J. KSIAM Vol.12, No.2, 55–68, 2008

SOLVING MATRIX POLYNOMIALS BY NEWTON'S METHOD WITH EXACT LINE SEARCHES

JONG HYEON SEO[1] AND HYUN-MIN KIM[2†]

[1]DEPARTMENT OF MATHEMATICS, PUSAN NATIONAL UNIVERSITY, SOUTH KOREA
E-mail address: hyeonni94@hanmail.net

[2]DEPARTMENT OF MATHEMATICS, PUSAN NATIONAL UNIVERSITY, SOUTH KOREA
E-mail address: hyunmin@pnu.edu

ABSTRACT. One of well known and much studied nonlinear matrix equations is the matrix polynomial which has the form $P(X) = A_0X^m + A_1X^{m-1} + \cdots + A_m$, where $A_0, A_1, \cdots, A_m$ and X are $n \times n$ complex matrices. Newton's method was introduced a useful tool for solving the equation $P(X) = 0$. Here, we suggest an improved approach to solve each Newton step and consider how to incorporate line searches into Newton's method for solving the matrix polynomial. Finally, we give some numerical experiment to show that line searches reduce the number of iterations for convergence.

1. INTRODUCTION

Nonlinear matrix equations often occur in applications and modeling of scientific problems. In this work we specially consider one of the nonlinear matrix equations which is called the matrix polynomial. A matrix polynomial can be defined by

$$P(X) = A_0X^m + A_1X^{m-1} + \cdots + A_m = 0, \tag{1}$$

where $A_0, A_1, \cdots, A_m, X \in \mathbb{C}^{n\times n}$. If $A_0 = I$ where I is an $n \times n$ identity matrix, then we say that $P(X)$ is monic. Matrix polynomials appear in the theory of differential equations, system theory, network theory, stochastic theory and other areas [8], [9], [10], [11]. A matrix S satisfying the equation $P(S) = 0$ is called a solvent, or more precisely, a right solvent of $P(X)$ to distinguish it from a left solvent, which is a solution of the related matrix equation

$$X^mA_0 + X^{m-1}A_1 + \cdots + A_m = 0.$$

In the quadratic case($m = 2$), Davis [1], [2] considered Newton's Method and Higham and Kim [5], [6] incorporated the exact line searches into Newton's method and developed functional iterations. For solving matrix polynomials Newton's method was considered by

2000 *Mathematics Subject Classification.* 65F30, 65H10.
Key words and phrases. matrix polynomial, solvent, Newton's method, line searches.
† Corresponding author: This author was supported for two years by Pusan National University Research Grant.

Kratz and Stickel [7] and Berlloulli's iteration was suggested by Dennis, Jr., Traub and Weber [3], [4].

Much motivation for studying the matrix polynomial comes from the polynomial eigenvalue problems

$$P(\lambda)\mathbf{v} = (A_0\lambda^m + A_1\lambda^{m-1} + \cdots + A_m)\mathbf{v} = 0. \tag{2}$$

If $P(\lambda_0)$ is singular, λ_0 is called a polynomial eigenvalue and a vector $\mathbf{v}_0$ corresponding λ_0 is called a polynomial eigenvector, or more precisely, a right polynomial eigenvector of $P(\lambda)$ to distinguish it from a left polynomial eigenvector, which is a vector corresponding λ_0 of the related equation

$$\mathbf{v}^T P(\lambda) = 0^T.$$

In this work, we extend the existence theory of solvent of the quadratic matrix equation which suggested by Xu and Lu [12] to the matrix polynomial (1) and improve Newton step using Schur decomposition. Also we show how to incorporate exact line searches into Newton's method for solving matrix polynomials.

2. THEORY

We deal primarily with the monic case for theoretical progress. First, we introduce the conditions of existence of solvents by spectral data of a given the equation $P(\lambda)$. The next result show how solvents can be constructed from eigenpairs of the polynomial eigenvalue problems.

Theorem 2.1. [3, Lem. 4.1] If a polynomial eigenvalue problem (2) has n linearly independent eigenvectors, $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n$ corresponding to distinct eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$, then $Q\Lambda Q^{-1}$ is a solvent to the matrix polynomial (1), where $Q = [\mathbf{v}_1, \cdots, \mathbf{v}_n]$ and $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$.

The following theorem gives information on the number of solvents of $P(X)$.

Theorem 2.2. [6, Thm. 2.1] *Suppose $P(\lambda)$ in (2) has p distinct eigenvalues $\{\lambda_i\}_{i=1}^p$, with $n \le p \le mn$, and the corresponding set of p eigenvectors $\{\mathbf{v}_i\}_{i=1}^p$ satisfies the Haar condition. Then there are at least $\binom{p}{n}$ different solvents of $P(X)$, and exactly this many if $p = mn$, which are given by*

$$S = Q\mathrm{diag}(\mu_i)Q^{-1}, \qquad Q = [\mathbf{q}_1, \cdots, \mathbf{q}_n],$$

where the eigenpairs $(\mu_i, \mathbf{q}_i)_{i=1}^n$ are chosen from among the eigenpairs $(\lambda_i, \mathbf{v}_i)_{i=1}^p$ of $P(\lambda)$.

