ALGORITHMS FOR SOLVENTS OF MATRIX POLYNOMIALS*

J. E. DENNIS, JR.,† J. F. TRAUB‡ AND R. P. WEBER¶

Abstract. In an earlier paper we developed the algebraic theory of matrix polynomials. Here we introduce two algorithms for computing "dominant" solvents. Global convergence of the algorithms under certain conditions is established.

1. Introduction. Let

\[ M(X) = X^m + A_1 X^{m-1} + \cdots + A_m, \]

where \( A_1, \cdots, A_m, X \) are \( n \times n \) complex matrices. We call \( M(X) \) a monic matrix polynomial of degree \( m \). A matrix \( S \) for which \( M(S) = 0 \) is called a right solvent (or briefly a solvent). In Dennis, Traub, and Weber [2] we studied the algebraic theory of matrix polynomials. In this paper we analyze two algorithms for calculating a "dominant" solvent. (Dominant solvent is a generalization of largest zero of a scalar polynomial. See Definition 2.1.)

Algorithm 1 of this paper is a generalization of an algorithm for scalar polynomials (Traub [5]). It is globally convergent in the following sense. If Stage 1 is done sufficiently long and if the hypotheses of Theorem 2.1 hold, then the iteration of Stage 2 is globally convergent. Stage One may be viewed as direct powering by a "block companion matrix". Algorithm 2 is a generalization of Bernoulli's algorithm. As in the scalar case, Bernoulli iteration may converge very slowly.

In Dennis, Traub, and Weber [1] the relation between "block eigenvalue" and solvent is explored and two algorithms for the calculation of "block eigenvectors" are given. We do not pursue this here.

\( M(\lambda I) \) is called a lambda-matrix and a scalar \( \lambda \) such that \( M(\lambda I) \) is singular is called a latent root. An application of solvents is to the calculation of latent roots. If all the latent roots are distinct, and if the hypotheses of Theorem 2.1 are satisfied, then the dominant solvent may be computed. The dominant solvent may be removed and Algorithm 1 or 2 applied to the deflated polynomial. If the successive deflated polynomials have dominant solvents the process can be repeated until all the latent roots have been computed. The stability of this process has not been investigated. Other proposed methods for the calculation of latent roots such as algorithms of Lancaster [4] and Kublanovskaya [3] are only locally convergent and do not have an associated method of deflation. See Dennis, Traub, and Weber [1, Appendix B] for additional material on algorithms for lambda-matrices.

In Dennis, Traub, and Weber [1] we give two globally convergent algorithms for calculating dominant latent roots. These results and their extensions will be reported in a future paper where we will also show how systems of polynomial equations may be solved using lambda-matrices.

We assume the reader is familiar with the notation and results of Dennis, Traub, and Weber [2]. For the reader's convenience, we state a number of definitions and results from the above cited paper crucial to this paper.

---

* Received by the editors November 12, 1975, and in final revised form July 24, 1977.
† Computer Science Department, Cornell University, Ithaca, New York 14853. This research was supported in part by the National Science Foundation under Grant GJ27528.
‡ Department of Computer Science, Carnegie—Mellon University, Pittsburgh, Pennsylvania 15213. This research was supported in part by the National Science Foundation under Grant GJ32111 and the Office of Naval Research under Contract N0014-67-A-0314-0010, NR 044-422.
¶ Bell Laboratories, Holmdel, New Jersey 07733.
If $S_1, \ldots, S_m$ are any $n$ by $n$ matrices the block Vandermonde matrix is defined by

$$
V(S_1, S_2, \ldots, S_m) = \begin{pmatrix}
I & I & \cdots & I \\
S_1 & S_2 & \cdots & S_m \\
\vdots & \vdots & \ddots & \vdots \\
S_1^{m-1} & S_2^{m-1} & \cdots & S_m^{m-1}
\end{pmatrix}.
$$

