}, \ i = 1, 2, \cdots, k-1, \\
&\beta_{i-1} = \frac{\left(\boldsymbol{r}_i, \boldsymbol{p}_{i-1}\right)_{\boldsymbol{A}^2}}{\left(\boldsymbol{p}_{i-1}, \boldsymbol{p}_{i-1}\right)_{\boldsymbol{A}^2}}.
\end{aligned}$$

注意到, 对这一组 $\widetilde{\boldsymbol{p}}_0, \widetilde{\boldsymbol{p}}_2, \cdots, \widetilde{\boldsymbol{p}}_{k-1}$ (已记为 $\boldsymbol{p}_0, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{k-1}$), 也有

$$\left[\boldsymbol{p}_0, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{k-1}\right] = \left[\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\right] \widehat{\boldsymbol{R}}_k = \boldsymbol{S}_k \widehat{\boldsymbol{R}}_k, \tag{4.126}$$

式中: $\widehat{\boldsymbol{R}}_k$ 为上三角矩阵, 对角元也为 $1, -\alpha_0, -\alpha_1, \cdots, -\alpha_{k-2}$. 由式 (4.125) 和式 (4.126), 得

$$\left[\boldsymbol{r}_0, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_{k-1}\right] = \left[\boldsymbol{p}_0, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{k-1}\right] \widehat{\boldsymbol{R}}_k^{-1} \widetilde{\boldsymbol{R}}_k,$$

式中: $\widehat{\boldsymbol{R}}_k^{-1} \widetilde{\boldsymbol{R}}_k$ 仍为上三角矩阵, 且对角元均为 1. 于是有

$$\boldsymbol{r}_i = \boldsymbol{p}_i + \sum_{s=0}^{i-1} c_s \boldsymbol{p}_s,$$

及

$$\beta_{i-1} = \frac{\left(\boldsymbol{r}_i, \boldsymbol{p}_{i-1}\right)_{\boldsymbol{A}^2}}{\left(\boldsymbol{p}_{i-1}, \boldsymbol{p}_{i-1}\right)_{\boldsymbol{A}^2}} = \frac{\left(\boldsymbol{A}\boldsymbol{r}_i, \boldsymbol{A}\boldsymbol{p}_{i-1}\right)}{\left(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1}\right)} = \frac{\left(\boldsymbol{A}\boldsymbol{r}_i, \frac{\boldsymbol{r}_i - \boldsymbol{r}_{i-1}}{\alpha_{i-1}}\right)}{\left(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1}\right)}.$$

由于

$$(\boldsymbol{A}\boldsymbol{r}_i, \boldsymbol{r}_{i-1}) = \Big(\boldsymbol{r}_i, \boldsymbol{A}\Big(\boldsymbol{p}_{i-1} + \sum_{s=0}^{i-2} c_s \boldsymbol{p}_s\Big)\Big) = 0,$$

因此

$$\beta_{i-1} = \frac{(\boldsymbol{A}\boldsymbol{r}_i, \boldsymbol{r}_i)}{\alpha_{i-1}(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1})}. \tag{4.127}$$

另外, 由 $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \alpha_{i-1}\boldsymbol{p}_{i-1}$ 及式 (4.123), 得

$$\begin{aligned}
\alpha_{i-1} &= \frac{(\boldsymbol{x}_i - \boldsymbol{x}_{i-1}, \boldsymbol{p}_{i-1})_{\boldsymbol{A}^2}}{(\boldsymbol{p}_{i-1}, \boldsymbol{p}_{i-1})_{\boldsymbol{A}^2}} = \frac{(\boldsymbol{A}(\boldsymbol{x}_i - \boldsymbol{x}_{i-1}), \boldsymbol{A}\boldsymbol{p}_{i-1})}{(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1})} \\
&= \frac{(\boldsymbol{r}_{i-1} - \boldsymbol{r}_i, \boldsymbol{A}\boldsymbol{p}_{i-1})}{(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1})} = \frac{(\boldsymbol{r}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1})}{(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1})} \\
&= \frac{(\boldsymbol{A}\boldsymbol{r}_{i-1}, \boldsymbol{p}_{i-1})}{(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1})} = \frac{(\boldsymbol{A}\boldsymbol{r}_{i-1}, \boldsymbol{r}_{i-1} + \beta_{i-2}\boldsymbol{p}_{i-2})}{(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1})} \\
&= \frac{(\boldsymbol{A}\boldsymbol{r}_{i-1}, \boldsymbol{r}_{i-1})}{(\boldsymbol{A}\boldsymbol{p}_{i-1}, \boldsymbol{A}\boldsymbol{p}_{i-1})}.
\end{aligned} \tag{4.128}$$

将式 (4.128) 代入式 (4.127), 得

$$\beta_{i-1} = \frac{(\boldsymbol{A}\boldsymbol{r}_i, \boldsymbol{r}_i)}{(\boldsymbol{A}\boldsymbol{r}_{i-1}, \boldsymbol{r}_{i-1})}.$$

于是可以得到极小残量法另一种形式的递推算法:

算法 4.10 (递推形式的 MINRES 方法) 给定 n 阶非奇异的实对称矩阵 $\boldsymbol{A}$, n 维向量 $\boldsymbol{b}$ 和允许误差 $\varepsilon > 0$. 本算法计算近似解向量 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{p}_0 = \boldsymbol{r}_0$;

for $k = 0, 1, \cdots,$

$\alpha_k = \dfrac{(\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_k)}{(\boldsymbol{A}\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_k)}$; $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$;

$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k$; $\beta_k = \dfrac{(\boldsymbol{A}\boldsymbol{r}_{k+1}, \boldsymbol{r}_{k+1})}{(\boldsymbol{A}\boldsymbol{r}_k, \boldsymbol{r}_k)}$;

$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k$; $\boldsymbol{A}\boldsymbol{p}_{k+1} = \boldsymbol{A}\boldsymbol{r}_{k+1} + \beta_k \boldsymbol{A}\boldsymbol{p}_k$;

end

从形式上看, 算法 4.10 每一步迭代也需要计算两次矩阵与向量的乘法 $\boldsymbol{A}\boldsymbol{p}_k$ 和 $\boldsymbol{A}\boldsymbol{r}_{k+1}$, 实际上可从 $\boldsymbol{A}\boldsymbol{p}_{k+1} = \boldsymbol{A}\boldsymbol{r}_{k+1} + \beta_k \boldsymbol{A}\boldsymbol{p}_k$ 来计算 $\boldsymbol{A}\boldsymbol{p}_{k+1}$, 这样就只需要计算一次矩阵与向量的乘法. 此外, 在计算中需要保存五个向量 $\boldsymbol{x}_k$, $\boldsymbol{p}_k$, $\boldsymbol{r}_k$, $\boldsymbol{A}\boldsymbol{p}_k$ 和 $\boldsymbol{A}\boldsymbol{r}_k$. 递推形式的 MINRES 方法的 MATLAB 程序如下:

```
%递推形式的极小残量法-minres1.m
function [x,iter,time,res,resvec]=minres1(A,b,x,max_it,tol)
%极小残量法求解对称不定方程组Ax=b
tic; r=b-A*x; p=r; mr=norm(r);
u=A*p; z=A*r; iter=1;
while (iter<max_it)
    alpha=(r'*u)/(u'*u);
    x=x+alpha*p; r1=r-alpha*u;
    z1=A*r1; beta=(z1'*r1)/(z'*r);
    p=r1+beta*p; u=z1+beta*u;
    res=norm(r1)/mr; resvec(iter)=res;
    if (res<tol), break; end
    r=r1; z=z1; iter=iter+1;
end
time=toc;
```

现在考虑预处理的极小残量法. 选择一个适当的对称正定矩阵 $\boldsymbol{M}$, 设 $\boldsymbol{M}$ 有分解 $\boldsymbol{M}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$, 则方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 经过预处理后变为

$$\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}}=\widetilde{\boldsymbol{b}}, \tag{4.129}$$

式中:

$$\widetilde{\boldsymbol{A}}=\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}},\ \widetilde{\boldsymbol{x}}=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x},\ \widetilde{\boldsymbol{b}}=\boldsymbol{L}^{-1}\boldsymbol{b}.$$

方程组 (4.129) 的系数矩阵 $\widetilde{\boldsymbol{A}}=\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}$ 仍为对称不定矩阵. 通过 $\boldsymbol{M}$ 和 $\boldsymbol{L}$ 的选取, 可以使得 $\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}$ 比 $\boldsymbol{A}$ 有更好的条件数. 现在对方程组 (4.129) 使用极小残量法, 得

$$\begin{aligned}
&\widetilde{\boldsymbol{r}}_0=\widetilde{\boldsymbol{b}}-\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}}_0,\ \widetilde{\boldsymbol{p}}_0=\widetilde{\boldsymbol{r}}_0,\\
&\alpha_k=\frac{(\widetilde{\boldsymbol{r}}_k,\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{p}}_k)}{(\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{p}}_k,\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{p}}_k)},\quad \widetilde{\boldsymbol{x}}_{k+1}=\widetilde{\boldsymbol{x}}_k+\alpha_k\widetilde{\boldsymbol{p}}_k,\\
&\widetilde{\boldsymbol{r}}_{k+1}=\widetilde{\boldsymbol{r}}_k-\alpha_k\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{p}}_k,\quad \beta_k=\frac{(\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{r}}_{k+1},\widetilde{\boldsymbol{r}}_{k+1})}{(\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{r}}_k,\widetilde{\boldsymbol{r}}_k)},\\
&\widetilde{\boldsymbol{p}}_{k+1}=\widetilde{\boldsymbol{r}}_{k+1}+\beta_k\widetilde{\boldsymbol{p}}_k,\quad k=0,1,\cdots.
\end{aligned}$$

为了在公式中使用原来变量, 作变换: $\boldsymbol{x}_k=\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{x}}_k$, $\boldsymbol{p}_k=\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{p}}_k$, 则 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k=\boldsymbol{L}\widetilde{\boldsymbol{r}}_k$.

$$\alpha_k=\frac{(\boldsymbol{L}^{-1}\boldsymbol{r}_k,(\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}})\boldsymbol{L}^{\mathrm{T}}\boldsymbol{p}_k)}{(\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{p}}_k,\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{p}}_k)}=\frac{(\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{L}^{-1}\boldsymbol{r}_k,\boldsymbol{A}\boldsymbol{p}_k)}{(\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{p}_k,\boldsymbol{A}\boldsymbol{p}_k)}=\frac{(\boldsymbol{M}^{-1}\boldsymbol{r}_k,\boldsymbol{A}\boldsymbol{p}_k)}{(\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{p}_k,\boldsymbol{A}\boldsymbol{p}_k)},$$

$$\widetilde{\boldsymbol{x}}_{k+1}=\widetilde{\boldsymbol{x}}_k+\alpha_k\widetilde{\boldsymbol{p}}_k\Longrightarrow\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}_{k+1}=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}_k+\alpha_k\boldsymbol{L}^{\mathrm{T}}\boldsymbol{p}_k\Longrightarrow\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\alpha_k\boldsymbol{p}_k,$$

$$\boldsymbol{L}^{-1}\boldsymbol{r}_{k+1}=\boldsymbol{L}^{-1}\boldsymbol{r}_k-\alpha_k(\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}})\boldsymbol{L}^{\mathrm{T}}\boldsymbol{p}_k\Longrightarrow\boldsymbol{r}_{k+1}=\boldsymbol{r}_k-\alpha_k\boldsymbol{A}\boldsymbol{p}_k,$$

$$\beta_k=\frac{((\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}})\widetilde{\boldsymbol{r}}_{k+1},\widetilde{\boldsymbol{r}}_{k+1})}{((\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}})\widetilde{\boldsymbol{r}}_k,\widetilde{\boldsymbol{r}}_k)}=\frac{(\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{r}}_{k+1},\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{r}}_{k+1})}{(\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{r}}_k,\boldsymbol{L}^{-\mathrm{T}}\widetilde{\boldsymbol{r}}_k)}$$

$$=\frac{(\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{L}^{-1}\boldsymbol{r}_{k+1},\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{L}^{-1}\boldsymbol{r}_{k+1})}{(\boldsymbol{A}\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{L}^{-1}\boldsymbol{r}_k,\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{L}^{-1}\boldsymbol{r}_k)}=\frac{(\boldsymbol{A}(\boldsymbol{M}^{-1}\boldsymbol{r}_{k+1}),\boldsymbol{M}^{-1}\boldsymbol{r}_{k+1})}{(\boldsymbol{A}(\boldsymbol{M}^{-1}\boldsymbol{r}_k),\boldsymbol{M}^{-1}\boldsymbol{r}_k)},$$

$$\widetilde{\boldsymbol{p}}_{k+1}=\widetilde{\boldsymbol{r}}_{k+1}+\beta_k\widetilde{\boldsymbol{p}}_k\Longrightarrow\boldsymbol{L}^{\mathrm{T}}\boldsymbol{p}_{k+1}=\boldsymbol{L}^{-1}\boldsymbol{r}_{k+1}+\beta_k\boldsymbol{L}^{\mathrm{T}}\boldsymbol{p}_k$$
$$\Longrightarrow\boldsymbol{p}_{k+1}=\boldsymbol{L}^{-\mathrm{T}}\boldsymbol{L}^{-1}\boldsymbol{r}_{k+1}+\beta_k\boldsymbol{p}_k\Longrightarrow\boldsymbol{p}_{k+1}=\boldsymbol{M}^{-1}\boldsymbol{r}_{k+1}+\beta_k\boldsymbol{p}_k.$$

这样, 就得到了预处理极小残量法的递推算法, 具体步骤为:

算法 4.11 (PMINRES 方法) 给定 n 阶非奇异的实对称矩阵 $\boldsymbol{A}$, n 维向量 $\boldsymbol{b}$, 允许误差 $\varepsilon>0$ 和预处理矩阵 $\boldsymbol{M}$(对称正定). 本算法计算近似解向量 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2\leqslant\varepsilon$, 其中 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0\in\mathbb{R}^n$; $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{z}_0=\boldsymbol{M}^{-1}\boldsymbol{r}_0$;

$\boldsymbol{p}_0=\boldsymbol{z}_0$; $\boldsymbol{u}_0=\boldsymbol{A}\boldsymbol{p}_0$; $\boldsymbol{w}_0=\boldsymbol{M}^{-1}\boldsymbol{u}_0$; $k=0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2>\varepsilon)$

$\alpha_k=\dfrac{(\boldsymbol{z}_k,\boldsymbol{u}_k)}{(\boldsymbol{w}_k,\boldsymbol{u}_k)}$; $\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\alpha_k\boldsymbol{p}_k$;

$\boldsymbol{r}_{k+1}=\boldsymbol{r}_k-\alpha_k\boldsymbol{u}_k$; $\boldsymbol{z}_{k+1}=\boldsymbol{M}^{-1}\boldsymbol{r}_k$;

$\beta_k=\dfrac{(\boldsymbol{A}\boldsymbol{z}_{k+1},\boldsymbol{z}_{k+1})}{(\boldsymbol{A}\boldsymbol{z}_k,\boldsymbol{z}_k)}$; $\boldsymbol{p}_{k+1}=\boldsymbol{z}_{k+1}+\beta_k\boldsymbol{p}_k$;

$\boldsymbol{u}_{k+1}=\boldsymbol{A}\boldsymbol{p}_{k+1}=\boldsymbol{A}\boldsymbol{z}_{k+1}+\beta_k\boldsymbol{u}_k$; $\boldsymbol{w}_{k+1}=\boldsymbol{M}^{-1}\boldsymbol{u}_{k+1}$;

$k=k+1$;

end

注 4.4 算法 4.11 仅与预处理子 $\boldsymbol{M}$ 有关, 而与 $\boldsymbol{L}$ 无关. 当 $\boldsymbol{M}$ 为单位阵时, 它就是没有经过预处理的极小残量法. 与共轭梯度法相比较, 预处理极小残量法增加了计算 $\boldsymbol{z}_{k+1}$ 和 $\boldsymbol{w}_{k+1}$ 的工作量, 这可以通过求解两个系数矩阵都是 $\boldsymbol{M}$ 的方程组来实现, 即每一步需要求解 $\boldsymbol{M}\boldsymbol{z}=\boldsymbol{r}_{k+1}$ 和 $\boldsymbol{M}\boldsymbol{w}=\boldsymbol{A}\boldsymbol{p}_{k+1}$ 来得到 $\boldsymbol{z}_{k+1}$ 和 $\boldsymbol{w}_{k+1}$. 尽管如此, 若预处理子 $\boldsymbol{M}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 是通过某种矩阵分解 (比如不完全 Cholesky 分解) 得到的, 此时 $\boldsymbol{L}$ 是下三角矩阵, 则 $\boldsymbol{M}\boldsymbol{z}=\boldsymbol{r}_{k+1}$ 可转化为求解两个三角形方程组 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{r}_{k+1}$ 和 $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{z}=\boldsymbol{y}$, 后者的运算量要比前者少得多.

PMINRES 方法的 MATLAB 程序如下:

```
%PMINRES方法程序-pminres.m
function [x,iter,time,res,resvec]=pminres1(A,b,x,M1,M2,max_it,tol)
%PMINRES方法求解对称不定方程组Ax=b, 预处理子M=M1*M2
```

```
tic; n=length(b);
r=b-A*x; z=M2\(M1\r); p=z; mr=norm(r);
u=A*p; v=u; w=M2\(M1\u); iter=1;
while (iter<max_it)
    alpha=(z'*u)/(w'*u); x=x+alpha*p;
    r=r-alpha*u; z1=M2\(M1\r); v1=A*z1;
    beta=(v1'*z1)/(v'*z); p=z1+beta*p;
    u=v1+beta*u; w=M2\(M1\u);
    res=norm(r)/mr; resvec(iter)=res;
    if (res<tol), break; end
    z=z1; v=v1; iter=iter+1;
end
time=toc;
```

例 4.9 仍考虑例 4.8 中的线性方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 取预处理矩阵 $\boldsymbol{M}\approx\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 为 $\boldsymbol{A}-\mu\boldsymbol{I}$ 的不完全 Cholesky 分解, 其中 $\mu=-1.0$ 是矩阵 $\boldsymbol{A}$ 的最小特征值的一个下界估计. 应用算法 4.11 到该线性方程组上, 第 79 步得到的 $\widehat{\boldsymbol{x}}=\boldsymbol{x}_{79}$ 满足

$$\|\widehat{\boldsymbol{x}}-\boldsymbol{x}^*\|_2=3.2414\times10^{-8}.$$

比较极小残量法和预处理极小残量法的数值表现, 迭代过程的收敛轨迹如图 4.10 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

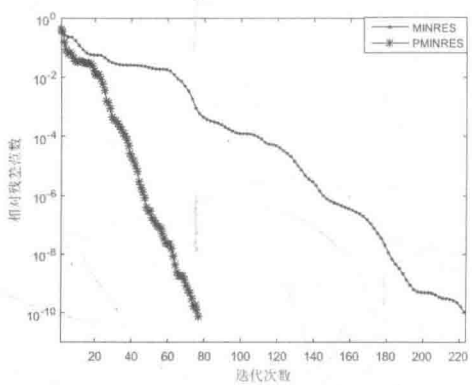

图 4.10 PMINRES 方法与 MINRES 方法的比较

相比于 MINRES 方法, PMINRES 方法尽管迭代次数得到了显著的下降, 从计算时间上看似乎没有优势, 原因可能是后者每一步需要求解两个以预处理子 $\boldsymbol{M}$ 为系数矩阵的线性方程组.

4.3.3 收敛性分析

与共轭梯度法一样, 虽然从理论上讲极小残量法是一种直接方法, 但实际使用时是把它作为迭代法来使用的. 因此, 下面简要说明其收敛特性.

由 Krylov 子空间的性质, $\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$ 的充分必要条件是存在 $\varphi \in \mathcal{P}_{k-1}$ 使得 $\boldsymbol{x} = \boldsymbol{x}_0 + \varphi(\boldsymbol{A})\boldsymbol{r}_0$. 注意到 $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$, $\boldsymbol{x}^* = \boldsymbol{A}^{-1}\boldsymbol{b}$, 因此, 有

$$\boldsymbol{x}^* - \boldsymbol{x}_k = \boldsymbol{A}^{-1}\boldsymbol{b} - \boldsymbol{x}_0 - \varphi(\boldsymbol{A})\boldsymbol{r}_0 = \boldsymbol{A}^{-1}(\boldsymbol{I} - \boldsymbol{A}\varphi(\boldsymbol{A}))\boldsymbol{r}_0 = \boldsymbol{A}^{-1}\psi(\boldsymbol{A})\boldsymbol{r}_0,$$

式中: $\psi(t) = 1 - t\varphi(t)$ 满足 $\psi(0) = 1$. 设 $\boldsymbol{A}$ 的谱分解为 $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$, 其中 $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n]$ 是正交矩阵, $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$, $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$. 现将 $\boldsymbol{r}_0$ 按 $\boldsymbol{A}$ 的特征向量展开, 有

$$\boldsymbol{r}_0 = \eta_1\boldsymbol{v}_1 + \eta_2\boldsymbol{v}_2 + \cdots + \eta_n\boldsymbol{v}_n = \sum_{i=1}^{n}\eta_i\boldsymbol{v}_i.$$

于是, 由式 (4.120), 有

$$\begin{aligned}
\|\boldsymbol{r}_k\|_2^2 &= \big(\boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x}_k), \boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x}_k)\big) = \big(\psi(\boldsymbol{A})\boldsymbol{r}_0, \psi(\boldsymbol{A})\boldsymbol{r}_0\big) \\
&= \left(\sum_{i=1}^{n}\eta_i\psi(\boldsymbol{A})\boldsymbol{v}_i, \sum_{j=1}^{n}\eta_j\psi(\boldsymbol{A})\boldsymbol{v}_j\right) \\
&= \left(\sum_{i=1}^{n}\eta_i\psi(\lambda_i)\boldsymbol{v}_i, \sum_{j=1}^{n}\eta_r\psi(\lambda_j)\boldsymbol{v}_j\right) \\
&= \sum_{i=1}^{n}\eta_i^2[\psi(\lambda_i)]^2 \leqslant \max_{1\leqslant i\leqslant n}[\psi(\lambda_i)]^2\sum_{i=1}^{n}\eta_i^2 \\
&= \max_{1\leqslant i\leqslant n}[\psi(\lambda_i)]^2\|\boldsymbol{r}_0\|_2^2.
\end{aligned}$$

由此可得

$$\|\boldsymbol{r}_k\|_2 \leqslant \min_{\psi\in\mathcal{P}_k^0}\max_{\lambda\in\lambda(\boldsymbol{A})}|\psi(\lambda)| \cdot \|\boldsymbol{r}_0\|_2, \tag{4.130}$$

式中: 集合 $\mathcal{P}_k^0$ 的定义为

$$\mathcal{P}_k^0 = \{\psi \in \mathcal{P}_k : \psi(0) = 1\}.$$

令

$$a = \min_{\lambda\in\lambda(\boldsymbol{A})}|\lambda|, \quad b = \max_{\lambda\in\lambda(\boldsymbol{A})}|\lambda|, \tag{4.131}$$

即 a 和 b 分别是 $\boldsymbol{A}$ 的特征值的最小模和最大模. 由 $\boldsymbol{A}$ 是非奇异的假定可知 $0 < a \leqslant b$. 下面假定 $a < b$, 当 $k = 2m$ 时, 特别取一个多项式

$$\widehat{\psi}(t) = C_m\left(\frac{b^2 + a^2 - 2t^2}{b^2 - a^2}\right)/C_m\left(\frac{b^2 + a^2}{b^2 - a^2}\right), \tag{4.132}$$

式中: C_m 为 m 次 Chebyshev 多项式, 即

$$C_m(t) = \frac{(t + \sqrt{t^2 - 1})^m + (t - \sqrt{t^2 - 1})^m}{2},$$

则 $\widehat{\psi}(t)$ 是一个 $2m$ 次多项式, 且 $\widehat{\psi}(0)=1$, 从而 $\widehat{\psi}\in\mathcal{P}_k^0$. 由 Chebyshev 逼近定理, $\widehat{\psi}(t)$ 在区间 $[-b,-a]\cup[a,b]$ 上均有

$$|\widehat{\psi}(t)|\leqslant 1/C_m\Big(1+\frac{2}{\kappa^2-1}\Big),\quad \forall\, t\in[-b,-a]\cup[a,b].$$

这样, 由式 (4.130) 即得

$$\frac{\|\boldsymbol{r}_{2m}\|_2}{\|\boldsymbol{r}_0\|_2}\leqslant\max_{\lambda\in\lambda(\boldsymbol{A})}|\widehat{\psi}(t)|=1/C_m\Big(\frac{b^2+a^2}{b^2-a^2}\Big)=1/C_m\Big(1+\frac{2}{\kappa^2-1}\Big),$$

式中: $\kappa=\|\boldsymbol{A}\|_2\|\boldsymbol{A}^{-1}\|_2=b/a$. 记 $\theta(\kappa)=1+\dfrac{2}{\kappa^2-1}$, 因 $\|\boldsymbol{r}_{2m+1}\|_2\leqslant\|\boldsymbol{r}_{2m}\|_2$, 故有

$$\frac{\|\boldsymbol{r}_k\|_2}{\|\boldsymbol{r}_0\|_2}\leqslant\frac{1}{C_{[k/2]}(\theta(\kappa))}.\tag{4.133}$$

当然式 (4.133) 的估计是十分粗糙的, 但也说明了极小残量法收敛得快慢与 $\boldsymbol{A}$ 的条件数有关. 事实上, 更精细的分析可以证明, 与共轭梯度法类似, 极小残量法的收敛快慢也与 $\boldsymbol{A}$ 的谱分布有关. 如果 $\boldsymbol{A}$ 的特征值除去少数几个外大多数集中在某数的附近, 则极小残量法将收敛得很快. 因此实际使用时也是将其与预处理技术相结合来使用的.

4.4 SYMMLQ 方法

SYMMLQ 方法与 MINRES 方法类似, 也是求解对称不定方程组的有效算法之一. 本节介绍这一算法的基本思想和导出过程, 并介绍与之关系密切的极小误差法, 以及这两个方法的收敛性及误差估计定理.

4.4.1 SYMMLQ 方法

对于对称不定方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},\tag{4.134}$$

令 $\boldsymbol{z}=\boldsymbol{x}-\boldsymbol{x}_0$, 则式 (4.134) 可改写为

$$\boldsymbol{A}\boldsymbol{z}=\boldsymbol{r}_0,\quad \boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0.$$

若

$$\mathcal{S}_k=\mathrm{span}\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k\}$$

张成一个 k 维子空间. 考虑二次泛函

$$J_k(\boldsymbol{z}_k)=(\boldsymbol{z}-\boldsymbol{z}_k)^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{z}-\boldsymbol{z}_k),\tag{4.135}$$

式中:

$$\boldsymbol{z}_k=\boldsymbol{V}_k\boldsymbol{y}_k,\quad \boldsymbol{V}_k=\big[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k\big].$$

那么二次泛函

$$J_k(\boldsymbol{z}_k)=\widetilde{J}_k(\boldsymbol{y}_k)=(\boldsymbol{A}^{-1}\boldsymbol{r}_0-\boldsymbol{V}_k\boldsymbol{y}_k)^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{A}^{-1}\boldsymbol{r}_0-\boldsymbol{V}_k\boldsymbol{y}_k)$$

的驻点 $\boldsymbol{y}_k$ 满足

$$\boldsymbol{V}_k^{\mathrm{T}}(\boldsymbol{r}_0-\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_k)=\boldsymbol{0},$$

或者

$$\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_k=\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{r}_0,\quad \boldsymbol{x}_k=\boldsymbol{x}_0+\boldsymbol{z}_k=\boldsymbol{x}_k+\boldsymbol{V}_k\boldsymbol{y}_k. \tag{4.136}$$

若取 $\boldsymbol{v}_1=\boldsymbol{r}_0/\|\boldsymbol{r}_0\|_2$, 则用对称 Lanczos 算法可得到 k 次 Krylov 子空间

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\mathrm{span}\{\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{r}_0,\cdots,\boldsymbol{A}^{k-1}\boldsymbol{r}_0\}$$

的正交规范基 $\{\boldsymbol{v}_i\}_{i=1}^k$.

假设已经得到了一个长度为 k 的 Lanczos 分解 (2.36):

$$\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_k\boldsymbol{T}_k+\beta_{k+1}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}},\quad \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k=\boldsymbol{I}_k,\quad \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{v}_{k+1}=\boldsymbol{0}, \tag{4.137}$$

式中: 三对角矩阵 $\boldsymbol{T}_k$ 通常记为

$$\boldsymbol{T}_k\equiv\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k=\begin{bmatrix}\alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} & \beta_k \\ & & & \beta_k & \alpha_k\end{bmatrix}.$$

将上式代入式 (4.136) 则得到 Galerkin 方程组

$$\boldsymbol{T}_k\boldsymbol{y}_k=\beta_1\boldsymbol{e}_1,\quad \boldsymbol{x}_k=\boldsymbol{x}_0+\boldsymbol{V}_k\boldsymbol{y}_k, \tag{4.138}$$

式中: $\beta_1=\|\boldsymbol{r}_0\|_2$.

因为 $\boldsymbol{A}$ 对称不定, 所以三对角矩阵 $\boldsymbol{T}_k$ 也对称不定. 对这样的 $\boldsymbol{T}_k$ 进行 Cholesky 分解可能失效. 因此要寻找 $\boldsymbol{T}_k$ 更为稳定的分解, 即正交三角分解 (LQ 分解):

$$\boldsymbol{T}_k=\widetilde{\boldsymbol{L}}_k\boldsymbol{Q}_k,\ \boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{Q}_k=\boldsymbol{I}_k, \tag{4.139}$$

式中: $\widetilde{\boldsymbol{L}}_k$ 为下三角矩阵.

利用 LQ 分解式 (4.139) 来求解式 (4.138). 和前面的做法一样, 不是直接求出 $\boldsymbol{y}_k$, 而是寻找求 $\boldsymbol{x}_k$ 递推公式.

因 $\boldsymbol{T}_k$ 是对称三对角矩阵, $\boldsymbol{Q}_k$ 可写为乘积

$$\boldsymbol{Q}_k=\boldsymbol{G}_{k-1,k}\cdots\boldsymbol{G}_{2,3}\boldsymbol{G}_{1,2}, \tag{4.140}$$

式中: $\boldsymbol{G}_{i,i+1}\,(i=1,2,\cdots,k-1)$ 为 $(i,i+1)$ Givens 矩阵, 它与单位矩阵不同的是其 i 和 $i+1$ 行和列组成的 2 阶矩阵是

$$\boldsymbol{G}_{i,i+1}=\begin{bmatrix}c_i & s_i \\ s_i & -c_i\end{bmatrix},\quad c_i^2+s_i^2=1.$$

将式 (4.139) 代入式 (4.138), 得

$$\widetilde{\boldsymbol{L}}_k(\boldsymbol{Q}_k\boldsymbol{y}_k)=\beta_1\boldsymbol{e}_1,\ \boldsymbol{x}_k=\boldsymbol{x}_0+\boldsymbol{V}_k\boldsymbol{Q}_k^{\mathrm{T}}(\boldsymbol{Q}_k\boldsymbol{y}_k).$$

令

$$\widetilde{\boldsymbol{W}}_k=\boldsymbol{V}_k\boldsymbol{Q}_k^{\mathrm{T}},\ \widetilde{\boldsymbol{z}}_k=\boldsymbol{Q}_k\boldsymbol{y}_k, \tag{4.141}$$

则得到方程组和迭代向量公式

$$\widetilde{\boldsymbol{L}}_k\widetilde{\boldsymbol{z}}_k=\beta_1\boldsymbol{e}_1,\ \boldsymbol{x}_k=\boldsymbol{x}_0+\widetilde{\boldsymbol{W}}_k\widetilde{\boldsymbol{z}}_k. \tag{4.142}$$

由 $\boldsymbol{Q}_k$ 的特殊形式 (4.140), $\widetilde{\boldsymbol{W}}_k$ 和 $\widetilde{\boldsymbol{z}}_k$ 可表示为

$$\widetilde{\boldsymbol{W}}_k=\left[\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_{k-1},\widetilde{\boldsymbol{w}}_k\right], \tag{4.143}$$

$$\widetilde{\boldsymbol{z}}_k=\left[\zeta_1,\zeta_2,\cdots,\zeta_{k-1},\widetilde{\zeta}_k\right]^{\mathrm{T}}. \tag{4.144}$$

而 LQ 分解式 (4.139) 中的 $\widetilde{\boldsymbol{L}}_k$ 和 $\boldsymbol{G}_{k,k+1}$ 的元素 c_k 和 s_k 可计算如下.

由式 (4.139), 公式

$$\widetilde{\boldsymbol{L}}_k=\boldsymbol{T}_k\boldsymbol{Q}_k^{\mathrm{T}}=\boldsymbol{T}_k\boldsymbol{G}_{1,2}\boldsymbol{G}_{2,3}\cdots\boldsymbol{G}_{k,k+1}$$

可表示为

$$\widetilde{\boldsymbol{L}}_k=\begin{bmatrix}\nu_1 & & & & \\ \delta_2 & \nu_2 & & & \\ \eta_3 & \delta_3 & \nu_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \eta_k & \delta_k & \widetilde{\nu}_k\end{bmatrix}. \tag{4.145}$$

现观察 $\widetilde{\boldsymbol{L}}_{k+1}$, 有

$$\widetilde{\boldsymbol{L}}_{k+1}=\boldsymbol{T}_{k+1}\boldsymbol{Q}_{k+1}^{\mathrm{T}}=\boldsymbol{T}_{k+1}\left[\begin{array}{c|c}\boldsymbol{Q}_k & \\ \hline & 1\end{array}\right]\boldsymbol{G}_{k,k+1}$$

$$=\left[\begin{array}{ccccc|cc}\nu_1 & & & & & & \\ \delta_2 & \nu_2 & & & & & \\ \eta_3 & \delta_3 & \nu_3 & & & & \\ & \ddots & \ddots & \ddots & & & \\ \hline & & \eta_k & \delta_k & & \widetilde{\nu}_k & \beta_{k+1} \\ & & & \eta_{k+1} & & \widetilde{\delta}_{k+1} & \alpha_{k+1}\end{array}\right]\left[\begin{array}{c|cc}\boldsymbol{I}_{k-1} & & \\ \hline & c_k & s_k \\ & s_k & -c_k\end{array}\right],$$

式中: c_k 和 s_k 由方程组

$$\widetilde{\nu}_kc_k+\beta_{k+1}s_k=\nu_k,\ \widetilde{\nu}_ks_k-\beta_{k+1}c_k=0$$

确定. 由此得 (注意到 $\widetilde{\nu}_1=\alpha_1,\ \widetilde{\delta}_2=\beta_2$)

$$\nu_k=\sqrt{\widetilde{\nu}_k^2+\beta_{k+1}^2},\ c_k=\frac{\widetilde{\nu}_k}{\nu_k},\ s_k=\frac{\beta_{k+1}}{\nu_k}, \tag{4.146}$$

以及

$$\delta_{k+1}=c_k\widetilde{\delta}_{k+1}+s_k\alpha_{k+1},\ \widetilde{\nu}_{k+1}=s_k\widetilde{\delta}_{k+1}-c_k\alpha_{k+1}, \tag{4.147}$$

$$\widetilde{\delta}_{k+2}=-c_k\beta_{k+2},\ \eta_{k+2}=s_k\beta_{k+2}. \tag{4.148}$$

由式 (4.143) 和式 (4.144), $\boldsymbol{x}_k$ 在式 (4.142) 中并不能递推地计算. 为此, 引进 $\boldsymbol{L}_k$, 它是 $\widetilde{\boldsymbol{L}}_k$ 以 ν_k 替换 $\widetilde{\nu}_k$ 而得. 相应地, 定义

$$\boldsymbol{W}_k=\left[\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_k\right],\ \ \boldsymbol{z}_k=\left[\zeta_1,\zeta_2,\cdots,\zeta_k\right]^{\mathrm{T}},$$

式中: $\boldsymbol{z}_k$ 满足三角方程组

$$\boldsymbol{L}_k\boldsymbol{z}_k=\beta_1\boldsymbol{e}_1. \tag{4.149}$$

由此得 (见 (4.146))

$$\zeta_k=\frac{\widetilde{\nu}_k}{\nu_k}\widetilde{\zeta}_k=c_k\widetilde{\zeta}_k. \tag{4.150}$$

最后, $\boldsymbol{W}_k$ 的列向量可按下列公式递推计算, 即

$$\left[\widetilde{\boldsymbol{w}}_k,\boldsymbol{v}_{k+1}\right]\begin{bmatrix}c_k & s_k\\ s_k & -c_k\end{bmatrix}=\left[\boldsymbol{w}_k,\widetilde{\boldsymbol{w}}_{k+1}\right],\ \widetilde{\boldsymbol{w}}_1=\boldsymbol{v}_1. \tag{4.151}$$

若 $\widetilde{\nu}_k=0$, 则 $\widetilde{\boldsymbol{L}}_k$ 是奇异的, 故式 (4.142) 不可解. 但由式 (4.146) 可知, 只要 $\beta_{k+1}\neq 0$, $\boldsymbol{L}_k$ 非奇异, 因而式 (4.149) 可解. 利用 $\boldsymbol{W}_k$ 和 $\boldsymbol{z}_k$ 递推地计算 $\widetilde{\boldsymbol{x}}_k$:

$$\widetilde{\boldsymbol{x}}_k=\boldsymbol{x}_0+\boldsymbol{W}_k\boldsymbol{z}_k=\widetilde{\boldsymbol{x}}_{k-1}+\zeta_k\boldsymbol{w}_k. \tag{4.152}$$

由式 (4.142), 式 (4.143), 式 (4.144) 和式 (4.152), 有

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_0+\widetilde{\boldsymbol{W}}_{k+1}\widetilde{\boldsymbol{z}}_{k+1}=\widetilde{\boldsymbol{x}}_k+\widetilde{\zeta}_{k+1}\widetilde{\boldsymbol{w}}_{k+1}. \tag{4.153}$$

式 (4.150) 只是给出了 ζ_k 与 $\widetilde{\zeta}_k$ 之间的关系. 还需要 ζ_k 或 $\widetilde{\zeta}_k\,(k=1,2,\cdots)$ 的计算公式. 由式 (4.149), 得

$$\begin{cases}\nu_1\zeta_1=\beta_1\Longrightarrow\zeta_1=\dfrac{\beta_1}{\nu_1},\\ \delta_2\zeta_1+\nu_2\zeta_2=0\Longrightarrow\zeta_2=-\dfrac{\delta_2\zeta_1}{\nu_2},\\ \eta_k\zeta_{k-2}+\delta_k\zeta_{k-1}+\nu_k\zeta_k=0\Longrightarrow\zeta_k=-\dfrac{\eta_k\zeta_{k-2}+\delta_k\zeta_{k-1}}{\nu_k},\ k\geqslant 3.\end{cases} \tag{4.154}$$

求解式 (4.149) 有比式 (4.142) 更好的数值性态. 由式 (4.152) 可求得 $\{\widetilde{\boldsymbol{x}}_k\}$, 若需要 $\{\boldsymbol{x}_k\}$ 可按式 (4.153) 得到. 所描述的算法称为 SYMMLQ 方法.

算法 4.12 (SYMMLQ 方法) 给定 n 阶非奇异的实对称矩阵 $\boldsymbol{A}$, n 维向量 $\boldsymbol{b}$ 和允许误差 $\varepsilon > 0$. 本算法计算向量 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\beta_1 = \|\boldsymbol{r}_0\|_2$; $\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta_1$;

$\alpha_1 = \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_1$; $\widetilde{\boldsymbol{v}}_2 = \boldsymbol{A}\boldsymbol{v}_1 - \alpha_1\boldsymbol{v}_1$; $\beta_2 = \|\widetilde{\boldsymbol{v}}_2\|_2$;

$\boldsymbol{v}_2 = \widetilde{\boldsymbol{v}}_2/\beta_2$; $\widetilde{\boldsymbol{w}}_1 = \boldsymbol{v}_1$;

$\widetilde{\nu}_1 = \alpha_1$; $\widetilde{\delta}_2 = \beta_2$; $\zeta_0 = 0$; $\eta_2 = 0$;

$\nu_1 = \sqrt{\alpha_1^2 + \beta_2^2}$; $c_1 = \widetilde{\nu}_1/\nu_1$; $s_1 = \beta_2/\nu_1$;

由 $\left[\widetilde{\boldsymbol{w}}_1, \boldsymbol{v}_2\right]\begin{bmatrix} c_1 & s_1 \\ s_1 & -c_1 \end{bmatrix} = \left[\boldsymbol{w}_1, \widetilde{\boldsymbol{w}}_2\right]$ 确定 $\boldsymbol{w}_1$ 和 $\widetilde{\boldsymbol{w}}_2$;

即 $\boldsymbol{w}_1 = c_1\widetilde{\boldsymbol{w}}_1 + s_1\boldsymbol{v}_2$; $\widetilde{\boldsymbol{w}}_2 = s_1\widetilde{\boldsymbol{w}}_1 - c_1\boldsymbol{v}_2$;

$\zeta_1 = \dfrac{\beta_1}{\nu_1}$; $\widetilde{\boldsymbol{x}}_1 = \boldsymbol{x}_0 + \zeta_1\boldsymbol{w}_1$; $k = 1$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 > \varepsilon)$

$\alpha_{k+1} = \boldsymbol{v}_{k+1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_{k+1}$;

$\widetilde{\boldsymbol{v}}_{k+2} = \boldsymbol{A}\boldsymbol{v}_{k+1} - \alpha_{k+1}\boldsymbol{v}_{k+1} - \beta_{k+1}\boldsymbol{v}_k$;

$\beta_{k+2} = \|\widetilde{\boldsymbol{v}}_{k+2}\|_2$; $\boldsymbol{v}_{k+2} = \widetilde{\boldsymbol{v}}_{k+2}/\beta_{k+2}$;

$\widetilde{\nu}_{k+1} = s_k\widetilde{\delta}_{k+1} - c_k\alpha_{k+1}$; $\nu_{k+1} = \sqrt{\widetilde{\nu}_{k+1}^2 + \beta_{k+2}^2}$;

$c_{k+1} = \dfrac{\widetilde{\nu}_{k+1}}{\nu_{k+1}}$; $s_{k+1} = \dfrac{\beta_{k+2}}{\nu_{k+1}}$;

$\delta_{k+1} = c_k\widetilde{\delta}_{k+1} + s_k\alpha_{k+1}$;

$\widetilde{\delta}_{k+2} = -c_k\beta_{k+2}$, $\eta_{k+2} = s_k\beta_{k+2}$;

$\zeta_{k+1} = -\dfrac{\eta_{k+1}\zeta_{k-1} + \delta_{k+1}\zeta_k}{\nu_{k+1}}$; $\widetilde{\zeta}_{k+1} = \zeta_{k+1}/c_{k+1}$;

由 $\left[\widetilde{\boldsymbol{w}}_{k+1}, \boldsymbol{v}_{k+2}\right]\begin{bmatrix} c_{k+1} & s_{k+1} \\ s_{k+1} & -c_{k+1} \end{bmatrix} = \left[\boldsymbol{w}_{k+1}, \widetilde{\boldsymbol{w}}_{k+2}\right]$ 确定 $\boldsymbol{w}_{k+1}$ 和 $\widetilde{\boldsymbol{w}}_{k+2}$;

即 $\boldsymbol{w}_{k+1} = c_{k+1}\widetilde{\boldsymbol{w}}_{k+1} + s_{k+1}\boldsymbol{v}_{k+2}$; $\widetilde{\boldsymbol{w}}_{k+2} = s_{k+1}\widetilde{\boldsymbol{w}}_{k+1} - c_{k+1}\boldsymbol{v}_{k+2}$;

$\boldsymbol{x}_{k+1} = \widetilde{\boldsymbol{x}}_k + \widetilde{\zeta}_{k+1}\widetilde{\boldsymbol{w}}_{k+1}$; $\widetilde{\boldsymbol{x}}_{k+1} = \widetilde{\boldsymbol{x}}_k + \zeta_{k+1}\boldsymbol{w}_{k+1}$;

$\boldsymbol{r}_{k+1} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1}$;

$k = k + 1$;

end

例 4.10 仍考虑例 4.8 中的线性方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, 其中矩阵 $\boldsymbol{A}$ 来自

http://www.cise.ufl.edu./research/sparse/matrices/PARSEC/SiNa.html,

它是一个 5743 阶的实对称不定稀疏矩阵, 右端项 $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{e}$, 其中 $\boldsymbol{e} = (1, 1, \cdots, 1)^{\mathrm{T}}$. 显然, 该方程组的真解为 $\boldsymbol{x}^* = (1, 1, \cdots, 1)^{\mathrm{T}}$. 将 SYMMLQ 方法应用到该线性方程组上, 迭代在 224 步后收敛 ($\varepsilon = 10^{-10}$), 计算得到的近似解 $\widehat{\boldsymbol{x}}$ 和真解 $\boldsymbol{x}^*$ 之间的绝对值误差为

$$\|\widehat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 2.8998 \times 10^{-8},$$

计算解 $\widehat{\boldsymbol{x}}$ 的残量满足

$$\|\boldsymbol{b}-\boldsymbol{A}\widehat{\boldsymbol{x}}\|_2 = 2.1331\times 10^{-8}.$$

将它与 MINRES 方法和 GMRES 方法进行了对比, 发现 SYMMLQ 方法与 MINRES 方法对本例有几乎完全等效的数值表现, 而 GMRES 方法则要略好一些. 迭代过程的收敛轨迹如图 4.11 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

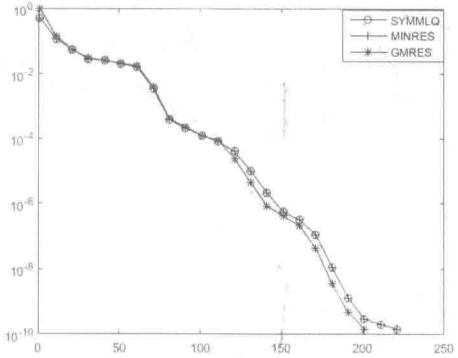

图 4.11 SYMMLQ 方法的收敛特性

4.4.2 收敛性分析

对于方程组

$$\boldsymbol{A}\boldsymbol{z}=\boldsymbol{r}_0,\quad \boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0,\quad \boldsymbol{z}=\boldsymbol{x}-\boldsymbol{x}_0,$$

考虑二次泛函

$$J_k^{(k)}(\boldsymbol{z}_k)=(\boldsymbol{z}-\boldsymbol{z}_k)^{\mathrm{T}}\boldsymbol{A}^k(\boldsymbol{z}-\boldsymbol{z}_k), \tag{4.155}$$

式中: k 为非负整数; $\boldsymbol{z}_k=\boldsymbol{V}_k\boldsymbol{y}_k$, $\boldsymbol{V}_k=\left[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k\right]$.

当 $k=2$ 时, 式 (4.155) 中的泛函

$$\begin{aligned} J_k^{(2)}(\boldsymbol{z}_k) &= (\boldsymbol{z}-\boldsymbol{z}_k)^{\mathrm{T}}\boldsymbol{A}^2(\boldsymbol{z}-\boldsymbol{z}_k)=\|\boldsymbol{A}(\boldsymbol{z}-\boldsymbol{z}_k)\|_2^2 \\ &= \|\boldsymbol{r}_0-\boldsymbol{A}(\boldsymbol{x}_k-\boldsymbol{x}_0)\|_2^2=\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k\|_2^2=\|\boldsymbol{r}_k\|_2^2 \end{aligned}$$

是正定二次泛函. 极小化这一正定二次泛函就是 4.3 节中极小残量法 (MINRES) 的基本思想.

当 $k=1$ 时, 式 (4.155) 中的泛函为

$$J_k^{(1)}(\boldsymbol{z}_k)=(\boldsymbol{z}-\boldsymbol{z}_k)^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{z}-\boldsymbol{z}_k)=(\boldsymbol{x}-\boldsymbol{x}_0-\boldsymbol{V}_k\boldsymbol{y}_k)^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}_0-\boldsymbol{V}_k\boldsymbol{y}_k).$$

求上述泛函稳定点 $\boldsymbol{y}_k$:

$$\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}_0-\boldsymbol{V}_k\boldsymbol{y}_k)=\boldsymbol{V}_k^{\mathrm{T}}(\boldsymbol{r}_0-\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_k)=\boldsymbol{0}$$

是本节 SYMMLQ 方法的出发点.

若在式 (4.155) 中取 $k=0$, 则有

$$J_k^{(0)}(\boldsymbol{z}_k)=(\boldsymbol{z}-\boldsymbol{z}_k)^{\mathrm{T}}(\boldsymbol{z}-\boldsymbol{z}_k)=\|\boldsymbol{x}-\boldsymbol{x}_k\|_2^2=\|\boldsymbol{\varepsilon}_k\|_2^2.$$

这也是一个正定二次泛函. 因而使 $J_k^{(0)}$ 极小化就是使误差 $\boldsymbol{\varepsilon}_k=\boldsymbol{x}-\boldsymbol{x}_k$ 的范数极小化.

为了使极小误差法可行, 取迭代所处的仿射子空间为

$$\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{A}\boldsymbol{r}_0),$$

也就是说

$$\boldsymbol{z}_k=\boldsymbol{x}_k-\boldsymbol{x}_0=\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_k.$$

由此, $\boldsymbol{y}_k$ 是最小二乘问题

$$\min\|\boldsymbol{x}-\boldsymbol{x}_0-\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_k\|_2=\|\boldsymbol{A}^{-1}\boldsymbol{r}_0-\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_k\|_2$$

的解. 由法方程可知 $\boldsymbol{y}_k$ 满足线性方程组

$$\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}^2\boldsymbol{V}_k\boldsymbol{y}_k=\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{r}_0=\beta_1\boldsymbol{e}_1,\quad \boldsymbol{x}_k=\boldsymbol{x}_0+\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_k. \tag{4.156}$$

由对称 Lanzcos 分解式 (4.137) 及 LQ 分解式 (4.139), 得

$$\begin{aligned}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}^2\boldsymbol{V}_k&=(\boldsymbol{A}\boldsymbol{V}_k)^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{V}_k)\\&=(\boldsymbol{V}_k\boldsymbol{T}_k+\beta_{k+1}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{V}_k\boldsymbol{T}_k+\beta_{k+1}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}})\\&=\boldsymbol{T}_k^2+\beta_{k+1}^2\boldsymbol{e}_k\boldsymbol{e}_k^{\mathrm{T}}=(\widetilde{\boldsymbol{L}}_k\boldsymbol{Q}_k)(\widetilde{\boldsymbol{L}}_k\boldsymbol{Q}_k)^{\mathrm{T}}+\beta_{k+1}^2\boldsymbol{e}_k\boldsymbol{e}_k^{\mathrm{T}}\\&=\widetilde{\boldsymbol{L}}_k\widetilde{\boldsymbol{L}}_k^{\mathrm{T}}+\beta_{k+1}^2\boldsymbol{e}_k\boldsymbol{e}_k^{\mathrm{T}}=\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}},\end{aligned}$$

最后一个等式成立是因为 $\widetilde{\boldsymbol{L}}_k\widetilde{\boldsymbol{L}}_k^{\mathrm{T}}$ 与 $\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}}$ 只是 (k,k) 处的那个元素不同, 前者是 $\eta_k^2+\delta_k^2+\widetilde{\nu}_k^2$, 而后者为 $\eta_k^2+\delta_k^2+\nu_k^2$. 注意到 $\nu_k^2=\widetilde{\nu}_k^2+\beta_{k+1}^2$, 即可得等式成立. 这样, 直接从 $\boldsymbol{T}_k$ 的 LQ 分解得到了 $\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}^2\boldsymbol{V}_k$ 的 Cholesky 分解. 代入式 (4.156), 需要求解如下线性方程组以得到 $\boldsymbol{y}_k$ 和 $\boldsymbol{x}_k$:

$$\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}}\boldsymbol{y}_k=\beta_1\boldsymbol{e}_1,\quad \boldsymbol{x}_k=\boldsymbol{x}_0+\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_k. \tag{4.157}$$

由上述方式得到近似解序列 $\{\boldsymbol{x}_k\}$ 的方法通常称为*极小误差法*. 下面证明这个极小误差解序列 $\{\boldsymbol{x}_k\}$ 就是 SYMMLQ 方法中的 $\{\widetilde{\boldsymbol{x}}_k\}$ (见式 (4.152)).

定理 4.9 SYMMLQ 方法中的序列 $\{\widetilde{\boldsymbol{x}}_k\}$ 是在 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{A}\boldsymbol{r}_0)$ 的极小误差逼近.

证明 由式 (4.157) 可知极小误差解 $\boldsymbol{x}_k$ 可表示为

$$\boldsymbol{x}_k=\boldsymbol{x}_0+\beta_1\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{L}_k^{-\mathrm{T}}\boldsymbol{L}_k^{-1}\boldsymbol{e}_1. \tag{4.158}$$

由式 (4.149) 和式 (4.152) 可知, $\widetilde{\boldsymbol{x}}_k$ 可表示为

$$\widetilde{\boldsymbol{x}}_k = \boldsymbol{x}_0 + \beta_1 \boldsymbol{W}_k \boldsymbol{L}_k^{-1} \boldsymbol{e}_1. \tag{4.159}$$

比较式 (4.158) 和式 (4.159) 右端, 只需证明

$$\boldsymbol{W}_k = \boldsymbol{A}\boldsymbol{V}_k \boldsymbol{L}_k^{-\mathrm{T}}, \tag{4.160}$$

即完成了定理的证明. 由式 (4.139) 和式 (4.141) 可知

$$\widetilde{\boldsymbol{W}}_{k+1} \widetilde{\boldsymbol{L}}_{k+1}^{\mathrm{T}} = \boldsymbol{V}_{k+1} (\widetilde{\boldsymbol{L}}_{k+1} \boldsymbol{Q}_{k+1})^{\mathrm{T}} = \boldsymbol{V}_{k+1} \boldsymbol{T}_{k+1}. \tag{4.161}$$

再由式 (4.137), 得

$$\widetilde{\boldsymbol{W}}_{k+1} = (\boldsymbol{A}\boldsymbol{V}_{k+1} - \beta_{k+2} \boldsymbol{v}_{k+2} \boldsymbol{e}_{k+1}^{\mathrm{T}}) \widetilde{\boldsymbol{L}}_{k+1}^{-\mathrm{T}}. \tag{4.162}$$

比较式 (4.162) 前 k 列, 并注意到 $\widetilde{\boldsymbol{L}}_{k+1}^{-\mathrm{T}}$ 是上三角矩阵, 可得式 (4.160). 证毕. □

下面分析极小误差法的误差估计. 令

$$\widetilde{\mathcal{P}}_{k-1}^{0}(t) = 1 - t^2 \mathcal{P}_{k-1}(t)$$

表示次数不超过 $k+1$, 在 $t=0$ 时值等于 1 且一阶导数等于 0 的多项式集合. 因 $\boldsymbol{x}_k - \boldsymbol{x}_0 \in \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{A}\boldsymbol{r}_0)$, 故存在 $p_{k-1} \in \mathcal{P}_{k-1}$, 使得 $\boldsymbol{x}_k - \boldsymbol{x}_0 = p_{k-1}(\boldsymbol{A})\boldsymbol{A}\boldsymbol{r}_0$. 所以误差 $\boldsymbol{\varepsilon}_k = \boldsymbol{x}^* - \boldsymbol{x}_k$ 满足

$$\begin{aligned}
\|\boldsymbol{\varepsilon}_k\|_2^2 &= \min_{p_{k-1} \in \mathcal{P}_{k-1}} \big(\boldsymbol{\varepsilon}_0 - p_{k-1}(\boldsymbol{A})\boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{\varepsilon}_0 - p_{k-1}(\boldsymbol{A})\boldsymbol{A}\boldsymbol{r}_0\big) \\
&= \min_{p_{k-1} \in \mathcal{P}_{k-1}} \big(\boldsymbol{\varepsilon}_0 - p_{k-1}(\boldsymbol{A})\boldsymbol{A}^2\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_0 - p_{k-1}(\boldsymbol{A})\boldsymbol{A}^2\boldsymbol{\varepsilon}_0\big) \\
&= \min_{p_{k-1} \in \mathcal{P}_{k-1}} \big([\boldsymbol{I} - \boldsymbol{A}^2 p_{k-1}(\boldsymbol{A})]\boldsymbol{\varepsilon}_0, [\boldsymbol{I} - \boldsymbol{A}^2 p_{k-1}(\boldsymbol{A})]\boldsymbol{\varepsilon}_0\big) \\
&= \min_{\widetilde{p}_{k-1} \in \widetilde{\mathcal{P}}_{k-1}^{0}} \big(\widetilde{p}_{k-1}(\boldsymbol{A})\boldsymbol{\varepsilon}_0, \widetilde{p}_{k-1}(\boldsymbol{A})\boldsymbol{\varepsilon}_0\big) \\
&\leqslant \min_{\widetilde{p}_{k-1} \in \widetilde{\mathcal{P}}_{k-1}^{0}} \max_{\lambda \in \sigma(\boldsymbol{A})} \widetilde{p}_{k-1}^2(\lambda) \|\boldsymbol{\varepsilon}_0\|_2^2.
\end{aligned} \tag{4.163}$$

因为 $\boldsymbol{A}$ 是对称不定的, 所以 $\sigma(\boldsymbol{A})$ 包含正的和负的特征值. 有下面的定理.

定理 4.10 *若 $\boldsymbol{A}$ 的谱 $\sigma(\boldsymbol{A}) \subset [-b, -a] \cup [a, b]$, 其中 $0 < a < b$. 则对极小误差迭代法成立*

$$\|\boldsymbol{\varepsilon}_k\|_2 \leqslant \frac{2\widetilde{r}^{\widetilde{k}}}{1 + \widetilde{r}^{2\widetilde{k}}} \|\boldsymbol{\varepsilon}_0\|_2, \tag{4.164}$$

式中:

$$\widetilde{r} = \frac{b-a}{b+a} = \frac{b/a - 1}{b/a + 1}, \quad \widetilde{k} = \left[\frac{k+1}{2}\right].$$

证明 取多项式

$$p_{k-1}(t)=\frac{C_{\widetilde{k}}(v(t))}{C_{\widetilde{k}}(v(0))},$$

式中: $C_{\widetilde{k}}(z)$ 为 $\widetilde{k}=[(k+1)/2]$ 次 Chebyshev 多项式, 且

$$v(t)=1-\frac{2(t^2-a^2)}{b^2-a^2},$$

它将区间 $[-b,-a]\cup[a,b]$ 映射到 $[-1,1]$.

显然, $p_{k-1}(t)$ 的次数不超过 $k+1$, 它是偶函数, 且 $p_{k-1}(0)=1$. 故 $p_{k-1}(t)\in\widetilde{P}_{k-1}^0$. 由式 (4.163), 有

$$\|\boldsymbol{\varepsilon}_k\|_2\leqslant\max_{t\in[-b,-a]\cup[a,b]}\left|\frac{C_{\widetilde{k}}(v(t))}{C_{\widetilde{k}}(v(0))}\right|\|\boldsymbol{\varepsilon}_0\|_2=\frac{1}{C_{\widetilde{k}}(v(0))}\|\boldsymbol{\varepsilon}_0\|_2.$$

不难推得

$$\begin{aligned}C_{\widetilde{k}}(v(0))&=C_{\widetilde{k}}\Big(\frac{b^2+a^2}{b^2-a^2}\Big)\\&=\frac{1}{2}\left\{\left[\frac{b^2+a^2}{b^2-a^2}+\sqrt{\Big(\frac{b^2+a^2}{b^2-a^2}\Big)^2-1}\right]^{\widetilde{k}}+\left[\frac{b^2+a^2}{b^2-a^2}+\sqrt{\Big(\frac{b^2+a^2}{b^2-a^2}\Big)^2-1}\right]^{-\widetilde{k}}\right\}\\&=\frac{1}{2}\left[\Big(\frac{b+a}{b-a}\Big)^{\widetilde{k}}+\Big(\frac{b+a}{b-a}\Big)^{-\widetilde{k}}\right]=\frac{1}{2}\left[\Big(\frac{1}{\widetilde{r}}\Big)^{\widetilde{k}}+\widetilde{r}^{\widetilde{k}}\right]=\frac{1+\widetilde{r}^{2\widetilde{k}}}{2\widetilde{r}^{\widetilde{k}}}.\end{aligned}$$

证毕. □

4.5 拟极小残量法

在 4.2 节中, 为了解决广义极小残量法的存储问题而采用了重新开始的方法, 进而导出了实际应用中常用的重开始广义极小残量法 GMRES(m). 本节介绍另一种解决存储问题的方法—拟极小残量法, 在目前的文献中常简称为 QMRES 方法 (Quasi-Minimal RESidual Approach).

GMRES 方法的存储量主要来自计算 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{b})$ 的正交基 $\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k$. 由于要求这些 $\boldsymbol{v}_i\,(i=1,2,\cdots,k)$ 相互正交, 故必须在计算中将这些向量都存储起来. 因此要使其存储量降低, 一个自然的想法是放弃正交性要求, 而选择其他适当的基向量.

类比于 MINRES 方法, 自然希望计算 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 的一组基向量 $\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k$, 使得

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\mathrm{span}\{\boldsymbol{v}_1,\cdots,\boldsymbol{v}_k\},\quad \boldsymbol{v}_1=\boldsymbol{r}_0/\|\boldsymbol{r}_0\|_2,\tag{4.165}$$

$$\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_{k+1}\widetilde{\boldsymbol{T}}_k,\tag{4.166}$$

式中: $\boldsymbol{V}_{k+1}=[\boldsymbol{V}_k,\boldsymbol{v}_{k+1}]=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k,\boldsymbol{v}_{k+1}]$ 列满秩, 而 $\widetilde{\boldsymbol{T}}_k$ 是 $(k+1)\times k$ 阶三对角矩阵.

如果能够找到一种方法来实现式 (4.165) 和式 (4.166) 中的分解, 则由于这样的基满足等式 $\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_{k+1}\widetilde{\boldsymbol{T}}_k$, 向量 $\boldsymbol{v}_i$ 就可由三项递推公式来确定, 从而所需的存储量只有 $O(n)$ 而并非 GMRES 方法的 $O(kn)$. 然而, 这样一来, 所要解决的极小化问题 (4.52) 就转化为求 $\boldsymbol{z}_k\in\mathbb{R}^k$, 使得

$$\|\boldsymbol{r}_k\|_2=\|\boldsymbol{V}_{k+1}(\beta\boldsymbol{e}_1-\widetilde{\boldsymbol{T}}_k\boldsymbol{z}_k)\|_2=\min, \tag{4.167}$$

式中: $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{z}_k$, $\beta=\|\boldsymbol{r}_0\|_2$. 但由于此时的 $\boldsymbol{V}_{k+1}$ 列向量并不正交, 故求解这一极小化问题就有很大的困难. 注意到

$$\|\boldsymbol{r}_k\|_2=\|\boldsymbol{V}_{k+1}(\beta\boldsymbol{e}_1-\widetilde{\boldsymbol{T}}_k\boldsymbol{z}_k)\|_2\leqslant\|\boldsymbol{V}_{k+1}\|_2\|\beta\boldsymbol{e}_1-\widetilde{\boldsymbol{T}}_k\boldsymbol{z}_k\|_2,$$

如果 $\|\boldsymbol{V}_{k+1}\|_2$ 不是特别大, 则只需求得最小二乘问题

$$\min\left\{\|\beta\boldsymbol{e}_1-\widetilde{\boldsymbol{T}}_k\boldsymbol{z}\|_2:\boldsymbol{z}\in\mathbb{R}^k\right\} \tag{4.168}$$

的解 $\boldsymbol{z}_k$. 虽然 $\|\boldsymbol{r}_k\|_2$ 并没有达到最小, 但也会相对很小, 这就是拟极小化方法的基本想法. 总结起来, 主要有两步:

第 1 步, 计算 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{b})$ 之满足条件 (4.165) 和 (4.166) 的一组基.

第 2 步, 求解最小二乘问题 (4.168).

关键是第 1 步如何实现, 第 2 步可用与 4.3 节中完全一样的方法来求解. 这里用非对称 Lanczos 方法实现 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 基向量的计算问题.

4.5.1 非对称 Lanczos 方法

非对称 Lanczos 方法是对称 Lanczos 方法的一种自然的推广, 它是同时计算 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v}_1)$ 和 $\mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}},\boldsymbol{w}_1)$ 的基向量 $\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k$ 和 $\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_k$, 使得

$$\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{v}_j=\begin{cases}0, & i\neq j,\\ 1, & i=j.\end{cases} \tag{4.169}$$

式 (4.169) 称为双正交性条件.

设矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{v}_1$ 已经给定, 并且假定 $\|\boldsymbol{v}_1\|_2=1$. 首先需选择一个 $\boldsymbol{w}_1\in\mathbb{R}^n$, 使得 $\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{w}_1=1$. 如果没有什么特别的信息可以利用的话, 可以简单地选择 $\boldsymbol{w}_1=\boldsymbol{v}_1$. 然后类比于对称 Lanczos 方法的计算过程, 也采用如下的三项递推方式, 即

$$\begin{cases}\beta_i\boldsymbol{v}_{i+1}=\boldsymbol{A}\boldsymbol{v}_i-\alpha_i\boldsymbol{v}_i-\gamma_{i-1}\boldsymbol{v}_{i-1},\\ \gamma_i\boldsymbol{w}_{i+1}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{w}_i-\alpha_i\boldsymbol{w}_i-\beta_{i-1}\boldsymbol{w}_{i-1}.\end{cases} \tag{4.170}$$

依次确定 $\alpha_i,\beta_i,\gamma_i,\boldsymbol{v}_{i+1}$ 和 $\boldsymbol{w}_{i+1}$. 这里假定 $\gamma_0\boldsymbol{v}_0=\beta_0\boldsymbol{w}_0=\boldsymbol{0}$, 即当 $i=1$ 时, 有

$$\beta_1\boldsymbol{v}_2=\boldsymbol{A}\boldsymbol{v}_1-\alpha_1\boldsymbol{v}_1,$$

$$\gamma_1 \boldsymbol{w}_2 = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{w}_1 - \alpha_1 \boldsymbol{w}_1.$$

由于希望计算得到的 $\boldsymbol{v}_2$ 和 $\boldsymbol{w}_2$ 满足 $\boldsymbol{v}_2 \perp \boldsymbol{w}_1$, $\boldsymbol{w}_2 \perp \boldsymbol{v}_1$, 故在上面的第 1 式两边左乘 $\boldsymbol{w}_1^{\mathrm{T}}$, 得

$$\alpha_1 = \boldsymbol{w}_1^{\mathrm{T}} \boldsymbol{A} \boldsymbol{v}_1. \tag{4.171}$$

此处利用了 $\boldsymbol{w}_1^{\mathrm{T}} \boldsymbol{v}_1 = 1$. 当然, 也可以在前面的第 2 式两边左乘 $\boldsymbol{v}_1^{\mathrm{T}}$, 得到 α_1 的表达式 (4.171). 一旦 α_1 确定, 便可计算

$$\begin{aligned} \beta_1 \boldsymbol{v}_2 &= \widetilde{\boldsymbol{v}}_2 = \boldsymbol{A} \boldsymbol{v}_1 - \alpha_1 \boldsymbol{v}_1, \\ \gamma_1 \boldsymbol{w}_2 &= \widetilde{\boldsymbol{w}}_2 = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{w}_1 - \alpha_1 \boldsymbol{w}_1. \end{aligned}$$

由于要求 $\boldsymbol{v}_2^{\mathrm{T}} \boldsymbol{w}_2 = 1$, 故必有

$$\beta_1 \gamma_1 = \widetilde{\boldsymbol{v}}_2^{\mathrm{T}} \widetilde{\boldsymbol{w}}_2 = \omega_1.$$

如果 $\omega_1 = 0$, 则只好结束; 否则可选择两个数 β_1 和 γ_1 使上式成立. 可选择

$$\beta_1 = \sqrt{|\omega_1|}, \quad \gamma_1 = \omega_1 / \beta_1. \tag{4.172}$$

这样一来, 所需的向量 $\boldsymbol{v}_2$ 和 $\boldsymbol{w}_2$ 即为

$$\boldsymbol{v}_2 = \widetilde{\boldsymbol{v}}_2 / \beta_1, \quad \boldsymbol{w}_2 = \widetilde{\boldsymbol{w}}_2 / \gamma_1. \tag{4.173}$$

假定已经确定了

$$\begin{aligned} &\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_i; \quad \boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_i; \\ &\alpha_1, \alpha_2, \cdots, \alpha_{i-1}; \quad \gamma_1, \gamma_2, \cdots, \gamma_{i-1}; \quad \beta_1, \beta_2, \cdots, \beta_{i-1}, \end{aligned}$$

下一步利用 $\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{v}_{i+1} = \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{v}_{i-1} = 0$ 和 $\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{v}_i = 1$, 从式 (4.170) 的第 1 式可导出

$$\alpha_i = \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{A} \boldsymbol{v}_i. \tag{4.174}$$

然后, 计算

$$\begin{cases} \widetilde{\boldsymbol{v}}_{i+1} = \boldsymbol{A} \boldsymbol{v}_i - \alpha_i \boldsymbol{v}_i - \gamma_{i-1} \boldsymbol{v}_{i-1}, \\ \widetilde{\boldsymbol{w}}_{i+1} = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{w}_i - \alpha_i \boldsymbol{w}_i - \beta_{i-1} \boldsymbol{w}_{i-1}, \\ \omega_i = \widetilde{\boldsymbol{v}}_{i+1}^{\mathrm{T}} \widetilde{\boldsymbol{w}}_{i+1}. \end{cases} \tag{4.175}$$

若 $\omega_i = 0$, 则结束; 否则, 计算

$$\beta_i = \sqrt{|\omega_i|}, \quad \gamma_i = \omega_i / \beta_i, \quad \boldsymbol{v}_{i+1} = \widetilde{\boldsymbol{v}}_{i+1} / \beta_i, \quad \boldsymbol{w}_{i+1} = \widetilde{\boldsymbol{w}}_{i+1} / \gamma_i. \tag{4.176}$$

假定上述过程共进行了 k 步, 现在令

$$\boldsymbol{V}_k = \big[\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\big], \quad \boldsymbol{W}_k = \big[\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_k\big],$$

$$\boldsymbol{T}_k = \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma_{k-1} \\ & & & \beta_{k-1} & \alpha_k \end{bmatrix}. \tag{4.177}$$

则易从上面的推导过程归纳地证明关于非对称 Lanczos 迭代的如下基本性质:

(1) $\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{T}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}$.

(2) $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}_k = \boldsymbol{W}_k\boldsymbol{T}_k^{\mathrm{T}} + \gamma_k\boldsymbol{w}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}$.

(3) $\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{W}_k = \boldsymbol{I}_k,\ \boldsymbol{v}_{k+1}^{\mathrm{T}}\boldsymbol{W}_k = \boldsymbol{0},\ \boldsymbol{w}_{k+1}^{\mathrm{T}}\boldsymbol{V}_k = \boldsymbol{0},\ \boldsymbol{w}_{k+1}^{\mathrm{T}}\boldsymbol{v}_{k+1} = 1$.

(4) $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}_1) = \mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$.

(5) $\mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}}, \boldsymbol{w}_1) = \mathrm{span}\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_k\}$.

其实 (1) 和 (2) 就是前面算法的矩阵表示. (4) 和 (5) 是前三条的直接推论. 例如 (4), 由 (1) 归纳地可证 $\boldsymbol{v}_i \in \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}_1), i = 1, 2, \cdots, k$, 而 (3) 又蕴涵着 $\boldsymbol{V}_k$ 是列满秩的 (即 $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k$ 是线性无关的), 从而必有 (4) 成立. 因此关键是 (3) 的证明.

下面用数学归纳法证明性质 (3). 首先由初值 $\boldsymbol{w}_1$ 的选择知, 自然有 $\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{v}_1 = 1$ 成立. 现在假设已经对自然数 i 证明了向量组 $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_i$ 和 $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_i$ 满足双正交性条件, 再来考虑 $i+1$ 的情形. 注意这里假定 $i+1 \leqslant k+1$.

首先由归纳法假设和 α_i 的定义, 有

$$\begin{aligned} \boldsymbol{v}_{i+1}^{\mathrm{T}}\boldsymbol{w}_i &= \frac{1}{\beta_i}(\boldsymbol{A}\boldsymbol{v}_i - \alpha_i\boldsymbol{v}_i - \gamma_{i-1}\boldsymbol{v}_{i-1})^{\mathrm{T}}\boldsymbol{w}_i \\ &= \frac{1}{\beta_i}\left(\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{w}_i - \alpha_i\right) = 0. \end{aligned}$$

对任意的 $l < i$, 有

$$\begin{aligned} \boldsymbol{v}_{i+1}^{\mathrm{T}}\boldsymbol{w}_l &= \frac{1}{\beta_i}\left(\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{w}_l - \alpha_i\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{w}_l - \gamma_{i-1}\boldsymbol{v}_{i-1}^{\mathrm{T}}\boldsymbol{w}_l\right) \\ &= \frac{1}{\beta_i}\left[\boldsymbol{v}_i^{\mathrm{T}}(\alpha_l\boldsymbol{w}_l + \beta_{l-1}\boldsymbol{w}_{l-1} + \gamma_l\boldsymbol{w}_{l+1}) - \gamma_{i-1}\boldsymbol{v}_{i-1}^{\mathrm{T}}\boldsymbol{w}_l\right]. \end{aligned}$$

当 $l < i-1$ 时, 由归纳法假设立即由上式知 $\boldsymbol{v}_{i+1}^{\mathrm{T}}\boldsymbol{w}_l = 0$. 而当 $l = i-1$ 时, 直接计算有

$$\boldsymbol{v}_{i+1}^{\mathrm{T}}\boldsymbol{w}_{i-1} = \frac{1}{\beta_i}\left(\alpha_{i-1}\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{w}_{i-1} + \beta_{i-2}\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{w}_{i-2} + \gamma_{i-1}\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{w}_i - \gamma_{i-1}\boldsymbol{v}_{i-1}^{\mathrm{T}}\boldsymbol{w}_{i-1}\right) = 0,$$

最后的等式用到了归纳法假设. 至于 $\boldsymbol{w}_{i+1}^{\mathrm{T}}\boldsymbol{v}_{i+1} = 1$ 可由计算过程立即得到. 这表明 (3) 对 $i+1$ 也成立. 因此由归纳法原理知 (3) 成立.

综合上面的讨论, 可得下面的非对称 Lanczos 方法.

算法 4.13 (非对称 Lanczos 方法) 给定矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 向量 $\boldsymbol{v}_1\in\mathbb{R}^n$ ($\|\boldsymbol{v}_1\|_2=1$) 和正整数 k. 本算法计算形如 (4.177) 的三个矩阵 $\boldsymbol{V}_k,\boldsymbol{W}_k$ 和 $\boldsymbol{T}_k$, 以及向量 $\boldsymbol{v}_{k+1},\boldsymbol{w}_{k+1}$ 和数 β_k,γ_k, 满足前面所述的 5 条性质.

选择向量 $\boldsymbol{w}_1\in\mathbb{R}^n$ 满足 $\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{v}_1=1$;
$\boldsymbol{v}=\boldsymbol{A}\boldsymbol{v}_1$; $\boldsymbol{w}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{w}_1$;
for $i=1:k$
  $\alpha_i=\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{v}$; $\boldsymbol{v}=\boldsymbol{v}-\alpha_i\boldsymbol{v}_i$; $\boldsymbol{w}=\boldsymbol{w}-\alpha_i\boldsymbol{w}_i$;
  if $\|\boldsymbol{v}\|_2=0$ 或者 $\|\boldsymbol{w}\|_2=0$
    stop
  else
    $\omega_i=\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}$;
  end
  if $\omega_i=0$
    stop
  else
    $\beta_i=\sqrt{|\omega_i|}$; $\gamma_i=\omega_i/\beta_i$;
    $\boldsymbol{v}_{i+1}=\boldsymbol{v}/\beta_i$; $\boldsymbol{w}_{i+1}=\boldsymbol{w}/\gamma_i$;
  end
  $\boldsymbol{v}=\boldsymbol{A}\boldsymbol{v}_{i+1}-\gamma_i\boldsymbol{v}_i$; $\boldsymbol{w}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{w}_{i+1}-\beta_i\boldsymbol{w}_i$;
end

注 4.5 非对称 Lanczos 迭代的基本性质 (1) $\sim$ (3) 蕴涵着

$$\boldsymbol{T}_k=\boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k.$$

这表明 $\boldsymbol{T}_k$ 正好是 $\boldsymbol{A}$ 沿着 $\mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}},\boldsymbol{w}_1)^{\perp}$ 到 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v}_1)$ 上的投影. 当然, $\boldsymbol{T}_k^{\mathrm{T}}$ 也是 $\boldsymbol{A}$ 沿着 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v}_1)^{\perp}$ 到 $\mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}},\boldsymbol{w}_1)$ 上的投影. 有关投影的几何解释见注 4.6.

注 4.6 设 $\boldsymbol{X},\boldsymbol{Y}\in\mathbb{R}^{n\times k}$ 满足 $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}=\boldsymbol{I}_k$, 并记

$$\mathcal{X}=\mathcal{R}(\boldsymbol{X}),\quad \mathcal{Y}=\mathcal{R}(\boldsymbol{Y}),\quad \boldsymbol{P}=\boldsymbol{X}\boldsymbol{Y}^{\mathrm{T}},$$

则容易验证

$$\mathcal{R}(\boldsymbol{P})=\mathcal{X},\quad \mathcal{N}(\boldsymbol{P})=\mathcal{Y}^{\perp},\quad \boldsymbol{P}^2=\boldsymbol{P},$$

这表明 $\boldsymbol{P}$ 是沿 $\mathcal{Y}^{\perp}$ 到 $\mathcal{X}$ 上的投影算子.

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 将 $\boldsymbol{A}$ 视作 $\mathbb{R}^n$ 到 $\mathbb{R}^n$ 的线性算子, 并记其在 $\mathcal{X}$ 上的限制为 $\boldsymbol{A}|_{\mathcal{X}}$, 则 $\boldsymbol{P}\circ\boldsymbol{A}|_{\mathcal{X}}$ 就是从 $\mathcal{X}$ 到 $\mathcal{X}$ 的线性算子. 由于

$$\boldsymbol{P}\circ\boldsymbol{A}|_{\mathcal{X}}(\boldsymbol{X}\boldsymbol{v})=\boldsymbol{X}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}\boldsymbol{v},\quad \forall\,\boldsymbol{v}\in\mathbb{R}^k,$$

故 $\boldsymbol{B}=\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{X}$ 正好是算子 $\boldsymbol{P}\circ\boldsymbol{A}|_{\mathcal{X}}$ 在给定基 $\boldsymbol{X}$ 之下的矩阵表示. 因此, 通常就称 $\boldsymbol{B}$ 是 $\boldsymbol{A}$ 沿 $\mathcal{Y}^{\perp}$ 到 $\mathcal{X}$ 上的投影.

从实际应用的角度来看, 非对称 Lanczos 方法比 Arnoldi 方法有着很大的优势, 这主要体现在它仅需存储六个 n 维向量 (因为 $k \ll n$, 所以 $\boldsymbol{T}_k$ 的存储量是微不足道的), 并不随着 k 的增加而增加.

但是, 从另一个角度来看, 非对称 Lanczos 算法会发生中断, 即会出现 $\widetilde{\boldsymbol{v}}_{i+1}$ 和 $\widetilde{\boldsymbol{w}}_{i+1}$ 均不为零, 而 $\widetilde{\boldsymbol{w}}_{i+1}^{\mathrm{T}}\widetilde{\boldsymbol{v}}_{i+1} = 0$ 的情形. 虽然在实际计算时出现 $\widetilde{\boldsymbol{w}}_{i+1}^{\mathrm{T}}\widetilde{\boldsymbol{v}}_{i+1} = 0$ 的概率很小, 但 $|\widetilde{\boldsymbol{w}}_{i+1}^{\mathrm{T}}\widetilde{\boldsymbol{v}}_{i+1}|$ 很小的情况还是经常会遇到的, 此时 $|\gamma_i|$ 和 $|\beta_i|$ 之中有一个就会变得很小, 从而导致在计算过程中引进较大的误差. 解决这一问题的一种方法就是采用 Look–ahead 技术. Look–ahead 方式是基于对算法 4.13 的仔细观察而得到的. 在遇到中断的时候, 虽然不能给出 $\boldsymbol{v}_{i+1}$ 和 $\boldsymbol{w}_{i+1}$, 但却常常可以给出 $\boldsymbol{v}_{i+2}$ 和 $\boldsymbol{w}_{i+2}$. 这就使得 Lanczos 方法可以继续进行下去. Look–ahead 技术虽然解决了非对称 Lanczos 算法的中断问题, 然而付出的代价也是不容忽视的: 一是使得实现过程变得更加复杂; 二是得到的投影矩阵已经不是三对角矩阵, 致使 QMRES 方法的运算复杂性大为增加. 因此, 就求解线性方程组而言, 通常是放弃使用 Look–ahead, 而是在遇到中断时简单地采用重新开始的办法.

4.5.2 QMRES 方法

从 $\boldsymbol{v}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|_2$ 出发, 用算法 4.13 计算得到分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{T}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}} = \boldsymbol{V}_{k+1}\widetilde{\boldsymbol{T}}_k \tag{4.178}$$

之后, QMRES 方法的下一步是求解最小二乘问题

$$\min\left\{\|\beta\boldsymbol{e}_1 - \widetilde{\boldsymbol{T}}_k\boldsymbol{z}\|_2 : \boldsymbol{z} \in \mathbb{R}^k\right\}, \tag{4.179}$$

式中: $\beta = \|\boldsymbol{r}_0\|_2$. 一旦这样的 $\boldsymbol{z}_k$ 求得, 则所寻求的 $\boldsymbol{x}_k$ 就是 $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k\boldsymbol{z}_k$.

极小化问题 (4.179) 是一个系数矩阵为三对角矩阵的最小二乘问题, 仍然用 QR 分解方法来求解.

由于 $\widetilde{\boldsymbol{T}}_k$ 具有式 (4.177) 所示的形状, 故可以计算 k 个 Givens 变换 $\boldsymbol{G}_1, \boldsymbol{G}_2, \cdots, \boldsymbol{G}_k$, 使得

$$\boldsymbol{G}_k\boldsymbol{G}_{k-1}\cdots\boldsymbol{G}_2\boldsymbol{G}_1\widetilde{\boldsymbol{T}}_k = \begin{bmatrix} \boldsymbol{R}_k \\ \boldsymbol{O} \end{bmatrix}, \tag{4.180}$$

式中:

$$\boldsymbol{G}_i = \mathrm{diag}\left(\boldsymbol{I}_{i-1}, \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix}, \boldsymbol{I}_{k-i}\right) \in \mathbb{R}^{(k+1)\times(k+1)}, \quad c_i^2 + s_i^2 = 1,$$

$$\boldsymbol{R}_k = \begin{bmatrix} \sigma_1 & \delta_1 & \varepsilon_1 & & \\ & \sigma_2 & \delta_2 & \ddots & \\ & & \ddots & \ddots & \varepsilon_{k-2} \\ & & & \sigma_{k-1} & \delta_{k-1} \\ & & & & \sigma_k \end{bmatrix}, \tag{4.181}$$

而且 $\beta_i \neq 0$ 蕴涵着 $\sigma_i \neq 0, i = 1, 2, \cdots, k$, 从而 $\boldsymbol{R}_k$ 是非奇异的.

令

$$\boldsymbol{G} = \boldsymbol{G}_k \boldsymbol{G}_{k-1} \cdots \boldsymbol{G}_2 \boldsymbol{G}_1, \quad \begin{bmatrix} \boldsymbol{t}_k \\ \rho_k \end{bmatrix} = \boldsymbol{G}(\beta \boldsymbol{e}_1), \quad \boldsymbol{t}_k = (\tau_1, \tau_2, \cdots, \tau_k)^{\mathrm{T}}, \tag{4.182}$$

则 $\boldsymbol{G}$ 是 $k+1$ 阶正交矩阵, 而且直接计算有

$$\begin{aligned} &\tau_1 = \beta c_1, \quad \tau_i = (-1)^{i-1} \beta s_1 s_2 \cdots s_{i-1} c_i, \quad i = 2, 3, \cdots, k, \\ &\rho_k = (-1)^k \beta s_1 s_2 \cdots s_k. \end{aligned} \tag{4.183}$$

这样, 利用式 (4.180) 和式 (4.182), 有

$$\begin{aligned} \|\widetilde{\boldsymbol{T}}_k \boldsymbol{z} - \beta \boldsymbol{e}_1\|_2^2 = \|\boldsymbol{G}(\widetilde{\boldsymbol{T}}_k \boldsymbol{z} - \beta \boldsymbol{e}_1)\|_2^2 &= \left\| \begin{bmatrix} \boldsymbol{R}_k \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{z} - \begin{bmatrix} \boldsymbol{t}_k \\ \rho_k \end{bmatrix} \right\|_2^2 \\ &= \|\boldsymbol{R}_k \boldsymbol{z} - \boldsymbol{t}_k\|_2^2 + \rho_k^2 \end{aligned}$$

对任意的 $\boldsymbol{z} \in \mathbb{R}^k$ 成立. 由此立即知道, 最小二乘问题 (4.179) 有唯一解

$$\boldsymbol{z}_k = \boldsymbol{R}_k^{-1} \boldsymbol{t}_k, \tag{4.184}$$

而且有

$$\|\widetilde{\boldsymbol{T}}_k \boldsymbol{z}_k - \beta \boldsymbol{e}_1\|_2 = |\rho_k|. \tag{4.185}$$

由式 (4.184) 求得 $\boldsymbol{z}_k$ 之后, 就可算出所求的 $\boldsymbol{x}_k$ 为 $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{z}_k$.

由式 (4.167) 和式 (4.185), 得

$$\|\boldsymbol{r}_k\|_2 = \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k\|_2 \leqslant \|\boldsymbol{V}_{k+1}\|_2 \|\beta \boldsymbol{e}_1 - \widetilde{\boldsymbol{T}}_k \boldsymbol{z}_k\|_2 = \|\boldsymbol{V}_{k+1}\|_2 |\rho_k|,$$

故在实际计算时, 可以用

$$|\rho_k| / \beta \leqslant \varepsilon \tag{4.186}$$

作为迭代终止的准则, 其中 $\varepsilon > 0$ 是给定的误差要求.

由式 (4.183) 可知, ρ_k 的值并不需要 $\boldsymbol{z}_k$ 和 $\boldsymbol{x}_k$ 的信息. 因此, 只需在 ρ_k 满足式 (4.186) 之后, 再去计算 $\boldsymbol{z}_k$ 和 $\boldsymbol{x}_k$ 即可.

将式 (4.184) 代入 $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{z}_k$, 得

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{R}_k^{-1} \boldsymbol{t}_k = \boldsymbol{x}_0 + \boldsymbol{P}_k \boldsymbol{t}_k, \tag{4.187}$$

式中: $\boldsymbol{P}_k = \boldsymbol{V}_k \boldsymbol{R}_k^{-1}$. 这样, 只要将 $\boldsymbol{P}_k$ 算出, 就可以通过式 (4.187) 来计算 $\boldsymbol{x}_k$. 令 $\boldsymbol{P}_k = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_k]$, 则比较 $\boldsymbol{P}_k \boldsymbol{R}_k = \boldsymbol{V}_k$ 两边的每一列, 得

$$\begin{aligned} &\sigma_1 \boldsymbol{p}_1 = \boldsymbol{v}_1, \\ &\delta_1 \boldsymbol{p}_1 + \sigma_2 \boldsymbol{p}_2 = \boldsymbol{v}_2, \end{aligned}$$

$$\varepsilon_{i-2}\boldsymbol{p}_{i-2}+\delta_{i-1}\boldsymbol{p}_{i-1}+\sigma_i\boldsymbol{p}_i=\boldsymbol{v}_i,\ i=3,4,\cdots,k.$$

由此可求得 $\boldsymbol{P}_k$ 的列向量为

$$\begin{cases}\boldsymbol{p}_1=\boldsymbol{v}_1/\sigma_1,\\ \boldsymbol{p}_2=(\boldsymbol{v}_2-\delta_1\boldsymbol{p}_1)/\sigma_2,\\ \boldsymbol{p}_i=(\boldsymbol{v}_i-\varepsilon_{i-2}\boldsymbol{p}_{i-2}-\delta_{i-1}\boldsymbol{p}_{i-1})/\sigma_i,\quad i=3,4,\cdots,k.\end{cases}\tag{4.188}$$

下面借助式 (4.183), 式 (4.187) 和式 (4.188) 导出计算 $\boldsymbol{x}_k$ 的递推公式.

首先注意到, 若非对称 Lanczos 分解的长度由 k 增加到 $k+1$, 则有

$$\widetilde{\boldsymbol{T}}_{k+1}=\left[\begin{array}{c|c}\widetilde{\boldsymbol{T}}_k & \widetilde{\boldsymbol{t}}_{k+1}\\ \hline \boldsymbol{0} & \beta_{k+1}\end{array}\right],\quad \widetilde{\boldsymbol{t}}_{k+1}=(0,\cdots,0,\gamma_k,\alpha_{k+1})^{\mathrm{T}},$$

于是有

$$\boldsymbol{R}_{k+1}=\left[\begin{array}{c|c}\boldsymbol{R}_k & \widetilde{\boldsymbol{r}}_{k+1}\\ \hline \boldsymbol{0} & \sigma_{k+1}\end{array}\right],\quad \widetilde{\boldsymbol{r}}_{k+1}=(0,\cdots,0,\varepsilon_{k-1},\delta_k)^{\mathrm{T}},\tag{4.189}$$

式中:

$$\begin{aligned}&\varepsilon_{k-1}=s_{k-1}\gamma_k,\quad \widehat{\gamma}_k=c_{k-1}\gamma_k,\\ &\delta_k=c_k\widehat{\gamma}_k+s_k\alpha_{k+1},\quad \widetilde{\alpha}_{k+1}=-s_k\widehat{\gamma}_k+c_k\alpha_{k+1}.\end{aligned}$$

上面的四个等式由如下 Givens 变换得到:

$$\widetilde{\boldsymbol{T}}_{k+1}=\left[\begin{array}{ccccccc}\alpha_1&\gamma_1&&&&&\\ \beta_1&\alpha_2&\gamma_2&&&&\\ &\ddots&\ddots&\ddots&&&\\ &&\beta_{k-3}&\alpha_{k-2}&\gamma_{k-2}&&\\ &&&\beta_{k-2}&\alpha_{k-1}&\gamma_{k-1}&\\ &&&&\beta_{k-1}&\alpha_k&\gamma_k\\ &&&&&\beta_k&\alpha_{k+1}\\ \hdashline &&&&&&\beta_{k+1}\end{array}\right]\xrightarrow{\boldsymbol{G}_1}\left[\begin{array}{ccccccc}\sigma_1&\delta_1&\varepsilon_1&&&&\\ &\widetilde{\alpha}_2&\widehat{\gamma}_2&&&&\\ &&\ddots&\ddots&\ddots&&\\ &&&\alpha_{k-2}&\gamma_{k-2}&&\\ &&&\beta_{k-2}&\alpha_{k-1}&\gamma_{k-1}&\\ &&&&\beta_{k-1}&\alpha_k&\gamma_k\\ &&&&&\beta_k&\alpha_{k+1}\\ \hdashline &&&&&&\beta_{k+1}\end{array}\right]$$

$$\xrightarrow{\boldsymbol{G}_2}\cdots\xrightarrow{\boldsymbol{G}_{k-2}}\left[\begin{array}{ccccccc}\sigma_1&\delta_1&\varepsilon_1&&&&\\ &\sigma_2&\delta_2&\varepsilon_2&&&\\ &&\ddots&\ddots&\ddots&&\\ &&&\sigma_{k-2}&\delta_{k-2}&\varepsilon_{k-2}&\\ &&&&\widetilde{\alpha}_{k-1}&\widehat{\gamma}_{k-1}&\\ &&&&\beta_{k-1}&\alpha_k&\gamma_k\\ &&&&&\beta_k&\alpha_{k+1}\\ \hdashline &&&&&&\beta_{k+1}\end{array}\right]$$

$$\xrightarrow{\boldsymbol{G}_{k-1}}\left[\begin{array}{ccccccc:c}
\sigma_1 & \delta_1 & \varepsilon_1 & & & & & \\
& \sigma_2 & \delta_2 & \varepsilon_2 & & & & \\
& & \ddots & \ddots & \ddots & & & \\
& & & \sigma_{k-2} & \delta_{k-2} & \varepsilon_{k-2} & & \\
& & & & \sigma_{k-1} & \delta_{k-1} & \varepsilon_{k-1} & \\
& & & & & \widetilde{\alpha}_k & \widehat{\gamma}_k & \\
& & & & & \beta_k & \alpha_{k+1} & \\ \hdashline
& & & & & & \beta_{k+1} &
\end{array}\right]\xrightarrow{\boldsymbol{G}_k}\left[\begin{array}{ccccccc:c}
\sigma_1 & \delta_1 & \varepsilon_1 & & & & & \\
& \sigma_2 & \delta_2 & \varepsilon_2 & & & & \\
& & \ddots & \ddots & \ddots & & & \\
& & & \sigma_{k-2} & \delta_{k-2} & \varepsilon_{k-2} & & \\
& & & & \sigma_{k-1} & \delta_{k-1} & \varepsilon_{k-1} & \\
& & & & & \sigma_k & \delta_k & \\
& & & & & & \widetilde{\alpha}_{k+1} & \\ \hdashline
& & & & & & \beta_{k+1} &
\end{array}\right],$$

即由

$$\left[\begin{array}{cc} c_{k-1} & s_{k-1} \\ -s_{k-1} & c_{k-1} \end{array}\right]\left[\begin{array}{c} 0 \\ \gamma_k \end{array}\right]=\left[\begin{array}{c} \varepsilon_{k-1} \\ \widehat{\gamma}_k \end{array}\right],\quad \left[\begin{array}{cc} c_k & s_k \\ -s_k & c_k \end{array}\right]\left[\begin{array}{c} \widehat{\gamma}_k \\ \alpha_{k+1} \end{array}\right]=\left[\begin{array}{c} \delta_k \\ \widetilde{\alpha}_{k+1} \end{array}\right]$$

得到. 而 σ_{k+1} 是在确定第 $k+1$ 个 Givens 变换 $\boldsymbol{G}_{k+1}$ 时得到的, 即计算 $c_{k+1}=\cos\theta_{k+1}$ 和 $s_{k+1}=\sin\theta_{k+1}$, 使得

$$\left[\begin{array}{cc} c_{k+1} & s_{k+1} \\ -s_{k+1} & c_{k+1} \end{array}\right]\left[\begin{array}{c} \widetilde{\alpha}_{k+1} \\ \beta_{k+1} \end{array}\right]=\left[\begin{array}{c} \sigma_{k+1} \\ 0 \end{array}\right]. \tag{4.190}$$

由式 (4.183) 可知 $\boldsymbol{t}_{k+1}=(\boldsymbol{t}_k^{\mathrm{T}},\tau_{k+1})^{\mathrm{T}}$, 其中

$$\tau_{k+1}=(-1)^k\beta s_1s_2\cdots s_kc_{k+1}=\rho_kc_{k+1}, \tag{4.191}$$

而

$$\rho_{k+1}=(-1)^{k+1}\beta s_1s_2\cdots s_ks_{k+1}=-\rho_ks_{k+1}. \tag{4.192}$$

由式 (4.188) 和式 (4.189), 有

$$\boldsymbol{P}_{k+1}=\boldsymbol{V}_{k+1}\boldsymbol{R}_{k+1}^{-1}=[\boldsymbol{V}_k,\boldsymbol{v}_{k+1}]\left[\begin{array}{c|c} \boldsymbol{R}_k^{-1} & -\boldsymbol{R}_k^{-1}\widetilde{\boldsymbol{r}}_{k+1}/\sigma_{k+1} \\ \hline \boldsymbol{0} & 1/\sigma_{k+1} \end{array}\right]=[\boldsymbol{P}_k,\boldsymbol{p}_{k+1}], \tag{4.193}$$

这里

$$\begin{aligned}\boldsymbol{p}_{k+1}&=(\boldsymbol{v}_{k+1}-\boldsymbol{V}_k\boldsymbol{R}_k^{-1}\widetilde{\boldsymbol{r}}_{k+1})/\sigma_{k+1}=(\boldsymbol{v}_{k+1}-\boldsymbol{P}_k\widetilde{\boldsymbol{r}}_{k+1})/\sigma_{k+1}\\&=(\boldsymbol{v}_{k+1}-\varepsilon_{k-1}\boldsymbol{p}_{k-1}-\delta_k\boldsymbol{p}_k)/\sigma_{k+1}.\end{aligned} \tag{4.194}$$

从而有

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_0+\boldsymbol{P}_{k+1}\boldsymbol{t}_{k+1}=\boldsymbol{x}_0+\left[\boldsymbol{P}_k,\boldsymbol{p}_{k+1}\right]\left[\begin{array}{c} \boldsymbol{t}_k \\ \tau_{k+1} \end{array}\right]=\boldsymbol{x}_k+\tau_{k+1}\boldsymbol{p}_{k+1}, \tag{4.195}$$

这就得到了 $\boldsymbol{x}_k$ 的递推公式.

