

基于复西尔维斯特矩阵方程的 改进双阶尺度分裂数值计算方法

杨创勋

广东工业大学数学与统计学院, 广东 广州

收稿日期: 2024年3月14日; 录用日期: 2024年5月17日; 发布日期: 2024年5月27日

摘要

连续时间的矩阵方程是矩阵方程中十分重要的一个类型, 在矩形域上的椭圆边值问题的数值解法、线性统计、振动结构的共振控制、二次矩阵的特征值配置问题、受噪声影响的图像复原等问题中有重要地位。由于连续时间的矩阵方程广泛的应用背景, 因此, 对连续时间的Sylvester方程的数值解法的研究具有重要的理论和实际意义。改进的双步尺度分裂(MDSS)方法是研究一类大型复杂对称线性系统的一种有效方法。在本文中, 我们将在双步尺度分裂(DSS)方法的基础上通过改进, 得到MDSS方法, 来求解复数域上的连续时间的西尔维斯特矩阵方程的近似解, 证明了该迭代序列在任意初始条件的情况下都收敛于西尔维斯特矩阵方程的唯一解, 并确定了其最优参数和相应的最优收敛因子。最后, 给出了一个测试问题来说明该新技术的有效性。

关键词

西尔维斯特矩阵方程, MDSS迭代法, 最优收敛因子

Improved Numerical Calculation Method for Second-Order Scale Splitting Based on the Complex Sylvester Matrix Equation

Chuangxun Yang

School of Mathematics and Statistics, Guangdong University of Technology,
Guangzhou Guangdong

Received: Mar. 14th, 2024; accepted: May 17th, 2024; published: May 27th, 2024

Abstract

The continuous time matrix equation is a very important type of matrix equation, which plays an important role in numerical solutions of elliptic boundary value problems in rectangular domains, linear statistics, resonance control of vibration structures, eigenvalue assignment of quadratic matrices, image restoration affected by noise, and other problems. Due to the wide application background of continuous time matrix equations, the study of numerical solutions for the continuous time Sylvester equation has important theoretical and practical significance. The improved two-step scale splitting (MDSS) method is an effective method for studying a class of large complex symmetric linear systems. In this article, we will improve the two-step scale splitting (DSS) method and obtain the MDSS method to solve the approximate solution of the continuous time Sylvester matrix equation in the complex field. We prove that the iterative sequence converges to a unique solution of the Sylvester matrix equation under any initial condition, and determine its optimal parameters and corresponding optimal convergence factors. Finally, a testing question was presented to demonstrate the effectiveness of the new technology.

Keywords

Sylvester Matrix Equation, MDSS Iterative Method, Optimal Parameters

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在这里，我们重点关注西尔维斯特矩阵方程：

$$AX + XB = C, A \in \mathbb{C}^{m \times m}, B \in \mathbb{C}^{n \times n}, C \in \mathbb{C}^{m \times n} \quad (1)$$

其中必须确定矩阵 $X \in \mathbb{C}^{m \times n}$ 。式(1)在控制理论和许多其他工程分支中具有非常重要的应用。矩阵方程在科学计算和工程技术等诸多领域都有着广泛的应用，如控制论[1]、系统理论、信号处理[2]、图像恢复[3]、极点配置[4]、扰动分析、模型降阶[5]、神经网络[6]、矩阵逼近问题[7]、电力系统[8]、微分方程的数值解[9]等。因此，对连续时间的 Sylvester 方程数值解法的研究具有重要的理论和实际意义。对于 Sylvester 矩阵方程，传统的求解方法是把这个矩阵方程用 Kronecker 积转化为形如的等价线性方程组，然后再用求解线性方程组的方法去计算其数值解。然而在实际问题中，遇到的系数矩阵与规模一般都是较大的，用 Kronecker 积转化后，会使阶数也变得非常大，在很大程度上增加了工作量及计算机的储存空间，而要得到其解析式也十分困难，此时这种传统的解决方法变得不可行。近年来，国内外许多学者致力于研究线性方程组的各种求解方法，并推广到矩阵方程的求解，这些方法可以概括为直接法和迭代法两大类。直接法是一种精确的求解方法，常见的直接法有 Bartels-stewart 法[10]和 Hessenberg-Schur 法[11]，其原理是利用正交相似变换将矩阵转化为三角矩阵或者上 Hessenberg 矩阵，然后再利用回代法求解线性方程。当求解一些小型线性矩阵方程时，我们通常采用直接法。但对于大型稀疏 Sylvester 矩阵方程，由于系数矩阵的规模非常大，直接法在求解方程时可能会产生许多非零元，将消耗大量的存储空间和计算时间，所以对于大型稀疏矩阵，我们一般采用迭代法来求解。古典迭代法是基于矩阵分裂的迭代算法，其基本思想是将线性方程组问题转化为求解一个不动点方程的问题，进而构造迭代格式。因此，分裂法是常见