If $S_1, \ldots, S_m$ are $n$ by $n$ matrices the fundamental matrix polynomials are a set of $m - 1$ degree matrix polynomials, $M_1, \ldots, M_m$, such that $M_i(S_j) = \delta_{ij} I$. Let

$$
M_i(X) = A_1^{(i)} X^{m-1} + \cdots + A_m^{(i)}.
$$

**Theorem 1.1.** If matrices $S_1, \ldots, S_m$ are such that $V(S_1, \ldots, S_m)$ is nonsingular, then there exist unique matrix polynomials $M_1, \ldots, M_m$ which are fundamental matrix polynomials. If, furthermore, $V(S_1, \ldots, S_{k-1}, S_{k+1}, \ldots, S_m)$ is nonsingular, then $A_k^{(k)}$ is nonsingular.

**Theorem 1.2.** If matrices $S_1, \ldots, S_m$ are such that $V(S_1, \ldots, S_m)$ is nonsingular and $M_1(X), \ldots, M_m(X)$ are a set of fundamental matrix polynomials, then an arbitrary matrix polynomial $G(X)$ with degree not exceeding $m - 1$ can be written as

$$
G(X) = \sum_{i=1}^{m} G(S_i) M_i(X).
$$

We summarize the remainder of this paper. In §§ 2 and 3 we state two algorithms for calculating solvents of matrix polynomials and prove global convergence of the algorithms for "dominant" solvents. Numerical examples are provided in the last section.

2. **A matrix polynomial algorithm.** We show that a generalization of Traub's scalar polynomial algorithm (Traub [5]) may be used to calculate a dominant solvent of the matrix polynomial problem. A dominant left solvent may also be computed (Dennis, Traub, and Weber [1, Corollary 5.1]).

**Algorithm 1.** Stage 1. Let $G_0(X) = I$ and define matrix polynomials $G_n(X)$ by

$$
G_{n+1}(X) = G_n(X)X - \Gamma_1^{(n)} M(X),
$$

for $n = 0, 1, \ldots, L - 1$, where

$$
G_n(X) = \Gamma_1^{(n)} X^{m-1} + \cdots + \Gamma_m^{(n)}.
$$

Stage 2. Let $X_0 = (\Gamma_1^{(L)}) (\Gamma_1^{(L-1)})^{-1}$ and define matrices $X_i$ by

$$
X_{i+1} = G_L(X_i) G_{L-1}(X_i) = \phi_i(X_i).
$$

**Note.** With $G_0(X) = I$, we find $G_{m-1}(X) = X^{m-1}$. We could of course take $G_0(X) = X^{m-1}$ and this is what we do in our programs. The choice $G_0(X) = I$ makes the proof of Theorem 2.1 slightly simpler.

Since $G_n(X)X$ is a shift, each step of Stage 1 can be performed with $m$ ($n$ by $n$) matrix multiplications and $m-1$ ($n$ by $n$) matrix additions. Each step of Stage 2 requires the solution of the $n$ by $n$ matrix equation $X_{i+1} G_{L-1}(X_i) = G_L(X_i)$.

Before proving convergence of Algorithm 1 we state a basic definition and prove a useful Lemma.

**Definition 2.1.** Matrix $A$ dominates matrix $B$ if all the eigenvalues of $A$ are greater, in modulus, than those of $B$. In particular, if the solvent $S_1$ dominates the
solvents $S_2, \ldots, S_n$ we say $S_1$ is a dominant solvent. (Note that a dominant solvent cannot be singular.)

**Lemma 2.1.** If matrix $A$ dominates matrix $B$, then $\lim_{n \to \infty} A^{-n} CB^n = 0$, for any constant matrix $C$.

**Proof.** For any $\epsilon > 0$, let

$$B = P_B(\epsilon) J_B(\epsilon) P_B(\epsilon)^{-1},$$

where $J_B(\epsilon)$ denotes a block diagonal matrix whose blocks are of the form

$$
\begin{pmatrix}
\lambda_B \\
\epsilon & \lambda_B \\
\epsilon & \lambda_B
\end{pmatrix}
$$

and where $\lambda_B$ denotes an eigenvalue of $B$. Then,

$$\|B^n\| \leq \|P_B(\epsilon)\| \|P_B(\epsilon)^{-1}\| (\epsilon + \max |\lambda_B|)^n,$$

using the infinity norm.

