ALGORITHMS FOR SOLVENTS OF MATRIX POLYNOMIALS

J. E. Dennis, Jr.
Cornell University

J. F. Traub
Carnegie-Mellon University

R. P. Weber
Bell Laboratories

October, 1975

J. E. Dennis, Jr. was supported in part by National Science Foundation Grant GJ27528. J. F. Traub 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.

1. INTRODUCTION

Let

$$M(X) = X^m + A_1 X^{m-1} + \ldots + A_m \qquad (1.1)$$

where $A_1, \ldots, A_m$, X are n by n complex matrices. We call M(X) a monic matrix polynomial of degree m. A matrix S for which $M(S) = \underline{\underline{0}}$ is called a right solvent (or briefly a solvent). In Dennis, Traub, and Weber [76] 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 [66]). It is globally convergent in the following sense. If Stage One is done sufficiently long and if the hypotheses of Theorem 2.1 hold, then the iteration of Stage Two is globally convergent. Stage One may be viewed as direct powering by a "block companion matrix". We have not succeeded in devising an inverse powering process as used by Jenkins and Traub [70]. 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 [71] 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 λ 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. The process can be repeated until all the latent roots have been computed. Other proposed methods for the calculation of latent roots such as algorithms of Lancaster [64] and Kublanovskaya [70] are only locally convergent and do not have an associated method of deflation. See Dennis, Traub, and Weber [71, Appendix B] for additional material on algorithms for lambda-matrices.

In Dennis, Traub, and Weber [71] 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 [76]. For the reader's convenience, we state a number of definitions and results from the above cited paper crucial to this paper.

If $S_1,\ldots,S_m$ are any n by n matrices the block Vandermonde matrix is defined by

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

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

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

We have

<u>Theorem 1.1.</u> If Matrices $S_1,\ldots,S_m$ are such that $V(S_1,\ldots,S_m)$ is nonsingular, then there exist unique matrix polynomials $M_i(X)$, $i=1,\ldots,m$ which are fundamental matrix polynomials. If, furthermore, $V(S_1,\ldots,S_{k-1},S_{k+1},\ldots,S_m)$ is nonsingular, then $A_1^{(k)}$ is nonsingular.

<u>Theorem 1.2.</u> If matrices $S_1,\ldots,S_m$ are such that $V(S_1,\ldots,S_m)$ is nonsingular and $M_1(X),\ldots,M_n(X)$ are a set of fundamental matrix polynomials, then an arbitrary matrix polynomial $G(X)$ can be written as

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

We summarize the remainder of this paper. In Sections 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 [66]) 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 [71, Corollary 5.1]).

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

$$G_{n+1}(X) = G_n(X)X - \alpha_1^n M(X), \tag{2.1}$$

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

$$G_n(X) \equiv \alpha_1^n X^{m-1} + \ldots + \alpha_m^n. \tag{2.2}$$

Stage 2. *Let* $X_0 = \left(\alpha_1^L\right)\left(\alpha_1^{L-1}\right)^{-1}$ *and define matrices* X_i *by*

$$X_{i+1} = G_L(X_i)G_{L-1}^{-1}(X_i) \equiv \phi_L(X_i) \tag{2.3}$$

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 wo do in our programs. The choice $G_0(X) = I$ makes the proof of Theorem 2.1 slightly simpler.

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 *strictly dominate, in modulus, those of* B.

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

Proof: For any $\epsilon > 0$, let

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

where

$$J_B(\epsilon) = \begin{pmatrix} \lambda_B & & & \\ \epsilon & \lambda_B & & \\ & \ddots & \ddots & \\ & & \epsilon & \lambda_B \end{pmatrix}$$

and where λ_B denotes eigenvalues of B. Then,

$$\|B^n\| \le \|P_B(\epsilon)\| \, \|P_B(\epsilon)^{-1}\| \, (\epsilon+\max|\lambda_B|)^n, \tag{2.4}$$

using the infinity norm.

