doi:10.3969/j.issn.1673-9833.2018.01.001

# 一类过阻尼系统二次方程的快速迭代算法

金继承，董　宁，蔡汝忠

（湖南工业大学 理学院，湖南 株洲 412007）

**摘　要**：考虑一类来自过阻尼系统的二次矩阵方程数值求解问题，针对方程系数矩阵的结构特点，设计了一种快速求解方程的迭代算法，给出了这类算法具体的迭代格式和收敛性。数值实验表明，提出的算法能够有效地求解此类方程具有实际意义的解。

**关键词**：过阻尼系统；二次方程；保结构加倍算法；Toeplitz 矩阵；Hankel 矩阵

**中图分类号**：O151.21；TP273　　**文献标志码**：A　　**文章编号**：1673-9833(2018)01-0001-05

# A Quick Iterative Algorithm for Quadratic Equations in an Overdamped System

JIN Jicheng，DONG Ning，CAI Ruzhong

（College of Science，Hunan University of Technology，Zhuzhou Hunan 412007，China）

**Abstract**：Aiming at the numerical solution of quadratic matrix equations in an overdamped system, and based on an analysis of the structural properties of the coefficient matrix of the equation, an iterative algorithm for a quick solution of the equation has thus been proposed, with its specific iterative format and convergence subsequently given. The numerical experiments show that the proposed algorithm is of practical significance for the effective solution of this kind of equation.

**Keywords**：overdamped system；quadratic equation；structure-preserving algorithm；Toeplitz matrix；Hankel matrix

## 1　研究背景

在研究阻尼系统的振动问题 [1-2] 时，需要求解式（1）所示的二次特征值问题（quadratic eigenvalue problem，QEP）。

$$Q(\lambda)\boldsymbol{x}=(\lambda^2\boldsymbol{M}+\lambda\boldsymbol{D}+\boldsymbol{K})\boldsymbol{x}=\boldsymbol{0},\ \boldsymbol{M},\boldsymbol{D},\boldsymbol{K}\in\mathbf{R}^{n\times n},\quad (1)$$

式中：$\lambda$ 为特征值；

$\boldsymbol{x}$ 为属于 $\lambda$ 的特征向量；

$\boldsymbol{M},\boldsymbol{D},\boldsymbol{K}$ 为系数矩阵，其结构为

$$\boldsymbol{M}=m\boldsymbol{I}_n,$$

$$\boldsymbol{D}=\boldsymbol{P}_n\mathrm{diag}(d,\cdots,d,0)\boldsymbol{P}_n^{\mathrm{T}}+\tau\boldsymbol{I}_n,$$

$$\boldsymbol{K}=\boldsymbol{P}_n\mathrm{diag}(k,\cdots,k,0)\boldsymbol{P}_n^{\mathrm{T}}+\kappa\boldsymbol{I}_n,$$

其中，$m$ 为物体的质量，$\boldsymbol{I}_n$ 为 $n$ 阶单位矩阵，$d,\tau$ 为阻尼系数，$k,\kappa$ 为弹性系数，$\boldsymbol{P}_n=\begin{bmatrix}1&0&\cdots&0\\-1&1&\ddots&\vdots\\\vdots&\ddots&\ddots&0\\0&\cdots&-1&1\end{bmatrix}$。

在质量 - 弹簧阻尼系统中，每个质量为 $m$ 的物体通过一个弹簧和一个阻尼器连接到下一个物体，其弹性系数和阻尼系数分别为 $k$ 和 $d$；同时每个物体也通过弹簧和阻尼器连接到地面，其弹性系数和阻尼系

**收稿日期**：2017-12-01

**基金项目**：湖南省自然科学基金资助项目（2017JJ2071），湖南省教育厅优秀青年基金资助项目（17B071），湖南省教育厅基金资助项目（17C0466）

**作者简介**：金继承（1963-），男，湖南汨罗人，湖南工业大学教授，博士生导师，主要从事偏微分方程数值方法理论及应用方面的教学与研究，E-mail：jcjin2008@sina.com

数分别为 $\kappa$ 和 $\tau$ [1-2]。在实际应用中，人们往往需要检测质量－弹簧系统是否处于过阻尼（overdamped）状态，即系统系数矩阵是否满足下面的定义 1。

**定义 1** 若 $\boldsymbol{M}>0$，$\boldsymbol{D}>0$，$\boldsymbol{K}\geqslant 0$，并且存在常数 $\mu>0$ 使得 $\boldsymbol{D}>\mu\boldsymbol{M}+\mu^{-1}\boldsymbol{K}$，则称阻尼系统处于过阻尼状态。这里 2 个对称矩阵 $\boldsymbol{M}_1$、$\boldsymbol{M}_2$ 满足不等式 $\boldsymbol{M}_1\geqslant\boldsymbol{M}_2$ (或 $\boldsymbol{M}_1>\boldsymbol{M}_2$) 表示 $\boldsymbol{M}_1-\boldsymbol{M}_2$ 为半正定（或正定）矩阵。

