The Modified Conjugate Residual Method for Partial Differential Equations

R. Chandra, S. C. Eisenstat, and M. H. Schultz

Research Report #107

May 1977

This research was supported by the Office of Naval Research under Grant N00014-76-C-0277, and Air Force Grant F49620-77-C-0037.

1. Introduction

In this paper we present the Modified Conjugate Residual (MCR) Method, a stabilized version of Luenberger's Method of Conjugate Residuals [6], for solving large sparse systems of linear equations. This iterative method has special significance when the system is not positive definite so that methods like Conjugate Gradients [2] are inapplicable. In the special case when the system is positive definite, MCR reduces to one of the family of general conjugate gradient methods discussed by Hestenes [4]. In section 2 we present the MCR method and its properties, in section 3 we derive general error bounds, and in section 4 we present an efficient computational version of the MCR algorithm. In section 5 we introduce a model indefinite problem, simple finite difference approximations to the forced vibration problem in two and three dimensions and apply the convergence results of section 3 to obtain error bounds. Although these problems are very specialized, nearly all our results are applicable to the solution of the linear systems arising from the application of finite difference or finite element methods to more general self-adjoint second order elliptic partial differential equations in more general domains. Finally in section 6 we describe the results of numerical experiments on the model problem. Proofs of the theoretical results and further details will appear in [1].

2. The Modified Conjugate Residual (MCR) Method

Consider the NxN linear system

(2.1) $$Ax = f$$

where A is non-singular and symmetric. To approximate x, the MCR method generates a sequence of direction vectors p_i which are mutually A^2-orthogonal, and computes the iterates $x_0, x_1, x_2, \ldots$ by moving at the $(i+1)^{st}$ step in the direction of p_i so that $x_{i+1} = x_i + a_i p_i$, where a_i is chosen to minimize the error functional

(2.2) $$E(a_i) = (f-Ax_{i+1},\ f-Ax_{i+1}).$$

Due to the special way in which the direction vectors are chosen the uni-directional minimizations correspond to minimization in the whole subspace spanned by the p_i's, so that the x_{i+1} obtained actually minimizes E(x) on the affine subspace $x_0 + \{p_0, p_1, p_2, \ldots, p_i\}$, where $\{\ldots\}$ denotes the subspace spanned by the vectors enclosed in the brackets.

In particular, we use the following method, based on the Lanczos algorithm [7] for generating the direction vectors:

Choose p_0, define $p_{-1} = 0$, and for i=0,1,..., let

(2.3a) $$p_{i+1} = Ap_i - \gamma_i p_i - \delta_i p_{i-1},$$

where

$$\gamma_i = (A^2 p_i, Ap_i)/(Ap_i, Ap_i) \qquad \text{and}$$

(2.3b)

$$\delta_i = (A^2 p_i, Ap_{i-1})/Ap_{i-1}, Ap_{i-1}).$$

We can show that

$$(p_i, A^2 p_j) = 0, \quad i \neq j. \tag{2.4}$$

We now consider the following scheme for computing the approximations x_i to x:

Given an arbitrary initial guess x_0 to x, compute $r_0 = f - Ax_0$, set $p_0 = r_0$ and for i = 0,1,2,... compute

$$\begin{aligned} a_i &= (r_i, Ap_i)/(Ap_i, Ap_i) \\ x_{i+1} &= x_i + a_i p_i \\ r_{i+1} &= r_i - a_i Ap_i \end{aligned} \tag{2.5}$$

and generate p_{i+1} by (2.3).

In section 3 we show that the above procedure can be made more efficient in certain cases. However, for now, we refer to (2.5) as the MCR algorithm and proceed to outline its properties and convergence results.

We can show that the following relations hold for the MCR method of (2.5):

$$Ap_i \in \{p_0, p_1, \ldots, p_{i+1}\} \tag{2.6a}$$

$$r_i \in \{p_0, p_1, \ldots, p_i\} \tag{2.6b}$$

$$\{p_0, p_1, \ldots, p_i\} = \{p_0, Ap_0, \ldots, A^i p_0\} = \{r_0, Ar_0, \ldots, A^i r_0\} \tag{2.6c}$$

$$(r_i, Ap_j) = 0, \quad j < i, \tag{2.6d}$$

$$(r_j, Ar_j) = 0, \quad i \neq j, \tag{2.6e}$$

$$(r_j, Ap_i) = (r_0, Ap_i), \quad j \leq i, \tag{2.6f}$$

$$\text{and } r_i \neq \underline{0} \Rightarrow p_i \neq \underline{0} \tag{2.6g}$$

($\underline{0}$ denotes the zero vector)

We also have the result:

Theorem 2.1: For each i, x_{i+1} minimizes E(x) over the affine subspace $x_0 + \{p_0, p_1, \ldots, p_i\}$.

