

中图分类号 O241.6  
学科分类号 070102

论文编号 1028708 10-0020

# 硕士学位论文

## 求解大型对称特征值问题的改进块 Jacobi-Davidson 方法

研究生姓名	康 艳 艳
学科、专业	计算数学
研究 方 向	数值代数
指 导 教 师	戴华 教授

南京航空航天大学

研究生院 理学院

二〇一〇年二月

Nanjing University of Aeronautics and Astronautics

The Graduate School

College of Science

**The improved block Jacobi-Davidson method for  
solving large symmetric eigenvalue problems**

A Thesis in

Computational Mathematics

by

Kang Yanyan

Advised by

Professor Dai Hua

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

February, 2010

## 承诺书

本人声明所呈交的硕士学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京航空航天大学或其他教育机构的学位或证书而使用过的材料。

本人授权南京航空航天大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本承诺书）

作者签名：\_\_\_\_\_

日 期：\_\_\_\_\_

## 摘 要

块 Jacobi-Davidson 方法是计算对称矩阵重或密集特征值的有效方法之一。该方法可同时计算若干个极端特征对,但迭代过程中所产生的一些已收敛的 Ritz 对仍然会参加后续的迭代运算,这降低了该方法的总体收敛速度,其次,块 Jacobi-Davidson 方法在计算内部特征值时计算量较大。

为了提高块 Jacobi-Davidson 方法的整体收敛速度,本文应用动态收缩技术,提出了动态收缩的块 Jacobi-Davidson 方法;为了计算大型对称矩阵的内部特征对,本文将调和 Rayleigh-Ritz 方法与块 Jacobi-Davidson 方法结合,提出了调和块 Jacobi-Davidson 方法,并将动态收缩技术应用于调和块 Jacobi-Davidson 方法,给出了动态收缩的调和块 Jacobi-Davidson 方法。

校正方程的求解是块 Jacobi-Davidson 类方法的关键,本文通过使用校正方程的等价增广形式将校正方程转化为经典的 saddle point 问题,并对其使用适当的预处理技术。

数值结果表明,动态收缩的块 Jacobi-Davidson 方法优于块 Jacobi-Davidson 方法,动态收缩的调和块 Jacobi-Davidson 方法能有效计算大型对称矩阵的内部重或密集特征值。

**关键词:** 对称矩阵; 特征值问题; 块 Jacobi-Davidson 方法; 收缩技术; 调和方法

## Abstract

Block Jacobi-Davidson method (BJD) is very efficient for computing the multiple or clustered eigenpairs of the symmetric eigenproblems. Several eigenpairs can be computed simultaneously by the method. However, some convergent Ritz pairs generated by the iteration will still participate in the follow-up iteration. The overall convergence rate of the method is reduced. In addition, its computational cost is very expensive when the method is applied to find interior eigenvalues.

In order to improve the overall convergence of the block Jacobi-Davidson method, the dynamic deflation technique is used to the method and the dynamic deflated block Jacobi-Davidson method (DBJD) is presented. In order to compute interior eigenvalues, combining the harmonic Rayleigh-Ritz procedure with the block Jacobi-Davidson method, the harmonic block Jacobi-Davidson method (HBJD) is proposed. Finally, applying the dynamic deflation technique to the harmonic block Jacobi-Davidson method, the dynamic deflated harmonic block Jacobi-Davidson method (DHBJD) is given.

Efficiently solving the correction equation is crucial for the block Jacobi-Davidson method. The correction equation in BJD is transformed into the classic saddle point system. The preconditioning techniques for the system are analyzed.

Numerical experiments show that the dynamic deflated block Jacobi-Davidson method is more efficient than the block Jacobi-Davidson method, moreover, the HBJD and the DHBJD is very efficient for computing the multiple or clustered interior eigenpairs of the symmetric eigenproblems.

**Key words:** symmetric matrix; eigenvalue problem; block Jacobi-Davidson method; deflation; harmonic method

# 目 录

第一章 绪论 .....	1
1.1 问题背景 .....	1
1.2 本文研究工作与内容安排 .....	3
1.3 符号约定 .....	4
第二章 预备知识 .....	5
2.1 投影方法 .....	5
2.1.1 正交投影方法 .....	5
2.1.2 斜投影方法 .....	7
2.1.3 精化投影方法 .....	8
2.2 JACOBI-DAVIDSON 方法 .....	9
2.2.1 JACOBI-DAVIDSON 方法的算法描述 .....	10
2.2.2 校正方程的求解与收敛性分析 .....	11
第三章 动态收缩的块 JACOBI-DAVIDSON 方法 .....	13
3.1 块 JACOBI-DAVIDSON 方法 .....	13
3.2 动态收缩的块 JACOBI-DAVIDSON 方法 .....	15
第四章 调和块 JACOBI-DAVIDSON 方法 .....	18
4.1 调和 RAYLEIGH-RITZ 方法 .....	18
4.2 调和块 JACOBI-DAVIDSON 方法 .....	20
第五章 算法的实现 .....	23
5.1 校正方程的求解 .....	23
5.1.1 等价增广形式的预处理解法 .....	23
5.1.2 对称正定情况下的 CG 解法 .....	25
5.2 块大小的选取 .....	34
5.3 重新开始策略 .....	34
第六章 数值结果 .....	35
第七章 总结 .....	39

参考文献 .....	40
致 谢 .....	44
在学期间的研究成果及发表的学术论文.....	45

# 第一章 绪论

## 1.1 问题背景

代数特征值问题是数值代数的一个重要领域，至今已有一百多年的历史，其研究具有重要的理论意义和应用价值。许多科学和工程问题会遇到特征值和特征向量的计算，如结构力学中的动力分析问题可转化为特征值问题，电子、机械、航空、航天、桥梁等结构振动问题以及控制系统中的稳定性问题最终都转化为特征值问题。因此，有效求解特征值问题具有广泛的应用价值。

矩阵  $A$  的特征值问题从表面上看是一个简单的非线性方程组

$$Ax = \lambda x,$$

其中  $\lambda$  是  $A$  的特征值， $x$  为  $A$  相应于  $\lambda$  的特征向量。计算特征值  $\lambda$  的最直接方法就是求解特征方程  $\det(\lambda I - A) = 0$ ，但是该方法只适合一些特殊的矩阵。首先，如果  $A$  的阶数比较大，则方程  $\det(\lambda I - A) = 0$  的计算量非常大；其次，特征方程  $\det(\lambda I - A) = 0$  的根通常是不稳定的，即方程系数的小扰动将导致方程根的较大误差。基于上述原因，我们需要寻求其它的求解方法。

目前，求解特征值问题的方法可以分为两大类：一类称为变换方法，另一类称为投影方法。变换方法是直接对原矩阵进行处理，通过一系列相似变换，使之变成一个易于计算特征值的形式，如 Jacobi 方法<sup>[1]</sup>、Givens 方法<sup>[1]</sup>、QR 方法<sup>[1, 2]</sup>等。变换方法可以求得矩阵的全部特征值，但通常需要存储矩阵的全部元素，以致存储量较大。因此，这类方法适用于求解中小规模的矩阵特征值问题。但在一些应用中所出现的特征值问题，其矩阵往往是大型稀疏的，并且只需要计算部分特征值和特征向量。对此类大规模矩阵特征值问题，变换方法往往因为存储量的制约而不适用。投影方法<sup>[3,4]</sup>是通过投影算子将大规模矩阵特征值问题投影到适当低维子空间中，即将大规模特征值问题转化为中、小型矩阵特征值问题，然后采用变换方法计算其特征对作为大型矩阵特征对的近似。投影方法的特点是仅利用矩阵  $A$  形成矩阵-向量乘积，不需要对矩阵  $A$  本身作变换，只需存储矩阵  $A$  的非零元，从而可减少内存需求，同时降低计算量。由于投影方法可采用收缩存储技术，因而它适合于求解大型稀疏矩阵的特征值问题。目前常用的求解大型稀疏矩阵特征值问题的投影方法有子空间迭代法<sup>[1-3]</sup>、Lanczos 方法<sup>[1,3-7]</sup>和 Arnoldi 方法<sup>[8-10]</sup>。子空间迭代法可同时求得几个端部特征值及特征向量。子空间迭代法的缺点是收敛速度较慢，从而运算量较大，并且随着迭代次数的增加，舍入误差的影响也会增大。Lanczos 方法利用三项递推公式产生 Krylov 子空间的标准正交基（即 Lanczos 向量）和三对角投影矩阵，存储量较小，计算速度较快。Arnoldi 方法是 Lanczos 方法在非对称矩阵情况下的推广，该方法利用 Arnoldi 过程产生 Krylov 子空间的标准正交基（即 Arnoldi 向量）和上 Hessenberg 投影矩阵。但是在迭



代过程中 Lanczos 向量和 Arnoldi 向量容易失去正交性,为解决此问题,往往需要进行重正交化,从而使计算量增加。

1975 年,为解决量子化学中出现的特征值问题,Davidson<sup>[11]</sup>提出了计算大型对称矩阵极端特征值的一种新方法——Davidson 方法。最初,Davidson 方法是通过摄动方法得到的,其主要思想是用一个包含特征方向更多信息的向量扩充搜索子空间以提高逼近效果。之后,许多学者对 Davidson 方法进行了研究和改进。Crouzeix 等人<sup>[12]</sup>分析了 Davidson 方法的收敛性,并将 Davidson 方法推广到块格式,提出了求解矩阵特征值问题的块 Davidson 方法。Davidson 方法已发展成为求解对角占优 Hermitian 矩阵若干个最小或最大特征值的有效方法。然而,对非对角占优 Hermitian 矩阵的特征值问题,Davidson 方法的计算效果不尽如人意,而且精度很高的预条件子会导致收敛减慢甚至停滞。

1996 年,Sleijpen 和 Van der Vorst<sup>[13]</sup>将 Jacobi 方法的校正思想和 Davidson 方法的内外迭代格式相结合,提出了 Jacobi-Davidson 方法。这一方法的核心是通过在近似特征向量的正交补空间上寻求对近似特征向量的修正向量,然后通过修正向量来扩展子空间,再在扩展的子空间中计算新的逼近效果更好的 Ritz 对作为特征对的近似。该方法克服了 Davidson 方法在扩充投影子空间方面存在的缺陷,具有较好的稳定性,对于非对角占优、非正规矩阵也能达到较快的收敛速度,因此 Jacobi-Davidson 方法被认为是计算矩阵特征值的最有效方法之一。Van Den Eshof<sup>[14]</sup>分析了 Jacobi-Davidson 方法的收敛性。

当要求的特征值是重特征值或密集特征值时,问题的性态可能会变坏,通常算法的有效性会下降。为解决这个问题,人们研究和开发了投影方法的块格式,比如块 Arnoldi 方法<sup>[15-17]</sup>,块 Lanczos 方法<sup>[18,19]</sup>,块 Davidson 方法<sup>[12,17]</sup>等。文献[20]通过组合块 Jacobi 方法和块 Davidson 方法,提出了求解对称矩阵特征值问题的块 Jacobi-Davidson 方法。该方法可同时计算多个特征值和相应的特征向量。如果块的大小选择适当,块 Jacobi-Davidson 方法将更有效,它不仅改进了 Jacobi-Davidson 方法的收敛性,而且适用于计算大型稀疏矩阵若干个最大或最小特征值和相应的特征向量。

近年来,诸多学者对上述各种块方法作出了大量的研究和改进。例如,为了进一步提高块方法的有效性,人们将收缩技术与块方法结合,将已经收敛的特征对提取出来,不参加下一轮迭代。为了减少存储量,所选的投影子空间的维数  $m$  通常较小,这样在经过  $m$  步迭代后,计算结果的精度往往不满足要求,所以还需要重新开始过程,并用 Ritz 向量作为重新开始过程的初始向量。但是对有些问题,虽然 Ritz 值收敛,但 Ritz 向量可能收敛比较慢甚至不收敛。鉴于此,为改善近似特征向量的收敛性,一些学者对重新开始初始向量的选取进行了研究,提出了一些较好的处理方法,如隐式重新开始技术<sup>[21-23]</sup>,动态稠密重新开始过程<sup>[24]</sup>,用 CG 递归进行重新开始<sup>[25]</sup>,组合 CG 递归和动态稠密重新开始<sup>[26]</sup>等。另外,为了改善投影方法中 Ritz 向量的收敛性,贾仲孝等<sup>[27-32]</sup>提出了精化策略以选择重新开始的初始向量,给出了精化 Arnoldi 方法,

精化块 Arnoldi 方法, 精化子空间迭代法, 精化 Jacobi-Davidson 方法等。

## 1.2 本文研究工作与内容安排

本文主要研究求解大型对称特征值问题的块 Jacobi-Davidson (BJD) 方法。

块 Jacobi-Davidson 方法是 Jacobi-Davidson 方法的块格式, 它不仅能有效计算对称矩阵重或密集特征值, 改进了 Jacobi-Davidson 方法的收敛性, 并且可以同时计算若干个极端特征对。但在实际计算中, 迭代过程中产生的一些已收敛的 Ritz 对仍然会参加后续的迭代运算, 这降低了该方法的总体收敛速度。其次, 近年来许多实际应用需要计算矩阵的内部特征值, 块方法虽然对重或密集特征值更有效, 但块 Jacobi-Davidson 方法在计算所需的内部特征值之前首先要把该特征值前面的所有特征值 (比如说大于该特征值的所有特征值) 计算出来, 这便带来许多不必要的计算量, 导致块 Jacobi-Davidson 方法计算矩阵内部特征值的工作量大增。再次, Jacobi-Davidson 方法和块 Jacobi-Davidson 方法的成功主要在于对投影子空间的有效扩充, 而投影子空间的扩充向量正是通过求解校正方程得到的, 因此求解校正方程组便是块 Jacobi-Davidson 方法以及其改进方法的关键。但块 Jacobi-Davidson 方法及其改进方法的校正方程由于系数矩阵随着逼近程度的越来越好, 有可能变得病态, 这便需要研究如何更有效地求解校正方程。

本文主要针对以上几个问题, 对块 Jacobi-Davidson 方法进行改进。为了提高块 Jacobi-Davidson 方法的整体收敛速度, 应用动态收缩技术, 提出了动态收缩的块 Jacobi-Davidson 方法; 为了计算大型对称矩阵的内部特征对, 本文将调和 Rayleigh-Ritz 方法与块 Jacobi-Davidson 方法结合, 提出了调和块 Jacobi-Davidson 方法, 并将动态收缩技术应用于调和块 Jacobi-Davidson 方法, 给出了动态收缩的调和块 Jacobi-Davidson 方法。对于校正方程的求解, 本文通过使用校正方程的等价增广形式将校正方程转化为经典的 saddle point 问题, 并对其使用适当的预处理技术。

本文的结构如下。第二章简单介绍了投影方法和 Jacobi-Davidson 方法, 为后面的研究做必要的准备。第三章, 在仔细研究块 Jacobi-Davidson 算法执行过程的基础上, 提出了动态收缩的块 Jacobi-Davidson 方法, 并给出了算法描述。第四章, 将调和 Rayleigh-Ritz 方法与块 Jacobi-Davidson 方法结合, 提出了调和块 Jacobi-Davidson 方法, 并将动态收缩技术应用于调和块 Jacobi-Davidson 方法, 给出了动态收缩的调和块 Jacobi-Davidson 方法, 并给出了各自的算法描述。第五章, 研究了各类算法实现的具体细节问题, 比如块大小的选取, 校正方程的有效求解, 重新开始的策略等。第六章通过 matlab 编写程序, 并通过若干数值例子, 在有效性和收敛性方面比较了改进的块 Jacobi-Davidson 方法和块 Jacobi-Davidson 方法, 揭示出新算法的优越性。第七章, 对本文工作作了总结, 并指出有待进一步研究的问题。

### 1.3 符号约定

这一节将给出本文所涉及到的记号和约定。

- “JD 方法”和“块 JD 方法”分别代表 Jacobi-Davidson 方法和块 Jacobi-Davidson 方法。
- $R^n$  表示  $n$  维实向量空间,  $R^{n \times m}$  表示所有  $n \times m$  实矩阵的全体。
- $x^T$  和  $X^T$  分别表示向量  $x$  和矩阵  $X$  的转置。
- 对  $n$  阶矩阵  $A = (a_{ij})$ , 记  $\text{diag}(A) = \text{diag}(a_{11}, \dots, a_{nn})$ 。
- $A$  表示  $n \times n$  实对称矩阵,  $\lambda_i$  表示  $A$  的第  $i$  个特征值。  $A$  的特征值排列次序为

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n。$$

- 任何对称矩阵的特征向量均取为规范正交特征向量。特别地, 取  $A$  对应于特征值  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  的规范正交特征向量系为  $\phi_1, \phi_2, \dots, \phi_n$ 。

- 如不特别声明, 所有的矩阵、向量范数均指 2-范数。
- $I_n$  表示  $n$  阶单位矩阵, 在不产生混淆时, 用  $I$  表示单位矩阵。
- MGS 表示 Modified Gram Schmidt 正交化过程。
- 列规范正交矩阵  $Q \in R^{n \times m}$  是指满足性质  $Q^T Q = I_m$  的矩阵。
- 如不特别声明,  $x \perp y$  表示向量  $x$  与  $y$  正交, 而  $x \perp U \in R^{n \times m}$  表示向量  $x$  和  $U$  的各列及