算法 4.14 (QMRES 方法) 给定 n 阶非奇异的实矩阵 $\boldsymbol{A}$, n 维向量 $\boldsymbol{b}$, 初始向量 $\boldsymbol{x}_0$ 和允许误差 $\varepsilon>0$. 本算法计算向量 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2\leqslant\varepsilon$, 其中 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$; $\beta=\|\boldsymbol{r}_0\|_2$; $\boldsymbol{v}_1=\boldsymbol{r}_0/\beta$;

取一个 $\boldsymbol{w}_1\in\mathbb{R}^n$, 使得 $\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{v}_1=1$ (例如可取 $\boldsymbol{w}_1=\boldsymbol{v}_1$);

$\alpha_1=\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_1$; $\boldsymbol{v}=\boldsymbol{A}\boldsymbol{v}_1-\alpha_1\boldsymbol{v}_1$; $\boldsymbol{w}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{w}_1-\alpha_1\boldsymbol{w}_1$;

$\omega=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{v}$;

if $\omega=0$

$\quad\boldsymbol{x}_1=\boldsymbol{x}_0+\boldsymbol{r}_0/\alpha_1$; 结束

else

$\quad\beta_1=\sqrt{|\omega|}$; $\gamma_1=\omega/\beta_1$;

$\quad\boldsymbol{v}_2=\boldsymbol{v}/\beta_1$; $\boldsymbol{w}_2=\boldsymbol{w}/\gamma_1$;

end

确定 $c_1=\cos\theta_1$ 和 $s_1=\sin\theta_1$, 使得

$$\begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix}\begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix}=\begin{bmatrix} \sigma_1 \\ 0 \end{bmatrix};$$

$\boldsymbol{p}_1=\boldsymbol{v}_1/\sigma_1$; $\rho_1=-\beta s_1$; $\tau_1=\beta c_1$;

$\boldsymbol{p}_0=\boldsymbol{0}$; $c_0=1$; $s_0=0$; $\boldsymbol{x}_1=\boldsymbol{x}_0+\tau_1\boldsymbol{p}_1$; $k=1$;

while $(|\rho_k|>\beta\varepsilon)$

$\quad\alpha_{k+1}=\boldsymbol{w}_{k+1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_{k+1}$;

$\quad\boldsymbol{v}=\boldsymbol{A}\boldsymbol{v}_{k+1}-\alpha_{k+1}\boldsymbol{v}_{k+1}-\gamma_k\boldsymbol{v}_k$;

$\quad\boldsymbol{w}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{w}_{k+1}-\alpha_{k+1}\boldsymbol{w}_{k+1}-\beta_k\boldsymbol{w}_k$;

$\quad\omega_{k+1}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{v}$;

$\quad$**if** $\omega_{k+1}=0$

$\quad\quad$**stop**

$\quad$**else**

$\quad\quad\beta_{k+1}=\sqrt{|\omega_{k+1}|}$; $\gamma_{k+1}=\omega_{k+1}/\beta_{k+1}$;

$\quad\quad\boldsymbol{v}_{k+2}=\boldsymbol{v}/\beta_{k+1}$; $\boldsymbol{w}_{k+2}=\boldsymbol{w}/\gamma_{k+1}$;

$\quad$**end**

$\quad\varepsilon_{k-1}=s_{k-1}\gamma_k$; $\widehat{\gamma}_k=c_{k-1}\gamma_k$;

$\quad\delta_k=c_k\widehat{\gamma}_k+s_k\alpha_{k+1}$; $\widetilde{\alpha}_{k+1}=-s_k\widehat{\gamma}_k+c_k\alpha_{k+1}$;

$\quad$确定 $c_{k+1}=\cos\theta_{k+1}$和 $s_{k+1}=\sin\theta_{k+1}$, 使得

$$\begin{bmatrix} c_{k+1} & s_{k+1} \\ -s_{k+1} & c_{k+1} \end{bmatrix}\begin{bmatrix} \widetilde{\alpha}_{k+1} \\ \beta_{k+1} \end{bmatrix}=\begin{bmatrix} \sigma_{k+1} \\ 0 \end{bmatrix};$$

$\quad\tau_{k+1}=\rho_k c_{k+1}$; $\rho_{k+1}=-\rho_k s_{k+1}$;

$\quad\boldsymbol{p}_{k+1}=(\boldsymbol{v}_{k+1}-\varepsilon_{k-1}\boldsymbol{p}_{k-1}-\delta_k\boldsymbol{p}_k)/\sigma_{k+1}$;

$\quad\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\tau_{k+1}\boldsymbol{p}_{k+1}$;

$\quad k=k+1$;

end

注 4.7 (1) 该算法只需存储 10 个 n 维向量即可, 并不随着 k 的增加而增加.

(2) 同样可以考虑预处理 QMRES 方法 (记为 PQMRES). 选定预处理矩阵 $\boldsymbol{M}$, 然后将 QMRES 方法应用到求解方程组 $\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{b}$. 这只需要在算法 4.14 中将非对称 Lanczos 过程中的矩阵 $\boldsymbol{A}$ 替换成 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 及初始残差 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$ 替换成 $\boldsymbol{r}_0 = \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0)$ 即可.

例 4.11 给定线性方程组的系数矩阵 $\boldsymbol{A}$ 和右端项分别为

$$\boldsymbol{A} = \begin{bmatrix} 4 & -1 & & \\ -2 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -2 & 4 \end{bmatrix}, \quad b = \boldsymbol{A}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix},$$

其中 $n = 1000$. 显然, 该方程组的真解为 $\boldsymbol{x}^* = (1, 1, \cdots, 1)^{\mathrm{T}}$. 应用 QMRES 方法到该方程组上 ($\varepsilon = 10^{-10}$), 迭代在 25 步后满足终止条件, 计算得到的近似解 $\widehat{\boldsymbol{x}} = \boldsymbol{x}_{25}$ 满足终止条件, 但

$$\|\widehat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 3.0263 \times 10^{-6}, \quad \|\boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}}\|_2 = 9.8572 \times 10^{-6}.$$

若选取预处理矩阵为 $\boldsymbol{M} = \mathrm{tril}(\boldsymbol{A})$ (即 $\boldsymbol{A}$ 的下三角部分). 应用 PQMRES 方法到该方程组上 ($\varepsilon = 10^{-10}$), 迭代在 12 步后满足终止条件, 计算得到的近似解 $\widetilde{\boldsymbol{x}} = \boldsymbol{x}_{12}$ 满足终止条件, 但

$$\|\widetilde{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 3.4207 \times 10^{-5}, \quad \|\boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}\|_2 = 2.3047 \times 10^{-4}.$$

若将 GMRES 法应用于此例, 迭代在 40 步后满足终止条件, 计算得到的近似解 $\widetilde{\boldsymbol{x}} = \boldsymbol{x}_{40}$ 满足

$$\|\widetilde{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 1.5159 \times 10^{-9}, \quad \|\boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}\|_2 = 2.1777 \times 10^{-9}.$$

迭代过程的收敛轨迹如图 4.12 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量. 值得注意的是, Jacobi 预条件子对本例无效 (矩阵 $\boldsymbol{A}$ 的对角元素相同), Gauss–Seidel 预条件子虽然对迭代的加速很明显, 但在计算时间上没有任何优势.

4.6 LSQR 方法

本节讨论用 LSQR 方法求解 n 阶非对称方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

或当 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ $(m > n)$ 时, 求解最小二乘问题

$$\min \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2. \tag{4.196}$$

这类方法基于 Lanczos 双对角化的思想, 是求解非对称方程组尤其是超定方程组 (线性最小二乘问题) 的一类十分有效的子空间方法.

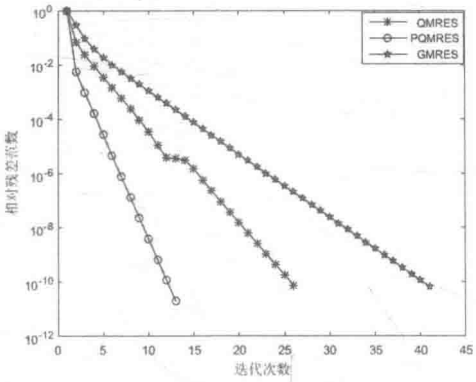

图 4.12 QMRES, PQMRES 和 GMRES 方法的收敛特性

4.6.1 Lanczos 双对角化方法

设 $\boldsymbol{A} \in \mathbb{R}^{m\times n}\,(m \geqslant n)$. 给定初始向量 $\boldsymbol{u}_1 \in \mathbb{R}^m$ 满足 $\|\boldsymbol{u}_1\|_2 = 1$. 双对角化方法产生 m 维向量组 $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots$ 和 n 维向量组 $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots$ 如下:

$$\alpha_i \boldsymbol{v}_i = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}_i - \beta_i \boldsymbol{v}_{i-1}\ (\beta_1 \boldsymbol{v}_0 = \boldsymbol{0}), \tag{4.197}$$

$$\beta_{i+1}\boldsymbol{u}_{i+1} = \boldsymbol{A}\boldsymbol{v}_i - \alpha_i \boldsymbol{u}_i,\quad i = 1, 2, \cdots, \tag{4.198}$$

式中: $\alpha_i, \beta_{i+1} \geqslant 0$, 且使 $\|\boldsymbol{u}_{i+1}\|_2 = \|\boldsymbol{v}_i\|_2 = 1$. 因此, 上述双对角方法可如下实现:

算法 4.15 (Lanczos 双对角化方法) 给定矩阵 $\boldsymbol{A} \in \mathbb{R}^{m\times n}\,(m \geqslant n)$, 初始向量 $\boldsymbol{u}_1 \in \mathbb{R}^m$ 满足 $\|\boldsymbol{u}_1\|_2 = 1$, $\boldsymbol{v}_0 = \boldsymbol{0}$ 和 $\beta_1 = 0$.

for $i = 1 : k$

$\widehat{\boldsymbol{v}}_i = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}_i - \beta_i \boldsymbol{v}_{i-1}$;

$\alpha_i = \|\widehat{\boldsymbol{v}}_i\|_2$; $\boldsymbol{v}_i = \widehat{\boldsymbol{v}}_i/\alpha_i$;

$\widehat{\boldsymbol{u}}_{i+1} = \boldsymbol{A}\boldsymbol{v}_i - \alpha_i \boldsymbol{u}_i$;

$\beta_{i+1} = \|\widehat{\boldsymbol{u}}_{i+1}\|_2$; $\boldsymbol{u}_{i+1} = \widehat{\boldsymbol{u}}_{i+1}/\beta_{i+1}$;

end

假设 $\alpha_i, \beta_{i+1} \neq 0, i = 1, 2, \cdots, k$, 则上述 Lanczos 双对角化过程可进行到第 k 步, 且若令

$$\boldsymbol{U}_k = \left[\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_k\right],\quad \boldsymbol{V}_k = \left[\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\right],$$

$$\boldsymbol{L}_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix},\quad \widetilde{\boldsymbol{L}}_k = \begin{bmatrix} \boldsymbol{L}_k \\ \beta_{k+1}\boldsymbol{e}_k^{\mathrm{T}} \end{bmatrix}, \tag{4.199}$$

则式 (4.197) 和式 (4.198) 可表示为

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}_k = \boldsymbol{V}_k\boldsymbol{L}_k^{\mathrm{T}}, \quad \boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{U}_k\boldsymbol{L}_k + \beta_{k+1}\boldsymbol{u}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}. \tag{4.200}$$

由此, 得

$$\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{U}_k\boldsymbol{L}_k + \beta_{k+1}\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{u}_{k+1}\boldsymbol{e}_k^{\mathrm{T}} = \boldsymbol{L}_k\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k. \tag{4.201}$$

若 $\alpha_{i+1} \neq 0$, 即式 (4.197) 对 $i = k+1$ 成立:

$$\alpha_{k+1}\boldsymbol{v}_{k+1} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}_{k+1} - \beta_{k+1}\boldsymbol{v}_k, \tag{4.202}$$

则由上式和式 (4.200), 得

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}_{k+1} = \boldsymbol{V}_k\widetilde{\boldsymbol{L}}_k^{\mathrm{T}} + \alpha_{k+1}\boldsymbol{v}_{k+1}\boldsymbol{e}_{k+1}^{\mathrm{T}}, \quad \boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{U}_{k+1}\widetilde{\boldsymbol{L}}_k. \tag{4.203}$$

由此, 得

$$\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}_{k+1} = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k\widetilde{\boldsymbol{L}}_k^{\mathrm{T}} + \alpha_{k+1}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}} = \widetilde{\boldsymbol{L}}_k^{\mathrm{T}}\boldsymbol{U}_{k+1}^{\mathrm{T}}\boldsymbol{U}_{k+1}. \tag{4.204}$$

基于以上分析, 可得下面的定理.

定理 4.11 由 Lanczos 双对角化过程产生的两个向量组 $\{\boldsymbol{u}_i\}_{i=1}^k$ 和 $\{\boldsymbol{v}_i\}_{i=1}^k$ 是规范正交的, 即

$$\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{U}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k = \boldsymbol{I}_k, \tag{4.205}$$

式中: $\boldsymbol{I}_k$ 为 k 阶单位矩阵.

证明 对 k 用归纳法. 当 $k = 1$ 时, 结论显然为真. 设式 (4.205) 对某个 $k \geqslant 1$ 成立. 利用归纳法假设, 由式 (4.201) 的第 2 个等式可知 $\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{u}_{k+1} = \boldsymbol{0}$, 即 $\boldsymbol{u}_{k+1}$ 与 $\boldsymbol{u}_i\,(i = 1, 2, \cdots, k)$ 均正交. 注意到 $\|\boldsymbol{u}_{k+1}\|_2 = 1$, 即得 $\boldsymbol{U}_{k+1}^{\mathrm{T}}\boldsymbol{U}_{k+1} = \boldsymbol{I}_{k+1}$. 再由式 (4.204) 的第 2 个等式可知 $\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{v}_{k+1} = \boldsymbol{0}$, 即知 $\boldsymbol{V}_{k+1}^{\mathrm{T}}\boldsymbol{V}_{k+1} = \boldsymbol{I}_{k+1}$. 证毕. □

由定理 4.11 可知, Lanczos 双对角化过程必在某个 $k \leqslant \min\{m, n\}$ 中断, 即或在式 (4.200) 中 $\beta_{k+1}\boldsymbol{u}_{k+1} = \boldsymbol{0}$, 或在式 (4.203) 中 $\alpha_{k+1}\boldsymbol{v}_{k+1} = \boldsymbol{0}$. 由此可知, 双对角化算法有两个中断状态:

(1) 由 $k \times k$ 阶矩阵 $\boldsymbol{L}_k$ 来表征 (见式 (4.200)):

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}_k = \boldsymbol{V}_k\boldsymbol{L}_k^{\mathrm{T}}, \quad \boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{U}_k\boldsymbol{L}_k, \quad \boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{U}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k = \boldsymbol{I}_k. \tag{4.206}$$

(2) 由 $(k+1) \times k$ 阶矩阵 $\widetilde{\boldsymbol{L}}_k$ 来表征 (见式 (4.203)):

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}_{k+1} = \boldsymbol{V}_k\widetilde{\boldsymbol{L}}_k^{\mathrm{T}}, \quad \boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{U}_{k+1}\widetilde{\boldsymbol{L}}_k, \quad \boldsymbol{U}_{k+1}^{\mathrm{T}}\boldsymbol{U}_{k+1} = \boldsymbol{I}_{k+1}, \quad \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k = \boldsymbol{I}_k. \tag{4.207}$$

定理 4.12 双对角化过程中断状态是 (1) 的充要条件是 $\boldsymbol{u}_1 \in \mathcal{R}(\boldsymbol{A})$, 而中断状态是 (2) 的充要条件是 $\boldsymbol{u}_1 \notin \mathcal{R}(\boldsymbol{A})$.

证明 考虑 $\widetilde{\boldsymbol{L}}_k$ 的奇异值分解. 注意到 $\widetilde{\boldsymbol{L}}_k$ 是列满秩的, 故有

$$\widetilde{\boldsymbol{L}}_k = \boldsymbol{P}_{k+1}\widetilde{\boldsymbol{\Sigma}}_k\boldsymbol{V}_k^{\mathrm{T}},\quad \boldsymbol{P}_{k+1}^{\mathrm{T}}\boldsymbol{P}_{k+1} = \boldsymbol{I}_{k+1},\quad \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k = \boldsymbol{I}_k, \tag{4.208}$$

式中: 矩阵 $\widetilde{\boldsymbol{\Sigma}}_k$ 为 $(k+1)\times k$ 阶对角阵,

$$\widetilde{\boldsymbol{\Sigma}}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \\ \hdashline 0 & \cdots & 0 \end{bmatrix},\ \sigma_i > 0,\ i = 1, 2, \cdots, k.$$

若式 (4.206) 成立, 则结合式 (4.207), 得

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1} = \boldsymbol{V}_k\boldsymbol{V}_k\widetilde{\boldsymbol{\Sigma}}_k^{\mathrm{T}},\quad \boldsymbol{A}\boldsymbol{V}_k\boldsymbol{V}_k = \boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1}\widetilde{\boldsymbol{\Sigma}}_k. \tag{4.209}$$

故有

$$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1} = \boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1}\widetilde{\boldsymbol{\Sigma}}_k\widetilde{\boldsymbol{\Sigma}}_k^{\mathrm{T}},\quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{V}_k\widetilde{\boldsymbol{\Sigma}}_k^{\mathrm{T}}\widetilde{\boldsymbol{\Sigma}}_k. \tag{4.210}$$

由式 (4.210) 可知, 当式 (4.207) 成立时, $\boldsymbol{A}$ 至少有 k 个奇异值 $\sigma_1, \sigma_2, \cdots, \sigma_k$ 非零, 且由式 (4.210) 的第 1 式可知

$$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1}\boldsymbol{e}_{k+1}) = \boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1}\widetilde{\boldsymbol{\Sigma}}_k\widetilde{\boldsymbol{\Sigma}}_k^{\mathrm{T}}\boldsymbol{e}_{k+1} = \boldsymbol{0},$$

即 $\boldsymbol{A}$ 至少有一个奇异值为零, 且 $\boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1}\boldsymbol{e}_{k+1}$ 是奇异向量. 再由式 (4.209) 的第 1 式可知

$$\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1}\boldsymbol{e}_{k+1}) = \boldsymbol{V}_k\boldsymbol{V}_k\widetilde{\boldsymbol{\Sigma}}_k^{\mathrm{T}}\boldsymbol{e}_{k+1} = \boldsymbol{0},$$

因此, $\boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1}\boldsymbol{e}_{k+1} \in \mathcal{N}(\boldsymbol{A}^{\mathrm{T}})$.

若 $\boldsymbol{u}_1 \in \mathcal{R}(\boldsymbol{A})$, 则由式 (4.198) 可知 $\boldsymbol{u}_i \in \mathcal{R}(\boldsymbol{A}), \forall\ i$. 于是有 $\boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1}\boldsymbol{e}_{k+1} \in \mathcal{R}(\boldsymbol{A})$. 但 $\mathcal{N}(\boldsymbol{A}^{\mathrm{T}})$ 是 $\mathcal{R}(\boldsymbol{A})$ 的正交补, 因此, $\boldsymbol{U}_{k+1}\boldsymbol{P}_{k+1}\boldsymbol{e}_{k+1} \notin \mathcal{N}(\boldsymbol{A}^{\mathrm{T}})$, 这就是说, 式 (4.207) 不可能成立. 所以必成立式 (4.206). 反之, 若 (1) 是中断状态, 即式 (4.206) 成立. 因 $\boldsymbol{L}_k$ 非奇异, 故 $\boldsymbol{U}_k = \boldsymbol{A}\boldsymbol{V}_k\boldsymbol{L}_k^{-1}$, 即 $\boldsymbol{u}_i \in \mathcal{R}(\boldsymbol{A}), i = 1, 2, \cdots, k$. 证毕. □

4.6.2 LSQR 算法

取 $\boldsymbol{u}_1 = \boldsymbol{r}_0/\beta_1\ (\beta_1 = \|\boldsymbol{r}_0\|_2)$ 作为双对角化算法 4.15 中的初始向量, 用以求解最小二乘问题 (4.196). 令

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k\boldsymbol{y}_k, \tag{4.211}$$

则由式 (4.203), 有

$$\begin{aligned} \boldsymbol{r}_k &= \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k = \boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_k \\ &= \beta_1\boldsymbol{u}_1 - \boldsymbol{U}_{k+1}\widetilde{\boldsymbol{L}}_k\boldsymbol{y}_k = \boldsymbol{U}_{k+1}(\beta_1\boldsymbol{e}_1 - \widetilde{\boldsymbol{L}}_k\boldsymbol{y}_k). \end{aligned}$$

若令

$$\boldsymbol{s}_{k+1} = \beta_1 \boldsymbol{e}_1 - \widetilde{\boldsymbol{L}}_k \boldsymbol{y}_k, \tag{4.212}$$

则

$$\boldsymbol{r}_k = \boldsymbol{U}_{k+1} \boldsymbol{s}_{k+1}. \tag{4.213}$$

因此极小化问题 (4.196) 等价于

$$\min \|\boldsymbol{s}_{k+1}\|_2 = \min \|\beta_1 \boldsymbol{e}_1 - \widetilde{\boldsymbol{L}}_k \boldsymbol{y}_k\|_2. \tag{4.214}$$

利用正交化方法求解最小二乘问题 (4.214):

作 $(k+1)\times(k+1)$ 阶矩阵 $\left[\widetilde{\boldsymbol{L}}_k, \beta_1 \boldsymbol{e}_1\right]$ 的 QR 分解:

$$\boldsymbol{G}_k \left[\widetilde{\boldsymbol{L}}_k, \beta_1 \boldsymbol{e}_1\right] = \begin{bmatrix} \boldsymbol{R}_k & \boldsymbol{z}_k \\ & \widetilde{\zeta}_{k+1} \end{bmatrix} \equiv \left[\begin{array}{ccccc:c} \rho_1 & \theta_2 & & & & \zeta_1 \\ & \rho_2 & \theta_3 & & & \zeta_2 \\ & & \ddots & \ddots & & \vdots \\ & & & \rho_{k-1} & \theta_k & \zeta_{k-1} \\ \hdashline & & & & \rho_k & \zeta_k \\ & & & & & \widetilde{\zeta}_{k+1} \end{array}\right], \tag{4.215}$$

式中: $\boldsymbol{G}_k \equiv \boldsymbol{G}_{k,k+1} \cdots \boldsymbol{G}_{2,3} \boldsymbol{G}_{1,2}$ 为 Givens 变换的乘积, 用以消去 $\widetilde{\boldsymbol{L}}_k$ 的次对角元 $\beta_2, \beta_3, \cdots$.

向量 $\boldsymbol{y}_k$ 和 $\boldsymbol{s}_{k+1}$ 可分别由

$$\boldsymbol{R}_k \boldsymbol{y}_k = \boldsymbol{z}_k, \quad \boldsymbol{z}_k = (\zeta_1, \zeta_2, \cdots, \zeta_k)^{\mathrm{T}} \tag{4.216}$$

和

$$\boldsymbol{s}_{k+1} = \boldsymbol{G}_k^{\mathrm{T}} \begin{bmatrix} \boldsymbol{0} \\ \widetilde{\zeta}_{k+1} \end{bmatrix} \tag{4.217}$$

求得.

直接求解式 (4.216), $\boldsymbol{y}_k$ 与 $\boldsymbol{y}_{k-1}$ 一般并无公共元素, 注意到 $[\boldsymbol{R}_k, \boldsymbol{z}_k]$ 可由 $[\boldsymbol{R}_{k-1}, \boldsymbol{z}_{k-1}]$ 加入新的一行和一列得到. 由此, 式 (4.211) 与式 (4.216) 有效结合的一个途径是引进矩阵 $\boldsymbol{H}_k$:

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{V}_k \boldsymbol{R}_k^{-1} \boldsymbol{z}_k \equiv \boldsymbol{x}_0 + \boldsymbol{H}_k \boldsymbol{z}_k, \tag{4.218}$$

其中 $\boldsymbol{H}_k = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_k]$ 的列可由 $\boldsymbol{H}_k \boldsymbol{R}_k = \boldsymbol{V}_k$ 逐列求得: 令 $\boldsymbol{h}_0 = \boldsymbol{0}$, 则成立

$$\theta_k \boldsymbol{h}_{k-1} + \rho_k \boldsymbol{h}_k = \boldsymbol{v}_k,$$

由此可得到递推式

$$\boldsymbol{h}_k = \frac{1}{\rho_k}(\boldsymbol{v}_k - \theta_k \boldsymbol{h}_{k-1}). \tag{4.219}$$

则由式 (4.218), 得

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \sum_{i=1}^{k-1} \zeta_i \boldsymbol{h}_i + \zeta_k \boldsymbol{h}_k = \boldsymbol{x}_{k-1} + \zeta_k \boldsymbol{h}_k. \tag{4.220}$$

现在需要确定式 (4.215) 的 QR 分解. Givens 变换 $\boldsymbol{G}_{k,k+1}$ 作用于前面 $k-1$ 步变换过的矩阵 $\left[\widetilde{\boldsymbol{L}}_k, \beta_1\boldsymbol{e}_1\right]$ 以消去 β_{k+1}, 这给出如下递推关系

$$\begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix} \begin{bmatrix} \widetilde{\rho}_k & 0 & \widetilde{\zeta}_k \\ \beta_{k+1} & \alpha_{k+1} & 0 \end{bmatrix} = \begin{bmatrix} \rho_k & \theta_{k+1} & \zeta_k \\ 0 & \widetilde{\rho}_{k+1} & \widetilde{\zeta}_{k+1} \end{bmatrix}, \tag{4.221}$$

其中 $\widetilde{\rho}_1 \equiv \alpha_1$, $\widetilde{\zeta}_1 \equiv \beta_1$. c_k, s_k 是 Givens 变换 $\boldsymbol{G}_{k,k+1}$ 中的元素. $\widetilde{\rho}_k$ 和 $\widetilde{\zeta}_k$ 是中间变量, 会被 ρ_k 和 ζ_k 所取代. 由 Givens 变换的定义, c_k, s_k 的选取应使得对向量 $[\widetilde{\rho}_k, \beta_{k+1}]^{\mathrm{T}}$ 作用后第 2 个分量为零, 且满足 $c_k^2 + s_k^2 = 1$. 为此, 必有

$$c_k = \frac{\widetilde{\rho}_k}{\sqrt{\widetilde{\rho}_k^2 + \beta_{k+1}^2}}, \quad s_k = \frac{\beta_{k+1}}{\sqrt{\widetilde{\rho}_k^2 + \beta_{k+1}^2}}. \tag{4.222}$$

由此, 式 (4.221) 即为

$$\begin{cases} \rho_k = \sqrt{\widetilde{\rho}_k^2 + \beta_{k+1}^2}, \ \theta_{k+1} = s_k\alpha_{k+1}, \ \zeta_k = c_k\widetilde{\zeta}_k, \\ \widetilde{\rho}_{k+1} = c_k\alpha_{k+1}, \ \widetilde{\zeta}_{k+1} = -s_k\widetilde{\zeta}_k, \ \widetilde{\rho}_1 \equiv \alpha_1, \ \widetilde{\zeta}_1 \equiv \beta_1. \end{cases} \tag{4.223}$$

此外, $\boldsymbol{G}_{k,k+1}$ 在式 (4.221) 中使用完之后即可丢弃而不必存储. 因此由 QR 分解算法产生 $\boldsymbol{R}_k$, $\boldsymbol{h}_k$ 和 $\widetilde{\zeta}_{k+1}$ 的工作量是很小的.

现将 LSQR 算法总结如下, 其中 $\boldsymbol{h}_k$ 用 $\boldsymbol{z}_k = \rho_k\boldsymbol{h}_k$ 代替.

算法 4.16 (LSQR 算法) 给定最小二乘问题 (4.196) 和容许误差限 $\varepsilon > 0$, 取初始向量 $\boldsymbol{x}_0$. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\beta_1 = \|\boldsymbol{r}_0\|_2$; $\boldsymbol{u}_1 = \boldsymbol{r}_0/\beta_1$;

$\widehat{\boldsymbol{v}}_1 = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}_1$; $\alpha_1 = \|\widehat{\boldsymbol{v}}_1\|_2$; $\boldsymbol{v}_1 = \widehat{\boldsymbol{v}}_1/\alpha_1$;

$\boldsymbol{z}_1 = \boldsymbol{v}_1$; $\widetilde{\zeta}_1 = \beta_1$; $\widetilde{\rho}_1 = \alpha_1$; $k = 1$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\| > \varepsilon)$

① 双对角化

$\widehat{\boldsymbol{u}}_{k+1} = \boldsymbol{A}\boldsymbol{v}_k - \alpha_k\boldsymbol{u}_k$;

$\beta_{k+1} = \|\widehat{\boldsymbol{u}}_{k+1}\|_2$; $\boldsymbol{u}_{k+1} = \widehat{\boldsymbol{u}}_{k+1}/\beta_{k+1}$;

$\widehat{\boldsymbol{v}}_{k+1} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}_{k+1} - \beta_{k+1}\boldsymbol{v}_k$;

$\alpha_{k+1} = \|\widehat{\boldsymbol{v}}_{k+1}\|_2$; $\boldsymbol{v}_{k+1} = \widehat{\boldsymbol{v}}_{k+1}/\alpha_{k+1}$;

② 构造和使用正交变换

$\rho_k = \sqrt{\widetilde{\rho}_k^2 + \beta_{k+1}^2}$; $\quad c_k = \widetilde{\rho}_k/\rho_k$; $s_k = \beta_{k+1}/\rho_k$;

$\theta_{k+1} = s_k\alpha_{k+1}$; $\widetilde{\rho}_{k+1} = c_k\alpha_{k+1}$;

$\zeta_k = c_k\widetilde{\zeta}_k$; $\widetilde{\zeta}_{k+1} = -s_k\widetilde{\zeta}_k$;

③ 更新 $\boldsymbol{x}_k$, $\boldsymbol{r}_k$ 和 $\boldsymbol{z}_k$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + (\zeta_k/\rho_k)\boldsymbol{z}_k;$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1} = \boldsymbol{r}_k - (\zeta_k/\rho_k)\boldsymbol{A}\boldsymbol{z}_k;$$

$$\boldsymbol{z}_{k+1} = \boldsymbol{v}_{k+1} - (\theta_{k+1}/\rho_k)\boldsymbol{z}_k;$$

④ 置 $k = k + 1$;

end

注 4.8 由双对角化过程可知

$$\boldsymbol{v}_k \in \mathrm{span}\{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}_1, (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}_1, \cdots, (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{k-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}_1\} \equiv \mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{u}_1).$$

因此, 由式 (4.211) 和 $\boldsymbol{u}_1 = \boldsymbol{r}_0/\beta_1$ 可知 LSQR 算法产生的迭代序列 $\boldsymbol{x}_k$ 满足

$$\begin{aligned}\boldsymbol{x}_k &\in \boldsymbol{x}_0 + \mathrm{span}\{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0, (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0, \cdots, (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{k-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0\} \\ &\equiv \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_0),\end{aligned} \tag{4.224}$$

且使

$$\|\boldsymbol{r}_k\|_2 = \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k\|_2 = \min\{\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2 : \boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b})\}. \tag{4.225}$$

LSQR 方法的 MATLAB 程序如下:

```
%LSQR方法程序-mlsqr.m
function [x,k,time,res,resvec]=mlsqr(A,b,x,max_it,tol)
tic; r=b-A*x; mr=norm(r); u=r/mr;
v=A'*u; alpha=norm(v); v=v/alpha;
z=v; zetat=mr; rhot=alpha;
resvec(1)=1; k=0;
while (k<=max_it)
    k=k+1;
    u=A*v-alpha*u; beta=norm(u); u=u/beta;
    v=A'*u-beta*v; alpha=norm(v); v=v/alpha;
    rho=sqrt(rhot^2+beta^2);
    c=rhot/rho; s=beta/rho;
    theta=s*alpha; rhot=c*alpha;
    zeta=c*zetat; zetat=-s*zetat;
    x=x+(zeta/rho)*z; r=b-A*x;
    z=v-(theta/rho)*z;
    res=norm(r)/mr; resvec(k+1)=res;
    if (res<tol), break; end
end
time=toc;
```

例 4.12 假设线性方程组的系数矩阵 $\boldsymbol{A}$ 为

$$\boldsymbol{A}=\begin{bmatrix} 4 & -1 & & & \\ -2 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -2 & 4 & -1 \\ & & & -2 & 4 \end{bmatrix}.$$

再假设真解 $\boldsymbol{x}^*$ 是分量都为 1 的向量, 从而该方程组的右端项为 $\boldsymbol{b}=\boldsymbol{A}\boldsymbol{x}^*$. 将 LSQR 算法应用到该线性方程组上, 迭代在 76 步后收敛 ($\varepsilon=10^{-10}$), 计算得到的近似解 $\widehat{\boldsymbol{x}}$ 和真解 $\boldsymbol{x}^*$ 之间的绝对值误差为

$$\|\widehat{\boldsymbol{x}}-\boldsymbol{x}^*\|_2=1.8330\times10^{-9},$$

但计算解 $\widehat{\boldsymbol{x}}$ 的残量满足

$$\|\boldsymbol{b}-\boldsymbol{A}\widehat{\boldsymbol{x}}\|_2=2.5663\times10^{-9}.$$

此外, 将 GMRES 方法应用于求解此例, 迭代 40 步满足终止准则. 对于此例, 虽然 LSQR 方法的迭代次数比 GMRES 方法多了将近一倍, 但计算时间不到 GMRES 方法的一半. 迭代过程的收敛轨迹如图 4.13 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

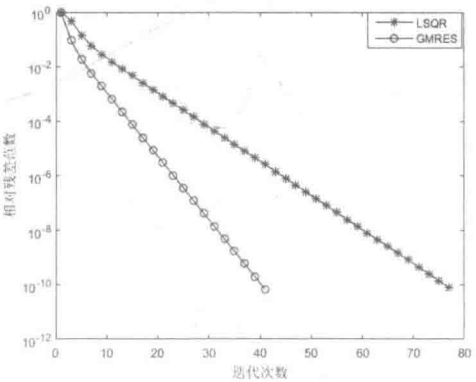

图 4.13 LSQR 算法的收敛特性

4.7 广义共轭残量法

本节考虑求解线性方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},\quad \boldsymbol{A}\in\mathbb{R}^{n\times n},\quad \boldsymbol{b}\in\mathbb{R}^n \tag{4.226}$$

的广义共轭残量法 (简记为 GCR 方法), 其中系数矩阵 $\boldsymbol{A}$ 是正定的, 即其对称部分 $\dfrac{1}{2}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})$ 是对称正定的.

4.7.1 GCR 方法

类似于对称正定方程组的共轭梯度法, 构造如下的广义共轭残量法.

算法 4.17 (GCR 方法) 给定方程组 (4.226) 和容许误差限 $\varepsilon > 0$, 取初始向量 $\boldsymbol{x}_0$. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{p}_0 = \boldsymbol{r}_0$; $k = 0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\| > \varepsilon)$

$k = k + 1$;

$$\alpha_k = \frac{(\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_k)}{(\boldsymbol{A}\boldsymbol{p}_k, \boldsymbol{A}\boldsymbol{p}_k)};$$

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$; $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k$;

for $i = 0, 1, \cdots, k$

$$\beta_i^{(k)} = -\frac{(\boldsymbol{A}\boldsymbol{r}_{k+1}, \boldsymbol{A}\boldsymbol{p}_i)}{(\boldsymbol{A}\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_i)};$$

end

$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \sum_{i=0}^{k} \beta_i^{(k)} \boldsymbol{p}_i;$$

$$\boldsymbol{A}\boldsymbol{p}_{k+1} = \boldsymbol{A}\boldsymbol{r}_{k+1} + \sum_{i=0}^{k} \beta_i^{(k)} \boldsymbol{A}\boldsymbol{p}_i;$$

end

从算法 4.17 可知, 计算过程中需要存储每一步迭代的 $\boldsymbol{p}_i$ 和 $\boldsymbol{A}\boldsymbol{p}_i$ $(i = 0, 1, \cdots, k)$. 其 MATLAB 程序如下:

```
%广义共轭残量法(GCR方法)程序-gcr.m
function [x,k,time,res,resvec]=gcr(A,b,x,max_it,tol)
tic; n=length(b); r=b-A*x; p=r; mr=norm(r);
resvec(1)=1; AP=zeros(n,max_it+1);
P=zeros(n,max_it+1);
AP(:,1)=A*p; P(:,1)=p; k=0;
while (k<max_it)
    k=k+1;
    alpha=(r'*AP(:,k))/(AP(:,k)'*AP(:,k));
    x=x+alpha*P(:,k);
    r=r-alpha*AP(:,k);
    Ar=A*r; s1=zeros(n,1); s2=zeros(n,1);
    for (i=1:k)
        b(i)=-(Ar'*AP(:,i))/(AP(:,i)'*AP(:,i));
        s1=s1+b(i)*P(:,i);
        s2=s2+b(i)*AP(:,i);
```

```
        end
        P(:,k+1)=r+s1;
        AP(:,k+1)=Ar+s2;
        res=norm(r)/mr; resvec(k+1)=res;
        if (res<tol), break; end
    end
    time=toc;
```

下面列出一些与共轭梯度法 (CG) 相类似的性质.

定理 4.13 设 $\boldsymbol{A}$ 正定, 即 $\boldsymbol{H}=\frac{1}{2}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})$ 是对称正定的. 则由算法 4.17 产生的 $\{\boldsymbol{x}_k\}$, $\{\boldsymbol{r}_k\}$ 和 $\{\boldsymbol{p}_k\}$ 具有下列性质:

(1) $(\boldsymbol{A}\boldsymbol{p}_k,\boldsymbol{A}\boldsymbol{p}_l)=0,\ k\neq l.$

(2) $(\boldsymbol{r}_k,\boldsymbol{A}\boldsymbol{p}_l)=0,\ k>l.$

(3) $(\boldsymbol{r}_k,\boldsymbol{A}\boldsymbol{p}_l)=(\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{p}_l),\ k\leqslant l.$

(4) $(\boldsymbol{r}_k,\boldsymbol{A}\boldsymbol{p}_k)=(\boldsymbol{r}_k,\boldsymbol{A}\boldsymbol{r}_k).$

(5) $(\boldsymbol{r}_k,\boldsymbol{A}\boldsymbol{r}_l)=0,\ k>l.$

(6) $\mathrm{span}\{\boldsymbol{p}_0,\boldsymbol{p}_1,\cdots,\boldsymbol{p}_k\}=\mathrm{span}\{\boldsymbol{p}_0,\boldsymbol{A}\boldsymbol{p}_0,\cdots,\boldsymbol{A}^k\boldsymbol{p}_0\}$
$=\mathrm{span}\{\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{r}_0,\cdots,\boldsymbol{A}^k\boldsymbol{r}_0\}=\mathrm{span}\{\boldsymbol{r}_0,\boldsymbol{r}_1,\cdots,\boldsymbol{r}_k\}.$

(7) 若 $\boldsymbol{r}_k\neq\boldsymbol{0}$, 则 $\boldsymbol{p}_k\neq\boldsymbol{0}$.

(8) $\boldsymbol{x}_{k+1}\in\mathcal{K}\equiv\boldsymbol{x}_0+\mathrm{span}\{\boldsymbol{p}_0,\boldsymbol{p}_1,\cdots,\boldsymbol{p}_k\}$, 且使其残量范数 $f(\boldsymbol{x}_{k+1})\equiv\|\boldsymbol{r}_{k+1}\|_2=\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_{k+1}\|_2$ 在仿射空间 $\mathcal{K}$ 中达到极小.

证明 (1) 首先验证 $(\boldsymbol{A}\boldsymbol{p}_1,\boldsymbol{A}\boldsymbol{p}_0)=0$ 成立. 事实上, 由 $\beta_i^{(k)}$ 更新规则, 有

$$\begin{aligned}(\boldsymbol{A}\boldsymbol{p}_1,\boldsymbol{A}\boldsymbol{p}_0)&=\big(\boldsymbol{A}(\boldsymbol{r}_1+\beta_0^{(0)}\boldsymbol{p}_0),\boldsymbol{A}\boldsymbol{p}_0\big)=\big(\boldsymbol{A}\boldsymbol{r}_1,\boldsymbol{A}\boldsymbol{p}_0\big)+\beta_0^{(0)}\big(\boldsymbol{A}\boldsymbol{p}_0,\boldsymbol{A}\boldsymbol{p}_0\big)\\&=\big(\boldsymbol{A}\boldsymbol{r}_1,\boldsymbol{A}\boldsymbol{p}_0\big)-\frac{(\boldsymbol{A}\boldsymbol{r}_1,\boldsymbol{A}\boldsymbol{p}_0)}{(\boldsymbol{A}\boldsymbol{p}_0,\boldsymbol{A}\boldsymbol{p}_0)}\big(\boldsymbol{A}\boldsymbol{p}_0,\boldsymbol{A}\boldsymbol{p}_0\big)=0.\end{aligned}$$

利用归纳法, 由 $\beta_i^{(k)}$ 的取法不难验证, 对任意的 $k\neq l$ 都有 $(\boldsymbol{A}\boldsymbol{p}_k,\boldsymbol{A}\boldsymbol{p}_l)=0$.

(2) 对 k 使用归纳法. 当 $k=0$ 时无需证明. 设 $k\leqslant s$ 时, 结论为真, 则

$$(\boldsymbol{r}_{s+1},\boldsymbol{A}\boldsymbol{p}_l)=(\boldsymbol{r}_s,\boldsymbol{A}\boldsymbol{p}_l)-\alpha_s(\boldsymbol{A}\boldsymbol{p}_s,\boldsymbol{A}\boldsymbol{p}_l).$$

若 $s>l$, 由归纳法假设及结论 (1), 上式右端为 0. 若 $s=l$, 由 α_s 的表达式, 上式右端也为 0.

(3) 对 $k\,(k\leqslant l)$ 使用归纳法证明. 当 $k=0$ 时, 结论平凡地成立. 设 $k=s<l$ 时结论成立, 则由结论 (1) 和归纳法假设, 有

$$(\boldsymbol{r}_{s+1},\boldsymbol{A}\boldsymbol{p}_l)=(\boldsymbol{r}_s,\boldsymbol{A}\boldsymbol{p}_l)-\alpha_s(\boldsymbol{A}\boldsymbol{p}_s,\boldsymbol{A}\boldsymbol{p}_l)=(\boldsymbol{r}_s,\boldsymbol{A}\boldsymbol{p}_l)=(\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{p}_l).$$

(4) 利用结论 (2), 有

$$(\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_k) = (\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{r}_k) + \sum_{i=0}^{k-1} \beta_i^{(k-1)} (\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_i) = (\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{r}_k).$$

(5) 由算法 4.17, 有

$$\boldsymbol{r}_l = \boldsymbol{p}_l - \sum_{i=0}^{l-1} \beta_i^{(l-1)} \boldsymbol{p}_i.$$

应用结论 (2), 对 $k > l$, 有

$$(\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{r}_l) = (\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_l) - \sum_{i=0}^{l-1} \beta_i^{(l-1)} (\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_i) = 0.$$

(6) 对 k 用归纳法证明. 当 $k = 0$ 时, 结论显然成立. 设对 $k \leqslant s$ 结论成立, 即

$$\boldsymbol{r}_i, \boldsymbol{p}_i \in \operatorname{span}\{\boldsymbol{p}_0, \boldsymbol{A}\boldsymbol{p}_0, \cdots, \boldsymbol{A}^s \boldsymbol{p}_0\}, \ i = 0, 1, \cdots, s.$$

由算法 4.17, 有

$$\boldsymbol{p}_{s+1} = \boldsymbol{r}_{s+1} + \sum_{i=0}^{s} \beta_i^{(s)} \boldsymbol{p}_i = \boldsymbol{r}_s - \alpha_s \boldsymbol{A}\boldsymbol{p}_s + \sum_{i=0}^{s} \beta_i^{(s)} \boldsymbol{p}_i,$$

故有

$$\boldsymbol{p}_{s+1} \in \operatorname{span}\{\boldsymbol{p}_0, \boldsymbol{A}\boldsymbol{p}_0, \cdots, \boldsymbol{A}^{s+1} \boldsymbol{p}_0\}.$$

由此, 得

$$\operatorname{span}\{\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_{s+1}\} \subset \operatorname{span}\{\boldsymbol{p}_0, \boldsymbol{A}\boldsymbol{p}_0, \cdots, \boldsymbol{A}^{s+1} \boldsymbol{p}_0\}.$$

但由结论 (1) 可知 $\{\boldsymbol{p}_i\}$ 线性无关, 故

$$\operatorname{span}\{\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_{s+1}\} = \operatorname{span}\{\boldsymbol{p}_0, \boldsymbol{A}\boldsymbol{p}_0, \cdots, \boldsymbol{A}^{s+1} \boldsymbol{p}_0\}.$$

同理, 可证

$$\operatorname{span}\{\boldsymbol{r}_0, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_k\} \subset \operatorname{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^k \boldsymbol{r}_0\}.$$

(7) 注意到 $\boldsymbol{H} = \dfrac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})$ 是对称正定的, 若 $\boldsymbol{r}_k \neq \boldsymbol{0}$, 则由结论 (4), 有

$$(\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_k) = (\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{r}_k) = (\boldsymbol{r}_k, \boldsymbol{H}\boldsymbol{r}_k) > 0,$$

故 $(\boldsymbol{r}_k, \boldsymbol{A}\boldsymbol{p}_k) \neq 0$, 由此 $\boldsymbol{p}_k \neq \boldsymbol{0}$.

(8) 由算法 4.17, 有

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_0 + \sum_{i=0}^{k} \alpha_i \boldsymbol{p}_i \in \boldsymbol{x}_0 + \operatorname{span}\{\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_k\}.$$

因此 $f^2(\boldsymbol{x}_{k+1}) = \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1}\|_2^2$ 是 $(\alpha_0, \alpha_1, \cdots, \alpha_k)$ 的二次泛函. 应用结论 (1), 得

$$f^2(\boldsymbol{x}_{k+1}) = \left\| \boldsymbol{r}_0 - \sum_{i=0}^{k} \alpha_i \boldsymbol{A}\boldsymbol{p}_i \right\|_2^2$$
$$= \|\boldsymbol{r}_0\|_2^2 - 2\sum_{i=0}^{k} \alpha_i (\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{p}_i) + \sum_{i=0}^{k} \alpha_i^2 (\boldsymbol{A}\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_i).$$

由 $\dfrac{\partial f^2(\boldsymbol{x}_{k+1})}{\partial \alpha_i} = 0\,(i = 0, 1, \cdots, k)$, 得

$$\alpha_i = \frac{(\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{p}_i)}{(\boldsymbol{A}\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_i)} = \frac{(\boldsymbol{r}_i, \boldsymbol{A}\boldsymbol{p}_i)}{(\boldsymbol{A}\boldsymbol{p}_i, \boldsymbol{A}\boldsymbol{p}_i)},\ i = 0, 1, \cdots, k.$$

这就证明了 $f(\boldsymbol{x}_{k+1})$ 在仿射空间 $\mathcal{K}$ 中被极小化. 证毕. □

由定理 4.13 可推得 GCR 方法的有限迭代终止性质.

推论 4.5 *设矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 是正定的 (即其对称部分是对称正定的), 则 GCR 方法至多迭代 n 步就给出精确解.*

证明 若当 $k \leqslant n-1$ 时有 $\boldsymbol{r}_k = \boldsymbol{0}$, 这表明 $\boldsymbol{x}_k$ 已是精确解. 若 $\boldsymbol{r}_k \neq \boldsymbol{0}$, $k = 0, 1, \cdots, n-1$, 则由定理 4.13 的结论 (7) 可知 $\boldsymbol{p}_k \neq \boldsymbol{0}$. 再由结论 (1) 知向量组 $\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_{n-1}$ 线性无关, 因此 $\mathrm{span}\{\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_{n-1}\} = \mathbb{R}^n$. 再由定理 4.13 的结论 (8), $\boldsymbol{x}_n$ 的残量范数有下述极小化性质:

$$\|\boldsymbol{r}_n\|_2 = \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n\|_2 = \min_{\alpha_i} \left\| \boldsymbol{r}_0 - \sum_{k=0}^{n-1} \alpha_k \boldsymbol{A}\boldsymbol{p}_k \right\|_2.$$

因 $\boldsymbol{A}$ 非奇异, 故 $\mathrm{span}\{\boldsymbol{A}\boldsymbol{p}_0, \boldsymbol{A}\boldsymbol{p}_1, \cdots, \boldsymbol{A}\boldsymbol{p}_{n-1}\} = \mathbb{R}^n$. 从而 $\|\boldsymbol{r}_n\|_2 = 0$, 故 $\boldsymbol{x}_n = \boldsymbol{A}^{-1}\boldsymbol{b}$. 证毕. □

现在考虑 GCR 方法作为迭代法时的迭代误差估计和收敛性. 令 $\mathcal{P}_k$ 是次数不超过 k 且满足 $p_k(0) = 1$ 的实系数多项式 $p_k(t)$ 的集合.

定理 4.14 *设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 正定, 即 $\boldsymbol{H} = \dfrac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})$ 是对称正定的. 则由算法 4.17 产生的残量序列 $\{\boldsymbol{r}_k\}$ 满足*

$$\|\boldsymbol{r}_k\|_2 \leqslant \min_{p_k \in \mathcal{P}_k} \|p_k(\boldsymbol{A})\|_2 \|\boldsymbol{r}_0\|_2 \leqslant \left[1 - \frac{\lambda_{\min}^2(\boldsymbol{H})}{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}\right]^{k/2} \|\boldsymbol{r}_0\|_2. \tag{4.227}$$

因此, GCR 方法收敛. 若 $\boldsymbol{A}$ 有完全特征向量集, 则

$$\|\boldsymbol{r}_k\|_2 \leqslant \kappa_2(\boldsymbol{T}) M_k \|\boldsymbol{r}_0\|_2,$$

式中: $\boldsymbol{T}$ 为 $\boldsymbol{A}$ 的 Jordan 标准形的变换矩阵, 而

$$M_k = \min_{p_k \in \mathcal{P}_k} \max_{\lambda \in \sigma(A)} |p_k(\lambda)|.$$

进一步, 若 $\boldsymbol{A}$ 是规范的, 则

$$\|\boldsymbol{r}_k\|_2 \leqslant M_k\|\boldsymbol{r}_0\|_2.$$

证明 由算法 4.17 和定理 4.13 的结论 (6), GCR 方法产生的残量 $\boldsymbol{r}_k$ 可表示为 $\boldsymbol{r}_k = \widetilde{p}_k(\boldsymbol{A})\boldsymbol{r}_0$, $\widetilde{p}_k \in \mathcal{P}_k$. 而任意的 $\boldsymbol{x} \in \boldsymbol{x}_0 + \mathrm{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\}$, 其对应的残量 $\boldsymbol{r}$ 可表示为 $\boldsymbol{r} = p_k(\boldsymbol{A})\boldsymbol{r}_0$, $p_k \in \mathcal{P}_k$. 于是由定理 4.13 的结论 (8), 有

$$\|\boldsymbol{r}_k\|_2 = \min_{p_k\in\mathcal{P}_k} \|p_k(\boldsymbol{A})\boldsymbol{r}_0\|_2 \leqslant \min_{p_k\in\mathcal{P}_k} \|p_k(\boldsymbol{A})\|_2\|\boldsymbol{r}_0\|_2,$$

这就证明了式 (4.227) 的第 1 个不等式.

现证明第 2 个不等式. 令

$$p_1(t) = 1 + \alpha t \in \mathcal{P}_1,$$

则 $p_1(t)^k \in \mathcal{P}_k$. 因此

$$\min_{p_k\in\mathcal{P}_k} \|p_k(\boldsymbol{A})\|_2 \leqslant \|p_1(\boldsymbol{A})^k\|_2 \leqslant \|p_1(\boldsymbol{A})\|_2^k. \tag{4.228}$$

下面估计 $\|p_1(\boldsymbol{A})\|_2$. 有

$$\begin{aligned}\|p_1(\boldsymbol{A})\|_2^2 &= \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{((\boldsymbol{I}+\alpha\boldsymbol{A})\boldsymbol{x}, (\boldsymbol{I}+\alpha\boldsymbol{A})\boldsymbol{x}))}{(\boldsymbol{x},\boldsymbol{x})} \\ &= \max_{\boldsymbol{x}\neq\boldsymbol{0}} \left\{1 + 2\alpha\frac{(\boldsymbol{x},\boldsymbol{A}\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} + \alpha^2\frac{(\boldsymbol{A}\boldsymbol{x},\boldsymbol{A}\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}\right\}. \end{aligned} \tag{4.229}$$

注意到

$$\frac{(\boldsymbol{A}\boldsymbol{x},\boldsymbol{A}\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} = \frac{(\boldsymbol{x},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} \leqslant \lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}) = \|\boldsymbol{A}\|_2^2.$$

利用 $\boldsymbol{H}$ 的对称正定性, 有

$$\frac{(\boldsymbol{x},\boldsymbol{A}\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} = \frac{1}{2}\left[\frac{(\boldsymbol{x},\boldsymbol{A}\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} + \frac{(\boldsymbol{x},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})}\right] = \frac{(\boldsymbol{x},\boldsymbol{H}\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} \geqslant \lambda_{\min}(\boldsymbol{H}) > 0.$$

若选取 $\alpha < 0$, 则由式 (4.229), 得

$$\|p_1(\boldsymbol{A})\|_2^2 \leqslant 1 + 2\lambda_{\min}(\boldsymbol{H})\,\alpha + \lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\,\alpha^2. \tag{4.230}$$

上式右端当

$$\alpha = -\frac{\lambda_{\min}(\boldsymbol{H})}{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}$$

时达到极小. 将这个 α 值代入式 (4.230), 得

$$\|p_1(\boldsymbol{A})\|_2^2 \leqslant 1 - \frac{\lambda_{\min}^2(\boldsymbol{H})}{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}.$$

由上式及式 (4.228) 即得式 (4.227) 的第 2 个不等式.

若 $\boldsymbol{A}$ 有完全特征向量集, 即 $\boldsymbol{A}$ 可对角化, 则 $\boldsymbol{\Lambda}=\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}$ 是对角矩阵, 则

$$\begin{aligned}\|\boldsymbol{r}_k\|_2 &= \min_{p_k\in\mathcal{P}_k}\|\boldsymbol{T}p_k(\boldsymbol{\Lambda})\boldsymbol{T}^{-1}\boldsymbol{r}_0\|_2 \leqslant \|\boldsymbol{T}\|_2\|\boldsymbol{T}^{-1}\|_2\min_{p_k\in\mathcal{P}_k}\|p_k(\boldsymbol{\Lambda})\|_2\|\boldsymbol{r}_0\|_2\\ &= \kappa_2(\boldsymbol{T})\min_{p_k\in\mathcal{P}_k}\max_{\lambda\in\sigma(\boldsymbol{A})}|p_k(\lambda)|\cdot\|\boldsymbol{r}_0\|_2.\end{aligned}$$

进一步, 若 $\boldsymbol{A}$ 还是规范的, 则可使 $\kappa_2(\boldsymbol{T})=1$. 证毕. □

例 4.13 考虑正定方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 其中

$$\boldsymbol{A}=\begin{bmatrix}12 & 3 & 2 & 1 & & & \\ -3 & 12 & 3 & 2 & 1 & & \\ -2 & -3 & \ddots & \ddots & \ddots & \ddots & \\ -1 & -2 & \ddots & \ddots & \ddots & \ddots & 1\\ & \ddots & \ddots & \ddots & \ddots & 3 & 2\\ & & -1 & -2 & -3 & 12 & 3\\ & & & -1 & -2 & -3 & 12\end{bmatrix}\in\mathbb{R}^{n\times n},\quad \boldsymbol{b}=\boldsymbol{A}\begin{bmatrix}1\\1\\\vdots\\\vdots\\\vdots\\1\\1\end{bmatrix}\in\mathbb{R}^n.$$

取 $n=1000$, 将 GCR 方法应用到该线性方程组上, 迭代在 20 步后收敛 ($\varepsilon=10^{-10}$), 计算得到的近似解 $\widehat{\boldsymbol{x}}$ 和真解 $\boldsymbol{x}^*$ 之间的绝对值误差为

$$\|\widehat{\boldsymbol{x}}-\boldsymbol{x}^*\|_2=2.0725\times10^{-9},$$

但计算解 $\widehat{\boldsymbol{x}}$ 的残量满足

$$\|\boldsymbol{b}-\boldsymbol{A}\widehat{\boldsymbol{x}}\|_2=2.7746\times10^{-8}.$$

迭代过程的收敛轨迹如图 4.14 所示, 其中横坐标为迭代步数 k, 纵坐标为 $\lg\|\boldsymbol{r}_k\|_2$, 其

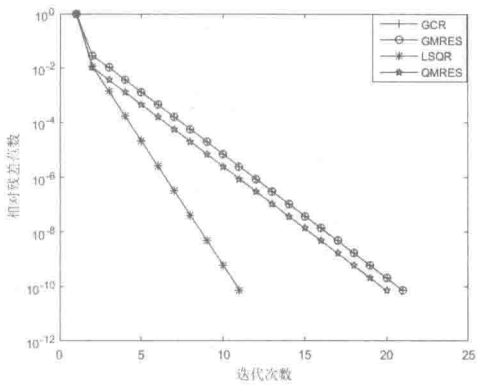

图 4.14 GCR 方法的收敛特性

中 $\boldsymbol{r}_k$ 是第 k 步得到的残量. 此外, 还给出了 GCR 方法与 GMRES, LSQR 及 QMRES 方法的比较, 结果如表 4.3 所示.

从表中可以看出, 对于此例, GCR 算法不如 LSQR 算法有效.

表 4.3 GCR 方法的数值结果

算法	迭代次数	CPU时间	相对残差	绝对误差
GCR	20	0.0028	7.3094e-11	2.0725e-09
GMRES	20	0.0018	7.3094e-11	2.0725e-09
LSQR	10	0.0007	7.7475e-11	2.1967e-09
QMRES	19	0.0010	7.3094e-11	2.0725e-09

4.7.2 GCR(m) 方法

注意到算法 4.17 (GCR 方法) 的执行需要存储所有的方向向量 $\boldsymbol{p}_k$. 如果在 GCR 方法中周期地重开始: 每 $m+1$ 次迭代, 用当前的第 i 次重开始的结果 $\boldsymbol{x}_{i(m+1)}$ 作为新的 $i+1$ 次重开始的初始向量, 这时只要存储 m 个方向向量. 这个重开始 CGR 方法称为 GCR(m). 下面给出重开始 GCR 方法的详细算法步骤.

算法 4.18 (GCR(m) 方法) 给定方程组 (4.226) 和容许误差限 $\varepsilon>0$, 取初始向量 $\boldsymbol{x}_0$ 及重开始数 m. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2\leqslant\varepsilon$, 其中 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{x}_1=\boldsymbol{x}_0$; $\boldsymbol{r}_1=\boldsymbol{r}_0$; $\boldsymbol{p}_1=\boldsymbol{r}_0$; $k=1$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|>\varepsilon)$

for $i=1:m$

$$\alpha_i=\frac{(\boldsymbol{r}_i,\boldsymbol{A}\boldsymbol{p}_i)}{(\boldsymbol{A}\boldsymbol{p}_i,\boldsymbol{A}\boldsymbol{p}_i)};$$

$$\boldsymbol{x}_{i+1}=\boldsymbol{x}_i+\alpha_i\boldsymbol{p}_i;\ \boldsymbol{r}_{i+1}=\boldsymbol{r}_i-\alpha_i\boldsymbol{A}\boldsymbol{p}_i;$$

for $s=1:i$

$$\beta_s^{(i)}=-\frac{(\boldsymbol{A}\boldsymbol{r}_{i+1},\boldsymbol{A}\boldsymbol{p}_s)}{(\boldsymbol{A}\boldsymbol{p}_s,\boldsymbol{A}\boldsymbol{p}_s)};$$

end

$$\boldsymbol{p}_{i+1}=\boldsymbol{r}_{i+1}+\sum_{s=0}^{i}\beta_s^{(i)}\boldsymbol{p}_s;$$

$$\boldsymbol{A}\boldsymbol{p}_{i+1}=\boldsymbol{A}\boldsymbol{r}_{i+1}+\sum_{s=0}^{i}\beta_i^{(i)}\boldsymbol{A}\boldsymbol{p}_s;$$

end

$\boldsymbol{x}_1=\boldsymbol{x}_{m+1}$; $\boldsymbol{r}_1=\boldsymbol{r}_{m+1}$; $\boldsymbol{p}_1=\boldsymbol{p}_{m+1}$; $\boldsymbol{A}\boldsymbol{p}_1=\boldsymbol{A}\boldsymbol{p}_{m+1}$;

$k=k+1$;

end

GCR(m) 方法的 MATLAB 程序如下:

```
%GCR(m)方法程序-gcrm.m
function [x,out,int,time,res,resvec]=gcrm(A,b,x,restrt,max_it,tol)
tic; n=length(b); m=restrt;
r=b-A*x; p=r; mr=norm(r); k=0;
AP=zeros(n,m+1); P=zeros(n,m+1);
```

```
AP(:,1)=A*p; P(:,1)=p; resvec(1)=1;
while (k<max_it)
    k=k+1;
    for i=1:m
        alpha=(r'*AP(:,i))/(AP(:,i)'*AP(:,i));
        x=x+alpha*P(:,i);
        r=r-alpha*AP(:,i);
        res=norm(r)/mr;
        resvec((k-1)*m+1+i)=res;
        if (res<tol), break;  end
        Ar=A*r; s1=zeros(n,1); s2=zeros(n,1);
        for (s=1:i)
            b(s)=-(Ar'*AP(:,s))/(AP(:,s)'*AP(:,s));
            s1=s1+b(s)*P(:,s);
            s2=s2+b(s)*AP(:,s);
        end
        P(:,i+1)=r+s1;
        AP(:,i+1)=Ar+s2;
    end
    if (res<tol),
         out=k; int=i;  break;
    end
    AP(:,1)=AP(:,m+1); P(:,1)=P(:,m+1);
end
time=toc;
```

例 4.14 仍考虑例 4.13 中的 $\boldsymbol{A}$ 和 $\boldsymbol{b}$. 将重开始 GCR 方法应用到该线性方程组上, 取重开始数 $m=6$, 在外迭代第 4 步内迭代第 2 步后收敛 ($\varepsilon=10^{-10}$), 计算得到的近似解 $\hat{\boldsymbol{x}}$ 和真解 $\boldsymbol{x}^*$ 之间的绝对值误差为

$$\|\hat{\boldsymbol{x}}-\boldsymbol{x}^*\|_2=2.0725\times10^{-9},$$

但计算解 $\hat{\boldsymbol{x}}$ 的残量满足

$$\|\boldsymbol{b}-\boldsymbol{A}\hat{\boldsymbol{x}}\|_2=2.7746\times10^{-8}.$$

迭代过程的收敛轨迹如图 4.15 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量.

若按公式

$$\text{总迭代次数}=(\text{外迭代次数}-1)\times\text{重开始数}+\text{内迭代数}$$

计算, 本例的总迭代次数为 20, 这跟例 4.13 中 *GCR* 方法一致.

此外, 还比较了 GCR(m) 与 GMRES(m) 算法. 从图 4.15 可以看出, 对于此例, GCR(m) 算法跟 GMRES(m) 算法的数值效果相仿.

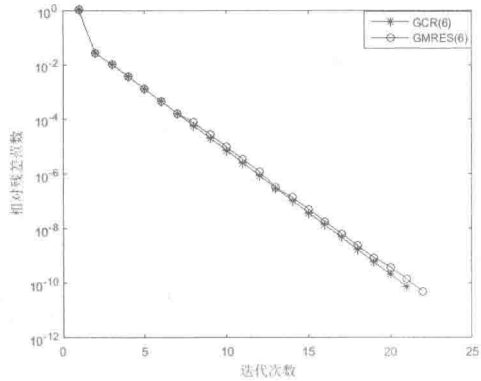

图 4.15 GCR(m) 方法的收敛特性

关于 GCR(m) 方法的收敛性, 有下面的结论.

定理 4.15 在定理 4.14 的假设下, 若 $\{\boldsymbol{r}_k\}$ 是 GCR(m) 的残量序列, 则

$$\|\boldsymbol{r}_{i(m+1)}\|_2 \leqslant \left[\min_{q_{m+1}\in\mathcal{P}_{m+1}} \|q_{m+1}(\boldsymbol{A})\|_2\right]^i \|\boldsymbol{r}_0\|_2, \tag{4.231}$$

由此, 得

$$\|\boldsymbol{r}_k\|_2 \leqslant \left[1-\frac{\lambda_{\min}^2(\boldsymbol{H})}{\lambda_{\max}(\boldsymbol{A}^\top\boldsymbol{A})}\right]^{k/2} \|\boldsymbol{r}_0\|_2. \tag{4.232}$$

因此, GCR(m) 方法收敛.

证明 由定理 4.14, 有

$$\begin{aligned}\|\boldsymbol{r}_{i(m+1)}\|_2 &\leqslant \left[\min_{q_{m+1}\in\mathcal{P}_{m+1}} \|q_{m+1}(\boldsymbol{A})\|_2\right] \|\boldsymbol{r}_{(i-1)(m+1)}\|_2\\ &\leqslant \cdots \leqslant \left[\min_{q_{m+1}\in\mathcal{P}_{m+1}} \|q_{m+1}(\boldsymbol{A})\|_2\right]^i \|\boldsymbol{r}_0\|_2.\end{aligned}$$

令 $k=i(m+1)+s$, $0\leqslant s\leqslant m$, 再由定理 4.14, 有

$$\|\boldsymbol{r}_k\|_2 \equiv \|\boldsymbol{r}_{i(m+1)+s}\|_2 \leqslant \left[1-\frac{\lambda_{\min}^2(\boldsymbol{H})}{\lambda_{\max}(\boldsymbol{A}^\top\boldsymbol{A})}\right]^{s/2} \|\boldsymbol{r}_{i(m+1)}\|_2. \tag{4.233}$$

由式 (4.231) 和定理 4.14, 得

$$\|\boldsymbol{r}_{i(m+1)}\|_2 \leqslant \left[1-\frac{\lambda_{\min}(\boldsymbol{H})^2}{\lambda_{\max}(\boldsymbol{A}^\top\boldsymbol{A})}\right]^{i(m+1)/2} \|\boldsymbol{r}_0\|_2.$$

将上式代入式 (4.233) 即得式 (4.232). 证毕. □

4.8 投影类方法

前几节所介绍的方法都是残量极小化类型的方法, 本节介绍第二类方法, 即残量正交化方法. 考虑如下的线性方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}, \tag{4.234}$$

式中: 矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 和向量 $\boldsymbol{b}\in\mathbb{R}^n$ 是已经给定的, 而 $\boldsymbol{x}\in\mathbb{R}^n$ 是待求的未知向量. 残量正交化方法是求 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 使得

$$\boldsymbol{r}_k\perp\mathcal{X}_k, \tag{4.235}$$

其中 $\boldsymbol{r}_k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$, 而 $\mathcal{X}_k\subset\mathbb{R}^n$ 是一个适当选择的 k 维子空间. 通常称式 (4.235) 为 Galerkin 条件.

设

$$\mathcal{X}_k=\operatorname{span}\{\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_k\},$$
$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\operatorname{span}\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k\},$$

并且记

$$\boldsymbol{V}_k=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_k],\quad \boldsymbol{W}_k=[\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_k],$$

则容易导出, 求 $\boldsymbol{x}_k\in\boldsymbol{x}_0+\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 满足式 (4.235) 就等价于求解线性方程组

$$\boldsymbol{H}_k\boldsymbol{z}=\boldsymbol{f}_k, \tag{4.236}$$

式中:

$$\boldsymbol{H}_k=\boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k,\quad \boldsymbol{f}_k=\boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{r}_0. \tag{4.237}$$

从几何上来看, 当 $\boldsymbol{W}_k^{\mathrm{T}}\boldsymbol{V}_k=\boldsymbol{I}_k$ 时, $\boldsymbol{H}_k$ 正好是 $\boldsymbol{A}$ 沿 $\mathcal{X}_k^{\perp}$ 到 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 上的投影 (见注 4.6), 故称式 (4.236) 为*投影方程*, 而称这类方法为*投影类方法*.

通常 $k\ll n$, 故式 (4.236) 有多种方法来求解. 这样, 实现这一方法的关键是如何选择子空间 $\mathcal{X}_k$ 以及如何有效地计算 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 和 $\mathcal{X}_k$ 的基向量.

本节取 $\mathcal{X}_k=\mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}},\widetilde{\boldsymbol{r}}_0)$, $\widetilde{\boldsymbol{r}}_0^{\mathrm{T}}\boldsymbol{r}_0\neq 0$, 并给出这方面的三类典型的方法: 双共轭梯度法 (BCG 方法)、共轭梯度平方法 (CGS 方法) 和稳定化共轭梯度法 (BCGSTAB 方法). BCG 方法是基础, 另外两个方法是通过对 BCG 方法的改进而得到的.

4.8.1 BCG 方法

BCG 方法 (双共轭梯度法, Bi-Conjugate Gradient) 是在式 (4.235) 中取子空间 $\mathcal{X}_k=\mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}},\widetilde{\boldsymbol{r}}_0)$ 而得到的一类投影方法. 当然, 可以利用非对称 Lanczos 方法来计算 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$ 和 $\mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}},\widetilde{\boldsymbol{r}}_0)$ 的基, 然后以类似于 CG 方法的推导过程导出 BCG 的基本迭代格式. 为了更清楚地展示这类方法与共轭梯度法的联系, 这里采用 CG 方法的一种等价描述来导出 BCG 的迭代格式.

1. CG 方法的等价描述

为了便于参照, 先将求对称正定线性方程组的 CG 方法的迭代格式重新叙述如下:

(1) $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{p}_{-1}=\boldsymbol{0}$; $\rho_{-1}=1$.

(2) $\rho_k=\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k$; $\beta_k=\rho_k/\rho_{k-1}$; $\boldsymbol{p}_k=\boldsymbol{r}_k+\beta_k\boldsymbol{p}_{k-1}$;

$\sigma_k=\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_k$; $\alpha_k=\rho_k/\sigma_k$; $\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\alpha_k\boldsymbol{p}_k$;

$\boldsymbol{r}_{k+1}=\boldsymbol{r}_k-\alpha_k\boldsymbol{A}\boldsymbol{p}_k$; $k=0,1,2,\cdots$.

若迭代进行到了 k 步, 则将产生三组向量:

① 近似解向量组 $\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_k$;

② 残量组 $\boldsymbol{r}_0,\boldsymbol{r}_1,\cdots,\boldsymbol{r}_k$;

③ 方向向量组 $\boldsymbol{p}_0,\boldsymbol{p}_1,\cdots,\boldsymbol{p}_k$.

它们具有如下的基本性质:

① $\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_j=\rho_i\delta_{ij}$ (残量相互正交);

② $\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}_j=\sigma_i\delta_{ij}$ ($\boldsymbol{A}$–共轭正交);

③ $\mathcal{K}_{k+1}(\boldsymbol{A},\boldsymbol{r}_0)=\mathrm{span}\{\boldsymbol{r}_0,\boldsymbol{r}_1,\cdots,\boldsymbol{r}_k\}=\mathrm{span}\{\boldsymbol{p}_0,\boldsymbol{p}_1,\cdots,\boldsymbol{p}_k\}$.

由性质 ③ 可知, 必存在 $\varphi_k,\psi_k\in\mathcal{P}_k$, 使得

$$\boldsymbol{r}_k=\varphi_k(\boldsymbol{A})\boldsymbol{r}_0,\quad \boldsymbol{p}_k=\psi_k(\boldsymbol{A})\boldsymbol{r}_0.$$

若记 $\varphi_0(t)=1$, $\psi_{-1}(t)=0$, $\vartheta(t)=t$, 则由前面所述的 CG 方法的迭代格式可导出

$$\begin{cases}\varphi_{k+1}=\varphi_k-\alpha_k\vartheta\psi_k,\\ \psi_k=\varphi_k+\beta_k\psi_{k-1},\quad k=0,1,2,\cdots.\end{cases}\tag{4.238}$$

现在在线性空间 $\mathcal{P}_n$ 上定义双线性泛函

$$[\varphi,\psi]=\boldsymbol{r}_0^{\mathrm{T}}\varphi(\boldsymbol{A})\psi(\boldsymbol{A})\boldsymbol{r}_0,\quad\forall\,\varphi,\psi\in\mathcal{P}_n.\tag{4.239}$$

注意, 这样定义的二元泛函除不能由 $[\varphi,\varphi]=0$ 导出 $\varphi=0$ 之外, 它具有内积的其他所有性质.

利用式 (4.239) 容易导出 CG 方法迭代格式中的几个参数的表达式, 即

$$\rho_k=[\varphi_k,\varphi_k],\quad \sigma_k=[\psi_k,\vartheta\psi_k].\tag{4.240}$$

将式 (4.238) 和式 (4.240) 相结合就得到了 CG 方法用多项式语言给出的等价描述:

(1) $\varphi_0=1$; $\psi_{-1}=0$; $\rho_{-1}=1$; (4.241a)

(2) $\rho_k=[\varphi_k,\varphi_k]$; $\beta_k=\rho_k/\rho_{k-1}$; (4.241b)

$\psi_k=\varphi_k+\beta_k\psi_{k-1}$; (4.241c)

$\sigma_k=[\psi_k,\vartheta\psi_k]$; $\alpha_k=\rho_k/\sigma_k$; (4.241d)

$\varphi_{k+1}=\varphi_k-\alpha_k\vartheta\psi_k$; $k=0,1,2,\cdots$. (4.241e)

显然, 由上面的迭代格式产生的多项式系列满足

$$[\varphi_i,\varphi_j]=\rho_i\delta_{ij},\quad [\psi_i,\vartheta\psi_j]=\sigma_i\delta_{ij}.\tag{4.242}$$

2. BCG 方法

类比于式 (4.239), 对给定的非奇异矩阵 $\boldsymbol{A}$ 以及向量 $\boldsymbol{r}_0$ 和 $\widetilde{\boldsymbol{r}}_0$, 在 $\mathcal{P}_n$ 上定义双线性泛函

$$\langle \varphi, \psi \rangle = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \varphi(\boldsymbol{A}) \psi(\boldsymbol{A}) \boldsymbol{r}_0, \quad \forall\, \varphi, \psi \in \mathcal{P}_n, \tag{4.243}$$

然后将式 (4.239) 中定义的双线性泛函 $[\cdot,\cdot]$ 换为现在定义的 $\langle \cdot,\cdot \rangle$, 再逐字照搬过来, 便有

$$\text{(1)}\ \ \varphi_0 = 1;\ \psi_{-1} = 0;\ \rho_{-1} = 1; \tag{4.244a}$$

$$\text{(2)}\ \ \rho_k = \langle \varphi_k, \varphi_k \rangle;\ \beta_k = \rho_k / \rho_{k-1}; \tag{4.244b}$$

$$\psi_k = \varphi_k + \beta_k \psi_{k-1}; \tag{4.244c}$$

$$\sigma_k = \langle \psi_k; \vartheta \psi_k \rangle;\ \alpha_k = \rho_k / \sigma_k; \tag{4.244d}$$

$$\varphi_{k+1} = \varphi_k - \alpha_k \vartheta \psi_k;\ k = 0, 1, 2, \cdots. \tag{4.244e}$$

当然, 也可用数学归纳法证明这样迭代产生的 $\{\varphi_i\}_{i=0}^{k}$ 和 $\{\psi_i\}_{i=0}^{k}$ 具有性质

$$\langle \varphi_i, \varphi_j \rangle = \rho_i \delta_{ij}, \quad \langle \psi_i, \vartheta \psi_j \rangle = \sigma_i \delta_{ij}. \tag{4.245}$$

现在定义

$$\boldsymbol{r}_k = \varphi_k(\boldsymbol{A}) \boldsymbol{r}_0, \quad \widetilde{\boldsymbol{r}}_k = \varphi_k(\boldsymbol{A}^{\mathrm{T}}) \widetilde{\boldsymbol{r}}_0, \tag{4.246}$$

$$\boldsymbol{q}_k = \psi_k(\boldsymbol{A}) \boldsymbol{r}_0, \quad \widetilde{\boldsymbol{q}}_k = \psi_k(\boldsymbol{A}^{\mathrm{T}}) \widetilde{\boldsymbol{r}}_0. \tag{4.247}$$

将式 (4.244) 改成向量形式, 便有

$$\text{(1)}\ \ \boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0;\ \widetilde{\boldsymbol{r}}_0 \text{ 满足 } \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_0 \neq 0;$$

$$\boldsymbol{q}_{-1} = \widetilde{\boldsymbol{q}}_{-1} = \boldsymbol{0};\ \rho_{-1} = 1;$$

$$\text{(2)}\ \ \rho_k = \widetilde{\boldsymbol{r}}_k^{\mathrm{T}} \boldsymbol{r}_k;\ \beta_k = \rho_k / \rho_{k-1}; \tag{4.248a}$$

$$\boldsymbol{q}_k = \boldsymbol{r}_k + \beta_k \boldsymbol{q}_{k-1}; \tag{4.248b}$$

$$\widetilde{\boldsymbol{q}}_k = \widetilde{\boldsymbol{r}}_k + \beta_k \widetilde{\boldsymbol{q}}_{k-1}; \tag{4.248c}$$

$$\sigma_k = \widetilde{\boldsymbol{q}}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{q}_k;\ \alpha_k = \rho_k / \sigma_k; \tag{4.248d}$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A} \boldsymbol{q}_k; \tag{4.248e}$$

$$\widetilde{\boldsymbol{r}}_{k+1} = \widetilde{\boldsymbol{r}}_k - \alpha_k \boldsymbol{A}^{\mathrm{T}} \widetilde{\boldsymbol{q}}_k;\ k = 0, 1, 2, \cdots. \tag{4.248f}$$

利用数学归纳法可证如下的结果.

定理 4.16 由式 (4.248) 产生的序列 $\{\boldsymbol{q}_k\}$, $\{\widetilde{\boldsymbol{q}}_k\}$, $\{\boldsymbol{r}_k\}$ 和 $\{\widetilde{\boldsymbol{r}}_k\}$ 具有如下性质:

$$\text{(1)}\ \ \widetilde{\boldsymbol{r}}_k^{\mathrm{T}} \boldsymbol{r}_l = \boldsymbol{r}_k^{\mathrm{T}} \widetilde{\boldsymbol{r}}_l = \rho_k \delta_{kl}. \tag{4.249a}$$

$$\text{(2)}\ \ \widetilde{\boldsymbol{q}}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{q}_l = \boldsymbol{q}_k^{\mathrm{T}} \boldsymbol{A} \widetilde{\boldsymbol{q}}_l = \sigma_k \delta_{kl}. \tag{4.249b}$$

$$\text{(3)}\ \ \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0) = \mathrm{span}\{\boldsymbol{q}_0, \boldsymbol{q}_1, \cdots, \boldsymbol{q}_{k-1}\} = \mathrm{span}\{\boldsymbol{r}_0, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_{k-1}\}. \tag{4.249c}$$

$$\text{(4)}\ \ \mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}}, \widetilde{\boldsymbol{r}}_0) = \mathrm{span}\{\widetilde{\boldsymbol{q}}_0, \widetilde{\boldsymbol{q}}_1, \cdots, \widetilde{\boldsymbol{q}}_{k-1}\} = \mathrm{span}\{\widetilde{\boldsymbol{r}}_0, \widetilde{\boldsymbol{r}}_1, \cdots, \widetilde{\boldsymbol{r}}_{k-1}\}. \tag{4.249d}$$

证明 首先对 k 用归纳法证明式 (4.249a) 和式 (4.249b). 显然, 当 $k=0$ 时, 结论是平凡的. 故设 $k+1\geqslant 1$. 假设结论对 k 成立. 由式 (4.248c) 和式 (4.248e), 得

$$\begin{aligned}\boldsymbol{r}_{k+1}^{\mathrm{T}}\widetilde{\boldsymbol{r}}_l &= \boldsymbol{r}_k^{\mathrm{T}}\widetilde{\boldsymbol{r}}_l - \alpha_k \boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{r}}_l \\ &= \boldsymbol{r}_k^{\mathrm{T}}\widetilde{\boldsymbol{r}}_l - \alpha_i \boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}(\widetilde{\boldsymbol{q}}_l - \beta_l\widetilde{\boldsymbol{q}}_{l-1}).\end{aligned} \tag{4.250}$$

在式 (4.250) 中, 当 $l=k$ 时, 由 α_k 的表达式和归纳假设, 得

$$\begin{aligned}\boldsymbol{r}_{k+1}^{\mathrm{T}}\widetilde{\boldsymbol{r}}_k &= \boldsymbol{r}_k^{\mathrm{T}}\widetilde{\boldsymbol{r}}_k - \frac{\widetilde{\boldsymbol{r}}_k^{\mathrm{T}}\boldsymbol{r}_k}{\widetilde{\boldsymbol{q}}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_k}(\boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{q}}_k - \beta_k\boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{q}}_{k-1}) \\ &= \boldsymbol{r}_k^{\mathrm{T}}\widetilde{\boldsymbol{r}}_k - \frac{\boldsymbol{r}_k^{\mathrm{T}}\widetilde{\boldsymbol{r}}_k}{\widetilde{\boldsymbol{q}}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_k}(\widetilde{\boldsymbol{q}}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_k - \beta_k\widetilde{\boldsymbol{q}}_{k-1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_k) = 0.\end{aligned}$$

当 $l<k$ 时, 由归纳法假设可得 $\boldsymbol{r}_{k+1}^{\mathrm{T}}\widetilde{\boldsymbol{r}}_l = 0$. 同理, 可证 $\widetilde{\boldsymbol{r}}_{k+1}^{\mathrm{T}}\boldsymbol{r}_l = 0\,(l\leqslant k)$. 这就证明了式 (4.249a).

由式 (4.248b) 和式 (4.248f), 得

$$\begin{aligned}\boldsymbol{q}_{k+1}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{q}}_l &= \boldsymbol{r}_{k+1}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{q}}_l + \beta_{k+1}\boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{q}}_l \\ &= \boldsymbol{r}_{k+1}^{\mathrm{T}}\Big(\frac{\widetilde{\boldsymbol{r}}_l - \widetilde{\boldsymbol{r}}_{l+1}}{\alpha_k}\Big) + \beta_{k+1}\boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{q}}_l.\end{aligned} \tag{4.251}$$

在式 (4.251) 中, 当 $l=k$ 时, 由 α_k 和 β_{k+1} 的表达式及归纳假设, 得

$$\begin{aligned}\boldsymbol{q}_{k+1}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{q}}_k &= \boldsymbol{r}_{k+1}^{\mathrm{T}}\Big(\frac{\widetilde{\boldsymbol{r}}_k - \widetilde{\boldsymbol{r}}_{k+1}}{\alpha_k}\Big) + \beta_{k+1}\frac{\boldsymbol{r}_k^{\mathrm{T}}\widetilde{\boldsymbol{r}}_k}{\alpha_k} \\ &= \boldsymbol{r}_{k+1}^{\mathrm{T}}\Big(\frac{\widetilde{\boldsymbol{r}}_k - \widetilde{\boldsymbol{r}}_{k+1}}{\alpha_k}\Big) + \frac{\boldsymbol{r}_{k+1}^{\mathrm{T}}\widetilde{\boldsymbol{r}}_{k+1}}{\boldsymbol{r}_k^{\mathrm{T}}\widetilde{\boldsymbol{r}}_k}\frac{\boldsymbol{r}_k^{\mathrm{T}}\widetilde{\boldsymbol{r}}_k}{\alpha_k} \\ &= \frac{\boldsymbol{r}_{k+1}^{\mathrm{T}}\widetilde{\boldsymbol{r}}_k}{\alpha_k} = 0.\end{aligned}$$

当 $l<k$ 时, 利用式 (4.249a) 和归纳法假设可得 $\boldsymbol{q}_{k+1}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{q}}_l = 0$. 同理, 可证 $\widetilde{\boldsymbol{q}}_{k+1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_l = 0\,(l\leqslant k)$. 这就证明了式 (4.249b).

由于

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0) = \mathrm{span}\{\boldsymbol{r}_0,\boldsymbol{r}_1,\cdots,\boldsymbol{r}_{k-1}\},\quad \mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}},\widetilde{\boldsymbol{r}}_0) = \mathrm{span}\{\widetilde{\boldsymbol{r}}_0,\widetilde{\boldsymbol{r}}_1,\cdots,\widetilde{\boldsymbol{r}}_{k-1}\},$$

且向量组 $\{\boldsymbol{q}_0,\boldsymbol{q}_1,\cdots,\boldsymbol{q}_{k-1}\}$ 与 $\{\boldsymbol{r}_0,\boldsymbol{r}_1,\cdots,\boldsymbol{r}_{k-1}\}$ 及向量组 $\{\widetilde{\boldsymbol{q}}_0,\widetilde{\boldsymbol{q}}_1,\cdots,\widetilde{\boldsymbol{q}}_{k-1}\}$ 与 $\{\widetilde{\boldsymbol{r}}_0,\widetilde{\boldsymbol{r}}_1,\cdots,\widetilde{\boldsymbol{r}}_{k-1}\}$ 可以相互线性表出, 故容易得到式 (4.249c) 和式 (4.249d). 证毕. □

此外, 若定义

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{q}_k,\quad k=0,1,2,\cdots,$$

则迭代式 (4.248e) 中的向量 $\boldsymbol{r}_k$ 刚好为 $\boldsymbol{r}_k = \boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_k$, 且由定理 4.16 的 (1) 和 (4) 可知

$$\boldsymbol{r}_k \perp \mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}},\widetilde{\boldsymbol{r}}_0),$$

这表明 $\boldsymbol{x}_k$ 满足残量正交化条件式 (4.235). 综上所述, 就得到了如下算法.

算法 4.19 (BCG 方法) 给定线性方程组 (4.234), 初始向量 $\boldsymbol{x}_0$ 和 $\varepsilon > 0$. 本算法计算向量 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{q}_{-1} = \widetilde{\boldsymbol{q}}_{-1} = \boldsymbol{0}$; $\rho_{-1} = 1$;

选择 $\widetilde{\boldsymbol{r}}_0$ 满足 $\widetilde{\boldsymbol{r}}_0^{\mathrm{T}}\boldsymbol{r}_0 \neq 0$; $k = 0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 > \varepsilon)$

$\rho_k = \widetilde{\boldsymbol{r}}_k^{\mathrm{T}}\boldsymbol{r}_k$; $\beta_k = \rho_k/\rho_{k-1}$;

$\boldsymbol{q}_k = \boldsymbol{r}_k + \beta_k\boldsymbol{q}_{k-1}$; $\widetilde{\boldsymbol{q}}_k = \widetilde{\boldsymbol{r}}_k + \beta_k\widetilde{\boldsymbol{q}}_{k-1}$;

$\sigma_k = \widetilde{\boldsymbol{q}}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_k$; $\alpha_k = \rho_k/\sigma_k$;

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{q}_k$; $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k\boldsymbol{A}\boldsymbol{q}_k$;

$\widetilde{\boldsymbol{r}}_{k+1} = \widetilde{\boldsymbol{r}}_k - \alpha_k\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{q}}_k$; $k = k+1$;

end

注 4.9 若计算过程中出现 $\rho_{k-1} = 0$ 或 $\sigma_k = 0$, 但还没有满足收敛性条件, 则此时算法就发生中断. 此外, 如果希望同时求解对偶方程 $\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}$, 则只需在上述算法中增加两句 $\widetilde{\boldsymbol{r}}_0 = \widetilde{\boldsymbol{b}} - \boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{x}}_0$ 和 $\widetilde{\boldsymbol{x}}_{k+1} = \widetilde{\boldsymbol{x}}_k + \alpha_k\widetilde{\boldsymbol{q}}_k$ 即可.

BCG 方法的 MATLAB 程序如下:

```
%BCG方法程序-bcg.m
function [x,k,time,res,resvec]=bcg(A,b,x,max_it,tol)
tic;n=length(b);
q=zeros(n,1);qt=q;rho=1;
r=b-A*x;mr=norm(r);k=0;
rt=ones(n,1);%rt的选取会影响收敛速度
%rt=r/mr; %rt=zeros(n,1);rt(1)=1;
while (k<max_it)
    k=k+1;
    res=norm(r)/mr;resvec(k)=res;
    if (res<tol), break; end
    rho1=rt'*r;beta=rho1/rho;
    q=r+beta*q;qt=rt+beta*qt;
    Aq=A*q; sigma=qt'*Aq;
    alpha=rho1/sigma;
    x=x+alpha*q;
    r=r-alpha*Aq;
    rt=rt-alpha*(A'*qt);
    rho=rho1;
end
time=toc;
```

例 4.15 假设线性方程组的系数矩阵 $\boldsymbol{A}$ 由 MATLAB 命令

$$gallery('lotkin', n), \quad n = 1000,$$

产生, 这是一个 Hilbert 矩阵 (即其 (i,j) 元素为 $1/(i+j-1)$), 但其第 1 行的元素都换成了 1. 这个矩阵非常病态, 其条件数为 6.65×10^{21}. 再假设真解 $\boldsymbol{x}^*$ 是分量都为 1 的向量, 从而该方程组的右端项为 $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^*$. 将 BCG 算法应用到该线性方程组上, 取算法中的 $\widetilde{\boldsymbol{r}}_0 = ones(n,1)$. 迭代在 60 步后收敛 ($\varepsilon = 10^{-10}$), 计算得到的近似解 $\widehat{\boldsymbol{x}}$ 和真解 $\boldsymbol{x}^*$ 之间的绝对误差为

$$\|\widehat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 0.6727,$$

但计算解 $\widehat{\boldsymbol{x}}$ 的残量满足

$$\|\boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}}\|_2 = 2.6526 \times 10^{-8}.$$

迭代过程的收敛轨迹如图 4.16 所示, 其中横坐标为迭代步数 k, 纵坐标为 $\lg \|\boldsymbol{r}_k\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残量. 请注意观察, 与前面几个方法相比, 此时 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$ 不再单调下降, 而是在振荡中下降着.

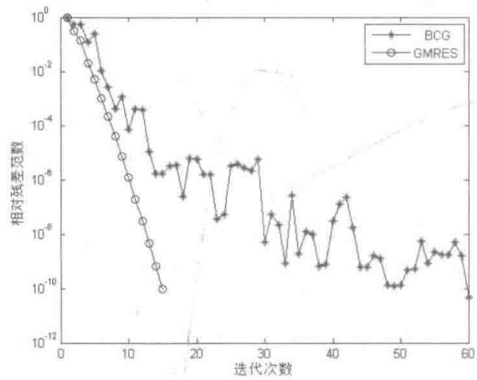

图 4.16 BCG 算法的收敛特性

4.8.2 CGS 方法

CGS 方法 (共轭梯度平方法, Conjugate Gradient Squared) 是为了避免 BCG 方法中 $\boldsymbol{A}^{\mathrm{T}}$ 的出现而提出的.

仔细观察 BCG 的迭代格式就会发现, $\boldsymbol{A}^{\mathrm{T}}$ 只在计算 $\widetilde{\boldsymbol{r}}_k$ 时用到, 而这一量又在计算 ρ_k 和 σ_k 时用到. 再利用导出 BCG 迭代格式的多项式形式的迭代格式 (4.244) 立即可以看出, 这两个量的计算完全可以避开 $\boldsymbol{A}^{\mathrm{T}}$ 的使用. 事实上, 有

$$\begin{cases} \rho_k = \langle \varphi_k, \varphi_k \rangle = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \varphi_k(\boldsymbol{A}) \varphi_k(\boldsymbol{A}) \boldsymbol{r}_0 = \langle \varphi_0, \varphi_k^2 \rangle, \\ \sigma = \langle \psi_k, \vartheta \psi_k \rangle = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \psi_k(\boldsymbol{A}) \boldsymbol{A} \psi_k(\boldsymbol{A}) \boldsymbol{r}_0 = \langle \psi_0, \vartheta \psi_k^2 \rangle. \end{cases} \tag{4.252}$$

因此, 若定义

$$\boldsymbol{r}_k = \varphi_k^2(\boldsymbol{A}) \boldsymbol{r}_0, \quad \boldsymbol{q}_k = \psi_k^2(\boldsymbol{A}) \boldsymbol{r}_0, \tag{4.253}$$

则有

$$\rho_k = \tilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_k, \quad \sigma_k = \tilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{A}\boldsymbol{q}_k. \tag{4.254}$$

这样一来, 要用式 (4.253) 来高效地计算 $\boldsymbol{r}_k$ 和 $\boldsymbol{q}_k$, 关键是如何高效地计算 φ_k^2 和 ψ_k^2. 由式 (4.244) 立即可导出

$$\psi_k^2 = (\varphi_k + \beta_k \psi_{k-1})^2 = \varphi_k^2 + 2\beta_k \varphi_k \psi_{k-1} + \beta_k^2 \psi_{k-1}^2, \tag{4.255}$$

$$\varphi_{k+1}^2 = (\varphi_k - \alpha_k \vartheta \psi_k)^2 = \varphi_k^2 - 2\alpha_k \vartheta \varphi_k \psi_k + \alpha_k^2 \vartheta^2 \psi_k^2, \tag{4.256}$$

其中又涉及 $\varphi_k \psi_{k-1}$ 和 $\varphi_k \psi_k$. 再利用式 (4.244), 有

$$\varphi_{k+1}\psi_k = \varphi_k \psi_k - \alpha_k \vartheta \psi_k^2, \tag{4.257}$$

$$\varphi_k \psi_k = \varphi_k^2 + \beta_k \varphi_k \psi_{k-1}. \tag{4.258}$$

再定义

$$\boldsymbol{p}_k = \varphi_k(\boldsymbol{A})\psi_{k-1}(\boldsymbol{A})\boldsymbol{r}_0, \quad \boldsymbol{u}_k = \varphi_k(\boldsymbol{A})\psi_k(\boldsymbol{A})\boldsymbol{r}_0, \tag{4.259}$$

由式 (4.255)~式 (4.258) 得到如下递推公式:

$$\begin{cases} \boldsymbol{q}_k = \boldsymbol{r}_k + 2\beta_k \boldsymbol{p}_k + \beta_k^2 \boldsymbol{q}_{k-1}, \\ \boldsymbol{r}_{k+1} = \boldsymbol{r}_k - 2\alpha_k \boldsymbol{A}\boldsymbol{u}_k + \alpha_k^2 \boldsymbol{A}^2 \boldsymbol{q}_k, \\ \boldsymbol{p}_{k+1} = \boldsymbol{u}_k - \alpha_k \boldsymbol{A}\boldsymbol{q}_k, \\ \boldsymbol{u}_k = \boldsymbol{r}_k + \beta_k \boldsymbol{p}_k. \end{cases} \tag{4.260}$$

由式 (4.260) 的第 1 式和第 4 式, 得

$$\boldsymbol{q}_k = \boldsymbol{u}_k + \beta_k(\boldsymbol{p}_k + \beta_k \boldsymbol{q}_{k-1}), \tag{4.261}$$

再由式 (4.260) 的第 2 式和第 3 式, 得

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}(\boldsymbol{u}_k + \boldsymbol{p}_{k+1}). \tag{4.262}$$

由式 (4.262) 可知, 若定义

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k(\boldsymbol{u}_k + \boldsymbol{p}_{k+1}), \tag{4.263}$$

则 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ 正好是对应近似解的残量.

综合上面的讨论, 便有如下的共轭梯度平方法.

算法 4.20 (CGS 方法) 给定线性方程组 (4.234), 初始向量 $\boldsymbol{x}_0$ 和容许误差 $\varepsilon > 0$. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{q}_{-1} = \boldsymbol{p}_0 = 0$; $\rho_{-1} = 1$;

选择 $\tilde{\boldsymbol{r}}_0$ 满足 $\tilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_0 \neq 0$; $k = 0$;

while $(\|\boldsymbol{r}_k\|_2/\|r_0\|_2 > \varepsilon)$

$\rho_k = \tilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_k$; $\beta_k = \rho_k/\rho_{k-1}$;

$\boldsymbol{u}_k = \boldsymbol{r}_k + \beta_k \boldsymbol{p}_k$;
$\boldsymbol{q}_k = \boldsymbol{u}_k + \beta_k(\boldsymbol{p}_k + \beta_k \boldsymbol{q}_{k-1})$;
$\boldsymbol{q}_k = \boldsymbol{A}\boldsymbol{q}_k$; $\sigma_k = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{q}_k$;
$\alpha_k = \rho_k / \sigma_k$; $\boldsymbol{p}_{k+1} = \boldsymbol{u}_k - \alpha_k \boldsymbol{q}_k$;
$\boldsymbol{z}_k = \alpha_k(\boldsymbol{u}_k + \boldsymbol{p}_{k+1})$;
$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{z}_k$; $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \boldsymbol{A}\boldsymbol{z}_k$;
$k = k + 1$;

end

算法 4.20 确实避免了 $\boldsymbol{A}^{\mathrm{T}}$ 的出现, 每次迭代只需作两次系数矩阵 $\boldsymbol{A}$ 与向量的乘积. 由前面的推导过程可知, 算法 4.20 产生的近似解的残量为 $\boldsymbol{r}_k^{\mathrm{CGS}} = \varphi_k^2(\boldsymbol{A})\boldsymbol{r}_0$, 而 BCG 算法产生的近似解的残量为 $\boldsymbol{r}_k^{\mathrm{BCG}} = \varphi_k(\boldsymbol{A})\boldsymbol{r}_0$. 当 $\boldsymbol{r}_k^{\mathrm{BCG}}$ 趋向于零时, $\varphi_k(\boldsymbol{A})$ 就是一个收缩因子. 从这个意义上来讲, $\boldsymbol{r}_k^{\mathrm{CGS}}$ 的收敛到零的速度应该是 $\boldsymbol{r}_k^{\mathrm{BCG}}$ 的两倍. 但这个算法的缺点是 $\|\boldsymbol{r}_k^{\mathrm{CGS}}\|_2$ 随着 k 的增加会发生激烈的抖动.

例 4.16 仍然考虑例 4.15 中的 $\boldsymbol{A}$ 和 $\boldsymbol{b}$. 将 CGS 算法应用到该线性方程组上, 取算法中的 $\widetilde{\boldsymbol{r}}_0 = (1, 0, \cdots, 0)^{\mathrm{T}}$, 迭代在 142 步后收敛 ($\varepsilon = 10^{-10}$), 计算得到的近似解 $\widehat{\boldsymbol{x}}$ 和真解 $\boldsymbol{x}^*$ 之间的误差为

$$\|\widehat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 0.8086,$$

但计算解 $\widehat{\boldsymbol{x}}$ 的残量满足

$$\|\boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}}\|_2 = 7.0413 \times 10^{-8}.$$

迭代过程的收敛轨迹如图 4.17 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2 / \|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量. 此时 $\|\boldsymbol{r}_k\|_2 / \|\boldsymbol{r}_0\|_2$ 与 BCG 方法一样也是在剧烈的振荡中艰难地下降着, 但注意到此例反映不出前面分析中 CGS 的收敛速度大约是 BCG 的两倍, 或许是受 $\widetilde{\boldsymbol{r}}_0$ 选取的影响.

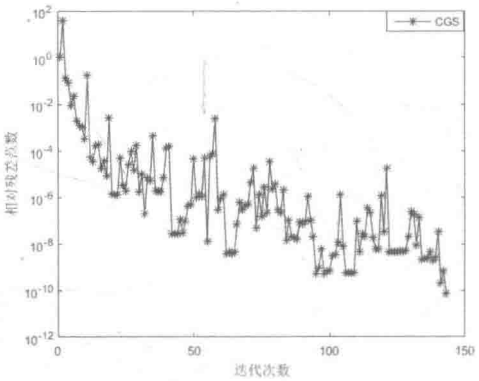

图 4.17 CGS 算法的收敛特性

4.8.3 BCGSTAB 方法

BCGSTAB 方法 (稳定化双共轭梯度法, Bi-Conjugate Gradient Stablized) 是为了改进 CGS 方法之残量的范数剧烈抖动而提出的. 这一方法的基本思想是 CGS 方法的残量 $\boldsymbol{r}_k^{\mathrm{CGS}}$ 满足

$$\boldsymbol{r}_k^{\mathrm{CGS}} = \varphi_k(\boldsymbol{A})\boldsymbol{r}_k^{\mathrm{BCG}} = \varphi_k^2(\boldsymbol{A})\boldsymbol{r}_0, \tag{4.264}$$

可以考虑不用多项式 φ_k, 而是选择一个其他的 k 次多项式 $\widetilde{\varphi}_k$, 使

$$\boldsymbol{r}_k = \widetilde{\varphi}_k(\boldsymbol{A})\boldsymbol{r}_k^{\mathrm{BCG}} = \widetilde{\varphi}_k(\boldsymbol{A})\varphi_k(\boldsymbol{A})\boldsymbol{r}_0, \tag{4.265}$$

以期待这样选取的相对残差范数 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$ 的振荡性有所改进.

