

计算矩阵 $M-P$ 广义逆的一种迭代法

陈 荣

长沙理工大学数学与统计学院, 湖南 长沙

收稿日期: 2023年6月14日; 录用日期: 2023年7月9日; 发布日期: 2023年7月19日

摘 要

本文提出了一种求解Moore-Penrose广义逆的高阶迭代法, 并证明了该方法具有局部九阶收敛速度。此外, 我们进行了数值实验, 与已有的迭代法进行计算时间对比, 结果表明该方法是有有效可行的。

关键词

Moore-Penrose广义逆, 迭代法, 局部收敛速度

An Iterative Method for Computing Matrix $M-P$ Generalized Inverse

Rong Chen

School of Mathematics and Statistics, Changsha University of Science and Technology, Changsha Hunan

Received: Jun. 14th, 2023; accepted: Jul. 9th, 2023; published: Jul. 19th, 2023

Abstract

In this paper, a high order iterative method for solving the Moore-Penrose generalized inverse is proposed, and it is proved that the method has local ninth order convergence rate. In addition, numerical experiments are carried out to compare the calculation time with the existing iterative methods, and the results show that the proposed method is effective and feasible.

Keywords

Moore-Penrose Generalized Inverse, Iterative Method, Local Convergence Rate



1. 引言

M - P 广义逆在求解线性方程组和线性最小二乘问题具有广泛应用, 例如关于求解以下线性方程组:

$$Ax = b, \quad (1.1)$$

其中 $A \in C^{m \times n}$ 是系数矩阵, b 为已知的 m 维列向量, x 为未知的 n 维列向量, 那么就可以通过计算系数矩阵 A 的 M - P 广义逆来求解。关于 M - P 广义逆的计算, 直接法例如奇异值分解法是较为常用的方法, 但当 A 的规模较大时, 直接法的计算成本很高, 因此, 计算 M - P 广义逆的高效迭代法十分重要。

1906 年, Moore [1] 注意到一个矩阵的广义逆的存在性, 并且用投影的方式定义每个矩阵的广义逆。1955 年, Penrose [2] 以不同的方式重新进行定义, 用同时满足四个方程的形式(简称 *Moore-Penrose* 方程), 引入唯一的广义逆。下面给出 M - P 广义逆的定义。

定义 设 A 是复数域上的 $m \times n$ 矩阵, 若存在一个矩阵 $X \in C^{m \times n}$, 使它满足以下 *Penrose* 方程:

$$\begin{cases} AXA = A, \\ XAX = X, \\ (AX)^* = AX, \\ (XA)^* = XA, \end{cases} \quad (1.2)$$

其中, $*$ 表示矩阵的共轭转置。那么可以证明 X 是存在且唯一的, X 被称为 A 的 M - P 广义逆, 记作 A^\dagger 。

对于许多的数学问题, 我们不能通过有限次算术运算及解析方法来求得它的精确解, 而只能通过对所求的解进行一次次估计, 从而逐步求得它的逼近解, 像这种方法我们称为迭代法。一直以来, 学者们对于使用迭代法计算广义逆做着许多研究。

学者们不断提出各种估计矩阵 M - P 逆的方法, 大多数这些迭代法都是基于求解以下定义的矩阵方程:

$$f(X) = I - AX = 0. \quad (1.3)$$

所求出来的矩阵 X 具有 A 的逆的部分性质, 验证该方法满足 M - P 逆的定义, 则此迭代法可以成为求广义逆的一种迭代法。

1933 年, Schulz [3] 提出迭代格式

$$X_{k+1} = X_k (2I - AX_k), k = 0, 1, 2, \dots, \quad (1.4)$$

其中 I 是 $m \times m$ 的单位矩阵, X_0 是矩阵 A^{-1} 的初始近似, 这就是著名的 *Schulz* 迭代法, 也是计算 M - P 逆常用的迭代法。该方案完全基于矩阵和矩阵相乘, 可以在并行机器上实现。而且如在文献[4]中所讨论的, 上述方法具有多对数复杂度, 并且在数值上是稳定的。

然而, 上述方案在开始过程中收敛比较缓慢, 导致计算工作量的增加。因此, 更高阶的方案被开发出来以降低计算复杂度。例如, W. Li 和 Z. Li [5] 提出以下三阶迭代方法:

$$X_{k+1} = X_k \left(3I - 3AX_k + (AX_k)^2 \right), k = 0, 1, 2, \dots \quad (1.5)$$

