Reducing Communication Costs in the Conjugate Gradient Algorithm on Distributed Memory Multiprocessors

E. F. D'Azevedo
C. H. Romine
This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831. Prices available from (615) 576-9401, FAX 576-9491.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.
REDUCING COMMUNICATION COSTS IN THE
CONJUGATE GRADIENT ALGORITHM ON
DISTRIBUTED MEMORY MULTIPROCESSORS

E. F. D'Azevedo
C. H. Romine

Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 6012
Oak Ridge, TN 37831-6367

Date Published: September 1992

Research supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
managed by
Martin Marietta Energy Systems, Inc.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400
Contents

1 Introduction ........................................... 1
2 The conjugate gradient algorithm .................... 2
3 The Lanczos algorithm ................................ 4
4 Numerical experiments on stability ................... 7
5 Parallel performance .................................. 14
6 Conclusion ............................................ 16
7 References ............................................. 16
List of Figures

4.1 Classical CG on Test 1 .............................................. 9
4.2 Modified CG on Test 1 .............................................. 9
4.3 Eijkhout Rearrangement on Test 1 ............................... 10
4.4 Meurant Rearrangement on Test 1 ................................. 10
4.5 Lanczos Rearrangement on Test 1 ................................. 11
4.6 Classical CG on Tests 2-4 ......................................... 11
4.7 Modified CG on Tests 2-4 ......................................... 12
4.8 Eijkhout Rearrangement on Tests 2-4 ............................. 12
4.9 Meurant Rearrangement on Tests 2-4 ............................. 13
4.10 Lanczos Rearrangement on Tests 2-4 ............................ 13
REDUCING COMMUNICATION COSTS IN THE CONJUGATE GRADIENT ALGORITHM ON DISTRIBUTED MEMORY MULTIPROCESSORS

E. F. D'Azevedo
C. H. Romine

Abstract

The standard formulation of the conjugate gradient algorithm involves two inner product computations. The results of these two inner products are needed to update the search direction and the computed solution. In a distributed memory parallel environment, the computation and subsequent distribution of these two values requires two separate communication and synchronization phases. In this paper, we present a mathematically equivalent rearrangement of the standard algorithm that reduces the number of communication phases. We give a second derivation of the modified conjugate gradient algorithm in terms of the natural relationship with the underlying Lanczos process. We also present empirical evidence of the stability of this modified algorithm.
1. Introduction

The conjugate gradient (CG) method is an effective iterative method for solving large sparse symmetric positive definite systems of linear equations. It is robust and, coupled with an effective preconditioner [19], is generally able to achieve rapid convergence to an accurate solution.

One drawback of the standard formulation of the conjugate gradient algorithm on distributed memory parallel machines is that it involves the computation of two separate inner products of distributed vectors. Moreover, the first inner product must be completed before the data are available for computing the second inner product. Hence, a distributed memory implementation of the standard conjugate gradient method has two separate communication phases for these two inner products. Since communication is quite expensive on the current generation of distributed memory multiprocessors, it is desirable to reduce the communication overhead by combining these two communication phases into one.

Saad [16,17] has shown one rearrangement of the computation that eliminates a communication phase by computing $\|r_k\|_2$ based on the relationship

$$\|r_{k+1}\|_2 = \alpha_k^2 \|Ap_k\|_2^2 - \|r_k\|_2^2$$

(1.1)

to be numerically unstable. Meurant [11] proposed using (1.1) as a predictor for $\|r_{k+1}\|$ and reevaluate the actual norm on the next iteration with an extra inner product. Van Rosendale [20] has proposed without numerical results an $m$-step conjugate gradient algorithm to increase parallelism.

The conjugate gradient algorithm is known to be closely related to the Lanczos algorithm for tridiagonalizing a matrix [5]. Paige [12,13,14] has done detailed analysis to show some variants of the Lanczos algorithm are unstable. Strakos [18] and Greenbaum [6,7] have considered the close connection between the Lanczos and CG algorithm in the analysis of stability of CG computations under perturbations in finite arithmetic.

In §2, we present a rearrangement of the conjugate gradient computation that
eliminates one communication phase by computing both inner products at once. We show a natural association between this rearrangement and the Lanczos algorithm in §3. A discussion of how this rearrangement of the computation affects the stability properties of the conjugate gradient algorithm and some MATLAB numerical experiments on the effectiveness of the rearrangement are included in §4.

2. The conjugate gradient algorithm

We begin by reviewing the standard conjugate gradient procedure [2,9] for solving the linear system

\[ Ax = b. \]  

(2.1)

For simplicity, we assume a zero initial guess, and residual vector \( r_1 = b \), with \( \langle x, y \rangle = x^t y \) as the usual inner product.

