Conjugate Gradient Methods for Partial Differential Equations[†]

R. Chandra, S.C. Eisenstat, and M.H. Schultz[*]

Research Report #49

---

† This paper has appeared in the Proceedings of the AICA International Symposium on Computer Methods for Partial Differential Equations, Bethlehem, Pennsylvania, June, 1975.

* This research supported in part by NSF Grant GJ-43157 and ONR Grant N0014-67-A-0097-0016.

1. Introduction

In this paper we consider two iterative methods for solving the large sparse, symmetric, positive definite linear systems one usually encounters in applying the finite difference or finite element method to approximate the solutions of partial differential equations. In particular, we discuss the conjugate gradient (CG) and the conjugate gradient, strongly implicit factorization (CGSIP) iterative methods for solving two model problems, the simplest finite difference approximations to Poisson's equation in a two dimensional square and a three dimensional cube. 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.

We present the algorithms and state theoretical and experimental results which show the superiority of the second algorithm. Moreover, the CGSIP algorithm is asymptotically more efficient with respect to storage and work than the sparse symmetric Gaussian elimination algorithm described in two companion papers in these proceedings, [7] and [8]. However, for problems of modest size in two dimensions and small problems in three dimensions the elimination algorithm seems to be more efficient.

In section 2, we present the two dimensional model problem in detail, while in sections 3 and 4 we describe the two algorithms.

Finally, in section 5, we discuss extensions to the three dimensional model problem.

2. The Two Dimensional Model Problem

In this section, we introduce a model problem, the five-point difference approximation to the Poisson equation on a square. In section 5, we give a generalization to the seven-point difference approximation to the Poisson equation defined on a cube.

We consider the problem

$$-\Delta w(x,y) = f(x,y), \quad (x,y) \in D \equiv (0,1)\times(0,1) \tag{2.1}$$

with Dirichlet boundary conditions

$$w(x,y) = 0, \quad (x,y) \in \partial D. \tag{2.2}$$

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 $1 \le i, j \le n$.

If we replace the differential operator by the familiar five-point difference approximation at each interior point, cf. [14], we obtain the system of linear equations

$$4W(i,j) - W(i,j-1) - W(i,j+1) - W(i-1,j) - W(i+1,j) = h^2F(i,j), \quad 1 \le i, j \le n, \tag{2.3}$$

where $W(i,j) = 0$ if $i = 0$ or $n+1$ or if $j = 0$ or $n+1$

and $F(i,j) = f(ih,jh)$.

If the unknowns $W(i,j)$ are ordered in the natural row-by-row fashion, the system (2.3) is reduced to an $n \times n$ block tridiagonal system of linear equations

$$A\,W \equiv \begin{bmatrix} T & -I & & 0 \\ -I & T & \ddots & \\ & \ddots & \ddots & -I \\ 0 & & -I & T \end{bmatrix} \qquad W = h^2 F$$

where T is the $n \times n$ tridiagonal matrix

$$T \equiv \begin{bmatrix} 4 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 4 \end{bmatrix}$$

and I is the $n \times n$ identity matrix. Equivalently

$$A \equiv \begin{bmatrix} b_1 & c_1 & d_1 & & 0 \\ c_1 & \ddots & \ddots & \ddots & d_{N-n} \\ d_1 & \ddots & \ddots & \ddots & c_{N-1} \\ & \ddots & \ddots & \ddots & b_N \\ 0 & d_{N-n} & c_{N-1} & b_N & \end{bmatrix} ,$$

where $N \equiv n^2$, $b_i \equiv 4$, $1 \le i \le N$,

$$c_i \equiv \begin{cases} 0 & \text{if } i = tn, \quad 1 \le t \le n-1 \\ -1 & \text{otherwise} , \end{cases}$$

and $d_i = -1$, $1 \le i \le N-n$. The matrix A has five nonzero diagonals and it is easy to verify that A is symmetric and positive definite, cf. [14].

3. The Conjugate Gradient Method

In this section, we introduce and discuss the conjugate gradient (CG) iterative method for solving a linear system with an arbitrary symmetric, positive definite coefficient matrix A. The CG method is easy to program, cf. [12], relatively efficient, cf. [11], and does not depend on a critical choice of iteration parameters.