2011 年, Li 等人[6]将迭代格式(1.5)改写成以下迭代格式:

$$X_{k+1} = X_k \left(3I - AX_k (3I - AX_k) \right), \quad (1.6)$$

这个也是著名的切比雪夫法。Krishnamurthy 和 Sen [7]提出以下的四阶迭代格式:

$$X_{k+1} = X_k \left(I + Y_k \left(I + Y_k \left(I + Y_k \right) \right) \right), \quad (1.7)$$

其中 $Y_k = I - AX_k$ 。

Soleymani [8]中提出了以下五阶迭代格式, 每次迭代需要六次矩阵相乘:

$$X_{k+1} = -\frac{1}{2} X_k \left(-11I + AX_k \left(25I + AX_k \left(-30I + AX_k \left(20I + AX_k \left(-7I + AX_k \right) \right) \right) \right) \right). \quad (1.8)$$

Soleymani [9]等人引入以下六阶方法, 每次迭代使用五次矩阵和矩阵相乘:

$$\begin{cases} B_k = AX_k, \\ C_k = B_k(-I + B_k), \\ X_{k+1} = X_k(2I - B_k)(3I - 2B_k + C_k)(I + C_k). \end{cases} \quad (1.9)$$

2012年, Soleymani [10]又开发七阶迭代方案, 每次迭代需要八次矩阵相乘:

$$X_{k+1} = \frac{1}{16} X_k \left[120I + AX_k \left(-393I + AX_k \left(735I + AX_k \left(-861I + AX_k \left(651I + AX_k \left(-315I + AX_k \left(93I + AX_k \left(-15I + AX_k \right) \right) \right) \right) \right) \right) \right) \right]. \quad (1.10)$$

1986年, Krishnamurthy 和 Sen [7]提出这类 Schulz 型迭代方案的一般形式。2013年, Li [11]等人对这类迭代方案进行理论研究, 证明 Schulz 型矩阵迭代法在数值上是稳定的。通过这种方法, 提出每个基于超幂级数的 p 阶迭代方法, 每次迭代需要 p 次矩阵和矩阵乘法。例如, 需要九个矩阵和矩阵乘法的九阶迭代方案定义如下:

$$X_{k+1} = X_k \left[I + Y_k \left(I + Y_k \left(I + Y_k \left(I + Y_k \left(I + Y_k \left(I + Y_k \left(I + Y_k \left(I + Y_k \right) \right) \right) \right) \right) \right) \right) \right], \quad (1.11)$$

其中 $Y_k = I - AX_k$ 。

本文构造一种新的高阶迭代方法来逼近 M - P 逆。首先, 从理论上证明提出的方案对非奇异矩阵收敛于九阶。之后, 我们研究生成的迭代序列到 M - P 逆的收敛性。通过计算效率, 我们将该迭代方案与先前的迭代方案进行理论比较。最后进行数值实验, 对于要求广义逆的给定矩阵 A , 我们的迭代方案所需迭代次数与迭代时间与原有的迭代格式进行对比, 验证该方案是有效可行的。

2. 算法

在本节中, 我们构造一种新的求矩阵广义逆的迭代方法。为此, 我们考虑以下新的有理迭代函数:

$$\begin{cases} Y_k = X_k - \frac{F(X_k)}{2F'(X_k)} \left(2 + \frac{F''(X_k)F(X_k)}{F'(X_k)^2} \right), \\ Z_k = Y_k - \frac{F(Y_k)}{F'(Y_k)}, \\ X_{k+1} = Y_k - \left(\frac{F(Y_k) - \frac{1}{5}F(Z_k)}{F'(Z_k)} \right) \left(1 - \frac{4}{5} \frac{F(Z_k)}{F(Y_k)} \right), k=0,1,2,\dots, \end{cases} \quad (2.1)$$

注意到, (2.1)在求解非线性方程的零点时具有局部九阶收敛性。

现在, 为了计算给定矩阵 A 的逆, 将迭代方案(2.1)应用于矩阵函数(1.3), 得到以下迭代格式:

$$\begin{aligned} X_{k+1} &= -\frac{1}{25} X_k (3I + AX_k (-3I + AX_k)) (-79I + AX_k (3I + AX_k (-3I + AX_k))) \\ &\quad \times (87I + AX_k (3I + AX_k (-3I + AX_k)) (-37I + 4AX_k (3I + AX_k (-3I + AX_k)))) \\ &= \frac{1}{25} X_k \left[237I - 1020AX_k + 2644(AX_k)^2 - 4626(AX_k)^3 + 5814(AX_k)^4 \right. \\ &\quad - 5460(AX_k)^5 + 3924(AX_k)^6 - 2169(AX_k)^7 + 901(AX_k)^8 \\ &\quad \left. - 264(AX_k)^9 + 48(AX_k)^{10} - 4(AX_k)^{11} \right], k = 0, 1, 2, \dots \end{aligned}$$