判别系统是否处于过阻尼状态需要求解对应的二次特征值问题（1）。传统的做法是将这一问题转化为一个 $2n$ 维的线性广义特征值问题 [3]，从而采用 QZ 算法进行求解 [3]，这种方法计算的复杂度一般较高。Guo C. H. 等 [4-5] 注意到系统（1）的过阻尼状态可以通过一种解元方法（solvent method）来判别，即先求解二次矩阵方程

$$Q(\boldsymbol{S})=\boldsymbol{M}\boldsymbol{S}^2+\boldsymbol{D}\boldsymbol{S}+\boldsymbol{K}=\boldsymbol{0} \tag{2}$$

的 2 个极端解（分别对应到系统（1）中 $n$ 个模最大的特征值和 $n$ 个模最小的特征值的 2 个解矩阵）；然后再分别求每个极端解对应的一个特征值来判别系统是否处于过阻尼状态，这样就避免了解一个 $2n$ 维广义特征值问题。这种方法的关键一步在于能否有效地求解二次矩阵方程（2）。一般来说常用的求解方法有不动点迭代法 [4]、牛顿法 [2]、循环约化方法 [6-7] 和保结构加倍算法 [8-9] 等。当系统处于过阻尼状态时，采用不动点方法求解方程一般是线性收敛的，而后 3 类方法都具有二次收敛性。保结构加倍算法和循环约化方法迭代格式已被证明是一致的 [4]，并且较牛顿法迭代格式更为简单，因此常被用来求解方程（2）。

值得注意的是，上述几种迭代算法一般来说每步迭代都需要 $O(n^3)$ 的计算复杂度，而在质量－弹簧系统中，系数矩阵 $\boldsymbol{M}$、$\boldsymbol{D}$ 与 $\boldsymbol{K}$ 往往都具有特殊的结构。因此，本文考虑利用潜在的矩阵结构来建立计算复杂度更低的迭代算法求解方程（2）。

本文将方程（2）的系数矩阵推广到了一类对称的 Toeplitz-plus-Hankel 矩阵（T+H 矩阵）[10-11]，并给出一个简单的矩阵求逆迭代方法来实现原有保结构加倍算法。由于这种矩阵求逆迭代是基于 T+H 矩阵的快速逆公式 [10]，因此本文给出的保结构加倍算法每步计算的复杂度能够降低到 $O(n^2)$。数值实验表明，本文提出的保结构加倍算法能够以更少的 CPU 处理时间达到与原来的保结构加倍算法差不多的计算精度。

## 2 Toeplitz-plus-Hankel 矩阵

为了更好地给出本文算法的迭代格式，本章介绍 Toeplitz-plus-Hankel 矩阵及其性质。

**定义 2** 设

$$\boldsymbol{T}=(t_{i-j})_{i,\,j=1}^{n}=\begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-(n-1)} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix},$$

$$\boldsymbol{H}=(h_{i+j-2})_{i,\,j=1}^{n}=\begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_1 & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & h_{2n-1} \\ h_{n-1} & \cdots & h_{2n-1} & h_{2n-2} \end{bmatrix},$$

分别为 $n$ 阶 Toeplitz 矩阵和 Hankel 矩阵，则 $\boldsymbol{M}=\boldsymbol{T}+\boldsymbol{H}$ 称为 $n$ 阶 Toeplitz-plus-Hankel（T+H）矩阵。

**引理 1**[11] $n$ 阶矩阵 $\boldsymbol{M}$ 为 T+H 矩阵，当且仅当 $\boldsymbol{M}$ 的 $n-2$ 阶中心子矩阵

$$\boldsymbol{W}_n\boldsymbol{M}-\boldsymbol{M}\boldsymbol{W}_n$$

为零矩阵，其中 $\boldsymbol{W}_n$ 为满足 $(\boldsymbol{W}_n)_{11}=(\boldsymbol{W}_n)_{nn}=(\boldsymbol{W}_n)_{i,\,i+1}=(\boldsymbol{W}_n)_{i+1,\,i}=1(i=1,2,\cdots,n-1)$，而其余元素为零的三对角矩阵。

由引理 1 知，$n$ 阶 T+H 矩阵 $\boldsymbol{M}$ 满足错位结构（displacement structure）[12]

$$\boldsymbol{W}_n\boldsymbol{M}-\boldsymbol{M}\boldsymbol{W}_n=\boldsymbol{G}\boldsymbol{F}^{\mathrm{T}}, \tag{3}$$