Given a linear system

$$A x = k \tag{3.1}$$

of N equations, whose coefficient matrix A is symmetric positive definite, and an arbitrary initial guess x_0 to x, we form the residual vector

$$r_0 \equiv k - Ax_0 \tag{3.2}$$

and the directional vector

$$p_0 = r_0 . \tag{3.3}$$

Then for $i = 0, 1, 2, \ldots$ we define the vectors x_{i+1}, r_{i+1}, and p_{i+1} and the scalars a_i and b_i using the relations

$$a_i \equiv (r_i, r_i)/(p_i, Ap_i) , \tag{3.4}$$

$$x_{i+1} \equiv x_i + a_i p_i \,, \tag{3.5}$$

$$r_{i+1} \equiv r_i - a_i A p_i \,, \tag{3.6}$$

$$b_i \equiv (r_{i+1}, r_{i+1})/(r_i, r_i) \tag{3.7}$$

and

$$p_{i+1} \equiv r_{i+1} + b_i p_i \,. \tag{3.8}$$

Table 1 summarizes the storage and the number of multiplications/iteration required by this algorithm for the model problem and also the corresponding information for the linear system arising from the application of the five-point finite difference approximation to any self-adjoint second order elliptic partial differential equation with variable coefficients in the unit square (referred to as the general problem). The qualitative and quantitative behavior of CG method is very well understood, cf. [3], [4], [9], and [11]. If we define the norms

$$\|y\|_A \equiv (y, Ay)^{1/2} \tag{3.9}$$

and

$$\|y\|_2 \equiv (y, y)^{1/2} \tag{3.10}$$

and the eigenvalues of A lie in the interval $[a,b]$, then

$$\|x_m - x\|_A \leq 2\left(\frac{1 - \sqrt{\alpha}}{1 + \sqrt{\alpha}}\right)^m \|x_0 - x\|_A \tag{3.11}$$

$$\|x_m - x\|_2 \leq 2\left(\frac{1 - \sqrt{\alpha}}{1 + \sqrt{\alpha}}\right)^m \|x_0 - x\|_2 \tag{3.12}$$

where $\alpha \equiv a/b$, cf. [3].

For the specific case of the model problem $\alpha = O(h^2)$, cf. [14], and we have the following convergence result, cf. [2].

Theorem 3.1. For the model problem, the CG method requires $O(n \log \varepsilon^{-1})$ iterations and $O(n^3 \log \varepsilon^{-1})$ multiplications to reduce the initial error by a factor of ε.

Computational results demonstrating the conclusions of this Theorem for the model problem are given in Table 3 and for the general problem in Table 4. The solution vector x was chosen to be a vector of N random numbers and the right hand side of (3.1) was obtained by taking its product with the matrix A. The initial guess x_0 was taken to be the zero vector and the error was computed as

$$\text{error} = \left(\frac{1}{N} \|x_j - x\|_2^2\right)^{1/2}.$$

The iteration process was terminated when the error was reduced to 10^{-6}.

From (3.11) and (3.12) it is clear, cf. [1], [2], and [10] for the details, that one way to modify the CG method to get better asymptotic convergence is to scale the linear system (3.1) by means of a symetric, positive definite matrix B^{-1}, i.e., to consider in place of (3.1)

$$B^{-1}A\,x = B^{-1}k\,, \tag{3.13}$$

where B is chosen so that the eigenvalues of $B^{-1}A$ lie in $[a',b']$ and $a'/b' > a/b$. We then apply the CG method (with respect to the

inner-product $(x,y)_B \equiv (x,By)$ instead of (x,y)) to (3.13) as before.

There is an extra vector $\tilde{r}_{i+1}$ introduced in this variant of the CG method and the only additional work involved is the solution of a linear system

$$B\, r_{i+1} = \tilde{r}_{i+1} \tag{3.14}$$

for each iteration. In practice, we recommend choosing $B = (LL^T)$ where L is lower triangular, and LL^T is an "approximate factorization" of A, i.e. $LL^T = A + R$ and R is small. With this form of B, (3.14) is equivalent to solving a linear system of equations whose coefficient matrix is given in factored form. As we shall see in the next section, this can lead to a significant reduction in the number of iterations.