For \( k = 1, 2, \ldots \)

\[ \gamma_k = \langle r_k, r_k \rangle \]
\[ \beta_k = \gamma_k / \gamma_{k-1} \quad (\beta_1 = 0) \]
\[ p_k = r_k + \beta_k p_{k-1} \quad (p_1 = r_1) \]
\[ v_k = Ap_k \]
\[ \sigma_k = \langle p_k, v_k \rangle \]
\[ \alpha_k = \gamma_k / \sigma_k \]
\[ x_{k+1} = x_k + \alpha_k p_k \]
\[ r_{k+1} = r_k - \alpha_k v_k. \]  

(2.2)

(2.3)

Saad [16,17] and Meurant [11] have considered eliminating the first inner product for \( \gamma_k = \langle r_k, r_k \rangle \). We propose eliminating the second communication phase by finding alternative expressions for \( \sigma_k \).
To further simplify (2.4), we rely on an intrinsic property of the CG procedure, the orthogonality of residual vectors (and equivalently the conjugacy of search directions [9, page 420])

\[
\frac{\langle r_k, r_{k+1} \rangle}{\langle r_k, r_k \rangle} = \frac{\langle p_k, Ap_{k+1} \rangle}{\langle p_k, Ap_k \rangle} = 0. \tag{2.5}
\]

(Eijkhout [4] suggests performing an extra inner product to explicitly evaluate \(\langle r_k, v_{k-1} \rangle\) for improved stability properties.) Thus from (2.3)

\[
r_k = r_{k-1} - \alpha_{k-1}v_{k-1} \\
\langle r_k, r_k \rangle = \langle r_k, r_{k-1} \rangle - \alpha_{k-1} \langle r_k, v_{k-1} \rangle \\
\gamma_k = 0 - \alpha_{k-1} \langle r_k, v_{k-1} \rangle. \tag{2.6}
\]

Therefore by (2.4), (2.6) and \(\beta_k = \gamma_k/\gamma_{k-1},\)

\[
\sigma_k = \langle r_k, Ar_k \rangle + 2\beta_k(-\gamma_k/\alpha_{k-1}) + \beta_k^2\sigma_{k-1} \\
\sigma_k = \delta_k - \beta_k^2\sigma_{k-1}, \text{ where } \delta_k = \langle r_k, Ar_k \rangle. \tag{2.7}
\]

We propose the following rearrangement of the conjugate gradient procedure. First initialize \(\sigma_1\) and \(v_1,\) by performing one step of the standard algorithm

\[
r_1 = b, \quad \gamma_1 = \langle r_1, r_1 \rangle, \quad p_1 = r_1, \quad v_1 = Ap_1 \\
\sigma_1 = \langle p_1, v_1 \rangle, \quad x_2 = (\gamma_1/\sigma_1)p_1
\]
For \( k = 2, 3, \ldots \)
\[
\begin{align*}
  s_k &= Ar_k \\
  \gamma_k &= \langle r_k, r_k \rangle \\
  \beta_k &= \gamma_k / \gamma_{k-1} \\
  p_k &= r_k + \beta_k p_{k-1} \\
  v_k &= s_k + \beta_k v_{k-1} \quad (v_k \equiv Ap_k) \\
  \sigma_k &= \delta_k - \beta_k^2 \sigma_{k-1} \\
  \alpha_k &= \gamma_k / \sigma_k \\
  x_{k+1} &= x_k + \alpha_k p_k \\
  r_{k+1} &= r_k - \alpha_k v_k .
\end{align*}
\]

Note that the above procedure requires extra storage for the vector \( s_k \) and extra work in updating the vector \( v_k \); in contrast to Meurant’s rearrangement, only two inner products and one matrix vector multiply are performed.

### 3. The Lanczos algorithm

In this section we present a natural derivation of the modified CG rearrangement from the Lanczos process. The Lanczos process is a tridiagonalization procedure to find orthogonal \( Q \) and tridiagonal \( T \) such that
\[
Q^T A Q = T = \begin{bmatrix}
\hat{\alpha}_1 & \hat{\beta}_1 \\
\hat{\beta}_1 & \hat{\alpha}_2 & \cdots \\
& \ddots & \ddots \\
& & \hat{\beta}_{n-1} & \hat{\alpha}_n \\
& & & \hat{\beta}_{n-1} & \hat{\alpha}_n
\end{bmatrix}, \quad Q^T Q = I .
\] (3.1)

We arrive at the Lanczos algorithm by equating columns in \( AQ = QT \), where \( q_j \)'s are columns of \( Q \), \( Q = [q_1 | \ldots | q_n] \),
\[
A q_j = \beta_{j-1} q_{j-1} + \hat{\alpha}_j q_j + \hat{\beta}_j q_{j+1}.
\]
Paige [12] presents the following stable arrangement of the Lanczos algorithm. Choose \( \tilde{r}_1 \) to be nonzero,

\[
\tilde{\beta}_1 = \pm \| \tilde{r}_1 \|_2, \quad q_1 = \tilde{r}_1 / \tilde{\beta}_1, \quad u_1 = Aq_1
\]

For \( j = 1, 2, \ldots \)

\[
\begin{align*}
\tilde{\alpha}_j &= \langle q_j, Aq_j \rangle = \langle \tilde{r}_j, A\tilde{r}_j \rangle / \langle \tilde{r}_j, \tilde{r}_j \rangle \\
\tilde{r}_{j+1} &= u_j - \tilde{\alpha}_j q_j \\
\tilde{\beta}_{j+1} &= \pm \| \tilde{r}_{j+1} \|_2 \\
q_{j+1} &= \tilde{r}_{j+1} / \tilde{\beta}_{j+1} \\
u_{j+1} &= Aq_{j+1} - \tilde{r}_j.
\end{align*}
\]

In the same paper, Paige [12] shows that computing with the equivalent formula \( \tilde{\beta}_j = \langle q_{j+1}, Aq_j \rangle \) for \( \| \tilde{r}_j \|_2 \) leads to poor stability.