列张成的子空间正交。

- 如果  $x_1, x_2, \dots, x_m$  是  $m$  个  $n$  维向量, 则  $Z = (x_1, x_2, \dots, x_m)$  意味着  $Z$  是一个  $n \times m$  矩阵, 其中  $x_j$  是  $Z$  的第  $j$  列。类似地, 如果  $X_1, X_2, \dots, X_m$  是  $m$  个  $n \times p$  矩阵, 则  $\bar{Z} = (X_1, X_2, \dots, X_m)$  表示一个  $n \times (p \times m)$  矩阵,  $\bar{Z}$  的前  $p$  列是  $X_1$ 。

- 如果  $Z = (x_1, x_2, \dots, x_m)$ , 则  $\text{span}(x_1, x_2, \dots, x_m)$  或  $\text{span}(Z)$  表示  $Z$  的列所张成的子空间。设  $v$  是  $n$  维向量,  $A$  是  $n$  阶矩阵,  $K_m(A, v) = \text{span}(v, Av, \dots, A^{m-1}v)$  称为 Krylov 子空间。

## 第二章 预备知识

### 2.1 投影方法

QR 方法<sup>[1, 2]</sup>是求解中、小规模矩阵特征值问题最有效的方法之一。目前, 不仅在 QR 方法的可靠性分析、数值稳定性研究等方面取得了许多的成果, 而且该方法已被成功地编制成了软件, 例如, MATLAB 软件包中计算矩阵全部特征对的子程序 eig.m 便是基于 QR 方法编制而成的。此外, Jacobi 方法<sup>[1]</sup>和分而治之方法<sup>[2]</sup>也是求解中、小规模矩阵特征值问题很有效的数值方法, 在某些方面具有各自的优点。因此, 从某种意义上说, 中、小规模矩阵特征值问题基本得到了圆满解决。

然而, 对于大规模矩阵特征值问题, 计算量、存储量等限制条件的存在使得对中、小规模矩阵特征值问题行之有效的方法在求解大规模问题时变得无能为力, 并且对于某些在实际问题中产生的大型稀疏矩阵, 我们有时仅需要计算其某些特征对。如果仍用 QR 方法或 Jacobi 方法求解, 即使可以计算其全部特征对, 但是会造成很大的浪费。正是由于这些因素的影响, 寻求和研究大规模矩阵部分特征值问题的数值方法成为数值代数领域的研究热点之一。

解决大规模矩阵  $A$  的部分特征值问题的最常用方法是投影类方法。其基本思想是将原矩阵  $A$  投影到某个低维子空间, 将大规模问题转化为中、小规模矩阵特征值问题, 然后采用常规方法计算其特征对, 并将这些特征对作为大规模矩阵  $A$  的部分特征对的近似。投影类方法大致可以分为如下三类:

#### 2.1.1 正交投影方法

**定义 2.1**<sup>[31]</sup> 设  $A$  为  $n$  阶矩阵, 给定  $m$  维子空间  $E$ , 在  $E$  上的正交投影方法就是寻找  $(\tilde{\lambda}_i, \tilde{\varphi}_i) (\|\tilde{\varphi}_i\| = 1)$  满足 Galerkin 投影条件:

$$\begin{cases} \tilde{\varphi}_i \in E \\ (A - \tilde{\lambda}_i I) \tilde{\varphi}_i \perp E \end{cases} \quad (2.1)$$

并用  $(\tilde{\lambda}_i, \tilde{\varphi}_i)$  作为  $A$  的特征对  $(\lambda_i, \varphi_i)$  的近似。

如果  $\{v_j\}_1^m$  是子空间  $E$  的一组标准正交基, 定义  $V_m = [v_1, v_2, \dots, v_m]$ , 则上述关系式可以写成:

$$\begin{cases} \tilde{\varphi}_i = V_m y_i \\ H y_i = \tilde{\lambda}_i y_i \end{cases} \quad (2.2)$$

其中  $H = V_m^T A V_m$  是  $A$  在标准正交基  $\{v_j\}_j^m$  下对子空间  $E$  的投影矩阵。若  $P_m$  表示子空间  $E$  上的

正交投影算子，则上面的条件 (2.1) 可改写为

$$P_m(A - \tilde{\lambda}_i I)\tilde{\varphi}_i = 0. \quad (2.3)$$

**定义 2.2**<sup>[2]</sup> 对  $n$  阶矩阵  $A$ ， $V_m$  是  $n \times m$  的列正交规范矩阵， $H = V_m^T A V_m$ ，假设  $H$  的某一特征对为  $(\tilde{\lambda}_i, y_i)$ ，则称  $\tilde{\lambda}_i$  为  $A$  在子空间  $E = \text{span}(V_m)$  上的 Ritz 值；称  $\tilde{\varphi}_i = V_m y_i$  为相应的 Ritz 向量。用  $\tilde{\lambda}_i$  和  $\tilde{\varphi}_i$  来逼近  $A$  的特征值和特征向量，此逼近称为 Rayleigh-Ritz 逼近。

事实上，正交投影方法就是 Rayleigh-Ritz 逼近。

设线性算子  $A_m = P_m A P_m$ ，于是有如下定理。

**定理 2.1**<sup>[3]</sup> 子空间  $E$  中的 Ritz 值  $\tilde{\lambda}_i$  和 Ritz 向量  $\tilde{\varphi}_i$  是  $A_m$  的特征值和特征向量。

利用正交投影方法求得的 Ritz 值  $\tilde{\lambda}_i$  和 Ritz 向量  $\tilde{\varphi}_i$  是否收敛于特征值问题 (2.1) 的特征值  $\lambda_i$  和特征向量  $\varphi_i$ ？实际上，当  $m = n$  时，Ritz 值  $\tilde{\lambda}_i$  和 Ritz 向量  $\tilde{\varphi}_i$  就是要求的特征值和特征向量。如果  $m \neq n$ ，并且  $A_m$  的特征值  $\tilde{\lambda}_i$  和特征向量  $\tilde{\varphi}_i$  接近  $A$  的特征值  $\lambda_i$  和特征向量  $\varphi_i$ ，则残量  $(A - \tilde{\lambda}_i I)\tilde{\varphi}_i$  是小量。关于  $\|(A - \tilde{\lambda}_i I)\tilde{\varphi}_i\|$ ，有如下估计式。

**定理 2.2**<sup>[3]</sup> 令  $\gamma_m = \|P_m A(I - P_m)\|$ ，则

$$\|(A - \tilde{\lambda}_i I)\tilde{\varphi}_i\| \leq \sqrt{|\lambda_i|^2 + \gamma_m^2} \|(I - P_m)\varphi_i\|. \quad (2.4)$$

由定理 2.2 可见，特征向量  $\varphi_i$  与子空间  $E$  之间的距离  $\|(I - P_m)\varphi_i\|$  是一个非常重要的量，通常  $\|(I - P_m)\varphi_i\|$  不是小量，当且仅当  $m = n$  时， $\|(I - P_m)\varphi_i\| = 0$ 。当  $m < n$  时， $I - P_m$  是一个正交投影算子， $\|(I - P_m)\varphi_i\|$  并不随  $m$  的增大而必然减小。但是如果子空间  $E$  比较特殊，可以保证  $\|(I - P_m)\varphi_i\|$  随  $m$  的增大而迅速减小。

假设子空间  $E$  是  $A$  的一个  $m$  维不变子空间，则必有  $m \times m$  矩阵  $H$ ，使得

$$R = A V_m - V_m H = 0 \quad (2.5)$$

显然，矩阵  $H$  是  $A$  对  $E$  的限制算子  $\pi_m$  的矩阵表示，并且  $H$  的特征值都是  $A$  的特征值， $\pi_m$  的特征值和特征向量都是  $A$  的特征值和特征向量。

若子空间  $E$  不是  $A$  的不变子空间，则对  $R(H) = A V_m - V_m H$  中矩阵  $H$  和  $A$  的特征值之间的关系，有下面的定理。

**定理 2.3**<sup>[1]</sup> 设  $V_m$  由  $m$  个  $n$  维正交规范向量组成， $H$  是  $m \times m$  矩阵，记

$$R(H) = A V_m - V_m H,$$

若  $A$  的特征值为  $\lambda_1, \lambda_2, \dots, \lambda_n$ ， $H$  的特征值为  $\tilde{\lambda}_1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_m$ ，则可以找到  $m$  个不同的自然数  $i_1, i_2, \dots, i_m$ ，使得

$$|\lambda_{i_k} - \tilde{\lambda}_k| \leq \|R(H)\|, k = 1, 2, \dots, m \quad (2.6)$$

上述定理就是著名的 Kahan 定理。此定理说明了矩阵  $H$  和  $A$  的特征值之间的关系，只要  $\|R(H)\|$  充分小， $H$  的特征值就可以作为  $A$  的特征值较好的近似。显然， $\|R(H)\|$  的大小与  $H$  的选取紧密相关。下面的定理给出了当  $H$  取何值时， $\|R(H)\|$  可取得最小值。

**定理 2.4**<sup>[1]</sup> 假设矩阵  $V_m$  和  $H$  满足定理 2.3 的条件, 记

$$R(H) = AV_m - V_m H, \quad Q = V_m^T AV_m$$

则  $\|R(Q)\| \leq \|R(H)\|$ 。

对形如  $AV_m - V_m H$  的矩阵, 只有当  $H = V_m^T AV_m$  时,  $AV_m - V_m H$  的范数才达到最小。因此, 正交投影方法是一种最佳逼近。在求解特征值问题时, 如果从子空间逼近, 一般都采用此方法。

选取不同的子空间  $E$  将会导出不同的正交投影方法。目前常用的正交投影方法有子空间迭代法, Lanczos 方法, Arnoldi 方法, Jacobi-Davidson 方法等。

对于求解大规模矩阵特征值问题的经典正交投影方法, 比如子空间迭代法和 Lanczos 方法等, 学术界已经进行了长达三十多年的深入研究。对大规模对称矩阵特征值问题的经典正交投影方法, 在算法的理论分析、数值性态分析和软件研制等方面, 已取得许多研究成果。而对于大规模矩阵非对称特征值问题的正交投影方法, 矩阵的非对称性给算法的设计和分析带来了一些本质的困难。

## 2.1.2 斜投影方法

给定两个  $m$ -维子空间  $K$  和  $L$ , 斜投影方法就是寻找  $(\tilde{\lambda}_i, \tilde{\varphi}_i)$  ( $\|\tilde{\varphi}_i\| = 1$ ) 满足下列条件

$$\begin{cases} \tilde{\varphi}_i \in K \\ (A - \tilde{\lambda}_i I)\tilde{\varphi}_i \perp L \end{cases} \quad (2.7)$$

并用  $(\tilde{\lambda}_i, \tilde{\varphi}_i)$  作为  $A$  的特征对  $(\lambda_i, \varphi_i)$  的近似。子空间  $K$  称为右子空间,  $L$  称为左子空间。

假设  $W_m$  和  $V_m$  均为  $n \times m$  矩阵, 它们的列向量分别是子空间  $K$  和  $L$  的一组基, 则上述关系式可以写成

$$\begin{cases} \tilde{\varphi}_i = V_m y_i \\ T y_i = \tilde{\lambda}_i W_m^H V_m y_i \end{cases} \quad (2.8)$$

其中  $T = W_m^H AV_m$ 。

$K$  和  $L$  的不同选取方式会导出不同的斜投影方法。研究较多的有两种, 一种为调和投影方法, 另一种为双正交 Lanczos 方法。

### 1) 调和投影方法

如果左、右子空间  $L$ 、 $K$  的基向量满足  $W_m = AV_m$ , 则该斜投影方法称为调和投影方法。若  $K = K_m(A, v_0)$ , 且  $K$  的基为标准正交基时, 则相应的调和投影方法称为调和 Arnoldi 方法<sup>[33-36]</sup>。调和 Arnoldi 方法是 Morgan<sup>[33]</sup>在研究计算大规模矩阵内部部分特征对时首先提出的。最早研究调和 Ritz 值的是 Morgan<sup>[33]</sup>和 Freund<sup>[37]</sup>, 而“调和”这一词是由 Paige 等人<sup>[35]</sup>引入到对称矩阵, 后来 Sleijpen 等人<sup>[13]</sup>将其引入到非对称情形。

调和投影方法的一个优点是可以通过恰当选取位移  $\alpha$  将原来密集特征值问题进行改善,使原问题特征值的分布更适于求解。众所周知,标准的投影方法并不适于内部特征值问题的求解。当然,我们可以通过位移求逆变换来求解内部特征对,但是这要求逆矩阵  $(A - \alpha I)^{-1}$ , 所需的运算量往往比较大。因此当矩阵阶数比较大或需要多次求逆的时候,位移求逆方法并不是一种理想的选择。调和投影方法的另一个优点就是可以将内部特征值问题转换为外部特征值问题,而不用显式求逆。近年来,调和投影方法已经成为计算矩阵内部特征对最常用和有效的方法之一。

## 2) 双正交 Lanczos 方法

当选取  $K = K_m(A, v_0)$ ,  $L = K_m(A^H, w_0)$  并且选取  $W_m$  和  $V_m$  的列向量为双正交向量(满足  $W_m^H V_m = I_m$ ) 时,斜投影方法称为双正交 Lanczos 方法<sup>[5, 38]</sup>。此方法由 Lanczos<sup>[5]</sup>于 1950 年提出,基本思想是利用两组正交的向量将矩阵  $A$  逐次地三对角化,然后计算此三对角阵的特征值,并将它们作为矩阵  $A$  的特征值的近似。双正交 Lanczos 方法只利用原矩阵形成矩阵-向量积  $Ax$  和矩阵转置-向量积  $A^H x$ ,从概念上讲,此方法本身只是一个将一般矩阵化成三对角阵的直接方法。在有限精度下,该方法的双正交性会很快失去,因而,双正交 Lanczos 方法提出后,很快便被更有效、更稳定的 Householder 变换和 Givens 变换取代。

当矩阵  $A$  对称时,Paige<sup>[6]</sup>在有限精度条件下,对 Lanczos 算法进行了系统分析。其分析结果表明正交性的损失并不是坏事,相反,正交性的损失必然伴随着至少一个 Ritz 值收敛于矩阵  $A$  的特征值,反之亦然。目前,虽然在理论上还有一些收敛性分析问题没有得到解决,但 Lanczos 算法已成为求解大型对称矩阵部分最大(最小)特征值及相应特征向量的非常可靠和健壮的技术之一。

然而,当矩阵  $A$  非对称时,在三对角化过程的每一步,都可能会发生严重中断<sup>[2]</sup>,并且严重中断的出现与舍入误差、问题的病态程度均无关。此时,问题变得非常复杂和困难。关于如何处理和避免严重中断,Taylor<sup>[39]</sup>和 Parlett<sup>[40]</sup>提出了 Lanczos 算法的“look-ahead”策略,它利用  $2 \times 2$  选主元来改进 Lanczos 算法。后来 Cullum<sup>[41]</sup>将不带再正交化的对称 Lanczos 算法推广至非对称情况并提出了一种新的矩阵三对角化方法。同时, Freund 等人<sup>[42]</sup>, Parlett<sup>[43]</sup>和 Saad<sup>[44]</sup>等人从不同的角度研究了此方法。在没有严重中断发生的前提下,Ye<sup>[45]</sup>分析了此方法的收敛性。Bai<sup>[46]</sup>的分析表明:在有限精度下,如果 Ritz 值的条件数合理并且没有严重中断发生,则 Ritz 值的收敛性即意味着双正交性的失去。

### 2.1.3 精化投影方法

对于非对称矩阵  $A$ ,用正交投影方法计算其部分特征对时,收敛速度可能非常慢甚至发散,收敛性态极不规则。为克服这一隐患,贾<sup>[47]</sup>提出对于确定的近似特征值,摒弃现有的由经典投影方法求得的近似特征向量,而用另一个在求解子空间上满足最优条件的新向量作为近似特征

向量。这种方法称为精化投影方法，而在求解子空间上满足最优条件的近似特征向量称为精化向量。理论分析表明这种新的方法实际为两个正交投影的复合<sup>[47]</sup>。

对任意求解子空间  $E$ ，以及一个近似特征值  $\tilde{\lambda}_i$ （可以是 Ritz 值、调和 Ritz 值或者由其它适当方法得到的近似特征值），精化投影方法<sup>[27-29]</sup>就是求满足

$$\|(A - \tilde{\lambda}_i I)\tilde{u}_i\| = \min_{u \in E, \|u\|=1} \|(A - \tilde{\lambda}_i I)u\| \quad (2.9)$$

的向量  $\tilde{u}_i$  ( $\|\tilde{u}_i\|=1$ ) 作为与近似特征值  $\tilde{\lambda}_i$  相对应的近似特征向量。称  $\tilde{u}_i$  为  $A$  在子空间  $E$  上与  $\tilde{\lambda}_i$  相对应的精化向量。对指定的求解子空间  $E$ ，精化向量  $\tilde{u}_i$  可以通过计算一个小规模矩阵的特征值或小规模矩阵的奇异值分解 (SVD) 获得。特别地，当  $\tilde{\lambda}_i$  为  $A$  在子空间上的 Ritz 值时，由关系式 (2.1) 和 (2.9)，精化投影方法可以表述为

$$\begin{cases} \tilde{\varphi}_i \in E \\ (A - \tilde{\lambda}_i I)\tilde{\varphi}_i \perp E \\ \|(A - \tilde{\lambda}_i I)\tilde{u}_i\| = \min_{u \in E, \|u\|=1} \|(A - \tilde{\lambda}_i I)u\| \end{cases} \quad (2.10)$$