However, the next example shows that even if we can find the spectral data for associated polynomial eigenvalue problem it is not always possible to construct the solvent.

Example 2.3. *Let*

$$P_M(X) \equiv X^2 + X + \begin{bmatrix} -6 & -5 \\ 0 & -6 \end{bmatrix} = 0.$$

Then, $P_M(X) = 0$ has two solvents

$$\begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} -3 & -1 \\ 0 & -3 \end{bmatrix}.$$

But there is only one eigenvector associative with $P_M(\lambda)\mathbf{v} = 0$, namely $(1, 0)^T$ for $\lambda = 2$ or $\lambda = -3$. Thus it is impossible to construct a solvent having the form $U\mathrm{daig}\{2, -3\}U^{-1}$, since the corresponding eigenvectors are not linearly independent.

Xu and Lu [12] provided an existence theorem of solvents to the equation $AX^2 + BX + C = 0$, where $A = I$, B and C are simultaneously triangularizable. We now extend these results to obtain some information of solvents from the restricted matrix polynomials, and reify the Theorem 2.1.

Theorem 2.4. *Let $P(X) = A_0X^m + A_1X^{m-1} + \cdots + A_m$ be a matrix polynomial with $A_p = UR_pU^*$ for $p = 0, 1, \ldots, m$. (U is unitary matrix and R_p's are upper triangular matrices). Then, if for $i, j = 1, \cdots, n$, $s_{i,j}$ is a solution of scalar coefficient polynomial $f_{i,j}(t) = \sum_{p=0}^{m}[R_p]_{ij}t^{m-p}$ and $i > j$ implies that $f_{j,j}(s_{i,i}) \neq 0$, then there exists a solvent S of $P(X) = 0$ having a form of $S = URU^*$ for some upper triangular matrix R.*

Proof. Define

$$N(X) \equiv \sum_{p=0}^{m} R_pX^{m-p}.$$

Then, we can easily verify that R is a solvent of $N(X) = 0$ implies that URU^* is a solvent of $P(X) = 0$. Therefore it is sufficient to prove that there exists a solvent of $N(X) = 0$.

For a associated polynomial eigenvalue problem $N(\lambda)$,

$$\det(N(\lambda)) = \prod_{j=1}^{n} f_{j,j}(\lambda).$$

Thus the set of all solutions of $f_{j,j}(\lambda) = 0$ for $j = 1, \cdots, n$ is also the that of eigenvalues of $N(\lambda)$. For each chosen eigenvalue $\lambda_i = s_{i,i}$ $(i = 1, 2, \cdots, n)$, a right eigenvector $\mathbf{x}_i$ can be obtained by solving the system

$$N(\lambda_i)\mathbf{x}_i = \begin{bmatrix} f_{1,1}(\lambda_i) & f_{1,2}(\lambda_i) & \cdots & \times & \times \\ 0 & f_{2,2}(\lambda_i) & \ddots & \times & \times \\ & 0 & \ddots & & \vdots \\ & & \ddots & f_{n-2,n-1}(\lambda_i) & \times \\ & & & f_{n-1,n-1}(\lambda_i) & \times \\ & & & 0 & f_{n,n}(\lambda_i) \end{bmatrix} \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_i^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Notice that $f_{1,1}(\lambda_i), \cdots, f_{i-1,i-1}(\lambda_i)$ are not zero by the hypothesis.

Since the matrix $N(\lambda_i)$ is upper triangular matrix, we may find a solution of the system by backward substitution. First put $x_k^{(i)} = 0$ for $k = i + 1, \cdots, n$. (It is trivial solution of the k-th linear equation. Next if $k = i$, then i-th linear equation has the form $0 \cdot x_i^{(i)} = 0$. Since we can choose $x_i^{(i)}$ arbitrary, take $x_i^{(i)} = 1$ for convenience. At last if $k < i$, then we can easily get $x_k^{(i)}$ by solving the k-th linear equation, since $f_k(\lambda_i) \neq 0$. Therefore for each $i = 1, 2, \cdots, n$, the eigenvector $\mathbf{x}_i$ has the form

$$\mathbf{x}_i \;=\; [x_1^{(i)} \quad x_2^{(i)} \quad \cdots \quad x_{i-1}^{(i)} \quad 1 \quad 0 \quad \cdots \quad 0]\ ,$$

where $x_k^{(i)}$ $(k = 1, 2, \cdots, i-1)$ is the solution of the linear equations such that

$$f_{k,k}(\lambda_i) \cdot x_k^{(i)} = -(f_{k,k+1}(\lambda_i) \cdot x_{k+1}^{(i)} + \cdots f_{k,n}(\lambda_i) \cdot x_n^{(i)}).$$