3. Error Bounds

The fundamental minimization property of Theorem 2.1 allows us to view the MCR method as one in which the error functional is minimized over subspaces of increasing dimension, and thereby to prove that it converges in at most N steps to the unique solution of $Ax = f$. Moreover, we can obtain error bounds by a procedure analogous to the one used for Conjugate Gradients [3].

Using Theorem 2.1 and property (2.6c) we can write

$$x_{i+1} = x_0 + \sum_{j=0}^{i} s_j A^j r_0 \text{ where } \{s_j\}_{j=o}^{i} \text{ are scalars}$$

$$= x_0 + P_i(A)\, r_0 \text{ where } P_i(A) \text{ is a polynomial of degree at most i in A.}$$

Then,

$$x - x_{i+1} = (I - P_i(A)A)\quad (x - x_0)$$

and

$$E(x - x_{i+1}) = (f - Ax_{i+1},\ f - Ax_{i+1}) \tag{3.1}$$

$$= (R_{i+1}(A)r_0, R_{i+1}(A)r_0)$$

where

$$R_{i+1}(A) = I - P_i(A)A.$$

Let us define $\bar{R}_m$ to be the set of polynomials $R_m(y)$ of degree at most m in y satisfying $R_m(0) = 1$. Then, due to (3.1) and Theorem 2.1, we can visualize MCR as a method that chooses, at the $(i+1)^{st}$ iteration, the particular polynomial $R_m \in \bar{R}_m$ that minimizes the functional E(x), i.e.,

$$E(x_{i+1}) = \min_{R_{i+1} \in \bar{R}_{i+1}} (R_{i+1}(A)r_0, R_{i+1}(A)r_0)$$

Since A is symmetric, there exist orthonormal eigenvectors v_j, $j = 1,2,\ldots,N$ of A such that

$$Av_j = \lambda_j v_j, \quad j = 1,2,\ldots,N,$$

where $\{\lambda_j\}_{j=1}^N$ are the eigenvalues of A.

We have $r_0 = \sum_{j=1}^N t_j v_j$ for some scalars $\{t_j\}_{j=1}^N$.

Then
$$R_{i+1}(A)r_0 = \sum_{j=1}^N t_j R_{i+1}(A)v_j = \sum_{j=1}^N t_j R_{i+1}(\lambda_j)v_j$$

and
$$E(x_{i+1}) = \min_{R_{i+1} \in \bar{R}_{i+1}} \left(\sum_{j=1}^N t_j R_{i+1}(\lambda_j)v_j, \sum_{j=1}^N t_j R_{i+1}(\lambda_j)v_j \right)$$

$$= \min_{R_{i+1} \in \bar{R}_{i+1}} \sum_{j=1}^N t_j^2 R_{i+1}^2(\lambda_j)$$

$$\leq \left(\min_{R_{i+1} \in \bar{R}_{i+1}} \max_{1 \leq j \leq N} |R_{i+1}(\lambda_j)| \right)^2 \sum_{j=1}^N t_j^2$$

$$= Q_{i+1}^2 E(x_0), \tag{3.2}$$

where $Q_{i+1} = \min_{R_{ix1} \in \bar{R}_{i+1}} \max_{1 \leq j \leq N} |R_{i+1}(\lambda_j)|$.

Depending upon the spectrum of A and what is known about it, we can use different techniques to get bounds on Q_{i+1}. We consider two approaches. First, we treat the case in which at most a few eigenvalues lie on one side of the origin and we know the end-points of the interval containing the eigenvalues on the other, for example, in the solution of difference approximations to the forced vibration problem. For this case we have the following theorem:

Theorem 3.1: If the eigenvalues of the matrix A lie in $\{\lambda_1,\lambda_2,\ldots,\lambda_m\} \cup [a,b]$ where $1 \leq m < N$, $a, b>0$, and $\lambda_i < 0$

for $1 \leq i \leq m$, then for $k \geq 0$ the iterates x_{k+m} given by MCR satisfy

$$(3.3) \qquad \| f - Ax_{k+m} \|_2 \leq 2C\left(\frac{1 - \sqrt{\alpha}}{1 + \sqrt{\alpha}}\right)^k \| f - Ax_0 \|_2$$

and

$$(3.4) \qquad \| x - x_{k+m} \|_2 \leq 2C \sqrt{\kappa}(A^2) \left(\frac{1 - \sqrt{\alpha}}{1 + \sqrt{\alpha}}\right)^k \| x - x_0 \|_2$$

where $\alpha = a/b$, $C = \prod_{j=1}^{m} \left(\frac{\lambda_j - b}{\lambda_j}\right)$ is a constant independent of k, and $\kappa(A^2)$ denotes the condition number of A^2.