其中 $\boldsymbol{G}=(\boldsymbol{g}_1,\boldsymbol{g}_2,\boldsymbol{g}_3,\boldsymbol{g}_4)$，$\boldsymbol{F}=(\boldsymbol{f}_1,\boldsymbol{f}_2,\boldsymbol{f}_3,\boldsymbol{f}_4)$，而列向量 $\boldsymbol{g}_i$，$\boldsymbol{f}_i\in\mathbf{R}^n$ $(i=1,2,3,4)$，即 T+H 矩阵的 $\boldsymbol{W}_n$ 错位秩（displacement rank）不超过 4。特别地，如果式（3）的右边为零，则称 $\boldsymbol{M}$ 为 $\boldsymbol{W}_n$ 可交换的 T+H 矩阵。下面的引理 2 给出了 $\boldsymbol{W}_n$ 可交换 T+H 矩阵中各元素间的关系。

**引理 2** 设矩阵 $\boldsymbol{M}=\boldsymbol{T}+\boldsymbol{H}=(t_{i-j})_{i,\,j=1}^{n}+(h_{i+j-2})_{i,\,j=1}^{n}$，则 $\boldsymbol{M}$ 为 $\boldsymbol{W}_n$ 可交换 T+H 矩阵，当且仅当

$$\begin{cases} t_1=t_{-1},\ h_n=h_{n-2}, \\ t_{1-i}+h_{i-1}=t_{-i}+h_{i-2}, \\ t_{i-1}+h_{2n-1-i}=t_i+h_{2n-i}, \quad i=2,3,\cdots,n-1。\\ t_{i-1}+h_{i-1}=t_i+h_{i-2}, \\ t_{1-i}+h_{2n-1-i}=t_{-i}+h_{2n-i}, \end{cases}$$

**证明** 由引理 1 可知，矩阵 $\boldsymbol{M}$ 的零元素只能出现在第一行（列）和最后一行（列）的位置。再由 $\boldsymbol{W}_n$ 可交换矩阵的定义，可以直接计算出上式中矩阵元素间的关系。证毕。

**定义 3**[11] 设矩阵

$$\boldsymbol{J}=(\delta_{i,\,n+1-j})_{i,\,j=1}^{n}=\begin{bmatrix} 0 & \cdots & 1 \\ \vdots & \iddots & \vdots \\ 1 & \cdots & 0 \end{bmatrix},$$

若 $n$ 阶 T+H 矩阵 $\boldsymbol{M}$ 满足等式 $\boldsymbol{J}\boldsymbol{M}\boldsymbol{J}=\boldsymbol{M}$，则称 $\boldsymbol{M}$ 为中心对称的（centrosymmetric）。

由定义 3 易知，若 $n$ 阶 T+H 矩阵 $\boldsymbol{M}$ 的 Toeplitz 矩阵部分是关于从左上到右下对角线对称，并且其 Hankel 矩阵部分是关于从右上到左下对角线对称，则 $\boldsymbol{M}$ 为中心对称的。

**引理 3**　1）设 $\boldsymbol{M}$ 为 $n$ 阶非奇异矩阵，则 $\boldsymbol{M}$ 为中心对称、$\boldsymbol{W}_n$ 可交换 T+H 矩阵，等价于 $\boldsymbol{M}^{-1}$ 为中心对称、$\boldsymbol{W}_n$ 可交换 T+H 矩阵。

2）设 $\boldsymbol{M}_1$、$\boldsymbol{M}_2$、$\boldsymbol{M}_3$、$\boldsymbol{M}_4$ 为 $n$ 阶中心对称的 $\boldsymbol{W}_n$ 可交换 T+H 矩阵，则 $\boldsymbol{M}_1\boldsymbol{M}_2+\boldsymbol{M}_3\boldsymbol{M}_4$ 也为一个 $n$ 阶中心对称的 $\boldsymbol{W}_n$ 可交换 T+H 矩阵。

**证明**　因为非奇异中心对称矩阵的逆、积、和及这些运算的线性组合形成的新矩阵仍为中心对称的，因此由 $\boldsymbol{W}_n$ 可交换 T+H 矩阵的定义可知，引理 3 的结论成立。证毕。

## 3　快速迭代算法

### 3.1　保结构加倍算法

保结构加倍算法和循环约化算法是目前求解二次矩阵方程（2）较为流行的算法。这 2 种算法和牛顿法一样都具有二次收敛性，但是其迭代格式较牛顿法更为简单。在过阻尼条件下，求解方程（2）的保结构加倍算法迭代格式和循环约化算法迭代格式本质上是一致的 [4]。因此，本文只考虑采用保结构加倍算法求解方程（2）。

**算法 1**　求解方程（2）的保结构加倍算法。

第 1 步：输入 $\boldsymbol{S}_0=\boldsymbol{D}$，$\boldsymbol{M}_0=\boldsymbol{M}$，$\boldsymbol{H}_0=\boldsymbol{0}$，$\boldsymbol{K}_0=\boldsymbol{K}$；

第 2 步：For $k=0, 1, 2, \cdots$, 直到收敛，

$$\boldsymbol{S}_{k+1}=\boldsymbol{S}_k-\boldsymbol{M}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{K}_k,$$
$$\boldsymbol{M}_{k+1}=\boldsymbol{M}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{M}_k,$$
$$\boldsymbol{H}_{k+1}=\boldsymbol{H}_k+\boldsymbol{K}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{M}_k,$$
$$\boldsymbol{K}_{k+1}=\boldsymbol{K}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{K}_k。$$

