DOE/ER/25048--32

CONF-8806154--

RECEIVED
NOV 20 1998
OSTI

CENTER FOR NUMERICAL ANALYSIS

THE UNIVERSITY OF TEXAS AT AUSTIN

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

The Search for "High Level" Parallelism for the Iterative Solution of Large Sparse Linear Systems*

by

David M. Young

July 1988 CNA-221

This paper was presented at the Second International Conference on Vector and Parallel Computing which was held at Tromso, Norway on June 6-10, 1988. The paper will appear in the Conference Proceedings.

The Search for "High-Level" Parallelism for the Iterative Solution of Large Sparse Linear Systems*

DAVID M. YOUNG

Center for Numerical Analysis
The University of Texas at Austin
Austin, Texas 78712

1. Introduction

In this paper we are concerned with the numerical solution, based on iterative methods, of large sparse systems of linear algebraic equations of the type which arise in the numerical solution of elliptic and parabolic partial differential equations by finite difference or finite element methods. We consider linear systems of the form

$$Au = b \tag{1.1}$$

where A is a given $N \times N$ matrix which is large and sparse and where b is a given $N \times 1$ column vector. We will assume that A is symmetric and positive definite (SPD). We consider iterative algorithms** for solving (1.1) which consist of a "basic iterative method," such as the Richardson, Jacobi, SSOR or incomplete Cholesky method, combined with an acceleration procedure such as Chebyshev acceleration or conjugate gradient acceleration.

It is often possible to achieve parallelism for iterative algorithms by subdividing the matrix problem into blocks and assigning each processor the task of handling

*The work was supported in part by the Department of Energy, under Grant DE-AS05-81ER10954, and by the National Science Foundation, under Grant MCS-8214731, with The University of Texas at Austin. Some of the work was done during the fall semester of 1987 while the author was visiting the Oak Ridge National Laboratory and the University of Tennessee under the Special Year on Numerical Linear Algebra.

**For a discussion of these procedures, see, e.g., Hageman and Young [5].

one or more blocks. For problems arising from partial differential equations this corresponds to subdividing the region into subregions. Such procedures lead to block iteration procedures and domain decomposition procedures, for example. One can often greatly increase the convergence of iterative algorithms by the use of multigrid techniques wherein one focusses on the use of several grids.

The object of this paper is, however, to examine some "high-level" methods for achieving parallelism. Such techniques involve only matrix/vector operations and do not involve working with blocks of the matrix, subdividing the region, or using different meshes. It is expected that if effective high-level methods could be developed, they could be combined with block and domain decomposition methods, and related methods, to obtain even greater speedups. It is also expected that by working at a higher level it will eventually be possible to develop general purpose software for parallel machines similar to the ITPACK software packages which have already been developed for sequential and vector machines; see Kincaid and Young [8].

Our discussion here is primarily devoted to describing various techniques which we and others have considered for obtaining high-level parallelism. We plan to continue research on these techniques and eventually to develop algorithms and programs for multiprocessors based on them.

In Section 2 we describe some "parallel iteration" techniques. Here several iteration procedures are applied in parallel and the results are combined periodically to yield (hopefully) faster convergence than that produced by any one of the individual procedures used. Similarly in Section 3 we consider "residual decomposition" techniques wherein an initial residual, corresponding to a given starting vector, is decomposed into the sum of several subresiduals. An iterative procedure is then applied to all of the subresiduals in parallel and the results combined to yield (hopefully) faster convergence. If the decomposition of the residual is carried out according to the eigenvalue spectrum of the matrix A then we refer to the procedure

as a "spectral decomposition method."*

For the methods used in both Section 2 and Section 3 several corrections to the initial approximation vector $u^{(0)}$ are obtained. A linear combination of these corrections is used in order to minimize a certain norm of the error. A procedure doing this is described in Section 4. This procedure is related to the *conjugate direction method* which is also described in Section 4 and which is contrasted to the conjugate gradient method.

The discusson of Section 4 is then applied, in Section 5 to the problem of solving a family of linear systems

$$(A + \rho I)u = b \tag{1.2}$$

where the scalar ρ and the vector b may vary. It is shown how, by the use of Arnoldi vectors and the conjugate direction method, only one set of matrix/vector multiplications involving the matrix A is required to solve all of the derived systems.

In Section 6 we consider the time-dependent problem defined by

$$\frac{du(t)}{dt} = -Au(t) + b \tag{1.3}$$

where A is a fixed $N \times N$ SPD matrix and where b is a fixed $N \times 1$ column vector. We consider the use of the backward difference method and the Crank-Nicolson method. Each scheme involves the repeated solution of systems of the form (1.2). Moreover it is shown that by the use of partial fraction representations of rational functions, several time steps can be carried out in parallel provided that the time steps are of different sizes.

It is well-known that there is a close relation between the solution of time-dependent problems of the form (1.3) and "steady state" problems of the form (1.1). This suggests the use of "rational" iteration techniques, described in Section 7. For these techniques we have $\varepsilon^{(n)} = R_n(A)\varepsilon^{(0)}$ where $\varepsilon^{(0)}$ is the initial error vector.

*Originally, this paper was to be primarily devoted to spectral decomposition methods. The title of the paper was originally "Spectral Decomposition Methods for the Numerical Solution of Partial Differential Equations Using Vector and Parallel Processors."