Letting

$$A^{-1} = P_A^{-1}(\epsilon) J_A^{-1}(\epsilon) P_A^{-1}(\epsilon)^{-1},$$

we similarly obtain

$$\|A^{-n}\| \leq \|P_A^{-1}(\epsilon)\| \|P_A^{-1}(\epsilon)^{-1}\| (\epsilon + \max |\lambda_A|)^n$$

$$= \|P_A^{-1}(\epsilon)\| \|P_A^{-1}(\epsilon)^{-1}\| \left( \epsilon + \frac{1}{\min |\lambda_A|} \right)^n.$$

Combining equations (2.4) and (2.5) we get

$$\|A^{-n} CB^n\| \leq k \left[ (\epsilon + \max |\lambda_B|) \left( \epsilon + \frac{1}{\min |\lambda_A|} \right) \right]^n,$$

where $k$, a function of $\epsilon$, is independent of $n$.

When $\epsilon = 0$, the constant to the $n$th power is less than one, since $\max |\lambda_B|/\min |\lambda_A| < 1$. By continuity, there exists an $\epsilon > 0$ so that the constant is still less than unity and the result follows. \( \square \)

We now state and prove the convergence theorem for Algorithm 1. Let

$$\tilde{G}_n(X) = (\Gamma^{(n)}_1)^{-1} G_n(X),$$

$$\tilde{M}_l(X) = (A^{(l)}_1)^{-1} M_l(X)$$

be monic matrix polynomials. Then we have the following theorem:

**Theorem 2.1.** If $M(X)$ is a matrix polynomial of degree $m$ such that

- (i) it has solvents $S_1, \ldots, S_m$,
- (ii) $S_1$ is a dominant solvent,
- (iii) $V(S_1, \ldots, S_m)$ and $V(S_2, \ldots, S_m)$ are nonsingular,

then

- (i) $\lim_{n \to \infty} \tilde{G}_n(X) = \tilde{M}_1(X),$
- (ii) for $L$ sufficiently large,

$$\lim_{i \to \infty} X_i = S_1.$$
Proof of part (i). From (2.1)

(2.7) \[ G_n(S_i) = S_i^n. \]

By Theorem 1.2 and (2.7)

(2.8) \[ G_n(X) = \sum_{i=1}^{m} G_n(S_i)M_i(X) = \sum_{i=1}^{m} S_i^nM_i(X), \]

and, thus,

(2.9) \[ \Gamma_1^{(n)} = \sum_{i=1}^{m} S_i^nA_1^{(i)}. \]

By Theorem 1.1, \( A_1^{(1)} \) is nonsingular. Since \( S_1 \) is also nonsingular there is an \( N \) such that for \( n \geq N \), \( \Gamma_1^{(n)} \) must be nonsingular, since by use of Lemma 2.1 and (2.9),

\[ \lim_{n \to \infty} \Gamma_1^{(n)}(S_1^nA_1^{(1)})^{-1} = I. \]

From (2.8), (2.9) and Lemma 2.1, we get, for \( n \geq N \),

\[ \tilde{G}_n(X) = \left( \sum_{i=1}^{m} S_i^nA_1^{(i)} \right)^{-1} S_1^n \left( \sum_{i=1}^{m} S_i^nM_i(X) \right) \]

\[ = \left( \sum_{i=1}^{m} S_1^{-1} S_i^n A_1^{(i)} \right)^{-1} \left( \sum_{i=1}^{m} S_1^{-1} S_i^nM_i(X) \right), \]

and the conclusion (i) follows by application of Lemma 2.1. \( \square \)

We defer the proof of part (ii) of the theorem to first obtain some lemmas needed in the proof. We assume the hypotheses of Theorem 2.1 hold.