上述算法在每步迭代中需计算矩阵 $\boldsymbol{S}_k-\boldsymbol{H}_k$ 的逆。实际上当阻尼系统（1）在过阻尼状态下，可以找到正常数 $u>0$，使得对第 $k$ 次迭代，矩阵序列满足

$$\boldsymbol{M}_k>0,\ \boldsymbol{K}_k>0,\ \boldsymbol{S}_k-\boldsymbol{H}_k>u^{2^k}\boldsymbol{M}_k+u^{-2^k}\boldsymbol{K}_k,$$

从而保结构加倍算法 1 是适定的（well-defined）[4]。由于上述算法的收敛性可以由过阻尼状态下循环约化算法收敛性得到 [4–5]，从而可得出定理 1。

**定理 1**　若过阻尼系统（1）处于过阻尼状态下，则保结构加倍算法 1 产生的矩阵迭代序列满足：

1）对任意的矩阵范数，$\|\boldsymbol{M}_k\|$ 和 $\|\boldsymbol{K}_k\|$ 二次收敛到零；

2）矩阵序列 $\{\boldsymbol{S}_k\}$ 二次收敛到一个非奇异的矩阵 $\boldsymbol{S}$，并且方程（2）的 2 个极端解可以表示为

$$\boldsymbol{S}_1=-\boldsymbol{S}^{-1}\boldsymbol{K},\ \boldsymbol{S}_2=-\boldsymbol{M}\boldsymbol{S}^{-1}。$$

### 3.2　快速迭代格式

保结构加倍算法 1 的第 2 步每次迭代需要计算一个逆矩阵和几个矩阵的乘积。若不考虑矩阵结构，需要对矩阵 $\boldsymbol{S}_k-\boldsymbol{H}_k$ 作 Cholesky 分解，即 $\boldsymbol{S}_k-\boldsymbol{H}_k=\boldsymbol{L}_k\boldsymbol{L}_k^{\mathrm{T}}$。若记 $\boldsymbol{V}_k=\boldsymbol{L}_k^{-1}\boldsymbol{M}_k$，$\boldsymbol{U}_k=\boldsymbol{L}_k^{-1}\boldsymbol{K}_k$，则算法第 2 步迭代中有如下分解：

$$\boldsymbol{M}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{K}_k=\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{U}_k,$$
$$\boldsymbol{K}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{M}_k=\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{V}_k,$$
$$\boldsymbol{M}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{M}_k=\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k,$$
$$\boldsymbol{K}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{K}_k=\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{U}_k。$$

因此，每步迭代计算的复杂度约为 $19n^3/3$。

然而阻尼系统的系数矩阵实际上具有特殊的矩阵结构，因此上述算法的计算复杂度有望进一步降低。实际上，方程（2）中系数矩阵 $\boldsymbol{M}$、$\boldsymbol{D}$、$\boldsymbol{K}$ 都具有如下 T+H 矩阵结构形式

$$\boldsymbol{R}=\left(r_{|i-j|}+\begin{cases} r_{i+j-1} & i+j\leqslant n+1,\\ r_{2n+1-i-j} & i+j>n+1 \end{cases}\right)_{i,\,j=1}^{n},\qquad（4）$$

其中下标 $(i,j)$ 表示第 $i$ 行第 $j$ 列的元素。

若在 $\boldsymbol{R}$ 中取 $r_0=m$，$r_1=r_2=\cdots=r_n=0$ 时，则 $\boldsymbol{R}=\boldsymbol{M}$；取 $r_0=2d+\tau$，$r_1=-d$，$r_2=r_3=\cdots=r_n=0$ 时，则 $\boldsymbol{R}=\boldsymbol{D}$；取 $r_0=2k+\kappa$，$r_1=-k$，$r_2=r_3=\cdots=r_n=0$ 时，则 $\boldsymbol{R}=\boldsymbol{K}$。此外，矩阵 $\boldsymbol{R}$ 的 Toeplitz 矩阵部分是关于左上到右下对角对称，且其 Hankel 矩阵部分是关于右上到左下对角对称的。因此由引理 2 知，$\boldsymbol{R}$ 是中心对称的 $\boldsymbol{W}_n$ 可交换 T+H 矩阵。基于上述观察和已有的关于 $\boldsymbol{W}_n$ 可交换 T+H 矩阵快速求逆方法 [10]，可得出定理 2。

**定理 2**　记 $\boldsymbol{e}_n$ 为 $n$ 阶单位矩阵的最后一列。若式（4）中 $\boldsymbol{W}_n$ 可交换 T+H 矩阵 $\boldsymbol{R}$ 非奇异，则其逆矩阵 $\boldsymbol{R}^{-1}$ 的列向量 $\boldsymbol{u}_i(i=1, 2, \cdots, n)$ 可以由如下公式计算：

$$\boldsymbol{u}_n=\boldsymbol{R}^{-1}\boldsymbol{e}_n,$$
$$\boldsymbol{u}_{n-1}=\boldsymbol{W}_n\boldsymbol{u}_n-\boldsymbol{u}_n,$$
$$\boldsymbol{u}_{i-1}=\boldsymbol{W}_n\boldsymbol{u}_i-\boldsymbol{u}_{i+1},$$