它用  $(\tilde{\lambda}_i, \tilde{\varphi}_i)$  作为  $(\lambda_i, \varphi_i)$  的近似值。

## 2.2 Jacobi-Davidson 方法

计算大规模矩阵部分特征值及相应的特征向量的常用方法是投影类方法。投影类方法不对矩阵本身进行处理，在计算过程中只需要形成矩阵-向量乘积的子程序（或者再加上矩阵转置-向量乘积的子程序），再利用其他一些处理技术构造一个尽可能多的包含待求特征对信息的低维子空间，并将矩阵投影到这个“适当的”低维子空间上，从而在这个低维子空间中计算出令人满意的近似解。因此，在投影方法的计算过程中最关键的步骤就是获得高“质量”的低维投影子空间。为了获得高“质量”的低维子空间，通常做法是对当前子空间不断的进行有效扩充，从而完成对近似特征向量的不断修正，使其逐渐逼近待求特征向量。在此过程中若能保持子空间的最大维数  $m$  最小就能大大节省计算时间及存储空间。

Davidson 方法<sup>[11]</sup>是由 E. R. Davidson 于 1975 年提出的一种计算大型稀疏对称矩阵极端特征值的有效方法，也是近年来研究活跃的投影方法之一。对于对角占优矩阵，Davidson 方法的收敛速度比 Lanczos 方法快。该方法提出后，很快便被推广到一般矩阵的情况，但它在投影子空间的扩充方面存在着缺陷，于是 Sleijpen 和 Van der Vorst<sup>[13]</sup>提出在近似特征向量的正交补空间中寻求特征向量新信息的 Jacobi-Davidson 方法。他们通过求解校正方程

$$(I - \tilde{u}\tilde{u}^T)(A - \tilde{\lambda}I)(I - \tilde{u}\tilde{u}^T)t = -r; \quad t \perp \tilde{u} \quad (2.11)$$

获得对  $\tilde{u}$  的一个修正向量  $t$ 。式中的  $(\tilde{\lambda}, \tilde{u})$  为当前子空间  $V_k$  上关于特征对  $(\lambda, u)$  的一个近似特征对， $r = A\tilde{u} - \tilde{\lambda}\tilde{u}$  为其残量。利用 (2.11) 的解  $t$  将投影子空间由  $V_k$  扩充至  $V_{k+1}$ ，并在  $V_{k+1}$  中



获得  $(\lambda, u)$  的一个更好的近似特征对。

不妨假设待求特征向量  $u$  可分解为  $u = \tilde{u} + z$  ( $z \perp \tilde{u}$ )，注意到  $\tilde{u}$  已在  $V_k$  中。为尽可能有效地扩充子空间，需要从  $[\text{span}(\tilde{u})]^\perp$  上找一个合理的向量，将其加入  $V_k$ 。若将  $u = \tilde{u} + z$  ( $z \perp \tilde{u}$ ) 代入  $Au = \lambda u$  中，则有

$$A(\tilde{u} + z) = \lambda(\tilde{u} + z), \quad (2.12)$$

即

$$(A - \lambda I)z = -(A - \lambda I)\tilde{u},$$

注意到  $r = A\tilde{u} - \tilde{\lambda}\tilde{u} = (A - \lambda I)\tilde{u} + (\lambda - \tilde{\lambda})\tilde{u}$ ，而  $(I - \tilde{u}\tilde{u}^T)r = r$ ，则

$$(I - \tilde{u}\tilde{u}^T)(A - \lambda I)\tilde{u} = r,$$

且有  $(I - \tilde{u}\tilde{u}^T)z = z$ ，于是可得

$$(I - \tilde{u}\tilde{u}^T)(A - \lambda I)(I - \tilde{u}\tilde{u}^T)z = -r. \quad (2.13)$$

由于  $\lambda$  是待求的，用 Ritz 值  $\tilde{\lambda}$  代替  $\lambda$  就可得到

$$(I - \tilde{u}\tilde{u}^T)(A - \tilde{\lambda}I)(I - \tilde{u}\tilde{u}^T)z = -r. \quad (2.14)$$

这便是 Jacobi-Davidson 方法校正方程的导出过程。文献[13]讨论了如此扩充投影子空间的优点，说明了它可以比较有效的避免 Davidson 方法可能出现的中断问题。

### 2.2.1 Jacobi-Davidson 方法的算法描述

求矩阵  $A$  的模最大特征对的 Jacobi-Davidson 方法可描述如下。

#### 算法 2.1 Jacobi-Davidson 方法

输入：矩阵  $A$ ，任意  $n \times 1$  模为 1 的初始向量  $v_1$ ，容许误差  $\varepsilon$

输出：最大特征对的一个近似  $(\theta, u)$

迭代：

1.  $V_1 = [v_1]$

2. For  $k = 1, 2, \dots$

2.1) 计算  $w = Av_k$ ;

2.2) 计算  $H_k = V_k^T AV_k$  的最后一列  $V_k^T w$ ，以及最后一行  $w^T V_k$ ;

2.3) 计算  $H_k$  的最大特征对  $(\theta_k, y_k)$ ，并计算  $u_k = V_k y_k$ ;

2.4) 计算残量  $r = Au_k - \theta_k u_k$ ;

2.5) 如果  $\|r\| \leq \varepsilon$ ，则停止；否则进行下一步；

2.6) 近似求解下面的校正方程，得到校正向量  $t$ ：

$$(I - u_k u_k^T)(A - \theta_k I)(I - u_k u_k^T)t = -r, \quad t \perp u_k$$

2.7) 如果  $k+1 \leq m$ ，则  $V_{k+1} = \text{mgs}(V_k, t)$ ，返回 2；否则  $V_1 = [u_k]$ ，返回 2。

上述算法是计算矩阵  $A$  的最大特征值及相应的特征向量的 Jacobi-Davidson 算法。如果需要

计算  $A$  的按实部（或按虚部、或按模）最大的特征值时，只要在本算法中做相应的调整即可。

为了减少运算过程中的存储量和计算量，算法中投影子空间的最大维数  $m$  不能很大，而算法在  $m$  步迭代中，一般满足不了收敛精度，所以几乎所有的投影算法都采用重新开始技术。随着人们对其不断研究，重新开始策略越来越丰富，较显著的有显示、隐式重新开始和动态稠密重新开始技术等。此外，在重新开始过程中，保留多少、保留哪些 Ritz 向量也是 JD 方法中一个非常重要的环节。这些改进使得 JD 方法变得更加灵活、更加有效。目前，JD 方法已成为计算大型对称和非对称稀疏矩阵极端特征值问题的最有效方法之一。

## 2.2.2 校正方程的求解与收敛性分析

对校正方程

$$(I - u_k u_k^T)(A - \theta_k I)(I - u_k u_k^T)t = -r$$

求解  $t \in [\text{span}(u_k)]^\perp$ ，我们可以把上式看作变换算子把  $(A - \theta_k I)$  限制到  $u_k$  的正交补上。在理论上有如下结果：

**引理 2.1** <sup>[48]</sup> (三次渐进收敛的 Rayleigh 商迭代) 如果矩阵  $A$  是正规矩阵，并且近似特征对  $(\mu_k, \varphi_k)$  收敛，则  $\lim_{k \rightarrow \infty} \|\varphi_{k+1} - v_k\| / \|\varphi_k - v_k\|^3 \leq 1$ 。

上述引理中，之所以能达到三次收敛是因为矩阵  $A$  是正规的，当矩阵  $A$  是非正规时，最好的结果是二次收敛。这时，人们提出了双边 Rayleigh 商迭代法，从而使算法达到局部三次收敛 <sup>[49]</sup>。

**定理 2.5** 设对  $v_k \in V_k$  ( $\|v_k\| = 1$ )，假设 Rayleigh 商  $\theta_k = v_k^H A v_k$  不是  $A$  的特征值，则在下一步迭代中要用到的 Rayleigh 商迭代方向包含在 JD 方法由本次迭代过程产生的子空间中。

证明：根据一般方程

$$\begin{cases} t \perp v_k \\ (I - \alpha v_k v_k^H)(A - \theta_k I)(I - \beta v_k v_k^H)t = -r \end{cases} \quad \forall \alpha \neq 0, \quad (2.15)$$

我们知道，Jacobi-Davidson 方法的校正方程恰好是方程 (2.15) 的特殊形式，所以我们来证明下面更强的结论：由方程 (2.15) 的解扩张的子空间包括下一步迭代中用到的 Rayleigh 商迭代方向。

事实上，在 Jacobi-Davidson 方法迭代中，JD 方法是用  $t$  来扩展投影子空间  $V_k$  的。因为  $v_k \in V_k$ ，所以我们用  $\tilde{t} = (A - \theta_k I)^{-1} v_k$  来作扩展向量可得到相同的投影子空间。从而 Rayleigh 商迭代产生的方向是

$$\tilde{v}_k = (A - \theta_k I)^{-1} v_k。$$

在  $v_k^H t = 0$  的限制下，方程 (2.15) 可化为

$$(A - \theta_k I)t = -r - \eta v_k。 \quad (2.16)$$

这实际上是广义 Davidson (GD) 方程。从 (2.16) 和 (2.12) 式中可得

$$t = (A - \theta_k I)^{-1} r - \eta (A - \theta_k I)^{-1} v_k = -v_k - \eta \tilde{v}_k, \quad (2.17)$$

其中  $\eta = -1/(v_k^H v_k)$  的选择保证  $v_k^H t = 0$  成立。

注意  $v_k \in V_k$ ，我们看到此时 Rayleigh 商迭代方向  $\tilde{v}_k \in V_{j+1}$ 。

这个定理说明，Jacobi-Davidson 方法能够保留 Rayleigh 商迭代令人满意的收敛速度，并且它通常比 Rayleigh 商迭代方法的收敛速度更快。

## 第三章 动态收缩的块 Jacobi-Davidson 方法

### 3.1 块 Jacobi-Davidson 方法

若要求的特征值分布很密集或矩阵的主特征值是重特征值时, Jacobi-Davidson 方法的有效性会下降。可能遇到的主要问题如下:一是如果当前的近似特征值很准确时,校正方程的系数矩阵会变得病态;二是如果当前的近似特征值是单的并且与其他特征值分离很好,则用校正方程的正交补可以很好的校正其病态情况,但对于重特征值这种校正效果会很差;三是校正方程的系数矩阵可能是奇异的,尽管校正方程的投影算子把近似特征向量的一个“坏”方向上的分量移除,但是其他的“坏”方向阻止了其余的重特征值的求解,也就是说其他的“坏”方向并没有得到修正,这可能导致 Jacobi-Davidson 方法的收敛速度变得很慢,甚至收敛结果会有较大的偏差;四是对于矩阵的良态密集特征值,用块格式可以达到很高的精度和收敛速度<sup>[2]</sup>。基于以上原因,文献[20]在对块 Jacobi 方法和块 Davidson 方法深入讨论的基础上,提出了块 Jacobi-Davidson 方法。

假定  $\lambda_1^{(k)}, \dots, \lambda_l^{(k)}$  和  $u_1^{(k)}, \dots, u_l^{(k)}$  是  $A$  的  $l$  个最大特征值和相应的特征向量的近似值, 并且  $U_k = [u_1^{(k)}, \dots, u_l^{(k)}]$  是列正交规范矩阵, 则

$$U_k^T A U_k = \Lambda^{(k)} = \text{diag}(\lambda_1^{(k)}, \dots, \lambda_l^{(k)}). \quad (3.1)$$

根据对块 Jacobi 方法和块 Davidson 方法的分析<sup>[20]</sup>, 实际要求的是对当前近似特征向量矩阵  $U_k$  的正交分量矩阵, 为了更好地利用当前的近似特征向量矩阵  $U_k$  的信息, 我们作  $A$  在子空间  $[\text{span}(U_k)]^\perp$  上的正交投影得

$$B_k = (I - U_k U_k^T) A (I - U_k U_k^T), \quad (3.2)$$

则

$$A = B_k + A U_k U_k^T + U_k U_k^T A - U_k \Lambda^{(k)} U_k^T. \quad (3.3)$$

实际上, 我们所要求的是  $A$  的接近于  $\Lambda^{(k)}$  的特征值  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_l)$  和相应的特征向量。因此, 我们希望求正交于  $U_k$  的校正向量矩阵  $T_k$  使得

$$A(U_k + T_k) = (U_k + T_k)\Lambda, \quad (3.4)$$

将 (3.3) 代入 (3.4), 并利用  $B_k U_k = 0$ ,  $U_k^T T_k = 0$ , 可得

$$B_k T_k - T_k \Lambda = -R_k + U_k (\Lambda - \Lambda^{(k)} - U_k^T A T_k), \quad (3.5)$$

其中  $R_k = A U_k - U_k \Lambda^{(k)}$ 。

因为  $U_k^T B_k = 0$ ,  $U_k^T T_k = 0$ ,  $U_k^T R_k = 0$ , 则

$$\Lambda - \Lambda^{(k)} - U_k^T A T_k = 0. \quad (3.6)$$

取  $\Lambda$  为  $\Lambda^{(k)}$ , 则求  $U_k$  的校正向量阵  $T_k$  转化为如下的矩阵方程

$$B_k T_k - T_k \Lambda^{(k)} = -R_k, \text{ 且 } U_k^T T_k = 0. \quad (3.7)$$

记  $T_k = [t_1^{(k)}, \dots, t_l^{(k)}]$ ,  $R_k = [r_1^{(k)}, \dots, r_l^{(k)}]$ , 则 (3.7) 式化为

$$\begin{cases} (B_k - \lambda_i^{(k)} I) t_i^{(k)} = r_i^{(k)} \\ U_k^T t_i^{(k)} = 0 \end{cases} \quad i=1, \dots, l. \quad (3.8)$$

由于

$$(I - U_k U_k^T) t_i^{(k)} = t_i^{(k)}, \quad i=1, \dots, l, \quad (3.9)$$

则 (3.8) 等价于

$$\begin{cases} (I - U_k U_k^T)(A - \lambda_i^{(k)} I)(I - U_k U_k^T) t_i^{(k)} = -r_i^{(k)} \\ U_k^T t_i^{(k)} = 0 \end{cases} \quad (i=1, 2, \dots, l). \quad (3.10)$$

因此, 块 Jacobi-Davidson 方法可描述如下:

### 算法 3.1 块 Jacobi-Davidson 方法

输入: 矩阵  $A$ , 投影子空间最大维数  $m$ , 块大小  $l$ .

输出:  $A$  的  $l$  个最大特征对的近似  $(\theta_i, u_i)$ ,  $1 \leq i \leq l$ .

1. 选取  $n \times l$  的列正交规范矩阵  $V_1$

2. 对  $k=1, 2, \dots$

2.1). 计算  $H_k = V_k^T A V_k$ ;

2.2). 求  $H_k$  的  $l$  个最大特征对,  $(\lambda_i^{(k)}, y_i^{(k)}) \quad (i=1, 2, \dots, l)$ ;

2.3). 计算 Ritz 向量  $u_i^{(k)} = V_k y_i^{(k)} \quad (i=1, 2, \dots, l)$ ;

2.4). 计算残量  $r_i^{(k)} = (A - \lambda_i^{(k)} I) u_i^{(k)} \quad (i=1, 2, \dots, l)$ , 若收敛则停止; 否则进行下一步;

2.5). 求新的修正向量矩阵  $T_k = [t_1^{(k)}, \dots, t_l^{(k)}]$ :

$$\begin{cases} (I - U^{(k)} U^{(k)T})(A - \lambda_i^{(k)} I)(I - U^{(k)} U^{(k)T}) t_i^{(k)} = -r_i^{(k)} \\ U^{(k)T} t_i^{(k)} = 0 \end{cases} \quad (i=1, 2, \dots, l);$$

其中  $U^{(k)} = [u_1^{(k)}, \dots, u_l^{(k)}]$

2.6). 若  $\dim(V_k) \leq m-l$ , 则  $V_{k+1} = [V_k, T_k]$ ; 否则  $V_{k+1} = [u_1^{(k)}, \dots, u_l^{(k)}, T_k]$ .