The function $R_n(A)$ is a rational function. For polynomial iteration, also called "polynomial acceleration," $R_n(A)$ is a polynomial in A. Rational iteration often converges very rapidly and has many other desirable properties. However, to carry out each iteration requires the solution of a linear system of the form (1.2), which may be very costly if ρ is small. It is hoped that techniques can be developed, possibly based on the use of the procedures developed in Section 4, to overcome this difficulty.

2. Parallel Iteration

Let us consider the following procedure for solving (1.1). We choose an initial approximation $u^{(0)}$ to the true solution $\bar{u} = A^{-1}b$ of (1.1). If $u^{(0)} \neq 0$ we may replace $u^{(0)}$ by $cu^{(0)}$ where c is a scalar chosen to minimize the error norm

$$\|cu^{(0)} - \bar{u}\|_{A^{1/2}} \tag{2.1}$$

where, in general, the $A^{1/2}$-norm of a vector v is given by

$$\|v\|_{A^{1/2}} = (v, Av)^{1/2} . \tag{2.2}$$

The choice of c to minimize (2.1) is

$$c = \frac{(b, u^{(0)})}{(u^{(0)}, Au^{(0)})} . \tag{2.3}$$

In the following discussion we will assume that $c = 1$.

The idea of parallel iteration is to carry out several iteration procedures starting with $u^{(0)}$, thus obtaining $u^{(1)}, u^{(2)}, \ldots, u^{(s)}$. One then chooses constants $c_1, c_2, \ldots, c_s$ so that $\|u - \bar{u}\|_{A^{1/2}}$ is minimized, where

$$u = \sum_{i=1}^{s} c_i u^{(i)} . \tag{2.4}$$

A procedure for finding the c_i is given in Section 4. Having determined u one can repeat the process, replacing $u^{(0)}$ by u. The hope would be that, by using s iterative methods in parallel, the rate of convergence of the overall procedure would be increased, ideally by a factor close to s.

Adams [1] considered "additive M-step preconditioners." Two iterative procedures were used — one based on the foward SOR method and the other based on the backward SOR method. The resulting iterative process, which could be carried out in parallel, was combined with polynomial preconditioning, [3] and [7]. The results obtained compared favorably with those obtained using the SSOR method. We note that the SSOR method can be regarded as a multiplicative, rather than an additive, preconditioning procedure since it involves a forward SOR iteration followed by a backward SOR iteration.

Very little appears to be known in general about the speedup attainable by parallel iteration. However, O'Leary and White [11] have proved some convergence results for iterative methods based on multi-splittings. Other results are given by Frommer and Mayer [4]. Research is needed to determine whether, for a given problem or class of problems, significant speedups are possible and, if so, how the iterative methods should be chosen.

3. Residual Decomposition

The idea of residual decomposition is somewhat similar to that of parallel iteration. Suppose we are given an initial approximation to $u^{(0)}$, to the true solution $\bar{u} = A^{-1}b$ of (1.1). We decompose the residual $r^{(0)} = b - Au^{(0)}$ into s "subresiduals" $r^{(0,1)}, r^{(0,2)}, \ldots, r^{(0,s)}$ such that

$$r^{(0)} = r^{(0,1)} + r^{(0,2)} + \cdots + r^{(0,s)} . \tag{3.1}$$

We then solve the systems

$$A\Delta^{(i)} = r^{(0,i)} , \qquad i = 1, 2, \ldots, s \tag{3.2}$$

to obtain the corrections $\Delta^{(1)}, \Delta^{(2)}, \ldots, \Delta^{(s)}$. If (3.2) is solved exactly then

$$\bar{u} = A^{-1}b = u^{(0)} + \Delta^{(1)} + \Delta^{(2)} + \cdots + \Delta^{(s)} . \tag{3.3}$$

However, if the $\Delta^{(i)}$ are solved only approximately we choose as our new approximate solution

$$\hat{u} = u^{(0)} + c_1\tilde{\Delta}^{(1)} + c_2\tilde{\Delta}^{(2)} + \cdots + c_s\tilde{\Delta}^{(s)} \tag{3.4}$$

where, for each i, $\tilde{\Delta}^{(i)}$ is an approximate solution of (3.2). We choose $c_1, c_2, \ldots, c_s$ to minimize $\|\hat{u} - \bar{u}\|_{A^{1/2}}$. Procedures for choosing the $\{c_i\}$ are given in the next section.

Spectral Decomposition

Let us now consider the possibility of decomposing $r^{(0)}$ on the basis of a decomposition of the spectrum of the coefficient matrix A of (1.1). As an example, let us consider a decomposition into three parts. As in Figure 3.1 we subdivide the interval $[m(A), M(A)]$, where $m(A)$ and $M(A)$ are, respectively, the smallest and largest eigenvalues of A, into three subintervals, namely, $I_1 = [\alpha_0, \alpha_1]$, $I_2 = [\alpha_1, \alpha_2]$, and $I_3 = [\alpha_2, \alpha_3]$ where $\alpha_0 = m(A)$ and $\alpha_3 = M(A)$. We write the residual $r^{(0)}$ in the form

$$r^{(0)} = \sum_{i=1}^{N} c_i v^{(i)} \tag{3.5}$$

where $v^{(i)}$ is the eigenvector of A associated with the eigenvalue ν_i. We seek to choose $r^{(0,k)}$, $k = 1, 2, 3$ so that if

$$r^{(0,k)} = \sum_{i=1}^{N} c_i^{(k)} v^{(i)} \tag{3.6}$$

then all values of $c_i^{(k)}$ are small except for values of i such that $\nu_i \in I_k$. If $\nu_i \in I_k$ we desire that $c_i^{(k)} = c_i$. The advantage of this is as follows. If we use the conjugate gradient method to solve

$$A\Delta = r^{(0)} \tag{3.7}$$

the number of iterations is of the order of $\sqrt{K(A)}$ where $K(A) = M(A)/m(A)$ is the condition number of A

