计算矩阵主平方根和符号函数的递推算法及其稳定性*

连 续 仁　　谢 两 参
（石油工业部地球物理勘探局）　（休斯顿大学电机系）

THE RECURSIVE ALGORITHMS FOR COMPUTING THE PRINCIPAL SQUARE ROOT OF A MATRIX AND THE MATRIX SIGN FUNCTION

Lian Sui-ren　　Shieh Leang-san
(*China Oil and Gas Co.*)　(*University of Houston*)

Abstract

This paper proposes the recursive algorithms that are fast convergent and numerically stable for computing the square root of a complex matrix and the associated matrix sign function. The stability of the proposed algorithms is investigated. An example is presented to demonstrate the effectiveness of the proposed algorithms.

一、引　言

矩阵的主平方根和矩阵符号函数在控制理论中有许多用途．例如求解矩阵的李亚普诺夫方程和矩阵的黎卡提方程[1-4]，大规模系统的降阶[5,6]和离散系统模型——连续系统模型的转换[7]等．常用的矩阵开方的算法有：从矩阵连分式导出的矩阵开方算法[8,9]，利用 Newton-Raphson 法得到的矩阵开方算法[10,11,12] 以及从矩阵符号函数导出的矩阵开方算法[13,14]．

若 $A\in C^{n\times n}$，其特征值为 $\sigma(A)=\{\lambda_i, i=1,2,\cdots,n\}$，其中 $\lambda_i\neq 0$，$\arg(\lambda_i)\neq\pi$，那么从 Newton-Raphson 法得到的计算矩阵平方根的算法为

$$G(k+1)=\frac{1}{2}[G(k)+G(k)^{-1}A],\quad G(0)=I_n,\quad \lim_{k\to\infty}G(k)=\sqrt{A}; \tag{1a}$$

* 1988 年 7 月 16 日收到．

或者

$$H(k+1)=\frac{1}{2}[H(k)+AH(k)^{-1}],\ H(0)=I_n,\ \lim_{k\to\infty}H(k)=\sqrt{A}. \tag{1b}$$

从矩阵符号函数导出的计算矩阵平方根的算法为

$$\begin{cases} M(k+1)=\frac{1}{2}[M(k)+N(k)^{-1}],\ M(0)=A,\ \lim M(k)=\sqrt{A}, \\ N(k+1)=\frac{1}{2}[M(k)^{-1}+N(k)],N(0)=I_n,\ \lim N(k)=(\sqrt{A})^{-1}; \end{cases} \tag{1c}$$

式中矩阵$\sqrt{A}$(≙$A^{\frac{1}{2}}$)表示矩阵A的主平方根. 它定义为$(\sqrt{A})^2=A$，并且

$$\arg(\sigma(\sqrt{A}))\ \in\left(-\frac{\pi}{2},\ \frac{\pi}{2}\right).$$

算法(1)具有 2 阶收敛速度并且算法(1c)数值稳定[12]. 如果矩阵A包含实的负特征值，算法(1)不能直接使用. 这时需要把A修改为 $\hat{A}=A\exp(-j\Delta\theta)$，使得 $\hat{A}$ 没有实的负特征值. 那么 $A^{\frac{1}{2}}=\hat{A}^{\frac{1}{2}}\exp\left(j\frac{\Delta\theta}{2}\right)$，其中 $\Delta\theta$ 是一个很小的角度.

为了改善收敛速度，文献[9]提出下述矩阵开方算法:

$$\begin{bmatrix}X_1(k+1)\\X_2(k+1)\end{bmatrix}=\begin{bmatrix}X_1(k) & AX_2(k)\\X_2(k) & X_1(k)\end{bmatrix}^{r-1}\begin{bmatrix}X_1(k)\\X_2(k)\end{bmatrix},\begin{bmatrix}X_1(0)\\X_2(0)\end{bmatrix}=\begin{bmatrix}I_n\\I_n\end{bmatrix}, \tag{2a}$$

$$\lim_{k\to\infty}X_1(k)X_2(k)^{-1}=\sqrt{A}, \tag{2b}$$

式中 $r=2,3,4\cdots$. 算法(2)具有 r 阶收敛速度. 算法(2)能够较快地收敛. 但是，如果矩阵A同时含有很大和很小的特征值，算法(2)会变得数值不稳定. 本文在算法(2)的基础上提出收敛快而且数值稳定的矩阵开方算法.