Letting

$$A^{-1} = P_{A^{-1}}(\epsilon)(J_{A^{-1}}(\epsilon))P_{A^{-1}}(\epsilon)^{-1},$$

we similarly obtain

$$\begin{aligned} \|A^{-n}\| &\le \|P_{A^{-1}}(\epsilon)\| \, \|P_{A^{-1}}(\epsilon)^{-1}\| \, (\epsilon+\max|\lambda_{A^{-1}}|)^n \\ &= \|P_{A^{-1}}(\epsilon)\| \, \|P_{A^{-1}}(\epsilon)^{-1}\| \, (\epsilon+\frac{1}{\min|\lambda_A|})^n. \end{aligned} \tag{2.5}$$

Combining equations (2.4) and (2.5) we get

$$\|A^{-n} \; C \, B^n\| \le k[(\epsilon+\max|\lambda_B|)(\epsilon + \frac{1}{\min|\lambda_A|})]^n \tag{2.6}$$

where k, a function of ϵ, 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. ■

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

$$\bar{G}_n(X) \equiv (\alpha_1^n)^{-1} G_n(X),$$

$$\bar{M}_i(X) = [A_1^{(i)}]^{-1} M_i(X)$$

be monic matrix polynomials. Then we have

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) $\lim_{n\to\infty} \bar{G}_n(X) = \bar{M}_1(X)$

(ii) for L sufficiently large,

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

Proof of part (i). From (2.1)

$$G_n(S_i) = S_i^n \,. \tag{2.7}$$

By Theorem 1.2 and (2.7)

$$G_n(X) = \sum_{i=1}^{m} G_n(S_i) M_i(X) = \sum_{i=1}^{m} S_i^n M_i(X), \tag{2.8}$$

and, thus,

$$\alpha_1^n = \sum_{i=1}^{m} S_i^n A_1^{(i)}. \tag{2.9}$$

By Theorem 1.1, $A_1^{(1)}$ is nonsingular. Since S_1 is also nonsingular there is an N such that for $n \geq N$, α_1^n must be nonsingular, since using Lemma 2.1 and (2.9),

$$\lim_{n\to\infty} \alpha_1^n (S_1^n A_1^{(1)})^{-1} = I.$$

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

$$\tilde{G}_n(X) = \left(\sum_{i=1}^{m} S_i^n A_1^{(i)}\right)^{-1} S_1^n S_1^{-n} \left(\sum_{i=1}^{m} S_i^n M_i(X)\right)$$

$$= \left(\sum_{i=1}^{m} S_1^{-n} S_i^n A_1^{(i)}\right)^{-1} \left(\sum_{i=1}^{m} S_1^{-n} S_i^n M_i(X)\right)$$

and the conclusion follows by an application of Lemma 2.1. ■

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 right solvent S.

Proof. The result follows from (2.7).

<u>Lemma 2.3</u>. <u>There exists a nontrivial ball</u> B, <u>centered at</u> S_1, <u>such that for</u> all $X \in B$

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

<u>and</u>

(ii) $\|M_j(X)\| \leq 1, \quad j \neq 1$.

<u>Proof</u>. 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 $M_j(S_1) = \underline{\underline{0}}$ for $j \neq 1$. ■

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

$$\|M_1^{-1}(X)\| \leq \frac{1}{1 - \|I-M_1(X)\|} \quad .$$

<u>Lemma 2.4</u>. <u>If</u> $X \in B$, <u>then there exists an</u> L' <u>such that</u> $\phi_L(X)$ <u>is defined for every</u> $L \geq L'$.

<u>Proof</u>. For $x \in B$, let

$$V_j(X) = M_j(X)M_1^{-1}(X), \; W_L(X) = \sum_{j=2}^{m} S_1^{-L}S_j^L V_j(X).$$

Then,

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

$$= S_1^{L-1}\left(I + \sum_{j=2}^{m} S_1^{-(L-1)}S_j^{L-1}V_j(X)\right) M_1(X)$$

$$= S_1^{L-1}(I+W_{L-1}(X))\; M_1(X). \qquad (2.10)$$

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

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

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. ■

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

$$\|S_j^L V_j(X) S_1^{-L}\| \le \tau\sigma^L \|M_j(X) M_1^{-1}(X)\| \le \frac{\tau\sigma^L}{1-K}, \tag{2.11}$$

where $0 \le \sigma < 1$, *and* τ *is a constant independent of* L *and* X.