In Lemma 2.2 we show that every right solvent is a fixed point of \( \phi_L(X) \) for each \( L \). Lemma 2.4 shows that \( \phi_L(X) \) is defined for all \( X \) in some neighborhood of the dominant solvent. Lemma 2.6 gives the local convergence of the second stage of Algorithm 1. Finally, Lemma 2.7 says that stage one will yield a point in the locally convergent region (Lemma 2.6) of the dominant solvent. Stage one supplies a sufficiently accurate starting value for the locally convergent stage two and, hence, the overall algorithm is globally convergent. The proof of part (ii) of Theorem 2.1 then immediately follows.

LEMMA 2.2. \( \phi_L(S) = S \) for all \( L \) and any nonsingular right solvent \( S \).

Proof. The result follows from (2.7).

LEMMA 2.3. There exists a nontrivial compact ball \( B \), centered at \( S_1 \), such that for all \( X \in B \)

(i) \( \| I - M_1(X) \| \leq K < 1, \)

and

(ii) \( \| M_f(X) \| \leq 1, j \neq 1. \)

Proof. A matrix polynomial is a continuous function of its matrix variable. The results thus follow from continuity and the facts that \( M_1(S_1) = I \) and \( M_f(S_1) = 0 \) for \( j \neq 1. \) \( \square \)

It follows from Lemma 2.3 that for all \( X \in B, M_1(X) \) is nonsingular and

\[ \| M_1^{-1}(X) \| \leq \frac{1}{1 - \| I - M_1(X) \|}. \]

LEMMA 2.4. If \( X \in B \), then there exists an \( L' \) such that \( \phi_L(X) \) is defined for every \( L \geq L' \).
Proof. For $X \in B$, let 

$$V_j(X) = M_j(X)M_1^{-1}(X), \quad W_L(X) = \sum_{j=2}^{m} S^{-L} S^L_j V_j(X).$$

Then,

$$G_{L-1}(X) = \sum_{j=1}^{m} S^L_j M_j(X)$$

(2.10) 

$$= S^{-L} \left( I + \sum_{j=2}^{m} S^{-1} S_j^{L-1} V_j(X) \right) M_1(X)$$

$$= S^{-L} (I + W_{L-1}(X)) M_1(X).$$

Note that $\lim_{L \to \infty} W_L(X) = 0$ uniformly for $X \in B$ since

$$\|V_j(X)\| = \|M_j(X)M_1^{-1}(X)\| \leq \frac{1}{1-K} < \infty$$

by Lemma 2.3. Thus, $I + W_L(X)$ is invertible for large $L$. By (2.10), $G_{L-1}(X)$ is invertible for large $L$ and the result follows. \(\square\)

**Lemma 2.5.** If $X \in B$, then

(2.11) 

$$\|S^L_j V_j(X)S_1^{-L}\| \leq \tau \sigma^L \|M_j(X)M_1^{-1}(X)\| \leq \frac{\tau \sigma^L}{1-K},$$

where $0 \leq \sigma < 1$, and $\tau$ is a constant independent of $L$ and $X$.

**Proof.** The result follows from (2.6), where

$$\sigma = (\varepsilon + \max |\lambda_{S_j}|) \left( \varepsilon + \frac{1}{\min |\lambda_{S_1}|} \right)$$

for $j \neq 1$, and $\varepsilon > 0$ such that $\sigma < 1$. \(\square\)

**Lemma 2.6.** If $X_0 \in B$ and $L$ is sufficiently large, then

$$\lim_{i \to \infty} X_i = S_1.$$

**Proof.** Let $X \in B$ and $L \equiv L'$ as in Lemma 2.4. Set

$$E_L(X) = \phi_L(X) - S_1.$$

Then, since

$$\phi_L(X) = G_L(X)G_{L-1}^{-1}(X)$$

$$= \left( \sum_{j=1}^{m} S^L_j V_j(X) \right) \left( \sum_{j=1}^{m} S_j^{L-1} V_j(X) \right)^{-1},$$

it follows that

$$E_L(X) \sum_{j=1}^{m} S_j^{L-1} V_j(X) = \sum_{j=2}^{m} (S_j - S_1) S_j^{L-1} V_j(X).$$