The set of eigenvectors $\mathbf{x}_i$ $(i = 1, \cdots, n)$ chosen by above method is linearly independent and that makes us possible to construct a nonsingular vector matrix such that

$$Q = \left[\begin{array}{c|c|c|c|c} & & & & \\ \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_n \\ & & & & \end{array}\right] = \left[\begin{array}{c|c|c|c|c} 1 & \times & \times & \cdots & \times \\ & 1 & \times & \cdots & \times \\ & & 1 & \ddots & \times \\ & & & \ddots & \vdots \\ & & & & 1 \end{array}\right].$$

By the Theorem 2.1, the upper triangular matrix R which is a solvent of $N(X) = 0$ can be obtained by

$$R = Q\,\mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_n\}\,Q^{-1}.$$

□

The next corollary is directly induced.

Corollary 2.5. *Let a matrix polynomial of which the coefficient matrices are all upper triangular (or lower triangular) be given and the additional hypothesis of Theorem 2.4 hold. Then there exists a solvent.*

Now our attention may be moved to the numerical methods for solving matrix polynomials.

3. Newton's Method

Newton's method is the most well-known and powerful numerical method for solving nonlinear equations. For this reason it is a natural approach that we apply Newton's method to solve the matrix polynomial (1). Newton's method for solving matrix polynomials was introduced by Kratz and Stickel [7]. In this section we will suggest an improved Newton step computationally using the Schur decomposition and show how to incorporate exact line searches when solving matrix polynomials by Newton's method.

We now construct the iterative method which has second order converges. Consider the nonlinear matrix equations

$$G(X) = 0, \tag{3}$$

where $G : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$. Define $H_k \in \mathbb{C}^{n\times n}$ as the solution of the linear equation $G(X_k) + G'(X_k)H_k = 0$, where the linear operator $G'(X)H : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is the Fréchet derivative of G at X in the direction H. Then Newton's method for the nonlinear matrix equation (3) can be defined by

$$X_0 \quad \text{given},$$
$$\left.\begin{array}{c} G(X_k) + G'(X_k)H_k = 0 \\ X_{k+1} = X_k + H_k \end{array}\right\}, \quad k = 0, 1, 2, \cdots .$$

Thus, each step of Newton's method involves finding the solution $H \in \mathbb{C}^{n\times n}$ of linear equation

$$G'(X)H = -G(X). \tag{4}$$

For the matrix polynomial $P(X) = 0$ the equation (4) can be explicitly represented by

$$P'(X)H = D_X(H) = \sum_{i=1}^{m} \left\{ \left(\sum_{j=0}^{m-i} A_j X^{m-(j+i)} \right) H X^{i-1} \right\} = -P(X). \tag{5}$$

We now have a natural question: when is the Fréchet derivative D_X regular, both at a solvent and an iterate, so that (5) has a unique solution? Unfortunately we do not have remarkable condition. Kratz [7] answered the questions in specific case and presented the Newton method to find solvents of the matrix polynomial (1) and proved that the algorithm converges quadratically near a simple solvent.

Finding the solvents of the matrix polynomials by Newton's method can be regarded as solving the linear equations (5). The general approach solving (5) is using vec function and Kroneckor product.

Using vec operator $\text{vec}(D_X(H))$ is represented by

$$\begin{aligned} \text{vec}(D_X(H)) &= \text{vec}(B_1 H + \cdots + B_m H X^{m-1}) \\ &= \text{vec}(B_1 H I) + \cdots + \text{vec}(B_m H X^{m-1}) \\ &= I \otimes B_1 \text{vec}(H) + \cdots (X^T)^{m-1} \otimes B_m \text{vec}(H) \\ &= (I \otimes B_1 + \cdots (X^T)^{m-1} \otimes B_m)\text{vec}(H) \\ &= \left(\sum_{i=1}^{m} (X^T)^{i-1} \otimes B_i \right) \text{vec}(H) \\ &= \left\{ \sum_{i=1}^{m} \left((X^T)^{i-1} \otimes \sum_{j=0}^{m-i} A_j X^{m-(j+i)} \right) \right\} \text{vec}(H) \\ &= \mathcal{D}\text{vec}(H) \end{aligned} \tag{6}$$

where

$$B_p = A_0 X^{m-p} + A_1 X^{m-(p+1)} + \cdots + A_p,\ p = 1, 2, \cdots, m, \tag{7}$$

and

$$\mathcal{D} = \sum_{i=1}^{m} \left((X^T)^{i-1} \otimes \sum_{j=0}^{m-i} A_j X^{m-(j+i)} \right).$$