Proof. The result follows from (2.6), where

$$\sigma = \max|\lambda_{S_j}|/\min|\lambda_{S_1}| < 1 \quad \text{for} \quad j \neq 1.$$ ■

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 $L \ge L'$ as in Lemma 2.4. Set

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

Then, since

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

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

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,

$$\|T_{j,L}(X)\| \leq \frac{\tau\sigma^{L}}{1 - K}.$$

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),

$$\|E_L(X)\| \leq \sum_{j=2}^{m} \frac{\|S_j - S_1\| \tau\sigma^{L-1} \|M_j(X)\| \; \|M_1^{-1}(X)\|}{1 - F}$$

for all $X \in B$. A matrix polynomial is continuously differentiable. Since $M_j(S_1) = \underline{\underline{0}}$ for $j \neq 1$, the result

$$\|M_j(X)\| \leq t\|X - S_1\|,$$

where $j \neq 1$, $t = \sup_{\substack{x \in B \\ j \neq 1}} \|M_j'(X)\|$, follows from the mean value theorem. Finally,

$$\|\phi_L(X) - S_1\| \leq c\sigma^{L-1} \|X - S_1\| \tag{2.12}$$

for all $X \in B$, where

$$c = \mathrm{abs}\left(\frac{\sum_{j=2}^{m} \|S_j - S_1\| \tau t}{(1-F)(1-K)}\right).$$

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,* $\left(\alpha_1^L\right)\left(\alpha_1^{L-1}\right)^{-1} \in B$.

Proof. Noting that $\alpha_1^L = \sum_{j=1}^{m} S_j^L A_1^{(j)}$, a proof similar to that in Lemma 2.6 yields

$$\lim_{L\to\infty} \left(\alpha_1^L\right)\left(\alpha_1^{L-1}\right)^{-1} = S_1. \tag{2.13}$$

■

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 $\lim_{i\to\infty} X_i = S_1$. ■

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 $\sigma = \max|\lambda_{S_j}|/\min|\lambda_{S_1}| < 1$, while that of stage two can be significantly faster 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 [71].

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* $X_0 = X_1 = \ldots = X_{m-2} = \underline{\underline{0}}$, $X_{m-1} = I$ *and define matrices* X_i *by*

$$X_{i+1} + A_1 X_i + \ldots + A_m X_{i-m+1} = \underline{\underline{0}}. \qquad (3.1)$$

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

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*

$$X_i = S_1^i \alpha_1 + \ldots + S_m^i \alpha_m \qquad (3.2)$$

is the general solution to the matrix difference equation (3.1), *where* $\alpha_1,\ldots,\alpha_m$ *are matrices determined by the initial conditions*.

Proof. Substitution of equation (3.2) into equation (3.1) yields

$$\sum_{j=0}^{m} A_j X_{i+j+1} = \sum_{j=0}^{m} A_j \sum_{k=1}^{m} S_k^{i+j+1} \alpha_k$$
$$= \sum_{k=1}^{m} \left(\sum_{j=0}^{m} A_j S_k^{m-j} \right) S_k^{i+2j-m+1} \alpha_k = \underline{\underline{0}}$$

where $A_0 = I$. The nonsingular block Vandermonde insures that $\alpha_1,\ldots,\alpha_m$ 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

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*

$$\lim_{n\to\infty} X_n X_{n-1}^{-1} = S_1.$$

Proof. From Theorem 3.1 we have

$$X_n = \sum_{i=1}^{m} S_i^n \alpha_i.$$

Combining this with the starting conditions that $X_0 = X_1 = \ldots = X_{m-2} = \underline{\underline{0}}$ and $X_{m-1} = I$ we get

$$\begin{pmatrix} I & \ldots & I \\ S_1 & \ldots & S_m \\ \vdots & & \vdots \\ S_1^{m-1} & \ldots & S_m^{m-1} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} X_0 \\ \vdots \\ X_{m-2} \\ X_{m-1} \end{pmatrix} = \begin{pmatrix} \underline{\underline{0}} \\ \vdots \\ \underline{\underline{0}} \\ I \end{pmatrix}$$

Partitioning, we get

$$\begin{pmatrix} I \\ S_1 \\ \cdot \\ \cdot \\ \cdot \\ S_1^{m-2} \end{pmatrix} \alpha_1 + V(S_2,\ldots,S_m) \begin{pmatrix} \alpha_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \underline{\underline{0}} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \underline{\underline{0}} \end{pmatrix}$$

and

$$S_1^{m-1}\alpha_1 + (S_2^{m-1},\ldots,S_m^{m-1}) \begin{pmatrix} \alpha_2 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_m \end{pmatrix} = I.$$

