J. KSIAM **Vol.5**, No.2, 63-69, 2001

# A PARALLEL PRECONDITIONER FOR GENERALIZED EIGENVALUE PROBLEMS BY CG-TYPE METHOD

SANGBACK MA AND HO-JONG JANG

Abstract. In this study, we shall be concerned with computing in parallel a few of the smallest eigenvalues and their corresponding eigenvectors of the eigenvalue problem, $Ax = \lambda Bx$, where $A$ is symmetric, and B is symmetric positive definite. Both A and B are large and sparse. Recently iterative algorithms based on the optimization of the Rayleigh quotient have been developed, and CG scheme for the optimization of the Rayleigh quotient has been proven a very attractive and promising technique for large sparse eigenproblems for small extreme eigenvalues. As in the case of a system of linear equations, successful application of the CG scheme to eigenproblems depends also upon the preconditioning techniques. A proper choice of the preconditioner significantly improves the convergence of the CG scheme. The idea underlying the present work is a parallel computation of the Multi-Color Block SSOR preconditioning for the CG optimization of the Rayleigh quotient together with deflation techniques. Multi-Coloring is a simple technique to obatin the parallelism of order $n$, where $n$ is the dimension of the matrix. Block SSOR is a symmetric preconditioner which is expected to minimize the interprocessor communication due to the blocking. We implemented the results on the CRAY-T3E with 128 nodes. The MPI(Message Passing Interface) library was adopted for the interprocessor communications. The test problems were drawn from the discretizations of partial differential equations by finite difference methods.

## 1. Introduction

Recently, there has been many efforts to find the small extreme eigenvalues of the generalized eigenproblem by iterative Rayleigh-quotient minimization methods by CG-type methods[2,4,5,12]. Such applications arise in many cases, such as in structural mechanics or computational chemistry, to name a few. In this paper we propose a parallel version of the same method with incomplete Cholesky factorization preconditioner on the CRAY-T3E. After the smallest eigenvalue is found, the PCG scheme together with deflation technique is used to compute the next eigenvalues. We offer a simple technique called the orthogonal deflation[6].

Iterative solution of eigenvalue problems or linear systems requires a preconditioning to accelerate the convergence. Incomplete Cholesky factorization is one of the most popular technique.

---

Key words: Generalized Eigenvalue Problem, CG, Preconditioning, Parallel, Multi-Color Block SSOR.

Parallel processing is a simple way to increase the speed. But incomplete Cholesky factorization is inherently *serial*. In this paper we use a block-type parallel preconditioner, Multi-Color Block SSOR(Symmetric Successive OverRelaxarion) preconditioner.

The CRAY-T3E computer in ETRI, Korea is a massively parallel message-passing machine with the 136 individual processing node(PE)s interconnected in a 3D-Torus structure. Each PE, a DEC Alpha EV5.6 chip, is capable of delivering up tp 900 Megaflops,amounting to 115 GigaFlops in total. Each PE has 128 MBs of core memory.

We present results from our numerical experiments drawn from the FDM discretizations of the elliptic partial differential equations.

## 2. Computation of the leftmost eigenpairs

Let $A$ and $B$ be sparse symmetric positive definite matrices of dimension $n$. Consider the generalized eigenvalue problem

$$Ax = \lambda Bx. \tag{1}$$

Denote by $0 < \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$ and $z_1, z_2, \cdots, z_n$ the eigenvalues and the corresponding eigenvectors.

We recall that the eigenvectors of (1) are the stationary points of the Rayleigh quotient

$$R(x) = \frac{x^T Ax}{x^T Bx}, \tag{2}$$

and the gradient of $R(x)$ is given by

$$g(x) = \frac{2}{x^T Bx}[Ax - R(x)Bx].$$

To simplify the notation we set $g^{(k)} = g(x^{(k)}) = 2r^{(k)}/x^T Bx,\ r^{(k)}$ being the residual vector. To compute a number of the leftmost eigenpairs of (1), the PCG with partial deflation was first proposed [12]. This scheme evaluates one eigenpair at a time by a deflation procedure requiring the assessment of a shifting parameter, which is problem dependent. Later, the PCG with orthogonal deflation has been developed [6], allowing for the simultaneous evaluation of the leftmost eigenpairs of (1). PCG with orthogonal deflation does not need any acceleratioin parameter, and is more suited to parallelization.