4. The Conjugate Gradient Strongly Implicit Factorization Methods

In this section, we present and study the conjugate gradient strongly implicit factorization (CGSIP) methods for solving the model problem. As with the CG method, the CGSIP method is easy to program, very efficient, and does not depend on a critical choice of iteration parameters. We will show that it compares favorably with other methods for the model problem.

In particular, instead of considering

$$A\, x = k \tag{4.1}$$

we consider

$$(LL^T)^{-1}A\,x \;=\; (LL^T)^{-1}\,k \tag{4.2}$$

where LL^T is an appropriate approximate factorization of A. Several procedures for constructing appropriate matrices L have appeared in the literature, cf. [6], [10], and [13]. We will discuss only the first of these due to Dupont, Kendall, and Rachford.

The matrix L is defined to have the same zero structure as the lower triangle of A and in practice, for computational efficiency, we actually factor the matrix LL^T as $\tilde{L}\,\tilde{U}$, where

$$\tilde{L} \equiv \begin{bmatrix} v_1 & & & 0 \\ c_1 & v_2 & & \\ & \ddots & \ddots & \\ d_1 & 0 & & \\ & \ddots & \ddots & v_N \\ 0 & d_{N-n} & c_{N-1} & \end{bmatrix},$$

$$\tilde{U} \equiv \begin{bmatrix} 1 & t_1 & g_1 & 0 \\ & \ddots & \ddots & \ddots \\ & & 0 & g_{N-n} \\ & & & t_{N-1} \\ 0 & & & 1 \end{bmatrix},$$

and the vectors c and d appear in the original matrix A. For reasons of efficiency, we compute and store the reciprocals $\tilde{v}_i$ of the v_i's using the relations

$$\tilde{v}_i \;=\; [4(1+\alpha_i) - c_{i-1}^2\,\tilde{v}_{i-1} + c_i\tilde{v}_{i-1} + c_{i-n}\tilde{v}_{i-n} - \tilde{v}_{i-n}]^{-1} \tag{4.3}$$

where $\tilde{v}_i = c_i = 0$ for $i < 1$ and the α_i are positive parameters. For the model problem, the coefficients of each of the $\tilde{v}_i$'s in (4.3) are

either 0 or 1 and the computation of $\tilde{L}$ is particularly simple. Moreover, the vectors t and g needed to define $\tilde{U}$ can be directly recovered from $\tilde{v}$ as follows:

$$t_i \equiv \begin{cases} 0 & \text{if } i = sn, \quad s = 1, 2, \ldots, n-1 \\ -\tilde{v}_i & \text{otherwise} \end{cases} \tag{4.4}$$

and

$$g_i \equiv -\tilde{v}_i \,. \tag{4.5}$$

In the case of an arbitrary self-adjoint, second order elliptic partial differential equation with variable coefficients, the analogous factorization algorithm is somewhat more complicated, cf. [6]. Work estimates for the factorization algorithm are given in Table 1.

Once we have computed the factorization, we solve (4.2) by the CG method (with respect to the inner product (x,LL^Ty)). This reduces to the following algorithm which is a variant of (3.2) - (3.8). Starting with an arbitrary initial guess x_0, form

$$\tilde{r}_0 \equiv k - A x_0, \tag{4.6}$$

solve

$$\tilde{L}\,\tilde{U}\, r_0 \equiv \tilde{r}_0, \tag{4.7}$$

and set

$$p_0 = r_0. \tag{4.8}$$

Then for $i = 0, 1, 2, \ldots$ form the vectors x_{i+1}, $\tilde{r}_{i+1}$, r_{i+1}, and p_{i+1} and the scalars a_i and b_i by means of

$$a_i \equiv (\tilde{r}_i, r_i)/(p_i, Ap_i) \,, \tag{4.9}$$

$$x_{i+1} \equiv x_i + a_i p_i \,, \tag{4.10}$$

$$\tilde{r}_{i+1} \equiv \tilde{r}_i - a_i A p_i \,, \tag{4.11}$$

$$\tilde{L}\,\tilde{U}\, r_{i+1} \equiv \tilde{r}_{i+1} \,, \tag{4.12}$$

$$b_i \equiv (\tilde{r}_{i+1}, r_{i+1})/(\tilde{r}_i, r_i) \tag{4.13}$$

and

$$p_{i+1} \equiv r_{i+1} + b_i p_i \,. \tag{4.14}$$

The storage and the number of multiplications/iteration required for this algorithm are given in Table 1. Moreover, we can prove the following result on the rate of convergence, cf. [2].