的有效迭代法, 包括 Jacobi 迭代法、Gauss-Seidel 迭代法、SOR 迭代法、交替方向法[12]、HSS 迭代法[13] 和 Smith 方法[14]等。在本文中, 我们将在双步尺度分裂(DSS)方法的基础上通过改进, 得到 MDSS 方法, 来求解复数域上的连续时间的西尔维斯特矩阵方程的近似解, 证明了该迭代序列在任意初始条件的情况下都收敛于西尔维斯特矩阵方程的唯一解, 并确定了其最优参数和相应的最优收敛因子。最后, 给出了一个测试问题来说明该新技术的有效性。

2. 复西尔维斯特矩阵方程的 MDSS 算法的研究

在本节中, 先给出本文中的一些基本定义和符号。

符号 \mathbb{C} 是所有复数的集合, 维数为 n 的复向量的集合用 \mathbb{C}^n 表示。符号 \mathbb{R} 是所有实数的集合, 并用 \mathbb{R}^n 表示维数为 n 的实向量的集合。具有 m 行和 n 列的复数(或实数)矩阵的集合用 $\mathbb{C}^{m \times n}$ (或 $\mathbb{R}^{m \times n}$) 表示。用 \otimes 表示克罗内克积, 用 I 表示单位矩阵, 用 0 表示零矩阵为 0 。此外, A^T 是 A 的转置, 以及我们得到 $A^H = \bar{A}^T$, 其中 \bar{A} 表示矩阵 A 的共轭。

对于向量 x , 我们定义 $\|x\|_2$ 为向量 x 的 2 范数, 以及我们定义 $\|\cdot\|_F$ 向量 x 的 Frobenius 范数, 其中对于任意矩阵 A , 我们定义 $\|A\|_F = \sqrt{\text{tr}(AA^H)}$ 。在这期间, 我们定义 $\text{tr}(A)$ 为矩阵 A 的迹。同样的, 我们定义 $Sp(A)$ 为矩阵 A 的谱, 则矩阵 A 的谱半径定义为 $\rho(A)$ 。我们使得矩阵 $A = (a_1, a_2, \dots, a_n) \in \mathbb{C}^{m \times n}$, 其中 a_k 是矩阵 A 的第 k 列。接着, 我们可以定义 $\text{vec}(A) = (a_1, a_2, \dots, a_n)^T$, 其中 $\text{vec}(A)$ 是 $\mathbb{C}^{m \times n}$ 中的向量。

在研究本文提出的新算法前, 我们将列出其他求解复西尔维斯特矩阵方程的算法, 我们计划简要描述一下求解矩阵方程(1)的 MHSS、PMHSS 和 GMHSS 迭代方法[15] [16]。

2.1. 解决矩阵方程(1)的 MHSS 算法

通过计算求解 $X^{(k+1)} \in \mathbb{C}^{m \times n}$, 再使用下列式子, 使得 $\{X^{(k)}\}_{k=0}^{\infty}$ 收敛。

$$\begin{cases} (\alpha I + W)X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\beta I + U) = (\alpha I - iT)X^{(k)} + X^{(k)}(\beta I - iV) + C \\ (\alpha I + T)X^{(k+1)} + X^{(k+1)}(\beta I + V) = (\alpha I + iW)X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\beta I + iU) - iC \end{cases} \quad (2)$$

其中, $\alpha > 0$ 和 $\beta > 0$, $X^{(0)} \in \mathbb{C}^{m \times n}$ 是一个初始矩阵, I 是单位矩阵。因此我们得到矩阵

$$M(\gamma) = (\gamma I + D)^{-1}(\gamma I + iH)(\gamma I + H)^{-1}(\gamma I - iD)$$