在上述的等式中, 矩阵和矩阵相乘的次数是十二, 这增加了每次迭代的计算量。而通过适当的因子分解技术, 可以使计算成本最小化, 从而提高计算效率。为此, 将上式改写成以下矩阵迭代格式:

$$\begin{cases} B_k = AX_k, \\ C_k = 3I + B_k (-3I + B_k), \\ S_k = B_k C_k, \\ X_{k+1} = -\frac{1}{25} X_k C_k (-79I + S_k (87I + S_k (-37I + 4S_k))), k = 0, 1, 2, \dots \end{cases} \quad (2.2)$$

在上述简化形式(2.2)中, 每次迭代时的矩阵相乘减少到七次。此外, 该迭代格式(2.2)也属于 Schulz 型方法。在下一节中, 我们将陈述并证明该方案在某些假设下具有九阶收敛性。

3. 理论分析

本节针对提出的迭代格式(2.2), 分别讨论其对于非奇异复矩阵以及一般复矩阵的收敛性分析。

3.1. 收敛性分析

定理 3.1 假设 A 是一个 $n \times n$ 阶非奇异复矩阵。如果初始迭代矩阵 X_0 满足以下条件:

$$\|I - AX_0\| < 1, \quad (3.1)$$

那么迭代格式(2.2)九阶收敛于 A^\dagger 。

证明: 假设不等式(3.1)成立, 设 $E_k = I - AX_k$ 为每步迭代的残差矩阵, 那么 $E_0 = I - AX_0$ 为初始残差矩阵。我们可以得到第 $k+1$ 步残差矩阵为:

$$\begin{aligned} E_{k+1} &= I - AX_{k+1} \\ &= I - \frac{1}{25} \left[237AX_k - 1020(AX_k)^2 + 2644(AX_k)^3 - 4626(AX_k)^4 \right. \\ &\quad + 5814(AX_k)^5 - 5460(AX_k)^6 + 3924(AX_k)^7 - 2169(AX_k)^8 \\ &\quad \left. + 901(AX_k)^9 - 264(AX_k)^{10} + 48(AX_k)^{11} - 4(AX_k)^{12} \right] \\ &= \frac{1}{25} (-I + AX_k)^9 (-25I + 4AX_k (3I + AX_k (-3I + AX_k))) \\ &= \frac{1}{25} (I - AX_k)^9 (21I + 4(I - AX_k)^3) \\ &= \frac{1}{25} E_k^9 (21I + 4E_k^3). \end{aligned}$$

在等式两边取矩阵范数得:

$$\|E_{k+1}\| \leq \left\| \frac{1}{25} E_k^9 (2I + 4E_k^3) \right\| = \frac{1}{25} [21\|E_k\|^9 + 4\|E_k\|^{12}]. \quad (3.2)$$

接下来我们使用数学归纳法, 由假设条件(3.1)知 $\|E_0\| < 1$, 代入(3.2)得到:

$$\|E_1\| \leq \frac{1}{25} [21\|E_0\|^9 + 4\|E_0\|^{12}] < 1.$$

现在, 我们假设 $\|E_k\| < 1$, 那么

$$\|E_{k+1}\| \leq \frac{1}{25} [21\|E_k\|^9 + 4\|E_k\|^{12}] \leq \|E_k\|^9.$$

然后, 我们可以得到

$$\|E_{k+1}\| \leq \|E_k\|^9 \leq \dots \leq \|E_0\|^{9^{k+1}} < 1.$$

因此, 当 $k \rightarrow \infty$ 时, $I - AX_k \rightarrow 0$, 也就是说当 $k \rightarrow \infty$ 时, $X_k \rightarrow A^{-1}$.

