NASA Contractor Report 178376

ICASE REPORT NO. 87-63

ICASE

POLYNOMIAL APPROXIMATION OF FUNCTIONS OF MATRICES AND ITS APPLICATION TO THE SOLUTION OF A GENERAL SYSTEM OF LINEAR EQUATIONS

Hillel Tal-Ezer

(NASA-CR-178376) POLYNOMIAL APPROXIMATION OF FUNCTIONS OF MATRICES AND ITS APPLICATION TO THE SOLUTION OF A GENERAL SYSTEM OF LINEAR EQUATIONS Final Report (NASA) 47 p
Avail: NTIS HC A03/MF A01 CSCL 12A G3/64

N87-30116
Unclas
0100127

Contract No. NAS1-18107
September 1987

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

POLYNOMIAL APPROXIMATION OF FUNCTIONS OF MATRICES AND ITS APPLICATION TO THE SOLUTION OF A GENERAL SYSTEM OF LINEAR EQUATIONS

Hillel Tal-Ezer*

Raymond and Beverly Sackler Faculty of Exact Sciences
Tel-Aviv University
Israel

and
Institute for Computer Applications in Science and Engineering

Abstract

Frequently, during the process of solving a mathematical model numerically, we end up with a need to operate on a vector v by an operator which can be expressed as $f(A)$ while A is $N \times N$ matrix (ex: $\exp(A)$, $\sin(A)$, A^{-1}). Except for very simple matrices, it is impractical to construct the matrix $f(A)$ explicitly. Usually an approximation to it is used. In the present research, we develop an algorithm which uses a polynomial approximation to $f(A)$. It is reduced to a problem of approximating $f(z)$ by a polynomial in z while z belongs to the domain D in the complex plane which includes all the eigenvalues of A. This problem of approximation is approached by interpolating the function $f(z)$ in a certain set of points which is known to have some maximal properties. The approximation thus achieved is "almost best." Implementing the algorithm to some practical problems is described.

Since a solution to a linear system $Ax = b$ is $x = A^{-1}b$, an iterative solution to it can be regarded as a polynomial approximation to $f(A) = A^{-1}$. Implementing our algorithm in this case is described too.

* Research was supported under the National Aeronautics and Space Administration under NASA Contract No. NAS1-18107 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA. 23665.Additional support was provided by the Air force Office of Scientific Research grant No. AF850303 and AFOSR 85-0189.

I. Introduction.

Let A be a $N \times N$ matrix whose elements belong to C, and $f(z)$ is a function such that

$$f(z) : C \to C \ . \tag{1.1}$$

The matrix $f(A)$ can be defined in the following way: If

$$f(z) = \sum_{i=0}^{\infty} a_i (z - z_0)^i \qquad |z - z_0| < r \tag{1.2}$$

then

$$f(A) = \sum_{i=0}^{\infty} a_i (A - z_0 I)^i \qquad |\lambda_k - z_0| < r \tag{1.3}$$

where λ_k is an eigenvalue of A. In (1.3), $f(A)$ is expressed as an infinite polynomial. It can be shown [Gant 59] that $f(A)$ as defined above can be written also as a finite polynomial of degree $\leq N - 1$ as follows:

$$f(A) = \sum_{k=1}^{s} \left[f(\lambda_k) H_{k1}(A) + \cdots + f^{(j_k - 1)}(\lambda_k) H_{kj_k}(A) \right] \tag{1.4}$$

where

$$\begin{array}{cc} \lambda_k & - \text{ an eigenvalue of } A \\ j_k & - \text{ multiplicity of } \lambda_k \\ H_{kj}(A) & - \text{ polynomial in } A \text{ of degree } \leq N - 1 \ . \end{array} \tag{1.5}$$

In many practical applications, we would like to compute a vector ω which results from operating with the matrix $f(A)$ on a vector v

$$\omega = [f(A)]v \ . \tag{1.6}$$

Using expression (1.4) for this purpose has two major disadvantages:

(a) The exact knowledge of the eigenvalues is required.

(b) $f(A)$ is expressed as a polynomial of degree $\leq N-1$ which can be large; thus it results in an highly time consuming operator.

In this paper, we would like to present an algorithm which approximates the matrix $f(A)$ by a polynomial $P_m(A)$, where $m << N$. Since (1.4), we have

$$E_m(A) = f(A) - P_m(A) = \sum_{k=1}^{s} \left[E(\lambda_k) H_{k1}(A) + \cdots + E^{(j_k - 1)}(\lambda_k) H_{kj_k}(A) \right] \tag{1.7}$$

while

$$E(z) = f(z) - P_m(z) \ . \tag{1.8}$$

Hence, our problem is reduced to a problem of approximating the function $f(z)$ by a polynomial $P_m(z)$ while z belongs to a domain in C which includes all the eigenvalues of A. It is obvious from (1.7) that in the case where $j_k > 1$, $P_m^{(j_k-1)}$ has to approximate $f^{(j_k-1)}$ at the point λ_k.

In Section 2, we describe briefly how to approach this problem of approximation. Using the results of this section we can construct a computational algorithm, once a suitable conformal mapping function is known. In very few cases, it is possible to get this function analytically. Usually we have to resort to a numerical method. A lot has been done in this area of conformal mapping and there are suitable routines. In our work, we have used a set of routines written by N. Trefethen based on his paper [Tref80], which are very efficient and reliable. The interested user can get the routines from the author upon request. The algorithm which results from making use of the approximating polynomial is presented in Section 3. In Section 4, we deal with the rate of convergence of the suggested method. Few examples of mathematical models for which the new algorithm can be implemented are given in Section 5. The numerical solution of a system of linear equations $Ax = b$ where A is a general nonsymmetric matrix can be regarded as a particular case of our problem where the function which has to be approximated is $f(z) = 1/z$. Section 6 is devoted to this subject. Using tools from the theory of approximation in the complex plane for inverting nonsymmetric matrices has been studied already by other researchers. A few of them are: T. Manteuffel [Mant77], [Mant78], D. Smolarski and P. Saylor [SmSa85], Gutknecht [Gutk86], Y. Saad [Saad87],and others. The algorithm discussed in our paper gains its simplicity from the fact that it is based on the powerful tool of interpolation. Thus, it can be implemented in a straightforward way, once the conformal mapping function is known. An important conclusion of the analysis is that for the general nonsymmetric case, the significant factor which governs the rate of convergence of most of the iterative algorithms is not the well known condition number $\|A\| \cdot \|A^{-1}\|$. It is shown that the relevant factor is ρ/R where ρ is a parameter which measures the size of the domain and R is related to the orientation of the domain with regard to the singular point of the function $\frac{1}{z}$ $(z = 0)$.

Based on this conclusion it is shown that some iterative algorithms, like standard SOR, are based on an incomplete optimization process. We conclude this paper in Section 7 by giving some numerical examples.

2. Polynomial Approximation in the Complex Plane

Let D be a bounded closed continuum in C such that the complement of D is simply connected in the extended plane and contains the point at ∞. Define now

$A(D)$ - the space of functions which are analytic at every point of D.

π_m - space of complex polynomials of degree $\leq m$.

Then it is well known [SmLe68] that for every $f \in A(D)$ there exists $P_m^* \in \pi_m$ such that

$$\|f - P_m^*\|_\infty \leq \|f - P_m\|_\infty \qquad \forall P_m \in \pi_m \ . \tag{2.1}$$

From algorithmic point of view, it is relatively complicated to find P_m^*. In many cases, it is quite simple to find a polynomial approximation which is "almost" as good as P_m^*. It is found by projecting $A(D)$ on π_m. If S_m is a projection operator

$$S_m : \ A(D) \rightarrow \pi_m \ , \tag{2.2}$$

then

$$f - S_m(f) = f - P_m^* + S_m(P_m^* - f); \tag{2.3}$$

thus

$$\|f - S_m(f)\| \leq (1 + \|S_m\|)\|f - P_m^*\| \ . \tag{2.4}$$

If $||S_m||$ is reasonably small, regarding $S_m(f)$ as "almost" as good as P_m^* is justified. For example, if D is the unit disc and

$$S_m(f) = \sum_{j=0}^{m} \frac{f^{(j)}(0)}{j!} z^j \ ; \tag{2.5}$$

then [GeMa75]

$$||S_m|| \leq \frac{4}{\pi^2} \log(m) + 1 + 0(1) \ . \tag{2.6}$$

If $D = [-1,1]$ and

$$S_m(f) = \sum_{k=0}^{m} a_k T_k(z) \tag{2.7}$$

while

$$T_k(z) = \cos(k(\cos^{-1} z)) \qquad \text{(Chebyshev polynomials)} , \tag{2.8}$$

then the bound on $||S_m||$ is exactly the same as in the previous case (2.6). A generalization to an arbitrary domain D (as defined in the beginning of this section) is given by a Faber polymomials expansion. Let $\phi(z)$ be a conformal mapping from z to ω which maps the complement of D to the complement of a disc of radius ρ such that

$$\lim_{z\to\infty} \frac{\phi(z)}{z} = 1 . \tag{2.9}$$

ρ is the logarithmic capacity or the transfinite diameter of D [SmLe68].

If the Laurent expansion at ∞ of $[\phi(z)]^m$ is

$$[\phi(z)]^m = z^m + c_{m-1}z^{m-1} + \cdots c_1 z + c_0 + c_{-1}z^{-1} + \cdots , \tag{2.10}$$

then Faber polynomial of degree m, $F_m(z)$, which corresponds to the domain D is the polynomial part of (2.10):

$$F_m(z) = z^m + c_{m-1}z^{m-1} + \cdots + c_0 . \tag{2.11}$$

We have

$$c_j = \frac{1}{2\pi i}\int_{|z|=R} \frac{[\phi(z)]^m}{z^{j+1}} dz \tag{2.12}$$

while R is chosen sufficiently large so that D is contained in the interior of the region bounded by the circle $|z| = R$ [SmLe68]. Given $f \in A(D)$ then

$$f(z) = \sum_{k=0}^{\infty} a_k F_k(z) . \tag{2.13}$$

The coefficients a_k are

$$a_k = \frac{1}{2\pi i}\int_{|\omega|=r} \frac{f(\psi(\omega))}{\omega^{k+1}} d\omega$$

where $\psi(\omega)$ is the inverse of $\phi(z)$ and $r > \rho$ is sufficiently small that f can be extended analytically to I_r. (I_r is the closed Jordan region with boundary Γ_r where Γ_r is the image under ψ of the circle $|\omega| = r$.) In [Ella83] it was shown how the c_j's and a_k's can be computed efficiently using F.F.T. Based on (2.13), the matrix $f(A)$ can be approximated by $P_m(A)$ where

$$P_m(A) = \sum_{k=0}^{m} a_k F_k(A) \ . \tag{2.14}$$

When D is an ellipse with axes parallel to the real and imaginary axes,then F_k is the translated and scaled Chebyshev polynomial T_k [Mark77]; thus it satisfies a simple three term recurrecnce relation. This recurrence relation enables us to compute

$$\omega = \left[\sum_{k=0}^{m} a_k F_k(A)\right] v \tag{2.15}$$

efficiently. A detailed description of these cases, approximating the operator e^A, can be found in [Tale85], [Tale86], [KoTa86], [TaKo84]. For more complicated domains, F_k satisfies a k terms recurrence relation [Ella83]; thus from memory point of view it is impractical to use an algorithm based on (2.15).