It is well known [5, page 370] that the CG algorithm is the Lanczos process where the normalized residual vectors form the \( Q \) matrix. From the CG procedure (2.2) we have

\[
p_1 = r_1, \quad p_j = r_j + \tilde{\beta}_j p_{j-1},
\]

or written in matrix form \( R_k = [r_1 | \ldots | r_k], P_k = [p_1 | \ldots | p_k], \)

\[
R_k = P_k L_k^T, \quad L_k = \begin{bmatrix} 1 & \beta_1 & \cdots & \beta_{k-2} & \beta_{k-1} \\ -\beta_1 & 1 & \cdots & \beta_{k-3} & \beta_{k-2} \\ \vdots & \ddots & \ddots & \ddots & \beta_{k-2} \\ -\beta_{k-1} & \cdots & \beta_1 & 1 \end{bmatrix}.
\]

Then \( Q_k = R_k \Delta_k^{-1} \) is orthogonal, where \( \Delta_k = \text{diag}(\|r_1\|_2, \ldots, \|r_k\|_2) \), and \( T_k = \)
\( Q_k^t AQ_k \) is tridiagonal,

\[
Q_k^t AQ_k = \left( R_k \Delta_k^{-1} \right)^t A \left( R_k \Delta_k^{-1} \right) = \Delta_k^{-1} P_k^t A P_k \Delta_k^{-1}
\]

\[
= \Delta_k^{-1} \left( P_k L_k \right)^t A \left( P_k L_k \right) \Delta_k^{-1} = \left( \Delta_k^{-1} L_k \right)^t \left( \Delta_k^{-1} L_k \right)^t
\]

\[
= \left( \Delta_k^{-1} L_k \right) D_k \left( \Delta_k^{-1} L_k \right)^t = T_k;
\]

\[
(D_k = \text{diag}(\sigma_1, \ldots, \sigma_k), \quad \sigma_j = \langle p_j, Ap_j \rangle)
\]

By equating the entries in (3.6) and (3.1), we have

\[
\hat{\alpha}_j = \frac{\sigma_j + \beta_j^2 \sigma_{j-1}}{\gamma_j} = \frac{1}{\alpha_j} + \frac{\beta_j}{\alpha_{j-1}}, \quad \hat{\beta}_j = \frac{-\beta_{j+1} \sigma_j}{\sqrt{\gamma_j} \sqrt{\gamma_{j+1}}} = \frac{-\sqrt{\beta_{j+1}}}{\alpha_j}.
\]

Note (3.6) with \( \hat{\alpha}_j = \langle q_j, Aq_j \rangle \) gives (2.7)

\[
\hat{\alpha}_j = \langle q_j, Aq_j \rangle = \frac{\langle r_j, Ar_j \rangle}{\gamma_j} = \frac{\delta_j}{\gamma_j} = \frac{\sigma_j + \beta_j^2 \sigma_{j-1}}{\gamma_j},
\]

\[
\langle r_j, Ar_j \rangle = \delta_j = \sigma_j + \beta_j^2 \sigma_{j-1}.
\]

Golub and Van Loan [5, page 342] present the application of the Lanczos process in solving linear equations, which is equivalent to the CG algorithm

\[
Q_k^t AQ_{ky_k} = Q_k^t b, \quad Q_{ky_k} = x_k
\]

\[
T_{ky_k} = \tilde{b}_k, \quad \tilde{b}_k = Q_k^t b
\]

The solution \( x_k = Q_{ky_k} \) can then be computed from (3.8) by the \( LDL^t \) factorization (3.6) of \( T_k \)

\[
x_k = Q_k T_k^{-1} Q_k^t b = Q_k T_k^{-1} \tilde{b}_k
\]

\[
= Q_k \left[ \left( \Delta_k^{-1} L_k \right) D_k \left( \Delta_k^{-1} L_k \right)^t \right]^{-1} \tilde{b}_k
\]

\[
= \left( P_k L_k \Delta_k^{-1} \right) \left[ \Delta_k L_k^{-t} D_k^{-1} L_k^{-1} \Delta_k \right] \tilde{b}_k,
\]

\[
= P_k a_k, \quad \text{where} \quad L_k D_k a_k = \Delta_k \tilde{b}_k.
\]
With further simplifications, one can show \( a_k = [\alpha_1, \ldots, \alpha_k]^t \) and this process can be rewritten in the more familiar form \( x_k = x_{k-1} + \alpha_k p_k \) of the CG algorithm.

