应用数学和力学编委会编
重 庆 出 版 社 出 版

文章编号: 1000_0887(2002) 11_1150_19

计算 Hamilton 矩阵特征值的一个稳定的有效的保结构的算法*

闫庆友[1,2], 熊西文[1]

(1. 大连大学 先进设计技术中心, 大连 116622;
2. 山东财政学院 经济统计系, 济南 250014)

(唐立民推荐)

摘要: 提出了一个稳定的有效的保结构的计算 Hamilton 矩阵特征值和特征不变子空间的算法, 该算法是由 SR 算法改进变形而得到的• 在该算法中, 提出了两个策略, 一个叫做消失稳策略, 另一个称为预处理技术• 在消失稳策略中, 通过求解减比方程和回溯彻底克服了 Bunser Gerstner 和 Mehrmann 提出的 SR 算法的严重失稳和中断现象的发生, 两种策略的实施的代价都非常低• 数值算例展示了该算法比其它求解 Hamilton 矩阵特征问题的算法更有效和可靠•

关 键 词: Hamilton 矩阵; QR 型算法; 特征值; 稳定性; 消失稳措施; 回溯技术; 减比

中图分类号: O241.6 文献标识码: A

引 言

在众多应用领域中, 例如, 在线性二次最优控制和实代数 Riccati 方程的求解中, 有关实 Hamilton 矩阵 $\boldsymbol{M}$ 的稳定特征值和稳定的不变子空间的数值计算都是十分重要的工作[1~10]• 起初, 人们自然的想法是利用 QR 算法对 $\boldsymbol{M}$ 的特征值和不变子空间进行数值计算, 但是这种方法完全没有利用 $\boldsymbol{M}$ 只含有 $2n^2+n$ 个自由参数而不是 $4n^2$ 个自由参数这一事实, 因而必然带来存储量和计算量的额外花费, 而且 QR 算法计算出的特征值可能失去 Hamilton 矩阵特征值正负成对的性质• 因此, 如果矩阵 $\boldsymbol{M}$ 的条件数很坏, 求解 $\boldsymbol{M}$ 的稳定的不变子空间和求解与之相应的代数 Riccati 方程就可能失败• 于是在随后的研究中人们便把精力集中在保 Hamilton 结构的算法的开发上了[3,11~14]•

保结构地计算 Hamilton 矩阵 $\boldsymbol{M}$ 的稳定的特征值和稳定的不变子空间基于以下事实: 辛相似变换是保 Hamilton 结构的• 1984 年 C. Van Loan 在文[14] 中提出了一个快速的能计算实 Hamilton 矩阵的所有特征值的算法• 该算法计算出的特征值是 $\boldsymbol{M}+\boldsymbol{E}$ 的精确特征值, 其中 $\|\boldsymbol{E}\|_2 = O(\varepsilon^{1/2}\|\boldsymbol{M}\|_2)$, ε 为机器精度, $\|\cdot\|_2$ 为矩阵的谱范数• 计算出的特征值的精度

* 收稿日期: 2001_02_27; 修订日期: 2002_06_28
基金项目: 国家重点基础研究项目(G1999032805); 博士点科研基金资助项目; 教育部优秀年轻教师基金资助项目
作者简介: 闫庆友(1963—), 男, 山东茌平人, 副教授, 博士(E_mail: yanqingyou@263.net).

不仅与其条件数有关而且与其范数有关• 当 λ 是依模较大的特征值时, λ 的计算精度是相当高的, 但当 λ 满足 $|\lambda| \leqslant \varepsilon^{1/2}\|M\|_2$ 时, 计算出的特征值将可能丢失一半的精度• 文[3]中给出了另外一种计算实 Hamilton 矩阵的特征值和不变子空间的可以称为是 QR 型的算法(SR 算法)• 该算法使用了三种辛相似变换把原矩阵 M 约化为 J_三对角 Hamilton 矩阵, 由于其中一种辛相似变换矩阵不是正交矩阵, 其相应矩阵的条件数为 $\sqrt{1+v^2}+|v| \approx 2|v|$, 容易指出, 通常在把原矩阵 M 约化为 J_三对角 Hamilton 矩阵开头几步中, 相应的 $|v|$ 往往是相当大的, 此时计算过程将出现严重失稳• 因此, 用此方法计算出的特征值有时仅有 4~5 个有效数字• 另一方面, 其非正交辛相似变换并不总是存在, 当该变换不存在时, 该算法将发生中断• 1989 年 A. Bunse_Gerstner, V. Mehrmann 和 D. Watkins 在文[12]中提出了另一种 SR 算法• 该算法完全基于非正交的辛 Gauss 消元• 由于其使用了类似选主元的策略, 可以使严重失稳现象部分地得到缓解, 然而由于使用的辛相似变换全是非正交的, 还是没有根本解决失稳问题, 而且该算法也存在中断的弊端•

Patel, Lin 和 Mistra 在[10]中提出了两种计算 Hamilton 矩阵特征值的性质借助正交相似变换计算出了 M 的稳定不变子空间• 由于使用的全是正交变换, 这两种方法实数值稳定的• 然而, 这些方法像 QR 算法一样视 M 为一般矩阵, 没有利用 Hamilton 的矩阵结构, 不能称为是有效的• 而且这两种算法是以预先知道特征值为前提的• 因此寻找一个有效而又稳定且又保 Hamilton 结构的算法在很长时间内被认为是一个悬而未决的问题[2]• 直到 1998 年, Benner, Mehrmann 和 Xu 在文[11]提出了一种计算 Hamilton 矩阵束和辛矩阵束的特征值方法, 才被认为解决了上述公开问题• 该方法是数值向后稳定的和保结构的•

本文所提算法乃是文[3]中所提 SR 算法的改进• 当用文[3]中所提算法对 Hamilton 矩阵进行 J_三对角化的过程中, 如果遇到原算法的计算过程中断或严重失稳时(即辛 Gauss 消元矩阵不存在或其条件数过大时), 我们提出了两个策略, 一个叫做消失稳策略, 另一个称为预处理技术• 在消失稳策略中, 通过求解减比方程和回溯彻底克服了 Bunser Gerstner 和 Mehrmann 提出的 SR 算法的严重失稳和中断现象的发生• 由于本文所提的措施将导致原来已 J_三对角化的极个别列需要再次 J_三对角化, 本文称之为回溯过程• 本文所提的回溯过程所需的工作量大约相当于对矩阵的一个列进行 J_三对角化所需的工作量• 因而本文所提的算法是迄今为止求 Hamilton 矩阵特征值的既能避免中断和失稳又保 Hamilton 结构的稳定的有效的算法• 数值算例展示了所提算法的有效性的稳定性•

1　记号及预备结果

用 $\mathbf{R}^{m\times n}$ 和 $\mathbf{C}^{m\times n}$ 分别表示所有实的和复的 $m\times n$ 矩阵所组成的集合($\mathbf{R}^{n\times 1}=\mathbf{R}^n$, $\mathbf{C}^{n\times 1}=\mathbf{C}^n$)• 用 I_n 表示 $n\times n$ 的单位矩阵, 并记 $J\in\mathbf{R}^{2n\times 2n}$ 为

$$J=\begin{bmatrix}\mathbf{0}_n & I_n\\ -I_n & \mathbf{0}_n\end{bmatrix}, \tag{1}$$

形如

$$M=\begin{bmatrix}A & F\\ Z & -A^{\mathrm{T}}\end{bmatrix} \tag{2}$$

的 $2n\times 2n$ 矩阵 M 称为实 Hamilton 矩阵, 其中 $A\in\mathbf{R}^{n\times n}$, $F=F^{\mathrm{T}}\in\mathbf{R}^{n\times n}$, $Z=Z^{\mathrm{T}}\in\mathbf{R}^{n\times n}$• 特别地, 如果 A, Z 为对角矩阵, F 为三对角矩阵, 则称 M 为 J_三对角 Hamilton 矩阵, 简称为 J_

三对角矩阵• 容易验证

$$\mathscr{H}=\left\{\boldsymbol{M}\in\mathbf{R}^{2n\times 2n}\mid \boldsymbol{J}\boldsymbol{M}=(\boldsymbol{J}\boldsymbol{M})^{\mathrm{T}}\right\}• \tag{3}$$

是所有实的$2n\times 2n$的Hamilton矩阵所组成的集合•

令$\lambda(\boldsymbol{T})$为任意方阵$\boldsymbol{T}$的特征谱,则当$\boldsymbol{M}\in\mathscr{H}$,且$\lambda\in\lambda(\boldsymbol{M})$时,有

$$-\lambda,\ \bar{\lambda},\ -\bar{\lambda}\in\lambda(\boldsymbol{M})• \tag{4}$$

设$\boldsymbol{S}\in\mathbf{R}^{2n\times 2n}$,且满足

$$\boldsymbol{J}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{J}=\boldsymbol{S}^{-\mathrm{T}}• \tag{5}$$

则称$\boldsymbol{S}$为辛矩阵,并记

$$\mathscr{S}=\left\{\boldsymbol{S}\in\mathbf{R}^{2n\times 2n}\mid \boldsymbol{J}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{J}=\boldsymbol{S}^{-\mathrm{T}}\right\}• \tag{6}$$

易知,当$\mathscr{S}\in\boldsymbol{S}$,且记$\boldsymbol{S}=\begin{bmatrix}\boldsymbol{S}_{11} & \boldsymbol{S}_{12}\\ \boldsymbol{S}_{21} & \boldsymbol{S}_{22}\end{bmatrix}$,其中$\boldsymbol{S}_{ij}(i,j=1,2)\in\mathbf{R}^{n\times n}$,则

$$\begin{bmatrix}\boldsymbol{S}_{11} & \boldsymbol{S}_{12}\\ \boldsymbol{S}_{21} & \boldsymbol{S}_{22}\end{bmatrix}^{-1}=\begin{bmatrix}\boldsymbol{S}_{22}^{\mathrm{T}} & -\boldsymbol{S}_{12}^{\mathrm{T}}\\ -\boldsymbol{S}_{21}^{\mathrm{T}} & \boldsymbol{S}_{11}^{\mathrm{T}}\end{bmatrix}• \tag{7}$$