The basic idea underlying PCG with orthogonal deflation is as follows. Assume that the eigenpairs $(\lambda_i, z_i),\ i = 1, \cdots, r-1$, have been computed. To avoid convergence toward one of the computed eigenvectors $z_i,\ i = 1, \cdots, r-1$, the next initial vector $\tilde{x}_r^{(0)}$ is chosen to be $B$-orthogonal to $Z_{r-1} = \text{span}\{z_i \mid i = 1, \cdots, r-1\}$. And the direction vector $\tilde{p}_r^{(k)}$ is evaluated by $B$-orthogonalizing $p_r^{(k)}$ with respect to $Z_{r-1}$. Also the new approximation vector $\tilde{x}_r^{(k)}$ is evaluated by $B$-normalizing $x_r^{(k)}$. Now from the characterization of the eigenvectors [8]

$$R(z_r) = \min_{x \perp_B Z_{r-1}} R(x),$$

$\tilde{x}_r^{(k)}$ converges toward $z_r$ as $k$ increases. That is, after $z_i,\ i = 1, \cdots, r-1$ have been evaluated, $z_r$ can be determined by minimizing $R(x)$ over the vector space which is the $B$-orthogonal complement to $Z_{r-1}$.

The minimization is performed by the PCG scheme. In present work, a Multi-Color Block SSOR preconditioner $M$, which will be discussed in §3, is used with parallel computation aspect.

To achieve convergence, usually a recurring 'restart' operation is used [6]. Alternatively an appropriate choice of the $\beta$ parameter in the PCG scheme may avoid the restart. Following [9],

$$\beta^{(k)} = \frac{g^{(k)^T} g^{(k)}}{g^{(k-1)^T} g^{(k-1)}} \tag{3}$$

is set.

The PCG scheme with orthogonal deflation consists of the following steps:

*Step 1.* Compute the preconditioner $M$.
*Step 2.* Give an initial vector $x^{(0)}$ such that $Z^T B x^{(0)} = 0$ (i.e. $x^{(0)}$ is taken to be $B$-orthgonal to $Z$. Choose a tolerance value and the allowed maximum number of iterations NMAX. Set $k = 0$ (iteration index).
*Step 3.* Construct the initial gradient direction $g^{(0)}$.

$$\text{Set } p^{(0)} = -g^{(0)} \text{ and } Mh^{(0)} = g^{(0)}.$$

*Step 4.* If $k = 0$ then set $\beta^{(k)} = 0$, otherwise evaluate

$$Mh^{(k)} = g^{(k)} \text{ and } \beta^{(k)} \text{ by (3).}$$

*Step 5.* Compute $\tilde{p}^{(k+1)} = h^{(k)} + \beta^{(k)} p^{(k)}$.
*Step 6.* Evaluate $p^{(k)}$ by $B$-orthogonalizing $\tilde{p}^{(k)}$ with respect to $Z$.
*Step 7.* Compute $\alpha^{(k+1)}$ by minimizing $R(x^{(k+1)})$ [7].
*Step 8.* Evaluate $\tilde{x}^{(k+1)} = x^{(k)} + \alpha^{(k+1)} p^{(k+1)}$.
*Step 9.* The new approximation $x^{(k+1)}$ is evaluated by $B$-normalizing $\tilde{x}^{(k+1)}$.
*Step 10.* Test on convergence.

## 3. Multi-Color Block SSOR preconditioning

Multi-Coloring is a way to achieve parallelism of order $N$, where $N$ is the order of the matrix. For example, it is known that for 5-point Laplacian we can order the matrix in 2-colors so that the nodes are not adjacent with the nodes with the same color. This is known as Red/Black ordering. For planar graphs maximum four colors are needed.

Blocked methods are useful in that they minimize the interprocessor communications, and increases the convergence rate as compared to point methods. SSOR is a symmetric preconditioner that is expected to perform as efficiently as Incomplete Cholesky factorization combined with blocking. Instead we need to invert the diagonal block. In this paper we used the MA48 package from the Harwell library, which is a direct method using reordering strategy to reduce the fill-ins. Since MA48 type employ some form of pivoting strategy, this is expected to perform better for ill-conditioned matrices than Incomplete Cholesky factorization, which does not adopt any type of pivoting strategy.