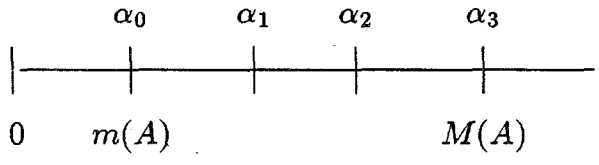

Figure 3.1. Decomposition of the Spectrum of A

On the other hand if we let

$$\alpha_1 = K(A)^{1/3} \quad , \quad \alpha_2 = K(A)^{2/3} \tag{3.8}$$

then we have

$$\frac{\alpha_3}{\alpha_2} = \frac{\alpha_2}{\alpha_1} = \frac{\alpha_1}{\alpha_0} = K(A)^{1/3} \ . \tag{3.9}$$

If we apply the conjugate gradient method to solve each of the systems

$$A\Delta^{(k)} = r^{(0,k)} \ , \tag{3.10}$$

which can be done in parallel, then the number of iterations will be on the order of $K(A)^{1/6}$. This would be true if all of the $c_i^{(k)}$ were to vanish *exactly* for all i such that $\nu_i \notin I_k$. Thus the procedure has considerable potential.

We have developed a program for splitting $r^{(0)}$. This program is based on the construction of orthogonal polynomials using a three term relation and on the determination of the characteristic function of each of the subintervals I_1, I_2 and I_3. By operating on $r^{(0)}$ by each of these characteristic functions, which are polynomials in A, we can get fairly good subresiduals $r^{(0,1)}$, $r^{(0,2)}$ and $r^{(0,3)}$. Unfortunately however, the components $c_i^{(k)}$ are not *exactly* zero outside of the k-th interval I_k. Our numerical experiments indicate that unless the $c_i^{(k)}$ are extremely close to zero outside of I_k, then the number of iterations required is on the order of $K(A)^{1/2}$ rather than $K(A)^{1/6}$. Thus the procedure, as it stands, does not appear to be practical. We are however, continuing our research on the development of practical methods for decomposing $r^{(0)}$.

We note that rather than using residual decomposition we can choose several starting vectors, say, $u^{(0,1)}, u^{(0,2)}, \ldots, u^{(0,s)}$. We can then carry out m steps of a given iterative procedure using each of the starting vectors, obtaining, say $u^{(m,1)}, u^{(m,2)}, \ldots, u^{(m,s)}$. This can be done in parallel. We can then choose scalars $k_1, k_2, \ldots, k_s$ so that $\|u - \bar{u}\|_{A^{1/2}}$ is minimized, where

$$u = \sum_{i=1}^{s} k_i u^{(m,i)} \ . \tag{3.11}$$

For the case where the iterative procedure used is the conjugate gradient method this scheme is related to the block conjugate gradient procedure described by O'Leary, [9], [10].

Research is needed to determine whether a significant speedup can be achieved by the use of several different starting vectors, and, if so, how the starting vectors should be chosen. One possibility for problems arising from partial differential equations might be to let the $\{u^{(0,i)}\}$ correspond to those elements of a hierarchical basis with large support. Alternatively, for the standard five-point finite difference representation of the Dirichlet problem in the unit square, for example, we might choose the comparatively "smooth" functions $\sin \pi x \sin \pi y$, $\sin 2\pi x \sin \pi y$, $\sin \pi x \sin 2\pi y$, etc.

4. Minimization Procedures: The Conjugate Direction Method

Let us assume that we wish to solve the system (1.1), where A is SPD and that we have an initial approximation $u^{(0)}$ to the solution vector $\bar{u} = A^{-1}b$. Let us also assume that we have $s+1$ linearly independent, "direction vectors" $v^{(0)}, v^{(1)}, \ldots, v^{(s)}$. We seek to determine a vector u^* of the form

$$u^* = u^{(0)} + \sum_{i=0}^{s} c_i v^{(i)} \tag{4.1}$$

such that $F(u^*)$ is minimized, where

$$F(u) = \|u - \bar{u}\|^2_{A^{1/2}} = \left(u - \bar{u}, A(u - \bar{u})\right) . \tag{4.2}$$

To minimize $F(u)$ we first construct a set of $s+1$ modified direction vectors $p^{(0)}, p^{(1)}, \ldots, p^{(s)}$ which are mutually A-orthogonal, or "conjugate," in the sense that

$$(p^{(i)}, Ap^{(j)}) = 0 , \qquad i \neq j . \tag{4.3}$$

To do this we use the Gram-Schmidt procedure. We have

$$
(4.4) \qquad \begin{cases} p^{(0)} = v^{(0)} \\ p^{(1)} = v^{(1)} - \dfrac{(v^{(1)}, Ap^{(0)})}{(p^{(0)}, Ap^{(0)})}\, p^{(0)} \\ p^{(2)} = v^{(2)} - \dfrac{(v^{(2)}, Ap^{(0)})}{(p^{(0)}, Ap^{(0)})}\, p^{(0)} - \dfrac{(v^{(2)}, Ap^{(1)})}{(p^{(1)}, Ap^{(1)})}\, p^{(1)} \\ \quad \cdots \end{cases}
$$

We then determine u^* by

$$
(4.5) \qquad u^* = u^{(0)} + \sum_{k=0}^{s} k_i p^{(i)}
$$

where

$$
(4.6) \qquad k_i = \frac{(p^{(i)}, r^{(0)})}{(p^{(i)}, Ap^{(i)})}
$$

and

$$
(4.7) \qquad r^{(0)} = b - Au^{(0)} \ .
$$

This follows since $A(u^{(0)} - \bar{u}) = Au^{(0)} - b = -r^{(0)}$ and since (4.3) holds.