$\boldsymbol{S}$对矩阵乘积运算都是封闭的• 而且$\boldsymbol{S}\in\mathscr{S}$,则有$\boldsymbol{S}^{-1},\boldsymbol{S}^{\mathrm{T}}\in\mathscr{S}$•

辛矩阵$\boldsymbol{S}\in\mathbf{R}^{2n\times 2n}$因具有以下性质,而使之成为一类极为重要的矩阵,即

$$\boldsymbol{M}\in\mathscr{H},\ \boldsymbol{S}\in\mathscr{S}\Rightarrow\boldsymbol{S}\boldsymbol{M}\boldsymbol{S}^{-1}\in\mathscr{H}• \tag{8}$$

如果$\boldsymbol{Q}\in\mathbf{R}^{2n\times 2n}$既是辛矩阵又是正交矩阵,即$\boldsymbol{J}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{J}=\boldsymbol{Q}^{-\mathrm{T}}=\boldsymbol{Q}$• 这意味着存在$\boldsymbol{Q}_1,\boldsymbol{Q}_2\in\mathbf{R}^{n\times n}$,满足

$$\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{Q}_2\\ -\boldsymbol{Q}_2 & \boldsymbol{Q}_1\end{bmatrix}• \tag{9}$$

本文经常用$\boldsymbol{M}\in\mathbf{R}^{2n\times 2n}$,$\boldsymbol{S}\in\mathbf{R}^{2n\times 2n}$分别表示Hamilton阵和辛阵• $\boldsymbol{M}$的第k列$(1\leqslant k\leqslant n)$通常称$\boldsymbol{M}$的前第$k$列,$\boldsymbol{M}$的第$n+k$叫做$\boldsymbol{M}$的后第$k$列,同样可以有关于$\boldsymbol{F}$的第$k$列和$-\boldsymbol{A}^{\mathrm{T}}$的第$k$列的类似的说法• 为节省篇幅,本文使用了Matlab语言中的表示方法• 例如(2)将被写为$\boldsymbol{M}=[\boldsymbol{A},\boldsymbol{F};\boldsymbol{Z},-\boldsymbol{A}^{\mathrm{T}}]$• 有关Matlab语言的书可参看[15]•

2 基本算法模块及计算流程

适当地选择(8)中的$\boldsymbol{S}$,可以使Hamilton阵$\boldsymbol{M}$用辛相似变换约化为J_三对角阵• 以下介绍三类Hamilton矩阵的约化中经常使用的基本辛矩阵及其各自对应的算法,参见文[1, 3, 11, 12, 14]•

1. 设$i\in\{1,2,\dots,n\}$,定义

$$\boldsymbol{J}(i,c,s)=\begin{bmatrix}\boldsymbol{C} & \boldsymbol{S}\\ -\boldsymbol{S} & \boldsymbol{C}\end{bmatrix}, \tag{10}$$

其中$\boldsymbol{S},\boldsymbol{C}\in\mathbf{R}^{n\times n}$是用下式描述的对角阵

$$\boldsymbol{C}=\boldsymbol{I}_n+(c-1)\boldsymbol{e}_i\boldsymbol{e}_i^{\mathrm{T}},\quad \boldsymbol{S}=s\boldsymbol{e}_i\boldsymbol{e}_i^{\mathrm{T}},\quad s^2+c^2=1• \tag{11}$$

当用$\boldsymbol{J}(i,c,s)$左乘$\boldsymbol{M}$时,$\boldsymbol{M}$中的第i行(或列)和第$n+i$行(或列)上的元素都将发生变化,特别当希望把$\boldsymbol{M}$的前第k列中的元系$\boldsymbol{Z}(i,k)$,$i>k$化为0时,c和s便可由以下的计算确定:

a) $\sigma=\sqrt{\boldsymbol{A}^2(i,k)+\boldsymbol{Z}^2(i,k)}$

b) 如果 $\sigma = 0$ 则有 $c = 1, s = 0$

否则 $c = \boldsymbol{A}(i, k)/\sigma, s = \boldsymbol{Z}(i, k)/\sigma$•

当需要把 $\boldsymbol{M}$ 的后第 k 列中的元素 $-\boldsymbol{A}^{\mathrm{T}}(i, k)$ 化为 0 时, 可以同样类似地利用 $F(i, k)$ 和 $-\boldsymbol{A}^{\mathrm{T}}(i, k)$ 来确定 c 和 s• 以后我们将称 $\boldsymbol{J}(i, c, s)$ 为辛 Jacobi 矩阵•

2. 设 $k \in \{1, 2, \cdots, n-2\}$, $\boldsymbol{w} \in \mathbf{R}^{n-k+1}$• 定义

$$\boldsymbol{H}(k, \boldsymbol{w}) = \begin{bmatrix} \operatorname{diag}(\boldsymbol{I}_{k-1}, \boldsymbol{P}) & \mathbf{0}_n \\ \mathbf{0}_n & \operatorname{diag}(\boldsymbol{I}_{k+1}, \boldsymbol{P}) \end{bmatrix}, \tag{12}$$

其中 $\boldsymbol{P} = \boldsymbol{I}_{n-k+1} - 2\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}/\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$• 显然, 如果 $\boldsymbol{w} = \mathbf{0}$, 则 $\boldsymbol{H}(k, \boldsymbol{w}) = \boldsymbol{I}_{2n}$• $k > 1$ 时, $\boldsymbol{H}(k, \boldsymbol{w})$ 的第一列为 $\boldsymbol{e}_1$•

当用 $\boldsymbol{H}(k, \boldsymbol{w})$ 左乘 $\boldsymbol{M}$ 时, $\boldsymbol{M}$ 的各列都将发生变化, 特别当希望把 $\boldsymbol{A}$ 的第 k 列中的元素 $\boldsymbol{A}[k+2:n, k]$ 化为 $\mathbf{0}$ 时, 则向量 $\boldsymbol{w}$ 便可如下确定:

a) $\sigma = \sqrt{\sum_{j=k+1}^{n} \boldsymbol{A}^2(j, k)}$,

b) $w_1 = \boldsymbol{A}(k+1, k) + \operatorname{sign}(\boldsymbol{A}(k+1, k)) \cdot \sigma$,

c) $w_{j-k} = \boldsymbol{A}(j, k) \quad (j = k+2, \cdots, n)$•

当希望把 $\boldsymbol{M}$ 的后第 k 列的 $\boldsymbol{F}$ 的元素 $\boldsymbol{F}[k+2:n, k]$ 化为 $\mathbf{0}$ 时, 也可类似地确定向量 $\boldsymbol{w}$• 以后称 $\boldsymbol{H}(k, \boldsymbol{w})$ 为辛 Householder 阵•

3. 设 $k \in \{1, 2, \cdots, n-1\}$, 定义

$$\boldsymbol{G}(k, v) = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{V} \\ \mathbf{0}_n & \boldsymbol{D}^{-1} \end{bmatrix}, \tag{13}$$

其中 $\boldsymbol{V} = \dfrac{v}{\sqrt[4]{1+v^2}}(\boldsymbol{e}_k\boldsymbol{e}_{k+1}^{\mathrm{T}} + \boldsymbol{e}_{k+1}\boldsymbol{e}_k^{\mathrm{T}})$, $\boldsymbol{D} = \boldsymbol{I}_n + \left(\dfrac{1}{\sqrt[4]{1+v^2}} - 1\right)(\boldsymbol{e}_k\boldsymbol{e}_{k+1}^{\mathrm{T}} + \boldsymbol{e}_{k+1}\boldsymbol{e}_k^{\mathrm{T}})$•

当 $Z(k, k) \neq 0$ 时, 可用 $\boldsymbol{G}(k, v)$ 左乘 $\boldsymbol{M}$ 且欲把 $\boldsymbol{A}$ 的第 k 列中的元素 $\boldsymbol{A}(k+1, k)$ 化为 0, 参数 v 仅需取为:

$$v = -\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)• \tag{14}$$

以后称 $\boldsymbol{G}(k, v)$ 为辛 Gauss 阵•

辛 Jacobi 阵和辛 Householder 阵都是辛正交阵, 即既是辛阵又是正交的矩阵, 而辛 Gauss 阵, 当 $v \neq 0$ 时, 却是非正交的矩阵• 显然, 当 $\boldsymbol{Z}(k, k) = 0$ 时, 相应的辛 Gauss 阵将不复存在, 此时 JTRDIAG[3] 将发生中断, 而当 $\boldsymbol{A}(k+1, k) \neq \mathbf{0}$, $|\boldsymbol{Z}(k, k)|$ 又相对 $\boldsymbol{A}(k+1, k)$ 很小时, 由于 $\boldsymbol{G}(k, v)$ 的条件数 $\mathrm{Cond}_2(\boldsymbol{G}(k, v)) = |v| + \sqrt{1+v^2}$ 将因 $|v|$ 很大而近似地变为 $2|v|$, 此时, 计算过程将遇到严重失稳的情况•

当计算过程中不发生中断和严重失稳的现象时, 任意实 Hamilton 阵可经由以下的 J_三对角化算法化为 J_三对角阵•

JTRIDIAG 算法

1. $k = 1$

2. 对于 $i = n: -1: k+1$

$\sigma = \sqrt{\boldsymbol{A}^2(i, k) + \boldsymbol{Z}^2(i, k)}$

$c = \boldsymbol{A}(i, K)/\sigma, s = \boldsymbol{Z}(i, k)/\sigma$

$\boldsymbol{J}(i, c, s)\boldsymbol{M}\boldsymbol{J}^{\mathrm{T}}(i, c, s) \Rightarrow \boldsymbol{M}$

结束•

3. $\sigma = \sqrt{\sum_{j=k+1}^{n} \boldsymbol{A}^2(j,k)}$

$\boldsymbol{w}_1 = A(k+1,k) + \mathrm{sign}(\boldsymbol{A}(k+1,k)) \cdot \sigma$

$\boldsymbol{w}_{j-k} = \boldsymbol{A}(j,k) \qquad (j = k+2, \cdots, n)$

$\boldsymbol{H}(k, \boldsymbol{w}) \boldsymbol{M} \boldsymbol{H}^{\mathrm{T}}(k, w) \Rightarrow \boldsymbol{M}$

4. $v = -\boldsymbol{A}(k+1,k)/\boldsymbol{Z}(k,k)$

$\boldsymbol{G}(k, \boldsymbol{v}) \boldsymbol{M} \boldsymbol{G}^{-1}(k, \boldsymbol{v}) \Rightarrow \boldsymbol{M}$