If the matrix A is positive definite, (3.3) and (3.4) hold with $C = 1$, and we have the same rate of convergence for Modified Conjugate Residuals as for Conjugate Gradients (cf. [3]).

In the second approach we assume that all we know is the end-points of the two intervals, one on either side of the origin, that contain the eigenvalues of A. Using Lebedev's results[5] we can show the following result.

Theorem 3.2: If the eigenvalues of the matrix A lie in $[-a,-b] \quad [c,d]$ where $a,b,c,d > 0$ and $a - b = d - c > 0$ then for $i \geq 0$ the iterates x_{i+1} given by MCR satisfy

$$\| f - Ax_{i+1} \|_2 \leq 2\left(\frac{1 - \sqrt{\beta}}{1 + \sqrt{\beta}}\right)^{\left\lfloor \frac{i+1}{2} \right\rfloor} \| f - Ax_0 \|_2$$

and

$$\| x - x_{i+1} \|_2 \leq 2 \sqrt{\kappa}(A^2) \left(\frac{1 - \sqrt{\beta}}{1 + \sqrt{\beta}}\right)^{\left\lfloor \frac{i+1}{2} \right\rfloor} \| x - x_0 \|_2$$

where $\beta = (bc)/(ad)$ and $\kappa(A^2)$ denotes the condition number of A^2.

We note that Theorem 3.2 requires that the two intervals on either side of the origin, known to contain the eigenvalues of A be of equal length. If we have that $a - b \neq d - c$ we can take the smaller of the two intervals and extend it away from the origin to get intervals of equal length on either side of the origin and then apply Theorem 3.2 to obtain bounds on the rate of convergence.

4. The MCR Algorithm - Computational Version

In this section we show that in certain cases we can reduce the cost of an iteration of the MCR method by using a less expensive process than (2.3) for generating the new direction vector. We present the computational version of the MCR method in its reduced form for both positive definite A and general symmetric A. The algorithms are numerically stable, easy to program and require very little storage and work per iteration.

In particular, the following identity holds.

Theorem 4.1: If $a_i \neq 0$,

(4.1a) $$\tilde{p}_{i+1} = r_{i+1} + b_i p_i,$$

where

(4.1b) $$b_i = (-Ar_{i+1}, Ap_i)/(Ap_i, Ap_i),$$

and p_{i+1} is given by relations (2.3), then

$$\tilde{p}_{i+1} = -a_i p_{i+1}.$$

The above theorem shows that if $a_i \neq 0$, equations (4.1) generate the same direction as the more expensive (2.3). Only the normalization is different, but since that does not effect the validity of orthogonality relations, it is clear that all the results of sections 2 and 3 hold.

In the special case of positive definite A, we can show (cf. [1]) that a_i is never zero and that (4.1) can be used to generate the direction vectors at every iteration. We also use alternative formulae for a_i and b_i summarized in (4.2):

MCR algorithm (A positive definite):

Choose x_0, compute $r_0 = f - Ax_0$, set $p_0 = r_0$ and for $i = 0,1,2,\ldots$ compute

(4.2)
$$a_i = (r_i, Ar_i)/(Ap_i, Ap_i)$$
$$x_{i+1} = x_i + a_i p_i$$
$$r_{i+1} = r_i - a_i Ap_i$$
$$b_i = (r_{i+1}, Ar_{i+1})/(r_i, Ar_i)$$
$$p_{i+1} = r_{i+1} + b_i p_i$$

The number of multiplications per iteration for the above procedure is 5N + 2 plus a matrix-vector product, viz. Ar_{i+1}; Ap_{i+1} is computed as

$$Ap_{i+1} = Ar_{i+1} + b_i Ap_i.$$

Storage requirements are 5N plus whatever is required to store the upper triangular part of A.