注:

1. 算法 3.1 的新修正方向是通过求解方程组 (3.10) 得到的, 可以看作是寻求对初始近似特征向量矩阵的块 Jacobi 方法和通过扩张搜索空间求近似特征向量矩阵的块 Davidson 方法的组合。
2. 算法 3.1 块大小为  $l$ , 而输出的特征对也为  $l$  个, 如果实际上需要特征对个数为  $r$  时, 为了保证可以计算出想要的特征对, 必须选择  $l \geq r$ 。
3. 通常情况下, 利用块 JD 算法求得的  $A$  的近似极端特征值和近似特征向量未必一次

就达到满意的精度。为克服这一困难，通常用对应于  $A$  的近似极端特征值的近似特征向量代替  $V_l$  重复块 JD 方法，这个过程称为迭代块 JD 方法。

计算矩阵  $A$  的  $l$  个模最大（或最小）特征值和相应特征向量的迭代块 JD 算法如下：

### 算法 3.2 迭代块 Jacobi-Davidson 方法

- (1) 选取块大小  $l$ ，块 JD 方法执行步数  $m$  和初始列正交规范矩阵  $V_l \in R^{n \times l}$ 。
- (2) 执行  $m$  步块 JD 算法，求得  $\bar{H}_m$ ， $\bar{V}_m$  及  $\bar{H}_m$  的特征对  $(\theta_i, y_i)$ ， $i = 1, 2, \dots, l$   
 计算 Ritz 向量  $u_i = \bar{V}_m y_i$ ， $i = 1, 2, \dots, l$ 。记  $U_m = [u_1, u_2, \dots, u_l]$   
 计算残量  $r_i = (A - \theta_i I)u_i$ ， $i = 1, 2, \dots, l$
- (3) 检查  $r_i$  ( $i = 1, 2, \dots, l$ ) 的收敛性，若满足精度要求，则停止；  
 否则，执行下一步。
- (4) 重新开始算法，令  $V_l = U_m$  转向步 (2)。

## 3.2 动态收缩的块 Jacobi-Davidson 方法

研究算法 3.1 的执行过程，我们可以看出，上述块 Jacobi-Davidson 方法中块的大小为  $l$ ，这便意味着每次迭代产生  $l$  个 Ritz 对，并判断这  $l$  个 Ritz 对是否收敛，只要存在一个不收敛，便要再计算  $l$  个校正向量用于扩张投影子空间或重新开始以执行下一步迭代，直到计算出的  $l$  个 Ritz 对都收敛才停止。实际上，这  $l$  个 Ritz 对的收敛速度通常并不相同，所以不妨假设在第  $s$  步时这  $l$  个 Ritz 对中已经出现了  $q$  ( $q < l$ ) 个收敛的 Ritz 对  $(\theta_i, u_i)$  ( $i = 1, 2, \dots, q$ )。按照上述算法的执行过程便意味着这  $q$  个已经收敛的 Ritz 对仍要参加以后的迭代运算，而计算误差和一些扰动的存在又很有可能导致已经收敛 Ritz 对又变得不收敛，所以又要重新计算，反复如此直到  $l$  个 Ritz 对都收敛。即使后面的迭代不会影响已收敛的 Ritz 对，但这样也使运算过程中增加了很多不必要的计算量。总之，就是这种在迭代过程中产生的已收敛的 Ritz 对仍要参加后续迭代运算的现象导致上述块 Jacobi-Davidson 方法计算量很大，整体收敛效果不好。

因此，为了使已经收敛的 Ritz 对不再参与后续迭代，一种有效的方法便是将已经收敛的特征向量提取出去，并将矩阵  $A$  限制在已收敛的 Ritz 向量的正交补空间上，然后将限制矩阵参加后续迭代，其中限制矩阵为  $\tilde{A} = (I - XX^T)A(I - XX^T)$ ， $X$  为  $n \times t$  列正交规范矩阵，且  $x_i$  ( $i = 1, 2, \dots, t$ ) 为矩阵  $A$  的已经收敛的 Ritz 向量。此种收缩技术易于实现，在很多方法中被很有效的用来计算次特征对。

本文采用动态收缩，其动态性主要体现为及时检测是否存在收敛的 Ritz 对，一旦出现收敛的 Ritz 对，便立即收缩，并适当改变迭代过程中块的大小  $l$  以及重新启动时初始子空间的维数  $p_0$ ，即块的大小和重新开始子空间维数是变化的。算法描述如下：

### 算法 3.3 基本的动态收缩块 Jacobi-Davidson 算法

输入：矩阵  $A$ ，投影子空间最大维数  $m$ ，块大小  $l$ ，初始子空间维数  $p_0$ ，通常取  $p_0 = 2l$ 。

输出： $A$  的  $l$  个最大特征对的近似  $(\theta_i, u_i)$ ， $1 \leq i \leq l$ 。

1) 选取一个  $n \times p_0$  的列正交规范矩阵  $V_1$ ；

2) 对  $k = 1, 2, \dots$

2.1) 将  $A$  投影到收敛特征向量张成子空间的正交补上得到  $\tilde{A}$ ，如果没有收敛的特征向量，则  $\tilde{A} = A$ ；

2.2) 计算  $H_k = V_k^T \tilde{A} V_k$ ；

2.3) 求  $H_k$  的  $l$  个最大特征对  $(\lambda_i^{(k)}, y_i^{(k)})$  ( $i = 1, 2, \dots, l$ )；

2.4) 计算 Ritz 向量  $u_i^{(k)} = V_k y_i^{(k)}$  ( $i = 1, 2, \dots, l$ )，并形成  $U^{(k)} = [u_1^{(k)}, \dots, u_l^{(k)}]$ ；

2.5) 计算残量  $r_i^{(k)} = (\tilde{A} - \lambda_i^{(k)} I) u_i^{(k)}$  ( $i = 1, 2, \dots, l$ )，逐个检验收敛性。不妨设  $q$  是收敛 Ritz 对的个数，将此  $q$  个收敛的近似特征对保存，并令  $l := l - q$ ， $p_0 = 2l$ 。若  $l = 0$ ，停止；否则，进行下一步；

2.6) 求新的修正向量矩阵  $T_k = [t_1^{(k)}, \dots, t_l^{(k)}]$ ：

$$\begin{cases} (I - U^{(k)} U^{(k)T})(\tilde{A} - \lambda_i^{(k)} I)(I - U^{(k)} U^{(k)T}) t_i^{(k)} = -r_i^{(k)} & (i = 1, 2, \dots, l), \\ U^{(k)T} t_i^{(k)} = 0 \end{cases}$$

其中  $\lambda_i^{(k)}$ ， $r_i^{(k)}$  对应于不收敛的 Ritz 对；

2.7) 若  $\dim(V_k) \leq m - l$ ，则  $V_{k+1} = MGS(V_k, T_k)$ ；否则在 Ritz 向量中选择  $p_0$  个向量形成重新开始子空间的正交基。

注：

1. 初始矩阵  $V_1$  的选取不当可引起算法失效。如果  $V_1$  的列的任何线性组合都不包含在对应于特征值  $\lambda_1, \dots, \lambda_r$  ( $r$  为要求特征值的个数) 的特征子空间上，则上述算法不能求得特征值  $\lambda_1, \dots, \lambda_r$  及其相应的特征向量  $x_1, \dots, x_r$ ，不过在实际计算中，出现这种情况的机会是极小的。
2. 投影子空间最大维数  $m$ ，块大小  $l$ ，初始子空间维数  $p_0$  的具体选取见第五章相关分析。
3. 上述算法存在和算法 3.1 注 2 相同的情况，即块大小为  $l$ ，而输出的特征对也为  $l$  个，如果实际上需要特征对个数为  $r$  时，为了保证可以计算出想要的特征对，必须选择  $l \geq r$ 。为了使算法更灵活，使块大小的确定不完全依赖待求特征值的个数  $r$  而可以被科学的选取，给出下述算法。

### 算法 3.4 动态收缩的块 Jacobi-Davidson 算法

输入：矩阵  $A$ ，待求特征值个数  $r$ ，投影子空间最大维数  $m$ ，块大小  $l$ ，初始子空间维数  $p_0$ ，通常取  $p_0 = 2l$ 。

输出： $A$  的  $r$  个最大特征对的近似  $(\theta_i, u_i)$ ， $1 \leq i \leq r$ 。

1) 选取一个  $n \times p_0$  的列正交规范矩阵  $V_1$ ；

2) 对  $k = 1, 2, \dots$

2.1) 将  $A$  投影到收敛的  $p$  个特征向量张成子空间的正交补上得到  $\tilde{A}$ ，如果没有收敛的特征向量即  $p = 0$ ，则  $\tilde{A} = A$ ；

2.2) 计算  $H_k = V_k^T \tilde{A} V_k$ ；

2.3) 求  $H_k$  的  $r - p$  个最大特征对  $(\lambda_i^{(k)}, y_i^{(k)})$  ( $i = 1, 2, \dots, r - p$ )；

2.4) 计算 Ritz 向量  $u_i^{(k)} = V_k y_i^{(k)}$  ( $i = 1, 2, \dots, r - p$ )，并形成  $U^{(k)} = [u_1^{(k)}, \dots, u_{r-p}^{(k)}]$ ；

2.5) 计算残量  $r_i^{(k)} = (\tilde{A} - \lambda_i^{(k)} I) u_i^{(k)}$  ( $i = 1, 2, \dots, m - r$ )，逐个检验收敛性。不妨设  $q$  是收敛 Ritz 对的个数，将此  $q$  个收敛的近似特征对保存，并令  $p = p + q$ 。如果  $l < r - p$ ， $l$  不变；如果  $l \geq r - p$ ， $l := r - p$ 。并令  $p_0 = 2l$ 。若  $l = 0$ ，停止；否则，进行下一步；

2.6) 求新的修正向量矩阵  $T_k = [t_1^{(k)}, \dots, t_l^{(k)}]$ ：

$$\begin{cases} (I - U^{(k)} U^{(k)T})(\tilde{A} - \lambda_i^{(k)} I)(I - U^{(k)} U^{(k)T}) t_i^{(k)} = -r_i^{(k)} & (i = 1, 2, \dots, l), \\ U^{(k)T} t_i^{(k)} = 0 \end{cases}$$

其中  $\lambda_i^{(k)}$ ， $r_i^{(k)}$  对应于不收敛的 Ritz 对；

2.7) 若  $\dim(V_k) \leq m - l$ ，则  $V_{k+1} = MGS(V_k, T_k)$ ；否则在 Ritz 向量中选择  $p_0$  个向量形成重新开始子空间的正交基。

注：该算法块大小  $l$  的选取与待求特征对个数  $r$  没有必然的关系，使块大小的确定不完全依赖待求特征值的个数  $r$ 。



## 第四章 调和块 Jacobi-Davidson 方法

内部特征值问题产生于许多实际问题中，近年来有着越来越多的应用。例如在分析潮汐运动时需要计算位于谱集内部的一些特征值；流体力学中的稳定性分析，电路系统模拟和天气预报模型的建立等都需要对某些内部特征值给出估计。

Morgan<sup>[33]</sup>提出一种特殊的斜投影方法用于求解大规模矩阵部分内部特征对。Paige 等人在 [35] 中将这种方法得到的近似特征对称为调和 Ritz 对，称这种特殊的斜投影方法为调和 Rayleigh-Ritz 投影方法，该方法将  $A$  的内部特征值转化为  $(A - \tau I)^{-1}$  ( $\tau$  为某一个目标数) 的端部特征值并且避免了对  $(A - \tau I)^{-1}$  进行直接计算，这是对计算大规模矩阵内部特征对问题的一大突破。调和 Rayleigh-Ritz 过程<sup>[33,35]</sup>是求解大规模矩阵部分特征对特别是部分内部特征对最常用的投影方法。而且调和 Rayleigh-Ritz 过程中的子空间的选取比较灵活，可以选取 Krylov 子空间，也可以选取其它含有更多待求特征向量信息的子空间。

事实证明，由块 Jacobi-Davidson 方法扩充方式得到的投影子空间包含较丰富的需求特征向量的信息。而调和 Rayleigh-Ritz 过程又可以一次性求出多个近似特征值和相应的特征向量。其次，块方法对密集特征值和重特征值问题更有效。所以，本章将块 Jacobi-Davidson 外部迭代应用于调和 Rayleigh-Ritz 过程，提出了调和块 Jacobi-Davidson 方法。首先，在本章第一节将简要介绍调和 Rayleigh-Ritz 方法，第二节根据块 Jacobi-Davidson 方法扩充子空间的方式，推导出调和块 Jacobi-Davidson 方法的校正方程，并给出调和块 Jacobi-Davidson 算法描述。

### 4.1 调和 Rayleigh-Ritz 方法

考虑如下的矩阵特征值问题

$$A\varphi_i = \lambda_i \varphi_i,$$

其中  $A$  是大型的  $n$  阶矩阵， $(\lambda_i, \varphi_i)$  是  $A$  的特征对 ( $\|\varphi_i\| = 1$ )。假设要计算矩阵  $A$  在  $\tau$  附近的  $l$  个特征对  $(\lambda_i, \varphi_i)$ ， $i = 1, 2, \dots, l$ 。

设  $K$  是  $m$  维子空间，并且  $\{v_i\}_{i=1}^m$  是它的一组标准正交基，令  $V_m = [v_1, v_2, \dots, v_m]$ ， $w_i = (A - \tau I)v_i$ ， $i = 1, 2, \dots, l$ ， $W_m = (A - \tau I)V_m = [w_1, w_2, \dots, w_m]$ ，子空间  $L = \text{span}\{(A - \tau I)v_1, (A - \tau I)v_2, \dots, (A - \tau I)v_m\} = \text{span}\{w_1, w_2, \dots, w_m\}$ ，则调和 Rayleigh-Ritz 方法为用满足

$$\begin{cases} \tilde{\varphi}_i \in K \\ (A - \tau I)\tilde{\varphi}_i - \tilde{\lambda}_i \tilde{\varphi}_i \perp L \end{cases} \quad (4.1)$$

的  $(\tilde{\lambda}_i + \tau, \tilde{\varphi}_i)$ ， $i = 1, 2, \dots, l$  作为  $A$  的近似特征对，称  $\tilde{\lambda}_i + \tau$  为调和 Ritz 值， $\tilde{\varphi}_i$  为调和 Ritz 向

量。\$(\tilde{\lambda}\_i + \tau, \tilde{\varphi}\_i)\$ 为 \$A\$ 在子空间 \$K\$ 上的调和 Ritz 对，它们是广义特征值问题

$$\begin{cases} V_m^T (A - \tau I)^T (A - \tau I) V_m y_i = \tilde{\lambda}_i V_m^T (A - \tau I)^T V_m y_i \\ \tilde{\varphi}_i = V_m y_i \end{cases} \quad (4.2)$$

的解。记 \$H\_1 = V\_m^T (A - \tau I)^T (A - \tau I) V\_m = W\_m^T W\_m\$, \$H\_2 = V\_m^T (A - \tau I)^T V\_m = W\_m^T V\_m\$, 则 (4.2) 为

$$\begin{cases} H_1 y_i = \tilde{\lambda}_i H_2 y_i \\ \tilde{\varphi}_i = V_m y_i \end{cases} \quad (4.3)$$

若假设矩阵 \$H\_2\$ 可逆，则关系式 (4.3) 即为特征问题

$$\begin{cases} H_2^{-1} H_1 y_i = \tilde{\lambda}_i y_i \\ \tilde{\varphi}_i = V_m y_i \end{cases} \quad (4.4)$$

\$H\_2^{-1} H\_1\$ 称为矩阵 \$A\$ 在子空间 \$K\$ 上的调和 Rayleigh 商矩阵，\$(\tilde{\lambda}\_i, y\_i)\$ 为 \$H\_2^{-1} H\_1\$ 的特征对。

实际上，我们可以廉价计算调和 Ritz 向量的 Rayleigh 商作为所求特征值的近似。\$\tilde{\varphi}\_i\$ 的 Rayleigh 商可以按下式廉价计算：

$$\rho_i = \tilde{\varphi}_i^T A \tilde{\varphi}_i = y_i^T V_m^T A V_m y_i = y_i^T H_2^T y_i + \tau \quad (4.5)$$

它不仅比调和 Ritz 值一般更精确，而且更可靠。贾<sup>[50]</sup>证明了，当目标点离所求的特征值很近时，该特征值会丢失掉，此时调和 Ritz 值无意义，但只要调和 Ritz 向量是好的近似，则 Rayleigh 商必然是特征值好的近似。因此，应该使用后者作为近似特征值。

#### 算法 4.1 调和 Rayleigh-Ritz 方法

- 1) 开始：选定 \$m\$ 维子空间 \$K\$ 的一组标准正交基 \$V\_m\$；
- 2) 形成小矩阵：\$H\_1 = V\_m^T (A - \tau I)^T (A - \tau I) V\_m\$, \$H\_2 = V\_m^T (A - \tau I)^T V\_m\$；
- 3) 计算调和 Ritz 对：解广义特征值问题 \$H\_1 y\_i = \tilde{\theta}\_i H\_2 y\_i\$，按需要选出 \$l\$ 个近似特征对 \$(\tilde{\theta}\_i + \tau, \tilde{u}\_i)\$，则 \$\tilde{\theta}\_i + \tau\$ 为调和 Ritz 值，\$\tilde{u}\_i\$ 为调和 Ritz 向量；
- 4) 计算近似特征值：按公式 \$\rho\_i = \tilde{u}\_i^T A \tilde{u}\_i = y\_i^T V\_m^T A V\_m y\_i = y\_i^T H\_2^T y\_i + \tau\$ 计算调和 Ritz 向量 \$\tilde{u}\_i\$ 的 Rayleigh 商 \$\rho\_i\$ 作为要求特征值 \$\theta\_i\$ 的近似；
- 5) 检测收敛：计算 \$l\$ 个近似特征对 \$(\rho\_i, \tilde{u}\_i)\$ 的残量。

调和 Rayleigh-Ritz 方法在数学上等价于对矩阵 \$(A - \tau I)^{-1}\$ 运用 Rayleigh-Ritz 方法。贾<sup>[50]</sup>建立了调和 Ritz 值与调和 Ritz 向量的误差界，指出在调和 Ritz 值收敛到特征值时，调和 Ritz 向量却经常出现不收敛的情况。但使用调和 Ritz 值计算得到的精化调和 Ritz 向量具有同调和 Ritz 值相同的收敛性质。

调和 Rayleigh-Ritz 过程避免了对 \$(A - \tau I)^{-1}\$ 进行直接求解而将 \$A\$ 的内部特征值转化为 \$(A - \tau I)^{-1}\$ 的端部特征值。但当 \$A\$ 的需求特征值分布比较密集时，必须选择出合适的目标值 \$\tau\$ 才