其中 $i=n-1, n-2, \cdots, 3, 2$。

**证明**　由式（3）知

$$\boldsymbol{W}_n\boldsymbol{R}^{-1}-\boldsymbol{R}^{-1}\boldsymbol{W}_n=-\sum_{i=1}^{4}\boldsymbol{x}_i\boldsymbol{y}_i,$$

其中 $\boldsymbol{x}_i$ 和 $\boldsymbol{y}_i$ 分别是方程 $\boldsymbol{R}\boldsymbol{x}_i=\boldsymbol{g}_i$ 和 $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{y}_i=\boldsymbol{f}_i$ 的解向量（$i=1, 2, 3, 4$）。

又由于 $\boldsymbol{R}$ 是中心对称的 T+H 矩阵，则上述方程等价于

$$\boldsymbol{R}\boldsymbol{x}_1=\boldsymbol{e}_1,\ \boldsymbol{R}\boldsymbol{x}_2=\boldsymbol{e}_n,\ \boldsymbol{R}\boldsymbol{x}_3=\boldsymbol{0},\ \boldsymbol{R}\boldsymbol{x}_4=\boldsymbol{0},\qquad（5）$$

且 $\boldsymbol{y}_1=\boldsymbol{x}_3$，$\boldsymbol{y}_2=\boldsymbol{x}_4$，$\boldsymbol{y}_3=\boldsymbol{x}_1$，$\boldsymbol{y}_4=\boldsymbol{x}_2$，其中 $\boldsymbol{e}_1$ 为单位

矩阵的第一列。

设 $\boldsymbol{u}_i$ 为方程 $\boldsymbol{Ru}_i=\boldsymbol{e}_i$ 的解，$\boldsymbol{e}_i$ 为单位矩阵的第 $i$ 列，由 $\boldsymbol{W}_n\boldsymbol{R}-\boldsymbol{RW}_n=\boldsymbol{0}$，知 $\boldsymbol{W}_n\boldsymbol{e}_i-\boldsymbol{RW}_n\boldsymbol{u}_i=\boldsymbol{0}$。

另一方面由直接运算可得

$$\boldsymbol{W}_n\boldsymbol{e}_n=\boldsymbol{e}_{n-1}+\boldsymbol{e}_n,$$
$$\boldsymbol{W}_n\boldsymbol{e}_i=\boldsymbol{e}_{i-1}+\boldsymbol{e}_{i+1},$$

其中 $i$=2, 3, $\cdots$, $n$−1。

因此有

$$\boldsymbol{R}(\boldsymbol{W}_n\boldsymbol{u}_n-\boldsymbol{u}_n)=\boldsymbol{e}_{n-1},$$
$$\boldsymbol{R}(\boldsymbol{W}_n\boldsymbol{u}_i-\boldsymbol{u}_{i+1})=\boldsymbol{e}_{i-1},$$

其中 $i$=2, 3,$\cdots$, $n$。从而定理得证。

由定理 2 知，在计算 $\boldsymbol{R}$ 的逆矩阵时，需要求解方程 $\boldsymbol{Rx}_2=\boldsymbol{e}_n$ 获得 $\boldsymbol{x}_2$（即 $\boldsymbol{u}_n$），而这可以通过一个具有 $O(n_2)$ 计算量的递归过程求出。为此先给出递归运算的引理 4。

**引理 4** 设 $\boldsymbol{R}=(r_{|i-j|}+r_{i+j-1})_{i,j=1}^{n}$，若其每个 $m$ 阶子矩阵 $\boldsymbol{R}^{(m)}=(r_{|i-j|}+r_{i+j-1})_{i,j=1}^{m}$ $(1\leqslant m\leqslant n)$ 可逆，且对应有 $m$ 维列向量 $\boldsymbol{u}^{(m)}$ 和 $\boldsymbol{v}^{(m)}$ 满足方程组

$$\begin{cases}\boldsymbol{R}^{(m)}\boldsymbol{v}^{(m)}=-\boldsymbol{g}^{(m)},\\ \boldsymbol{R}^{(m)}\boldsymbol{u}^{(m)}=-\boldsymbol{e}_m^{(m)},\end{cases}\tag{6}$$