For general symmetric A, we could have a_i equal to zero in which case the longer iteration (2.3) would have to be used. However, each time that $a_i \neq 0$ we could use the shorter iteration and save some work. In practice, though, due to finite precision and roundoff limitations, it may not be possible to decide whether a_i is zero or not. We can show that if $a_i = 0$ and we use the shorter version (4.1) to generate the direction vector, the algorithm may get tuck at a point and serious instabilities may occur. Therefore, we decide the use the shorter version only when $|a_i| > \varepsilon$ where $\varepsilon > 0$ is some threshold chosen a priori and large enough such that roundoff and precision will not cause us to choose (4.1) when $a_i = 0$. We note, however, that if $0 < a_i < \varepsilon$ we are using the longer iteration at step i, even though the shorter one

is sufficient.

We can also show that,

$$\delta_i = c_i(Ap_i, Ap_i)/(Ap_{i-1}, Ap_{i-1})$$

where $$c_i = \begin{cases} 1 \text{ if } |a_{i-1}| \leq \varepsilon \\ -1/a_{i-1} \text{ if } |a_{i-1}| > \varepsilon \end{cases}$$

We thus have the following computational version of the Modified Conjugate Residual Method for any symmetric indefinite linear system $Ax = f$:

MCR algorithm (A symmetric):

Choose x_0, and $\varepsilon > 0$; compute $r_0 = f - Ax_0$, set $p_0 = r_0$ and for $i = 0,1,2, \ldots$ compute

$$a_i = (r_i, Ap_i)/(Ap_i, Ap_i)$$
$$x_{i+1} = x_i + a_i p_i$$
$$r_{i+1} = r_i - a_i Ap_i$$

If $|a_i| \leq \varepsilon$,

(4.3)
$$p_{i+1} = Ap_i - \gamma_i p_i - \delta_i p_{i-1}$$
where $\gamma_i = (A^2 p_i, Ap_i)/(Ap_i, Ap_i)$
and $\delta_i = c_i(Ap_i, Ap_i)/(Ap_{i-1}, Ap_{i-1})$

$$\text{where } c_i = \begin{cases} 1 \text{ if } |a_{i-1}| \leq \varepsilon \\ -1/a_{i-1} \text{ if } |a_{i-1}| > \varepsilon \end{cases}$$

If $|a_i| > \varepsilon$,

$$p_{i+1} = r_{i+1} + b_i p_i$$
where $b_i = (-Ar_{i+1}, Ap_i)/(Ap_i, Ap_i)$

Each iteration of (4.3) requires only one matrix-vector product. To see this, suppose that we have already computed and stored the vectors x_i, r_i, p_i, Ap_i, p_{i-1} and Ap_{i-1} and the scalar (Ap_{i-1}, Ap_{i-1}). If $|a_i| \leq \varepsilon$ the only matrix-vector product performed at the $(i+1)^{st}$ iteration is $A(Ap_i)$ required in the computation of Ap_{i+1} which is done as follows:

$$Ap_{i+1} = A(Ap_i) - \gamma_i Ap_i - \delta_i Ap_{i-1}$$

If $|a_i| > \varepsilon$, then Ar_{i+1} is computed and Ap_{i+1} is obtained as

$$Ap_{i+1} = Ar_{i+1} + b_i Ap_i.$$

It is easy to see that in addition to the matrix-vector product, the iteration requires $7N + 2$ multiplications if $|a_i| > \varepsilon$ and $9N + 4$ multiplications of $|a_i| \leq \varepsilon$.

Storage is required for the six vectors x_i, r_i, p_i, Ap_i, p_{i-1} and Ap_{i-1}, each of which gets overwritten by its successor at the $(i+1)^{st}$ iteration, and a seventh vector that stores $A^2 p_i$ if $|a_i| \leq \varepsilon$ and Ar_{i+1} if $|a_i| > \varepsilon$. The matrix A is not modified during the iteration process. Since it is symmetric, only the upper (or lower) triangle may be stored. If A is sparse, sparse storage schemes may be used.

We note here the close connection between MCR and Luenberger's method of Conjugate Residuals [6]. Indeed, taking $\varepsilon = 0$ we can rewrite the algorithm (4.3) to give Luenberger's method. In this case, at iteration $k + 1$ we are required to answer the question -- "Is $a_k = 0$?" As pointed out earlier, this decision is not easy in the presence of

roundoff error so that Luenberger's method suffers from certain unresolved computational difficulties. By choosing ϵ sufficiently large we have the MCR method which is free from these problems.

5. The Model Problems

In this section we introduce model problems in two and three dimensions-- simple finite difference approximations to indefinite self-adjoint second order elliptic partial differential equations in a square and cube respectively (the forced vibration problem).