By the equality (6), the linear equation $D_X(H) = -P(X)$ is changed by the $n^2 \times n^2$ linear system which is

$$\mathcal{D}\,\mathrm{vec}(H) = \mathrm{vec}(-P(X)). \tag{8}$$

Note that D_X is regular if and only if $\mathcal{D}$ is a nonsingular matrix, and then $\inf_{\|H\|=1} \|D_X(H)\| = \min_{\|H\|=1} \|\mathcal{D}\,\mathrm{vec}(H)\| > 0$ [7, Lem. 1]. If D_X is regular, then the linear equation $D_X(H) = -P(X)$ has a unique solution H, where $\mathrm{vec}(H)$ solves the $n^2 \times n^2$ linear system $\mathcal{D}\,\mathrm{vec}(H) = \mathrm{vec}(-P(X))$. It seems to be reasonable, however it is nonsense from the viewpoint of numerical analysis. In the Newton step, we need to reduce the system size of the equation (8) to $n \times n$. Here is the useful algorithm using Schur decomposition. Given $X \in \mathbb{C}^{n\times n}$, compute the Schur decomposition of X

$$Q^* X Q = R,$$

where Q is unitary and R is upper triangular. Then, substituting this into (5) transforms the system to

$$D_R(H') = B_1 H' + B_2 H' R + \cdots + B_m H' R^{m-1} = C \tag{9}$$

where $B_i = \sum_{j=0}^{m-i} A_j X^{m-j+i}$, $H' = HQ$ and $C = -P(X)Q$. So, taking the vec function both sides of (9) makes a linear system such that

$$\mathrm{vec}(D_R(H')) = \tilde{\mathbf{D}}\mathrm{vec}(H) \tag{10}$$

where the matrix $\tilde{\mathbf{D}} \in \mathbb{C}^{n^2\times n^2}$ is given by

$$\tilde{\mathbf{D}} = \sum_{i=1}^{m} ((R^T)^{i-1} \otimes B_i). \tag{11}$$

If we define $\tilde{D}_{ij} = \sum_{k=1}^{m} [R^{k-1}]_{ji} B_k$, then $\tilde{\mathbf{D}}$ in (11) is represented by

$$\tilde{\mathbf{D}} = \begin{pmatrix} \widetilde{D}_{11} & & & \\ \widetilde{D}_{21} & \widetilde{D}_{22} & & \\ \vdots & & \ddots & \\ \widetilde{D}_{n1} & \cdots & \cdots & \widetilde{D}_{nn} \end{pmatrix}.$$

Since $\tilde{\mathbf{D}}$ is a block lower triangular matrix, using the block forward substitution, the equation (10) is changed to n linear systems with size $n \times n$ such that

$$\begin{aligned}
\mathbf{h}'_1 &= \widetilde{D}_{11}^{-1}\mathbf{c}_1 \\
\mathbf{h}'_2 &= \widetilde{D}_{22}^{-1}(\mathbf{c}_2 - \widetilde{D}_{21}\mathbf{h}'_1) \\
&\ \vdots \\
\mathbf{h}'_n &= \widetilde{D}_{nn}^{-1}(\mathbf{c}_2 - \widetilde{D}_{n1}\mathbf{h}'_1 - \cdots - \widetilde{D}_{n,n-1}\mathbf{h}'_{n-1}),
\end{aligned}$$

where $\mathbf{h}'_i$ and $\mathbf{c}_i$ are i-th columns of H' and C, respectively.

Algorithm 3.1. *Given matrices $R, B_1, B_2, \cdots, B_m$ which is defined by* (7) *and* (9)*and positive integers i, j ($i \le j$ and $i, j \le n$), the following algorithm computes $\tilde{D}_{ij}$*

```
for $r = 1 : m$
    $\tilde{D}_{ij} \leftarrow \tilde{D}_{ij} + R^{r-1}(j,i)B_r$
end
```

Algorithm 3.2 (Solving the equation $D_X(H) = -P(X)$ by Schur decomposition)**.** $A_0, \cdots, A_m, X \in \mathbb{C}^{n\times n}$ *is given and $X = QRQ^*$ is the Schur decomposition of X. This algorithm finds $H \in \mathbb{C}^{n\times n}$ which is the solution of $D_X(H) = -P(X)$ in* (5)

```
$B_m \leftarrow A_0, R_1 \leftarrow I$
for $i = 1 : m-1$
    $B_{m-i} \leftarrow B_{m-i+1}X + A_i$
    $R_{i+1} \leftarrow R_iR$
end
$C \leftarrow -(B_1X + A_m)Q$
for $i = 1 : n$
    $\mathbf{d} \leftarrow 0$
    for $j = 1 : i-1$
        Use algorithm (3.1) to compute $\tilde{D}_{ij}$
        $\mathbf{d} \leftarrow \mathbf{d} + \tilde{D}_{ij}H(:,j)$
    end
    Use algorithm (3.1) to compute $\tilde{D}_{ii}$
    $H(:,i) \leftarrow \tilde{D}_{ii}^{-1}(C(:,i) - \mathbf{d})$
end
$H \leftarrow HQ^*$
```

Let the matrix polynomial (1) have only real coefficient matrices. Although the desired solvent is real above method may required complex arithmetic. We will use the technique of Schur algorithm to desire an algorithm for computation a solvent that use only real arithmetic if the given coefficient matrices are real.