为 MHSS 方法的迭代矩阵, 其中为 $H = I \otimes W + U \otimes I$, $D = I \otimes T + V \otimes I$ 和 $\gamma = \alpha + \beta$ 。此外, 还证明了 $\rho(M(\gamma)) \leq \sigma(\gamma) < 1$, 其中

$$\sigma(\gamma) := \max \frac{\sqrt{\gamma^2 + (\lambda_i^W + \lambda_j^U)^2}}{\gamma + \lambda_i^W + \lambda_j^U}$$

因此, MHSS 迭代收敛于西尔维斯特矩阵等式(1)的精确解 $X^* \in \mathbb{C}^{m \times n}$, 其中得到最优参数 $\tilde{\gamma} = \sqrt{\lambda_{\min}^W \lambda_{\max}^U}$, 其中 $\lambda_{\min}^W = \lambda_{\min}^W + \lambda_{\min}^U$, $\lambda_{\max}^W = \lambda_{\max}^W + \lambda_{\max}^U$, 同时 $\lambda_{\min}^W = \min Sp(W)$, $\lambda_{\min}^U = \min Sp(U)$, $\lambda_{\max}^U = \max Sp(U)$ 以及 $\lambda_{\max}^W = \max Sp(W)$ 。此外还有

$$\sigma(\tilde{\gamma}) = \frac{\sqrt{\kappa(H) + 1}}{\sqrt{\kappa(H) + 1}}$$

其中

$$\kappa(H) = \frac{\lambda_{\min}}{\lambda_{\max}}$$

2.2. 解决矩阵方程(1)的 PMHSS 算法

通过计算求解 $X^{(k+1)} \in \mathbb{C}^{m \times n}$ ，再使用下列式子，使得 $\{X^{(k)}\}_{k=0}^{\infty}$ 收敛。

$$\begin{cases} (\gamma P_1 + W)X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\gamma P_2 + U) = (\gamma P_1 - iT)X^{(k)} + X^{(k)}(\gamma P_2 - iV) + C \\ (\gamma P_1 + T)X^{(k+1)} + X^{(k+1)}(\gamma P_2 + V) = (\gamma P_1 + iW)X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\gamma P_2 + iU) - iC \end{cases} \quad (3)$$

其中 $\gamma > 0$ 和 $X^{(0)} \in \mathbb{C}^{m \times n}$ 是一个初始矩阵， P_1 和 P_2 是对称正定矩阵。而且

$$P(\gamma) = (\gamma P + D)^{-1}(\gamma P + iH)(\gamma P + H)^{-1}(\gamma P - iD)$$

为 PMHSS 方法的迭代矩阵，其中 $P = I \otimes P_1 + P_2^T \otimes I$ 。

2.3. 解决矩阵方程(1)的 GMHSS 算法

通过计算求解 $X^{(k+1)} \in \mathbb{C}^{m \times n}$ ，再使用下列式子，使得 $\{X^{(k)}\}_{k=0}^{\infty}$ 收敛。

$$\begin{cases} (\alpha I + W)X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\beta I + U) = (\alpha I - iT)X^{(k)} + X^{(k)}(\beta I - iV) + C \\ (\tau I + T)X^{(k+1)} + X^{(k+1)}(\mu I + V) = (\tau I + iW)X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\mu I + iU) - iC \end{cases} \quad (4)$$

其中， α 、 β 、 τ 和 μ 为实正常数。GMHSS 方法的迭代矩阵为

$$M(\gamma, \theta) = (\theta I + D)^{-1}(\theta I + iH)(\gamma I + H)^{-1}(\gamma I - iD)$$

因此，对于不同的参数 α 、 β 、 τ 和 μ ，提出了不同的方法。

接着，便是本文提出算法的研究：

2.4. 解决矩阵方程(1)的 MDSS 算法

在本节中，我们尝试应用 MDSS 迭代方法来求解矩阵方程(1)。首先，设 $A = W + iT$ 和 $B = U + iV$ ，其中 W 、 T 、 U 和 V 是实对称正定矩阵，根据上述等式以及变换，我们可以将矩阵方程(1)写成

$$(W + iT)X + X(U + iV) = C. \quad (5)$$

现在假设 α 和 β 是一个正的实常数。通过在等式(6)中分别乘以式子 $(\alpha - \beta i)$ 和 $(\beta - \alpha i)$ ，我们得到

$$(\alpha W + i\alpha T - i\beta W + \beta T)X + X(\alpha U + i\alpha V - i\beta U + \beta V) = (\alpha - i\beta)C \quad (6)$$