This major drawback can be overcome by using a different approach to the approximating problem. It uses interpolation as the projection operator S_m. Using this tactic we are confronted with the following question: Which are the interpolating points in D, such that the interpolating polynomial will be "almost" as good as P_m^*? It is known [SmLe68] that if D is the unit disc, then two possible sets of points are:

(a) $z_j = 0 \qquad j = 1, \ldots, m \qquad$ (zeroes of z^m)

(b) The m roots of unity.

In a similar way, for a general domain D, a two possible sets of points are [SmLe68]:

(a) the m zeroes of $F_m(z)$ (2.14a)

(b) $z_j = \psi(\omega_j) j = 1, \ldots, m$, while ω_j are the m roots (2.14b)
of the equation $\omega^m = \rho$.

In the first case, it can be shown [GeMa75] that

$$||S_m|| \leq \frac{4}{\pi^2} \log(m) + 1 + 0(1) \tag{2.15}$$

while in the second case

$$||S_m|| \leq \frac{2}{\pi}\log(m) + 1 + 0(1) \ . \tag{2.16}$$

Since, interpolating at $z_j = \psi(\omega_j)$ $j = 0, \ldots, m$ does not involve computing Faber polynomials, it is the simplest and most efficient approach to this problem of approximation. The interpolating polynomial in Newton form is:

$$P_m(z) = a_0 + a_1(z - z_0) + a_2(z - z_0)(z - z_1) + \cdots + a_m(z - z_0)\cdots(z - z_{m-1}) \tag{2.17}$$

where a_k is the divided difference of order k

$$a_k = f[z_0, \ldots, z_k] \qquad k = 0, \ldots, m \ . \tag{2.18}$$

When ρ (the logorithmic capacity of D) is large, a_i will be very small; thus it is preferable to make the following change of variables

$$\hat{z} = z/\rho \ . \tag{2.19}$$

Hence

$$\hat{f}(\hat{z}) = f(z) = f(\rho \cdot \hat{z}) \tag{2.20}$$

and

$$P_m(\hat{z}) \cong \hat{f}(\hat{z}) \tag{2.21}$$

where

$$P_m(\hat{z}) = b_0 + b_1(\hat{z} - \hat{z}_0) + \cdots + b_m(\hat{z} - \hat{z}_0)\cdots(\hat{z} - \hat{z}_{m-1}) \tag{2.22a}$$

$$b_k = \hat{f}[\hat{z}_0, \ldots, \hat{z}_k] \qquad k = 0, \ldots, m \ . \tag{2.22b}$$

The only difficulty in finding $P_m(\hat{z})$ is to get the function $\psi(\omega)$. For simple domains, it is possible to find this function analytically. For more complicated domains one has to resort to a numerical approach.

When the domain D is a polygon, the mapping function is Schwartz-Cristoffel transformation. In [Tref80], a very reliable and efficient algorithm for mapping the interior of the unit disc to the interior of the polygon is described. Since, in our case, the mapping of the exteriors is needed, the routines should be modified accordingly.

Separated domains: In some very important physical phenomenon, we have a situation where the eignevalues of the matrix A are clustered in two or more separated domains which can be far apart (for example: stiff O.D.E.'s problems of parazitic waves in meteorological sciences, spectral methods for nonperiodic boundary values problems, etc.). Hence, the domain D is a union of k subdomains:

$$D = \cup_{i=1}^{k} D_i \ .$$

In this case, the complement of D is not simply connected any more but just connected. The basic theory regarding the simply connected case extends to the more general one. A detailed analysis of this problem is carried out in our next paper. It is shown there that the interpolating points can be taken as a union of sets of points achieved by considering each domain separately. In Section 7, we bring some numerical examples of this case.

3. The Algorithms

Based on (2.22), we will approximate the operator $f(A)$ by $P_m(\hat{A})$ while

$$\begin{aligned} P_m(\hat{A}) =& b_0 I + b_1(\hat{A} - \hat{z}_0 I) + b_2(\hat{A} - \hat{z}_0 I)(\hat{A} - \hat{z}_1 I) + \\ & \cdots + b_m(\hat{a} - \hat{z}_0 I) \cdots (\hat{A} - \hat{z}_{m-1} I) \ . \end{aligned} \tag{3.1}$$

$\hat{A} = (1/\rho)\ A$

Operating with $P_m(\hat{A})$ on a vector v is carried out by the following algorithm

Algorithm 1:

$$\begin{aligned} & u = v \\ & \omega = b_0 v \\ & \text{For } i = 1, \ldots, m \quad \text{do} \\ & \quad u = (\hat{A} - \hat{z}_{i-1} I) u \\ & \quad w = w + b_i u \end{aligned} \tag{3.2}$$

The output of Algorithm 1 is the vector ω which satisfies

$$\omega = P_m(\hat{A}) v \ . \tag{3.3}$$

Roundoff errors of Algorithm 1 depend strongly on the arrangement of the interpolating points.

In Appendix A we describe how to arrange the points, taking into account this phenomenon of roundoff errors.

In many practical problems, we have the following situation:

1) A is a real matrix (3.4a)

2) $f(\overline{z}) = \overline{f(z)}$. (3.4b)

In this case, it is possible to design an algorithm similar to (3.2) which will be carried out in real arithmetic even so $\hat{z}_i$ and b_i are complex numbers. This result is based on the following two theorems:

Theorem 1: *Let $z_0, z_1, z_2, \overline{z}_2, \ldots, z_k, \overline{z}_k$ be $2k$ interpolating points where z_0 and z_1 are real numbers. If $P_{2k-1}(z)$,*

$$P_{2k-1}(z) = a_0 + a_1 z + \cdots a_{2k-1} z^{2k-1} \tag{3.5}$$

is the interpolating polynomial of a function $f(z)$ which satisfies (3.4b) then $a_0, a_1, \ldots, a_{2k-1}$ are real numbers.

Proof. The Lagrange formula of $P_{2k-1}(z)$ is

$$P_{2k-1}(z) = \sum_{j=0}^{k} \ell_j(z) f(z_j) + \sum_{j=2}^{k} \ell_j^*(z) f(\overline{z}_j) \tag{3.6}$$

where

$$\ell_j(z) = \frac{Q_{2k}(z)}{Q'_{2k}(z_j)} \frac{1}{z - z_j} \quad j = 0, \ldots, k \tag{3.7}$$

$$\ell_j^*(z) = \frac{Q_{2k}(z)}{Q'_{2k}(\overline{z}_j)} \frac{1}{z - \overline{z}_j} \quad j = 2, \ldots, k \tag{3.8}$$

$$Q_{2k}(z) = \prod_{i=0}^{k} (z - z_i) \prod_{i=2}^{k} (z - \overline{z}_i) \, . \tag{3.9}$$

Since (3.9) we have

$$Q'_{2k}(z_j) = \prod_{\substack{i=0 \\ i \neq j}}^{k} (z_j - z_i) \prod_{i=2}^{k} (z_j - \overline{z}_i) \quad j = 0, \ldots, k \tag{3.10}$$

$$Q'_{2k}(\overline{z}_j) = \prod_{i=0}^{k}(\overline{z}_j - z_i) \prod_{\substack{i=2 \\ i\neq j}}^{k} (\overline{z}_j - \overline{z}_i) \quad j = 2, \ldots, k \ . \tag{3.11}$$

Hence

$$\begin{aligned} \overline{Q'_{2k}(z_j)} &= \prod_{\substack{i=0 \\ i\neq j}}^{k} (\overline{z}_j - \overline{z}_i) \prod_{i=2}^{k}(\overline{z}_j - z_i) = \\ &= \prod_{\substack{i=2 \\ i\neq j}}^{k} (\overline{z}_j - \overline{z}_i) \prod_{i=0}^{k}(\overline{z}_j - z_i) = Q'_{2k}(\overline{z}_j) \end{aligned} \tag{3.12}$$

$$\begin{aligned} \overline{Q_{2k}(z)} &= \prod_{i=0}^{k}(\overline{z} - \overline{z}_i) \prod_{i=2}^{k}(\overline{z} - z_i) = \prod_{i=2}^{k}(\overline{z} - \overline{z}_i) \prod_{i=0}^{k}(\overline{z} - z_i) = \\ &= Q_{2k}(\overline{z}) \ . \end{aligned} \tag{3.13}$$

Therefore,

$$\overline{\ell_j(z)} = \frac{Q_{2k}(\overline{z})}{Q'_{2k}(\overline{z}_j)} \frac{1}{\overline{z} - \overline{z}_j} = \begin{cases} \ell_j(\overline{z}) & j = 0, k \\ \ell_j^*(\overline{z}) & 2 \leq j \leq 1 \end{cases} \tag{3.14a}$$

$$\overline{\ell_j^*(z)} = \frac{Q_{2k}(\overline{z})}{Q'_{2k}(z_j)} \frac{1}{\overline{z} - z_j} = \ell_j(\overline{z}) \qquad 2 \leq j \leq k \ . \tag{3.14b}$$

Using (3.4b) and (3.14) in (3.6) we get

$$\overline{P_{2k-1}}(z) = \sum_{j=2}^{k} \ell_j^*(\overline{z}) f(\overline{z}_j) + \sum_{j=0}^{k} \ell(\overline{z}) f(z_j) = P_{2k-1}(\overline{z}) \tag{3.15}$$

and the proof is concluded.