可能将它们转换为分离较好的特征值,如果是重特征,该方法的有效性会有所降低。而且当 $\tau$ 非常靠近或者等于 $A$ 的特征值时,调和 Rayleigh-Ritz 的有效性也会有所降低,但是该过程还是有益的<sup>[34]</sup>。关于 $\tau$ 的选取对调和 Ritz 向量的影响以及调和 Ritz 值和对应的 Ritz 向量的误差分析可见文献[51]。

在理论上,调和 Rayleigh-Ritz 过程等价于对矩阵 $(A - \tau I)^{-1}$ 运用 Rayleigh-Ritz 方法。对 Rayleigh-Ritz 过程,若投影子空间中有所求特征向量的信息越丰富,则在该空间上的 Ritz 值的近似程度就越好<sup>[17]</sup>。因此对调和 Rayleigh-Ritz 过程同样地有,如果投影子空间中有所求特征向量的信息越丰富,则在该子空间上的调和 Ritz 值的近似程度越好。

## 4.2 调和块 Jacobi-Davidson 方法

假定 $\theta_1^{(k)}, \dots, \theta_l^{(k)}$ 和 $u_1^{(k)}, \dots, u_l^{(k)}$ 分别是由调和 Rayleigh-Ritz 方法在投影子空间 $V_k$ 上求得的 $A$ 的最靠近 $\tau$ 的 $l$ 个特征值和相应的特征向量的近似值,并且 $U_k = [u_1^{(k)}, \dots, u_l^{(k)}]$ 是列正交规范矩阵,则

$$U_k^T A U_k = \Lambda^{(k)} = \text{diag}(\theta_1^{(k)}, \dots, \theta_l^{(k)}). \quad (4.6)$$

根据块 Jacobi-Davidson 方法对投影子空间扩充的方式,即用当前近似特征向量矩阵 $U_k$ 的正交分量矩阵来扩充投影子空间。为了更好地利用当前的近似特征向量矩阵 $U_k$ 的信息,我们作 $A$ 在子空间 $[\text{span}(U_k)]^\perp$ 上的正交投影

$$B_k = (I - U_k U_k^T) A (I - U_k U_k^T), \quad (4.7)$$

$$\text{则} \quad A = B_k + A U_k U_k^T + U_k U_k^T A - U_k \Lambda^{(k)} U_k^T. \quad (4.8)$$

我们所要求的是 $A$ 的接近于 $\Lambda^{(k)}$ 的特征值 $\Lambda = \text{diag}(\theta_1, \dots, \theta_l)$ 和相应的特征向量。因此,我们希望求正交于 $U_k$ 的校正向量矩阵 $T_k$ 使得

$$A(U_k + T_k) = (U_k + T_k)\Lambda, \quad (4.9)$$

将(4.8)代入(4.9),并利用 $B_k U_k = 0$ ,  $U_k^T T_k = 0$ , 得

$$B_k T_k - T_k \Lambda = -R_k + U_k (\Lambda - \Lambda^{(k)} - U_k^T A T_k), \quad (4.10)$$

其中 $R_k = A U_k - U_k \Lambda^{(k)}$ 。

因为 $U_k^T B_k = 0$ ,  $U_k^T T_k = 0$ ,  $U_k^T R_k = 0$ , 则

$$\Lambda - \Lambda^{(k)} - U_k^T A T_k = 0, \quad (4.11)$$

取 $\Lambda$ 为 $\Lambda^{(k)}$ , 则求 $U_k$ 的校正向量矩阵 $T_k$ 转化为如下的矩阵方程

$$B_k T_k - T_k \Lambda^{(k)} = -R_k, \text{ 且 } U_k^T T_k = 0. \quad (4.12)$$

记 $T_k = [t_1^{(k)}, \dots, t_l^{(k)}]$ ,  $R_k = [r_1^{(k)}, \dots, r_l^{(k)}]$ , 则(4.12)式化为

$$\begin{cases} (B_k - \theta_i^{(k)} I) t_i^{(k)} = r_i^{(k)} & i = 1, \dots, l. \\ U_k^T t_i^{(k)} = 0 \end{cases} \quad (4.13)$$

由于

$$(I - U_k U_k^T) t_i^{(k)} = t_i^{(k)}, i = 1, \dots, l, \quad (4.14)$$

则 (4.13) 等价于

$$\begin{cases} (I - U_k U_k^T)(A - \theta_i^{(k)} I)(I - U_k U_k^T) t_i^{(k)} = -r_i^{(k)} \\ U_k^T t_i^{(k)} = 0 \end{cases} \quad (i = 1, 2, \dots, l). \quad (4.15)$$

因此, 调和块 Jacobi-Davidson 方法可描述如下:

#### 算法 4.2 调和块 Jacobi-Davidson 算法

输入: 矩阵  $A$ , 投影子空间最大维数  $m$ , 目标数  $\tau$ , 待求的特征对的个数  $l$ , 初始子空间维数  $p_0$ , 通常取  $p_0 = 2l$ 。

输出:  $A$  的最靠近  $\tau$  的  $l$  个特征对的近似  $(\theta_i, u_i)$ ,  $i = 1, 2, \dots, l$ 。

1) 选取  $n \times p_0$  的列正交规范矩阵  $V_1$ ,  $W_1 = (A - \tau I)V_1$ ;

2) 对  $k = 1, 2, \dots$

2.1) 根据  $V_k, W_k$  用算法 4.1 计算得到  $l$  个近似特征对  $(\theta_i^{(k)}, u_i^{(k)}) = (\theta_i^{(k)}, V y_i^{(k)})$

$i = 1, 2, \dots, l$ ;

2.2) 计算残量  $r_i^{(k)} = (A - \theta_i^{(k)} I)u_i^{(k)}$  ( $i = 1, 2, \dots, l$ ), 若收敛则停止; 否则进行下一步;

2.3) 求新的修正向量矩阵  $T_k = [t_1^{(k)}, \dots, t_l^{(k)}]$ :

$$\begin{cases} (I - U^{(k)} U^{(k)T})(A - \theta_i^{(k)} I)(I - U^{(k)} U^{(k)T}) t_i^{(k)} = -r_i^{(k)} \\ U^{(k)T} t_i^{(k)} = 0 \end{cases} \quad (i = 1, 2, \dots, l),$$

其中  $U^{(k)} = [u_1^{(k)}, \dots, u_l^{(k)}]$ ;

2.4) 若  $\dim(V) \leq m - l$ , 则  $V_{k+1} = MGS(V_k, T_k)$ ,  $W_{k+1} = (A - \tau I)V_{k+1}$ ;

否则  $V_{k+1} = MGS(u_1^{(k)}, \dots, u_l^{(k)}, T_k)$ ,  $W_{k+1} = (A - \tau I)V_{k+1}$ 。

算法 4.2 虽然能很好的计算出最靠近目标数  $\tau$  的  $l$  个近似特征对, 但是也存在和算法 3.1 相同的问题, 即迭代过程中产生的已经收敛的 Ritz 对有可能参加后续的迭代运算。所以, 为了提高算法 4.2 的整体收敛速度, 我们将收缩技巧应用在算法 4.2 中, 并考虑到块大小的选取与待求特征值个数  $r$  的情况, 得到如下的动态收缩的调和块 Jacobi-Davidson 方法。

#### 算法 4.3 动态收缩的调和块 Jacobi-Davidson 算法

输入: 矩阵  $A$ , 投影子空间最大维数  $m$ , 目标数  $\tau$ , 待求的特征对的个数  $r$ , 块大小  $l$ 。

输出:  $A$  的最靠近  $\tau$  的  $r$  个特征对的近似  $(\theta_i, u_i)$ ,  $i = 1, 2, \dots, r$ 。

1) 选取一个  $n \times p_0$  的列正交规范矩阵  $V_1$ , 并计算  $W_1 = (A - \tau I)V_1$ ;

2) 对  $k = 1, 2, \dots$

2.1) 将  $A$  投影到收敛的  $p$  个特征向量子空间的正交补上得到  $\tilde{A}$ , 如果没有收敛的特征向量即  $p = 0$  则  $\tilde{A} = A$ ;

2.2) 根据  $V_k, W_k$  用算法 4.1 计算得到  $l$  个近似特征对  $(\theta_i^{(k)}, u_i^{(k)}) = (\theta_i^{(k)}, V y_i^{(k)})$ ,  $i = 1, 2, \dots, r - p$ ;

2.3) 计算残量  $r_i^{(k)} = (\tilde{A} - \theta_i^{(k)} I) u_i^{(k)}$  ( $i = 1, 2, \dots, r - p$ ), 逐个检验收敛性。不妨设  $q$  是收敛 Ritz 对的个数, 将此  $q$  个收敛的近似特征对保存, 并令  $p = p + q$ 。如果  $l < r - p$ ,  $l$  不变; 如果  $l \geq r - p$ ,  $l := r - p$ 。并令  $p_0 = 2l$ 。若  $l = 0$ , 停止; 否则, 进行下一步;

2.4) 求新的修正向量矩阵  $T_k = [t_1^{(k)}, \dots, t_l^{(k)}]$ :

$$\begin{cases} (I - U^{(k)} U^{(k)T})(\tilde{A} - \theta_i^{(k)} I)(I - U^{(k)} U^{(k)T}) t_i^{(k)} = -r_i^{(k)} & (i = 1, 2, \dots, l), \\ U^{(k)T} t_i^{(k)} = 0 \end{cases}$$

其中  $U^{(k)} = [u_1^{(k)}, \dots, u_l^{(k)}]$ , 且  $\theta_i^{(k)}, r_i^{(k)}$  对应于不收敛的 Ritz 对;

2.5) 若  $\dim(V) \leq m - l$ , 则  $V_{k+1} = MGS(V_k, T_k)$ ,  $W_{k+1} = (\tilde{A} - \tau I) V_{k+1}$ ;

否则  $V_{k+1} = MGS(u_1^{(k)}, \dots, u_l^{(k)}, T_k)$ ,  $W_{k+1} = (\tilde{A} - \tau I) V_{k+1}$ 。

## 第五章 算法的实现

### 5.1 校正方程的求解

#### 5.1.1 等价增广形式的预理解法

无论是在块 Jacobi-Davidson 方法中还是在改进的块 Jacobi-Davidson 方法中,为了得到有效扩充子空间的修正向量,都需要求解如下形式的方程组:

$$\begin{cases} (I - UU^T)(A - \theta_i I)(I - UU^T)t_i = -r_i \\ U^T t_i = 0 \end{cases} \quad (i=1, 2, \dots, l), \quad (5.1)$$

其中  $U^T U = I$ 。

于是,有效求解方程组 (5.1) 便是块 Jacobi-Davidson 类方法的关键。但由于方程的系数矩阵随着逼近程度的提高,有可能变得病态,所以这里需要用到预处理技术。一般情况下,预处理算子的选取都是通过将  $A - \theta_i I$  的近似矩阵  $K_i$  投影到  $[\text{span}(U)]^\perp$  上实现的,即预处理算子为  $\tilde{K}_i \equiv (I - UU^T)K_i(I - UU^T)$ 。然而,由于投影的存在,使得预处理效果不明显。为了避免投影,我们使用校正方程的等价增广形式将校正方程转化为经典的 saddle point 问题,并对其使用适当的预处理技术。

**定理 5.1** 若记

$$(a) \quad t_i = (I - UU^T)t_i, \quad (b) \quad F_i t_i = -r_i, \quad (5.2)$$

其中  $F_i = (I - UU^T)(A - \theta_i I)(I - UU^T)$ ,  $i=1, 2, \dots, l$  且  $U^T U = I$ , 则 (5.2) 式等价于

$$\begin{bmatrix} A - \theta_i I & U \\ U^T & 0 \end{bmatrix} \begin{bmatrix} t_i \\ f_i \end{bmatrix} = \begin{bmatrix} -r_i \\ 0 \end{bmatrix}, i=1, \dots, l, \quad (5.3)$$

即  $t_i$  是 (5.2) 式的解当且仅当  $t_i$  是 (5.3) 式的解。

证明: “ $\Rightarrow$ ”若  $t_i = (I - UU^T)t_i$ ,  $F_i t_i = -r_i$ , 则

$$(I - UU^T)(A - \theta_i I)(I - UU^T)t_i = -r_i,$$

即

$$(I - UU^T)(A - \theta_i I)t_i = -r_i,$$

$$(A - \theta_i I)t_i - UU^T(A - \theta_i I)t_i = -r_i.$$

令  $f_i = -U^T(A - \theta_i I)t_i$ ,

则

$$\begin{cases} (A - \theta_i I)t_i + Uf_i = -r_i \\ U^T t_i = 0 \end{cases},$$

即 
$$\begin{bmatrix} A - \theta_i I & U \\ U^T & 0 \end{bmatrix} \begin{bmatrix} t_i \\ f_i \end{bmatrix} = \begin{bmatrix} -r_i \\ 0 \end{bmatrix}.$$

“ $\Leftarrow$ ”若  $\begin{bmatrix} A - \theta_i I & U \\ U^T & 0 \end{bmatrix} \begin{bmatrix} t_i \\ f_i \end{bmatrix} = \begin{bmatrix} -r_i \\ 0 \end{bmatrix}$ , 则

$$\begin{cases} (A - \theta_i I)t_i + Uf = -r_i \\ U^T t_i = 0 \end{cases}.$$

因为  $U^T r_i = 0$ , 则

$$(I - UU^T)[(A - \theta_i I)t_i + Uf] = -r_i,$$

即 
$$(I - UU^T)(A - \theta_i I)t_i = -r_i,$$

由  $U^T t_i = 0$ , 则  $t_i = (I - UU^T)t_i$ , 且  $F_i t_i = -r_i$ 。

**引理 5.1** 对于  $Z \in R^{n \times p}$  和  $Y \in R^{n \times p}$ , 有

$$\begin{bmatrix} I_n & Y \\ Z^T & 0 \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ Z^T & I_p \end{bmatrix} \begin{bmatrix} I_n & Y \\ 0 & -V \end{bmatrix}, \quad (5.4)$$

其中  $V \equiv Z^T Y$ , 当且仅当  $Z^T Y$  非奇异时,  $\begin{bmatrix} I_n & Y \\ Z^T & 0 \end{bmatrix}$  可逆。

引理 5.1 显然成立。

**定理 5.2** 如果  $K_i$  是  $(A - \theta_i I)$  的近似矩阵且易于求逆, 那么

$$\begin{bmatrix} K_i & U \\ U^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} I_n & -K_i^{-1}U \\ 0 & I_p \end{bmatrix} \begin{bmatrix} I_n & 0 \\ (U^T K_i^{-1}U)^{-1}U^T & -(U^T K_i^{-1}U)^{-1} \end{bmatrix} \begin{bmatrix} K_i^{-1} & 0 \\ 0 & I_p \end{bmatrix}, \quad (5.5)$$

且

$$\begin{aligned} & \begin{bmatrix} K_i & U \\ U^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} A - \theta_i I & U \\ U^T & 0 \end{bmatrix} = \\ & \begin{bmatrix} I_n & -K_i^{-1}U \\ 0 & I_p \end{bmatrix} \begin{bmatrix} I_n & 0 \\ (U^T K_i^{-1}U)^{-1}U^T & -(U^T K_i^{-1}U)^{-1} \end{bmatrix} \begin{bmatrix} K_i^{-1}(A - \theta_i I) & K_i^{-1}U \\ U^T & 0 \end{bmatrix}. \end{aligned} \quad (5.6)$$

证明: 因为

$$\begin{bmatrix} K_i & U \\ U^T & 0 \end{bmatrix} = \begin{bmatrix} K_i & 0 \\ 0 & I_p \end{bmatrix} \begin{bmatrix} I_n & K_i^{-1}U \\ U^T & 0 \end{bmatrix},$$

则 
$$\begin{bmatrix} K_i & U \\ U^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} I_n & K_i^{-1}U \\ U^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} K_i^{-1} & 0 \\ 0 & I_p \end{bmatrix}. \quad (5.7)$$

又由引理 5.1 得

$$\begin{bmatrix} I_n & Y \\ Z^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} I_n & Y \\ 0 & -V \end{bmatrix}^{-1} \begin{bmatrix} I_n & 0 \\ -Z^T & I_p \end{bmatrix} = \begin{bmatrix} I_n & YV^{-1} \\ 0 & -V^{-1} \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -Z^T & I_p \end{bmatrix},$$

所以

$$\begin{aligned} \begin{bmatrix} I_n & K_i^{-1}U \\ U^T & 0 \end{bmatrix}^{-1} &= \begin{bmatrix} I_n & K_i^{-1}U (U^T K_i^{-1}U)^{-1} \\ 0 & -(U^T K_i^{-1}U)^{-1} \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -U^T & I_p \end{bmatrix} \\ &= \begin{bmatrix} I_n & -K_i^{-1}U \\ 0 & I_p \end{bmatrix} \begin{bmatrix} I_n & 0 \\ 0 & -(U^T K_i^{-1}U)^{-1} \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -U^T & I_p \end{bmatrix} \quad (5.8) \\ &= \begin{bmatrix} I_n & -K_i^{-1}U \\ 0 & I_p \end{bmatrix} \begin{bmatrix} I_n & 0 \\ (U^T K_i^{-1}U)^{-1}U^T & (U^T K_i^{-1}U)^{-1} \end{bmatrix} \end{aligned}$$