We note that the above process requires computation of \( \tilde{\beta}_j = \pm \langle \tilde{r}_j, \tilde{r}_j \rangle \) and \( \tilde{\alpha}_j = \langle \tilde{r}_j, A\tilde{r}_j \rangle / \langle \tilde{r}_j, \tilde{r}_j \rangle \); as in the modified CG rearrangement (2.8), both inner products can be computed together. Moreover, the use of the alternative formula \( \tilde{\beta}_j = \langle q_{j+1}, Aq_j \rangle \) for \( \| \tilde{r}_j \|_2 \) leads to instability, as in the case of Saad’s rearrangement (1.1). Hence, algorithm (3.3) with (3.9) is a CG-like algorithm that differs from (2.8) in the computing of \( r_j \) from \( q_j \). The formulae

\[
p_j = r_j + \beta_j p_{j-1}, \quad r_{j+1} = r_j - \alpha_j A p_j
\]

from (2.8) can be shown to be equivalent to the three-term recurrence relation (3.2) of the Lanczos algorithm. We have

\[
r_{j+1} = r_j - \alpha_j A \left( r_j + \beta_j p_{j-1} \right) = r_j - \alpha_j Ar_j - (\alpha_j \beta_j / \alpha_{j-1}) (r_{j-1} - r_j) \\
= (1 + \alpha_j \beta_j / \alpha_{j-1}) r_j - \alpha_j Ar_j - (\beta_j \alpha_j / \alpha_{j-1}) r_{j-1} \\
= \tilde{\alpha}_j \alpha_j r_j - \alpha_j Ar_j - (\beta_j \alpha_j / \alpha_{j-1}) r_{j-1} \\
Ar_j = (\beta_j / \alpha_{j-1}) r_{j-1} + \tilde{\alpha}_j r_j - (1/\alpha_j) r_{j+1} \quad .
\]

By (3.6) and the relationships \( q_j = \tilde{r}_j / \tilde{\beta}_j = r_j / \sqrt{\gamma_j} \), (3.10) simplifies to (3.2). Thus in this light, the modified CG rearrangement (2.8) is naturally associated with the Lanczos process.

### 4. Numerical experiments on stability

The aim of the following experiments is to determine the stability and convergence properties of the modified conjugate gradient procedures.

We performed a number of MATLAB experiments in solving \( Ax = b \) by the conjugate gradient procedure to study the convergence behavior on different distributions of eigenvalues of the preconditioned matrix. In Eijkhout’s rearrangement, \( \langle r_k, v_{k-1} \rangle \) is computed by an extra inner product. Meurant’s rearrangement
is taken from [11] and the Lanczos rearrangement is adapted from [5, page 342] by evaluating the two inner products for \( \hat{\alpha}_j \) together as \( \hat{\alpha}_j = \langle \hat{r}_j, A\hat{r}_j \rangle / \langle \hat{r}_j, \hat{r}_j \rangle. \)

**Test 1**

The matrices considered have the eigenspectrum used by Strakos [18] and Greenbaum and Strakos [7]

\[
\lambda_i = \lambda_1 + \frac{i-1}{n-1} (\lambda_n - \lambda_1) \rho^{n-i}, \quad i = 2, \ldots, n, \quad \rho \in (0, 1). \tag{4.1}
\]

We have used \( n = 100, \lambda_1 = 1E-3, \kappa = \lambda_n / \lambda_1 = 1E5 \) and \( \rho = 0.6, 0.8, 0.9, 1.0 \) in the experiments. For \( \rho = 1 \), we have a uniformly distributed spectrum, and \( \rho < 1 \) describes quantitatively the clustering at \( \lambda_1 \).

**Test 2**

The eigenspectrum has a gap, \( \{1, \ldots, 50, 10051, \ldots, 10100\} \).

**Test 3**

The eigenspectrum has double eigenvalues, \( \{1, 1, 2, 3, \ldots, 50, 50\} \).

**Test 4**

The eigenspectrum consists of the roots of the Chebyshev polynomial \( T_n(x) \) shifted from \([-1, 1]\) to the interval \([a, b]\)

\[
\lambda_i = \frac{(b - a)}{2} \cos \left( \frac{\pi/2 + (i - 1)\pi}{n} \right) + \frac{(b + a)}{2}, \quad i = 1, \ldots, n. \tag{4.2}
\]

We have used \( n = 100, a = 1, b = 1E5. \)

As done in Hageman and Young [8], Greenbaum [6] and Strakos [18], we operate on diagonal matrices. This procedure is equivalent to representing all vectors over the basis of eigenvectors of matrix \( A \). In all cases, a random\(^1\) right

---

\(^1\)uniform over \([-1, 1]\)
Figure 4.1: Classical CG on Test 1. Dashed curve: $\rho = 0.6$; dotted curve: $\rho = 0.8$; dash-dot curve: $\rho = 0.9$; solid curve: $\rho = 1$.

Figure 4.2: Modified CG on Test 1. Dashed curve: $\rho = 0.6$; dotted curve: $\rho = 0.8$; dash-dot curve: $\rho = 0.9$; solid curve: $\rho = 1$. 
Figure 4.3: Eijkhout Rearrangement on Test 1. Dashed curve: $\rho = 0.6$; dotted curve: $\rho = 0.8$; dash-dot curve: $\rho = 0.9$; solid curve: $\rho = 1$.

Figure 4.4: Meurant Rearrangement on Test 1. Dashed curve: $\rho = 0.6$; dotted curve: $\rho = 0.8$; dash-dot curve: $\rho = 0.9$; solid curve: $\rho = 1$. 
Figure 4.5: Lanczos Rearrangement on Test 1. Dashed curve: \( \rho = 0.6 \); dotted curve: \( \rho = 0.8 \); dash-dot curve: \( \rho = 0.9 \); solid curve: \( \rho = 1 \).