以及式子

$$(\beta W + i\beta T - i\alpha W + \alpha T)X + X(\beta U + i\beta V - i\alpha U + \alpha V) = (\beta - i\alpha)C \quad (7)$$

在这里，我们使得 $C_1 = (\alpha - i\beta)C$ ， $C_2 = (\beta - i\alpha)C$ ，然后我们得到

$$(\alpha W + \beta T)X + X(\alpha U + \beta V) = i(\beta W - \alpha T)X + iX(\beta U - \alpha V) + C_1 \quad (8)$$

以及

$$(\alpha T + \beta W)X + X(\alpha V + \beta U) = i(\alpha W - \beta T)X + iX(\alpha U - \beta V) + C_2 \quad (9)$$

然后通过上述式子，我们得到了求解矩阵方程(1)的算法。

通过计算求解 $X^{(k+1)} \in \mathbb{C}^{m \times n}$ ，再使用下列式子，使得 $\{X^{(k)}\}_{k=0}^{\infty}$ 收敛：

$$\begin{cases} (\alpha W + \beta T)X^{(k+\frac{1}{2})} + X^{(k+\frac{1}{2})}(\alpha U + \beta V) = i(\beta W - \alpha T)X^{(k)} + iX^{(k)}(\beta U - \alpha V) + C_1 \\ (\alpha T + \beta W)X^{(k+1)} + X^{(k+1)}(\alpha V + \beta U) = i(\alpha W - \beta T)X^{(k+\frac{1}{2})} + iX^{(k+\frac{1}{2})}(\alpha U - \beta V) + C_2 \end{cases} \quad (10)$$

其中, $\alpha > 0$ 和 $\beta > 0$, $X^{(0)} \in \mathbb{C}^{m \times n}$ 是一个初始矩阵, I 是单位矩阵。很明显, 矩阵 $\alpha W + \beta T$, $\alpha U + \beta V$, $\alpha T + \beta W$ 和 $\alpha V + \beta U$ 是对称正定的。

对此, 我们可以得到以下定理。

定理 2.1 首先, 我们定义 $E = D + iH$, 得到

$$D = I \otimes W + U \otimes I, H = I \otimes T + V \otimes I \quad (11)$$

其中, I 是单位矩阵。根据上述式子, 便得到了 MDSS 方法的迭代矩阵, 如下式所示:

$$K(\alpha, \beta) = (\alpha H + \beta D)^{-1}(\alpha D - \beta H)(\alpha D + \beta H)^{-1}(\beta D - \alpha H) \quad (12)$$

对此, 我们得到迭代矩阵 $K(\alpha, \beta)$ 的谱半径为

$$\rho(K(\alpha, \beta)) = \max \left| \frac{(\alpha\lambda - \beta)(\alpha - \beta\lambda)}{(\alpha\lambda + \beta)(\alpha + \beta\lambda)} \right| < 1, \forall \alpha, \beta > 0 \quad (13)$$

其中 $Sp(DH^{-1})$ 表示矩阵 DH^{-1} 的谱, 则 MDSS 迭代(11)在任意初始条件下都收敛于矩阵方程(1)的唯一精确解 $X^* \in \mathbb{C}^{m \times n}$ 。

证明: 通过使用 Kronecker 积, 我们可以按照以下的计算结构编写算法(3.1):

$$\begin{cases} (I \otimes (\alpha W + \beta T) + (\alpha U + \beta V)^T \otimes I)X^{(k+\frac{1}{2})} = (I \otimes i(\beta W - \alpha T) + i(\beta U - \alpha V) \otimes I)X^{(k)} + C_1 \\ (I \otimes (\alpha T + \beta W) + (\alpha V + \beta U) \otimes I)X^{(k+1)} = (I \otimes i(\alpha W - \beta T) + i(\alpha U - \beta V) \otimes I)X^{(k+\frac{1}{2})} + C_2 \end{cases} \quad (14)$$