于是, 将 (5.8) 式代入 (5.7) 式便得到 (5.5) 式, 从而可得 (5.5) 式。

因此, 根据定理 5.2 我们可以得到左预处理的增广校对方程如下:

$$\begin{aligned} \begin{bmatrix} I_n & -K_i^{-1}U \\ 0 & I_p \end{bmatrix} \begin{bmatrix} I_n & 0 \\ (U^T K_i^{-1}U)^{-1}U^T & -(U^T K_i^{-1}U)^{-1} \end{bmatrix} \begin{bmatrix} K_i^{-1}(A - \theta_i I) & K_i^{-1}U \\ U^T & 0 \end{bmatrix} \begin{bmatrix} r_i \\ f_i \end{bmatrix} \\ = \begin{bmatrix} -K_i^{-1}r_i + (K_i^{-1}U)(U^T K_i^{-1}U)^{-1}(U^T K_i^{-1})r_i \\ -(U^T K_i^{-1}U)^{-1}U^T K_i^{-1}r_i \end{bmatrix} \quad (5.9) \end{aligned}$$

那么, 在算法 3.1、3.3、3.5 以及算法 4.2、4.3 的实现中, 为了更有效地得到扩充子空间的修正向量, 将求解校对方程的步骤都可以用求解 (5.9) 式来替换。而对于 (5.9) 式的求解, 我们可以使用 Krylov 子空间方法求解, 如 GMRES 方法, CG 方法等。对于预处理矩阵  $K_i$  的选取, 存在多种形式, 比如对角预处理、不完全分解预处理、SSOR 预处理等。另外, (5.9) 中涉及到对  $(U^T K_i^{-1}U)$  求逆, 因为其阶数很小, 所以比较容易。而对  $K_i$  的求逆, 因为它是预处理矩阵, 所以也比较容易。

### 5.1.2 对称正定情况下的 CG 解法

Jacobi-Davidson 类方法的核心是通过求解校对方程扩充搜索子空间, 事实上, 正如文献[52]所指出, 校对方程无需精确求解。只需少数的迭代和适中的精度求解校对方程, 便能够使得 Jacobi-Davidson 类方法有效收敛。但是通常并不知道校对方程的求解达到怎样的精度便能够保证 Jacobi-Davidson 类方法全局有效。

文献[25]利用共轭梯度法求解 Jacobi-Davidson 方法的校对方程, 并根据 Jacobi-Davidson 方法和共轭梯度法的性质建立了校对方程的残量与外部过程收敛性的关系。这一关系的建立, 使得在内部迭代过程中, 可以随时监控外部的收敛情况, 从而能够建立合理的停止标准, 在适当的时候停止内迭代, 避免不必要的计算量, 并且保证 Jacobi-Davidson 方法的有效性。



本节将用共轭梯度法求解块 Jacobi-Davidson 类方法的校对方程 (5.1) 式, 并期望建立内迭代中校对方程残量与外部过程收敛性的关系, 推导出理论结果, 建立出合理的停止标准, 有效地保证块 Jacobi-davidson 方法的全局表现。

我们在本节集中分析由以下算法给出的简化 BJD 过程, 在该过程中  $u_i^{(k+1)}$  直接由  $u_i^{(k)}$  加上修正向量  $t_i^{(k)}$  得到。因为完整的 BJD 方法在一般情况下期望是较好的, 所以关于下面简化方法的相关分析可以看作是相对于完整版本的最坏情况的估计。

#### 算法 5.1 简化 block Jacobi-Davidsons 方法

·初始化 :

选取  $n \times l$  列正交规范矩阵  $V^{(0)}$ ,  $V^{(0)} = [v_1^{(0)}, \dots, v_l^{(0)}]$ ;

·迭代 ( $m=1, 2, \dots$ ):

1. 计算  $H_m = V^{(m)T} A V^{(m)}$ ;
2. 求  $H_m$  的  $l$  个最大特征对,  $(\lambda_i^{(m)}, y_i^{(m)})$  ( $i=1, 2, \dots, l$ );
3. 计算 Ritz 向量  $u_i^{(m)} = V^{(m)} y_i^{(m)}$  ( $\theta_i^{(m)} = (u_i^{(m)}, A u_i^{(m)}) = \lambda_i^{(m)}$ ) ( $i=1, 2, \dots, l$ );
4. 计算残量  $r_i^{(m)} = (A - \theta_i^{(m)} I) u_i^{(m)}$ , 若收敛则停止 ( $i=1, 2, \dots, l$ );

否则进行下一步;

5. 选取  $\eta_i$  使得  $(A - \eta_i I)$  正定 (注: 选取方法参见文献[47]);

6. 近似地求解  $t_i^{(m)}$  ( $i=1, 2, \dots, l$ )

$$\begin{cases} (I - U^{(m)} U^{(m)T})(A - \eta_i I)(I - U^{(m)} U^{(m)T}) t_i^{(m)} = -r_i^{(m)} \\ U^{(m)T} t_i^{(m)} = 0 \end{cases}; \quad (5.10)$$

7.  $u_i^{(m+1)} = \frac{u_i^{(m)} + t_i^{(m)}}{\|u_i^{(m)} + t_i^{(m)}\|}$  ( $i=1, 2, \dots, l$ ),

令  $U^{(m+1)} = [u_1^{(m+1)}, \dots, u_l^{(m+1)}]$ , 正交化得到  $\bar{U}^{(m+1)}$ ;

8.  $V^{(m+1)} = \bar{U}^{(m+1)}$ 。

上述算法的校对方程为 (5.10), 通常情况下为了迭代地求解此校对方程, 寻找一个好的预条件子很必要。因为投影矩阵的预处理不明显, 所以通过以下方法建立投影矩阵  $A_{\eta, U} = (I - U U^T)(A - \eta I)(I - U U^T)$  的预条件 (为了表述方便, 去掉一些下标): 即首先对  $(A - \eta I)$  建立一个预条件子  $K$ , 然后对  $K$  应用同样的投影, 从而产生如下形式的预条件:

$$M_U = (I - U U^T) K (I - U U^T)。$$

于是预处理的校对方程变为 (为了表述方便, 去掉一些下标)

$$\begin{cases} (I - U U^T) K (I - U U^T) t = g \\ U^T t = 0 \end{cases}, \quad (5.11)$$

其中  $U$  为列正交规范向量矩阵,  $K$  对称正定。

若记

$$Y = K^{-1}U,$$

$$H = U^T Y,$$

则关于预处理的校正方程 (5.11) 的解可由下面的引理得到。

**引理 5.2** 如果  $H$  非奇异, 则有

$$(I - YH^{-1}U^T)(I - UU^T) = (I - UU^T), \quad (5.12)$$

$$(I - UU^T)(I - YH^{-1}U^T) = (I - YH^{-1}U^T), \quad (5.13)$$

$$(I - YH^{-1}U^T)K^{-1}(I - UU^T) = (I - YH^{-1}U^T)K^{-1}. \quad (5.14)$$

**证明:** 首先证明 (5.12) 式:

$$\begin{aligned} (I - YH^{-1}U^T)(I - UU^T) &= (I - YH^{-1}U^T) - (I - YH^{-1}U^T)UU^T \\ &= (I - YH^{-1}U^T) - (UU^T - YH^{-1}U^T) \\ &= (I - UU^T) \end{aligned}$$

同理可证 (5.13), (5.14) 式成立。

**引理 5.3** 如果  $H$  非奇异, 则 (5.11) 式的解为  $t = (I - YH^{-1}U^T)K^{-1}g$ 。

**证明:** 对 (5.11) 式中  $(I - UU^T)K(I - UU^T)t = g$  两边同时乘以  $(I - YH^{-1}U^T)K^{-1}$  得:

$$\begin{aligned} (I - YH^{-1}U^T)K^{-1}(I - UU^T)K(I - UU^T)t &= (I - YH^{-1}U^T)K^{-1}g \\ (I - YH^{-1}U^T)K^{-1}K(I - UU^T)t &= (I - YH^{-1}U^T)K^{-1}g \\ (I - YH^{-1}U^T)(I - UU^T)t &= (I - YH^{-1}U^T)K^{-1}g \\ (I - UU^T)t &= (I - YH^{-1}U^T)K^{-1}g \end{aligned}$$

又因为  $U^T t = 0$ , 于是有

$$t = (I - YH^{-1}U^T)K^{-1}g。$$

## 算法 5.2 预优共轭梯度法

·初始化 ( $i = 1, 2, \dots, l$ )

$$t_{i,0}^{(m)} = d_{-1} = 0,$$

$$g_{i,0}^{(m)} = -r_i^{(m)} \quad (i = 1, 2, \dots, l);$$

·迭代 ( $k = 0, 1, \dots$ ) 对一固定的  $i$ , ( $i = 1, 2, \dots, l$ ):

1. 求解  $M_{U^{(m)}} w_{i,k} = g_{i,k}^{(m)}$ ;  $w_{i,k} \perp U^{(m)}$  ( $w_{i,k} = (I - Y_m H_m^{-1} U^{(m)T}) K^{-1} g_{i,k}^{(m)}$ );
2.  $\rho_{i,k} = (g_{i,k}^{(m)}, w_{i,k})$ ;
3.  $v_{i,k} = A_{\eta, U^{(m)}} d_{i,k}$ ;
4.  $\alpha_{i,k} = (d_{i,k}, v_{i,k})$ ;

$$5. \quad t_{i,k}^{(m+1)} = t_{i,k}^{(m)} + \frac{\rho_{i,k}}{\alpha_{i,k}} d_{i,k} \quad ;$$

$$6. \quad g_{i,k}^{(m+1)} = g_{i,k}^{(m)} - \frac{\rho_{i,k}}{\alpha_{i,k}} v_{i,k} \quad ;$$

$A$ :  $n \times n$  对称矩阵;

$$U^{(m)} = [u_1^{(m)}, \dots, u_l^{(m)}]$$

$$\theta_i^{(m)} = (u_i^{(m)}, Au_i^{(m)}) \quad (i = 1, 2, \dots, l)$$

$$r_i^{(m)} = Au_i^{(m)} - \theta_i^{(m)} u_i^{(m)} \quad (i = 1, 2, \dots, l)$$

$$A_{\eta_i, U^{(m)}} = (I - U^{(m)} U^{(m)T})(A - \eta_i I)(I - U^{(m)} U^{(m)T}) \quad (i = 1, 2, \dots, l)$$

$$M_{U^{(m)}} = (I - U^{(m)} U^{(m)T}) K_i (I - U^{(m)} U^{(m)T}) \quad (i = 1, 2, \dots, l)$$

$$K_i \text{ 为对称正定矩阵} \quad (i = 1, 2, \dots, l)$$

注: 对于步骤 1, 求解  $M_{U^{(m)}} w_{i,k} = g_{i,k}^{(m)}$ ;  $w_{i,k} \perp U^{(m)}$ , 事实上由引理 5.2 和引理 5.3 可以得到  $w_{i,k} = (I - Y_m H_m^{-1} U^{(m)T}) K^{-1} g_{i,k}^{(m)}$ 。

下面我们来推导本节主要结论。

对于某个固定的  $i$ , ( $i = 1, 2, \dots, l$ ), 我们首先定义

$$u_{i,k}^{(m+1)} = \frac{u_i^{(m)} + t_{i,k}^{(m)}}{\|u_i^{(m)} + t_{i,k}^{(m)}\|}, \quad (5.15)$$

$$r_{i,k}^{(m+1)} = Au_i^{(m+1)} - (u_i^{(m+1)}, Au_i^{(m+1)}) u_i^{(m+1)}. \quad (5.16)$$

因此,  $r_{i,k}^{(m+1)}$  为当内迭代停止于第  $k$  步时, 算法 5.1 中下一步的残量表示。我们也定义精确求解校正方程时  $\hat{t}_i^{(m)}$  对应的值:

$$\hat{u}_i^{(m+1)} = \frac{u_i^{(m)} + \hat{t}_i^{(m)}}{\|u_i^{(m)} + \hat{t}_i^{(m)}\|}, \quad (5.17)$$

$$\hat{r}_i^{(m+1)} = A\hat{u}_i^{(m+1)} - (\hat{u}_i^{(m+1)}, A\hat{u}_i^{(m+1)}) \hat{u}_i^{(m+1)}. \quad (5.18)$$

下面的定理包括这一部分的主要结果, 注意到我们在算法 5.2 中假设初始近似等于 0, 实际上, 非 0 情形没有意义, 因为任何含丰富信息的非 0 初始估计都可以被立即加到  $u_i^{(m)}$  上。

**定理 5.3** 考虑应用算法 5.2, 对于某个特定的  $i$ , ( $i = 1, 2, \dots, l$ ), 并假设  $\alpha_{i,j} \neq 0$ ,  $j = 0, \dots, k-1$ , 其中  $k$  为某个正整数, 令  $r_{i,k}^{(m+1)}$  如 (5.14), (5.15) 所定义。于是有

$$\|r_{i,k}^{(m+1)}\|^2 = \frac{\|g_{i,k}^{(m)}\|^2}{1 + \|t_{i,k}^{(m)}\|^2} + \left( \frac{\|t_{i,k}^{(m)}\|}{1 + \|t_{i,k}^{(m)}\|^2} (\theta_i^{(m)} - \eta_i + \beta_{i,k}) \right)^2, \quad (5.19)$$

其中

$$\beta_{i,k} = (u_i^{(m)}, (A - \eta_i I)t_{i,k}^{(m)}) = -\sum_{j=0}^{k-1} \frac{\rho_{i,j}^2}{\alpha_{i,j}}. \quad (5.20)$$

还假设  $A_{\eta_i, U^{(m)}}$  在子空间  $U^{(m)\perp}$  上正定, 令  $\hat{t}_i^{(m)}$  为  $A_{\eta_i, U^{(m)}} t_i^{(m)} = -r_i^{(m)}$  的精确解, 且  $t_i^{(m)} \perp U^{(m)}$ ,

并令  $\hat{r}_i^{(m+1)}$  如 (5.17), (5.18) 所定义, 则

$$\left| \frac{\|t_{i,k}^{(m)}\|}{1 + \|t_{i,k}^{(m)}\|^2} (\theta_i^{(m)} - \eta_i + \beta_{i,k}) \right| \leq \|\hat{r}_i^{(m+1)}\| \cdot \max \left( \frac{\|t_{i,k}^{(m)}\|}{\|\hat{t}_i^{(m)}\|}, \frac{\|\hat{t}_i^{(m)}\|}{\|t_{i,k}^{(m)}\|} \right) (1 + f_{i,k}), \quad (5.21)$$

且

$$f_{i,k} \leq \begin{cases} 0, & \text{if } \eta_i = 0 \\ \frac{\|\hat{t}_i^{(m)} - t_{i,k}^{(m)}\|_{A_{\eta_i, U^{(m)}}}^2}{\|\hat{t}_i^{(m)}\|_{A_{\eta_i, U^{(m)}}}^2} \cdot \frac{\theta_i^{(m)} - \lambda_1}{\lambda_1 - \eta_i}, & \text{if } \eta_i < \lambda_1. \end{cases} \quad (5.22)$$

证明: 我们首先证明

$$(d_{i,k}, A_{\eta_i, U^{(m)}} d_{i,j}) = 0 \quad \text{for } (j = 1, \dots, k-1), \quad (5.24)$$

$$(g_{i,k}^{(m)}, d_{i,j}) = 0 \quad \text{for } (j = 1, \dots, k-1), \quad (5.25)$$

$$(g_{i,k}^{(m)}, t_{i,k}^{(m)}) = 0, \quad (5.26)$$

$$(g_{i,0}^{(m)}, t_{i,k}^{(m)}) = \sum_{j=0}^{k-1} \frac{\rho_{i,j}^2}{\alpha_{i,j}}, \quad (5.27)$$

$$(g_{i,k}^{(m)}, \hat{t}_i^{(m)} - t_{i,k}^{(m)}) = (g_{i,0}^{(m)}, \hat{t}_i^{(m)} - t_{i,k}^{(m)}). \quad (5.28)$$

事实上, (5.24), (5.25) 为共轭梯度法的性质. 等式 (5.26) 可由  $t_{i,k}^{(m)} = \sum_{j=0}^{k-1} \frac{\rho_{i,j}^2}{\alpha_{i,j}} d_{i,j}$  直接得

到, (5.27) 式可以随之得到, 因为对于所有的  $j \leq k$ , 根据 (5.24) 得:

$$\rho_{i,j} = (g_{i,j}^{(m)}, w_{i,j}) = (g_{i,j}^{(m)}, d_{i,j}) = (g_{i,0}^{(m)}, d_{i,j}) - \sum_{t=0}^{j-1} \frac{\rho_{i,t}^2}{\alpha_{i,t}} (d_{i,t}, A_{\eta_i, U^{(m)}} d_{i,j}) = (g_{i,0}^{(m)}, d_{i,j}).$$

最后由

$$g_{i,0}^{(m)} - g_{i,k}^{(m)} = \sum_{j=0}^{k-1} \frac{\rho_{i,j}}{\alpha_{i,j}} A_{\eta_i, U^{(m)}} d_{i,j},$$

以及

$$\left( A_{\eta_i, U^{(m)}} d_{i,j}, \hat{t}_i^{(m)} - t_{i,k}^{(m)} \right) = \left( d_{i,j}, A_{\eta_i, U^{(m)}} (\hat{t}_i^{(m)} - t_{i,k}^{(m)}) \right) = (d_{i,j}, g_{i,k}^{(m)}) = 0$$