现在我们证明了在假设(3.1)成立的情况下, 由迭代格式(2.2)生成的矩阵序列 $\{X_k\}_{k=0}^{k=\infty}$ 收敛于矩阵 A^{-1} . 接下来, 我们证明该序列的收敛阶数至少为九阶. 为此, 我们将第 k 步迭代的误差矩阵定义为:

$$\varepsilon_k = A^{-1} - X_k,$$

那么就有 $A\varepsilon_k = I - AX_k = E_k$. 因此在第 $k+1$ 步迭代时有:

$$A\varepsilon_{k+1} = E_{k+1} = \frac{1}{25} E_k^9 (2I + 4E_k^3) = \frac{1}{25} [21(A\varepsilon_k)^9 + 4(A\varepsilon_k)^{12}].$$

由于 A 是可逆的, 由上式可得:

$$\varepsilon_{k+1} = \frac{1}{25} [21\varepsilon_k (A\varepsilon_k)^8 + 4\varepsilon_k (A\varepsilon_k)^{11}].$$

对两边取矩阵范数, 我们可以得到:

$$\|\varepsilon_{k+1}\| \leq \left(\frac{1}{25} [21\|A\|^8 + 4\|A\|^{11} \|\varepsilon_k\|^3] \right) \|\varepsilon_k\|^9.$$

不等式表明, 在假设条件(3.1)成立的情况下, 由迭代格式(2.2)生成的矩阵序列 $\{X_k\}_{k=0}^{k=\infty}$ 至少以九阶收敛于 A^{-1} , 证毕.

在定理 3.1 中分析, 一定条件下, 新构造的迭代方法(2.2)对于非奇异复矩阵是收敛的. 因此, 初始迭代矩阵 X_0 对该方法的收敛性起着重要的作用.

Pan 和 Schreiber 提出, 可以选择

$$X_0 = \alpha A^*, \quad (3.3)$$

作为初始迭代矩阵, 其中 α 是一个足够小的正数, A^* 是给定矩阵 A 的共轭转置矩阵. 式(3.3)中 α 的最优选择是 $\frac{2}{\sigma_1^2 + \sigma_r^2}$, 其中 $\sigma_1 = \|A\|_2$, $\sigma_r = \frac{1}{\|A^*\|_2}$. 然而, σ_r 较难获得, 但是我们可以知道

$$\sigma_{\min} \leq \sigma_1^2 \leq \sigma_r^2 \leq \sigma_{\max},$$

其中, σ_{\min} 和 σ_{\max} 分别是矩阵 A 的奇异值的上界和下界. 根据上述不等式, 可以得到最优一般初始选择:

$$X_0 = \frac{2}{\sigma_{\min}^2 + \sigma_{\max}^2} A^*.$$

除此之外, Pan 和 Schreiber 还提出了 α 的次优值为

$$\alpha = \frac{1}{\|A\|_1 \|A\|_\infty}.$$

另外地, 对于不同类型的矩阵, 可以选取不同的初始矩阵, 可见文献[12]。

3.2. Moore-Penrose 广义逆的收敛性

本小节讨论当矩阵 A 为任意 $m \times n$ 阶复矩阵时, 上述所提出的迭代方案对 Moore-Penrose 逆的收敛性, 我们还将证明它的收敛速度保持不变。接下来, 用引理说明所提出的方案满足 Penrose 方程。

引理 3.2 对于由迭代格式(2.2)以及初始矩阵 $X_0 = \alpha A^*$ 生成的矩阵迭代序列 $\{X_k\}_{k=0}^{k=\infty}$, 以下矩阵方程组成立:

$$\begin{cases} \text{(a)} & X_k A A^\dagger = X_k, \\ \text{(b)} & A^\dagger A X_k = X_k, \\ \text{(c)} & (A X_k)^* = A X_k, \\ \text{(d)} & (X_k A)^* = X_k A. \end{cases}$$

证明: 使用数学归纳法证明引理。对于等式(a), 我们首先证明 $k=0$ 时, 等式成立:

$$X_0 A A^\dagger = \alpha A^* A A^\dagger = \alpha A^* (A A^\dagger)^* = \alpha A^* (A^\dagger)^* A^* = \alpha (A A^\dagger A)^* = \alpha A^* = X_0.$$

假设当 $k=n$, 等式(a)成立, 即 $X_n A A^\dagger = X_n$ 。那么我们要证明当 $k=n+1$ 时, 等式(a)成立, 即 $X_{n+1} A A^\dagger = X_{n+1}$ 。使用迭代格式(2.2), 我们得到:

$$\begin{aligned} X_{n+1} A A^\dagger &= \frac{1}{25} X_n \left[237I - 1020 A X_n + 2644 (A X_n)^2 - 4626 (A X_n)^3 + 5814 (A X_n)^4 \right. \\ &\quad \left. - 5460 (A X_n)^5 + 3924 (A X_n)^6 - 2169 (A X_n)^7 + 901 (A X_n)^8 - 264 (A X_n)^9 \right. \\ &\quad \left. + 48 (A X_n)^{10} - 4 (A X_n)^{11} \right] A A^\dagger \\ &= \frac{1}{25} [237 X_n A A^\dagger - 1020 X_n A X_n A A^\dagger + 2644 X_n A X_n (A X_n A A^\dagger) - 4626 X_n (A X_n)^2 (A X_n A A^\dagger) \\ &\quad + 5814 X_n (A X_n)^3 (A X_n A A^\dagger) - 5460 X_n (A X_n)^4 (A X_n A A^\dagger) + 3924 X_n (A X_n)^5 (A X_n A A^\dagger) \\ &\quad - 2169 X_n (A X_n)^6 (A X_n A A^\dagger) + 901 X_n (A X_n)^7 (A X_n A A^\dagger) - 264 X_n (A X_n)^8 (A X_n A A^\dagger) \\ &\quad + 48 X_n (A X_n)^9 (A X_n A A^\dagger) - 4 X_n (A X_n)^{10} (A X_n A A^\dagger)]. \end{aligned}$$

由已知得 $X_n A A^\dagger = X_n$, 将其代入上式可得:

$$\begin{aligned} X_{n+1} A A^\dagger &= \frac{1}{25} \left[237 X_n - 1020 X_n A X_n + 2644 X_n (A X_n)^2 - 4626 X_n (A X_n)^3 \right. \\ &\quad \left. + 5814 X_n (A X_n)^4 - 5460 X_n (A X_n)^5 + 3924 X_n (A X_n)^6 - 2169 X_n (A X_n)^7 \right. \\ &\quad \left. + 901 X_n (A X_n)^8 - 264 X_n (A X_n)^9 + 48 X_n (A X_n)^{10} - 4 X_n (A X_n)^{11} \right] \\ &= X_n. \end{aligned}$$

因此, 由数学归纳法, 对于任意 $k \in N$, 条件(a)得证。

对于第三个条件等式(c), 首先, 我们验证当 $k=0$ 时, 等式(c)成立:

$$(AX_0)^* = (A\alpha A^*)^* = \alpha(A^*)^* A^* = \alpha AA^* = A\alpha A^* = AX_0.$$

假设当 $k = n$ ，等式(c)成立，即 $(AX_n)^* = AX_n$ 。那么我们要证明当 $k = n + 1$ 时，等式(c)成立，即 $(AX_{n+1})^* = AX_{n+1}$ 。因此，使用迭代格式(2.2)，我们得到：

$$\begin{aligned} (AX_{n+1})^* &= \frac{1}{25} \left[237(AX_k)^* - 1020((AX_k)^2)^* + 2644((AX_k)^3)^* - 4626((AX_k)^4)^* \right. \\ &\quad + 5814((AX_k)^5)^* - 5460((AX_k)^6)^* + 3924((AX_k)^7)^* - 2169((AX_k)^8)^* \\ &\quad \left. + 901((AX_k)^9)^* - 264((AX_k)^{10})^* + 48((AX_k)^{11})^* - 4((AX_k)^{12})^* \right]. \end{aligned}$$

由已知得 $(AX_{n+1})^* = AX_{n+1}$ 成立，那么上式可以写成：

$$\begin{aligned} (AX_{n+1})^* &= \frac{1}{25} \left[237AX_k - 1020(AX_k)^2 + 2644(AX_k)^3 - 4626(AX_k)^4 \right. \\ &\quad \left. + 5814(AX_k)^5 - 5460(AX_k)^6 + 3924(AX_k)^7 - 2169(AX_k)^8 \right]. \end{aligned}$$

因此，由数学归纳法，对于任意 $k \in N$ ，条件(c)得证。同样地，条件(b)和条件(d)也可以类似地证明。综上所述，引理得证。

在下一个定理中，我们将证明迭代格式(2.2)对于 M - P 逆的收敛速度。

引理 3.3 对于任意给定的复矩阵 A ，由迭代格式(2.2)以及初始矩阵 $X_0 = \alpha A^*$ 生成矩阵迭代序列 $\{X_k\}_{k=0}^{k=\infty}$ ，该序列以九阶收敛于给定矩阵的 M - P 逆 A^\dagger 。

证明：我们设 $E_k = X_k - A^\dagger$ ，把它作为用迭代格式(2.2)逼近 M - P 逆时第 k 步的误差矩阵。那么我们可以得到：

$$\begin{aligned} AE_{k+1} &= AX_{k+1} - AA^\dagger = AX_{k+1} - I + I - AA^\dagger = -E_{k+1} + I - AA^\dagger \\ &= \frac{1}{25} (21E_k^9 + 4E_k^{12}) + I - AA^\dagger. \end{aligned}$$