Let

$$T_{j,L}(X) = S_j^{L-1} V_j(X)S_1^{-(L-1)}.$$
Thus, by Lemma 2.5,
\[ \|T_{j,L}(X)\| \leq \frac{\sigma^{L-1}}{1-K}. \]

Choose \( L \) large enough so that
\[ \sum_{j=2}^{m} \|T_{j,L}(X)\| \leq F < 1 \]
for all \( X \in B \). Then,
\[ E_L(X)\left[ I + \sum_{j=2}^{m} T_{j,L}(X) \right] = \sum_{j=2}^{m} (S_j - S_1)T_{j,L}(X) \]
gives, by (2.11),
\[ \|E_L(X)\| \leq \sum_{j=2}^{m} \|S_j - S_1\| \sigma^{L-1} \|M_j(X)\| \|M_1^{-1}(X)\| \frac{1-F}{1-F}, \]
for all \( X \in B \). A matrix polynomial is continuously differentiable. Since \( M_j(S_1) = 0 \) for \( j \neq 1 \), we have
\[ \|M_j(X)\| \leq t\|X - S_1\|, \]
where \( t \) is independent of \( X \) and \( S_1 \), and \( j \neq 1 \). Finally,
\[ \|\phi_L(X) - S_1\| \leq c\sigma^{L-1}\|X - S_1\| \]
for all \( X \in B \), where
\[ c = \frac{\sum_{j=2}^{m} \|S_j - S_1\| p_t}{(1-F)(1-K)}. \]

The result follows from (2.12), since \( 0 \leq \sigma < 1 \) and \( L \) can be taken large enough so that \( c\sigma^{L-1} < 1 \). □

The preceding lemma gave convergence for the second stage of Algorithm 1 if \( X_0 \in B \). The next lemma shows that \( X_0 \) is in \( B \) if the first stage is continued long enough.

**Lemma 2.7.** For \( L \) sufficiently large, \( (I'IL)(I'IL)^{-1} \in B \).

**Proof.** We note that \( I^{(L)} = \sum_{j=1}^{m} S_j^{(L)}A_j^{(L)} \); a proof similar to that in Lemma 2.6 yields
\[ \lim_{L \to \infty} (I^{(L)})(I^{(L)})^{-1} = S_1. \]
□

The second part of Theorem 2.1 can now be easily proved using these lemmas.

**Proof of part (ii) of Theorem 2.1.** For \( L \) sufficiently large, \( X_0 \in B \) by Lemma 2.7. Lemma 2.6 then shows that \( \lim_{L \to \infty} X_L = S_1 \). □

Equation (2.12) reveals the rate of convergence.

**Corollary 2.1.** \[ \|\phi_L(X) - S_1\| \leq c\sigma^{L-1}\|X - S_1\| \] for all \( X \in B \), where \( 0 \leq \sigma < 1 \).

This corollary shows that even though the second stage is only linearly convergent, the asymptotic error constant can be made as small as desired by increasing the number of iterations of the first stage. The asymptotic error constant for stage one depends on \( \max |\lambda S_1|/\min |\lambda S_1| < 1 \), while that of stage two can be significantly smaller.
than stage one. This is the purpose of the second stage, for (2.13) shows that stage one can also yield $S_1$.

Computational considerations of Algorithm 1, a flow-chart, and an APL program may be found in Dennis, Traub, and Weber [1].

3. The block Bernoulli algorithm. A generalization of Bernoulli’s scalar polynomial algorithm may be used to calculate a dominant solvent of the matrix polynomial problem. A relation between the block Bernoulli algorithm and the first stage of Algorithm 1 is established at the end of the section.