Theorem 4.1. Let $\alpha_i = kh^2$, $1 \le i \le N$, where k is a positive constant independent of h. For the model problem, the CGSIP method requires $O(n^{1/2} \log \varepsilon^{-1})$ iterations and $O(n^{5/2} \log \varepsilon^{-1})$ multiplications to reduce the initial error by a factor of ε.

Computational results verifying the conclusions of this Theorem for a variety of values of n are given for the CGSIP method applied to the model problem in Table 3 and for general problems in Table 4. The error was computed as in section 3 and the same termination criterion was used. These results should be compared with the corresponding results for the CG method and for other methods.

5. Three Dimensional Problems

The algorithms and results of the previous sections can be generalized to apply to the solution of self-adjoint, second order elliptic partial differential equations in three (or higher) dimensions. We put an $n \times n \times n$ 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

$$A x = k , \tag{5.1}$$

where A is symmetric, positive definite and has its nonzeroes restricted to seven diagonals at distances zero (the main diagonal), 1, $n+1$, and n^2+1 from the main diagonal.

For the case of the Poisson equation we have

$$A = \begin{bmatrix} B_1 & -I & & 0 \\ -I & B_2 & \ddots & \\ & \ddots & \ddots & -I \\ 0 & & -I & B_n \end{bmatrix}$$

where the B_i's are $n^2 \times n^2$ matrices of the form

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

and D is an $n \times n$ tridiagonal matrix with -1's on the sub- and superdiagonals and 6's on the diagonal. The I's denote identity matrices of suitable order.

Proceeding in an analogous way to the two dimensional case, we can use the CG and CGSIP methods, cf. [5]. Likewise one obtains the following convergence result, cf. [2] and [6].

Theorem 5.1. For the model problem in the cube, the CG method requires $0(n \log \varepsilon^{-1})$ iterations and $0(n^4 \log \varepsilon^{-1})$ multiplications to reduce the error by a factor of ε and the CGSIP method requires $0(n^{1/2} \log \varepsilon^{-1})$ iterations and $0(n^{7/2} \log \varepsilon^{-1})$ multiplications to reduce the error by a factor of ε.

A summary of the storage and multiplications/iteration required for both methods applied to the model and general problems is given in Table 2. Numerical results for both methods on the model problem are reported in Table 5 and numerical results for both methods on the general problem are reported in Table 6. The error was computed as in sections 3 and 4 and the same termination criterion was used. These results compare quite favorably with the corresponding results for other methods.

Two Dimensional Problems

Equation	Vectors stored	Storage required	No. of mults./ iteration	Mults. for factorization
Poisson:				
CG	r, x, p, Ap	4N	6N + 2	0
CGSIP	r, x, p, Ap, $\tilde{v}$	5N	8N + 1	2N
General:				
CG	r, x, p, Ap, b, c, d	7N - 2n	10N - 4n + 2	0
CGSIP	r, x, p, Ap, b, c, d, v, t, g	10N - 4n	15N - 8n + 2	5N - 4n

Table 1: Storage requirements and multiplication counts for N×N ($N = n^2$) systems.