In two dimensions we consider the problem:

(5.1a) $$-\Delta w(x,y) - \sigma w(x,y) = g(x,y), \quad (x,y) \in D \equiv (0,1) \times (0,1)$$

with Dirichlet boundary conditions

(5.1b) $$w(x,y) = 0, \quad (x,y) \in \delta D$$

where σ is a constant.

To approximate the solution to this problem, we cover the domain D with a uniform mesh with mesh-width $h \equiv 1/(n+1)$ and seek a mesh function $W(i,j)$ which is an approximation to $w(ih,jh)$ for each $1 \leq i, j \leq n$. If we replace the differential operator by the familiar five-point difference approximation at each interior mesh point, cf. [8], we obtain the system of linear equations

$$(4-\sigma h^2)\, W(i,j) - W(i,j-1) - W(i,j+1) - W(i-1,j) - W(i+1,j) = h^2 G(i,j), \tag{5.2}$$
$$1 \leq i, j \leq n,$$

where $W(i,j) = 0$ if $i = 0$ or $n+1$ or if $j = 0$ or $n+1$ and $G(i,j) = g(ih,jh)$.

If the unknowns $W(i,j)$ are ordered in the natural row-by-row fashion, the system (5.2) is reduced to a nxn block tridiagonal system of linear equations

$$AW \equiv \begin{bmatrix} T & -I & 0 \\ -I & \ddots & -I \\ 0 & -I & T \end{bmatrix} W = h^2 G$$

where T is the nxn tridiagonal matrix

$$T \equiv \begin{bmatrix} (4-\sigma h^2) & -1 & 0 \\ -1 & \ddots & -1 \\ 0 & -1 & (4-\sigma h^2) \end{bmatrix}$$

and I is the nxn identity matrix. The matrix A has five non-zero diagonals and is symmetric.

In three dimensions, we consider the problem:

$$-\Delta w(x,y,z) - \sigma w(x,y,z) = f(x,y,z), \quad (x,y,z) \in D \equiv (0,1) \times (0,1) \times (0,1)$$

with Dirichlet boundary conditions

$$w(x,y,z) = 0, \qquad (x,y,z) \in \partial D$$

where σ is a constant.

Proceeding in an analogous way to the two dimensional case, we put an (n+1) x (n+1) x (n+1) grid on the unit cube and use a seven-point finite difference approximation to the differential operator. This yields an $n^3 \times n^3$ linear system Ax = f, where A is symmetric and has its non-zeroes restricted to seven diagonals at distances zero (the main diagonal), 1, n+1, and n^2+1 from the main diagonal. We have

$$A = \begin{bmatrix} B & -I & 0 \\ -I & \ddots & -I \\ 0 & -I & B \end{bmatrix}$$

where B is a $n^2 x n^2$ matrix of the form

$$B = \begin{bmatrix} D & -I & 0 \\ -I & \ddots & -I \\ 0 & -I & D \end{bmatrix} ,$$

D is the nxn tridiagonal matrix,

$$D = \begin{bmatrix} (6-\sigma h^2) & -1 & 0 \\ -1 & (6-\sigma h^2) & \ddots \\ 0 & \ddots & -1 \\ & & -1 \quad (6-\sigma h^2) \end{bmatrix} ,$$

$h = 1/(n+1)$ and the I's denote identity matrices of suitable order.

If $\sigma \leq 0$, the corresponding matrices A for the two and three dimensional model problems are positive definite. However, if σ is sufficiently positive, the corresponding matrices are strictly indefinite, i.e., they have both positive and negative eigenvalues. We also have, in the notation of Theorem 3.1, $\alpha = 0(h^2)$. Applying Theorem 3.1, we obtain the following convergence results:

Theorem 5.1: For the model problems, MCR requires $0(n \log n)$ iterations to reduce the initial error by a factor n^{-p}, $p > 0$. The number of multiplications required to reduce the error by the same factor are $0(n^3 \log n)$ in two and $0(n^4 \log n)$ in three dimensions.

In the next section we present results of numerical experiments with MCR for the two-and-three-dimensional model problems for various values of σ and h.

