Parallel Computing 14 (1990) 191-198
North-Holland

Vectorization of some block preconditioned conjugate gradient methods *

L. BRUGNANO and M. MARRONE

Dipartimento di Matematica, Università di Bari, Via G. Fortunato, 70125 Bari, Italy

Received June 1989
Revised November 1989

Abstract. The block preconditioned conjugate gradient methods are very effective to solve the linear systems arising from the discretization of elliptic PDE. Nevertheless, the solution of the linear system $Ms = r$, to get the preconditioned residual, is a 'bottleneck', on vector processors. In this paper, we show how to modify the algorithm, in order to get better performances, on such computers. Numerical tests carried out on a CRAY X-MP/48 are presented, in order to give numerical evidence.

Keywords. Linear algebra, block preconditioning, preconditioned conjugate gradient methods, vector processors

1. Introduction: Block preconditioners

Let us consider the block tridiagonal, symmetric, nonsingular M-matrix [7]

$$A = \begin{bmatrix} S_1 & F_2^{\mathrm{T}} & & & & \\ F_2 & S_2 & F_3^{\mathrm{T}} & & & \\ & F_3 & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & F_n^{\mathrm{T}} \\ & & & & F_n & S_n \end{bmatrix}, \quad S_i,\ F_i \in \mathbb{R}^{k \times k}. \tag{1.1}$$

Let us suppose A be weakly diagonally dominant, and consider the case in which the blocks S_i are tridiagonal, and the blocks F_i are diagonal. Very effective preconditioners are obtained by using sparse block factorizations of A:

$$\begin{aligned} T_1 &= S_1, \\ T_i &= S_i - F_i \Sigma_{i-1} F_i, \quad i = 2, \ldots, n, \end{aligned} \tag{1.2}$$

Σ_{i-1} being a sparse approximation of T_{i-1}^{-1}. Different choices of Σ_{i-1} will give different preconditioners [1-3]. In such a way, we get

$$M = LD^{-1}L^{\mathrm{T}} = A + R, \tag{1.3}$$

* Work performed within the activities of the project "Matematica Computazionale" supported by the "Ministero della Pubblica Istruzione 40%".

0167-8191/90/$03.50 © 1990 - Elsevier Science Publishers B.V. (North-Holland)

where $D = \mathrm{diag}(T_i)$, and L is a lower block bidiagonal matrix, with the main diagonal equal to the one of D, and the lower diagonal equal to the one of A. It is simple to show that $R = \mathrm{diag}(R_i)$ [3],

$$\begin{aligned} &R_1 = T_1 - S_1 = 0, \\ &R_i = F_i\left(T_{i-1}^{-1} - \Sigma_{i-1}\right)F_i, \quad i = 2,\ldots,n. \end{aligned} \tag{1.4}$$

We consider the case in which the blocks T_i have the same sparsity as the diagonal blocks S_i of A. In particular, we are going to consider the standard INV and MINV preconditioner [2–4], even if the analysis could be extended to other block preconditioners.

The INV preconditioner is obtained by considering $\Sigma_{i-1} = \mathrm{trid}(T_{i-1}^{-1})$, where $\mathrm{trid}(T_{i-1}^{-1})$ is the tridiagonal part of T_{i-1}^{-1}. The MINV preconditioner is the modified version of INV, obtained by imposing that Σ_{i-1} and T_{i-1}^{-1} have the same row-sums [3]. We consider again the tridiagonal part of the inverse, in (1.2), but the neglected elements of the inverse are summed, on each row, to the corresponding diagonal element.

One can show [3] that the blocks T_i of INV and MINV are symmetric diagonally dominant M-matrices.