一种选择 $\widetilde{\varphi}_k$ 的方法是根据递推公式

$$\widetilde{\varphi}_0(t) = 1, \quad \widetilde{\varphi}_{k+1}(t) = (1-\omega_{k+1}t)\widetilde{\varphi}_k(t), \tag{4.266}$$

式中: ω_{k+1} 为待定参数. BCGSTAB 方法正是利用这一参数的可选择性来改进其相对残差范数的振荡性的.

利用式 (4.244) 和式 (4.266), 可导出 $\widetilde{\varphi}_{k+1}\varphi_{k+1}$ 的递推计算公式

$$\begin{aligned}\widetilde{\varphi}_{k+1}\varphi_{k+1} &= (1-\omega_{k+1}\vartheta)\widetilde{\varphi}_k(\varphi_k - \alpha_k\vartheta\psi_k)\\ &= (1-\omega_{k+1}\vartheta)(\widetilde{\varphi}_k\varphi_k - \alpha_k\vartheta\widetilde{\varphi}_k\psi_k),\end{aligned} \tag{4.267}$$

$$\begin{aligned}\widetilde{\varphi}_k\psi_k &= \widetilde{\varphi}_k(\varphi_k + \beta_k\psi_{k-1})\\ &= \widetilde{\varphi}_k\varphi_k + \beta_k(1-\omega_k\vartheta)\widetilde{\varphi}_{k-1}\psi_{k-1}.\end{aligned} \tag{4.268}$$

现定义

$$\boldsymbol{r}_k = \widetilde{\varphi}_k(\boldsymbol{A})\varphi_k(\boldsymbol{A})\boldsymbol{r}_0, \quad \boldsymbol{p}_k = \widetilde{\varphi}_k(\boldsymbol{A})\psi_k(\boldsymbol{A})\boldsymbol{r}_0, \tag{4.269}$$

则由式 (4.267) 和式 (4.268), 得

$$\boldsymbol{r}_{k+1} = (\boldsymbol{I}-\omega_{k+1}\boldsymbol{A})(\boldsymbol{r}_k - \alpha_k\boldsymbol{A}\boldsymbol{p}_k), \tag{4.270}$$

$$\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_k(\boldsymbol{I}-\omega_k\boldsymbol{A})\boldsymbol{p}_{k-1}. \tag{4.271}$$

下面来考虑 α_k 和 β_k 的计算问题. 由式 (4.244) 可知

$$\alpha_k = \frac{\rho_k}{\sigma_k}, \quad \beta_k = \frac{\rho_k}{\rho_{k-1}}, \tag{4.272}$$

式中:

$$\rho_k = \langle\varphi_k, \varphi_k\rangle, \quad \sigma_k = \langle\psi_k, \vartheta\psi_k\rangle. \tag{4.273}$$

由定理 4.16 可知

$$\begin{aligned}&\varphi_k(\boldsymbol{A})\boldsymbol{r}_0 = \boldsymbol{r}_k^{\mathrm{BCG}} \perp \mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}}, \widetilde{\boldsymbol{r}}_0),\\ &\psi_k(\boldsymbol{A})\boldsymbol{r}_0 = \boldsymbol{q}_k^{\mathrm{BCG}} \perp \boldsymbol{A}^{\mathrm{T}}\mathcal{K}_k(\boldsymbol{A}^{\mathrm{T}}, \widetilde{\boldsymbol{r}}_0).\end{aligned}$$

于是对任意的 $\psi \in \mathcal{P}_{k-1}$, 有

$$\langle \varphi_k, \psi \rangle = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \psi(\boldsymbol{A}) \varphi_k(\boldsymbol{A}) \boldsymbol{r}_0 = \big(\psi(\boldsymbol{A}^{\mathrm{T}}) \widetilde{\boldsymbol{r}}_0\big)^{\mathrm{T}} \big(\varphi_k(\boldsymbol{A}) \boldsymbol{r}_0\big) = 0, \tag{4.274}$$

$$\langle \psi_k, \vartheta\psi \rangle = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \psi(\boldsymbol{A}) \boldsymbol{A} \psi_k(\boldsymbol{A}) \boldsymbol{r}_0 = \big(\boldsymbol{A}^{\mathrm{T}} \psi\big(\boldsymbol{A}^{\mathrm{T}}\big) \widetilde{\boldsymbol{r}}_0\big)^{\mathrm{T}} \big(\psi_k(\boldsymbol{A}) \boldsymbol{r}_0\big) = 0. \tag{4.275}$$

设 φ_k 和 $\widetilde{\varphi}_k$ 的首项系数分别为 ξ_k 和 η_k, 则由式 (4.244) 和式 (4.266) 可知, 它们可递推地计算

$$\xi_{k+1} = -\alpha_k \xi_k, \quad \eta_{k+1} = -\omega_{k+1} \eta_k, \tag{4.276}$$

其中 $\xi_0 = \eta_0 = 1$, 这里用到了

$$\psi_k = \varphi_k + \beta_k \psi_{k-1}$$

蕴涵着 ψ_k 和 φ_k 有相同的首项系数. 这样

$$\psi = \psi_k - \frac{\xi_k}{\eta_k} \widetilde{\varphi}_k \quad \text{和} \quad \varphi = \varphi_k - \frac{\xi_k}{\eta_k} \widetilde{\varphi}_k$$

均为 $k-1$ 次多项式, 从而利用式 (4.274) 和式 (4.275) 有

$$\begin{aligned} \rho_k &= \langle \varphi_k, \varphi_k \rangle = \langle \varphi_k, \varphi + \frac{\xi_k}{\eta_k} \widetilde{\varphi}_k \rangle = \langle \varphi_k, \frac{\xi_k}{\eta_k} \widetilde{\varphi}_k \rangle \\ &= \frac{\xi_k}{\eta_k} \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \widetilde{\varphi}_k(\boldsymbol{A}) \varphi_k(\boldsymbol{A}) \boldsymbol{r}_0 = \frac{\xi_k}{\eta_k} \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_k, \end{aligned} \tag{4.277}$$

$$\begin{aligned} \sigma_k &= \langle \psi_k, \vartheta\psi_k \rangle = \langle \psi + \frac{\xi_k}{\eta_k} \widetilde{\varphi}_k, \vartheta\psi_k \rangle = \langle \frac{\xi_k}{\eta_k} \widetilde{\varphi}_k, \vartheta\psi_k \rangle \\ &= \frac{\xi_k}{\eta_k} \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{A} \widetilde{\varphi}_k(\boldsymbol{A}) \psi_k(\boldsymbol{A}) \boldsymbol{r}_0 = \frac{\xi_k}{\eta_k} \widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{A} \boldsymbol{p}_k. \end{aligned} \tag{4.278}$$

因此,

$$\alpha_k = \frac{\rho_k}{\sigma_k} = \frac{\widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_k}{\widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{A} \boldsymbol{p}_k}, \tag{4.279}$$

$$\beta_k = \frac{\rho_k}{\rho_{k-1}} = \frac{\xi_k}{\xi_{k-1}} \frac{\eta_{k-1}}{\eta_k} \frac{\widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_k}{\widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_{k-1}} = \frac{\alpha_{k-1}}{\omega_k} \frac{\widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_k}{\widetilde{\boldsymbol{r}}_0^{\mathrm{T}} \boldsymbol{r}_{k-1}}, \tag{4.280}$$

其中最后一个等式用到了式 (4.276).

到此为止 ω_{k+1} 仍然是自由的, 下面确定 ω_{k+1}. 为了符号简单起见, 记

$$\boldsymbol{s}_k = \boldsymbol{r}_k - \alpha_k \boldsymbol{A} \boldsymbol{p}_k, \tag{4.281}$$

有

$$\boldsymbol{r}_{k+1} = (\boldsymbol{I} - \omega_{k+1} \boldsymbol{A}) \boldsymbol{s}_k. \tag{4.282}$$

自然选择 ω_{k+1} 使

$$\|\boldsymbol{r}_{k+1}\|_2 = \min_{\omega} \|(\boldsymbol{I} - \omega \boldsymbol{A}) \boldsymbol{s}_k\|_2.$$

解此最小二乘问题, 得

$$\omega_{k+1} = \frac{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{s}_k}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{s}_k}. \tag{4.283}$$

正是 ω_{k+1} 的这种极小化取法 $\|\boldsymbol{r}_{k+1}\|_2$ 而使得的振荡性有所改善.

注意: 式 (4.282) 又可写为

$$\boldsymbol{r}_{k+1} = \boldsymbol{s}_k - \omega_{k+1}\boldsymbol{A}\boldsymbol{s}_k = \boldsymbol{r}_k - \alpha_k\boldsymbol{A}\boldsymbol{p}_k - \omega_{k+1}\boldsymbol{A}\boldsymbol{s}_k, \tag{4.284}$$

故可定义

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k + \omega_{k+1}\boldsymbol{s}_k. \tag{4.285}$$

这样定义 $\boldsymbol{x}_{k+1}$ 后, 其残量正好是式 (4.282) 所定义的 $\boldsymbol{r}_{k+1}$.

综合上面的讨论, 便得到如下的稳定化双共轭梯度算法.

算法 4.21 (BCGSTAB 方法) 给定线性方程组 (4.234), 初始向量 $\boldsymbol{x}_0$ 和 $\varepsilon > 0$. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{p}_0 = \boldsymbol{r}_0$;

选择 $\widetilde{\boldsymbol{r}}_0$ 使 $\rho_0 = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}}\boldsymbol{r}_0 \neq 0$; $k = 0$;

while $(\|\boldsymbol{r}_k\|_2 > \|\boldsymbol{r}_0\|_2\varepsilon)$

$k = k + 1$;

$\boldsymbol{u}_k = \boldsymbol{A}\boldsymbol{p}_k$; $\sigma_k = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}}\boldsymbol{u}_k$;

$\alpha_k = \rho_k/\sigma_k$; $\boldsymbol{s}_k = \boldsymbol{r}_k - \alpha_k\boldsymbol{u}_k$;

$\boldsymbol{q}_k = \boldsymbol{A}\boldsymbol{s}_k$; $\omega_{k+1} = (\boldsymbol{s}_k^{\mathrm{T}}\boldsymbol{q}_k)/(\boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{q}_k)$;

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k + \omega_{k+1}\boldsymbol{s}_k$;

$\boldsymbol{r}_{k+1} = \boldsymbol{s}_k - \omega_{k+1}\boldsymbol{q}_k$;

$\rho_{k+1} = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}}\boldsymbol{r}_{k+1}$;

$\beta_{k+1} = (\alpha_k\rho_{k+1})/(\omega_{k+1}\rho_k)$;

$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_{k+1}(\boldsymbol{p}_k - \omega_{k+1}\boldsymbol{u}_k)$;

end

算法 4.21 也避免了 $\boldsymbol{A}^{\mathrm{T}}$ 的出现, 而且每次迭代也只需作两次矩阵向量乘法. 实际应用显示这种方法收敛得较快, 而且残量的范数较平稳, 是一种比较好的算法.

BCGSTAB 方法的 MATLAB 程序如下:

```
%BCGSTAB方法程序-bcgstab.m
function [x,k,time,res,resvec]=bcgstab(A,b,x,max_it,tol)
tic; r=b-A*x; p=r; mr=norm(r);
rt=ones(length(b),1); %rt=r;
%rt=zeros(length(b),1); rt(1)=1;
rho=rt'*r; k=0;
while (k<=max_it)
    k=k+1; u=A*p; sigma=rt'*u;
```

```
    alpha=rho/sigma; s=r-alpha*u;
    v=A*s; omega=(s'*v)/(v'*v);
    x=x+alpha*p+omega*s;
    r=s-omega*v; rho1=rt'*r;
    beta=(alpha*rho1)/(omega*rho);
    p=r+beta*(p-omega*u);
    res=norm(r)/mr;resvec(k)=res;
    if (res<tol), break; end
    rho=rho1;
end
time=toc;
```

例 4.17 仍然考虑例 4.15 中的 $\boldsymbol{A}$ 和 $\boldsymbol{b}$. 将 BCGSTAB 算法应用到该线性方程组上, 取算法中的 $\widetilde{\boldsymbol{r}}_0 = \text{ones}(n,1)$, 迭代在 101 步后收敛 ($\varepsilon = 10^{-10}$), 计算得到的近似解 $\widehat{\boldsymbol{x}}$ 和真解 $\boldsymbol{x}^*$ 之间的绝对误差为

$$\|\widehat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 0.9536,$$

但计算解 $\widehat{\boldsymbol{x}}$ 的残量满足

$$\|\boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}}\|_2 = 7.8561 \times 10^{-8}.$$

迭代过程的收敛轨迹如图 4.18 所示, 其中横坐标为迭代步数 k, 纵坐标为相对残差 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$, 这里 $\boldsymbol{r}_k$ 是第 k 步得到的残差向量. 此时的收敛速度与 CGS 方法基本一样, 但是注意 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$ 已经变为呈平稳下降.

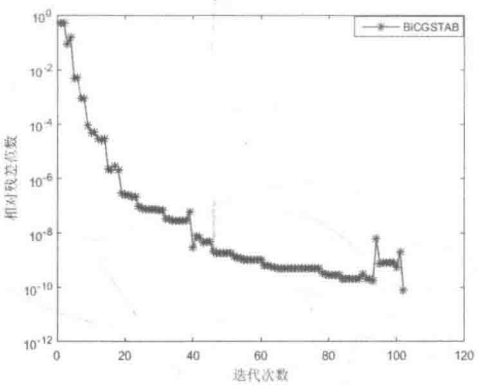

图 4.18 BCGSTAB 算法的收敛特性

最后, 关于预处理 BCGSTAB 算法. 对预处理方程组

$$\widehat{\boldsymbol{A}}\widehat{\boldsymbol{x}} = \widehat{\boldsymbol{b}}, \tag{4.286}$$

式中:

$$\widehat{\boldsymbol{A}} = \boldsymbol{M}_1^{-1}\boldsymbol{A}\boldsymbol{M}_2^{-1},\ \ \widehat{\boldsymbol{x}} = \boldsymbol{M}_2\boldsymbol{x},\ \ \widehat{\boldsymbol{b}} = \boldsymbol{M}_1^{-1}\boldsymbol{b}.$$

对预处理方程组 (4.286) 应用 BCGSTAB 算法, 再作代换:

$$\widehat{\boldsymbol{u}}_k = \boldsymbol{M}_1^{-1}\boldsymbol{u}_k,\ \widehat{\boldsymbol{s}}_k = \boldsymbol{M}_1^{-1}\boldsymbol{s}_k,\ \widehat{\boldsymbol{q}}_k = \boldsymbol{M}_1^{-1}\boldsymbol{q}_k,$$
$$\widehat{\boldsymbol{r}}_k = \boldsymbol{M}_1^{-1}\boldsymbol{r}_k,\ \widehat{\boldsymbol{p}}_k = \boldsymbol{M}_1^{-1}\boldsymbol{p}_k,\ \widehat{\boldsymbol{x}}_k = \boldsymbol{M}_2\boldsymbol{x}_k,\ \widehat{\widetilde{\boldsymbol{r}}}_0 = \boldsymbol{M}_1^{\mathrm{T}}\widetilde{\boldsymbol{r}}_0.$$

便得到如下预处理稳定化双共轭梯度算法.

算法 4.22 (PBCGSTAB 方法) 给定线性方程组 (4.234), 初始向量 $\boldsymbol{x}_0$ 和 $\varepsilon > 0$ 及预处理矩阵 $\boldsymbol{M} = \boldsymbol{M}_1\boldsymbol{M}_2$. 本算法计算 $\boldsymbol{x}_k$, 使得 $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

选取 $\boldsymbol{x}_0$; 计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$; $\boldsymbol{p}_0 = \boldsymbol{r}_0$;

选择 $\widetilde{\boldsymbol{r}}_0$ 使 $\rho_0 = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}}\boldsymbol{r}_0 \neq 0$; $k = 0$;

while $(\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2 > \varepsilon)$

$k = k + 1$;

由 $\boldsymbol{M}\boldsymbol{y}_k = \boldsymbol{p}_k$ 解得 $\boldsymbol{y}_k$;

$\boldsymbol{u}_k = \boldsymbol{A}\boldsymbol{y}_k$; $\sigma_k = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}}\boldsymbol{u}_k$;

$\alpha_k = \rho_k/\sigma_k$; $\boldsymbol{s}_k = \boldsymbol{r}_k - \alpha_k\boldsymbol{u}_k$;

由 $\boldsymbol{M}\boldsymbol{z}_k = \boldsymbol{s}_k$ 解得 $\boldsymbol{z}_k$; $\boldsymbol{q}_k = \boldsymbol{A}\boldsymbol{z}_k$;

由 $\boldsymbol{M}_1[\boldsymbol{\xi}_k, \boldsymbol{\eta}_k] = [\boldsymbol{q}_k, \boldsymbol{s}_k]$ 解得 $[\boldsymbol{\xi}_k, \boldsymbol{\eta}_k]$;

$\omega_{k+1} = (\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\eta}_k)/(\boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{\xi}_k)$;

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{y}_k + \omega_{k+1}\boldsymbol{z}_k$;

$\boldsymbol{r}_{k+1} = \boldsymbol{s}_k - \omega_{k+1}\boldsymbol{q}_k$;

$\rho_{k+1} = \widetilde{\boldsymbol{r}}_0^{\mathrm{T}}\boldsymbol{r}_{k+1}$; $\beta_{k+1} = (\alpha_k\rho_{k+1})/(\omega_{k+1}\rho_k)$;

$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_{k+1}(\boldsymbol{p}_k - \omega_{k+1}\boldsymbol{u}_k)$;

end

若考虑例 4.15 中的 $\boldsymbol{A}$ 和 $\boldsymbol{b}$. 将 PBCGSTAB 方法应用到该线性方程组上, 取算法中的 $\widetilde{\boldsymbol{r}}_0 = \boldsymbol{r}_0$, 预处理矩阵 $\boldsymbol{M} = \boldsymbol{M}_1\boldsymbol{M}_2$ 为矩阵 $\boldsymbol{A}$ 的不完全 LU 分解, 则只需迭代 1 步即满足终止准则 ($\varepsilon = 10^{-10}$), 计算得到的近似解 $\widehat{\boldsymbol{x}}$ 和真解 $\boldsymbol{x}^*$ 之间的误差为

$$\|\widehat{\boldsymbol{x}} - \boldsymbol{x}^*\|_2 = 5.7303 \times 10^{-15},$$

计算解 $\widehat{\boldsymbol{x}}$ 的残量满足

$$\|\boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}}\|_2 = 6.0738 \times 10^{-15}.$$

习题 4

4.1 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 是对称正定矩阵, $\boldsymbol{p}_1, \cdots, \boldsymbol{p}_k$ 是相互共轭正交的向量组, 即满足 $\boldsymbol{p}_i^{\mathrm{T}} \boldsymbol{A} \boldsymbol{p}_j = 0\,(i \neq j)$. 试证明: $\boldsymbol{p}_1, \cdots, \boldsymbol{p}_k$ 是线性无关的.

4.2 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 是对称正定矩阵, $\mathcal{X}$ 是 $\mathbb{R}^n$ 的一个 k 维子空间. 试证明: $\boldsymbol{x}_k \in \mathcal{X}$ 满足

$$\|\boldsymbol{x}_k - \boldsymbol{A}^{-1}\boldsymbol{b}\|_{\boldsymbol{A}} = \min_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{A}^{-1}\boldsymbol{b}\|_{\boldsymbol{A}}$$

的充分必要条件是 $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ 垂直于子空间 $\mathcal{X}$, 其中 $\boldsymbol{b} \in \mathbb{R}^n$ 是任意给定的向量.

4.3 设对称正定矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 至多有 l 个互不相同的特征值. 试证明: 在没有舍入误差的前提下, 共轭梯度法至多 l 步即可得到方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 的精确解.

4.4 设对称矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 只有 l 个互不相同的特征值, $\boldsymbol{b} \in \mathbb{R}^n$ 是任一向量, 试证明: 子空间 $\mathcal{X} = \mathrm{span}\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{n-1}\boldsymbol{b}\}$ 的维数至多为 l.

4.5 设 $\boldsymbol{r}_k$ 和 $\boldsymbol{z}_k$ 是由预处理共轭梯度法 (算法 4.2) 产生的. 证明: 若 $\boldsymbol{r}_k \neq \boldsymbol{0}$, 则必有 $\boldsymbol{z}_k^{\mathrm{T}}\boldsymbol{r}_k \neq 0$.

4.6 设

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

取 $m = 2$, $\boldsymbol{x}_0 = \boldsymbol{0}$. 分别用 GMRES 方法和 GMRES($m$) 方法计算出 $\boldsymbol{x}_1$.

4.7 证明: 若 GMRES 方法中出现 $h_{k+1,k} = 0$, 则相应的近似解 $\boldsymbol{x}_k$ 为精确解. 反之亦然.

4.8 证明: BCG 方法的残量 $\boldsymbol{r}_k$ 也有三项递推公式.

4.9 设矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 非奇异. 证明 GMRES 方法中的最小二乘问题 (4.55) 是满秩的.

4.10 设矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 非奇异. 对于方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, 使用内积 $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\cdot, \cdot)$, 导出在 Krylov 子空间

$$\mathrm{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \cdots, \boldsymbol{A}^k\boldsymbol{r}_0\}$$

上的最佳近似解的计算公式.

第 5 章　线性最小二乘问题的数值解法

当给定的矩阵 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 非奇异时, 对任何向量 $\boldsymbol{b} \in \mathbb{C}^n$, 线性方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 总有唯一解 $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$. 但在许多实际问题中, 矩阵 $\boldsymbol{A}$ 不是方阵, 甚至不是满秩的, 而且 $\boldsymbol{b} \notin \mathcal{R}(\boldsymbol{A})$. 此时, 方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 可能无解或者有无穷多个解. 本章将讨论这样的一类线性方程组的有关理论及数值方法.

5.1　线性最小二乘问题的数学性质

本节讨论线性最小二乘问题的有关性质. 首先给出最小二乘问题的定义.

定义 5.1　设 $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, $\boldsymbol{b} \in \mathbb{C}^m$, 确定 $\boldsymbol{x} \in \mathbb{C}^n$ 使得

$$\|\boldsymbol{Ax} - \boldsymbol{b}\|_2 = \min_{\boldsymbol{z}\in\mathbb{C}^n} \|\boldsymbol{Az} - \boldsymbol{b}\|_2. \tag{5.1}$$

问题 (5.1) 称为线性最小二乘问题 (Least Squares, LS 问题), 而 $\boldsymbol{x}$ 则称为最小二乘解或极小解. 称 $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{b} - \boldsymbol{Ax}$ 为残差向量 (简称残量).

所有最小二乘解的集合记为 S_{LS}, 即

$$S_{\mathrm{LS}} = \{\boldsymbol{x} \in \mathbb{C}^n : \boldsymbol{x} \text{ 满足 } (5.1)\}. \tag{5.2}$$

S_{LS} 中 2–范数最小者称为极小范数解, 记为 $\boldsymbol{x}_{\mathrm{LS}}$, 即

$$\|\boldsymbol{x}_{\mathrm{LS}}\|_2 = \min\{\|\boldsymbol{x}\|_2 : \boldsymbol{x} \in S_{\mathrm{LS}}\}.$$

线性最小二乘问题 (5.1) 的解 $\boldsymbol{x}$ 又可称为线性方程组

$$\boldsymbol{Ax} = \boldsymbol{b}, \quad \boldsymbol{A} \in \mathbb{C}^{m\times n},\ \boldsymbol{b} \in \mathbb{C}^m \tag{5.3}$$

的最小二乘解, 即 $\boldsymbol{x}$ 在残量 $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{b} - \boldsymbol{Ax}$ 的 2–范数最小的意义下满足方程组 (5.3). 当 $m > n$ 时称为超定方程组或矛盾方程组; 当 $m < n$ 时称为欠定方程组.

不难发现, 若将矩阵 $\boldsymbol{A}$ 写成 $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n]$, $\boldsymbol{a}_i \in \mathbb{C}^m$, $i = 1, 2, \cdots, n$, 则求解最小二乘问题 (5.1) 等价于求 $\{\boldsymbol{a}_i\}_{i=1}^n$ 的线性组合使之与向量 $\boldsymbol{b}$ 之差的 2–范数达到最小. 可分为两种情况: 第一种是 $\{\boldsymbol{a}_i\}_{i=1}^n$ 线性无关, 即 $\boldsymbol{A}$ 为列满秩; 第二种是 $\{\boldsymbol{a}_i\}_{i=1}^n$ 线性相关, 即 $\boldsymbol{A}$ 为秩亏的. 下面分别针对这两种情形讨论最小二乘问题 (5.1) 极小解的数值解法.

5.1.1　最小二乘解的特征及一般表示

矩阵的广义逆是研究线性方程组 (5.3) 最小二乘解的一个重要而有力的工具. 下面讨论:

(1) 当方程组 (5.3) 有解时, 如何确定 $\boldsymbol{x}_0 \in \mathbb{C}^n$, 使得

$$\|\boldsymbol{x}_0\|_2 = \min_{\boldsymbol{Ax}=\boldsymbol{b}} \|\boldsymbol{x}\|_2,$$

称这样的 $\boldsymbol{x}_0$ 为方程组 (5.3) 的极小范数解.

(2) 当方程组 (5.3) 无解时, 如何确定 $\boldsymbol{x}_0 \in \mathbb{C}^n$, 使得

$$\|\boldsymbol{x}_0\|_2 = \min_{\min \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2} \|\boldsymbol{x}\|_2,$$

称这样的 $\boldsymbol{x}_0$ 为方程组 (5.3) 的极小范数最小二乘解.

引理 5.1 设 $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, 则有

(1) $[\mathcal{R}(\boldsymbol{A})]^{\perp} = \mathcal{N}(\boldsymbol{A}^{\mathrm{H}})$, 并且 $\mathbb{C}^m = \mathcal{R}(\boldsymbol{A}) \bigoplus \mathcal{N}(\boldsymbol{A}^{\mathrm{H}})$ (或 $\mathcal{R}(\boldsymbol{A}) = [\mathcal{N}(\boldsymbol{A}^{\mathrm{H}})]^{\perp}$).

(2) $[\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})]^{\perp} = \mathcal{N}(\boldsymbol{A})$, 并且 $\mathbb{C}^n = \mathcal{R}(\boldsymbol{A}^{\mathrm{H}}) \bigoplus \mathcal{N}(\boldsymbol{A})$ (或 $\mathcal{N}(\boldsymbol{A}) = [\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})]^{\perp}$).

证明 (1) 将 $\boldsymbol{A}$ 按列划分为 $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n]$, 其中 $\boldsymbol{a}_i \in \mathbb{C}^m\,(i = 1, 2, \cdots, n)$, 由于

$$\mathcal{R}(\boldsymbol{A}) = \mathrm{span}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\},$$

所以

$$\begin{aligned}
[\mathcal{R}(\boldsymbol{A})]^{\perp} &= \{\boldsymbol{x} : \boldsymbol{x} \perp \boldsymbol{a}_i,\ i = 1, 2, \cdots, n\} \\
&= \{\boldsymbol{x} : \boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{x} = 0,\ i = 1, 2, \cdots, n\ \} \\
&= \{\boldsymbol{x} : \boldsymbol{A}^{\mathrm{H}}\boldsymbol{x} = \boldsymbol{0}\} = \mathcal{N}(\boldsymbol{A}^{\mathrm{H}}),
\end{aligned}$$

即 $\mathcal{R}(\boldsymbol{A}) \perp \mathcal{N}(\boldsymbol{A}^{\mathrm{H}})$, 从而 $\mathcal{R}(\boldsymbol{A}) + \mathcal{N}(\boldsymbol{A}^{\mathrm{H}})$ 为直和. 再由

$$\dim \mathcal{R}(\boldsymbol{A}) + \dim \mathcal{N}(\boldsymbol{A}^{\mathrm{H}}) = \mathrm{rank}(\boldsymbol{A}) + [m - \mathrm{rank}(\boldsymbol{A})] = m,$$

可得 $\mathbb{C}^m = \mathcal{R}(\boldsymbol{A}^{\mathrm{H}}) \bigoplus \mathcal{N}(\boldsymbol{A})$.

(2) 在 (1) 中以 $\boldsymbol{A}^{\mathrm{H}} \in \mathbb{C}^{n\times m}$ 代替 $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ 即可得 (2). 证毕. □

设矩阵 $\boldsymbol{A}$ 的奇异值分解为

$$\boldsymbol{A} = \boldsymbol{U} \begin{bmatrix} \boldsymbol{\Sigma}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \boldsymbol{V}^{\mathrm{H}},$$

式中: $\boldsymbol{U} = [\boldsymbol{U}_1, \boldsymbol{U}_2]$, $\boldsymbol{V} = [\boldsymbol{V}_1, \boldsymbol{V}_2]$ 为酉矩阵, $\boldsymbol{U}_1 = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_r]$, $\boldsymbol{U}_2 = [\boldsymbol{u}_{r+1}, \cdots, \boldsymbol{u}_n]$, $\boldsymbol{V}_1 = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_r]$, $\boldsymbol{V}_2 = [\boldsymbol{v}_{r+1}, \cdots, \boldsymbol{v}_n]$; $\boldsymbol{\Sigma}_r = \mathrm{diag}(\sigma_1, \cdots, \sigma_r)$, $\sigma_1 \geqslant \cdots \geqslant \sigma_r > 0$. 于是

$$\boldsymbol{A} = \boldsymbol{U}_1\boldsymbol{\Sigma}_r\boldsymbol{V}_1^{\mathrm{H}}, \quad \boldsymbol{A}^{\dagger} = \boldsymbol{V}_1\boldsymbol{\Sigma}_r^{-1}\boldsymbol{U}_1^{\mathrm{H}}, \quad \boldsymbol{A}^{\mathrm{H}} = \boldsymbol{V}_1\boldsymbol{\Sigma}_r\boldsymbol{U}_1^{\mathrm{H}}. \tag{5.4}$$

容易验证

$$\boldsymbol{A}\boldsymbol{A}^{\dagger} = \boldsymbol{U}_1\boldsymbol{U}_1^{\mathrm{H}}, \quad \boldsymbol{A}^{\dagger}\boldsymbol{A} = \boldsymbol{V}_1\boldsymbol{V}_1^{\mathrm{H}}, \quad \boldsymbol{A}^{\mathrm{H}}\boldsymbol{A} = \boldsymbol{V}_1\boldsymbol{\Sigma}_r^2\boldsymbol{V}_1^{\mathrm{H}}. \tag{5.5}$$

由于 $\{\boldsymbol{u}_i\}_{i=1}^r$ 和 $\{\boldsymbol{v}_i\}_{i=r+1}^n$ 分别是 $\mathcal{R}(\boldsymbol{A})$ 和 $\mathcal{N}(\boldsymbol{A})$ 的标准正交基, 故 $\mathcal{R}(\boldsymbol{A})$ 和 $\mathcal{N}(\boldsymbol{A})$ 上的正交投影矩阵分别为:

$$\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})} = \boldsymbol{U}_1\boldsymbol{U}_1^{\mathrm{H}} = \boldsymbol{A}\boldsymbol{A}^{\dagger}, \quad \boldsymbol{P}_{\mathcal{N}(\boldsymbol{A})} = \boldsymbol{V}_2\boldsymbol{V}_2^{\mathrm{H}} = \boldsymbol{I}_n - \boldsymbol{V}_1\boldsymbol{V}_1^{\mathrm{H}} = \boldsymbol{I}_n - \boldsymbol{A}^{\dagger}\boldsymbol{A}. \tag{5.6}$$

而 $\mathcal{R}(\boldsymbol{A})^{\perp}=\mathcal{N}(\boldsymbol{A}^{\mathrm{H}})$ 和 $\mathcal{N}(\boldsymbol{A})^{\perp}=\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})$ 上的正交投影矩阵分别为

$$\boldsymbol{P}_{\mathcal{R}(\boldsymbol{A})^{\perp}}=\boldsymbol{I}_m-\boldsymbol{A}\boldsymbol{A}^{\dagger},\quad \boldsymbol{P}_{\mathcal{N}(\boldsymbol{A})^{\perp}}=\boldsymbol{A}^{\dagger}\boldsymbol{A}. \tag{5.7}$$

有下面的定理.

定理 5.1 设 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$, $\boldsymbol{b}\in\mathbb{C}^m$, 则线性方程组 (5.3) 有解的充要条件是

$$\boldsymbol{A}\boldsymbol{A}^{\dagger}\boldsymbol{b}=\boldsymbol{b}, \tag{5.8}$$

并且在有解时, 其通解为

$$\boldsymbol{x}=\boldsymbol{A}^{\dagger}\boldsymbol{b}+(\boldsymbol{I}-\boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{z}, \tag{5.9}$$

其中 $\boldsymbol{z}\in\mathbb{C}^n$ 任意.

证明 若 $\boldsymbol{A}\boldsymbol{A}^{\dagger}\boldsymbol{b}=\boldsymbol{b}$, 则显然方程组 (5.3) 有解 $\boldsymbol{x}=\boldsymbol{A}^{\dagger}\boldsymbol{b}$. 反之, 若 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 则

$$\boldsymbol{A}\boldsymbol{A}^{\dagger}\boldsymbol{b}=\boldsymbol{A}\boldsymbol{A}^{\dagger}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}.$$

下面证明其通解为式 (5.9). 事实上, 根据式 (5.6), 有

$$\mathcal{N}(\boldsymbol{A})=\{(\boldsymbol{I}_n-\boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{z}:\boldsymbol{z}\in\mathbb{C}^n\}.$$

故线性方程组 (5.3) 的通解可以表示为 $\boldsymbol{x}=\boldsymbol{A}^{\dagger}\boldsymbol{b}+(\boldsymbol{I}-\boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{z}$, 其中 $\boldsymbol{z}\in\mathbb{C}^n$ 任意. 证毕. □

注 5.1 式 (5.9) 表明: $\boldsymbol{x}_0=\boldsymbol{A}^{\dagger}\boldsymbol{b}$ 是方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的一个解, 而 $(\boldsymbol{I}-\boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{z}$ 是对应齐次方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0}$ 的通解.

定理 5.2 如果方程组 (5.3) 有解, 则它的极小范数解 $\boldsymbol{x}_0$ 唯一, 并且 $\boldsymbol{x}_0=\boldsymbol{A}^{\dagger}\boldsymbol{b}$.

证明 先证 $\boldsymbol{x}_0\in\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})$. 若 $\boldsymbol{x}_0\notin\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})$, 则由引理 5.1 (2) 作向量分解, 得

$$\boldsymbol{x}_0=\boldsymbol{y}_0+\boldsymbol{y}_1,\quad \boldsymbol{y}_0\in\mathcal{R}(\boldsymbol{A}^{\mathrm{H}}),\quad \boldsymbol{y}_1\in\mathcal{N}(\boldsymbol{A})=[\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})]^{\perp},\quad \boldsymbol{y}_1\neq 0.$$

由于 $\boldsymbol{y}_0\perp\boldsymbol{y}_1$, 所以

$$\|\boldsymbol{x}_0\|_2^2=\|\boldsymbol{y}_0\|_2^2+\|\boldsymbol{y}_1\|_2^2>\|\boldsymbol{y}_0\|_2^2.$$

但是 $\boldsymbol{A}\boldsymbol{y}_0=\boldsymbol{A}\boldsymbol{y}_0+\boldsymbol{A}\boldsymbol{y}_1=\boldsymbol{A}\boldsymbol{x}_0=\boldsymbol{b}$, 所以 $\boldsymbol{x}_0$ 不是方程组 (5.3) 的极小范数解. 这与前提冲突, 故 $\boldsymbol{x}_0\in\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})$.

再证 $\boldsymbol{x}_0$ 唯一. 若 $\boldsymbol{z}_0$ 也是方程 (5.3) 的极小范数解, 则 $\boldsymbol{z}_0\in\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})$, 从而

$$\boldsymbol{x}_0-\boldsymbol{z}_0\in\mathcal{R}(\boldsymbol{A}^{\mathrm{H}})=[\mathcal{N}(\boldsymbol{A})]^{\perp}.$$

另外, 由于 $\boldsymbol{A}(\boldsymbol{x}_0-\boldsymbol{z}_0)=\boldsymbol{0}$, 所以 $\boldsymbol{x}_0-\boldsymbol{z}_0\in\mathcal{N}(\boldsymbol{A})$, 故只能有 $\boldsymbol{x}_0-\boldsymbol{z}_0=\boldsymbol{0}$. 即 $\boldsymbol{x}_0=\boldsymbol{z}_0$.

最后证 $\boldsymbol{x}_0=\boldsymbol{A}^{\dagger}\boldsymbol{b}$. 根据定理 5.1 可得方程组 (5.3) 的通解为

$$\boldsymbol{x}=\boldsymbol{A}^{\dagger}\boldsymbol{b}+(\boldsymbol{I}-\boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{z}\ \ (\boldsymbol{z}\in\mathbb{C}^n\text{ 任意}).$$

取 $\boldsymbol{z}=\boldsymbol{0}$, 则 $\boldsymbol{x}_0=\boldsymbol{A}^{\dagger}\boldsymbol{b}$ 是方程组 (5.3) 的一个解. 因为

$$\begin{aligned}(\boldsymbol{x}_0,(\boldsymbol{I}-\boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{z}) &= \boldsymbol{z}^{\mathrm{H}}(\boldsymbol{I}-\boldsymbol{A}^{\dagger}\boldsymbol{A})^{\mathrm{H}}\boldsymbol{x}_0\\ &= \boldsymbol{z}^{\mathrm{H}}(\boldsymbol{I}-\boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{A}^{\dagger}\boldsymbol{b}\\ &= \boldsymbol{z}^{\mathrm{H}}(\boldsymbol{A}^{\dagger}-\boldsymbol{A}^{\dagger}\boldsymbol{A}\boldsymbol{A}^{\dagger})\boldsymbol{b}=\boldsymbol{0},\end{aligned}$$

所以

$$\|\boldsymbol{x}\|_2^2=\|\boldsymbol{x}_0\|_2^2+\|(\boldsymbol{I}-\boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{z}\|_2^2\geqslant\|\boldsymbol{x}_0\|_2^2.$$

故 $\boldsymbol{x}_0=\boldsymbol{A}^{\dagger}\boldsymbol{b}$ 是方程组 (5.3) 的极小范数解. 证毕. □

定理 5.3 *如果线性方程组 (5.3) 无解, 则它的极小范数最小二乘解 $\boldsymbol{x}_0$ 唯一, 并且 $\boldsymbol{x}_0=\boldsymbol{A}^{\dagger}\boldsymbol{b}$.*

证明 由于方程组 (5.3) 无解, 所以 $\boldsymbol{b}\notin\mathcal{R}(\boldsymbol{A})$. 根据引理 5.1(1) 作向量分解 $\boldsymbol{b}=\boldsymbol{b}_1+\boldsymbol{b}_2$, 其中 $\boldsymbol{b}_1\in\mathcal{R}(\boldsymbol{A})$, $\boldsymbol{b}_2\in\mathcal{N}(\boldsymbol{A}^{\mathrm{H}})=[\mathcal{R}(\boldsymbol{A})]^{\perp}$ 且 $\boldsymbol{b}_2\neq\boldsymbol{0}$. 注意到 $\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}_1\in\mathcal{R}(\boldsymbol{A})$ 及 $\boldsymbol{b}_2\in[\mathcal{R}(\boldsymbol{A})]^{\perp}$, 可得 $(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}_1)\perp\boldsymbol{b}_2$, 于是有

$$\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2=\|(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}_1)+(-\boldsymbol{b}_2)\|_2^2=\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}_1\|_2^2+\|\boldsymbol{b}_2\|_2^2,$$

这表明 $\min\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$ 等价于 $\min\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}_1\|_2$.

因为 $\boldsymbol{b}_1\in\mathcal{R}(\boldsymbol{A})$, 所以方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}_1$ 有解, 根据定理 5.2 可得它的唯一极小范数解为 $\boldsymbol{A}^{\dagger}\boldsymbol{b}_1$. 由于 $\boldsymbol{b}_2\in\mathcal{N}(\boldsymbol{A}^{\mathrm{H}})$, 所以 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{b}_2=\boldsymbol{0}$. 于是得

$$\boldsymbol{A}^{\dagger}\boldsymbol{b}_2=\boldsymbol{A}^{\dagger}\boldsymbol{A}\boldsymbol{A}^{\dagger}\boldsymbol{b}_2=\boldsymbol{A}^{\dagger}(\boldsymbol{A}\boldsymbol{A}^{\dagger})^{\mathrm{H}}\boldsymbol{b}_2=\boldsymbol{A}^{\dagger}(\boldsymbol{A}^{\dagger})^{\mathrm{H}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{b}_2=\boldsymbol{0},$$

所以 $\boldsymbol{x}_0=\boldsymbol{A}^{\dagger}\boldsymbol{b}=\boldsymbol{A}^{\dagger}(\boldsymbol{b}_1+\boldsymbol{b}_2)=\boldsymbol{A}^{\dagger}\boldsymbol{b}_1$ 是方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}_1$ 的唯一极小范数解, 也是方程组 (5.3) 的唯一极小范数最小二乘解. 证毕. □

5.1.2 线性 LS 的等价性问题

1. *法方程*

下面的定理给出了最小二乘问题极小解的一个刻画.

定理 5.4 *$\boldsymbol{x}$ 是最小二乘问题 (5.1) 的极小解, 即 $\boldsymbol{x}\in S_{\mathrm{LS}}$ 的充分必要条件是 $\boldsymbol{x}$ 为方程*

$$\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}^{\mathrm{H}}\boldsymbol{b} \tag{5.10}$$

的解, 其中式 (5.10) 称为最小二乘问题的法方程.

证明 注意到最小二乘问题 (5.1) 等价于极小化问题

$$\min\varphi(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2=\frac{1}{2}\boldsymbol{x}^{\mathrm{H}}(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A})\boldsymbol{x}-(\boldsymbol{b}^{\mathrm{H}}\boldsymbol{A})\boldsymbol{x}+\frac{1}{2}\|\boldsymbol{b}\|_2^2.$$

由于 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 是半正定矩阵, 因此 n 元实函数 $\varphi(\boldsymbol{x})$ 是凸函数, 故 $\boldsymbol{x}$ 是最小二乘问题 (5.1) 的极小解等价于

$$\nabla\varphi(\boldsymbol{x}) = \boldsymbol{A}^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = \boldsymbol{0}.$$

证毕. □

定理 5.4 说明最小二乘问题 (5.1) 与法方程 (5.10) 是等价的, 即法方程 (5.10) 与最小二乘问题 (5.1) 有相同的解集.

2. KKT 方程

下面的定理给出最小二乘问题 (5.1) 的另一个等价性问题.

定理 5.5 设 $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^n$. 则 $\boldsymbol{x}$ 和 $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$ 分别为最小二乘问题 (5.1) 的极小解和残量的充分必要条件是 $\boldsymbol{x}$ 和 $\boldsymbol{r}$ 为鞍点系统

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{A} \\ \boldsymbol{A}^{\mathrm{H}} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \tag{5.11}$$

的解. 上述线性系统称为最小二乘问题的 Karush–Kuhn–Tucker 方程 (KKT 方程).

证明 若 $\boldsymbol{x}$ 为最小二乘问题 (5.1) 的极小解, 而 $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$ 为其残量, 则

$$\boldsymbol{x} = \boldsymbol{A}^{\dagger}\boldsymbol{b} + (\boldsymbol{I} - \boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{z}, \quad \boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{A}^{\dagger}\boldsymbol{b} = (\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\dagger})\boldsymbol{b}.$$

由广义逆 $\boldsymbol{A}^{\dagger}$ 的性质及等式 $\boldsymbol{r} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, 得

$$\boldsymbol{A}^{\mathrm{H}}\boldsymbol{r} = \boldsymbol{A}^{\mathrm{H}}(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\dagger})\boldsymbol{b} = \left[(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\dagger})\boldsymbol{A}\right]^{\mathrm{H}}\boldsymbol{b} = \boldsymbol{0}.$$

故式 (5.11) 是相容的线性系统, 且 $\boldsymbol{x}$ 和 $\boldsymbol{r}$ 满足式 (5.11).

反之, 通过验证广义逆的四个条件, 可以验证

$$\boldsymbol{B}^{\dagger} \equiv \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A} \\ \boldsymbol{A}^{\mathrm{H}} & \boldsymbol{O} \end{bmatrix}^{\dagger} = \begin{bmatrix} \boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\dagger} & (\boldsymbol{A}^{\dagger})^{\mathrm{H}} \\ \boldsymbol{A}^{\dagger} & -\boldsymbol{A}^{\dagger}(\boldsymbol{A}^{\dagger})^{\mathrm{H}} \end{bmatrix}.$$

故式 (5.11) 的任一解向量 $\boldsymbol{x}, \boldsymbol{r}$ 有如下形式

$$\begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{x} \end{bmatrix} = \boldsymbol{B}^{\dagger}\begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} + (\boldsymbol{I} - \boldsymbol{B}^{\dagger}\boldsymbol{B})\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^{\dagger})\boldsymbol{b} \\ \boldsymbol{A}^{\dagger}\boldsymbol{b} - (\boldsymbol{I} - \boldsymbol{A}^{\dagger}\boldsymbol{A})\boldsymbol{z} \end{bmatrix},$$

式中: $\boldsymbol{y} \in \mathbb{C}^m$, $\boldsymbol{z} \in \mathbb{C}^n$ 为任意向量. 故满足式 (5.11) 的任一组向量 $\boldsymbol{x}, \boldsymbol{r}$ 分别为式 (5.1) 的极小解和残量. 证毕. □

5.1.3 线性最小二乘问题的正则化

设矩阵 $\boldsymbol{A} \in \mathbb{C}_r^{m\times n}$ 的奇异值分解为

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}, \tag{5.12}$$

式中: $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_m]$ 为 m 阶酉矩阵; $\boldsymbol{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n]$ 为 n 阶酉矩阵; $l=\min\{m,n\}$; $\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_r>0=\sigma_{r+1}=\cdots=\sigma_l=0$,

$$\boldsymbol{\Sigma}=\operatorname{diag}(\sigma_1,\sigma_2,\cdots,\sigma_l). \tag{5.13}$$

则最小二乘问题 (5.1) 的极小范数最小二乘解 $\boldsymbol{x}_{\mathrm{LS}}$ 可表示为

$$\boldsymbol{x}_{\mathrm{LS}}=\boldsymbol{A}^{\dagger}\boldsymbol{b}=\sum_{i=1}^{r}\frac{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{b}}{\sigma_i}\boldsymbol{v}_i. \tag{5.14}$$

由于舍入误差的存在, 所有用于计算 $\boldsymbol{A}^{\dagger}$ 和 $\boldsymbol{x}_{\mathrm{LS}}$ 的算法, 实际上是计算某个扰动矩阵 $\widehat{\boldsymbol{A}}=\boldsymbol{A}+\Delta\boldsymbol{A}$ 的广义逆 $\widehat{\boldsymbol{A}}^{\dagger}$ 以及 $\widehat{\boldsymbol{x}}_{\mathrm{LS}}=\widehat{\boldsymbol{A}}^{\dagger}\widehat{\boldsymbol{b}}$.

当 $\boldsymbol{A}$ 为病态的满秩矩阵, 即存在 $k\,(1\leqslant k<l)$, 使得 $\sigma_k\gg\sigma_{k+1}\approx 0$. 由于在极小解 $\boldsymbol{x}_{\mathrm{LS}}$ 的表达式中存在项

$$\frac{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{b}}{\sigma_i}\boldsymbol{v}_i.\ i=k+1,\cdots,l,$$

因此, 对 $\boldsymbol{A}$ 的较小扰动, 将会使极小解 $\boldsymbol{x}_{\mathrm{LS}}$ 产生很大的误差. 此时, 称 $\boldsymbol{A}$ 为*数值秩亏*的. 当矩阵 $\boldsymbol{A}$ 秩亏而 $\Delta\boldsymbol{A}$ 很小, 则广义逆的不连续性表明了原来意义上的秩不再适用于数值计算. 当 $\widehat{\boldsymbol{A}}$ 和 $\boldsymbol{A}$ 的秩不相同时, $\widehat{\boldsymbol{A}}^{\dagger}$ 和 $\boldsymbol{A}^{\dagger}$, $\widehat{\boldsymbol{x}}_{\mathrm{LS}}$ 和 $\boldsymbol{x}_{\mathrm{LS}}$ 可能会相差很大, 而且扰动 $\Delta\boldsymbol{A}$ 越小, 其相差的程度会越大.

1. *截断的 LS 问题*

最小二乘问题的第一种正则化方法是截断的 LS 问题. 首先给出矩阵 $\boldsymbol{A}$ 的 δ 秩的定义.

定义 5.2 *设 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$ 和 $\delta>0$ 给定. 称数*

$$k=\min_{\boldsymbol{B}\in\mathbb{C}^{m\times n}}\{\operatorname{rank}(\boldsymbol{B}):\|\boldsymbol{A}-\boldsymbol{B}\|_2\leqslant\delta\} \tag{5.15}$$

为矩阵 $\boldsymbol{A}$ 的 δ 秩.

由定义 5.2 和矩阵的降秩最佳逼近定理, 当 $k<l$ 时, 有

$$\|\boldsymbol{A}-\boldsymbol{A}_k\|_2=\min_{\operatorname{rank}(\boldsymbol{B})\leqslant k}\|\boldsymbol{A}-\boldsymbol{B}\|_2=\sigma_{k+1},$$

式中:

$$\boldsymbol{A}_k=\sum_{i=1}^{k}\sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^{\mathrm{H}}.$$

因此, 矩阵 $\boldsymbol{A}$ 的 δ 秩为 k 的充分必要条件是

$$\sigma_1\geqslant\cdots\geqslant\sigma_k>\sigma_{k+1}\geqslant\cdots\geqslant\sigma_l.$$

对于最小二乘问题 (5.1), 截断的 LS 问题为

$$\|\boldsymbol{A}_k\boldsymbol{x}-\boldsymbol{b}\|_2=\min_{\boldsymbol{z}\in\mathbb{C}^n}\|\boldsymbol{A}_k\boldsymbol{z}-\boldsymbol{b}\|_2, \tag{5.16}$$

式中: $\boldsymbol{A}_k$ 为 $\boldsymbol{A}$ 的最佳秩-k 逼近. 此时, 式 (5.16) 的极小范数最小二乘解 $\bar{\boldsymbol{x}}_{\mathrm{LS}}$ 可表示为

$$\bar{\boldsymbol{x}}_{\mathrm{LS}} = \boldsymbol{A}_k^{\dagger}\boldsymbol{b} = \sum_{i=1}^{k} \frac{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{b}}{\sigma_i}\boldsymbol{v}_i. \tag{5.17}$$

2. Tikhonov 正则化

最小二乘问题的另一种正则化方法是 Tikhonov 正则化, 即考虑如下的正则化问题:

$$\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \tau^2\|\boldsymbol{D}\boldsymbol{x}\|_2^2 = \min_{\boldsymbol{z}\in\mathbb{C}^n} \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{b}\|_2^2 + \tau^2\|\boldsymbol{D}\boldsymbol{z}\|_2^2, \tag{5.18}$$

式中: $\tau > 0$; $\boldsymbol{D} = \mathrm{diag}(d_1, d_2, \cdots, d_n)$ 为正定的对角矩阵.

易见, 式 (5.18) 等价于

$$\left\| \begin{bmatrix} \tau\boldsymbol{D} \\ \boldsymbol{A} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b} \end{bmatrix} \right\|_2^2 = \min_{\boldsymbol{z}\in\mathbb{C}^n} \left\| \begin{bmatrix} \tau\boldsymbol{D} \\ \boldsymbol{A} \end{bmatrix} \boldsymbol{z} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b} \end{bmatrix} \right\|_2^2, \tag{5.19}$$

当 $\tau > 0$ 时, 系数矩阵是列满秩的, 因此有唯一的最小二乘解.

设 $m \geqslant n$. 则当 $\boldsymbol{D} = \boldsymbol{I}$ 时, 式 (5.19) 系数矩阵的奇异值为 $\tilde{\sigma}_i = \sqrt{\sigma_i + \tau^2}$, $i = 1, 2, \cdots, n$. 此时式 (5.18) 的解可表示为

$$\boldsymbol{x}(\tau) = \sum_{i=1}^{n} \frac{(\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{b})\sigma_i}{\sigma_i^2 + \tau^2}\boldsymbol{v}_i = \sum_{i=1}^{n} \frac{(\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{b})\eta_i}{\sigma_i}\boldsymbol{v}_i, \quad \eta_i = \frac{\sigma_i^2}{\sigma_i^2 + \tau^2}, \tag{5.20}$$

式中: η_i 为滤波因子. 当 $\tau \ll \sigma_i$ 时, $\eta_i \approx 1$; 当 $\tau \gg \sigma$ 时, $\eta_i \ll 1$.

正则化问题 (5.18) 的优点是它的解可以通过 QR 分解

$$\begin{bmatrix} \tau\boldsymbol{D} \\ \boldsymbol{A} \end{bmatrix} = \boldsymbol{Q}\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix}$$

得到.

注 5.2 最小二乘问题的正则化已经把原来的最小二乘问题化为新的最小二乘问题. 事实上, 由式 (5.17) 和式 (5.20) 可以看出, 截断的 LS 问题和 Tikhonov 正则化的 LS 问题, 与最小二乘问题 (5.1) 的极小范数最小二乘解以及相应的残量各不相同. 正则化方法通常用于处理病态最小二乘问题和反问题等不适定问题.

5.2 求解满秩最小二乘问题的数值方法

本节假定最小二乘问题 (5.1) 中的矩阵 $\boldsymbol{A} \in \mathbb{C}^{m\times n}(m \geqslant n)$ 为列满秩矩阵, 即 $\mathrm{rank}(\boldsymbol{A}) = n$. 此时最小二乘问题 (5.1) 有唯一的极小范数最小二乘解 $\boldsymbol{x}_{\mathrm{LS}}$, 并且连续地依赖给定的数据 $\boldsymbol{A}$ 和 $\boldsymbol{b}$. 对于这类问题, 介绍两种最基本的数值方法.

5.2.1 法方程方法

由定理 5.4, 最小二乘问题 (5.1) 等价于其法方程

$$\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}^{\mathrm{H}}\boldsymbol{b}.$$

当 $\mathrm{rank}(\boldsymbol{A})=n$ 时, $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 为 Hermite 正定矩阵, 法方程 (5.10) 的唯一解可以用 Cholesky 分解法求得. 写出算法步骤如下:

算法 5.1 (法方程 Cholesky 分解法) 给定 $\boldsymbol{A}\in\mathbb{C}^{m\times n}(m\geqslant n)$ 为列满秩矩阵, $\boldsymbol{b}\in\mathbb{C}^{m}$. 本算法计算 $\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$ 的极小解 $\boldsymbol{x}_{\mathrm{LS}}$.

步 1, 对 n 阶 Hermite 正定矩阵 $\boldsymbol{B}=\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 作 Cholesky 分解 $\boldsymbol{B}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{H}}$, 其中 $\boldsymbol{L}$ 为下三角矩阵.

步 2, 依次解 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{A}^{\mathrm{H}}\boldsymbol{b}$, $\boldsymbol{L}^{\mathrm{H}}\boldsymbol{x}=\boldsymbol{y}$ 得到最小二乘问题 (5.1) 的解 $\boldsymbol{x}_{\mathrm{LS}}$.

n 阶 Hermite 正定矩阵 $\boldsymbol{B}$ 的 Cholesky 分解需要 $O(n^3/3)$ 次乘法. 而计算 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 需要 mn^2 次乘法. 在算法 5.1 中第 2 步的计算量 (约 $O(n^2)$ 次乘法) 与第 1 步相比可以忽略. 所以该算法需要的乘法次数为 $O(mn^2+n^3/3)$. 当 $m\gg n$ 时, 算法的主要工作量在于 $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}$ 的计算.

5.2.2 QR 分解方法

本节考虑最小二乘问题 (5.1) 中的矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times n}(m\geqslant n)$ 的情形. 根据正交矩阵保持向量 2-范数不变的性质, 对于任意的正交矩阵 $\boldsymbol{Q}\in\mathbb{R}^{m\times m}$, 最小二乘问题 (5.1) 等价于

$$\|\boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})\|_2=\min\left\{\|\boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{z}-\boldsymbol{b})\|_2:\boldsymbol{z}\in\mathbb{R}^{n}\right\}. \tag{5.21}$$

这样, 就可望通过适当选取正交矩阵 $\boldsymbol{Q}$, 使原问题 (5.1) 转化为较为容易求解的最小二乘问题 (5.21), 这就是正交化方法—QR 分解方法的基本思想.

设 $\boldsymbol{A}$ 有 QR 分解:

$$\boldsymbol{A}=\boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix}=\boldsymbol{Q}_1\boldsymbol{R}, \tag{5.22}$$

式中: $\boldsymbol{Q}\in\mathbb{R}^{m\times m}$ 为正交矩阵; $\boldsymbol{Q}_1$ 为 $\boldsymbol{Q}$ 的前 n 列组成的矩阵; $\boldsymbol{R}\in\mathbb{R}^{n\times}$ 为对角元均为正的上三角矩阵.

现取式 (5.21) 中的正交矩阵为分解式 (5.22) 中的 $\boldsymbol{Q}$, 并记

$$\boldsymbol{f}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}=\begin{bmatrix}\boldsymbol{Q}_1^{\mathrm{T}}\\ \boldsymbol{Q}_2^{\mathrm{T}}\end{bmatrix}\boldsymbol{b}=\begin{bmatrix}\boldsymbol{f}_1\\ \boldsymbol{f}_2\end{bmatrix}\begin{matrix}n\\ m-n\end{matrix}, \tag{5.23}$$

则对任意的 $\boldsymbol{x}\in\mathbb{C}^{n}$, 有

$$\begin{aligned}\|\boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})\|_2^2&=\left\|\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix}\boldsymbol{x}-\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}\right\|_2^2=\left\|\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix}\boldsymbol{x}-\begin{bmatrix}\boldsymbol{f}_1\\ \boldsymbol{f}_2\end{bmatrix}\right\|_2^2\\ &=\|\boldsymbol{R}\boldsymbol{x}-\boldsymbol{f}_1\|_2^2+\|\boldsymbol{f}_2\|_2^2.\end{aligned}$$

由此可知, $\boldsymbol{x}$ 是最小二乘问题 (5.1) 的解, 当且仅当 $\boldsymbol{x}$ 是上三角形方程组 $\boldsymbol{R}\boldsymbol{x}=\boldsymbol{f}_1$ 的解.

根据上述讨论, 有如下的 QR 分解算法. 在实施过程中, 采取对增广矩阵 $\widetilde{\boldsymbol{A}}=[\boldsymbol{A},\boldsymbol{b}]$ 进行 QR 分解:

$$\boldsymbol{H}_n\cdots\boldsymbol{H}_2\boldsymbol{H}_1[\boldsymbol{A},\boldsymbol{b}]=[\boldsymbol{R},\widetilde{\boldsymbol{b}}]\Longrightarrow\boldsymbol{Q}^{\mathrm{T}}[\boldsymbol{A},\boldsymbol{b}]=[\boldsymbol{R},\widetilde{\boldsymbol{b}}],\tag{5.24}$$

得

$$\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R},\quad\widetilde{\boldsymbol{b}}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b},$$

式中: $\boldsymbol{Q}=\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_n$. 式 (5.24) 表明, 对增广矩阵 $\widetilde{\boldsymbol{A}}=[\boldsymbol{A},\boldsymbol{b}]$ 实施 QR 分解, 当把矩阵 $\boldsymbol{A}$ 约化为上三角矩阵 $\boldsymbol{R}$ 时, 向量 $\boldsymbol{b}$ 约化成了 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}$, 这正是所需要的.

算法 5.2 (LS 问题 QR 分解法) 给定 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ $(m\geqslant n)$ 为列满秩矩阵, $\boldsymbol{b}\in\mathbb{R}^m$. 本算法计算 $\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$ 的极小解 $\boldsymbol{x}_{\mathrm{LS}}$.

步 1, 计算增广矩阵 $\widetilde{\boldsymbol{A}}=[\boldsymbol{A},\boldsymbol{b}]$ 的 QR 分解并覆盖 $\widetilde{\boldsymbol{A}}$.

步 2, 用回代法求解上三角形方程组 $\mathrm{triu}(\widetilde{\boldsymbol{A}}(1:n.1:n))\boldsymbol{x}=\widetilde{\boldsymbol{A}}(1:n,n+1)$, 得到最小二乘解 $\boldsymbol{x}_{\mathrm{LS}}$, 其中记号 $\mathrm{triu}(\boldsymbol{B})$ 表示提取矩阵 $\boldsymbol{B}$ 的上三角部分组成的上三角形矩阵.

用算法 5.2 求解满秩最小二乘问题需要 $2mn^2-2n^3/3$ 个 flop. 更新 $\boldsymbol{b}$ 所需的 $O(mn)$ 个 flop 和回代法所需的 $O(n^2)$ 个 flop 与分解 $\boldsymbol{A}$ 所需的工作量相比是微不足道的.

注 5.3 算法 5.2 的主要工作量集中在增广矩阵 $\widetilde{\boldsymbol{A}}$ 的 QR 分解. 可用如下三种方法之一实现这一分解:

(1) Householder 方法.

(2) Givens 正交化方法.

(3) 修正的 Gram-Schmit 方法.

根据算法 5.2, 采用 Householder 变换 QR 分解编制 MATLAB 程序如下:

```
function [x]=ls_houseqr(A,b)
%用Householder变换QR分解求最小二乘问题min||Ax-b||
[m,n]=size(A);
[A]=house_qr([A,b]);  %调用Householder变换QR分解程序
x=triu(A(1:n,1:n))\A(1:n,n+1);
```

例 5.1 用 MATLAB 程序 ls_houseqr.m 求解最小二乘问题 $\min\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$, 其中

$$\boldsymbol{A}=\begin{bmatrix}2&3&4&5\\4&3&2&1\\4&5&6&7\\9&5&7&2\\4&2&5&3\end{bmatrix},\quad\boldsymbol{x}=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix},\quad\boldsymbol{b}=\begin{bmatrix}20\\22\\35\\42\\50\end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[2 3 4 5; 4 3 2 1; 4 5 6 7; 9 5 7 2; 4 2 5 3];
>> b=[20 22 35 42 50]';
>> x=ls_houseqr(A,b)
```

即可得计算结果.

5.3 求解秩亏最小二乘问题的数值解法

本节讨论秩亏最小二乘问题 $\min \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$ 的数值解法. 如果 $\boldsymbol{A}\in\mathbb{R}_r^{m\times n}\,(m\geqslant n)$ 是秩亏损的, 即 $\operatorname{rank}(\boldsymbol{A})=r<n$, 此时, 最小二乘问题 (5.1) 有无穷多个解, 且 5.2 节所介绍的处理满秩最小二乘问题的法方程方法和 QR 分解法都不再有效. 因此, 本节专门讨论秩亏最小二乘问题的数值求解方法.

5.3.1 列主元 QR 分解法

当矩阵 $\boldsymbol{A}$ 秩亏损时, 其 QR 分解不一定能给出列空间 $\mathcal{R}(\boldsymbol{A})$ 的一组标准正交基. 此时可计算经过列置换之后的矩阵 $\widetilde{\boldsymbol{A}}=\boldsymbol{A}\boldsymbol{P}$ 的 QR 分解来解决, 即 $\boldsymbol{A}\boldsymbol{P}=\boldsymbol{Q}\boldsymbol{R}$, 其中 $\boldsymbol{P}$ 是置换矩阵.

下面介绍列主元 QR 分解法来解决秩亏的最小二乘问题. 设最小二乘问题 (5.1) 中的已知向量 $\boldsymbol{b}$ 分解为

$$\boldsymbol{b}=\boldsymbol{b}_1+\boldsymbol{b}_2,\quad \boldsymbol{b}_1\in\mathcal{R}(\boldsymbol{A}),\ \boldsymbol{b}_2\in\mathcal{R}(\boldsymbol{A})^{\perp}. \tag{5.25}$$

易证问题 (5.1) 等价于

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}_1. \tag{5.26}$$

现假定 $\mathcal{R}(\boldsymbol{A})=\mathcal{R}(\boldsymbol{Q}_1)$, 其中 $\boldsymbol{Q}_1\in\mathbb{R}^{m\times r}$ 且 $\boldsymbol{Q}_1^{\mathrm{T}}\boldsymbol{Q}_1=\boldsymbol{I}_r$, 即 $\boldsymbol{Q}_1$ 的列构成 $\mathcal{R}(\boldsymbol{A})$ 的一组标准正交基. 则存在矩阵 $\boldsymbol{S}\in\mathbb{R}^{r\times n}$ 和向量 $\boldsymbol{h}\in\mathbb{R}^r$, 使得

$$\boldsymbol{A}=\boldsymbol{Q}_1\boldsymbol{S},\quad \boldsymbol{b}_1=\boldsymbol{Q}_1\boldsymbol{h}. \tag{5.27}$$

将式 (5.27) 代入式 (5.26), 并注意到 $\boldsymbol{Q}_1$ 的列线性无关, 可知 (5.26) 等价于

$$\boldsymbol{S}\boldsymbol{x}=\boldsymbol{h}. \tag{5.28}$$

显然方程组 (5.28) 总是有解的. 进一步, 由式 (5.27) 可知

$$\boldsymbol{S}=\boldsymbol{Q}_1^{\mathrm{T}}\boldsymbol{A},\quad \boldsymbol{h}=\boldsymbol{Q}_1^{\mathrm{T}}\boldsymbol{b}_1=\boldsymbol{Q}_1^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{b}_2)=\boldsymbol{Q}_1^{\mathrm{T}}\boldsymbol{b}. \tag{5.29}$$

因此, 只要求得 $\mathcal{R}(\boldsymbol{A})$ 的一组标准正交基, 就可以通过式 (5.29) 和式 (5.28) 求得最小二乘问题 (5.1) 的任一解.

现由于 $\operatorname{rank}(\boldsymbol{A})=r<n$, 2.2 节介绍的 QR 分解式 (2.9) 一般不能产生 $\mathcal{R}(\boldsymbol{A})$ 的一组标准正交基. 但如果先对 $\boldsymbol{A}$ 的列进行适当的排列使其前 r 列线性无关, 然后再进行 QR 分解, 则仍然可以产生 $\mathcal{R}(\boldsymbol{A})$ 的一组标准正交基.

设有分解式

$$\boldsymbol{AP} = \boldsymbol{Q}\begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}\begin{matrix} r \\ m-r \end{matrix}, \tag{5.30}$$
$$\quad\ r \quad n-r$$

式中: $\boldsymbol{P}$ 为置换矩阵; $\boldsymbol{Q}$ 为正交矩阵; $\boldsymbol{R}_{11}$ 为非奇异的上三角矩阵. 则 $\boldsymbol{Q}$ 的前 r 列就是 $\mathcal{R}(\boldsymbol{A})$ 的一组标准正交基.

一旦求出式 (5.30) 的分解式, 则由式 (5.26), 有

$$(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{AP})(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}) = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}_1,$$

由式 (5.30) 并令 $\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{z} \end{bmatrix}$, 得

$$\begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{z} \end{bmatrix} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}_1 = \begin{bmatrix} \boldsymbol{Q}_1^{\mathrm{T}} \\ \boldsymbol{Q}_2^{\mathrm{T}} \end{bmatrix}\boldsymbol{b}_1 = \begin{bmatrix} \boldsymbol{h} \\ \boldsymbol{g} \end{bmatrix},$$

即

$$\begin{bmatrix} \boldsymbol{R}_{11}\boldsymbol{w} + \boldsymbol{R}_{12}\boldsymbol{z} \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h} \\ \boldsymbol{g} \end{bmatrix}.$$

由此, 得

$$\boldsymbol{w} = \boldsymbol{R}_{11}^{-1}(\boldsymbol{h} - \boldsymbol{R}_{12}\boldsymbol{z}), \quad \boldsymbol{g} = \boldsymbol{0}.$$

故

$$\boldsymbol{x} = \boldsymbol{P}\begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{z} \end{bmatrix} = \boldsymbol{P}\begin{bmatrix} \boldsymbol{R}_{11}^{-1}(\boldsymbol{h} - \boldsymbol{R}_{12}\boldsymbol{z}) \\ \boldsymbol{z} \end{bmatrix}, \quad \boldsymbol{z} \in \mathbb{R}^{n-r},$$

这就是最小二乘问题 (5.1) 的通解. 注意到上式还可以写为

$$\boldsymbol{x} = \boldsymbol{x}_b + \boldsymbol{P}\begin{bmatrix} -\boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{12} \\ \boldsymbol{I}_{n-r} \end{bmatrix}\boldsymbol{z}, \quad \boldsymbol{x}_b = \boldsymbol{P}\begin{bmatrix} \boldsymbol{R}_{11}^{-1}\boldsymbol{h} \\ \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{z} \in \mathbb{R}^{n-r}, \tag{5.31}$$

式中: $\boldsymbol{x}_b$ 为最小二乘问题的基本解.

下面讨论分解式 (5.30) 的具体实施过程. 类似于分解式 (2.9) 的计算过程, 这一分解可用 Householder 变换和适当的初等列变换相结合逐步求得. 假设对某一正整数 k, 已经求得 $k-1$ 个 Householder 变换 $\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_{k-1}$ 和 $k-1$ 个初等变换矩阵 $\boldsymbol{P}_1, \boldsymbol{P}_2, \cdots, \boldsymbol{P}_{k-1}$, 使得

$$\boldsymbol{R}_{k-1} = (\boldsymbol{H}_{k-1}\cdots\boldsymbol{H}_2\boldsymbol{H}_1)\boldsymbol{A}(\boldsymbol{P}_1\boldsymbol{P}_2\cdots\boldsymbol{P}_{k-1})$$

$$= \begin{array}{c} \\ \\ k-1 \end{array} \begin{array}{c} k-1 \quad\quad n-k+1 \\ \left[\begin{array}{cc} \boldsymbol{R}_{11}^{(k-1)} & \boldsymbol{R}_{12}^{(k-1)} \\ \boldsymbol{O} & \boldsymbol{R}_{22}^{(k-1)} \end{array}\right] \\ m-k+1 \end{array}, \tag{5.32}$$

式中: $\boldsymbol{R}_{11}^{(k-1)}$ 为非奇异的上三角矩阵. 现记

$$\boldsymbol{R}_{22}^{(k-1)} = \left[\boldsymbol{v}_k^{(k-1)}, \boldsymbol{v}_{k+1}^{(k-1)}, \cdots, \boldsymbol{v}_n^{(k-1)}\right],$$

即 $\boldsymbol{v}_i^{(k-1)}$ 表示 $\boldsymbol{R}_{22}^{(k-1)}$ 的第 $i-k+1$ 列. 下一步是首先确定指标 $p\,(k \leqslant p \leqslant n)$ 满足

$$\left\|\boldsymbol{v}_p^{(k-1)}\right\|_2 = \max\left\{\|\boldsymbol{v}_k^{(k-1)}\|_2, \|\boldsymbol{v}_{k+1}^{(k-1)}\|_2, \cdots, \|\boldsymbol{v}_n^{(k-1)}\|_2\right\}. \tag{5.33}$$

若 $\|\boldsymbol{v}_p^{(k-1)}\|_2 = 0$, 停止计算. 否则取 $\boldsymbol{P}_k$ 为第 k 列与第 p 列交换的初等变换矩阵, 并确定一个 Householder 变换 $\widetilde{\boldsymbol{H}}_k \in \mathbb{R}^{(m-k+1)\times(m-k+1)}$, 使得

$$\widetilde{\boldsymbol{H}}_k \boldsymbol{v}_p^{(k-1)} = \gamma_{kk}\boldsymbol{e}_1.$$

令 $\boldsymbol{H}_k = \mathrm{diag}(\boldsymbol{I}_{k-1}, \widetilde{\boldsymbol{H}}_k)$, 则有

$$\boldsymbol{R}_k = \begin{array}{c} \\ \\ k \end{array} \begin{array}{c} k \quad\quad n-k \\ \left[\begin{array}{cc} \boldsymbol{R}_{11}^{(k)} & \boldsymbol{R}_{12}^{(k)} \\ \boldsymbol{O} & \boldsymbol{R}_{22}^{(k)} \end{array}\right] \\ m-k \end{array}, \tag{5.34}$$

式中: $\boldsymbol{R}_{11}^{(k)}$ 为 $k \times k$ 阶非奇异上三角矩阵.

这样, 从 $k=1$ 出发, 依次进行 $r\,(=\mathrm{rank}(\boldsymbol{A}))$ 次“列选主元”的 Householder 变换, 即可得到分解式 (5.30). 值得一提的是, 如果只是利用这一分解来求秩亏的最小二乘问题的解 (通常是用来求基本解), 那么正交矩阵 $\boldsymbol{Q} = \boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_r$ 没有必要显式地计算, 只需将计算过程中每一步的 Householder 变换同步地作用在向量 $\boldsymbol{b}$ 上即可.

此外, 为了减少“列选主元”的计算量, 不必每一步都按照 2–范数的定义去计算式 (5.33) 中的范数, 因为对于任何正交矩阵 $\boldsymbol{Q} \in \mathbb{R}^{l\times l}$ 均成立

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{x} = \begin{array}{c} \\ \left[\begin{array}{c} \alpha \\ \boldsymbol{z} \end{array}\right] \\ \\ \end{array} \begin{array}{c} 1 \\ l-1 \end{array} \Longrightarrow \|\boldsymbol{z}\|_2^2 = \|\boldsymbol{x}\|_2^2 - \alpha^2.$$

这样, 可以通过修正旧的范数来得到新的范数, 即

$$\|\boldsymbol{x}^{(j)}\|_2^2 = \|\boldsymbol{x}^{(j-1)}\|_2^2 - a_{kj}^2.$$

综上所述, 可写出计算分解式 (5.30) 的算法并用以求解秩亏最小二乘问题的详细步骤如下.

算法 5.3 (秩亏最小二乘问题的列主元 QR 分解法)

步 1, 输入矩阵 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{m\times n}$, $\boldsymbol{b}\in\mathbb{R}^m$. 计算

$$p(j):=j,\quad \sigma_j=\boldsymbol{A}(:,j)^{\mathrm{T}}\boldsymbol{A}(:,j),\quad j=1,2,\cdots,n.$$

置 $k:=1$.

步 2, 确定 r 使满足 $\sigma_r=\max\limits_{k\leqslant j\leqslant n}\sigma_j$. 若 $\sigma_r=0$, 停算.

步 3, 交换 $p(r)$ 与 $p(k)$, σ_r 与 σ_k 以及 $\boldsymbol{A}(:,r)$ 与 $\boldsymbol{A}(:,k)$.

步 4, 确定一个 Householder 矩阵 $\widetilde{\boldsymbol{H}}_k$ 使得

$$\widetilde{\boldsymbol{H}}_k\boldsymbol{A}(k:m,k)=\gamma_{kk}\boldsymbol{e}_1,$$

并计算

$$[\boldsymbol{A},\boldsymbol{b}]:=\mathrm{diag}(\boldsymbol{I}_{k-1},\widetilde{\boldsymbol{H}}_k)[\boldsymbol{A},\boldsymbol{b}],\quad \sigma_j:=\sigma_j-a_{kj}^2,\quad j=k+1,\cdots,n.$$

步 5, 置 $k:=k+1$, 转步 2.

算法 5.3 的运算量是 $2mnr-r^2(m+n)+2r^3/3$ 次乘除法, 其中 $r=\mathrm{rank}(\boldsymbol{A})$. 分解式 (5.30) 中的上三角矩阵 $\boldsymbol{R}$ 存储在 $\boldsymbol{A}$ 的上三角部分, 初等变换矩阵 $\boldsymbol{P}$ 由整数向量 $\boldsymbol{p}(1:n)$ 来记录, 即 $\boldsymbol{P}=\boldsymbol{P}_1\boldsymbol{P}_2\cdots\boldsymbol{P}_r$, 其中 $\boldsymbol{P}_k$ 是通过单位矩阵交换第 k 列与第 $p(k)$ 列而得到. 而第 k 个 Householder 向量的 $k+1:n$ 分量存储在 $\boldsymbol{A}(k+1:m,k)$ 中.

根据算法 5.3 编制 MATLAB 程序如下:

```
function [x,fval,P]=piv_house_qr(A,b)
%秩亏最小二乘问题的列主元QR分解法
[m,n]=size(A); %Q=eye(m);
A1=A; b1=b;
P=eye(n);
for j=1:n
    c(j)=A(1:m,j)'*A(1:m,j);
end
[cr,r]=max(c);
for k=1:n
    if (cr<=0), break; end
    c([k r])=c([r k]);
    P(:,[k r])=P(:,[r k]);
    A(1:m,[k r])= A(1:m,[r k]);
    [v,beta]=house(A(k:m,k));
    H=eye(m-k+1)-beta*v*v';
    A(k:m,k:n)=H*A(k:m,k:n);
    b(k:m)=H*b(k:m);
    for j=k+1:n
        c(j)=c(j)-A(k,j)^2;
    end
    [cr,r]=max(c(k+1:n));
```

```
        r=r+k;
    end
    for i=1:m
        if sum(abs(A(i,:)))<1.0e-10
            rank=i-1; break;
        end
    end
    R11=A(1:rank,1:rank); c=b(1:rank);
    x=P*[R11\c; zeros(rank,1)];
    fval=norm(A1*x-b1);
```

例 5.2 用 MATLAB 程序 piv_house_qr.m 求解超定方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的最小二乘解, 其中

$$\boldsymbol{A}=\begin{bmatrix}1&2&3&4\\1&4&5&6\\1&5&6&7\\1&8&9&10\\1&11&12&13\end{bmatrix},\quad \boldsymbol{x}=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}11\\13\\15\\18\\20\end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[1 2 3 4; 1 4 5 6; 1 5 6 7; 1 8 9 10; 1 11 12 13];
>> b=[11 13 15 18 20]';
>> [x,fval,P]=piv_house_qr(A,b)
```

即可得计算结果.

此外, 如果希望求出秩亏最小二乘问题的极小范数解, 则还需将分解式 (5.30) 中的 $\boldsymbol{R}_{12}$ 约化为零矩阵. 这可通过 r 次 Householder 变换来完成. 即确定 r 个 Householder 变换 $\boldsymbol{Z}_1,\cdots,\boldsymbol{Z}_r$ 和一个置换矩阵 $\widetilde{\boldsymbol{P}}$ 使得

$$[\boldsymbol{R}_{11},\boldsymbol{R}_{12}]\boldsymbol{Z}_1\cdots\boldsymbol{Z}_r\widetilde{\boldsymbol{P}}=[\boldsymbol{T},\boldsymbol{O}],$$

式中: $\boldsymbol{T}\in\mathbb{R}^{r\times r}$ 为上三角矩阵.

现令 $\boldsymbol{Z}=\boldsymbol{P}\boldsymbol{Z}_1\cdots\boldsymbol{Z}_r\widetilde{\boldsymbol{P}}$, 则有

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Z}=\begin{bmatrix}\boldsymbol{T}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}\end{bmatrix}\begin{matrix}r\\m-r\end{matrix},\qquad(5.35)$$

$$\begin{matrix}r & n-r\end{matrix}$$

由此, 得

$$\boldsymbol{x}_{\mathrm{LS}}=\boldsymbol{Z}\begin{bmatrix}\boldsymbol{T}^{-1}\boldsymbol{c}\\\boldsymbol{0}\end{bmatrix},$$

式中: $\boldsymbol{c}$ 为由 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}$ 前 r 个分量构成的 r 维向量.

5.3.2 奇异值分解法

设 $\boldsymbol{A}\in\mathbb{R}^{m\times n}\,(m>n)$ 的奇异值分解为

$$\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}},\quad \boldsymbol{\Sigma}=\begin{array}{c}\left[\begin{array}{cc}\boldsymbol{\Sigma}_r & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O}\end{array}\right]\begin{array}{c}r\\ m-r\end{array}\\ \begin{array}{cc}r & n-r\end{array}\end{array},\tag{5.36}$$

式中: $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_m]$ 和 $\boldsymbol{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n]$ 为正交矩阵; $\boldsymbol{\Sigma}_r=\mathrm{diag}(\sigma_1,\cdots,\sigma_r)$, $\sigma_1\geqslant\cdots\geqslant\sigma_r>0$. 则由定理 5.2 和定理 5.3 可知

$$\boldsymbol{x}_{\mathrm{LS}}=\boldsymbol{A}^{\dagger}\boldsymbol{b}=\boldsymbol{V}\left[\begin{array}{cc}\boldsymbol{\Sigma}_r^{-1} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O}\end{array}\right]\boldsymbol{U}^{\mathrm{T}}=\sum_{i=1}^{r}\frac{\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{b}}{\sigma_i}\boldsymbol{v}_i.\tag{5.37}$$

因此, 一旦求出分解式 (5.36), 就可由式 (5.37) 容易地求出最小二乘问题 (5.1) 的极小范数解 $\boldsymbol{x}_{\mathrm{LS}}$.

例 5.3 用 MATLAB 系统自带的奇异值分解函数 svd.m 求解超定方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 其中

$$\boldsymbol{A}=\left[\begin{array}{cccc}1&2&3&4\\1&4&5&6\\1&5&6&7\\1&8&9&10\\1&11&12&13\end{array}\right],\quad \boldsymbol{x}=\left[\begin{array}{c}x_1\\x_2\\x_3\\x_4\end{array}\right],\quad \boldsymbol{b}=\left[\begin{array}{c}11\\13\\15\\18\\20\end{array}\right].$$

解 编制 MATLAB 程序, 并存成文件名 ex53.m:

```
%例5.3 ex53.m
A=[1 2 3 4; 1 4 5 6; 1 5 6 7; 1 8 9 10; 1 11 12 13];
b=[11 13 15 18 20]';
[m,n]=size(A);
[U,S,V]=svd(A);
% x=(V*pinv(S)*U')*b;
for i=1:n
    if abs(S(i,i))<1.0e-6
        r=i-1; break;
    end
end    %r=rank(S);
x=zeros(n,1);
for i=1:r
    x=x+(U(:,i)'*b/S(i,i))*V(:,i);
end
x
fval=norm(A*x-b)
```

然后在命令窗口输入 ex53 即可得计算结果.

观察到求出的最小二乘解与例 5.2 不一样. 原因是本例求出的是极小范数解, 而例 5.2 只是一个基解. 但它们对应的最优值是相同的.

5.4 求解最小二乘问题的迭代方法

对于某些大型稀疏的最小二乘问题, 前面介绍的直接解法不一定有效. 此时, 迭代法应该是最好的选择. 本节讨论大型稀疏最小二乘问题

$$\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2=\min_{\boldsymbol{z}\in\mathbb{R}^n}\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{b}\|_2 \tag{5.38}$$

解的迭代算法, 包括矩阵分裂迭代法和 Krylov 子空间迭代法. 本节假定矩阵 $\boldsymbol{A}$ 是 $m\times n\,(m\geqslant n)$ 阶的列满秩实矩阵, $\boldsymbol{b}$ 是 m 维实向量, 即 $\boldsymbol{A}\in\mathbb{R}_n^{m\times n}$, $\boldsymbol{b}\in\mathbb{R}^m$. 于是, 最小二乘问题 (5.38) 有唯一的最小二乘解 $\boldsymbol{x}_{\mathrm{LS}}$. 这里所考虑的迭代算法基于最小二乘问题的法方程

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}, \tag{5.39}$$

或基于其 KKT 方程

$$\begin{bmatrix}\boldsymbol{I}_m & \boldsymbol{A}\\ \boldsymbol{A}^{\mathrm{T}} & \boldsymbol{O}\end{bmatrix}\begin{bmatrix}\boldsymbol{r}\\ \boldsymbol{x}\end{bmatrix}=\begin{bmatrix}\boldsymbol{b}\\ 0\end{bmatrix}. \tag{5.40}$$

5.4.1 基于法方程的矩阵分裂迭代法

由于 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ 是列满秩的, 因此, 原则上讲, 任何适用于求解对称正定方程组的迭代方法都可用于求解最小二乘问题的法方程 (5.39), 从而得到相应的求解最小二乘问题 (5.38) 的迭代算法.

为了方便起见, 将 $\boldsymbol{A}$ 按列分块为 $\boldsymbol{A}=[\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n]$, 并记 $d_i=\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{a}_i$, $i=1,2,\cdots,n$. $\boldsymbol{D}=\mathrm{diag}(d_1,d_2,\cdots,d_n)$. 于是可将矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 分裂为

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{D}-\boldsymbol{L}-\boldsymbol{L}^{\mathrm{T}}, \tag{5.41}$$

这里 $-\boldsymbol{L}$ 是其严格下三角部分.

1. Jacobi 迭代法

在分裂 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{N}$ 中, 令 $\boldsymbol{M}=\boldsymbol{D}$, $\boldsymbol{N}=\boldsymbol{L}+\boldsymbol{L}^{\mathrm{T}}$, 则得到 Jacobi 迭代格式为

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{D}^{-1}\big[(\boldsymbol{D}-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{x}^{(k)}+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}\big],\quad k=0,1,\cdots, \tag{5.42}$$

或等价地, 有

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\boldsymbol{D}^{-1}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k)}),\quad k=0,1,\cdots. \tag{5.43}$$

显然, Jacobi 迭代法的迭代矩阵为 $\boldsymbol{G}_J=\boldsymbol{I}-\boldsymbol{D}^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$, 且迭代过程不需要显式地计算 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$.

在实际计算中, 可以对每个分量进行迭代. 令 $\boldsymbol{r}^{(k)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k)}$, 执行下面的算法:

$\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)}$; $k = 0$;
while $(\|\boldsymbol{r}^{(k)}\|_2/\|\boldsymbol{r}^{(0)}\|_2 \leqslant \varepsilon)$
  for $i = 1 : n$
    $d_i = \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{a}_i$;
    $x_i^{(k+1)} = x_i^{(k)} + \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{r}^{(k)}/d_i$;
  end
  $\boldsymbol{r}^{(k+1)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k+1)}$;
  $k = k + 1$;
end

Jacobi 迭代法的 MATLAB 程序如下:

```
function [x,fval,k,time]=nls_jacobi(A,b,x,tol,max_it)
%Jacobi算法求解法方程A'Ax=A'b
tic; n=size(A,2); r=b-A*x;
d=zeros(n,1); nr=norm(A'*r); k=0;
for i=1:n
    d(i)=A(:,i)'*A(:,i);
end
while (k<=max_it)
    k=k+1;
    for i=1:n
       x(i)=x(i)+A(:,i)'*r/d(i);
    end
    r=b-A*x; s=A'*r;
    if (norm(s)/nr<tol)
        break;
    end
end
fval=norm(r); time=toc;
```

例 5.4 利用 Jacobi 迭代法的 MATLAB 程序, 计算最小二乘问题 $\min \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2$ 的极小解, 其中

$$\boldsymbol{A} = \begin{bmatrix} 4 & -1 & 0 & 0 \\ 1 & 4 & -1 & 0 \\ 0 & 1 & 4 & -1 \\ 0 & 0 & 1 & 4 \\ 1 & 3 & 2 & 1 \\ 0 & 2 & 0 & 3 \\ 8 & 2 & 3 & 1 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 9 \\ 12 \\ 11 \\ 13 \\ 17 \\ 15 \\ 19 \end{bmatrix}.$$

取初始点为零向量, 容许误差为 10^{-10}.

解 编写 MATLAB 脚本文件 ex54.m, 并在命令窗口运行之, 迭代 708 次, 得到极小解和极小值

$$\boldsymbol{x}^* = (1.1019, 2.7905, 1.9907, 2.6509)^{\mathrm{T}}, \quad \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^*\|_2 = 9.2279.$$

2. Gauss–Seidel 迭代法

在分裂 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ 中, 令 $\boldsymbol{M} = \boldsymbol{D} - \boldsymbol{L}$, $\boldsymbol{N} = \boldsymbol{L}^{\mathrm{T}}$, 则得到 Gauss–Seidel 迭代格式为

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{D}^{-1}\big[\boldsymbol{L}\boldsymbol{x}^{(k+1)} + (\boldsymbol{D} - \boldsymbol{L} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{x}^{(k)} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}\big], \quad k = 0, 1, \cdots, \tag{5.44}$$

或等价地, 有

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}), \quad k = 0, 1, \cdots. \tag{5.45}$$

不难发现, Gauss–Seidel 迭代法的迭代矩阵为 $\boldsymbol{G}_S = (\boldsymbol{D} - \boldsymbol{L})^{-1}(\boldsymbol{D} - \boldsymbol{L} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}) = (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{L}^{\mathrm{T}}$, 并需要显式地计算 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$.

为了避免 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的显式计算, 可在第 k 步引入辅助变量 $\boldsymbol{z}_1 = \boldsymbol{x}^{(k)}$, $\boldsymbol{r}_1 = \boldsymbol{r}^{(k)}$, 则式 (5.44) 可化为

$$\begin{aligned}\boldsymbol{x}^{(k+1)} &= \boldsymbol{D}^{-1}\big[\boldsymbol{L}\boldsymbol{x}^{(k+1)} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} + (\boldsymbol{D} - \boldsymbol{L} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{x}^{(k)}\big] \\ &= \boldsymbol{x}^{(k)} + \boldsymbol{D}^{-1}\big[\boldsymbol{L}(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) + \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)})\big] \\ &= \boldsymbol{z}_1 + \boldsymbol{D}^{-1}\big[\boldsymbol{L}(\boldsymbol{x}^{(k+1)} - \boldsymbol{z}_1) + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_1\big].\end{aligned} \tag{5.46}$$

详细的算法步骤如下:

$\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)}$; $k = 0$;
while $(\|\boldsymbol{r}^{(k)}\|_2/\|\boldsymbol{r}^{(0)}\|_2 \leqslant \varepsilon)$
  $\boldsymbol{r}_1 = \boldsymbol{r}^{(k)}$; $\boldsymbol{z}_1 = \boldsymbol{x}^{(k)}$;
  for $i = 1 : n$
    $d_i = \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{a}_i$; $\alpha_i = \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{r}_i/d_i$;
    $\boldsymbol{z}_{i+1} = \boldsymbol{z}_i + \alpha_i\boldsymbol{e}^{(i)}$;
    $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i\boldsymbol{a}_i$;
  end
  $\boldsymbol{x}^{(k+1)} = \boldsymbol{z}_{n+1}$;
  $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}_{n+1}$;
  $k = k + 1$;
end

以上算法不再需要显式地计算 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$.

Gauss–Seidel 迭代法的 MATLAB 程序如下:

```
function [x,fval,k,time]=nls_seidel(A,b,x,tol,max_it)
%Gauss--Seidel算法求解法方程A'Ax=A'b
```

```
tic;n=size(A,2);
r=b-A*x;d=zeros(n,1);
nr=norm(A'*r);k=0;E=eye(n);
for (i=1:n
    d(i)=A(:,i)'*A(:,i);
end
D=diag(d); Z=[x]; R=[r];
while (k<=max_it)
    k=k+1;
    for i=1:n
        delta=A(:,i)'*R(:,i)/d(i);
        z=Z(:,i)+delta*E(:,i);
        r=R(:,i)-delta*A(:,i);
        Z=[Z,z]; R=[R,r];
    end
    x=Z(:,n+1);r=b-A*x;
    Z=[x];R=[r];
    if(norm(A'*r)/nr<tol),break;end
end
fval=norm(r); time=toc;
```

例 5.5 利用 Gauss–Seidel 迭代法的 MATLAB 程序, 计算例 5.4 中的最小二乘问题 $\min \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2$ 的极小解. 取初始点为零向量, 容许误差为 10^{-10}.

解 编写 MATLAB 脚本文件 ex55.m, 并在命令窗口运行之, 迭代 19 次, 得到极小解和极小值

$$\boldsymbol{x}^* = (1.1019, 2.7905, 1.9907, 2.6509)^{\mathrm{T}}, \quad \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^*\|_2 = 9.2279.$$

3. SOR 迭代法

在分裂 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ 中, 令

$$\boldsymbol{M} = \frac{1}{\omega}(\boldsymbol{D} - \omega\boldsymbol{L}), \ \boldsymbol{N} = \frac{1}{\omega}\big[(1-\omega)\boldsymbol{D} + \omega\boldsymbol{L}^{\mathrm{T}})\big], \ \ \omega \neq 0,$$

则得到 SOR 迭代格式为

$$\boldsymbol{x}^{(k+1)} = \omega\boldsymbol{D}^{-1}\big(\boldsymbol{L}\boldsymbol{x}^{(k+1)} + \boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}^{(k)} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}\big) + (1-\omega)\boldsymbol{x}^{(k)}, \ \ k = 0, 1, \cdots, \tag{5.47}$$

式中: ω 为松弛因子. 或等价地, 有

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \omega\boldsymbol{D}^{-1}\big[\boldsymbol{L}(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) + \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)})\big], \ \ k = 0, 1, \cdots. \tag{5.48}$$

上述迭代法也需要显式地计算 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$, 其迭代矩阵为 $\boldsymbol{G}_\omega = (\boldsymbol{D}-\omega\boldsymbol{L})^{-1}\big[(1-\omega)\boldsymbol{D}+\omega\boldsymbol{L}^{\mathrm{T}}\big)$.

同样, 为了避免显式计算 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$, 可在第 k 步引入辅助变量 $\boldsymbol{z}_1=\boldsymbol{x}^{(k)}$, $\boldsymbol{r}_1=\boldsymbol{r}^{(k)}$, 则式 (5.47) 可化为

$$\begin{aligned}\boldsymbol{x}^{(k+1)}&=\boldsymbol{x}^{(k)}+\omega\boldsymbol{D}^{-1}\big[\boldsymbol{L}(\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)})+\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k)})\big]\\&=\boldsymbol{z}_1+\omega\boldsymbol{D}^{-1}\big[\boldsymbol{L}(\boldsymbol{x}^{(k+1)}-\boldsymbol{z}_1)+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}_1\big].\end{aligned}\tag{5.49}$$

详细的算法步骤如下:

$\boldsymbol{r}^{(0)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(0)}$; $k=0$;
while $(\|\boldsymbol{r}^{(k)}\|_2/\|\boldsymbol{r}^{(0)}\|_2\leqslant\varepsilon)$
  $\boldsymbol{r}_1=\boldsymbol{r}^{(k)}$; $\boldsymbol{z}_1=\boldsymbol{x}^{(k)}$;
  for $i=1:n$
    $d_i=\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{a}_i$; $\alpha_i=\omega\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{r}_i/d_i$;
    $\boldsymbol{z}_{i+1}=\boldsymbol{z}_i+\alpha_i\boldsymbol{e}^{(i)}$;
    $\boldsymbol{r}_{i+1}=\boldsymbol{r}_i-\alpha_i\boldsymbol{a}_i$;
  end
  $\boldsymbol{x}^{(k+1)}=\boldsymbol{z}_{n+1}$, $\boldsymbol{r}^{(k+1)}=\boldsymbol{r}_{n+1}$;
  $k=k+1$;
end

注 5.4 关于上述三种迭代法的收敛性定理, 跟第 3 章中求解线性方程组的 Jacobi 迭代法、Gauss-Seidel 迭代法和 SOR 迭代法相类似, 只需将此处的 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 和 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ 分别视作方程 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 中的 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 即可得到类似的结论.

SOR 迭代法的 MATLAB 程序如下:

```
function [x,fval,k,time]=nls_sor(A,b,w,x,tol,max_it)
%SOR迭代法求解法方程A'Ax=A'b
tic;n=size(A,2);r=b-A*x;d=zeros(n,1);
nr=norm(A'*r);k=0; E=eye(n);
for(i=1:n),d(i)=A(:,i)'*A(:,i);end
D=diag(d);Z=[x];R=[r];
while (k<=max_it)
    k=k+1;
    for i=1:n
        delta=w*A(:,i)'*R(:,i)/d(i);
        z=Z(:,i)+delta*E(:,i);
        r=R(:,i)-delta*A(:,i);
        Z=[Z,z]; R=[R,r];
    end
    x=Z(:,n+1);r=b-A*x;Z=[x];R=[r];
    if(norm(A'*r)/nr<tol),break;end
end
fval=norm(r);time=toc;
```

例 5.6 利用 SOR 迭代法的 MATLAB 程序, 计算最小二乘问题 $\min \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2$ 的极小解, 其中

$$\boldsymbol{A}=\begin{bmatrix} 1 & 23.73 & 5.49 & 1.21 \\ 1 & 22.34 & 4.32 & 1.35 \\ 1 & 28.84 & 5.04 & 1.92 \\ 1 & 27.67 & 4.72 & 1.49 \\ 1 & 20.83 & 5.35 & 1.56 \\ 1 & 22.27 & 4.27 & 1.50 \\ 1 & 27.57 & 5.25 & 1.85 \\ 1 & 28.01 & 4.62 & 1.51 \end{bmatrix}, \quad \boldsymbol{x}=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad \boldsymbol{b}=\begin{bmatrix} 15.02 \\ 12.62 \\ 14.86 \\ 13.98 \\ 15.91 \\ 12.47 \\ 15.80 \\ 14.32 \end{bmatrix}.$$

取初始点为零向量, 松弛因子 $\omega=1.06$, 容许误差为 10^{-6}.

解 编写 MATLAB 脚本文件 ex56.m, 并在命令窗口运行之, 迭代 1134 次, 得到极小解和极小值

$$\boldsymbol{x}^*=(-0.0054, 0.0169, 2.4468, 1.2954)^{\mathrm{T}}, \quad \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^*\|_2=0.9959.$$

5.4.2 基于法方程的共轭梯度法

由于 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ $(m\geqslant n)$ 列满秩, 故最小二乘问题的法方程 (5.39) 是对称正定方程组. 因此, 法方程 (5.39) 等价于极小化问题

$$\min f(\boldsymbol{x})=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{x}-(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b})^{\mathrm{T}}\boldsymbol{x}. \tag{5.50}$$

事实上, $f(\boldsymbol{x})$ 的梯度为

$$\nabla f(\boldsymbol{x})=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b},$$

且对任意给定的非零向量 $\boldsymbol{p}$ 和实数 α, 有

$$f(\boldsymbol{x}+\alpha\boldsymbol{p})=f(\boldsymbol{x})+\alpha\nabla f(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{p}+\frac{1}{2}\alpha^2\boldsymbol{p}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}.$$

若 $\boldsymbol{x}^*$ 是法方程 (5.39) 的解, 则有 $\nabla f(\boldsymbol{x}^*)=\boldsymbol{0}$. 因此, 对任意非零向量 $\boldsymbol{p}\in\mathbb{R}^n$, 有

$$f(\boldsymbol{x}^*+\alpha\boldsymbol{p})\begin{cases} >f(\boldsymbol{x}^*), & \alpha\neq 0, \\ =f(\boldsymbol{x}^*), & \alpha=0. \end{cases}$$

故 $\boldsymbol{x}^*$ 是 $f(\boldsymbol{x})$ 的极小点. 反之, 因 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 对称正定, 可知 $f(\boldsymbol{x})$ 在 $\mathbb{R}^n$ 中有唯一的极小点. 若 $\boldsymbol{x}^*$ 是 $f(\boldsymbol{x})$ 的极小点, 则必有 $\nabla f(\boldsymbol{x}^*)=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}^*-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}=\boldsymbol{0}$, 即 $\boldsymbol{x}^*$ 是法方程 (5.39) 的解.