The storage and work requirements for these problems are given in Table 1. For the model problems, the diagonal elements of A are all equal and all the non-zero off-diagonal elements are just -1's so that no storage is required for A, and computing a matrix-vector product requires only N multiplications. Table 1 also gives the storage and work requirements for a general problem. A general problem is one that arises from the application of the simplest (i.e., five-point in two and seven-point in three dimensions) finite difference approximations to any second-order self-adjoint elliptic partial differential equation where A has the same non-zero structure as for the corresponding model problem. A is still symmetric, but the off-diagonal non-zero entries need not be -1's so that the storage and work costs are higher than for the corresponding model problem.

6. Numerical Results

Numerical experiments were performed to demonstrate the performance of the MCR method on the two-and three-dimensional model problems for various σ and h. The solution was chosen to be

$$w(x,y) = 3e^x e^y (x-x^2)(y-y^2)$$

in two and

$$w(x,y,z) = 3e^x e^y e^z (x-x^2)(y-y^2)(z-z^2)$$

in three dimensions. If $\sigma = 0$, all the eigenvalues of A are positive. The first few eigenvalues of the differential operator $-\Delta$ in two dimensions are 19.7, 49.4, 79.0, 98.7,..., cf. [8], so that for $\sigma = 30$ and 90 A has one and three negative eigenvalues respectively. Similarly, in three dimensions, the eigenvalues of $-\Delta$ are 29.6, 59.2, 88.8, 108.6, ... so that for $\sigma = 50$ and 100 A again has one and three negative eigenvalues respectively.

The initial guess x_0 was taken to be the zero vector and the error at iteration i was defined as

$$\text{error} = \| f - Ax_i \|_2 / \| f - Ax_0 \|_2$$

and was computed as $((r_i,r_i)/(r_0,r_0))^{1/2}$. We chose $\varepsilon = 10^{-4}$; the computations were carried out on a PDP-10 (word-length 36 bits) in single precision.

Figures 1a and 1b show the error versus number of iterations for the model problems. Tables 2 and 3 give the total work required to reduce the error to 10^{-6}. In these experiments it turned out that

MCR always chose the shorter iteration.

Since the difference approximations are only second-order accurate, we also computed the number of iterations required to reduce the error by a factor of $1/n^2$ for various h and σ. We see from Figures 2a and 2b that for a fixed problem (i.e., fixed σ) the plot of the number of iterations against n log n is a straight line. This illustrates the conclusions of Theorem 5.1.

Problem	Two dimensions ($N=n^2$) Storage reqd.	No. of mults./ iteration $\|a_i\|>\varepsilon$	No. of mults./ iteration $\|a_i\|\leq\varepsilon$	Three dimensions ($N=n^3$) Storage reqd.	No. of mults./ iteration $\|a_i\|>\varepsilon$	No. of mults./ iteration $\|a_i\|\leq\varepsilon$
Model	7N	8N+2	10N+4	7N	8N+2	10N+4
General	10N-2n	12N-4n+2	14N-4n+4	$11N-3n^2$	$14N-6n^2+2$	$16N-6n^2+4$

Table 1: Storage requirements and multiplication counts for N x N systems.

Mesh-width h	No. of mults./ iteration $\|a_i\|>\varepsilon$	No. of mults./ iteration $\|a_i\|\leq\varepsilon$	σ = 30 No. of iters.	σ = 30 Total no. of mults.	σ = 90 No. of iters.	σ = 90 Total no. of mults.
1/8	394	494	21	8,175	29	11,327
1/16	1,802	2,254	52	93,253	63	113,075
1/32	7,690	9,614	108	828,597	131	1,005,467

Table 2: Work required to reduce the error to 10^{-6} for two-dimensional model problems on n x n mesh ($h = 1/(n+1)$).

Mesh-width h	No. of mults./ iteration $\|a_i\|>\varepsilon$	No. of mults./ iteration $\|a_i\|\leq\varepsilon$	σ = 50 No. of iters.	σ = 50 Total no. of mults.	σ = 100 No. of iters.	σ = 100 Total no. of mults.
1/4	218	274	9	1,907	8	1,689
1/8	2,746	3,434	32	87,185	52	142,105
1/16	27,002	33,754	71	1,910,391	93	2,504,435

Table 3: Work required to reduce the error to 10^{-6} for three-dimensional model problems on n x n x n mesh ($h=1/(n+1)$).