Combining these equations, we get

$$\left[S_1^{m-1} - (S_2^{m-1},\ldots,S_m^{m-1})\, V^{-1}(S_2,\ldots,S_m) \begin{pmatrix} I \\ S_1 \\ \vdots \\ S_1^{m-2} \end{pmatrix} \right] \alpha_1 = I$$

and hence α_1 is nonsingular. Now observe that

$$\begin{aligned} X_n X_{n-1}^{-1} &= \left(\sum_{i=1}^{m} S_i^n \alpha_i \right) \left(\sum_{i=1}^{m} S_i^{n-1} \alpha_i \right)^{-1} \\ &= \left(S_1 + W_n S_1^{n-1} \alpha_1^{-1} S_1^{-(n-1)} \right) \left(I + V_n S_1^{n-1} \alpha_1^{-1} S_1^{-(n-1)} \right)^{-1}, \end{aligned}$$

where

$$W_n = \sum_{j=2}^{m} S_j^n \alpha_j S_1^{-(n-1)}, \quad V_n = \sum_{j=2}^{m} S_j^{n-1} \alpha_j S_1^{-(n-1)}.$$

Since

$$\lim_{n\to\infty} W_n S_1^{n-1} \alpha_1^{-1} S_1^{-(n-1)} = \underline{\underline{0}}, \quad \lim_{n\to\infty} V_n S_1^{n-1} \alpha_1^{-1} S_1^{-(n-1)} = \underline{\underline{0}},$$

the result follows. ■

Note that the theorem could have been made somewhat more general by removing the condition on $V(S_2,\ldots,S_m)$ and relaxing the initial conditions on $X_0,\ldots,X_{m-1}$ to just insuring α_1 being nonsingular.

The quantity $X_{n-1}^{-1}X_n$ also converges, but not to S_1. See Dennis, Traub, and Weber [71, Theorem 6.2].

The block Bernoulli iteration (3.1) can also be written as

$$\begin{pmatrix} X_{i-m+2} \\ \vdots \\ X_i \\ X_{i+1} \end{pmatrix} = \begin{pmatrix} 0 & I & & \\ \vdots & & \ddots & \\ 0 & & & I \\ -A_m & -A_{m-1} & \cdots & -A_1 \end{pmatrix} \begin{pmatrix} X_{i-m+1} \\ \vdots \\ X_{i-1} \\ X_i \end{pmatrix} \tag{3.3}$$

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 [71, Chapter 8]. We do not pursue this here.

Consider the same power-like method on the transpose of the matrix in equation (3.3). That is,

$$\begin{pmatrix} W_m^{i+1} \\ \vdots \\ W_2^{i+1} \\ W_1^{i+1} \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 & -A_m^T \\ I & & & -A_{m-1}^T \\ & \ddots & & \vdots \\ & & I & -A_1^T \end{pmatrix} \begin{pmatrix} W_m^i \\ \vdots \\ W_2^i \\ W_1^i \end{pmatrix}.$$

Multiplying out, the system

$$\begin{array}{lcccc} W_m^{i+1} & = & & - & A_m^T \; W_1^i \\ W_{m-1}^{i+1} & = & W_m^i & - & A_{m-1}^T W_1^i \\ \vdots & & \vdots & & \vdots \quad \vdots \\ W_1^{i+1} & = & W_2^i & - & A_1^T \; W_1^i \end{array}$$

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

$$G_{i+1}(X) = G_i(X)X - (W_1^i)^T M(X),$$

where

$$G_i(X) \equiv (W_1^i)^T X^{m-1} + \ldots + (W_m^i)^T.$$

This is precisely stage one of Algorithm 1. These results are generalizations of what occurs in the scalar case. See Traub [66].

4. NUMERICAL EXAMPLES

Four numerical examples of Algorithm 1 are given. The first illustrates the case where the convergence theorem applies and the iteration converges. In the second example convergence occurs even though hypothesis (iii) of Theorem 2.1 is violated. 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 [71].