类似于用共轭梯度法求解对称正定方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的思想, 设 $\boldsymbol{x}^{(0)}\in\mathbb{R}^n$ 是任意给定的一个初始向量, 对于 $k=0,1,2,\cdots$, 从 $\boldsymbol{x}^{(k)}$ 出发, 沿方向 $\boldsymbol{p}^{(k)}$ 求函数 $f(\boldsymbol{x})$ 在直线 $\boldsymbol{x}=\boldsymbol{x}^{(k)}+\alpha\boldsymbol{p}^{(k)}$ 上的极小点, 得

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\alpha_k\boldsymbol{p}^{(k)},\quad \boldsymbol{r}^{(k)}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k)}, \tag{5.51}$$

$$\boldsymbol{s}^{(k)}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}^{(k)},\quad \alpha_k=\frac{(\boldsymbol{s}^{(k)},\boldsymbol{p}^{(k)})}{(\boldsymbol{p}^{(k)},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)})}, \tag{5.52}$$

称向量 $\boldsymbol{p}^{(k)}$ 为*搜索方向*. 若向量组 $\boldsymbol{p}^{(0)},\boldsymbol{p}^{(1)},\cdots,\boldsymbol{p}^{(k-1)}$ 满足

$$\left(\boldsymbol{p}^{(i)},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(j)}\right)=0,\ i\neq j, \tag{5.53}$$

且 $\boldsymbol{p}^{(k)}\neq\boldsymbol{0}$, $k=0,1,\cdots,n-1$, 则称向量组 $\{\boldsymbol{p}^{(k)}\}$ 为 $\mathbb{R}^n$ 中关于 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的一个共轭向量组, 称迭代法 (5.51) 为*共轭方向法*. 特别地, 若取 $\boldsymbol{p}^{(0)}=\boldsymbol{s}^{(0)}$,

$$\boldsymbol{p}^{(k+1)}=\boldsymbol{s}^{(k+1)}+\beta_k\boldsymbol{p}^{(k)},\quad \beta_k=-\frac{(\boldsymbol{s}^{(k+1)},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)})}{(\boldsymbol{p}^{(k)},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)})}, \tag{5.54}$$

则称为*共轭梯度法*.

由式 (5.51)、式 (5.52) 及式 (5.54) 可知, 若存在 $k\geqslant 0$, 使得 $\boldsymbol{s}^{(k)}=\boldsymbol{0}$, 则 $\boldsymbol{x}^{(k)}$ 为最小二乘问题的解, 且有 $\alpha_k=\beta_k=0$, $\boldsymbol{s}^{(k+1)}=\boldsymbol{p}^{(k+1)}=\boldsymbol{0}$. 此外, 在上述的共轭梯度法中每次迭代需要计算 4 次矩阵与向量的乘法, 因此有必要对 α_k 和 β_k 的计算公式进行化简.

定理 5.6 *在共轭梯度法 (5.51)、(5.52) 和 (5.54) 中, 若 $k>0$, $\boldsymbol{s}^{(k)}\neq\boldsymbol{0}$, 则有*

$$\begin{cases}(\boldsymbol{s}^{(k)},\boldsymbol{s}^{(i)})=(\boldsymbol{s}^{(k)},\boldsymbol{p}^{(i)})=(\boldsymbol{p}^{(k)},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(i)})=0,\ 0\leqslant i<k,\\ (\boldsymbol{p}^{(k)},\boldsymbol{s}^{(i)})=(\boldsymbol{s}^{(k)},\boldsymbol{s}^{(k)}),\ 0\leqslant i\leqslant k.\end{cases} \tag{5.55}$$

证明 由定理的条件, 有 $\alpha_i\neq 0$, $i=0,1,\cdots,k$. 根据迭代公式, 得

$$\begin{cases}\boldsymbol{s}^{(k+1)}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}^{(k+1)}=\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^{(k+1)})=\boldsymbol{s}^{(k)}-\alpha_k\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)},\\ \boldsymbol{p}^{(k+1)}=\boldsymbol{s}^{(k+1)}+\beta_k\boldsymbol{p}^{(k)}=\boldsymbol{s}^{(k)}-\alpha_k\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)}+\beta_k\boldsymbol{p}^{(k)}.\end{cases} \tag{5.56}$$

下面用归纳法证明定理的结论. 注意到

$$\begin{aligned}(\boldsymbol{s}^{(1)},\boldsymbol{s}^{(0)})&=(\boldsymbol{s}^{(1)},\boldsymbol{p}^{(0)})=(\boldsymbol{s}^{(0)},\boldsymbol{p}^{(0)})-\alpha_0(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(0)},\boldsymbol{p}^{(0)})=0,\\ (\boldsymbol{p}^{(1)},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(0)})&=(\boldsymbol{s}^{(1)},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(0)})+\beta_0(\boldsymbol{p}^{(0)},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(0)})=0,\\ (\boldsymbol{p}^{(1)},\boldsymbol{s}^{(0)})&=(\boldsymbol{p}^{(1)},\boldsymbol{s}^{(1)}+\alpha_0\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(0)})=(\boldsymbol{p}^{(1)},\boldsymbol{s}^{(1)})\\ &=(\boldsymbol{s}^{(1)}+\beta_0\boldsymbol{p}^{(0)},\boldsymbol{s}^{(1)})=(\boldsymbol{s}^{(1)},\boldsymbol{s}^{(1)}),\end{aligned}$$

即 $k=1$ 时结论成立. 现假定一直到 $k\,(>1)$ 时, 结论成立. 则

$$(\boldsymbol{s}^{(k+1)},\boldsymbol{p}^{(k)})=(\boldsymbol{s}^{(k)},\boldsymbol{p}^{(k)})-\alpha_k(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)},\boldsymbol{p}^{(k)})=0,$$

$$
\begin{aligned}
(\boldsymbol{p}^{(k+1)}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)}) &= (\boldsymbol{s}^{(k+1)}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)}) + \beta_k(\boldsymbol{p}^{(k)}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)}) = 0,\\
(\boldsymbol{s}^{(k+1)}, \boldsymbol{s}^{(k)}) &= (\boldsymbol{s}^{(k+1)}, \boldsymbol{p}^{(k)} - \beta_{k-1}\boldsymbol{p}^{(k-1)}) = -\beta_{k-1}(\boldsymbol{s}^{(k+1)}, \boldsymbol{p}^{(k-1)})\\
&= -\beta_{k-1}(\boldsymbol{s}^{(k)} - \alpha_k\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)}, \boldsymbol{p}^{(k-1)}) = 0.
\end{aligned}
$$

当 $i < k$ 时, 有

$$
\begin{aligned}
(\boldsymbol{s}^{(k+1)}, \boldsymbol{p}^{(i)}) &= (\boldsymbol{s}^{(k)}, \boldsymbol{p}^{(i)}) - \alpha_k(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)}, \boldsymbol{p}^{(i)}) = 0,\\
(\boldsymbol{p}^{(k+1)}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(i)}) &= (\boldsymbol{s}^{(k+1)}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(i)}) + \beta_k(\boldsymbol{p}^{(k)}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(i)})\\
&= (\boldsymbol{s}^{(k+1)}, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(i)}) = (\boldsymbol{s}^{(k+1)}, (\boldsymbol{s}^{(i)} - \boldsymbol{s}^{(i+1)})/\alpha_i) = 0,\\
(\boldsymbol{s}^{(k+1)}, \boldsymbol{s}^{(i)}) &= (\boldsymbol{s}^{(k+1)}, \boldsymbol{p}^{(i)} - \beta_{i-1}\boldsymbol{p}^{(i-1)}) = -\beta_{i-1}(\boldsymbol{s}^{(k+1)}, \boldsymbol{p}^{(i-1)})\\
&= -\beta_{i-1}(\boldsymbol{s}^{(k)} - \alpha_k\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)}, \boldsymbol{p}^{(i-1)}) = 0.
\end{aligned}
$$

又当 $i \leqslant k+1$ 时, 有

$$
\begin{aligned}
(\boldsymbol{p}^{(k+1)}, \boldsymbol{s}^{(i)}) &= \Big(\boldsymbol{p}^{(k+1)}, \boldsymbol{s}^{(k+1)} + \sum_{l=i}^{k}\alpha_l\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(l)}\Big) = (\boldsymbol{p}^{(k+1)}, \boldsymbol{s}^{(k+1)})\\
&= (\boldsymbol{s}^{(k+1)} + \beta_k\boldsymbol{p}^{(k)}, \boldsymbol{s}^{(k+1)}) = (\boldsymbol{s}^{(k+1)}, \boldsymbol{s}^{(k+1)}).
\end{aligned}
$$

由归纳法原理, 定理得证. □

利用式 (5.56) 的第 1 式, 当 $\alpha_k \neq 0$ 时, 有

$$
\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{p}^{(k)} = \frac{\boldsymbol{s}^{(k)} - \boldsymbol{s}^{(k+1)}}{\alpha_k},
$$

代入 α_k 和 β_k 的表达式并化简, 得

$$
\alpha_k = \frac{\|\boldsymbol{s}^{(k)}\|_2^2}{\|\boldsymbol{A}\boldsymbol{p}^{(k)}\|_2^2}, \quad \beta_k = \frac{\|\boldsymbol{s}^{(k+1)}\|_2^2}{\|\boldsymbol{s}^{(k)}\|_2^2}. \tag{5.57}
$$

算法 5.4 (CGLS) 给定矩阵 $\boldsymbol{A} \in \mathbb{R}_n^{m\times n}$, 向量 $\boldsymbol{b} \in \mathbb{R}^m$ 和容许误差 $\varepsilon > 0$. 本算法计算向量 $\boldsymbol{x}^{(k)}$, 使得 $\|\boldsymbol{s}^{(k)}\|_2/\|\boldsymbol{s}^{(0)}\|_2 \leqslant \varepsilon$, 其中 $\boldsymbol{r}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}$, $\boldsymbol{s}^{(k)} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}^{(k)}$.

取初始向量 $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$;

计算 $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)}$; $\boldsymbol{p}^{(0)} = \boldsymbol{s}^{(0)} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}^{(0)}$; $\gamma_0 = \|\boldsymbol{s}^{(0)}\|_2^2$; $k = 0$;

while $(\|\boldsymbol{s}^{(k)}\|_2/\|\boldsymbol{s}^{(0)}\|_2 \leqslant \varepsilon)$

$\boldsymbol{q}^{(k)} = \boldsymbol{A}\boldsymbol{p}^{(k)}$; $\alpha_k = \gamma_k/\|\boldsymbol{q}^{(k)}\|_2^2$;

$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k\boldsymbol{p}^{(k)}$;

$\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k\boldsymbol{q}^{(k)}$; $\boldsymbol{s}^{(k+1)} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}^{(k+1)}$;

$\gamma_{k+1} = \|\boldsymbol{s}^{(k+1)}\|_2^2$; $\beta_k = \gamma_{k+1}/\gamma_k$;

$\boldsymbol{p}^{(k+1)} = \boldsymbol{s}^{(k+1)} + \beta_k\boldsymbol{p}^{(k)}$;

$k = k + 1$;

end

算法 5.4 的有限终止性、收敛性和收敛速度的分析与求解对称正定方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的共轭梯度法相类似, 此处不再赘述. 此外, 不难发现算法 5.4 的每次迭代已经减少到只需计算两次矩阵与向量的乘法.

算法 5.4 的 MATLAB 程序如下:

```
function [x,fval,k,time]=nls_cg(A,b,x,tol,max_it)
%CG算法求解法方程A'Ax=A'b
tic; n=size(A,2); r=b-A*x;
s=A'*r; p=s; gama=s'*s;
nr=norm(s); k=0;
while (k<=max_it)
    k=k+1; q=A*p; alpha=gama/(q'*q);
    x=x+alpha*p; r=r-alpha*q; s=A'*r;
    if (norm(s)/nr<tol), break; end
    gama1=s'*s; beta=gama1/gama;
    p=s+beta*p; gama=gama1;
end
fval=norm(r); time=toc;
```

例 5.7 利用 CG 迭代法的 MATLAB 程序, 计算例 5.6 中的最小二乘问题 $\min\|\boldsymbol{b}-\boldsymbol{Ax}\|_2$ 的极小解, 并与 SOR 迭代法进行比较 (取初始点为零向量, 松弛因子 $\omega=1.06$, 容许误差 $\varepsilon=10^{-10}$).

解 编写 MATLAB 脚本文件 ex57.m, 并在命令窗口运行之, 得到数值结果如表 5.1 所示.

表 5.1 CG 法与 SOR 法的数值结果

算法	迭代次数	CPU时间	极小解	极小值
SOR	2889	0.2275	$(-0.0309, 0.0171, 2.4509, 1.2954)^{\mathrm{T}}$	0.9959
CG	5	0.0031	$(-0.0309, 0.0171, 2.4509, 1.2954)^{\mathrm{T}}$	0.9959

5.4.3 基于 KKT 方程的 SOR 类迭代法

由于 SOR 迭代法要求方程组系数矩阵的对角元均不为零, 故不能直接将 SOR 迭代法用于求解最小二乘问题 (5.38) 的 KKT 方程 (5.40), 但可以通过适当的等价变形使之满足 SOR 迭代法的条件.

事实上, KKT 方程 (5.40) 等价于

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{I}_m \\ \boldsymbol{O} & \boldsymbol{A}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{r} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}. \tag{5.58}$$

由于 $\boldsymbol{A}\in\mathbb{R}_n^{m\times n}$, 不失一般性, 可将 $\boldsymbol{A}$ 进行如下分块:

$$\boldsymbol{A}=\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \begin{matrix} n \\ m-n \end{matrix}, \tag{5.59}$$

式中: $\boldsymbol{A}_1 \in \mathbb{R}^{n\times n}$ 为非奇异阵. 再将 $\boldsymbol{r}$ 和 $\boldsymbol{b}$ 进行相应的分块, 即

$$\boldsymbol{r} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix},$$

式中: $\boldsymbol{v}, \boldsymbol{b}_1 \in \mathbb{R}^n$, $\boldsymbol{w}, \boldsymbol{b}_2 \in \mathbb{R}^{m-n}$. 则式 (5.58) 变成

$$\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{I}_n & \boldsymbol{O} \\ \boldsymbol{A}_2 & \boldsymbol{O} & \boldsymbol{I}_{m-n} \\ \boldsymbol{O} & \boldsymbol{A}_1^{\mathrm{T}} & \boldsymbol{A}_2^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{0} \end{bmatrix}. \tag{5.60}$$

式 (5.60) 可进一步等价变形为

$$\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{O} & \boldsymbol{I}_n \\ \boldsymbol{A}_2 & \boldsymbol{I}_{m-n} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_2^{\mathrm{T}} & \boldsymbol{A}_1^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{w} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{0} \end{bmatrix}. \tag{5.61}$$

记

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{O} & \boldsymbol{I}_n \\ \boldsymbol{A}_2 & \boldsymbol{I}_{m-n} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_2^{\mathrm{T}} & \boldsymbol{A}_1^{\mathrm{T}} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{w} \\ \boldsymbol{v} \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{0} \end{bmatrix},$$

则方程组 (5.61) 可写成

$$\boldsymbol{B}\boldsymbol{y} = \boldsymbol{c}. \tag{5.62}$$

由于 $\boldsymbol{A}_1$ 是非奇异的, 不难验证矩阵 $\boldsymbol{B}$ 也是非奇异的.

现按照矩阵 $\boldsymbol{B}$ 的自然分块, 有

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{m-n} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_1^{\mathrm{T}} \end{bmatrix}, \ \boldsymbol{L} = -\begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{A}_2 & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_2^{\mathrm{T}} & \boldsymbol{O} \end{bmatrix}, \ \boldsymbol{U} = -\begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_n \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}. \tag{5.63}$$

以及

$$\mathcal{L} = \boldsymbol{D}^{-1}\boldsymbol{L} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ -\boldsymbol{A}_2 & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{A}_1^{-\mathrm{T}}\boldsymbol{A}_2^{\mathrm{T}} & \boldsymbol{O} \end{bmatrix}, \ \mathcal{U} = \boldsymbol{D}^{-1}\boldsymbol{U} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{A}_1^{-1} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}. \tag{5.64}$$

由此, 可得以下结论:

(1) Jacobi 迭代格式为

$$\boldsymbol{y}^{(k+1)} = \mathcal{G}_J\boldsymbol{y}^{(k)} + \boldsymbol{f}_J, \tag{5.65}$$

式中: 迭代矩阵 $\mathcal{G}_J$ 和右端向量 $\boldsymbol{f}_J$ 分别为

$$\mathcal{G}_J=\mathcal{L}+\mathcal{U}=\begin{bmatrix}\boldsymbol{O}&\boldsymbol{O}&-\boldsymbol{A}_1^{-1}\\-\boldsymbol{A}_2&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&-(\boldsymbol{A}_2\boldsymbol{A}_1^{-1})^{\mathrm{T}}&\boldsymbol{O}\end{bmatrix},\quad \boldsymbol{f}_J=D^{-1}\boldsymbol{c}=\begin{bmatrix}\boldsymbol{A}_1^{-1}\boldsymbol{b}_1\\\boldsymbol{b}_2\\\boldsymbol{0}\end{bmatrix}. \tag{5.66}$$

详细的迭代格式为

$$\begin{cases}\boldsymbol{x}^{(k+1)}=\boldsymbol{A}_1^{-1}(\boldsymbol{b}_1-\boldsymbol{v}^{(k)}),\\\boldsymbol{w}^{(k+1)}=\boldsymbol{b}_2-\boldsymbol{A}_2\boldsymbol{x}^{(k)},\qquad k=0,1,2,\cdots.\\\boldsymbol{v}^{(k+1)}=-\boldsymbol{A}_1^{-\mathrm{T}}\boldsymbol{A}_2^{\mathrm{T}}\boldsymbol{w}^{(k)},\end{cases} \tag{5.67}$$

此外, 若记 $\boldsymbol{A}_3=\boldsymbol{A}_2\boldsymbol{A}_1^{-1}$, 直接计算, 得

$$\begin{aligned}\mathcal{G}_J^3&=-\begin{bmatrix}\boldsymbol{A}_1^{-1}\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_2&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{A}_2\boldsymbol{A}_1^{-1}\boldsymbol{A}_3^{\mathrm{T}}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_2\boldsymbol{A}_1^{-1}\end{bmatrix}\\&=-\begin{bmatrix}\boldsymbol{A}_1^{-1}\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_3\boldsymbol{A}_1&\boldsymbol{O}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{A}_3\boldsymbol{A}_3^{\mathrm{T}}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_3\end{bmatrix}.\end{aligned} \tag{5.68}$$

(2) SOR 迭代格式为

$$\boldsymbol{y}^{(k+1)}=\mathcal{G}_\omega\boldsymbol{y}^{(k)}+\boldsymbol{f}_\omega, \tag{5.69}$$

式中: 迭代矩阵 $\mathcal{G}_\omega$ 和右端向量 $\boldsymbol{f}_\omega$ 分别为

$$\mathcal{G}_\omega=(\boldsymbol{I}-\omega\mathcal{L})^{-1}\big[(1-\omega)\boldsymbol{I}+\omega\mathcal{U}\big],\quad \boldsymbol{f}_\omega=\omega(\boldsymbol{I}-\omega\mathcal{L})^{-1}\boldsymbol{f}_J. \tag{5.70}$$

仍记 $\boldsymbol{A}_3=\boldsymbol{A}_2\boldsymbol{A}_1^{-1}$, 直接计算, 得

$$\begin{aligned}\mathcal{G}_\omega&=\begin{bmatrix}\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}\\\omega\boldsymbol{A}_2&\boldsymbol{I}&\boldsymbol{O}\\\boldsymbol{O}&\omega\boldsymbol{A}_3^{\mathrm{T}}&\boldsymbol{I}\end{bmatrix}^{-1}\begin{bmatrix}(1-\omega)\boldsymbol{I}&\boldsymbol{O}&-\omega\boldsymbol{A}_1^{-1}\\\boldsymbol{O}&(1-\omega)\boldsymbol{I}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&(1-\omega)\boldsymbol{I}\end{bmatrix}\\&=\begin{bmatrix}\boldsymbol{I}&\boldsymbol{O}&\boldsymbol{O}\\-\omega\boldsymbol{A}_2&\boldsymbol{I}&\boldsymbol{O}\\\omega^2\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_2&-\omega\boldsymbol{A}_3^{\mathrm{T}}&\boldsymbol{I}\end{bmatrix}\begin{bmatrix}(1-\omega)\boldsymbol{I}&\boldsymbol{O}&-\omega\boldsymbol{A}_1^{-1}\\\boldsymbol{O}&(1-\omega)\boldsymbol{I}&\boldsymbol{O}\\\boldsymbol{O}&\boldsymbol{O}&(1-\omega)\boldsymbol{I}\end{bmatrix}\\&=\begin{bmatrix}(1-\omega)\boldsymbol{I}&\boldsymbol{O}&-\omega\boldsymbol{A}_1^{-1}\\-\omega(1-\omega)\boldsymbol{A}_2&(1-\omega)\boldsymbol{I}&\omega^2\boldsymbol{A}_3\\\omega(1-\omega)^2\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_2&-\omega(1-\omega)\boldsymbol{A}_3^{\mathrm{T}}&(1-\omega)\boldsymbol{I}-\omega^3\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_3\end{bmatrix}\\&:=(1-\omega)\boldsymbol{I}+\mathcal{K}_\omega,\end{aligned}$$

这里

$$\mathcal{K}_\omega = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & -\omega \boldsymbol{A}_1^{-1} \\ -\omega(1-\omega)\boldsymbol{A}_2 & \boldsymbol{O} & \omega^2 \boldsymbol{A}_3 \\ \omega(1-\omega)^2 \boldsymbol{A}_3^{\mathrm{T}} \boldsymbol{A}_2 & -\omega(1-\omega)\boldsymbol{A}_3^{\mathrm{T}} & -\omega^3 \boldsymbol{A}_3^{\mathrm{T}} \boldsymbol{A}_3 \end{bmatrix},$$

$$\begin{aligned} \boldsymbol{f}_\omega &= \omega \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ \omega \boldsymbol{A}_2 & \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \omega \boldsymbol{A}_3^{\mathrm{T}} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}_1^{-1} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{0} \end{bmatrix} \\ &= \omega \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ -\omega \boldsymbol{A}_2 & \boldsymbol{I} & \boldsymbol{O} \\ \omega^2 \boldsymbol{A}_3^{\mathrm{T}} \boldsymbol{A}_2 & -\omega \boldsymbol{A}_3^{\mathrm{T}} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_1^{-1} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{0} \end{bmatrix} \\ &= \begin{bmatrix} \omega \boldsymbol{A}_1^{-1} \boldsymbol{b}_1 \\ \omega(\boldsymbol{b}_2 - \omega \boldsymbol{A}_3 \boldsymbol{b}_1) \\ -\omega^2 \boldsymbol{A}_3^{\mathrm{T}} (\boldsymbol{b}_2 - \omega \boldsymbol{A}_3 \boldsymbol{b}_1) \end{bmatrix}. \end{aligned}$$

则详细的 SOR 迭代格式为

$$\begin{cases} \boldsymbol{x}^{(k+1)} = (1-\omega)\boldsymbol{x}^{(k)} + \omega \boldsymbol{A}_1^{-1}(\boldsymbol{b}_1 - \boldsymbol{v}^{(k)}), \\ \boldsymbol{w}^{(k+1)} = (1-\omega)\boldsymbol{w}^{(k)} - \omega(1-\omega)\boldsymbol{A}_2 \boldsymbol{x}^{(k)} - \omega^2 \boldsymbol{A}_3(\boldsymbol{b}_1 - \boldsymbol{v}^{(k)}) + \omega \boldsymbol{b}_2, \\ \boldsymbol{v}^{(k+1)} = (1-\omega)\boldsymbol{v}^{(k)} + \omega(1-\omega)^2 \boldsymbol{A}_3^{\mathrm{T}} \boldsymbol{A}_2 \boldsymbol{x}^{(k)} - \omega(1-\omega)\boldsymbol{A}_3^{\mathrm{T}} \boldsymbol{w}^{(k)} \\ \qquad -\omega^2 \boldsymbol{A}_3^{\mathrm{T}} \big[\boldsymbol{b}_2 - \omega \boldsymbol{A}_3(\boldsymbol{b}_1 - \boldsymbol{v}^{(k)})\big], \quad k = 0, 1, 2, \cdots. \end{cases} \tag{5.71}$$

特别地, 在上式中取 $\omega = 1$, 可得 Gauss–Seidel 迭代格式:

$$\begin{cases} \boldsymbol{x}^{(k+1)} = \boldsymbol{A}_1^{-1}(\boldsymbol{b}_1 - \boldsymbol{v}^{(k)}), \\ \boldsymbol{w}^{(k+1)} = \boldsymbol{b}_2 - \boldsymbol{A}_3(\boldsymbol{b}_1 - \boldsymbol{v}^{(k)}), \\ \boldsymbol{v}^{(k+1)} = -\boldsymbol{A}_3^{\mathrm{T}} \big[\boldsymbol{b}_2 - \boldsymbol{A}_3(\boldsymbol{b}_1 - \boldsymbol{v}^{(k)})\big], \quad k = 0, 1, 2, \cdots. \end{cases} \tag{5.72}$$

(3) SSOR 的迭代矩阵 $\mathcal{H}_\omega$ 和右端向量 $\boldsymbol{g}_\omega$ 分别为

$$\begin{aligned} \mathcal{H}_\omega &= (\boldsymbol{I} - \omega \mathcal{U})^{-1} \big[(1-\omega)\boldsymbol{I} + \omega \mathcal{L}\big](\boldsymbol{I} - \omega \mathcal{L})^{-1} \big[(1-\omega)\boldsymbol{I} + \omega \mathcal{U}\big] \\ &= (\boldsymbol{I} - \omega \mathcal{U})^{-1} (\boldsymbol{I} - \omega \mathcal{L})^{-1} \big[(1-\omega)\boldsymbol{I} + \omega \mathcal{L}\big] \big[(1-\omega)\boldsymbol{I} + \omega \mathcal{U}\big], \end{aligned} \tag{5.73}$$

$$\boldsymbol{g}_\omega = \omega(2-\omega)(\boldsymbol{I} - \omega \mathcal{U})^{-1} (\boldsymbol{I} - \omega \mathcal{L})^{-1} \boldsymbol{f}_J. \tag{5.74}$$

显然, $\mathcal{H}_\omega$ 相似于

$$\begin{aligned} \widehat{\mathcal{H}}_\omega &= (\boldsymbol{I} - \omega \mathcal{L})^{-1} \big[(1-\omega)\boldsymbol{I} + \omega \mathcal{L}\big] \big[(1-\omega)\boldsymbol{I} + \omega \mathcal{U}\big](\boldsymbol{I} - \omega \mathcal{U})^{-1} \\ &= (\boldsymbol{I} - \omega \mathcal{L})^{-1} \big[(1-\omega)\boldsymbol{I} + \omega \mathcal{L}\big](\boldsymbol{I} - \omega \mathcal{U})^{-1} \big[(1-\omega)\boldsymbol{I} + \omega \mathcal{U}\big] \\ &:= \mathcal{W}(\mathcal{L})\mathcal{W}(\mathcal{U}), \end{aligned} \tag{5.75}$$

这里

$$\mathcal{W}(\mathcal{Z}) = (\boldsymbol{I} - \omega\mathcal{Z})^{-1}\big[(1-\omega)\boldsymbol{I} + \omega\mathcal{Z}\big]. \tag{5.76}$$

直接计算, 得

$$\mathcal{W}(\mathcal{L}) = \begin{bmatrix} (1-\omega)\boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ -\omega(2-\omega)\boldsymbol{A}_2 & (1-\omega)\boldsymbol{I} & \boldsymbol{O} \\ \omega^2(2-\omega)\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_2 & -\omega(2-\omega)\boldsymbol{A}_3^{\mathrm{T}} & (1-\omega)\boldsymbol{I} \end{bmatrix},$$

$$\mathcal{W}(\mathcal{U}) = \begin{bmatrix} (1-\omega)\boldsymbol{I} & \boldsymbol{O} & -\omega(2-\omega)\boldsymbol{A}_1^{-1} \\ \boldsymbol{O} & (1-\omega)\boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & (1-\omega)\boldsymbol{I} \end{bmatrix}.$$

于是, 有

$$\widehat{\mathcal{H}}_\omega = \mathcal{W}(\mathcal{L})\mathcal{W}(\mathcal{U}) = (1-\omega)^2\boldsymbol{I} + \mathcal{T}_\omega, \tag{5.77}$$

式中:

$$\mathcal{T}_\omega = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & -\sigma\boldsymbol{A}_1^{-1} \\ -\sigma\boldsymbol{A}_2 & \boldsymbol{O} & \omega^2(2-\omega^2)\boldsymbol{A}_3 \\ \omega\sigma\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_2 & -\sigma\boldsymbol{A}_3^{\mathrm{T}} & -\omega^2(2-\omega)^2\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_3 \end{bmatrix},$$

这里 $\sigma = \omega(1-\omega)(2-\omega)$.

$$\begin{aligned} \boldsymbol{g}_\omega &= \omega(2-\omega)(\boldsymbol{I} - \omega\mathcal{U})^{-1}(\boldsymbol{I} - \omega\mathcal{L})^{-1}\boldsymbol{f}_J \\ &= \omega(2-\omega)\begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} & \omega\boldsymbol{A}_1^{-1} \\ \boldsymbol{O} & \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I} \end{bmatrix}^{-1}\begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ \omega\boldsymbol{A}_2 & \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \omega\boldsymbol{A}_3^{\mathrm{T}} & \boldsymbol{I} \end{bmatrix}^{-1}\boldsymbol{f}_J \\ &= \omega(2-\omega)\begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} & -\omega\boldsymbol{A}_1^{-1} \\ \boldsymbol{O} & \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I} \end{bmatrix}\begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ -\omega\boldsymbol{A}_2 & \boldsymbol{I} & \boldsymbol{O} \\ \omega^2\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_2 & -\omega\boldsymbol{A}_3^{\mathrm{T}} & \boldsymbol{I} \end{bmatrix}\boldsymbol{f}_J \\ &= \omega(2-\omega)\begin{bmatrix} \boldsymbol{A}_1^{-1}\boldsymbol{b}_1 + \omega^2\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_3\boldsymbol{b}_1 + \omega^2\boldsymbol{A}_1^{-1}\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{b}_2 \\ \boldsymbol{b} - \omega\boldsymbol{A}_3\boldsymbol{b}_1 \\ \omega^2\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{A}_3\boldsymbol{b}_1 - \omega\boldsymbol{A}_3^{\mathrm{T}}\boldsymbol{b}_2 \end{bmatrix}. \end{aligned}$$

据此即可写出 SSOR 迭代法的迭代格式

$$\boldsymbol{y}^{(k+1)} = \mathcal{T}_\omega\boldsymbol{y}^{(k)} + \boldsymbol{g}_\omega. \tag{5.78}$$

上述 Jacobi 迭代、SOR 迭代和 SSOR 迭代的收敛性分析可参考文献 [23].

5.4.4 基于 KKT 方程的 HSS 迭代法

现在考虑最小二乘问题 (5.38) 的 KKT 方程

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{A} \\ \boldsymbol{A}^{\mathrm{T}} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \tag{5.79}$$

的迭代解法. 由于式 (5.79) 的系数矩阵是对称不定的, 为了设计更有效的迭代算法, 将其改成成如下的等价形式:

$$\boldsymbol{B}\boldsymbol{z} := \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A} \\ -\boldsymbol{A}^{\mathrm{T}} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} := \boldsymbol{f}, \tag{5.80}$$

此时的 KKT 矩阵是非奇异的非对称半正定矩阵.

下面将白中治等学者提出的关于非对称正定线性方程组的 HSS (Hermite/skew-Hermite Splitting) 迭代法应用于求解上述 KKT 方程组 (5.80).

选取参数 $\alpha > 0$, 将矩阵 $\boldsymbol{B}$ 分裂成

$$\boldsymbol{B} = (\alpha\boldsymbol{I} + \boldsymbol{H}) - (\alpha\boldsymbol{I} - \boldsymbol{S}) = (\alpha\boldsymbol{I} + \boldsymbol{S}) - (\alpha\boldsymbol{I} - \boldsymbol{H}),$$

式中:

$$\boldsymbol{H} = \frac{1}{2}(\boldsymbol{B} + \boldsymbol{B}^{\mathrm{T}}) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}, \quad \boldsymbol{S} = \frac{1}{2}(\boldsymbol{B} - \boldsymbol{B}^{\mathrm{T}}) = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{A} \\ -\boldsymbol{A}^{\mathrm{T}} & \boldsymbol{O} \end{bmatrix}.$$

则针对 KKT 方程组 (5.80) 的 HSS 迭代格式为

$$\begin{cases} (\alpha\boldsymbol{I} + \boldsymbol{H})\boldsymbol{z}^{(k+\frac{1}{2})} = (\alpha\boldsymbol{I} - \boldsymbol{S})\boldsymbol{z}^{(k)} + \boldsymbol{f}, \\ (\alpha\boldsymbol{I} + \boldsymbol{S})\boldsymbol{z}^{(k+1)} = (\alpha\boldsymbol{I} - \boldsymbol{H})\boldsymbol{z}^{(k+\frac{1}{2})} + \boldsymbol{f}. \end{cases} \tag{5.81}$$

更详细地, 即

$$\begin{cases} \begin{bmatrix} (\alpha+1)\boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \alpha\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}^{(k+\frac{1}{2})} \\ \boldsymbol{x}^{(k+\frac{1}{2})} \end{bmatrix} = \begin{bmatrix} \alpha\boldsymbol{I} & -\boldsymbol{A} \\ \boldsymbol{A}^{\mathrm{T}} & \alpha\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}^{(k)} \\ \boldsymbol{x}^{(k)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}, \\ \begin{bmatrix} \alpha\boldsymbol{I} & \boldsymbol{A} \\ -\boldsymbol{A}^{\mathrm{T}} & \alpha\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}^{(k+1)} \\ \boldsymbol{x}^{(k+1)} \end{bmatrix} = \begin{bmatrix} (\alpha-1)\boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \alpha\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}^{(k+\frac{1}{2})} \\ \boldsymbol{x}^{(k+\frac{1}{2})} \end{bmatrix} + \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}. \end{cases} \tag{5.82}$$

应用 HSS 迭代法 (5.81) 时, 求解两个子问题可采用直接法或内迭代法进行计算. 此外, 不难发现, 迭代法 (5.81) 的迭代矩阵为

$$\mathcal{L}(\alpha) = (\alpha\boldsymbol{I} + \boldsymbol{S})^{-1}(\alpha\boldsymbol{I} - \boldsymbol{H})(\alpha\boldsymbol{I} + \boldsymbol{H})^{-1}(\alpha\boldsymbol{I} - \boldsymbol{S}). \tag{5.83}$$

收敛性定理如下.

定理 5.7 设 $\boldsymbol{A}\in\mathbb{R}_n^{m\times n}$, $\sigma_k\,(k=1,2,\cdots,n)$ 为矩阵 $\boldsymbol{A}$ 的 (正) 奇异值. 则 HSS 迭代法 (5.82) 的迭代矩阵 $\mathcal{L}(\alpha)$ 有 $m-n$ 重特征值为 $\dfrac{\alpha-1}{\alpha+1}$, 和 $2n$ 个特征值为

$$\frac{1}{(\alpha+1)(\alpha^2+\sigma_k^2)}\left[\alpha(\alpha^2-\sigma_k^2)\pm\sqrt{(\alpha^2+\sigma_k^2)^2-4\alpha^4\sigma_k^2}\right],\quad k=1,2,\cdots,n.$$

因此, 有

$$\rho(\mathcal{L}(\alpha))<1,\quad\forall\,\alpha>0,$$

即 HSS 迭代收敛到 KKT 方程组 (5.79) 的准确解.

证明 由于

$$\alpha\boldsymbol{I}\pm\boldsymbol{H}=\begin{bmatrix}(\alpha\pm1)\boldsymbol{I} & \boldsymbol{O}\\ \boldsymbol{O} & \alpha\boldsymbol{I}\end{bmatrix},\quad \alpha\boldsymbol{I}\pm\boldsymbol{S}=\begin{bmatrix}\alpha\boldsymbol{I} & \pm\boldsymbol{A}\\ \mp\boldsymbol{A}^{\mathrm{T}} & \alpha\boldsymbol{I}\end{bmatrix},$$

且有

$$\alpha\boldsymbol{I}+\boldsymbol{S}=\begin{bmatrix}\boldsymbol{I} & \boldsymbol{O}\\ -\alpha^{-1}\boldsymbol{A}^{\mathrm{T}} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\alpha\boldsymbol{I} & \boldsymbol{A}\\ \boldsymbol{O} & \boldsymbol{S}(\alpha)\end{bmatrix},$$

这里 $\boldsymbol{S}(\alpha)=\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$. 直接计算, 得

$$\mathcal{L}(\alpha)=\begin{bmatrix}\mathcal{L}_1(\alpha) & \mathcal{L}_2(\alpha)\\ \mathcal{L}_3(\alpha) & \mathcal{L}_4(\alpha)\end{bmatrix},$$

式中:

$$\mathcal{L}_1(\alpha)=\frac{\alpha-1}{\alpha+1}\boldsymbol{I}-\frac{2}{\alpha+1}\boldsymbol{A}\boldsymbol{S}(\alpha)^{-1}\boldsymbol{A}^{\mathrm{T}},\quad \mathcal{L}_2(\alpha)=-\frac{2\alpha}{\alpha+1}\boldsymbol{A}\boldsymbol{S}(\alpha)^{-1},$$
$$\mathcal{L}_3(\alpha)=\frac{2\alpha}{\alpha+1}\boldsymbol{S}(\alpha)^{-1}\boldsymbol{A}^{\mathrm{T}},\quad \mathcal{L}_4(\alpha)=-\frac{\alpha-1}{\alpha+1}\boldsymbol{I}+\frac{2\alpha^2}{\alpha+1}\boldsymbol{S}(\alpha)^{-1}.$$

设矩阵 $\boldsymbol{A}$ 的奇异值分解为

$$\boldsymbol{A}=\boldsymbol{U}\begin{bmatrix}\boldsymbol{\Sigma}\\ \boldsymbol{O}\end{bmatrix}\boldsymbol{V}^{\mathrm{T}},\quad \boldsymbol{\Sigma}=\mathrm{diag}\,(\sigma_1,\sigma_2,\cdots,\sigma_n),$$

式中: $\boldsymbol{U}\in\mathbb{R}^{m\times m}$, $\boldsymbol{V}\in\mathbb{R}^{n\times n}$ 为正交矩阵. 于是有

$$\boldsymbol{S}(\alpha)=\alpha\boldsymbol{I}+\frac{1}{\alpha}\boldsymbol{V}\begin{bmatrix}\boldsymbol{\Sigma} & \boldsymbol{O}\end{bmatrix}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\begin{bmatrix}\boldsymbol{\Sigma}\\ \boldsymbol{O}\end{bmatrix}\boldsymbol{V}^{\mathrm{T}}=\boldsymbol{V}\Big(\alpha\boldsymbol{I}+\frac{1}{\alpha}\boldsymbol{\Sigma}^2\Big)\boldsymbol{V}^{\mathrm{T}},$$

及

$$\mathcal{L}_1(\alpha)=\boldsymbol{U}\begin{bmatrix}\dfrac{\alpha-1}{\alpha+1}\boldsymbol{I}-\dfrac{2}{\alpha+1}\big(\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{\Sigma}^2\big)^{-1}\boldsymbol{\Sigma}^2 & \boldsymbol{O}\\ \boldsymbol{O} & \dfrac{\alpha-1}{\alpha+1}\boldsymbol{I}\end{bmatrix}\boldsymbol{U}^{\mathrm{T}},$$

$$\mathcal{L}_2(\alpha)=\boldsymbol{U}\begin{bmatrix}-\dfrac{2\alpha}{\alpha+1}\boldsymbol{\Sigma}\big(\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{\Sigma}^2\big)^{-1}\\ \boldsymbol{O}\end{bmatrix}\boldsymbol{V}^{\mathrm{T}},$$

$$\mathcal{L}_3(\alpha)=\boldsymbol{V}\begin{bmatrix}\dfrac{2\alpha}{\alpha+1}\boldsymbol{\Sigma}\big(\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{\Sigma}^2\big)^{-1}, & \boldsymbol{O}\end{bmatrix}\boldsymbol{U}^{\mathrm{T}},$$

$$\mathcal{L}_4(\alpha)=\boldsymbol{V}\left[-\frac{\alpha-1}{\alpha+1}\boldsymbol{I}+\frac{2\alpha^2}{\alpha+1}\big(\alpha\boldsymbol{I}+\frac{1}{\alpha}\boldsymbol{\Sigma}^2\big)^{-1}\right]\boldsymbol{V}^{\mathrm{T}}.$$

令

$$\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{U} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{V}\end{bmatrix},$$

则 $\mathcal{L}(\alpha)$ 经过正交相似变换 $\widetilde{\mathcal{L}}(\alpha)=\boldsymbol{Q}^{\mathrm{T}}\mathcal{L}(\alpha)\boldsymbol{Q}$ 后, 化为

$$\widetilde{\mathcal{L}}(\alpha)=\begin{bmatrix}\dfrac{\alpha-1}{\alpha+1}\boldsymbol{I}-\dfrac{2}{\alpha+1}\big(\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{\Sigma}^2\big)^{-1}\boldsymbol{\Sigma}^2 & \boldsymbol{O} & -\dfrac{2\alpha}{\alpha+1}\boldsymbol{\Sigma}\big(\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{\Sigma}^2\big)^{-1}\\ \boldsymbol{O} & \dfrac{\alpha-1}{\alpha+1}\boldsymbol{I} & \boldsymbol{O}\\ \dfrac{2\alpha}{\alpha+1}\boldsymbol{\Sigma}\big(\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{\Sigma}^2\big)^{-1} & \boldsymbol{O} & -\dfrac{\alpha-1}{\alpha+1}\boldsymbol{I}+\dfrac{2\alpha^2}{\alpha+1}\big(\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{\Sigma}^2\big)^{-1}\end{bmatrix},$$

从而可得 $\mathcal{L}(\alpha)$ 有 $m-n$ 重特征值 $\dfrac{\alpha-1}{\alpha+1}$ (显然其绝对值<1), 和矩阵

$$\begin{aligned}\mathcal{L}_k(\alpha)&=\begin{bmatrix}\dfrac{\alpha-1}{\alpha+1}-\dfrac{2}{\alpha+1}\big(\alpha+\dfrac{1}{\alpha}\sigma_k^2\big)^{-1}\sigma_k^2 & -\dfrac{2\alpha}{\alpha+1}\sigma_k\big(\alpha+\dfrac{1}{\alpha}\sigma_k^2\big)^{-1}\\ \dfrac{2\alpha}{\alpha+1}\sigma_k\big(\alpha+\dfrac{1}{\alpha}\sigma_k^2\big)^{-1} & -\dfrac{\alpha-1}{\alpha+1}+\dfrac{2\alpha^2}{\alpha+1}\big(\alpha+\dfrac{1}{\alpha}\sigma_k^2\big)^{-1}\end{bmatrix}\\ &=\frac{1}{(\alpha+1)(\alpha^2+\sigma_k^2)}\begin{bmatrix}(\alpha-1)\alpha^2-(\alpha+1)\sigma_k^2 & -2\alpha^2\sigma_k\\ 2\alpha^2\sigma_k & (\alpha+1)\alpha^2-(\alpha-1)\sigma_k^2\end{bmatrix}\end{aligned}$$

的特征值, $k=1,2,\cdots,n$. 而二阶矩阵 $\mathcal{L}_k(\alpha)$ 的特征方程为

$$\left(\lambda-\frac{(\alpha-1)\alpha^2-(\alpha+1)\sigma_k^2}{(\alpha+1)(\alpha^2+\sigma_k^2)}\right)\left(\lambda-\frac{(\alpha+1)\alpha^2-(\alpha-1)\sigma_k^2}{(\alpha+1)(\alpha^2+\sigma_k^2)}\right)+\frac{4\alpha^4\sigma_k^2}{(\alpha+1)^2(\alpha^2+\sigma_k^2)^2}=0,$$

化简, 得

$$\lambda^2-\frac{2\alpha(\alpha^2-\sigma_k^2)}{(\alpha+1)(\alpha^2+\sigma_k^2)}\lambda+\frac{\alpha-1}{\alpha+1}=0. \tag{5.84}$$

它的两个根为

$$\lambda_{1,2}^{(k)}=\frac{\alpha(\alpha^2-\sigma_k^2)\pm\sqrt{(\alpha^2+\sigma_k^2)^2-4\alpha^4\sigma_k^2}}{(\alpha+1)(\alpha^2+\sigma_k^2)}. \tag{5.85}$$

不难验证, 当 $\alpha^2+\sigma_k^2>2\alpha^2\sigma_k$ 时, 有

$$|\lambda_{1,2}^{(k)}|\leqslant\frac{1}{(\alpha+1)(\alpha^2+\sigma_k^2)}\left[\alpha|\alpha^2-\sigma_k^2|+\sqrt{(\alpha^2+\sigma_k^2)^2-4\alpha^4\sigma_k^2}\right]$$

$$= \frac{\alpha}{\alpha+1}\left[\frac{|\alpha^2-\sigma_k^2|}{\alpha^2+\sigma_k^2}+\sqrt{\frac{1}{\alpha^2}-\frac{4\alpha^2\sigma_k^2}{(\alpha^2+\sigma_k^2)^2}}\right]$$
$$< \frac{\alpha}{\alpha+1}\left(1+\frac{1}{\alpha}\right)=1.$$

当 $\alpha^2+\sigma_k^2=2\alpha^2\sigma_k$ 时, 有

$$|\lambda_{1,2}^{(k)}| = \frac{\alpha|\alpha^2-\sigma_k^2|}{(\alpha+1)(\alpha^2+\sigma_k^2)} \leqslant \frac{\alpha}{\alpha+1} < 1.$$

当 $\alpha^2+\sigma_k^2<2\alpha^2\sigma_k$ 时, $\lambda_{1,2}^{(k)}$ 为一对共轭复根, 此时有

$$|\lambda_{1,2}^{(k)}| = \sqrt{\lambda_1^{(k)}\lambda_2^{(k)}} = \frac{|\alpha-1|}{\alpha+1} < 1.$$

综合上述, $\rho(\mathcal{L}(\alpha))<1, \forall\ \alpha>0$. 证毕. □

此外, 还可以讨论 HSS 迭代法 (5.81) 中的最优参数 α 的选取, 详细的分析过程可参考文献 [22], 其结论如下.

定理 5.8 设 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ 列满秩, $\alpha>0$ 是给定的常数. 若 $\sigma_k\,(k=1,2,\cdots,n)$ 是矩阵 $\boldsymbol{A}$ 的正奇异值, $\sigma_{\min}=\min\limits_{1\leqslant k\leqslant n}\{\sigma_k\}$, $\sigma_{\max}=\max\limits_{1\leqslant k\leqslant n}\{\sigma_k\}$, 则解线性方程组 (5.80) 的 HSS 迭代法的最优参数 α 为

$$\alpha^* = \arg\min_{\alpha>0}\rho(\mathcal{L}(\alpha)) = \sqrt{\sigma_{\min}\sigma_{\max}},$$

并且

$$\rho(\mathcal{L}(\alpha^*)) = \frac{\sigma_{\max}-\sigma_{\min}}{\sigma_{\max}+\sigma_{\min}}.$$

下面考虑迭代法 (5.82) 的执行. 这一算法的计算量主要集中在求解第 2 个子方程组, 因为其系数矩阵是一个分块的反对称矩阵. 但由于其对角块的特殊性, 可以将其降阶处理. 具体迭代格式如下:

$$\begin{cases} \boldsymbol{r}^{(k+\frac{1}{2})} = \dfrac{1}{\alpha+1}\left(\alpha\boldsymbol{r}^{(k)} - \boldsymbol{A}\boldsymbol{x}^{(k)} + \boldsymbol{b}\right), \\ \boldsymbol{x}^{(k+\frac{1}{2})} = \boldsymbol{x}^{(k)} + \dfrac{1}{\alpha}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}^{(k)}, \\ \left(\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\right)\boldsymbol{r}^{(k+1)} = (\alpha-1)\boldsymbol{r}^{(k+\frac{1}{2})} - \boldsymbol{A}\boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{b}, \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k+\frac{1}{2})} + \dfrac{1}{\alpha}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}^{(k+1)}. \end{cases} \tag{5.86}$$

迭代格式 (5.86) 的主要工作量集中在第 3 式, 即求解以 $\alpha\boldsymbol{I}+\dfrac{1}{\alpha}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 为系数矩阵的方程组. 但由于该矩阵是对称正定的, 故可使用共轭梯度法进行非精确求解, 相应的迭代格式称为非精确 HSS 迭代法 (简记为 IHSS).

IHSS 迭代格式的 MATLAB 程序如下:

```
function [x,fval,k,time]=kkt_ihss(A,b,alpha,x,tol,max_it)
%IHSS算法求解KKT方程r=b-Ax, A'r=0
if nargin<6, max_it=1000; end
if nargin<5, tol=1.e-6; end
if nargin<4, x=zeros(size(A,2),1); end
tic; m=size(A,1); r=b-A*x;
s=A'*r; nr=norm(s); k=0;
B=alpha*eye(m)+A*A'/alpha;
while (k<=max_it)
    k=k+1;
    r=(alpha*r-A*x+b)/(alpha+1);
    x=x+s/alpha;
    c=(alpha-1)*r-A*x+b;
    r=eq_cg(B,c);  %CG法求解子问题
    s=A'*r; x=x+s/alpha;
    if (norm(s)/nr<tol), break; end
end
fval=norm(r);
time=toc;
```

例 5.8 利用 HSS 迭代法的 MATLAB 程序, 计算例 5.6 中的最小二乘问题 $\min \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2$ 的极小解, 并与 CG 迭代法进行比较 (取初始点为零向量, 参数 $\alpha = 6.0$, 容许误差 $\varepsilon = 10^{-10}$).

解 编写 MATLAB 脚本文件 ex58.m, 并在命令窗口运行之, 得到数值结果如表 5.2 所示.

表 5.2 IHSS, HSS 与 CG 方法的数值结果

算法	迭代次数	CPU时间	极小解	极小值
IHSS	741	0.0003	$(-0.0303, 0.0171, 2.4508, 1.2953)^{\mathrm{T}}$	0.9959
HSS	743	0.0159	$(-0.0303, 0.0171, 2.4508, 1.2953)^{\mathrm{T}}$	0.9959
CG	5	0.0031	$(-0.0309, 0.0171, 2.4509, 1.2954)^{\mathrm{T}}$	0.9959

习题 5

5.1 设

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}.$$

求 LS 问题 (5.1) 的 LS 解 $\boldsymbol{x}_{\mathrm{LS}}$.

5.2 设

$$\boldsymbol{A}=\begin{bmatrix}1&3&1&1\\2&0&0&0\\1&0&0&0\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}1\\1\\1\end{bmatrix}.$$

求 LS 问题的全部解.

5.3 证明问题 (5.18) 和问题 (5.19) 的等价性.

5.4 利用等式

$$\|\boldsymbol{A}(\boldsymbol{x}+\alpha\boldsymbol{w})-\boldsymbol{b}\|_2^2=\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+2\alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}-b)+\alpha^2\|\boldsymbol{A}\boldsymbol{w}\|_2^2.$$

证明: 若 $\boldsymbol{x}\in S_{\mathrm{LS}}$, 则 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$.

5.5 给定矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$, 其 Moore-Penrose 广义逆矩阵记为 $\boldsymbol{A}^{\dagger}$. 试证明:

(1) 若 $\mathrm{rank}(\boldsymbol{A})=n$, 则 $\boldsymbol{A}^{\dagger}=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}$;

(2) 若 $\mathrm{rank}(\boldsymbol{A})=r\leqslant n$, 且 $\boldsymbol{A}=\boldsymbol{B}\boldsymbol{C}$, 其中 $\boldsymbol{B}\in\mathbb{R}^{m\times r}$, $\boldsymbol{C}\in\mathbb{R}^{r\times n}$, 则 $\boldsymbol{A}^{\dagger}=\boldsymbol{C}^{\dagger}\boldsymbol{B}^{\dagger}$;

(3) $\boldsymbol{A}^{\dagger}\boldsymbol{b}$ 是超定方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的极小范数最小二乘解.

5.6 设矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$, 且存在 $\boldsymbol{X}\in\mathbb{C}^{n\times m}$ 使得对每个 $\boldsymbol{b}\in\mathbb{R}^m$, $\boldsymbol{x}=\boldsymbol{X}\boldsymbol{b}$ 均极小化 $\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$. 试证明: $\boldsymbol{A}\boldsymbol{X}\boldsymbol{A}=\boldsymbol{A}$ 和 $(\boldsymbol{A}\boldsymbol{X})^{\mathrm{T}}=\boldsymbol{A}\boldsymbol{X}$.

5.7 给定 $\boldsymbol{f}\in\mathbb{R}^m$, $\boldsymbol{g}\in\mathbb{R}^n$. 定义

$$\mathcal{X}=\{\boldsymbol{X}\in\mathbb{R}^{m\times n}:\boldsymbol{X}^{\mathrm{T}}\boldsymbol{f}=\boldsymbol{g}\}.$$

证明: 当且仅当 $\boldsymbol{g}\boldsymbol{f}^{\dagger}\boldsymbol{f}=\boldsymbol{g}$ 时, $\mathcal{X}\neq\emptyset$; 且当 $\mathcal{X}\neq\emptyset$ 时, 任一 $\boldsymbol{X}\in\mathcal{X}$ 都可表示成

$$\boldsymbol{X}=(\boldsymbol{f}^{\mathrm{T}})^{\dagger}\boldsymbol{g}^{\mathrm{T}}+(\boldsymbol{I}-\boldsymbol{f}\boldsymbol{f}^{\dagger})\boldsymbol{Z},$$

其中 $\boldsymbol{Z}\in\mathbb{R}^{m\times n}$.

5.8 给定矩阵 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$, $\boldsymbol{B}\in\mathbb{C}^{n\times n}$ 和 $\boldsymbol{C}\in\mathbb{C}^{m\times n}$. 求矩阵 $\boldsymbol{X}\in\mathbb{C}^{m\times n}$, 使其满足

$$\|\boldsymbol{A}\boldsymbol{X}-\boldsymbol{X}\boldsymbol{B}-\boldsymbol{C}\|_{\mathrm{F}}=\min,$$

并讨论 $\boldsymbol{A}\boldsymbol{X}-\boldsymbol{X}\boldsymbol{B}=\boldsymbol{C}$ 相容的条件.

5.9 证明: 如果 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ 的秩为 n, 则只要 $\|\boldsymbol{E}\|_2\|\boldsymbol{A}^{\dagger}\|_2<1$, 就有 $\boldsymbol{A}+\boldsymbol{E}$ 的秩也为 n.

5.10 证明: 如果 $\boldsymbol{A}_k\to\boldsymbol{A}$ 和 $\boldsymbol{A}_k^{\dagger}\to\boldsymbol{A}^{\dagger}$ 成立, 则存在正整数 k_0, 使得 $\forall k\geqslant k_0$, 秩 $\mathrm{rank}(\boldsymbol{A}_k)$ 是常数.

5.11 设 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$, 定义 $\boldsymbol{B}(t)=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+t\boldsymbol{I})^{-1}\boldsymbol{A}^{\mathrm{T}}$, 其中 $t>0$. 试证明

$$\|\boldsymbol{B}(t)-\boldsymbol{A}^{\dagger}\|_2=\frac{t}{\sigma_r(\boldsymbol{A})[\sigma_r(\boldsymbol{A})^2+t]},\quad r=\mathrm{rank}(\boldsymbol{A},$$

从而, 当 $t\to0$ 时, 有 $\boldsymbol{B}(t)\to\boldsymbol{A}^{\dagger}$.

5.12 假定 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{f}$, $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{E})\widehat{\boldsymbol{x}}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{f}$, $2\|\boldsymbol{E}\|_2\leqslant\sigma_n(\boldsymbol{A})^2$. 试证明: 如果 $\boldsymbol{r}=\boldsymbol{f}-\boldsymbol{A}\boldsymbol{x}$, $\widehat{\boldsymbol{r}}=\boldsymbol{f}-\boldsymbol{A}\widehat{\boldsymbol{x}}$, 则有 $\widehat{\boldsymbol{r}}-\boldsymbol{r}=\boldsymbol{A}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{E})^{-1}\boldsymbol{E}\boldsymbol{x}$ 和

$$\|\widehat{\boldsymbol{r}}-\boldsymbol{r}\|_2\leqslant2\kappa_2(\boldsymbol{A})\frac{\|\boldsymbol{E}\|_2}{\boldsymbol{A}\|_2}\|\boldsymbol{x}\|_2.$$

第 6 章　解线性方程组的直接法

本章考虑如下 n 阶线性方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \tag{6.1}$$

的直接法, 其中 $\boldsymbol{A}=(a_{ij})$ 称为方程组的系数矩阵, $\boldsymbol{b}=(b_1,b_2,\cdots,b_n)^{\mathrm{T}}$ 称为方程组的右端项, 向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ 为所求的解. 若系数矩阵 $\boldsymbol{A}$ 是非奇异的, 则线性方程组 (6.1) 存在唯一解.

所谓直接法, 是指在没有舍入误差的情况下经过有限次运算可求得方程组的精确解的方法. 本章主要介绍求解线性方程组 (6.1) 最基本的直接法—Gauss 消去法和 LU 分解法以及舍入误差分析等.

6.1　Gauss 消去法

6.1.1　顺序 Gauss 消去法

Gauss 消去法的基本思想是: 首先使用初等行变换将方程组转化为一个同解的上三角形方程组 (称为消元), 再通过回代法求解该三角形方程组 (称为回代). 按行原先的位置进行消元的 Gauss 消去法称为顺序 Gauss 消去法.

例 6.1　用顺序 Gauss 消去法解线性方程组

$$\begin{cases} x_1+x_2+x_3+x_4=6, \\ -x_1+2x_2-3x_3+x_4=-2, \\ 3x_1-3x_2+6x_3-2x_4=7, \\ -4x_1+5x_2+2x_3-3x_4=7. \end{cases}$$

解　(1) 消元过程:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 6 \\ -1 & 2 & -3 & 1 & -2 \\ 3 & -3 & 6 & -2 & 7 \\ -4 & 5 & 2 & -3 & 7 \end{bmatrix} \xrightarrow[r_4+4r_1]{\substack{r_2+r_1 \\ r_3-3r_1}} \begin{bmatrix} 1 & 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & 2 & 4 \\ 0 & -6 & 3 & -5 & -11 \\ 0 & 9 & 6 & 1 & 31 \end{bmatrix} \xrightarrow[r_4-3r_2]{r_3+2r_2}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & 2 & 4 \\ 0 & 0 & -1 & -1 & -3 \\ 0 & 0 & 12 & -5 & 19 \end{bmatrix} \xrightarrow{r_4+12r_3} \begin{bmatrix} 1 & 1 & 1 & 1 & 6 \\ 0 & 3 & -2 & 2 & 4 \\ 0 & 0 & -1 & -1 & -3 \\ 0 & 0 & 0 & -17 & -17 \end{bmatrix}.$$

(2) 回代过程:

$$\left\{\begin{array}{r} x_1+x_2+x_3+x_4=6,\\ 3x_2-2x_3+2x_4=4,\\ -x_3-x_4=-3,\\ -17x_4=-17. \end{array}\right. \Rightarrow \left\{\begin{array}{l} x_4=1,\\ x_3=3-x_4=2,\\ x_2=(4+2x_3-2x_4)/3=2,\\ x_1=6-x_2-x_3-x_4=1. \end{array}\right.$$

故原方程组的解是: $x_1=1,\ x_2=2,\ x_3=2,\ x_4=1.$

对于一般线性方程组, 使用顺序 Gauss 消去法求解

$$\left\{\begin{array}{l} a_{11}^{(1)}x_1+a_{12}^{(1)}x_2+\cdots+a_{1n}^{(1)}x_n=b_1^{(1)},\\ a_{21}^{(1)}x_1+a_{22}^{(1)}x_2+\cdots+a_{2n}^{(1)}x_n=b_2^{(1)},\\ \qquad\qquad\vdots\\ a_{n1}^{(1)}x_1+a_{n2}^{(1)}x_2+\cdots+a_{1n}^{(1)}x_n=b_n^{(1)}. \end{array}\right. \tag{6.2}$$

(1) 消元过程:

$$\left[\begin{array}{ccccccc} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)}\\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)}\\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} & b_3^{(1)}\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ a_{n1}^{(1)} & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right] \to \left[\begin{array}{ccccccc} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)}\\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)}\\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & b_3^{(2)}\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right]$$

$$\to\cdots\to \left[\begin{array}{ccccccc} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)}\\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)}\\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} & b_3^{(3)}\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} & b_n^{(n)} \end{array}\right],$$

式中:

$$a_{ij}^{(2)}=a_{ij}^{(1)}-m_{i1}a_{1j}^{(1)},\quad b_i^{(2)}=b_i^{(1)}-m_{i1}b_1^{(1)},\quad m_{i1}=\frac{a_{i1}^{(1)}}{a_{11}^{(1)}},\quad i,j=2,\cdots,n.$$

一般地, 有

$$a_{ij}^{(k+1)}=a_{ij}^{(k)}-m_{ik}a_{kj}^{(k)},\quad b_i^{(k+1)}=b_i^{(k)}-m_{ik}b_k^{(k)},\quad m_{ik}=\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}},$$

$$i,j=k+1,\cdots,n;\ k=1,\cdots,n-1. \tag{6.3}$$

(2) 回代过程:

$$\begin{cases} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n = b_1^{(1)}, \\ \qquad\quad a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)}, \\ \qquad\qquad\qquad\qquad \vdots \\ \qquad\qquad\qquad\quad a_{nn}^{(n)}x_n = b_n^{(n)}, \end{cases}$$

$$\Longrightarrow \begin{cases} x_n = b_n^{(n)}/a_{nn}^{(n)}, \\ x_k = \Big(b_k^{(k)} - \sum\limits_{j=k+1}^{n} a_{kj}^{(k)}x_j\Big)/a_{kk}^{(k)}, \quad k = n-1, \cdots, 2, 1. \end{cases} \tag{6.4}$$

在此基础上, 得到顺序 Gauss 消去法的具体算法步骤.

算法 6.1 (顺序 Gauss 消去法)

步1, 输入系数矩阵 $\boldsymbol{A}$, 右端项 $\boldsymbol{b}$, 置 $k := 1$.

步2, 消元: 对 $k = 1, \cdots, n-1$, 计算

$$m_{ik} = a_{ik}^{(k)}/a_{kk}^{(k)}, \quad a_{ik}^{(k+1)} = 0,$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}, \quad b_i^{(k+1)} = b_i^{(k)} - m_{ik}b_k^{(k)}.$$

$$(i = k+1, \cdots, n; \quad j = k+1, \cdots, n.)$$

步3, 回代:

$$x_n = b_n^{(n)}/a_{nn}^{(n)},$$

$$x_k = \Big(b_k^{(k)} - \sum_{j=k+1}^{n} a_{kj}^{(k)}x_j\Big)/a_{kk}^{(k)}, \ k = n-1, \cdots, 1.$$

可以统计顺序 Gauss 消去法的计算量. 由于加减法的计算量可忽略不计, 此处只统计乘除法次数.

消元过程: 第 $k\,(k = 1, \cdots, n-1)$ 步消元, 有

$$(n-k)(n-k+1) + (n-k) = (n-k)(n-k+2)$$

次乘除法, 共

$$N_1 = \sum_{k=1}^{n-1}(n-k)(n-k+2) = \sum_{i=1}^{n-1}(i^2 + 2i)$$

$$= \frac{n(n-1)(2n-1)}{6} + n(n-1) = \frac{n(n-1)(2n+5)}{6}$$

次乘除法.

回代过程: 计算 $x_k\,(k = n, \cdots, 2, 1)$ 时, 有 $n-k+1$ 次乘除法, 共

$$N_2 = \sum_{k=1}^{n}(n-k+1) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

次乘除法.

消元和回代过程共计

$$N_1+N_2=\frac{n(n-1)(2n+5)}{6}+\frac{n(n+1)}{2}=\frac{n^3}{3}+n^2-\frac{n}{3}$$

次乘除法.

可见消元过程的计算量为 $O(n^3)$, 而回代过程的计算量为 $O(n^2)$, 因此顺序 Gauss 消去法的计算量主要在消元过程部分.

算法 6.1 要求对所有的 $k=1,\cdots,n$, $a_{kk}^{(k)}\neq 0$, 此时称顺序 Gauss 消去法是可行的. 一般将 $a_{kk}^{(k)}$ 称为第 k 步消元的主元. 下面的定理给出了顺序 Gauss 消去法可行的一个充要条件.

定理 6.1 证明: *顺序 Gauss 消去法可行的充分必要条件是系数矩阵 $\boldsymbol{A}$ 的所有顺序主子式 $\boldsymbol{D}_k\neq 0$, $k=1,2,\cdots,n$.*

证明 必要性. 若顺序 Gauss 消去法是可行的, 即 $a_{kk}^{(k)}\neq 0$, 则可进行消去法的 $k-1$ 步 ($k\leqslant n$). 由于 $\boldsymbol{A}^{(k)}$ 是由 $\boldsymbol{A}$ 逐行实行初等变换 (某数乘以某一行加到另一行) 得到的, 这些运算不改变相应顺序主子式的值, 故有

$$\boldsymbol{D}_k=\begin{vmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2k}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{kk}^{(k)} \end{vmatrix}=a_{11}^{(1)}a_{22}^{(2)}\cdots a_{kk}^{(k)}\neq 0,\quad k=1,2,\cdots,n.$$

充分性. 用归纳法证明. 当 $k=1$ 时显然成立. 设命题对 $k-1$ 成立. 现设 $\boldsymbol{D}_1\neq 0$, $\cdots$, $\boldsymbol{D}_{k-1}\neq 0$, $\boldsymbol{D}_k\neq 0$. 由归纳法假设有 $a_{11}^{(1)}\neq 0,\cdots,a_{k-1,k-1}^{(k-1)}\neq 0$. 因此, 消去法可以进行第 $k-1$ 步, $\boldsymbol{A}$ 约化为

$$\boldsymbol{A}^{(k)}=\begin{bmatrix} \boldsymbol{A}_{11}^{(k-1)} & \boldsymbol{A}_{12}^{(k-1)} \\ & \boldsymbol{A}_{22}^{(k)} \end{bmatrix},$$

式中: $\boldsymbol{A}_{11}^{(k-1)}$ 是对角元为 $a_{11}^{(1)},\cdots,a_{k-1,k-1}^{(k-1)}$ 的上三角矩阵. 因 $\boldsymbol{A}^{(k)}$ 是通过行初等变换由 $\boldsymbol{A}$ 逐步得到的, 故 $\boldsymbol{A}$ 的 k 阶顺序主子式与 $\boldsymbol{A}^{(k)}$ 的 k 阶顺序主子式相等, 即

$$\boldsymbol{D}_k=\det\begin{bmatrix} \boldsymbol{A}_{11}^{(k-1)} & \alpha_{12}^{(k-1)} \\ & a_{kk}^{(k)} \end{bmatrix}=a_{11}^{(1)}\cdots a_{k-1,k-1}^{(k-1)}a_{kk}^{(k)},$$

式中: $\alpha_{12}^{(k-1)}$ 为 $\boldsymbol{A}_{12}^{(k-1)}$ 的第 1 列. 故由 $\boldsymbol{D}_k\neq 0$ 及归纳法假设可推出 $a_{kk}^{(k)}\neq 0$. 证毕.□

根据算法 6.1, 可编制 MATLAB 程序如下:

```
%顺序Gauss消去法程序-msgauss.m
function x=msgauss(A,b,flag)
```

```
%输入:A为系数矩阵,b为右端项,若flag=0(默认),
%      则不显示中间过程,否则显示中间过程
%输出:x为解向量
if nargin<3, flag=0; end
n=length(b);
for k=1:(n-1)  %消元过程
   m=A(k+1:n,k)/A(k,k);
   A(k+1:n,k+1:n)=A(k+1:n,k+1:n)-m*A(k,k+1:n);
   b(k+1:n)=b(k+1:n)-m*b(k);
   A(k+1:n,k)=zeros(n-k,1);
   if flag~=0, Ab=[A,b], end
end
x=zeros(n,1);  %回代过程
x(n)=b(n)/A(n,n);
for k=n-1:-1:1
   x(k)=(b(k)-A(k,k+1:n)*x(k+1:n))/A(k,k);
end
```

例 6.2 利用程序 msgauss.m 计算下列方程组的解

$$\begin{cases} x_1+x_2+x_3+x_4=6, \\ -x_1+2x_2-3x_3+x_4=-2, \\ 3x_1-3x_2+6x_3-2x_4=7, \\ -4x_1+5x_2+2x_3-3x_4=7. \end{cases}$$

解 在 MATLAB 命令窗口输入:

```
>> A=[1 1 1 1;-1 2 -3 1;3 -3 6 -2;-4 5 2 -3];
>> b=[6 -2 7 7]';
>> x=msgauss(A,b)
```

可得计算结果 (略).

6.1.2 列主元 Gauss 消去法

一般来说, 顺序 Gauss 消去法的计算过程是不可靠的, 一旦出现 $a_{kk}^{(k)}=0$, 计算就无法进行下去. 另外, 即使对所有 $k=1,2,\cdots,n$, $a_{kk}^{(k)}\neq 0$, 也不能保证计算过程是数值稳定的.

例 6.3 设有线性方程组

$$\begin{cases} 0.0001x_1+1.0x_2=1.0, \\ 1.0x_1+1.0x_2=2.0. \end{cases}$$

其精确解为

$$x_1 = \frac{10000}{9999} \approx 1.00010, \quad x_2 = \frac{9998}{9999} \approx 0.99990.$$

现在假定用尾数为 4 位十进制字长的浮点数来求解.

解 (1) 消元过程: 根据 4 位浮点数运算规则 $1.0 - 10000.0 = (0.00001 - 0.1)10^5 = (0.0000 - 0.1)10^5 = -10000.0$ (舍入). 同理, $2.0 - 10000.0 = -10000.0$.

$$\begin{bmatrix} 0.0001 & 1.0 & 1.0 \\ 1.0 & 1.0 & 2.0 \end{bmatrix} \xrightarrow{r_2 - 10^4 r_1} \begin{bmatrix} 0.0001 & 1.0 & 1.0 \\ 0 & 1.0 - 10000.0 & 2.0 - 10000.0 \end{bmatrix}$$

$$\xrightarrow{\text{舍入}} \begin{bmatrix} 0.0001 & 1.0 & 1.0 \\ 0 & -10000.0 & -10000.0 \end{bmatrix}.$$

(2) 回代过程:

$$\begin{cases} 0.0001x_1 + 1.0x_2 = 1.0, \\ \quad -10000.0x_2 = -10000.0. \end{cases} \Longrightarrow \begin{cases} x_2 = 1.0, \\ x_1 = 0.0. \end{cases}$$

代入原方程组验算, 发现结果严重失真.

分析结果失真的原因发现, 由于第 1 列的主元素 0.0001 绝对值过于小, 当它在消元过程中作分母时把中间过程数据放大 10000 倍, 使中间结果“吃”掉了原始数据, 从而造成数值不稳定.

针对以上问题, 考虑选用绝对值大的数作为主元素.

(1) 消元过程:

$$\begin{bmatrix} 0.0001 & 1.0 & 1.0 \\ 1.0 & 1.0 & 2.0 \end{bmatrix} \xrightarrow{r_1 \leftrightarrow r_2} \begin{bmatrix} 1.0 & 1.0 & 2.0 \\ 0.0001 & 1.0 & 1.0 \end{bmatrix}$$

$$\xrightarrow{r_2 - 0.0001 r_1} \begin{bmatrix} 1.0 & 1.0 & 2.0 \\ 0 & 1.0 - 0.0001 & 1.0 - 0.0002 \end{bmatrix}$$

$$\xrightarrow{\text{舍入}} \begin{bmatrix} 1.0 & 1.0 & 2.0 \\ 0 & 1.0 & 1.0 \end{bmatrix}.$$

这里, 舍入过程 $1.0 - 0.0001 = (0.1 - 0.00001)10^1$ (舍入), 同理 $1.0 - 0.0002 = 1.0$.

(2) 回代过程:

$$\begin{cases} 1.0x_1 + 1.0x_2 = 2.0, \\ \quad 1.0x_2 = 1.0. \end{cases} \Longrightarrow \begin{cases} x_2 = 1.0, \\ x_1 = 1.0. \end{cases}$$

代入原方程组验算, 发现结果基本合理.

例 6.3 说明了选主元素的重要性. 下面阐述列主元 Gauss 消去法的基本思想. 记 $\boldsymbol{A}^{(1)}=\boldsymbol{A}$, 在消元过程的第 1 步, 取第 1 列中绝对值最大的元素 $a_{r_1 1}^{(1)}$, 即

$$a_{r_1 1}^{(1)}=\max_{1\leqslant i\leqslant n}|a_{i1}^{(1)}|$$

作为主元素. 若 $r_1>1$, 交换第 r_1 行和第 1 行.

一般地, 在消元过程的第 k 步, 取

$$a_{r_k k}^{(k)}=\max_{k\leqslant i\leqslant n}|a_{ik}^{(k)}| \tag{6.5}$$

作为主元素. 若 $r_k>k$, 交换第 r_k 行和第 k 行.

列主元 Gauss 消去法的算法具体步骤如下.

算法 6.2 (列主元 Gauss 消去法)

步1, 输入系数矩阵 $\boldsymbol{A}$, 右端项 $\boldsymbol{b}$, 置 $k:=1$.

步2, 对 $k=1,\cdots,n-1$ 进行如下操作:

① 选列主元, 确定 r_k, 使

$$a_{r_k k}^{(k)}=\max_{k\leqslant i\leqslant n}|a_{ik}^{(k)}|,$$

若 $a_{r_k k}^{(k)}=0$, 则停止计算, 否则, 进行下一步.

② 若 $r_k>k$, 交换 $(A^{(k)},b^{(k)})$ 的第 k, r_k 两行.

③ 消元: 对 $i,j=k+1,\cdots,n$, 计算

$$m_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)},\quad a_{ik}^{(k+1)}=0,$$

$$a_{ij}^{(k+1)}=a_{ij}^{(k)}-m_{ik}a_{kj}^{(k)},\quad b_i^{(k+1)}=b_i^{(k)}-m_{ik}b_k^{(k)}.$$

步3, 回代:

$$x_n=b_n^{(n)}/a_{nn}^{(n)},$$

$$x_k=\Big(b_k^{(k)}-\sum_{j=k+1}^{n}a_{kj}^{(k)}x_j\Big)/a_{kk}^{(k)},\quad k=n-1,\cdots,1.$$

根据算法 6.2, 编制 MATLAB 程序如下:

```
%列主元Gauss消去法程序-mgauss2.m
function x=mgauss2(A,b,flag)
%输入:A为系数矩阵,b为右端项,若flag=0(默认),
%      则不显示中间过程,否则显示中间过程
%输出:x为解向量
if nargin<3, flag=0; end
n=length(b);
for k=1:(n-1)  %选主元
   [ap,p]=max(abs(A(k:n,k)));p=p+k-1;
   if p>k
      A([k p],:)=A([p k],:);b([k p],:)=b([p k],:);
```

```
    end
    m=A(k+1:n,k)/A(k,k);  %消元
    A(k+1:n,k+1:n)=A(k+1:n,k+1:n)-m*A(k,k+1:n);
    b(k+1:n)=b(k+1:n)-m*b(k);A(k+1:n,k)=zeros(n-k,1);
    if flag~=0, Ab=[A,b], end
end
x=zeros(n,1); x(n)=b(n)/A(n,n); %回代
for k=n-1:-1:1
    x(k)=(b(k)-A(k,k+1:n)*x(k+1:n))/A(k,k);
end
```

例 6.4 利用程序 mgauss2.m 计算下列线性方程组的解

$$\begin{bmatrix} 2 & -1 & 4 & -3 & 1 \\ -1 & 1 & 2 & 1 & 3 \\ 4 & 2 & 3 & 3 & -1 \\ -3 & 1 & 3 & 2 & 4 \\ 1 & 3 & -1 & 4 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 11 \\ 14 \\ 4 \\ 16 \\ 18 \end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[2 -1 4 -3 1;-1 1 2 1 3;4 2 3 3 -1;-3 1 3 2 4;1 3 -1 4 4];
>> b=[11 14 4 16 18]';
>> x=mpgauss(A,b); x=x'
```

得计算结果:

```
x =
   1.0000   2.0000   1.0000  -1.0000   4.0000
```

6.2 LU 分解法

把一个 n 阶矩阵分解成两个三角形矩阵的乘积称为矩阵的三角分解. 本节介绍矩阵的 LU 分解 $\boldsymbol{A}=\boldsymbol{LU}$, 其中 $\boldsymbol{L}$ 是单位下三角矩阵, $\boldsymbol{U}$ 是上三角矩阵. 这种形式的分解对于求解方程组 (6.1) 是十分有用的. 事实上, 若 $\boldsymbol{A}=\boldsymbol{LU}$ 是一个 LU 分解, 此时线性方程组

$$\boldsymbol{Ax}=\boldsymbol{b} \Longrightarrow \boldsymbol{LUx}=\boldsymbol{b} \Longrightarrow \begin{cases} \boldsymbol{Ly}=\boldsymbol{b} \\ \boldsymbol{Ux}=\boldsymbol{y} \end{cases} \tag{6.6}$$

转化为 $\boldsymbol{Ly}=\boldsymbol{b}$ 及 $\boldsymbol{Ux}=\boldsymbol{y}$ 两个三角形方程组. 由于三角形方程组很容易通过向前消去法或回代方法求解, 且只有 $O(n^2)$ 的计算量, 故研究矩阵的 LU 分解十分有意义.

6.2.1 顺序 LU 分解法

首先讨论矩阵的 LU 分解. 设 $\boldsymbol{A}=\boldsymbol{LU}$, 其中 $\boldsymbol{L}$ 为一个单位下三角矩阵, $\boldsymbol{U}$ 为一个上三角矩阵, 即

$$\boldsymbol{L}=\begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix},\quad \boldsymbol{U}=\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}. \tag{6.7}$$

下面推导三角形矩阵 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的元素的计算公式. 由等式 $\boldsymbol{A}=\boldsymbol{LU}$, 得

$$a_{ij}=\begin{bmatrix} l_{i1}, & \cdots, & l_{i,i-1}, & 1, & 0, & \cdots, & 0 \end{bmatrix}\begin{bmatrix} u_{1j} \\ \vdots \\ u_{j-1,j} \\ u_{jj} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{6.8}$$

当 $j\geqslant i$ 时, 有

$$a_{ij}=l_{i1}u_{1j}+\cdots+l_{i,i-1}u_{i-1,j}+u_{ij},$$

于是

$$u_{ij}=a_{ij}-\sum_{r=1}^{i-1}l_{ir}u_{rj};$$

当 $j<i$ 时, 有

$$a_{ij}=l_{i1}u_{1j}+\cdots+l_{i,j-1}u_{j-1,j}+l_{ij}u_{jj},$$

于是

$$l_{ij}=\Big(a_{ij}-\sum_{r=1}^{j-1}l_{ir}u_{rj}\Big)/u_{jj}.$$

即

$$u_{1j}=a_{1j},\quad j=1,\cdots,n;\quad l_{i1}=a_{i1}/u_{11},\quad i=2,\cdots,n; \tag{6.9}$$

$$u_{ij}=a_{ij}-\sum_{r=1}^{i-1}l_{ir}u_{rj},\quad i=2,\cdots,n;\ j=i,\cdots,n; \tag{6.10}$$

$$l_{ij}=\Big(a_{ij}-\sum_{r=1}^{j-1}l_{ir}u_{rj}\Big)/u_{jj},\quad i=3,\cdots,n;\ j=2,\cdots,i-1. \tag{6.11}$$

为了便于编程计算, 将式 (6.9) 第 1 式中的下标 j 换成 i, 式 (6.10) 中的下标 i 换成 k, 下标 j 换成 i, 式 (6.11) 中的下标 j 换成 k, 则有

$$u_{1i} = a_{1i}, \quad i = 1, \cdots, n; \quad l_{i1} = a_{i1}/u_{11}, \quad i = 2, \cdots, n; \tag{6.12}$$

$$u_{ki} = a_{ki} - \sum_{r=1}^{k-1} l_{kr} u_{ri}, \quad k = 2, \cdots, n; \ i = k, \cdots, n; \tag{6.13}$$

$$l_{ik} = \Big(a_{ik} - \sum_{r=1}^{k-1} l_{ir} u_{rk}\Big)/u_{kk}, \quad k = 2, \cdots, n-1; \ i = k+1, \cdots, n. \tag{6.14}$$

上述这种按矩阵 $\boldsymbol{A}$ 元素的自然顺序进行分解的方法称为顺序 LU 分解法. 下面是用顺序 LU 分解求解线性方程组的算法步骤.

算法 6.3 (顺序 LU 分解法)

步1, 输入系数矩阵 $\boldsymbol{A}$, 右端项 $\boldsymbol{b}$.

步2, LU 分解:

$u_{1i} = a_{1i}, \quad i = 1, \cdots, n;$

$l_{i1} = a_{i1}/u_{11}, \quad i = 2, \cdots, n;$

对 $k = 2, \cdots, n$, 计算

$$u_{ki} = a_{ki} - \sum_{r=1}^{k-1} l_{kr} u_{ri}, \quad i = k, \cdots, n;$$

$$l_{ik} = \big(a_{ik} - \sum_{r=1}^{k-1} l_{ir} u_{rk}\big)/u_{kk}, \quad i = k+1, \cdots, n.$$

步3, 用向前消去法解下三角方程组 $\boldsymbol{Ly} = \boldsymbol{b}$:

$$y_1 = b_1; \ y_k = b_k - \sum_{i=1}^{k-1} l_{ki} y_i, \quad k = 2, \cdots, n.$$

步4, 用回代法解上三角方程组 $\boldsymbol{Ux} = \boldsymbol{y}$:

$$x_n = y_n/u_{nn}; \ x_k = \Big(y_k - \sum_{i=k+1}^{n} u_{ki} x_i\Big)\Big/u_{kk}, \quad k = n-1, \cdots, 1.$$

注 6.1 可以看出, 利用 LU 分解, 分开了系数矩阵的计算和对右端项的计算. 正是这一特点, 使得 LU 分解法特别适用于求解系数矩阵相同而右端项不同的一系列方程组, 而控制论等领域中刚好存在这样的实际问题.

下面考虑顺序 LU 分解的程序实现. 注意到 LU 分解后, 原系数矩阵 $\boldsymbol{A}$ 的数据不再需要保留, 因此, 为了节省存储空间, 可在 MATLAB 程序中将分解后的单位下三角矩阵 $\boldsymbol{L}$ 和上三角矩阵 $\boldsymbol{U}$ 分别存放在系数矩阵 $\boldsymbol{A}$ 的严格下三角和上三角部分 (单位下三角矩阵的对角线元素 1 不需存储), 而不再为其开辟额外的存储单元.

算法 6.3 的 MATLAB 程序如下:

```
%顺序LU分解法程序-mslu.m
function [x,A]=mslu(A,b)
```

```
%输入:A为系数矩阵,b为右端向量
%输出:x为解向量,L和U分别存放在A的严格下三角和上三角部分
n=length(b);
for k=1:n  %顺序LU分解
   A(k:n,k)=A(k:n,k)-A(k:n,1:k-1)*A(1:k-1,k);
   A(k+1:n,k)=A(k+1:n,k)/A(k,k);  %乘子向量
   A(k,k+1:n)=A(k,k+1:n)-A(k,1:k-1)*A(1:k-1,k+1:n);
end
y=zeros(n,1);
for k=1:n, %解下三角矩阵Ly=b
   y(k)=b(k)-A(k,1:k-1)*y(1:k-1);
end
x=zeros(n,1);
for k=n:-1:1, %解上三角方程组Ux=y
   x(k)=(y(k)-A(k,k+1:n)*x(k+1:n))/A(k,k);
end
```

例 6.5 利用程序 mslu.m 计算例 6.4 中的线性方程组的解.

解 在 MATLAB 命令窗口输入:

```
>> A=[2 -1 4 -3 1;-1 1 2 1 3;4 2 3 3 -1;-3 1 3 2 4;1 3 -1 4 4];
>> b=[11 14 4 16 18]';
>> [x,A]=mslu(A,b)
```

可得计算结果 (略).

6.2.2 列主元 LU 分解法

顺序 LU 分解法在本质上与顺序 Gauss 消去法是一致的. 由算法 6.1 可知, 顺序 Gauss 消去法的第 1 步消元相当于用矩阵

$$\boldsymbol{M}_1 = \begin{bmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & & 1 & & \\ \vdots & & & \ddots & \\ -m_{n1} & & & & 1 \end{bmatrix}$$

左乘 $[\boldsymbol{A}^{(1)}, \boldsymbol{b}^{(1)}]$, 这里 m_{i1} 由式 (6.3) 所定义, 即

$$[\boldsymbol{A}^{(2)}, \boldsymbol{b}^{(2)}] = \boldsymbol{M}_1[\boldsymbol{A}^{(1)}, \boldsymbol{b}^{(1)}].$$

第 2 步消元相当于用矩阵

$$\boldsymbol{M}_2=\begin{bmatrix}1 & & & & \\ & 1 & & & \\ & -m_{32} & 1 & & \\ & \vdots & & \ddots & \\ & -m_{n2} & & & 1\end{bmatrix}$$

左乘 $[\boldsymbol{A}^{(2)},\boldsymbol{b}^{(2)}]$, 即

$$[\boldsymbol{A}^{(3)},\boldsymbol{b}^{(3)}]=\boldsymbol{M}_2[\boldsymbol{A}^{(2)},\boldsymbol{b}^{(2)}]=\boldsymbol{M}_2\boldsymbol{M}_1[\boldsymbol{A}^{(1)},\boldsymbol{b}^{(1)}].$$

一般地, 第 k 步相当于用矩阵

$$\boldsymbol{M}_k=\begin{bmatrix}1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -m_{k+1,\,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -m_{nk} & & & 1\end{bmatrix}$$

左乘 $[\boldsymbol{A}^{(k)},\boldsymbol{b}^{(k)}]$, 即

$$\begin{aligned}\left[\boldsymbol{A}^{(k+1)},\boldsymbol{b}^{(k+1)}\right]&=\boldsymbol{M}_k\left[\boldsymbol{A}^{(k)},\boldsymbol{b}^{(k)}\right]=\cdots\\&=\boldsymbol{M}_k\cdots\boldsymbol{M}_2\boldsymbol{M}_1\left[\boldsymbol{A}^{(1)},\boldsymbol{b}^{(1)}\right],\quad k=1,\cdots,n-1.\end{aligned}$$

由顺序 Gauss 消去法可知, 经过 $n-1$ 步消元后, 系数矩阵 $\boldsymbol{A}$ 被化成了上三角矩阵, 即 $\boldsymbol{A}^{(n)}=\boldsymbol{U}$, 从而

$$\boldsymbol{U}=\boldsymbol{A}^{(n)}=\boldsymbol{M}_{n-1}\boldsymbol{A}^{(n-1)}=\boldsymbol{M}_{n-1}\cdots\boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{A},$$

于是

$$\boldsymbol{A}=\boldsymbol{M}_1^{-1}\boldsymbol{M}_2^{-1}\cdots\boldsymbol{M}_{n-1}^{-1}\boldsymbol{U}=\boldsymbol{L}\boldsymbol{U},$$

式中:

$$\boldsymbol{L}=\boldsymbol{M}_1^{-1}\boldsymbol{M}_2^{-1}\cdots\boldsymbol{M}_{n-1}^{-1}=\begin{bmatrix}1 & & & & \\ m_{21} & 1 & & & \\ \vdots & & \ddots & & \\ m_{n-1,\,1} & m_{n-1\,2} & \cdots & 1 & \\ m_{n1} & m_{n2} & \cdots & m_{n,\,n-1} & 1\end{bmatrix}$$

为单位下三角矩阵. 由此可见, 顺序 Gauss 消去法实际上就是将方程组的系数矩阵分解成单位下三角矩阵与上三角矩阵的乘积. 对比算法 6.1 和算法 6.3, 不难看出, 顺序 Gauss 消去法的消元过程相当于 LU 分解过程和 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{b}$ 的求解, 而回代过程则相当于解线性方程组 $\boldsymbol{U}\boldsymbol{x}=\boldsymbol{y}$.

定理 6.2 若 n 阶方阵 $\boldsymbol{A}$ 的所有顺序主子式都不等于零, 则 $\boldsymbol{A}$ 存在唯一的 LU 分解 $\boldsymbol{A}=\boldsymbol{LU}$.

证明 因顺序 LU 分解本质上等同于顺序 Gauss 消去法, 故存在性由定理 6.1 立即可得. 下面证明唯一性. 事实上, 若 $\boldsymbol{A}$ 存在两种不同的三角分解:

$$\boldsymbol{A}=\boldsymbol{LU}=\boldsymbol{L}_1\boldsymbol{U}_1,$$

式中: $\boldsymbol{L}$ 和 $\boldsymbol{L}_1$ 均为单位下三角矩阵; $\boldsymbol{U}$ 和 $\boldsymbol{U}_1$ 均为上三角矩阵. 因 $\boldsymbol{A}$ 是非奇异的, 故 $\boldsymbol{U}$ 和 $\boldsymbol{U}_1$ 也是非奇异的. 于是由上式, 得

$$\boldsymbol{L}_1^{-1}\boldsymbol{L}=\boldsymbol{U}_1\boldsymbol{U}^{-1}.$$

注意到上式的左边是单位下三角矩阵, 而右边则是上三角矩阵, 故必有

$$\boldsymbol{L}_1^{-1}\boldsymbol{L}=\boldsymbol{U}_1\boldsymbol{U}^{-1}=\boldsymbol{I}\ (\text{单位阵}),$$

即 $\boldsymbol{L}_1=\boldsymbol{L}, \boldsymbol{U}_1=\boldsymbol{U}$. 证毕. □

根据上面的分析, 既然顺序 LU 分解法本质上等同于顺序 Gauss 消去法, 因此, 为了提高计算的数值稳定性, 有必要考虑列主元 LU 分解技术. 这只需要在一般 LU 分解的第 k 步避免绝对值较小的 u_{kk} 作除数即可. 假设第 $k-1$ 步已经完成, 在进行第 k 步分解之前进行选主元的操作. 可以引入

$$s_i=a_{ik}-\sum_{r=1}^{k-1}l_{ir}u_{rk},\quad i=k,k+1,\cdots,n,$$

且令

$$|s_{i_k}|=\max_{k\leqslant i\leqslant n}|s_i|.$$

然后用 s_{i_k} 作为 u_{kk} 并交换增广矩阵 $[\boldsymbol{A},\boldsymbol{b}]$ 的第 k 行和第 i_k 行, 于是有 $|l_{ik}|\leqslant 1\,(i=k+1,\cdots,n)$, 再进行第 k 步分解. 算法如下:

算法 6.4 (列主元 LU 分解法)

步1, 输入系数矩阵 $\boldsymbol{A}$, 右端项 $\boldsymbol{b}$.

步2, 列主元 LU 分解:

对 $k=1,\cdots,n$,

① 计算 $s_i=a_{ik}-\sum\limits_{r=1}^{k-1}l_{ir}u_{rk}\Longrightarrow a_{ik},\ \ i=k,k+1,\cdots,n.$

② 选主元 $|s_{i_k}|=\max\limits_{k\leqslant i\leqslant n}|s_i|$, 并记录 i_k, $s_{i_k}\Longrightarrow u_{kk}$.

③ 交换 $[\boldsymbol{A},\boldsymbol{b}]$ 的第 k 行和第 i_k 行元素.

④ 计算 $\boldsymbol{L}$ 的第 k 列元素: $l_{ik}=s_i/u_{kk}=a_{ik}/a_{kk}\Longrightarrow a_{ik},\ \ i=k+1,\cdots,n.$

⑤ 计算 $\boldsymbol{U}$ 的第 k 行元素: $u_{kj}=a_{kj}-\sum\limits_{r=1}^{k-1}l_{kr}u_{rj}\Longrightarrow a_{kj},\ \ j=k+1,\cdots,n.$

步3, 用向前消去法解下三角方程组 $\boldsymbol{Ly}=\boldsymbol{b}$:

$$y_1=b_1;\ y_k=b_k-\sum_{j=1}^{k-1}l_{kj}y_j,\quad k=2,\cdots,n.$$

步4, 用回代法解上三角方程组 $\boldsymbol{Ux}=\boldsymbol{y}$:

$$x_n=y_n/u_{nn};;\ x_k=\Big(y_k-\sum_{j=k+1}^{n}u_{kj}x_j\Big)/u_{kk},\quad k=n-1,\cdots,1.$$

下面给出列主元 LU 分解法的 MATLAB 程序:

```
%列主元LU分解法程序-mplu.m
function [x,A,P]=mplu(A,b)
%列主元LU分解PA=LU
%输入:A为系数矩阵,b为右端向量
%输出:x返回解向量,L和U分别存放在A的严格下三角和
%      上三角部分,P返回选主元时记录行交换的置换阵
n=length(b);
P=eye(n);  %P记录选择主元时候所进行的行变换
for k=1:n  %列主元LU分解
   A(k:n,k)=A(k:n,k)-A(k:n,1:k-1)*A(1:k-1,k);
   [s,m]=max(abs(A(k:n,k)));  %选列主元
   m=m+k-1;
   if m~=k
      A([k m],:)=A([m k],:);P([k m],:)=P([m k],:);
   end
   A(k+1:n,k)=A(k+1:n,k)/A(k,k);
   A(k,k+1:n)=A(k,k+1:n)-A(k,1:k-1)*A(1:k-1,k+1:n);
end
b=P*b; y=zeros(n,1);
for k=1:n, %解下三角矩阵Ly=b
   y(k)=b(k)-A(k,1:k-1)*y(1:k-1);
end
x=zeros(n,1);
for k=n:-1:1, %解上三角方程组Ux=y
   x(k)=(y(k)-A(k,k+1:n)*x(k+1:n))/A(k,k);
end
```

例 6.6 利用程序 mplu.m 计算下列线性方程组的解

$$\begin{bmatrix}2&-1&4&-3&1\\-1&1&2&1&3\\4&2&3&3&-1\\-3&1&3&2&4\\1&3&-1&4&4\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{bmatrix}=\begin{bmatrix}11\\14\\4\\16\\18\end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[2 -1 4 -3 1;-1 1 2 1 3;4 2 3 3 -1;-3 1 3 2 4;1 3 -1 4 4];
>> b=[11 14 4 16 18]';
>> [x,A,P]=mplu(A,b)
```

可得计算结果 (略).

6.2.3 不完全 LU 分解

当 $\boldsymbol{A}$ 的非零元较少且按一定规则分布时 (称为非零元的稀疏模式), 其 LU 分解 (又称完全 LU 分解) 产生的单位下三角矩阵 $\boldsymbol{L}$ 和上三角矩阵 $\boldsymbol{U}$ 一般不能保持和 $\boldsymbol{A}$ 相同的稀疏模式. 本节讨论矩阵 $\boldsymbol{A}$ 的不完全 LU 分解, 即在保持稀疏模式的前提下进行 LU 分解, 这样分解后 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的乘积是 $\boldsymbol{A}$ 的一个近似: $\boldsymbol{A} \approx \boldsymbol{LU}$. 这种分解得到的 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的逆矩阵往往用来对方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 作预处理:

$$\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{b}}, \quad \widetilde{\boldsymbol{A}} = \boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{U}^{-1}, \quad \widetilde{\boldsymbol{x}} = \boldsymbol{U}\boldsymbol{x}, \quad \widetilde{\boldsymbol{b}} = \boldsymbol{L}^{-1}\boldsymbol{b}.$$

使得预处理后方程组的系数矩阵有更好的条件数.

定义 6.1 设 $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$, 非对角元的位置和除对角线之外的其他非零元的位置分别记为下面的两个集合

$$\mathrm{ND} = \{(i,j) : 1 \leqslant i,j \leqslant n \text{ 且 } i \neq j\}, \quad \mathrm{NZ} = \{(i,j) \in \mathrm{ND} : a_{ij} \neq 0\}.$$

设指标集 $\mathcal{F}$ 满足 $\mathrm{NZ} \subseteq \mathcal{F} \subseteq \mathrm{ND}$, $0 \leqslant \omega \leqslant 1$. 若矩阵 $\boldsymbol{A}$ 有分解

$$\boldsymbol{A} = \boldsymbol{LU} + \boldsymbol{R},$$

式中: $\boldsymbol{L} = (l_{ij}) \in \mathbb{R}^{n\times n}$ 为单位下三角阵且满足

$$l_{ij} = 0, \quad i > j, \quad (i,j) \notin \mathcal{F}; \tag{6.15}$$

$\boldsymbol{U} = (u_{ij}) \in \mathbb{R}^{n\times n}$ 为上三角阵且满足

$$u_{ij} = 0, \quad i < j, \quad (i,j) \notin \mathcal{F}; \tag{6.16}$$

矩阵 $\boldsymbol{R} = (r_{ij}) \in \mathbb{R}^{n\times n}$ 满足

$$r_{ij} = 0, \quad (i,j) \in \mathcal{F}, \tag{6.17}$$

$$r_{ii} = -\omega \sum_{j=1,\, j\neq i}^{n} r_{ij}, \quad i = 1,2,\cdots,n. \tag{6.18}$$

则称 $\boldsymbol{A}$ 有关于 $\mathcal{F}$ 和 ω 的松弛不完全 LU 分解, ω 称为松弛参数.

若 $\mathcal{F} = \mathrm{ND}$, 由式 (6.17) 表明 $\boldsymbol{R}$ 的非对角元为 0, 式 (6.18) 表明 $\boldsymbol{R}$ 的对角元也为 0, 所以 $\boldsymbol{R} = \boldsymbol{O}$. 同时式 (6.15) 和式 (6.16) 也不需要满足, 此时的松弛不完全 LU 分解

即为完全 LU 分解. 若 $\mathcal{F}=\mathrm{NZ}$, 式 (6.17) 和式 (6.18) 表明 $\boldsymbol{L}$ 的下三角部分以及 $\boldsymbol{U}$ 的上三角部分和 $\boldsymbol{A}$ 的零元分布是一样的. 在分解中可以把集合

$$\mathrm{ND}\backslash\mathcal{F}=\{(i,j):1\leqslant i\neq j\leqslant n,(i,j)\notin\mathcal{F}\}$$

看成矩阵 $\boldsymbol{A}$ 的零模式, 而 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 保持了这种零模式. 上述松弛不完全 LU 分解中, 若参数 $\omega=0$, 相应的分解称为不完全 LU 分解 (ILU). 若参数 $\omega=1$, 相应的分解称为修正的不完全 LU 分解 (MILU).

下面给出矩阵 $\boldsymbol{A}$ 关于 $\mathcal{F}$ 和 ω 的松弛不完全 LU 分解, 该分解可以通过 $n-1$ 步 Gauss 消去法和 $n-1$ 次修正来实现, 称为修正的 Gauss 消去法. 具体步骤如下: 记 $\boldsymbol{A}^{(1)}=(a_{ij}^{(1)})=\boldsymbol{A}$, 假定已经执行了 $k-1$ 步修正的 Gauss 消去法并产生了 $\boldsymbol{A}^{(k)}=(a_{ij}^{(k)})$, 则第 k 步修正的 Gauss 消去法如下.