二、算法(2)的稳定性分析

令 $X(k)$≙$X_1(k)X_2(k)^{-1}$，那么算法(2)在 $r=2,3,4,5$ 时成为:

$r=2$ 时，

$$X(k+1)=X(k)[I_n+AX(k)^{-2}]/2,\ X(0)=I_n. \tag{3}$$

$r=3$ 时，

$$X(k+1)=X(k)[3I_n+AX(k)^{-2}]^{-1}[I_n+3AX(k)^{-2}],X(0)=I_n. \tag{4}$$

$r=4$ 时，

$$X(k+1)=X(k)[4I_n+4AX(k)^{-2}]^{-1}[I_n+6AX(k)^{-2}+(AX(k)^{-2})^2],X(0)=I_n. \tag{5}$$

$r=5$ 时:

$$X(k+1)=X(k)[5I_n+10AX(k)^{-2}+(AX(k)^{-2})^2]^{-1}\cdot[I_n+10AX(k)^{-2}+5(AX(k)^{-2})^2],\ X(0)=I_n. \tag{6}$$

算法(3)—(6)分别具有 2,3,4,5 阶收敛速度.

定理1. 如果矩阵A可以变换为对角线矩阵，那么在某次递推时存在舍入误差，而在以后的递推中不产生舍入误差的假定下，算法(3)—(6)为数值稳定的充分条件是A的特征值满足

$$|K_{ij}| = \left| 1 - \frac{\sqrt{\frac{\lambda_i}{\lambda_j}} + \frac{\lambda_i}{\lambda_j}}{2} \right| \leqslant 1, \quad \begin{matrix} i,j = 1,2,\cdots,n, \\ i \neq j. \end{matrix} \tag{7}$$

证明 我们来分析存在舍入误差时算法(3)的数值稳定性. 假定矩阵A可以变换成对角线矩阵，即存在矩阵M使得

$$M^{-1}AM = \mathrm{diag}\{\lambda_i, i = 1,2,\cdots,n\} \triangleq \Lambda. \tag{8}$$

于是，可以把算法(3)对角线化得到

$$D(k+1) = D(k)[I_n + \Lambda D(k)^{-2}]/2, \quad D(0) = I_n, \tag{9a}$$

其中 $D(k) = M^{-1}X(k)M = \mathrm{diag}\{d_i(k),\ i = 1,2,\cdots,n\}$，并且

$$\lim_{k\to\infty} D(k) = \mathrm{diag}\{\sqrt{\lambda_i},\ i = 1,2,\cdots,n\} \triangleq \mathrm{diag}\{d_i,\ i = 1,2,\cdots,n\}, \tag{9b}$$

这里$\sqrt{\lambda_i} \triangleq d_i$是$\lambda_i$的主平方根.

如果在第k_0次递推时存在舍入误差$E(k_0)$，经过舍入后的递推结果记为$\tilde{X}(k_0)$，那么

$$\tilde{X}(k_0) = X(k_0) + E(k_0). \tag{10}$$

第k次($k > k_0$)递推结果记以$\tilde{X}(k)$，相应的误差记以$E(k)$，那么

$$\tilde{X}(k) = X(k) + E(k). \tag{11}$$

我们的目的是分析在$k > k_0$时$E(k_0)$是如何传播的，所以进一步假定在$k > k_0$时用下式由$\tilde{X}(k)$计算$\tilde{X}(k+1)$时不产生舍入误差

$$\tilde{X}(k+1) = \tilde{X}(k)[I_n + A\tilde{X}(k)^{-2}]/2, \quad k = k_0, k_0+1,\cdots \tag{12}$$

这个假定表示$E(k)$($k > k_0$)完全是由于$E(k_0)$引起的. 式(12)的初值是$\tilde{X}(k_0)$. 把式(11)代入式(12)，再将所得结果减去式(3)，利用下面的矩阵摄动公式

$$(H+\Delta)^{-1} = H^{-1} - H^{-1}\Delta H^{-1} + O(\|\Delta^2\|), \tag{13}$$

其中H和Δ是矩阵，$O(\|\Delta^2\|)$是Δ的高阶小量，略去$E(k)$的高阶小量后得到

$$\begin{aligned} E(k+1) \cong \{&E(k)[I_n + AX(k)^{-2}] - X(k)AX(k)^{-1}E(k)X(k)^{-2} \\ &- X(k)AX(k)^{-2}E(k)X(k)^{-1}\}/2, \quad k = k_0, k_0+1,\cdots \end{aligned} \tag{14}$$