2. Vectorizing the solution of $Ms = r$

To get the preconditioned residual s, in a preconditioned conjugate gradient method, we must solve $Ms = r$, where r is the current residual:

$$\begin{aligned} &T_1 y_1 = r_1, && s_n = y_n, \\ &T_i y_i = r_i - F_i y_{i-1}, \quad i = 2,\ldots,n, && T_i(s_i - y_i) = -F_{i+1}s_{i+1}, \quad i = n-1,\ldots,1. \end{aligned} \tag{2.1}$$

Let us consider, now, the generic block T_i:

$$T_i = \begin{bmatrix} a_1^{(i)} & -b_2^{(i)} & & \\ -b_2^{(i)} & \ddots & \ddots & \\ & \ddots & \ddots & -b_k^{(i)} \\ & & -b_k^{(i)} & a_k^{(i)} \end{bmatrix}. \tag{2.2}$$

It could be factorized as $T_i = L_{T_i} D_{T_i} L_{T_i}^{\mathrm{T}}$, where

$$L_{T_i} = \begin{bmatrix} d_1^{(i)} & & & \\ -b_2^{(i)} & \ddots & & \\ & \ddots & \ddots & \\ & & -b_k^{(i)} & d_k^{(i)} \end{bmatrix}, \qquad D_{T_i} = diag\left(d_1^{(i)},\ldots,d_k^{(i)}\right),$$

$$\begin{aligned} &d_1^{(i)} = a_1^{(i)}, \\ &d_j^{(i)} = a_j^{(i)} - \frac{\left(b_j^{(i)}\right)^2}{d_{j-1}^{(i)}}, \quad j = 2,\ldots,k. \end{aligned} \tag{2.3}$$

If we scale T_i to get $D_{T_i} = I$, we get $T_i = (I - E_i)(I - E_i^{\mathrm{T}})$, where

$$E_i = \begin{bmatrix} 0 & & & \\ -\tilde{b}_2^{(i)} & 0 & & \\ & \ddots & \ddots & \\ & & -\tilde{b}_k^{(i)} & 0 \end{bmatrix}. \tag{2.4}$$

From the diagonal dominance of T_i, one gets $\rho(E_i) < 1$. It follows that

$$T_i^{-1} = \left(I - E_i^{\mathrm{T}}\right)^{-1}\left(I - E_i\right)^{-1} = \left(I + E_i^{\mathrm{T}} + \cdots + E_i^{\mathrm{T}^{k-1}}\right)\left(I + E_i + \cdots + E_1^{k-i}\right). \tag{2.5}$$

We can use a truncated expansion of (2.5) as approximation of T_i^{-1}:

$$T_i^{-1} \approx \tilde{T}_i^{-1} = (I + E^{\mathrm{T}} + \cdots + E^{\mathrm{T}^m})(I + E + \cdots + E^m), \tag{2.6}$$

where, obviously $m \ll k - 1$. It follows that the solution of the tridiagonal subsystems in (2.1) can be replaced with vectorizable operations. Now we want to get an estimate of the error due to the use of (2.6), instead of (2.5). For sake of simplicity, let us assume $F_i = -I$ (this is a common case). In such a way, instead of M we get

$$\tilde{M} = \begin{bmatrix} \tilde{T}_1 & -I & & & \\ -I & \tilde{T}_2 + \tilde{T}_1^{-1} & -I & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & \tilde{T}_n + \tilde{T}_{n-1}^{-1} \end{bmatrix}$$

$$= M + \begin{bmatrix} \tilde{T}_1 - T_1 & & \\ & \ddots & \\ & & \tilde{T}_n - T_n \end{bmatrix} + \begin{bmatrix} 0 & & \\ \tilde{T}_1^{-1} - T_1^{-1} & & \\ & \ddots & \\ & & \tilde{T}_{n-1}^{-1} T_{n-1}^{-1} \end{bmatrix}$$

$$= M + P_1 + P_2 = M + \tilde{R}.$$

Referring to (1.3), we choose m so that $\|\tilde{R}\| \leqslant \|R\|$ (from now on, $\|\cdot\|$ denotes $\|\cdot\|_\infty$), that is, the error introduced by the truncation must be at most the error made in the construction of the preconditioner. It follows that we need an estimate for $\|P_1\|$, $\|P_2\|$ and $\|R\|$ (while $\|R\|$ is available immediately at run time, if the MINV preconditioner is used; this is not true for INV).