Let $X \in \mathbb{R}^{n\times n}$ be given and $X = Q\widehat{R}Q^T$ be a real Schur decomposition with

$$\widehat{R} = \begin{bmatrix} R_{11} & \cdots & \cdots & R_{1p} \\ & R_{22} & & \vdots \\ & & \ddots & \vdots \\ & & & R_{pp} \end{bmatrix} \tag{12}$$

where R_{ii} is 1×1 or 2×2 matrices. Substituting $X = Q\widehat{R}Q^T$ into (5) transforms the system to

$$D_R(H') = B_1 H' + B_2 H'\widehat{R} + \cdots + B_m H'\widehat{R}^{m-1} = \widehat{C} \tag{13}$$

where $B_i = \sum_{j=0}^{m-i} A_j X^{m-j+i}$, $H' = HQ$, and $\widehat{C} = -P(X)Q$. By taking the vec function both sides of the equation (13), we have the linear system given by

$$\widehat{\mathbf{D}}\mathrm{vec}(\widehat{H}) = \mathrm{vec}(\widehat{C}) \tag{14}$$

with

$$\widehat{\mathbf{D}} = \begin{pmatrix} \widehat{D}_{11} & & & \\ \widehat{D}_{21} & \widehat{D}_{22} & & \\ \vdots & & \ddots & \\ \widehat{D}_{p1} & \cdots & \cdots & \widehat{D}_{pp} \end{pmatrix}$$

where

$$\widehat{D}_{ij} = \begin{cases} (I)_{\Psi(i)} \otimes B_1 + \sum_{k=2}^{m} (R_{ji})^{k-1})^T \otimes B_k & \text{if} \quad i == j \\ \sum_{k=2}^{m} (R_{ji}^{k-1})^T \otimes B_k & \text{else.} \end{cases}$$

and $\Psi(i)$ is the number of columns of block matrix R_{ii} in (12).

By quasi-block-forward substitution, we have p pieces of linear system such that

$$\widehat{D}_{ii}\widehat{\mathbf{h}}_i = \widehat{\mathbf{c}}_i - \sum_{k=1}^{i-1} \widehat{D}_{ik}\widehat{\mathbf{h}}_k, \tag{15}$$

where $\begin{bmatrix} \widehat{\mathbf{h}}_1 \\ \vdots \\ \widehat{\mathbf{h}}_p \end{bmatrix} = \mathrm{vec}(\widehat{H})$, $\begin{bmatrix} \widehat{\mathbf{c}}_1 \\ \vdots \\ \widehat{\mathbf{c}}_p \end{bmatrix} = \mathrm{vec}(\widehat{C})$, and $\widehat{\mathbf{c}}_i$, $\widehat{\mathbf{h}}_i \in \mathbb{C}^{\Psi(i)n}$ for $i = 1, 2, \cdots, p$. Note that $\widehat{D}_{ii}$'s are either $2n\times 2n$ or $n\times n$ matrix, the index of element $[R_{ij}]_{11}$ $(i \le j)$ of $\widehat{R}$ is $\left(\sum_{k=1}^{i-1}\Psi(k)+1, \sum_{k=1}^{j-1}\Psi(k)+1\right)$ and the size of R_{ij} is $\Psi(i)\times\Psi(j)$.

Algorithm 3.3. *Given matrices $\widehat{R}, B_1, B_2, \cdots, B_m$ which is defined by* (13) *and* (7) *and positive integers i, j ($i \le j$ and $i, j \le p$), the following algorithm computes $\widehat{D}_{ij}$*

```
**if** $i == j$
    $\widehat{D}_{ij} \leftarrow I_2 \otimes B_1$ ($I_2$ *is* $2\times 2$ *identity matrix.*)
**else**
    $\widehat{D}_{ij} \leftarrow O_{n\times\Psi(i),\, n\times\Psi(j)}$ ($O_{nm}$ *is* $n\times m$ *zero matrix.*)
```

```
for $r = 2 : m$
  $\widehat{D}_{ij} \leftarrow \widehat{D}_{ij} + (R_{ji}^{r-1})^T \otimes B_r$
end
```