式(14)的初值是$E(k_0)$. 用矩阵M对(14)进行变换，得到

$$\begin{aligned} \varepsilon(k+1) \cong \{&\varepsilon(k)[I_n + \Lambda D(k)^{-2}] - D(k)\Lambda D(k)^{-1}\varepsilon(k)D(k)^{-2} \\ &- D(k)\Lambda D(k)^{-2}\varepsilon(k)D(k)^{-1}\}/2, \quad k = k_0, k_0+1,\cdots, \end{aligned} \tag{15}$$

其中$\varepsilon(k) = M^{-1}E(k)M$，$\varepsilon(k_0) = M^{-1}E(k_0)M$. 矩阵方程式(15)等价于下面$n^2$个数量差分方程式

$$\begin{aligned} \varepsilon_{ij}(k+1) \cong \varepsilon_{ij}(k)[1 + \lambda_j d_j(k)^{-2} - \lambda_i d_i(k)^{-2} - \lambda_i d_i(k)^{-1}d_j(k)^{-1}]/2, \\ i,j = 1,2,\cdots,n, \quad k = k_0, k_0+1,\cdots, \end{aligned} \tag{16}$$

其中$\varepsilon_{ij}(k)$和$\varepsilon_{ij}(k_0)$分别是矩阵$\varepsilon(k)$和$\varepsilon(k_0)$的元素. 当$k \to \infty$时，

$$d_i(k) \to \sqrt{\lambda_i}, \quad i = 1,2,\cdots,n.$$

于是式(16)成为

$$\varepsilon_{ij}(k+1)=K_{ij}\varepsilon_{ij}(k),\quad k=k_0,\ k_0+1,\cdots, \tag{17a}$$

$$K_{ij}=1-\frac{\sqrt{\frac{\lambda_i}{\lambda_j}}+\frac{\lambda_i}{\lambda_j}}{2}. \tag{17b}$$

差分方程式(17a)的解是

$$\varepsilon_{ij}(k)=K_{ij}^{k-k_0}\varepsilon_{ij}(k_0),\quad k=k_0,\ k_0+1,\cdots. \tag{17c}$$

在式(17)中 $i,j=1,2,\cdots,n$. 注意 $K_{ii}=0, i=1,2,\cdots,n$. 如果条件(7)成立，则算法(3)数值稳定，即第 k_0 次递推产生的舍入误差不会使以后的递推结果的误差无限增长.

由于

$$\begin{aligned}E(k)&=M\varepsilon(k)M^{-1}\\&=M\{K_{ij}^{k-k_0}\varepsilon_{ij}(k_0),i,j=1,2,\cdots,n\}M^{-1}\\&\triangleq\{\delta_{pq}(k),p,q=1,2,\cdots,n\},\end{aligned} \tag{18a}$$

其中 $\delta_{pq}(k)$ 可以写成

$$\delta_{pq}(k)=\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_{ij}^{pq}K_{ij}^{k-k_0}\varepsilon_{ij}(k_0), \tag{18b}$$

α_{ij}^{pq} 是决定于矩阵M的常数. 式(18)还可以进一步改写成

$$E(k)=\sum_{i=1}^{n}\sum_{j=1}^{n}\{\alpha_{ij}^{pq},p,q=1,2,\cdots,n\}K_{ij}^{k-k_0}\varepsilon_{ij}(k_0). \tag{19}$$

如果某一个 $K_{i^*j^*}$ 有 $|K_{i^*j^*}|>1$，但其余的 K_{ij} 有 $|K_{ij}|\leqslant 1$ $(i\neq i^*,\ j\neq j^*)$，并且式(19)中 $K_{i^*j^*}$ 的系数矩阵 $\{\alpha_{i^*j^*}^{pq},p,q=1,2,\cdots,n\}$ 正好是零矩阵(这是可能的)，那么尽管条件(7)不成立，算法(3)仍是数值稳定的. 所以，条件(7)并不是必要条件.