Figure 4.6: Classical CG on Tests 2–4. Solid curve: Test 2; dashed curve: Test 3; dotted curve Test 4.
Figure 4.7: Modified CG on Tests 2–4. Solid curve: Test 2; dashed curve: Test 3; dotted curve Test 4.

Figure 4.8: Eijkhout Rearrangement on Tests 2–4. Solid curve: Test 2; dashed curve: Test 3; dotted curve Test 4.
Figure 4.9: Meurant Rearrangement on Tests 2–4. Solid curve: Test 2; dashed curve: Test 3; dotted curve Test 4.

Figure 4.10: Lanczos Rearrangement on Tests 2–4. Solid curve: Test 2; dashed curve: Test 3; dotted curve Test 4.
hand side and zero initial guess are used.

We display the decrease of $A$-norm of the error at each iteration divided by the $A$-norm of the initial error

$$
\frac{(\hat{x} - x_k, A(\hat{x} - x_k))^{1/2}}{(\hat{x} - x_0, A(\hat{x} - x_0))^{1/2}}, \quad \hat{x} = A^{-1}b. \quad (4.3)
$$

Figures 4.1-4.5, display the convergence results from Test 1. Note that for $\rho = 0.8, 0.9$ both the standard and modified CG procedures exhibit similar slow convergence behavior. Figures 4.6-4.10 display the convergence results on Tests 2-4. For Test 1 with $\rho = 0.6$, the standard CG algorithm shows the best convergence properties. Eijkhout’s rearrangement has slightly better stability properties than modified CG. The other results are essentially the same.

All the results on Tests 2-4 again show similar convergence behavior among the standard CG and the different rearrangements of CG.

5. Parallel performance

To gauge the effectiveness of the modified CG procedure, we performed a number of experiments in comparing the run-time in standard CG and modified CG. The test matrices are chosen from the Harwell-Boeing Test Collection [3]. The experiments are performed on 16 nodes of the iPSC/860 hypercube. Each matrix is first reordered by the bandwidth reducing Reverse Cuthill-McKee ordering [10]. The matrix is then equally block partitioned by rows and distributed across the processors in ELLPACK format [15]. In all cases, a random right hand side and zero initial guess are used, and convergence is assumed when

$$
\|r_k\|_2 \leq 10^{-8}\|r_0\|_2. \quad (5.1)
$$

The conjugate gradient procedure is rarely used without some form of preconditioning to accelerate convergence. In the tests described below, we use a block preconditioner derived as follows: Let $A_i$ be the diagonal block of the
1.5

Table 5.1: Description of test problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Order</th>
<th>Nonzeros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCSSTK13</td>
<td>2003</td>
<td>11973</td>
<td>Fluid Flow Generalized Eigenvalues</td>
</tr>
<tr>
<td>BCSSTK14</td>
<td>1806</td>
<td>32630</td>
<td>Root of Omni Coliseum, Atlanta</td>
</tr>
<tr>
<td>BCSSTK15</td>
<td>3948</td>
<td>60882</td>
<td>Module of an Offshore Platform</td>
</tr>
<tr>
<td>BCSSTK18</td>
<td>11948</td>
<td>80519</td>
<td>R.E.Ginna Nuclear Power Station</td>
</tr>
</tbody>
</table>

Table 5.2: Timing results.

<table>
<thead>
<tr>
<th>Problem</th>
<th>standard CG</th>
<th>modified CG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iterations</td>
<td>Time</td>
</tr>
<tr>
<td>BCSSTK13</td>
<td>1007</td>
<td>19.56</td>
</tr>
<tr>
<td>BCSSTK14</td>
<td>232</td>
<td>2.72</td>
</tr>
<tr>
<td>BCSSTK15</td>
<td>376</td>
<td>7.60</td>
</tr>
<tr>
<td>BCSSTK18</td>
<td>697</td>
<td>34.55</td>
</tr>
</tbody>
</table>

matrix $A$ contained in processor $i$, and write $A_i = L_i + D_i + L_i^T$ where $L_i$ is strictly lower triangular and $D_i$ is diagonal. Then the preconditioning matrix $M$ is $M = \text{diag}(M_1, M_2, \ldots, M_p)$, where $M_i = (L_i + D_i)D_i^{-1}(L_i + D_i)^T$. As shown in Axelsson and Barker [1], this corresponds to each processor doing a single SSOR step (with $\omega = 1$) on its diagonal block $A_i$. This preconditioner requires no added communication among the processors when implemented in parallel.

Table 5.1 is a brief description of the problems selected from the Harwell-Boeing Test Collection. Table 5.2 shows the number of iterations and time (in seconds) required to solve the corresponding problems. In all cases, the modified CG shows an improvement in the time required for solution, ranging from 5% to 13%. Moreover, the modified CG rearrangement shows no unstable behavior since it takes almost exactly the same number of iterations as standard CG.
6. Conclusion

We have presented a rearrangement of the standard conjugate gradient procedure that eliminates one synchronization point by performing two inner products at once. The rearrangement has a natural connection with the Lanczos process for solving linear equations. Although not a proof, MATLAB simulations indicate that the rearrangement is stable. Moreover, computational experiments using parallel versions of both the modified and standard conjugate gradient algorithms show that the modified version reduces the execution time by as much as 13% on an Intel iPSC/860 with 16 processors.