我们可以证明 $(I - AA^\dagger)^t = I - AA^\dagger, t \in N$ ，以及 $(I - AA^\dagger)AE_k = 0$ ，把它们代入上式得到：

$$AE_{k+1} = \frac{1}{25} (21(AE_k)^9 - 4(AE_k)^{12}). \tag{3.4}$$

设 $P = AA^\dagger$ 以及 $S = I - AX_0$ ，那么有 $P^2 = P$ ，还得到：

$$\begin{aligned} PS &= AA^\dagger(I - AX_0) = AA^\dagger - AA^\dagger AX_0 = AA^\dagger - AX_0 \\ &= AA^\dagger - AX_0 AA^\dagger = (I - AX_0)AA^\dagger = SP. \end{aligned}$$

另外，Stanimirović 和 Cvetković-Ilić 表明，若存在 $P \in \mathbb{C}^{n \times n}$ 和 $S \in \mathbb{C}^{n \times n}$ ，使得 $P = P^2$ 和 $PS = SP$ 成立，那么有 $\rho(PS) \leq \rho(S)$ 。因此，我们可以得到：

$$\rho(A(X_0 - A^\dagger)) = \rho(A(\alpha A^* - A^\dagger)) \leq \rho(I - \alpha AA^*) = \max_{1 \leq i \leq r} |1 - \lambda_i(\alpha AA^*)| < 1,$$

其中， r 为奇异值的个数， ρ 为矩阵的谱半径。我们知道，存在一个正常数 ϵ 和一个矩阵范数 $\|\cdot\|$ ，使得：

$$A(X_0 - A^\dagger) \leq \rho(A(X_0 - A^\dagger)) + \epsilon < 1,$$

这也意味着 $\|AE_0\| < 1$ 。此外，对(3.4)两边同时使用矩阵范数可得：

$$\|A\mathbb{E}_{k+1}\| \leq \frac{1}{25} \left[21\|A\mathbb{E}_k\|^9 + 4\|A\mathbb{E}_k\|^{12} \right] \leq \|A\mathbb{E}_k\|^9 \leq \|A\|^9 \|\mathbb{E}_k\|^9. \quad (3.5)$$

现在, 我们提出的迭代格式(2.2)在计算 M - P 逆时的误差为:

$$\|X_{k+1} - A^\dagger\| = \|A^\dagger AX_{k+1} - A^\dagger AA^\dagger\| \leq \|A^\dagger\| \|AX_{k+1} - AA^\dagger\| = \|A^\dagger\| \|A\| \|\mathbb{E}_{k+1}\|.$$

将不等式(3.5)代入到上式中, 可以得到:

$$\|\mathbb{E}_{k+1}\| = \|X_{k+1} - A^\dagger\| \leq \|A\| \|A\|^9 \|\mathbb{E}_k\|^9. \quad (3.6)$$

因此, 当 $k \rightarrow \infty$ 时, $\|X_{k+1} - A^\dagger\| \rightarrow 0$. 因此, (3.6)证明由迭代格式(2.2)生成的矩阵序列 $\{X_k\}_{k=0}^{k=\infty}$ 在假设条件下, 至少以九阶收敛于给定矩阵 A 的 M - P 广义逆。证毕。

3.3. 计算效率

如第 1 节所述, 高阶求矩阵广义逆的最一般的方法是利用 p 次矩阵和矩阵相乘构造局部 p 阶矩阵迭代, 而本文构造的方法(2.2)仅需要使用七次矩阵相乘便具有局部九阶收敛性。Traub 将 p 阶迭代方法的计算效率指数(CEI)定义为:

$$CEI = p^{\frac{1}{\theta}},$$

其中 θ 表示算法总的计算成本。每个 Schulz 型迭代方法的计算成本都由矩阵和矩阵相乘控制, 让我们假设每次矩阵相乘的成本是统一的。那么对于每次迭代需要 μ 次矩阵相乘, 迭代次数为 S 的迭代算法, 其计算成本为 $\theta = \mu S$, Soleymani [8] 提出 p 阶 Schulz 型迭代方法大约需要以下迭代次数:

$$S = 2 \log_p \kappa_2(A),$$

其中 κ_2 是给定矩阵在 $\|\cdot\|_2$ 下的条件数。因此, p 阶迭代方法的计算效率指数为:

$$CEI = p^{\frac{1}{\mu(2 \log_p \kappa_2(A))}}. \quad (3.7)$$

利用计算效率指数这一概念, 我们将迭代格式(1.4)、(1.6)、(1.8)、(1.9)、(1.10)、(1.11)与新构造格式(2.2)进行比较。分别用 1、2、3、4、5、6 以及 new 代表上述格式, 由图 1 可以看出, 在不同的条件数下, 我们提出的方法都具有较高的计算效率。