接着, 根据算法演练, 上述两个等式, 可以分别重写为

$$(\alpha D + \beta H)x^{(k+\frac{1}{2})} = i(\beta D - \alpha H)x^{(k+1)} + C_1 \quad (15)$$

$$(\alpha H + \beta D)x^{(k+1)} = i(\alpha D - \beta H)x^{(k+\frac{1}{2})} + C_2 \quad (16)$$

其中, 我们设定 $C_1 = \text{vec}(C_1)$ 、 $C_2 = \text{vec}(C_2)$ 、 $x^{(k)} = \text{vec}(x^{(k)})$ 。根据上述等式转换, 我们得到

$$Q_1 = (\alpha D + \beta H)^{-1}(\alpha H - \beta D) \quad (17)$$

以及

$$Q_2 = (\alpha H + \beta D)^{-1}(\alpha D - \beta H) \quad (18)$$

因此, 算法 MDSS 过程的迭代矩阵为

$$K(\alpha, \beta) = Q_2 Q_1 = (\alpha H + \beta D)^{-1}(\alpha D - \beta H)(\alpha D + \beta H)^{-1}(\alpha H - \beta D)$$

接着, 我们假设 $\lambda_{p,q}^H$, λ_p^T 和 λ_q^V 分别表示 H , T 和 V 的特征值 ($p = 1, \dots, m$, $q = 1, \dots, n$)。可以得到

$$\lambda_{p,q}^H = \lambda_p^T + \lambda_q^V > 0, p = 1, \dots, m; q = 1, \dots, n$$

其中 H 是非奇异的, 因此, 对于算法 MDSS 的迭代矩阵 $K(\alpha, \beta)$ 的谱半径 $\rho(K(\alpha, \beta))$, 我们可以写出来, 得到以下结果

$$\begin{aligned}
 \rho(K(\alpha, \beta)) &= \rho\left((\alpha H + \beta D)^{-1}(\alpha D - \beta H)(\alpha D + \beta H)^{-1}(\alpha H - \beta D)\right) \\
 &= \rho\left(H^{-1}(\alpha I + \beta DH^{-1})^{-1}(\alpha DH^{-1} - \beta I)HH^{-1}(\alpha DH^{-1} + \beta I)^{-1}(\alpha I - \beta DH^{-1})H\right) \\
 &= \rho\left((\alpha I + \beta DH^{-1})^{-1}(\alpha DH^{-1} - \beta I)(\alpha DH^{-1} + \beta I)^{-1}(\alpha I - \beta DH^{-1})\right) \\
 &= \max_{\lambda \in Sp(DH^{-1})} \left| \frac{\alpha - \beta\lambda}{\alpha + \beta\lambda} \times \frac{\alpha\lambda - \beta}{\alpha\lambda + \beta} \right| \\
 &= \max_{\lambda \in Sp(DH^{-1})} \left| \frac{\alpha^2\lambda + \beta^2\lambda - \alpha\beta - \alpha\beta\lambda^2}{\alpha^2\lambda + \beta^2\lambda + \alpha\beta + \alpha\beta\lambda^2} \right| \\
 &= \max_{\lambda \in Sp(DH^{-1})} \left| \frac{\alpha\lambda \left[\left(\alpha + \frac{\beta^2}{\alpha} \right) - \beta \left(\lambda + \frac{1}{\lambda} \right) \right]}{\alpha\lambda \left[\left(\alpha + \frac{\beta^2}{\alpha} \right) + \beta \left(\lambda + \frac{1}{\lambda} \right) \right]} \right| \tag{19} \\
 &= \max_{\lambda \in Sp(DH^{-1})} \left| \frac{\left(\alpha + \frac{\beta}{\alpha} \right) - \beta \left(\lambda + \frac{1}{\lambda} \right)}{\left(\alpha + \frac{\beta}{\alpha} \right) + \beta \left(\lambda + \frac{1}{\lambda} \right)} \right|
 \end{aligned}$$

根据上述一系列等式，其中等式中任何一个 $\alpha, \beta, \lambda > 0$ ，因此我们可以得到下列不等式

$$\left(\alpha + \frac{\beta}{\alpha} \right) - \beta \left(\lambda + \frac{1}{\lambda} \right) < \left(\alpha + \frac{\beta}{\alpha} \right) + \beta \left(\lambda + \frac{1}{\lambda} \right) \tag{20}$$