令 $|K_{ij\max}|\triangleq\mathrm{Max}(|K_{ij}|,i,j=1,2,\cdots,n)$，矩阵 $\{\alpha_{ij}^{pq},p,q=1,2,\cdots,n\}_{\max}$ 是式(19)中 $K_{ij\max}$ 的系数矩阵；令 $|K_{ij_{sub}\max}|\triangleq Sub\max(|K_{ij}|,i,j=1,2,\cdots,n)$，矩阵 $\{\alpha_{ij}^{pq},p,q=1,2,\cdots,n\}_{sub\max}$ 为式(19)中 $K_{ij_{sub}\max}$ 的系数矩阵. 若 $|K_{ij\max}|>1$，并且 $\{\alpha_{ij}^{pq},p,q=1,2,\cdots,n\}_{\max}\neq 0_n$，那么当 $k\to\infty$ 时从式(19)可得

$$\|E(k)\|\infty\|\{\alpha_{ij}^{pq},\ p,q=1,2,\cdots,n\}\max\|\cdot|K_{ij\max}|^{k-k_0}. \tag{20}$$

算法(3)不稳定时误差发散速度定义为

$$R\triangleq\|E(k+1)\|/\|E(k)\|.$$

从式(20)可得

$$R=|K_{ij\max}|. \tag{21}$$

倘若 $\{\alpha_{ij}^{pq},\ p,q=1,2,\cdots,n\}_{\max}=0_n$ 而 $\{\alpha_{ij}^{pq},\ p,q=1,2,\cdots,n\}_{sub\max}\neq 0_n$ 且 $|K_{ij_{sub}\max}|>1$，则当 $k\to\infty$ 时

$$\|E(k)\|\infty\|\{\alpha_{ij}^{pq},p,q=1,2,\cdots,n\}_{sub\max}\|\cdot|K_{ij_{sub}\max}|^{k-k_0}. \tag{22}$$

此时，误差发散速度为

$$R=|K_{ij_{sub}\max}|. \tag{23}$$

把同样的方法用于算法(4)、(5)、(6)可以导出同样的结果. 定理 1 证毕. 由定理 1 直接

得到下面的结论：

推论 1. 如果矩阵 A 的所有特征值是正实数，则算法(3)—(6)为数值稳定的充分条件是

$$\left|\frac{\lambda_{max}}{\lambda_{min}}\right| \leqslant 2.438, \tag{24}$$

其中 λ_{max} 和 λ_{min} 分别是矩阵 A 的最大和最小特征值．解不等式

$$\left|1-\frac{\sqrt{\frac{\lambda_{max}}{\lambda_{min}}}+\frac{\lambda_{max}}{\lambda_{min}}}{2}\right| \leqslant 1$$

便得到(24)．

作者做许多实例，表明条件(7)、(24)正确．

三、收敛较快并且数值稳定的矩阵开方的算法

为了得到收敛快而且数值稳定的矩阵开方算法，把算法(2)修改如下：

用对角线方块矩阵 block diag $[X_1(k)^{-r}, X_1(k)^{-r}]$ 乘式(2a)的两边，并定义

$$X_1(k)^{-r}X_1(k+1)\triangleq\hat{X}_1(k+1),\quad X_1(k)^{-r}X_2(k+1)\triangleq\hat{X}_2(k+1),$$
$$X_1(k)X_2(k)^{-1}=\hat{X}_1(k)\hat{X}_2(k)^{-1}\triangleq X(k),$$

我们得到

$$\begin{bmatrix}\hat{X}_1(k+1)\\ \hat{X}_2(k+1)\end{bmatrix}=\begin{bmatrix}I_n & AX(k)^{-1}\\ X(k)^{-1} & I_n\end{bmatrix}^{r-1}\begin{bmatrix}I_n\\ X(k)^{-1}\end{bmatrix},\ X(0)=I_n, \tag{25a}$$

$$X(k)=\hat{X}_1(k)\hat{X}_2(k)^{-1}, \tag{25b}$$

$$\lim_{k\to\infty}X(k)=\sqrt{A}. \tag{25c}$$

定义

$$\begin{bmatrix}P_r(k)\\ X(k)^{-1}Q_r(k)\end{bmatrix}\triangleq\begin{bmatrix}I_n & AX(k)^{-1}\\ X(k)^{-1} & I_n\end{bmatrix}^{r-1}\begin{bmatrix}I_n\\ X(k)^{-1}\end{bmatrix}=\begin{bmatrix}\hat{X}_1(k+1)\\ \hat{X}_2(k+1)\end{bmatrix}, \tag{26a}$$