7. References


INTERNAL DISTRIBUTION

1. B.R. Appleton
2-3. T.S. Darland
4-8. E.F. D'Azevedo
9. J.M. Donato
10. J.J. Dongarra
11. G.A. Geist
12. M.R. Leuze
13. E.G. Ng
14. C.E. Oliver
15. B.W. Peyton
16-20. S.A. Raby
21-25. C.H. Romine
26. T.H. Rowan
27-31. R.F. Sincovec
32-36. R.C. Ward
37. P.H. Worley
38. Central Research Library
39. ORNL Patent Office
40. K-25 Appl Tech Library
41. Y-12 Technical Library
42. Lab Records Dept - RC
43-44. Laboratory Records Dept

EXTERNAL DISTRIBUTION

45. Cleve Ashcraft, Boeing Computer Services, P.O. Box 24346, M/S 7L-21, Seattle, WA 98124-0346
46. Donald M. Austin, 6196 EECS Bldg., University of Minnesota, 200 Union St., S.E., Minneapolis, MN 55455
47. Robert G. Babb, Oregon Graduate Institute, CSE Department, 19600 N.W. von Neumann Drive, Beaverton, OR 97006-1999
48. Lawrence J. Baker, Exxon Production Research Company, P.O. Box 2189, Houston, TX 77252-2189
49. Jesse L. Barlow, Department of Computer Science, Pennsylvania State University, University Park, PA 16802
50. Edward H. Barsis, Computer Science and Mathematics, P.O. Box 5800, Sandia National Laboratories, Albuquerque, NM 87185
51. Chris Bischof, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
52. Ake Bjorck, Department of Mathematics, Linkoping University, S-581 83 Linkoping, Sweden
53. Jean R. S. Blair, Department of Computer Science, Ayres Hall, University of Tennessee, Knoxville, TN 37996-1301
54. Roger W. Brockett, Wang Professor of Electrical Engineering and Computer Science, Division of Applied Sciences, Harvard University, Cambridge, MA 02138
55. James C. Browne, Department of Computer Science, University of Texas, Austin, TX 78712
56. Bill L. Buzbee, Scientific Computing Division, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307
57. Donald A. Calahan, Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109
59. Ian Cavers, Department of Computer Science, University of British Columbia, Vancouver, British Columbia V6T 1W5, Canada
60. Tony Chan, Department of Mathematics, University of California, Los Angeles, 405 Hilgard Avenue, Los Angeles, CA 90024
61. Jagdish Chandra, Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709
62. Eleanor Chu, Department of Mathematics and Statistics, University of Guelph, Guelph, Ontario, Canada N1G 2W1
63. Melvyn Ciment, National Science Foundation, 1800 G Street N.W., Washington, DC 20550
64. Tom Coleman, Department of Computer Science, Cornell University, Ithaca, NY 14853
65. Paul Concus, Mathematics and Computing, Lawrence Berkeley Laboratory, Berkeley, CA 94720
66. Andy Conn, IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598
67. John M. Conroy, Supercomputer Research Center, 17100 Science Drive, Bowie, MD 20715-4300
68. Jane K. Cullum, IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598
69. George Cybenko, Center for Supercomputing Research and Development, University of Illinois, 104 S. Wright Street, Urbana, IL 61801-2932
70. George J. Davis, Department of Mathematics, Georgia State University, Atlanta, GA 30303
71. Tim A. Davis, Computer and Information Sciences Department, 301 CSE, University of Florida, Gainesville, FL 32611-2024
72. John J. Dorning, Department of Nuclear Engineering Physics, Thornton Hall, McCormick Road, University of Virginia, Charlottesville, VA 22901
73. Iain Duff, Atlas Centre, Rutherford Appleton Laboratory, Didcot, Oxon OX11 0QX, England
74. Patricia Eberlein, Department of Computer Science, SUNY at Buffalo, Buffalo, NY 14260
75. Stanley Eisenstat, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520
76. Lars Elden, Department of Mathematics, Linkoping University, 581 83 Linkoping, Sweden
77. Howard C. Elman, Computer Science Department, University of Maryland, College Park, MD 20742
78. Albert M. Erisman, Boeing Computer Services, Engineering Technology Applications, ETA Division, P.O. Box 24346, MS-7L-20 Seattle, WA 98124-0346
79. Geoffrey C. Fox, Northeast Parallel Architectures Center, 111 College Place, Syracuse University, Syracuse, NY 13244-4100
80. Paul O. Frederickson, NASA Ames Research Center, R1ACS, M/S T045-1, Moffett Field, CA 94035
81. Fred N. Fritsch, L-316, Computing and Mathematics Research Division, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550
82. Robert E. Funderlic, Department of Computer Science, North Carolina State University, Raleigh, NC 27650
83. K. Gallivan, Computer Science Department, University of Illinois, Urbana, IL 61801
84. Dennis B. Gannon, Computer Science Department, Indiana University, Bloomington, IN 47405
85. Feng Gao, Department of Computer Science, University of British Columbia, Vancouver, British Columbia V6T 1W5, Canada
86. David M. Gay, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974
87. C. William Gear, NEC Research Institute, 4 Independence Way, Princeton, NJ 08540
88. W. Morven Gentleman, Division of Electrical Engineering, National Research Council, Building M-50, Room 344, Montreal Road, Ottawa, Ontario, Canada K1A 0R8
89. J. Alan George, Vice President, Academic and Provost, Needles Hall, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
90. John R. Gilbert, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304
91. Gene H. Golub, Department of Computer Science, Stanford University, Stanford, CA 94305
92. Joseph F. Grcar, Division 8245, Sandia National Laboratories, Livermore, CA 94551-0909
93. John Gustafson, Ames Laboratory, Iowa State University, Ames, IA 50011
94. Per Christian Hansen, UCI* C Lyngby, Building 305, Technical University of Denmark, DK-2800 Lyngby, Denmark
95. Richard Hanson, IMSL Inc., 2500 Park West Tower One, 2500 City West Blvd., Houston, TX 77042-3020
96. Michael T. Heath, National Center for Supercomputing Applications, 4157 Beckman Institute, University of Illinois, 405 North Mathews Avenue, Urbana, IL 61801-2300
97. Don E. Heller, Physics and Computer Science Department, Shell Development Co., P.O. Box 481, Houston, TX 77001
98. Nicholas J. Higham, Department of Mathematics, University of Manchester, Grt
Manchester, M13 9PL, England