**Algorithm 2.** Let $X_0 = X_1 = \cdots = X_{m-2} = 0$, $X_{m-1} = I$ and define matrices $X_i$, $i \geq m - 1$, by

\begin{equation}
X_{i+1} + A_1X_i + \cdots + A_mX_{i-m+1} = 0.
\end{equation}

The general solution to the matrix recursion (3.1) is given by the following theorem.

**Theorem 3.1.** If $S_1, \ldots, S_m$ are right solvents of $M(X)$, such that $V(S_1, \ldots, S_m)$ is nonsingular, then

\begin{equation}
X_i = S_i^1\Omega_1 + \cdots + S_m^i\Omega_m,
\end{equation}

is the general solution to the matrix difference equation (3.1), where $\Omega_1, \ldots, \Omega_m$ are matrices determined by the initial conditions.

**Proof.** Substitution of equation (3.2) into equation (3.1) yields for $i \geq m - 1$

\[
\sum_{j=0}^{m} A_jX_{i-j+1} = \sum_{j=0}^{m} A_j \sum_{k=1}^{m} S_k^{j-1-i+1}\Omega_k
\]

\[
= \sum_{k=1}^{m} \left( \sum_{j=0}^{m} A_jS_k^{j-1} \right) S_k^{i-m+1}\Omega_k = 0
\]

where $A_0 = I$. The nonsingular block Vandermonde insures that $\Omega_1, \ldots, \Omega_m$ can be uniquely calculated in terms of $X_0, X_1, \ldots, X_{m-1}$. If $\hat{X}_i$ is the general solution to equation (3.1) and $X_i = \hat{X}_i$ for the first $m$ consecutive subscripts, then $X_i = \hat{X}_i$ for all $i$.

In the scalar Bernoulli method, if there is a dominant root, then the ratio of the Bernoulli iterates converges to the root. This is generalized to matrix polynomials by the following theorem.

**Theorem 3.2.** If $M(X)$ is a matrix polynomial of degree $m$ such that

(i) it has solvents $S_1, \ldots, S_m$,

(ii) $S_1$ is a dominant solvent,

(iii) $V(S_1, \ldots, S_m)$ and $V(S_2, \ldots, S_m)$ are nonsingular, then

\[
\lim_{n \to \infty} X_nX_{n-1}^{-1} = S_1.
\]

**Proof.** From Theorem 3.1 we have

\[
X_n = \sum_{i=1}^{m} S_i^n\Omega_i.
\]

Combining this with the starting conditions that $X_0 = X_1 = \cdots = X_{m-2} = 0$ and
\[ X_{m-1} = I \] we get
\[
\begin{pmatrix}
I & \cdots & I \\
S_1 & \cdots & S_m \\
\vdots & \ddots & \vdots \\
S_1^{m-1} & \cdots & S_m^{m-1}
\end{pmatrix}
\begin{pmatrix}
\Omega_1 \\
\Omega_2 \\
\vdots \\
\Omega_m
\end{pmatrix} =
\begin{pmatrix}
X_0 \\
X_{m-2} \\
\vdots \\
X_{m-1}
\end{pmatrix} = 0.
\]

Partitioning, we get
\[
\begin{pmatrix}
I \\
S_1 \\
\vdots \\
S_1^{m-2}
\end{pmatrix}
\begin{pmatrix}
\Omega_1 + V(S_2, \cdots, S_m) \\
\Omega_2 \\
\vdots \\
\Omega_m
\end{pmatrix} = 0.
\]

and
\[
S_1^{m-1} \Omega_1 + (S_2^{m-1}, \cdots, S_m^{m-1}) = I.
\]