5. 对于 $i = n: -1: k+1$

$\sigma = \sqrt{\boldsymbol{F}^2(i,k) + \boldsymbol{A}^2(k,i)}$

$c = \boldsymbol{F}(i,k)/\sigma, \quad s = -\boldsymbol{A}(k,i)/\sigma$

$\boldsymbol{J}(i,c,s) \boldsymbol{M} \boldsymbol{J}^{\mathrm{T}}(i,c,s) \Rightarrow \boldsymbol{M}$

结束•

6. 如果 $k < n-1$

$\sigma = \sqrt{\sum_{j=k+1}^{n} \boldsymbol{F}^2(j,k)}$

$\boldsymbol{w}_1 = \boldsymbol{F}(k+1,k) + \mathrm{sign}(\boldsymbol{F}(k+1,k)) \cdot \sigma$

$\boldsymbol{w}_{j-k} = \boldsymbol{F}(j,k) \qquad (j = k+2, \cdots, n)$

$\boldsymbol{H}(k, \boldsymbol{w}) \boldsymbol{M} \boldsymbol{H}^{\mathrm{T}}(k, \boldsymbol{w}) \Rightarrow \boldsymbol{M}$•

结束•

7. $k < n-1$

$k = k+1$ 　转第 2 步

结束

JTRIDIAG 是计算 Hamilton 矩阵全部特征值的 SR 算法[3]的核心模块, 此处与之并列的还有压缩分析与处理模块 Compac 和隐式位移迭代模块 Implishift• 本文中不拟讨论这两个模块, 但在编程时, 将使用有关文献中这两个模块中的理论成果[3,16~18]•

3 失稳和中断问题的处理措施

3.1 失稳和中断问题

正如我们在后面的数值实验中将要看到的, 在 $\boldsymbol{M}$ 的 J_三对角化过程中, 假设不发生中断和严重失稳的情况是不现实的• 因而如何避免严重失稳和中断问题应是所有讨论保Hamilton结构的算法都不能回避的问题• 文[3]在发生中断现象时, 算法失败• 而对严重失稳的情形提出了如下修正措施: 当用隐式位移方式计算 Hamilton 矩阵的特征值时, 如果在 $\boldsymbol{M}$ 的 J_三对角化过程中出现了严重失稳的情况, 此时可随机生成一个辛矩阵 $\boldsymbol{S} = \boldsymbol{I} - \boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{J}$, 其中 $\boldsymbol{w}$ 为一随机产生的单位向量, 并用 $\boldsymbol{S}^{-1}\boldsymbol{M}\boldsymbol{S}$ 代替$\boldsymbol{M}$, 算法重新开始• 其意图在于使下一轮的J_三对角化过程中, 不再发生严重失稳的情况• 虽然, 这已为某些计算实例所证实, 但这里仍存有一定的不确定因素, 即并不能保证在随后的J_三对角化过程中不发生严重失稳和中断现象• 最致命的是: J_三对角化的重新启动, 意味着以前计算的信息被全部废弃• 因而不能不提出在追求避免失稳的同时, 还要追求算法的有效性问题• 文[12]中提出的基于Gauss 消元的算法, 可

以使严重失稳现象部分地得到缓解，但由于整个 J_三对角化过程实质上全部使用非正交辛相似变换，最终还是没有使失稳问题得到根本解决。 特别是，当在 $\boldsymbol{M}$ 的前第 k 列的 J_三对角化过程中，遇到 $g_{kk}=0$ 的极端情况时，此算法也完全失效。 综上所述，在现今的SR算法中，如何有效的克服中断和避免严重失稳现象的发生是两个关系到 SR 型算法生死存亡的问题。

本文采用了不同于上述两种方法的措施。 即在算 JTRIDIAG 中，当前 k－1 列已经做完 J_三对角化，而且在处理 $\boldsymbol{M}$ 的前第 k 列时循环中 2, 3 步也已经做完，此时 $\boldsymbol{M}$ 的前第 k 列的属于 $\boldsymbol{A}$ 块和 $\boldsymbol{Z}$ 块的部分已经化为

$$\begin{cases}\boldsymbol{A}[:, k]=[0, \cdots, 0, a_{kk}, a_{k+1, k}, 0, \cdots, 0]^{\mathrm{T}}, \\ \boldsymbol{Z}[:, k]=[0, \cdots, 0, z_{kk}, 0, 0, \cdots, 0]^{\mathrm{T}}.\end{cases} \tag{15}$$

此时，若 $z_{kk}\neq 0$，则令 $v=-a_{k+1, k}/z_{kk}$。

如果 $|v|>\mathrm{tol}$（这里 tol 是用来判别是否严重失稳的常数）或 $z_{kk}=0$（此时就认为是遇到了严重失稳或中断的情况），本文采取如下克服中断和避免严重失稳的措施（DUS）：

1．$\boldsymbol{P}(k+1)\boldsymbol{M}\boldsymbol{P}^{\mathrm{T}}(k+1)\Rightarrow\boldsymbol{M}$，其中 $\boldsymbol{P}(k+1)=[\boldsymbol{P}_{11}, \boldsymbol{P}_{12}; \boldsymbol{P}_{21}, \boldsymbol{P}_{22}]$。

而 $\boldsymbol{P}_{11}=\boldsymbol{I}_n-\boldsymbol{e}_{k+1}\boldsymbol{e}_{k+1}^{\mathrm{T}}$, $\boldsymbol{P}_{12}=\boldsymbol{e}_{k+1}\boldsymbol{e}_{k+1}^{\mathrm{T}}$, $\boldsymbol{P}_{21}=-\boldsymbol{P}_{12}$, $\boldsymbol{P}_{22}=\boldsymbol{P}_{11}$。

该措施的实质是要把 $\boldsymbol{M}$ 的第 $k+1$ 行与第 $n+k+1$ 行，第 $k+1$ 列和第 $n+k+1$ 列进行交换，使原来依模很大的 $a_{k+1, k}$ 变换到 $z_{k+1, k}$ 的位置（为了保证是辛相似变换 $a_{k+1, k}$ 变换到 $z_{k+1, k}$ 的位置时，其符号已发生变化）。

2．如果 $z_{k+1, k}<0$，则 $\boldsymbol{S}(k+1)\boldsymbol{M}\boldsymbol{S}^{\mathrm{T}}(k+1)\Rightarrow\boldsymbol{M}$，

其中 $\boldsymbol{S}(k+1)=[\boldsymbol{S}_{11}, \boldsymbol{0}; \boldsymbol{0}, \boldsymbol{S}_{11}]$，而 $\boldsymbol{S}_{11}=\boldsymbol{I}_n-2\boldsymbol{e}_{k+1}\boldsymbol{e}_{k+1}^{\mathrm{T}}$。

该措施的实质是保证新的 $z_{k+1, k}$ 是绝对值很大的正数。

3．$\boldsymbol{HMH}\Rightarrow\boldsymbol{M}$，其中 $\boldsymbol{H}=[\boldsymbol{H}, \boldsymbol{0}; \boldsymbol{0}, \boldsymbol{H}]$，

而 $\boldsymbol{H}=[\boldsymbol{I}_{k-1}, \boldsymbol{0}, \boldsymbol{0}; \boldsymbol{0}, \boldsymbol{H}_0, \boldsymbol{0}; \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{I}_{n-k-1}]$，$\boldsymbol{H}_0=\begin{bmatrix} c & s \\ s & -c\end{bmatrix}$ 满足 $c^2+s^2=1$。

该措施中的 c, s 如何选取问题留待第 3.2 部分去讨论。

注：以上三种措施中使用的矩阵全部为辛正交阵。

3.2 减比与回溯

如果我们把第 3.1 部分的 DUS 措施的第 3 步执行前后的属于 $\boldsymbol{A}, \boldsymbol{F}, \boldsymbol{Z}$ 块的元素分别记为 a_{ij}, f_{ij}, z_{ij} 和 a_{ij}, f_{ij}, z_{ij} $(i, j=1, 2, \cdots, n)$，且令 $c=\cos\theta$, $s=\sin\theta$，则有

$$a_{kk}=a_{kk}\cos^2\theta+a_{k, k+1}\sin\theta\cos\theta+a_{k+1, k+1}\sin^2\theta, \tag{16}$$

$$a_{k, k+1}=(a_{kk}-a_{k+1, k+1})\sin\theta\cos\theta-a_{k, k+1}\cos^2\theta, \tag{17}$$

$$a_{k+1, k}=(a_{kk}-a_{k+1, k+1})\sin\theta\cos\theta+a_{k, k+1}\sin^2\theta, \tag{18}$$

$$a_{i, k}=a_{i, k+1}\sin\theta, \quad a_{i, k+1}=-a_{i, k+1}\cos\theta \qquad (i=k+2, \cdots, n), \tag{19}$$

$$z_{kk}=\frac{1}{2}(z_{kk}+z_{k+1, k+1})+r\cos(2\theta-\phi), \tag{20}$$

$$z_{k, k+1}=z_{k+1, k}=r\sin(2\theta-\phi), \tag{21}$$

$$z_{k+1, k+1}=z_{kk}\sin^2\theta+z_{k+1, k+1}\cos^2\theta, \tag{22}$$

$$z_{ik}=z_{i, k+1}\sin\theta, z_{i, k+1}=-z_{i, k+1}\cos\theta \qquad (i=k+2, \cdots, n), \tag{23}$$

其中

$$r = \sqrt{z_{k+1,k}^2 + \frac{1}{4}(z_{kk} - z_{k+1,k+1})^2}, \tag{24}$$

$$\cos\phi = (z_{kk} - z_{k+1,k+1})/(2r), \tag{25}$$

$$\sin\phi = z_{k+1,k}/r. \tag{26}$$

此外，还特别需要注意的是

$$f_{k,k+1} = f_{k,k-1}\cos\theta,\ f_{k+1,k-1} = f_{k,k-1}\sin\theta. \tag{27}$$