Theoretically, it is possible to write an algorithm based on (3.5). It means to approximate $f(A)$ by $P_m(A)$ where

$$P_m(A) = a_0 I + a_1 A + \cdots + a_m A^m \ . \tag{3.16}$$

This algorithm cannot be used for practical problem since huge roundoff errors result. We would like to stick to the Newton-form which is much more robust from the roundoff errors point of view. For this purpose, we state and prove the following theorem:

Theorem 2: *Let $P_{2k-1}(z)$ be the interpolating polynomial as defined in Theorem 1, written in Newton-form*

$$P_{2k-1}(z) = b_0 + b_1(z - z_0) + (z - z_0)(z - z_1) \prod_{i=1}^{k-1} S_i(z) R_i(z) \tag{3.17}$$

$$S_i(z) = b_{2i} + b_{2i+1}(z - z_{i+1}) \tag{3.18a}$$

$$R_i(z) = \prod_{j=2}^{i}(z - z_j)(z - \overline{z}_j) \qquad i = 2, \ldots, k-1 \tag{3.18b}$$

$$R_1(z) = 1 \ . \tag{3.18c}$$

then

$$P_{2k-1}(z) = b_0 + b_1(z - z_0) + (z - z_0)(z - z_1) \prod_{i=1}^{k-1} S_i^R(z) R_i(z) \tag{3.19}$$

where

$$S_i^R(z) = b_{2i}^R + b_{2i+1}^R(z - z_{i+1}^R) \tag{3.20a}$$

$$b_{2i}^R = \text{Real } (b_{2i}) \tag{3.20b}$$

$$b_{2i+1}^R = \text{Real } (b_{2i+1}) = b_{2i+1} \tag{3.20c}$$

$$z_{i+1}^R = \text{Real } (z_{i+1}). \tag{3.20d}$$

Proof. We have

$$P_{2k+1}(z) = P_{2k-1}(z) + (z - z_0)(z - z_1) S_k(z) R_k(z). \tag{3.21}$$

Define

π - The set of all polynomials with real coefficients.

Then, by theorem 1

$$P_{2k+1}(z), \ P_{2k-1}(z) \in \pi \ .$$

$R_k(z) \in \pi$ by definition.

Since

$$S_k(z) = [P_{2k+1}(z) - P_{2k-1}(z)]/(z - z_0)(z - z_1) R_k(z) \tag{3.22}$$

then

$$S_k \in \pi \ .$$

Thus

$$b_{2k+1}^R = b_{2k+1} \tag{3.23}$$

$$b_{2k} - b_{2k+1}z_{k+1} = b_{2k}^R - b_{2k+1}^R z_{k+1}^R \tag{3.24}$$

and the proof is concluded.

Based on (3.19), the vector ω

$$\omega = [P_{2k-1}(\hat{A})]v \tag{3.25}$$

is computed by the following algorithm:

Algorithm 2.

$$\omega = [b_0 I + b_1(\hat{A} - z_0 I)]v \tag{3.26a}$$

$$r = (\hat{A} - \hat{z}_0 I)(\hat{A} - \hat{z}_1 I)v. \tag{3.26b}$$

For $i = 1,,\ldots,k-1$ do:

$$\tilde{r} = (\hat{A} - \hat{z}_{i+1}^R I)r \tag{3.26c}$$

$$\omega = w + b_{2i}^R r + b_{2i+1}^R \tilde{r} \tag{3.26d}$$

$$r = (\hat{A} - \hat{z}_{i+1}^R I)\tilde{r} + (\hat{z}_{i+1}^I)^2 r \qquad (\hat{z}_{i+1}^I = I_m(\hat{z}_{i+1})). \tag{3.26e}$$

This algorithm requires three vectors. It is possible to save one vector by using an algorithm based on (3.16). As mentioned previously, this algorithm has the disadvantage of sensitivity to roundoff errors. Thus,it can be used only with low degree polynomials. Another possible way of saving one vector and which is much more robust from the roundoff errors point of view is to apply (3.19) through calculating the roots of $P_{2k-1}(z)$. Since $P_{2k-1}(z)$ is a polynomial with real coefficients, we have the following set of roots

$$\lambda_1, \lambda_2, \ldots, \lambda_r, \mu_1, \overline{\mu}_1, \ldots, \mu_s, \overline{\mu}_s \tag{3.27}$$

where

$$r + 2s = 2k - 1; \qquad \lambda_i \qquad \text{are real .} \tag{3.28}$$

Hence

$$P_{2k-1}(A) = \alpha \prod_{i=1}^{r}(A - \lambda_i I) \prod_{i=1}^{s}(A^2 - 2Re\mu_i A + |\mu_i|^2 I) \tag{3.29}$$

$$\alpha = \frac{P_{2k-1}(0)}{\beta} : \qquad \beta = (-1)^r \prod_{i=1}^{r} \lambda_i \prod_{i=1}^{s} |\mu_i|^2 . \tag{3.30}$$

Operating with (3.29) requires only two vectors. This approach is limited mainly by the sensitivity of the algorithm which finds the roots of $P_{2k-1}(z)$.

4. Rate of Convergence.

We have the following definitions:

Definition (4.1): Let Γ_R be the image under ψ of the circle $|\omega| = R(R > \rho)$ and I_R is the closed Jordan region whose boundary is Γ_R. If $f(z)$ is single valued and analytic on I_R, then the sequence of polynomials $P_m(z)$ is said to converge to $f(z)$ on I_R ***maximally*** if

$$|f(z) - P_m(z)| \leq C(\rho/R)^m \qquad z \in I_R \tag{4.1}$$

where C depends on ρ/R but not on m or z.

Definition (4.2): The set of interpolating points $z_j = \psi(\omega_j)$ is said to be ***uniformly distributed*** on Γ_D (the boundary of D) if ω_j are equally distributed on the circle $|\omega| = \rho$.

Using these two definitions, we can quote the following [Wals56]:

Theorem: Let D be a closed Jordan region. Let the points $\beta_k^{(m)}$ lie on the boundary Γ_D of D. A necessary and sufficient condition that the sequence of polynomials $P_m(z)$ of degree m found by interpolation to a function $f(z)$ analytic on D in the points $\beta_k^{(m)}$ converges uniformly to $f(z)$ on D is that the points $\beta_k^{(m)}$ be uniformly distributed on Γ_D. If this condition is satisfied, the sequence $P_m(z)$ converges ***maximally*** to $f(z)$ on D.

Given a domain D and a function $f(z)$, ρ and R are defined explicitly and we have

$$\rho/R = \text{the asymptotic rate of convergence.}$$

If $f(z)$ is an entire function, (4.1) is satisfied for arbitrary R. Using Theorem (1), we can expect that the algorithm described in Section 3 will converge very rapidly for the approximation of the operator $\exp(A)$. On the other end, for the operator A^{-1}, the rate of convergence will depend strongly on the size of D (the parameter ρ) and on its orientation with regard to the singular point $z = 0$ (the parameter $R = |\phi(0)|$). Since the set of interpolating points $z_j = \psi\left(\rho e^{ij\frac{2\pi}{m}}\right) \qquad j = 0, \ldots, m-1$ depend on m, given a

desired accuracy ε, one has to decide a priory on the degree of the polynomial. Deciding on m can be done in the following ways:

1. Getting the parameters ρ and R (analytically or numerically) and choosing m such that

$$(\rho/R)^m \cong \varepsilon \ . \tag{4.2}$$

2. Computing the error numerically on a set of check points on the boundary for different m's, and choosing m which will satisfy the desired accuracy.

Using 1 or 2 gives us information on $|f(z) - P_m(z)|_\infty$. Substituting it for $\|f(A) - P_m(A)\|_\infty$ is relatively accurate when A is a normal matrix, since in this case we have

$$\begin{aligned} \|f(A) - P_m(A)\|_\infty &\leq |f(z) - P_m(z)|_\infty \|T\|^{-1}\|T\| \\ &= |f(z) - P_m(z)|_\infty \end{aligned}$$

where T and T^{-1} are the unitary matrices which diagnolize A. When A is "far" from normality, m should be increased by an amount which depends on $\|T\| \cdot \|T^{-1}\|$.

5. Applications.

Frequently, while solving a system of O.D.E.'s or P.D.E.'s, we end up with the following set of differential equations

$$\frac{d}{dt}U_N - G_N U_N = S_N(x,t) \tag{5.1}$$

$$U_N(0) = U_N^0$$

where U_N and S_N are vectors of dimension N and G_N is a $N \times N$ matrix. Expanding $S_N(x,t)$ as

$$S_N(x,t) = \sum_{j=1}^{k} \alpha_j(t) S_N^j(x)$$

enables us to write the formal solution of (5.1) as

$$U_N(t) = \exp(G_N t)U_N^0 + \sum_{j=1}^{k} f_j(G_N t) S_N^j(x) \tag{5.2}$$

where

$$f_j(G_N t) = \int_0^t \exp(G_N \tau)\alpha_j(t-\tau)d\ .\tau \qquad 1 \le j \le k \tag{5.3}$$

$U_N(t)$ can be approximated by the algorithm discussed in Section 3 where the functions which have to be interpolated are

$$f_0(z) = \exp(tz) \tag{5.4a}$$

$$f_j(z) = \int_0^t \exp(z\tau)\alpha_j(t-\tau)d\tau \qquad 1 \le j \le k \tag{5.4b}$$

In the case where (5.1) is originated from a system of hyperbolic P.D.E.'s, the domain D is on the imaginary axis (in the constant coefficients case) or close to it (in the variable coefficients case). When (5.1) is originated from a set of parabolic equations the domain D is on the negative real axis or close to it. In these two cases, the Faber polynomials are scaled and translated Chebyshev polynomials. Thus, an efficient algorithm, which makes use of the three terms recurrence relations, can be implemented. These two cases are described and treated in details in [Tale86], [KoTa86], [TaKo84], [Tale85].

In the more general situation, when we have both parabolic and hyperbolic terms in the equation, the domain D is more complicated. Let us look at the following simple $1-D$ equation

$$u_t = au + bu_x + cu_{xx} \qquad (a < 0, c > 0)\ . \tag{5.5}$$

If the solution is periodic in space, then a good approximation can be achieved by pseudospectral Fourier descretization. We get the following semidiscrete representation of (5.5)

$$(U_N)_t = a\ U_N + b(U_N)_x + c(U_N)_{xx} \tag{5.6}$$

where

$$U_N = \sum_{k=-N}^{N} a_k e^{ikx} \tag{5.7}$$

is the projected solution. (5.6) can be written as

$$(U_N)t = G_N U_N \tag{5.8}$$

while G_N is the finite domain operator

$$G_N = aI + b\frac{\partial}{\partial x} + c\frac{\partial^2}{\partial x^2} \ . \tag{5.9}$$

The function e^{ikx} is an eigenfunction of G_N. Thus, if λ_k is an eigenvalue of G_N, then

$$\lambda_k = a - ck^2 + ibk \qquad |k| \leq N \ . \tag{5.10}$$

Since $a < 0$ and $c > 0$, we get that the domain D is the following parabola in the complex plane:

$$D = \{z = x + iy | x = a - ck^2;\ y = bk \qquad |k| \leq N\} \ . \tag{5.11}$$

In real applications a, b, and c are not constant, but have space dependence. Therefore, the domain D will vary accordingly. In section 7 we report on some numerical experiments treating this problem.