Algorithm 3.4 (Solving the equation $D_X(H) = -P(X)$ by real Schur decomposition)**.** $A_0, \cdots, A_m, X \in \mathbb{R}^{n\times n}$ *is given and* $X = Q\widehat{R}Q^T$ *is the real Schur decomposition of* X *in* (13)*. This algorithm find* $H \in \mathbb{R}^{n\times n}$ *which is the solution of* $D_X(H) = -P(X)$ *in* (5)

```
$B_m = A_0$, $\widehat{R}_0 = I$
for $i = 1 : m - 1$
  $B_{m-i} \leftarrow B_{m-i+1}X + A_i$
  $\widehat{R}_i \leftarrow \widehat{R}_{j-1}\widehat{R}$
end
$\widehat{C} \leftarrow -(B_1 X + A_m)Q$
$pos \leftarrow 0, i \leftarrow 1$
while $pos < n$
  if $\widehat{R}(pos + 2, pos + 1) == 0$ then $\Psi(i) \leftarrow 1$
  else $\Psi(i) \leftarrow 2$
  end
  $pos \leftarrow \Psi(i) + pos$
  $i \leftarrow i + 1$
end
$pos \leftarrow 1$
for $i = 1 : p$ ($p$ is the maximum of domain of $\Psi$.)
  $\mathbf{d} \leftarrow O_{n\times\Psi(i),1}$
  for $j = 1 : i - 1$
    Use algorithm (3.3) to compute $\widehat{D}_{ij}$
    $\mathbf{d} \leftarrow \mathbf{d} + \widehat{D}_{ij}$
  end
  Use algorithm (3.3) to compute $\widehat{D}_{ii}$
  $\widehat{\mathbf{h}}_i \leftarrow \widehat{D}_{ii}^{-1}$ $[\text{vec}\{\text{C}(:, \text{pos} : \text{pos} + \Psi(\text{i}) - 1)\} - \mathbf{d}]$
  if $\Psi_{(i)} == 1$
    $H(:, pos) \leftarrow \widehat{\mathbf{h}}_i$
  else
    $H(:, pos) \leftarrow \widehat{\mathbf{h}}_i(1 : n, 1)$
    $H(:, pos + 1) \leftarrow \widehat{\mathbf{h}}_i(n + 1, 2n, 1)$
  end
  $pos \leftarrow pos + \Psi(i)$
end
$H = HQ^T$
```

We now show how to incorporate exact line searches when solving a matrix polynomial (1) by Newton's method. For implementation we need to find the exact expansion of $P(X + tH)$.

Let μ, ν be the permutations of $1, 2, \cdots, n$, and integer $m(1 \le m \le n)$ be given. Then the relation $\mu \sim_m \nu$ given by $\mu(i) > m$ if and only if $\nu(i) > m$ and $\mu(i) \le m$ if and only if $\nu(i) \le m$ for all $i = 1, 2, \cdots, n$ is an equivalent relation. So we can define the equivalent class of permutations μ's by $[\mu]_m$. Note that the number of $[\mu]_m$ is $n!/m!(n-m)!$. Let $r_1 = r_2 = \cdots = r_m = X \in \mathbb{C}^{n\times n}$, $r_{m+1} = r_{m+2} = \cdots = r_n = Y \in \mathbb{C}^{n\times n}$ then the function $\Phi_{X,Y} : \mathbb{N} \times \mathbb{N} \to \mathbb{C}^{n\times n}$ can be defined by

(1) $\Phi_{X,Y}[0,0] \equiv I$,
(2) $\Phi_{X,Y}[m, n-m] \equiv \sum_{[\mu]_m} r_{[\mu]_m(1)} r_{[\mu]_m(2)} \cdots r_{[\mu]_m(n)}$.

The function $\Phi_{X,Y}$ is the sum of the products of all repeated permutations of X and Y. By using the notation of Φ, we can describe the expansions of $(X+tH)^k$ for $k = 0, 1, 2, \cdots, m$

$$\begin{aligned} A_m &= A_m \Phi_{XH}[0,0], \\ A_{m-1}(X+tH) &= A_{m-1}X + tA_{m-1}H, \\ &= A_{m-1}\Phi_{X,H}[1,0] + tA_{m-1}\Phi_{X,H}[0,1], \\ A_{m-2}(X+tH)^2 &= A_{m-2}X^2 + A_{m-2}(XH+HX) + t^2A_{m-2}H^2 \\ &= A_{m-2}\Phi_{X,H}[2,0] + tA_{m-2}\Phi_{X,H}[1,1] + t^2A_{m-2}\Phi_{X,H}[0,2], \\ &\vdots \\ A_{m-k}(X+tH)^m &= A_{m-k}\Phi_{X,H}[m,0] + tA_{m-k}\Phi_{X,H}[m-1,1] + \cdots + t^m\Phi_{X,H}[0,m]. \end{aligned}$$

Therefore we obtain the expansion formula

$$(X+tH)^k = \sum_{i=0}^{k} t^i \Phi_{X,H}[k-i, i] \tag{16}$$

and easily verify

$$(\Phi_{X,H}[i,j])^* = \Phi_{X^*,Y^*}[i,j].$$

By using the formula (16) and equality $D_X(H) + P(X) = 0$, $P(X+tH)$ can be represented by

$$\begin{aligned} P(X+tH) &= \sum_{i=0}^{m}\sum_{j=0}^{i} t^j \Phi_{X,H}[i-j, j] \\ &= P(X) + tD_X(H) + \sum_{i=2}^{m}\sum_{j=2}^{i} t^j \Phi_{X,H}[i-j,j] \\ &= (1-t)P(X) + \sum_{i=2}^{m}\sum_{j=2}^{i} t^j \Phi_{X,H}[i-j,j]. \end{aligned}$$