由于 $\boldsymbol{HMH} \Rightarrow \boldsymbol{M}$ 被执行，$\boldsymbol{M}$ 的前第 k 列上比未执行前又增加了一些新的非零元。当 $\boldsymbol{M}$ 的前第 k 列被再一次J_三对角化时，如果把 $\boldsymbol{M}$ 的前第 k 列再一次被辛消元时的元素 $\boldsymbol{A}[k+1, k]$ 记为 $a_{k+1,k}$，则需要被辛Gauss消元的总量是

$$a_{k+1,k} = \sqrt{z_{k+1,k}^2 + \alpha_{k+1}^2\sin^2\theta + a_{k+1,k}^2}, \tag{28}$$

这里 $\alpha_{k+1} = \sqrt{\sum_{i=k+2}^{n}(a_{i,k+1}^2 + z_{i,k+1}^2)}$。

此时比值 $|a_{k+1,k}|/|z_{kk}|$ 是 θ 的函数。我们必须选取 θ，使得 $|a_{k+1,k}|/|z_{kk}|$ 小于tol才是被允许的。因此，当此 $|a_{k+1,k}|/|z_{kk}|$ 被指定为小于tol的 T 时，我们的问题化为求以 θ 为自变量的超越方程

$$a_{k+1,k}^2 - T^2 z_{kk}^2 = 0 \tag{29}$$

的实根。

当 T 被指定后，上面方程是含7个参数 a_{kk}, $a_{k,k+1}$, $a_{k+1,k+1}$, z_{kk}, $z_{k+1,k}$, $z_{k+1,k+1}$, α_{k+1} 关于 θ 的超越方程。明显地，当 $\theta = 0$ 时，$|a_{k+1,k}/z_{kk}| = |z_{k+1,k}/z_{kk}|$。因此研究上述超越方程，其实质是研究应如何选取 θ，方能使比值 $|a_{k+1,k}/z_{kk}|$ 从 $|z_{k+1,k}/z_{kk}|$ 减少到tol以下，并研究 T 应小到何种程度才合适的问题。因此，我们不妨把方程(29)，称为本问题中的减比方程，把减比方程左端的函数称为减比函数，而把 $\boldsymbol{HMH} \Rightarrow \boldsymbol{M}$ 称为减比辛相似变换。

使 $|z_{k+1,k}/z_{kk}|$ 降到tol以下意味着失稳现象的消除，因而我们把减比辛相似变换的实施，称为本算法中的消失稳措施(DUS)。由于执行了减比辛相似变换，由(27)知 $f_{k+1,k-1}$ 将由0变换成 $f_{k,k-1}\sin\theta$，当此值不为0时，这意味着原来已经J_三对角化的 M 的后第 $k-1$ 列失去了已经J_三对角化的形式。而且由于 $\boldsymbol{HMH} \Rightarrow \boldsymbol{M}$ 的被执行，使 M 的前第 k 列中已经化为0的许多元素又重新变成非零元素。当使用JTRIDIAG算法中的2, 3, 4步使这些元素再一次化为0时，$\boldsymbol{M}$ 的后第 $k-1$ 列的上一半和下一半的列向量还会增加更多的非零元素。于是 $\boldsymbol{M}$ 的后第 $k-1$ 列面临着需要再一次J_三对角化的问题。本文中，我们把 $\boldsymbol{M}$ 中某个已经J_三对角化的列，由于减比的原因而又失去了J_三对角化的形式，需要再一次对之进行J_三对角化的计算工作，称为对该列进行的必需的回溯。当前要对 $\boldsymbol{M}$ 的后第 $k-1$ 列进行回溯，只需机械地执行JTRIDIAG算法中的第5, 6两步即可。但对 $\boldsymbol{M}$ 的后第 $k-1$ 列的回溯，又将引起对 $\boldsymbol{M}$ 的前第 k 列进行必需的回溯，如果在新一轮的对 $\boldsymbol{M}$ 的前第 k 列进行回溯中所产生的比值：$\sqrt{\sum_{i=k+1}^{n}(a_{ik}^2 + z_{ik}^2)}/|z_{kk}| \leqslant \mathrm{tol}$，此后便不再有减比问题，因而也不再发生对 $\boldsymbol{M}$ 的后第 $k-1$ 列的回溯问题，此时便可说一次回溯成功(关于一次回溯成功的充要条件将在第3.3部分讨论)。在JTRIDIAG算法的第 k 个循环中，一次回溯成功所必需付出的工作量大约是同一轮循环中需减比与回溯时所付出的工作量的1倍稍多一点。于是下一个值得关注的问题是：如何才能做到一次回溯成功。

3.3 一次回溯成功的充要条件

如果我们用 $\boldsymbol{M}^{(0)}$ 表示减比相似变换刚被执行后的矩阵 $\boldsymbol{M}$; 用 $\boldsymbol{M}^{(1)}$ 表示对 $\boldsymbol{M}^{(0)}$ 的前第 k 列用一系列辛 Jacobi 相似变换使 $z_{ik}^{(0)}(i=+\ 1, \cdots, n)$ 化为0 后的矩阵; 用 $\boldsymbol{M}^{(2)}$ 表示对 $\boldsymbol{M}^{(1)}$ 的前第 k 列用辛Householder变换使 $a_{ik}^{(1)}(i=+\ 2, \cdots, n)$ 化为0后所得矩阵; 用 $\boldsymbol{M}^{(3)}$ 表示对 $\boldsymbol{M}^{(2)}$ 的前第 k 列用辛 Gauss 变换使 $a_{k+1,k}^{(2)}$ 化为 0后所得矩阵; 用 $\boldsymbol{M}^{(4)}$ 表示对 $\boldsymbol{M}^{(3)}$ 的后第 $k-1$ 列用一系列辛Jacobi 变换使 $a_{k-1,i}^{(3)}(i=k, \cdots, n)$ 化为0后所得的矩阵; 用 $\boldsymbol{M}^{(5)}$ 表示对 $\boldsymbol{M}^{(4)}$ 的后第 $k-1$ 列用辛 Householder 变换使 $f_{ik}^{(4)}(i=k+2, \cdots, n)$ 化为 0后所得矩阵• 于是, 利用第 3.2 部分中公式(16) ~ (27), 当把写有"–" 处的元素, 比如 a_{jk}, 改写为 $a_{jk}^{(0)}$, 便得到了减比相似变换执行前后的元素间的关系•

当 $\boldsymbol{M}^{(4)}$ 形成后, $\boldsymbol{M}^{(4)}$ 的后第 $k-1$ 列的上一半的向量的形式为

$$\boldsymbol{M}^{(4)}[1:n, n+k-1] = \boldsymbol{F}^{(4)}[1:n, k-1] = [0, \cdots, 0, f_{k-2,k-1}^{(4)}, f_{k-1,k-1}^{(4)}, f_{k,k-1}^{(4)}, \cdots, f_{n,k-1}^{(4)}]^{\mathrm{T}}, \tag{30}$$

记 $\boldsymbol{x} = [f_{k,k-1}^{(4)}, \cdots, f_{n,k-1}^{(4)}]^{\mathrm{T}} = [x_1, \cdots, x_{n-k+1}]^{\mathrm{T}}$•

下一步, 将用 $\boldsymbol{x}$ 形成的 Householder 向量 $\boldsymbol{w}$

1. $\sigma = \sqrt{\sum_{i=1}^{n-k+1} x_i^2}$
2. $w_1 = x_1 + \operatorname{sign}(x_1) \cdot \sigma$
3. $w_j = x_j (j=2, \cdots, n-k+1)$

和 Householder 矩阵 $\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}} / \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$, 并对 $\boldsymbol{M}^{(4)}$ 做辛相似变换, 以使

$$f_{k,k-1}^{(5)} = \operatorname{sign}(f_{k,k-1}^{(4)}) \|\boldsymbol{x}\|_2,$$

$$f_{i,k-1}^{(5)} = 0 \qquad (i=k+1, \cdots, n)•$$

此时, $\boldsymbol{A}^{(5)}[k:n, k:n] = \boldsymbol{H}\boldsymbol{A}^{(4)}[k:n, k:n]\boldsymbol{H}$,

$$\boldsymbol{Z}^{(5)}[k:n, k:n] = \boldsymbol{H}\boldsymbol{Z}^{(4)}[k:n, k:n]H•$$

于是, 便可把一次回溯成功的充要条件描述为:

$$\mathrm{tol} \cdot | z_{kk}^{(5)} | \geqslant \sqrt{\sum_{i=k+1}^{n} ((a_{ik}^{(5)})^2 + (z_{ik}^{(5)})^2)}• \tag{31}$$

或

$$\mathrm{tol} \cdot | \boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{H}\boldsymbol{Z}^{(4)}[k:n, k:n]\boldsymbol{H}\boldsymbol{e}_1 | \geqslant$$

$$\sqrt{\sum_{i=2}^{n-k+1} ((\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{H}\boldsymbol{A}^{(4)}[k:n, k:n]\boldsymbol{H}\boldsymbol{e}_1)^2 + (\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{H}\boldsymbol{Z}^{(4)}[k:n, k:n]\boldsymbol{H}\boldsymbol{e}_1)^2)}• \tag{32}$$

由于对称阵 $\boldsymbol{Z}^{(4)}[k, n:k, n]$ 是一阶块阵 $z_{kk}^{(4)}$ 与对称阵$\boldsymbol{Z}^{(4)}[k+1:n, k+1:n]$ 的直和, 于是 $\boldsymbol{Z}^{(4)}[k:n, k:n]$ 可以相应的直交分解成

$$\boldsymbol{Z}^{(4)}[k:n, k:n] = \operatorname{diag}(1, \boldsymbol{Q}) \Lambda (\operatorname{diag}(1, Q))^{\mathrm{T}}, \tag{33}$$

这里 $\boldsymbol{Q}$ 是使对称阵$\boldsymbol{Z}^{(4)}[k+1:n, k+1:n]$ 化为对角阵的$(n-k)\times(n-k)$ 阶正交阵, $\Lambda = \operatorname{diag}(z_{kk}^{(4)}, \lambda_1, \cdots, \lambda_{n-k})$, 即满足

$$\boldsymbol{Z}^{(4)}[k+1:n, k+1:n] = \boldsymbol{Q}\operatorname{diag}(z_{kk}^{(4)}, \lambda_1, \cdots, \lambda_{n-k})\boldsymbol{Q}^{\mathrm{T}}, \tag{34}$$