因此，我们得到，迭代矩阵 $K(\alpha, \beta)$ 的谱半径 $\rho(K(\alpha, \beta))$ ，有以下关系式：

$$\rho(K(\alpha, \beta)) = \max_{\lambda \in Sp(DH^{-1})} \left| \frac{(\alpha\lambda - \beta)(\alpha - \beta\lambda)}{(\alpha\lambda + \beta)(\alpha + \beta\lambda)} \right| < 1, \forall \alpha, \beta > 0 \tag{21}$$

根据上述不等式(22)，表明了 MDSS 迭代(11)在任意初始条件下都收敛于矩阵方程(1)的唯一精确解 $X^* \in \mathbb{C}^{m \times n}$ 。定理得证。

根据上述定理，我们得到该算法可以在任意初始条件下都收敛于唯一的精确解。对此，我们还需得到参数，使得算法的谱半径 $\rho(K(\alpha, \beta))$ 取得最小值。因此，我们需要得到定理，并证明定理的可行性。

定理 2.2 此外，使函数 $\rho(K(\alpha, \beta))$ 最小化的参数 α, β 为

$$\frac{\alpha^*}{\beta^*} = \frac{\sqrt{uv} + \sqrt{uv} - 4}{2} \tag{22}$$

其中

$$\begin{aligned}
 u &= \min_{\lambda_{\min} \leq \zeta \leq \lambda_{\max}} \{ \zeta + \zeta^{-1} \} \\
 v &= \max_{\lambda_{\min} \leq \zeta \leq \lambda_{\max}} \{ \zeta + \zeta^{-1} \}
 \end{aligned} \tag{23}$$

并设为

$$\lambda_{\min} = \min_{\lambda_j \in Sp(DH^{-1})} \{ \lambda_j \}, \lambda_{\max} = \max_{\lambda_j \in Sp(DH^{-1})} \{ \lambda_j \} \tag{24}$$

接着，我们得到收敛因子 $\rho(K(\alpha, \beta))$

$$\rho(K(\alpha, \beta)) = \frac{\sqrt{\kappa-1}}{\sqrt{\kappa+1}} \quad (25)$$

其中,

$$\kappa = \frac{u}{v} \quad (26)$$

证明: 从上述所得到的等式与结果上可以看出

$$\rho(K(\alpha, \beta)) = \max \left\{ \frac{\alpha + \beta^2 \alpha^{-1} - \beta u}{\alpha + \beta^2 \alpha^{-1} + \beta u}, \frac{\alpha + \beta^2 \alpha^{-1} - \beta v}{\alpha + \beta^2 \alpha^{-1} + \beta v} \right\} \quad (27)$$

接着, 我们可以根据下列等式得到其最小值

$$\frac{\alpha^* + \beta^{*2} \alpha^{*-1} - \beta^* u}{\alpha^* + \beta^{*2} \alpha^{*-1} + \beta^* u} = -\frac{\alpha^* + \beta^{*2} \alpha^{*-1} - \beta^* v}{\alpha^* + \beta^{*2} \alpha^{*-1} + \beta^* v} \quad (28)$$

当上述等式成立时, 我们便可以得到谱半径 $\rho(K(\alpha, \beta))$ 的最小值。根据上述关系, 我们得到

$$(\alpha^* + \beta^{*2} \alpha^{*-1})^2 = \beta^{*2} uv \quad (29)$$

接着得到下列等式

$$\alpha^{*2} - \alpha^* \beta^* \sqrt{uv} + \beta^{*2} = 0 \quad (30)$$

因此, 通过求解上述等式, 对于谱半径 $\rho(K(\alpha, \beta))$, 我们得到

$$\rho(K(\alpha^*, \beta^*)) = \frac{\alpha^* + \beta^{*2} \alpha^{*-1} - \beta^* v}{\alpha^* + \beta^{*2} \alpha^{*-1} + \beta^* v} \quad (31)$$

根据上述结果以及等式, 我们最终可以得到最小化的谱半径的值

$$\rho(K(\alpha^*, \beta^*)) = \frac{\sqrt{uv} - u}{\sqrt{uv} + u} = \frac{\sqrt{\frac{v}{u}} - 1}{\sqrt{\frac{v}{u}} + 1} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \quad (32)$$