(1) Gauss 消去法. 把 $\boldsymbol{A}^{(k)}$ 的第 k 列中第 $k+1$ 个元素到第 n 个元素变为 0, 令

$$\widetilde{\boldsymbol{A}}^{(k)}=\boldsymbol{L}_k\boldsymbol{A}^{(k)}=\boldsymbol{A}^{(k)}-\boldsymbol{l}_k\boldsymbol{a}_k,$$

式中: $\boldsymbol{L}_k=\boldsymbol{I}-\boldsymbol{l}_k\boldsymbol{e}_k^{\mathrm{T}}$, 且

$$\begin{aligned}&\boldsymbol{l}_k=(0,\cdots,0,l_{k+1,k},\cdots,l_{nk})^{\mathrm{T}},\ \ l_{ik}=a_{ik}^{(k-1)}/a_{kk}^{(k-1)},\ i=k+1,\cdots,n,\\&\boldsymbol{a}_k=\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{A}^{(k)}=(0,\cdots,0,a_{k,k}^{(k)},\cdots,a_{kn}^{(k)})^{\mathrm{T}}.\end{aligned}$$

(2) 修正 $\widetilde{\boldsymbol{A}}^{(k)}=(\widetilde{a}_{ij}^{(k)})$ 右下角的 $n-k$ 阶子矩阵. 将 $\widetilde{\boldsymbol{A}}^{(k-1)}$ 修正为

$$\boldsymbol{A}^{(k+1)}=\widetilde{\boldsymbol{A}}^{(k)}+\boldsymbol{R}_k,$$

式中: $\boldsymbol{R}_k=(r_{ij}^{(k)})$ 为

$$r_{ij}^{(k)}=\begin{cases}\widetilde{a}_{ij}^{(k-1)},\ \ k+1\leqslant i,j\leqslant n,\ i\neq j,\ (i,j)\notin\mathcal{F};\\-\omega\sum\limits_{\substack{l=k+1\\l\neq i,\,(i,l)\notin\mathcal{F}}}\widetilde{a}_{il}^{(k-1)},\ \ k+1\leqslant i,j\leqslant n,\ i=j;\\0,\ \ \text{其他}.\end{cases}$$

第 $n-1$ 步修正的 Gauss 消去法后, 可得单位下三角矩阵

$$\boldsymbol{L}=(\boldsymbol{L}_{n-1}\boldsymbol{L}_{n-2}\cdots\boldsymbol{L}_1)^{-1}$$

和上三角阵 $\boldsymbol{U}=\boldsymbol{A}^{(n)}$. 令

$$\boldsymbol{R}=\sum_{k=1}^{n-1}\boldsymbol{R}_k,\quad \boldsymbol{R}_k=(r_{ij}^{(k)}),$$

则可得如下定理 (见文献 [13]).

定理 6.3 设 $a_{kk}^{(k)}\neq 0$, $k=1,2,\cdots,n$, 则上述修正的 Gauss 消去法得到的 $\boldsymbol{A}=\boldsymbol{LU}+\boldsymbol{R}$ 就是关于 $\mathcal{F}$ 和 ω 的松弛不完全 LU 分解.

下面给出修正 Gauss 消去法可以进行下去的充分条件.

定义 6.2 若矩阵 $\boldsymbol{A}$ 满足

(1) $a_{ii} > 0\,(i = 1, 2, \cdots, n-1)$, $a_{nn} \geqslant 0$.

(2) $a_{ij} \leqslant 0\,(i \neq j)$, $1 \leqslant i, j \leqslant n$.

(3) $n(i) = \max\{j : 1 \leqslant j \leqslant n,\ a_{ij} \neq 0\} > i,\quad i = 1, 2, \cdots, n-1$.

则称 $\boldsymbol{A}$ 为 $\hat{M}$ 矩阵.

有下面的定理, 详细证明参考文献 [6].

定理 6.4 设 $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$ 为弱严格对角占优的 $\hat{M}$ 矩阵, 对任意指标集 $\mathcal{F}$ (NZ $\subset \mathcal{F} \subset$ ND) 和 ω $(0 \leqslant \omega \leqslant 1)$, 修正的 Gauss 消去法给出了 $\boldsymbol{A}$ 关于 $\mathcal{F}$ 和 ω 的松弛不完全 LU 分解.

定义非零模式

$$\widetilde{\mathcal{F}} = \mathcal{F} \cup \big\{(i,i) : i = 1, 2, \cdots, n\big\},$$

它是零模式的补集. 加入修正的 Gauss 消去法是为了确保每步的零模式和非零模式都保持不变. 所以每次修正消元实际上只是对非零模式 $\widehat{\mathcal{F}}$ 的元素进行了 Gauss 消元, 同时将对角元素进行相应的修正. 下面给出松弛不完全 LU 分解具体的算法, 其中 $\boldsymbol{L}$ 存放在 $\boldsymbol{A}$ 的严格下三角部分, $\boldsymbol{U}$ 存放在 $\boldsymbol{A}$ 的上三角部分.

算法 6.5 (松弛不完全 LU 分解) 给定矩阵 $\boldsymbol{A}$, 非零模式 $\widetilde{\mathcal{F}}$ 和松弛参数 ω.

```
for k = 1 : n − 1
  for i = k + 1 : n
    if (i, k) ∈ F̃ (非零模式)
      a_ik = a_ik / a_kk;
      for j = k + 1 : n
        a_ij = a_ij − a_ik · a_kj;
      end
      for j = k + 1 : n
        if (i, j) ∉ F̃ (零模式)
          r_ij = r_ij + a_ij; r_ii = r_ii − ω ∗ a_ij;
          a_ii = a_ii + ω ∗ a_ij; a_ij = 0;
        end
      end
    end
  end
end
```

下面给出 ILU 分解的 MATLAB 程序如下:

```
function [L,U,R]=milu(A,NF,omega)
%矩阵A的松弛不完全LU分解
%输入:矩阵A,非零模式NF,松弛参数omega
%输出:单位下三角阵L,上三角阵U,剩余矩阵R,满足A=LU+R
n=size(A,1); R=zeros(n);
for k=1:n-1
   for i=k+1:n
      if (ismember(i+k*sqrt(-1),NF)==1)  %(i,k)属于集合NF
         A(i,k)=A(i,k)/A(k,k);
         for j=k+1:n
            A(i,j)=A(i,j)-A(i,k)*A(k,j);
         end
         for j=k+1:n
            if (ismember(i+j*sqrt(-1),NF)==0) %(i,j)不属于集合NF
               R(i,j)=R(i,j)+A(i,j);R(i,i)=R(i,i)-omega*A(i,j);
               A(i,i)=A(i,i)+omega*A(i,j);A(i,j)=0;
            end
         end
      end
   end
end
L=tril(A,-1)+eye(n); U=triu(A);
```

由于实际应用中一般不需要剩余矩阵 $\boldsymbol{R}$ 的信息, 为了节省存储量, 可以不对其计算及保存, 这只需在上述算法程序中将涉及计算 $\boldsymbol{R}$ 的有关语句去掉即可.

例 6.7 利用程序 milu.m 对矩阵 $\boldsymbol{A}$ 进行松弛不完全 LU 分解, 其中

$$\boldsymbol{A}=\begin{bmatrix} 4 & 0 & 0 & 1 & 0 \\ 0 & 3 & 2 & 0 & 1 \\ 8 & 0 & 2 & 0 & 0 \\ 0 & 9 & 0 & 5 & 0 \\ 0 & 0 & 2 & 0 & 6 \end{bmatrix},$$

$$\mathcal{F}=\{(1,1),(1,4),(2,2),(2,3),(2,5),(3,1),(3,3),(4,2),(4,4),(5,3),(5,5)\},\quad \omega=0.$$

解 编写 MATLAB 脚本文件 ex67.m, 在命令窗口运行之, 可得相应的计算结果 (略).

6.3 对称正定方程组的直接法

前面讨论的 Gauss 消去法和 LU 分解法, 都是求解一般方程组的方法, 它们均不考虑方程组系数矩阵本身的特点. 但在实际应用中经常会遇到一些特殊类型的方程组,

其系数矩阵具有某种特殊性, 如对称正定矩阵、稀疏 (带状) 矩阵等. 对于这些方程组, 若还用原有的一般方法来求解, 势必造成存储空间和计算的浪费. 因此, 有必要构造适合特殊方程组的求解方法. 本节主要介绍解对称正定方程组的 Cholesky 分解法.

6.3.1 Cholesky 分解法

当线性方程组的系数矩阵 $\boldsymbol{A}$ 是对称正定矩阵时, 可利用对称正定的特点使 LU 分解减少计算量, 从而节省存储空间. 由于对称正定矩阵的所有顺序主子式都大于零, 故由定理 6.2 可知 $\boldsymbol{A}$ 存在唯一的 LU 分解. 由于 $\boldsymbol{A}$ 是对称的, 即 $a_{ij}=a_{ji}$, $i,j=1,2,\cdots,n$. 由 LU 分解式 (6.12)~式 (6.14), 有

$$u_{1i}=a_{1i},\quad i=1,\cdots,n;\quad l_{i1}=\frac{a_{i1}}{a_{11}},\quad i=2,\cdots,n,$$

则

$$l_{i1}=\frac{a_{i1}}{a_{11}}=\frac{a_{1i}}{a_{11}}=\frac{u_{1i}}{u_{11}},\quad i=2,\cdots,n. \tag{6.19}$$

若已求得第 1 步到第 $k-1$ 步的 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的元素有如下关系, 即

$$l_{ij}=\frac{u_{ji}}{u_{jj}},\quad j=1,\cdots,k-1;\ i=j+1,\cdots,n, \tag{6.20}$$

则对于第 k 步, 由式 (6.13)、式 (6.14) 和式 (6.20), 得

$$u_{ki}=a_{ki}-\sum_{r=1}^{k-1}l_{kr}u_{ri}=a_{ki}-\sum_{r=1}^{k-1}\frac{u_{rk}u_{ri}}{u_{rr}},\quad i=k,\cdots,n;$$

$$\begin{aligned}l_{ik}&=\Big(a_{ik}-\sum_{r=1}^{k-1}l_{ir}u_{rk}\Big)\Big/u_{kk}\\&=\Big(a_{ik}-\sum_{r=1}^{k-1}\frac{u_{rk}u_{ri}}{u_{rr}}\Big)\Big/u_{kk}=\frac{u_{ki}}{u_{kk}},\quad i=k+1,\cdots,n.\end{aligned}$$

由此, 得

$$l_{ik}=\frac{u_{ki}}{u_{kk}},\quad k=1,\cdots,n-1;\ i=k+1,\cdots,n. \tag{6.21}$$

这样, 利用式 (6.21) 计算 $\boldsymbol{L}$ 的元素可节省工作量, 计算量节省了将近一半, 而 $\boldsymbol{U}$ 的元素仍用式 (6.13) 计算:

$$u_{ki}=a_{ki}-\sum_{r=1}^{k-1}l_{kr}u_{ri},\quad k=2,\cdots,n;\ i=k,\cdots,n.$$

注 6.2 由式 (6.21), 得

$$u_{ki}=u_{kk}l_{ik},\ k=1,\cdots,n-1;\ i=k+1,\cdots,n,$$

此即

$$\boldsymbol{U}=\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}},$$

式中: $\boldsymbol{D}$ 为以 $u_{kk}\,(k=1,\cdots,n)$ 为对角元的对角矩阵. 这样, 就把对称正定矩阵 $\boldsymbol{A}$ 分解成了

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}=\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$$

的形式. 这种分解方法称为 Cholesky 分解法.

下面建立用 Cholesky 分解法求解对称正定方程组的算法步骤.

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}\Longrightarrow\begin{cases}\boldsymbol{A}=\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}},\\ \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{b},\end{cases}\Longrightarrow\begin{cases}\boldsymbol{L}\boldsymbol{y}=\boldsymbol{b},\\ \boldsymbol{D}\boldsymbol{z}=\boldsymbol{y},\\ \boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{z}.\end{cases}\tag{6.22}$$

算法 6.6 (Cholesky 分解法)

步 1, 输入对称正定矩阵 $\boldsymbol{A}$ 和右端向量 $\boldsymbol{b}$.

步 2, Cholesky 分解:

$u_{1i}=a_{1i},\quad i=1,\cdots,n;$

$l_{i1}=u_{1i}/u_{11},\quad i=2,\cdots,n.$

对 $k=2,\cdots,n$, 计算

$$u_{ki}=a_{ki}-\sum_{r=1}^{k-1}l_{kr}u_{ri},\quad i=k,\cdots,n;$$

$$l_{ik}=u_{ki}/u_{kk},\quad i=k+1,\cdots,n.$$

步 3, 用向前消去法解下三角方程组 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{b}$:

$y_1=b_1,$

对 $k=2,\cdots,n$, 计算 $y_k=b_k-\sum\limits_{i=1}^{k-1}l_{ki}y_i.$

步 4, 解对角形方程组 $\boldsymbol{D}\boldsymbol{z}=\boldsymbol{y}$:

对 $k=1,\cdots,n$, 计算 $z_k=y_k/d_k.$

步 5, 用回代法解上三角方程组 $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{z}$:

$x_n=z_n,$

对 $k=n-1,\cdots,1$, 计算 $x_k=z_k-\sum\limits_{i=k+1}^{n}l_{ik}x_i.$

根据算法 6.6, 编制 MATLAB 程序如下:

```
%Cholesky分解法程序-mschol.m
function [x,L,D]=mschol(A,b)
%用Cholesky分解法解对称正定方程组Ax=b
%输入:系数矩阵A,右端项b
%输出:解向量x,单位下三角阵L,对角阵D
n=size(A,1); D=zeros(1,n); L=eye(n,n);
U(1,:)=A(1,:); L(2:n,1)=U(1,2:n)/U(1,1);  %Cholesky分解
for k=2:n
   U(k,k:n)=A(k,k:n)-L(k,1:k-1)*U(1:k-1,k:n);
```

```
    L(k+1:n,k)=U(k,k+1:n)/U(k,k);
end
%求解下三角方程组Ly=b(向前消去法)
y=zeros(n,1); y(1)=b(1);
for k=2:n,
    y(k)=b(k)-L(k,1:k-1)*y([1:k-1]);
end
%求解对角方程组Dz=y
D=diag(diag(U));
for k=1:n,
    z(k)=y(k)/D(k,k);
end
%求解上三角方程组L'x=z(回代法)
x=zeros(n,1); U=L'; x(n)=z(n);
for k=(n-1):-1:1,
    x(k)=z(k)-U(k,k+1:n)*x(k+1:n);
end
```

例 6.8 利用程序 mschol.m 计算下列对称正定方程组的解

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & 2 & -3 \\ -1 & -3 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -3 \\ 2 \end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[1,1,-1; 1,2,-3; -1,-3,3];
>> b=[0,-3,2]';
>> [x,L,D]=mschol(A,b)
```

可得计算结果 (略).

6.3.2 不完全 Cholesky 分解

在预处理共轭梯度法中, 往往需要对称正定矩阵 $\boldsymbol{A}$ 的不完全 Cholesky 分解 $\boldsymbol{A} \approx \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$ 来获得预处理子 $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$. 由于 $\boldsymbol{A}$ 的对称性, 此时, 假定指标集 $\mathcal{F}$ 也是对称正定的, 即若 $(i,j) \in \mathcal{F}$, 则必有 $(j,i) \in \mathcal{F}$. 相对于非对称情形的松弛 ILU 分解, 可以考虑 $\boldsymbol{A}$ 关于 $\mathcal{F}$ 和 ω 的松弛不完全 Cholesky 分解 (简称 RIC 分解):

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}} + \boldsymbol{R}, \tag{6.23}$$

式中: $\boldsymbol{L}$ 为单位下三角矩阵且满足式 (6.15); $\boldsymbol{R}$ 为剩余矩阵并满足式 (6.17) 和式 (6.18); $\boldsymbol{D}$ 为对角矩阵.

易知, 此时 $\boldsymbol{R}$ 也是对称的, 而且也可以采用前面介绍的修正 Gauss 消去法实现分解 (6.23), 即先用修正 Gauss 消去法求 $\boldsymbol{A}$ 的 RILU 分解

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U} + \boldsymbol{R},$$

然后取 $\boldsymbol{D} = \mathrm{diag}(u_{11}, u_{22}, \cdots, u_{nn})$ 为 $\boldsymbol{U}$ 的对角元素所构成的对角矩阵. 这样便得到了式 (6.23) 中的 $\boldsymbol{L}$, $\boldsymbol{D}$ 和 $\boldsymbol{R}$.

下面的定理给出了对称正定矩阵 $\boldsymbol{A}$ 关于 $\mathcal{F}$ 和 ω 的松弛不完全 Cholesky 分解 (6.23) 中 $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$ 正定的一个充分条件.

可以在算法 6.5 的基础上, 利用 $\boldsymbol{A}$ 的对称性给出求 $\boldsymbol{A}$ 的 RIC 分解的算法. 也可从 Cholesky 分解出发稍加修正求得. 下面是不完全 Cholesky 分解的一个简易可行的算法.

算法 6.7 (不完全 Cholesky 分解法) 给定对称正定矩阵 $\boldsymbol{A}$. 下面的算法计算 $\boldsymbol{A}$ 的不完全 Cholesky 分解: $\boldsymbol{A} \approx \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$.

$n = \mathrm{size}(\boldsymbol{A}, 1)$;
for $k = 1 : n$
  $l_{kk} = \left(a_{kk} - \sum_{r=1}^{k-1} l_{kr}^2\right)^{1/2}$;
  for $i = k+1 : n$
    if $(a_{ik} = 0)$
      $l_{ik} = 0$;
    else
      $l_{ik} = \left(a_{ik} - \sum_{r=1}^{k-1} l_{ir} l_{kr}\right) \Big/ l_{kk}$;
    end
  end
end

显然, 由算法 6.7 得到的不完全分解 $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 与 $\boldsymbol{A}$ 有相同的稀疏性, 但这一算法并不总是稳定的 (当 l_{kk} 很小的时候, $k = 1, 2, \cdots, n$).

算法 6.7 的 MATLAB 程序如下:

```
%不完全Cholesky分解程序-michol.m
function [L]=michol(A)
%输入:对称正定矩阵A
%输出:下三角矩阵L,满足A=LL'+R
n=size(A,1); L=zeros(n,n);
L(1,1)=sqrt(A(1,1)); L(2,1)=A(2,1)/L(1,1);
for k=2:n
    s=0;
    for(p=1:k-1),s=s+L(k,p)^2;end
    L(k,k)=sqrt(A(k,k)-s);
```

```
    for i=k+1:n
        if (A(i,k)~=0)
            s=0;
            for (p=1:k-1),s=s+L(i,p)*L(k,p); end
            L(i,k)=(A(i,k)-s)/L(k,k);
        end
    end
end
```

例 6.9 利用程序 michol.m 对矩阵 $\boldsymbol{A}$ 进行松弛不完全 Cholesky 分解, 其中

$$\boldsymbol{A}=\begin{bmatrix} 8 & -2 & -1 & 0 & 0 & 1 \\ -2 & 8 & -2 & 0 & 3 & 0 \\ -1 & -2 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & -2 & -1 \\ 0 & 3 & 0 & -2 & 8 & -2 \\ 1 & 0 & 0 & -1 & -2 & 8 \end{bmatrix}.$$

解 编写 MATLAB 脚本文件 ex69.m, 在命令窗口运行之, 可得相应的计算结果 (略).

6.4 带状线性方程组的直接法

6.4.1 三对角方程组

在科学与工程计算中, 经常遇到求解三对角方程组的问题. 例如, 三次样条插值计算, 以及用有限差分法求解二阶常系数线性常微分方程的边值问题和热传导问题, 经常需要求解三对角方程组 (即系数矩阵为三对角矩阵的方程组). 三对角矩阵属于所谓的“带状矩阵”, 在大多数应用中, 带状矩阵是严格对角占优的或正定的. 下面给出带状矩阵的定义.

定义 6.3 n 阶矩阵称为带状矩阵, 如果存在正整数 $p, q\,(1<p,q<n)$, 当 $j\geqslant i+p$ 或 $i\geqslant j+q$ 时, 有 $a_{ij}=0$, 并称 $w=p+q-1$ 为该带状矩阵的“带宽”. n 阶矩阵称为带状矩阵, 如果存在正整数 $p, q\,(1<p,q<n)$, 当 $i+p\leqslant j$ 或 $j+q\leqslant i$ 时, 有 $a_{ij}=0$, 并称 $w=p+q-1$ 为该带状矩阵的“带宽”.

三对角方程组的一般形式是

$$\boldsymbol{A}\boldsymbol{x}:=\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}=\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}:=\boldsymbol{f}. \tag{6.24}$$

显然三对角矩阵的带宽为 3. 本节介绍求解方程组 (6.24) 的追赶法和变参数追赶法.

1. 追赶法

将顺序 LU 分解法应用于三对角方程组得到所谓的“追赶法”. 事实上, 一方面, 可将三对角矩阵 $\boldsymbol{A}$ 分解为

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}, \tag{6.25}$$

式中:

$$\boldsymbol{L}=\begin{bmatrix} l_1 & & & & \\ a_2 & l_2 & & & \\ & \ddots & \ddots & & \\ & & a_{n-1} & l_{n-1} & \\ & & & a_n & l_n \end{bmatrix},\quad \boldsymbol{U}=\begin{bmatrix} 1 & u_1 & & & \\ & 1 & u_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & u_{n-1} \\ & & & & 1 \end{bmatrix}.$$

比较式 (6.25) 两端的对应元素, 得

$$b_1=l_1,\ c_{k-1}=l_{k-1}u_{k-1},\ b_k=a_ku_{k-1}+l_k,\ k=2,\cdots,n.$$

于是有

$$l_1=b_1,\ u_{k-1}=\frac{c_{k-1}}{l_{k-1}},\ l_k=b_k-a_ku_{k-1},\ k=2,\cdots,n. \tag{6.26}$$

另一方面, 方程组 (6.24) 等价于求解 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{f}$ 和 $\boldsymbol{U}\boldsymbol{x}=\boldsymbol{y}$, 其中 $\boldsymbol{y}=(y_1,y_2,\cdots,y_n)^{\mathrm{T}}$. 分别比较 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{f}$ 和 $\boldsymbol{U}\boldsymbol{x}=\boldsymbol{y}$ 两端的对应元素, 得

$$\begin{cases} l_1y_1=f_1,\quad a_ky_{k-1}+l_ky_k=f_k,\quad k=2,3,\cdots,n, \\ x_k+u_kx_{k+1}=y_k,\quad k=1,2,\cdots,n-1,\quad x_n=y_n. \end{cases} \tag{6.27}$$

结合式 (6.26) 和式 (6.27) 可得下面的算法.

算法 6.8 (追赶法) 计算三对角方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}$ 的解.

$l_1=b_1;\ y_1=\dfrac{f_1}{l_1}$;

for $k=2:n$

$\quad u_{k-1}=\dfrac{c_{k-1}}{l_{k-1}}$;

$\quad l_k=b_k-a_ku_{k-1}$;

$\quad y_k=\dfrac{f_k-a_ky_{k-1}}{l_k}$;

end

$x_n=y_n$;

for $k=n-1:-1:1$

$\quad x_k=y_k-u_kx_{k+1}$;

end

算法 6.8 被称为“追赶法”的原因: 一是第 1 步关于指标 k 由小到大计算 l_k, u_k 和 y_k 这三个量, 这是“向前追”的过程; 二是第 2 步关于指标 k 从大到小计算方程组的解 x_k, 此即“往回赶”的过程. 追赶法只有 $5n-4$ 次乘除法运算和 $3n-3$ 次加减法运算, 且当系数矩阵对角占优时数值稳定, 是解三对角方程组的优秀算法. 编程计算时, 可将 l_k, u_k 依次存放在 b_k, c_k 的位置, 而将 y_k 和 x_k 先后存放在 f_k 的位置, 因此整个计算过程只需 $4n$ 个存储单元.

追赶法的 MATLAB 程序如下:

```
%追赶法程序-mchase.m
function [f]=mchase(a,b,c,f)
%用追赶法解三对角方程组Ax=f
%输入:a为A的下对角线,b为A的主对角线,c为A的次上对角线,f为右端向量
%输出:解向量f(LU分解中的l(k),u(k)存放在b(k),c(k)的位置,
%      y(k)和x(k)先后存放在f(k)的位置)
n=length(b); f(1)=f(1)/b(1);
for k=2:n
    c(k-1)=c(k-1)/b(k-1);
    b(k)=b(k)-a(k)*c(k-1);
    f(k)=(f(k)-a(k)*f(k-1))/b(k);
end
for k=n-1:-1:1
    f(k)=f(k)-c(k)*f(k+1);
end
```

定理 6.5 *若方程组 (6.24) 的系数矩阵的元素满足条件*

$$|b_1|>|c_1|>0,\ |b_n|>|a_n|>0,\ |b_i|>|a_i|+|c_i|,\quad i=2,\cdots,n-1,$$

则追赶法是可行的.

证明 由式 (6.26) 和式 (6.27) 可知, 只需证明 $l_k\neq 0\,(k=1,2,\cdots,n)$ 即可. 显然 $l_1=b_1\neq 0$, 且 $|l_1|=|b_1|>|c_1|$. 设 $|l_{k-1}|>|c_{k-1}|$, 则

$$\begin{aligned}|l_k|=|b_k-a_ku_{k-1}|&=\left|b_k-\frac{a_k}{l_{k-1}}c_{k-1}\right|\\&\geqslant |b_k|-|a_k|\cdot\left|\frac{c_{k-1}}{l_{k-1}}\right|>|b_k|-|a_k|\\&>\begin{cases}|c_k|, & k<n,\\ 0, & k=n,\end{cases}\end{aligned}$$

即 $l_k\neq 0$, $k=2,\cdots,n$. 从而, 追赶法是可行的. 证毕. □

注 6.3 *满足定理 6.5 条件的三对角矩阵即为严格对角占优的, 例 6.5 说明对于严格对角占优的三对角矩阵, 追赶法总是可行的.*

2. 变参数追赶法

需要指出的是, 追赶法的本质是没有选取主元的 Gauss 消去法, 因此对于一般的三对角矩阵, 不一定存在如式 (6.25) 的三角分解. 或者三角分解可以进行下去, 但计算过程不是数值稳定的, 因而最终得到的解并不可靠. 这就促使考虑对追赶法进行改进和修正.

将三对角矩阵 $\boldsymbol{A}$ 分解为

$$\boldsymbol{A}=\boldsymbol{D}\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{U}}, \tag{6.28}$$

式中: $\boldsymbol{D}=\operatorname{diag}(d_1,d_2,\cdots,d_n)$,

$$\widetilde{\boldsymbol{L}}=\begin{bmatrix} l_1 & 1 & & & \\ & l_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{bmatrix}_{n\times(n+1)},\quad \widetilde{\boldsymbol{U}}=\begin{bmatrix} u_1 & & & & \\ 1 & u_2 & & & \\ & 1 & \ddots & & \\ & & \ddots & u_n & \\ & & & 1 & \end{bmatrix}_{(n+1)\times n}.$$

比较式 (6.25) 两端的元素, 得

$$\begin{cases} b_k=d_k(l_ku_k+1),\ k=1,2,\cdots,n; \\ a_k=d_kl_k,\ c_{k-1}=d_{k-1}u_k,\ k=2,\cdots,n. \end{cases}$$

选取 l_1, u_1 使得 $l_1u_1+1\neq 0$, 则有

$$\begin{cases} d_1=\dfrac{b_1}{l_1u_1+1}, \\ u_k=\dfrac{c_{k-1}}{d_{k-1}},\ d_k=b_k-a_ku_k,\ l_k=\dfrac{a_k}{d_k}, \\ \qquad (k=2,3,\cdots,n). \end{cases}$$

注意到, 求解 $\boldsymbol{Ax}=\boldsymbol{f}$ 等价于求解 $\widetilde{\boldsymbol{L}}\boldsymbol{y}=\boldsymbol{D}^{-1}\boldsymbol{f}:=\boldsymbol{g}$ 和 $\widetilde{\boldsymbol{U}}\boldsymbol{x}=\boldsymbol{y}$, 其中

$$\boldsymbol{y}=(y_1,y_2,\cdots,y_{n+1})^{\mathrm{T}},\quad \boldsymbol{g}=\left(\frac{f_1}{d_1},\frac{f_2}{d_2},\cdots,\frac{f_n}{d_n}\right)^{\mathrm{T}}.$$

比较 $\widetilde{\boldsymbol{L}}\boldsymbol{y}=\boldsymbol{g}$ 和 $\widetilde{\boldsymbol{U}}\boldsymbol{x}=\boldsymbol{y}$ 两端的元素, 得

$$l_ky_k+y_{k+1}=g_k,\ k=1,2,\cdots,n; \tag{6.29}$$

$$\begin{cases} u_1x_1=y_1, \\ x_k+u_{k+1}x_{k+1}=y_{k+1},\ k=1,2,\cdots,n-1, \\ x_n=y_{n+1}. \end{cases} \tag{6.30}$$

由式 (6.29) 和式 (6.30) 可以发现, 只要计算出 x_1, 其他的可以依次计算出来. 下面推导计算 x_1 的公式.

由式 (6.30), 得

$$\begin{aligned}x_1 &= y_2 - u_2x_2 = y_2 - u_2(y_3 - u_3x_3)\\ &= y_2 + (-u_2)y_3 + (-u_2)(-u_3)x_3\\ &= \cdots\\ &= y_2 + (-u_2)y_3 + (-u_2)(-u_3)y_4\\ &\quad + \cdots + (-u_2)(-u_3)\cdots(-u_{n-1})y_n + (-u_2)(-u_3)\cdots(-u_n)y_{n+1}.\end{aligned} \tag{6.31}$$

由式 (6.29), 得

$$\begin{aligned}y_{k+1} &= g_k - l_ky_k = g_k - l_k(g_{k-1} + l_{k-1}y_{k-1})\\ &= g_k + (-l_k)g_{k-1} + (-l_k)(-l_{k-1})y_{k-1}\\ &= \cdots\\ &= g_k + (-l_k)g_{k-1} + (-l_k)(-l_{k-1})g_{k-2}\\ &\quad + \cdots + (-l_k)(-l_{k-1})\cdots(-l_2)g_1\\ &\quad + (-l_k)(-l_{k-1})\cdots(-l_1)(u_1x_1),\ k = 1, 2, \cdots, n.\end{aligned} \tag{6.32}$$

将式 (6.32) 代入式 (6.31) 并整理, 得

$$\begin{aligned}&[1 + u_1l_1 + (u_1u_2)(l_2l_1) + \cdots + (u_1\cdots u_n)(l_n\cdots l_1)]x_1\\ &= [1 + u_2l_2 + \cdots + (u_2\cdots u_n)(l_n\cdots l_2)]g_1\\ &\quad + (-u_2)[1 + u_3l_3 + \cdots + (u_3\cdots u_n)(l_n\cdots l_3)]g_2\\ &\quad + \cdots + [(-u_2)\cdots(-u_{n-1})](1 + u_nl_n)g_{n-1} + [(-u_2)\cdots(-u_n)]g_n.\end{aligned}$$

令

$$\begin{aligned}&s_k = 1 + u_kl_k + \cdots + (u_k\cdots u_n)(l_n\cdots l_k),\ k = 1, 2, \cdots, n,\\ &t_1 = s_2g_1 + (-u_2)s_3g_2 + \cdots + [(-u_2)\cdots(-u_{n-1})]s_ng_{n-1} + [(-u_2)\cdots(-u_n)]g_n.\end{aligned}$$

则

$$x_1 = \frac{t_1}{s_1},$$

且有递推公式

$$\begin{aligned}&s_n = 1 + u_nl_n,\ \ s_k = 1 + u_ks_{k+1}l_k,\ \ k = n-1, n-2, \cdots, 1,\\ &t_n = g_n,\ \ t_k = s_{k+1}g_k - u_{k+1}t_{k+1},\ \ k = n-1, n-2, \cdots, 1.\end{aligned}$$

综合上述, 可得下面的变参数追赶法.

算法 6.9 (变参数追赶法) 计算三对角方程组 $\boldsymbol{Ax} = \boldsymbol{f}$ 的解. 选取 l_1 和 u_1 使得 $l_1u_1 + 1 \neq 0$.

$d_1 = \dfrac{b_1}{l_1 u_1 + 1}$; $g_1 = \dfrac{f_1}{d_1}$;

for $k = 2 : n$

$\quad u_k = \dfrac{c_{k-1}}{d_{k-1}}$; $d_k = b_k - a_k u_k$; $l_k = \dfrac{a_k}{d_k}$; $g_k = \dfrac{f_k}{d_k}$;

end

$s_n = 1 + u_n l_n$; $t_n = g_n$;

for $k = n - 1 : -1 : 1$

$\quad s_k = 1 + u_k s_{k+1} l_k$; $t_k = s_{k+1} g_k - u_{k+1} t_{k+1}$;

end

$x_1 = \dfrac{t_1}{s_1}$; $y_1 = u_1 x_1$;

for $k = 1 : n$

$\quad y_{k+1} = g_k - l_k y_k$;

end

$x_n = y_{n+1}$;

for $k = n - 1 : -1 : 2$

$\quad x_k = y_{k+1} - u_{k+1} x_{k+1}$;

end

算法 6.9 需 $10n - 8$ 次乘除运算, $5n - 5$ 次加减运算. 由于算法中的 l_1 和 u_1 的选取只需使得 $l_1 u_1 + 1 \neq 0$, 可以认为它们是两个可变的参数, 故称为“变参数追赶法”.

变参数追赶法的 MATLAB 程序如下:

```
%变参数追赶法程序-mchase_var.m
function [x]=mchase_var(a,b,c,f,l1,u1)
%用变参数追赶法解三对角方程组Ax=f
%输入:a为A的下对角线,b为A的主对角线,c为A的次
%      上对角线,f为右端向量;l1,u1满足l1*u1+1不为0.
%输出:解向量x
n=length(b);x=zeros(n,1);l=zeros(n,1);
u=zeros(n,1);l(1)=l1; u(1)=u1;
d(1)=b(1)/(l(1)*u(1)+1); g(1)=f(1)/d(1);
for k=2:n
    u(k)=c(k-1)/d(k-1);d(k)=b(k)-a(k)*u(k);
    l(k)=a(k)/d(k);g(k)=f(k)/d(k);
end
s(n)=1+u(n)*l(n); t(n)=g(n);
for k=n-1:-1:1
    s(k)=1+u(k)*s(k+1)*l(k);
    t(k)=s(k+1)*g(k)-u(k+1)*t(k+1);
```

```
end
x(1)=t(1)/s(1);y(1)=u(1)*x(1);
for k=1:n
    y(k+1)=g(k)-l(k)*y(k);
end
x(n)=y(n+1);
for k=n-1:-1:2
    x(k)=y(k+1)-u(k+1)*x(k+1);
end
```

例 6.10 设

$$\boldsymbol{A}=\begin{bmatrix} 4 & -1 & & & \\ -2 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -2 & 4 & -1 \\ & & & -2 & 4 \end{bmatrix}, \quad \boldsymbol{f}=\begin{bmatrix} 3 \\ 1 \\ \vdots \\ 1 \\ 2 \end{bmatrix}.$$

用追赶法和变参数追赶法 (取 $l_1=u_1=1$) 求解三对角方程组 $\boldsymbol{Ax}=\boldsymbol{f}$, 并计算实际误差 $\|\boldsymbol{f}-\boldsymbol{Ax}\|_2$.

解 利用程序 mchase.m 和 mchase_var.m, 编写 MATLAB 脚本文件 ex610.m, 并对不同的维数 n, 在命令窗口运行该程序, 得到数值结果如表 6.1 所示.

表 6.1 追赶法和变参数追赶法的数值结果

维数	追赶法		变参数追赶法	
	误差	计算时间	误差	计算时间
1024	7.1650e-15	0.0056	1.2212e-15	0.0103
2048	1.0091e-14	0.0058	1.2212e-15	0.0106
4096	1.4241e-14	0.0064	1.2212e-15	0.0109
8192	2.0118e-14	0.0066	1.2212e-15	0.0128

例 6.11 设

$$\boldsymbol{A}=\begin{bmatrix} 2 & 3 & & & \\ 6 & 9 & 3 & & \\ & \ddots & \ddots & \ddots & \\ & & 6 & 9 & 3 \\ & & & 6 & 9 \end{bmatrix}, \quad \boldsymbol{f}=\begin{bmatrix} 5 \\ 10 \\ \vdots \\ 10 \\ 12 \end{bmatrix}.$$

试用追赶法和变参数追赶法 (取 $l_1=u_1=1$) 求解三对角方程组 $\boldsymbol{Ax}=\boldsymbol{f}$, 并计算实际误差 $\|\boldsymbol{f}-\boldsymbol{Ax}\|_2$.

解 对于此例, 追赶法已经失效, 但变参数追赶法仍然非常有效, 数值结果 (运行 MATLAB 脚本文件 ex611.m) 如表 6.2 所示.

表 6.2 变参数追赶法的数值结果

维数	追赶法	变参数追赶法	
		误差	计算时间
1024	失效	5.7293e-14	0.0101
4096	失效	1.1391e-13	0.0111

6.4.2 块三对角方程组

用有限差分法五点格式离散二维 Poisson 方程, 得到的代数方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 的系数矩阵是一个块三对角矩阵, 即

$$\begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{C}_1 & & & \\ \boldsymbol{A}_2 & \boldsymbol{B}_2 & \boldsymbol{C}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{A}_{m-1} & \boldsymbol{B}_{m-1} & \boldsymbol{C}_{m-1} \\ & & & \boldsymbol{A}_m & \boldsymbol{B}_m \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{m-1} \\ \boldsymbol{x}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_{m-1} \\ \boldsymbol{f}_m \end{bmatrix}, \tag{6.33}$$

式中: $\boldsymbol{A}_k, \boldsymbol{B}_k, \boldsymbol{C}_k$ 均为 r 阶方阵; $\boldsymbol{x}_k, \boldsymbol{f}_k$ 为 r 维列向量. 当 $\boldsymbol{A}$ 可逆时, 方程组有唯一解 $\boldsymbol{x}^*$. 本节介绍求解方程组 (6.33) 的块追赶法和双参数法.

1. 块追赶法

将三对角方程矩阵 $\boldsymbol{A}$ 分解为 $\boldsymbol{A}=\boldsymbol{LU}$, 其中

$$\boldsymbol{L}=\begin{bmatrix} \boldsymbol{L}_1 & & & & \\ \boldsymbol{A}_2 & \boldsymbol{L}_2 & & & \\ & \ddots & \ddots & & \\ & & \boldsymbol{A}_{m-1} & \boldsymbol{L}_{m-1} & \\ & & & \boldsymbol{A}_m & \boldsymbol{L}_m \end{bmatrix}, \quad \boldsymbol{U}=\begin{bmatrix} \boldsymbol{I} & \boldsymbol{U}_1 & & & \\ & \boldsymbol{I} & \boldsymbol{U}_2 & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{I} & \boldsymbol{U}_{m-1} \\ & & & & \boldsymbol{I} \end{bmatrix}.$$

比较 $\boldsymbol{A}=\boldsymbol{LU}$ 两端的对应的子矩阵, 得

$$\boldsymbol{B}_1=\boldsymbol{L}_1,\ \boldsymbol{C}_{k-1}=\boldsymbol{L}_{k-1}\boldsymbol{U}_{k-1},\ \boldsymbol{B}_k=\boldsymbol{A}_k\boldsymbol{U}_{k-1}+\boldsymbol{L}_k,\ k=2,\cdots,n.$$

于是有

$$\boldsymbol{L}_1=\boldsymbol{B}_1,\ \boldsymbol{U}_{k-1}=\boldsymbol{L}_{k-1}^{-1}\boldsymbol{C}_{k-1},\ \boldsymbol{L}_k=\boldsymbol{B}_k-\boldsymbol{A}_k\boldsymbol{U}_{k-1},\ k=2,\cdots,n. \tag{6.34}$$

求解三对角方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 等价于求解

$$\boldsymbol{Ly}=\boldsymbol{f} \quad 和 \quad \boldsymbol{Ux}=\boldsymbol{y},$$

式中: $\boldsymbol{y}=\left[\boldsymbol{y}_1^{\mathrm{T}},\boldsymbol{y}_2^{\mathrm{T}},\cdots,\boldsymbol{y}_m^{\mathrm{T}}\right]^{\mathrm{T}}$. 分别比较 $\boldsymbol{Ly}=\boldsymbol{f}$ 和 $\boldsymbol{Ux}=\boldsymbol{y}$ 两端的对应元素, 得

$$\begin{cases} \boldsymbol{L}_1\boldsymbol{y}_1=\boldsymbol{f}_1, \quad \boldsymbol{A}_k\boldsymbol{y}_{k-1}+\boldsymbol{L}_k\boldsymbol{y}_k=\boldsymbol{f}_k, \quad k=2,3,\cdots,m, \\ \boldsymbol{x}_k+\boldsymbol{U}_k\boldsymbol{x}_{k+1}=\boldsymbol{y}_k, \quad k=1,2,\cdots,m-1, \quad \boldsymbol{x}_m=\boldsymbol{y}_m. \end{cases} \tag{6.35}$$

结合式 (6.34) 和式 (6.35) 可得下面的算法.

算法 6.10 (块追赶法) 计算块三对角方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 的解.

$\boldsymbol{L}_1=\boldsymbol{B}_1$; $\boldsymbol{y}_1=\boldsymbol{L}_1^{-1}\boldsymbol{f}_1$;

for $k=2:m$

$\quad \boldsymbol{U}_{k-1}=\boldsymbol{L}_{k-1}^{-1}\boldsymbol{C}_{k-1}$; $\boldsymbol{L}_k=\boldsymbol{B}_k-\boldsymbol{A}_k\boldsymbol{U}_{k-1}$; $\boldsymbol{y}_k=\boldsymbol{L}_k^{-1}(\boldsymbol{f}_k-\boldsymbol{A}_k\boldsymbol{y}_{k-1})$;

end

$\boldsymbol{x}_m=\boldsymbol{y}_m$;

for $k=m-1:-1:1$

$\quad \boldsymbol{x}_k=\boldsymbol{y}_k-\boldsymbol{U}_k\boldsymbol{x}_{k+1}$;

end

使用块追赶法求解三对角方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 需要的乘除法次数约为

$$mr^2\Big(\frac{10}{3}r+3\Big)-r^2(3r+2).$$

编程计算时, 可将 $\boldsymbol{L}_k$ 和 $\boldsymbol{U}_k$ 依次存放在 $\boldsymbol{B}_k$ 和 $\boldsymbol{C}_k$ 的位置, 而将 $\boldsymbol{y}_k$ 和 $\boldsymbol{x}_k$ 先后存放在 $\boldsymbol{f}_k$ 的位置.

块追赶法的 MATLAB 程序如下:

```
%块追赶法程序-mchase_block.m
function [fi]=mchase_block(Ai,Bi,Ci,fi,m)
%用块追赶法解块三对角方程组Ax=f
%输入:Ai为A的次下对角块,Bi为A的主对角块,
%     Ci为A的次上对角块,fi为右端向量,m为分块数
%输出:解向量fi,其中LU分解的L{k},U{k}存放在Bi{k},
%     Ci{k}的位置,y{k}和x{k}先后存放在fi{k}的位置
fi{1}=Bi{1}\fi{1};
for k=2:m
    Ci{k-1}=Bi{k-1}\Ci{k-1};Bi{k}=Bi{k}-Ai{k}*Ci{k-1};
    fi{k}=Bi{k}\(fi{k}-Ai{k}*fi{k-1});
end
for k=m-1:-1:1
    fi{k}=fi{k}-Ci{k}*fi{k+1};
end
```

2. 双参数法

注意到块三角对方程组 (6.33) 的前 $m-1$ 个子方程为

$$\begin{cases} \boldsymbol{B}_1\boldsymbol{x}_1+\boldsymbol{C}_1\boldsymbol{x}_2=\boldsymbol{f}_1, \\ \boldsymbol{A}_2\boldsymbol{x}_1+\boldsymbol{B}_2\boldsymbol{x}_2+\boldsymbol{C}_2\boldsymbol{x}_3=\boldsymbol{f}_2, \\ \qquad\vdots \\ \boldsymbol{A}_{m-1}\boldsymbol{x}_{m-2}+\boldsymbol{B}_{m-1}\boldsymbol{x}_{m-1}+\boldsymbol{C}_{m-1}\boldsymbol{x}_m=\boldsymbol{f}_{m-1}. \end{cases}$$

将上式中的 $\boldsymbol{x}_2,\boldsymbol{x}_3,\cdots,\boldsymbol{x}_m$ 都用 $\boldsymbol{x}_1$ 表示出来, 即

$$\begin{aligned} \boldsymbol{x}_2&=\boldsymbol{C}_1^{-1}(\boldsymbol{f}_1-\boldsymbol{B}_1\boldsymbol{x}_1):=\boldsymbol{s}_2+\boldsymbol{T}_2\boldsymbol{x}_1, \\ \boldsymbol{x}_3&=\boldsymbol{C}_2^{-1}(\boldsymbol{f}_2-\boldsymbol{A}_2\boldsymbol{x}_1-\boldsymbol{B}_2\boldsymbol{x}_2) \\ &=\boldsymbol{C}_2^{-1}(\boldsymbol{f}_2-\boldsymbol{B}_2\boldsymbol{s}_2)-\boldsymbol{C}_2^{-1}(\boldsymbol{A}_2+\boldsymbol{B}_2\boldsymbol{T}_2)\boldsymbol{x}_1:=\boldsymbol{s}_3+\boldsymbol{T}_3\boldsymbol{x}_1, \\ &\vdots \\ \boldsymbol{x}_m&=\boldsymbol{C}_{m-1}^{-1}(\boldsymbol{f}_{m-1}-\boldsymbol{A}_{m-1}\boldsymbol{x}_{m-2}-\boldsymbol{B}_{m-1}\boldsymbol{x}_{m-1}):=\boldsymbol{s}_m+\boldsymbol{T}_m\boldsymbol{x}_1. \end{aligned}$$

记 $\boldsymbol{s}_1=\boldsymbol{0}$, $\boldsymbol{T}_1=\boldsymbol{I}_r$ (r 阶单位阵), 则 $\boldsymbol{x}_1=\boldsymbol{s}_1+\boldsymbol{T}_1\boldsymbol{x}_1$. 于是形式上有

$$\boldsymbol{x}_k=\boldsymbol{s}_k+\boldsymbol{T}_k\boldsymbol{x}_1,\quad k=1,2,\cdots,m. \tag{6.36}$$

下面推导参数序列 $\{\boldsymbol{s}_k\}$ 和 $\{\boldsymbol{T}_k\}$ 的递推计算公式. 注意到由第 k 个子方程 $\boldsymbol{A}_k\boldsymbol{x}_{k-1}+\boldsymbol{B}_k\boldsymbol{x}_k+\boldsymbol{C}_k\boldsymbol{x}_{k+1}=\boldsymbol{f}_k$ 可得

$$\begin{aligned} \boldsymbol{x}_{k+1}&=\boldsymbol{C}_k^{-1}(\boldsymbol{f}_k-\boldsymbol{A}_k\boldsymbol{x}_{k-1}-\boldsymbol{B}_k\boldsymbol{x}_k) \\ &=\boldsymbol{C}_k^{-1}(\boldsymbol{f}_k-\boldsymbol{A}_k\boldsymbol{s}_{k-1}-\boldsymbol{B}_k\boldsymbol{s}_k)-\boldsymbol{C}_k^{-1}(\boldsymbol{A}_k\boldsymbol{T}_{k-1}+\boldsymbol{B}_k\boldsymbol{T}_k)\boldsymbol{x}_1. \end{aligned}$$

将上式与式 (6.36) 比较, 得

$$\begin{aligned} &\boldsymbol{s}_1=\boldsymbol{0},\quad \boldsymbol{s}_2=\boldsymbol{C}_1^{-1}\boldsymbol{f}_1, \\ &\boldsymbol{s}_{k+1}=\boldsymbol{C}_k^{-1}(\boldsymbol{f}_k-\boldsymbol{A}_k\boldsymbol{s}_{k-1}-\boldsymbol{B}_k\boldsymbol{s}_k),\ k=2,\cdots,m-1, \\ &\boldsymbol{T}_1=\boldsymbol{I}_r,\quad \boldsymbol{T}_2=-\boldsymbol{C}_1^{-1}\boldsymbol{B}_1, \\ &\boldsymbol{T}_{k+1}=-\boldsymbol{C}_k^{-1}(\boldsymbol{A}_k\boldsymbol{T}_{k-1}+\boldsymbol{B}_k\boldsymbol{T}_k),\ k=2,\cdots,m-1. \end{aligned}$$

将式 (6.36) 代入式 (6.33) 的第 m 个子方程, 得

$$(\boldsymbol{A}_m\boldsymbol{T}_{m-1}+\boldsymbol{B}_m\boldsymbol{T}_m)\boldsymbol{x}_1=\boldsymbol{f}_m-\boldsymbol{A}_m\boldsymbol{s}_{m-1}-\boldsymbol{B}_m\boldsymbol{s}_m. \tag{6.37}$$

若 r 阶方程组 (6.37) 有解 $\boldsymbol{x}_1$, 则可由式 (6.36) 确定方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}$ 的唯一解. 称这种方法为解块三对角方程组的*双参数法*.

因为 $\boldsymbol{A}$ 可逆, 故方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}$ 有唯一解, 从而 $\boldsymbol{x}_1$ 存在. 假设方程组 (6.37) 有无穷多组解, 则对每一个解 $\boldsymbol{x}_1$, 由式 (6.36) 都可求得方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}$ 的一组解, 这与

方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 有唯一解矛盾. 因此 $\boldsymbol{x}_1$ 存在且唯一, 从而方程组 (6.37) 的系数矩阵 $\boldsymbol{A}_m\boldsymbol{T}_{m-1}+\boldsymbol{B}_m\boldsymbol{T}_m$ 非奇异, 且有

$$\boldsymbol{x}_1=(\boldsymbol{A}_m\boldsymbol{T}_{m-1}+\boldsymbol{B}_m\boldsymbol{T}_m)^{-1}(\boldsymbol{f}_m-\boldsymbol{A}_m\boldsymbol{s}_{m-1}-\boldsymbol{B}_m\boldsymbol{s}_m),$$

代入 (6.36) 即可求得方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 的唯一解. 算法如下.

算法 6.11 (双参数法) 计算块三对角方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 的解.

$\boldsymbol{s}_1=\boldsymbol{0}$; $\boldsymbol{s}_2=\boldsymbol{C}_1^{-1}\boldsymbol{f}_1$; $\boldsymbol{T}_1=\boldsymbol{I}_r$; $\boldsymbol{T}_2=-\boldsymbol{C}_1^{-1}\boldsymbol{B}_1$;

for $k=2:m-1$

$\quad\boldsymbol{s}_{k+1}=\boldsymbol{C}_k^{-1}(\boldsymbol{f}_k-\boldsymbol{A}_k\boldsymbol{s}_{k-1}-\boldsymbol{B}_k\boldsymbol{s}_k)$;

$\quad\boldsymbol{T}_{k+1}=-\boldsymbol{C}_k^{-1}(\boldsymbol{A}_k\boldsymbol{T}_{k-1}+\boldsymbol{B}_k\boldsymbol{T}_k)$;

end

$\boldsymbol{x}_1=(\boldsymbol{A}_m\boldsymbol{T}_{m-1}+\boldsymbol{B}_m\boldsymbol{T}_m)^{-1}(\boldsymbol{f}_m-\boldsymbol{A}_m\boldsymbol{s}_{m-1}-\boldsymbol{B}_m\boldsymbol{s}_m)$;

for $k=2:m$

$\quad\boldsymbol{x}_k=\boldsymbol{s}_k+\boldsymbol{T}_k\boldsymbol{x}_1$;

end

使用双参数法求解三对角方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 需要的乘除法次数约为

$$mr^2\Big(\frac{13}{3}r+4\Big)-r^2(4r+3).$$

双参数法的 MATLAB 程序如下:

```
%双参数法程序-mdouble_par.m
function [x]=mdouble_par(Ai,Bi,Ci,fi,m)
%用双参数法解三对角方程组Ax=f
%输入:Ai为A的次下对角块,Bi为A的主对角块,
%     Ci为A的次上对角块,fi为右端向量
%输出:解向量x
x=cell(m,1); s=cell(m,1); T=cell(m,1);
s{1}=zeros(3,1); s{2}=Ci{1}\fi{1};
T{1}=eye(3); T{2}=-Ci{1}\Bi{1};
for k=2:m-1
    s{k+1}=Ci{k}\(fi{k}-Ai{k}*s{k-1}-Bi{k}*s{k});
    T{k+1}=-Ci{k}\(Ai{k}*T{k-1}+Bi{k}*T{k});
end
x{1}=(Ai{m}*T{m-1}+Bi{m}*T{m})\(fi{m}-Ai{m}*s{m-1}-Bi{m}*s{m});
for k=2:m
    x{k}=s{k}+T{k}*x{1};
end
```

例 6.12 选取子矩阵分别为

$$\boldsymbol{B}_k=\begin{bmatrix}4&-1&0\\-1&4&-1\\0&-1&4\end{bmatrix},\quad \boldsymbol{A}_k=\boldsymbol{C}_k^{\mathrm{T}}=\begin{bmatrix}13&0&0\\0&11&0\\1&0&12\end{bmatrix},\ k=1,2,\cdots,m.$$

分别对 $m=10^3,5\times10^3,10^4,5\times10^4,10^5,5\times10^5$ 使用块追赶法和双参数法求解块三对角矩阵 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}$, 其中 $\boldsymbol{f}_k=(1,0,1)^{\mathrm{T}}$. 并将所得数值解 $\boldsymbol{x}$ 代入 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}$, 各子方程的误差为

$$\begin{aligned}\delta_1&=\|\boldsymbol{B}_1\boldsymbol{x}_1+\boldsymbol{C}_1\boldsymbol{x}_2-\boldsymbol{f}_1\|_2,\\ \delta_k&=\|\boldsymbol{A}_k\boldsymbol{x}_{k-1}+\boldsymbol{B}_k\boldsymbol{x}_k+\boldsymbol{C}_k\boldsymbol{x}_{k+1}-\boldsymbol{f}_k\|_2,\quad k=2,\cdots,m-1,\\ \delta_m&=\|\boldsymbol{A}_m\boldsymbol{x}_{m-1}+\boldsymbol{B}_m\boldsymbol{x}_m-\boldsymbol{f}_m\|_2.\end{aligned}$$

两种方法的计算时间 (PC Intel Core i5-3470 CPU 3.2GHz, MATLAB R2015b) 和各子方程的误差最大值 $\max\{\delta_k\}_{k=1}^m$ 如表 6.3 所示.

表 6.3 块追赶法和双参数法的数值结果

维数	块追赶法		双参数法	
	误差	计算时间	误差	计算时间
1×10^3	1.9956e-11	0.0319	5.5943e-16	0.0235
5×10^3	1.8646e-11	0.1277	7.0217e-16	0.0821
1×10^4	1.2669e-10	0.2503	4.9772e-16	0.1551
5×10^4	5.1618e-11	1.2301	8.8991e-16	0.7532
1×10^5	1.2903e-09	2.4382	8.9509e-16	1.4845
5×10^5	8.6523e-10	12.1268	6.2942e-16	7.4758

从表 6.3 可以看出, 对于此例, 双参数法要比块追赶法有效得多.

例 6.13 选取子矩阵分别为

$$\boldsymbol{B}_k=\begin{bmatrix}2&-1&0\\-2&1&0\\0&0&3\end{bmatrix},\quad \boldsymbol{A}_k=\boldsymbol{C}_k=\begin{bmatrix}2&0&0\\0&2&0\\0&0&2\end{bmatrix},\ k=1,2,\cdots,m.$$

分别对 $m=10^3,5\times10^3,10^4,5\times10^4,10^5,5\times10^5$ 使用块追赶法和双参数法求解块三对角矩阵 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}$, 其中 $\boldsymbol{f}_k=(1,2,1)^{\mathrm{T}}$. 两种方法的计算时间 (PC Intel Core i5-3470 CPU 3.2GHz, MATLAB R2015b) 和各子方程的误差最大值 $\max\{\delta_k\}_{k=1}^m$ 如表 6.4 所示.

从表 6.4 可以看出, 当块追赶法失效时, 双参数法依然十分有效. 当然, 也可以构造出块追赶法有效而双参数法失效的算例. 由此可见, 块追赶法和双参数法可以互为补充.

表 6.4 双参数法的数值结果

维数	块追赶法	双参数法	
		误差	计算时间
1×10^3	失效	4.4409e-16	0.0240
5×10^3	失效	4.4409e-16	0.0848
1×10^4	失效	5.5511e-16	0.1622
5×10^4	失效	4.4409e-16	0.7594
1×10^5	失效	6.6613e-16	1.5165
5×10^5	失效	5.5511e-16	7.4960

6.5 直接法的舍入误差分析

本节对用直接法求解方程组得到的解进行舍入误差分析. 下面先介绍矩阵的条件数, 它是判断矩阵病态与否的一种度量.

6.5.1 矩阵的条件数

定义 6.4 设 $\boldsymbol{A}$ 为非奇异矩阵, 称 $\kappa(\boldsymbol{A})=\|\boldsymbol{A}^{-1}\|\cdot\|\boldsymbol{A}\|$ 为矩阵 $\boldsymbol{A}$ 的条件数, 这里 $\|\cdot\|$ 是任意的算子范数.

从定义 6.4 知道, $\kappa(\boldsymbol{A})=\kappa(\boldsymbol{A}^{-1})$. 常用的条件数有

(1) 无穷范数条件数

$$\kappa(\boldsymbol{A})_\infty=\|\boldsymbol{A}^{-1}\|_\infty\cdot\|\boldsymbol{A}\|_\infty.$$

(2) 谱条件数

$$\kappa(\boldsymbol{A})_2=\|\boldsymbol{A}^{-1}\|_2\cdot\|\boldsymbol{A}\|_2=\sqrt{\frac{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}{\lambda_{\min}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}}.$$

当 $\boldsymbol{A}$ 为对称矩阵时, $\kappa(\boldsymbol{A})_2=|\lambda_1|/|\lambda_n|$, 其中 λ_1 和 λ_n 分别是 $\boldsymbol{A}$ 的绝对值最大和绝对值最小的特征值.

容易验证, 矩阵的条件数有如下性质:

(1) 任意非零矩阵的条件数 $\kappa(\boldsymbol{A})\geqslant1$.

(2) 若 $c\neq0$ 为常数, 则 $\kappa(c\boldsymbol{A})=\kappa(\boldsymbol{A})$.

(3) 若 $\boldsymbol{A}$ 为正交矩阵, 则 $\kappa(\boldsymbol{A})_2=1$.

(4) 若 $\boldsymbol{A}$ 为非奇异矩阵, $\boldsymbol{Q}$ 为正交矩阵, 则

$$\kappa(\boldsymbol{QA})_2=\kappa(\boldsymbol{AQ})_2=\kappa(\boldsymbol{A})_2.$$

6.5.2 矩阵条件数的估算

本节介绍一种非常实用的方法来估计矩阵无穷范数条件数 $\kappa(\boldsymbol{A})_\infty$. 显然

$$\|\boldsymbol{A}\|_\infty=\max_{1\leqslant i\leqslant n}\sum_{j=1}^{n}|a_{ij}|$$

可以直接计算, 主要的困难在于计算 $\|\boldsymbol{A}^{-1}\|_\infty$.

下面介绍的算法是由 Cline 等[24] 给出的, 它在列主元 LU 分解 $\boldsymbol{PA}=\boldsymbol{LU}$ 的基础上, 再增加 $O(n^2)$ 的工作量, 就可以给出条件数的量级上的估计, 关键的是这个过程中不需要求 $\boldsymbol{A}^{-1}$.

设 $\boldsymbol{y}$ 为方程组 $\boldsymbol{Ay}=\boldsymbol{d}$ 的解, 则

$$\|\boldsymbol{A}^{-1}\|_\infty \geqslant \frac{\|\boldsymbol{y}\|_\infty}{\|\boldsymbol{d}\|_\infty}.$$

上式表明, 线性方程组的解和右端项在无穷范数意义下的比是 $\|\boldsymbol{A}^{-1}\|_\infty$ 的一个下界. 如果能选取向量 $\boldsymbol{d}$ 使得 $\|\boldsymbol{y}\|_\infty/\|\boldsymbol{d}\|_\infty$ 尽量大, 那么这个比值将会是 $\|\boldsymbol{A}^{-1}\|_\infty$ 的一个很好的估计. 下面讨论如何选取 $\boldsymbol{d}$. 设 $\boldsymbol{A}$ 的奇异值分解为

$$\boldsymbol{A}=\boldsymbol{U\Sigma V}^{\mathrm{T}},$$

式中: $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 为正交阵; $\boldsymbol{\Sigma}=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_n)$ 为由 $\boldsymbol{A}$ 的奇异值构成的对角矩阵.

不失一般性, 设 $\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_n>0$. 设 $\boldsymbol{d}$ 为方程组 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{d}=\boldsymbol{b}$ 的解, $\boldsymbol{b}\in\mathbb{R}^n$. 令 $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{b}=(\alpha_1,\alpha_2,\cdots,\alpha_n)^{\mathrm{T}}$, 则有

$$\|\boldsymbol{d}\|_2^2=\|\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{b}\|_2^2=\|\boldsymbol{U\Sigma}^{-1}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{b}\|_2^2=\|\boldsymbol{\Sigma}^{-1}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{b}\|_2^2=\sum_{i=1}^{n}(\sigma_i^{-1}\alpha_i)^2, \tag{6.38}$$

及

$$\|\boldsymbol{y}\|_2^2=\|\boldsymbol{A}^{-1}\boldsymbol{d}\|_2^2=\|\boldsymbol{A}^{-1}(\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{b})\|_2^2=\|\boldsymbol{V\Sigma}^{-2}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{b}\|_2^2=\|\boldsymbol{\Sigma}^{-2}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{b}\|_2^2=\sum_{i=1}^{n}(\sigma_i^{-2}\alpha_i)^2, \tag{6.39}$$

注意到

$$\sum_{i=1}^{n}(\sigma_i^{-2}\alpha_i)^2\leqslant\frac{1}{\sigma_n^2}\sum_{i=1}^{n}(\sigma_i^{-1}\alpha_i)^2,$$

从而有

$$\frac{1}{\sigma_n}\geqslant\frac{\|\boldsymbol{y}\|_2}{\|\boldsymbol{d}\|_2}.$$

从上式可知, $1/\sigma_n$ 是 $\|\boldsymbol{y}\|_2/\|\boldsymbol{d}\|_2$ 的上界, 且当 α_n 越大时, $\|\boldsymbol{y}\|_2/\|\boldsymbol{d}\|_2$ 越接近 $1/\sigma_n(=\|\boldsymbol{A}^{-1}\|_2)$. 利用范数的等价性, $\|\boldsymbol{y}\|_\infty/\|\boldsymbol{d}\|_\infty$ 也会越大. 假设 $\boldsymbol{A}$ 有列主元 LU 分解 $\boldsymbol{PA}=\boldsymbol{LU}$. 令 $\widetilde{\boldsymbol{d}}=\boldsymbol{Pd}$, 那么有

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{d}=\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\widetilde{\boldsymbol{d}}=\boldsymbol{b},$$

而方程组 $\boldsymbol{Ay}=\boldsymbol{d}$ 等价于

$$\boldsymbol{PAy}=\boldsymbol{LUy}=\boldsymbol{Pd}=\widetilde{\boldsymbol{d}},$$

注意到 $\|\boldsymbol{Pd}\|_\infty=\|\boldsymbol{d}\|_\infty$, 则计算 $\|\boldsymbol{y}\|_\infty/\|\boldsymbol{d}\|_\infty$ 可以通过下面几步求得:

(1) 求解 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{b}$ 和 $\boldsymbol{L}^{\mathrm{T}}\widetilde{\boldsymbol{d}}=\boldsymbol{x}$.

(2) 求解 $\boldsymbol{Lz}=\widetilde{\boldsymbol{d}}$ 和 $\boldsymbol{Uy}=\boldsymbol{z}$.

(3) 计算 $\|\boldsymbol{y}\|_\infty/\|\widetilde{\boldsymbol{d}}\|_\infty$.

现考虑向量 $\boldsymbol{b}$ 的选取. 由式 (6.38) 可知, 若 α_n 越大, $\|\boldsymbol{d}\|_2$ 也越大, 所以可以考虑选取向量 $\boldsymbol{b}$ 使得 $\boldsymbol{x}$ 的无穷范数尽可能大. 此外, 由于 $\boldsymbol{b}$ 乘上一个非零常数对 $\|\boldsymbol{y}\|_\infty/\|\boldsymbol{d}\|_\infty$ 的值没有影响, 不妨要求 $\boldsymbol{b}$ 的分量为 1 或 -1. 求解 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{b}$ 的过程可通过下面算法给出:

$x_1=b_1/u_{11}$; $\beta_1=0$;

for $k=2:n$

$$\beta_k=\sum_{i=1}^{k-1}u_{ik}x_i;\ \ x_k=(b_k-\beta_k)/u_{kk};$$

end

为了使得 $\|\boldsymbol{x}\|_\infty$ 尽可能大, 取 $b_k=-\mathrm{sign}(\beta_k)$, 则

$$|x_k|=\frac{|\mathrm{sign}(\beta_k)+\beta_k|}{|u_{kk}|}=\frac{1+|\beta_k|}{|u_{kk}|},$$

所以当 $|\beta_k|$ 越大时, $|x_k|$ 也会越大.

矩阵无穷范数条件数估算的 MATLAB 程序如下:

```
function [kappa]=cond_inf(A)
%输入:矩阵A
%输出:矩阵A无穷范数条件数的估计值
n=size(A,1);[L,U]=lu(A); %列主元LU分解PA=LU
%求解U'x=b
beta(1)=0; x=zeros(n,1);b(1)=1; x(1)=b(1)/U(1,1);
for k=2:n
    beta(k)=U(1:k-1,k)'*x(1:k-1);
    a=beta(k); b(k)=-sign(a);
    x(k)=(b(k)-beta(k))/U(k,k);
end
dw=L'\x; z=L\dw; y=U\z;
h2=norm(y,'inf')/norm(dw,'inf');
h1=norm(A,'inf');kappa=h1*h2;  %矩阵A的无穷范数条件数
```

例 6.14 利用程序 cond_inf.m 估算下面矩阵 $\boldsymbol{A}$ 的无穷范数条件数 $\kappa(A)_\infty$:

$$\boldsymbol{A}=\begin{bmatrix}2&-1&&&\\-1&2&-1&&\\&\ddots&\ddots&\ddots&\\&&-1&2&-1\\&&&-1&2\end{bmatrix}_{n\times n}.$$

解 取 $n=128$, 在 MATLAB 命令窗口依次输入:

```
>> n=128;e=ones(n-1,1);A=2*eye(n)+diag(e,-1)+diag(e,1);
>> [kappa]=cond_inf(A)
```

即可得到数值结果 (略).

6.5.3 舍入误差对解的影响

用直接法解线性方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 由于测量误差或舍入误差的存在, 导致 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 有误差 $\delta\boldsymbol{A}$ 和 $\delta\boldsymbol{b}$. $\delta\boldsymbol{A}$ 和 $\delta\boldsymbol{b}$ 同计算机运算和精度有关, 计算精度越高, $\|\delta\boldsymbol{A}\|$ 和 $\|\delta\boldsymbol{b}\|$ 必然越小. 下面分析 $\|\delta\boldsymbol{A}\|$ 和 $\|\delta\boldsymbol{b}\|$ 对解的扰动的影响.

定义 6.5 *如果矩阵 $\boldsymbol{A}$ 或 $\boldsymbol{b}$ 的微小变化, 引起方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的解的巨大变化, 则称方程组是病态的, 称矩阵 $\boldsymbol{A}$ 为病态矩阵. 否则, 称方程组是良态的, 称矩阵 $\boldsymbol{A}$ 为良态矩阵.*

当 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 都有微小的变化时, 解的变化可以用下面定理描述.

定理 6.6 *设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 为非奇异矩阵, $\delta\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 满足 $\|\delta\boldsymbol{A}\|\cdot\|\boldsymbol{A}^{-1}\|<1$. 若 $\boldsymbol{x}$ 和 $\delta\boldsymbol{x}$ 满足*

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},\quad (\boldsymbol{A}+\delta\boldsymbol{A})(\boldsymbol{x}+\delta\boldsymbol{x})=\boldsymbol{b}+\delta\boldsymbol{b}, \tag{6.40}$$

则

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\leqslant\frac{\kappa(\boldsymbol{A})}{1-\kappa(\boldsymbol{A})\varepsilon_r(\boldsymbol{A})}\big(\varepsilon_r(\boldsymbol{A})+\varepsilon_r(\boldsymbol{b})\big), \tag{6.41}$$

式中: $\varepsilon_r(\boldsymbol{A})=\dfrac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}$, $\varepsilon_r(\boldsymbol{b})=\dfrac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}$, 这里的矩阵范数是由向量范数诱导出来的算子范数.

证明 由于 $\boldsymbol{A}+\delta\boldsymbol{A}=(\boldsymbol{I}+\delta\boldsymbol{A}\boldsymbol{A}^{-1})\boldsymbol{A}$ 和 $\|\delta\boldsymbol{A}\boldsymbol{A}^{-1}\|\leqslant\|\delta\boldsymbol{A}\|\cdot\|\boldsymbol{A}^{-1}\|<1$, 则 $\boldsymbol{A}+\delta\boldsymbol{A}$ 可逆且

$$\|(\boldsymbol{I}+\delta\boldsymbol{A}\boldsymbol{A}^{-1})^{-1}\|\leqslant\frac{1}{1-\|\boldsymbol{A}^{-1}\|\cdot\|\delta\boldsymbol{A}\|},$$

可知

$$\|(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\|=\|\boldsymbol{A}^{-1}(\boldsymbol{I}+\delta\boldsymbol{A}\boldsymbol{A}^{-1})^{-1}\|\leqslant\frac{\|\boldsymbol{A}^{-1}\|}{1-\|\boldsymbol{A}^{-1}\|\cdot\|\delta\boldsymbol{A}\|}.$$

此外, 由式 (6.40), 得

$$\begin{aligned}\delta\boldsymbol{x}&=(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}(\boldsymbol{b}+\delta\boldsymbol{b})-\boldsymbol{x}\\&=(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\boldsymbol{b}+(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\delta\boldsymbol{b}-\boldsymbol{A}^{-1}\boldsymbol{b}\\&=\big((\boldsymbol{A}+\delta\boldsymbol{A})^{-1}-\boldsymbol{A}^{-1}\big)\boldsymbol{b}+(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\delta\boldsymbol{b}\\&=-(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\delta\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{b}+(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\delta\boldsymbol{b}\\&=-(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\delta\boldsymbol{A}\boldsymbol{x}+(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\delta\boldsymbol{b}.\end{aligned}$$

从而

$$\|\delta\boldsymbol{x}\|\leqslant\|(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\|\cdot\|\delta\boldsymbol{A}\|\cdot\|\boldsymbol{x}\|+\|(\boldsymbol{A}+\delta\boldsymbol{A})^{-1}\|\cdot\|\delta\boldsymbol{b}\|$$

$$\leqslant \frac{\|\boldsymbol{A}^{-1}\|}{1-\|\boldsymbol{A}^{-1}\|\cdot\|\delta\boldsymbol{A}\|}(\|\delta\boldsymbol{A}\|\cdot\|\boldsymbol{x}\|+\|\delta\boldsymbol{b}\|),$$

两边除以 $\|\boldsymbol{x}\|$, 注意到 $\|\boldsymbol{b}\|=\|\boldsymbol{A}\boldsymbol{x}\|\leqslant\|\boldsymbol{A}\|\cdot\|\boldsymbol{x}\|$, 可得

$$\begin{aligned}\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} &\leqslant \frac{\|\boldsymbol{A}^{-1}\|}{1-\|\boldsymbol{A}^{-1}\|\cdot\|\delta\boldsymbol{A}\|}\Big(\|\delta\boldsymbol{A}\|+\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{x}\|}\Big)\\ &= \frac{\|\boldsymbol{A}^{-1}\|\cdot\|\boldsymbol{A}\|}{1-\|\boldsymbol{A}^{-1}\|\cdot\|\boldsymbol{A}\|\cdot\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}}\Big(\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}+\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{A}\|\cdot\|\boldsymbol{x}\|}\Big)\\ &\leqslant \frac{\|\boldsymbol{A}^{-1}\|\cdot\|\boldsymbol{A}\|}{1-\|\boldsymbol{A}^{-1}\|\cdot\|\boldsymbol{A}\|\cdot\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}}\Big(\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}+\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}\Big),\end{aligned}$$

即得到式 (6.41). 证毕. □

根据定理 6.6, 当 $\kappa(\boldsymbol{A})$ 越大时, $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 扰动之后得到的解的相对误差也越大, 相应的方程组也越病态.

对于病态的方程组, 为了得到较精确的近似解, 可以采用如下措施来减少舍入误差的影响: ① 采用高精度计算; ② 采用数值稳定性较好的算法; ③ 采用迭代改善计算解的办法.

设 $\boldsymbol{x}$ 是方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的一个近似解, 如果它没有达到精度要求, 怎样产生一个更好的近似解呢? 记 $\boldsymbol{r}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$, 由于 $\boldsymbol{x}$ 为近似解, 所以 $\boldsymbol{r}\neq\boldsymbol{0}$. 考虑方程组 $\boldsymbol{A}\boldsymbol{z}=\boldsymbol{r}$, 如果 $\boldsymbol{z}$ 是 $\boldsymbol{A}\boldsymbol{z}=\boldsymbol{r}$ 的精确解, 则 $\boldsymbol{A}(\boldsymbol{x}+\boldsymbol{z})=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{r}=\boldsymbol{b}$, 即 $\boldsymbol{x}+\boldsymbol{z}$ 为 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的一个精确解. 实际中 $\boldsymbol{z}$ 可能还是 $\boldsymbol{A}\boldsymbol{z}=\boldsymbol{r}$ 的近似解. 当 $\boldsymbol{x}+\boldsymbol{z}$ 还是不够精确时, 把 $\boldsymbol{x}+\boldsymbol{z}$ 看作上述的 $\boldsymbol{x}$, 重复上述过程. 设矩阵 $\boldsymbol{A}$ 有列主元 LU 分解 $\boldsymbol{P}\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}$, 下面给出对近似解的迭代改进:

(1) $\boldsymbol{r}:=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$; (2) 解方程组 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{P}\boldsymbol{r}$, 得到 $\boldsymbol{y}$;

(3) 解方程组 $\boldsymbol{U}\boldsymbol{z}=\boldsymbol{y}$, 得到 $\boldsymbol{z}$; (4) 令 $\boldsymbol{x}:=\boldsymbol{x}+\boldsymbol{z}$;

(5) 若 $\boldsymbol{x}$ 达到精度的要求, 停算; 否则, 转 (1).

习题 6

6.1 用列主元 Gauss 消去法解下面的方程组

$$(1)\ \begin{cases}-3x_1+2x_2+6x_3=4,\\ 10x_1-7x_2=7,\\ 5x_1-x_2+5x_3=6;\end{cases}\qquad (2)\ \begin{cases}x_1+2x_2+3x_3=6,\\ 5x_1-6x_2+9x_3=8,\\ 3x_1-2x_2+x_3=2.\end{cases}$$

6.2 顺序 Gauss 消去法可行的条件是 $a_{11}^{(1)},a_{22}^{(2)},\cdots,a_{n-1,n-1}^{(n-1)}$ 都不为零. 试证明顺序 Gauss 消去法可行的充要条件是 $\boldsymbol{A}$ 的顺序主子式 $\boldsymbol{D}_k\neq 0$, $1\leqslant k\leqslant n$.

6.3 设 $\boldsymbol{A}=(a_{ij})$, $a_{11}\neq 0$, 经一步 Gauss 消元, 得

$$\boldsymbol{A}^{(2)}=\begin{bmatrix} a_{11} & \boldsymbol{\alpha}_1^{\mathrm{T}} \\ \boldsymbol{O} & \boldsymbol{A}_2 \end{bmatrix},\quad \text{其中}\quad \boldsymbol{A}_2=\begin{bmatrix} a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \ddots & \vdots \\ a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}.$$

试证明:

(1) 若 $\boldsymbol{A}$ 对称, 则 $\boldsymbol{A}_2$ 也对称;

(2) 若 $\boldsymbol{A}$ 对称正定, 则 $\boldsymbol{A}_2$ 也对称正定.

6.4 对于 n 阶矩阵 $\boldsymbol{A}=(a_{ij})$, 若

$$|a_{ii}|>\sum_{j=1,j\neq i}^{n}|a_{ij}|,\quad i=1,2,\cdots,n,$$

则称 $\boldsymbol{A}$ 是严格对角占优矩阵. 证明: 若 $\boldsymbol{A}$ 是严格对角占优矩阵, 则经一步顺序 Gauss 消元过程后, 得到的 $\boldsymbol{A}^{(1)}$ 仍为严格对角占优矩阵.

6.5 已知方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{f}$, 其中

$$\boldsymbol{A}=\begin{bmatrix} 2 & -1 & b \\ -1 & 2 & a \\ b & -1 & 2 \end{bmatrix},\quad \boldsymbol{f}=\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

(1) 试问参数满足什么条件式时, 可选用 Cholesky 分解法求解该方程组?

(2) 取 $b=0$, $a=1$, 试用追赶法求解该方程组.

6.6 试证明:

(1) 正定矩阵必存在 LU 分解;

(2) 如果对称矩阵的各阶顺序主子式不等于零, 则必存在 LU 分解.

6.7 证明: 非奇异矩阵 $\boldsymbol{A}$ 不一定有 LU 分解.

6.8 证明: 非奇异矩阵 $\boldsymbol{A}\in\mathbf{R}^{n\times n}$ 有唯一 LDU 分解的充要条件是 $\boldsymbol{A}$ 的顺序主子式 $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, $\cdots$, $\boldsymbol{D}_{n-1}$ 都是非零的.

6.9 已知方程组

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 2 & 1 \\ 0 & 2 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 1/2 \\ 1/3 \\ -2/3 \end{bmatrix}$$

的解为 $\boldsymbol{x}=(1/2,-1/3,0)^{\mathrm{T}}$. 如果右端有微小扰动 $\|\delta\boldsymbol{b}\|_\infty=0.5\times10^{-6}$, 估计由此引起的解的相对误差.

6.10 方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 其中 $\boldsymbol{A}$ 为 $m\times n$ 阶对称且非奇异矩阵. 设 $\boldsymbol{A}$ 有误差 $\delta\boldsymbol{A}$, 则原方程组变化为 $(\boldsymbol{A}+\delta\boldsymbol{A})(\boldsymbol{x}+\delta\boldsymbol{x})=\boldsymbol{b}$, 其中 $\delta\boldsymbol{x}$ 为解的误差向量. 证明:

$$\frac{\|\delta\boldsymbol{x}\|_2}{\|\boldsymbol{x}+\delta\boldsymbol{x}\|_2}\leqslant\left|\frac{\lambda_1}{\lambda_n}\right|\frac{\|\delta\boldsymbol{A}\|_2}{\|\boldsymbol{A}\|_2},$$

式中: λ_1 和 λ_n 分别为 $\boldsymbol{A}$ 的按模最大和最小的特征值.

第 7 章　矩阵特征值问题的数值方法

许多工程实际问题的求解, 如振动问题、稳定性问题等, 最终都归结为求某些矩阵的特征值和特征向量的问题. n 阶方阵 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 的特征值与特征向量, 是满足如下两个方程的数 $\lambda \in \mathbb{C}$ 和非零向量 $\boldsymbol{x} \in \mathbb{C}^n$:

$$p(\lambda) = \det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0, \tag{7.1}$$

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x} \quad 或 \quad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}. \tag{7.2}$$

式 (7.1) 称为矩阵 $\boldsymbol{A}$ 的特征方程, $\boldsymbol{I}$ 是 n 阶单位阵, $\det(\boldsymbol{A} - \lambda \boldsymbol{I})$ 表示方阵 $\boldsymbol{A} - \lambda \boldsymbol{I}$ 的行列式, 它是 λ 的 n 次代数多项式, 当 n 较大时其零点难以准确求解. 因此, 从数值计算的观点来看, 用特征多项式来求矩阵特征值的方法并不可取, 必须建立有效的数值方法.

在实际应用中, 求矩阵的特征值和特征向量通常采用迭代法. 其基本思想是, 将特征值和特征向量作为一个无限序列的极限来求得. 舍入误差对这类方法的影响很小, 但通常计算量较大.

根据具体问题的需要, 有些实际问题只要计算模最大的特征值. 当然, 更多的问题则要求计算全部特征值和特征向量. 本章介绍几种目前在计算机上比较常用的矩阵特征值问题的数值方法.

7.1　矩阵的特征值估计和隔离

在数值计算和其他学科中, 往往需要估计一个矩阵的特征值在复平面上的位置. 例如, 在研究一个迭代算法时, 需要判断迭代矩阵 (或迭代函数的 Jacobi 矩阵) 的特征值是否全部落在单位圆内. 又如, 分析一个非线性系统是否稳定时, 需要知道有关矩阵的特征值是否全部落在复平面的左半部. 本节扼要介绍一些这方面的结果.

首先讨论 Hermite 矩阵的特征值表示问题. 由于 Hermite 矩阵 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 的特征值均为实数, 故可约定其 n 个特征值的排列次序为:

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n. \tag{7.3}$$

定义 7.1　设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 为 Hermite 矩阵, 对任意的非零向量 $\boldsymbol{x} \in \mathbb{C}^n$, 称

$$R(\boldsymbol{x}) = \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})}$$

为 $\boldsymbol{x}$ 的 Rayleigh 商.

下面的定理说明了 Hermite 矩阵的特征值可以用 Rayleigh 商的极大值和极小值来描述.

定理 7.1　设矩阵 $\boldsymbol{A}^{\mathrm{H}} = \boldsymbol{A} \in \mathbb{C}^{n\times n}$, 则

$$\max_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathbb{C}^n} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} = \lambda_{\max}(\boldsymbol{A}), \quad \min_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathbb{C}^n} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} = \lambda_{\min}(\boldsymbol{A}). \tag{7.4}$$

证明 设 $\boldsymbol{A}$ 的特征值为 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$, 则存在酉矩阵 $\boldsymbol{Q}$ 使得

$$\boldsymbol{A} = \boldsymbol{Q}\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{Q}^{\mathrm{H}},$$

故

$$\begin{aligned}\frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} &= \frac{(\boldsymbol{Q}\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x})} \\ &= \frac{(\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x})}{(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x})}.\end{aligned}$$

令

$$\boldsymbol{y} = \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x} = (y_1, y_2, \cdots, y_n)^{\mathrm{T}},$$

则

$$\begin{aligned}\frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} &= \frac{(\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{y}, \boldsymbol{y})}{(\boldsymbol{y}, \boldsymbol{y})} \\ &= \frac{\lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2}{y_1^2 + y_2^2 + \cdots + y_n^2}.\end{aligned}$$

由于

$$\lambda_n(y_1^2 + y_2^2 + \cdots + y_n^2) \leqslant \lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2 \leqslant \lambda_1(y_1^2 + y_2^2 + \cdots + y_n^2),$$

故

$$\lambda_n \leqslant \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \leqslant \lambda_1, \quad \forall\, \boldsymbol{0} \neq \boldsymbol{x} \in \mathbb{C}^n.$$

取 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_n$ 分别为对应于 λ_1 和 λ_n 的特征向量, 则

$$\frac{(\boldsymbol{A}\boldsymbol{x}_1, \boldsymbol{x}_1)}{(\boldsymbol{x}_1, \boldsymbol{x}_1)} = \lambda_1, \quad \frac{(\boldsymbol{A}\boldsymbol{x}_n, \boldsymbol{x}_n)}{(\boldsymbol{x}_n, \boldsymbol{x}_n)} = \lambda_n,$$

因此结论成立. 证毕. □

定理 7.2 $\boldsymbol{A}^{\mathrm{H}} = \boldsymbol{A} \in \mathbb{C}^{n\times n}$ 的特征值如式 (7.3), 则对 $1 \leqslant k \leqslant n$, 有

$$\lambda_k = \max_{\mathcal{V}_k} \min_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})}, \tag{7.5}$$

式中: $\mathcal{V}_k$ 为 $\mathbb{C}^n$ 的任意一个 k 维子空间.

证明 设 $\boldsymbol{A}$ 的对应于特征值 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 的特征向量依次为 $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots \boldsymbol{p}_n$, 且标准正交, 构造子空间 $\mathcal{W}_k$=span$\{\boldsymbol{p}_k, \boldsymbol{p}_{k+1}, \cdots, \boldsymbol{p}_n\}$, 那么 $\dim(\mathcal{W}_k) = n - k + 1$. 由于 $\mathcal{V}_k + \mathcal{W}_k \subset \mathbb{C}^n$, 利用子空间的维数公式求得

$$n \geqslant \dim(\mathcal{V}_k + \mathcal{W}_k) = \dim(\mathcal{V}_k) + \dim(\mathcal{W}_k) - \dim(\mathcal{V}_k \cap \mathcal{W}_k)$$

$$= n + 1 - \dim(\mathcal{V}_k \cap \mathcal{W}_k),$$

即 $\dim(\mathcal{V}_k \cap \mathcal{W}_k) \geqslant 1$. 于是存在 $\boldsymbol{x}_0 \in \mathcal{V}_k \cap \mathcal{W}_k \subset \mathcal{W}_k$ 满足 $\|\boldsymbol{x}_0\|_2 = 1$, 且有

$$\boldsymbol{x}_0 = c_k\boldsymbol{p}_k + c_{k+1}\boldsymbol{p}_{k+1} + \cdots + c_n\boldsymbol{p}_n, \quad (c_k^2 + c_{k+1}^2 + \cdots + c_n^2 = 1),$$

$$(\boldsymbol{A}\boldsymbol{x}_0, \boldsymbol{x}_0) = \lambda_k c_k^2 + \lambda_{k+1}c_{k+1}^2 + \cdots + \lambda_n c_n^2 \leqslant \lambda_k.$$

因此

$$\min_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \leqslant \frac{(\boldsymbol{A}\boldsymbol{x}_0, \boldsymbol{x}_0)}{(\boldsymbol{x}_0, \boldsymbol{x}_0)} \leqslant \lambda_k.$$

由 $\mathcal{V}_k$ 的任意性, 得

$$\max_{\mathcal{V}_k} \min_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \leqslant \lambda_k.$$

另外, 取 k 维子空间 $\mathcal{V}_k = \mathrm{span}\{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_k\}$, 当 $\boldsymbol{x} \in \mathcal{V}_k$ 满足 $\|\boldsymbol{x}\|_2 = 1$ 时, 有

$$\boldsymbol{x} = c_1\boldsymbol{p}_1 + c_2\boldsymbol{p}_2 + \cdots + c_k\boldsymbol{p}_k, \quad (c_1^2 + c_2^2 + \cdots + c_k^2 = 1),$$

$$(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}) = \lambda_1 c_1^2 + \lambda_2 c_2^2 + \cdots + \lambda_k c_k^2 \geqslant \lambda_k.$$

由 $\boldsymbol{x} \in \mathcal{V}_k$ 的任意性, 得

$$\min_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \geqslant \lambda_k.$$

从而有

$$\max_{\mathcal{V}_k} \min_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \geqslant \lambda_k.$$

综上所述, 式 (7.5) 成立. 证毕. □

定理 7.3 设 n 阶 Hermite 矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{E}$ 的特征值依次为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n, \quad \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_n.$$

则有

$$\varepsilon_n \leqslant \mu_k - \lambda_k \leqslant \varepsilon_1, \tag{7.6}$$

式中: ε_1 和 ε_n 分别为 Hermite 矩阵 $\boldsymbol{E}$ 的最大特征值和最小特征值.

证明 下面约定向量 $\boldsymbol{x}$ 满足 $\|\boldsymbol{x}\|_2 = 1$. 在式 (7.5) 中固定 $\mathcal{V}_k$, 利用式 (7.4), 得

$$\mu_k \geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{B}\boldsymbol{x}, \boldsymbol{x}) \geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}) + \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{E}\boldsymbol{x}, \boldsymbol{x})$$

$$\geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}) + \min_{\boldsymbol{x}\in\mathbb{C}^n}(\boldsymbol{E}\boldsymbol{x}, \boldsymbol{x}) = \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}) + \varepsilon_n,$$

$$\lambda_k \geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}) \geq \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{B}\boldsymbol{x}, \boldsymbol{x}) + \min_{\boldsymbol{x}\in\mathcal{V}_k}(-\boldsymbol{E}\boldsymbol{x}, \boldsymbol{x})$$

$$\geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{B}\boldsymbol{x}, \boldsymbol{x}) + \min_{\boldsymbol{x}\in\mathbb{C}^n}(-\boldsymbol{E}\boldsymbol{x}, \boldsymbol{x}) = \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{B}\boldsymbol{x}, \boldsymbol{x}) + (-\varepsilon_1).$$

由于 $\mathcal{V}_k \subset \mathbb{R}^n$ 任意, 所以

$$\mu_k \geqslant \max_{\mathcal{V}_k} \min_{\boldsymbol{x} \in \mathcal{V}_k} (\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x}) + \varepsilon_n = \lambda_k + \varepsilon_n,$$

$$\lambda_k \geqslant \max_{\mathcal{V}_k} \min_{\boldsymbol{x} \in \mathcal{V}_k} (\boldsymbol{B}\boldsymbol{x}, \boldsymbol{x}) - \varepsilon_1 = \mu_k - \varepsilon_1,$$

即式 (7.6) 成立. 证毕. □

注 7.1 定理 7.3 可用来讨论实对称矩阵的摄动问题. 可将 $\boldsymbol{E}$ 看作由于种种原因在矩阵 $\boldsymbol{A}$ 的元素中产生的误差构成的"误差矩阵", 而 $\boldsymbol{B}$ 就是实际计算时的矩阵. 设对 $\boldsymbol{E}$ 的每个元素 e_{ij} 都有 $|e_{ij}| \leqslant \varepsilon$, 其中 $\varepsilon > 0$ 是某个常数, 则 $\boldsymbol{B}$ 的特征值 $\lambda(\boldsymbol{B}) = \lambda(\boldsymbol{A}) + \Delta\lambda$ 与 $\boldsymbol{A}$ 的特征值 $\lambda(\boldsymbol{A})$ 之间满足 (7.6) 式, 即 $\varepsilon_1 \geqslant \Delta\lambda \geqslant \varepsilon_n$. 因此, 有

$$|\Delta\lambda| \leqslant \rho(\boldsymbol{E}) \leqslant \|\boldsymbol{E}\|_F \leqslant n\varepsilon.$$

这表明, 当 $\boldsymbol{A}$ 的诸元素有一个小扰动时, 在 $\boldsymbol{A}$ 的特征值中造成的扰动与其同阶.

定理 7.4 (分隔定理) 设 $\boldsymbol{A}$ 是 n 阶实对称矩阵, $\boldsymbol{B} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$, 其中 $\boldsymbol{Q} \in \mathbb{R}^{n\times(n-1)}$ 满足 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \boldsymbol{I}_{n-1}$. 再设 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的特征值分别为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \quad 和 \quad \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_{n-1},$$

则有

$$\lambda_1 \geqslant \mu_1 \geqslant \lambda_2 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_{n-1} \geqslant \lambda_n. \tag{7.7}$$

特别地, 在定理 7.4 中选取 $\boldsymbol{Q} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_{i-1}, \boldsymbol{e}_{i+1}, \cdots, \boldsymbol{e}_n]$, 即得如下结论.

推论 7.1 设 $\boldsymbol{B}$ 是 n 阶实对称矩阵 $\boldsymbol{A}$ 的一个 $n-1$ 阶主子阵, 并假定 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的特征值分别为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \quad 和 \quad \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_{n-1},$$

则有

$$\lambda_1 \geqslant \mu_1 \geqslant \lambda_2 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_{n-1} \geqslant \lambda_n.$$

反复应用推论 7.1, 便有以下推论.

推论 7.2 设 $\boldsymbol{A}$ 是一个 n 阶实对称矩阵, $\boldsymbol{B}$ 是 $\boldsymbol{A}$ 的一个 k 阶主子阵 ($1 \leqslant k \leqslant n-1$), 并假定 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的特征值分别为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \quad 和 \quad \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_k,$$

则有

$$\lambda_i \geqslant \mu_i \geqslant \lambda_{n-k+i}, \quad i = 1, 2, \cdots, k. \tag{7.8}$$

为了细致描述 n 阶矩阵的特征值在复平面上的分布范围, 下面引进 Gerschgorin 圆盘 (简称盖尔圆或盖氏圆).

定义 7.2 设 $\boldsymbol{A}=(a_{ij})\in\mathbb{C}^{n\times n}$, 令 $R_i=\sum\limits_{j=1,\,j\neq i}^{n}|a_{ij}|$, 则称

$$G_i=\{z|z\in\mathbb{C}:|z-a_{ii}|\leqslant R_i\},\quad i=1,2,\cdots n \tag{7.9}$$

为 $\boldsymbol{A}$ 的第 i 个盖氏圆.

定理 7.5 设 λ 是 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的任一特征值, 则 $\lambda\in\bigcup\limits_{i=1}^{n}G_i$, 即 $\boldsymbol{A}$ 的任一特征值都落在它的 n 个盖氏圆盘的并集内.

证明 设 $\boldsymbol{A}$ 的对应于特征值 λ 的特征向量为 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$. 选取 i_0 使得 $|x_{i_0}|=\max\limits_{1\leqslant i\leqslant n}|x_i|$, 则由 $\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}$ 可得

$$\sum_{j=1}^{n}a_{i_0j}x_j=\lambda x_{i_0}\Longrightarrow(\lambda-a_{i_0i_0})x_{i_0}=\sum_{j=1,j\neq i_0}^{n}a_{i_0j}x_j$$

$$\Longrightarrow\ |\lambda-a_{i_0i_0}|=\left|\sum_{j=1,j\neq i_0}^{n}a_{i_0j}\frac{x_j}{x_{i_0}}\right|\leqslant R_{i_0},$$

即 $\lambda\in G_{i_0}\subset\bigcup\limits_{i=1}^{n}G_i$. 证毕. □

定理 7.5 用一组圆盘覆盖矩阵的特征值分布区域, 下面介绍用另外一组几何图形覆盖矩阵的特征值分布区域的定理, 后者可以看作是前者的推广.

定理 7.6 设 λ 是 $\boldsymbol{A}\in C^{n\times n}\,(n>1)$ 的任一特征值, 则 λ 位于某个

$$\Omega_{ij}=\left\{z\,|\,z\in\mathbb{C},|z-a_{ii}||z-a_{jj}|\leq R_iR_j,\,i\neq j;\,i,j=1,2,\cdots,n\right\}$$

之中, 称 $\Omega_{ij}(i\neq j)$ 为 $\boldsymbol{A}$ 的 Cassini (卡西尼) 卵形.

证明 设 $\boldsymbol{A}$ 的对应于特征值 λ 的特征值向量为 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$. 选取 $i_0\neq j_0$ 满足 $|x_{i_0}|\geqslant|x_{j_0}|\geqslant|x_k|(k\neq i_0,j_0)$, 下证 $\lambda\in\Omega_{i_0j_0}$.

(1) 如果 $x_{j_0}=0$, 则 $x_{i_0}\neq0$, $x_k=0\ (k\neq i_0)$. 由 $\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}$ 可得

$$\lambda x_{i_0}=\sum_{k=1}^{n}a_{i_0k}x_k=a_{i_0i_0}x_{i_0}\Rightarrow\lambda=a_{i_0i_0}.$$

故

$$|\lambda-a_{i_0i_0}||\lambda-a_{j_0j_0}|=0\leq R_{i_0}R_{j_0}.$$

(2) 如果 $x_{j_0}\neq0$, 则 $|x_{i_0}|\geqslant|x_{j_0}|>0$, 再由 $\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}$ 可得

$$(\lambda-a_{ii})x_i=\sum_{k\neq i}a_{ik}x_k,\ (i=1,2,\cdots,n).$$

取 $i=i_0$ 时, 可得

$$|\lambda-a_{i_0i_0}||x_{i_0}|\leq\sum_{k\neq i_0}|a_{i_0k}||x_k|\leq|x_{j_0}|R_{i_0}.$$

取 $i=j_0$ 时, 可得

$$|\lambda-a_{j_0j_0}||x_{j_0}|\leq\sum_{k\neq j_0}|a_{j_0k}||x_k|\leq|x_{i_0}|R_{j_0}.$$

因此 $|\lambda-a_{i_0i_0}||\lambda-a_{j_0j_0}|\leq R_{i_0}R_{j_0}$. 综述 (1) 和 (2) 即得 $\lambda\in\Omega_{i_0j_0}$. 证毕. □

推论 7.3 设 $\boldsymbol{A}=(a_{ij})\in\mathbb{C}^{n\times n}\,(n>1)$ 满足 $|a_{ii}||a_{jj}|>R_iR_j\,(i\neq j)$, 则 $\det(\boldsymbol{A})\neq 0$.

证明 设 λ 是 $\boldsymbol{A}$ 的任一特征值, 那么必有 Ω_{ij} 使得 $\lambda\in\Omega_{ij}$, 即

$$|\lambda-a_{ii}||\lambda-a_{jj}|\leq R_iR_j.$$

如果 $\lambda=0$, 则有 $|a_{ii}||a_{jj}|\leq R_iR_j$, 这与题设矛盾, 故 $\lambda\neq 0$. 从而 $\det(\boldsymbol{A})\neq 0$. 证毕. □

7.2 幂法和反幂法

7.2.1 幂法

幂法是通过求矩阵的特征向量来求出特征值的一种迭代法. 它主要用来求按模最大的特征值和相应的特征向量的. 其优点是算法简单, 容易计算机实现, 缺点是收敛速度慢, 其有效性依赖于矩阵特征值的分布情况.

适于使用幂法的常见情形是: $\boldsymbol{A}$ 的特征值可按模的大小排列为 $|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$, 且其对应特征向量 $\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\cdots,\boldsymbol{\xi}_n$ 线性无关. 此时, 任意非零向量 $\boldsymbol{x}^{(0)}$ 均可用 $\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\cdots,\boldsymbol{\xi}_n$ 线性表示, 即

$$\boldsymbol{x}^{(0)}=\alpha_1\boldsymbol{\xi}_1+\alpha_2\boldsymbol{\xi}_2+\cdots+\alpha_n\boldsymbol{\xi}_n,\tag{7.10}$$

且 $\alpha_1,\ \alpha_2,\cdots,\alpha_n$ 不全为零. 作向量序列 $\boldsymbol{x}^{(k)}=\boldsymbol{A}^k\boldsymbol{x}^{(0)}$, 则

$$\begin{aligned}\boldsymbol{x}^{(k)}=\boldsymbol{A}^k\boldsymbol{x}^{(0)}&=\alpha_1\boldsymbol{A}^k\boldsymbol{\xi}_1+\alpha_2\boldsymbol{A}^k\boldsymbol{\xi}_2+\cdots+\alpha_nA^k\boldsymbol{\xi}_n\\&=\alpha_1\lambda_1^k\boldsymbol{\xi}_1+\alpha_2\lambda_2^k\boldsymbol{\xi}_2+\cdots+\alpha_n\lambda_n^k\boldsymbol{\xi}_n\\&=\lambda_1^k\Big[\alpha_1\boldsymbol{\xi}_1+\alpha_2\Big(\frac{\lambda_2}{\lambda_1}\Big)^k\boldsymbol{\xi}_2+\cdots+\alpha_n\Big(\frac{\lambda_n}{\lambda_1}\Big)^k\boldsymbol{\xi}_n\Big].\end{aligned}$$

由此可见, 若 $\alpha_1\neq 0$, 则因 $k\to\infty$ 时, 有

$$\Big(\frac{\lambda_i}{\lambda_1}\Big)^k\to 0,\quad(i=2,\cdots,n),$$

故当 k 充分大时, 必有

$$\boldsymbol{x}^{(k)}\approx\lambda_1^k\alpha_1\boldsymbol{\xi}_1,$$

即 $\boldsymbol{x}^{(k)}$ 可以近似看成 λ_1 对应的特征向量; 而 $\boldsymbol{x}^{(k)}$ 与 $\boldsymbol{x}^{(k-1)}$ 分量之比为

$$\frac{\boldsymbol{x}_i^{(k)}}{\boldsymbol{x}_i^{(k-1)}}\approx\frac{\lambda_1^k\alpha_1(\boldsymbol{\xi}_1)_i}{\lambda_1^{k-1}\alpha_1(\boldsymbol{\xi}_1)_i}=\lambda_1.$$

于是利用向量序列 $\{\boldsymbol{x}^{(k)}\}$ 既可求出按模最大的特征值 λ_1, 又可求出对应的特征向量 $\boldsymbol{\xi}_1$.

在实际计算中, 考虑到当 $|\lambda_1|>1$ 时, $\lambda_1^k\to\infty$; 当 $|\lambda_1|<1$ 时, $\lambda_1^k\to 0$, 因而计算 $\boldsymbol{x}^{(k)}$ 时可能会导致计算机"上溢"或"下溢"现象发生, 故采取每步将 $\boldsymbol{x}^{(k)}$ 归一化处理的办法, 即将 $\boldsymbol{x}^{(k)}$ 的各分量都除以模最大的分量, 使 $\|\boldsymbol{x}^{(k)}\|_\infty=1$. 于是, 求 $\boldsymbol{A}$ 按模最大的特征值 λ_1 和对应的特征向量 $\boldsymbol{\xi}_1$ 的算法, 可归纳为如下步骤.

算法 7.1 (幂法)

步 1, 输入矩阵 $\boldsymbol{A}$, 初始向量 $\boldsymbol{v}^{(0)}$, 误差限 ε, 最大迭代次数 N. 记 m_0 是 $\boldsymbol{v}^{(0)}$ 按模最大的分量, $\boldsymbol{x}^{(0)}=\boldsymbol{v}^{(0)}/m_0$. 置 $k:=0$.

步 2, 计算 $\boldsymbol{v}^{(k+1)}=\boldsymbol{A}\boldsymbol{x}^{(k)}$. 记 m_{k+1} 是 $\boldsymbol{v}^{(k+1)}$ 按模最大的分量, $\boldsymbol{x}^{(k+1)}=\boldsymbol{v}^{(k+1)}/m_{k+1}$.

步 3, 若 $|m_{k+1}-m_k|<\varepsilon$, 停算, 输出近似特征值 m_{k+1} 和近似特征向量 $\boldsymbol{x}^{(k+1)}$; 否则, 转步 4.

步 4, 若 $k<N$, 置 $k:=k+1$, 转步 2; 否则输出计算失败信息, 停算.

算法 7.1 称为幂法. 幂法的算法结构简单, 容易编程实现. 其 MATLAB 程序如下:

```
function [lam,v,k]=mypower(A,x,tol,N)
%用幂法计算矩阵的模最大特征值和对应的特征向量
%输入:A为n阶方阵,x为初始向量,tol为控制精度,N为最大迭代次数
%输出:lam为按模最大的特征值,v为对应的特征向量,k为迭代次数
if nargin<4, N=1000; end
if nargin<3, tol=1e-6; end
m=0; k=0;
while(k<N)
   v=A*x;
   [m1,t]=max(abs(v));
   m1=v(t); x=v/m1;
   err=abs(m1-m);
   if err<tol, break; end
   m=m1; k=k+1;
end
lam=m1; v=x;
```

例 7.1 利用程序 mypower.m 求矩阵 $\boldsymbol{A}$ 按模最大的特征值 λ_1 和对应的特征向量 $\boldsymbol{\xi}_1$, 其中

$$\boldsymbol{A}=\begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 99 & -1 \\ & & & -1 & 100 \end{bmatrix}.$$

解 用幂法的 MATLAB 程序, 编写 MATLAB 脚本文件 ex71.m, 取容许误差为 $\varepsilon = 10^{-6}$, 在命令窗口运行之, 迭代 661 次得到模最大的特征值 $\lambda_1 = 100.7461$.

下面证明算法 7.1 的收敛性定理.

定理 7.7 设矩阵 $\boldsymbol{A}$ 的特征值可按模的大小排列为 $|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|$, 且其对应特征向量 $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_n$ 线性无关. 序列 $\{\boldsymbol{x}^{(k)}\}$ 由算法 7.1 产生, 则有

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \frac{\boldsymbol{\xi}_1}{\max\{\boldsymbol{\xi}_1\}} := \boldsymbol{\xi}_1^0, \quad \lim_{k\to\infty} m_k = \lambda_1, \tag{7.11}$$

式中: $\boldsymbol{\xi}_1^0$ 为将 $\boldsymbol{\xi}_1$ 归一化后得到的向量; $\max\{\boldsymbol{\xi}_1\}$ 为向量 $\boldsymbol{\xi}_1$ 模最大的分量.

证明 由算法 7.1 的步 2 和步 3 知

$$\boldsymbol{x}^{(k)} = \frac{\boldsymbol{v}^{(k)}}{m_k} = \frac{\boldsymbol{A}\boldsymbol{x}^{(k-1)}}{m_k} = \frac{\boldsymbol{A}^2\boldsymbol{x}^{(k-2)}}{m_k m_{k-1}} = \cdots = \frac{\boldsymbol{A}^k\boldsymbol{x}^{(0)}}{m_k m_{k-1}\cdots m_1}.$$

由于 $\boldsymbol{x}^{(k)}$ 的最大分量为 1, 即 $\max\{\boldsymbol{x}^{(k)}\} = 1$, 故

$$m_k m_{k-1}\cdots m_1 = \max\{\boldsymbol{A}^k\boldsymbol{x}^{(0)}\}.$$

从而

$$\begin{aligned}\boldsymbol{x}^{(k)} &= \frac{\boldsymbol{A}^k\boldsymbol{x}^{(0)}}{\max\{\boldsymbol{A}^k\boldsymbol{x}^{(0)}\}} = \frac{\lambda_1^k\Big[\alpha_1\boldsymbol{\xi}_1 + \sum\limits_{i=2}^{n}\alpha_i\Big(\dfrac{\lambda_i}{\lambda_1}\Big)^k\boldsymbol{\xi}_i\Big]}{\max\Big\{\lambda_1^k\Big[\alpha_1\boldsymbol{\xi}_1 + \sum\limits_{i=2}^{n}\alpha_i\Big(\dfrac{\lambda_i}{\lambda_1}\Big)^k\boldsymbol{\xi}_i\Big]\Big\}} \\ &= \frac{\alpha_1\boldsymbol{\xi}_1 + \sum\limits_{i=2}^{n}\alpha_i\Big(\dfrac{\lambda_i}{\lambda_1}\Big)^k\boldsymbol{\xi}_i}{\max\Big\{\alpha_1\boldsymbol{\xi}_1 + \sum\limits_{i=2}^{n}\alpha_i\Big(\dfrac{\lambda_i}{\lambda_1}\Big)^k\boldsymbol{\xi}_i\Big\}}.\end{aligned}$$

可见

$$\lim_{k\to\infty}\boldsymbol{x}^{(k)} = \frac{\alpha_1\boldsymbol{\xi}_1}{\max\{\alpha_1\boldsymbol{\xi}_1\}} = \frac{\boldsymbol{\xi}_1}{\max\{\boldsymbol{\xi}_1\}} = \boldsymbol{\xi}_1^0.$$

又

$$\begin{aligned}\boldsymbol{v}^{(k)} = \boldsymbol{A}\boldsymbol{x}^{(k-1)} &= \frac{\boldsymbol{A}^k\boldsymbol{x}^{(0)}}{m_{k-1}\cdots m_1} = \frac{\boldsymbol{A}^k\boldsymbol{x}^{(0)}}{\max\{\boldsymbol{A}^{k-1}\boldsymbol{x}^{(0)}\}} \\ &= \frac{\lambda_1^k\Big[\alpha_1\boldsymbol{\xi}_1 + \sum\limits_{i=2}^{n}\alpha_i\Big(\dfrac{\lambda_i}{\lambda_1}\Big)^k\boldsymbol{\xi}_i\Big]}{\lambda_1^{k-1}\max\Big\{\alpha_1\boldsymbol{\xi}_1 + \sum\limits_{i=2}^{n}\alpha_i\Big(\dfrac{\lambda_i}{\lambda_1}\Big)^{k-1}\boldsymbol{\xi}_i\Big\}},\end{aligned}$$

注意到 m_k 是 $\boldsymbol{v}^{(k)}$ 模最大的分量, 即有

$$m_k = \max\{\boldsymbol{v}^{(k)}\} = \lambda_1\frac{\max\Big\{\alpha_1\boldsymbol{\xi}_1 + \sum\limits_{i=2}^{n}\alpha_i\Big(\dfrac{\lambda_i}{\lambda_1}\Big)^k\boldsymbol{\xi}_i\Big\}}{\max\Big\{\alpha_1\boldsymbol{\xi}_1 + \sum\limits_{i=2}^{n}\alpha_i\Big(\dfrac{\lambda_i}{\lambda_1}\Big)^{k-1}\boldsymbol{\xi}_i\Big\}},$$

从而 $\lim\limits_{k\to\infty} m_k = \lambda_1$ 成立. 证毕. □

7.2.2 幂法的加速技术

定理 7.8 在定理 7.7 的条件下, 幂法 (算法 7.1) 是线性收敛的.

证明 设 k 充分大时, $\boldsymbol{A}^k\boldsymbol{x}^{(0)}$ 的按模最大分量是它的第 j 个分量, 则

$$\begin{aligned} m_k - \lambda_1 &= \max\{\boldsymbol{v}^{(k)}\} - \lambda_1 = \frac{\max\{\boldsymbol{A}^k\boldsymbol{x}^{(0)}\}}{\max\{\boldsymbol{A}^{k-1}\boldsymbol{x}^{(0)}\}} - \lambda_1 \\ &= \frac{\left[\beta_1\lambda_1^k\boldsymbol{\xi}_1 + \beta_2\lambda_2^k\boldsymbol{\xi}_2 + \cdots + \beta_n\lambda_n^k\boldsymbol{\xi}_n\right]_j}{\left[\beta_1\lambda_1^{k-1}\boldsymbol{\xi}_1 + \beta_2\lambda_2^{k-1}\boldsymbol{\xi}_2 + \cdots + \beta_n\lambda_n^{k-1}\boldsymbol{\xi}_n\right]_j} - \lambda_1 \\ &= \frac{\left[\beta_2\lambda_2^{k-1}(\lambda_2-\lambda_1)\boldsymbol{\xi}_2 + \cdots + \beta_n\lambda_n^{k-1}(\lambda_n-\lambda_1)\boldsymbol{\xi}_n\right]_j}{\left[\beta_1\lambda_1^{k-1}\boldsymbol{\xi}_1 + \beta_2\lambda_2^{k-1}\boldsymbol{\xi}_2 + \cdots + \beta_n\lambda_n^{k-1}\boldsymbol{\xi}_n\right]_j}. \end{aligned}$$

于是有

$$\begin{aligned} m_k - \lambda_1 &= \left(\frac{\lambda_2}{\lambda_1}\right)^{k-1} \frac{\left[\beta_2(\lambda_2-\lambda_1)\boldsymbol{\xi}_2 + \sum\limits_{i=3}^{n}\beta_i\left(\frac{\lambda_i}{\lambda_2}\right)^{k-1}(\lambda_i-\lambda_1)\boldsymbol{\xi}_i\right]_j}{\left[\beta_1\boldsymbol{\xi}_1 + \sum\limits_{i=2}^{n}\beta_i\left(\frac{\lambda_i}{\lambda_1}\right)^{k-1}\boldsymbol{\xi}_i\right]_j} \\ &= \left(\frac{\lambda_2}{\lambda_1}\right)^{k-1} M_k, \quad M_k \to M, \end{aligned}$$

式中: M 为常数. 所以, 当 $k\to\infty$时, 有

$$\frac{|m_{k+1}-\lambda_1|}{|m_k-\lambda_1|} = \left|\frac{M_{k+1}(\lambda_2/\lambda_1)^k}{M_k(\lambda_2/\lambda_1)^{k-1}}\right| \to \left|\frac{\lambda_2}{\lambda_1}\right|,$$

这就证明了幂法的线性收敛速度. 证毕. □

定理 7.8 表明, 幂法的收敛速度与比值 $|\lambda_2/\lambda_1|$ 的大小有关, $|\lambda_2/\lambda_1|$ 越小, 收敛速度越快, 当此比值接近于 1 时, 收敛速度是非常缓慢的. 因此, 可以对矩阵作一原点位移, 令

$$\boldsymbol{B} = \boldsymbol{A} - \alpha\boldsymbol{I},$$

式中: α 为参数. 选择此参数可使矩阵 $\boldsymbol{B}$ 的上述比值更小, 以加快幂法的收敛速度. 设矩阵 $\boldsymbol{A}$ 的特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$, 对应的特征向量为 $\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\cdots,\boldsymbol{\xi}_n$, 则矩阵 $\boldsymbol{B}$ 的特征值为 $\lambda_1-\alpha,\lambda_2-\alpha,\cdots,\lambda_n-\alpha$, $\boldsymbol{B}$ 的特征向量和 $\boldsymbol{A}$ 的特征向量相同. 假设原点位移后, $\boldsymbol{B}$ 的特征值 $\lambda_1-\alpha$ 仍为模最大的特征值, 选择 α 的目的是使

$$\max_{2\leqslant i\leqslant n}\frac{|\lambda_i-\alpha|}{|\lambda_1-\alpha|} < \left|\frac{\lambda_2}{\lambda_1}\right|. \tag{7.12}$$

适当地选择 α 可使幂法的收敛速度得到加速. 此时 $m_k \to \lambda_1 - \alpha$, $m_k + \alpha \to \lambda_1$, 而 $\boldsymbol{x}^{(k)}$ 仍然收敛于 $\boldsymbol{A}$ 的特征向量 $\boldsymbol{\xi}_1^0$. 这种加速收敛的方法称为原点位移法.

在实际计算中, 由于矩阵的特征值分布情况事先一般是不知道的, 参数 α 的选取存在困难, 因为 α 的选取要保证 $\lambda_1 - \alpha$ 仍然是矩阵 $\boldsymbol{B}(=\boldsymbol{A}-\alpha\boldsymbol{I})$ 模最大的特征值, 故原点位移法是很难实现的. 但是, 在反幂法中, 原点位移参数 α 是非常容易选取的, 因此, 带原点位移的反幂法已成为改进特征值和特征向量精度的标准算法. 采用原点位移加速技术的幂法 MATLAB 程序如下:

```
function [lam,v,k]=mopower(A,x,alpha,tol,N)
%用原点位移幂法求矩阵的模最大特征值和对应的特征向量
%输入:A为n阶方阵,x为初始向量,tol为控制精度,
%     N最大迭代次数,alpha为原点位移参数
%输出:lam返回按模最大的特征值,v返回对应的特征向量,k返回迭代次数
if nargin<5, N=1000; end
if nargin<4, tol=1e-6; end
m=0; k=0;
A=A-alpha*eye(length(x));
while(k<N)
    v=A*x;
    [m1,t]=max(abs(v));
    m1=v(t); x=v/m1;
    err=abs(m1-m);
    if err<tol, break; end
    m=m1; k=k+1;
end
lam=m1+alpha;v=x;
```

例 7.2 利用原点位移幂法通用程序, 取 $\alpha = 50$, 求例 7.1 中的矩阵 $\boldsymbol{A}$ 按模最大的特征值 λ_1 和对应的特征向量 $\boldsymbol{\xi}_1$.

解 用原点位移幂法的 MATLAB 程序, 编写 M 文件 ex72.m, 在命令窗口运行该文件, 迭代 353 次得到模最大的特征值 $\lambda_1 = 100.7461$.

由计算结果可以看出, 在同样的精度控制下, 带原点位移加速的幂法只需要迭代 353 次, 而纯粹的幂法则需要迭代 661 次 (例 7.1).

7.2.3 反幂法

设 $\boldsymbol{A}$ 可逆, 则对 $\boldsymbol{A}$ 的逆阵 $\boldsymbol{A}^{-1}$ 施以幂法称为反幂法. 由于 $\boldsymbol{A}\boldsymbol{\xi}_i = \lambda_i\boldsymbol{\xi}_i$ 时, 成立 $\boldsymbol{A}^{-1}\boldsymbol{\xi}_i = \lambda_i^{-1}\boldsymbol{\xi}_i$. 因此, 若 $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_{n-1}| > |\lambda_n|$, 则 λ_n^{-1} 是 $\boldsymbol{A}^{-1}$ 按模最大的特征值, 此时按反幂法, 必有

$$m_k \to \lambda_n^{-1}, \quad \boldsymbol{x}^{(k)} \to \boldsymbol{\xi}_n^0,$$

且其收敛率为 $|\lambda_n/\lambda_{n-1}|$. 任取初始向量 $\boldsymbol{x}^{(0)}$, 构造向量序列

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{A}^{-1}\boldsymbol{x}^{(k)}, \quad k = 0, 1, 2, \cdots \tag{7.13}$$

按幂法计算即可. 但用式 (7.13) 计算, 首先要求 $\boldsymbol{A}^{-1}$, 这比较麻烦而且是不经济的. 实际计算中, 通常用解方程组的办法, 即用

$$\boldsymbol{A}\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}, \quad k = 0, 1, 2, \cdots, \tag{7.14}$$

求 $\boldsymbol{x}^{(k+1)}$. 为防止计算机溢出, 实际计算时所用的公式为

$$\begin{cases} \boldsymbol{v}^{(k)} = \boldsymbol{x}^{(k)}/\max(\boldsymbol{x}^{(k)}), \\ \boldsymbol{A}\boldsymbol{x}^{(k+1)} = \boldsymbol{v}^{(k)}, \end{cases} \quad k = 0, 1, 2, \cdots, \tag{7.15}$$

式中: $\max(\boldsymbol{x}^{(k)})$ 为 $\boldsymbol{x}^{(k)}$ 模最大的分量.

反幂法主要用于已知矩阵的近似特征值为 α 时, 求矩阵的特征向量并提高特征值的精度. 此时, 可以用原点位移法来加速迭代过程, 于是式 (7.15) 相应为

$$\begin{cases} \boldsymbol{v}^{(k)} = \boldsymbol{x}^{(k)}/\max(\boldsymbol{x}^{(k)}), \\ (\boldsymbol{A} - \alpha\boldsymbol{I})\boldsymbol{x}^{(k+1)} = \boldsymbol{v}^{(k)}, \end{cases} \quad k = 0, 1, 2, \cdots. \tag{7.16}$$

反幂法的计算步骤如下.

算法 7.2 (反幂法)

步 1, 选取初值 $\boldsymbol{x}^{(0)}$, 近似值 α, 误差限 ε, 最大迭代次数 N. 记 m_0 为 $\boldsymbol{x}^{(0)}$ 中按模最大的分量, $\boldsymbol{v}^{(0)} = \boldsymbol{x}^{(0)}/m_0$. 置 $k := 0$.

步 2, 解方程组 $(\boldsymbol{A} - \alpha\boldsymbol{I})\boldsymbol{x}^{(k+1)} = \boldsymbol{v}^{(k)}$ 得 $\boldsymbol{x}^{(k+1)}$.

步 3, 记 m_{k+1} 为 $\boldsymbol{x}^{(k+1)}$ 中按模最大的分量, $\boldsymbol{v}^{(k+1)} = \boldsymbol{x}^{(k+1)}/m_{k+1}$.

步 4, 若 $|m_{k+1}^{-1} - m_k^{-1}| < \varepsilon$, 则置 $\lambda := m_{k+1}^{-1} + \alpha$, 输出 λ 和 $\boldsymbol{x}^{(k+1)}$, 停算; 否则, 转步 5.

步 5, 若 $k < N$, 置 $k := k + 1$, 转步 2, 否则输出计算失败信息, 停算.

注 7.2 (1) 算法 7.2 计算出与数 α 最接近的特征值及相应的特征向量. 若取 $\alpha = 0$, 则求出 $\boldsymbol{A}$ 的按模最小的特征值.

(2) 通常首先用幂法求出 $\boldsymbol{A}$ 的按模最大的近似特征值作为算法 7.2 中的 α 值, 再使用该算法对 α 和相应的特征向量进行精确化.

(3) 为节省计算量, 通常先用列主元 LU 分解将矩阵 $\boldsymbol{A} - \alpha\boldsymbol{I}$ 分解为下三角矩阵 $\boldsymbol{L}$ 和上三角矩阵 $\boldsymbol{U}$, 这样在迭代过程中每一步就只需求解两个三角方程组即可.

反幂法的 MATLAB 程序如下:

```
function [lam,v,k]=mvpower(A,x,alpha,tol,N)
%用反幂法计算矩阵与alpha最接近的特征值和对应的特征向量
%输入:A为n阶方阵,x为初始向量,tol为精度,N为最大迭代数,alpha为某个常数
%输出:lam返回与alpha最接近的特征值,v返回对应的特征向量,k返回迭代次数
if nargin<5, N=500; end
if nargin<4, tol=1e-5; end
m=0.5; k=0;
A=A-alpha*eye(length(x));
[L,U,P]=lu(A);
while (k<N)
   [m1,t]=max(abs(x));
   m1=x(t); v=x/m1;
   z=L\(P*v); x=U\z;
   err=abs(1/m1-1/m);
   if err<=tol, break; end
   k=k+1; m=m1;
end
lam=alpha+1/m;
```

例 7.3 利用反幂法程序, 求例 7.1 中的矩阵 $\boldsymbol{A}$ 最接近 101, 99, 2 和 0 的特征值和对应的特征向量.

解 注意到此处 α 的值分别取 $101, 99, 2, 0$. 用反幂法的 MATLAB 程序, 编写 M 文件 ex73.m, 取容许误差为 $\varepsilon = 10^{-6}$, 在命令窗口运行该文件, 得到 4 个近似特征值: 100.7462; 99.2107; 1.7893; 0.2538, 这是矩阵 $\boldsymbol{A}$ 的两个模最大和两个模最小的特征值.

7.3 Jacobi 方法

Jacobi 方法用于求解实对称矩阵的全部特征值和对应的特征向量. 其数学原理如下:

(1) n 阶实对称矩阵的特征值全为实数, 其对应的特征向量线性无关且两两正交.

(2) 相似矩阵具有相同的特征值.

(3) 若 n 阶实矩阵 $\boldsymbol{A}$ 是对称的, 则存在正交矩阵 $\boldsymbol{Q}$, 使得 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{D}$, 其中 $\boldsymbol{D}$ 是一个对角矩阵, 它的对角元素 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 就是 $\boldsymbol{A}$ 的特征值, $\boldsymbol{Q}$ 的第 i 列向量就是 λ_i 对应的特征向量.

Jacobi 方法就是基于上述原理, 用一系列正交变换对角化 $\boldsymbol{A}$, 即逐步消去 $\boldsymbol{A}$ 的非对角元, 从而得到 $\boldsymbol{A}$ 的全部特征值.

7.3.1 实对称矩阵的旋转正交相似变换

首先回顾一下第 2 章介绍过的一种正交变换—Givens 变换, 它是 Jacobi 方法的基本工具.

定义 7.3 设 $1 \leqslant i < j \leqslant n$, 则称矩阵

$$\boldsymbol{G}_{ij} = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & \cos\varphi & & & & \sin\varphi & \\ & & & 1 & & & & \\ & & & & \ddots & & & \\ & & & & & 1 & & \\ & & -\sin\varphi & & & & \cos\varphi & \\ & & & & & & & \ddots \\ & & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \end{matrix} \tag{7.17}$$

$$\begin{matrix} & & i & & j & \end{matrix}$$

为 (i,j) 平面的旋转矩阵, 或 Givens 变换矩阵.

显然, $\boldsymbol{G} = \boldsymbol{G}_{ij}$ 为正交矩阵, 即 $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} = \boldsymbol{I}$. 对于向量 $\boldsymbol{x} \in \mathbb{R}^n$, 由线性变换 $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}$ 得到的 $\boldsymbol{y}$ 的分量为

$$\begin{cases} y_i = x_i \cos\varphi + x_j \sin\varphi, \\ y_j = -x_i \sin\varphi + x_j \cos\varphi, \\ y_k = x_k, \quad k \neq i, j, \end{cases} \tag{7.18}$$

即用 $\boldsymbol{G}_{ij}$ 对向量 $\boldsymbol{x}$ 作用, 只改变其第 i,j 两个分量.

由矩阵 $\boldsymbol{G} = \boldsymbol{G}_{ij}$ 确定的正交变换 $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}$ 称为平面旋转变换, 或 Givens 变换. 根据式 (7.18) 容易验证, 矩阵 $\boldsymbol{G}_{ij}$ 具有下列基本性质.

定理 7.9 设 $\boldsymbol{x} \in \mathbb{R}^n$ 的第 j 个分量 $x_j \neq 0$, $1 \leqslant i < j \leqslant n$. 若令

$$c = \cos\varphi = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad s = \sin\varphi = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}, \tag{7.19}$$

则 $\boldsymbol{y} = \boldsymbol{G}_{ij}\boldsymbol{x}$ 的分量为

$$\begin{cases} y_i = \sqrt{x_i^2 + x_j^2}, \quad y_j = 0, \\ y_k = x_k, \quad k \neq i, j. \end{cases} \tag{7.20}$$

定理 7.9 表明, 可以用 Givens 变换将向量的某个分量变为零元素.

下面讨论 Givens 变换对实对称矩阵的作用. 用旋转矩阵 $\boldsymbol{G}_{ij}$ 对实对称矩阵 $\boldsymbol{A} = (a_{ij})_{n\times n}$ 作正交相似变换, 所得矩阵记为 $\boldsymbol{A}_1$, 即

$$\boldsymbol{A}_1 = \boldsymbol{G}_{ij}\boldsymbol{A}\boldsymbol{G}_{ij}^{\mathrm{T}} = (a_{ij}^{(1)}).$$

显然

$$\boldsymbol{A}_1^{\mathrm{T}} = (\boldsymbol{G}_{ij}\boldsymbol{A}\boldsymbol{G}_{ij}^{\mathrm{T}})^{\mathrm{T}} = \boldsymbol{G}_{ij}\boldsymbol{A}\boldsymbol{G}_{ij}^{\mathrm{T}} = \boldsymbol{A}_1,$$

即 $\boldsymbol{A}_1$ 仍为实对称矩阵. 直接计算, 得

$$\begin{cases} a_{ii}^{(1)} = a_{ii}\cos^2\varphi + a_{jj}\sin^2\varphi + 2a_{ij}\cos\varphi\sin\varphi, \\ a_{jj}^{(1)} = a_{ii}\sin^2\varphi + a_{jj}\cos^2\varphi - 2a_{ij}\cos\varphi\sin\varphi, \\ a_{ij}^{(1)} = a_{ji}^{(1)} = a_{ij}(\cos^2\varphi - \sin^2\varphi) - (a_{ii} - a_{jj})\cos\varphi\sin\varphi, \\ a_{il}^{(1)} = a_{li}^{(1)} = a_{il}\cos\varphi + a_{jl}\sin\varphi, \quad l \neq i, j, \\ a_{jl}^{(1)} = a_{lj}^{(1)} = -a_{il}\sin\varphi + a_{jl}\cos\varphi, \quad l \neq i, j, \\ a_{lm}^{(1)} = a_{ml}^{(1)} = a_{ml}, \quad m, l \neq i, j. \end{cases} \tag{7.21}$$

不难看出, $\boldsymbol{A}$ 经过 $\boldsymbol{G}_{ij}$ 的正交相似变换后, $\boldsymbol{A}_1$ 的元素和 $\boldsymbol{A}$ 的元素相比, 只有第 i 行, 第 j 行和第 i 列, 第 j 列元素发生了变化, 而其他元素和 $\boldsymbol{A}$ 是相同的.

由式 (7.21) 的最后一个等式可知, 若 $a_{ij} \neq 0$, 则可适当选取 φ 的值, 使得 $a_{ij}^{(1)} = a_{ji}^{(1)} = 0$. 事实上, 令

$$a_{ij}(\cos^2\varphi - \sin^2\varphi) - (a_{ii} - a_{jj})\cos\varphi\sin\varphi = 0,$$

解得

$$\cot 2\varphi = \frac{a_{ii} - a_{jj}}{2a_{ij}} = \frac{1 - \tan^2\varphi}{2\tan\varphi}, \quad -\frac{\pi}{4} < \varphi \leqslant \frac{\pi}{4}. \tag{7.22}$$

在 Jacobi 方法中, 总是按上式选取 φ. 在实际计算时, 为避免使用三角函数, 可令

$$t = \tan\varphi, \quad c = \cos\varphi, \quad s = \sin\varphi, \quad d = \frac{a_{ii} - a_{jj}}{2a_{ij}}. \tag{7.23}$$

由式 (7.22), 得

$$t^2 + 2dt - 1 = 0. \tag{7.24}$$

式 (7.24) 有两个根, 取其绝对值最小者为 t, 即

$$t = \begin{cases} -d + \sqrt{d^2+1}, & d \geqslant 0, \\ -d - \sqrt{d^2+1}, & d < 0. \end{cases} \tag{7.25}$$

若记

$$c = \cos\varphi = \frac{1}{\sqrt{1+t^2}}, \quad s = \sin\varphi = \frac{t}{\sqrt{1+t^2}} = ct, \tag{7.26}$$

这时, 式 (7.21) 可写为

$$\begin{cases} a_{ii}^{(1)} = a_{ii}c^2 + a_{jj}s^2 + 2csa_{ij}, \\ a_{jj}^{(1)} = a_{ii}s^2 + a_{jj}c^2 - 2csa_{ij}, \quad a_{ij}^{(1)} = a_{ji}^{(1)} = 0, \\ a_{il}^{(1)} = a_{li}^{(1)} = ca_{il} + sa_{jl}, \ a_{jl}^{(1)} = a_{lj}^{(1)} = -sa_{il} + ca_{jl}, \quad l \neq i, j; \\ a_{lm}^{(1)} = a_{ml}^{(1)} = a_{ml}, \quad m, l \neq i, j. \end{cases} \tag{7.27}$$

利用等式 $a_{ij}(c^2-s^2)-(a_{ii}-a_{jj})cs=0$, 可以验证

$$\left(a_{ii}^{(1)}\right)^2+\left(a_{jj}^{(1)}\right)^2=a_{ii}^2+a_{jj}^2+2a_{ij}^2, \tag{7.28}$$

即 $\boldsymbol{A}$ 经过一次这种正交相似变换后, 所得到的矩阵 $\boldsymbol{A}_1$ 的对角元素平方和增加了 $2a_{ij}^2$.

7.3.2 Jacobi 方法及其收敛性

选择 $\boldsymbol{A}_0=\boldsymbol{A}$ 中一对非零的非对角元素 a_{ij}, a_{ji}, 使用平面旋转矩阵 $\boldsymbol{G}_{ij}$ 作正交相似变换得 $\boldsymbol{A}_1$, 可使 $\boldsymbol{A}_1$ 的这对非对角元素 $a_{ij}^{(1)}=a_{ji}^{(1)}=0$; 再选择 $\boldsymbol{A}_1$ 中一对非零的非对角元素作上述旋转正交相似变换得 $\boldsymbol{A}_2$, 可使 $\boldsymbol{A}_2$ 的这对非对角元素为 0. 如此不断地作旋转正交相似变换, 可产生一个矩阵序列 $\boldsymbol{A}=\boldsymbol{A}_0, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_k, \cdots$. 虽然 $\boldsymbol{A}$ 至多只有 $n(n-1)/2$ 对非零非对角元素, 但不能期望通过 $n(n-1)/2$ 次旋转正交相似变换使其对角化. 因为每次旋转变换虽然能使一对特定的非对角元素化为 0, 但这次变换可能将前面已经化为 0 了的一对非对角元素变成非 0.

但是, 在 Jacobi 方法中的每一步, 如由 $\boldsymbol{A}_{k-1}$ 变成 $\boldsymbol{A}_k$, 取其绝对值最大的一对非零非对角元素, 即取

$$\left|a_{i_kj_k}^{(k-1)}\right|=\max_{\substack{1\leqslant i,j\leqslant n\\ i\neq j}}\left|a_{ij}^{(k-1)}\right| \tag{7.29}$$

作旋转相似变换, 这时记旋转矩阵 $\boldsymbol{G}_{ij}=\boldsymbol{G}_{i_kj_k}$. 后面将证明, 这样产生的矩阵序列 $\boldsymbol{A}_0, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_k, \cdots$ 趋向于对角矩阵, 即 Jacobi 方法是收敛的.

在实际计算中, 可预先取一个小的控制量 $\varepsilon>0$, 若成立

$$\left|a_{ij}^{(k)}\right|<\varepsilon,\quad i,j=1,2,\cdots,n,\ i\neq j, \tag{7.30}$$

则可视 $\boldsymbol{A}_k$ 为对角矩阵, 从而结束计算. $\boldsymbol{A}_k$ 的对角元素可视为 $\boldsymbol{A}$ 的特征值.

Jacobi 方法也可以求 $\boldsymbol{A}$ 的所有特征向量. 事实上, 由

$$\begin{aligned}\boldsymbol{A}_k&=\boldsymbol{G}_k\boldsymbol{A}_{k-1}\boldsymbol{G}_k^{\mathrm{T}}=\boldsymbol{G}_k\boldsymbol{G}_{k-1}\boldsymbol{A}_{k-2}\boldsymbol{G}_{k-1}^{\mathrm{T}}\boldsymbol{G}_k^{\mathrm{T}}=\cdots\\&=\boldsymbol{G}_k\boldsymbol{G}_{k-1}\cdots\boldsymbol{G}_1\boldsymbol{A}\boldsymbol{G}_1^{\mathrm{T}}\cdots\boldsymbol{G}_{k-1}^{\mathrm{T}}\boldsymbol{G}_k^{\mathrm{T}},\end{aligned}$$

若记

$$\boldsymbol{Q}_k=\boldsymbol{G}_1^{\mathrm{T}}\cdots\boldsymbol{G}_{k-1}^{\mathrm{T}}\boldsymbol{G}_k^{\mathrm{T}}, \tag{7.31}$$

则

$$\boldsymbol{A}_k=\boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_k. \tag{7.32}$$

式中: $\boldsymbol{Q}_k$ 为正交矩阵.

若 $\boldsymbol{A}_k$ 可视为对角矩阵, 其对角元即为 $\boldsymbol{A}$ 的特征值, 其第 i 个对角元 $a_{ii}^{(k)}$ 对应的特征向量就是 $\boldsymbol{Q}_k$ 的第 i 列元素构成的向量. $\boldsymbol{Q}_k$ 的计算可与 $\boldsymbol{A}$ 的旋转相似变换同步进行. 若令 $\boldsymbol{Q}_0=\boldsymbol{I}$, 则

$$\boldsymbol{Q}_k=\boldsymbol{Q}_{k-1}\boldsymbol{G}_k^{\mathrm{T}}. \tag{7.33}$$

若 $\boldsymbol{G}_k = \boldsymbol{G}_{ij}$, 得 $\boldsymbol{Q}_k$ 的计算公式如下:

$$\begin{cases} q_{li}^{(k)} = q_{li}^{(k-1)}c + q_{lj}^{(k-1)}s, & l = 1, 2, \cdots, n, \\ q_{lj}^{(k)} = -q_{li}^{(k-1)}s + q_{lj}^{(k-1)}c, & l = 1, 2, \cdots, n, \\ q_{km}^{(k)} = q_{km}^{(k-1)}, & k, m \neq i, j. \end{cases} \tag{7.34}$$

也就是说, 除了第 i, j 列元素发生变化外, 其他元素不变. 若不需要计算特征向量, 则可省略此步.

根据上述讨论, 可得 Jacobi 方法的计算步骤如下.

算法 7.3 (Jacobi 方法)

步 1, 输入矩阵 $\boldsymbol{A}$, $\boldsymbol{Q} = \boldsymbol{I}$, 初始向量 $\boldsymbol{x}$, 误差限 ε, 最大迭代次数 N. 置 $k := 1$.

步 2, 在矩阵中找绝对值最大的非对角元

$$\mu = |a_{i_r j_r}| = \max_{\substack{1 \leqslant i, j \leqslant n \\ i \neq j}} |a_{ij}|,$$

置 $i := i_r$, $j := j_r$.

步 3, 按式 (7.23) $\sim$ 式 (7.27) 计算 d, t, c, s 的值和矩阵 $\boldsymbol{A}_1$ 的元素 $a_{lm}^{(1)}$, $l, m = 1, 2, \cdots, n$.

步 4, 更新 $\boldsymbol{Q}$ 的元素:

$$\begin{cases} q_{li} := q_{li}c + q_{lj}s, \\ q_{lj} := -q_{li}s + q_{lj}c, \end{cases} \quad l = 1, 2, \cdots, n.$$

步 5, 若 $\mu < \varepsilon$, 输出 $\boldsymbol{A}_1$ 的对角元和 $\boldsymbol{Q}$ 的列向量, 停算; 否则, 转步 6.

步 6, 若 $k < N$, 置 $k := k + 1$, 转步 2; 否则输出计算失败信息, 停算.

根据算法 7.3, 编制 Jacobi 方法的 MATLAB 程序如下:

```
%Jacobi方法程序-Jacobi_eig.m
function [lambda,Q]=Jacobi_eig(A,tol)
%用Jacobi方法求实对称矩阵A的全部特征值和特征向量
%输入:A为n阶对称方阵,tol为容许误差
%输出:lambda为向量,其分量为A的特征值,
%     Q为矩阵,其元素为矩阵A的n个特征向量
if nargin<2, tol=1e-6; end
[n]=size(A,1); Q=eye(n);
%计算A的非对角元绝对值最大元素所在的行p和列q
[w1,p]=max(abs(A-diag(diag(A))));
[w2,q]=max(w1); p=p(q);
while(1)
  d=(A(p,p)-A(q,q))/(2*A(p,q));
```

```
    if(d>=0)
       t=-d+sqrt(d^2+1);
    else
       t=-d-sqrt(d^2+1);
    end
    c=1/sqrt(t^2+1); s=c*t; G=[c s; -s c];
    A([p q],:)=G*A([p q],:);
    A(:,[p q])=A(:,[p q])*G';
    Q(:,[p q])=Q(:,[p q])*G';
    [w1,p]=max(abs(A-diag(diag(A))));
    [w2,q]=max(w1); p=p(q);
    if (abs(A(p,q))<tol*sqrt(sum(diag(A).^2)/n))
       break;
    end
  end
  lambda=sort(diag(A));
```

例 7.4 利用 Jacobi 方法程序, 求例 7.1 中的矩阵 $\boldsymbol{A}$ 的全部特征值和对应的特征向量.

解 在 MATLAB 命令窗口执行程序 ex74.m 得到矩阵 $\boldsymbol{A}$ 的全部特征值 $\hat{\boldsymbol{\lambda}}$. 此外, 利用 MATLAB 自带的函数 eig 求得其特征值为 $\boldsymbol{\lambda}$, 计算其误差 $\|\hat{\boldsymbol{\lambda}}-\boldsymbol{\lambda}\|_2 = 3.1756\times 10^{-9}$. 特征值的分布如图 7.1 所示.

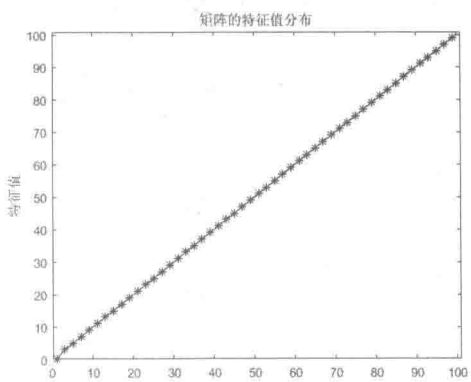

图 7.1 矩阵 $\boldsymbol{A}$ 的特征值分布

下面考虑算法 7.3 (Jacobi 方法) 的收敛性. 记实对称矩阵 $\boldsymbol{A}$ 的非对角元素的平方和为

$$S(\boldsymbol{A}) = \sum_{\substack{l,m=1\\ l\neq m}}^{n} a_{lm}^2. \tag{7.35}$$

设 $\boldsymbol{A}_{k+1}=\boldsymbol{G}_{ij}\boldsymbol{A}_k\boldsymbol{G}_{ij}^{\mathrm{T}}$, 则由式 (7.27) 不难验证

$$S(\boldsymbol{A}_{k+1})=S(\boldsymbol{A}_k)-2\big[a_{ij}^{(k)}\big]^2. \tag{7.36}$$

即经过这种正交相似变换后, $\boldsymbol{A}_{k+1}$ 的非对角元素平方和减少了 $2[a_{ij}^{(k)}]^2$. 同时, 由式 (7.28), 有

$$\big[a_{ii}^{(k+1)}\big]^2+\big[a_{jj}^{(k+1)}\big]^2=\big[a_{ii}^{(k)}\big]^2+\big[a_{jj}^{(k)}\big]^2+2\big[a_{ij}^{(k)}\big]^2, \tag{7.37}$$

即对角元素的平方和增加了 $2[a_{ij}^{(k)}]^2$. 若在 Jacobi 方法中, 每次旋转正交相似变换使 $\boldsymbol{A}_k$ 的绝对值最大的非对角元素化为 0, 则成立以下定理.

定理 7.10 *记实对称矩阵 $\boldsymbol{A}=\boldsymbol{A}_0$, 若在 Jacobi 方法中, 每次旋转正交相似变换使 $\boldsymbol{A}_k$ 的绝对值最大的非对角元素化为 0, 则得到的矩阵序列 $\{\boldsymbol{A}_k\}$ 趋向于对角矩阵.*

证明 设 $\boldsymbol{A}_k$ 的绝对值最大的非对角元素为 $a_{ij}^{(k)}$, 故有

$$\big[a_{ij}^{(k)}\big]^2\geqslant\frac{1}{n(n-1)}S(\boldsymbol{A}_k).$$

用旋转正交相似变换将其化为 0, 得 $\boldsymbol{A}_{k+1}$, 此时

$$\begin{aligned}S(\boldsymbol{A}_{k+1})&=S(\boldsymbol{A}_k)-2\big[a_{ij}^{(k)}\big]^2\leqslant S(\boldsymbol{A}_k)-\frac{2}{n(n-1)}S(\boldsymbol{A}_k)\\&=\Big[1-\frac{2}{n(n-1)}\Big]S(\boldsymbol{A}_k)\leqslant\Big[1-\frac{2}{n(n-1)}\Big]^{k+1}S(\boldsymbol{A}).\end{aligned}$$

由于

$$1-\frac{2}{n(n-1)}<1,$$

所以

$$\lim_{k\to\infty}S(\boldsymbol{A}_{k+1})=0,$$

即 $\boldsymbol{A}_{k+1}$ 趋向于对角矩阵, 故 Jacobi 方法是收敛的. 证毕. □

7.4 QR 方法

QR 方法用于求一般矩阵的全部特征值, 是目前最有效的方法之一. 本节就实矩阵的情形进行介绍.

众所周知, 对于任何实对称矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 存在正交矩阵 $\boldsymbol{Q}$ 使得

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n),$$

式中: $\lambda_1,\lambda_2,\cdots,\lambda_n$ 为 $\boldsymbol{A}$ 的全部特征值; $\boldsymbol{Q}$ 的列向量为对应的特征向量.

而对于一般的 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 有下面的实 Schur 分解定理.

定理 7.11 对于任何实矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 存在正交矩阵 $\boldsymbol{Q}$ 使得

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{R}_{11} & \boldsymbol{R}_{12} & \cdots & \boldsymbol{R}_{1m}\\ & \boldsymbol{R}_{22} & \cdots & \boldsymbol{R}_{2m}\\ & & \ddots & \vdots\\ & & & \boldsymbol{R}_{mm}\end{bmatrix}, \tag{7.38}$$

式中: 对角块 $\boldsymbol{R}_{ii}\,(i=1,2,\cdots,m)$ 为 1×1 或 2×2 的子矩阵, 1 阶子矩阵的元素是 $\boldsymbol{A}$ 的实特征值, 2 阶子矩阵的两个特征值是 $\boldsymbol{A}$ 的一对复共轭特征值.

式 (7.38) 通常称为矩阵 $\boldsymbol{A}$ 的实 Schur 分解, 而右边的分块上三角矩阵称为 $\boldsymbol{A}$ 的实 Schur 标准形. 显然, 只要求得一个实矩阵的实 Schur 标准形, 就很容易求得它的全部特征值.

因此, 通常希望构造一种迭代 (相似变换), 希望它能逼近矩阵 $\boldsymbol{A}$ 的实 Schur 标准形. 例如, 对于给定的矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 令 $\boldsymbol{A}_1:=\boldsymbol{A}$, 构造迭代:

$$\begin{cases}\boldsymbol{A}_k=\boldsymbol{Q}_k\boldsymbol{R}_k,\\ \boldsymbol{A}_{k+1}=\boldsymbol{R}_k\boldsymbol{Q}_k,\end{cases}\qquad k=1,2,\cdots, \tag{7.39}$$

式中: $\boldsymbol{Q}_k$ 为正交矩阵; $\boldsymbol{R}_k$ 为上三角矩阵.

可以证明, 在一定条件下, 由式 (7.39) 产生的矩阵序列 $\{\boldsymbol{A}_k\}$ 将“逼近”于 $\boldsymbol{A}$ 的实 Schur 标准形.

然而, 式 (7.39) 作为一种实用的迭代法是没有竞争力的. 一是每步迭代的运算量太大 (大约是 $O(n^3)$); 二是收敛速度太缓慢 (依赖于特征值的分离程度). 因此, 要想其成为一种高效的迭代方法, 必须设法尽可能地减少每步迭代的运算量, 提高其收敛速度. 一个可行的办法是, 首先把矩阵 $\boldsymbol{A}$ 正交相似变换为上 Hessenberg 形, 然后再用正交相似变换对它进行迭代. 下面就循着这个思路介绍实用的 QR 方法来求实矩阵 $\boldsymbol{A}$ 的全部特征值.

7.4.1 化一般矩阵为上 Hessenberg 矩阵

在用 QR 方法求矩阵特征值时, Householder 矩阵有两个作用: 一是对 $\boldsymbol{A}$ 作正交相似变换, 把 $\boldsymbol{A}$ 化为上 Hessenberg 矩阵; 二是对矩阵作正交三角分解.

首先讨论把 $\boldsymbol{A}$ 化为上 Hessenberg 矩阵. 设 $\boldsymbol{A}_1=\boldsymbol{A}=(a_{ij}^{(1)})$ 是 n 阶实方阵, 取 $\boldsymbol{x}=(0,a_{21}^{(1)},\cdots,a_{n1}^{(1)})^{\mathrm{T}}$, 记 $a_1=\mathrm{sgn}(x_2)\|\boldsymbol{x}\|_2$, 则由定理 2.2 和定理 2.3 构造 Householder 矩阵

$$\boldsymbol{H}_1=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & * & \cdots & *\\ \vdots & \vdots & \ddots & \vdots\\ 0 & * & \cdots & *\end{bmatrix},$$

使得

$$\boldsymbol{H}_1\boldsymbol{x}=a_1\boldsymbol{e}_2.$$

所以 $\boldsymbol{H}_1\boldsymbol{A}_1$ 的第 1 列为

$$\boldsymbol{H}_1\begin{bmatrix} a_{11}^{(1)} \\ a_{21}^{(1)} \\ \vdots \\ a_{n1}^{(1)} \end{bmatrix} = \boldsymbol{H}_1\boldsymbol{x} + \boldsymbol{H}_1\begin{bmatrix} a_{11}^{(1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = a_1\boldsymbol{e}_2 + \begin{bmatrix} a_{11}^{(1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} \\ a_1 \\ \vdots \\ 0 \end{bmatrix}.$$

因为用 $\boldsymbol{H}_1$ 右乘一个矩阵不改变该矩阵的第 1 列, 于是

$$\boldsymbol{A}_2 = \boldsymbol{H}_1\boldsymbol{A}_1\boldsymbol{H}_1 = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ a_1 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}.$$

再取 $\boldsymbol{x} = (0, 0, a_{32}^{(2)}, \cdots, a_{n2}^{(2)})^{\mathrm{T}}$, 记 $a_2 = \mathrm{sgn}(x_3)\|\boldsymbol{x}\|_2$, 构造 $\boldsymbol{H}_2$ 为

$$\boldsymbol{H}_2 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix},$$

使得

$$\boldsymbol{H}_2\boldsymbol{x} = a_2\boldsymbol{e}_3.$$

所以 $\boldsymbol{H}_2\boldsymbol{A}_2$ 的第 1 列与 $\boldsymbol{A}_2$ 的第 1 列相同, 而 $\boldsymbol{H}_2\boldsymbol{A}_2$ 的第 2 列变为

$$\boldsymbol{H}_2\boldsymbol{x} + \boldsymbol{H}_2\begin{bmatrix} a_{12}^{(2)} \\ a_{22}^{(2)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = a_2\boldsymbol{e}_3 + \begin{bmatrix} a_{12}^{(2)} \\ a_{22}^{(2)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} a_{12}^{(2)} \\ a_{22}^{(2)} \\ a_2 \\ \vdots \\ 0 \end{bmatrix}.$$

而用 $\boldsymbol{H}_2$ 右乘一个矩阵不改变该矩阵的第 1 列和第 2 列, 于是

$$\boldsymbol{A}_3 = \boldsymbol{H}_2\boldsymbol{A}_2\boldsymbol{H}_2 = \begin{bmatrix} * & * & * & \cdots & * \\ a_1 & * & * & \cdots & * \\ 0 & a_2 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix}.$$

这样下去, 经过 $n-2$ 次变换后, $\boldsymbol{A}_1$ 就化为上 Hessenberg 矩阵 $\boldsymbol{A}_{n-1}$, 即

$$\boldsymbol{A}_{n-1} = \boldsymbol{H}_{n-2}\cdots\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A}_1\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2}$$

$$= \begin{bmatrix} * & * & * & * & \cdots & * \\ a_1 & * & * & * & \cdots & * \\ & a_2 & * & * & \cdots & * \\ & & a_3 & * & \cdots & * \\ & & & \ddots & \ddots & \vdots \\ & & & & a_{n-1} & * \end{bmatrix}.$$

如果 $\boldsymbol{A}_1$ 是对称矩阵, 则 $\boldsymbol{A}_{n-1}$ 仍是对称矩阵, 此时 $\boldsymbol{A}_{n-1}$ 将是对称三对角矩阵:

$$\boldsymbol{A}_{n-1} = \begin{bmatrix} * & a_1 & & & \\ a_1 & * & a_2 & & \\ & a_2 & * & a_3 & \\ & & \ddots & \ddots & a_{n-1} \\ & & & a_{n-1} & * \end{bmatrix}.$$

以上利用 Householder 变换约化 $\boldsymbol{A}$ 为上 Hessenberg 形的方法, 可总结为如下实用算法.

算法 7.4 (上 Hessenberg 化)

步 1, 输入 $\boldsymbol{A} = (a_{ij})$, $k := 1$.

步 2, 计算 $n-k$ 阶 Householder 矩阵 $\widetilde{\boldsymbol{H}}_k$, 使

$$\widetilde{\boldsymbol{H}}_k \begin{bmatrix} a_{k+1,k} \\ a_{k+2,k} \\ \vdots \\ a_{n,k} \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

置 $\boldsymbol{A} := \boldsymbol{H}_k\boldsymbol{A}\boldsymbol{H}_k$, 其中 $\boldsymbol{H}_k = \mathrm{diag}(\boldsymbol{I}_k, \widetilde{\boldsymbol{H}}_k)$.

步 3, 若 $k < n-2$, 则 $k := k+1$, 转步 2; 否则输出有关信息, 停算.

算法 7.4 计算出 $\boldsymbol{A}$ 的上 Hessenberg 形就存在 $\boldsymbol{A}$ 所对应的存储单元内, 所需运算量是 $5n^3/3$. 如果需要累计 $\boldsymbol{Q} = \boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2}$, 则还需增加 $2n^3/3$ 的运算量.

将矩阵 $\boldsymbol{A}$ 化为上 Hessenberg 矩阵 (算法 7.4) 的 MATLAB 程序如下:

```
function [A,Q]=mhessen(A)
%用Householder变换化n阶矩阵A为上Hessenberg矩阵
%调用函数:mhouse.m
```

```
n=size(A,1); Q=eye(n);
for k=1:(n-2)
    x=A(k+1:n,k); [v,beta]=mhouse(x);
    H=(eye(length(v))-beta*v*v');
    A(k+1:n,1:n)=H*A(k+1:n,1:n);
    A(1:n,k+1:n)=A(1:n,k+1:n)*H;
    Q=Q*blkdiag(eye(k),H);
end
```

例 7.5 利用 MATLAB 程序 mhessen.m, 将下列矩阵化为上 Hessenberg 矩阵:

$$\boldsymbol{A}=\begin{bmatrix} -1 & 2 & 3 & 5 \\ 2 & -3 & 8 & 1 \\ 3 & 8 & -2 & 7 \\ 5 & 1 & 7 & 6 \end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[-1 2 3 5; 2 -3 8 1; 3 8 -2 7; 5 1 7 6];
>> [A,Q]=mhessen(A)
A =
   -1.0000   -6.1644    0.0000    0.0000
   -6.1644   11.7368    1.8380    0.0000
    0.0000    1.8380   -6.5929    5.9938
    0.0000    0.0000    5.9938   -4.1439
Q =
    1.0000         0         0         0
         0    0.3244   -0.0418   -0.9450
         0    0.4867    0.8640    0.1289
         0    0.8111   -0.5017    0.3007
```

一般来说, 上 Hessenberg 分解是不唯一的, 但可以证明下面的结果.

定理 7.12 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 有如下两个上 Hessenberg 分解:

$$\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{U}=\boldsymbol{H},\quad \boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}=\boldsymbol{G}, \tag{7.40}$$

式中: $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n]$ 和 $\boldsymbol{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n]$ 为 n 阶正交矩阵; $\boldsymbol{H}=(h_{ij})$ 和 $\boldsymbol{G}=(g_{ij})$ 为上 Hessenberg 矩阵. 若 $\boldsymbol{u}_1=\boldsymbol{v}_1$, 且 $\boldsymbol{H}$ 的次对角元 $h_{i+1,i}\neq 0\,(i=1,2,\cdots,n)$, 则存在对角元均为 1 或 -1 的对角矩阵 $\boldsymbol{D}$, 使得

$$\boldsymbol{U}=\boldsymbol{V}\boldsymbol{D},\quad \boldsymbol{H}=\boldsymbol{D}\boldsymbol{G}\boldsymbol{D}, \tag{7.41}$$

即 $\boldsymbol{u}_i=\pm\boldsymbol{v}_i$, $|h_{ij}|=|g_{ij}|$, $i,j=1,2,\cdots,n$.

证明 对矩阵的阶数 n 用归纳法. 对于 $n=1$ 时结论显然成立. 假设 $n=m$ 时结论成立, 即

$$\boldsymbol{u}_i=\varepsilon_i\boldsymbol{v}_i,\quad i=1,2,\cdots,m, \tag{7.42}$$

式中: $\varepsilon_1=1$, $\varepsilon_i=1$ 或 -1, $i=2,\cdots,m$. 下面证明存在 ε_{m+1} 为 1 或 -1 使得

$$\boldsymbol{u}_{m+1}=\varepsilon_{m+1}\boldsymbol{v}_{m+1}.$$

由式 (7.40), 得

$$\boldsymbol{AU}=\boldsymbol{UH},\quad \boldsymbol{AV}=\boldsymbol{VG}.$$

分别比较上面两个矩阵等式两边的第 m 列, 可得

$$\boldsymbol{A}\boldsymbol{u}_m=h_{1m}\boldsymbol{u}_1+\cdots+h_{mm}\boldsymbol{u}_m+h_{m+1,m}\boldsymbol{u}_{m+1}, \tag{7.43}$$

$$\boldsymbol{A}\boldsymbol{v}_m=g_{1m}\boldsymbol{v}_1+\cdots+g_{mm}\boldsymbol{v}_m+g_{m+1,m}\boldsymbol{v}_{m+1}. \tag{7.44}$$

分别在式 (7.43) 和式 (7.44) 两边左乘 $\boldsymbol{u}_i^{\mathrm{T}}$ 和 $\boldsymbol{v}_i^{\mathrm{T}}$ $(i=1,2,\cdots,m)$, 得

$$h_{im}=\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}_m,\quad i=1,2,\cdots,m, \tag{7.45}$$

$$g_{im}=\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_m,\quad i=1,2,\cdots,m. \tag{7.46}$$

由式 (7.42)、式 (7.45) 和式 (7.46), 得

$$h_{im}=\varepsilon_i\varepsilon_m g_{im},\ i=1,2,\cdots,m. \tag{7.47}$$

将式 (7.47) 代入式 (7.43), 并利用式 (7.42) 和式 (7.44), 得

$$\begin{aligned}h_{m+1,m}\boldsymbol{u}_{m+1}&=\boldsymbol{A}\boldsymbol{u}_m-\varepsilon_1\varepsilon_m g_{1m}\boldsymbol{u}_1-\cdots-\varepsilon_m\varepsilon_m g_{mm}\boldsymbol{u}_m\\&=\varepsilon_m(\boldsymbol{A}\boldsymbol{v}_m-\varepsilon_1^2 g_{1m}\boldsymbol{v}_1-\cdots-\varepsilon_m^2 g_{mm}\boldsymbol{v}_m)\\&=\varepsilon_m(\boldsymbol{A}\boldsymbol{v}_m-g_{1m}\boldsymbol{v}_1-\cdots-g_{mm}\boldsymbol{v}_m)\\&=\varepsilon_m g_{m+1,m}\boldsymbol{v}_{m+1}.\end{aligned} \tag{7.48}$$

上式两边取范数, 得

$$|h_{m+1,m}|=|g_{m+1,m}|.$$

由于 $h_{m+1,m}\neq 0$, 故式 (7.48) 蕴含着

$$\boldsymbol{u}_{m+1}=\varepsilon_{m+1}\boldsymbol{v}_{m+1},$$

式中: $\varepsilon_{m+1}=1$ 或 -1. 证毕. □

注 7.3 一个上 Hessenberg 矩阵 $\boldsymbol{H}=(h_{ij})$, 如果其次对角元均不为零, 即 $h_{i+1,i}\neq 0$, $i=1,2,\cdots,n-1$, 则它是不可约的. 定理 7.12 表明, 如果 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\boldsymbol{H}$ 为不可约的上 Hessenberg 矩阵, 则 $\boldsymbol{Q}$ 和 $\boldsymbol{H}$ 完全由 $\boldsymbol{Q}$ 的第 1 列确定 (这里是在相差一个正负号意义下的唯一).

7.4.2 上 Hessenberg 矩阵的 QR 分解

对于上 Hessenberg 矩阵

$$\boldsymbol{H}=\begin{bmatrix} h_{11}^{(1)} & h_{12}^{(1)} & h_{13}^{(1)} & \cdots & h_{1n}^{(1)} \\ h_{21}^{(1)} & h_{22}^{(1)} & h_{23}^{(1)} & \cdots & h_{2n}^{(1)} \\ & h_{32}^{(1)} & h_{33}^{(1)} & \cdots & h_{2n}^{(1)} \\ & & \ddots & \ddots & \vdots \\ & & & h_{n,n-1}^{(1)} & h_{nn}^{(1)} \end{bmatrix},$$

通常可以通过 $n-1$ 次 Givens 变换将它化成上三角矩阵, 从而得到 $\boldsymbol{H}$ 的 QR 分解式. 具体步骤是:

(1) 记 $\boldsymbol{H}_1=\boldsymbol{H}$. 设 $h_{21}^{(1)}\neq 0$ (否则可进行下一步), 取 Givens 矩阵

$$\boldsymbol{G}_{21}=\begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix},$$

式中:

$$c_1=\frac{h_{11}^{(1)}}{r_1},\quad s_1=\frac{h_{21}^{(1)}}{r_1},\quad r_1=\sqrt{(h_{11}^{(1)})^2+(h_{21}^{(1)})^2}.$$

则

$$\boldsymbol{G}_{21}\boldsymbol{H}_1=\begin{bmatrix} r_1 & h_{12}^{(2)} & h_{13}^{(2)} & \cdots & h_{1n}^{(2)} \\ 0 & h_{22}^{(2)} & h_{23}^{(2)} & \cdots & h_{2n}^{(2)} \\ & h_{32}^{(2)} & h_{33}^{(2)} & \cdots & h_{3n}^{(2)} \\ & & \ddots & \ddots & \vdots \\ & & & h_{n,n-1}^{(2)} & h_{nn}^{(2)} \end{bmatrix}:=\boldsymbol{H}_2.$$

(2) 设 $h_{32}^{(1)}\neq 0$ (否则可进行下一步), 再取 Givens 矩阵

$$\boldsymbol{G}_{32}=\begin{bmatrix} 1 & & & & & \\ & c_2 & s_2 & & & \\ & -s_2 & c_2 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix},$$

式中:

$$c_2=\frac{h_{22}^{(2)}}{r_2},\quad s_2=\frac{h_{32}^{(2)}}{r_2},\quad r_2=\sqrt{(h_{22}^{(2)})^2+(h_{32}^{(2)})^2}.$$

则

$$\boldsymbol{G}_{32}\boldsymbol{H}_2=\begin{bmatrix} r_1 & h_{12}^{(3)} & h_{13}^{(3)} & \cdots & h_{1,n-1}^{(3)} & h_{1n}^{(3)} \\ 0 & r_2 & h_{23}^{(3)} & \cdots & h_{2,n-1}^{(3)} & h_{2n}^{(3)} \\ & 0 & h_{33}^{(3)} & \cdots & h_{3,n-1}^{(3)} & h_{3n}^{(3)} \\ & & h_{43}^{(3)} & \cdots & h_{4,n-1}^{(3)} & h_{4n}^{(3)} \\ & & & \ddots & \vdots & \vdots \\ & & & & h_{n,n-1}^{(3)} & h_{nn}^{(3)} \end{bmatrix}:=\boldsymbol{H}_3.$$

(3) 假设上述过程已经进行了 $k-1$ 步, 有

$$\boldsymbol{H}_k=\boldsymbol{G}_{k,k-1}\boldsymbol{H}_{k-1}=\begin{bmatrix} r_1 & \cdots & h_{1,k-1}^{(k)} & h_{1k}^{(k)} & \cdots & h_{1,n-1}^{(k)} & h_{1n}^{(k)} \\ & \ddots & & & & & \\ & & r_{k-1} & h_{k-1,k}^{(k)} & \cdots & h_{k-1,n-1}^{(k)} & h_{k-1,n}^{(k)} \\ & & & h_{kk}^{(k)} & \cdots & h_{k,n-1}^{(k)} & h_{kn}^{(k)} \\ & & & h_{k+1,k}^{(k)} & \cdots & h_{k+1,n-1}^{(k)} & h_{k+1,n}^{(k)} \\ & & & & \ddots & \vdots & \vdots \\ & & & & & h_{n,n-1}^{(k)} & h_{nn}^{(k)} \end{bmatrix}.$$

设 $h_{k+1,k}^{(k)}\neq 0$, 取 Givens 矩阵

$$\boldsymbol{G}_{k+1,k}=\begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & c_k & s_k & & & \\ & & & -s_k & c_k & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{bmatrix},$$

式中:

$$c_k=\frac{h_{kk}^{(k)}}{r_k},\quad s_k=\frac{h_{k+1,k}^{(k)}}{r_k},\quad r_k=\sqrt{(h_{kk}^{(k)})^2+(h_{k+1,k}^{(k)})^2}.$$

于是

$$\boldsymbol{G}_{k+1,k}\boldsymbol{H}_k=\begin{bmatrix} r_1 & \cdots & h_{1,k}^{(k+1)} & h_{1,k+1}^{(k+1)} & \cdots & h_{1,n-1}^{(k+1)} & h_{1n}^{(k+1)} \\ & \ddots & & & & & \\ & & r_k & h_{k,k+1}^{(k+1)} & \cdots & h_{k,n-1}^{(k+1)} & h_{kn}^{(k+1)} \\ & & & h_{k+1,k+1}^{(k+1)} & \cdots & h_{k+1,n-1}^{(k+1)} & h_{k+1,n}^{(k+1)} \\ & & & h_{k+2,k+1}^{(k+1)} & \cdots & h_{k+2,n-1}^{(k+1)} & h_{k+2,n}^{(k+1)} \\ & & & & \ddots & \vdots & \vdots \\ & & & & & h_{n,n-1}^{(k+1)} & h_{nn}^{(k+1)} \end{bmatrix}:=\boldsymbol{H}_{k+1}.$$

因此, 最多作 $n-1$ 次 Givens 变换, 即得

$$\boldsymbol{G}_{n,n-1}\cdots\boldsymbol{G}_{32}\boldsymbol{G}_{21}\boldsymbol{H}=\begin{bmatrix} r_1 & h_{12}^{(n)} & h_{13}^{(n)} & \cdots & h_{1n}^{(n)} \\ & r_2 & h_{23}^{(n)} & \cdots & h_{2n}^{(n)} \\ & & r_3 & \cdots & h_{3n}^{(n)} \\ & & & \ddots & \vdots \\ & & & & r_n \end{bmatrix}=\boldsymbol{R}.$$

因为 $\boldsymbol{G}_{k,k-1}\,(k=2,\cdots,n)$ 均为正交阵, 故

$$\boldsymbol{H}=\boldsymbol{G}_{21}^{\mathrm{T}}\boldsymbol{G}_{32}^{\mathrm{T}}\cdots\boldsymbol{G}_{n,n-1}^{\mathrm{T}}\boldsymbol{R}=\boldsymbol{Q}\boldsymbol{R},$$

式中: $\boldsymbol{Q}=\boldsymbol{G}_{21}^{\mathrm{T}}\boldsymbol{G}_{32}^{\mathrm{T}}\cdots\boldsymbol{G}_{n,n-1}^{\mathrm{T}}$ 仍为正交阵.

可算出完成这一过程的运算量约为 $4n^2$, 比一般矩阵的 QR 分解的运算量 $O(n^3)$ 少了一个数量级.

值得注意的是, 可以证明 $\widetilde{\boldsymbol{H}}=\boldsymbol{R}\boldsymbol{Q}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{Q}$ 仍为上 Hessenberg 矩阵, 于是可按上述步骤一直迭代下去, 直到 $\boldsymbol{H}$ 正交相似于上三角矩阵或块上三角矩阵 (对角块为 1×1 或 2×2 矩阵) 为止, 从而求得矩阵 $\boldsymbol{H}$ 的全部特征值和相应的特征向量.

上 Hessenberg 矩阵 QR 分解的 MATLAB 程序如下:

```
function A=hessen_qrtran(A,m)
%本程序输入n阶上Hessenberg矩阵A,用Givens变换对其左上角m阶主子块
%进行QR分解,再作相似变换,最后输出变换后的上Hessenberg矩阵A.
Q=eye(m);
for i=1:m-1
   xi=A(i,i); xk=A(i+1,i);
   if xk~=0
      d=sqrt(xi^2+xk^2);
```

```
        c=xi/d; s=xk/d;
        G=[c, s; -s, c];
        A(i:i+1,i:m)=G*A(i:i+1,i:m);
        Q(1:m,i:i+1)=Q(1:m,i:i+1)*G';
    end
end
%Q*A,  %验证Q*R=A
A(1:m,1:m)=A(1:m,1:m)*Q;
```

例 7.6 利用程序将上 Hessenberg 矩阵 $\boldsymbol{A}$ 进行 QR 变换, 其中

$$\boldsymbol{A}=\begin{bmatrix} 2 & 3 & 5 & 7 & 8 \\ 4 & 2 & 3 & 5 & 9 \\ 0 & 8 & 3 & 6 & 2 \\ 0 & 0 & 7 & 1 & 3 \\ 0 & 0 & 0 & 6 & 9 \end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[2 3 5 7 8; 4 2 3 5 9; 0 8 3 6 2; 0 0 7 1 3; 0 0 0 6 9];
>> A=hessen_qrtran(A,5)
A =
    4.8000    5.3682    7.6664   10.5793   -4.9445
    7.3321    2.7238    4.9535    0.3480   -6.8327
         0    7.2218    1.1745    2.7581   -3.2311
         0         0    6.0298    7.7534   -4.9785
         0         0         0   -1.5283    0.5483
```

7.4.3 基本 QR 方法

本节介绍求一般方阵全部特征值的 QR 方法. 令 $\boldsymbol{A}_1=\boldsymbol{A}$, 对 $\boldsymbol{A}_1$ 作 QR 分解:

$$\boldsymbol{A}_1=\boldsymbol{Q}_1\boldsymbol{R}_1,$$

然后令 $\boldsymbol{A}_2=\boldsymbol{R}_1\boldsymbol{Q}_1$, 再对 $\boldsymbol{A}_2$ 作 QR 分解:

$$\boldsymbol{A}_2=\boldsymbol{Q}_2\boldsymbol{R}_2,$$

并令 $\boldsymbol{A}_3=\boldsymbol{R}_2\boldsymbol{Q}_2$, 这样下去就得到一个矩阵序列 $\{\boldsymbol{A}_k\}$, 其产生过程可概述如下:

$$\begin{cases} \boldsymbol{A}_1=\boldsymbol{A}, \\ \boldsymbol{A}_k=\boldsymbol{Q}_k\boldsymbol{R}_k, \qquad k=1,2,\cdots. \\ \boldsymbol{A}_{k+1}=\boldsymbol{R}_k\boldsymbol{Q}_k, \end{cases} \tag{7.49}$$

容易证明, $\boldsymbol{A}_{k+1}$ 与 $\boldsymbol{A}_k$ 相似, 故 $\{\boldsymbol{A}_k\}$ 有相同的特征值.

在一定条件下, $\{\boldsymbol{A}_k\}$ 本质上收敛于上三角矩阵 (或分块上三角矩阵). 若它们收敛于上三角矩阵, 则该上三角矩阵的对角元就是原矩阵 $\boldsymbol{A}$ 的全部特征值; 若收敛于分块上三角矩阵, 则这些分块矩阵的特征值也就是 $\boldsymbol{A}$ 的特征值.

由于当 $\boldsymbol{A}$ 为一般的实矩阵时, $\{\boldsymbol{A}_k\}$ 的收敛速度较慢, 故在 QR 方法的实际应用中, 通常先将 $\boldsymbol{A}$ 化为相似的上 Hessenberg 矩阵, 再求特征值以加快收敛速度. 它的计算过程如下.

算法 7.5 (基本 QR 方法)

步 1, 输入上 Hessenberg 矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$.

步 2, 记 $\boldsymbol{A}_1 := \boldsymbol{A}$. 对于 $k = 1, 2, \cdots$, 有

(1) $\boldsymbol{A}_k = \boldsymbol{Q}_k \boldsymbol{R}_k$ (QR 分解).

(2) $\boldsymbol{A}_{k+1} = \boldsymbol{Q}_k^{\mathrm{T}} \boldsymbol{A}_k \boldsymbol{Q}_k = \boldsymbol{R}_k \boldsymbol{Q}_k$ (正交相似变换).

基本 QR 方法的 MATLAB 程序如下:

```
function [iter,D]=qr_eig(A,tol,N)
%用基本QR算法求n阶实方阵A的全部特征值
%输入:A为实对称矩阵,tol为控制精度,N为最大迭代次数
%输出:iter为迭代次数,D为A的全部特征值
%调用函数:mhessen.m,hessen_qrtran.m,eig-仅用于1,2矩阵
if nargin<3, N=500; end
if nargin<2, tol=1e-5; end
n=size(A,1); D=zeros(n,1);
i=n; m=n; iter=0;  %初始化
A=mhessen(A);  %化矩阵A为Hessenberg矩阵
while (iter<=N)  %用基本QR算法进行迭代
    iter=iter+1;
    if m<=2
        la=eig(A(1:m,1:m)); D(1:m)=la';
        break;
    end
   %对上Hessenberg矩阵作QR分解并作正交相似变换
   A=hessen_qrtarn(A,m);
   %下面的程序段判断是否终止
   for k=m-1:-1:1
      if abs(A(k+1,k))<tol
          if m-k<=2
             la=eig(A(k+1:m,k+1:m));
             j=i-m+k+1; D(j:i)=la';
             i=j-1; m=k; break;
```

```
        end
    end
  end
end
```

例 7.7 利用程序 qr_eig.m, 求下列矩阵的全部特征值:

$$\boldsymbol{A}=\begin{bmatrix} 3 & 2 & 3 & 4 & 5 & 6 & 7 \\ 11 & 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 8 & 9 & 1 & 2 & 3 & 4 \\ -4 & 2 & 9 & 11 & 13 & 15 & 8 \\ -1 & -2 & -3 & -1 & -1 & -1 & -1 \\ 3 & 2 & 3 & 4 & 13 & 15 & 8 \\ -2 & -2 & -3 & -4 & -5 & -3 & -3 \end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[3 2 3 4 5 6 7;11 1 2 3 4 5 6;2 8 9 1 2 3 4; ...
      -4 2 9 11 13 15 8; -1 -2 -3 -1 -1 -1 -1; ...
      3 2 3 4 13 15 8; -2 -2 -3 -4 -5 -3 -3];
>> [iter,D]=qr_eig(A)
iter =
      622
D =
   18.4123 + 0.0000i
   11.1805 + 0.0000i
    1.7099 - 4.2522i
    1.7099 + 4.2522i
    4.4983 + 0.0000i
   -2.2327 + 0.0000i
   -0.2783 + 0.0000i
```

下面分析基本 QR 方法的收敛性.

定义 7.4 若由 QR 方法产生的序列 $\{\boldsymbol{A}_k\}$ 当 $k\to\infty$ 收敛于分块上三角矩阵 (对角块为一阶或二阶子块), 则称 QR 方法是收敛的. 若序列 $\{\boldsymbol{A}_k\}$ 当 $k\to\infty$ 时, 其对角元均收敛且严格下三角部分元素收敛于 0, 则称 $\{\boldsymbol{A}_k\}$ 基本收敛到上三角阵.

值得注意的是, 基本收敛的概念并未指出 $\{\boldsymbol{A}_k\}$ 严格上三角部分元素是否收敛. 但对求矩阵 $\boldsymbol{A}$ 的特征值而言, 基本收敛足够了.

算法 7.5 具有下列性质.

性质 7.1 *在算法 7.5 中, 若记*

$$\widetilde{\boldsymbol{Q}}_k = \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_k, \quad \widetilde{\boldsymbol{R}}_k = \boldsymbol{R}_k\boldsymbol{R}_{k-1}\cdots\boldsymbol{R}_1, \tag{7.50}$$

显然 $\widetilde{\boldsymbol{Q}}_k$ 为正交矩阵, $\widetilde{\boldsymbol{R}}_k$ 为上三角矩阵. 则有

(1) $\boldsymbol{Q}_k$ *和* $\boldsymbol{A}_{k+1}$ *都是上* Hessenberg *矩阵.*

(2) $\boldsymbol{A}_{k+1} = \widetilde{\boldsymbol{Q}}_k^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}}_k \sim \boldsymbol{A}$ *(相似性).* (7.51)

(3) $\boldsymbol{A}^k = \widetilde{\boldsymbol{Q}}_k\widetilde{\boldsymbol{R}}_k$ *($\boldsymbol{A}$ 的 k 次幂 $\boldsymbol{A}^k$ 的* QR *分解).* (7.52)

证明 用归纳法证明性质 (3). 当 $k=1$ 时, 有

$$\boldsymbol{A} = \boldsymbol{A}_1 = \boldsymbol{Q}_1\boldsymbol{R}_1 = \widetilde{\boldsymbol{Q}}_1\widetilde{\boldsymbol{R}}_1.$$

设 $\boldsymbol{A}^{k-1} = \widetilde{\boldsymbol{Q}}_{k-1}\widetilde{\boldsymbol{R}}_{k-1}$, 则

$$\begin{aligned}\boldsymbol{A}^k &= \boldsymbol{A}(\widetilde{\boldsymbol{Q}}_{k-1}\widetilde{\boldsymbol{R}}_{k-1}) = \widetilde{\boldsymbol{Q}}_{k-1}(\widetilde{\boldsymbol{Q}}_{k-1}^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}}_{k-1})\widetilde{\boldsymbol{R}}_{k-1}\\ &= \widetilde{\boldsymbol{Q}}_{k-1}\boldsymbol{A}_k\widetilde{\boldsymbol{R}}_{k-1} = \widetilde{\boldsymbol{Q}}_{k-1}\boldsymbol{Q}_k\boldsymbol{R}_k\widetilde{\boldsymbol{R}}_{k-1} = \widetilde{\boldsymbol{Q}}_k\widetilde{\boldsymbol{R}}_k.\end{aligned}$$

证毕. □

性质 (1) 称为上 Hessenberg 形在 QR 变换下的不变性. 它的意义是, 算法 7.5 可以始终对上 Hessenberg 矩阵进行. 这时, QR 分解中每列的消元只要作一次 Givens 变换, 从而简化了 QR 变换的计算. 性质 (2) 表示, 由 QR 方法生成的矩阵序列 $\{\boldsymbol{A}_k\}$ 保持原矩阵 $\boldsymbol{A}$ 的特征值不变. 性质 (3) 说明, QR 方法即 QR 变换过程, 实质上是对 $\boldsymbol{A}$ 的 k 次幂 $\boldsymbol{A}^k$ 进行 QR 分解的过程. 由此可见, QR 方法与幂法有内在的联系. 下面给出 QR 方法的一个最简单的收敛性定理, 它表明, 在一定条件下可以把 QR 方法看成幂法的推广. 为此, 先给出下面的引理.

引理 7.1 *设 $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{Q}_{n-1}] \in \mathbb{R}^{n\times n}$ 为正交矩阵, 其中 $\boldsymbol{q}_1$ 为 $\boldsymbol{Q}$ 的第 1 列. 对于矩阵 $\boldsymbol{A}$, 若记*

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1 & \boldsymbol{\beta}^{\mathrm{T}} \\ \boldsymbol{\alpha} & \boldsymbol{C} \end{bmatrix}, \quad \text{其中 } \boldsymbol{\alpha} = \boldsymbol{Q}_{n-1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1 \in \mathbb{R}^{n-1},$$

则有

$$\|\boldsymbol{\alpha}\|_2 = \|\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1\|_2. \tag{7.53}$$

证明 因为对任意的 $\boldsymbol{x} \in \mathbb{R}^n$ 都有 $\|\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{x}\|_2 = \|\boldsymbol{x}\|_2$. 故

$$\begin{aligned}&\|\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1\|_2 = \|[\boldsymbol{q}_1, \boldsymbol{Q}_{n-1}]^{\mathrm{T}}[\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1]\|_2\\ =& \left\| \begin{bmatrix} \boldsymbol{q}_1^{\mathrm{T}}[\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1] \\ \boldsymbol{Q}_{n-1}^{\mathrm{T}}[\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1] \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} 0 \\ \boldsymbol{\alpha} \end{bmatrix} \right\|_2 = \|\boldsymbol{\alpha}\|_2.\end{aligned}$$

证毕. □

定理 7.13 设对称矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 满足

$$|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|>0,$$

对应的规范正交化特征向量为 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$. 如果单位坐标向量

$$\boldsymbol{e}_1=(1,0,\cdots,0)^{\mathrm{T}}=\sum_{i=1}^{n}\alpha_i\boldsymbol{x}_i$$

中的 $\alpha_1\neq 0$, 那么由算法 7.5 生成的矩阵序列 $\{\boldsymbol{A}_k\}$ 具有收敛性质

$$\lim_{k\to\infty}\boldsymbol{A}_k\boldsymbol{e}_1=\lambda_1\boldsymbol{e}_1. \tag{7.54}$$

证明 记 $\widetilde{\boldsymbol{Q}}_k$ 的第 1 列为 $\widetilde{\boldsymbol{q}}_1^{(k)}=\widetilde{\boldsymbol{Q}}_k\boldsymbol{e}_1$, $\widetilde{\boldsymbol{R}}_k$ 的第 1 个对角元为 $\widetilde{r}_{11}^{(k)}$, 则由式 (7.52) 可知

$$\boldsymbol{A}^k\boldsymbol{e}_1=\widetilde{\boldsymbol{Q}}_k\widetilde{\boldsymbol{R}}_k\boldsymbol{e}_1=\widetilde{\boldsymbol{Q}}_k(\widetilde{r}_{11}^{(k)}\boldsymbol{e}_1)=\widetilde{r}_{11}^{(k)}\widetilde{\boldsymbol{q}}_1^{(k)}.$$

注意到 $\|\widetilde{\boldsymbol{q}}_1^{(k)}\|_2=1$, $\widetilde{r}_{11}^{(k)}\neq 0$ (因矩阵 $\boldsymbol{A}$ 非奇异). 从而 $|\widetilde{r}_{11}^{(k)}|=\|\boldsymbol{A}^k\boldsymbol{e}_1\|_2$. 于是, 根据幂法的收敛性, 有

$$\lim_{k\to\infty}\widetilde{\boldsymbol{q}}_1^{(k)}=\lim_{k\to\infty}\frac{\boldsymbol{A}^k\boldsymbol{e}_1}{\widetilde{r}_{11}^{(k)}}=\boldsymbol{z}_1, \tag{7.55}$$

式中: $\boldsymbol{z}_1$ 为矩阵 $\boldsymbol{A}$ 的对应于 λ_1 的规范化特征向量 (可以相差一个常数因子 ±1). 进而, 由式 (7.51), 可以把 $\boldsymbol{A}_{k+1}$ 写成

$$\begin{aligned}\boldsymbol{A}_{k+1}&=\widetilde{\boldsymbol{Q}}_k^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}}_k=[\widetilde{\boldsymbol{q}}_1^{(k)},\widetilde{\boldsymbol{Q}}_{k-1}]^{\mathrm{T}}\boldsymbol{A}[\widetilde{\boldsymbol{q}}_1^{(k)},\widetilde{\boldsymbol{Q}}_{k-1}]\\&=\begin{bmatrix}a_{11}^{(k+1)}&*\\ \boldsymbol{\alpha}^{(k+1)}&*\end{bmatrix},\end{aligned}$$

式中: $a_{11}^{(k+1)}=(\widetilde{\boldsymbol{q}}_1^{(k)})^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{q}}_1^{(k)}$, $\boldsymbol{\alpha}^{(k+1)}=(\widetilde{\boldsymbol{Q}}_{k-1}^{(k)})^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{q}}_1^{(k)}$. 根据式 (7.55), 得

$$\lim_{k\to\infty}a_{11}^{(k+1)}=\lim_{k\to\infty}(\widetilde{\boldsymbol{q}}_1^{(k)})^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{q}}_1^{(k)}=\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{z}_1=\lambda_1.$$

故由引理 7.1, 得

$$\begin{aligned}\lim_{k\to\infty}\|\boldsymbol{\alpha}^{(k+1)}\|_2&=\lim_{k\to\infty}\|\boldsymbol{A}\widetilde{\boldsymbol{q}}_1^{(k)}-[(\widetilde{\boldsymbol{q}}_1^{(k)})^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{q}}_1^{(k)}]\widetilde{\boldsymbol{q}}_1^{(k)}\|_2\\&=\|\boldsymbol{A}\boldsymbol{z}_1-[\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{z}_1]\boldsymbol{z}_1\|_2=0.\end{aligned}$$

因此, $\lim\limits_{k\to\infty}\boldsymbol{\alpha}^{(k+1)}=\boldsymbol{0}$. 于是有

$$\lim_{k\to\infty}\boldsymbol{A}_{k+1}=\begin{bmatrix}\lambda_1&*\\ \boldsymbol{0}&*\end{bmatrix},$$

即式 (7.54) 成立. 证毕. □

作为定理 7.13 的推广, 有下面的收敛性结果.

定理 7.14 设 $\boldsymbol{A}=\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$, 其中 $\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)$. 如果 ① $|\lambda_1|>|\lambda_2|>\cdots>|\lambda_n|>0$; ② $\boldsymbol{X}^{-1}$ 具有 LU 分解: $\boldsymbol{X}^{-1}=\boldsymbol{L}\boldsymbol{U}$. 则 QR 方法基本收敛于上三角矩阵.

证明 由式 (7.51) 可知, 只需分析 $\widetilde{\boldsymbol{Q}}_k$ 的极限情况. 而由式 (7.52), 只需分析 $\boldsymbol{A}^k$ 的极限情况. 注意到

$$\boldsymbol{A}^k=\boldsymbol{X}\boldsymbol{\Lambda}^k\boldsymbol{X}^{-1}=\boldsymbol{X}\boldsymbol{\Lambda}^k\boldsymbol{L}\boldsymbol{U}=\boldsymbol{X}(\boldsymbol{\Lambda}^k\boldsymbol{L}\boldsymbol{\Lambda}^{-k})\boldsymbol{\Lambda}^k\boldsymbol{U}.$$

令

$$\boldsymbol{\Lambda}^k\boldsymbol{L}\boldsymbol{\Lambda}^{-k}=\boldsymbol{I}+\boldsymbol{E}_k,$$

则

$$\boldsymbol{A}^k=\boldsymbol{X}(\boldsymbol{I}+\boldsymbol{E}_k)\boldsymbol{\Lambda}^k\boldsymbol{U}.$$

由于 $\boldsymbol{L}$ 是单位下三角形, 故

$$(\boldsymbol{E}_k)_{ij}=\begin{cases}0, & i\leqslant j,\\ l_{ij}(\lambda_i/\lambda_j)^k, & i>j.\end{cases}$$

由假设条件 ① 知, $\boldsymbol{E}_k\to\boldsymbol{O}$ 且 $(\boldsymbol{E}_k)_{ij}$ 的收敛速度是 $|\lambda_i/\lambda_j|$.

设 $\boldsymbol{X}=\boldsymbol{Q}\boldsymbol{R}$ 且 $\boldsymbol{R}$ 的对角元均为正数, 则有

$$\begin{aligned}\boldsymbol{A}^k&=\boldsymbol{Q}\boldsymbol{R}(\boldsymbol{I}+\boldsymbol{E}_k)\boldsymbol{\Lambda}^k\boldsymbol{U}\\&=\boldsymbol{Q}(\boldsymbol{I}+\boldsymbol{R}\boldsymbol{E}_k\boldsymbol{R}^{-1})\boldsymbol{R}\boldsymbol{\Lambda}^k\boldsymbol{U}.\end{aligned}$$

因为 $\boldsymbol{E}_k\to\boldsymbol{O}\,(k\to\infty)$, 故当 k 充分大时, $\boldsymbol{I}+\boldsymbol{R}\boldsymbol{E}_k\boldsymbol{R}^{-1}$ 非奇异, 所以有唯一的 QR 分解 $\boldsymbol{I}+\boldsymbol{R}\boldsymbol{E}_k\boldsymbol{R}^{-1}=\widehat{\boldsymbol{Q}}_k\widehat{\boldsymbol{R}}_k$ ($\widehat{\boldsymbol{R}}_k$ 的对角元为正), 而且当 $k\to\infty$ 时, $\widehat{\boldsymbol{Q}}_k\to\boldsymbol{I}$, $\widehat{\boldsymbol{R}}_k\to\boldsymbol{I}$. 此时, $\boldsymbol{A}^k$ 有如下分解

$$\boldsymbol{A}^k=(\boldsymbol{Q}\widehat{\boldsymbol{Q}}_k)(\widehat{\boldsymbol{R}}_k\boldsymbol{R}\boldsymbol{\Lambda}^k\boldsymbol{U}).$$

妨碍上式成为 $\boldsymbol{A}^k$ 的 QR 分解的仅仅是上式右端第 2 个因子 (上三角矩阵) 的对角元可能非正. 为补救这一点, 可引入两个对角正交矩阵

$$\boldsymbol{D}_1=\mathrm{diag}\left(\frac{\lambda_1}{|\lambda_1|},\cdots,\frac{\lambda_n}{|\lambda_n|}\right),\quad \boldsymbol{D}_2=\mathrm{diag}\left(\frac{\boldsymbol{U}_{11}}{|\boldsymbol{U}_{11}|},\cdots,\frac{\boldsymbol{U}_{nn}}{|\boldsymbol{U}_{nn}|}\right),$$

式中: $\boldsymbol{U}_{ii}\,(i=1,2,\cdots,n)$ 为矩阵 $\boldsymbol{U}$ 的对角元. 于是

$$\boldsymbol{A}^k=\big((\boldsymbol{Q}\widehat{\boldsymbol{Q}}_k)(\boldsymbol{D}_2\boldsymbol{D}_1^k)\big)\big(\boldsymbol{D}_1^{-k}\boldsymbol{D}_2^{-1}\widehat{\boldsymbol{R}}_k\boldsymbol{R}\boldsymbol{\Lambda}^k\boldsymbol{U}\big)$$

是 $\boldsymbol{A}^k$ 的唯一 QR 分解. 从而由式 (7.51), 有

$$\boldsymbol{A}_{k+1}=\big(\boldsymbol{Q}\widehat{\boldsymbol{Q}}_k\boldsymbol{D}_2\boldsymbol{D}_1^k\big)^{\mathrm{T}}\boldsymbol{A}\big(\boldsymbol{Q}\widehat{\boldsymbol{Q}}_k\boldsymbol{D}_2\boldsymbol{D}_1^k\big).$$

将 $\boldsymbol{A}=\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}=\boldsymbol{Q}\boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1}\boldsymbol{Q}^{-1}$ 代入上式, 得

$$\boldsymbol{A}_{k+1}=\big(\boldsymbol{D}_2\boldsymbol{D}_1^k\big)^{\mathrm{T}}\big(\widehat{\boldsymbol{Q}}_k^{-1}\boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1}\widehat{\boldsymbol{Q}}_k\big)\big(\boldsymbol{D}_2\boldsymbol{D}_1^k\big).$$

因为 $\widehat{\boldsymbol{Q}}_k^{-1}\boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1}\widehat{\boldsymbol{Q}}_k\to\boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1}\equiv\bar{\boldsymbol{R}}$ (上三角矩阵), 所以 $\boldsymbol{A}_k$ 的对角线以下元素收敛于 0. 因为 $\boldsymbol{D}_1^k$ 可能不收敛, 故 $\boldsymbol{A}_k$ 基本收敛于 $\bar{\boldsymbol{R}}$. 证毕. □

7.4.4 带原点位移的 QR 方法

定理 7.14 表明 $\boldsymbol{A}_k$ 的对角元 $a_{ii}^{(k)} \to \lambda_i\,(k \to \infty)$. 从证明过程中可以看出 $\boldsymbol{A}_k$ 的下三角部分的元素趋于 0 的速度由 $k \to \infty$ 时, $\widehat{\boldsymbol{Q}}_k \to \boldsymbol{I}$ 和 $\widehat{\boldsymbol{R}}_k \to \boldsymbol{I}$ 的速度, 亦即由 $\boldsymbol{E}_k \to \boldsymbol{O}$ 的速度所决定. 而从矩阵 $\boldsymbol{E}_k$ 的构成可以看出, $\boldsymbol{E}_k$ 的第 i 行元素趋于 0 的速度由 $|\lambda_i/\lambda_{i-1}|$ 决定, 第 i 列元素趋于 0 的速度由 $|\lambda_{i+1}/\lambda_i|$ 决定. 可以证明, $\boldsymbol{A}_k$ 的下三角部分的元素趋于 0 的情况也是这样. 所以 $a_{ii}^{(k)} \to \lambda_i$ 的速度由 $\boldsymbol{A}_k$ 的第 i 行和第 i 列的下三角部分的元素趋于 0 的速度确定, 这个速度即为 $O(\rho_i)$, 其中

$$\rho_i = \max\left\{\frac{|\lambda_i|}{|\lambda_{i-1}|},\ \frac{|\lambda_{i+1}|}{|\lambda_i|},\ \lambda_0 = +\infty,\ \lambda_{n+1} = 0\right\}.$$

在实际计算中, 线性收敛速度是不令人满意的, 特别是当 ρ_i 不算很小时, 收敛将是十分缓慢的. 为此, 将使用原点位移的方法进行加速. 现在 $\rho_n = |\lambda_n/\lambda_{n-1}|$, 如果将算法用于矩阵 $\boldsymbol{A} - s\boldsymbol{I}$, 则 $a_{nn}^{(k)}$ 将以商 $|\lambda_n - s|/|\lambda_{n-1} - s|$ 线性收敛于 $\lambda_n - s$. 当 s 是 λ_n 的一个较好的近似时, 收敛是很快的. 基于这个想法, 可构造原点位移 QR 方法如下.

算法 7.6 (原点位移 QR 方法)

步 1, 输入上 Hessenberg 矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$.

步 2, 记 $\boldsymbol{A}_1 := \boldsymbol{A}$. 对于 $k = 1, 2, \cdots$,

(1) $\boldsymbol{A}_k - s_k\boldsymbol{I} = \boldsymbol{Q}_k\boldsymbol{R}_k$ (QR 分解).

(2) $\boldsymbol{A}_{k+1} = \boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}_k\boldsymbol{Q}_k = \boldsymbol{R}_k\boldsymbol{Q}_k + s_k\boldsymbol{I}$ (正交相似变换).

其中, 由 $\boldsymbol{A}_k$ 到 $\boldsymbol{A}_{k+1}$ 的变换称为原点位移的 QR 变换.

与算法 7.5 相类似, 由此生成的矩阵序列 $\{\boldsymbol{A}_k\}$ 和 $\{\boldsymbol{Q}_k\}$ 都是上 Hessenberg 形, 并且 $\boldsymbol{A}_k$ 与 $\boldsymbol{A}$ 相似. 现在的问题是如何选择 s_k 使得收敛速度加快. 下面讨论一类特殊情形.

假设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 满足定理 7.13 的条件, 并设由基本 QR 方法生成的 $\boldsymbol{A}_k$ 右下角 2 阶子矩阵

$$\boldsymbol{J}_k = \begin{bmatrix} a_{n-1,n-1}^{(k)} & a_{n-1,n}^{(k)} \\ a_{n,n-1}^{(k)} & a_{n,n}^{(k)} \end{bmatrix} \tag{7.56}$$

的特征值为 $\lambda_{n-1}^{(k)}$ 和 $\lambda_n^{(k)}$, 则有

$$\lim_{k\to\infty} a_{n,n}^{(k)} = \lambda_n,\quad \lim_{k\to\infty} \lambda_{n-1}^{(k)} = \lambda_{n-1},\quad \lim_{k\to\infty} \lambda_n^{(k)} = \lambda_n. \tag{7.57}$$

下面考虑位移量 s_k 的取法. 因为矩阵 $\boldsymbol{A}_k - s_k\boldsymbol{I}$ 的特征值 $\mu_k = \lambda_k - s_k$, 因此, 如果取 $s_k \approx \lambda_n$, 那么在定理 7.13 的条件下, 有

$$\left|\frac{\mu_{k+1}}{\mu_k}\right| = \left|\frac{\lambda_{k+1} - \lambda_n}{\lambda_k - \lambda_n}\right| < \left|\frac{\lambda_{k+1}}{\lambda_k}\right|,\quad k = 1, 2, \cdots, n-1.$$

这表明算法 7.6 的收敛速度比算法 7.5 快. 因此, 由子矩阵 (7.56) 的收敛性质 (7.57), 位移量 s_k 有下列两种取法:

(1) $s_k = a_{nn}^{(k)}$.

(2) 当 $\lambda_{n-1}^{(k)}$ 和 $\lambda_n^{(k)}$ 为实数时, 取 s_k 为其中与 $a_{nn}^{(k)}$ 最接近的一个.

这两种位移策略特别适用于对称矩阵, 因为此时子矩阵 $\boldsymbol{J}_k$ 对称, 从而它的两个特征值都是实数. 可以证明, 采用这种位移策略的算法 7.6, $\boldsymbol{A}_k$ 基本收敛于上三角矩阵, 收敛是二阶的, 并且 $a_{n,n-1}^{(k)}$ 最先趋于零, 从而首先得到绝对值最小的特征值 λ_n.

7.4.5 双重步位移隐式 QR 方法

对于一般的实方阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 前面讨论的算法 7.5 和算法 7.6 都不太好. 首先, 这两种方法都是显式方法, 即每次迭代都需要明显地作矩阵的 QR 分解, 并用所得到的正交矩阵作相似变换 (实际上是作三角矩阵与正交矩阵的乘积), 计算量和存储量都很大. 其次, 基本 QR 算法 7.5 若收敛则是线性的, 而原点位移 QR 算法 7.6 只有当特征值是实数时才有可能改善收敛性. 这是可以理解的, 因为这两种算法都是在实数域上进行运算, 因此当实矩阵 $\boldsymbol{A}$ 有复特征值时, 即使收敛也是很缓慢的.

理论分析和实际计算的经验表明: QR 迭代产生的矩阵序列右下角最先显露 $\boldsymbol{A}$ 的特征值. 在原点位移 QR 方法中正是利用这一特点来选取位移参数 s_k 的. 如果显露的是 $\boldsymbol{A}$ 的实特征值, 即 $a_{nn}^{(k)}$ 是 $\boldsymbol{A}$ 的较好的近似特征值时, 就可以简单地选取 $s_k = a_{nn}^{(k)}$. 然而, 如果显露的是 $\boldsymbol{A}$ 的复共轭特征值时, 即式 (7.56) 中的矩阵 $\boldsymbol{J}_k$ 的特征值是一对互相共轭的复数 μ_1 和 μ_2, 且与 $\boldsymbol{A}$ 的特征值比较接近时, 就应该选择 $\boldsymbol{J}_k$ 某一特征值 μ_i 作为位移参数. 但这样一来, 就引进了复运算, 而这是所不希望的.

实数运算的优点是, 计算简单, 工作量小. 为了用实数运算来求一般实方阵的全部特征值, 特别是复特征值, 可以引进双重步位移隐式 QR 方法. 它的基本思想是: 首先把原点位移推广到复数域 (当然包括实数), 并且每次迭代作两次原点位移的 QR 变换, 因此称为双重步 QR 方法, 这时的运算都可以是复数; 然后把这两次 QR 变换合在一起, 转化成实数运算, 并且构成隐式方法, 即不明显地进行 QR 变换. 它的优点是, 迭代过程都是实数运算, 原点位移加速了收敛性, 隐式方法可减少计算工作量并节省存储空间.