其中 $\lambda_1, \cdots, \lambda_{n-k}$ 是对称阵$\boldsymbol{Z}^{(4)}[k+1:n, k+1:n]$ 的特征值• 于是可导出

$$z_{kk}^{(5)} = \boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{H}\operatorname{diag}(1, \boldsymbol{Q}) \Lambda (\operatorname{diag}(1, \boldsymbol{Q}))^{\mathrm{T}}\boldsymbol{H}\boldsymbol{e}_1 =$$

$$z_{kk}^{(4)}h_{11}^2+\sum_{i=2}^{n-k+1}\lambda_{i-1}h_{i1}^2=z_{kk}^{(4)}h_{11}^2+\lambda\sum_{i=2}^{n-k+1}h_{i1}^2=$$
$$z_{kk}^{(4)}x_1^2/\sigma^2+\lambda(x_2^2+,\ \cdots,x_{n-k+1}^2)/\sigma^2,\tag{35}$$

这里 λ 满足 $\min\left\{\lambda_1,\ \cdots,\ \lambda_{n-k}\right\}\leqslant\lambda\leqslant\max\left\{\lambda_1,\ \cdots,\ \lambda_{n-k}\right\}$,

3.4 减比方程的建立与求解

减比方程的建立与求解是讨论应如何选取减比方程中的 T, 使得既满足 $T<$ tol 又能满足一次回溯成功・ 从一次回溯成功的充要条件(31) 及(35) 里, 我们看到, 非常关键的问题是尽量使 $f_{k,k-1}^{(4)}$ 即 x_1 在向量 $\boldsymbol{F}^{(4)}[k:n,k-1]$ 中占有相当大的份额・ $\boldsymbol{F}[k:n,k-1]$ 中的元素在执行减比相似变换后发生了变化, 由式(27) 知其中的 $f_{k,k-1}$ 将由原来的 $f_{k,k-1}$ 变为 $f_{k,k-1}\cos\theta$, 即

$$f_{k,k-1}^{(0)}=f_{k,k-1}\cos\theta,\tag{36}$$

这里 θ 是由减比方程求出的・ 因为 $f_{k,k-1}^{(4)}=f_{k,k-1}^{(0)}$, 所以, 由上式可以看出, 把 $|\theta|$ 取的很小, 例如 $|\theta|<\pi/180$ 或更小将有利于一次回溯成功・ 另一方面, 由(35) 还可以看出, 一次回溯成功的可能性还有赖于 $z_{kk}^{(4)}$ 不是过分的小・ 由于 $z_{kk}^{(4)}=z_{kk}^{(3)}=z_{kk}^{(2)}=z_{kk}^{(1)}=z_{kk}^{(0)}$ 由(20) 知

$$z_{kk}^{(4)}=\frac{1}{2}(z_{kk}+z_{k+1,k+1})+r\cos(2\theta-\phi)\tag{37}$$

由于当 $\theta=0$ 时, $|z_{kk}^{(4)}(\theta)|=|z_{kk}|$ 是很小的数, 然而由于 r 很大, θ 的很小变化, 将引起 $z_{kk}^{(4)}$ 的显著变化・ 于是, 使 $f_{k,k-1}^{(4)}$ 在向量 $\boldsymbol{F}^{(4)}[k:n,k-1]$ 中占有大的份额和 $z_{kk}^{(4)}$ 不是过分的小, 这两个要求是并行不悖的, 即存在 θ 使得这两个要求可以同时达到・ 由于 θ 的值将随减比方程中的 T 的取值而确定, 我们曾就减比方程的7个参数, 依符号、不同数量级的变化, 作了许多实验, 发现减比方程左端函数当 T 取值偏大时, 相应的 $|\theta|$ 值将减小・ 并且当 $\sqrt{T}$ 的取值远远小于失稳时的 $|v|$ 时, θ 完全可以取到接近于0, 而且此时求得的 θ 与满足一次回溯成功的要求・ 在第6部分数值实验中取 tol(判别是否严重失稳的控制常数) 为 $10^6/\|\boldsymbol{M}\|_\infty$, 即当 JTRIDIAG 算法的第4步中的 $Z(k,k)=0$ 或 $|v|=|\boldsymbol{A}(k+1,k)/\boldsymbol{Z}(k,k)|\geqslant$ tol 时, 就求解减比方程以确定 θ 和执行减比辛相似变换, 进而回溯・ 一般情况下, 减比方程中的 T 取为 $\sqrt[8]{v^3}\sim\sqrt[7]{v^3}$, 当因 $\boldsymbol{Z}(k,k)=0$ 而发生中断时, 取 $T=100$, 数值算例证实了这种选取方案就是合适可行的・

需要说明的是, 当 T 取定以后, 减比方程可能有很多解 θ, 我们仅需选取依模最小的那个, 由于减比函数在 $\theta=0$ 附近, 变化很剧烈, 因此减比方程的数值解必需具有很高的精度・ 为此, 使用Newton切线法求解减比方程, 也需要一个好的开始点・ 在后面的数值实验中, 我们先用减比函数在 $\theta=0$ 处的函数值、导数值、二阶导数值构造一个二次逼近函数, 然后求解该二次函数的根, 并用绝对值较小的根作为初始点・ 值得一提的是, 按这种方案求解减比方程, 从没失败过, 这些工作虽然也耗费一些工作量, 但对阶数稍大一点的 $\boldsymbol{M}$ 矩阵来说, 这和做一次辛 Jacobi 相似变换的工作量相比还是微不足道的・

4 带消失稳与回溯措施的JTRIDIAG算法

下面我们给出带消失稳与回溯措施的 JTRIDIAG 算法(MJTRDIAG)・

MJTRIDIAG 算法

1. $k=1$

2. 对 $i = n: -1: k+1$,

$\sigma = \sqrt{\boldsymbol{A}^2(i, k) + \boldsymbol{Z}^2(i, k)}$,

$c = \boldsymbol{A}(i, k)/\sigma$, $s = \boldsymbol{Z}(i, k)/\sigma$

$\boldsymbol{J}(i, c, s)\boldsymbol{M}\boldsymbol{J}^{\mathrm{T}}(i, c, s) \Rightarrow \boldsymbol{M}$

结束•

3. 如果 $k < n-1$

$$\sigma = \sqrt{\sum_{j=k+1}^{n} \boldsymbol{A}^2(j, k)}$$

$w_1 = A(k+1, k) + \mathrm{sign}(A(k+1, k)) \cdot \sigma$

$w_{j-k} = A(j, k)$, $\quad (j = k+2, \cdots, n)$

$\boldsymbol{H}(k, w)\boldsymbol{M}\boldsymbol{H}^{\mathrm{T}}(k, w) \Rightarrow \boldsymbol{M}$

结束•

4. 如果 $Z(k, k) \neq 0$ 且 $|\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)| \leqslant \mathrm{tol}$,

$v = -\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)$,

$\boldsymbol{G}(k, v)\boldsymbol{M}\boldsymbol{G}^{-1}(k, v) \Rightarrow \boldsymbol{M}$•

否则

a) $\boldsymbol{P}(k+1)\boldsymbol{M}\boldsymbol{P}^{\mathrm{T}}(k+1) \Rightarrow \boldsymbol{M}$

b) 如果 $z_{k+1,k} < 0$, 则 $\boldsymbol{S}(k+1)\boldsymbol{M}\boldsymbol{S}^{\mathrm{T}}(k+1) \Rightarrow \boldsymbol{M}$

c) 解减比方程确定 $\boldsymbol{H}$, $\boldsymbol{HMH} \Rightarrow \boldsymbol{M}$

d) 对于 $i = n: -1: k+1$

$\sigma = \sqrt{\boldsymbol{A}^2(i, k) + \boldsymbol{Z}^2(i, k)}$

$c = \boldsymbol{A}(i, k)/\sigma$, $s = \boldsymbol{Z}(i, k)/\sigma$

$\boldsymbol{J}(i, c, s)\boldsymbol{M}\boldsymbol{J}^{\mathrm{T}}(i, c, s) \Rightarrow \boldsymbol{M}$

结束•

e) $$\sigma = \sqrt{\sum_{j=k+1}^{n} A^2(j, k)}$$

$w_1 = A(k+1, k) + \mathrm{sign}(A(k+1, k)) \cdot \sigma$

$w_{j-k} = A(j, k) \quad (j = k+2, \cdots, n)$

$\boldsymbol{H}(k, w)\boldsymbol{M}\boldsymbol{H}^{\mathrm{T}}(k, w) \Rightarrow \boldsymbol{M}$

f) $v = -\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)$

$\boldsymbol{G}(k, v)\boldsymbol{M}\boldsymbol{G}^{-1}(k, v) \Rightarrow \boldsymbol{M}$•

g) 对于 $i = n: -1: k+1$,

$\sigma = \sqrt{\boldsymbol{F}^2(i, k-1) + \boldsymbol{A}^2(k-1, i)}$,

$c = \boldsymbol{F}(i, k-1)/\sigma$, $s = -\boldsymbol{A}(k-1, i)/\sigma$,

$\boldsymbol{J}(i, c, s)\boldsymbol{M}\boldsymbol{J}^{\mathrm{T}}(i, c, s) \Rightarrow \boldsymbol{M}$•

结束•

h) $$\sigma = \sqrt{\sum_{j=k}^{n} F^2(j, k-1)}$$

$w_1 = F(k, k-1) + \mathrm{sign}(F(k, k-1)) \cdot \sigma$

$w_{j-k+1} = F(j, k) \qquad (j = k+1, \cdots, n)$

$\boldsymbol{H}(k-1, \boldsymbol{w})\boldsymbol{M}\boldsymbol{H}^{\mathrm{T}}(k-1, \boldsymbol{w}) \Rightarrow \boldsymbol{M}$ 转第2步

结束•

5. 对于 $i = n: -1: k+1$

$\sigma = \sqrt{F^2(i, k) + A^2(k, i)}$

$c = F(i, k)/\sigma, s = -A(k, i)/\sigma$

$\boldsymbol{J}(i, c, s)\boldsymbol{M}\boldsymbol{J}^{\mathrm{T}}(i, c, s) \Rightarrow \boldsymbol{M}$

结束•

6. 如果 $k < n-1$

$\sigma = \sqrt{\sum_{j=k+1}^{n} F^2(j, k)}$

$w_1 = F(k+1, k) + \operatorname{sign}(F(k+1, k)) \cdot \sigma$

$w_{j-k} = F(j, k) \qquad (j = k+2, \cdots, n)$

$\boldsymbol{H}(k, w)\boldsymbol{M}\boldsymbol{H}^{\mathrm{T}}(k, w) \Rightarrow \boldsymbol{M}$