其中 $\boldsymbol{g}^{(m)}=(r_m+r_{m+1}-r_{m-1}-r_m, r_{m-1}+r_{m+2}-r_{m-2}-r_{m+1},\cdots, r_1+r_{2m}-r_0-r_{2m-1})^{\mathrm{T}}$，$\boldsymbol{e}_m^{(m)}$ 为 $m$ 维单位矩阵 $\boldsymbol{I}_m$ 的最后一列。则方程组（6）中的 $\boldsymbol{v}^{(m)}$ 有递归公式

$$\boldsymbol{v}^{(m+1)}=\left(\boldsymbol{W}^{(m+1)}-\left(1+\frac{\gamma_m}{\delta_m}-\frac{\lambda_m}{\delta_{m-1}}\right)\boldsymbol{I}_{m+1}\right)\begin{bmatrix}\boldsymbol{v}^{(m)}-\boldsymbol{e}_m^{(m)}\\1\end{bmatrix}-\frac{\delta_m}{\delta_{m-1}}\begin{bmatrix}\boldsymbol{v}^{(m-1)}-\boldsymbol{e}_{m-1}^{(m-1)}\\1\\0\end{bmatrix}。$$

其中 $\gamma_m=\left(\boldsymbol{g}^{(m+1)}\right)^{\mathrm{T}}\begin{bmatrix}\boldsymbol{v}^{(m)}-\boldsymbol{e}_m^{(m)}\\1\end{bmatrix}$；

$\boldsymbol{W}^{(m)}$ 为与 $\boldsymbol{W}_n$ 具有相同结构的 $m$ 阶矩阵；

$$\lambda_m=\left(\boldsymbol{f}^{(m+1)}\right)^{\mathrm{T}}\begin{bmatrix}\boldsymbol{v}^{(m)}-\boldsymbol{e}_m^{(m)}\\1\end{bmatrix},$$

$$\delta_m=\left(\boldsymbol{f}^{(m)}\right)^{\mathrm{T}}\left(\boldsymbol{v}^{(m)}-\boldsymbol{e}_m^{(m)}\right)+r_0+r_{2m+1}\neq 0,$$
$$\boldsymbol{f}^{(m)}=(r_m+r_{m+1}, r_{m-1}+r_{m+2},\cdots, r_1+r_{2m})^{\mathrm{T}}。$$

**证明** 首先递归公式中 $\delta_m$ 为非零的数，否则 $\boldsymbol{R}^{(m+1)}\left(\left(\boldsymbol{v}^{(m)}-\boldsymbol{e}_m^{(m)}\right)^{\mathrm{T}},1\right)^{\mathrm{T}}=\boldsymbol{0}$，即 $\boldsymbol{R}^{(m+1)}$ 为奇异矩阵，矛盾。其次由等式 $\boldsymbol{g}^{(m)}=\boldsymbol{f}^{(m)}-\boldsymbol{R}^{(m)}\boldsymbol{e}_m^{(m)}$ 得

$$\boldsymbol{R}^{(m+1)}\begin{bmatrix}\boldsymbol{v}^{(m)}-\boldsymbol{e}_m^{(m)}\\1\end{bmatrix}=\begin{bmatrix}\boldsymbol{0}^{(m)}\\\delta_m\end{bmatrix},\tag{7}$$

$$\boldsymbol{R}^{(m+1)}\begin{bmatrix}\boldsymbol{v}^{(m-1)}-\boldsymbol{e}_{m-1}^{(m-1)}\\1\\0\end{bmatrix}=\begin{bmatrix}\boldsymbol{0}^{(m-1)}\\\delta_{m-1}\\\lambda_{m-1}\end{bmatrix}。$$

综合上面 2 个方程可得等式

$$\boldsymbol{R}^{(m+1)}\left\{\left(1-\frac{\lambda_{m-1}}{\delta_{m-1}}\right)\begin{bmatrix}\boldsymbol{v}^{(m)}-\boldsymbol{e}_m^{(m)}\\1\end{bmatrix}+\frac{\delta_m}{\delta_{m-1}}\begin{bmatrix}\boldsymbol{v}^{(m-1)}-\boldsymbol{e}_{m-1}^{(m-1)}\\1\\0\end{bmatrix}\right\}=\begin{bmatrix}\boldsymbol{0}^{(m-1)}\\\delta_m\\\delta_m\end{bmatrix}。$$

另一方面，有错位方程

$$\boldsymbol{W}^{(m+1)}\boldsymbol{R}^{(m+1)}-\boldsymbol{R}^{(m+1)}\boldsymbol{W}^{(m+1)}=$$
$$\boldsymbol{g}^{(m+1)}\left(\boldsymbol{e}_{m+1}^{(m+1)}\right)^{\mathrm{T}}-\boldsymbol{e}_{m+1}^{(m+1)}\left(\boldsymbol{g}^{(m+1)}\right)^{\mathrm{T}},$$