那么

$$\begin{bmatrix}P_{r-1}(k)\\ X(k)^{-1}Q_{r-1}(k)\end{bmatrix}\triangleq\begin{bmatrix}I_n & AX(k)^{-1}\\ X(k)^{-1} & I_n\end{bmatrix}^{r-2}\begin{bmatrix}I_n\\ X(k)^{-1}\end{bmatrix}. \tag{26b}$$

合并(26a)和(26b)得到

$$\begin{bmatrix}P_r(k)\\ Q_r(k)\end{bmatrix}=\begin{bmatrix}I_n & AX(k)^{-2}\\ I_n & I_n\end{bmatrix}\begin{bmatrix}P_{r-1}(k)\\ Q_{r-1}(k)\end{bmatrix},\ \begin{bmatrix}P_1(k)\\ Q_1(k)\end{bmatrix}=\begin{bmatrix}I_n\\ I_n\end{bmatrix}. \tag{26c}$$

利用 $\hat{X}_1(k+1)=P_r(k)$，$\hat{X}_2(k+1)=X(k)^{-1}Q_r(k)$ 和(25b)得到

$$X(k+1)=X(k)Q_r(k)^{-1}P_r(k),\ X(0)=I_n, \tag{27a}$$

展开(26c)得到

$$P_i(k)=P_{i-1}(k)+AX(k)^{-2}Q_{i-1}(k),\ P_1(k)=I_n, \tag{27b}$$

$$Q_i(k)=P_{i-1}(k)+Q_{i-1}(k),\ Q_1(k)=I_n,\ i=2,3,\cdots,r \tag{27c}$$

并且

$$\lim_{k\to\infty} X(k) = \sqrt{A}. \tag{27d}$$

在式(27)中取 $r = 2,3,4,5$ 便得到(3),(4),(5)和(6).

定义 $G(k)\triangleq AX(k)^{-2}$,从式(27a)得到

$$G(k+1) = G(k)[Q_r(k)P_r(k)^{-1}]^2,\ G(0) = A. \tag{28}$$

由式(27)和(28)得到

$$X(k+1) = X(k)Q_r(k)^{-1}P_r(k),\ X(0) = I_n, \tag{29a}$$

$$G(k+1) = G(k)[Q_r(k)P_r(k)^{-1}]^2,\ G(0) = A, \tag{29b}$$

$$P_l(k) = P_{l-1}(k) + G(k)Q_{l-1}(k),\ P_1(k) = I_n, \tag{29c}$$

$$Q_l(k) = P_{l-1}(k) + Q_{l-1}(k),\ Q_1(k) = I_n, \tag{29d}$$

其中 $k = 0,1,2,\cdots,\ l = 2,3,\cdots,r$.

$$\lim_{k\to\infty} X(k) = \sqrt{A}. \tag{29e}$$

由 $G(k)$ 的定义和式(29e)得到

$$\lim_{k\to\infty} G(k) = I_n. \tag{29f}$$

在(29)中取 $r = 2$ 得到

$$X(k+1) = X(k)[I_n + G(k)]/2,\ X(0) = I_n, \tag{30a}$$

$$G(k+1) = G(k)\{[2I_n][I_n + G(k)]^{-1}\}^2, G(0) = A. \tag{30b}$$

取 $r = 3$ 得到

$$X(k+1) = X(k)[3I_n + G(k)]^{-1}[I_n + 3G(k)],\ X(0) = I_n, \tag{31a}$$

$$G(k+1) = G(k)\{[3I_n + G(k)][I_n + 3G(k)]^{-1}\}^2,\ G(0) = A. \tag{31b}$$

取 $r = 4$ 得到

$$X(k+1) = X(k)[4I_n + 4G(k)]^{-1}[I_n + 6G(k) + G(k)^2],\ X(0) = I_n, \tag{32a}$$

$$G(k+1) = G(k)\{[4I_n + 4G(k)][I_n + 6G(k) + G(k)^2]^{-1}\}^2,\ G(0) = A. \tag{32b}$$