定理得证。

3. 数值例子

为了测试文中提出的新方法的有效性, 我们给出了一个求解西尔维斯特矩阵等式的测试问题(1.1)的方法。所有的计算和绘图都是在 MATLAB 软件上完成的。我们使用 $X^{(0)} = 0$ (零矩阵)来进行初始迭代, 如果当前的迭代满足以下条件, 则终止迭代

$$\frac{\|R(X^{(K)})\|_F}{\|R(X^{(0)})\|_F} \leq 10^{-10}$$

其中 $R(X^{(K)}) = C - AX^{(K)} - X^{(K)}B$ 。在测试过程中, 我们会报告迭代次数(IT)和计算时间(以秒为单位)。对于西尔维斯特矩阵方程 $AX + XB = C$, 与 $A = W + iT$, $B = U + iV$, 其中, 我们假设

$$W = U = \left(K + \frac{3 - \sqrt{3}}{\theta} I_n \right), T = V = \left(K + \frac{3 + \sqrt{3}}{\theta} I_n \right)$$

其中, 矩阵 K 的形式为 $K = I_n \otimes V_m + V_m \otimes I_m$, 其中 $V_m = h^{-2} \text{Tri}(-1, 2, -1) \in \mathbb{R}^{m \times m}$ 。因此, K 是一个大小为 $n \times n$ 的对角矩阵, 具有 $n = m^2$ 。 I_n 和 I_m 分别是维数为 n 和 m 的单位矩阵。在这里, 我们设置了

$$\theta = h = \frac{1}{m+1}。$$

此外, 对于矩阵 C , 我们设置 $C = K$ 。同时, 我们通过把矩阵方程(1)乘以 h^2 来将它归一化。最后, 我们用 α^* 和 β^* 表示了 MDSS 方法中的最优参数。这些参数通过实验被指定, 使每种方法的迭代次数最少。

在数值实验中, 我们设置 $m = 2$ 。通过输入上述参数, 在 MATLAB 中得到如下图形:

从图 1 可以看出, 本文提出的 MDSS 算法在任意的初始条件下, 可以收敛于唯一的数值解, 说明算法在实际应用中的可行性也可以得到验证。

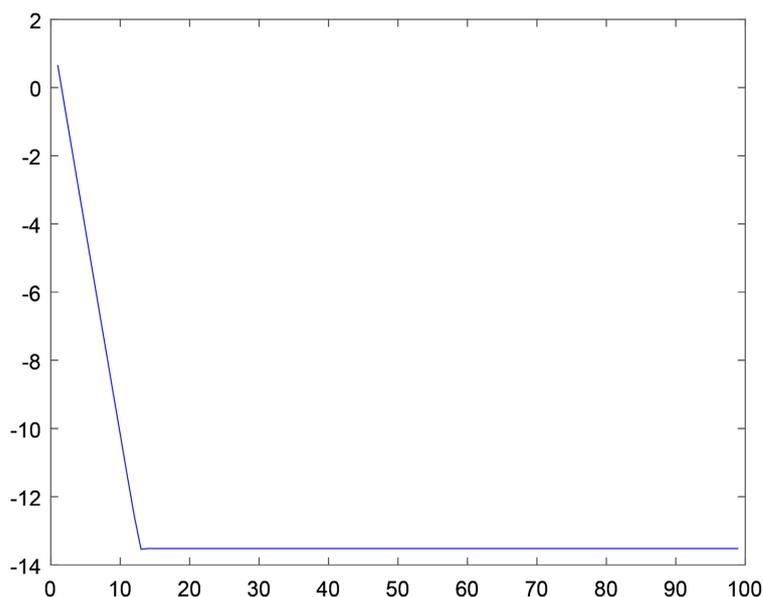


Figure 1. The number of iterations of MDSS algorithm under logarithmic conditions

图 1. MDSS 算法在对数条件下的迭代次数

4. 结论

在本文中, 我们在双步尺度分裂(DSS)方法的基础上通过改进, 得到 MDSS 方法, 来求解复数域上的连续时间的西尔维斯特矩阵方程的近似解, 证明了该迭代序列在任意初始条件的情况下都收敛于西尔维斯特矩阵方程的唯一解, 并确定了其最优参数和相应的最优收敛因子。最后, 通过数值实验可以证明该算法的可行性。