Conjugate Direction Method

Let us again assume that we wish to solve the linear system (1.1) where A is SPD and that we have a set of N linearly independent (direction) vectors $p^{(0)}, p^{(1)}, \ldots, p^{(N-1)}$ which are mutually A-orthogonal. (If we have N linearly independent vectors which are not mutually A-orthogonal then one can, in principle at least, obtain mutually A-orthogonal vectors by the Gram-Schmidt process described above.) The conjugate direction method can be defined by

$$
(4.8) \qquad \begin{cases} u^{(0)} \ \text{ is arbitrary} \\ r^{(0)} = b - Au^{(0)} \\ u^{(n+1)} = u^{(n)} + \lambda_n p^{(n)} \quad , \qquad n = 0, 1, \ldots, N-1 \\ \lambda_n = \dfrac{(p^{(n)}, r^{(0)})}{(p^{(n)}, Ap^{(n)})} \end{cases}
$$

We remark that the conjugate gradient method [6] is a special case of the conjugate direction method where the direction vectors are computed sequentially. Thus for the conjugate gradient method we have

$$
\left\{
\begin{aligned}
& u^{(0)} \text{ is arbitrary} \\
& r^{(0)} = b - Au^{(0)} \\
& p^{(n)} = r^{(n)} + \alpha_n p^{(n-1)} \\
& u^{(n+1)} = u^{(n)} + \lambda_n p^{(n)} \\
& r^{(n)} = b - Au^{(n)} \\
& \alpha_n = \frac{(r^{(n)}, r^{(n)})}{(r^{(n-1)}, r^{(n-1)})} \\
& \lambda_n = \frac{(r^{(n)}, p^{(n)})}{(p^{(n)}, Ap^{(n)})}
\end{aligned}
\right. \tag{4.9}
$$

The direction vectors and residuals are computed in the order $r^{(0)}, p^{(0)}, r^{(1)}, p^{(1)}, \ldots$.

5. The Solution of Related Linear Systems

Let us suppose that we wish to solve a family of linear systems of the form (1.2). We assume that A is a fixed SPD matrix and that the nonnegative constant ρ and the vector b may vary. If all of the ρ's and b's were known in advance the solutions could be obtained in parallel using the conjugate gradient method. However, we assume that we need to solve the systems sequentially so that ρ and b are not known in advance.

We propose the following strategy. We first choose a vector $w^{(0)}$ and a value of ρ, say ρ_0. For a given integer, s, we construct a set of vectors $w^{(1)}, w^{(2)}, \ldots, w^{(s)}$ called "Arnoldi vectors," which span $K_s(w^{(0)}, A+\rho_0 I) = Sp(w^{(0)}, (A+\rho_0 I)w^{(0)}, \ldots, (A+\rho_0 I)^{s-1}w^{(0)})$ and which are mutually orthogonal but not, in general, mutually $(A+\rho_0 I)$-orthogonal. We then show that the $\{w^{(i)}\}$ are independent of ρ_0. Then, for any given ρ we construct a set of direction vectors $p^{(0)}, p^{(1)}, \ldots, p^{(s)}$ which are mutually $(A+\rho I)$-orthogonal. The conjugate direction method is then applied to obtain an approximate solution of (1.2).

The Arnoldi Vectors

Given $w^{(0)}$ and ρ_0 we construct the Arnoldi vectors using the formula

$$w^{(i)} = (A + \rho_0 I)w^{(i-1)} + \beta_{i,i-1}w^{(i-1)} + \beta_{i,i-2}w^{(i-2)} \tag{5.1}$$

where

$$\begin{cases} \beta_{i,i-1} = -\dfrac{\left((A + \rho_0 I)w^{(i-1)}, w^{(i-1)}\right)}{\left(w^{(i-1)}, w^{(i-1)}\right)} \\ \beta_{i,i-2} = -\dfrac{\left((A + \rho_0 I)w^{(i-1)}, w^{(i-2)}\right)}{\left(w^{(i-2)}, w^{(i-2)}\right)} \end{cases} \tag{5.2}$$

It can easily be shown that the $\{w^{(i)}\}$ are mutually orthogonal. It can also be shown that the $\{w^{(i)}\}$ are independent of ρ_0 since we have

$$\begin{cases} \beta_{i,i-1}(\rho_0) = \beta_{i,i-1}(0) - \rho_0 \\ \beta_{i,i-2}(\rho_0) = \beta_{i,i-2}(0) \end{cases} \tag{5.3}$$

It should be noted that only $s_0 + 1$ vectors can be obtained by the process where s_0 is the smallest integer such that the vectors $w^{(i)}, (A + \rho_0 I)w^{(0)}, \ldots, (A + \rho_0 I)^{s_0} w^{(0)}$ are linearly dependent. Evidently $s_0 \leq N - 1$.