取 $r = 5$ 得到

$$X(k+1) = X(k)[5I_n + 10G(k) + G(k)^2]^{-1}[I_n + 10G(k) + 5G(k)^2],$$
$$X(0) = I_n, \tag{33a}$$

$$G(k+1) = G(k)\{[5I_n + 10G(k) + G(k)^2][I_n + 10G(k) + 5G(k)^2]^{-1}\}^2,$$
$$G(0) = A. \tag{33b}$$

算法(30),(31),(32),(33)分别具有 2, 3,4, 5 阶收敛速度.

定理 2. 算法(29)具有 r 阶收敛速度并且在下述意义下数值稳定: 第 k 次递推存在舍入误差,在以后的递推中不产生递推误差,则第 k 次递推的舍入误差不会引起以后的递推误差无限增加.

证明. 算法(29)是从算法(2)导出的,所以它的收敛速度与算法(2)的收敛速度相同. 算法(29)的数值稳定性证明如下:

考虑 $r = 2$ 的情形即算法(30).第 k 次递推时 $G(k)$和 $X(k)$的舍入误差分别是 $E(k)$ 和 $F(k)$,经过舍入后的递推结果分别记为 $\tilde{G}(k)$ 和 $\tilde{X}(k)$. 那么

$$\tilde{G}(k) = G(k) + E(k), \tag{34a}$$

$$\tilde{X}(k) = X(k) + F(k). \tag{34b}$$

我们的目的是分析误差矩阵 $E(k)$ 和 $F(k)$ 如何传播到第$(k+1)$次递推结果. 因此，进一步假定在用 $\tilde{G}(k)$ 和 $\tilde{X}(k)$ 计算 $\tilde{G}(k+1)$ 和 $\tilde{X}(k+1)$ 时没有舍入误差，那么

$$\tilde{G}(k+1) = \hat{G}(k)\{[2I_n][I_n + \hat{G}(k)]^{-1}\}^2, \tag{35a}$$

$$\tilde{X}(k+1) = \tilde{X}(k)[I_n + \tilde{G}(k)]/2. \tag{35b}$$

把(34)代入(35)，利用矩阵摄动公式(13)并略去 $E(k)$ 和 $F(k)$ 的高阶小量，得到

$$\begin{aligned} E(k+1) = 4\{&E(k)[I_n + G(k)]^{-2} - G(k)[I_n + G(k)]^{-1}E(k)[I_n + G(k)]^{-2} \\ &- G(k)[I_n + G(k)]^{-2}E(k)[I_n + G(k)]^{-1}\}, \end{aligned} \tag{36a}$$

$$F(k+1) = \{F(k)[I_n + G(k)] + X(k)E(k)\}/2. \tag{36b}$$

当 $k \to \infty$ 时 $X(k) \to \sqrt{A}$，$G(k) = AX(k)^{-2} \to I_n$，于是式(36)成为

$$E(k+1) = O_n, \tag{37a}$$

$$F(k+1) = F(k) + X(k)E(k)/2. \tag{37b}$$

在(37a)中系统矩阵为零矩阵，在(37b)中系统矩阵的特征值为1，因此矩阵状态方程式(37)是稳定的. 进一步假定第$(k+2)$次递推时没有舍入误差，那么(37b)成为

$$F(k+2) = F(k+1) + X(k+1)E(k+1)/2 = F(k+1).$$

这表明第 k 次递推时的舍入误差不会引起以后各次递推误差无限增加. 用同样的方法可以证明算法(29)在 $r > 2$ 时也是数值稳定的.

四、矩阵的符号函数

矩阵的符号函数在控制系统中有许多应用[1-6,13,14]. 矩阵开方算法的一个重要应用是计算矩阵的符号函数. 略加修改算法(29)便可以得到收敛较快而且数值稳定的计算矩阵符号函数的算法. 矩阵 A 的符号函数定义为

$$\mathrm{Sign}(A) = A(\sqrt{A^2})^{-1} = A^{-1}(\sqrt{A^2}),$$

其中 $\sqrt{A^2}$ 是矩阵 A^2 的主平方根.