Finally the merit function $p(t)$ can be obtained by

$$\begin{aligned}
p(t) &= ||P(X+tH)||_2^F \\
&= \operatorname{trace}\left((1-t)P(X)^* + \sum_{i=2}^{m}\sum_{j=2}^{i}\Phi_{X^*,Y^*}[i-j,j]\right) \\
&\qquad\qquad \left((1-t)P(X) + \sum_{i=2}^{m}\sum_{j=2}^{i}\Phi_{X,Y}[i-j,j]\right) \\
&= (1-t)^2||P(X)||_F^2 + \cdots + t^{2m}\,\Phi_{X^*,Y^*}[0,m]\,\Phi_{X,Y}[0,m] \\
&= (1-t)^2||P(X)||_F^2 + \cdots + t^{2m}||H^m||_F^2.
\end{aligned}$$

Since $p'(0) = -2\|P(X)\|_F^2 < 0$ and leading coefficient of $p'(t)$ is positive, p' has a real zero in positive real number, and this zero corresponds to a minimum or a point of inflection of p. If $p(t)$ has the minimum for large t, then the Newton step with line search may have numerically a bad effect on convergence. So we must restrict our attention to the appropriate interval $(0, k]$. Because $t = 1$ corresponds to a pure Newton step, k must be greater than 1. Thus we define t by

$$p(t) = \min_{x\in(0,k]} p(x) \text{ with } k > 1.$$

Roughly speaking, exact line searches means finding scalar t that makes $X + tH$ closer point in direction of H from solvent.

4. Numerical Experiments and Conclusions

In this section we show and compare some experimental results using Newton's method with and without line searches. Our experiments were done in MATLAB. Iterations for Newton's method with and without exact line searches are terminated when the residual $P(X_k)$ is of the same order of magnitude as the round error in computing it, namely when the relative residual $\rho(X_k)$ satisfies

$$\rho(X_k) = \frac{||fl(P(X_k))||_F}{||A_0||_F||X_k||_F^m + \cdots + ||A_m||_F} \leq nu, \tag{17}$$

where $u = 2^{-53} \simeq 1.1 \times 10^{-16}$ is unit round off. MATLAB codes for Newton's method has an option to choose whether to use exact line searches. Suggested examples are in Kratz [7] solved by pure Newton's method. We will compare the results with exact line searches.

Consider a matrix equation

$$P_1(X) = X^3 + \begin{bmatrix} -6 & 6 \\ -3 & -15 \end{bmatrix} X^2 + \begin{bmatrix} 2 & -42 \\ 21 & 65 \end{bmatrix} X + \begin{bmatrix} 18 & -66 \\ 33 & 81 \end{bmatrix} = 0. \tag{18}$$

We obtained two solvents such that

$$S_1 = \begin{bmatrix} 4 & -2 \\ 1 & 7 \end{bmatrix}, \qquad S_2 = \begin{bmatrix} 0 & -2 \\ 1 & 3 \end{bmatrix}.$$

with starting matrices $218I_2$ and $-218I_2$, respectively. Figure 1 shows Newton's method with exact line searches converges faster than Newton's method without exact line searches for the two starting matrices.

Here, we consider the matrix differential equation

$$\mathbf{y}^{(4)} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{y}^{(2)} + \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ -1 & 0 & 1 \end{bmatrix} \mathbf{y}' + \begin{bmatrix} -20 & 2 & 1 \\ 2 & -20 & 0 \\ 1 & 0 & -20 \end{bmatrix} \mathbf{y} = 0.$$

Such equations occur in connection with vibrating systems. The characteristic polynomial is given by

$$P_2(X) = X^4 + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} X^2 + \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ -1 & 0 & 1 \end{bmatrix} X + \begin{bmatrix} -20 & 2 & 1 \\ 2 & -20 & 0 \\ 1 & 0 & -20 \end{bmatrix} = 0. \tag{19}$$

Figure 2 shows the convergence of Newton's method with and without exact line searches in $P_2(X) = 0$ with starting matrices $24I$ and $-24I$.

Next, we choose starting matrices

$$X_0 = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{pmatrix}, \qquad -100 \le x_1, \cdots, x_6 \le 100$$

with an equally spaces grid of 100 random points $(x_1, \cdots, x_9)$. Figure 3 shows how many times a solvent is produced within 30, 50, and 100 iterations with and without exact line searches. For Newton's method with and without exact line searches, exact line searches give

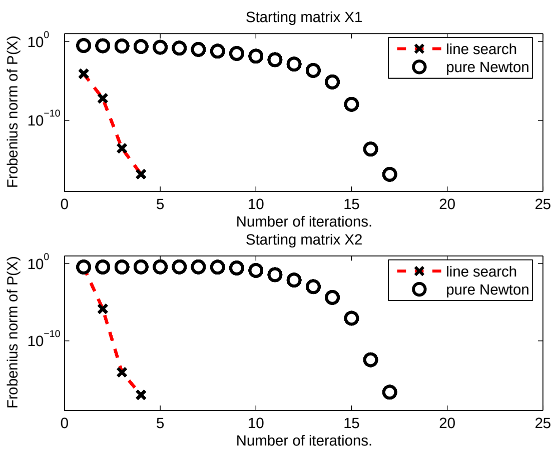

FIGURE 1. Convergence for example (18) with and without Newton's method.

the result in more frequent convergence. Convergence of Newton's methods with and without exact line searches is obtained to four different solvents depending on the starting matrix.