In a joint work with Dr. Dan Kosloff and his colleagues, we investigate the implementation of the present algorithm to some real geophysical problems which can be represented as (5.1). The eigenvalues are scattered close to a T shape domain D

$$D = \{z | z \in D_1 \cup D_2 \ ; \quad D_1 = [-a, 0] \ ; \quad D_2 = [-ib, ib]\}$$

In this case, we have an analytic expression for $\psi(\omega)$. (Thanks to Nick Trefethen.) The conformal mapping $\tilde{\psi}(\omega)$ which maps the complement of the unit disc on the complement of D is:

$$\begin{aligned} \tilde{\psi}(\omega) =& \frac{b}{2}\left\{\left[\frac{1}{2}(1+E)(\omega + \frac{1}{\omega}) + 1 - E\right]^2 - 4\right\}^{1/2} \qquad |\omega| = 1 \ . \\ E =& \sqrt{a^2 + b^2}/b \end{aligned} \tag{5.12}$$

Hence

$$\psi(\omega) = \tilde{\psi}(\omega/\rho) \tag{5.13}$$

where

$$\rho = \frac{b(1+E)}{4} \tag{5.14}$$

is the logarithmic capacity of D. Numerical results for this domain are presented in section 7.

A very important area where our method can be implemented is the one of solving general nonsymmetric systems of equations. This is topic of the next section.

6. Linear System.

The iterative solution to a set of linear equations which can be written in matrix form

$$Ax = b \tag{6.1}$$

is a well treated problem in the numerical analysis literature. While very efficient algorithms have been developed for the case when A is a symmetric positive definite matrix, the general nonsymmetric case is still a challenging problem.

In this section, we would describe an iterative algorithm for the solution of (6.1) based on the new approach. Since

$$x = A^{-1}b \tag{6.2}$$

we can write the numerical approximation X^k as

$$x^k = P_k(A)b \tag{6.3}$$

where $P_k(z)$ is "almost best" approximation to the function $\frac{1}{z}$. z belongs to the domain D which includes all the eigenvalues of A.

In [Mant77], [Mant78], T.A. Manteuffel has described an iterative algorithm based on enclosing the eigenvalues in ellipses in the complex plane, thus getting an approximation based on scaled and translated Chebyshev polynomials expansion. The algorithm described in our paper is more general since it can be implemented to any domain in the complex plane. Its advantage will be significant when the discrepancy between the best ellipse as defined in [Mant77] and the domain is relatively large.

A standard approach for solving (6.1) is known as Richardson algorithm. It takes the following form:

$$x^{k+1} = x^k - \alpha_k(Ax^k - I)\ . \tag{6.4}$$

If

$$E^k = x - x^k \tag{6.5}$$

we get

$$E^k = [P_k^*(A)]E^0 \tag{6.6}$$

where

$$P_k^*(A) = \prod_{j=0}^{k-1}(I - \alpha_j A) \tag{6.7}$$

and α_j are optimal in the sense that

$$\max_{z\in D}|P_k^*(z)| \leq \max_{z\in D}|P_k(z)| \qquad \underset{P_k(0)=1}{\forall} P_k \in \pi_k \tag{6.8}$$

(π_k is the space of all polynomials of degree k). We would like to show that an algorithm which results from implementating our approach is equivalent to algorithm based on (6.4)-(6.8). For this purpose we have

Theorem 3. *Let $T_k(z)$ be a polynomial of degree k defined on C such that $T_k(0) = 1$ and let $Q_{k-1}(z)$ be the interpolant of the function $\frac{1}{z}$ at the roots of T_k then.*

$$1 - z\ Q_{k-1}(z) = T_k(z)\ . \tag{6.9}$$

Proof. We have

$$Q_{k-1}(z_i) = \frac{1}{z_i} \qquad i = 1, \ldots, k \tag{6.10}$$

where z_i is a root of T_k. Therefore, the polynomial $R_k(z)$

$$R_k(z) = T_k(z) - (1 - z\ Q_{k-1}(z)) \tag{6.11}$$

vanishes at $k+1$ points: $0, z_1, \ldots, z_k$ and thus it is identical zero. Hence,

$$T_k(z) = 1 - z\ Q_{k-1}(z) \tag{6.12}$$

and the proof is concluded.

If x^0 is the initial guess, we have

$$Ax^0 = b^0 \tag{6.13}$$

$$A(x - x^0) = b - b^0 \tag{6.14}$$

Thus

$$x = A^{-1}(b - b^0) + x^0 \ . \tag{6.15}$$

Approximating A^{-1} by the interpolant $Q_{k-1}(A)$ results in

$$x^k = Q_{k-1}(A)(b - b^0) + x^0 \tag{6.16}$$

using (6.5) and (6.14) we get

$$E^k = [I - AQ_{k-1}(A)]E^0 \ . \tag{6.17}$$

Since (6.9), the equivalence is established.

We have shown that using the algorithm based on interpolating the function $\frac{1}{z}$ at $z_1, \ldots, z_k$ will reproduce identical results to Richardson iterations

$$x^{j+1} = x^j - \alpha_j(Ax^j - b) \tag{6.18}$$

with

$$\alpha_j = \frac{1}{z_j} \qquad j = 0, \ldots, k-1 \ . \tag{6.19}$$

Writing (6.18), (6.19) as an algorithm with real arithmetic takes the following form:

Algorithm 3 (Richardson).

(The parameters are $\alpha_j = 1./z_j$ and z_j are defined in Appendix A.)

$$x^1 = x^0 - \alpha_0(Ax^0 - b) \tag{6.20a}$$

$$x^2 = x^1 - \alpha_1(Ax^1 - b) \tag{6.20b}$$

for $i = 1, \ldots, \frac{k-1}{2}$

$$R^i = Ax^{2i-1} - b \tag{6.21a}$$

$$x^{2i+1} = x^{2i-1} - [2\alpha_i^R - |\alpha_i|^2 A]R^i \qquad (\alpha_i^R = Re(\alpha_i)) \tag{6.21b}$$

Preconditioning.

Usually, solving a linear system of equations (6.1) is composed of two stages:

1. Modifying the original system to an equivalent one

$$\tilde{A}x = \tilde{b} \tag{6.22}$$

such that (6.22) will converge faster than (6.1).

2. Solving (6.22).

In many cases, we have a family of matrices $\tilde{A}$ which depend on a parameter ω such that

$$\tilde{A}_\omega x = \tilde{b}_\omega \tag{6.23}$$

and we would like to choose the optimal ω. Based on Section 4, it is obvious that the significant factor which determines the convergence is

$$\lambda = \rho/R \ . \tag{6.24}$$

Optimization is achieved when

$$|\lambda(\omega_{opt})| \leq |\lambda(\omega)| \qquad \forall\omega \ . \tag{6.25}$$

Therefore, we conclude that for matrices whose eigenvalues are scattered in the complex plane, one should consider the factor ρ/R rather than the standard condition number $\text{cond}(A) = ||A||\ ||A^{-1}||$. It is possible to have two matrices A and B such that

$$\text{cond}\ (A) < \text{cond}\ (B) \tag{6.26}$$

and on the other hand

$$[\rho/R](A) > [\rho/R](B) \ . \tag{6.27}$$

For example: Let A be a normal matrice whose spectrum $r(A)$ is

$$r(A) = \{z|\ 1 \leq |z| \leq 2;\ Re\ z > 0\} \ .$$

Let B be a normal matrice whose spectrum is

$$r(B) = \{z|\ |z-3| \leq 2\} \ .$$

We have

$$\begin{aligned} \text{cond}(A) &= 2 \\ \text{cond}(B) &= 5 \ . \end{aligned} \tag{6.28}$$

On the other hand

$$\begin{aligned} [\rho/R](A) &= 0.8547 \\ [\rho/R](B) &= 0.6666\cdots \ . \end{aligned} \tag{6.29}$$

Since (6.29), Richardson algorithm for B will converge much faster than for A, even so $\text{cond}(B) > \text{cond}(A)$. Based on this discussion, we would like to show also that the standard S.O.R. procedure can be considered as an incomplete optimization process. Solving (6.1) by S.O.R., we use the following iterative procedure

$$x^{k+1} = T_\omega x^k + b_\omega \tag{6.30}$$

where

$$\begin{aligned} T_\omega &= M_\omega^{-1} N_\omega \\ b_w &= M_\omega^{-1} b \\ M_\omega &= D + \omega L \\ N_\omega &= (1-\omega)D - \omega U \ . \end{aligned}$$

U, D, L are the lower, diagonal, and upper parts of the matrice A respectively. The optimal ω_{sor} is chosen such that

$$|r(T_{\omega_{sor}})| = \text{minimal} \ . \tag{6.32}$$

($r(A)$ is the spectral radius).

If x is a solution of (6.30), then

$$A_\omega x = b_\omega \tag{6.33}$$

where

$$A_\omega = I - T_\omega \ . \tag{6.34}$$

Thus, it is evident that rather than using an optimization procedure based on (6.32) one should use the more general one, based on (6.25). A well-known example treated in the literature for demonstrating the features of S.O.R. is the problem of solving Laplace

equation in a rectangle. In [Youn71, p. 205], it is shown that for a certain discretization of the rectangle, the optimal ω is

$$\omega_{\text{sor}} = 1.25 \ .$$

In this case, all the eigenvalues λ_i of T_ω satisfy

$$|\lambda_i| = 0.25 \ .$$

Hence, the rate of convergence is 0.25. Optimizing with respect to (6.25) can result with a different solution. We do not intend to carry out this optimization process, but to point out a different possible parameter ω. For this problem, we can choose $\overline{\omega}$

$$\overline{\omega} = 1 + \varepsilon \qquad\qquad 0 < \varepsilon < 0.25$$

such that the domain D which includes all the eigenvalues of $A_{\overline{\omega}}$ will be

$$D = B_1 \cup B_2$$

where

$$B_1 = \{\lambda_0\}$$

$$B_2 = \{z| \ |z-1| < \varepsilon\}$$

and

$$0 < \lambda_0 < 1 - \varepsilon \ .$$

Since the complement of D is not a simply connected domain, it is not included in the theory discussed in this paper. Implementing our approach to these types of domains will be carried out in a future paper. For the time being, let us mention that the basic results extend. Thus, since B_1 is composed of only one point and B_2 is a circle of radius ε around 1, it can be shown that the rate of convergence ρ/R is

$$\rho/R(A_{\overline{\omega}}) = \varepsilon < 0.25$$

and λ_0 will enter into the constant C (4.1). The set of interpolating points is the union of sets of both domains. Since in B_1 we have only one point, we will get the following points

$$\lambda_0, \mu_1, \ldots, \mu_n$$

while μ_j can be chosen as equally distributed on the circle $|z-1| = \varepsilon$

$$\mu_j = \varepsilon \ \exp(2\pi ij/n) \qquad\qquad j = 1, \ldots, n$$

or as the n zeros of Faber polynomial of degree m which is $(z-1)^n$; thus

$$\mu_j = 1 \qquad\qquad j = 1, \ldots, n \ .$$

The efficiency of using an algorithm based on $\overline{\omega}$ rather then ω_{sor} depends on the accuracy needed, since the constant C has been increased by a factor of $1/\lambda_0$.