令 $S(k) \triangleq A^{-1}X(k)$，$G(k) = A^2X(k)^{-2}$，取 $X(0) = A^2$，$G(0) = A^{-2}$，从算法(29)可得计算矩阵符号函数的算法如下：

$$S(k+1) = S(k)Q_r(k)^{-1}P_r(k), \quad S(0) = A, \tag{38a}$$

$$P_l(k) = P_{l-1}(k) + S(k)^{-2}Q_{l-1}(k), \quad P_1(k) = I_n, \tag{38b}$$

$$Q_l(k) = P_{l-1}(k) + Q_{l-1}(k), \quad Q_1(k) = I_n, \tag{38c}$$

其中 $k = 0,1,2\cdots$，$l = 2,3,\cdots,r$.

$$\lim_{k\to\infty} S(k) = \mathrm{Sign}(A). \tag{38d}$$

式(38)中 r 是算法的收敛阶数.

推论 2. 算法(38)具有 r 阶收敛速度并且在下述意义下数值稳定：第 k 次递推存在舍入误差，在以后的递推中不产生舍入误差，则第 k 次递推的舍入误差不会引起以后的递推误差无限增加.

在算法(38)中令 $r=2,3,4,5$ 得到：

$r=2$ 时，

$$S(k+1)=[I_n+S(k)^2][2S(k)]^{-1}=\frac{1}{2}[S(k)+S(k)^{-1}],\quad S(0)=A. \tag{39}$$

$r=3$ 时，

$$S(k+1)=S(k)[3I_n+S(k)^2][I_n+3S(k)^2]^{-1},\quad S(0)=A. \tag{40}$$

$r=4$ 时，

$$S(k+1)=[I_n+6S(k)^2+S(k)^4][4S(k)+4S(k)^3]^{-1},\quad S(0)=A. \tag{41}$$

$r=5$ 时，

$$S(k+1)=S(k)[5I_n+10S(k)^2+S(k)^4][I_n+10S(k)^2+5S(k)^4]^{-1},\quad S(0)=A. \tag{42}$$

算法(39),(40),(41),(42)分别具有 2,3, 4, 5 阶收敛速度. 算法(39)是通常使用的计算矩阵符号函数的算法[13].

五、例　题

考虑矩阵 A

$$A=\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0.01 & 0 & 0 \\ -1 & -1 & 100 & 100 \\ -1 & -1 & -100 & 100 \end{bmatrix}.$$

它的主平方根$\sqrt{A}$是

$$\sqrt{A}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.909090 & 0.1 & 0 & 0 \\ -0.045509 & -0.045506 & 10.987 & 4.5509 \\ -0.108960 & 0.10888 & -4.5509 & 10.987 \end{bmatrix}.$$

矩阵 A 的特征值是 $\sigma(A)=\{0.01,1,100\pm j100\}$，矩阵$\sqrt{A}$ 的特征值是

$$\sigma(\sqrt{A})=\{0.1,1,10.987\pm j4.5509\}.$$

使用算法 (1a)、(1b)、(1c)、(2)、(30)、(31)、(32)、(33) 来计算矩阵 A 的主平方根. 第 k 次递推的误差 $e(k)$ 定义为 $e(k)=\|X(k)-\sqrt{A}\|$，其中 $X(k)$ 是上述算法在第 k 次递推时的计算结果. 在 VAX11/750 计算机上进行计算. 算法 (1a)，在 $k=8$ 时误差下降到 $e(k)=0.2816\times10^{-13}$，然后单调上升；在 $k=9$ 时的误差 $e(k)=0.1268\times10^{-12}$，在 $k=25$ 时误差增加到 $e(k)=0.3585\times10^{-2}$. 算法(1b)，在 $k=8$ 时的误差下降到 $e(k)=0.8905\times10^{-8}$，然后发散；在 $k=9$ 时的误差是 $e(k)=0.4142\times10^{-6}$，当 $k=16$ 时误差迅速上升到 $e(k)=0.1295\times10^{7}$. 可见，算法(1a)、(1b)是数值不稳定的. 算法(1c)在 $k=9$ 时误差下降到 $e(k)=0.2220\times10^{-15}$，在 $k\geq9$ 时的误差不变. 可见，