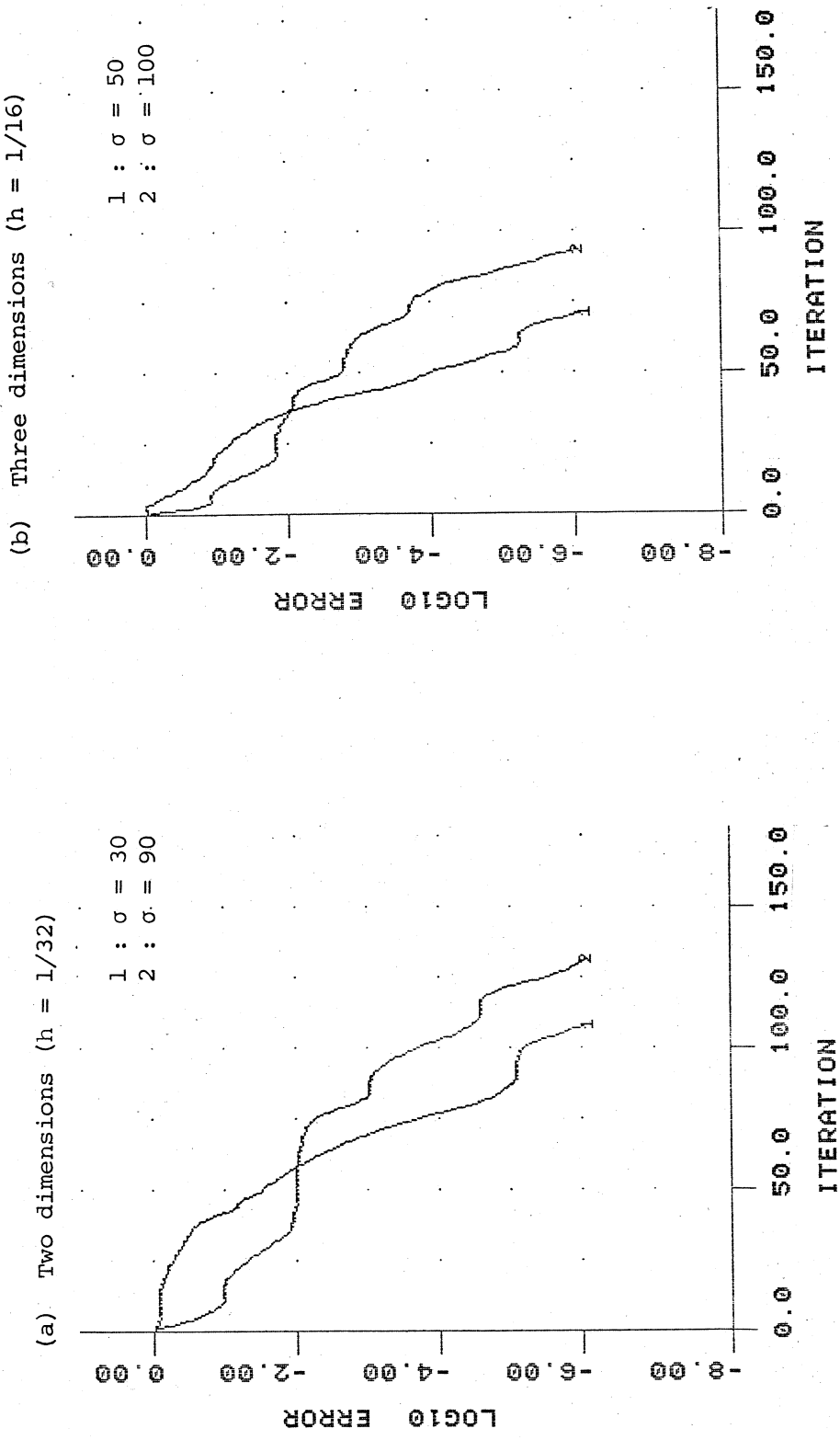

Figure 1: Error versus number of iterations for two and three-dimensional model problems.