7. Numerical Results.

I. Time dependent problem-Parabolic type.

In this subsection we present numerical results for the following problem

$$u_t - Gu = S(x,t) \tag{7.1a}$$

$$u(x,0) = 0 \ .$$

Where

$$G = a(x)\frac{\partial^2}{\partial x^2} + b(x)\frac{\partial}{\partial x} + c(x) \tag{7.1b}$$

$$S(x,t) = \sin(kx) - t\left\{[-k^2 a(x) + c(x)]\sin kx + kb(x)\cos kx\right\} \ . \tag{7.1c}$$

The exact solution of (7.1) is

$$u(x,t) = t \ \sin kx \ . \tag{7.2}$$

In order to solve (7.1) numerically, we first approximate the space operator G, by the finite difference operator G_N. Assuming periodicity we have

$$(G_N u)_j = a(x_j)\frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} + b(x_j)\frac{u_{j+1} - u_{j-1}}{2\Delta x} + c(x_j)u_j \qquad j = 0, \ldots, N-1 \tag{7.3}$$

where N is the number of grid points and

$$\Delta x = 2\pi/N \tag{7.4}$$

$$x_j = j \cdot \Delta x \qquad j = 0, \ldots, N \; . \tag{7.5}$$

Hence G_N is a $N \times N$ matrix. The semidescrete representation of (7.1) is

$$(u_N)_t - G_N u_N = S_N^1 + t S_N^2 \tag{7.6a}$$

$$(u_N^0)_j = 0 \qquad j = 0, \ldots, N \tag{7.6b}$$

$$(S_N^1)_j = \sin\,(kx_j) \qquad j = 0, \ldots, N \tag{7.6c}$$

$$(S_N^2)_j = [k^2 a(x_j) - c(x_j)] \sin\, kx_j - kb(x_j) \cos kx_j \quad j = 0, \ldots, N. \tag{7.6d}$$

The formal solution of (7.6) is

$$u_N(t) = f_1(G_N t) S_N^1 + f_2(G_N t) S_N^2 \tag{7.7}$$

where

$$f_1(G_N t) = \int_0^t \exp(G_N \tau) d\tau = [\exp(G_N t) - I]/G_N \tag{7.8a}$$

$$f_2(G_N t) = \int_0^t \exp(G_N \tau)(t - \tau) d\tau = [\exp(G_N t) - G_N t - I]/G_N^2. \tag{7.8b}$$

In order to implement our algorithm we have to get an approximation of the domain D which includes the eigenvalues of G_N. One way to get it is by doing a Fourier analysis of the constant coefficients operator.

$$(\tilde{G}_N u)_j = a \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} + b \frac{u_{j+1} - u_{j-1}}{2\Delta x} - c u_j \tag{7.9a}$$

where

$$a = \max_{0 < x \leq 2\pi} |a(x)|; \qquad b = \max_{0 < x \leq 2\pi} |b(x)|; \qquad c = \max_{0 < x \leq 2\pi} |c(x)|; \text{ or } c = \min_{0 < x \leq 2\pi} |c(x)| \; . \tag{7.9b}$$

Let v_k be an eigenvector of $\tilde{G}_N$; then

$$(v_k)_j = e^{ikj\Delta x} \tag{7.10}$$

and

$$\lambda_k = \frac{2a}{\Delta x^2}(\cos(k\Delta x) - 1) - i \frac{b}{\Delta x} \sin(k\Delta x) - c \tag{7.11}$$

is an eigenvalue of $\tilde{G}_N$.

The tables in this subsection present numerical results for the fully descrete solution of (7.1) where

$$k = 3$$

$$a(x) = 1./(2 + \cos x) \tag{7.12}$$

$$b(x) = 1./(2 + \sin(x)) \tag{7.13}$$

$$c(x) = -20./(2 + \cos x). \tag{7.14}$$

From (7.11) we get

$$A \leq Re\ \lambda_k \leq B \tag{7.15}$$

$$|I_m \lambda_k| \leq C \tag{7.16}$$

while

$$A = -\frac{4}{\Delta x^2} - 20\ ; \quad B = -20/3\ ; \quad C = \frac{1}{\Delta x}\ . \tag{7.17}$$

Since

$$|B - A| \gg C \tag{7.18}$$

taking $\tilde{D}$ as

$$\tilde{D} = \{x | A \leq x \leq B\} \tag{7.19}$$

will be a relatively good approximation to the domain D. We have

$$\rho(\tilde{D}) = (B - A)/4\ . \tag{7.20}$$

Hence, in order to implement the new algorithm we have to interpolate the functions

$$f_1(\rho\hat{z}t) = [\exp(\rho\hat{z}t) - 1]/\rho\hat{z}$$

$$f_2(\rho\hat{z}t) = [\exp(\rho\hat{z}t) - \rho\hat{z}t - 1]/(\rho\hat{z})^2$$

at points on $\frac{1}{\rho}\tilde{D}$.

In this case we can take the M interpolating points as the extreme of the scaled and translated Chebychev polynomial of degree M [Rivl74]. Thus

$$\tilde{z}_j = \frac{1}{2}[(B - A)\tilde{x}_j + B + A] \qquad j = 1, \ldots, M$$

where

$$\tilde{x}_j = \cos\left[\frac{(j-1)\pi}{M-1}\right] .$$

In order to avoid roundoff errors one should arrange the interpolating points as described in Appendix A. Hence

$$z_j = \frac{1}{2}[(B-A)x_j + B + A] \qquad j = 1,\ldots,M \tag{7.21}$$

where

$$x_1 = 1 \quad ; \quad x_2 = -1 \tag{7.22a}$$

$$x_j = Re(\omega_{2j-3}) \qquad j = 3,\ldots,M \tag{7.22b}$$

and ω_{2j-3} are as in Figure (A.1) for the case $M = 7$.

Since z_j are real we can use algorithm 1 (3.2). The next two tables present the numerical results using algorithm 1 with z_j defined by (7.21)-(7.22).

Table 1.

Mesh refinement chart - problem (7.1)

Using algorithm 1. $t = 1$

N	M	$L_2 I_n$
32	24	.1108-01
64	52	.2592-02
128	112	.7033-03

N - Number of grid points.

M - Number of matrix-vector multiplications (Each evolution operator is approximated by a polynomial of degree $M/2$)

$L_2 I_n$ - L_2 Error at $t = 1$.

For sake of comparison we present in Table 2 a similar chart while using a standard second order in time scheme

$$u^{n+1} = u^n + \Delta t G_N u^n + \frac{\Delta t^2}{2} G_N^2 u^n \qquad (t = n \cdot \Delta t). \tag{7.23}$$

Table 2.

Mesh refinement chart - problem (7.1)

Using (7.23) $t = 1$

N	M	L_2T_a
32	102	.1108-01
64	408	.2729-02
128	1632	.6828-03

(In this case $M/2$ is the number of time steps.)

Comparing Tables 1 and 2 we observe that while in the standard second order scheme (7.23) M is proportional to N^2 (which results from the fact that $\Delta t = 0(\Delta x^2)$) for the new algorithm, M is "almost" proportional to N. This phenomenon is explained and proved in [Tale 85]. For $t = 20$ we have the following results;

Table 3.

Mesh refinement chart - problem (7.1)

Using algorithm 1 $t = 20$

N	M	L_2I_n
32	24	.1322-01
64	52	.3407-02
128	112	.1180-02

Observing Tables 1 and 3 we notice that M does not depend on t. It can be explained as follows: For large t, $||\exp(G_N t)||$ is much smaller than the error which results from the space descretization. Thus, since (7.8) we have

$$f_1(G_N t) \cong -1/G_N$$

$$f_2(G_N t) \cong -(G_N t + I)/G_N^2$$

which means that for large t, approximating $f_1(G_N t)$ is equivalent to inverting G_N,and approximating $f(G_N t)$ is equivalent to inverting G_N^2.

We did not implement the second order algorithm (7.27) for $t = 20$, but it is straightforward (by stability considerations) that for $N = 32, 64, 128$ we have $M = 2040, 8160, 32640$ respectively. Hence for $t = 20$ and $N = 128$ the new algorithm is more efficient than the second order scheme by the impressive factor of $32640/112 \cong 290$.

II. Viscoelastic Wave Propogation.

We have worked on this model with the Geophysical group in Tel- Aviv University headed by Dr. Dan Kosloff. A detailed report on the results for a general 2-D problems will be published elsewhere. In this subsection we describe the implementation of the algorithm for the simple 1-D model

$$U_t = GU \qquad 0 \leq X \leq L \tag{7.24}$$

where

$$G = \begin{pmatrix} 0 & 1 & 0 \\ m_u \nabla & 0 & \nabla \\ \alpha & 0 & -1/\tau \end{pmatrix} \quad ; \quad U = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \tag{7.25}$$

$$\nabla = \frac{\partial^2}{\partial x^2} \tag{7.26a}$$

$$m_u = c^2 \left[1 - (1 - \frac{\tau}{\eta})\right] \tag{7.26b}$$

$$c = 2000 \qquad \text{(the speed of sound)} \tag{7.26c}$$

$$\alpha = c^2(1 - \tau/\eta)/\eta \tag{7.26d}$$

u_1 is the pressure,u_2 is the pressure time derivative,u_3 is a memory variable and τ, η are parameters of the problem. The initial data are

$$u_1(x, 0) = \exp\left[-\frac{1}{2}(x - \frac{1}{2}L)^2\right] \tag{7.27a}$$

$$u_2(x, 0) = u_3(x, 0) = 0. \tag{7.27b}$$

The space descretization is done by Fourier method (GoOr81). Thus the semidescrete representation is

$$(U_N)_t = G_N U_N \tag{7.28a}$$

$$\begin{aligned} U_N^1(x,0) &= P_N[\exp(-\frac{1}{2}(x-\frac{1}{2}L)^2] \; . \\ U_N^2(x,0) &= U_N^3(x,0) = 0 \; . \end{aligned} \tag{7.28b}$$

P_N is the pseudospectral projection operator:

$$P_N f = \sum_{k=-N}^{N} a_k \exp[i(2\pi k/L)x] \tag{7.29}$$

$$a_k = \frac{1}{2NC_k} \sum_{j=0}^{2N-1} f(x_j) \exp[-i(2\pi k/L)x_j] \qquad \begin{cases} C_k = 1 \text{ for } |k| \neq N \\ C_N = C_{-N} = 2 \end{cases} \tag{7.30}$$

$$\begin{aligned} x_j &= j\Delta x \\ j &= 0, \ldots, 2N-1 \\ \Delta x &= L/2N \end{aligned} \tag{7.31}$$

and

$$G_N = P_N G P_N \; . \tag{7.32}$$

(G_N is a $6N \times 6N$ matrix.)