Three Dimensional Problems

Equation	Vectors stored	Storage required	No. of mults./ iteration	Mults. for factorization
Poisson:				
CG	r, x, p, Ap	4N	6N + 2	0
CGSIP	r, x, p, Ap, $\tilde{v}$	5N	8N + 1	2N
General:				
CG	r, x, p, Ap, b, c, f, d	$8N - 3n^2$	$12N - 6n^2 + 2$	0
CGSIP	r, x, p, Ap, b, c, f, d, v, t, g, s	$12N - 6n^2$	$19N - 12n^2 + 2$	$7N - 6n^2$

Table 2: Storage requirements and multiplication counts for N×N ($N = n^3$) systems.

n	CG No. of iterations	CG No. of mults./ iteration	CG Total no. of mults.	CGSIP No. of iterations	CGSIP No. of mults./ iteration	CGSIP Total no. of mults.
16	45	1538	69722	14	2049	29965
32	89	6146	549042	22	8193	185365
48	131	13826	1815814	26	18433	490777
64	175	24578	4309342	31	32769	1036318

Table 3: Work required to reduce the error to 1E-6 for the Poisson equation on n×n mesh.

n	CG No. of iterations	CG No. of mults./ iteration	CG Total no. of mults.	CGSIP No. of iterations	CGSIP No. of mults./ iteration	CGSIP Total no. of mults.
16	45	2498	113882	14	3714	55644
32	89	10114	906162	22	15106	347308
48	131	22850	3006982	26	34178	922612
64	175	40706	7147870	31	60930	1949502

Table 4: Work required to reduce the error to 1E-6 for the General Equation on n×n mesh.

n	CG No. of iterations	CG No. of mults./ iteration	CG Total no. of mults.	CGSIP No. of iterations	CGSIP No. of mults./ iteration	CGSIP Total no. of mults.
4	14	386	5532	8	513	4423
8	29	3074	90170	12	4097	51723
12	42	10370	438996	15	13825	216014
16	54	24578	1335404	18	32769	610321

Table 5: Work required to reduce the error to 1E-6 for the Poisson Equation on n×n×n mesh.

n	CG			CGSIP		
	No. of iterations	No. of mults./ iteration	Total no. of mults.	No. of iterations	No. of mults./ iteration	Total no. of mults.
4	14	674	9852	8	1026	9264
8	29	5762	170810	12	8962	117144
12	42	19874	847668	15	31106	500286
16	54	47618	2602604	18	74754	1426980

Table 6: Work required to reduce the error to 1E-6 for the General Equation on n×n×n mesh.

References

[1] R. Bartels and J.W. Daniel. A conjugate gradient approach to nonlinear elliptic boundary value problems in irregular regions. Report #CNA 63, Center for Numerical Analysis, University of Texas at Austin, 1973.

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

[3] J.W. Daniel. The conjugate gradient method for linear and nonlinear operator equations. *SIAM J. Numer. Anal.* 4:10-26, 1967.

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

[5] T. Dupont. A factorization procedure for the solution of elliptic difference equations. *SIAM J. Numer. Anal.*5:753-782, 1968.

[6] T. Dupont, R.P. Kendall and H.H. Rachford, Jr. An approximate factorization procedure for solving self-adjoint elliptic difference equations. *SIAM J. Numer. Anal.* 5:559-573, 1968.

[7] S.C. Eisenstat, M.H. Schultz, and A.H. Sherman. Application of Sparse Matrix Methods to Partial Differential Equations. In the Proceedings of the AICA International Symposium on Computer Methods for Partial Differential Equations, Bethlehem, Penn., 1975.

[8] S.C. Eisenstat, M.H. Schultz, and A.H. Sherman. Efficient Implementation of Sparse Symmetric Gaussian Elimination. Same as above.

[9] M.R. Hestenes and E. Stiefel. Method of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards*, Sect. B, 49:409-436, 1952.

[10] J.A. Meijerink and H.A. Van der Vorst. Iterative solution of linear systems arising from discrete approximations to partial differential equations. A.C.C.U., Budapestlaan 6, "De Uithof," Utrecht, the Netherlands, 1974.

[11] J.K. Reid. On the method of conjugate gradients for the solution of large sparse systems of linear equations. Proc. Conference on "Large Sparse Sets of Linear Equations," Academic Press, New York, 1971.

[12] J.K. Reid. A Fortran subroutine for the solution of large sparse sets of linear equations by conjugate gradients. Research Group Report, Theoretical Physics Division, Atomic Energy Research Establishment, Harwell, G.B., 1970.

[13] H.L. Stone. Iterative solution of implicit approximations of multidimensional partial differential equations. *SIAM J. Numer. Anal.* 5:530-558, 1968.

[14] R.S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, 1962.