便可得到 (5.28)。

接下来我们证明 (5.20)，因为  $r_i^{(m)} \perp u_i^{(m)}$ ，所以有

$$-g_{i,0}^{(m)} = r_i^{(m)} = (I - u_i^{(m)} u_i^{(m)T}) r_i^{(m)} = (I - u_i^{(m)} u_i^{(m)T}) (A - \eta_i I) u_i^{(m)}, \quad (5.29)$$

以及  $t_{i,k}^{(m)} \perp U^{(m)}$  进而  $t_{i,k}^{(m)} \perp u_i^{(m)}$ ，我们可以由 (5.27) 推出

$$\begin{aligned} (u_i^{(m)}, (A - \eta_i I) t_{i,k}^{(m)}) &= ((A - \eta_i I) u_i^{(m)}, t_{i,k}^{(m)}) \\ &= ((I - u_i^{(m)} u_i^{(m)T}) (A - \eta_i I) u_i^{(m)}, t_{i,k}^{(m)}) \\ &= (-g_{i,0}^{(m)}, t_{i,k}^{(m)}) \\ &= -\sum_{j=0}^{k-1} \frac{\rho_{i,j}^2}{\alpha_{i,j}} \end{aligned}$$

令  $\theta_{i,k}^{(m+1)} = (u_{i,k}^{(m+1)}, A u_{i,k}^{(m+1)})$ ，因为  $\|u_i^{(m)}\| = 1$  且  $u_i^{(m)}$  垂直于  $t_{i,k}^{(m)}$ ，所以有

$$\|u_i^{(m)} + t_{i,k}^{(m)}\|^2 = 1 + \|t_{i,k}^{(m)}\|^2,$$

进而

$$\begin{aligned} \|r_{i,k}^{(m+1)}\|^2 &= \|(A - \theta_{i,k}^{(m+1)} I) u_{i,k}^{(m+1)}\|^2 \\ &= \|(A - \eta_i I) u_{i,k}^{(m+1)}\|^2 - (\theta_{i,k}^{(m+1)} - \eta_i)^2 \\ &= \left(1 + \|t_{i,k}^{(m)}\|^2\right)^{-1} \|(A - \eta_i I)(u_i^{(m)} + t_{i,k}^{(m)})\|^2 - (\theta_{i,k}^{(m+1)} - \eta_i)^2 \end{aligned} \quad (5.30)$$

另一方面，注意到 (5.29) 可得

$$(I - u_i^{(m)} u_i^{(m)T}) (A - \eta_i I) (u_i^{(m)} + t_{i,k}^{(m)}) = r_i^{(m)} + A_{\eta_i, U^{(m)}} t_{i,k}^{(m)} = -g_{i,k}^{(m)}. \quad (5.31)$$

因此有，

$$\begin{aligned} \|(A - \eta_i I) u_i^{(m)} + t_{i,k}^{(m)}\|^2 &= \|(I - u_i^{(m)} u_i^{(m)T}) (A - \eta_i I) (u_i^{(m)} + t_{i,k}^{(m)})\|^2 \\ &\quad + \|u_i^{(m)} u_i^{(m)T} (A - \eta_i I) (u_i^{(m)} + t_{i,k}^{(m)})\|^2 \\ &= \|g_{i,k}^{(m)}\|^2 + (\theta_i^{(m)} - \eta_i + \beta_{i,k})^2 \end{aligned} \quad (5.32)$$

而且，利用 (5.31) 和 (5.26)，

$$\begin{aligned}
\left(1 + \|t_{i,k}^{(m)}\|^2\right) \left(\theta_{i,k}^{(m+1)} - \eta_i\right) &= \left(u_i^{(m)} + t_{i,k}^{(m)}, (A - \eta_i I)(u_i^{(m)} + t_{i,k}^{(m)})\right) \\
&= \left(\theta_i^{(m)} - \eta_i + \beta_{i,k}\right) + \left(t_{i,k}^{(m)}, (A - \eta_i I)(u_i^{(m)} + t_{i,k}^{(m)})\right) \\
&= \left(\theta_i^{(m)} - \eta_i + \beta_{i,k}\right) - \left(t_{i,k}^{(m)}, g_{i,k}^{(m)}\right) \\
&= \left(\theta_i^{(m)} - \eta_i + \beta_{i,k}\right)
\end{aligned}$$

再根据 (5.30) (5.32) 便可得出等式 (5.19)。

为了证明 (5.21)，首先在 (5.19) 中令  $g_{i,k}^{(m)} \rightarrow 0$ ,  $t_{i,k}^{(m)} \rightarrow \hat{t}_i^{(m)}$ ，我们可以得到

$$\left\| \hat{r}_i^{(m+1)} \right\| = \left| \frac{\left\| \hat{t}_i^{(m)} \right\|^2}{1 + \left\| \hat{t}_i^{(m)} \right\|^2} \left( u_i^{(m)}, (A - \eta_i I)(u_i^{(m)} + \hat{t}_i^{(m)}) \right) \right|.$$

因为对于任意正数  $x$  和  $y$ ，总有  $\frac{x(1+y^2)}{y(1+x^2)} \leq \max(\frac{x}{y}, \frac{y}{x})$ ，因此 (5.21) 在

$$f_{i,k} = - \frac{\left( u_i^{(m)}, (A - \eta_i I)(\hat{t}_i^{(m)} - t_{i,k}^{(m)}) \right)}{\left( u_i^{(m)}, (A - \eta_i I)(u_i^{(m)} + \hat{t}_i^{(m)}) \right)} \text{ 的情况下成立。}$$

进一步，根据 (5.28) 和 (5.29)，

$$\begin{aligned}
-\left( u_i^{(m)}, (A - \eta_i I)(\hat{t}_i^{(m)} - t_{i,k}^{(m)}) \right) &= -\left( (I - u_i^{(m)} u_i^{(m)T})(A - \eta_i I)u_i^{(m)}, \hat{t}_i^{(m)} - t_{i,k}^{(m)} \right) \\
&= \left( g_{0,i}^{(m)}, \hat{t}_i^{(m)} - t_{i,k}^{(m)} \right) \\
&= \left( g_{i,k}^{(m)}, \hat{t}_i^{(m)} - t_{i,k}^{(m)} \right), \\
&= \left\| \hat{t}_i^{(m)} - t_{i,k}^{(m)} \right\|_{A_{\eta_i, U^{(m)}}}^2
\end{aligned}$$

又

$$\begin{aligned}
\left( u_i^{(m)}, (A - \eta_i I)(u_i^{(m)} + \hat{t}_i^{(m)}) \right) &= \theta_i^{(m)} - \eta_i + \left( (I - u_i^{(m)} u_i^{(m)T})(A - \eta_i I)u_i^{(m)}, \hat{t}_i^{(m)} \right) \\
&= \theta_i^{(m)} - \eta_i + \left( r_i^{(m)}, \hat{t}_i^{(m)} \right), \\
&= \theta_i^{(m)} - \eta_i + \left\| \hat{t}_i^{(m)} \right\|_{A_{\eta_i, U^{(m)}}}^2
\end{aligned}$$

$$\text{因此, } f_{i,k} = \frac{\left\| \hat{t}_i^{(m)} - t_{i,k}^{(m)} \right\|_{A_{\eta_i, U^{(m)}}}^2}{\left\| \hat{t}_i^{(m)} \right\|_{A_{\eta_i, U^{(m)}}}^2} \cdot \left( \frac{\theta_i^{(m)} - \eta_i}{\left( u_i^{(m)}, (A - \eta_i I)(u_i^{(m)} + \hat{t}_i^{(m)}) \right)} - 1 \right).$$

这便证明了当  $\eta_i = \theta_i^{(m)}$  时 (5.22) 成立。

对于算法 5.2 中一些中间变量的实际计算可以由下面分析简化。

首先因为

$$\begin{aligned} d_{i,k}^{(m)} &\perp U^{(m)} \quad (i=1,2,\dots,l) \text{ (因为 } d_{i,k}^{(m)} \text{ 是在 } (I-U^{(m)}U^{(m)T}) \text{ 空间上求得的),} \\ v_{i,k} &= (I-U^{(m)}U^{(m)T})(A-\eta_i I)(I-U^{(m)}U^{(m)T})d_{i,k}^{(m)}, \end{aligned}$$

若记  $\tilde{v}_{i,k} = (A-\eta_i I)d_{i,k}^{(m)}$ ,

则

$$v_{i,k} = (I-U^{(m)}U^{(m)T})\tilde{v}_{i,k} = \tilde{v}_{i,k} - U^{(m)}U^{(m)T}\tilde{v}_{i,k}.$$

如果我们跳过计算  $v_{i,k}$  而直接计算  $\tilde{v}_{i,k}$ , 则在计算  $\alpha_{i,k}$  时不产生任何问题, 因为  $d_{i,k}^{(m)} \perp U^{(m)}$ ,

$$\alpha_{i,k} = (d_{i,k}^{(m)}, v_{i,k}) = (d_{i,k}^{(m)}, \tilde{v}_{i,k}).$$

接着我们考虑不计算  $g_{i,k}^{(m)}$  而根据下式直接计算  $\tilde{g}_{i,k}^{(m)}$ :

$$\tilde{g}_{i,k+1}^{(m)} = \tilde{g}_{i,k}^{(m)} - \frac{\rho_{i,k}^{(m)}}{\alpha_{i,k}} \tilde{v}_{i,k},$$

注意到  $\tilde{g}_{i,k}^{(m)} = g_{i,0}^{(m)} - (A-\eta_i I)t_{i,k}^{(m)}$ , 从而  $(I-U^{(m)}U^{(m)T})\tilde{g}_{i,k}^{(m)} = g_{i,k}^{(m)}$ , 即

$$\begin{aligned} w_{i,k} &= \left( I - Y_i^{(m)} \left( H_i^{(m)} \right)^{-1} U^{(m)T} \right) K_i^{-1} g_{i,k}^{(m)} \\ &= \left( I - Y_i^{(m)} \left( H_i^{(m)} \right)^{-1} U^{(m)T} \right) K_i^{-1} (I - U^{(m)}U^{(m)T}) \tilde{g}_{i,k}^{(m)}, \end{aligned}$$

由引理 5.3 得

$$w_{i,k} = \left( I - Y_i^{(m)} \left( H_i^{(m)} \right)^{-1} U^{(m)T} \right) K_i^{-1} \tilde{g}_{i,k}^{(m)}.$$

因为  $w_{i,k} \perp U^{(m)}$

$$\rho_{i,k} = (g_{i,k}^{(m)}, w_{i,k}) = ((I - U^{(m)}U^{(m)T})\tilde{g}_{i,k}^{(m)}, w_{i,k}) = (\tilde{g}_{i,k}^{(m)}, w_{i,k}).$$

可见直接计算  $\tilde{v}_{i,k}$ ,  $\tilde{g}_{i,k}^{(m)}$  并不影响  $\alpha_{i,k}$ ,  $w_{i,k}$  以及  $\rho_{i,k}$ 。

而  $g_{i,k}^{(m)}$  的模可以通过下式来计算:

$$\|g_{i,k}^{(m)}\|^2 = \|(I - U^{(m)}U^{(m)T})\tilde{g}_{i,k}^{(m)}\|^2.$$

上述考虑结果更利于求解校正方程的共轭梯度法的实现。因此可给出下述算法 5.3, 其中应用了上述结果, 并采用了内迭代的停止策略 (详见[25])。

#### 算法 5.3 (关于校正方程的共轭梯度法: 详细算法)

对于固定的  $i$  ( $i=1,2,\dots,l$ ):

·初始化

$$t_{i,0}^{(m)} = d_{i,-1} = 0$$

$$\tilde{g}_{i,0}^{(m)} = -r_i^{(m)}$$

$$Y_i^{(m)} = K_i^{-1}U^{(m)}$$

$$H_i^{(m)} = U^{(m)T} Y_i^{(m)}$$

$$\rho_{i,-1} = 1; \quad \beta_{i,0} = 0$$

·迭代 ( $k = 0, 1, \dots$ )

$$g_{i,k} = \left\| \left( I - U^{(m)} U^{(m)T} \right) \tilde{g}_{i,k}^{(m)} \right\|$$

$$r_{i,k} = \sqrt{\frac{g_{i,k}^2}{1 + \|t_{i,k}^{(m)}\|^2} + \left( \frac{\|t_{i,k}^{(m)}\|^2}{1 + \|t_{i,k}^{(m)}\|^2} (\theta_i^{(m)} - \eta_i + \beta_{i,k}) \right)^2}$$

如果  $r_{i,k} \leq \varepsilon$  , 以  $t_{i,k}^{(m)}$  退出;

如果  $g_{i,k}/g_{i,0} \leq 0.5$  , 则

如果  $r_{i,k} \geq r_{i,k-1}$  , 则以  $t_{i,k-1}^{(m)}$  退出;

如果  $(r_{i,k}/r_{i,k-1}) > (g_{i,k}/g_{i,k-1})^a$  , 则以  $t_{i,k}^{(m)}$  退出。

$$w_{i,k} = \left( I - Y_i^{(m)} \left( H_i^{(m)} \right)^{-1} U^{(m)T} \right) K_i^{-1} \tilde{g}_{i,k}^{(m)}$$

$$\rho_{i,k} = (\tilde{g}_{i,k}^{(m)}, w_{i,k})$$

$$d_{i,k} = w_{i,k} + \frac{\rho_{i,k}}{\rho_{i,k-1}} d_{i,k-1}$$

$$\tilde{v}_{i,k} = (A - \eta I) d_{i,k}$$

$$\alpha_{i,k} = (d_{i,k}^{(m)}, \tilde{v}_{i,k})$$

如果  $\alpha_{i,k} \leq 0$  : 则以  $t_{i,k}^{(m)}$  退出;

$$t_{i,k+1}^{(m)} = t_{i,k}^{(m)} + \frac{\rho_{i,k}}{\alpha_{i,k}} d_{i,k}$$

$$\tilde{g}_{i,k+1}^{(m)} = \tilde{g}_{i,k}^{(m)} - \frac{\rho_{i,k}}{\alpha_{i,k}} \tilde{v}_{i,k}$$

$$\beta_{i,k+1} = \beta_{i,k} - \frac{\rho_{i,k}^2}{\alpha_{i,k}}$$

$K_i^{-1}$  为对称正定矩阵

$\varepsilon$  为期望精度,  $a$  为小于 1 的参数 (经常取  $a = 0.9$ )。



## 5.2 块大小的选取

选取块大小的最有用的信息是知道矩阵的谱分布情况，然而这正是我们要确定的。对于一个矩阵，块大小的最佳选取依赖于待求特征值的数目  $r$  和进行块 Jacobi-Davidson 方法的步数  $s$ 。

对于确定的块大小  $l$ ，块 Jacobi-Davidson 过程的步数  $s$  的选取受计算机内存能力的限制。一般式根据计算机内存能力，用最大可能的  $s$  作为块 Jacobi-Davidson 方法的步数。

实际上，块 Jacobi-Davidson 方法的块大小  $l$  和步数  $s$  的乘积受内存能力的限制。如果我们增大  $l$  的值，则  $s$  的值通常必须减小，反之亦然。

关于块大小  $l$  的选取可以从以下几个方面考虑：

1. 如果我们预先知道要求的特征值中有重特征值，则块大小  $l$  至少应取要求特征值中最大重数。
2. 如果有几个极端密集特征值，并且他们和不要求的特征值有较好的分离，则我们应取块大小  $l$  至少等于这簇密集特征值的个数。

计算经验表明，按上述策略选取块大小  $l$  和块 Jacobi-Davidson 方法及其改进方法的步数  $s$  是比较合适的。

## 5.3 重新开始策略

投影方法为了减少基向量的存储量与正交化过程的运算量，在达到收敛之前一般都需要使用重新开始，即在已得到的投影子空间中选取好的特征向量信息构筑重新开始子空间然后重新进行迭代。在重新开始时必然舍弃了一部分有用的特征向量信息，因而会使收敛速度减慢，为了改进收敛速度，针对不同的方法已有多种手段可供使用，比如关于 Arnoldi 方法，有隐式重新启动策略、加权组合策略等。Davidson 类方法由于对投影子空间性质没有额外的限制，重新开始的方式则较为灵活。在文献[13]中 Sleijpen 等人提出稠密重新开始方法，即在重新开始时保留多于需求个数的特征向量以增大需求特征值与其他特征值之间的分离度。在文献[24]中 Stathopoulos 等人提出动态稠密重新开始技术，对于求矩阵端部特征对效果比较好。

如果要计算矩阵  $A$  的端部  $l$  个特征对，稠密重新开始则是在重新开始时使用  $p_0$  (大于  $l$ ) 个端部 Ritz 值对应的 Ritz 向量张成的子空间作为重新开始子空间。

如果要计算矩阵  $A$  的最接近  $\tau$  的  $l$  个特征对，稠密重新开始则是在重新开始时使用  $p_0$  (大于  $l$ ) 个最接近  $\tau$  的 Ritz 值对应的 Ritz 向量张成的子空间作为重新开始子空间。

本文因为考虑到修正向量的重要性，所以在重新开始时使用 Ritz 向量及 Ritz 向量相应的修正向量张成的子空间作为重新开始子空间。即  $p_0 = 2l$ ，重新开始子空间中，Ritz 向量的个数和修正向量的个数分别为  $l$ 。数值结果表明，这种重新开始方式是合理的。