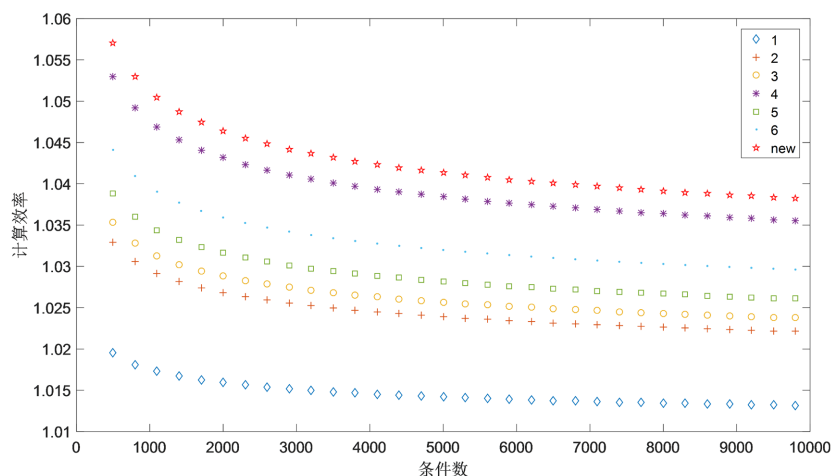


Figure 1. Comparison of calculation efficiency of different methods

图 1. 不同方法计算效率比较

因此,新构造的方法从效率上来说理论有效的。

4. 数值实验

本节给出构造格式(2.2)的数值实验结果,并且与迭代格式(1.4)、(1.6)、(1.8)、(1.9)、(1.10)、(1.11)进行比较,主要进行迭代次数与迭代时间的对比,展现此迭代格式的优势。算法在 MATLABR2016a 编程实现,数值实验在 PC 机上 Windows 系统中进行。

在本节中选择初始迭代矩阵为 $X_0 = \frac{2}{\sigma_{\min}^2 + \sigma_{\max}^2} A^*$, 终止条件为 $\|X_{k+1} - X_k\| < \varepsilon$ [13], 数值实验每次进行多次实验取平均值。数值实验如下:

例 4.1 考虑如下 5 阶病态 Hilbert 矩阵(文献[14]中例 1)

$$A = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}.$$

数值实验结果如表 1 所示,其中迭代格式(1.10)不收敛,迭代格式(2.2)平均迭代次数最小,需要 25 次,平均时间也为最少。

Table 1. Example 4.1 experimental comparison

表 1. 例 4.1 实验对比

	(1.4)	(1.6)	(1.8)	(1.9)	(1.10)	(1.11)	(2.2)
平均迭代次数	78	30	33	31	发散	26	25
平均所需时间(s)	0.00035	0.00045	0.00051	0.00037	-	0.00032	0.00031

例 4.2 考虑如下 5 阶秩亏矩阵(文献[15]中例 4)

$$A = \begin{bmatrix} 0 & 0 & 0 & 2 & 0 \\ 4 & 1 & 0 & 2 & 0 \\ 0 & -2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 2 & 1 & 4 & -3 & 1 \end{bmatrix}$$

数值实验结果如表 2 所示,其中迭代格式(2.2)平均迭代次数最小,需要 29 次,平均时间也为最少。

Table 2. Example 4.2 experimental comparison

表 2. 例 4.2 实验对比

	(1.4)	(1.6)	(1.8)	(1.9)	(1.10)	(1.11)	(2.2)
平均迭代次数	105	40	43	44	37	36	29
平均所需时间(s)	0.00198	0.00214	0.00341	0.00182	0.00220	0.00177	0.00141

例 4.3 考虑随机 $n \times (n+20)$ 阶矩阵 ($n=200$)

数值实验结果如表 3 所示, 其中迭代格式(2.2)平均迭代次数最小, 需要 9 次, 平均时间也为最少。

Table 3. Example 4.3 experimental comparison

表 3. 例 4.3 实验对比

	(1.4)	(1.6)	(1.8)	(1.9)	(1.10)	(1.11)	(2.2)
平均迭代次数	27	13	12	11	10	9	9
平均所需时间(s)	0.01838	0.02107	0.02681	0.01437	0.02218	0.01859	0.01341