The Direction Vectors

Let us now construct the direction vectors $p^{(0)}, p^{(1)}, \ldots, p^{(s)}$ corresponding to a given value of ρ which will differ from ρ_0. We define $p^{(0)}, p^{(1)}, \ldots$ by

$$\begin{cases} p^{(0)} = w^{(0)} \\ p^{(1)} = w^{(1)} + a_1 p^{(0)} \\ p^{(2)} = w^{(2)} + a_2 p^{(1)} \\ \quad \ldots \end{cases} \tag{5.4}$$

where

$$a_n = -\frac{(w^{(n)}, (A + \rho I)p^{(n-1)})}{(p^{(n-1)}, (A + \rho I)p^{(n-1)})} \tag{5.5}$$

We first show that $(w^{(i)}, p^{(j)}) = 0$ for $j < i$. But $p^{(j)}$ is a linear combination of $w^{(0)}, w^{(1)}, \ldots, w^{(j)}$ so that the result follows from the orthogonality of the $\{w^{(i)}\}$.

We next show that $(p^{(n)}, (A+\rho I)p^{(i)}) = 0$ for $i = 0, 1, \ldots, n-1$. This is true for $i = n-1$ by (5.5). For $i \leq n-2$ we have

$$\begin{aligned} (p^{(n)}, (A+\rho I)p^{(i)}) &= (w^{(n)} + a_n p^{(n-1)}, (A+\rho I)p^{(i)}) \\ &= (w^{(n)}, (A+\rho I)p^{(i)}) \end{aligned} \tag{5.6}$$

since $i < n-1$. But $p^{(i)}$ is a linear combination of $w^{(i)}, w^{(i-1)}, \ldots, w^{(0)}$. Hence $(A+\rho I)w^{(i)}$ is a linear combination of $w^{(i+1)}, w^{(i)}, \ldots, w^{(0)}$. Since $i < n-1$ the result follows from the orthogonality of the $\{\omega^{(i)}\}$.

Suppose now that we wish to solve a specific linear system (1.2) and that we have already computed and stored the Arnoldi vectors as well as the $\{(A+\rho_0 I)w^{(i)}\}$. We use the conjugate direction method with direction vectors given by (5.4). We show that this does not require any additional matrix/vector multiplication involving A. This is possible since by (5.4) we have

$$(A+\rho I)p^{(i)} = (A+\rho I)w^{(i)} + (A+\rho I)a_i p^{(i-1)} \ . \tag{5.7}$$

Thus we can compute $(A+\rho I)p^{(i)}, (A+\rho I)p^{(1)}$, etc. recursively using $(A+\rho I)w^{(0)}, (A+\rho I)w^{(1)}$, etc. (We note that $(A+\rho I)w^{(i)} = (A+\rho_0 I)w^{(i)} + (\rho - \rho_0)w^{(i)}$.)

For a given initial approximation $u^{(0)}$ to the true solution $\bar{u}$ we compute $r^{(0)} = b - (A+\rho I)u^{(i)}$ and $u^{(1)}, u^{(2)}, \ldots$ by

$$u^{(n+1)} = u^{(n)} + \lambda_n p^{(n)} \ , \qquad n = 0, 1, 2, \ldots \tag{5.8}$$

where

$$\lambda_n = \frac{(p^{(n)}, r^{(0)})}{(p^{(n)}, (A+\rho I)p^{(n)})} \tag{5.9}$$

We remark that the above scheme may break down if only s_0 linearly independent Arnoldi vectors $\{w^{(i)}\}$ can be generated and if the conjugate direction method does not yield sufficient accuracy within s_0 iterations. If this happens one could generate a new set of Arnoldi vectors based on $r^{(0)} = b - (A+\rho I)u^{(0)}$. However, the savings of the matrix/vector multiplication by $A+\rho I$ would be lost.

6. Time Dependent Problems

Let us now consider the time dependent problem (1.3). Such a problem arises, for example, from the standard five-point difference equation representation, with respect to the space variables, of a problem involving the diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x,y) \tag{6.1}$$

over a rectangle where the values of u are given and fixed for all t on the boundary of the rectangle.

We consider two alternative descretizations for solving (1.3). The first corresponds to the so-called *backward difference method* defined by

$$\frac{u(t+\Delta t) - u(t)}{\Delta t} = -Au(t+\Delta t) + b \tag{6.2}$$

or

$$(A + \rho I)u(t+\Delta t) = \rho u(t) + b \tag{6.3}$$

where

$$\rho = \frac{1}{\Delta t}\,. \tag{6.4}$$

The second corresponds to the *Crank-Nicolson method* defined by

$$\frac{u(t+\Delta t) - u(t)}{\Delta t} = -A\left\{\frac{u(t+\Delta t) + u(t)}{2}\right\} + b \tag{6.5}$$

or

$$(A + \rho I)u(t+\Delta t) = 2b - (A - \rho I)u(t) \tag{6.6}$$

where

$$\rho = \frac{2}{\Delta t}\,. \tag{6.7}$$

Let us now focus on the backward difference equation (6.2). Suppose we use the time steps $(\Delta t)_1$ and $(\Delta t)_2$ where $(\Delta t)_1 \neq (\Delta t)_2$. We have by (6.3)

$$u\big(t+(\Delta t)_1+(\Delta t)_2\big) = (A+\rho_2 I)^{-1}(A+\rho_1 I)^{-1}\{\rho_1\rho_2 u(t)+\rho_2 b\} + (A+\rho_2 I)^{-1}b\ . \tag{6.8}$$