SSOR needs a $\omega$ parameter for overrelaxation. However, it is known that the convergence rate is not so sensitive to the $\omega$ parameter.

Let the domain be divided into $L$ blocks. Suppose that we apply a multi-coloring technique, such as a greedy algorithm described in [10], to these blocks so that a block of one color has no coupling with a block of the same color. Let $D_j$ be the coupling within the block $j$, and color(j) be the color of the j-th block. We denote by $U_{j,k}, k = 1, q, j < k$ and $L_{j,k}, k < j$ the couplings between the $j$-th color block and the k-th block.

Then, we can describe the Multi-Color Block SSOR as follows.

ALGORITHM **3.1. Multi-Color Block SSOR**

*Let q be the total number of colors, and color(i), i=1, L, be the array of the color for each block.*

1. *Choose* $u_0$, *and* $\omega > 0$.
2. *For* $i > 0$ *Until Convergence Do*
3. *For kolor = 1, q Do*
4. *For j = 1, L Do*
5. *if(color(j) == kolor) then*
6. $(u_{i+1/2})_j = D_j{}^{-1}(b - \omega * \sum_{k \neq kolor}^{k=q} L_{j,k} u_{i+1/2})$.
7. *endif*
8. *Endfor*
9. *For kolor = 1, q Do*
10. *For j = 1, L Do*
11. *if(color(j) == kolor) then*
12. $(u_{i+1})_j = D_j{}^{-1}(u_{i+1/2} - \omega * \sum_{k \neq kolor}^{k=q} U_{j,k} u_{i+1})$.
13. *endif*
14. *Endfor*
15. *Endfor*
16. *Endfor*

Note that the innermost loop in line six and seven can be executed in parallel.

## 4. Test problems

- **Problem 1** Poisson Equation on a Square

$$\begin{aligned} -\triangle u &= f, \;\; \Omega = (0,1) \times (0,1) \\ u &= 0 \;\text{ on }\; \delta\Omega \\ f &= x(1-x) + y(1-y) \end{aligned}$$

- **Problem 2**[3] Elman's problem

$$\begin{aligned} -(bu_x)_x - (cu_y)_y + (du)_x + du_x + (eu)_y + eu_y + fu &= g \\ \Omega = (0,1) \times (0,1) & \\ u = 0 \;\; on \;\; \delta\Omega & \end{aligned} \tag{4}$$

where $b = \exp(-xy),\ c = \exp(xy),\ d = \beta(x+y)$,
$e = \gamma(x+y),\ f = \frac{1}{(1+xy)}$,
and $g$ is such that exact solution $u = x\exp(xy)\sin(\pi x)\sin(\pi y)$

## 5. Results

Tables 1 - 3 contain the timings for the cases without preconditioning and with Multi-Color Block SSOR preconditioning. All of our test problems assume $B{=}I$. We used MPI(Message Passing Machine) library for the interprocessor communications. For the first two problems we used the Block-Row mapping for the graph partitioning of the matrix. For the third problem we have used the Metis code developed by V. Kumar of the University of Minnesota. The number of colors needed is two for the first two problems and reaches 6 for the three dimensional problem of problem three. For the multi-coloring we have used the greedy heuristic as described in [10].

For the first problem the matrix is derogatory but not defective, i.e, the matrix has $N$ linearly independent eigenvectors, but the matrix has two eigenvectors for some eigenvalues. With the orthogonal deflation strategy the convergence is very slow, especially with the second eigenvector for the eigenvalue with two eigenvectors. For this problem the timing gets worse with the preconditioning. For the second problem we get a normal acceleration with the preconditioning. The third problem is very ill-conditioned, coming from Cylinder Shell problem of Harwell/Boeing collection. The condition numbers of 's1rmq4m1' is $1.8^6$, while that of 's3dkq4m2' is $1.9^{11}$. It is reported in [2] that for the third problem Incomplete Cholesky Factorization preconditioning does not achieve the convergence. But our preconditioner does achieve the convergence. As for the $\omega$ parameter we set $\omega$ to be 1.

For the inversion of diagonal blocks in Block SSOR method, we have used the MA48 routine of the Harwell library, which adopts direct methods for sparse matrices with the reordering strategy reducing fill-ins. The cost of the MA48 is roughly proportional to $L^2$, where $L$ is the size of the matrix. Since $L$ is roughly $N/p$, we expect a quadratic decrease with the increasing number of processors.