In our experiments we took

$$N = 64 \; . \tag{7.33}$$

$$\Delta x = 20 \; . \tag{7.34}$$

Thus

$$L = 2N \times \Delta x = 2560 \; . \tag{7.35}$$

Since $N = 64$, G_N is a 384×384 matrix.

Approximating the domain D,which includes the eigenvalues of G_N,is done by making use of the following idea. Let us assume that the number of points K which are needed to resolve the coefficients of the operator G are relatively small. In this case, decreasing the space domain by a factor of K/N and using the same Δx gives us the matrix G_K. Since

K is small, one can use a library routine to compute the eigenvalues of G_K. The domain D_K,which includes this set of eigenvalues,is relatively a good approximation of D. In the present case G_N is a constant coefficient matrix,and the domain D_4 achieved by computing the eigenvalues of G_4 (a 12×12 matrix) is exactly the same as D_{128}. The domain D_4 is

$$D_4 = \{z | z \in [-a, 0] \text{ or } z \in [-ib, ib]\} \tag{7.36}$$

where a and b depend on τ, η. For this domain we have an analytic expression of the conformal mapping which maps the complement of the unit disc on the complement of D_4 (5.12). The logarithmic capacity is given by (5.14).

We ran two sets of numerical experiments. In the first one we took

$$\tau = 0.1600890 \times 10^{-3}$$

$$\eta = 0.1582262 \times 10^{-3}$$

and we have gotten

$$a = 6320 \qquad ; \qquad b = 317 \ .$$

Using algorithm 2 (3.26) for computing the solution at time level $t = 0.1$ results in

N	M	$L_2 I_n$
128	130	.1588-02
128	150	.2904-05

(Since we do not have an analytic expression for the exact solution, we computed $L_2 I_n$ by comparing the numerical solution to another numerical solution achieved by using algorithm 2 with $M = 300$.) Similarly, using the second order in time algorithm (7.23) results in

N	M	$L_2 T_a$
128	632	.1902-1

(For $M < 632$ the scheme is unstable.)

In the next experiment we took

$$\tau = 0.1600890 \times 10^{-4}$$

$$\eta = 0.1582262 \times 10^{-4}$$

and we have gotten

$$a = 63201 \qquad ; \qquad b = 317 \ .$$

The results at $t = .1$ were

N	M	$L_2 I_n$
128	400	.2200-02
128	450	.1096-04

While using (7.23) results in

N	M	$L_2 Ta$
128	6320	.1892-3

(For $M < 6320$ the scheme is unstable.)

III. Linear Systems.

For the numerical experiments reported in this subsection, we have used a test matrix $A_{N\times N}$ which is block diagonal. Each block is of the following shape

$$\begin{pmatrix} a_j & b_j^2 \\ -c_j^2 & a_j \end{pmatrix} . \tag{7.37}$$

Hence, the eigenvalues are

$$\lambda_j = a_j \pm i|b_j c_j| \qquad j = 1, \ldots, N/2 \ . \tag{7.38}$$

Two kinds of experiments have been carried out:

$b_j = c_j \Rightarrow$ the matrix is normal.

$b_j \neq c_j \Rightarrow$ the matrix is not normal.

We solve

$$Av = w \qquad (A \text{ is a } 1000 \times 1000 \text{ matrix}) \tag{7.39}$$

where

$$v = [v_1, \ldots, v_{1000}]^T$$

and

$$v_j = (-1)^j \qquad j = 1, \ldots, 1000 \tag{7.40}$$

and the initial guess v^0 is such that

$$v_j^0 = 0 \qquad j = 1, \ldots, 1000\ . \tag{7.41}$$

Three methods have been tested:

(1) I_n - the method described in this paper

(2) C_b - Chebyshev approach (Mant78)

(3) M_r - Minimum residual.

The minimum residual algorithm is defined as follows:

Given initial guess x^0 for $k = 0, 1, \ldots,$ until satisfied do

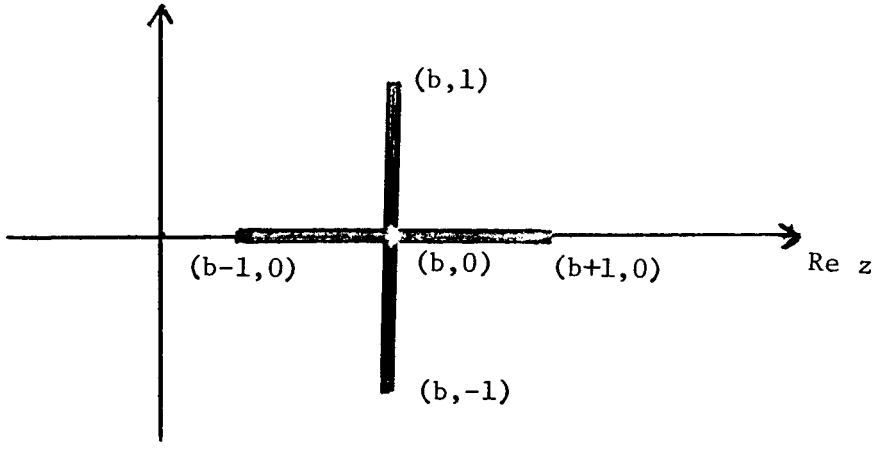
$$r^k = Ax^k - b$$

$$\alpha_k = (r^k, Ar^k)/(Ar^k, Ar^k)$$

$$x^{k+1} = x^k - \alpha_k r^k$$

1. First case - Cross shaped domain.

In [SmSa85], the authors treat an example given by Hageman [HaYo81] which originates from the solution of the neutron diffusion equation, where the eigenvalues of the Jacobi iteration matrix may be shown to lie on a the following Cross-shaped domain D:



The conformal mapping from the exterior of the unit disc to the exterior of D is

$$z = \tilde{\psi}(w) = b + \frac{1}{\sqrt{2}}\sqrt{w^2 + \frac{1}{w^2}} \; . \tag{7.42}$$

Since (7.42) we get that

$$\rho = 1/\sqrt{2} \; . \tag{7.43}$$

Thus the mapping function from the exterior of disc of radius ρ to the exterior of D is

$$z = \psi(w) = b + \sqrt{w^2 + \frac{1}{2w^2}} \; . \tag{7.44}$$

Since

$$R = |\psi^{-1}(0)| \tag{7.45}$$

we have

$$R = \left[\frac{b^2 + (b^4 - 1)^{1/2}}{2}\right]^{1/2} \; . \tag{7.46}$$

Using (7.43) and (7.46) we get that the asymptotic rate of convergence is

$$\rho/R = \left[b^2 + (b^4 - 1)^{1/2}\right]^{-1/2} \; . \tag{7.47}$$

According to the theory (4.1)

$$|f(z) - P_M(z)| \leq C(\rho/R)^M \; ; \tag{7.48}$$

thus we can predict that after M iterations, the accuracy (in the normal case) will be

$$L_2\text{Error}(I_n) \cong (\rho/R)^M \ . \tag{7.49}$$

An improved prediction, which includes the constant C is:

$$L_2\text{Error}(I_n) \cong C(\rho/R)^M_{ap} \cong \prod_{i=1}^{M}(1 - \frac{z}{z_i}) \tag{7.50}$$

where z is any point at the domain D and z_i are the interpolating points.

(In the next two tables we have chosen $z = (b, 0)$.)

The numerical results are

Table 7.

Solution of (7.39) with $b = 1.004$

M	$(\rho/R)^M$	$C(\rho/R)^M_{ap}$	L_2Error (I_n)		L_2Error (C_b)		L_2Error (M_r)	
			Normal	N. Normal	Normal	N. Normal	Normal	N. Normal
102	1.575-3	3.16-3	3.87-3	1.02+0	1.85-1	9.32+0	9.67-2	6.55+0
201	2.8-6	5.63-6	6.9-6	4.41-3	1.04-1	10.31+0	5.37-2	7.34+0
402	9.0-12	1.8-11	2.2-11	2.29-8	3.9-2	7.65+0	1.9-2	7.13+0

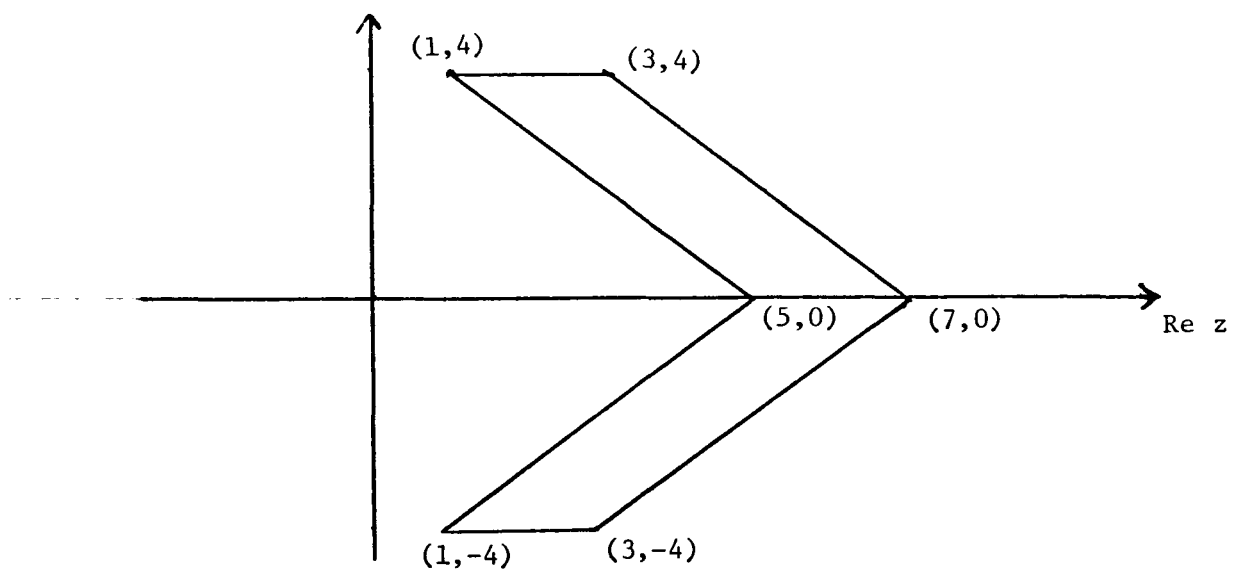
Table 8.