结束

7. 如果 $k < n-1$

$k = k+1$ 转第2步

结束•

5 预处理问题

由于在隐式位移的SR算法[3]中,要求 $\boldsymbol{M}$ 在J_三对角化过程中的所有辛相似变换的变换矩阵的乘积的第一列,始终应保持为 $r\boldsymbol{e}_1$ 的形式,这里 $\boldsymbol{e}_1$ 是 $2n \times 2n$ 阶单位矩阵的第一列, r 是可以变化的非零常数• 这就要求 $\boldsymbol{M}$ 在J_三对角化过程中,当对 $\boldsymbol{M}$ 的前第1列进行J_三对角化时,必须不发生减比的问题,否则在相似变换矩阵的乘积中将有一个乘积因子的第1列具有 $[c, s, 0, \cdots, 0]^{\mathrm{T}}$ 的形式,其中 $c^2 + s^2 = 1$,而 $s \neq 0$•

为使入机矩阵在前第1列不发生减比问题,应使入机矩阵的 z_{11} 一开始就比较大,为此我们提出如下预处理措施• 当对 $\boldsymbol{M}$ 的前第1列已经约化成

$$[a_{11}, a_{21}, 0, \cdots, 0 \vdots z_{11}, 0, \cdots, 0]^{\mathrm{T}}$$

的形式时,可强制执行一次辛行交换的相似变换,使该列化为

$$[a_{11}, 0, 0, \cdots, 0 \vdots z_{11}, z_{21}, 0, \cdots, 0]^{\mathrm{T}}$$

的形式,此时的 $|z_{21}|$ 是相当大的•

$$\boldsymbol{Z}[1: 2; 1: 2] = [z_{11}, z_{21}; z_{21}, z_{22}] \cdot \tag{38}$$

然后对 $\boldsymbol{M}$ 强制执行一次减比相似变换,所不同的是减比相似变换中的 c, s 的选取应使新的 z_{11},即 $\boldsymbol{z}_{11}$ 成为原来的二阶对称矩阵 $[z_{11}, z_{21}; z_{21}, z_{22}]$ 依模最大的特征值,而此时的 z_{21} 将变为0• 因为原来的 z_{21} 绝对值很大,对称矩阵的对角化过程采用正交变换,所以变换后的 z_{11},即 z_{11} 的绝对值会很大• 当 $\boldsymbol{M}$ 的前第1列的非对角元比较大而对角元不大时,预处理可以只进行一次,如果第2列的非对角元也比较大时,预处理不妨进行两次• 预处理工作,利用Matlab的现成语句是容易实现的,且代价很小•

6　数值实验

为了对本文算法的有效性及计算精度进行检测, 我们用Matlab语言编制了测试程序, 在Inter Pentium 100Hmz 微机上(机器精度 $\varepsilon \approx 2.2 \times 10^{-16}$) 进行了大量数值实验• 入机的Hamilton矩阵是随即生成的; 矩阵无纯虚特征值且预先知道所有的特征值; $\boldsymbol{M}$ 的阶数在10~100之间; 矩阵中的元素依模的最大值与最小值之比约在1~ 2 000 之间; tol 的取值在 $10^6/\|\boldsymbol{M}\|_\infty$~ $10^7/\|\boldsymbol{M}\|_\infty$ 之间; 压缩分析与处理中的收敛标准取为 $\varepsilon\|\boldsymbol{M}\|_\infty$; 当发生严重失稳现象时, 减比方程中的 T 取为 $\sqrt[8]{v^3}$~ $\sqrt[7]{v^3}$, 当发生中断时, 取 $T = 100$• 对上百例的随机矩阵计算表明: 由于tol取值不算大而元素间的大小悬殊较大, 因此算例几乎都出现减比和回溯过程• 如所希望的那样, 需要减比与回溯的列, 一般只出现在算法的开头几步, 而每个列都没有发生一次以上的回溯的现象• 预处理可以减少严重失稳发生的次数, 而且使计算过程更加稳定• 每算出一个特征值大约需0.6~ 1.1 次迭代• 每个计算的特征值与精确的特征值相比, 一般都10~ 15 个以上的有效数字•

例1　$\boldsymbol{M} = [\boldsymbol{A}, \boldsymbol{F}; \boldsymbol{Z}, -\boldsymbol{A}']$ 为一18×18阶Hamilton矩阵• 由于篇幅原因具体数据放在了附录•

表1给出了带消失稳和回溯措施的MJTRIDIAG 算法、预处理后的MJTRIDIAG 算法和不带消失稳和回溯措施的JTRIDIAG算法在有中断现象时在消除失稳和避免中断方面的比较• 其中第一列的 k 是J_三对角化过程中的列号, 之所以出现了三次 $k = 2$, 是因为在矩阵的第二列实行了消失稳和回溯措施• 第二列给出的是带消失稳和回溯措施的MJTRIDIAG 算法相应的 $|\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)|$• 第三列给出的是使用第5部分的预处理后再用带消失稳和回溯措施的MJTRIDIAG算法的相应的 $|\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)|$, 出乎意料的是: 预处理后就完全控制了失稳现象• 第四列给出的是不带消失稳和回溯措施的JTRIDIAG 算法相应的 $|\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)|$• 可见不带消失稳和回溯措施的JTRIDIAG 算法在 $k = 4$时, 遇到了 $\boldsymbol{Z}(4, 4) = 0$ 的情况, 算法完全失败• 本文算法不仅不会中断, 还有效地控制了严重失稳现象•

表2给出了用带消失稳和回溯措施的MJTRIDIAG 算法计算出的特征值、预处理后的本算法计算出的特征值的绝对误差• 其中第一列是MJTRIDIAG 算法计算出的结果, 第二列给出的是预处理后的本算法计算出的结果, 第三列是本问题的精确特征值(由于Hamilton 矩阵的特征值具有(4)的性质, 我们只给出了部分特征值)• 可见本文算法计算出的特征值具有10~ 13 有效数字, 而预处理后本文算法计算出的特征值具有12~ 15 有效数字, 几乎和QR 算法相媲美•

例2　$\boldsymbol{M} = [\boldsymbol{A}, \boldsymbol{F}; \boldsymbol{Z}, -\boldsymbol{A}']$ 为一30×30阶Hamilton矩阵, 由于篇幅原因, 矩阵省略, 只给出计算结果•

表3给出了带消失稳和回溯措施的MJTRIDIAG 算法、预处理后的本算法(MJTRIDIAG with Pre)和不带消失稳和回溯措施的JTRIDIAG算法在无中断现象时的比较, 其中第一列的 k 是J_三对角化过程中的列号, 之所以出现了三次 $k = 2$, 是因为在矩阵的第二列实行了消失稳和回溯措施, 第二列给出的是带消失稳和回溯措施的MJTRIDIAG 算法相应的 $|\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)|$• 第三列给出的是使用第5部分的预处理后再用带消失稳和回溯措施的MJTRIDIAG 算法的相应的 $|\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)|$• 第四列给出的是不带消失稳和回溯措施的JTRIDIAG算法相应的 $|\boldsymbol{A}(k+1, k)/\boldsymbol{Z}(k, k)|$• 可见不带消失稳和回溯措施的JTRIDIAG算法虽无中断, 但某些Gauss矩阵的条件数比带消失稳和回溯措施的MJTRIDIAG算法要大得多• 且带预处理

的本算法比不带预处理的本算法稳定性更好• 值得注意的是, 预处理后就没有再发生严重失稳现象• 大量的数值实验表明预处理可以减少消失稳和回溯次数•

表4给出了用带消失稳和回溯措施的MJTRIDIAG算法计算出的特征值、预处理后的本算法计算出的特征值、不带消失稳和回溯措施的JTRIDIAG算法计算出的特征值的绝对误差的比较• 其中第一列是本算法计算出的特征值 λ的误差• 第二列给出的是预处理后的本算法计算出的特征值 λ的误差• 第三列给出的是不带消失稳和回溯措施的JTRIDIAG算法计算出的特征值, 可见由于多次发生严重失稳现角, 原算法计算出的特征值已毫无意义• 第四列给出的是本问题的精确特征值(由于Hamilton矩阵的特征值具有(4) 的性质, 我们只给出了部分特征值)• 可见本文算法计算出的特征值具有10~ 12个有效数字• 不带消失稳和回溯措施的JTRIDIAG算法计算出的特征值除一个实特征值外, 其他特征值已完全失真• 而预处理后本文算法计算出的特征值具有12~ 15个有效数字, 几乎达到机器精度•

例3取自[11]算例1, 例4取自于[11]算例2• 表5和表6的结构和意义除了第六列增加了本文算法的计算结果外与[11]中的表1的结构和意义完全相同•

我们还对阶数不同, 条件数不同的矩阵进行了大量的数值实验, 所有算例皆证实了本文的算法具有高的精度和良好的稳定性•

表1　　例1中 $|A(k+-1, k)/Z(k, k)|$ 的值

k	$\vert A(k+1, k)/Z(k, k)\vert$	$\vert A(k+1, k)/Z(k, k)\vert$	$\vert A(k+1, k)/Z(k, k)\vert$
1	2.592 114 234 111 782E-003	2.111 590 797 329 785E-003	2.592 114234 111 782E-003
2	1.777861 196 843 929E+016	7.205 842 436 811 865E+000	1.777 861 196 843 929E+016
2	9.891 210 147 724 910E+005		
2	6.158 529 349 595 562E-001		
3	4.494 212 651 080 097E+006	1.206 479 743 872 030E+000	1.332 329 676 810 446E+015
4	7.643 139 769 370 089E+005	3.654 884 617 478 983E+000	$Z(4,4)=0$, 中断
5	1.150 954 205 960 697E+003	8.008 002 825 341 964E-002	
6	3.783 353 623 763 489E+000	4.014 867 667 764 965E+000	
7	1.905 483 706 800 735E+002	1.846 977 991 787 582E+000	
8	6.298 205 947 195 798E+001	1.470 110 716 790 187E-001	