1. 双重步位移隐式 QR 变换

双重步位移隐式 QR 方法的核心是双重步位移隐式 QR 变换. 设矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 是不可约的上 Hessenberg 矩阵, 它的右下角的 2 阶子矩阵

$$\begin{bmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{n,n} \end{bmatrix}$$

的特征值为 μ_1 和 μ_2, 它们可能都是实数, 也可能是一对共轭复数. 记

$$s = a_{n-1,n-1} + a_{n,n}, \quad t = a_{n-1,n-1}a_{n,n} - a_{n,n-1}a_{n-1,n},$$

s 和 t 都是实数, 则有

$$\mu_1 + \mu_2 = s, \quad \mu_1\mu_2 = t. \tag{7.58}$$

现在取 μ_1 和 μ_2 作为位移量, 作两次原点位移的 QR 变换:

$$\begin{cases} \boldsymbol{A}-\mu_1\boldsymbol{I}=\boldsymbol{Q}_1\boldsymbol{R}_1,\ \boldsymbol{B}=\boldsymbol{Q}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_1=\boldsymbol{R}_1\boldsymbol{Q}_1+\mu_1\boldsymbol{I}, \\ \boldsymbol{B}-\mu_2\boldsymbol{I}=\boldsymbol{Q}_2\boldsymbol{R}_2,\ \boldsymbol{C}=\boldsymbol{Q}_2^{\mathrm{H}}\boldsymbol{B}\boldsymbol{Q}_2=\boldsymbol{R}_2\boldsymbol{Q}_2+\mu_2\boldsymbol{I}, \end{cases} \tag{7.59}$$

式中: $\boldsymbol{Q}_1,\boldsymbol{Q}_2$ 为酉矩阵, 也是上 Hessenberg 矩阵; $\boldsymbol{B}$ 和 $\boldsymbol{C}$ 都是上 Hessenberg 矩阵; $\boldsymbol{R}_1$ 和 $\boldsymbol{R}_2$ 都是上三角矩阵. 双重步位移的 QR 变换 (7.59) 具有如下性质: 若记

$$\boldsymbol{H}=(\boldsymbol{A}-\mu_2\boldsymbol{I})(\boldsymbol{A}-\mu_1\boldsymbol{I}),\ \ \boldsymbol{Q}=\boldsymbol{Q}_1\boldsymbol{Q}_2,\ \ \boldsymbol{R}=\boldsymbol{R}_2\boldsymbol{R}_1, \tag{7.60}$$

式中: $\boldsymbol{Q}$ 为酉矩阵; $\boldsymbol{R}$ 为上三角矩阵. 则有

$$\boldsymbol{H}=\boldsymbol{Q}\boldsymbol{R},\ \ \boldsymbol{C}=\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}. \tag{7.61}$$

事实上, 有

$$\begin{aligned} \boldsymbol{H}&=(\boldsymbol{A}-\mu_2\boldsymbol{I})(\boldsymbol{A}-\mu_1\boldsymbol{I})=(\boldsymbol{A}-\mu_2\boldsymbol{I})\boldsymbol{Q}_1\boldsymbol{R}_1 \\ &=\boldsymbol{Q}_1(\boldsymbol{Q}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_1-\mu_2\boldsymbol{I})\boldsymbol{R}_1=\boldsymbol{Q}_1(\boldsymbol{B}-\mu_2\boldsymbol{I})\boldsymbol{R}_1 \\ &=\boldsymbol{Q}_1(\boldsymbol{Q}_2\boldsymbol{R}_2)\boldsymbol{R}_1=\boldsymbol{Q}\boldsymbol{R}, \\ \boldsymbol{C}&=\boldsymbol{Q}_2^{\mathrm{H}}\boldsymbol{B}\boldsymbol{Q}_2=\boldsymbol{Q}_2^{\mathrm{H}}(\boldsymbol{Q}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_1)\boldsymbol{Q}_2=\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}. \end{aligned}$$

进一步, 注意到

$$\begin{aligned} \boldsymbol{H}&=\boldsymbol{A}^2-(\mu_1+\mu_2)\boldsymbol{A}+\mu_1\mu_2\boldsymbol{I}=\boldsymbol{A}^2-s\boldsymbol{A}+t\boldsymbol{I} \\ &=\begin{bmatrix} h_{11} & \times & \times & \times & \cdots & \times \\ h_{21} & \times & \times & \times & \cdots & \times \\ h_{31} & \times & \times & \times & \cdots & \times \\ & \times & \times & \times & \cdots & \times \\ & & \ddots & \ddots & \ddots & \vdots \\ & & & \times & \times & \times \end{bmatrix} \end{aligned} \tag{7.62}$$

是实矩阵, 其中

$$\begin{cases} h_{11}=a_{11}^2+a_{12}a_{21}-sa_{11}+t, \\ h_{21}=a_{21}(a_{11}+a_{22}-s), \\ h_{31}=a_{21}a_{32}. \end{cases} \tag{7.63}$$

所以式 (7.61) 的第 1 式 $\boldsymbol{H}=\boldsymbol{Q}\boldsymbol{R}$ 是实的 QR 分解, 即其中的酉矩阵 $\boldsymbol{Q}$ 必定是正交矩阵, $\boldsymbol{R}$ 必是实上三角矩阵. 从而式 (7.61) 的第 2 式为正交相似变换 $\boldsymbol{C}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$.

综上所述, 为了确保计算得到的 $\boldsymbol{C}$ 仍为实矩阵, 根据式 (7.61), 自然考虑按如下的步骤来计算 $\boldsymbol{C}$:

(1) 计算 $\boldsymbol{H}=\boldsymbol{A}^2-s\boldsymbol{A}+t\boldsymbol{I}$.

(2) 计算 $\boldsymbol{H}$ 的 QR 分解:$\boldsymbol{H}=\boldsymbol{Q}\boldsymbol{R}$.

(3) 计算 $\boldsymbol{C}$ 的正交相似变换: $\boldsymbol{C}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$.

然而, 如此计算的第 1 步形成 $\boldsymbol{H}$ 的运算量就为 $O(n^3)$, 这使得前面为减少每次迭代所需运算量所做的努力付之东流. 幸运的是, 定理 7.12 表明, 不论采取什么方法计算正交矩阵 $\widetilde{\boldsymbol{Q}}$ 使得 $\widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}}=\widetilde{\boldsymbol{C}}$ 成为上 Hessenberg 矩阵, 只要保证 $\widetilde{\boldsymbol{Q}}$ 的第 1 列与 $\boldsymbol{Q}$ 的第 1 列一样, 则 $\widetilde{\boldsymbol{C}}$ 就与 $\boldsymbol{C}$ 在本质上是一样的 (所有元素的绝对值相等). 因此, 可以有很大的自由度去寻求更有效的方法来实现 $\boldsymbol{A}$ 到 $\boldsymbol{C}$ 的变换.

首先, 从式 (7.61) 的第 1 式 $\boldsymbol{H}=\boldsymbol{Q}\boldsymbol{R}$ 知, $\boldsymbol{Q}$ 的第 1 列与 $\boldsymbol{H}$ 的第 1 列共线, 即 $\boldsymbol{Q}\boldsymbol{e}_1$ 由 $\boldsymbol{H}\boldsymbol{e}_1$ 单位化得到. 而由式 (7.62) 容易算出

$$\boldsymbol{H}\boldsymbol{e}_1=(h_{11},h_{21},h_{31},0,\cdots,0)^{\mathrm{T}},$$

式中: h_{11},h_{21},h_{31} 由式 (7.63) 得出.

其次, 如果 Householder 变换 $\boldsymbol{P}_0$ 将 $\boldsymbol{H}\boldsymbol{e}_1$ 变为 $\alpha\boldsymbol{e}_1$, 即 $\boldsymbol{P}_0(\boldsymbol{H}\boldsymbol{e}_1)=\alpha\boldsymbol{e}_1$, 其中 $\alpha\in\mathbb{R}$, 则易知, $\boldsymbol{P}_0$ 的第 1 列就与 $\boldsymbol{H}\boldsymbol{e}_1$ 共线, 从而 $\boldsymbol{P}_0\boldsymbol{e}_1=\boldsymbol{Q}\boldsymbol{e}_1$. 而由 Householder 变换的性质, $\boldsymbol{P}_0$ 可按如下方式确定:

$$\boldsymbol{P}_0=\mathrm{diag}(\widetilde{\boldsymbol{P}}_0,\boldsymbol{I}_{n-3}),$$

式中:

$$\widetilde{\boldsymbol{P}}_0=\boldsymbol{I}_3-\beta\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}},\quad \beta=2/\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v},$$

$$\boldsymbol{v}=(h_{11}+\alpha\mathrm{sign}(h_{11}),h_{21},h_{31})^{\mathrm{T}},\ \alpha=\sqrt{h_{11}^2+h_{21}^2+h_{31}^2}.$$

现令

$$\boldsymbol{D}_1=\boldsymbol{P}_0\boldsymbol{A}\boldsymbol{P}_0,$$

那么只要找到第 1 列为 $\boldsymbol{e}_1$ 的正交矩阵 $\widetilde{\boldsymbol{Q}}$ 使 $\widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{D}_1\widetilde{\boldsymbol{Q}}=\widetilde{\boldsymbol{H}}$ 为上 Hessenberg 矩阵, 那么 $\widetilde{\boldsymbol{H}}$ 就是希望得到的矩阵 $\boldsymbol{C}$. 这只需确定 $n-1$ 个 Householder 变换 $\boldsymbol{P}_1,\cdots,\boldsymbol{P}_{n-1}$, 使

$$(\boldsymbol{P}_{n-1}\cdots\boldsymbol{P}_1)\boldsymbol{D}_1(\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-1})=\widetilde{\boldsymbol{H}}$$

为上 Hessenberg 矩阵, 即有 $\widetilde{\boldsymbol{Q}}=\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-1}$ 的第 1 列为 $\boldsymbol{e}_1$. 而且由于 $\boldsymbol{D}_1$ 所具有的特殊性质, 实现这一约化过程所需的运算量仅为 $O(n^2)$.

事实上, 由于用 $\boldsymbol{P}_0$ 将 $\boldsymbol{A}$ 相似变换为 $\boldsymbol{D}_1$ 只改变了 $\boldsymbol{A}$ 的前三行和前三列, 故 $\boldsymbol{D}_1$ 具有如下形状, 即

$$\boldsymbol{D}_1=\boldsymbol{P}_0\boldsymbol{A}\boldsymbol{P}_0=\begin{bmatrix}\times&\times&\times&\times&\cdots&\times&\times\\ \times&\times&\times&\times&\cdots&\times&\times\\ \oplus&\times&\times&\times&\cdots&\times&\times\\ \oplus&\oplus&\times&\times&\cdots&\times&\times\\ &&&\times&\cdots&\vdots&\vdots\\ &&&&\ddots&\vdots&\vdots\\ &&&&&\times&\times\end{bmatrix},$$

仅比上 Hessenberg 形多 3 个非零元“$\oplus$”. 由 $\boldsymbol{D}_1$ 的这种特殊性, 易知用来约化 $\boldsymbol{D}_1$ 为上 Hessenberg 形的第一个 Householder 变换 $\boldsymbol{P}_1$ 具有如下形状, 即

$$\boldsymbol{P}_1 = \mathrm{diag}(1, \widetilde{\boldsymbol{P}}_1, \boldsymbol{I}_{n-4}),$$

式中: $\widetilde{\boldsymbol{P}}_1$ 为 3 阶 Householder 变换, 而且 $\boldsymbol{P}_1\boldsymbol{D}_1\boldsymbol{P}_1$ 具有如下形状, 即

$$\boldsymbol{D}_2 = \boldsymbol{P}_1\boldsymbol{D}_1\boldsymbol{P}_1 = \boldsymbol{P}_1\boldsymbol{P}_0\boldsymbol{A}\boldsymbol{P}_0\boldsymbol{P}_1 = \begin{bmatrix} \times & \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \times & \cdots & \times & \times \\ 0 & \times & \times & \times & \cdots & \times & \times \\ 0 & \oplus & \times & \times & \cdots & \times & \times \\ & \oplus & \oplus & \times & \cdots & \vdots & \vdots \\ & & & & \ddots & \vdots & \vdots \\ & & & & & \times & \times \end{bmatrix},$$

如此递推地进行下去, 不难发现, 第 k 次约化所用的 Householder 变换 $\boldsymbol{P}_k$ 具有如下形状, 即

$$\boldsymbol{P}_k = \mathrm{diag}(\boldsymbol{I}_k, \widetilde{\boldsymbol{P}}_k, \boldsymbol{I}_{n-k-3}),$$

式中: $\widetilde{\boldsymbol{P}}_k$ 为 3 阶 Householder 变换, $k = 2, \cdots, n-3$, 而且 $\boldsymbol{D}_{n-2} := \boldsymbol{P}_{n-3}\boldsymbol{D}_{n-2}\boldsymbol{P}_{n-3}$ 具有如下形状

$$\begin{aligned} \boldsymbol{D}_{n-2} &= \boldsymbol{P}_{n-3}\boldsymbol{D}_{n-3}\boldsymbol{P}_{n-3} \\ &= \boldsymbol{P}_{n-3}\boldsymbol{P}_{n-4}\cdots\boldsymbol{P}_1\boldsymbol{D}_1\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-4}\boldsymbol{P}_{n-3} \\ &= \begin{bmatrix} \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \cdots & \times & \times \\ & \times & \times & \cdots & \times & \times \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & \times & \times & \times \\ & & & \oplus & \times & \times \end{bmatrix}. \end{aligned}$$

因此, 最后一次约化所用的 Householder 变换 $\boldsymbol{P}_{n-2}$ 具有如下形状, 即

$$\boldsymbol{P}_{n-1} = \mathrm{diag}(\boldsymbol{I}_{n-2}, \widetilde{\boldsymbol{P}}_{n-2}),$$

式中: $\widetilde{\boldsymbol{P}}_{n-2}$ 为 2 阶 Householder 变换. 而且 $\boldsymbol{D}_{n-1} := \boldsymbol{P}_{n-2}\boldsymbol{D}_{n-2}\boldsymbol{P}_{n-2}$ 具有如下形状, 即

$$\begin{aligned} \boldsymbol{D}_{n-1} &= \boldsymbol{P}_{n-2}\boldsymbol{D}_{n-2}\boldsymbol{P}_{n-2} \\ &= \boldsymbol{P}_{n-2}\cdots\boldsymbol{P}_1\boldsymbol{P}_0\boldsymbol{A}\boldsymbol{P}_0\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-2} \end{aligned}$$

$$= \begin{bmatrix} \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \cdots & \times & \times \\ & \times & \times & \cdots & \times & \times \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & \times & \times & \times \\ & & & & \times & \times \end{bmatrix} := \boldsymbol{D}. \tag{7.64}$$

这样, 就找到了一种由 $\boldsymbol{A}$ 到 $\boldsymbol{C}$ 的变换方法, 它既避免了复运算的出现, 又减少了运算量. 当然, 这一变换过程对 $\boldsymbol{J}_k$ 的两个特征值都是实数的情形也是可行的. 因此, 不必在选取位移参数时区别显露的是实特征值还是复共轭特征值的情形, 而只需取作 $\boldsymbol{J}_k$ 的两个特征值即可.

定理 7.15 由式 (7.64) 得到的矩阵 $\boldsymbol{D}$"基本收敛"于变换 (7.61) 中的矩阵 $\boldsymbol{C}$.

证明 若记正交矩阵

$$\boldsymbol{U} = \boldsymbol{P}_0\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-2},$$

则式 (7.64) 即为 $\boldsymbol{D} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{U}$. 易知, $\boldsymbol{U}$ 的第 1 列与 $\boldsymbol{P}_0$ 的第 1 列相同, 即

$$\boldsymbol{U}\boldsymbol{e}_1 = \boldsymbol{P}_0\boldsymbol{e}_1.$$

另外, 用 Householder 变换对矩阵 $\boldsymbol{H}$ 作 QR 分解的过程是

$$\widetilde{\boldsymbol{P}}_{n-1}\cdots\widetilde{\boldsymbol{P}}_2\widetilde{\boldsymbol{P}}_1\boldsymbol{H} = \boldsymbol{R},$$

式中: $\widetilde{\boldsymbol{P}}_1 = \boldsymbol{P}_0$. 对于 $i = 2, \cdots, n-2$, 有

$$\widetilde{\boldsymbol{P}}_i = \begin{bmatrix} \boldsymbol{I}_{i-1} & & \\ & \widetilde{\boldsymbol{V}}_i & \\ & & \boldsymbol{I}_{n-i-2} \end{bmatrix},$$

式中: $\widetilde{\boldsymbol{V}}_i \in \mathbb{R}^{3\times 3}$ 为 Householder 矩阵.

当 $i = n-1$ 时, 有

$$\widetilde{\boldsymbol{P}}_{n-1} = \begin{bmatrix} \boldsymbol{I}_{n-2} & \\ & \widetilde{\boldsymbol{V}}_{n-1} \end{bmatrix},$$

式中: $\widetilde{\boldsymbol{V}}_{n-1} \in \mathbb{R}^{2\times 2}$ 为 Householder 矩阵. 若记正交矩阵

$$\boldsymbol{Q} = \widetilde{\boldsymbol{P}}_1\widetilde{\boldsymbol{P}}_2\cdots\widetilde{\boldsymbol{P}}_{n-1} = \boldsymbol{P}_0\widetilde{\boldsymbol{P}}_2\cdots\widetilde{\boldsymbol{P}}_{n-1},$$

则由 $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}$. 注意到 $\boldsymbol{Q}$ 的第 1 列

$$\boldsymbol{Q}\boldsymbol{e}_1 = \boldsymbol{P}_0\boldsymbol{e}_1 = \boldsymbol{U}\boldsymbol{e}_1,$$

根据定理 7.12, 在化上 Hessenberg 形 $\boldsymbol{C}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$ 的变换中, 矩阵 $\boldsymbol{Q}$ 和 $\boldsymbol{C}$ 本质上由 $\boldsymbol{Q}$ 的第 1 列唯一确定 (相差一个 1 或 -1 因子的意义下). 因此, 定理的结论成立. 证毕. □

综上所述, 可得双重步位移的 QR 变换的迭代过程.

算法 7.7 (双重步位移隐式 QR 变换)

步 1, 输入不可约上 Hessenberg 矩阵 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$.

步 2, $k:=0$; $m:=n-1$; $s:=a_{mm}+a_{nn}$;
$t:=a_{mm}a_{nn}-a_{mn}a_{nm}$;
$x:=a_{11}^2+a_{12}a_{21}-sa_{11}+t$;
$y:=a_{21}(a_{11}+a_{22}-s)$; $z:=a_{21}a_{32}$.

步 3, 若 $k=n-2$, 则转步 5; 否则, 确定 Householder 矩阵 $\widetilde{\boldsymbol{P}}_k\in\mathbb{R}^{3\times 3}$ 使

$$\widetilde{\boldsymbol{P}}_k\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}*\\0\\0\end{bmatrix},$$

置 $\boldsymbol{A}:=\boldsymbol{P}_k\boldsymbol{A}\boldsymbol{P}_k$, 其中 $\boldsymbol{P}_k=\mathrm{diag}(\boldsymbol{I}_k,\widetilde{\boldsymbol{P}}_k,\boldsymbol{I}_{n-k-3})$.

步 4, 更新:

$$x:=a_{k+2,k+1},\quad y:=a_{k+3,k+1},\quad z:=\begin{cases}a_{k+4,k+1}, & k<n-3\\ 0, & k=n-3.\end{cases}$$

置 $k:=k+1$, 转步 3.

步 5 确定 Householder 矩阵 $\widetilde{\boldsymbol{P}}_{n-2}\in\mathbb{R}^{2\times 2}$ 使

$$\widetilde{\boldsymbol{P}}_{n-2}\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}*\\0\end{bmatrix},$$

置 $\boldsymbol{A}:=\boldsymbol{P}_{n-2}\boldsymbol{A}\boldsymbol{P}_{n-2}$, 其中 $\boldsymbol{P}_k=\mathrm{diag}(\boldsymbol{I}_{n-2},\widetilde{\boldsymbol{P}}_{n-2})$. 迭代结束.

算法 7.7 的运算量是 $6n^2$. 如果需要把正交变换累积起来, 还需再增加运算量 $6n^2$.

双重步位移隐式 QR 变换的 MATLAB 程序如下:

```
function A=ddi_qrtran(A,n)
%双重位移隐式QR变换
%输入:A为不可约n阶上Hessenberg矩阵,其最后2x2阶主子阵有特征值a和b
%输出:A=Q^TAQ,这里Q=P1...P_{n-2}是一系列Householder矩阵的乘积
%      且Q^T(A-aI)(A-bI)是上三角形矩阵
I3=eye(3); I2=eye(2); s=A(n-1,n-1)+A(n,n);
```

```
t=A(n-1,n-1)*A(n,n)-A(n-1,n)*A(n,n-1);
x=A(1,1)^2+A(1,2)*A(2,1)-s*A(1,1)+t;
y=A(2,1)*(A(1,1)+A(2,2)-s); z=A(2,1)*A(3,2);
for k=0:n-3
   [v,beta]=mhouse([x,y,z]'); q=max(1,k);
   A(k+1:k+3,q:n)=(I3-beta*v*v')*A(k+1:k+3,q:n);
   r=min(k+4,n);
   A(1:r,k+1:k+3)=A(1:r,k+1:k+3)*(I3-beta*v*v');
   x=A(k+2,k+1); y=A(k+3,k+1);
   if (k<n-3), z=A(k+4,k+1); end
end
[v,beta]=mhouse([x,y]');
A(n-1:n,n-2:n)=(I2-beta*v*v')*A(n-1:n,n-2:n);
A(1:n,n-1:n)=A(1:n,n-1:n)*(I2-beta*v*v');
```

2. 双重步位移隐式 QR 方法

前面的讨论已经解决了用 QR 方法求一个给定实矩阵的实 Schur 分解的几个关键性问题. 然而, 作为一种实用的算法, 还需给出一种有效的判定准则, 来判定迭代过程中所产生的上 Hessenberg 矩阵的次对角元素何时可以忽略不计. 一种简单而实用的准则是: 当

$$|a_{i+1,i}| \leqslant (|a_{ii}| + |a_{i+1,i+1}|)\varepsilon \tag{7.65}$$

时, 就将 $a_{i+1,i}$ 看作是 0.

将算法 7.4 和算法 7.7 与收敛准则 (7.65) 结合起来, 就得到了双重步位移隐式 QR 方法. 这一算法是计算一给定的 n 阶实矩阵 $\boldsymbol{A}$ 实 Schur 分解: $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{T}$, 其中 $\boldsymbol{Q}$ 为正交矩阵, $\boldsymbol{T}$ 为拟上三角矩阵, 即对角块为 1×1 或 2×2 方阵的块上三角矩阵. 由于其中的 QR 变换不明显, 故称为隐式方法.

算法 7.8 (双重步位移隐式 QR 方法)

步 1, 输入矩阵 $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$.

步 2, 上 Hessenberg 化. 用算法 7.4 计算 $\boldsymbol{A}$ 的上 Hessenberg 分解 $\boldsymbol{A} := \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$.

步 3, 若满足收敛性准则, 停算; 否则返回步 2 继续进行双重步位移隐式 QR 迭代.

双重步位移隐式 QR 方法的 MATLAB 程序如下:

```
function [iter,D]=ddiqr_eig(A,tol)
%用双重步位移隐式QR方法求实方阵的全部特征值
%输入:A为n阶上Hessenberg形实方阵,tol为控制精度(默认是1e-5)
%输出:iter为迭代次数,D为A的全部特征值
```

```
if nargin<2, tol=1e-5; end
n=size(A,1);
D=zeros(n,1); i=n; m=n; iter=0;  %初始化
[A]=hessenb(A);  %化矩阵A为Hessenberg矩阵
while (m>0)
    %用双重位移隐式QR方法进行迭代
    if m<=2
        la=eig(A(1:m,1:m));
        D(1:m)=la'; break;
    end
    iter=iter+1;
    A=ddiqr_tran(A,m);  %对上Hessenberg 矩阵作QR分解,并作正交相似变换
    for k=m-1:-1:1  %下面的程序段是判断是否终止
        if abs(A(k+1,k))<tol
            if m-k<=2
                la=eig(A(k+1:m,k+1:m));
                j=i-m+k+1; D(j:i)=la';
                i=j-1; m=k; break;
            end
        end
    end
end
```

例 7.8 用双重步位移隐式 QR 方法求例 7.7 中矩阵 $\boldsymbol{A}$ 的全部特征值.

解 编写 M 文件 ex78.m, 在 MATLAB 命令窗口输入 ex78 得

```
>> ex78
算   法        迭代次数    CPU时间
基本QR方法      622        0.0702
双位移QR方法    8          0.0201
基本QR方法特征值    双位移QR方法特征值
18.4123 + 0.0000i   18.4123 + 0.0000i
11.1805 + 0.0000i   11.1805 + 0.0000i
 1.7099 - 4.2522i    1.7099 - 4.2522i
 1.7099 + 4.2522i    1.7099 + 4.2522i
 4.4983 + 0.0000i   -2.2327 + 0.0000i
-2.2327 + 0.0000i    4.4983 + 0.0000i
-0.2783 + 0.0000i   -0.2783 + 0.0000i
```

7.4.6 特征向量的计算方法

本节讨论在用 (双重步位移隐式) QR 方法求得给定矩阵的特征值之后, 如何求其

对应的特征向量. 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 并假定已用 QR 方法求得 $\boldsymbol{A}$ 的特征值 λ 的一个近似 $\widetilde{\lambda}$. 现在讨论如何求对应于 λ 的特征向量.

目前, 解决这一问题最好的方法是*带原点位移的反幂法* (也称为*反迭代法*, *逆迭代法*), 其基本迭代格式如下:

$$(\boldsymbol{A}-\alpha\boldsymbol{I})\boldsymbol{y}^{(k)}=\boldsymbol{x}^{(k-1)}, \tag{7.66a}$$

$$\boldsymbol{x}^{(k)}=\boldsymbol{y}^{(k)}/\|\boldsymbol{y}^{(k)}\|_2,\quad k=1,2,\cdots, \tag{7.66b}$$

式中: α 为选定的位移参数; $\boldsymbol{x}^{(0)}$ 为给定的初始向量.

从式 (7.66a) 可以看出, 每迭代一次就需要解一个线性方程组, 这要比幂法的运算量大得多. 但由于方程组的系数矩阵不随 k 变化, 故可事先对它进行列主元 LU 分解, 然后每次迭代只需解两个三角形方程组即可. 顺便指出, 式 (7.66b) 只是为了防止迭代"溢出"而做的归一化处理, 在实际计算时可以用 $\|\cdot\|_\infty$ 进行归一化.

现假定 $\boldsymbol{A}$ 是非亏损的, 即存在 $\boldsymbol{X}=[\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\cdots,\boldsymbol{\xi}_n]\in\mathbb{C}^{n\times n}$ 非奇异, 使得

$$\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}=\boldsymbol{\Lambda}\equiv\operatorname{diag}\{\lambda_1,\lambda_2,\cdots,\lambda_n\}. \tag{7.67}$$

而且不失一般性, 还可以假设 $\|\boldsymbol{\xi}_i\|_2=1\,(i=1,2,\cdots,n)$. 现将初始向量 $\boldsymbol{x}^{(0)}$ 按 $\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\cdots,\boldsymbol{\xi}_n$ 展开

$$\boldsymbol{x}^{(0)}=\sum_{i=1}^{n}\beta_i\boldsymbol{\xi}_i. \tag{7.68}$$

再假定 α 与 $\boldsymbol{A}$ 的特征值 λ_s 最靠近, 并有

$$0<|\alpha-\lambda_s|<|\alpha-\lambda_i|,\quad i\neq s, \tag{7.69a}$$

$$\beta_s\neq 0. \tag{7.69b}$$

由式 (7.66a)、式 (7.67) 和式 (7.68), 得

$$\begin{aligned}\boldsymbol{y}^{(k)}&=\theta_k(\boldsymbol{A}-\alpha\boldsymbol{I})^{-k}\boldsymbol{x}^{(0)}\\&=\theta_k\sum_{i=1}^{n}\beta_k(\boldsymbol{A}-\alpha\boldsymbol{I})^{-k}\boldsymbol{\xi}_i=\theta_k\sum_{i=1}^{n}\beta_k(\lambda_i-\alpha)^{-k}\boldsymbol{\xi}_i\\&=\theta_k\beta_s(\lambda_s-\alpha)^{-k}\left[\boldsymbol{\xi}_s+\sum_{i\neq s}^{n}\left(\frac{\lambda_s-\alpha}{\lambda_i-\alpha}\right)^k\frac{\beta_i}{\beta_s}\boldsymbol{\xi}_i\right]\\&:=\theta_k\beta_s(\lambda_s-\alpha)^{-k}(\boldsymbol{\xi}_s+\boldsymbol{u}_k),\end{aligned} \tag{7.70}$$

式中: θ_k 为正数, 而 $\boldsymbol{u}_k\to 0\,(k\to\infty)$, 其收敛速度依赖于

$$\frac{|\lambda_s-\alpha|}{\min\limits_{i\neq s}|\alpha-\lambda_i|}$$

的大小. 将式 (7.70) 代入式 (7.66b), 得

$$\boldsymbol{x}^{(k)}=\frac{\boldsymbol{y}^{(k)}}{\|\boldsymbol{y}^{(k)}\|_2}=\frac{\eta_k}{\|\boldsymbol{\xi}_s+\boldsymbol{u}_k\|_2}(\boldsymbol{\xi}_s+\boldsymbol{u}_k),$$

式中:

$$\eta_k = \frac{\theta_k \beta_s (\lambda_s - \alpha)^{-k}}{|\theta_k \beta_s (\lambda_s - \alpha)^{-k}|} = \frac{\beta_s}{|\beta_s|} \left(\frac{\lambda_s - \alpha}{|\lambda_s - \alpha|} \right)^{-k}$$

为满足 $|\eta_k| = 1$ 的复数. 从而有

$$\begin{aligned} \operatorname{dist}(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}_s) &= \|\mathcal{P}_{\boldsymbol{x}^{(k)}} - \mathcal{P}_{\boldsymbol{\xi}_s}\|_2 = \|\boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^{\mathrm{H}} - \boldsymbol{\xi}_s \boldsymbol{\xi}_s^{\mathrm{H}}\|_2 \\ &= \left\| \frac{(\boldsymbol{\xi}_s + \boldsymbol{u}_k)(\boldsymbol{\xi}_s + \boldsymbol{u}_k)^{\mathrm{H}}}{(\boldsymbol{\xi}_s + \boldsymbol{u}_k)^{\mathrm{H}}(\boldsymbol{\xi}_s + \boldsymbol{u}_k)} - \boldsymbol{\xi}_s \boldsymbol{\xi}_s^{\mathrm{H}} \right\|_2 \\ &\to 0, \quad k \to \infty, \end{aligned}$$

收敛速度依赖于 $|\lambda_s - \alpha| / \min\limits_{i \neq s} |\alpha - \lambda_i|$ 的大小. 换言之, 即 $\boldsymbol{x}^{(k)}$ 将按方向收敛于 $\boldsymbol{A}$ 的特征向量. α 与 λ_s 越靠近, 收敛速度越快.

由此可见, 从收敛速度的角度来考虑, 用式 (7.66) 迭代时, 自然是 α 取得越靠近 $\boldsymbol{A}$ 的某个特征值越好. 但是, 当 α 与 $\boldsymbol{A}$ 的某个特征值很靠近时, $\boldsymbol{A} - \alpha \boldsymbol{I}$ 就很接近于一个奇异矩阵, 每一步迭代就需要解一个非常病态的线性方程组. 幸运的是理论分析以及大量的计算实践表明: $\boldsymbol{A} - \alpha \boldsymbol{I}$ 的病态性并不影响其收敛速度, 而且当 α 很靠近 $\boldsymbol{A}$ 的某个特征值时, 常常只需要一次迭代就可以得到相当好的近似特征向量. 为此, 下面进行简要的理论分析.

设 λ 是 $\boldsymbol{A}$ 的特征值, $\boldsymbol{y} \in \mathbb{C}^n$ 满足 $\|\boldsymbol{y}\|_2 = 1$. 定义

$$\boldsymbol{r} = (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{y} \tag{7.71}$$

为向量 $\boldsymbol{y}$ 的残差向量. 由式 (7.71), 得

$$(\boldsymbol{A} - \boldsymbol{r}\boldsymbol{y}^{\mathrm{H}})\boldsymbol{y} = \lambda \boldsymbol{y},$$

即 $\boldsymbol{y}$ 是矩阵 $\boldsymbol{A} - \boldsymbol{r}\boldsymbol{y}^{\mathrm{H}}$ 对应于 λ 的特征向量. 如果 $\|\boldsymbol{r}\|_2$ 很小, 则 $\|\boldsymbol{r}\boldsymbol{y}^{\mathrm{H}}\|_2$ 也会很小. 从而, 当 $\|\boldsymbol{r}\|_2$ 很小时, $\boldsymbol{y}$ 就是 $\boldsymbol{A}$ 的对应于 λ 的一个很好的近似特征向量. 即可用 $\boldsymbol{y}$ 的残差向量大小来衡量 $\boldsymbol{y}$ 可否作为对应于 λ 的近似特征向量.

进一步, 假定

$$|\alpha - \lambda| = \min_{\widetilde{\lambda} \in \lambda(\boldsymbol{A})} |\alpha - \widetilde{\lambda}| \leqslant \varepsilon_1, \tag{7.72}$$

式中: ε_1 为很小的正数, 通常 $\varepsilon_1 = O(\epsilon)$; ϵ 为机器精度. 再假定对给定的初始向量 $\boldsymbol{x}^{(0)}$, 用列主元 Gauss 消去法求解方程组 (7.66a) 得到向量 $\boldsymbol{y}^{(1)}$ 的计算值 $\widetilde{\boldsymbol{y}}^{(1)}$. 则由 Gauss 消去法的舍入误差分析结果可知

$$(\boldsymbol{A} - \alpha \boldsymbol{I} + \boldsymbol{E})\widetilde{\boldsymbol{y}}^{(1)} = \boldsymbol{x}^{(0)}, \tag{7.73}$$

式中: $\|\boldsymbol{E}\|_2 \leqslant \varepsilon_2$ (通常 $\varepsilon_2 = O(\epsilon)$). 这样, 由式 (7.66b) 计算得

$$\boldsymbol{x}^{(1)} = \widetilde{\boldsymbol{y}}^{(1)} / \|\widetilde{\boldsymbol{y}}^{(1)}\|_2, \tag{7.74}$$

这里忽略了计算 $\boldsymbol{x}^{(1)}$ 所产生的误差.

利用式 (7.73), 可得向量 $\boldsymbol{x}^{(1)}$ 的残差向量为

$$\boldsymbol{r}=(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{x}^{(1)}=(\alpha-\lambda)\boldsymbol{x}^{(1)}-\boldsymbol{E}\boldsymbol{x}^{(1)}+\frac{\boldsymbol{x}^{(0)}}{\|\widetilde{\boldsymbol{y}}^{(1)}\|_2}.$$

于是有

$$\|\boldsymbol{r}\|_2\leqslant\varepsilon_1+\varepsilon_2+\|\widetilde{\boldsymbol{y}}^{(1)}\|_2^{-1}, \tag{7.75}$$

这里假定 $\|\boldsymbol{x}^{(0)}\|_2=1$.

由此可见, 如果计算得到的 $\widetilde{\boldsymbol{y}}^{(1)}$ 具有很大的范数, 则由式 (7.66) 迭代一次所得的向量就有很小的残差向量, 从而在特征值问题不是十分病态的条件下, 就得到了很好的近似特征向量. 因此, 只需说明在式 (7.72) 成立的前提下确有 $\widetilde{\boldsymbol{y}}^{(1)}$ 的范数很大.

事实上, 设 $\boldsymbol{A}-\alpha\boldsymbol{I}+\boldsymbol{E}$ 的奇异值分解为

$$\boldsymbol{A}-\alpha\boldsymbol{I}+\boldsymbol{E}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}, \tag{7.76}$$

式中: $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n]$, $\boldsymbol{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n]$; $\boldsymbol{\Sigma}=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_n)$; $\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_n>0$. 由 $\lambda-\alpha$ 是矩阵 $\boldsymbol{A}-\alpha\boldsymbol{I}$ 的特征值, 得

$$\sigma_n(\boldsymbol{A}-\alpha\boldsymbol{I})\leqslant|\lambda-\alpha|\leqslant\varepsilon_1,$$

式中: $\sigma_n(\boldsymbol{A}-\alpha\boldsymbol{I})$ 为 $\boldsymbol{A}-\alpha\boldsymbol{I}$ 的最小奇异值. 则由特征值的分离理论, 有

$$\sigma_n\leqslant\sigma_n(\boldsymbol{A}-\alpha\boldsymbol{I})+\|\boldsymbol{E}\|_2\leqslant\varepsilon_1+\varepsilon_2. \tag{7.77}$$

将 $\boldsymbol{x}^{(0)}$ 按 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n$ 展开, 有

$$\boldsymbol{x}^{(0)}=\sum_{i=1}^{n}\beta_i\boldsymbol{u}_i,\quad\sum_{i=1}^{n}|\beta_i|^2=\|\boldsymbol{x}^{(0)}\|_2^2=1.$$

从而有

$$\begin{aligned}\widetilde{\boldsymbol{y}}^{(1)}&=(\boldsymbol{A}-\alpha\boldsymbol{I}+\boldsymbol{E})^{-1}\boldsymbol{x}^{(0)}\\&=\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\mathrm{H}}\Big(\sum_{i=1}^{n}\beta_i\boldsymbol{u}_i\Big)=\sum_{i=1}^{n}\frac{\beta_i}{\sigma_i}\boldsymbol{v}_i.\end{aligned} \tag{7.78}$$

于是有

$$\|\widetilde{\boldsymbol{y}}^{(1)}\|_2=\left(\sum_{i=1}^{n}\left|\frac{\beta_i}{\sigma_i}\right|^2\right)^{\frac{1}{2}}\geqslant\frac{|\beta_n|}{\sigma_n}\geqslant\frac{|\beta_n|}{\varepsilon_1+\varepsilon_2}.$$

这样一来, 只要 $|\beta_n|$ 不是很小 (即 $\boldsymbol{x}^{(0)}$ 在 $\boldsymbol{u}_n$ 方向上不是十分亏损), $\|\widetilde{\boldsymbol{y}}^{(1)}\|_2$ 就会很大. 因此, 通常反幂法只需要迭代一次就足够了.

上述分析表明, 利用反幂法求特征向量时, 位移量 α 取为较精确的近似特征值最好. 此时, 一般只需要迭代一次就可以得到很好的近似特征向量. 因此, 通常总是在用某种方法求得 $\boldsymbol{A}$ 的近似特征值之后, 再利用反幂法求对应的特征向量. 连同 QR 方法一起来使用反幂法的基本步骤如下.

算法 7.9

步 1, 输入矩阵 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$.

步 2, 上 Hessenberg 化. 用算法 7.4 计算 $\boldsymbol{A}$ 的上 Hessenberg 分解 $\boldsymbol{A}:=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$.

步 3, 用双重步位移隐式 QR 方法 (算法 7.7) 求出 $\boldsymbol{A}$ 的特征值, 而不累积正交变换矩阵.

步 4, 对每个计算得到的近似特征值 $\widetilde{\lambda}$, 在式 (7.66) 中取位移参数 $\alpha=\widetilde{\lambda}$ 进行迭代, 求出特征向量 $\boldsymbol{z}$.

步 5, 计算 $\boldsymbol{x}=\boldsymbol{Q}\boldsymbol{z}$ (则 $\boldsymbol{x}$ 就是对应于 $\widetilde{\lambda}$ 的近似特征向量).

算法 7.9 的 MATLAB 程序如下:

```
function [Lam,V,iter,ki]=ddiqr_eigvec(A,tol)
%用双重步位移隐式QR方法求实方阵的全部特征值和相应的特征向量
%输入:A为n阶实方阵,tol为控制精度(默认是1.e-5)
%输出:Lam为A的全部特征值,V为A的全部特征向量,iter为迭代次数
if nargin<2, tol=1e-5; end
n=size(A,1); x=rand(n,1); %x=ones(n,1);
Lam=zeros(n,1); V=zeros(n);
[A,Q]=mhessen(A);  %调用上Hessenberg化程序
%调用双重步位移隐式QR方法求全部特征值
[iter,lambda]=ddiqr_eig(A,tol);
for i=1:n
    [lam,v,k]=mvpower(A,x,lambda(i));  %调用反幂法程序
    V(:,i)=v; ki(i)=k; Lam(i)=lam;
end
V=Q*V;  %V的每一列为特征向量
```

例 7.9 用算法 7.9 求例 7.7 中矩阵 $\boldsymbol{A}$ 的全部特征值和特征向量.

解 编写脚本 M 文件 ex79.m, 然后在 MATLAB 命令窗口执行之, 得计算结果:

```
>> ex79
双位移QR方法结果     eig函数计算结果
18.4123 + 0.0000i  18.4123 + 0.0000i
11.1805 + 0.0000i  11.1805 + 0.0000i
 1.7099 - 4.2522i   4.4983 + 0.0000i
 1.7099 + 4.2522i   1.7099 + 4.2522i
-2.2327 + 0.0000i   1.7099 - 4.2522i
 4.4983 + 0.0000i  -2.2327 + 0.0000i
-0.2783 + 0.0000i  -0.2783 + 0.0000i
iter =
```

```
     8
ki =
     2     2     2     2     2     2     2
err =
   1.7959e-05
```

7.5 Givens–Householder 方法

7.3 节中的 Jacobi 方法通过 Givens 变换构造一个正交相似矩阵序列 $\{\boldsymbol{A}_{k+1}\}(\boldsymbol{A}_{k+1}=\boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}_k\boldsymbol{Q}_k)$, 使得 $\boldsymbol{A}_{k+1}$ 趋于一个对角矩阵, 从而得到 $\boldsymbol{A}$ 的全部特征值 $\lambda_i\,(i=1,2,\cdots,n)$. Givens 指出, 如果不要求 $\boldsymbol{A}_k$ 趋于对角矩阵而是三对角矩阵, 则这个过程可以是有限步的. 而 Householder 建议如果 $\boldsymbol{Q}_k$ 取为形如 $\boldsymbol{H}_k=\boldsymbol{I}-2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ 的 Householder 矩阵, 则可以更有效地实现这个三对角化的过程.

将 $\boldsymbol{A}$ 化为三对角矩阵 $\boldsymbol{T}=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{H}$ 后, 还需要求 $\boldsymbol{T}$ 的特征值. 为此, Givens 根据 $\boldsymbol{T}-\lambda\boldsymbol{I}$ 的顺序主子式构成的 Sturm 序列这一事实提出了计算 $\boldsymbol{T}$ 的特征值的二分法.

为计算 $\boldsymbol{A}$ 的特征向量, 要先计算 $\boldsymbol{T}$ 的特征向量. 一个有效的方法是反迭代法.

这种先通过 Householder 变换化对称矩阵 $\boldsymbol{A}$ 为对称三对角矩阵 $\boldsymbol{T}$, 然后通过求 $\boldsymbol{T}$ 的特征值和特征向量来求得 $\boldsymbol{A}$ 的特征值和特征向量的方法称为 Givens–Householder *方法*. 对称矩阵的三对角化即为上 Hessenberg 化, 这在 7.4.1 节中已经讨论过, 可以执行算法 7.4 得到. 下面只需考虑如何计算对称三对角矩阵 $\boldsymbol{T}$ 的特征值.

7.5.1 求对称三对角矩阵特征值的二分法

考虑实对称三对角矩阵

$$\boldsymbol{T}=\begin{bmatrix}\alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n\end{bmatrix}.$$

假定 $\boldsymbol{T}$ 是不可约的, 即 $\beta_i\neq 0\,(i=2,\cdots,n)$. 否则, 可以将 $\boldsymbol{T}$ 分解成几个阶数更小的不可约三对角矩阵.

令 $p_k(\lambda)$ 表示矩阵 $\boldsymbol{T}-\lambda\boldsymbol{I}$ 的 k 阶顺序主子式, 即

$$p_k(\lambda)=\det(\boldsymbol{T}-\lambda\boldsymbol{I})_k$$

$$= \det \begin{bmatrix} \alpha_1 - \lambda & \beta_2 & & & \\ \beta_2 & \alpha_2 - \lambda & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} - \lambda & \beta_k \\ & & & \beta_k & \alpha_k - \lambda \end{bmatrix}, \quad k = 1, 2, \cdots, n.$$

若令 $p_0(\lambda) \equiv 1$, 则有递推式

$$\begin{aligned} & p_0(\lambda) = 1, \ p_1(\lambda) = \alpha_1 - \lambda, \\ & p_{k+1}(\lambda) = (\alpha_{k+1} - \lambda) p_k(\lambda) - \beta_{k+1}^2 p_{k-1}(\lambda), \quad k = 1, \cdots, n-1. \end{aligned} \tag{7.79}$$

容易证明, 多项式序列 $\{p_k(\lambda)\}_0^n$ 具有如下四个特征.

引理 7.2 *序列 $\{p_k(\lambda)\}_0^n$ 满足*

$$p_k(-\infty) > 0, \quad p_k(+\infty) \begin{cases} > 0, & k \text{ 为偶数}, \\ < 0, & k \text{ 为奇数}. \end{cases} \tag{7.80}$$

证明 首先用归纳法证明

$$p_k(\lambda) = (-1)^k \lambda^k + g_{k-1}(\lambda), \tag{7.81}$$

式中: $g_{k-1}(\lambda)$ 为 λ 的 $k-1$ 次多项式.

事实上, 当 $k = 0$ 时, 显然式 (7.81) 成立. 假设 $k-1$ 时结论成立. 则对 k, 有

$$\begin{aligned} p_k(\lambda) &= (\alpha_k - \lambda) p_{k-1}(\lambda) - \beta_k^2 p_{k-2}(\lambda) \\ &= (\alpha_k - \lambda) \left[(-1)^{k-1} \lambda^{k-1} + g_{k-2}(\lambda) \right] - \beta_k^2 p_{k-2}(\lambda) \\ &= (-1)^k \lambda^k + (-1)^{k-1} \alpha_k \lambda^{k-1} + (\alpha_k - \lambda) g_{k-2}(\lambda) - \beta_k^2 p_{k-2}(\lambda) \\ &= (-1)^k \lambda^k + g_{k-1}(\lambda). \end{aligned}$$

注意到 k 次多项式 $p_k(\lambda)$ 的符号在 $|\lambda|$ 充分大时决定于其首项的符号, 故由上式立即得到引理的结论. 证毕. □

引理 7.3 *$p_0(\lambda)$ 无零点, 相邻两个多项式 $p_k(\lambda)$ 与 $p_{k+1}(\lambda)$ 无公共零点.*

证明 $p_0(\lambda)$ 无零点是显然的. 设 $p_k(\lambda)$ 与 $p_{k+1}(\lambda)$ 有公共零点 $\bar{\lambda}$, 则由递推式 (7.79) 可知, $\bar{\lambda}$ 是 p_{k-1} 的零点 (因 $\beta_{k+1} \neq 0$). 依次类推, 可知 $\bar{\lambda}$ 是 $p_{k-2}(\lambda), \cdots, p_1(\lambda), p_0(\lambda)$ 的零点, 与 $p_0(\lambda) = 1$ 矛盾. 证毕. □

引理 7.4 *设 $\bar{\lambda}$ 是 $p_k(\lambda) (0 < k < n)$ 的零点, 则 $p_{k-1}(\bar{\lambda}) p_{k+1}(\bar{\lambda}) < 0$.*

证明 因 $\bar{\lambda}$ 是 p_k 的零点, 由引理 7.3 可知, $p_{k-1}(\bar{\lambda}) \neq 0$, $p_{k+1}(\bar{\lambda}) \neq 0$. 由式 (7.79) 可知

$$p_{k+1}(\bar{\lambda}) = -\beta_{k+1}^2 p_{k-1}(\bar{\lambda}),$$

故 $p_{k-1}(\bar{\lambda}) p_{k+1}(\bar{\lambda}) = -\beta_{k+1}^2 p_{k-1}(\bar{\lambda})^2 < 0$. 证毕. □

引理 7.5 $p_k(\lambda)\,(0<k<n)$ 的零点全是单重的, 并且 $p_k(\lambda)$ 的零点把 $p_{k+1}(\lambda)$ 的零点严格地隔离开来.

证明 对 k 使用归纳法. 事实上, 当 $k=1$ 时, $p_1(\lambda)=\alpha_1-\lambda$, 其零点为 α_1. 另外, $p_2(\alpha_1)=-\beta_2^2<0$, 且由式 (7.80) 可知, $p_2(-\infty)>0$, $p_2(+\infty)>0$. 故 $p_2(\lambda)$ 的根必定在区间 $(-\infty,\alpha_1)$ 与 $(\alpha_1,+\infty)$ 内, 即当 $k=1$ 时, 引理结论成立.

假设当 $k=i-1$ 时结论成立, 即 $p_{i-1}(\lambda)$ 和 $p_i(\lambda)$ 的根全是单根且 $p_{i-1}(\lambda)$ 的根把 $p_i(\lambda)$ 的根严格隔开. 记 $\lambda_s^{(l)}$ 为 $p_l(\lambda)\,(l=i-1,i)$ 的第 s 个根. 则有

$$\lambda_1^{(i)}<\lambda_1^{(i-1)}<\lambda_2^{(i)}<\lambda_2^{(i-1)}<\cdots<\lambda_{i-1}^{(i)}<\lambda_{i-1}^{(i-1)}<\lambda_i^{(i)}.$$

由于

$$p_{i-1}(-\infty)>0,\ \ p_{i-1}(\lambda_1^{(i-1)})=0,\ \ p_{i-1}(\lambda_2^{(i-1)})=0,\ \cdots,\ p_{i-1}(\lambda_{i-1}^{(i-1)})=0,$$

所以

$$p_{i-1}(\lambda_1^{(i)})>0,\ \ p_{i-1}(\lambda_2^{(i)})<0,\ \ p_{i-1}(\lambda_3^{(i)})>0,\cdots,$$

即 $p_{i-1}(\lambda_s^{(i)})$ 的符号为 $(-1)^{s+1}$. 又因

$$p_{i+1}(\lambda_s^{(i)})=-\beta_{i+1}^2p_{i-1}(\lambda_s^{(i)}),$$

故

$$p_{i+1}(-\infty)>0,\ \ p_{i+1}(\lambda_1^{(i)})<0,\ \ p_{i+1}(\lambda_2^{(i)})>0,\ \ p_{i+1}(\lambda_3^{(i)})<0,\cdots.$$

于是由零点存在定理, $p_{i+1}(\lambda)$ 在区间

$$(-\infty,\lambda_1^{(i)}),\ (\lambda_1^{(i)},\lambda_2^{(i)}),\cdots,(\lambda_{i-1}^{(i)},\lambda_i^{(i)}),\ (\lambda_i^{(i)},+\infty)$$

内都有根. 这样的区间共有 $i+1$ 个, 而 $p_{i+1}(\lambda)$ 恰有 $i+1$ 个根, 故在每个区间内有且仅有一个根, 即当 $k=i$ 时结论成立. 由归纳法原理, 引理得证. □

定义 7.5 对于一个多项式序列 $\{p_k(\lambda)\}_0^n$, 如果它具有引理 7.3 至引理 7.5 的性质, 则称这个多项式序列为 Sturm 序列.

对于 Sturm 序列, 任意取定 α, 则

$$p_0(\alpha),\ p_1(\alpha),\ \cdots,p_k(\alpha),\ \ k\leqslant n \tag{7.82}$$

是一个有限序列. 用 $s_k(\alpha)$ 表示式 (7.82) 中每相邻两数符号一致的数目, 并称其为该序列的同号数. 若序列中某一项 $p_k(\alpha)=0$, 则约定 $p_k(\alpha)$ 的符号与 $p_{k-1}(\alpha)$ 的符号相同. 根据引理 7.3, 此时必有 $p_{k-1}(\alpha)\neq 0$. 例如, 数列 $\{2,\,4,\,8,\,16,\,-10,\,-12,\,14\}$ 的同号数 $s_6=4$, 而数列 $\{-1,\,-3,\,-5,0,\,6,\,7\}$ 的同号数 $s_5=4$.

下面的定理是求对称三对角矩阵特征值二分法的理论基础.

定理 7.16 设 $\boldsymbol{T}$ 是不可约对称三对角矩阵, α 是任意实数, 则 $s_k(\alpha)$ 等于 $p_k(\lambda)=0$ 在区间 $[\alpha,+\infty)$ 内根的个数.

证明 用归纳法. 当 $k=0$ 时, 由于 $p_0(\alpha)=1$, 故结论显然成立. 当 $k=1$ 时, $p_1(\lambda)=\alpha_1-\lambda$. 若 $p_1(\alpha)<0$, 则 $s_1(\alpha)=0$. 注意到 $p_1(-\infty)>0$, 可知 $p_1(\lambda)=0$ 的根必在 $(-\infty,\alpha)$ 内, 因此在 $[\alpha,+\infty)$ 无根, 即根的个数为 0 个. 若 $p_1(\alpha)\geqslant 0$, 则 $s_1(\alpha)=1$. 注意到 $p_1(+\infty)<0$, 可知此时 $p_1(\lambda)=0$ 的唯一根必在 $[\alpha,+\infty)$ 内. 故当 $k=1$ 时, 结论为真.

现假定 $k\leqslant l$ 时结论成立. 记 $s_l(\alpha)=r$, $p_l(\lambda)$ 的零点为

$$\xi_l<\xi_{l-1}<\cdots<\xi_1.$$

由归纳法假设, 有

$$\xi_l<\xi_{l-1}<\cdots<\xi_{r+1}<\alpha\leqslant\xi_r<\cdots<\xi_1. \tag{7.83}$$

设 $p_{l+1}(\lambda)$ 的零点为

$$\mu_{l+1}<\mu_l<\cdots<\mu_{r+1}<\mu_r<\cdots<\mu_1,$$

则显然有

$$p_l(\alpha)=\prod_{i=1}^{l}(\xi_i-\alpha),\quad p_{l+1}(\alpha)=\prod_{i=1}^{l+1}(\mu_i-\alpha). \tag{7.84}$$

且据引理 7.5 有 $p_l(\lambda)$ 与 $p_{l+1}(\lambda)$ 的零点相互隔离:

$$\mu_{l+1}<\xi_l<\mu_l<\cdots<\xi_{r+1}<\mu_{r+1}<\xi_r<\mu_r<\cdots<\xi_1<\mu_1, \tag{7.85}$$

由式 (7.83) 和式 (7.85) 可知, $\xi_{r+1},\mu_{r+1},\alpha,\xi_r$ 之间的相互位置只能出现下面四种情况:

(1) $\xi_{r+1}<\alpha<\mu_{r+1}$. 此时 $p_{l+1}(\lambda)$ 在 $[\alpha,+\infty)$ 上恰有 $r+1$ 个零点. 而由式 (7.84) 可知 $p_l(\alpha)$ 的符号为 $(-1)^{l-r}$ 与 $p_{l+1}(\alpha)$ 的符号 $(-1)^{l+1-(r+1)}$ 相同, 故 $s_{l+1}(\alpha)=s_l(\alpha)+1=r+1$.

(2) $\mu_{r+1}<\alpha<\xi_r$. 此时 $p_{l+1}(\lambda)$ 在 $[\alpha,+\infty)$ 上恰有 r 个零点. 而由式 (7.84) 可知 $p_l(\alpha)$ 的符号 $(-1)^{l-r}$ 与 $p_{l+1}(\alpha)$ 的符号 $(-1)^{(l+1)-r}$ 相反, 于是 $s_{l+1}(\alpha)=s_l(\alpha)=r$.

(3) $\mu_{r+1}=\alpha<\xi_r$. 此时 $p_{l+1}(\lambda)$ 在 $[\alpha,+\infty)$ 上恰有 $r+1$ 个零点. 因为此时 $p_{l+1}(\alpha)=0$, 按约定 $p_{l+1}(\alpha)$ 与 $p_l(\alpha)$ 符号相同, 故仍有 $s_{l+1}(\alpha)=s_l(\alpha)+1=r+1$.

(4) $\mu_{r+1}<\alpha=\xi_r$. 此时 $p_{l+1}(\lambda)$ 在 $[\alpha,+\infty)$ 上恰有 r 个零点. 注意到 $p_l(\alpha)=0$, 按约定, 它与 $p_{l-1}(\alpha)$ 符号相同. 而由引理 7.4 知, $p_{l+1}(\alpha)$ 与 $p_{l-1}(\alpha)$ 反号, 故 $p_{l+1}(\alpha)$ 与 $p_l(\alpha)$ 反号. 从而 $s_{l+1}(\alpha)=s_l(\alpha)=r$.

这就对 $k=l+1$ 证明了结论成立. 由归纳法原理知定理对一切 $k\leqslant n$ 都成立. □

推论 7.4 *设 $\boldsymbol{T}$ 是不可约对称三对角矩阵, $p_k(\lambda)$ 是 $\boldsymbol{T}-\lambda\boldsymbol{I}$ 的前 k 阶主子式, $s_n(\alpha)$ 是序列 $\{p_k(\alpha)\}_0^n$ 的同号数, 则 $s_n(\alpha)$ 等于 $\boldsymbol{T}$ 在区间 $[\alpha,+\infty)$ 中特征值的个数.*

利用推论 7.4, 可以确定在任意区间 $(\alpha,\beta]$ 中所含 $\boldsymbol{T}$ 的特征值的数目. 实际上, 它就等于 $s_n(\alpha)-s_n(\beta)$. 而且若 $\boldsymbol{T}$ 的特征值排列为

$$\lambda_n<\lambda_{n-1}<\cdots<\lambda_m<\cdots<\lambda_1,$$

$s_n(\alpha)\geqslant m>s_n(\beta)$, 则 $\lambda_m\in[\alpha,\beta)$.

综上所述, 可以给出计算 $\boldsymbol{T}$ 的特征值 λ_m 的二分法.

算法 7.10 (计算三对角矩阵特征值的二分法)

步 1, 输入包含 λ_m 的初始区间 $[a_0, b_0]$ (如取 $a_0 = -\|\boldsymbol{T}\|, b_0 = \|\boldsymbol{T}\|$). 置 $k := 0$.

步 2, 计算 $[a_k, b_k]$ 的中点 $c_k = \dfrac{a_k + b_k}{2}$ 及 $\{p_i(c_k)\}_{i=0}^{n}$ 的同号数 $s_n(c_k)$.

步 3, 若 $s_n(c_k) \geqslant m$, 则 $a_{k+1} := c_k$, $b_{k+1} := b_k$; 否则, $a_{k+1} := a_k$, $b_{k+1} := c_k$.

步 4, 若 $|b_{k+1} - a_{k+1}| \leqslant \varepsilon$, 则令 $\lambda_m \approx \dfrac{1}{2}(a_{k+1} + b_{k+1})$, 停算. 否则, 置 $k := k + 1$, 转步 2.

在算法 7.10 中, λ_m 始终属于区间 $[a_k, b_k]\,(k = 0, 1, \cdots)$. 当 k 充分大时, $[a_k, b_k]$ 的长度 $\dfrac{b_0 - a_0}{2^k}$ 可以小于任意指定的精度, 此时区间中点 c_k 作为 λ_m 的近似值, 其误差不会超过 $\varepsilon/2$. 若二分法的计算是精确的, 则可以得到任意指定精度的近似特征值. 但实际计算中会有舍入误差的影响. 虽然如此, 利用后验误差分析的方法可以证明二分法是一个数值稳定的方法.

注 7.4 在算法 7.10 中, 每一步都要计算 n 个多项式 $p_i(\lambda)\,(i = 1, 2, \cdots, n)$ 在区间 $[a_k, b_k]$ 的中点 c_k 的值 (因 $p_0(\lambda) = 1$, 故不需要计算), 这很容易发生"上溢"或"下溢"现象.

下面讨论 $s = s_n(\alpha)$ 的计算. 由于

$$p_k(\alpha) = (\alpha_k - \alpha)p_{k-1}(\alpha) - \beta_k^2 p_{k-2}(\alpha),\ k = 1, 2, \cdots, n,$$

这里 $p_0(\alpha) = 1$, $\beta_1 = 0$, $p_{-1}(\alpha) = 0$. 令 $s = 0$. 若 $p_{k-1}(\alpha)p_k(\alpha) > 0$, s 加 1. 若 $p_{k-1}(\alpha)p_k(\alpha) < 0$, s 不变. 若 $p_{k-1}(\alpha)p_k(\alpha) = 0$, 由于 $p_{k-1}(\alpha)$ 与 $p_k(\alpha)$ 不能同时为 0, 分为两种情况. 第一种情况是 $p_{k-1}(\alpha) = 0$, $p_k(\alpha) \neq 0$, 此时 $p_{k-1}(\alpha)$ 与 $p_{k-2}(\alpha)\,(\neq 0)$ 同号. 若 $p_{k-2}(\alpha)p_k(\alpha) > 0$, 则 s 加 1, 否则 s 不变. 第二种情况是 $p_{k-1}(\alpha) \neq 0$, $p_k(\alpha) = 0$. 此时 $p_k(\alpha)$ 与 $p_{k-1}(\alpha)\,(\neq 0)$ 同号, s 加 1.

因此, 如果引入

$$q_k(\alpha) = p_k(\alpha)/p_{k-1}(\alpha),\quad k = 1, 2, \cdots, n,$$

则

$$q_k(\alpha) = (\alpha_k - \alpha) - \beta_k^2/q_{k-1}(\alpha),\quad k = 1, 2, \cdots, n. \tag{7.86}$$

对于任意的 $1 \leqslant k \leqslant n$, ① 若 $q_{k-1}(\alpha) \neq 0$, 当式 (7.86) 的计算结果 $q_k(\alpha) \geqslant 0$ 时, s 加 1, 否则, s 不变; ② 若 $q_{k-1}(\alpha) = 0$, 这表明 $p_{k-1}(\alpha) = 0$, 从而有 $p_k(\alpha) = -\beta_k^2 p_{k-2}(\alpha)$. 由于 $\beta_k \neq 0$, $p_k(\alpha)$ 与 $p_{k-2}(\alpha)$ 异号, 故 $p_k(\alpha)$ 与 $p_{k-1}(\alpha)$ 同号, s 加 1.

为此, 定义一个新的序列 $\{q_k(\lambda)\}_{k=1}^{n}$ 如下:

$$q_1(\lambda) = \alpha_1 - \lambda,$$

$$q_k(\lambda) = \begin{cases} \alpha_k - \lambda - \dfrac{\beta_k^2}{q_{k-1}(\lambda)}, & q_{k-1}(\lambda) \neq 0, \\ 1, & q_{k-1}(\lambda) = 0, \end{cases}$$

$$k = 2, 3, \cdots, n.$$

则 $\{p_k(\lambda)\}_{k=0}^{n}$ 的同号数 $s_n(\lambda)$ 等于 $\{q_k(\lambda)\}_{k=1}^{n}$ 中非负数的数目. 故在二分法的实际计算中, 总是通过计算 $\{q_k(\lambda)\}_{k=1}^{n}$ 的非负数的数目来代替计算 $s_n(\lambda)$.

7.5.2 二分法的程序实现

本节考虑算法 7.10 的程序实现. 首先考虑多项式序列 $\{p_k(\lambda)\}_{k=1}^{n}$ 和 $\{q_k(\lambda)\}_{k=1}^{n}$. 编制 MATLAB 程序如下:

```
%本函数计算Sturm序列{p_k(t)}的值
function p=pkfun(t,a,b)
n=length(a);
p(1)=a(1)-t; p(2)=(a(2)-t)*p(1)-b(1);
for k=3:n
    p(k)=(a(k)-t)*p(k-1)-b(k-1)^2*p(k-2);
end
%本函数计算防止溢出的多项式序列{q_k(t)}的值
function q=qkfun(t,a,b)
n=length(a);
q(1)=a(1)-t;
for k=2:n
    if q(k-1)==0
        q(k)=1;
    else
        q(k)=a(k)-t-b(k-1)^2/q(k-1);
    end
end
```

用数据

$$a = [0.8147, 0.9058, 0.1270, 0.9134, 0.6324, 0.0975, 0.2785, 0.5469]^{\mathrm{T}},$$

$$b = [0.9575, 0.9649, 0.1576, 0.9706, 0.9572, 0.4854, 0.8003]^{\mathrm{T}}, \quad t = 6.8,$$

测试两个程序, 得

```
>> a=[0.8147,0.9058,0.1270,0.9134,0.6324,0.0975,0.2785,0.5469]';
>> b=[0.9575,0.9649,0.1576,0.9706,0.9572,0.4854,0.8003]';
>> p=pkfun(6.8,a,b)
p =
    1.0e+06 *
   -0.0000  0.0000 -0.0002  0.0013 -0.0079  0.0517 -0.3355  2.0646
>> q=qkfun(6.8,a,b)
q =
   -5.9853 -5.7410 -6.5108 -5.8828 -6.0075 -6.5500 -6.4855 -6.1543
```

可以看出序列 $\{q_k(\lambda)\}_{k=1}^n$ 的计算的确能有效地防止“溢出”.

现在结合算法 7.10 和注 7.4, 编制 MATLAB 程序如下:

```
%Givens-Householder方法程序-givens_househ.m
function [lambda,k]=givens_househ(T,m,tol,max_it)
%用Givens-Householder求实对称三对角矩阵的第m个特征值
%输入:T为n阶实对称方阵,m特征值序号,
%     tol为容许误差,max_it为最大迭代次数
%输出:lambda为第m个特征值,k为满足精度迭代次数
if nargin<4, max_it=500; end
if nargin<3, tol=1e-6; end
n=size(T,1); k=0;
b=norm(T,inf); a=-b;
alpha=diag(T); beta=diag(T,1);
while(k<max_it)
    k=k+1; c=(a+b)/2;
    q=qkfun(c,alpha,beta);
    tt=find(q>=0);
    sn=length(tt);
    if(sn>=m), a=c; else, b=c; end
    if abs(b-a)<=tol,
        lambda=c; break;
    end
end
```

下面看一个计算实例.

例 7.10 用 Givens–Householder 方法计算矩阵

$$\boldsymbol{A}=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \cdots & n \end{bmatrix}$$

的全部特征值. 取 $n=12$.

解 编制 MATLAB 脚本文件 ex710.m, 然后在命令窗口执行之, 即可得到 $\boldsymbol{A}$ 全部特征值以及计算时间.

7.5.3 特征向量的计算

关于 $\boldsymbol{A}$ 的特征向量的计算, 既可以直接从 $\boldsymbol{A}$ 出发借助已求得的近似特征值利用反迭代法计算, 也可以先用反迭代法求出 $\boldsymbol{T}$ 的相应特征向量, 然后再利用三对角化 (上

Hessenberg 化) 过程所得到的变换矩阵 $\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_{n-2}$ 将其变换为 $\boldsymbol{A}$ 的特征向量. 后者由于其方法简单、计算量小而得到推荐.

设 $(\lambda, \boldsymbol{x})$ 是矩阵 $\boldsymbol{T}$ 的一个特征对, 即 $\boldsymbol{T}\boldsymbol{x} = \lambda\boldsymbol{x}$. 由于

$$\boldsymbol{T} = (\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2})^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2}),$$

故

$$\boldsymbol{A}(\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2})\boldsymbol{x} = \lambda(\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2})\boldsymbol{x},$$

于是

$$\boldsymbol{z} = (\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2})\boldsymbol{x} \tag{7.87}$$

即为 $\boldsymbol{A}$ 的相应于 λ 的特征向量.

实际计算时不是直接按式 (7.87) 进行 $n-2$ 个矩阵乘向量的运算, 而是利用 Householder 矩阵 $\boldsymbol{H}_k = \boldsymbol{I} - 2\boldsymbol{u}_k\boldsymbol{u}_k^{\mathrm{T}}$ 的特殊形式按下列公式计算:

$$\begin{cases} \boldsymbol{y}_{n-1} = \boldsymbol{x}, \\ \boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{y}_{k+1} = \boldsymbol{y}_{k+1} - 2(\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{y}_{k+1})\boldsymbol{u}_k, \\ \boldsymbol{z} = \boldsymbol{y}_1. \end{cases}$$

因此, 问题归结为求不可约对称三对角矩阵 $\boldsymbol{T}$ 的特征向量 $\boldsymbol{x}$. 通常采用所谓的*反迭代法*. 反迭代法本质上是用原点位移的反幂法求矩阵的特征向量, 其基本迭代格式如下:

$$\begin{cases} (\boldsymbol{T} - \mu\boldsymbol{I})\boldsymbol{v}_k = \boldsymbol{z}_{k-1}, \\ \boldsymbol{z}_k = \boldsymbol{v}_k/\|\boldsymbol{v}_k\|_2, \quad k = 1, 2, \cdots, \end{cases} \tag{7.88}$$

式中: μ 为事先取定的位移参数; $\boldsymbol{z}_0$ 为初始向量.

通常用 Givens–Householder 方法 (或其他方法, 如 QR 方法等) 求得某个特征值的近似值后, 即用其作为位移参数, 然后再用式 (7.88) 进行迭代, 会有很快的收敛速度. 反迭代法的基本步骤如下.

算法 7.11 (计算特征向量的反迭代法)

步 1, 计算 $\boldsymbol{A}$ 的上 Hessenberg 分解: $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{T}$.

步 2, 使用算法 7.10 计算 $\boldsymbol{T}$ 的近似特征值 $\bar{\lambda}_1 > \bar{\lambda}_2 > \cdots > \bar{\lambda}_n$.

步 3, 对每个计算得到的特征值 $\bar{\lambda}_i\,(i = 1, 2, \cdots, n)$, 在式 (7.88) 中取 $\mu := \bar{\lambda}_i$ 进行迭代, 求出特征向量 $\boldsymbol{z}_i$.

步 4, 计算 $\boldsymbol{x}_i = \boldsymbol{Q}\boldsymbol{z}_i$ (则 $\boldsymbol{x}_i$ 就是对应于 $\bar{\lambda}_i$ 的近似特征向量).

Givens–Householder 方法可以用来计算对称矩阵的全部特征值问题. 但它可以灵活地求解各种部分特征值问题, 如求若干最大 (最小) 特征值, 求位于给定区间 $[\alpha, \beta]$ 上的特征值等. 这一点它比其他部分特征值问题的算法显得更为优越, 所以这一方法通常也用来计算对称矩阵的部分特征值问题.

算法 7.11 的 MATLAB 程序如下:

```
%基于Givens-Householder方法的求特征向量的反迭代法程序-ghvector.m
function  [Lam,V,ki]=ghvector(A,tol,max_it)
%用反迭代法求实对称矩阵A的第m大的特征向量
%输入:A为n阶对称方阵,m为按降幂排列特征值序号,
%     tol为容许误差,max_it为最大迭代次数
%输出:lambda为第m大的特征值,x为对应的特征向量,k为迭代次数
n=size(A,1); Lam=zeros(n,1); x=rand(n,1); %x=ones(n,1);
[T,Q]=mhessen(A);  %调用上Hessenberg化程序
for i=1:n
    [la]=givens_househ(T,i,tol,max_it); Lam(i)=la;
    [lam,v,k2]=mvpower(T,x,Lam(i));  %调用反幂法程序
    Lam(i)=lam; V(:,i)=v; ki(i)=k2;
    norm(T*v-Lam(i)*v),
end
V=Q*V; %V的每一列为特征向量
```

例 7.11 用算法 7.11 计算例 7.11 中矩阵 $\boldsymbol{A}$ 的全部特征值和特征向量. 取 $n=15$.

解 编制 MATLAB 脚本文件 ex711.m, 然后在命令窗口执行之, 即可得到 $\boldsymbol{A}$ 全部特征值和特征向量以及计算时间.

7.6 Krylov 子空间方法

由于目前计算机存储空间和运算速度的限制, QR 方法以及由其导出的各类方法只适用于小型矩阵特征值和特征向量的计算, 而并不能用于求解大型矩阵的特征值问题. 对于大型稀疏矩阵的特征值问题而言, 目前较为有效和实用的一种求解方法就是 Krylov 子空间方法. 特别是近几年, 随着重开始技术的不断完善, 使得这类方法的效率越来越高, 适用范围越来越广. 本节介绍与 Krylov 子空间方法有关的一些基本概念和重要理论, 并且详细地阐述目前较为成熟的 Arnoldi 方法和 Lanczos 方法.

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 是一个大型稀疏的矩阵, 即 $n\gg 1$ (如 $n\geqslant 10^6$), 而且 $\boldsymbol{A}$ 的非零元很少 (如不超过 10%). 对于这样矩阵的特征值问题, 一般来讲用 QR 方法来求解已经是不可能的事情. 例如, 假如希望用 QR 方法来计算 $\boldsymbol{A}$ 的特征值, 则完成这一计算任务所需的运算量约为 $10n^3$. 如果 $n=10^7$, 而且假定用一个万亿次的计算机来执行这一计算任务, 则所需的计算时间为

$$\frac{10\times 10^{21}}{10^{12}\times 3600\times 24\times 365}\approx 317\ (\text{年})$$

此外, QR 方法所需的存储量是 $O(n^2)=O(10^{14})=O(15\text{G})$, 这也是现在的计算机上无法实现的事情.

那么如何来求解大型的矩阵计算问题呢? 一个直观而朴素的想法是: 先用一个小型的问题"逼近"所考虑的大型问题, 然后用这个小型问题的解近似所求的解. 实现这

一想法的一个途径是, 先选择一个适当的 k 维子空间 $\mathcal{X}_k \subset \mathbb{R}^n, k \ll n$, 然后将所考虑的问题在某种特定的意义下“投影”到 $\mathcal{X}_k$ 上变成一个小型的问题.

目前, 常用的子空间 $\mathcal{X}_k$ 是 Krylov 子空间:

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \operatorname{span}\{\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{v}\},$$

式中: $\boldsymbol{v} \in \mathbb{R}^n$ 是人为给定的.

选择这一空间是因为它有一些特有的优点:

(1) 所需信息量少: 只需一个矩阵 $\boldsymbol{A}$, 一个向量 $\boldsymbol{v}$ 和一个正整数 k.

(2) 计算简单: 只涉及矩阵乘向量, 可以充分利用 $\boldsymbol{A}$ 所具有的稀疏性.

(3) 容易扩张: $\mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) + \operatorname{span}\{\boldsymbol{A}^k\boldsymbol{v}\}$.

当所有的子空间确定之后, 下一步的任务就是如何将所考虑的大型问题“投影”到 $\mathcal{X}_k$ 上. 设 $\boldsymbol{V}_k \in \mathbb{R}^{n\times k}$ 是 $\mathcal{X}_k$ 上的一组标准正交基构成的矩阵, 即若记 $\boldsymbol{V}_k = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k]$, 则有

$$\mathcal{X}_k = \operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}, \quad \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k = \boldsymbol{I}. \tag{7.89}$$

有了 $\boldsymbol{V}_k$ 之后, 就可以计算 $\boldsymbol{A}$ 在 $\mathcal{X}_k$ 上的投影

$$\boldsymbol{H}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k.$$

现在来看 $\boldsymbol{H}_k$ 的几何解释. 令 $\mathcal{A}: \mathbb{R}^n \mapsto \mathbb{R}^n$ 为

$$\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}, \quad \forall\, \boldsymbol{x} \in \mathbb{R}^n.$$

则有

$$\begin{array}{ccc} \mathcal{X}_k & \xrightarrow{\mathcal{A}|_{\mathcal{X}_k}} & \mathbb{R}^n \\ & \searrow & \big\downarrow \mathcal{P}_{\mathcal{X}_k} \\ & \mathcal{P}_{\mathcal{X}_k}\circ\mathcal{A}|_{\mathcal{X}_k} & \mathcal{X}_k \end{array}.$$

而且 $\boldsymbol{H}_k$ 正好是线性算子 $\mathcal{P}_{\mathcal{X}_k} \circ \mathcal{A}|_{\mathcal{X}_k}$ 在基向量 $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$ 下的矩阵表示, 其中 $\mathcal{A}|_{\mathcal{X}_k}$ 表示算子 $\mathcal{A}$ 在子空间 $\mathcal{X}_k$ 上的限制, $\mathcal{P}_{\mathcal{X}_k}$ 表示 $\mathcal{X}_k$ 上的正交投影算子, $\mathcal{P}_{\mathcal{X}_k} \circ \mathcal{A}|_{\mathcal{X}_k}$ 表示算子 $\mathcal{P}_{\mathcal{X}_k}$ 和 $\mathcal{A}|_{\mathcal{X}_k}$ 的复合.

事实上, 只要注意到: ① $\boldsymbol{x} \in \mathcal{X}_k$ 当且仅当存在 $\boldsymbol{y} \in \mathbb{R}^k$ 使得 $\boldsymbol{x} = \boldsymbol{V}_k\boldsymbol{y}$; ② $\mathcal{P}_{\mathcal{X}_k}$ 的矩阵表示为 $\boldsymbol{V}_k\boldsymbol{V}_k^{\mathrm{T}}$. 有

$$\mathcal{P}_{\mathcal{X}_k} \circ \mathcal{A}|_{\mathcal{X}_k}(\boldsymbol{V}_k\boldsymbol{y}) = \boldsymbol{V}_k\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}, \quad \forall\, \boldsymbol{y} \in \mathbb{R}^k,$$

即

$$\mathcal{P}_{\mathcal{X}_k} \circ \mathcal{A}|_{\mathcal{X}_k}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k.$$

这表明, $\boldsymbol{H}_k$ 正好是线性算子 $\mathcal{A}|_{\mathcal{X}_k}$ 在基向量 $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$ 下的矩阵表示.

只要有了 $\boldsymbol{H}_k$ 和 $\boldsymbol{V}_k$, 涉及大矩阵 $\boldsymbol{A}$ 的计算问题就可用相应的小矩阵 $\boldsymbol{H}_k$ 对应的问题来解决.

设 $(\lambda, \boldsymbol{x})$ 是矩阵 $\boldsymbol{A}$ 的一个特征对, 即

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}. \tag{7.90}$$

再假定

$$\boldsymbol{x} = \boldsymbol{V}_k \boldsymbol{y} + \boldsymbol{v} \approx \boldsymbol{V}_k \boldsymbol{y}.$$

于是, 由式 (7.90), 有

$$\boldsymbol{A}\boldsymbol{V}_k \boldsymbol{y} \approx \lambda \boldsymbol{V}_k \boldsymbol{y}.$$

从而有

$$\boldsymbol{H}_k \boldsymbol{y} = \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A}\boldsymbol{V}_k \boldsymbol{y} \approx \lambda \boldsymbol{y}.$$

因此, 若 $(\lambda, \boldsymbol{y})$ 是 $\boldsymbol{H}_k = \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A}\boldsymbol{V}_k$ 的特征对, 则可用 $(\lambda, \boldsymbol{V}_k \boldsymbol{y})$ 近似 $\boldsymbol{A}$ 的特征对. 这样就将大型特征值问题 (7.90) 转化为一个小型特征值问题:

$$\boldsymbol{H}_k \boldsymbol{y} = \lambda \boldsymbol{y}.$$

这就是用 Krylov 子空间方法求解大规模特征值问题的基本思想.

7.6.1 Rayleigh–Ritz 投影方法

基于 Krylov 子空间来设计特征值问题数值解法的最基本的技术就是投影. 下面介绍 Rayleigh–Ritz 投影方法及其基本性质.

设 $\boldsymbol{A} \in \mathbb{R}^{n \times n}, \mathcal{V} \subset \mathbb{C}^n$ 是一个 m 维子空间 (不一定是 Krylov 子空间), 借助 $\mathcal{V}$ 给出 $\boldsymbol{A}$ 的某些近似特征对. 达到这一目的的一种途径就是正交投影法, 其基本思想是选择 $\mu \in \mathbb{C}$ 和 $\boldsymbol{u} \in \mathcal{V}$ 使得

$$(\boldsymbol{A}\boldsymbol{u} - \mu \boldsymbol{u}) \perp \mathcal{V}. \tag{7.91}$$

通常称条件 (7.91) 为 Galerkin 条件. 这一条件可以等价地表述为

$$(\boldsymbol{A}\boldsymbol{u} - \mu \boldsymbol{u}, \boldsymbol{v}) = 0, \quad \forall\, \boldsymbol{v} \in \mathcal{V}. \tag{7.92}$$

设 $\mathcal{V}$ 的一组标准正交基为 $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m$, 并令 $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m]$, 则 $\boldsymbol{u}$ 可表示为 $\boldsymbol{u} = \boldsymbol{V}_m \boldsymbol{y}$, 其中 $\boldsymbol{y} \in \mathbb{C}^m$. 这样式 (7.92) 可以等价地表示为

$$\boldsymbol{V}_m^{\mathrm{H}} (\boldsymbol{A}\boldsymbol{V}_m \boldsymbol{y} - \mu \boldsymbol{V}_m \boldsymbol{y}) = \boldsymbol{0},$$

即

$$\boldsymbol{A}_m \boldsymbol{y} = \mu \boldsymbol{y}, \tag{7.93}$$

其中 $\boldsymbol{A}_m = \boldsymbol{V}_m^{\mathrm{H}} \boldsymbol{A}\boldsymbol{V}_m$ 正好是 $\boldsymbol{A}$ 关于 $\boldsymbol{V}_m$ 的 Rayleigh 商. 这里, 对 $\boldsymbol{A}_m$ 的每一个特征对 $(\mu, \boldsymbol{y})$, 称 μ 为 Ritz 值, 而称 $\boldsymbol{u} = \boldsymbol{V}_m \boldsymbol{y}$ 为 Ritz 向量, 称 $(\mu, \boldsymbol{V}_m \boldsymbol{y})$ 为 Ritz 对. 在一定条件下, Ritz 对 $(\mu, \boldsymbol{V}_m \boldsymbol{y})$ 将是 $\boldsymbol{A}$ 的一个很好的近似特征对.

上述的这一求 $\boldsymbol{A}$ 的某些近似特征值和近似特征向量的方法就称为 Rayleigh–Ritz *投影方法*, 其基本步骤如下.

算法 7.12 (Rayleigh–Ritz 投影方法)

步 1, 求 $\mathcal{V}$ 的一组标准正交基 $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m\}$, 并记 $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m]$.

步 2, 计算 Rayleigh 商 $\boldsymbol{A}_m = \boldsymbol{V}_m^{\mathrm{H}} \boldsymbol{A} \boldsymbol{V}_m$.

步 3, 计算 $\boldsymbol{A}_m$ 的特征值, 并选择其中若干所需的作为 $\boldsymbol{A}$ 的近似值: $\mu_1, \mu_2, \cdots, \mu_k$.

步 4, 计算 μ_i 所对应的特征向量 $\boldsymbol{y}_i$, 并形成 Ritz 向量 $\boldsymbol{u}_i = \boldsymbol{V}_m \boldsymbol{y}_i, i = 1, 2, \cdots, k$.

一般来讲, 有 $m \ll n$. 因此, $\boldsymbol{A}_m$ 将是一个阶数很小的矩阵. 这样, 就有很多成熟的数值方法用来完成上述方法中步 3 和步 4 的计算任务. 例如, 可先用 QR 方法来计算 $\boldsymbol{A}_m$ 特征值, 然后再用反幂法来求其对应的特征向量. 另外需要说明的一点是, 由于特征向量比 Schur 向量对扰动敏感得多, 因此, 在某些情况下, 步 4 中的"求 Ritz 向量"可以用"求 Schur 向量"来代替 (所谓 Schur 向量是指实矩阵 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 的实 Schur 分解 $\boldsymbol{A} = \boldsymbol{U}^{\mathrm{T}} \boldsymbol{T} \boldsymbol{U}$ 中正交矩阵 $\boldsymbol{U}$ 的列向量, 其中 $\boldsymbol{T}$ 为对角块为 1×1 或 2×2 子矩阵的拟上三角矩阵).

下面给出用 Ritz 对来近似特征对的一个理论依据. 这里假定 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 且 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, 即 $\boldsymbol{A}$ 是 n 阶实对称矩阵.

设 $\boldsymbol{V} = [\boldsymbol{V}_k, \boldsymbol{V}_u]$ 是一个 n 阶正交矩阵, 其中 $\boldsymbol{V}_k \in \mathbb{R}^{n \times k}, \boldsymbol{V}_u \in \mathbb{R}^{n \times (n-k)}$. 在实际应用中, 通常 $\boldsymbol{V}_k$ 是已知的, 是由 Lanczos 方法产生的 Krylov 子空间的一组标准正交基构成的, 而 $\boldsymbol{V}_u$ 通常是不知道的. 在下面的讨论中, 将使用如下的符号:

$$\boldsymbol{T} = \boldsymbol{V}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_k & \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_u \\ \boldsymbol{V}_u^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_k & \boldsymbol{V}_u^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_u \end{bmatrix} := \begin{bmatrix} \boldsymbol{T}_k & \boldsymbol{T}_{uk} \\ \boldsymbol{T}_{ku} & \boldsymbol{T}_u \end{bmatrix}. \tag{7.94}$$

Ritz 对作为 $\boldsymbol{A}$ 的特征对的"最佳"逼近有多种理由. 其中之一可以从上面的矩阵看出, 由于 $\boldsymbol{T}_{uk}$ 和 $\boldsymbol{T}_u$ 是不知道的, 而 $\boldsymbol{T}_k$ 是唯一知道的, 因此自然只能借助 $\boldsymbol{T}_k$ 的特征对来构造出 $\boldsymbol{A}$ 的近似特征对. 另一个理由是下面的定理 7.17 所描述的 Rayleigh 商的最佳逼近性.

如所周知, 若对某个 k 阶矩阵 $\boldsymbol{R}$ 有 $\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k \boldsymbol{R}$, 则 $\mathcal{R}(\boldsymbol{V}_k)$ 就是 $\boldsymbol{A}$ 的一个不变子空间, 从而 $\boldsymbol{R}$ 的特征值全都是 $\boldsymbol{A}$ 的特征值, 即 $\lambda(\boldsymbol{R}) \subset \lambda(\boldsymbol{A})$. 如果上述等式不能成立, 自然希望找到一个 k 阶矩阵 $\boldsymbol{R}$, 使得

$$\|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k \boldsymbol{R}\|_2 = \min,$$

然后用 $\boldsymbol{R}$ 的特征值来近似 $\boldsymbol{A}$ 的特征值. 下面的定理表明 Rayleigh 商 $\boldsymbol{T}_k = \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_k$ 就是满足此条件的矩阵.

定理 7.17 设 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 满足 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, $\boldsymbol{V}_k \in \mathbb{R}^{n \times k}$ 满足 $\boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{V}_k = \boldsymbol{I}$, 则有

(1) $\boldsymbol{T}_k$ 满足

$$\begin{aligned} \|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k \boldsymbol{T}_k\|_2 &= \min\{\|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k \boldsymbol{S}\|_2 : \boldsymbol{S}^{\mathrm{T}} = \boldsymbol{S} \in \mathbb{R}^{k \times k}\} \\ &= \|\boldsymbol{T}_{ku}\|_2. \end{aligned} \tag{7.95}$$

(2) 若 $\boldsymbol{T}_k$ 的谱分解为 $\boldsymbol{T}_k = \boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}}$, 其中 $\boldsymbol{Y}$ 是正交矩阵, $\boldsymbol{M}$ 是对角矩阵, 则

$$\|\boldsymbol{A}(\boldsymbol{V}_k \boldsymbol{Y}) - (\boldsymbol{V}_k \boldsymbol{Y})\boldsymbol{M}\|_2 = \min\{\|\boldsymbol{A}\boldsymbol{P}_k - \boldsymbol{P}_k \boldsymbol{D}\|_2 : \boldsymbol{P}_k \in \mathcal{P},\ \boldsymbol{D} \in \mathcal{D}\}$$

$$= \|\boldsymbol{T}_{ku}\|_2, \tag{7.96}$$

式中:

$$\mathcal{P} = \{\boldsymbol{P}_k \in \mathbb{R}^{n\times k} : \boldsymbol{P}_k^{\mathrm{T}}\boldsymbol{P}_k = \boldsymbol{I},\ \ \mathcal{R}(\boldsymbol{P}_k) = \mathcal{R}(\boldsymbol{V}_k)\},$$

$$\mathcal{D} = \{\boldsymbol{D} \in \mathbb{R}^{k\times k} : \boldsymbol{D} \text{ 是对角矩阵}\}.$$

证明 (1) 对任意的 $\boldsymbol{S}^{\mathrm{T}} = \boldsymbol{S} \in \mathbb{R}^{k\times k}$, 有

$$\begin{aligned}\|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k\boldsymbol{S}\|_2 &= \|\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k\boldsymbol{S})\|_2 = \|[\boldsymbol{V}_k, \boldsymbol{V}_u]^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k\boldsymbol{S})\|_2 \\ &= \left\|\begin{bmatrix} \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{S} \\ \boldsymbol{V}_u^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_u^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{S} \end{bmatrix}\right\|_2 = \left\|\begin{bmatrix} \boldsymbol{T}_k - \boldsymbol{S} \\ \boldsymbol{T}_{ku} \end{bmatrix}\right\|_2 \\ &\geqslant \|\boldsymbol{T}_{ku}\|_2,\end{aligned}$$

而且当 $\boldsymbol{S} = \boldsymbol{T}_k$ 时上述不等式等号成立. 因此, 定理的结论 (1) 成立.

(2) 任意取 $\boldsymbol{P}_k \in \mathcal{P}$ 和 $\boldsymbol{D} \in \mathcal{D}$, 由 $\mathcal{P}$ 的定义可知, 必存在一个 k 阶正交矩阵 $\boldsymbol{U}$ 使得 $\boldsymbol{P}_k = \boldsymbol{V}_k\boldsymbol{U}$. 再由结论 (1) 所证和谱范数的酉不变性, 有

$$\begin{aligned}\|\boldsymbol{A}\boldsymbol{P}_k - \boldsymbol{P}_k\boldsymbol{D}\|_2 &= \|\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{U} - \boldsymbol{V}_k\boldsymbol{U}\boldsymbol{D}\|_2 = \|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathrm{T}}\|_2 \\ &\geqslant \|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k\boldsymbol{T}_k\|_2 = \|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k\boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}}\|_2 \\ &= \|\boldsymbol{A}(\boldsymbol{V}_k\boldsymbol{Y}) - (\boldsymbol{V}_k\boldsymbol{Y})\boldsymbol{M}\|_2 = \|\boldsymbol{T}_{ku}\|_2.\end{aligned}$$

注意到 $\boldsymbol{P}_k$ 和 $\boldsymbol{D}$ 的任意性, 可知式 (7.96) 成立. 证毕. □

注 7.5 从定理 7.17 的证明可以看出, 将谱范数换为 Frobenius 范数定理的结论仍然成立. 定理的结论 (1) 说明, 对于实对称矩阵 $\boldsymbol{A}$ 来说, Rayleigh 商 $\boldsymbol{T}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k$ 具有最佳逼近性; 而定理的结论 (2) 又说明用 Ritz 对作为其近似特征对是最优的.

一般来讲, $\boldsymbol{T}_{ku}$ 是未知的, 但如果 $\boldsymbol{V}_k$ 是由 Lanczos 过程 (即算法 2.13) 产生的, 则有

$$\boldsymbol{T} = \left[\begin{array}{c|c} \boldsymbol{T}_k & \boldsymbol{T}_{uk} \\ \hline \boldsymbol{T}_{ku} & \boldsymbol{T}_u \end{array}\right] = \left[\begin{array}{ccccc|ccccc} \alpha_1 & \beta_1 & & & & & & & & \\ \beta_1 & \ddots & \ddots & & & & & & & \\ & \ddots & \ddots & \beta_{k-1} & & & & & & \\ & & \beta_{k-1} & \alpha_k & \beta_k & & & & & \\ \hline & & & \beta_k & \alpha_{k+1} & \beta_{k+1} & & & & \\ & & & & \beta_{k+1} & \ddots & \ddots & & & \\ & & & & & \ddots & \ddots & \beta_{n-1} & & \\ & & & & & & \beta_{n-1} & \alpha_n & & \end{array}\right], \tag{7.97}$$

从而 $\boldsymbol{T}_{ku} = \beta_k\boldsymbol{e}_1\boldsymbol{e}_k^{\mathrm{T}}$, 其中 β_k 已由算法 2.13 算出, 于是此时有 $\|\boldsymbol{T}_{ku}\|_2 = \beta_k$. 此外, 利用 $\boldsymbol{T}_{ku}$ 还可以给出 Ritz 对作为近似特征对的绝对误差.

定理 7.18 假设符号同前, 并设 $\boldsymbol{T}_k$ 的谱分解为

$$\boldsymbol{T}_k = \boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}},$$

式中: $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_k]$ 为正交矩阵; $\boldsymbol{M} = \mathrm{diag}(\mu_1, \mu_2, \cdots, \mu_k)$. 则有

(1) 存在 $\boldsymbol{A}$ 的 k 个特征值 $\lambda_1, \cdots, \lambda_k$, 使得

$$|\lambda_i - \mu_i| \leqslant \|\boldsymbol{T}_{ku}\|_2, \quad i = 1, 2, \cdots, k. \tag{7.98}$$

(2) 令 $\boldsymbol{z}_i = \boldsymbol{V}_k \boldsymbol{y}_i$, 则

$$\|\boldsymbol{A}\boldsymbol{z}_i - \mu_i \boldsymbol{z}_i\|_2 = \|\boldsymbol{T}_{ku}\boldsymbol{y}_i\|_2, \quad i = 1, 2, \cdots, k. \tag{7.99}$$

(3) 若 $\boldsymbol{V}_k$ 是由 Lanczos 过程 (算法 2.13) 得到的, 则有

$$\|\boldsymbol{T}_{ku}\|_2 = |\beta_k|, \quad \|\boldsymbol{T}_{ku}\boldsymbol{y}_i\|_2 = |\beta_k (\boldsymbol{y}_i)_k|, \tag{7.100}$$

式中: $(\boldsymbol{y}_i)_k$ 为 $\boldsymbol{y}_i$ 的最后一个分量.

证明 (1) 令 $\widehat{\boldsymbol{T}} = \mathrm{diag}(\boldsymbol{T}_k, \boldsymbol{T}_u)$, 则

$$\|\boldsymbol{T} - \widehat{\boldsymbol{T}}\|_2 = \left\| \begin{bmatrix} \boldsymbol{O} & \boldsymbol{T}_{uk} \\ \boldsymbol{T}_{ku} & \boldsymbol{O} \end{bmatrix} \right\|_2 = \|\boldsymbol{T}_{ku}\|_2,$$

然后应用 Weyl 定理即知式 (7.98) 成立.

(2) 直接计算, 有

$$\begin{aligned} \|\boldsymbol{A}\boldsymbol{z}_i - \mu_i \boldsymbol{z}_i\|_2 &= \|\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_i) - \mu_i \boldsymbol{V}^{\mathrm{T}}(\boldsymbol{V}_k\boldsymbol{y}_i)\|_2 \\ &= \left\| \begin{bmatrix} \boldsymbol{T}_k\boldsymbol{y}_i \\ \boldsymbol{T}_{ku}\boldsymbol{y}_i \end{bmatrix} - \begin{bmatrix} \mu_i\boldsymbol{y}_i \\ \boldsymbol{0} \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{T}_{ku}\boldsymbol{y}_i \end{bmatrix} \right\|_2 = \|\boldsymbol{T}_{ku}\boldsymbol{y}_i\|_2, \end{aligned}$$

故式 (7.99) 成立.

(3) 由式 (7.97) 可得式 (7.100). 证毕. □

7.6.2 Lanczos 方法

Lanczos 方法是目前计算大型稀疏实对称矩阵少数几个特征值和对应特征向量最常用的 Krylov 子空间方法之一. 本节介绍 Lanczos 方法的详细算法及其相关理论.

1. 经典 Lanczos 方法

设 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A} \in \mathbb{R}^{n\times n}$ 已经给定, 希望求 $\boldsymbol{A}$ 的几个最大特征对或最小特征对. 将算法 2.13 与 7.6.1 节所介绍的 Rayleigh–Ritz 方法相结合就可得到完成这一计算任务的经典 Lanczos 方法.

算法 7.13 (经典 Lanczos 方法) 给定一个 n 阶实对称矩阵 $\boldsymbol{A}$. 本算法计算 $\boldsymbol{A}$ 的少数几个两端特征对.

步 1, 选择初始向量 $\boldsymbol{v}$, 并令 $\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|_2$.

步 2, 应用算法 2.13 产生一个长度为 k 的 Lanczos 分解:

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{T}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}.$$

步 3, 计算 $\boldsymbol{T}_k$ 的特征值, 并选择其中满足要求的记为 μ_i, $i = 1, 2, \cdots, \ell$.

步 4, 计算 $\boldsymbol{T}_k$ 对应于 μ_i 的特征向量 $\boldsymbol{y}_i$, 并形成 Ritz 向量 $\boldsymbol{u}_i = \boldsymbol{V}_k\boldsymbol{y}_i$, $i = 1, 2, \cdots, \ell$.

步 5, 检验是否已经满足要求. 如果不满足要求, 则增加 k, 再返回到步 2, 重复以上各步.

注 7.6 定理 7.18 说明, 若 Rayleigh 商 $\boldsymbol{T}_k$ 的一个特征对 $(\mu, \boldsymbol{y})$ 使得 $|\beta_k|\cdot|\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y}|$ 很小, 则 μ 就是 $\boldsymbol{A}$ 的某个特征值的很好近似. 因此, 通常在算法 7.13 中就是以 $|\beta_k|\cdot|\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y}|$ 是否已经足够小来判断 Ritz 对 $(\mu, \boldsymbol{V}_k\boldsymbol{y})$ 是否可以作为 $\boldsymbol{A}$ 的近似特征对.

经典 Lanzcos 方法的 MATLAB 程序如下:

```
function [mu,U,lambda]=Class_Lanczos(A,v,k)
%输入:A为对称矩阵,v为初始向量,k为子空间的维数
%输出:lambda和U分别是A的k个特征值及相应的特征向量
tol=1.e-12;
v=v/norm(v); V(:,1)=v;
for i=1:k
    [V,T,beta]=Lanczos2(A,v,i);
    [Y,Mu]=eig(T);
    mu{i}=diag(Mu);
    U=V*Y;
    if beta*abs(Y(i,i))<tol
       k=i; break;
    end
end
lambda=mu{k};
```

例 7.12 令 $\boldsymbol{A} \in \mathbb{R}^{500\times 500}$ 是一个随机产生的对角矩阵, 其对角元服从正态分布. 对于 k 从 1 到 40, 利用算法 7.13 计算出 $\boldsymbol{T}_k$ 的特征值, 并标注在图 7.2 中. 图中第 k 列 "+" 表示 $\boldsymbol{T}_k$ 的所有特征值, 最后一列 "+" 表示 $\boldsymbol{A}$ 的精确特征值.

从这一数值例子的计算结果可以看出:

(1) 两端特征值 (即最大和最小特征值) 收敛最快, 而内部特征值收敛较慢.

(2) $\boldsymbol{T}_k$ 的第 i 个最大 (或最小) 特征值单调增加 (或减少) 地收敛到 $\boldsymbol{A}$ 的第 i 个最大 (或最小) 特征值.

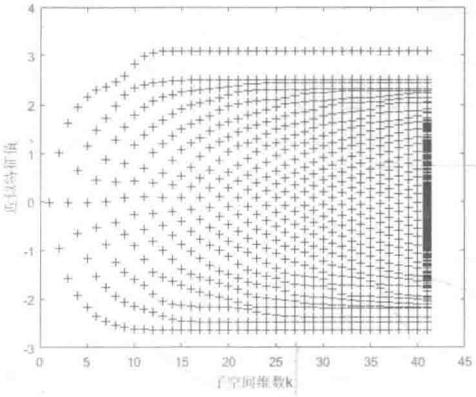

图 7.2 Lanczos 算法的收敛特性

2. 收敛性理论

下面对 Lanczos 方法及其收敛性进行一些理论上的分析. 为此, 先引进两个基本概念.