## 第六章 数值实验

本文的各个算法均由 Matlab7.0 实现, 在 Ath.64 TK55 机器上运行, 运算使用双精度。本文所有试验的初始矩阵  $V_1$  均为随机产生, 其列向量规范正交,  $l$  是块大小,  $m$  用于限制迭代子空间的维数,  $A$  为  $n$  阶对称矩阵,  $tol$  表示由 Ritz 对  $(\theta_i, u_i)$  作为  $A$  的特征对时其残量范数误差限。为表述方便, 块 Jacobi-Davidson 方法记为 BJD, 动态收缩的块 Jacobi-Davidson 方法记为 DBJD, 调和块 Jacobi-Davidson 方法记为 HBJD, 动态收缩的调和块 Jacobi-Davidson 方法记为 DHBJD。块 Jacobi-Davidson 方法和所有改进的块 Jacobi-Davidson 方法中校对方程组的求解均使用求解 5.1.1 节 (5.9) 式代替, 具体解法使用预处理的 GMRES 方法。如果校对方程组解的残量范数小于  $10^{-3}$ , 则终止内迭代, 预处理矩阵使用  $A - \theta I$  的不完全 LU (ILU) 分解。

**例 1**  $A$  是 1000 阶对称矩阵

$$A = \begin{bmatrix} 1 & 1.2 & 0.42 & 0.8 & & & \\ 1.2 & 2 & \ddots & . & \ddots & & \\ 0.42 & . & \ddots & \ddots & . & 0.8 & \\ 0.8 & \ddots & . & . & \ddots & 0.42 & \\ & \ddots & \ddots & . & \ddots & 1.2 & \\ & & 0.8 & 0.42 & 1.2 & 1000 & \end{bmatrix}。$$

首先, 利用 Matlab 软件包中自带的程序在双精度条件下计算  $A$  的 3 个最大特征值如下:

$\lambda_1 = 1.001766606194563$ ,  $\lambda_2 = 0.999252233924983$ ,  $\lambda_3 = 0.998098478188296$ 。

令  $l = 3, m = 30, tol = 1e - 6$ , 则分别用 BJD, DBJD 算法计算的结果如下:

程序名	特征值	相应残量范数 ( $1.0e-06 *$ )	相应 Ritz 对收 敛时迭代次数	迭代 总数	CPU 时间 (秒)
BJD	1.001766606187512	0.019387841946675	542	542	293.7462
	0.999252233924932	0.035484943904647			
	0.998098478153587	0.215904645078175			
DBJD	1.001766605259754	0.027643052862445	10	14	32.2743
	0.999252233227136	0.286815246674748	14		
	0.998098472188153	0.093638408655121	14		

下面将用 HBJD 和 DHBJD 计算矩阵  $A$  的 5 个最靠近某个目标数  $\tau$  的特征值,  $\tau$  的值在表格中给出。下表的第一列给出  $A$  的 5 个最靠近  $\tau$  的特征值是通过 Matlab 自带的程序计算得到的, 并以此作为评价 HBJD 和 DHBJD 可靠性的参照。程序中令  $l = 3, m = 30, tol = 1e - 8$ 。

$\tau = 0.2$				
最靠近 $\tau$ 的 5 个特征值	HBJD		DHBJD	
	特征值	迭代数	特征值	迭代数
0.1980000000000000 0.1990000000000000 0.2000000000000000 0.2010000000000000 0.2020000000000000	0.1980000000190700	337	0.1980000000019070	40
	0.19900000000223161		0.19900000001923011	40
	0.20000000000380289		0.20000000002316133	41
	0.20100000000009532		0.20100000000095327	45
	0.20200000000242740		0.20200000000242739	47
$\tau = 0.7$				
最靠近 $\tau$ 的 5 个特征值	HBJD		DHBJD	
	特征值	迭代数	特征值	迭代数
0.6980000000000001 0.6990000000000001 0.7000000000000000 0.7010000000000000 0.7020000000000000	0.6980000000011110	341	0.6980000000000242	38
	0.6990000000000327		0.6990000000000192	39
	0.7000000001540341		0.7000000000011540	39
	0.7010000002218815		0.7010000002095803	43
	0.7020000000013566		0.702000000001013	43

例 2  $A$  是 1600 阶矩阵,  $B$  为 40 阶

$$A = \begin{bmatrix} B & -I & & & \\ -I & B & -I & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & -I \\ & & & & -I & B \end{bmatrix}, \text{ 其中 } B = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & -1 \\ & & & & -1 & 4 \end{bmatrix}$$

此问题来源于二维 Laplace 方程  $-\frac{\partial^2 u(x, y)}{\partial x^2} - \frac{\partial^2 u(x, y)}{\partial y^2} = \lambda u(x, y)$ , 边界条件为:

$u(x, y)|_{\Omega} = 0$ , 其中  $\Omega = \{(x, y) | 0 < x < 1, 0 < y < 1\}$ 。

首先, 利用 Matlab 软件包中自带的程序在双精度条件下计算  $A$  的 5 个最大特征值如下:

$\lambda_1=7.988263204734929$ ,  $\lambda_2=7.970692449928202$ ,  $\lambda_3=7.970692449928170$ ,  
 $\lambda_4=7.953121695121424$ ,  $\lambda_5=7.941522450123081$ 。

令  $l = 3, m = 30, tol = 1e - 8$ ，则分别用 BJD, DBJD 算法计算的结果如下：

程序名	特征值	相应的残量范数 ( $1.0e-08 *$ )	相应 Ritz 对收 敛时迭代次数	迭代 总数	CPU 时间 (秒)
BJD	7.988263204657986	0.048263250962968	943	943	506.63343
	7.970692449991355	0.172542504002686			
	7.970692449891345	0.126937198139814			
	7.953121695124728	0.012968255875065			
	7.941522450119568	0.138790666570529			
DBJD	7.988263204723878	0.031198813388626	17	28	67.52124
	7.970692449937218	0.065717861997048	17		
	7.970692449919204	0.052547342950137	21		
	7.953121695231406	0.066453586582849	28		
	7.941522450312022	0.259906777672120	28		

下面将用 HBJD 和 DHBJD 计算矩阵  $A$  的 3 个最靠近某个目标数  $\tau$  的特征值， $\tau$  的值在下表中给出。下表第一列给出的 3 个最靠近  $\tau$  的  $A$  的特征值是通过 Matlab 自带的程序计算得到的，并以此作为评价 HBJD 和 DHBJD 可靠性的参照。程序中令  $l = 5, m = 30, tol = 1e - 8$ 。

$\tau = 2$				
最靠近 $\tau$ 的 3 个特征值	HBJD		DHBJD	
	特征值	迭代数	特征值	迭代数
1.993314842851863 1.993314842851868 2.016601748510065	1.993314842936258	233	1.993314842819872	31
	1.993314842798513		1.993314842805656	31
	2.016601748105657		2.016601748143033	35
$\tau = 6$				
最靠近 $\tau$ 的 3 个特征值	HBJD		DHBJD	
	特征值	迭代数	特征值	迭代数
5.983398251489929 6.006685157148141 6.006685157148144	5.983398251484843	143	5.983398251451302	36
	6.006685157109907		6.006685157105878	39
	6.006685157123692		6.006685157184279	39

例 1 和例 2 数值结果表明动态收缩块 Jacobi-Davidson 方法的收敛效果远比块 Jacobi-Davidson 方法好。对于内部特征值的求解，数值结果表明调和块 Jacobi-Davidson 方法 (HBJD) 和动态收缩的调和块 Jacobi-Davidson 方法 (DHBJD) 方法的可靠性都比较理想，但动态收缩的调和块 Jacobi-Davidson 方法比调和块 Jacobi-Davidson 方法的收敛速度要快很多，并且

调和块 Jacobi-Davidson 方法和动态收缩的调和块 Jacobi-Davidson 方法对内部重特征值或密集特征值都是有效的。

## 第七章 总结

本文主要研究求解大型对称特征值问题的块 Jacobi-Davidson (BJD) 方法。提出了动态收缩的块 Jacobi-Davidson 方法和计算矩阵内部特征对的调和块 Jacobi-Davidson 方法，并将动态收缩技术应用于调和块 Jacobi-Davidson 方法，提出了动态收缩的调和块 Jacobi-Davidson 方法。数值结果显示了新方法在收敛速度和可靠性两方面的优越性。同时，本文对块 Jacobi-Davidson 方法的具体实现过程也进行了一定的研究，比如校正方程的求解，重新开始策略，块大小的选择等。这些举措对收敛速度的提高、计算量的减少均有一定的帮助。

本文仅对所提的方法的有效性和可靠性在数值结果上进行了验证，理论上有待深入研究。另外，可将本文所提出的方法和思想推广到非对称矩阵特征值问题和广义特征值问题。

## 参考文献

- [1] 曹志浩. 矩阵特征值问题. 上海: 上海科学技术出版社, 1980.
- [2] Wilkinson J H. The Algebraic Eigenvalue Problem. Oxford: Clarendon Press, 1965.
- [3] Saad Y. Numerical Methods for Large Eigenvalue Problems. Manchester: Manchester University Press, 1992.
- [4] 戴华. 求解大规模矩阵问题的 Krylov 子空间方法. 南京航空航天大学学报, 2001, 33(2): 139-145.
- [5] Lanczos C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J Res Nat Bur Stand, 1950, 45: 225-280.
- [6] Paige C C. The computation of eigenvalues and eigenvectors of very large sparse matrices. Ph. D. Thesis, University of London, 1971.
- [7] Jia Z. The convergence of generalized Lanczos methods for large unsymmetric eigenproblems. SIAM J Matrix Anal Appl, 1995, 16: 843-862.
- [8] Arnoldi W E. The principle of minimized iteration in the solution of the matrix eigenvalue problem. Quart Appl Math, 1951, 5: 17-29.
- [9] Saad Y. Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. Linear Algebra Appl, 1980, 34: 269-295.
- [10] 张勇, 廉庆荣. 解大型非对称特征值问题的精化块不完全正交化算法. 高等学校计算数学学报, 1997, 19: 273-279.
- [11] Davidson E R. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices. J Comput Phys, 1975, 17: 87-94.
- [12] Crouzeix M, Philippe B, Sadkane M. The Davidson method. SIAM J Sci Comput, 1994, 15: 62-76.
- [13] Sleijpen G L G, Van der Vorst H A. A Jacobi-Davidson method for linear eigenvalue problems. SIAM J Matrix Anal Appl, 1996, 17: 401-425.
- [14] Van Den Eshof J. The convergence of Jacobi-Davidson iterations for Hermitian eigenproblem. Numer Linear Algebra Appl, 2002, 9: 163-179.
- [15] 戴华. 求解大型非对称特征问题的块 Arnoldi 方法. 南京航空航天大学学报, 1987, 19: 100-108.
- [16] Jia Z. A variation on the block Arnoldi method for large unsymmetric matrix eigenproblems. Acta

Maths Appl Sinica, 1998, 14: 425-432.

- [17] Sadkane M. Block-Arnoldi and Davidson methods for unsymmetric large eigenvalue problems. Numer Math, 1993, 64: 195-212.
- [18] Underwood R. An iterative block Lanczos method for the solution of large sparse symmetric eigenproblems. Ph. D. Thesis, Stanford University, 1975.
- [19] Jia Z. Generalized block Lanczos methods for large unsymmetric eigenproblems. Numer Math, 1998, 80: 239-266.
- [20] 王岩青. 实对称矩阵特征值问题的迭代块 Jacobi-Davidson 方法. 解放军理工大学学报(自然科学版), 2002, 3(1): 93-96.
- [21] Sorensen D C. Implicit application of polynomial filters in a k-step Arnoldi method. SIAM J Matrix Anal Appl, 1992, 13: 357-385.
- [22] Lehoucq R B. Analysis and implementation of an implicitly restarted Arnoldi iteration. Ph. D. Thesis, Rice University, 1995.
- [23] Lehoucq R B, Sorensen D C. Deflation techniques for an implicitly restarted Arnoldi iteration. SIAM J Matrix Anal Appl, 1996, 17: 789-821.
- [24] Stathopoulos A, Saad Y, Wu K. Dynamic thick restarting of the Davidson and implicitly restarted Arnoldi methods. SIAM J Sci Comput, 1998, 19: 227-245.
- [25] Notay Y. Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem. Numer Linear Algebra Appl, 2002, 9: 21-44.
- [26] Stathopoulos A, Saad Y. Restarting techniques for the (Jacobi-)Davidson symmetric eigenvalue methods. ETNA, 1998, 17: 163-181.
- [27] Jia Z. Refined iterative algorithms based on Arnoldi process for large unsymmetric eigenproblems. Linear Algebra Appl, 1997, 259: 1-23.
- [28] Jia Z. Refined iterative algorithms based on the block Arnoldi process for large unsymmetric eigenproblems. Linear Algebra Appl, 1998, 270: 171-189.
- [29] Jia Z. A refined subspace iteration algorithm for large sparse eigenproblems. Appl Numer Math, 2000, 32: 35-52.
- [30] 陈桂芝. 解大规模非对称矩阵问题的一些精化算法. 大连: 大连理工大学博士学位论文, 2000.
- [31] 贾仲孝, 陈桂芝. 解大规模非对称矩阵特征值问题的精化 Arnoldi 方法的一种变形. 数值计算与计算机应用, 2003, 24(2): 101-110.
- [32] Feng S Q, Jia Z X. A refined Jacobi-Davidson method and its correction equation. Computers



and Mathematics with Applications, 2005, 49: 417-427.

- [33] Morgan R B. Computing interior eigenvalues of large matrices. Num. Linear Algebra Appl, 1991, 154-156: 289-309.
- [34] Morgan R B, Zeng M. Harmonic projection methods for large non-symmetric eigenvalue problems. Num Linear Algebra Appl, 1998, 5: 33-55.
- [35] Paige C C, Parlett B N, Van der vorst H A. Approximate solutions and eigenvalue bounds Krylov subspaces. Num Linear Algebra Appl, 1995, 2: 115-133.
- [36] Morgan R B. Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equation. SIAM J Matrix Anal Appl, 2000, 21: 1112-1135.
- [37] Freund R W. Quasi-kernel polynomials and their use in non-Hermitian matrices. J Comp Appl Math, 1992, 43: 135-158.
- [38] Lanczos C. Solution of systems of linear equations by minimized iterations. J Res Nat Bur Stand, 1952, 49: 33-53.
- [39] Taylor D R. Analysis of the look ahead Lanczos algorithm. Ph. D. Thesis, University of California, Berkeley, 1982.
- [40] Parlett B N. A new look at the Lanczos algorithm for solving symmetric systems of linear equations. Linear Algebra Appl, 1980, 29: 323-346.
- [41] Cullum J, Willoughby R A. Lanczos Algorithms for Large Symmetric Eigenvalue Computation. Vol.1 Theory, Birkhauser, Boston, 1985.
- [42] Freund R W, Gutknecht M H, Nachtigal N M. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. SIAM J Sci Comput, 1993, 14: 137-158.
- [43] Parlett B N. Reduction to tridiagonal form and minimal realization. SIAM J Matrix Anal Appl, 1992, 13: 567-593.
- [44] Saad Y. On the rates of convergence of the Lanczos and the block-Lanczos methods. SIAM J Numer Anal, 1980, 17: 687-706.
- [45] Ye Q. A convergence analysis for nonsymmetric Lanczos algorithms. Math Comput, 1991, 56: 677-691.
- [46] Bai Z. Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem. Math Comput, 1994, 62: 209-226.
- [47] Jia Z. Composite orthogonal projection methods for large eigenproblems. Science in China (Series A), 1999, 42: 577-585.
- [48] Brandts J. The Riccati algorithm for eigenvalues and invariant subspaces of matrices with

inexpensive action. *Linear Algebra Appl*, 2003, 358: 335-365.

- [49] Golub G H, Van Loan C F. *Matrix Computations*. 2 nd ed., Baltimore: Johns Hopkins University Press, 1989.
- [50] Jia Z. The Convergence of harmonic Ritz Values, Harmonic Ritz Vectors and Refined harmonic Ritz Vectors. Technical Report, Department of Mathematical Sciences, Tsinghua University, China, 2002.
- [51] Sleijpen G L G, Jasper Van Den Eshof . Accurate approximations to eigenpairs using the Harmonic Rayleigh-Ritz method. Department of Mathematics, Universiteit Utrecht, Preprint nr. 1184, 2001.
- [52] Sleijpen G L G, Van der Vorst H A, Meijerink E. Efficient expansion of subspaces in the Jacobi-Davidson method for standard and generalized eigenproblems. *ETNA*, 1998, 1: 75-89.

## 致 谢

本文是在导师戴华教授的悉心指导和帮助下完成的。在论文的写作过程中，从资料收集、文献阅读和讨论到论文校阅无不倾注了戴老师的辛勤汗水。同时，戴老师高尚的品德、渊博的学识、严谨的治学态度以及为人师表的作风给我以深刻的教育和影响。“对人有仁爱之心、对事有责任心、对学习有恒心、做研究有钻研精神”是戴老师对学生们的基本希望，这一直以来激励我在各方面不断提高，学习中更加奋发进取、严谨踏实。这一切将使我受益终生。在此，我向戴老师表示衷心的感谢和崇高的敬意！

同时，还要感谢理学院高雄、高征同学，以及吕良福、毛晓彬师兄。

此外，也感谢所有帮助过和关心过我的老师和同学们。

## 在学期间的研究成果及发表的学术论文

- [1] 康艳艳. 求解大型对称特征值问题的改进块 Jacobi-Davidson 方法. 西安文理学院学报（自然科学版）. (已录用).