Combining these equations, we get
\[
\begin{pmatrix}
S_1^{m-1} - (S_2^{m-1}, \cdots, S_m^{m-1}) V^{-1}(S_2, \cdots, S_m) \\
I \\
S_1 \\
\vdots \\
S_1^{m-2}
\end{pmatrix}
\begin{pmatrix}
\Omega_1 \\
\Omega_2 \\
\vdots \\
\Omega_m
\end{pmatrix} = \begin{pmatrix}
I \\
0 \\
0 \\
0
\end{pmatrix}
\]

and hence \( \Omega_1 \) is nonsingular. Now observe that
\[
X_n X_{n-1}^{-1} = \left( \sum_{i=1}^{m} S_i^n \Omega_i \right) \left( \sum_{i=1}^{m} S_i^{n-1} \Omega_i \right)^{-1}
\]
\[
= (S_1 + W_n S_1^{n-1} \Omega_1^{-(n-1)} S_1^{-(n-1)}) (I + V_n S_1^{n-1} \Omega_1^{-(n-1)} S_1^{-(n-1)})^{-1},
\]

where
\[
W_n = \sum_{j=2}^{m} S_j^n \Omega_j S_1^{-(n-1)}, \quad V_n = \sum_{j=2}^{m} S_j^{n-1} \Omega_j S_1^{-(n-1)}.
\]

Since
\[
\lim_{n \to \infty} W_n S_1^{n-1} \Omega_1^{-(n-1)} S_1^{-(n-1)} = 0, \quad \lim_{n \to \infty} V_n S_1^{n-1} \Omega_1^{-(n-1)} S_1^{-(n-1)} = 0,
\]

the result follows. \( \square \)

Note that the theorem could have been made somewhat more general by removing the condition on \( V(S_2, \cdots, S_m) \) and relaxing the initial conditions on \( X_0, \cdots, X_{m-1} \) to just insure \( \Omega_1 \) being nonsingular.

The quantity \( X_{n-1} X_n \) also converges, but not to \( S_1 \). See Dennis, Traub, and Weber [1, Thm. 6.2].
The block Bernoulli iteration (3.1) can also be written as

\[
\begin{pmatrix}
X_{i-m+2} \\
\vdots \\
X_i \\
X_{i+1}
\end{pmatrix}
\begin{pmatrix}
0 & I \\
\vdots & \ddots \\
0 & -A_m & -A_{m-1} & \cdots & -A_1
\end{pmatrix}
\begin{pmatrix}
X_{i-m+1} \\
\vdots \\
X_{i-1} \\
X_i
\end{pmatrix}
\]

(3.3)

where \(X_i\) is a matrix of order \(n\). Equation (3.3) looks like eigenvector powering except

\[
\begin{pmatrix}
X_{i-m+1} \\
\vdots \\
X_{i-1} \\
X_i
\end{pmatrix}
\]

is not a vector in the usual sense. A theory of such power methods is studied in Dennis, Traub, and Weber [1, Chapter 8] We do not pursue this here.

Consider the same power-like method on the transpose of the matrix in equation (3.3). With a superscript on \(W\) denoting iteration count, we have

\[
\begin{pmatrix}
W_{m+1}^j \\
\vdots \\
W_2^j \\
W_1^j
\end{pmatrix}
\begin{pmatrix}
0 & \cdots & 0 & -A_m^T \\
I & \cdots & -A_{m-1}^T \\
\vdots & \ddots & \vdots & \vdots \\
I & -A_1^T
\end{pmatrix}
\begin{pmatrix}
W_m^j \\
\vdots \\
W_2^j \\
W_1^j
\end{pmatrix}
\]

Multiplying out, we obtain the system

\[
\begin{align*}
W_m^{j+1} &= -A_m^T W_1^j \\
W_{m-1}^{j+1} &= W_m^j - A_{m-1}^T W_1^j \\
& \quad \vdots \\
W_1^{j+1} &= W_2^j - A_1^T W_1^j
\end{align*}
\]

Multiply the \(j\)th equation on the left by \((X^T)^{j-1}\) and add. The result is

\[
G_{i+1}(X) = G_i(X) X - (W_i^j)^T M(X),
\]

where

\[
G_i(X) = (W_i^j)^T X^{m-1} + \cdots + (W_m^j)^T.
\]

This is precisely stage one of Algorithm 1. These results are generalizations of what occurs in the scalar case. See Traub [5].