Now we show how to get an estimate for $\|P_1\|$. Observe that

$$(I + E + \cdots + E^m)^{-1} = (I - E)(I - E^{m+1})^{-1} \approx (I - E)(I + E^{m+1}).$$

It follows, from (2.6),

$$\tilde{T}_i \approx T_i + (I - E_i)\left(E_i^{m+1} + E_i^{\mathrm{T}^{m+1}}\right)\left(I - E_i^{\mathrm{T}}\right),$$

that is

$$\|P_1\| \approx 2\beta_1^{m+1}(1 + \beta_1)^2, \tag{2.7}$$

where

$$\beta_1 = \max_{i=1,\dots,n} \|E_i\|. \tag{2.8}$$

Now we are looking for an estimate of $\| P_2 \|$. If we define the matrices

$$S_i = T_i^{-1} - \tilde{T}_i^{-1},$$

we get

$$\| S_i \| < 2 \left\| E_i^{m+1} \right\| \left\| T_i^{-1} \right\| = \eta_i.$$

It follows that

$$\| P_2 \| < \max_{i=1,\dots,n-1} \eta_i \leqslant 2\beta_2^{m+1}\gamma, \tag{2.9}$$

where

$$\beta_2 = \max_{i=1,\dots,n-1} \| E_i \|, \qquad \gamma = \max_{i=1,\dots,n-1} \left\| T_i^{-1} \right\|.$$

Finally, we get

$$\| \tilde{R} \| \approx 2\left(\beta_1^{m+1}(1+\beta_1)^2 + \gamma\beta_2^{m+1} \right) = \xi(m). \tag{2.10}$$

At last, let us obtain an estimate of $\| R \|$ for INV.

Let T_i the generic diagonal block of the INV factorization, whose structure is given in (2.2), and let $T_i^{-1} = (t_{rs}^{(i)})$. One gets [4]

$$t_{kk}^{(i)} = 1/d_k^{(i)},$$

$$t_{rk}^{(i)} = t_{r+1,k}^{(i)} b_{r+1}^{(i)} / d_r^{(i)}, \quad r = k-1,\dots,1,$$

$$\text{for } s = k-1,\dots,1,$$

$$t_{ss}^{(i)} = \left(1 + t_{s,s+1}^{(i)} b_{s+1}^{(i)}\right)/d_s^{(i)},$$

$$t_{sr}^{(i)} = t_{rs}^{(i)} = t_{r+1,s}^{(i)} b_{r+1}^{(i)} / d_r^{(i)}, \quad r = s-1,\dots,1.$$

With the assumption of T_i being diagonally dominant, we get that the elements on the rows of T_i^{-1} become smaller and smaller, as we go away from the main diagonal. Moreover, as we are considering the case in which $F_i = -I$, we get (see (1.4))

$$\| R \| = \max_{i=1,\dots,n-1} \left\| T_i^{-1} - \operatorname{trid}\left(T_i^{-1}\right) \right\|.$$

It follows that

$$\delta_{\min} \sum_{j=2}^{k-1} \sigma_{\min}^j \leqslant \| R \| \leqslant 2\delta_{\max} \sum_{j=2}^{(k-1)/2} \sigma_{\max}^j,$$

where

$$\delta_{\min} = \min_{i,j}\left(t_{jj}^{(i)}\right), \qquad \delta_{\max} = \max_{i,j}\left(t_{jj}^{(i)}\right),$$

$$\sigma_{\min} = \min_{i,j}\left(b_{j+1}^{(i)}/d_j^{(i)}\right), \quad \sigma_{\max} = \max_{i,j}\left(b_{j+1}^{(i)}/d_j^{(i)}\right),$$

that is

$$r_1 = \delta_{\min}\sigma_{\min}^2 \frac{1-\sigma_{\min}^{k-2}}{1-\sigma_{\min}} \leqslant \| R \| \leqslant 2\delta_{\max}\sigma_{\max}^2 \frac{1-\sigma_{\max}^{(k-2)/2}}{1-\sigma_{\max}} = r_2. \tag{2.11}$$

In such a way, we get an interval estimate for $\| R \|$. Moreover, with reference to (2.9) we get

$$\gamma \leqslant 2\delta_{\max} \frac{1-\sigma_{\max}^{k/2}}{1-\sigma_{\max}} = \gamma_1.$$

If the interval $\mathscr{L}=[r_1, r_2]$ is 'sufficiently small', any value of m for which $\xi(m)\in\mathscr{L}$ (see (2.10)) is acceptable. Otherwise, we should remain as much as possible close to r_1.