The matrix polynomial in (1) arises in some applications, for example, the stochastic problems and the polynomial eigenvalue problems. Specially, the polynomial eigenvalue problems can be solved by the $n \times n$ standard eigenvalue problems, if we can find a solvent of associated matrix polynomials.

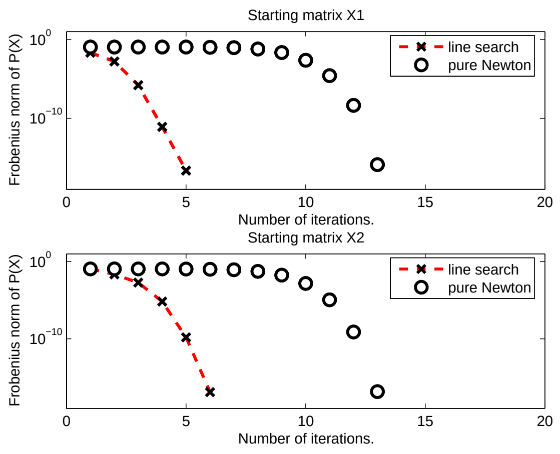

FIGURE 2. Convergence for example (19) with and without Newton's method.

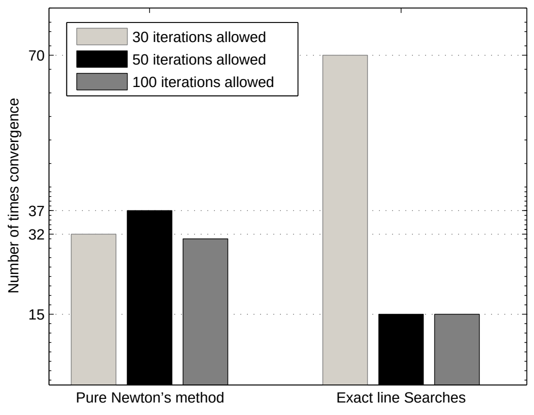

FIGURE 3. Number of times convergence obtained for problem (19) with 100 different arbitrary starting matrices.

We described existence theorems for solvents of the matrix polynomials (1) using spectral theory for the polynomial eigenvalue problems. And we had constructed the solvents for some special matrix polynomial. We also derived Newton's method and improved this algorithm by Schur decomposition and incorporated exact line searches into Newton's method. Finally, we experimented with some examples. Newton's method with exact line searches frequently reduced the number of iterations.

REFERENCES

[1] George J. Davis, *Numerical solution of a quadratic matrix equation*, SIAM J. Sci. Stat. Comput., **2** (1981), 164–175.

[2] George J. Davis, *An alogrithm to compute solvents of the matrix equation* $AX^2 + BX + C = 0$, ACM Trans. Math. Software, **9** (1983), 246–254.

[3] J. E. Dennis, Jr., J. F. Traub, and R. P. Weber, *The algebraic theory of matrix polynomials*, Numerical Algorithms, **13** (1976), 831–845.

[4] J. E. Dennis, Jr. and Robert B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.

[5] N. J. Higham and H.-M. Kim, *Numerical analysis of a quadratic matrix equation*, IMA J. Numer. Anal., **20** (2000), 499–519.

[6] Nicholas J. Higham and Hyun-Min Kim, *Solving a quadratic matrix equation by Newton's method with exact line searches*, SIAM J. Matrix Anal. Appl., **23** (2001), 303–316.

[7] W. Kratz and E. Stickel, *Numerical solution of matrix polynomial equations by Newton's method*, IMA J. Numer. Anal., **7** (1987), 355–369.

[8] P. Lancaster, I. Gohberg and L. Rodman, *Matrix Polynomials*, Academic press, New York, 1982.

[9] P. Lancaster and M. Tismenetsky, *The Theory of Matrices with Applications, 2nd ed.*, Academic press, New York, 1985.

[10] Peter Lancaster, *Lambda-matrices and Vibrating Systems*, Pergamon Press, Oxford, 1996.

[11] Guy Latouche and V.Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, Society for Industrial and Applied Mathematics, 1999.

[12] Hong guo Xu and Lin zhang Lu, *Properties of a quadratic matrix equation and the solution of the continuous-time algebraic Riccati equation*, Linear Algebra Appl., **222** (1995), 127–145.