从以上数值例子可以看出, 迭代格式(2.2)所使用的平均迭代次数和平均使用时间都是最少的, 因此该算法是有效可行的。

5. 结论

本文构造了一种计算给定复矩阵的 M - P 逆的高阶迭代方案。在一定条件下, 该方法至少能达到九阶收敛性。理论分析表明, 该方案采用了较小的矩阵相乘数, 从而提高其计算效率指标。图 1 表明, 相对于现有的迭代方法, 该方法有着较高的计算效率指标。数值实验结果表明, 该方案在较小的计算时间内准确地得到期望的结果。因此, 研究结果支持并证明所提出的方法的有效性。

参考文献

- [1] Moore, E.H. (1920) On the Reciprocal of the General Algebraic Matrix. *Bulletin of the American Mathematical Society*, **26**, 394-395. <https://doi.org/10.1090/S0002-9904-1920-03332-X>
- [2] Penrose, R. (1955) A Generalized Inverse for Matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, **51**, 406-413. <https://doi.org/10.1017/S0305004100030401>
- [3] Schulz, G. (1933) Iterative Berechnung der reziproken Matrix. *Journal of Applied Mathematics and Mechanics*, **13**, 57-59. <https://doi.org/10.1002/zamm.19330130111>
- [4] Penrose, R. (1956) On Best Approximate Solutions of Linear Matrix Equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, **52**, 17-19. <https://doi.org/10.1017/S0305004100030929>
- [5] Li, W.G. and Li, Z. (2010) A Family of Iterative Methods for Computing the Approximate Inverse of a Square Matrix and Inner Inverse of a Non-Square Matrix. *Applied Mathematics and Computation*, **215**, 3433-3442. <https://doi.org/10.1016/j.amc.2009.10.038>
- [6] Li, H.B., Huang, T.Z., Zhang, Y., Liu, X.P. and Gu, T.X. (2011) Chebyshev-Type Methods and Preconditioning Techniques. *Applied Mathematics and Computation*, **218**, 260-270. <https://doi.org/10.1016/j.amc.2011.05.036>
- [7] Krishnamurthy, E.V. (1986) Numerical Algorithms: Computations in Science and Engineering. Affiliated East-West Press, New Delhi.
- [8] Soleymani, F. (2014) On Finding Robust Approximate Inverses for Large Sparse Matrices. *Linear and Multilinear Algebra*, **62**, 1314-1334. <https://doi.org/10.1080/03081087.2013.825910>
- [9] Soleymani, F., Stanimirović, P.S. and Ullah, M.Z. (2013) An Accelerated Iterative Method for Computing Weighted Moore—Penrose Inverse. *Applied Mathematics and Computation*, **222**, 365-371. <https://doi.org/10.1016/j.amc.2013.07.039>
- [10] Soleymani, F. (2012) A Rapid Numerical Algorithm to Compute Matrix Inversion. *International Journal of Mathematics and Mathematical Sciences*, **2012**, Article ID: 134653. <https://doi.org/10.1155/2012/134653>
- [11] Weiguo, L., Juan, L. and Tiantian, Q. (2013) A Family of Iterative Methods for Computing Moore—Penrose Inverse of a Matrix. *Linear Algebra and Its Applications*, **438**, 47-56. <https://doi.org/10.1016/j.laa.2012.08.004>
- [12] Pan, V.Y. (2010) Newton's Iteration for Matrix Inversion, Advances and Extensions. In: Olshevsky, V. and Tyrtyshnikov, E., Eds., *Matrix Methods: Theory, Algorithms and Applications*, Imperial College Press, London, 364-381. https://doi.org/10.1142/9789812836021_0024
- [13] Wei, Y.M. (2000) Successive Matrix Squaring Algorithm for Computing the Drazin Inverse. *Applied Mathematics and Computation*, **108**, 67-75. [https://doi.org/10.1016/S0096-3003\(98\)10118-2](https://doi.org/10.1016/S0096-3003(98)10118-2)

- [14] Iram, Z.A. and Zein, A. (2022) A New Family of Second-Order Iterative Methods for Computing the Moore Penrose Inverse Based on Penrose Equations. *International Journal of Applied Mathematics*, **35**, 365-380. <https://doi.org/10.12732/ijam.v35i3.1>
- [15] Kaur, M., Kansal, M. and Kumar, S. (2021) An Efficient Matrix Iterative Method for Computing Moore—Penrose Inverse. *Mediterranean Journal of Mathematics*, **18**, Article No. 42. <https://doi.org/10.1007/s00009-020-01675-4>