We now consider the partial fraction representation of $(x+\rho_2)^{-1}(x+\rho_1)^{-1}$, which is given by

$$\frac{1}{(x+\rho_2)(x+\rho_1)} = \frac{1}{\rho_1-\rho_2}\Big\{\frac{1}{x+\rho_2}-\frac{1}{x+\rho_1}\Big\}\ . \tag{6.9}$$

Thus we have

(6.10)

$$u\big(t+(\Delta t)_1+(\Delta t)_2\big) = \frac{1}{\rho_1-\rho_2}\big\{(A+\rho_2 I)^{-1}-(A+\rho_1 I)^{-1}\big\}\rho_1\rho_2 u(t)+\rho_2 b\} + (A+\rho_2 I)^{-1}b\ .$$

Evidently to carry out the above process we have to solve two systems of the from (1.2) for two sets of values of ρ and b. These systems can be solved in parallel using the techniques of the previous section. The idea can be extended to allow for several time steps $(\Delta t)_1, (\Delta t)_2, \ldots, (\Delta t)_s$ provided that the $\{(\Delta t)_i\}$ are distinct.

We remark that the idea of using partial fractions for the parallel solution of a system of the form

$$\prod_{i=1}^{s}(A+\rho_i I)x = y \tag{6.11}$$

has been used by Sweet [**13**] in connection with the cyclic reduction procedure.

7. Rational Iteration

Since A is SPD, the solution $u(t)$ of (1.3) converges to the steady state solution $\bar{u} = A^{-1}b$ as $t \to \infty$. Moreover, we can regard the time dependent schemes considered in the previous section as iterative procedures for solving the linear system (1.1). Thus, corresponding to the backward difference method and the Crank-Nicolsen method, respectively, we have the iterative methods

$$(A + \rho I)u^{(n+1)} = \rho u^{(n)} + b \tag{7.1}$$

and

$$(A + \rho I)u^{(n+1)} = -(A - \rho I)u^{(n)} + 2b\ . \tag{7.2}$$

These two "rational" iterative methods correspond to the matrix splittings

$$A = (A + \rho I) - \rho I \tag{7.3}$$

and

$$A = \tfrac{1}{2}(A + \rho I) - \left[-\tfrac{1}{2}(A - \rho I)\right] \tag{7.4}$$

respectively.

One can also consider non-stationary iterative methods where ρ varies. Thus, for example, one could apply (7.1), first with ρ_1 and then with ρ_2 obtaining

$$(A + \rho_1 I)u^{(n+1)} = \rho_1 u^{(n)} + b \tag{7.5}$$

and

$$(A + \rho_2 I)u^{(n+2)} = \rho_2 u^{(n+1)} + b\ . \tag{7.6}$$

From this it follows that

$$u^{(n+2)} = (A + \rho_2 I)^{-1}(A + \rho_1 I)^{-1}(\rho_1\rho_2 u^{(n)} + \rho_2 b) + (A + \rho_2 I)^{-1}b \tag{7.7}$$

and

$$\begin{aligned} u^{(n+2)} - \bar{u} &= (A + \rho_2 I)^{-1}(A + \rho_1 I)^{-1}\rho_1\rho_2(u^{(n)} - \bar{u}) \\ &= R(A)(u^{(n)} - \bar{u}) \end{aligned} \tag{7.8}$$

where $R(x)$ is the rational function

$$R(x) = \frac{\rho_1 \rho_2}{(x+\rho_1)(x+\rho_2)} \,. \tag{7.9}$$

Since $R(x)$ is a rational function we refer to the above procedure as "rational iteration." In the case of polynomial acceleration $R(x)$ would be a polynomial.

At this point we note that the ordinary extrapolated Richardson's method can be derived by applying the *forward difference method* to (1.3). Thus we have

$$\frac{u(t+\Delta t) - u(t)}{\Delta t} = -Au(t) + b \tag{7.10}$$

or

$$u(t+\Delta t) = \big(I - (\Delta t)A\big)u(t) + b\Delta t \,. \tag{7.11}$$

This corresponds to the extrapolated Richardson's method defined by

$$\begin{aligned} u^{(n+1)} &= (I - \gamma A)u^{(n)} + \gamma b \\ &= G_{[\gamma]}u^{(n)} + b_{[\gamma]} \end{aligned} \tag{7.12}$$

where $\gamma = \Delta t$ is the *extrapolation factor*. If the eigenvalues ν of A lie in the range $0 < m(A) \leqq \nu \leqq M(A)$ then the optimum extrapolation factor γ^* is given by

$$\gamma^* = \frac{2}{M(A) + m(A)} \tag{7.13}$$

and the corresponding spectral radius of $G_{[\gamma^*]}$ is

$$S(G_{[\gamma^*]}) = \frac{M(A) - m(A)}{M(A) + m(A)} = \frac{K(A) - 1}{K(A) + 1} \tag{7.14}$$

where $K(A) = M(A)/m(A)$ is the *condition number* of A. The number of iterations needed for convergence using the extrapolated Richardson's method is asymptotically proportional to $K(A)$. If conjugate gradient acceleration is used the number of iterations is asymptotically proportional to $K(A)^{1/2}$.