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 one with $G_0(X) = X^2$. After five steps of Stage one we have

$$\overline{G}_5(X) = X^2 + \begin{pmatrix} -2.026 & 3.711 \\ -1.856 & -7.593 \end{pmatrix} X + \begin{pmatrix} -1.715 & -9.193 \\ 4.597 & 12.075 \end{pmatrix}.$$

Stage two then gives

$$X_o = \begin{pmatrix} 3.9925 & -2.4261 \\ 1.2131 & 7.6317 \end{pmatrix},$$

$$X_1 = \begin{pmatrix} 3.9729 & -2.0892 \\ 1.0446 & 7.1067 \end{pmatrix},$$

$$X_2 = \begin{pmatrix} 3.9927 & -2.0179 \\ 1.0089 & 7.0195 \end{pmatrix},$$

$$X_3 = \begin{pmatrix} 3.9985 & -2.0034 \\ 1.0017 & 7.0035 \end{pmatrix},$$

$$X_4 = \begin{pmatrix} 3.9997 & -2.0006 \\ 1.0003 & 7.0006 \end{pmatrix},$$

and

$$X_5 = \begin{pmatrix} 3.9999 & -2.0001 \\ 1.0001 & 7.0001 \end{pmatrix}.$$

$S_1 = \begin{pmatrix} 4 & -2 \\ 1 & 7 \end{pmatrix}$ is a dominant right solvent of the matrix polynomial.

<u>Example 4.2</u>. Consider the matrix polynomial

$$M(X) = X^3 + \begin{pmatrix} -11.44382802 & 3.420249653 \\ 0.8613037448 & -5.556171983 \end{pmatrix} X^2$$

$$+ \begin{pmatrix} 41.02912621 & -20.93481276 \\ 0.5533980583 & 7.332871012 \end{pmatrix} X$$

$$+ \begin{pmatrix} -39.65603329 & 23.56171983 \\ 0.6074895978 & -3.386962552 \end{pmatrix}.$$

It has a solvent, $S_1 = \begin{pmatrix} 7 & 2 \\ -1 & 4 \end{pmatrix}$ and $S_2 = S_3 = \begin{pmatrix} 4 & -2 \\ 3 & -1 \end{pmatrix}$. The eigenvalues of S_1 are 5 and 6, while while the eigenvalues of S_2 are 1 and 2. Clearly, $V(S_1,S_2,S_3)$ and $V(S_2,S_3)$ are singular. With $L = 6$, we get

$$X_o = \begin{pmatrix} 6.7783 & 1.2464 \\ -1.0231 & 3.9215 \end{pmatrix},$$

$$X_1 = \begin{pmatrix} 6.9896 & 1.9764 \\ -1.0011 & 3.9975 \end{pmatrix},$$

and

$$X_2 = \begin{pmatrix} 6.9997 & 1.9995 \\ -1.0000 & 3.9999 \end{pmatrix}.$$

The convergence is fast, though linear, since the asymptotic error constant is $(.4)^6$. ■

Example 4.3.

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

The corresponding lambda-matrix has latent roots -16,05113, -.4215 and -.2637 ± 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 4.4. 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 $S_1 = \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}$ and $S_2 = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$. 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.

BIBLIOGRAPHY

Dennis, Traub, and Weber [71] Dennis, J. E., Traub, J. F., and Weber, R. P., "On the Matrix Polynomial, Lambda-Matrix and Block Eigenvalue Problems." Cornell University, Carnegie-Mellon University, Computer Science Department Technical Report, 1971.

Dennis, Traub, and Weber [76] Dennis, J. E., Traub, J. F., and Weber, R. P., "The Algebraic Theory of Matrix Polynomials." To appear in SIAM J. Num. Anal. 1976. Also available as Technical Report, Computer Science Department, Carnegie-Mellon University, December 1974.

Jenkins and Traub [70] Jenkins, M. and Traub, J. F., "A Three-Stage Variable-Shift Iteration for Polynomial Zeros and its Relation to Generalized Rayleigh Iteration," Numerische Mathematik, 14 (1970), 252-261.

Kublanovskaya [70] Kublanovskaya, V., "On an Approach to the Solution of the Generalized Latent Value Problem for λ-Matrices," SIAM J. Numer. Anal., 7 (1970), 532-537.

Lancaster [64] Lancaster, P., "Algorithms for Lambda-Matrices," Numerische Mathematik, 6 (1964), 338-394.

Traub [66] Traub, J. F., "A Class of Globally Convergent Iteration Functions for the Solution of Polynomial Equations," Mathematics of Computation, 20 (1966), 113-138.