TABLE 1. Problem 1, with FDM

| | p = 4 | p = 8 | p = 16 | p = 32 | p = 64 |
|---|---|---|---|---|---|
| | No Preconditioning/MC-BSSOR Preconditioning | | | | |
| N=$128^2$ | 14.4/21.2 | 78.2/8.29 | 13.1/13.8 | 17.8/18.3 | 33.3/31.0 |
| N=$256^2$ | 146/446 | 107/64.8 | 47.0/53.4 | 357/607 | 113/134 |
| N=$512^2$ | | | 5500/ 841 | 172/ 1049 | 138/8164 |

TABLE 2. Problem 2 with FDM

| | p = 4 | p = 8 | p = 16 | p = 32 | p = 64 |
|---|---|---|---|---|---|
| | Cpu time/Iterations | | | | |
| N=$128^2$ | 29.4/19.0 | 21.2/9.7 | 21.6/10.0 | 22.8/5.13 | 33.4/6.63 |
| N=$256^2$ | 191/139 | 107/61.7 | 75.8/56.6 | 64.7/19.6 | 77.4/15.1 |
| N=$512^2$ | | | 376/225 | 242/117 | 211/66.9 |

TABLE 3. Cylinder Shell problem from the Harwell/Boeing Collection

| | p = 4 | p = 8 | p = 16 | p = 32 | p = 64 |
|---|---|---|---|---|---|
| | Cpu time/Iterations | | | | |
| s1rmq4m1 | SL/102 | SL/52.1 | SL/SL | SL/SL | SL/SL |
| s3dkq4m2 | | | | SL/6043 | SL/2977 |

## 6. CONCLUSIONS

- Except for the first derogatory matrix, our preconditioner shows a normal behaviour even for the ill-conditioned matrices of the third problem.
- For the first problem our algorithm needs an improvement.
- Due to the nature of MA48 library, we expect our preconditioner to be scalable with the increasing number of processors.

## REFERENCES

[1] W. W. BRADBURY AND R. FLETCHER, *New iterative methods for the solution of the eigenproblem*, Numer. Math., 9(1966), pp. 259–267.

[2] Y. CHO AND Y. K. YONG, *A multi-mesh, preconditioned conjugate gradient solver for eigenvalue problems in finite element models*, Comput. Struct., 58(1996), pp. 575–583.

[3] H. Elman, *Iterative methods for large, sparse, nonsymmetric systems of linear equations*, Ph. D Thesis, Yale University, 1982

[4] Y. T. Feng and D. R. J. Owen, *Conjugate gradient methods for solving the smallest eigenpair of large symmetric eigenvalue problems*, Internat. J. Numer. Methods Engrg., 39(1996), pp. 2209–2229.

[5] G. Gambolati, G. Pini, and M. Putti, *Nested iterations for symmetric eigenproblems*, SIAM J. Sci. Comput., 16(1995), pp. 173–191.

[6] G. Gambolati, F. Sartoretto, and P. Florian, *An orthogonal accelerated deflation technique for large symmetric eigenproblems*, Comput. Methods Appl. Mech. Engrg., 94(1992), pp. 13–23.

[7] D.E. Longsine and S.F. McCormick, *Simultaneous Rayleigh quotient minimization methods for* $Ax = \lambda Bx$ , Linear Algebra Appl., 34(1980), pp. 195–234.

[8] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[9] A. Ruhe, *Computation of eigenvalues and eigenvectors*, in Sparse Matrix Techniques, V. A. Baker, ed., Springer-Verlag, Berlin(1977), pp. 130–184.

[10] Y. Saad, *Highly parallel preconditioner for general sparse matrices*, in Recent Advances in Iterative Methods, IMA Volumes in Mathematics and its Applications, Vol. 60, G. Golub, M. Luskin and A. Greenbaum, eds, Springer-Verlag, Berlin, 1994, pp. 165-199.

[11] F. Sartoretto, G. Pini and G. Gambolati, *Accelerated simultaneous iterations for large finite element eigenproblems*, J. Comput. Phys., 81(1989), pp. 53–69.

[12] H. R. Schwarz, *Eigenvalue problems and preconditioning*, ISNM, 96(1991), pp. 191–208.

School of Electrical and Computer Engineering
Hanyang University, Ansan, Korea Department of Mathematics

Hanyang University, Seoul, Korea