In Figure 7.1 we plot the eigenvalues of the extrapolated Richardson's method, $(\lambda_1 = 1 - \gamma\nu)$, the eigenvalues of the iterative method of (7.1), $(\lambda_2 = \rho/(\rho + \nu))$ and the eigenvalues of the iterative method of (7.3), $(\lambda_3 = (1 - \nu/\rho)/(1 + \nu/\rho))$. It should be noted that λ_1 vanishes for $\nu = \gamma^{-1}$ but if γ is very large $|\lambda_1|$ is greater than one for large ν. On the other hand, λ_3 vanishes for $\nu = \rho$ but $|\lambda_3| \leqq 1$ for all ν in the range $m(A) \leqq \nu \leqq M(A)$. It should be noted that λ_2 is a positive monotone decreasing function of ν for $\nu \geqq 0$ and that $\lambda_2(0) = 1$.

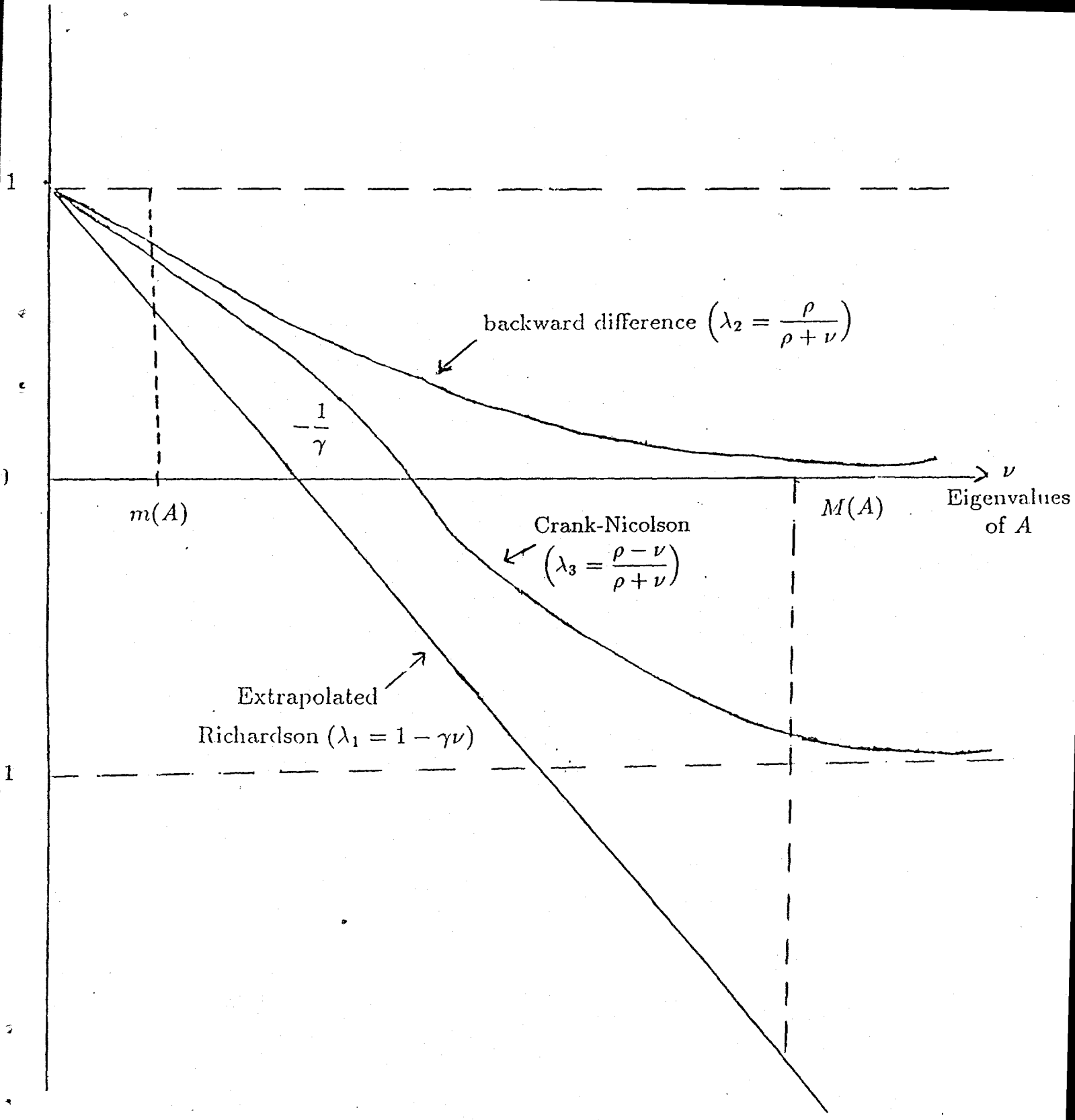

Figure 7.1. Eigenvalues of Iterative Procedures

Let us now consider the iterative method defined by s iterations of (7.3) with variable ρ. The eigenvalues λ of the method are given by

$$\lambda = \frac{\rho_1 - \nu}{\rho_2 + \nu} \frac{\rho_2 - \nu}{\rho_2 + \nu} \cdots \frac{\rho_s - \nu}{\rho_s + \nu} .$$

The rational function given above is the same as that frequently used in the analysis of the Peaceman-Rachford [**12**] alternating direction implicit scheme, see *e.g.*, Birkhoff, Varga and Young [**2**]. It can be shown that for a suitable choice of the $\{\rho_i\}$ the number of iterations required for convergence is asymptotically proportional to $\log K(A)$.