3. Numerical tests

We have considered three test problems. For each problem, we have compared the original preconditioner, INV or MINV, with the truncated ones, for various choices of m. Moreover, we have compared those preconditioners with the one suggested by Meurant [5], in which the inverse of the generic tridiagonal block is approximated by a band matrix, with the main seven diagonals coinciding with those of the exact inverse.

Let us briefly examine the computational cost of the considered preconditioned conjugate gradient methods, in terms of requested memory and operations (per iterate). The acrostic TRUNC(m) is for the method using the truncated preconditioner, while MEUR is for the one proposed by Meurant. The acrostics MTRUNC and MMEUR are for the modified versions. N is the dimension of the problem. See Table 1.

The tests were carried out on a single processor of a CRAY X-MP/48. The language used is FORTRAN (CFT). To get the execution time (in seconds), we got, for each problem and method used, the minimum time of 50 executions.

Test problem 1. The first test problem derives from the discretization of

$$-\Delta u=f, \quad \text{on } \Omega=(0, 1)\times(0, 1),$$
$$u\,|_{\partial\Omega}=0.$$

The usual 5-points scheme is used, with step size $h=(n+1)^{-1}$, $N=n^2$, where n is the mesh size, $n=100$.

Test problem 2. The second problem derives from the discretization of

$$-\lambda\ \Delta u=f \quad \text{on } \Omega=(0, 1)\times(0, 1),$$
$$u\,|_{\partial\Omega}=0,$$

where

$$\lambda=\begin{cases}1000 & \text{on } \Omega_1\\ 1 & \text{on } \Omega_2\end{cases},$$

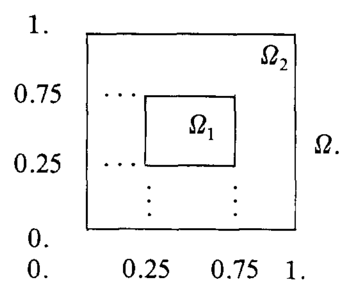


Table 1

Method	Memory	Vectorizable operations	Non vectorizable operations
INV/MINV	$10N$	$32N$	$8N$
TRUNC/MTRUNC(3)	$11N$	$40N$	–
TRUNC/MTRUNC(7)	$12N$	$48N$	–
TRUNC/MTRUNC(15)	$13N$	$56N$	–
MEUR/MMEUR	$12N$	$48N$	–

The usual 5-points scheme is used, with step size $h = (n+1)^{-1}$, $N = n^2$, where n is the mesh size, $n = 100$.

In both Problems 1 and 2 we got for INV the following estimates (see (2.11) and (2.10)):

$$r_1 = 0.089, \quad r_2 = 0.4915, \quad \| R \| = 0.4915 \text{ (true value)},$$
$$\xi(3) = 0.1639, \quad \xi(7) = 0.0225, \quad \xi(16) = 4\text{E}{-7}.$$

For MINV we got

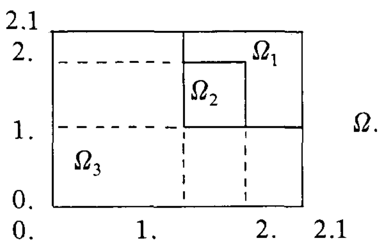
$$r_1 = 0.178, \quad r_2 = 1.8656, \quad \| R \| = 1.8656 \text{ (true value)},$$
$$\xi(3) = 0.4178, \quad \xi(7) = 0.0115, \quad \xi(16) = 8\text{E}{-6}.$$

Test problem 3. The third problem derives from the discretization of

$$-\lambda\, \Delta u + \sigma u = \sigma \quad \text{on } \Omega = (0, 2.1) \times (0, 2.1),$$
$$\left. \frac{\partial u}{\partial z} \right|_{\partial\Omega} = 0,$$

where z the unit vector normal to $\partial\Omega$, and

$$\lambda = \begin{cases} 1 & \text{on } \Omega_1, \\ 2 & \text{on } \Omega_2, \\ 3 & \text{on } \Omega_3, \end{cases} \qquad \sigma = \begin{cases} 0.02 & \text{on } \Omega_1, \\ 0.03 & \text{on } \Omega_2, \\ 0.05 & \text{on } \Omega_3, \end{cases}$$



The usual 5-points scheme is used, with step size $h = (n+1)^{-1}$, $N = n^2$, where n is the mesh size, $n = 90$.