设 $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ 是两个非零向量, 定义 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 之间的夹角为

$$\theta := \theta(\boldsymbol{x}, \boldsymbol{y}) = \arccos \frac{|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}|}{\|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2}, \tag{7.101}$$

其几何解释如图 7.3 所示.

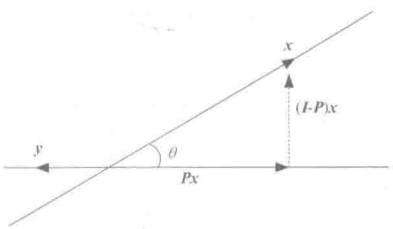

图 7.3 向量 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 之间的夹角

令

$$\boldsymbol{P} = \frac{1}{\|\boldsymbol{y}\|_2^2} \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}},$$

即 $\boldsymbol{P}$ 是到 $\mathrm{span}\{\boldsymbol{y}\}$ 上的正交投影, 则有

$$\tan\theta = \frac{\|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}\|_2}{\|\boldsymbol{P}\boldsymbol{x}\|_2}, \quad \sin\theta = \frac{\|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}, \quad \cos\theta = \frac{\|\boldsymbol{P}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}.$$

设 $\boldsymbol{x} \in \mathbb{R}^n$ 是一个非零向量, $\mathcal{V}$ 是一个子空间, 则定义 $\boldsymbol{x}$ 与 $\mathcal{V}$ 之间的夹角为

$$\theta(\boldsymbol{x}, \mathcal{V}) = \min\{\theta(\boldsymbol{x}, \boldsymbol{u}) : \boldsymbol{0} \neq \boldsymbol{u} \in \mathcal{V}\}. \tag{7.102}$$

再假设 $\boldsymbol{P}$ 是 $\mathcal{V}$ 上的正交投影, 则有

$$\tan\theta(\boldsymbol{x},\mathcal{V})=\frac{\|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}\|_2}{\|\boldsymbol{P}\boldsymbol{x}\|_2}. \tag{7.103}$$

式 (7.103) 的几何解释如图 7.4 所示.

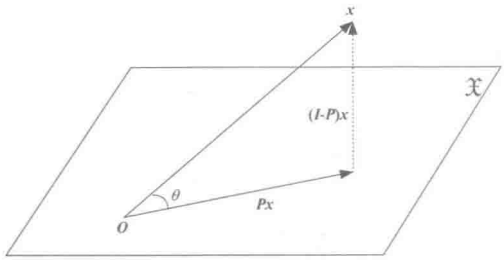

图 7.4 向量 $\boldsymbol{x}$ 与子空间 $\mathcal{X}$ 之间的夹角

事实上, 可以证明 $\theta(\boldsymbol{x},\mathcal{V})=\theta$, 其中 θ 如图 7.4 所示. 记 $\boldsymbol{x}_1=\boldsymbol{P}\boldsymbol{x},\boldsymbol{x}_2=(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}$, 则有

$$\boldsymbol{x}=\boldsymbol{x}_1+\boldsymbol{x}_2,\quad \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2=0,\quad \boldsymbol{x}_2\perp\mathcal{V},$$

于是对任意的 $\boldsymbol{u}\in\mathcal{V}$, 有

$$|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{u}|=|\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{u}|\leqslant\|\boldsymbol{x}_1\|_2\|\boldsymbol{u}\|_2,$$

从而

$$\frac{|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{u}|}{\|\boldsymbol{x}\|_2\|\boldsymbol{u}\|_2}\leqslant\frac{\|\boldsymbol{x}_1\|_2}{\|\boldsymbol{x}\|_2}=\frac{|\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1|}{\|\boldsymbol{x}\|_2\|\boldsymbol{x}_1\|_2}=\frac{|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}_1|}{\|\boldsymbol{x}\|_2\|\boldsymbol{x}_1\|_2}.$$

于是有

$$\theta(\boldsymbol{x},\boldsymbol{u})\geqslant\theta(\boldsymbol{x},\boldsymbol{x}_1)=\theta.$$

注意到 $\boldsymbol{u}$ 的任意性, 便有 $\theta=\min\limits_{\boldsymbol{u}\in\mathcal{X}}\theta(\boldsymbol{x},\boldsymbol{u})=\theta(\boldsymbol{x},\mathcal{V})$.

为了叙述简洁, 先将下面所要使用的符号作一说明. 对给定的 $\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 和 $\boldsymbol{0}\neq\boldsymbol{v}\in\mathbb{R}^n$, 简记

$$\mathcal{K}_k=\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v}),$$

并假定与其对应的长度为 k 的 Lanczos 分解为

$$\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_k\boldsymbol{T}_k+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}},$$

式中: $\boldsymbol{v}_1=\boldsymbol{v}/\|\boldsymbol{v}\|_2$. 再假定 $\boldsymbol{A}$ 和 $\boldsymbol{T}_k$ 的谱分解分别为

$$\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}\ \text{ 和 }\ \boldsymbol{T}_k=\boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}},$$

式中:

$$\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n),\quad \lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n,$$

$$\boldsymbol{M}=\mathrm{diag}(\mu_1,\mu_2,\cdots,\mu_k),\quad \mu_1\geqslant\mu_2\geqslant\cdots\geqslant\mu_k,$$
$$\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n],\quad \boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_k].$$

如果没有特别说明, 在本节后续部分中所遇到的这些符号均为上面所述的含义.

首先给出一个简单的结果. 若 $(\lambda,\boldsymbol{u})$ 是 $\boldsymbol{A}$ 的一个特征对, 而且如果 $\boldsymbol{u}\in\mathcal{K}_k$, 则 λ 必是 $\boldsymbol{T}_k$ 的特征值. 事实上, $\boldsymbol{u}\in\mathcal{K}_k$ 蕴涵着必存在 $\boldsymbol{y}$ 使得 $\boldsymbol{u}=\boldsymbol{V}_k\boldsymbol{y}$, 这样利用 Lanczos 分解有

$$\lambda\boldsymbol{V}_k\boldsymbol{y}=\lambda\boldsymbol{u}=\boldsymbol{A}\boldsymbol{u}=\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}=\boldsymbol{V}_k\boldsymbol{T}_k\boldsymbol{y}+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y}.$$

在上面的等式两边左乘 $\boldsymbol{V}_k^{\mathrm{T}}$, 并注意到 $\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{v}_{k+1}=\boldsymbol{0}$, 有

$$\lambda\boldsymbol{y}=\boldsymbol{T}_k\boldsymbol{y},$$

即 λ 是 $\boldsymbol{T}_k$ 的特征值 (这里 $\boldsymbol{u}\neq\boldsymbol{0}$ 蕴涵着 $\boldsymbol{y}\neq\boldsymbol{0}$). 因此, 如果 $\boldsymbol{u}\notin\mathcal{K}_k$, 但若 $\boldsymbol{u}$ 与 $\mathcal{K}_k$ 很靠近的话, 可望 $\boldsymbol{T}_k$ 有特征值是 λ 的很好近似. 进一步, 可以证明如下结果.

定理 7.19 令 $\boldsymbol{A}\boldsymbol{u}=\lambda\boldsymbol{u}$, $\|\boldsymbol{u}\|_2=1$, 则必存在 $\boldsymbol{T}_k$ 的一个特征值 μ, 使得

$$|\mu-\lambda|\leqslant\|\boldsymbol{A}\|_2\tan\theta(\boldsymbol{u},\mathcal{K}_k),\tag{7.104}$$

式中: $\mathcal{K}_k=\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})$.

证明 令 $\boldsymbol{P}=\boldsymbol{V}_k\boldsymbol{V}_k^{\mathrm{T}}$ 是 $\mathcal{K}_k$ 上的正交投影, 并记

$$\boldsymbol{u}_1=\boldsymbol{P}\boldsymbol{u},\quad \boldsymbol{u}_2=(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{u},$$

则有

$$\tan\theta(\boldsymbol{u},\mathcal{K}_k)=\frac{\|\boldsymbol{u}_2\|_2}{\|\boldsymbol{u}_1\|_2},\tag{7.105}$$

而且

$$\begin{aligned}\lambda\boldsymbol{u}_1+\lambda\boldsymbol{u}_2=\lambda\boldsymbol{u}&=\boldsymbol{A}\boldsymbol{u}=\boldsymbol{A}\boldsymbol{u}_1+\boldsymbol{A}\boldsymbol{u}_2\\&=\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1+\boldsymbol{A}\boldsymbol{u}_2\\&=\boldsymbol{V}_k\boldsymbol{T}_k\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1+\boldsymbol{A}\boldsymbol{u}_2.\end{aligned}$$

在上式两边左乘 $\boldsymbol{V}_k^{\mathrm{T}}$, 得

$$\lambda\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1=\boldsymbol{T}_k\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1+\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}_2,$$

从而有

$$(\lambda\boldsymbol{I}-\boldsymbol{T}_k)\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1=\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}_2.$$

于是

$$\min_{\mu\in\lambda(\boldsymbol{T}_k)}|\lambda-\mu|\cdot\|\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2=\min_{\mu\in\lambda(\boldsymbol{T}_k)}|\lambda-\mu|\cdot\|\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2$$

$$\leqslant \|(\lambda \boldsymbol{I} - \boldsymbol{M})\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2 \tag{7.106}$$

$$= \|(\lambda \boldsymbol{I} - \boldsymbol{T}_k)\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2 = \|\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}_2\|_2$$

$$\leqslant \|\boldsymbol{V}_k^{\mathrm{T}}\|_2\|\boldsymbol{A}\|_2\|\boldsymbol{u}_2\|_2 = \|\boldsymbol{A}\|_2\|\boldsymbol{u}_2\|_2.$$

此外, 由于 $\boldsymbol{u}_1 \in \mathcal{K}_k$, 故存在 $\boldsymbol{y}$ 使得 $\boldsymbol{u}_1 = \boldsymbol{V}_k\boldsymbol{y}$, 从而有

$$\|\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2 = \|\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{y}\|_2 = \|\boldsymbol{y}\|_2 = \|\boldsymbol{u}_1\|_2, \tag{7.107}$$

将式 (7.105)、式 (7.106) 与式 (7.107) 相结合便得到所要证的不等式 (7.104). 证毕. □

注 7.7 定理 7.19 是说, 如果对应于特征值 λ 的特征向量 $\boldsymbol{u}$ 与 Krylov 子空间 $\mathcal{K}_k$ 之间的距离 $\tan\theta(\boldsymbol{u}, \mathcal{K}_k)$ 很小, 则 $\boldsymbol{T}_k$ 就必有一个特征值 μ 和 λ 很靠近. 因此, 可望由 Lanczos 算法得到 λ 的一个很好的近似值.

下面的定理给出了第 i 个特征向量 $\boldsymbol{u}_i$ 与 $\mathcal{K}_k$ 之间的距离 $\tan\theta(\boldsymbol{u}_i, \mathcal{K}_k)$ 的一个可计算的上界估计.

定理 7.20 对给定的 $i\,(1 \leqslant i < k)$, 若

$$\lambda_{i-1} > \lambda_i, \quad \lambda_{i+1} > \lambda_n, \quad \boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}_i \neq 0,$$

则有

$$\tan\theta(\boldsymbol{u}_i, \mathcal{K}_k) \leqslant \frac{\xi_i}{C_{k-i}(1+2\delta_i)}\tan\theta(\boldsymbol{u}_i, \boldsymbol{v}), \tag{7.108}$$

式中: $\mathcal{K}_k = \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$; $\delta_i = \dfrac{\lambda_i - \lambda_{i+1}}{\lambda_{i+1} - \lambda_n}$; $C_{k-i}(t)$ 为 $k-i$ 次 Chebyshev 多项式; 而

$$\xi_i = \begin{cases} 1, & i = 1, \\ \prod\limits_{s=1}^{i-1} \frac{\lambda_s - \lambda_n}{\lambda_s - \lambda_i}, & i > 1. \end{cases} \tag{7.109}$$

证明 记 $\boldsymbol{P}_i = \boldsymbol{u}_i\boldsymbol{u}_i^{\mathrm{T}}$, 即 $\boldsymbol{P}_i$ 是到子空间 $\mathrm{span}\{\boldsymbol{u}_i\}$ 上的正交投影, 因此, 有

$$\tan\theta(\boldsymbol{u}_i, \boldsymbol{v}) = \frac{\|(\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{v}\|_2}{\|\boldsymbol{P}_i\boldsymbol{v}\|_2} \quad (\boldsymbol{P}_i\boldsymbol{v} \neq \boldsymbol{0}). \tag{7.110}$$

此外, 任取 $\boldsymbol{x} \in \mathcal{K}_k$, 则必存在一个 $p \in \mathcal{P}_{k-1}$, 使得 $\boldsymbol{x} = p(\boldsymbol{A})\boldsymbol{v}$. 当然这里假定 $\boldsymbol{x} \neq \boldsymbol{0}$. 注意到 $\boldsymbol{P}_i\boldsymbol{A} = \boldsymbol{A}\boldsymbol{P}_i$, 便有

$$\boldsymbol{P}_i\boldsymbol{x} = p(\boldsymbol{A})\boldsymbol{P}_i\boldsymbol{v} = p(\lambda_i)\boldsymbol{P}_i\boldsymbol{v},$$
$$(\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{x} = p(\boldsymbol{A})(\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{v} = p(\boldsymbol{A})\boldsymbol{v}_i,$$

式中: $\boldsymbol{v}_i = (\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{v}$. 这样

$$\tan\theta(\boldsymbol{u}_i, \boldsymbol{x}) = \frac{\|(\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{x}\|_2}{\|\boldsymbol{P}_i\boldsymbol{x}\|_2} = \frac{\|p(\boldsymbol{A})\boldsymbol{v}_i\|_2}{|p(\lambda_i)| \cdot \|\boldsymbol{P}_i\boldsymbol{v}\|_2}$$

$$= \left\| \frac{p(\boldsymbol{A})}{p(\lambda_i)} \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|_2} \right\|_2 \frac{\|(\boldsymbol{I}-\boldsymbol{P}_i)\boldsymbol{v}\|_2}{\|\boldsymbol{P}_i\boldsymbol{v}\|_2}$$

$$= \|\widehat{p}(\boldsymbol{A})\widehat{\boldsymbol{v}}_i\|_2 \tan\theta(\boldsymbol{u}_i, \boldsymbol{v}), \tag{7.111}$$

式中: $\widehat{p}(t) = p(t)/p(\lambda_i)$; $\widehat{\boldsymbol{v}}_i = \boldsymbol{v}_i/\|\boldsymbol{v}_i\|_2$. 这里最后一个等式用到了式 (7.110).

现将 $\widehat{\boldsymbol{v}}_i$ 用 $\boldsymbol{A}$ 的特征向量展开, 即令

$$\widehat{\boldsymbol{v}}_i = \alpha_1\boldsymbol{u}_1 + \alpha_2\boldsymbol{u}_2 + \cdots + \alpha_n\boldsymbol{u}_n.$$

而 $\widehat{\boldsymbol{v}}_i \perp \boldsymbol{u}_i$, 故必有 $\alpha_i = 0$. 此外, $\alpha_1^2 + \cdots + \alpha_n^2 = \|\widehat{\boldsymbol{v}}_i\|_2^2 = 1$. 这样, 有

$$\|\widehat{p}(\boldsymbol{A})\widehat{\boldsymbol{v}}_i\|_2^2 = \sum_{s=1}^{n} \alpha_s^2 \widehat{p}(\lambda_s)^2 \leqslant \max_{1\leqslant s\leqslant n, s\neq i} |\widehat{p}(\lambda_s)|^2. \tag{7.112}$$

现在取一个特殊的多项式 $p_* \in \mathcal{P}_{k-1}$:

$$p_*(t) = \frac{(\lambda_1 - t)(\lambda_2 - t)\cdots(\lambda_{i-1} - t)}{(\lambda_1 - \lambda_i)(\lambda_2 - \lambda_i)\cdots(\lambda_{i-1} - \lambda_i)} v(t),$$

这里

$$v(t) = C_{k-i}\Big(\frac{2t - (\lambda_{i+1} + \lambda_n)}{\lambda_{i+1} - \lambda_n}\Big) / C_{k-i}(1 + 2\delta_i).$$

容易验证, 对于这个多项式 p_*, 有

$$p_*(\lambda_i) = 1, \quad p_*(\lambda_s) = 0, \quad s = 1, 2, \cdots, i-1.$$

从而有 $p_*(t) \equiv \widehat{p}_*(t)$, 而且

$$\max_{1\leqslant s\leqslant n,\, s\neq i} |p_*(\lambda_s)| = \max_{i+1\leqslant s\leqslant n} |p_*(\lambda_s)| \leqslant \xi_i \max_{i+1\leqslant s\leqslant n} |v(\lambda_s)|$$

$$\leqslant \frac{\xi_i}{C_{k-i}(1 + 2\delta_i)}, \tag{7.113}$$

这里最后一个等式用到 Chebyshev 多项式的性质. 现将式 (7.111)、式 (7.112) 和式 (7.113) 相结合, 有

$$\tan\theta(\boldsymbol{u}_i, \mathcal{K}_k) = \min_{x\in\mathcal{K}_k} \tan\theta(\boldsymbol{u}_i, \boldsymbol{x})$$

$$\leqslant \max_{1\leqslant s\leqslant n,\, s\neq i} |p_*(\lambda_s)| \tan\theta(\boldsymbol{u}_i, \boldsymbol{v})$$

$$\leqslant \frac{\xi_i}{C_{k-i}(1 + 2\delta_i)} \tan\theta(\boldsymbol{u}_i, \boldsymbol{v}).$$

这正是要证的结论. 证毕. □

定理 7.21 (Kaniel–Saad 定理) 对于任意给定的 $i\,(1 \leqslant i < k)$, 若

$$\lambda_i \neq \mu_s, \quad s = 1, 2, \cdots, i-1, \quad \lambda_{i+1} > \lambda_n,$$

而且 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}_i \neq 0$, 则有

$$0 \leqslant \lambda_i - \mu_i \leqslant (\lambda_i - \lambda_n)\left[\frac{\zeta_i}{C_{k-i}(1+2\delta_i)}\tan\theta(\boldsymbol{u}_i, \boldsymbol{v})\right]^2, \tag{7.114}$$

其中 δ_i 由定理 7.20 给出, 而

$$\zeta_i = \begin{cases} 1, & i = 1, \\ \max\limits_{i+1\leqslant \ell \leqslant n} \prod\limits_{s=1}^{i-1} \frac{\mu_s - \lambda_\ell}{\mu_s - \lambda_i}, & i > 1. \end{cases} \tag{7.115}$$

注 7.8 不等式 (7.114) 右端可分为三部分: 第一部分为 $(\lambda_i - \lambda_n)\zeta_i^2$, 主要是由 λ_i 与 $\mu_1, \cdots, \mu_{i-1}$ 的分离程度确定; 第二部分为 $\tan^2\theta(\boldsymbol{u}_i, \boldsymbol{v})$, 是由初始向量 $\boldsymbol{v}$ 所确定, 反映了 $\boldsymbol{v}$ 中含有特征向量 $\boldsymbol{u}_i$ 的成分的大小; 第三部分为 $[C_{k-i}(1+2\delta_i)]^{-2}$, 这是关键的一部分, 反映了 Lanczos 方法的收敛性, 当 i 较小且 $0 < \delta_i < 0.1$ 时, 由 Chebyshev 多项式的性质可知, 此时随着 k 的增加, 这个值减少得非常快.

注 7.9 定理 7.21 实质上表明, 随着 Lanczos 迭代次数 k 的增加, $\boldsymbol{T}_k$ 的几个最大特征值 $\mu_1, \cdots, \mu_s$ (s 较小) 将非常快地收敛到 $\boldsymbol{A}$ 的前几个最大的特征值 $\lambda_1, \cdots, \lambda_s$. 特别地, 对于 $i = 1$, 不等式 (7.114) 就变为

$$0 \leqslant \lambda_1 - \mu_1 \leqslant (\lambda_1 - \lambda_n)\tan^2\theta(\boldsymbol{u}_1, \boldsymbol{v})[C_{k-1}(1+2\delta_1)]^{-2},$$

由此更容易看出, 随着 k 的增加, μ_1 与 λ_1 之间的差距迅速地缩小.

注 7.10 对 $-\boldsymbol{A}$ 应用定理 7.21 可知, $\boldsymbol{T}_k$ 的几个最小的特征值 $\mu_{k-s}, \mu_{k-s+1}, \cdots, \mu_k$ 将随着 k 的增加很快地收敛到 $\boldsymbol{A}$ 的几个最小的特征值 $\lambda_{n-s}, \lambda_{n-s+1}, \cdots, \lambda_n$. 特别地, 有

$$0 \leqslant \mu_k - \lambda_n \leqslant (\lambda_1 - \lambda_n)\tan^2\theta(\boldsymbol{u}_n, \boldsymbol{v})[C_{k-1}(1+2\widetilde{\delta}_1)]^{-2},$$

式中: $\widetilde{\delta}_1 = \frac{\lambda_{n-1} - \lambda_n}{\lambda_1 - \lambda_{n-1}}$. 随着 k 的增加, $\boldsymbol{T}_k$ 的最小特征值 μ_k 将很快地收敛到 $\boldsymbol{A}$ 的最小特征值 λ_n.

为了证明定理 7.21, 先将证明中需要的几个基本结果总结在下面的引理中.

引理 7.6 符号如前所述, 则有

(1) $\boldsymbol{T}_k$ 的每个特征值 μ_i 可表示为

$$\mu_i = \max_{\boldsymbol{0}\neq \boldsymbol{u}\in\mathcal{U}_{k-i+1}} \frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}}, \quad i = 1, 2, \cdots, k, \tag{7.116}$$

式中:

$$\mathcal{U}_{k-i+1} = \mathrm{span}\{\boldsymbol{V}_k\boldsymbol{y}_i, \boldsymbol{V}_k\boldsymbol{y}_{i+1}, \cdots, \boldsymbol{V}_k\boldsymbol{y}_k\}. \tag{7.117}$$

(2) 由式 (7.117) 所定义的子空间 $\mathcal{U}_{k-i+1}$ 可表示为

$$\mathcal{U}_{k-i+1} = \{p(\boldsymbol{A})\boldsymbol{v} : p \in \mathcal{P}_{k-1},\ p(\mu_s) = 0,\ s = 1, 2, \cdots, i-1\}. \tag{7.118}$$

证明 (1) 由 $\boldsymbol{T}_k$ 的谱分解易证

$$\mu_i = \max_{\boldsymbol{0}\neq\boldsymbol{y}\in\mathcal{Y}_{k-i+1}} \frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{T}_k\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}}, \tag{7.119}$$

式中: $\mathcal{Y}_{k-i+1} = \mathrm{span}\{\boldsymbol{y}_i, \boldsymbol{y}_{i+1}, \cdots, \boldsymbol{y}_k\}$. 注意到

$$\frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{T}_k\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}} = \frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{y}} = \frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}}, \quad \boldsymbol{u} = \boldsymbol{V}_k\boldsymbol{y},$$

以及 $\boldsymbol{V}_k\mathcal{Y}_{k-i+1} = \mathcal{U}_{k-i+1}$, 由式 (7.119) 可知式 (7.116) 成立.

(2) 由于

$$\mathcal{U}_{k-i+1} \subset \mathcal{R}(\boldsymbol{V}_k) = \mathcal{K}_k = \{p(\boldsymbol{A})\boldsymbol{v} : p \in \mathcal{P}_{k-1}\},$$

$$\mathcal{U}_{k-i+1}^{\perp} = \mathrm{span}\{\boldsymbol{V}_k\boldsymbol{y}_1, \boldsymbol{V}_k\boldsymbol{y}_2, \cdots, \boldsymbol{V}_k\boldsymbol{y}_{i-1}\},$$

故 $\boldsymbol{u} \in \mathcal{U}_{k-i+1}$ 的充分必要条件是存在 $p \in \mathcal{P}_{k-1}$ 满足

$$\boldsymbol{v}^{\mathrm{T}}p(\boldsymbol{A})\boldsymbol{V}_k\boldsymbol{y}_s = 0, \quad s = 1, 2, \cdots, i-1, \tag{7.120}$$

使得 $\boldsymbol{u} = p(\boldsymbol{A})\boldsymbol{v}$.

注意到 $\boldsymbol{v}_{k+1} \perp \mathcal{K}_k$, 从 Lanczos 分解出发, 归纳地可以证明

$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^s\boldsymbol{V}_k = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{T}_k^s, \quad s = 0, 1, \cdots, k-1.$$

从而, 对任意的 $p \in \mathcal{P}_{k-1}$, 有

$$\boldsymbol{v}^{\mathrm{T}}p(\boldsymbol{A})\boldsymbol{V}_k = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{V}_kp(\boldsymbol{T}_k). \tag{7.121}$$

上式两边右乘 $\boldsymbol{y}_s$, 得

$$\boldsymbol{v}^{\mathrm{T}}p(\boldsymbol{A})\boldsymbol{V}_k\boldsymbol{y}_s = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{V}_kp(\mu_s)\boldsymbol{y}_j = \|\boldsymbol{v}\|_2p(\mu_j)\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{y}_s,$$

这里用到了 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{V}_k = \|\boldsymbol{v}\|_2\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{V}_k = \|\boldsymbol{v}\|_2\boldsymbol{e}_1^{\mathrm{T}}$. 再注意到 $\boldsymbol{T}_k$ 是不可约的对称三对角矩阵蕴含着 $\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{y}_s \neq 0$, 便知式 (7.120) 成立的充分必要条件是

$$p(\mu_s) = 0, \quad s = 1, 2, \cdots, i-1.$$

由此可知式 (7.118) 成立. 证毕. □

有了前面的准备工作, 下面证明定理 7.21.

定理 7.21 的证明 注意到式 (7.94), 并且应用推论 7.2, 即知 $\lambda_i - \mu_i \geqslant 0$. 再应用引理 7.6 的结论 (1), 有

$$\begin{aligned} 0 \leqslant \lambda_i - \mu_i &= \lambda_i - \max_{\boldsymbol{0}\neq\boldsymbol{u}\in\mathcal{U}_{k-i+1}} \frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}} \\ &= \min_{\boldsymbol{0}\neq\boldsymbol{u}\in\mathcal{U}_{k-i+1}} \frac{\boldsymbol{u}^{\mathrm{T}}(\lambda_i\boldsymbol{I} - \boldsymbol{A})\boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}}. \end{aligned} \tag{7.122}$$

现在定义 $\widehat{p}(t)=g(t)h(t)$, 其中

$$g(t)=\begin{cases}1, & i=1,\\ \dfrac{(\mu_1-t)\cdots(\mu_{i-1}-t)}{(\mu_1-\lambda_i)\cdots(\mu_{i-1}-\lambda_i)}, & i>1,\end{cases}$$

$$h(t)=C_{k-i}\left(\frac{2t-\lambda_{i+1}-\lambda_n}{\lambda_{i+1}-\lambda_n}\right)\Big/C_{k-i}(1+2\delta_i).$$

容易验证, 这样定义的多项式 $\widehat{p}(t)$ 满足 $\widehat{p}(t)\in\mathcal{P}_{k-1}$, 且

$$\widehat{p}(\lambda_i)=1,\ \ \widehat{p}(\mu_s)=0,\ \ s=1,2,\cdots,i-1.$$

再利用引理 7.6 的结论 (2), 即有 $\widehat{\boldsymbol{u}}=\widehat{p}(\boldsymbol{A})\boldsymbol{v}\in\mathcal{U}_{k-i+1}$.

设 $\boldsymbol{v}=\gamma_1\boldsymbol{u}_1+\cdots+\gamma_n\boldsymbol{u}_n$. 直接计算有

$$\widehat{\boldsymbol{u}}^{\mathrm{T}}\widehat{\boldsymbol{u}}=\boldsymbol{v}^{\mathrm{T}}\widehat{p}(\boldsymbol{A})^2\boldsymbol{v}=\gamma_1^2\,\widehat{p}(\lambda_1)^2+\cdots+\gamma_n^2\,\widehat{p}(\lambda_n)^2\geqslant\gamma_i^2 \tag{7.123}$$

和

$$\begin{aligned}\widehat{\boldsymbol{u}}^{\mathrm{T}}(\lambda_i\boldsymbol{I}-\boldsymbol{A})\widehat{\boldsymbol{u}}&=\boldsymbol{v}^{\mathrm{T}}\widehat{p}(\boldsymbol{A})(\lambda_i\boldsymbol{I}-\boldsymbol{A})\widehat{p}(\boldsymbol{A})\boldsymbol{v}\\&=\sum_{s=1}^{n}\gamma_s^2\,\widehat{p}(\lambda_s)^2(\lambda_i-\lambda_s)\\&=\sum_{s=1}^{i-1}\gamma_s^2\,\widehat{p}(\lambda_s)^2(\lambda_i-\lambda_s)+\sum_{s=i+1}^{n}\gamma_s^2\,\widehat{p}(\lambda_s)^2)(\lambda_i-\lambda_s)\\&\leqslant\sum_{s=i+1}^{n}\gamma_s^2\,\widehat{p}(\lambda_s)^2(\lambda_i-\lambda_s)\\&\leqslant(\lambda_i-\lambda_n)\sum_{s=i+1}^{n}\gamma_s^2\,\widehat{p}(\lambda_s)^2,\end{aligned} \tag{7.124}$$

其中不等式 (7.123) 用到了 $\widehat{p}(\lambda_i)=1$.

将式 (7.122)、式 (7.123) 和式 (7.124) 结合, 有

$$\begin{aligned}0\leqslant\lambda_i-\mu_i&\leqslant\frac{\widehat{\boldsymbol{u}}^{\mathrm{T}}(\lambda_i\boldsymbol{I}-\boldsymbol{A})\widehat{\boldsymbol{u}}}{\widehat{\boldsymbol{u}}^{\mathrm{T}}\widehat{\boldsymbol{u}}^{\mathrm{T}}}\\&\leqslant\frac{\lambda_i-\lambda_n}{\gamma_i^2}\sum_{s=i+1}^{n}\gamma_s^2\,\widehat{p}(\lambda_s)^2\\&\leqslant\frac{\lambda_i-\lambda_n}{\gamma_i^2}\max_{i+1\leqslant s\leqslant n}\widehat{p}(\lambda_s)^2\sum_{s=i+1}^{n}\gamma_s^2.\end{aligned} \tag{7.125}$$

此外, 由 $\widehat{p}(t)$ 的定义, 有

$$\max_{i+1\leqslant s\leqslant n}|\widehat{p}(\lambda_s)|\leqslant\max_{i+1\leqslant s\leqslant n}|g(\lambda_s)|\max_{i+1\leqslant s\leqslant n}|h(\lambda_s)|$$

$$\leqslant \frac{\zeta_i}{C_{k-i}(1+2\delta_i)}. \tag{7.126}$$

将式 (7.126) 代入式 (7.125), 并注意到

$$\frac{1}{\gamma_i^2}\sum_{s=i+1}^{n}\gamma_s^2 \leqslant \frac{1}{\gamma_i^2}\left(\sum_{s=1}^{i-1}\gamma_s^2+\sum_{s=i+1}^{n}\gamma_s^2\right)$$

$$= \frac{\|(\boldsymbol{I}-\boldsymbol{P}_i)\boldsymbol{v}\|_2^2}{\|\boldsymbol{P}_i\boldsymbol{v}\|_2^2} = \tan^2\theta(\boldsymbol{u}_i,\boldsymbol{v}),$$

便知式 (7.114) 成立. 证毕. □

3. 重开始 Lanczos 方法

从理论上来讲, 由 Lanczos 算法产生的矩阵 $\boldsymbol{V}_k$ 的列向量是相互正交的. 然而, 当其在计算机上运行时, 产生的 $\boldsymbol{V}_k$ 很快就失去了其正交性. 正交性的损失会导致 Ritz 值的收敛发生紊乱现象: 会有多个 Ritz 值收敛到同一个特征值, 或者有的特征值迟迟没有逼近它的 Ritz 值出现. 如果并不关心特征值的重数, 又能忍受长时间的等待, 则正交性的损失并不影响所要达到的目的. 但如果对特征值精确的重数很感兴趣, 或者对计算时间要求很高, 则就必须去弥补 $\boldsymbol{V}_k$ 的正交性损失. 目前已有各种各样的重正交化技术来达到这一目标, 最直接的方法就是采用算法 2.14 后所述的完全重正交化技术. 这样做的结果是 $\boldsymbol{V}_k$ 的正交性可以达到机器精度, 但为此付出的代价是:

(1) 存储量从 $O(n)$ 增加到 $O(kn)$.

(2) 计算量从 $O(kn)$ 增加到 $O(k^2n)$.

由于计算机存储空间的限制, k 不能无限制地增加, 这样就会出现 k 已经达到了最大的可能, 但 $\boldsymbol{T}_k$ 中还没有足够的信息来解决希望解决的问题.

解决这一问题的方法主要有两种: 一种是放弃完全重正交化, 而采用有选择性的重正交化技术, 从而减少存储量的要求; 另一种是使用重开始技术.

假定已经得到了一个长度为 k 的 Lanczos 分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{T}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}, \tag{7.127}$$

而希望求的特征值是 $\widehat{m}$ 个 (如 $\widehat{m}$ 个最大特征值, 或者 $\widehat{m}$ 个最小特征值). 现在取一个正整数 m 略大于 $\widehat{m}$, 如 $m=\widehat{m}+2$. k 常取作 $2m$ 或 $5m$ 等.

下面介绍最近由 Stewart 给出的 Krylov–Schur 重开始方法, 其基本步骤如下.

第 1 步, 计算 $\boldsymbol{T}_k$ 的谱分解 $\boldsymbol{T}_k = \boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}}$, 其中

$$\boldsymbol{Y} = [\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_k],\quad \boldsymbol{M} = \mathrm{diag}(\mu_1,\mu_2,\cdots,\mu_k),$$

然后适当地调整特征值 μ_i 的次序, 使得 $\mu_{m+1},\cdots,\mu_k$ 是与要计算的特征值差距较大者, 希望将其剔除. 例如, 要计算 $\boldsymbol{A}$ 的几个最大特征值, 则将 μ_i 排序为

$$\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_m \geqslant \cdots \geqslant \mu_k$$

即可.

第 2 步, 将 $\boldsymbol{T}_k$ 的谱分解代入式 (7.127), 并右乘 $\boldsymbol{Y}$, 得

$$\boldsymbol{A}(\boldsymbol{V}_k\boldsymbol{Y}) = (\boldsymbol{V}_k\boldsymbol{Y})\boldsymbol{M} + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{Y}.$$

比较上式两边的前 m 列, 得

$$\boldsymbol{A}\widetilde{\boldsymbol{V}}_m = \widetilde{\boldsymbol{V}}_m\boldsymbol{M}_1 + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{z}_m^{\mathrm{T}}, \tag{7.128}$$

式中:

$$\widetilde{\boldsymbol{V}}_m = \boldsymbol{V}_k\boldsymbol{Y}(:,1:m),\quad \boldsymbol{z}_m^{\mathrm{T}} = \boldsymbol{Y}(k,1:m),\quad \boldsymbol{M}_1 = \mathrm{diag}(\mu_1,\cdots,\mu_m).$$

第 3 步, 计算一个正交矩阵 $\boldsymbol{Q}\in\mathbb{R}^{m\times m}$, 使得

$$\boldsymbol{Q}\boldsymbol{M}_1\boldsymbol{Q}^{\mathrm{T}} = \widehat{\boldsymbol{T}}_m,\quad \boldsymbol{Q}\boldsymbol{z}_m = \alpha\boldsymbol{e}_m, \tag{7.129}$$

式中: $\widehat{\boldsymbol{T}}_m$ 为对称三对角矩阵; $\alpha = \|\boldsymbol{z}_m\|_2$. 则在式 (7.128) 两边右乘 $\boldsymbol{Q}^{\mathrm{T}}$, 可得如下长度为 m 的 Lanczos 分解:

$$\boldsymbol{A}\widehat{\boldsymbol{V}}_m = \widehat{\boldsymbol{V}}_m\widehat{\boldsymbol{T}}_m + \widehat{\beta}_m\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathrm{T}}, \tag{7.130}$$

式中:

$$\widehat{\boldsymbol{V}}_m = \widetilde{\boldsymbol{V}}_m\boldsymbol{Q}^{\mathrm{T}},\quad \widehat{\beta}_m = \alpha\beta_k,\quad \boldsymbol{v}_{m+1} = \boldsymbol{v}_{k+1}.$$

然后从式 (7.130) 出发, 再应用重正交化的 Lanczos 迭代, 使其扩展为一个长度为 k 的 Lanczos 分解. 如果仍不满足要求, 则可再用上面介绍的方法将其收缩为一个长度为 m 的 Lanczos 分解. 如此反复进行, 最终会得到一个 $\widehat{\boldsymbol{T}}_m$, 它的特征值含有所需要的特征值的很好的近似.

现在的问题是如何实现式 (7.129). 这一计算任务的具体计算过程可从下面的示例中明白. 例如 $m=5$, 且已经将 $\boldsymbol{M}_1$ 和 $\boldsymbol{z}_m$ 约化为如下形式, 即

$$\boldsymbol{M}_1 \to \widetilde{\boldsymbol{M}}_1 = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & & \\ & & & \alpha_4 & \\ & & & & \alpha_5 \end{bmatrix},\quad \boldsymbol{z}_m \to \widetilde{\boldsymbol{z}}_m = \begin{bmatrix} 0 \\ 0 \\ \widetilde{z}_3 \\ z_4 \\ z_5 \end{bmatrix}.$$

下一步, 先确定一个 Givens 变换 $\boldsymbol{G}_3 = \boldsymbol{G}(3,4;c_3,s_3)$, 使得 $\widetilde{\boldsymbol{z}}_m$ 的第 3 个分量为零, 然后计算 $\widehat{\boldsymbol{z}}_m = \boldsymbol{G}_3\widetilde{\boldsymbol{z}}_m$ 和 $\widehat{\boldsymbol{M}}_1 = \boldsymbol{G}_3\widetilde{\boldsymbol{M}}_1\boldsymbol{G}_3^{\mathrm{T}}$, 则 $\widehat{\boldsymbol{z}}_m$ 和 $\widehat{\boldsymbol{M}}_1$ 有如下形式, 即

$$\widehat{\boldsymbol{M}}_1 = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & \gamma & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & \gamma & \beta_3 & \alpha_4 & \\ & & & & \alpha_5 \end{bmatrix},\quad \widehat{\boldsymbol{z}}_m = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \widetilde{z}_4 \\ z_5 \end{bmatrix}.$$

在 $\widehat{\boldsymbol{M}}_1$ 的 (2,4) 和 (4,2) 位置上出现了两个不希望有的可能非零元“γ”. 然后, 就可以采用类似于隐式对称 QR 方法中使用的技术, 计算 (2,3) 平面和 (1,2) 平面的两个 Givens 变换

$$\boldsymbol{G}_2 = \boldsymbol{G}(2,3;c_2,s_2) \quad 和 \quad \boldsymbol{G}_1 = \boldsymbol{G}(1,2;c_1,s_1),$$

将这两个不需要的元素消去, 即

$$\widehat{M}_1 \xrightarrow{G_2} \begin{bmatrix} \alpha_1 & \beta_1 & \gamma & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ \gamma & \beta_2 & \alpha_3 & \beta_3 & \\ & & \beta_3 & \alpha_4 & \\ & & & & \alpha_5 \end{bmatrix} \xrightarrow{G_1} \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \beta_3 & \alpha_4 & \\ & & & & \alpha_5 \end{bmatrix}.$$

再注意到 $\boldsymbol{G}_1\boldsymbol{G}_2\widehat{\boldsymbol{z}}_m = \widehat{\boldsymbol{z}}_m = \widehat{\boldsymbol{z}}_m$, 就完成了一步约化, 即在 $\boldsymbol{z}_m$ 上又多引进了一个零元.

具体的算法可总结如下.

算法 7.14 给定 m 阶对角矩阵 $\boldsymbol{M} = \mathrm{diag}(\mu_1, \cdots, \mu_m)$ 和 m 维向量 $\boldsymbol{z}$. 本算法计算正交矩阵 $\boldsymbol{Q}$, 对称三对角矩阵 $\boldsymbol{T}$ 和实数 α, 使得 $\boldsymbol{Q}\boldsymbol{z} = \tau\boldsymbol{e}_m$, $\boldsymbol{Q}\boldsymbol{M}\boldsymbol{Q}^{\mathrm{T}} = \boldsymbol{T}$.

function $[\boldsymbol{Q}, \boldsymbol{T}, \tau] = \mathbf{TriReduce}(\boldsymbol{M}, \boldsymbol{z})$

$\alpha_i = \mu_i,\ i = 1, 2, \cdots, m;$

$\beta_i = 0,\ i = 1, 2, \cdots, m-1;$

$\boldsymbol{Q} = \boldsymbol{I}_m;$

for $i = 1 : m-1$

确定 $c = \cos\theta$ 和 $s = \sin\theta$, 使得

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} z_i \\ z_{i+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \delta \end{bmatrix};\quad \begin{bmatrix} z_i \\ z_{i+1} \end{bmatrix} := \begin{bmatrix} 0 \\ \delta \end{bmatrix};$$

$$\begin{bmatrix} \alpha_i & \beta_i \\ \beta_i & \alpha_{i+1} \end{bmatrix} := \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \alpha_i & \beta_i \\ \beta_i & \alpha_{i+1} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{\mathrm{T}};$$

$\gamma := -s\beta_{i-1};\ \beta_{i-1} := c\beta_{i-1};$

$\boldsymbol{Q} := \boldsymbol{G}(i, i+1; c, s)\boldsymbol{Q};$

if $i > 1$

for $r = i : -1 : 2$

确定 $c = \cos\theta$ 和 $s = \sin\theta$, 使得

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \gamma \\ \beta_r \end{bmatrix} = \begin{bmatrix} 0 \\ \sigma \end{bmatrix};\ \beta_r := \sigma;$$

$$\begin{bmatrix} \alpha_{r-1} & \beta_{r-1} \\ \beta_{r-1} & \alpha_r \end{bmatrix} := \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \alpha_{r-1} & \beta_{r-1} \\ \beta_{r-1} & \alpha_r \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{\mathrm{T}};$$

if $r > 2$

$\gamma := -s\beta_{r-2};\ \beta_{r-2} := c\beta_{r-2};$

end

$\boldsymbol{Q} := \boldsymbol{G}(r-1, r; c, s)\boldsymbol{Q};$

end

end

end

$\tau = z_m;$

注 7.11 需要注意的是, 这里对于对称三对角矩阵 $\boldsymbol{T}$, 只存储其对角元和次对角元: $\alpha_1, \alpha_2, \cdots, \alpha_m;\ \beta_1, \beta_2, \cdots, \beta_{m-1}$.

算法 7.14 的 MATLAB 程序如下:

```
function [Q,alpha,beta,tau]=trireduce(M,z)
%输入:M为m阶对角矩阵,z为m维向量
%输出:Q为正交矩阵,alpha为对称三对角矩阵T的对角元,
%      beta为对称三对角矩阵T的次对角元,使得Qz=tau*e_m,QMQ'=T
m=size(M,1);
for i=1:m
    alpha(i)=M(i,i);
end
beta=zeros(m-1,1); Q=eye(m);
for i=1:m-1
    delta=sqrt(z(i)^2+z(i+1)^2);
    c=z(i+1)/delta; s=-z(i)/delta;
    z(i+1)=delta; z(i)=0;
    a=[c,s;-s,c]*[alpha(i),beta(i);
    beta(i),alpha(i+1)]*[c,s;-s,c]';
    alpha(i)=a(1,1); beta(i)=a(1,2); alpha(i+1)=a(2,2);
    Q(i:i+1,:)=[c,s;-s,c]*Q(i:i+1,:);
    if i>1
        gamma=-s*beta(i-1); beta(i-1)=c*beta(i-1);
        for r=i:-1:2
            sigma=sqrt(gamma^2+beta(r)^2);
            c=beta(r)/sigma; s=-gamma/sigma;
            a=[c,s;-s,c]*[alpha(r-1),beta(r-1);
            beta(r-1),alpha(r)]*[c,s;-s,c]';
```

```
            alpha(r-1)=a(1,1); beta(r-1)=a(1,2);
            alpha(r)=a(2,2); beta(r)=sigma;
            if r>2
                gamma=-s*beta(r-2); beta(r-2)=c*beta(r-2);
            end
            Q(r-1:r,:)=[c,s;-s,c]*Q(r-1:r,:);
        end
    end
end
tau=z(m);
```

在上面的讨论中, 已经提到了用 Lanczos 过程将长度为 m 的 Lanczos 分解扩展到长度为 k 的 Lanczos 分解, 下面算法给出了一种实现方法.

算法 7.15 给定对称矩阵 $\boldsymbol{A}$ 的长度为 m 的 Lanczos 分解

$$\boldsymbol{A}\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{T}_m + \beta_m\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathrm{T}} = \boldsymbol{V}_{m+1}\widehat{\boldsymbol{T}}_m,$$

本算法将其扩展为一个长度为 k 的 Lanczos 分解.

function $[\boldsymbol{V}_{k+1}, \widehat{\boldsymbol{T}}_k] = \mathbf{ExpandLan}(\boldsymbol{A}, \boldsymbol{V}_{m+1}, \widehat{\boldsymbol{T}}_m, m, k)$
for $i = m+1 : k$
  $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}_i - \beta_{i-1}\boldsymbol{v}_{i-1}$; $\alpha_i = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}_i$;
  $\boldsymbol{u} = \boldsymbol{u} - \alpha_i\boldsymbol{v}_i$;
  $\boldsymbol{u} = \boldsymbol{u} - (\boldsymbol{v}_{i-1}^{\mathrm{T}}\boldsymbol{u})\boldsymbol{v}_{i-1}$; (局部重正交化)
  $\boldsymbol{u} = \boldsymbol{u} - (\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{u})\boldsymbol{v}_i$;
  $\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{V}_i(\boldsymbol{V}_i^{\mathrm{T}}\boldsymbol{u})$; (完全重正交化)
  $\beta_i = \|\boldsymbol{u}\|_2$;
  if $\beta_i = 0$
    stop; (已经得到了 $\boldsymbol{A}$ 的一个不变子空间)
  else
    $\boldsymbol{v}_{i+1} = \boldsymbol{u}/\beta_i$;
    $\boldsymbol{V}_{i+1} = [\boldsymbol{V}_i, \boldsymbol{v}_{i+1}]$;
  end
end

算法 7.15 的 MATLAB 程序如下:

```
function [V,alpha,beta]=expand_lanczos(A,V,alpha,beta,m,k)
%给定对称矩阵A的长度为m的Lanczos分解,
%本算法将其扩展为一个长度为k的Lanczos分解
for i=m+1:k
    u=A*V(:,i)-beta(i-1)*V(:,i-1);
    alpha(i)=u'*V(:,i); u=u-alpha(i)*V(:,i);
```

```
    u=u-(u'*V(:,i-1))*V(:,i-1); %局部重正交化
    u=u-(u'*V(:,i))*V(:,i);
    u=u-V(:,1:i)*(V(:,1:i)'*u); %完全重正交化
    beta(i)=norm(u);
    if (beta(i)==0)
        return;
    else
        V(:,i+1)=u/beta(i);
        V=[V, V(:,i+1)];
    end
end
```

综合上面的讨论, 就得到了如下的隐式重开始完全重正交化 Lanczos 方法.

算法 7.16 (隐式重开始 Lanczos 方法) 给定一个大型稀疏的对称矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. 本算法计算 $\boldsymbol{A}$ 的 $\widehat{m}$ 个两端特征对 ($\widehat{m} \ll n$).

步 1, 初始化

(1) $m = \widehat{m} + 2$; $k = 2m$ (或者 $5m$, 或者取存储所能允许的最大整数).

(2) 选取一个适当的非零向量 $\boldsymbol{v} \in \mathbb{R}^n$ (一般随机选取).

(3) 计算

$$\begin{aligned} &\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|_2; \quad \boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}_1; \quad \alpha_1 = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}_1; \\ &\boldsymbol{u} = \boldsymbol{u} - \alpha_1\boldsymbol{v}_1; \quad \boldsymbol{u} = \boldsymbol{u} - (\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}_1)\boldsymbol{v}_1; \\ &\beta_1 = \|\boldsymbol{u}\|_2; \quad \boldsymbol{v}_2 = \boldsymbol{u}/\beta_1; \quad \boldsymbol{V}_2 = [\boldsymbol{v}_1, \boldsymbol{v}_2]. \end{aligned}$$

(4) $[\boldsymbol{V}_{m+1}, \widehat{\boldsymbol{T}}_m] = \mathbf{ExpandLan}(\boldsymbol{A}, \boldsymbol{V}_2, \widehat{\boldsymbol{T}}_1, 1, m)$.

步 2, 迭代

(1) $[\boldsymbol{V}_{k+1}, \widehat{\boldsymbol{T}}_k] = \mathbf{ExpandLan}(\boldsymbol{A}, \boldsymbol{V}_{m+1}, \widehat{\boldsymbol{T}}_m, m, k)$.

(2) 用对称 QR 方法 (或其他方法) 计算 $\boldsymbol{T}_k$ 的谱分解:

$$\boldsymbol{T}_k = \boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}}, \quad \boldsymbol{M} = \mathrm{diag}(\mu_1, \cdots, \mu_k), \quad \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} = \boldsymbol{I}_k.$$

(3) 若 $\boldsymbol{T}_k$ 的特征值中已含有所求特征值的很好近似, 则输出有关消息, 结束; 否则适当调整 μ_i 的顺序, 使得 $\boldsymbol{T}_k$ 的两端特征值位于前面, 而后 $k-m$ 个特征值为需要剔除的对象.

(4) $[\boldsymbol{Q}, \boldsymbol{T}_m, \alpha] = \mathbf{TriReduce}(\boldsymbol{M}_1, \boldsymbol{z})$, 其中

$$\boldsymbol{M}_1 = \mathrm{diag}(\mu_1, \cdots, \mu_k), \quad \boldsymbol{z}^{\mathrm{T}} = \boldsymbol{Y}(k, 1:m).$$

(5) $\boldsymbol{V}_{m+1} = [\boldsymbol{V}_k\boldsymbol{Y}(1:k, 1; m)\boldsymbol{Q}^{\mathrm{T}}, \boldsymbol{v}_{k+1}]$; $\beta_m = \alpha\beta_k$. 然后, 转步 2.

注 **7.12** 假如希望计算的是 $\boldsymbol{A}$ 之位于某一实数 μ 附近的几个特征对, 则需要应用上一节最后所介绍的位移求逆技术, 即应用算法 7.16 到矩阵 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 上.

算法 7.16 的 MATLAB 程序如下:

```
function [M,V,iter]=irest_Lanczos(A,m)
%给定一个大型稀疏的对称矩阵A,
%本算法计算A的m个两端特征对(m<<n).
m=m+2; k=2*m; V=[ ]; tol=1.0e-30; iter=0;
v=rand(size(A,1),1); v=v/norm(v); V=[V,v];
u=A*v; alpha(1)=u'*v;
u=u-alpha(1)*v; u=u-(u'*v)*v;
beta(1)=norm(u); v=u/beta(1); V=[V,v];
[V,alpha,beta]=expand_lanczos(A,V,alpha,beta,1,m);
while(iter<1000)
    iter=iter+1;
    [V,alpha,beta]=expand_lanczos(A,V,alpha,beta,m,k);
    betak=beta(end);  %将beta的最后一个分量存起来
    vk1=V(:,end);  %将V的最后一列存起来
    T=diag(alpha)+diag(beta(1:end-1),1)+diag(beta(1:end-1),-1);
    [Y,M]=eig(T);  %M的对角元按升序按列
    [M,I]=sort(diag(M),'descend');  %对M的对角元按降序按列
    M=diag(M); Y=Y(:,I);  %第k列到了第1列
    V=V(:,1:k)*Y(:,1:m); z=Y(k,1:m)'; M=M(1:m,1:m);
    if abs(beta(end))*abs(Y(k,1))<tol
       break;
    end
    [V,alpha,beta,theta]=trireduce2(M,z);
    V=V*V'; V=[V,vk1];
    beta(m)=theta*betak;
end
```

例 **7.13** 用算法 7.16 计算矩阵

$$\boldsymbol{A}=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \cdots & n \end{bmatrix} \tag{7.131}$$

的 m 个模最大的特征值. 取 $n=2000$, $m=5$.

解 利用算法 7.16, 编写 MATLAB 脚本文件 ex713.m,

```
%例7.13-ex713.m
clear all
n=2000; m=5; A=zeros(n,n);
for i=1:n
    for j=1:i
        A(i,j)=j; A(j,i)=A(i,j);
    end
end
ev=eig(A); ev=sort(ev,'descend'); lam=ev(1:m);
[M,V,iter]=irest_Lanczos(A,m); iter
mu=diag(M(1:m,1:m))
err1=norm(mu-lam)
err2=norm(A*V(:,1)-M(1,1)*V(:,1))
```

然后在 MATLAB 命令窗口运行之, 即得所要求的结果.

7.6.3 Arnoldi 方法

Arnoldi 方法是最经典的一类 Krylov 子空间法, 它主要用于计算一个大型稀疏非对称矩阵的少数几个特征值和对应的特征向量. 随着重开始技术的不断完善, Arnoldi 方法的应用范围变得越来越广. 本节将阐述这些重开始技术以及与此相结合所产生的重开始 Arnoldi 方法.

1. 经典 Arnoldi 方法

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 已经给定, 希望计算的是 $\boldsymbol{A}$ 在某一指定范围内的少数几个特征值及对应的特征向量, 其中 $\boldsymbol{A}$ 是大型的稀疏矩阵. 将算法 2.12 与 Rayleigh–Ritz 方法相结合就可得到完成这一任务的 Arnoldi 算法.

算法 7.17 (经典 Arnoldi 方法) 给定一个 n 阶实矩阵 $\boldsymbol{A}$. 本算法计算 $\boldsymbol{A}$ 的少数几个特征对.

步 1, 选择初始向量 $\boldsymbol{v}$, 并令 $\boldsymbol{v}_1=\boldsymbol{v}/\|\boldsymbol{v}\|_2$.

步 2, 利用算法 2.12 产生一个长度为 k 的 Arnoldi 分解, 即

$$\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_k\boldsymbol{H}_k+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}.$$

步 3, 计算 $\boldsymbol{H}_k$ 的特征值, 并选择其中若干个满足要求的记为 $\mu_1,\cdots,\mu_l$.

步 4, 计算 μ_i 所对应的特征向量 $\boldsymbol{y}_i$, 并形成 Ritz 向量 $\boldsymbol{u}_i=\boldsymbol{V}_k\boldsymbol{y}_i, i=1,2,\cdots,l$.

步 5, 如果不满足要求, 增加 k, 再返回到步 2, 重复以上各步.

注 7.13 算法 7.17 只涉及矩阵乘向量 $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}$, 因此可充分利用 $\boldsymbol{A}$ 的稀疏性 $\boldsymbol{A}$ 所具有的特殊结构.

注 7.14 设 $\boldsymbol{H}_k\boldsymbol{y}=\mu\boldsymbol{y}$, 在 Arnoldi 分解的两边右乘 $\boldsymbol{y}$, 得

$$\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}=\mu\boldsymbol{V}_k\boldsymbol{y}+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y},$$

于是有

$$\|\boldsymbol{A}(\boldsymbol{V}_k\boldsymbol{y})-\mu(\boldsymbol{V}_k\boldsymbol{y})\|_2=|\beta_k|\cdot|\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y}|. \tag{7.132}$$

由此可知, 可以由 $|\beta_k|\cdot|\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y}|$ 的大小来决定 $(\mu,\boldsymbol{V}_k\boldsymbol{y})$ 是否可以作为 $\boldsymbol{A}$ 的近似特征对.

为了对算法 7.17 的数值特性有个直观的了解, 下面看一个简单的数值例子.

例 7.14 令

$$\boldsymbol{A}=\boldsymbol{P}^{-1}\mathrm{diag}(-24,-23,\cdots,24,25)\boldsymbol{P}\in\mathbb{R}^{50\times 50},$$

式中: $\boldsymbol{P}$ 为随机产生的可逆矩阵.

对于 k 从 1 到 30, 应用算法 7.17 于给定的矩阵 $\boldsymbol{A}$ 上, 并且将计算结果标注在了图 7.5 中, 其中横坐标表示迭代步数 k, 第 k 列的点表示 $\boldsymbol{H}_k$ 的所有实特征值, 最后一列的点表示 $\boldsymbol{A}$ 的精确特征值.

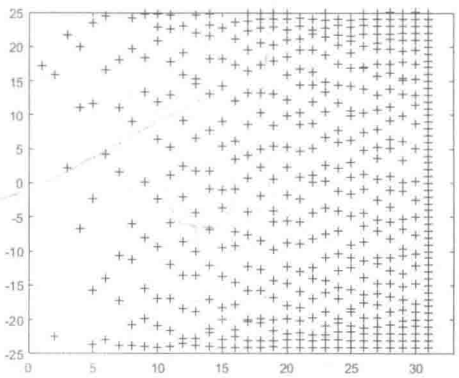

图 7.5 Arnoldi 算法的收敛特性

请注意观察该例所展示出的算法 7.17 的收敛特性: $k=10$ 时, $\boldsymbol{H}_{10}$ 就有两个最大特征值收敛到了 $\boldsymbol{A}$ 的两个最大特征值; 而到了 $k=21$ 时, $\boldsymbol{H}_{21}$ 就有 3 个最小特征值和 2 个最大特征值分别收敛到了 $\boldsymbol{A}$ 的 3 个最小特征值和 2 个最大特征值; 而位于区间内部的有些特征值, 直到 $k=30$ 还没有 Ritz 值收敛到它们.

虽然例 7.14 仅仅是一个可对角化的特殊矩阵, 但其所展示出的算法 7.17 的收敛特性对一般情形也是成立的, 即外围特征值 (exterior eigenvalues) 最先收敛, 而内部特征值收敛特别缓慢. 事实上, 这一结果也可以从理论上解释如下.

为了叙述简洁起见, 假设 $\boldsymbol{A}$ 是可对角化的. 设 $\boldsymbol{A}$ 的特征对为

$$(\lambda_j,\boldsymbol{x}_j),\quad j=1,2,\cdots,n,$$

则 $\boldsymbol{A}$ 可对角化的假定蕴含着 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ 是线性无关的. 现将初始向量 $\boldsymbol{v}$ 按特征向量展开:

$$\boldsymbol{v} = \gamma_1 \boldsymbol{x}_1 + \gamma_2 \boldsymbol{x}_2 + \cdots + \gamma_n \boldsymbol{x}_n.$$

根据定理 2.10 的结论 (5), $\boldsymbol{u} \in \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ 的充分必要条件是存在 $p \in \mathcal{P}_{k-1}$, 使得 $\boldsymbol{u} = p(\boldsymbol{A})\boldsymbol{v}$, 从而有

$$\boldsymbol{u} = \gamma_1 p(\lambda_1)\boldsymbol{x}_1 + \gamma_2 p(\lambda_2)\boldsymbol{x}_2 + \cdots + \gamma_n p(\lambda_n)\boldsymbol{x}_n.$$

由此可知, 欲使 $\boldsymbol{u}$ 是某一特征向量 $\boldsymbol{x}_j$ 的很好近似, 则必须有 $|p(\lambda_j)|$ 相对要比其他的 $|p(\lambda_i)|$ 大得多. 而由复变函数的最大模原理可知, 这只有 λ_j 位于谱集的边缘才能达到. 这也就是 Arnoldi 方法为什么先收敛到外围特征值的缘故.

前面已经列举了 Arnoldi 算法的优点, 但它也有致命的不足之处. 这一算法随着 k 的增加其存储空间需求越来越大, 因此 k 的大小受到了计算机存储量的限制. 这样一来就会出现这样的问题: 希望计算的特征值还没有得到, 而 k 就不能再增大了. 解决这一问题的方法就是现在已经逐渐成熟的重开始技术.

2. 隐式重开始 Arnoldi 方法

假设希望计算 $\boldsymbol{A}$ 的 $\widehat{m}$ 个模最大的特征值 (其他情况可通过注 7.15 所述的"位移求逆技术"归结为这种情形). 再假定已得到一个长度为 k 的 Arnoldi 分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k \boldsymbol{H}_k + \beta_k \boldsymbol{v}_{k+1} \boldsymbol{e}_k^{\mathrm{T}}, \quad \beta_k = h_{k+1,k}. \tag{7.133}$$

由于计算机存储空间的限制, k 已经不能再增加了. 但由 $\boldsymbol{H}_k$ 所提供的 Ritz 值还不满足精度要求. 面临这样的处境, 下一步没有别的选择, 只有重新开始, 即再选择一个新的初始向量 $\widehat{\boldsymbol{v}}$, 重新再来计算新的 Arnoldi 分解. 但是并不想把前面所做的一切全部扔掉, 最好能够对重新选择 $\widehat{\boldsymbol{v}}$ 提供一些有用的信息, 使新选择的 $\widehat{\boldsymbol{v}}$ 比 $\boldsymbol{v}$ 更好一些. 这正是发展重开始技术的最原始想法.

目前存在两种隐式重开始 Arnoldi 方法: 一种是 Sorensen (索仑森) 提出的基于"过滤多项式"技术的隐式重开始方法; 另一种是 Stewart (史都瓦) 提出的基于实 Schur 分解的所谓 Krylov–Schur 重开始方法. 在此着重介绍后者, 其基本步骤有如下 3 步:

第 1 步, 计算 $\boldsymbol{H}_k$ 的实 Schur 分解: $\boldsymbol{H}_k = \boldsymbol{U}_1 \boldsymbol{R} \boldsymbol{U}_1^{\mathrm{T}}$, 其中 $\boldsymbol{U}_1 \in \mathbb{R}^{k\times k}$ 是正交矩阵, $\boldsymbol{R} \in \mathbb{R}^{k\times k}$ 是拟上三角矩阵 (即 $\boldsymbol{R}$ 是分块上三角矩阵, 其对角块是 1×1 或 2×2). 这一步可由著名的 QR 方法实现.

第 2 步, 重排 $\boldsymbol{R}$ 的对角块, 即计算一个正交矩阵 $\boldsymbol{U}_2 \in \mathbb{R}^{k\times k}$, 使得

$$\boldsymbol{U}_2^{\mathrm{T}} \boldsymbol{R} \boldsymbol{U}_2 = \begin{bmatrix} \boldsymbol{R}_1 & * \\ \boldsymbol{O} & \boldsymbol{R}_2 \end{bmatrix},$$

式中: $\boldsymbol{R}_1$ 和 $\boldsymbol{R}_2$ 均为实 Schur 标准形, 而且

$$\lambda(\boldsymbol{R}_1) = \{\mu_1, \cdots, \mu_m\}, \quad \lambda(\boldsymbol{R}_2) = \{\mu_{m+1}, \cdots, \mu_k\}.$$

第 1 步和第 2 步计算完成之后, 有

$$\boldsymbol{H}_k = \boldsymbol{U}_1\boldsymbol{U}_2 \begin{bmatrix} \boldsymbol{R}_1 & * \\ \boldsymbol{O} & \boldsymbol{R}_2 \end{bmatrix} (\boldsymbol{U}_1\boldsymbol{U}_2)^{\mathrm{T}}.$$

将其代入式 (7.133) 并且右乘 $\boldsymbol{U}_1\boldsymbol{U}_2$, 得

$$\boldsymbol{A}(\boldsymbol{V}_k\boldsymbol{U}_1\boldsymbol{U}_2) = \boldsymbol{V}_k\boldsymbol{U}_1\boldsymbol{U}_2 \begin{bmatrix} \boldsymbol{R}_1 & * \\ \boldsymbol{O} & \boldsymbol{R}_2 \end{bmatrix} + \beta_k \boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}(\boldsymbol{U}_1\boldsymbol{U}_2).$$

比较上式两边的前 m 列, 得

$$\boldsymbol{A}\overline{\boldsymbol{V}}_m = \overline{\boldsymbol{V}}_m\boldsymbol{R}_1 + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{z}_m^{\mathrm{T}}, \tag{7.134}$$

式中: $\overline{\boldsymbol{V}}_m$ 为 $\boldsymbol{V}_k\boldsymbol{U}_1\boldsymbol{U}_2$ 的前 m 列构成的矩阵; $\boldsymbol{z}_m^{\mathrm{T}}$ 为 $\boldsymbol{U}_1\boldsymbol{U}_2$ 的最后一行的前 m 个元素构成的向量.

第 3 步, 计算一个正交矩阵 $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$, 使得

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{Q} = \widehat{\boldsymbol{H}}_m, \quad \boldsymbol{z}_m^{\mathrm{T}}\boldsymbol{Q} = \alpha\boldsymbol{e}_m^{\mathrm{T}}, \tag{7.135}$$

式中: $\widehat{\boldsymbol{H}}_m$ 为上 Hessenberg 矩阵.

事实上, 式 (7.135) 可按如下方式实现:

(1) 首先计算一个 Householder 变换 $\boldsymbol{P}_1$, 使得

$$\boldsymbol{z}_m^{\mathrm{T}}\boldsymbol{P}_1 = \alpha\boldsymbol{e}_m^{\mathrm{T}}.$$

(2) 然后依次计算 $m-2$ 个 Householder 变换

$$\boldsymbol{P}_i \in \mathbb{R}^{(m-i-1)\times(m-i+1)}, \quad i = 2, \cdots, m-1,$$

使得

$$\boldsymbol{Q}_{m-1}^{\mathrm{T}} \cdots \boldsymbol{Q}_2^{\mathrm{T}}\boldsymbol{P}_1^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{P}_1\boldsymbol{Q}_2 \cdots \boldsymbol{Q}_{m-1} = \widehat{\boldsymbol{H}}_m$$

为上 Hessenberg 矩阵, 其中 $\boldsymbol{Q}_i = \mathrm{diag}(\boldsymbol{P}_i, \boldsymbol{I}_{i-1})$.

其实, 这一约化过程本质上就是在 7.4.1 节中介绍的约化一个矩阵为上 Hessenberg 形的方法. 只是此处是由矩阵的行向量来确定约化所需的 Housheulder 变换 $\boldsymbol{P}_i$, 而且是从最后一行开始约化的; 而那里是由其列向量来确定, 而且是从第 1 列开始约化的. 显然, 这样得到的 $\boldsymbol{Q}_i$ 满足 $\boldsymbol{e}_m^{\mathrm{T}}(\boldsymbol{Q}_2 \cdots \boldsymbol{Q}_{m-1}) = \boldsymbol{e}_m^{\mathrm{T}}$, 故令 $\boldsymbol{Q} = \boldsymbol{P}_1\boldsymbol{Q}_2 \cdots \boldsymbol{Q}_{m-1}$, 则该正交矩阵就是式 (7.135) 中所需的正交矩阵.

一旦式 (7.135) 已经得到, 则在式 (7.134) 两边右乘 $\boldsymbol{Q}$, 得

$$\boldsymbol{A}\widehat{\boldsymbol{V}}_m = \widehat{\boldsymbol{V}}_m\widehat{\boldsymbol{H}}_m + \widehat{\beta}_m\widehat{\boldsymbol{v}}_{m+1}\boldsymbol{e}_m^{\mathrm{T}}, \tag{7.136}$$

式中: $\widehat{\boldsymbol{V}}_m = \overline{\boldsymbol{V}}_m\boldsymbol{Q}$, $\widehat{\boldsymbol{v}}_{m+1} = \boldsymbol{v}_{k+1}$, $\widehat{\beta}_m = \alpha\beta_k$.

这就是 Stewart 的 Krylov–Schur 重开始方法最终计算得到的结果: 将一个长度为 k 的 Arnoldi 分解收缩为一个长度为 m 的 Arnoldi 分解.

下面考虑 Krylov–Schur 重开始方法的收敛性问题. 为了符号简单, 将式 (7.134) 改写为

$$\boldsymbol{A}\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{R} + \beta_k \boldsymbol{v}_{k+1}\boldsymbol{z}^{\mathrm{T}}. \tag{7.137}$$

设

$$\boldsymbol{z} = (0, \cdots, 0, z_{\ell+1}, \cdots, z_m)^{\mathrm{T}}. \tag{7.138}$$

实际计算时, 要给定一个收敛准则, 可将 $\boldsymbol{z}$ 之小到一定程度的分量均置零. 现假定对此 ℓ, 矩阵 $\boldsymbol{R}$ 有如下的分块表示:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{O} & \boldsymbol{R}_{22} \end{bmatrix} \begin{matrix} \ell \\ m-\ell \end{matrix}.$$
$$\begin{matrix} \ell & m-\ell \end{matrix}$$

事实上, 若 $\boldsymbol{R}$ 的第 ℓ 行正好对应于它的一个 2×2 阶对角块的第 1 行, 则上述分块就不存在, 此时可取 $\ell-1$ 作为 ℓ.

再对 $\boldsymbol{V}_m$ 作相应的分块, 即

$$\boldsymbol{V}_m = \begin{bmatrix} \boldsymbol{V}_{m1} & \boldsymbol{V}_{m2} \end{bmatrix}.$$
$$\begin{matrix} \ell & m-\ell \end{matrix}$$

由于 $\boldsymbol{z}$ 有式 (7.138) 的形状, 故式 (7.135) 中的正交矩阵就可取作如下形式:

$$\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{I}_\ell, \boldsymbol{Q}_1),$$

式中: $\boldsymbol{Q}_1$ 为 $m-\ell$ 阶正交矩阵. 对于这样的 $\boldsymbol{Q}$, 有

$$\widehat{\boldsymbol{V}}_m = \boldsymbol{V}_m\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{V}_{m1} & \boldsymbol{V}_{m2}\boldsymbol{Q}_1 \end{bmatrix},$$

$$\widehat{\boldsymbol{H}}_m = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12}\boldsymbol{Q}_1 \\ \boldsymbol{O} & \boldsymbol{Q}_1^{\mathrm{T}}\boldsymbol{R}_{22}\boldsymbol{Q}_1 \end{bmatrix}.$$

这表明, 在进行这一步变换时, $\boldsymbol{V}_m$ 的前 ℓ 列和 $\boldsymbol{R}$ 的 ℓ 阶顺序主子阵并没有改变. 在 Arnoldi 分解式 (7.136) 中 $\widehat{\boldsymbol{V}}_m$ 的前 ℓ 列 $\boldsymbol{V}_{m1}$ 就是 $\boldsymbol{A}$ 的一个不变子空间的一组标准正交基, $\widehat{\boldsymbol{H}}_m$ 的 ℓ 阶顺序主子阵 $\boldsymbol{R}_{11}$ 的特征值就全部是 $\boldsymbol{A}$ 的特征值, 而且后继部分的计算再也不会涉及 $\boldsymbol{V}_{m1}$ 和 $\boldsymbol{R}_{11}$, 因此就称它们已经被锁定. 随着迭代的进行, 锁定的部分会越来越大, 最终达成收敛.

综合上面的讨论, 可得如下的隐式重开始 Arnoldi 算法.

算法 7.18 (隐式重开始 Arnoldi 方法) 给定 n 阶实矩阵 $\boldsymbol{A}$, 正整数 k 和 $\widehat{m}$ ($\widehat{m} < k$). 本算法计算 $\boldsymbol{A}$ 的 $\widehat{m}$ 个模最大的特征值和对应的特征向量.

步 1, 初始化

(1) 选择初始向量 $\boldsymbol{v}$, 并令 $\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|_2$; $m = \widehat{m}+2$.

(2) 应用算法 2.12 于数据 $(\boldsymbol{A}, \boldsymbol{v}_1, k)$ 上, 产生一个长度为 k 的 Arnoldi 分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}.$$

步 2, 收敛性判定

(1) 计算 $\boldsymbol{H}_k$ 的实 Schur 分解: $\boldsymbol{H}_k = \boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^{\mathrm{T}}$, 其中 $\boldsymbol{U}$ 是正交矩阵, $\boldsymbol{R}$ 是实 Schur 标准形.

(2) 若 $\boldsymbol{H}_k^{\mathrm{T}}$ 的 $\widehat{m}$ 个模最大的特征值已经收敛, 则计算相应 Ritz 向量, 停算; 否则, 进行下一步.

步 3, 重开始 Arnoldi 过程

(1) 利用 Krylov-Schur 重开始方法产生一个长度为 m 的 Arnoldi 分解

$$\boldsymbol{A}\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{H}_m + \beta_m\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathrm{T}}.$$

(2) 利用 Arnoldi 过程将该分解扩展为一个长度为 k 的 Arnoldi 分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}.$$

然后转步 2.

注 7.15 注意到算法 7.18 只能计算 $\boldsymbol{A}$ 的几个模最大的特征值和对应的特征向量. 若要计算 $\boldsymbol{A}$ 在某个数 μ 附近的几个特征值和对应的特征向量, 通常是采取位移求逆的方法, 即将算法 7.18 应用到矩阵 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 上. 一旦 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 的几个模最大的特征值 $\widehat{\lambda}_1, \cdots, \widehat{\lambda}_m$ 已经求得, 则

$$\lambda_i = \mu + 1/\widehat{\lambda}_i, \quad i = 1, 2, \cdots, m$$

就是 $\boldsymbol{A}$ 的最靠近 μ 的几个特征值.

需注意的是, 应用算法 7.18 于 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 上时, 需要计算形如

$$\boldsymbol{y} = (\boldsymbol{A}-\mu\boldsymbol{I})^{-1}\boldsymbol{x}$$

的矩阵乘向量, 即需要求解形如

$$(\boldsymbol{A}-\mu\boldsymbol{I})\boldsymbol{y} = \boldsymbol{x} \tag{7.139}$$

的线性方程组. 由于 Arnoldi 方法对 $\boldsymbol{y}$ 的误差十分敏感, 因此通常采用稀疏列主元的 LU 分解来求解方程组 (7.139). 当然, 在整个迭代过程中只需要作一次分解就够了.

7.6.4 Jacobi–Davidson 方法

7.6.1 节中的 Rayleigh–Ritz 投影方法其实只是子空间迭代方法的一般框架, 它并没有给出子空间的具体选取方法. 7.6.2 节的 Lanzcos 方法 (矩阵 $\boldsymbol{A}$ 对称) 和 7.6.3 节的 Arnoldi 方法 (矩阵 $\boldsymbol{A}$ 非对称) 其实都是将投影子空间取为 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ 的

Rayleigh–Ritz 投影法. 本节将给出投影子空间的另一个取法, 相应的方法称为 Jacobi–Davidson 方法.

设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, $\mathcal{K}=\operatorname{span}\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m\}\subset\mathbb{C}^n$, $\{\boldsymbol{v}_i\}_{i=1}^m$ 为 $\mathcal{K}$ 中的标准正交基. 记 $\boldsymbol{V}_m=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m]$. 选择 $\mu\in\mathbb{C}$ 和 $\boldsymbol{x}\in\mathcal{K}$ 使得

$$(\boldsymbol{A}\boldsymbol{x}-\mu\boldsymbol{x})\perp\mathcal{K}.$$

上述条件可以等价地表述为

$$(\boldsymbol{A}\boldsymbol{x}-\mu\boldsymbol{x},\boldsymbol{v})=0,\quad\forall\,\boldsymbol{v}\in\mathcal{K}.$$

注意到 $\boldsymbol{x}$ 可表示为 $\boldsymbol{x}=\boldsymbol{V}_m\boldsymbol{y}$, 其中 $\boldsymbol{y}\in\mathbb{C}^m$, 故上述条件又可以等价地表示为

$$\boldsymbol{V}_m^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{V}_m\boldsymbol{y}-\mu\boldsymbol{V}_m\boldsymbol{y})=\boldsymbol{0},$$

即

$$\boldsymbol{A}_m\boldsymbol{y}=\mu\boldsymbol{y},$$

其中 $\boldsymbol{A}_m=\boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m$ 正好是 $\boldsymbol{A}$ 关于 $\boldsymbol{V}_m$ 的 Rayleigh 商. 设 $(\mu,\boldsymbol{y})\,(\|\boldsymbol{y}\|_2=1)$ 是 $\boldsymbol{A}_m$ 的特征对, $(\mu,\boldsymbol{V}_m\boldsymbol{y})$ 是 $\boldsymbol{A}$ 的 Ritz 对, 它是 $\boldsymbol{A}$ 的某个特征对的近似. 若记 $\boldsymbol{u}=\boldsymbol{V}_m\boldsymbol{y}$, 则显然有 $\|\boldsymbol{u}\|_2=1$. 现取向量 $\boldsymbol{z}\in\mathbb{C}^n$ 满足 $\boldsymbol{z}^{\mathrm{H}}\boldsymbol{u}=0$, 且

$$\boldsymbol{A}(\boldsymbol{u}+\boldsymbol{z})=\lambda(\boldsymbol{u}+\boldsymbol{z})\Longleftrightarrow(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{z}=-(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{u}.\tag{7.140}$$

上式表明 $\boldsymbol{A}$ 的近似特征向量 $\boldsymbol{u}$ 加上一个和它正交的向量 $\boldsymbol{z}$ 后变为 $\boldsymbol{A}$ 的特征向量. 这启发把上述方程限制在与 $\boldsymbol{u}$ 正交的子空间上, 得

$$(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A}-\lambda\boldsymbol{I})(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})\boldsymbol{z}=-(\boldsymbol{A}-\mu\boldsymbol{I})\boldsymbol{u}.\tag{7.141}$$

事实上, 考虑到

$$\boldsymbol{u}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{u}=(\boldsymbol{V}_m\boldsymbol{y})^{\mathrm{H}}\boldsymbol{A}(\boldsymbol{V}_m\boldsymbol{y})=\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}_m\boldsymbol{y}=\mu,\tag{7.142}$$

$\boldsymbol{u}$ 与 $\boldsymbol{z}$ 正交以及式 (7.140), 有

$$\begin{aligned}&(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A}-\lambda\boldsymbol{I})(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})\boldsymbol{z}\\&=(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{z}=-(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{u}\\&=-(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{u}+(\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{u}-\lambda\boldsymbol{u})=-(\boldsymbol{A}-\mu\boldsymbol{I})\boldsymbol{u}.\end{aligned}$$

由于式 (7.141) 中的 λ 未知, 用 μ 近似它, 得

$$\widetilde{\boldsymbol{A}}\boldsymbol{z}=-\widetilde{\boldsymbol{r}},\quad\boldsymbol{z}^{\mathrm{H}}\boldsymbol{u}=0,\tag{7.143}$$

式中:

$$\widetilde{\boldsymbol{A}}=(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A}-\mu\boldsymbol{I})(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})$$

为 $\boldsymbol{A}$ 在和 $\boldsymbol{u}$ 正交的子空间上的限制; $\widetilde{\boldsymbol{r}}=(\boldsymbol{A}-\mu\boldsymbol{I})\boldsymbol{u}$ 表示 Ritz 对 $(\mu,\boldsymbol{u})$ 的残差. 由式 (7.142) 可知

$$\boldsymbol{u}^{\mathrm{H}}\widetilde{\boldsymbol{r}}=\boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A}-\mu\boldsymbol{I})\boldsymbol{u}=0,$$

即 $\widetilde{\boldsymbol{r}} \in \mathrm{span}\{\boldsymbol{u}\}^{\perp} = \mathcal{R}(\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})$. 式 (7.143) 也可表示为

$$(\boldsymbol{A} - \mu\boldsymbol{I})\boldsymbol{z} = -\widetilde{\boldsymbol{r}} + \alpha\boldsymbol{u}, \quad \alpha = \boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A} - \mu\boldsymbol{I})\boldsymbol{z}, \quad \boldsymbol{z}^{\mathrm{H}}\boldsymbol{u} = 0.$$

设 μ 不是 $\boldsymbol{A}$ 的特征值, 则有

$$\boldsymbol{z} = (\boldsymbol{A} - \mu\boldsymbol{I})^{-1}(-\widetilde{\boldsymbol{r}} + \alpha\boldsymbol{u}). \tag{7.144}$$

根据求出的 $\boldsymbol{z}$, 得

$$\boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A} - \mu\boldsymbol{I})\boldsymbol{z} = \boldsymbol{u}^{\mathrm{H}}(-\widetilde{\boldsymbol{r}} + \alpha\boldsymbol{u}) = \alpha.$$

注意到 $\boldsymbol{u}$ 与 $\boldsymbol{z}$ 正交, 则由式 (7.144), 有

$$\alpha = \frac{\boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A} - \mu\boldsymbol{I})^{-1}\widetilde{\boldsymbol{r}}}{\boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A} - \mu\boldsymbol{I})^{-1}\boldsymbol{u}}.$$

这表明, 当 μ 不是 $\boldsymbol{A}$ 的特征值时, 方程组 (7.143) 有唯一解.

上面讨论了如何从一个子空间 $\mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m\}$ 出发求 Ritz 向量 $\boldsymbol{u}$, 并把它进行校正写为 $\boldsymbol{u} + \boldsymbol{z}$, 这里 $\boldsymbol{u}$ 在该子空间中. 现在利用 $\boldsymbol{z}$ 构造一个“更大”的子空间, 取

$$\boldsymbol{v}_{m+1} = \boldsymbol{z} - \sum_{i=1}^{m}(\boldsymbol{z}^{\mathrm{H}}\boldsymbol{v}_i)\boldsymbol{v}_i$$

的单位化向量. 下面给出 Jacobi–Davidson 方法的详细算法步骤.

算法 7.19 (Jacobi–Davidson 方法) 给定 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\boldsymbol{v}_1 \in \mathbb{C}^n$ 满足 $\|\boldsymbol{v}_1\|_2 = 1$.

For $m = 1, 2, \cdots$ **Do**

(1) 根据 $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m]$ 产生 $\boldsymbol{A}_m = \boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m \in \mathbb{C}^{m\times m}$.

(2) 求 $\boldsymbol{A}_m$ 的某个特征对 $(\mu, \boldsymbol{y})$. 令 $\boldsymbol{u} = \boldsymbol{V}_m\boldsymbol{y}$.

(3) 计算 $\widetilde{\boldsymbol{r}} = \boldsymbol{A}\boldsymbol{u} - \mu\boldsymbol{u}$. 若 $\|\widetilde{\boldsymbol{r}}\|_2 \leqslant \varepsilon$, 停算.

(4) 解方程组 (7.143) 得到 $\boldsymbol{z}$.

(5) 取 $\widetilde{\boldsymbol{v}}_{m+1} := \boldsymbol{z} - \sum_{i=1}^{m}(\boldsymbol{z}^{\mathrm{H}}\boldsymbol{v}_i)\boldsymbol{v}_i$, $\boldsymbol{v}_{m+1} = \widetilde{\boldsymbol{v}}_{m+1}/\|\widetilde{\boldsymbol{v}}_{m+1}\|_2$.

Enddo

注 7.16 注意到

$$\boldsymbol{A}_{m+1} = \boldsymbol{V}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_{m+1} = \begin{bmatrix} \boldsymbol{V}_m^{\mathrm{H}} \\ \boldsymbol{v}_{m+1}^{\mathrm{H}} \end{bmatrix} \boldsymbol{A}\big[\boldsymbol{V}_m, \ \boldsymbol{v}_{m+1}\big]$$

$$= \begin{bmatrix} \boldsymbol{A}_m & \boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{v}_{m+1} \\ \boldsymbol{v}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m & \boldsymbol{v}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{v}_{m+1} \end{bmatrix},$$

故算法 7.19 在第 $m+1$ 次循环时, 只需计算 $\boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{v}_{m+1}$, $\boldsymbol{v}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m$ 和 $\boldsymbol{v}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{v}_{m+1}$ 便产生了 $\boldsymbol{A}_{m+1}$.

注 7.17 算法 7.19 的计算量主要集中在第 4 步解方程组 (7.143). 可以使用子空间迭代法 GMRES 方法或预处理 GMRES 方法来求解.

算法 7.19 的 MATLAB 程序如下:

```
%Jacobi-Davidson方法程序-jacobidavidson.m
function [mu,u,Vm]=jacobidavidson(A,v,tol,max_it)
%用Jacobi-Davidson求矩阵A的模最大的特征值及相应的特征向量
%输入:A为n阶实方阵,v为初始向量,tol为容许误差,max_it为子空间的最大维数
%输出:mu返回A的模最大特征值,u为相应的特征向量,Vm为子空间基矩阵
if nargin<4, max_it=ceil(size(A,1)/2); end
if nargin<3, tol=1e-6; end
n=size(A,1); m=0; Vm=[ ]; I=eye(n);
x0=rand(n,1); v=v/norm(v);
Vm=[Vm,v]; Am=Vm'*A*Vm;
while(m<max_it)
    m=m+1;
    [Y,D]=eig(Am); [s,t]=max(abs(diag(D)));
    mu=D(t,t); u=Vm*Y(:,t); r=A*u-mu*u;
    if (norm(r)<=tol), break; end
    At=(I-u*u')*(A-mu*I)*(I-u*u');
    [z]=gmres(At,-r);  %系统自带的GMRES函数
    v=z;  %正交化
    for i=1:m
        v=v-(z'*Vm(:,i))*Vm(:,i);
    end
    v=v/norm(v);
    Am=[Am,Vm'*A*v;v'*A*Vm,v'*A*v];
    Vm=[Vm,v];
end
```

例 7.15 用算法 7.19 计算矩阵

$$\boldsymbol{A}=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \cdots & n \end{bmatrix}$$

的模最大特征值. 取 $n=1000$.

解 编写 MATLAB 脚本程序 ex715.m, 然后命令窗口运行该程序可得到相应的结果. 当投影子空间的维数增加到 $m=4$ 时, 迭代已满足终止条件, 得到近似特征值 $\widetilde{\lambda}$ 和

对应的特征向量 $\boldsymbol{u}$ 满足 $\|\boldsymbol{A}\boldsymbol{u}-\widetilde{\lambda}\boldsymbol{u}\|_2=5.3620\times10^{-9}$.

假设希望计算矩阵 $\boldsymbol{A}$ 的某个与 α 最接近的特征值, 可以考虑求 $(\boldsymbol{A}-\alpha\boldsymbol{I})^{-1}$ 模最大的特征值 μ, 那么 $\widetilde{\lambda}=\dfrac{1}{\mu}+\alpha$ 即为 $\boldsymbol{A}$ 的某个与 α 最接近的特征值. 也就是说, 只需用 $\boldsymbol{B}=(\boldsymbol{A}-\alpha\boldsymbol{I})^{-1}$ 去调用算法 7.19 的程序 jacobidavidson.m 即可. 例如, 要计算例 7.15 中的矩阵与 $\alpha=18$ 最接近的特征值和相应的特征向量, 可以编写如下 MATLAB 脚本程序:

```
tic; n=1000; A=zeros(n,n);
for i=1:n,
    for j=1:i
        A(i,j)=j; A(j,i)=A(i,j);
    end
end
v=rand(n,1); I=eye(n);
alpha=18; B=(A-alpha*I)\I;
[mu,u,Vm]=jacobidavidson(B,v);
lambda=alpha+1/mu,
err1=norm(A*u-lambda*u),d=eig(A);
err2=norm(lambda-d(925)),toc
```

执行上述程序语句, 得到 $\lambda=17.8762$, $\|\boldsymbol{A}\boldsymbol{u}-\lambda\boldsymbol{u}\|_2=8.1153\times10^{-9}$.

习题 7

7.1 试证明: 定理 7.6 中的集合 $\bigcup\limits_{i\neq j}\Omega_{ij}$ 是定理 7.5 中的集合 $\bigcup\limits_{i=1}^{n}G_i$ 的子集.

7.3 设 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$, $\boldsymbol{B}\in\mathbb{R}^{n\times m}$. 试证明: 矩阵 $\boldsymbol{AB}$ 与 $\boldsymbol{BA}$ 具有相同的非零特征值.

7.3 设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, $\xi\pm\mathrm{i}\eta$ 为 $\boldsymbol{A}$ 的一对共轭特征值, 对应的特征向量为 $\boldsymbol{x}\pm\mathrm{i}\boldsymbol{y}$, 其中 $\xi,\eta\in\mathbb{R}$, $\eta\neq0$, $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^n$. 试证明:

(1) $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 线性无关;

(2) 存在正交矩阵 $\boldsymbol{Q}\in\mathbb{R}^{n\times n}$, 使满足

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{P} & \boldsymbol{S}\\ \boldsymbol{O} & \boldsymbol{T}\end{bmatrix},$$

其中 $\boldsymbol{P}\in\mathbb{R}^{2\times2}$.

7.4 矩阵的任一特征值及其相应的特征向量称为矩阵的一个特征对. 设 $(\lambda,\boldsymbol{v})$ 是矩阵 $\boldsymbol{A}$ 的特征对, 试证明:

(1) 对于任意的常数 α, 则 $(\lambda-\alpha,\boldsymbol{v})$ 是矩阵 $\boldsymbol{A}-\alpha\boldsymbol{I}$ 的特征对;

(2) 若 $\lambda \neq 0$, 则 $(1/\lambda, \boldsymbol{v})$ 是矩阵 $\boldsymbol{A}^{-1}$ 的特征对;

(3) 若 $\alpha \neq \lambda$, 则 $(1/(\lambda-\alpha), \boldsymbol{v})$ 是矩阵 $(\boldsymbol{A}-\alpha\boldsymbol{I})^{-1}$ 的特征对.

7.5 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$. 对于给定的非零向量 $\boldsymbol{x} \in \mathbb{C}^n$, 定义

$$\rho(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{H}}\boldsymbol{x}},$$

称之为 $\boldsymbol{x}$ 对 $\boldsymbol{A}$ 的 Rayleigh 商. 试证明: 对任意的非零向量 $\boldsymbol{x} \in \mathbb{C}^n$, 有

$$\|\boldsymbol{A}\boldsymbol{x} - \rho(\boldsymbol{x})\boldsymbol{x}\|_2 = \inf_{\mu\in\mathbb{C}} \|\boldsymbol{A}\boldsymbol{x} - \mu\boldsymbol{x}\|_2.$$

7.6 设 $\boldsymbol{A}$ 为奇异的不可约上 Hessenberg 矩阵. 试证明: 进行一次基本的 QR 迭代后, $\boldsymbol{A}$ 的零特征值将出现.

7.7 设 $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$, $\boldsymbol{C} = \boldsymbol{A} + \mathrm{i}\boldsymbol{B}$, $\mathrm{i} = \sqrt{-1}$, $\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{A} \end{bmatrix}$.

(1) 证明 $\boldsymbol{C}$ 为 Hermite 矩阵的充分必要条件是 $\boldsymbol{M}$ 为实对称矩阵;

(2) 指出 $\boldsymbol{C}$ 的特征值和特征向量与 $\boldsymbol{M}$ 的特征值和特征向量的关系.

7.8 设 $\boldsymbol{A}$ 为实对称矩阵, $\{\boldsymbol{A}_k\}$ 是按算法 7.3 (Jacobi 方法) 产生的矩阵序列, 记

$$S(\boldsymbol{A}_k) = \sum_{\substack{i,j=1 \\ i\neq j}}^{n} \left[a_{ij}^{(k)}\right]^2,$$

证明:

$$\lim_{k\to\infty} S(\boldsymbol{A}_k) = 0.$$

7.9 设

$$\boldsymbol{A} = \begin{bmatrix} (\boldsymbol{A}_{11})_{3\times 3} & \boldsymbol{O}_{3\times 2} \\ \boldsymbol{O}_{2\times 3} & (\boldsymbol{A}_{22})_{2\times 2} \end{bmatrix}.$$

又设 λ_i 为 $\boldsymbol{A}_{11}$ 的特征值, λ_j 为 $\boldsymbol{A}_{22}$ 的特征值, $\boldsymbol{x}_i = (\alpha_1, \alpha_2, \alpha_3)^{\mathrm{T}}$ 是 $\boldsymbol{A}_{11}$ 对应于 λ_i 的特征向量, $\boldsymbol{y}_j = (\beta_1, \beta_2)^{\mathrm{T}}$ 是 $\boldsymbol{A}_{22}$ 对应于 λ_j 的特征向量. 求证:

(1) λ_i 和 λ_j 是 $\boldsymbol{A}$ 的特征值;

(2) $\bar{\boldsymbol{x}}_i = (\alpha_1, \alpha_2, \alpha_3, 0, 0)^{\mathrm{T}}$ 是 $\boldsymbol{A}$ 对应于 λ_i 的特征向量, $\bar{\boldsymbol{y}}_j = (0, 0, 0, \beta_1, \beta_2)^{\mathrm{T}}$ 是 $\boldsymbol{A}$ 对应于 λ_j 的特征向量.

7.10 设 $\boldsymbol{A}$ 为不可约对称三对角矩阵, 对其实行 QR 方法时, 每个迭代矩阵 $\boldsymbol{A}_k = \boldsymbol{Q}_k\boldsymbol{R}_k$, $k = 0, 1, \cdots$, 其中 $\boldsymbol{A}_0 = \boldsymbol{A}$. 试证明:

(1) $\boldsymbol{R}_k = (r_{ij})$ 为上三角矩阵, 且只有 r_{ij}, $i \leqslant j \leqslant i+2$, $i, j = 1, 2, \cdots, n$ 可能不为零;

(2) $\boldsymbol{Q}_k$ 为不可约上 Hessenberg 矩阵;

(3) $\boldsymbol{A}_{k+1}$ 仍为不可约对称三对角矩阵.

7.11 设三对角矩阵

$$\boldsymbol{A}=\begin{bmatrix} a_1 & c_1 & & & \\ b_1 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & a_{n-1} & c_{n-1} \\ & & & b_{n-1} & a_n \end{bmatrix} \in \mathbb{R}^{n\times n},$$

其中 $b_ic_i>0$, $i=1,2,\cdots,n-1$. 试证明: 存在非奇异的对角矩阵 $\boldsymbol{D}$, 使得 $\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{D}$ 为实对称三对角矩阵.

7.12 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 有实特征值且满足 $\lambda_1>\lambda_2\geqslant\cdots\geqslant\lambda_{n-1}>\lambda_n$. 现应用幂法于矩阵 $\boldsymbol{A}-\mu\boldsymbol{I}$, 试证明: 选择 $\mu=\frac{1}{2}(\lambda_2+\lambda_n)$ 时, 所产生的向量序列收敛到属于 λ_1 的特征向量的速度最快.

7.13 设 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$, $\boldsymbol{x}\in\mathbb{C}^n$, 且 $\boldsymbol{X}=[\boldsymbol{x},\boldsymbol{A}\boldsymbol{x},\cdots,\boldsymbol{A}^{n-1}\boldsymbol{x}]$. 试证明: 若 $\boldsymbol{X}$ 非奇异, 则 $\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$ 为上 Hessenberg 矩阵.

7.14 设 $\boldsymbol{A}$ 是上 Hessenberg 矩阵, 其列主元 LU 分解为 $\boldsymbol{P}\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}$, 这里 $\boldsymbol{P}$ 是置换矩阵, $\boldsymbol{L}$ 为单位下三角矩阵, $\boldsymbol{U}$ 为上三角矩阵. 试证明: $\widetilde{\boldsymbol{A}}=\boldsymbol{U}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{L})$ 仍为上 Hessenberg 矩阵, 并且相似于 $\boldsymbol{A}$.

7.15 若给定 $\boldsymbol{A}_0:=\boldsymbol{A}$, 并由

$$\boldsymbol{A}_k-\mu_k\boldsymbol{I}=\boldsymbol{Q}_k\boldsymbol{R}_k,\quad \boldsymbol{A}_{k+1}=\boldsymbol{R}_k\boldsymbol{Q}_k+\mu_k\boldsymbol{I}$$

产生矩阵序列 $\{\boldsymbol{A}_k\}$. 试证明:

$$(\boldsymbol{Q}_0\boldsymbol{Q}_1\cdots\boldsymbol{Q}_l)(\boldsymbol{R}_l\cdots\boldsymbol{R}_1\boldsymbol{R}_0)=(\boldsymbol{A}-\mu_0\boldsymbol{I})(\boldsymbol{A}-\mu_1\boldsymbol{I})\cdots(\boldsymbol{A}-\mu_l\boldsymbol{I}).$$

参考文献

[1] 孙继广. 矩阵扰动分析. 北京: 科学出版社, 1987.

[2] 蔡大用. 数值代数. 北京: 清华大学出版社, 1987.

[3] 胡家赣. 线性代数方程组的迭代解法. 北京: 科学出版社, 1991.

[4] 何旭初, 孙文瑜. 广义逆矩阵引论. 南京: 江苏科学技术出版社, 1991.

[5] 王国荣. 矩阵及算子广义逆. 北京: 科学出版社, 1994.

[6] 徐树方. 矩阵计算的理论与方法. 北京: 北京大学出版社, 1995.

[7] 曹志浩. 数值线性代数. 上海: 复旦大学出版社, 1996.

[8] 戴华. 矩阵论. 北京: 科学出版社, 2001.

[9] 曹志浩. 变分迭代法. 北京: 科学出版社, 2005.

[10] 张凯院, 徐仲. 数值代数. 2 版. 北京: 科学出版社, 2006.

[11] 魏木生. 广义最小二乘问题的理论与计算. 北京: 科学出版社, 2006.

[12] 蒋尔雄. 矩阵计算. 北京: 科学出版社, 2008.

[13] 李大明. 数值线性代数. 北京: 清华大学出版社, 2010.

[14] 徐树方. 控制论中的矩阵计算. 北京: 高等教育出版社, 2011.

[15] 徐树方, 高立, 张平文. 数值线性代数. 2 版. 北京: 北京大学出版社, 2013.

[16] 徐仲, 陆全, 张凯院, 等. H-类矩阵的理论与应用. 北京: 科学出版社, 2013.

[17] 徐树方, 钱江. 矩阵计算六讲. 北京: 高等教育出版社, 2014.

[18] 谷同祥, 安恒斌, 刘兴平, 等. 迭代方法和预处理技术 (上册). 北京: 科学出版社, 2015.

[19] 谷同祥, 徐小文, 刘兴平, 等. 迭代方法和预处理技术 (下册). 北京: 科学出版社, 2016.

[20] Golub G H, Van Loan C F. Matrix Computations. 3 ed (中译本). 袁亚湘, 等译. 北京: 人民邮电出版社, 2011.

[21] Bai Z Z, Golub G H, Ng M K. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl., 2003, 24(3): 603–626.

[22] Bai Z Z, Golub G H, Pan J Y. Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math., 2004, 98(1): 1–32.

[23] Niethammer W, Pillis J de, Varga R S. Convergence of block iterative methods applied to sparse least-squares problems. Linear Algebra Appl., 1984, 58(1): 327–341.

[24] Cline A K, Moler C B, Stewart G W, Wilkinson J H. An estimate for the condition number of a matrix. SIAM J. Numer. Anal., 1979, 16(2): 368–375.

[25] Freund R W, Golub G H, Nachtigal N M. Iterative solution of linear systems. Acta Numer., 1992, 1(1): 57–100.

[26] Paige C C, Saunders M A. Solution of sparse indefinite systems of linear equations. SIAM J. Numer. Anal., 1975, 12(4): 617–629.

[27] Varga R S, Gillis J. Matrix Iterative Analysis. 2 ed. Springer, 2000.

[28] Young D M. Iterative Solution of Large Linear Systems. Dover Publications, 2003.

[29] Saad Y. Iterative Methods for Sparse Linear Systems. 2 ed. SIAM, 2003.