Let us consider the case where the linear system is derived from the standard five-point finite difference representation of the Poisson equation $u_{xx}+u_{yy} = f(x,y)$ in the unit square $0 \leqq x \leqq 1, 0 \leqq y \leqq 1$ with Dirichlet boundary conditions. In this case the eigenvalues ν of A lie in the range

$$m(A) = 8 \sin^2 \frac{\pi h}{2} \leqq \nu \leqq 8 \cos^2 \frac{\pi h}{2} = M(A) \tag{7.15}$$

where h is the mesh size. The condition number $K(A)$ of A is given by

$$K(A) = \frac{M(A)}{m(A)} = \cot^2 \frac{\pi h}{2} \approx \frac{4}{\pi^2 h^2} = O(h^{-2}) . \tag{7.16}$$

Thus using the extrapolated Richardson's method the number of iterations is $O(h^{-2})$ whereas if conjugate gradient acceleration is used the number of iterations is $O(h^{-1})$. Using the Peaceman-Rachford scheme with good parameters the number of iterations is $O(\log h^{-1})$.

Evidently rational iteration has many attractive properties as compared with polynomial acceleration. Unfortunately there is one serious drawback, namely, the amount of work needed to carry out each iteration. Thus if ρ is very small, the amount of work required to solve a system of the form

$$(A + \rho I)x = y \tag{7.17}$$

for x, given y, may be comparable with that needed to solve the original system (1.1).

As an example, consider the use of a single value of ρ. It can be shown, see, *e.g.*, Birkhoff, Varga and Young [**2**] that the optimum single value of ρ is given by

$$\rho^* = \sqrt{M(A)m(A)} \,. \tag{7.18}$$

For the model problem $\rho^* = 4 \sin \pi h = O(h)$. It can be shown that for the model problem defined above if conjugate gradient acceleration is applied, the number of iterations is $O(h^{-1/2})$ instead of $O(h^{-1})$ as with the Richardson's method with conjugate gradient acceleration. However, the number of iterations needed to solve* each system of the form

$$(A + \rho^* I)x = y \tag{7.19}$$

for x, given y, is also $O(h^{-1/2})$. Thus with $O(h^{-1/2})$ systems each taking $O(h^{-1/2})$ iterations we again have $O(h^{-1})$ for the overall process.

We are now investigating several procedures for overcoming this difficulty. One possibility for solving the system $(A+\rho I)x = y$ for x, given y, would be to do so for a relatively large value of ρ and then to let ρ decrease using some kind of continuation method. Another procedure would be a sort of nesting procedure involving a set of increasing values of ρ. Whatever procedure is used would involve the solution of many systems of the form $(A + \rho I)x = y$ where ρ and y vary. It is hoped that the procedures given in Section 4 will reduce the time needed to solve each system to the point that the overall scheme will become practical.

Bibliography

1. Adams, Loyce M. [1985], "Additive M-Step Preconditioners", Technical Report 85-6, Department of Applied Mathematics, University of Washington, Seattle, Washington.
2. Birkhoff, G., Varga, R.S., and Young, D.M. [1962], "Alternating Direction Implicit Methods", *Advances in Computers*, **3**, 189–273.
3. Dubois, P., Greenbaum, A., and Rodrique, G. [1978], "Approximating the Inverse of a Matrix for Use in Iterative Algorithms on Vector Processors", *Computing*, **22**, 257–268.
4. Frommer, Andreas and Mayer, Gunter [1988], "Convergence of Relaxed Parallel Multisplitting Methods", unpublished manuscript.

*Here conjugate gradient acceleration would be applied to Richardson's method. The formulas are given in (4.9).

5. Hageman, L.A. and Young, D.M. [1981], *Applied Iterative Methods*, Academic Press, New York.
6. Hestenes, M.R. and Stiefel, E.L. [1952], "Methods of Conjugate Gradients for Solving Linear Systems", *J. Res. Nat. Bur. Standards*, **49**, 409–436.
7. Johnson, O., Micchelli, C. and Paul, G. [1983], "Polynomial Preconditioners for Conjugate Gradient Calculations", *SIAM J. Numer. Anal.*, **20**, 362–376.
8. Kincaid, D.R. and Young, D.M. [1988], "A Review of the ITPACK Project", Report CNA-217, Center for Numerical Analysis, The University of Texas, Austin, Texas.
9. O'Leary, Dianne P. [1980], "The Block Conjugate Gradient Algorithm and Related Methods", *Linear Algebra Appl.*, **29**, 293–322.
10. O'Leary, Dianne P. [1987], "Parallel Implementation of the Block Conjugate Gradient Algorithm", *Parallel Computing*, **5**, 127–139.
11. O'Leary, Dianne P. and White, R.E. [1985], "Multi-splittings of Matrices and Parallel Solution of Linear Systems", *SIAM J. Alg. Disc. Meth.*, **6**, 630–640.
12. Peaceman, D.W. and Rachford, H.H., Jr. [1955], "The Numerical Solution of Parabolic and Elliptic Differential Equations", *J. Soc. Indus. Appl. Math.*, **3**, 28–41.
13. Sweet, R. [1988], "A Parallel and Vector Variant of the Cyclic Reduction Algorithm", to appear in the *SIAM Journal of Scientific and Statistical Computing.*