For this problem the value of β_1 (see (2.8)) in the relation (2.7) is obtained in correspondence of the last element of the last block. We get

$$\| P_1 \| = \| T_n - \tilde{T}_n \| \approx \| I + E_n \|^2 \| E_n^{m+1} \|$$

where

$$E_n = \begin{bmatrix} 0 & & & & \\ \tilde{\beta}_1 & 0 & & & \\ & \ddots & \ddots & & \\ & & \tilde{\beta}_1 & 0 & \\ & & & \beta_1 & 0 \end{bmatrix},$$

and (see (2.9) and (2.4))

$$\tilde{\beta}_1 = \max\Big(\beta_2, \max_{i=2,\dots,k} \{ \tilde{b}_i^{(n)} \} \Big), \quad \beta_1 - \tilde{\beta}_1 = \epsilon > 0.$$

It follows that, instead of (2.7), we get

$$\| P_1 \| \approx 2 \tilde{\beta}_1^m \beta_1 (1 + \beta_1)^2.$$

Table 2
Test problem 1

Method	Speedup	Time	Iterates
INV	1.0	0.656	28
TRUNC(3)	3.2	0.206	31
TRUNC(7)	3.0	0.220	28
TRUNC(15)	2.6	0.248	28
MEUR	2.9	0.228	30
MINV	1.0	0.464	20
MTRUNC(3)	3.3	0.142	22
MTRUNC(7)	2.5	0.184	21
MTRUNC(15)	2.5	0.183	20
MMEUR	2.8	0.167	22

Table 3
Test problem 2

Method	Speedup	Time	Iterates
INV	1.0	0.712	30
TRUNC(3)	3.1	0.227	34
TRUNC(7)	3.0	0.236	30
TRUNC(15)	2.7	0.266	30
MEUR	2.8	0.252	33
MINV	1.0	0.440	19
MTRUNC(3)	2.7	0.163	24
MTRUNC(7)	2.6	0.171	20
MTRUNC(15)	2.5	0.176	19
MMEUR	2.4	0.183	23

We obtain for INV,

$$r_1 = 0.2457, \qquad r_2 = 2.7676, \qquad \| R \| = 0.8446 \text{ (true value)},$$

$$\xi(3) = 9.5059, \qquad \xi(7) = 2.5945, \qquad \xi(16) = 0.1933.$$

Observe that, even if $\xi(3) \notin [r_1, r_2]$, the truncated preconditioner TRUNC(3) is effective, as we are going to see.

For each method, and each problem, we give the speedup with respect to the original preconditioner (INV/MINV), the execution time, and the number of iterations to get convergence, see Tables 2–4. The stopping criterion used is $\| r_i \|_2 / \| r_0 \|_2 < 10^{-6}$, where r_i is the residual at the ith step, and r_0 the initial residual. The initial point is $x_0 = 0$.

Table 4
Test problem 3

Method	Speedup	Time	Iterates
INV	1.0	1.349	69
TRUNC(3)	3.0	0.448	79
TRUNC(7)	2.9	0.460	69
TRUNC(15)	2.5	0.530	69
MEUR	2.4	0.569	80

References

[1] O. Axelson, A survey of preconditioned iterative methods for linear systems of algebraic equations, *BIT* **25** (1985) 166–187.

[2] L. Brugnano and M. Marrone, Metodi dei gradienti coniugati: Teoria, Tecniche di precondizionamento, ed Implementazione su calcolatori vettoriali, Quaderno n.ro 1/89 del Dipartimento di Matematica dell' Università di Bari, Italy.

[3] P. Concus, G.H. Golub and G. Meurant, Block preconditioning for the conjugate gradient method, *SIAM J. Sci. Statist. Comput.* **6** (1) (1985) 220–252.

[4] P. Concus and G. Meurant, On computing the INV block preconditionings for the conjugate gradient method, *BIT* **26** (1986) 493–504.

[5] G. Meurant, Multitasking the conjugate gradient method on the CRAY X-MP/48, *Parallel Comput.* **5** (1987) 267–280.

[6] H.A. Van Der Vorst, A vectorizable variant of some ICCG methods, *SIAM J. Sci. Statist. Comput.* **3** (3) (1982) 350–356.

[7] R.S. Varga, *Matrix Iterative Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1962).