参考文献

- [1] Zhou, K., Doyle, J.C. and Glover, K. (1996) Robust and Optimal Control. Prentice-Hall, Upper Saddle River, New Jersey.
- [2] Anderson, B.D.O., Agathoklis, P., Jury, E.I. and Mansour, M. (1986) Stability and the Matrix Lyapunov Equation for Discrete 2-Dimensional Systems. *IEEE Transactions on Circuits and Systems*, **33**, 261-267. <https://doi.org/10.1109/TCS.1986.1085912>
- [3] Calvetti, D. and Reichel, L. (1996) Application of ADI Iterative Methods to the Restoration of Noisy Images. *SIAM*

-
- Journal on Matrix Analysis and Applications*, **17**, 165-186. <https://doi.org/10.1137/S0895479894273687>
- [4] Bru, R., Mas, J. and Urbano, A.M. (1994) An Algorithm for the Single-Input Pole Assignment Problem. *SIAM Journal on Matrix Analysis and Applications*, **15**, 393-407. <https://doi.org/10.1137/S0895479889161125>
 - [5] Baur, U. and Benner, P. (2008) Cross-Gramian Based Model Reduction for Data-Sparse Systems. *Electronic Transactions on Numerical Analysis*, **31**, 256-270. <https://doi.org/10.1137/070711578>
 - [6] Zhang, Y.N., Jiang, D.C. and Wang, J. (2002) A Recurrent Neural Network for Solving Sylvester Equation with Time-Varying Coefficients. *IEEE Transactions on Neural Networks*, **13**, 1053-1063. <https://doi.org/10.1109/TNN.2002.1031938>
 - [7] Liao, A.P., Bai, Z.Z. and Lei, Y. (2005) Best Approximate Solution of Matrix Equation $AXB + CYD = E$. *SIAM Journal on Matrix Analysis and Applications*, **27**, 675-688. <https://doi.org/10.1137/040615791>
 - [8] Ilic, M. (1989) New Approaches to Voltage Monitoring and Control. *IEEE Control Systems Magazine*, **9**, 5-11. <https://doi.org/10.1109/37.16743>
 - [9] Bai, Z.Z., Huang, Y.M. and Ng, M.K. (2007) On Preconditioned Iterative Methods for Burgers Equations. *SIAM Journal on Scientific Computing*, **29**, 415-439. <https://doi.org/10.1137/060649124>
 - [10] Bartels, R.H. and Stewart, G.W. (1972) Solution of the Matrix Equation $AX + XB = C$. *Communications of the ACM*, **15**, 820-826. <https://doi.org/10.1145/361573.361582>
 - [11] Golub, G.H., Nash, S. and Loan, C.V. (1979) A Hessenberg-Schur Method for the Problem $AX + XB = C$. *IEEE Transactions on Automatic Control*, **24**, 909-913. <https://doi.org/10.1109/TAC.1979.1102170>
 - [12] Benner, P., Li, R.C. and Truhar, N. (2009) On the ADI Method for Sylvester Equations. *Journal Computational and Applied Mathematics*, **233**, 1035-1045. <https://doi.org/10.1016/j.cam.2009.08.108>
 - [13] Bai, Z.-Z. (2011) On Hermitian and Skew-Hermitian Splitting Iteration Methods for Continuous Sylvester Equations. *Journal of Computation and Mathematics*, **29**, 185-198. <https://doi.org/10.4208/jcm.1009-m3152>
 - [14] Smith, R.A. (1968) Matrix Equation $XA + BX = C$. *SIAM Journal on Applied Mathematics*, **16**, 198-201. <https://doi.org/10.1137/0116017>
 - [15] Bai, Z.-Z., Benzi, M. and Chen, F. (2010) Modified HSS Iteration Methods for a Class of Complex Symmetric Linear Systems. *Computing*, **87**, 93-111. <https://doi.org/10.1007/s00607-010-0077-0>
 - [16] Hezari, D., Salkuyeh, D.K. and Edalatpour, V. (2016) A New Iterative Method for Solving a Class of Complex Symmetric System of Linear Equations. *Numerical Algorithms*, **73**, 927-955. <https://doi.org/10.1007/s11075-016-0123-x>