其中 $\boldsymbol{g}^{(m+1)}$ 为方程组（6）中定义的向量。

在上述错位方程两端同时右乘向量 $\begin{bmatrix}\boldsymbol{v}^{(m)}-\boldsymbol{e}_m^{(m)}\\1\end{bmatrix}$，得到等式

$$\boldsymbol{R}^{(m+1)}\left\{\left(\boldsymbol{W}^{(m+1)}-\frac{\gamma_m}{\delta_m}\boldsymbol{I}_m\right)\begin{bmatrix}\boldsymbol{v}^{(m)}-\boldsymbol{e}_m^{(m)}\\1\end{bmatrix}\right\}-\begin{bmatrix}\boldsymbol{0}^{(m-1)}\\\delta_m\\\delta_m\end{bmatrix}=-\boldsymbol{g}^{(m+1)}。$$

最后，联合两个等式即可以证得引理 4 中的递归公式。

基于上述引理 4，有如下定理 3。

**定理 3** 线性方程 $\boldsymbol{Rx}_2=\boldsymbol{e}_n$ 的解为

$$\boldsymbol{x}_2=\frac{1}{\delta_{n-1}}\begin{bmatrix}\boldsymbol{v}^{(n-1)}-\boldsymbol{e}_{n-1}^{(n-1)}\\1\end{bmatrix},$$

其中 $\boldsymbol{e}_{n-1}^{(n-1)}$ 为 $n$−1 阶单位矩阵 $\boldsymbol{I}_{n-1}$ 的最后一列，而 $n$−1 维列向量 $\boldsymbol{v}^{(n-1)}$ 和常量 $\delta_{n-1}$ 由引理 4 给出。

**证明** 在式（7）中，令 $m$=$n$−1，两边同除 $\delta_{n-1}$ 即定理得证。

根据定理 3，可以给出算法 1 的一个结构化的计算版本，其每步迭代只有 $O(n^2)$ 计算复杂度。

**算法 2** 快速求解方程（2）的迭代格式

第 1 步：输入 $\boldsymbol{S}_0=\boldsymbol{D}$，$\boldsymbol{M}_0=\boldsymbol{M}$，$\boldsymbol{H}_0=\boldsymbol{0}$，$\boldsymbol{K}_0=\boldsymbol{K}$。

第 2 步：For $k$= 0, 1, 2, $\cdots$，直到收敛，

1）用定理 2 和定理 3 快速计算 $(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}$；

2）快速计算矩阵 $\boldsymbol{M}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{K}_k$，$\boldsymbol{M}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{M}_k$，$\boldsymbol{K}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{K}_k$ 的乘积；

3）履行算法 1 的第 2 步迭代格式。

在上述算法中，由定理 3 知计算向量 $\boldsymbol{x}_2$ 的复杂度约为 $13n^2/2$；再由定理 2 的递推公式知，计算逆矩

阵 $(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}$ 总的复杂度约为 $7n^2$；而在第 2 步中计算 $\boldsymbol{M}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{K}_k$、$\boldsymbol{M}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{M}_k$ 和 $\boldsymbol{K}_k(\boldsymbol{S}_k-\boldsymbol{H}_k)^{-1}\boldsymbol{K}_k$ 的乘积的复杂度约为 $25n^2/2$，从而整个算法的计算复杂度大约为 $52n^2/2$，即 $O(n^2)$ 计算复杂度。

## 4　数值实验

本章通过数值实验来说明快速保结构加倍算法 2 的有效性。采用 Matlab 编程，将算法 2 与原来的保结构加倍算法 1，在计算时间和计算精度两方面进行比较。程序运行的 PC 环境为具有 4 GB 内存和 Intel 2.3 GHz CPU 的 Windows 系统，误差精度为 $eps=2^{-53}\approx 1.1\times 10^{-16}$。两种算法的终止准则为

$$\|\boldsymbol{S}_{k+1}-\boldsymbol{S}_k\|_1/\|\boldsymbol{S}_k\|_1 \leqslant n\times eps, \tag{8}$$

式中：$\|\bullet\|_1$ 表示矩阵的 1 范数；

$n$ 为矩阵的阶数。

当式（8）满足时，取 $\boldsymbol{S}_{k+1}$ 作为方程（2）的近似解，并计算此时方程的相对残量

$$r=\|Q(\boldsymbol{S}_{k+1})\|_1/\left(\|\boldsymbol{M}\|_1\|\boldsymbol{S}_{k+1}\|_1^2+\|\boldsymbol{D}\|_1\|\boldsymbol{S}_{k+1}\|_1+\|\boldsymbol{K}\|_1\right)。$$

**例 1**　考虑阻尼系统[11]对应的二次方程（2），其对应的系数矩阵

$$\boldsymbol{M}=\boldsymbol{I}_n,$$
$$\boldsymbol{D}=\beta\bullet\text{tri diag}(-10, 30, -10),$$
$$\boldsymbol{K}=\text{tri diag}(-5, 15, -5),$$

其中 tri diag$(a, b, c)$ 表示三对角矩阵，主对角元为 $b$，上次对角为 $c$，下次对角为 $a$；$\beta$ 为确定系统是否处于过阻尼状态的正数。试验证算法 1 和算法 2 的有效性。