Solution of (7.39) with $b = 1.1$

M	$(\rho/R)^M$	$C(\rho/R)^M_{ap}$	L_2Error (I_n)		L_2Error (C_b)		L_2Error (M_r)	
			Normal	N. Normal	Normal	N. Normal	Normal	N. Normal
10	4.13-2	8.99-2	1.10-1	9.69-1	1.9-1	9.9-1	1.57-1	8.0-1
22	9.03-4	1.8-3	2.22-3	6.37-2	5.01-2	5.56-1	4.07-2	4.29-1
42	1.5-6	3.1-6	3.8-6	2.82-4	6.34-3	1.33-1	5.1-3	9.98-2
82	4.5-12	9.0-12	1.1-11	2.15-9	1.2-4	4.81-3	9.4-5	3.27-3

2. Second Case - Boomerang shape domain.

Another example reported in [SmSa85] is Van der Vorst's example. Let M be the matrix arising from the descretization of the P.D.E. operator $u_{xx} + u_{yy} + \beta_1 u_x + \beta_2 u_y$ and let K be the incomplete Cholesbey factorization; then the eigenvalues of the preconditioned matrix $K^{-1}M$, are sometimes observed to form the following boomerang-shaped profile



Since, in this case we do not have an explicit expression for the conformal mapping, we have used numerical algorithm [Tref88] for computing the interpolating points z_i. The predicted accuracy is

$$C(\rho/R)^M_{ap} \cong \prod_{i=1}^{M}(1 - \frac{z}{z_i}) \tag{7.51}$$

while

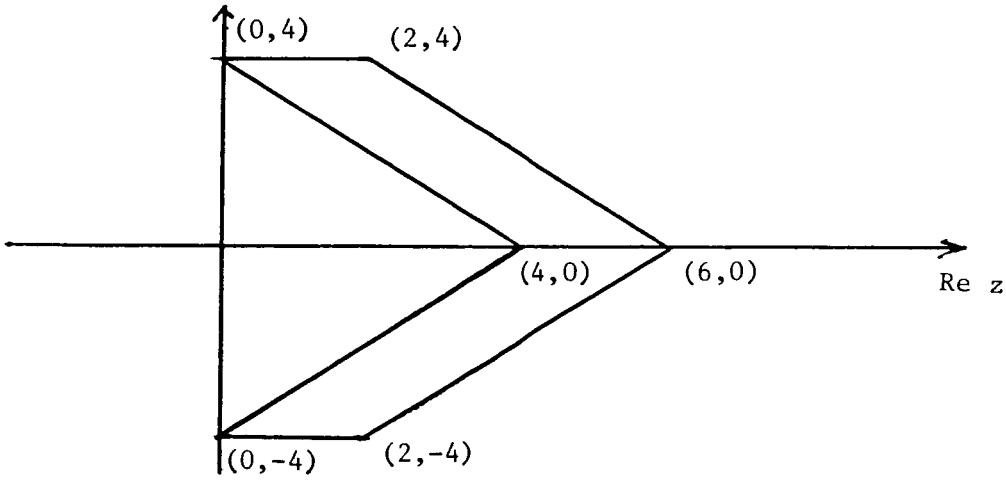
$$z = (6, 0) \ . \tag{7.52}$$

Table 9 summerizes the results of this case.

Table 9.

M	$C(\rho/R)^M_{ap}$	L_2Error (I_n)		L_2Error (C_b)		L_2Error (M_r)	
		Normal	N. Normal	Normal	N. Normal	Normal	N. Normal
20	1.23-4	5.4-4	8.8-4	1.18-1	1.33-1	2.42-1	2.35+0
40	7.58-8	2.36-7	5.14-7	4.3-2	5.8-2	1.13-1	2.35+0
60	3.78-11	1.03-10	3.05-10	1.71-2	4.24-2	5.59-2	2.35+0

In the next experiment we have shifted the domain to the left.



According to the theory, the two methods: C_b and M_r would not converge in this case. Table 10 verified this fact.

Table 10.

M	$C(\rho/R)_{ap}^{M}$	L_2Error (I_n)		L_2Error (C_b)		L_2Error (M_r)	
		Normal	N. Normal	Normal	N. Normal	Normal	N. Normal
20	1.56-3	6.65-3	1.08-2	1.15+0	1.31+0	7.09-1	3.62+0
40	1.22-5	3.8-5	8.25-5	4.93+0	6.67+0	6.76-1	3.62+0
60	7.91-8	2.16-7	6.38-7	23.0+0	56.9+0	6.58-1	3.62+0

3. Third Case - disjoint intervals.

As mentioned previously (Section 2), since the complement of D is not simply connected anymore one should use a slightly different theory. In a future paper we will carry out a detailed analysis.

In this subsection we report on a few experiments where D is

$$D = I_1 \cup I_2 \tag{7.53}$$

$$I_1 = [\alpha_1, \alpha_2] \; ; \quad I_2 = [\alpha_3, \alpha_4] \tag{7.54}$$

$$\alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 \; . \tag{7.55}$$

The interpolating points are

$$z_1^1, z_2^1, \ldots, z_{N_1}^1, z_1^2, z_2^2, \ldots, Z_{N_2}^2 \tag{7.56}$$

where

$$z_i^1 = \frac{1}{2}\left[(\alpha_2 - \alpha_1)\cos\frac{\pi(i-1)}{(N_1-1)} + \alpha_2 + \alpha_1\right] \qquad i = 1, \ldots, N_1 \tag{7.57}$$

$$z_i^2 = \frac{1}{2}\left[(\alpha_4 - \alpha_3)\cos\frac{\pi(i-1)}{(N_2-1)} + \alpha_3 + \alpha_4\right] \qquad i = 1, \ldots, N_2 \; . \tag{7.58}$$

The error polynomial $P_N(z)$ is

$$P_N(z) = P_{N_1}(z) \cdot P_{N_2}(z) \qquad (N = N_1 + N_2) \tag{7.59}$$

while

$$P_{N_1}(z) = \prod_{i=1}^{N_1}(1 - z/z_i) \tag{7.60}$$

$$P_{N_2}(z) = \prod_{i=1}^{N_2}(1 - z/z_i) \ . \tag{7.61}$$

We have

$$|P_{N_1}(z)| \sim (\rho_1/R_1)^{N_1} \qquad z \in I_1 \ . \tag{7.62}$$

Since the conformal mapping from the exterior of a unit disc to the exterior of an interval $[a, b]$ is

$$\psi^{-1}(w) = \frac{1}{2}\left[\frac{1}{2}(b-a)(w + \frac{1}{w}) + b + a\right] \ , \tag{7.63}$$

we get

$$\rho = (b-a)/4 \tag{7.64}$$

$$R = (a + b + \sqrt{ab})/4 \ . \tag{7.65}$$

Thus

$$|P_{N_1}(z)| \sim (\rho_1/R_1)^{N_1} = \left(\frac{\sqrt{\alpha_2} - \sqrt{\alpha_1}}{\sqrt{\alpha_2} + \sqrt{\alpha_1}}\right)^{N_1} \qquad z \in I_1 \ . \tag{7.66}$$

$P_{N_2}(z)$ satisfies

$$|P_{N_2}(z)| < 1 \qquad z \in I_1 \ . \tag{7.67}$$

Therefore if

$$\left(\frac{\sqrt{\alpha_2} - \sqrt{\alpha_1}}{\sqrt{\alpha_2} + \sqrt{\alpha_1}}\right)^{N_1} = \varepsilon \tag{7.68}$$

then

$$|P_N(z)| < \varepsilon \qquad z \in I_1 \ . \tag{7.69}$$

Now, for $z \in I_2$ we have

$$\max\ |P_{N_1}(z)| \sim k\alpha_4^{N_1} \qquad \left(k = 1/\prod_{i=1}^{N_1} z_i^1\right) \tag{7.70}$$

$$|P_{N_2}(z)| \sim (\rho_2/R_2)^{N_2} \tag{7.71}$$

where

$$\rho_2/R_2 = \frac{\sqrt{\alpha_4} - \sqrt{\alpha_3}}{\sqrt{\alpha_4} - \sqrt{\alpha_3}} \ . \tag{7.72}$$

Therefore, if

$$k\alpha_4^{N_1}(\rho_2/R_2)^{N_2} < \varepsilon \tag{7.73}$$

then

$$|P_N(z)| < \varepsilon \qquad z \in I_2 \ . \tag{7.74}$$

Using (7.66), (7.68),and (7.73),we get that N_2 has to satisfy

$$N_2 > \frac{\log(\rho_1/R_1\alpha_4)}{\log(\rho_2/R_2)} N_1 + \frac{\log(1/k)}{\log(\rho_2/R_2)} \ . \tag{7.75}$$

In Table (11) we report on experiments where α_1, α_2 are fixed, and we increase α_3, α_4 such that $\alpha_4 - \alpha_3$ is constant. According to (7.75), it is easily verified that in this case $N_2 \to N_1$. The predicted error is

$$E_r = \max\ |P_N(z)| = P_N(\alpha_4) = \prod_{i=1}^{N_1}(1 - \alpha_4/z_i^1) \prod_{i=1}^{N_2}(1 - \alpha_4/z_i^2) \ . \tag{7.76}$$

Table 11.

$\alpha_1 = 1 \quad \alpha_2 = 3 \quad \alpha_4 = \alpha_3 + 2$

α_3	N_1	N_2	E_r	L_2 Error (I_n)		L_2 Error (C_b)		L_2 Error (M_r)	
				Normal	N. Normal	Normal	N. Normal	Normal	N. Normal
20	12	12	1.9-7	1.1-7	1.7-5	6.6-5	3.5-3	28-3	1.0-2
40	12	12	3.2-7	1.9-7	5.4-4	9.7-4	8.7-2	2.2-3	4.1-2
80	12	12	4.4-7	2.5-7	1.4-4	75-3	1.0+0	5.4-2	1.3-1
60	12	12	5.1-7	2.9-7	3.2-4	4.3-2	5.9+0	2.8-2	1.6-1
320	12	12	5.5-7	3.2-7	6.9-4	7.4-2	28.8+0	1.2-1	1.6-1

In the next set of experiments,we also increase the distance between α_3 and α_4. The results are reported in Table 12.