表2　　例1中特征值的绝对误差

$\vert\lambda-\lambda\vert$	$\vert\lambda-\lambda\vert$	λ
1.568 301 044 585 496E-010	1.891 820 033 961 267E-013	-6.688 650
1.790 567 694 115 452E-012	3.907 985 046 680 550E-014	-10.628 00
1.078 603 872 883 832E-010	2.131 628 207 280 300E-014	-21.996 20
4 707 036 208 675 495-010	3.526 656 667 363 483E-012	-10.698 15-32 174 60i
8.462 066 602 987 760E-010	2.131 628 207 280 300E-014	-39.443 10
7.389 909 099 141 649E-010	3.418 458 319 741 152E-012	-36.331 60-20 597 65i
5.618 829 845 843 720E-010	2.131 628 207 280 300E-014	-38.397 50

表 3 例 2 中 | $A(k+1, k)/Z(k, k)$ | 的值

k	$\vert A(k+1, k)/Z(k, k)\vert$	$\vert A(k+1, k)/Z(k, k)\vert$	$\vert A(k+1, k)/Z(k, k)\vert$
1	5.586 100 393 682 330E-003	7.898 022 493 159 091E-003	5.586 100 393 682 330E-003
2	5.543 319 113 477 248E+016	1.183 198 817 511 806E+000	5.543 319 113 477 247E+016
2	9.891 210 138 201 758E+005		
2	6.331 423 334 020 061E-001		
3	2.267 463 435 230 546E+005	5.249 702 478 939 774E-001	2.161 039 984 649 396E+015
4	1.904 860 520 260 430E+005	1.776 772 019 645 638E+000	4.030 994 052 857 680E+014
5	2.221 355 254 397 536E+005	4.296 784 516 168 871E+000	5.146 835 112 550 907E+014
6	3.319 213 796 838 172E+005	8.731 489 522 467 358E-001	5.939 691 883 416 462E+013
7	4.515 396 905 222 706E+003	9.304 681 083 670 656E-001	8.748 806 836 660 709E+012
8	3.207 527 118 243 190E+004	1.280 179 121 824 412E+001	6.267 064 881 083 799E+011
9	8.791 685 681 548 655E+005	3.037 726 932 586 039E+000	2.150 817 733 954 547E+003
10	6.222 735 925 240 173E+001	3.927 240 287 923 688E-001	3.752 433 904 531 520E+002
11	4.752 546 914 860 273E+000	1.271 562 773 692 639E+000	2.255 529 029 448 146E+000
12	1.016 536 745 651 306E+001	1.144 651 878 076 591E+000	4.193 500 191 454 012E+000
13	1.440 129 791 961 942E+000	3.106 793 448 456 030E-001	7.151 244 169 839 161E+001
14	3.314 576 386 426 958E+000	1.648 433 561 793 554E+000	3.920 303 937 400 028E+000

表 4 例 2 中特征值的绝对误差

$\vert\lambda-\lambda\vert$	$\vert\lambda-\lambda\vert$
1.537 134 863 838 218E-010	1.012 523 398 458 143E-013
5.562 872 287 009 197E-011	2.017 206 494 842 224E-013
2.561 984 244 286 338E-010	3.027 789 680 898 332E-013
2.465 228 021 719 668E-010	2.131 628 207 280 300E-014
2.685 218 467 912 810E-010	5.211 723 197 616 109E-013
4.250 028 488 565 310E-010	3.586 657 574 068 102E-013
4.506 190 975 916 979E-010	7.389 644 451 905 042E-013
4.836 309 700 930 275E-010	7.122 191 489 040 444E-012
5.868 958 916 466 293E-010	7.611 103 479 104 570E-013
λ	λ
-5.731 063 078 263 221	-6.944 55
-1.738 213 795 651 647-18.135 250 350 991 04i	-2.894 55-17.643i
-32.877 933 727 163 58-6.679 220 127 497 273i	-30.189 6-13.609 4i
-58.525 220 713 554 04	-36.910 35
-35.023 864 402 804 30-19.977 470 799 141 32i	-37.339 3-22.254 8i
-38.691 762 126 354 70-26.203 658 057 848 62i	-38.104 85-22.823 4i
-30.342 150 000 001 71	-30.342 15
-21.932 269 371 100 00-32.657 495 244 856 68i	-22.235 15-30.771 6i
-34.114 117 644 517 23-40.565 835 104 692 62i	-46.773 5-45.845 2i

表5 例3中绝对误差 $|\lambda - \tilde{\lambda}|$

λ	URVHQR	URVPSD	SQRED	LAPACK	MJTRIDIG with Pre
1	0	0	0	7.8E-16	3.3E-16
1E-2	5.5E-16	5.5E-16	5.5E-16	5.0E-17	2.0E-17
1E-4	7.7E-14	1.6E-18	1.6E-14	2.6E-18	5.0E-18
1E-6	4.1E-12	1.0E-18	1.5E-11	8.4E-18	2.2E-17
1E-8	1.7E-09	3.1E-17	2.2E-09	4.7E-17	4.6E-17

表6 例4中绝对误差 $|\tilde{\lambda} - \lambda^{QR}|$

$\lambda \approx$	$s(\lambda)$	URVHQR	URVPSD	SQRED	MJTRIDIG with Pre
0.284 7	1.8E-6	1.8E-9	2.7E-11	2.8E-9	9.9E-10
0.143 6	1.8E-6	2.7E-8	9.9E-10	7.6E-8	1.0E-8
0.081 22	3.8E-8	1.4E-7	5.9E-9	5.6E-7	3.8E-8
0.049 5	2.6E-8	2.3E-7	9.8E-9	1.4E-6	5.5E-8
0.031 02	5.5E-8	1.2E-7	5.0E-9	1.1E-6	2.5E-8

7 结　　论

本文提出的算法是一种保结构的计算Hamilton矩阵特征值的稳定的有效的算法. 它是一种QR型算法,在此算法中使用了消除严重失稳和避免中断的技术. 从而克服了同类算法会发生中断和严重失稳的致命弊端. 如果MJTRIDIAG算法与预处理措施结合使用,则往往大大改善后继列Jtridiag化的稳定性. 对一般问题往往只需采取一次减比与回溯的措施,就能有效地计算出问题的所有特征值. 由于采用了很少几次减比与回溯,计算量增加很少,计算出的特征值一般具有10~15个有效数字. 由于利用了Hamilton矩阵特征值的性质,采用了二次或四次位移,计算出一个特征值的平均迭代次数在0.6~1.1之间. 本文所提算法和其它保结构计算Hamilton矩阵特征值、特征不变子空间的算法([19, 12, 14])的算法结合将可能是一个极有价值的课题.

附　　录

A 的第1~3列为

1.124 124 238 400 000E+001	1.488 148 800 000 000E-002	1.967 911 200 000 000E-002
1.488 148 800 000 000E-002	-1.054 563 758 400 000E+001	3.048 081 600 000 000E-002
1.967 911 200 000 000E-002	3.048 081 600 000 000E-002	-2.512 718 111 999 999E+000
1.949 532 000 000 000E-002	1.070 620 800 000 000E-002	3.218 710 817 600 000E+001
2.462 164 800 000 000E-002	1.802 930 400 000 000E-002	3.658 303 200 000 000E-002
1.639 965 600 000 000E-002	2.044 099 200 000 000E-002	1.841 731 200 000 000E-002
2.491 656 000 000 000E-002	2.515 994 400 000 000E-002	2.478 520 800 000 000E-002
1.165 879 200 000 000E-002	2.125 504 800 000 000E-002	2.443 958 840 000 000E-002

2.346 278 400 000 000E-002	2.128 166 400 000 000E-002	1.114 276 800 000 000E-002

A 的第 4~6 列为

1.949 532 000 000 000E-002	2.462 164 800 000 000E-002	1.639 965 600 000 000E-002
1.070 620 800 000 000E-002	1.802 930 400 000 000E-002	2.044 099 200 000 000E-002
-3.216 209 182 400 000E+001	3.658 303 200 000 000E-002	1.841 731 200 000 000E-002
-2.483 051 040 000 000E+000	2.333 556 000 000 000E-002	1.934 143 200 000 000E-002
2.333 556 000 000 000E-002	1.066 507 691 200 000E+001	2.063 380 403 200 000E+001
1.934 143 200 000 000E-002	-2.056 149 596 800 000E+001	1.066 201 057 600 000E+001
1.902 242 400 000 000E-002	1.433 827 200 000 000E-002	2.969 061 600 000 000E-002
1.966 298 400 000 000E-002	2.316 412 800 000 000E-002	2.512 212 000 000 000E-002
6.193 368 000 000 000E-003	4.145 169 600 000 000E-002	1.442 301 600 000 000E-002

A 的第 7~9 列为

2.491 656 000 000 000E-002	1.165 879 200 000 000E-002	2.346 278 400 000 000E-002
2.515 994 400 000 000E-002	2.125 504 800 000 000E-002	2.128 166 400 000 000E-002
2.478 520 800 000 000E-002	2.443 958 400 000 000E-002	1.114 276 800 000 000E-002
1.902 242 400 000 000E-002	1.966 298 400 000 000E-002	6.193 368 000 000 000E-003
1.433 827 200 000 000E-002	2.316 412 800 000 000E-002	4.145 169 600 000 000E-002
2.969 061 600 000 000E-002	2.512 212 000 000 000E-002	1.442 301 600 000 000E-002
6.667 329 200 000 000E+000	8.583 048 000 000 000E-003	2.228 282 400 000 000E-002
8.583 048 000 000 000E-003	-2.464 840 112 000 000E+000	4.084 308 000 000 000E-002
2.228 282 400 000 000E-002	4.084 308 000 000 000E-002	-1.359 830 448 000 000E+000

F 的第 1~3 列为

3.722 905 678 800 000E+001	1.116 111 600 000 000E-002	1.475 933 400 000 000E-002
1.116 111 600 000 000E-002	-3.749 155 318 800 000E+001	2.286 061 200 000 000E-002
1.475 933 400 000 000E-002	2.286 061 200 000 000E-002	-9.908 151 084 000 000E+000
1.462 149 000 000 000E-002	8.029 656 000 000 000E-003	9.381 132 000 005 010E-003
1.846 623 600 000 000E-002	1.352 197 800 000 000E-002	2.743 727 400 000 000E-002
1.229 974 200 000 000E-002	1.533 074 400 000 000E-002	1.381 298 400 000 000E-002
1.868 742 000 000 000E-002	1.886 995 800 000 000E-001	1.858 890 600 000 000E-002
8.744 094 000 000 001E-003	1.594 128 600 000 000E-002	1.832 968 800 000 000E-002
1.759 708 800 000 000E-002	1.596 124 800 000 000E-002	8.357 076 000 000 000E-003