算法(1c)是数值稳定的。算法(2)(取 $r=2$),在 $k=4$ 时误差下降到 $e(k)=2.192$,然后迅速增大,在 $k=5$ 时的误差是 $e(k)=0.5629\times10^{15}$。可见算法(2)是数值不稳定的。本文提出的算法(30)在 $k=6$ 时的误差下降至 $e(k)=0.5439\times10^{-14}$,在 $k\geqslant6$ 时误差不变。算法(31)在 $k=5$ 时的误差下降至 $e(k)=0.3640\times10^{-11}$,在 $k\geqslant5$ 时误差不变。算法(32)在 $k=4$ 时的误差减小到 $e(k)=0.1251\times10^{-12}$,在 $k\geqslant4$ 时误差不变。算法(33)在 $k=3$ 时的误差下降到 $e(k)=0.9772\times10^{-9}$,在 $k\geqslant3$ 时误差不变。可见本文提出的算法不仅收敛较快而且数值稳定。

六、结 论

本文提出了收敛较快而且数值稳定的计算矩阵主平方根的递推算法并举例说明其效果。在此基础上导出了收敛较快的计算矩阵符号函数的递推算法。证明了这些递推算法的数值稳定性。还分析了现有的从矩阵连分式法导出的收敛较快的矩阵开方算法的稳定性并得到稳定条件。

参 考 文 献

[1] E. D. Denman, A. N. Beavers, The matrix sign function and computations in systems, Appl. Math. Comput., Vol. 2, 1976 pp. 63—94.

[2] G. J. Bierman, Computational aspects of the matrix sign function solution to the ARE, Proc. of 23rd Conference on Decision and Control, 1984 pp. 514—519.

[3] J. D. Gardiner, A. J. Laub, A generalization of the matrix sign function solution for algebraic Riccati Equations, Proc. of 24th Conference on Decision and Control, 1985, pp. 1233—1235.

[4] L. S. Shieh, C. T. Wang, Y. T. Tasy, Fast suboptimal state-space selftuner for linear stochastic multivariable systems, Proc. IEE-D, Vol. 130, 1983, pp. 143—154.

[5] L. S. Shieh, Y. T. Tsay, Algebra-geometric approach for the model reduction of large-scale multivariable systems, Proc. IEE-D. Vol. 131, 1984, pp. 23—26.

[6] L. S. Shieh, H. M. Dib, R. E. Yates, Separation of matrix eigenvalues and structural decomposition of large-scale systems, Proc. IEE-D. Vol. 133, 1986, pp. 90—96.

[7] L. S. Shieh, J. S. H. Tsai, S. R. Lian, Determining continuous-time state equations from discrete-time state equations via the principal qth root method, IEEE Trans. Automat. Contr., Vol. AC-31, 1986, pp. 454—457.

[8] L. S. Shieh, Y. t. Tsay, R. E. Yates, Computation of the principal nth roots of complex matrices, IEEE Trans. Automat. Contr., Vol. AC-30, 1985, pp. 606—608.

[9] Y. T. Tsay, L. S. Shieh, J. S. H. Tsai, A fast method for computing the principal nth roots of complex matrices, Linear Algebra Appl., Vol. 76, 1986, pp. 205—221.

[10] W. D. Hoskins, D. J. Walton, A fast method of computing the square root of a matrix, IEEE Trans. Automat. Contr., AC-23, 1978, pp. 494—495.

[11] W. D. Hoskins D. J. Walton, A fast, more stable method for computing the pth roots of positive defenite matrices, Linear Algebra Appl., Vol. 26, 1979, pp. 139—163.

[12] N. J. Higham, Newton's method for the matrix square root, Mathematics of Computation, Vol. 46, 1986, pp. 537—549.

[13] J. D. Roberts, Linear model reduction and solution of the algebraic Riccati equation by use of the sign function, CUED/B-Control/TR 13 Report, Cambridge University, 1971; also, Int. J. Control, Vol. 32, 1980, pp. 677—687.

[14] L. S. Shieh, Y. T. Tsay R. E. Yates, Some properties of matrix sign function derived from continued fractions, Proc. IEE-d, Vol. 130, 1983, pp. 111—118.