Air Force Base, Washington, DC 20332

100. Robert E. Huddleston, Computation Department, Lawrence Livermore National
Laboratory, P.O. Box 808, Livermore, CA 94550

101. Ilse Ipsen, Department of Computer Science, Yale University, P.O. Box 2158 Yale
Station, New Haven, CT 06520

102. Barry Joe, Department of Computer Science, University of Alberta, Edmonton,
Alberta T6G 2H1, Canada

103. Lennart Johnsson, Thinking Machines Inc., 245 First Street, Cambridge, MA
02142-1214

104. Harry Jordan, Department of Electrical and Computer Engineering, University of
Colorado, Boulder, CO 80309

105. Bo Kagstrom, Institute of Information Processing, University of Umea, 5-901 87
Umea, Sweden

106. Malvyn H. Kalos, Cornell Theory Center, Engineering and Theory Center Bldg.,
Cornell University, Ithaca, NY 14853-3901

107. Hans Kaper, Mathematics and Computer Science Division, Argonne National Lab-
oratory, 9700 South Cass Avenue, Bldg. 221, Argonne, IL 60439

108. Linda Kaufman, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

109. Robert J. Kee, Division 8245, Sandia National Laboratories, Livermore, CA 94551-
0969

110. Kenneth Kennedy, Department of Computer Science, Rice University, P.O. Box
1892, Houston, TX 77001

111. Thomas Kitchens, Department of Energy, Scientific Computing Staff, Office of
Energy Research, ER-7, Office G-236 Germantown, Washington, DC 20585

112. Richard Lau, Office of Naval Research, Code 111MA, 800 Quincy Street, Boston
Tower 1, Arlington, VA 22217-5000

113. Alan J. Laub, Department of Electrical and Computer Engineering, University of
California, Santa Barbara, CA 93106

114. Robert L. Launer, Army Research Office, P.O. Box 12211, Research Triangle Park,
NC 27709

115. Charles Lawson, MS 301-490, Jet Propulsion Laboratory, 4800 Oak Grove Drive,
Pasadena, CA 91109

116. Peter D. Lax, Courant Institute of Mathematical Sciences, New York University,
251 Mercer Street, New York, NY 10012