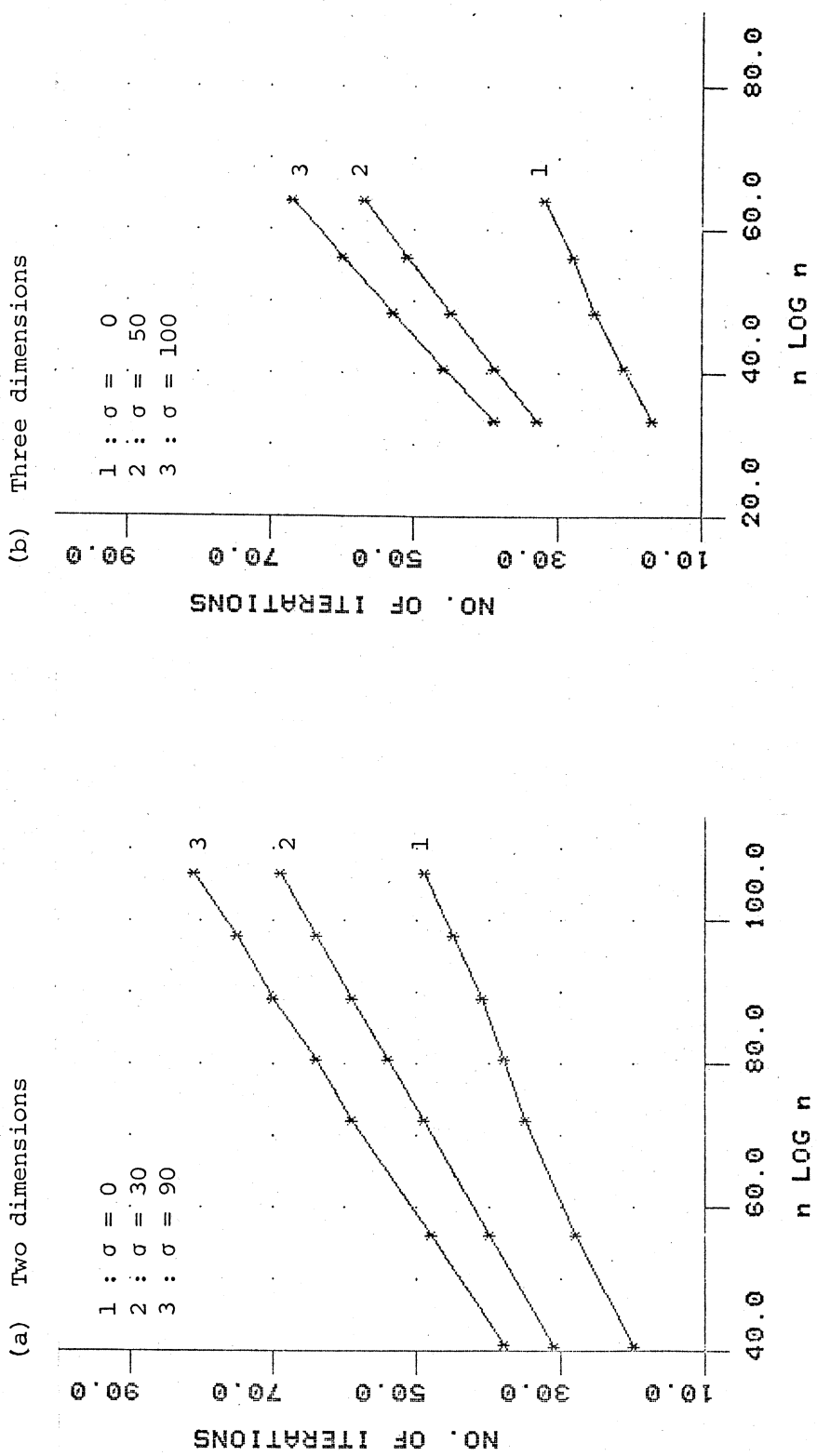

Figure 2: Graphs showing the number of iterations required to reduce the error by $1/(n^2)$ vs. n log n for two-and three-dimensional model problems.

References

[1] R. Chandra. Ph.D. Dissertation. Department of Computer Science, Yale University. To appear.

[2] R. Chandra, S. C. Eisenstat, and M. H Schultz. Conjugate gradient methods for partial differential equations. Proceedings of the AICA Intertational Symposium on Computer Methods for Partial Differential Equations, Bethlehem, Pennsylvania, 1975.

[3] J. W. Daniel. The Approximate Minimization of Functionals. Prentice-Hall, INc., 1971.

[4] M. R. Hestenes. The conjugate gradient method for solving linear systems. Proc. Symposia in Appl. Math. 6:83-102, Numerical Analysis. McGraw-Hill, New York, 1956.

[5] V. I. Lebedev. Iterative methods for the solution of operator equations with their spectrum lying on several intervals. Journal of Computational Mathematics and Mathematical Physics 9(6):1247-1252, 1969.

[6] D. G. Luenberger. The conjugate residual method for constrained minimizations problems. SIAM J. Numer. Anal. 7:390-398, 1970.

[7] C. C. Paige. Computational variants of the Lanczos method for the eigenproblem. J. Inst. Math. Appl. 10:373-381, 1972.

[8] R. S. Varga. Matrix Iterative Analysis. Prentice-Hall, Inc., 1962.