4. Numerical examples. Three numerical examples of Algorithm 1 are given. The first illustrates the case where the convergence theorem applies and the iteration converges. In the last two examples hypothesis (ii) is violated and Algorithm 1 does not converge. Modifications of Algorithm 1 are discussed.

Additional numerical examples may be found in Dennis, Traub, and Weber [1].

**Example 4.1.** Consider the monic cubic matrix polynomial

\[
M(X) = X^3 + \begin{pmatrix} -6 & 6 \\ -3 & -15 \end{pmatrix} X^2 + \begin{pmatrix} 2 & -42 \\ 21 & 65 \end{pmatrix} X + \begin{pmatrix} 18 & 66 \\ -33 & -81 \end{pmatrix}
\]
Start Stage 1 with $G_0(X) = X^2$. After five steps of Stage 1 we have

$$\tilde{G}_S(X) = X^2 + \begin{pmatrix} -2.026 & 3.711 \\ -1.856 & -7.593 \end{pmatrix} X + \begin{pmatrix} -1.715 & -9.193 \\ 4.597 & 12.075 \end{pmatrix}.$$  

Stage 2 then gives

$$X_0 = \begin{pmatrix} 3.9925 \\ 1.2131 \end{pmatrix},$$

$$X_1 = \begin{pmatrix} 3.9729 \\ 1.0446 \end{pmatrix},$$

$$X_2 = \begin{pmatrix} 3.9927 \\ 1.0089 \end{pmatrix},$$

$$X_3 = \begin{pmatrix} 3.9985 \\ 1.0017 \end{pmatrix},$$

$$X_4 = \begin{pmatrix} 3.9997 \\ 1.0003 \end{pmatrix},$$

and

$$X_5 = \begin{pmatrix} 3.9999 \\ 1.0001 \end{pmatrix}$$

is a dominant right solvent of the matrix polynomial.

**Example 4.2.**

$$M(X) = X^2 + \begin{pmatrix} 7 & 8 \\ 8 & 10 \end{pmatrix} X + \begin{pmatrix} 9 & 3 \\ 4 & 4 \end{pmatrix}.$$  

The corresponding lambda-matrix has latent roots $-16.05113$, $-.4215$, and $-.2637 \pm 1.8649i$. There exist two solvents having these as their eigenvalues, but neither can dominate, since there is a complex pair of latent roots whose absolute value is between the two other latent roots. Algorithm 1 does not converge. A complex shift of the variable in the lambda-matrix can be used to break up complex pairs of latent roots. With a shift of $i$, Algorithm 1 converges with no difficulties.

**Example 4.3.** Consider the quadratic

$$M(X) = X^2 + \begin{pmatrix} -1 & -6 \\ 2 & -9 \end{pmatrix} X + \begin{pmatrix} 0 & 12 \\ -2 & 14 \end{pmatrix}.$$  

The corresponding lambda-matrix has latent roots 1, 2, 3, 4 with corresponding latent vectors $(1, 0)^T$, $(0, 1)^T$, $(1, 1)^T$, $(1, 1)^T$. The problem has solvents

$$S_1 = \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}.$$  

$M(X)$ has additional solvents with eigenvalues 1, 2; 1, 4 and 2, 3. The only pair chosen from 1, 2, 3, 4 which cannot be the eigenvalues of a solvent is 3, 4. Thus, there is no dominant solvent and Algorithm 1 did not converge.
Reversing the order of matrix coefficients has the effect of making the latent roots the reciprocals of the original latent roots. The right solvents are the inverses of the original ones. Thus, 1 and $\frac{1}{2}$ are the new dominant latent roots. Algorithm 1 converges to

$$\begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix},$$

and, hence, the solvent

$$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

is found for the original problem.

REFERENCES