117. James E. Leiss, Rt. 2, Box 142C, Broadway, VA 22815

118. John G. Lewis, Boeing Computer Services, P.O. Box 24346, M/S 7L-21, Seattle,
WA 98124-0346
119. Jing Li, IMSL Inc., 2500 Park West Tower One, 2500 City West Blvd., Houston, TX 77042-3020
120. Heather M. Liddell, Center for Parallel Computing, Department of Computer Science and Statistics, Queen Mary College, University of London, Mile End Road, London E1 4NS, England
121. Arno Liegmann, c/o ETH Rechenzentrum, Claustiusstr. 55, CH-8092 Zurich, Switzerland
122. Joseph Liu, Department of Computer Science, York University, 4700 Keele Street, North York, Ontario, Canada M3J 1P3
123. Robert F. Lucas, Supercomputer Research Center, 17100 Science Drive, Bowie, MD 20715-4300
124. Franklin Luk, Department of Computer Science, Amos Eaton Building - #131, Rensselaer Polytechnic Institute, Troy, NY 12180-3590
125. Thomas A. Manteuffel, Department of Mathematics, University of Colorado - Denver, Campus Box 170, P.O. Box 173364, Denver, CO 80217-3364
126. Consuelo Maulino, Universidad Central de Venezuela, Escuela de Computacion, Facultad de Ciencias, Apartado 47002, Caracas 1041-A, Venezuela
127. James McGraw, Lawrence Livermore National Laboratory, L-306, P.O. Box 808, Livermore, CA 94550
129. Cleve Moler, The Mathworks, 325 Linfield Place, Menlo Park, CA 94025
130. Neville Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green Street, Urbana, IL 61801
131. Dianne P. O'Leary, Computer Science Department, University of Maryland, College Park, MD 20742
132. James M. Ortega, Department of Applied Mathematics, Thornton Hall, University of Virginia, Charlottesville, VA 22901
134. Chris Paige, McGill University, School of Computer Science, McConnell Engineering Building, 3480 University Street, Montreal, Quebec, Canada H3A 2A7
135. Roy P. Pargas, Department of Computer Science, Clemson University, Clemson, SC 29634-1906
136. Beresford N. Parlett, Department of Mathematics, University of California, Berkeley, CA 94720
137. Merrell Patrick, Department of Computer Science, Duke University, Durham, NC 27706
138. Robert J. Plemmons, Departments of Mathematics and Computer Science, Box 7311, Wake Forest University, Winston-Salem, NC 27109
139. Jesse Poore, Department of Computer Science, Ayres Hall, University of Tennessee, Knoxville, TN 37996-1301
140. Alex Pothen, Department of Computer Science, Pennsylvania State University, University Park, PA 16802
141. Yuanchang Qi, IBM European Petroleum Application Center, P.O. Box 585, N-4040 Hafrsfjord, Norway
142. Giuseppe Radicati, IBM European Center for Scientific and Engineering Computing, via del Giorgione 159, I-00147 Roma, Italy
143. John K. Reid, Numerical Analysis Group, Central Computing Department, Atlas Centre, Rutherford Appleton Laboratory, Didcot, Oxon OX11 0QX, England
144. Werner C. Rheinboldt, Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, PA 15260
145. John R. Rice, Computer Science Department, Purdue University, West Lafayette, IN 47907
146. Garry Rodrigue, Numerical Mathematics Group, Lawrence Livermore Laboratory, Livermore, CA 94550
147. Donald J. Rose, Department of Computer Science, Duke University, Durham, NC 27706
148. Edward Rothberg, Department of Computer Science, Stanford University, Stanford, CA 94305
149. Axel Ruhe, Dept. of Computer Science, Chalmers University of Technology, S-41296 Goteborg, Sweden
150. Joel Saltz, ICASE, MS 132C, NASA Langley Research Center, Hampton, VA 23665
151. Ahmed H. Sameh, Center for Supercomputing R&D, 1384 W. Springfield Avenue, University of Illinois, Urbana, IL 61801
152. Michael Saunders, Systems Optimization Laboratory, Operations Research Department, Stanford University, Stanford, CA 94305
153. Robert Schreiber, RIACS, Mail Stop 230-5, NASA Ames Research Center, Moffett Field, CA 94035
154. Martin H. Schultz, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520
155. David S. Scott, Intel Scientific Computers, 15201 N.W. Greenbrier Parkway, Beaverton, OR 97006
156. Lawrence F. Shampine, Mathematics Department, Southern Methodist University, Dallas, TX 75275
157. Andy Sherman, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520
158. Kermit Sigmon, Department of Mathematics, University of Florida, Gainesville, FL 32611
159. Horst Simon, Mail Stop T045-1, NASA Ames Research Center, Moffett Field, CA 94035
160. Anthony Skjellum, Lawrence Livermore National Laboratory, 7000 East Ave., L-316, P.O. Box 808 Livermore, CA 94551
161. Danny C. Sorensen, Department of Mathematical Sciences, Rice University, P.O. Box 1892, Houston, TX 77251
162. G. W. Stewart, Computer Science Department, University of Maryland, College Park, MD 20742
163. Paul N. Swartztrauber, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307
164. Philippe Toint, Dept. of Mathematics, University of Namur, FUNOP, 61 rue de Bruxelles, B-Namur, Belgium
165. Bernard Tourancheau, LIP, ENS-Lyon, 69364 Lyon cedex 07, France
166. Hank Van der Vorst, Dept. of Techn. Mathematics and Computer Science, Delft University of Technology, P.O. Box 356, NL-2600 AJ Delft, The Netherlands
167. Charles Van Loan, Department of Computer Science, Cornell University, Ithaca, NY 14853
168. Jim M. Varah, Centre for Integrated Computer Systems Research, University of British Columbia, Office 2053-2324 Main Mall, Vancouver, British Columbia V6T 1W5, Canada
169. Udaya V. Vemulapati, Dept. of Computer Science, University of Central Florida, Orlando, FL 32816-0362
170. Robert G. Voigt, ICASE, MS 132-C, NASA Langley Research Center, Hampton, VA 23665
171. Phuong Vu, Cray Research, Inc., 19607 Franz Rd., Houston, TX 77084
172. Daniel D. Warner, Department of Mathematical Sciences, O-104 Martin Hall, Clemson University, Clemson, SC 29631
173. Mary F. Wheeler, Rice University, Department of Mathematical Sciences, P.O. Box 1892, Houston, TX 77251
174. Andrew B. White, Computing Division, Los Alamos National Laboratory, P.O. Box 1663, MS-265, Los Alamos, NM 87545
175. Margaret Wright, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974
176. David Young, University of Texas, Center for Numerical Analysis, RLM 13.150, Austin, TX 78731
177. Earl Zmijewski, Department of Computer Science, University of California, Santa Barbara, CA 93106
179–188. Office of Scientific & Technical Information, P.O. Box 62, Oak Ridge, TN 37831