**解**　取 $\beta=1$（完全过阻尼状态）和 $\beta=0.448$（接近弱过阻尼状态）两种情况，检测算法 2 的数值表现。计算的方程阶数 $n$ 分别取值 500, 1 000, 1 500, 2 000。当算法终止时，设算法的迭代次数为 $x$，方程的相对残量为 $r$，CPU 消耗的计算时间为 $t$，其结果分别如表 1 和表 2 所示。

**表 1　例 1 中 $\beta$=1 时两种算法的结果比较**

Table 1　Comparison of two algorithm results in example 1 when $\beta$=1

| $n$ | 算法 1 | | | 算法 2 | | |
|---|---|---|---|---|---|---|
| | $x$ | $t$/s | $r$ | $x$ | $t$/s | $r$ |
| 500 | 5 | 1.7 | 8.83e−17 | 5 | 1.1 | 2.64e−15 |
| 1 000 | 5 | 19.9 | 5.57e−17 | 5 | 4.6 | 3.27e−15 |
| 1 500 | 5 | 56.1 | 4.40e−17 | 5 | 16.3 | 3.02e−15 |
| 2 000 | 5 | 95.9 | 3.35e−17 | 5 | 36.7 | 2.53e−15 |

从表 1 和表 2 可以看出，保结构加倍算法 1 和本文给出的算法 2，在系统处于完全过阻尼状态或接近弱过阻尼状态时，都能求得方程（2）的解。但在两种状态下，本文给出的算法 2 与原保结构加倍算法 1 相比，能以更少的时间计算出方程的解，获得的相对残量却与原保结构算法 1 相差不大。

**表 2　例 1 中 $\beta$=0.448 时两种算法的结果比较**

Table 2　Comparison of two algorithm results in example 1 when $\beta$=0.448

| $n$ | 算法 1 | | | 算法 2 | | |
|---|---|---|---|---|---|---|
| | $x$ | $t$/s | $r$ | $x$ | $t$/s | $r$ |
| 500 | 9 | 2.9 | 3.58e−17 | 9 | 1.9 | 9.63e−15 |
| 1 000 | 9 | 31.3 | 6.12e−17 | 9 | 11.6 | 1.47e−14 |
| 1 500 | 9 | 97.1 | 8.67e−16 | 9 | 23.7 | 1.78e−14 |
| 2 000 | 9 | 135.0 | 7.64.e−16 | 9 | 40.6 | 2.03e−14 |

**例 2**　考虑二次方程 $Q(\boldsymbol{S})=\boldsymbol{MS}^2+\boldsymbol{DS}+\boldsymbol{K}=\boldsymbol{0}$，系数矩阵 $\boldsymbol{M}$ 和 $\boldsymbol{K}$ 都具有式（4）中矩阵 $\boldsymbol{R}$ 的形式，其元素 $r_i$ 为区间 $(-1, 0)$ 中随机分布的数 $(1\leqslant i\leqslant n)$，$r_0=2n$。矩阵 $\boldsymbol{D}=\mu\boldsymbol{M}+\mu^{-1}\boldsymbol{K}+10^{-3}\boldsymbol{I}_n$，其中 $\mu$ 为正实数。试验证算法 1 和算法 2 的有效性。

**解**　由定义 1 知，此时方程处于过阻尼状态。取 $\mu=1$ 和不同的方程阶数 $n(500, 1\ 000, 1\ 500, 2\ 000)$ 来检测保结构加倍算法 1 和算法 2 的数值表现。计算结果如表 3 所示。

**表 3　例 2 中两种算法的结果比较**

Table 3　Comparison of the two algorithm results in example 2

| $n$ | 算法 1 | | | 算法 2 | | |
|---|---|---|---|---|---|---|
| | $x$ | $t$/s | $r$ | $x$ | $t$/s | $r$ |
| 500 | 4 | 0.4 | 2.67e−16 | 4 | 0.9 | 4.72e−15 |
| 1 000 | 4 | 2.9 | 2.00e−16 | 4 | 1.8 | 1.49e−15 |
| 1 500 | 4 | 8.3 | 1.32e−16 | 4 | 3.6 | 1.39e−15 |
| 2 000 | 4 | 13.5 | 1.47e−16 | 4 | 5.8 | 2.09e−15 |

从表 3 可以看出，保结构加倍算法 1 和本文给出的算法 2，在系统处于过阻尼状态下能够求得方程的解。在 $n$ 较大且方程相对残量差不多的情况下，算法 2 比原来的保结构加倍算法 1 求解方程所消耗的时间更少。

**参考文献：**

[1]　TISSEUR F. Backward Error and Condition of Polynomial Eigenvalue Problems[J]. Linear Algebra and Its Applications，2000，309(1/2/3)，339−361.

[2]　TISSEUR F，MEERBERGEN K. The Quadratic Eigenvalue Problems[J]. SIAM Review，2001，43(2)，235−286.

[3]　GOLUB G H，LOAN C F V. Matrix Computations[M]. 3rd ed. Baltimore：Johns Hopkins University Press，1996：375−387.
（下转第 15 页）