Table 12.

$\alpha_1 = 1 \quad \alpha_1 = 3 \quad \alpha_4 = 3\alpha_3$

α_3	N_1	N_2	E_r	L_2 Error (I_n)		L_2 Error (C_b)		L_2 Error (M_r)	
				Normal	N. Normal	Normal	N. Normal	Normal	N. Normal
10	10	32	1.6-7	1.3-7	1.3-5	7.7-7	3.6-5	3.8-4	2.5-4
20	10	38	8.4-7	2.9-7	9.7-6	2.0-5	1.1-3	4.2-4	1.5-3
40	10	44	2.2-6	6.0-7	9.8-6	3.6-4	2.2-2	4.9-3	1.3-2
80	10	50	3.9-6	9.6-7	1.5-5	4.2-3	2.8-1	1.9-2	2.5-2

In the last set of experiments, we have negatives eigenvalues as well. In this case, one cannot use the two methods: C_b, M_r.

Table 13.

$$\alpha_1 = 2\alpha_2 \quad \alpha_3 = -\alpha_2 \quad \alpha_4 = -\alpha_1$$

α_1	N_1	N_2	E_r	L_2 Error (I_n)	
				Normal	N. Normal
2	16	16	9.7-7	3.5-7	2.5-5
4	16	16	9.7-7	3.5-7	2.5-5
8	16	16	9.7-7	3.5-7	2.5-5

8. Conclusions

It has been shown that having an a priori knowledge on the distribution of the eigenvalues of a matrix A, it is possible to construct an efficient algorithm for approximating $f(A)$. The more accurate we locate the domain of the e.v., the more efficient is the algorithm. Two methods addressing this problem were described in Section 7. It seems to us that once this problem of finding the domain of the e.v. is solved satisfactorily, the algorithms described in the paper can be used as a numerical tool for many practical problems.

Appendix A.

The set of interpolating points satisfy

$$z_j = \tilde{\psi}(w_j) \qquad j = 1, \ldots, N \tag{A.1}$$

while w_j are roots of the equation

$$w^N = 1 \, . \tag{A.2}$$

The problem is: how to arrange w_j such that the roundoff errors will be minimum.

In a future paper we will carry out a detailed analysis addressing this problem. According to the solution described there, we have

$$w_1, w_2, \ldots, w_N \tag{A.3}$$

equally distributed on the unit circle, where

$$w_1 = 1 \, ; \qquad w_2 = -1$$

and each "new" point w_j is chosen such that the partial set

$$w_1, w_2, \ldots, w_j \tag{A.4}$$

will be as equally distributed as possible. For example, when $N = 12$ we have

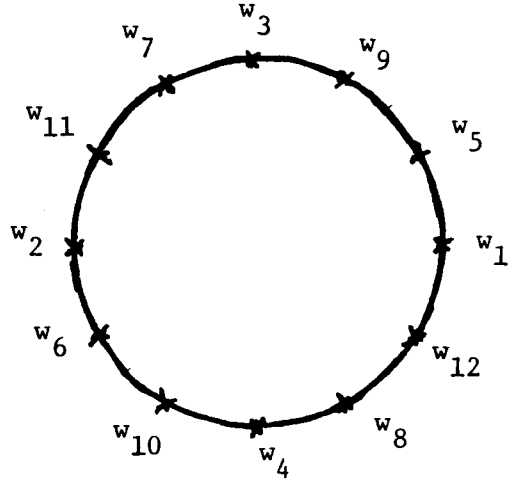


Figure (A.1).

In the case of algorithm (3.27), (A.3) has to be modified as follows

$$w_1, w_2, w_3, w_4, \ldots, w_{2j-1}, w_{2j}, \ldots, w_N \tag{A.5}$$

while

$$w_1 = 1\ ; \qquad w_2 = -1 \tag{A.6}$$

$$w_{2j} = \overline{w}_{2j-1} \qquad 2 \le j \le \frac{N}{2} \tag{A.7}$$

and each "new" pair of points is chosen such that the partial set

$$w_1, w_2, \ldots, w_{2j-1}, w_{2j} \tag{A.8}$$

will be as equally distributed as possible. Thus, for the case $N = 12$ we have

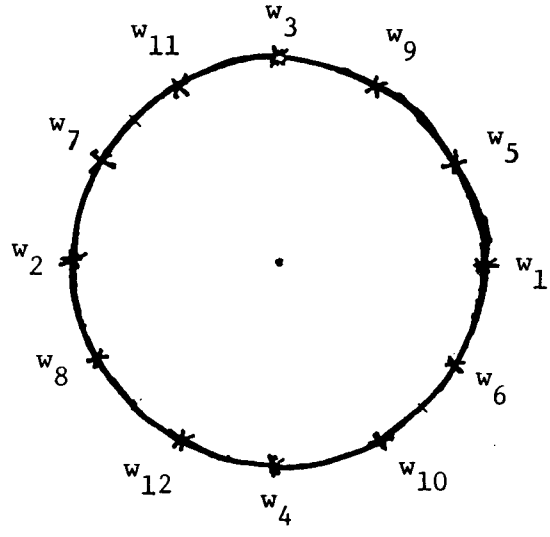


Figure (A.2).

Acknowledge: I would like to thank Prof. Eitan Tadmor and Dr. David Levin for many stimulating discussions. I would also like to thank Prof. Martin Schultz who reinforced my interest in using polynomial approximation in the complex plane for solving general systems of linear equations.

References

1. [Ella83] S.W. Ellacott, Computation of Faber Series With Applications to Numerical Polynomial Approximation in the Complex Plane, Math of Comp. 40 (1983) 162, p. 575-587.

2. [Gant59] F.R. Gantnacher, The Theory of Matrices, Vol. 1, Chelsea Publishing Company, New York, 1959.

3. [GeMa75] K.O. Geddes, J.C. Mason, Polynomial Approximation by Projection on the Unit Circle, SIAM J. Numer. Anal. 12, No. 1, (1975), pp. 111-120.

4. [GoOr81] D. Gottlieb, S. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM Publisher, Philadelphia, PA. (1977).

5. [Gutk86] M.H. Gutknect, An Iterative Method for Solving Linear Equations Based on Minimum Norm Pick-Nevanlinna Interpolation, Private Communication, January, 1986.

6. [KoTa86] R.Kosloff, H. Tal-ezer, A Direct Relaxation Method for Calculating Eigenfucntions and Eigenvalues of the Schrodinger Equation on a Grid, Chemical Physics Letters 127, No. 3 (1986), pp. 223-230.

7. [HaYo81] L. Hayeman, D. Young, Applied Iterative Methods, Academic Press, New York, 1981.

8. [Mant77] T.A. Manteuffel, The Tchebychev Iteration for Nonsymmetric Linear, System Numer. Math. 28 (1977), pp. 307-327.

9. [Mant78] T.A. Manteuffel, Adaptive Procedure for Estimating Parameters for the Nonsymmetric Tchebychev Iteration, Numer. Math. 31 (1978), pp. 183-208.

10. [Mark77] A.I. Markushevich, Theory of Functions of a Complex Variable, Chelsea, New York (1977).

11. [Saad87] Y. Saad, Least Squares Polynomials in the Complex Plane and Their Use for Solving Nonsymetric Linear Systems, SIAM J. Numer. Anal., Vol.

24, No. 1, (1987), pp. 155-169.

12. [SmSa85] D.C. Smolarski, P.E. Saylor, An Optimum Semi- Iterative Method for Solving Any Linear Set With a Square Matrix, Report No. UIUCDCS-R-85-1218, Department of Computer Science, University of Illinois (1985).

13. [SmLe68] V.I. Smirnov, N.A. Lebedov, Functions of a Complex Variable, London ILIFFE Books Ltd. (1968).

14. [Tale86] H. Tal-Ezer, Spectral Methods in Time for Hyperbolic Equations, SIAM Jour. Numer. Anal., 23, No. 1 (1986), pp. 11-26.

15. [Tale85] H. Tal-Ezer, Spectral Methods in Time for Parabolic Problems, ICASE Report 85-9, Langley Research Center, Hampton, Va. (1985).

16. [TaKo84] H. Tal-Ezer, R. Kosloff, An accurate, New and HIghly Efficient Scheme for Propagating the Time-Dependent Schorohinger Equation, Journal of Chem. Phys. (1984), pp. 3967-3971.

17. [Tref80] L.N. Trefethen, Numerical computation of the Schwarz-Christoffel transformation,SIAM J. Sci. Stat. Comput. 1 (1980), 82-102.

18. [Wals56] J.L. Walsh, Interpolation and Approximation by Rational Functions in the Complex Domain, American Mathematical Society, Providence, Rhode Island. 1956.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-178376 ICASE Report No. 87-63		

4. Title and Subtitle
POLYNOMIAL APPROXIMATION OF FUNCTIONS OF MATRICES AND ITS APPLICATION TO THE SOLUTION OF A GENERAL SYSTEM OF LINEAR EQUATIONS

5. Report Date
September 1987

6. Performing Organization Code

7. Author(s)
Hillel Tal-Ezer

8. Performing Organization Report No.
87-63

10. Work Unit No.
505-90-21-01

9. Performing Organization Name and Address
Institute for Computer Applications in Science and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23665-5225

11. Contract or Grant No.
NAS1-18107

13. Type of Report and Period Covered
Contractor Report

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

14. Sponsoring Agency Code

15. Supplementary Notes
Langley Technical Monitor:
Richard W. Barnwell

Final Report

Submitted to SIAM J. Numer. Anal.

16. Abstract

Frequently, during the process of solving a mathematical model numerically, we end up with a need to operate on a vector v by an operator which can be expressed as $f(A)$ while A is $N \times N$ matrix (ex: $\exp(A)$, $\sin(A)$, A^{-1}). Except for very simple matrices, it is impractical to construct the matrix $f(A)$ explicitly. Usually an approximation to it is used. In the present research, we develop an algorithm which uses a polynomial approximation to $f(A)$. It is reduced to a problem of approximating $f(z)$ by a polynomial in z while z belongs to the domain D in the complex plane which includes all the eigenvalues of A. This problem of approximation is approached by interpolating the function $f(z)$ in a certain set of points which is known to have some maximal properties. The approximation thus achieved is "almost best." Implementing the algorithm to some practical problems is described.

Since a solution to a linear system $Ax = b$ is $x = A^{-1}b$, an iterative solution to it can be regarded as a polynomial approximation to $f(A) = A^{-1}$. Implementing our algorithm in this case is described too.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
iterative solution, Richardson's iteration, nonsymmetric matrices, Faber polynomials, polynomial interpolation	64 - Numerical Analysis Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	46	A03

NASA FORM 1626 OCT 86