F 的第 4~6 列为

1.462 149 000 000 000E-002	1.846 623 600 000 000E-002	1.229 974 200 000 000E-002
8.029 656 000 000 000E-003	1.352 197 800 000 000E-002	1.533 074 400 000 000E-002
9.381 131 999 994 352E-003	2.743 727 400 000 000E-002	1.381 298 400 000 000E-002
-9.885 900 780 000 000E+000	1.750 167 000 000 000E-002	1.450 607 400 000 000E-002
1.750 167 000 000 000E-002	3.524 750 768 400 000E+001	2.711 552 400 000 094E-002
1.450 607 400 000 000E-002	2.711 552 399 999 740E-002	3.524 520 793 200 000E+001

1.426 681 800 000 000E-002	1.075 370 400 000 000E-002	2.226 796 200 000 000E-002
1.474 723 800 000 000E-002	1.737 309 600 000 000E-002	1.884 159 000 000 000E-002
4.645 026 000 000 001E-003	3.108 877 200 000 000E-002	1.081 726 200 000 000E-002

F 的第 7～9 列为

1.868 742 000 000 000E-002	8.744 094 000 000 001E-003	1.759 708 800 000 000E-002
1.886 995 800 000 000E-002	1.594 128 600 000 000E-002	1.596 124 800 000 000E-002
1.858 890 600 000 000E-002	1.832 968 800 000 000E-002	8.357 076 000 000 000E-003
1.426 681 800 000 000E-002	1.474 723 800 000 000E-002	4.645 026 000 000 000E-003
1.075 370 400 000 000E-002	1.737 309 600 000 000E-002	3.108 877 200 000 000E-002
2.226 796 200 000 000E-002	1.884 159 000 000 000E-002	1.081 726 200 000 000E-002
2.149 764 690 000 000E+001	6.437 286 000 000 000E-003	1.671 211 800 000 000E-002
6.437 286 000 000 000E-003	-9.819 630 084 000 000E+000	3.063 231 000 000 000E-002
1.671 211 800 000 000E-002	3.063 231 000 000 000E-002	-6.036 360 336 000 000E+000

Z 的第 1～3 列为

3.620 834 348 800 001E+001	-1.984 198 400 000 000E-002	-2.623 881 600 000 000E-002
-1.984 198 400 000 000E-002	-3.852 994 988 800 000E+001	-4.064 108 800 000 000E-002
-2.623 881 600 000 000E-002	-4.064 108 800 000 000E-002	-1.091 390 918 400 000E+001
-2.599 376 000 000 000E-002	-1.427 494 400 000 000E-002	-1.667 756 800 000 397E-002
-3.282 886 400 000 000E-002	-2.403 907 200 000 000E-002	-4.877 737 600 000 000E-002
-2.186 620 800 000 000E-002	-2.725 465 600 000 000E-002	-2.455 641 600 000 000E-002
-3.322 208 000 000 000E-002	-3.354 659 200 000 000E-002	-3.304 694 400 000 000E-002
-1.554 505 600 000 000E-002	-2.834 006 400 000 000E-002	-3.258 611 200 000 000E-002
-3.128 371 200 000 000E-002	-2.837 555 200 000 000E-002	-1.485 702 400 000 000E-002

Z 的第 4～6 列为

-2.599 376 000 000 000E-002	-3.282 886 400 000 000E-002	-2.186 620 800 000 000E-002
-1.427 494 400 000 000E-002	-2.403 907 200 000 000E-002	-2.725 465 600 000 000E-002
-1.667 756 799 999 687E-002	-4.877 737 600 000 000E-002	-2.455 641 600 000 000E-002
-1.095 346 652 800 000E+001	-3.111 408 000 000 000E-002	-2.578 857 600 000 000E-002
-3.111 408 000 000 000E-002	3.422 203 078 400 000E-001	-4.820 537 600 000 030E-002
-2.578 857 600 000 000E-002	-4.820 537 600 000 000E-002	3.422 611 923 200 000E+001
-2.536 323 200 000 000E-002	-1.911 769 600 000 000E-002	3.958 748 800 000 000E-002
-2.621 731 200 000 000E-002	-3.088 550 400 000 000E-002	-3.349 616 000 000 000E-002
-8.257 824 000 000 002E-003	-5.526 892 799 999 999E-002	-1.923 068 800 000 000E-002

Z 的第 7～9 列为

-3.322 208 000 000 000E-002	-1.554 505 600 000 000E-002	-3.128 371 200 000 000E-002
-3.354 659 200 000 000E-002	-2.834 006 400 000 000E-002	-2.837 555 200 000 000E-002
-3.304 694 400 000 000E-002	-3.258 611 200 000 000E-002	-1.485 702 400 000 000E-002
-2.536 323 200 000 000E-002	-2.621 731 200 000 000E-002	-8.257 824 000 000 000E-003
-1.911 769 600 000 000E-002	-3.088 550 400 000 000E-002	-5.526 892 799 999 999E-002

- 3. 958 748 800 000 000E- 002	- 3. 349 616 000 000 000E- 002	- 1. 923 068 800 000 000E- 002
2. 043 849 440 000 000E+ 001	- 1. 144 406 400 000 000E- 002	- 2. 971 043 200 000 000E- 002
- 1. 144 406 400 000 000E- 002	- 1. 088 421 318 400 000E+ 001	- 5. 445 744 000 000 000E- 002
- 2. 971 043 200 000 000E- 002	- 5. 445 744 000 000 000E- 002	- 7. 105 092 736 000 000E+ 000

致谢　本文作者感谢唐立民教授提出一个如此值得研究的课题, 感谢贾仲孝教授有洞察力的分析与指导•

[参　考　文　献]

[1] Byers R. A Hamiltonian QR_algirithm[J]. SIAM J Sci Statist Comput, 1986, **7**: 212—229.

[2] Bunse Gerstner A, Byers R, Mehrmann V. A chat of numerical methods for structured eigenvalue problems[J]. SIAM J Matrix Anal Appl, 1992, **13**: 419—453.

[3] Bunse_Gerstner A, Mehrmann V. A symplectic QR_like algorithm for the solution of the real algebraic Riccati equation[J]. IEEE Trans Automat Control, 1986, **31**: 1104—1113.

[4] Hench J J, Laub A J. Numerical solution of the discrete_time periadic Riccati equation[J]. IEEE Trans Automat Control, 1994, **39**: 1197—1210.

[5] Lin W W. A new method for computing the closed loop eigenvalues of a discrete_time algebraic Riecati equation[J]. Linear Algebra Appl, 1987, **96**: 157—180.

[6] Lu L Z, Lin W W. An iterative algorithm for the solution of a discrete_time algebraic Riccati equation [J]. Linear Algebra Appl, 1993, **188/189**: 465—488.

[7] Lin W W, Wang C. On computing stable Lagrangian subspaces of Hamiltonian martices and symplectic pencils[J]. SIAM J Matrix Anal Appl, 1997, **18**: 590—614.

[8] Pappas C, Laub A J, Sandell N R. On the numerical solution of the discrete_time algebraic Reiccati equation[J]. IEEE Trans Autom Control, 1980, **25**: 631—641.

[9] Patel R V. On computing the eigenvalues of a symplectic pencils[J]. Linear Algebra Appl, 1993, **188**: 591—611.

[10] Patel R V, Lin Z, Misra P. Computation of stable invariant subspaces of Hamiltonian matrices[J]. SIAM J Matrix Anal Appl, 1994, **15**: 284—298.

[11] Benner P, Mehrmann V, Xu H. A numerically stable, structure preserving method for computing the eigenvalues oy real Hamiltonian or symplectic pencils[J]. Numer Math, 1998, **78**: 329—358.

[12] Bunse Gerstner A, Mehrmann V, Watkins D. An SR algorithm for Hamiltonian matrices, based on Gaussian elimination[J]. Methods Oper Res, 1989, **58**: 339—358.

[13] Mehrmann V. A symplectic orthogonal method for single input or single output discrete time optimal quadrtic control problems[J]. SIAM J Matrix Anal Appl, 1988, **9**: 221—247.

[14] Van Loan C. A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix[J]. Linear Algebra Appl, 1984, **16**: 233—251.

[15] 许波, 刘征. Matab 工程数学应用[M]. 北京: 清华大学出版社, 2000.

[16] Golub G H, Van Loan C. Matrix Computations[M]. Baltimore: The Johns Hopkins University Press, 1996.

[17] Stewart G W. Introduction to Matrix Computations[M]. New York: Academic, 1973.

[18] Wilkinson J H. The Algebraic Eigenvalue Problem[M]. Clarendon: Oxford, 1965.

[19] Benner P, Faåbender H. An implicity restarted symplectic lanczos method for the Hamiltonian eigenvalue problem[J]. Linear Algebra Appl, 1997, **263**: 75—111.

An Effcient and Stable Structure Preserving Algorithm for Computing the Eigenvalues of a Hamiltonian Matrix

YAN Qing_you[1,2], XIONG Xi_wen[1]
(1. Center of Advanced Design Technology, Dalian University, Dalian 116622 P R China;
2. Department of Economics and Statistics, Shandong Finance Institute, Jinan 250014, P R China)

Abstract: An efficient and stable structure preserving algorithm, which is a variant of the QR like (SR) algorithm due to Bunse_Gerstner and Mehrmann, is presented for computing the eigenvalues and stable invariant subspaces of a Hamiltonian matrix. In the algorithm two strategies are employed, one of which is called dis_unstabilization technique and the other is preprocessing technique. Together with them, a so called ratio_reduction equation and a backtrack technique are introduced to avoid the instability and breakdown in the original algorithm. It is shown that the new algorithm can overcome the instability and breakdown at low cost. Numerical results have demonstrated that the algorithm is stable and can compute the eigenvalues to very high accuracy.

Key words: Hamiltonian matrix; QR like algorihm; eigenvalue; stability; dis_unstabilization; backtrack technique; ratio_reduction