数理解析研究所講究録
第832巻 1993年 64-74

Eigenvalue analysis and Parallelization of the Multi-grid Preconditioned Conjugate Gradient Method

Osamu Tatebe* (建部 修見)
Department of Information Science, Faculty of Science, the University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113, JAPAN

1 Introduction

The multi-grid preconditioned conjugate gradient method(MGCG method)[12] is the PCG method with the multi-grid method[11] as a preconditioner. The MGCG method has the following properties.

1. Even in the case that the problem is stiff and ill-conditioned, the MGCG method converges fast.
2. The number of iterations of the CG method is very few.
3. The number of iterations does not depend upon a grid size.

In order to explain this phenomenon, this paper considers eigenvalue analysis, which is closely related to the number of CG iterations. Since the MGCG method has these characters, it is an ideal method which solves the Poisson's equations, and no other method has these characters. Next parallelization of the MGCG method is considered.

In the section 2, the model of problems is described. In the sections 3 and 4, the preconditioned conjugate gradient method and the multi-grid method which are the basis of the MGCG method are explained and the section 5 presents the MGCG method. In the section 6, eigenvalue analysis is shown that the multi-grid preconditioner suits the conjugate gradient method. In the section 7, parallelization of the MGCG method is considered. Under this consideration, the section 8 presents implementation of the MGCG method on the Fujitsu multicomputer AP1000 and its evaluation.

2 The model of problems

The target problem is two-dimensional elliptic partial differential equation and the numerical experiment is performed for the following Poisson's equation with Dirichlet boundary conditions

$$-\nabla(k\nabla u) = f \quad \text{in} \quad \Omega \subset \Re^2$$

$$\text{with} \quad u = f_\Gamma \quad \text{on} \quad \Gamma$$

*E-mail: tatebe@is.s.u-tokyo.ac.jp

where k is a real function, Ω is a domain in the two-dimensional space, Γ is its boundary. This equation is discretized by the finite element method and the system of linear equations

$$A\boldsymbol{x} = \boldsymbol{f}$$

is obtained. This coefficient matrix A is sparse, symmetric and positive definite.

3 The preconditioned conjugate gradient method

If a real $n \times n$ matrix A is symmetric and positive definite, the solution of the linear system $A\boldsymbol{x} = \boldsymbol{f}$ is equivalent to the minimization of the quadratic function

$$Q(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - \boldsymbol{f}^T\boldsymbol{x}. \tag{1}$$

The conjugate gradient method is one of the minimization method and use A-conjugate vectors as the direction vectors which are generated sequentially. Theoretically this method has the striking property that the number of steps until convergence is at most rank(A) steps. And this method can be adapted successfully to the parallel and vector computation, since one CG iteration requires only one product of the matrix and the vector, two inner products, tree linked triads, two scalar divides and one scalar compare operation.

Next the preconditioned conjugate gradient method is explained. Let U be a nonsingular matrix, and define $\tilde{A} = UAU^T$, then solve $\tilde{A}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{f}}$ by the use of plain conjugate gradient method. Let $\boldsymbol{x}^0$ be an initial approximate vector, then an initial residual $\boldsymbol{r}^0$ is $\boldsymbol{r}^0 = \boldsymbol{f} - A\boldsymbol{x}^0$. Let $M = U^TU$, $\tilde{\boldsymbol{r}}^0 = M\boldsymbol{r}^0$ and an initial direction vector $\boldsymbol{p}^0 = \tilde{\boldsymbol{r}}^0$. The PCG algorithm is described by Program 1.

```
i = 0;
while ( !convergence ) {
  α_i = (r̃_i, r_i)/(p_i, Ap_i);
  x_{i+1} = x_i + α_i p_i;
  r_{i+1} = r_i − α_i Ap_i;
  convergence test;
  r̃_{i+1} = M r_{i+1};
  β_i = (r̃_{i+1}, r_{i+1})/(r̃_i, r_i);
  p_{i+1} = r̃_{i+1} + β_i p_i;
  i++;
}
```

Program 1: the PCG iteration

In the loop block of Program 1, an matrix M at $\tilde{\boldsymbol{r}}_{i+1} = M\boldsymbol{r}_{i+1}$ is not actually calculated, and in ordinary circumstances, such as incomplete Cholesky decomposition CG method, only M^{-1} is got. Hence the following equation:

$$M^{-1}\tilde{\boldsymbol{r}}_{i+1} = \boldsymbol{r}_{i+1} \tag{2}$$

must be solved.

This paper focuses on a solver of this equation (2) and new proposal is the PCG method exploiting the multi-grid method.

4 The multi-grid method

In case of the iterative method, the residual reduces most rapidly on the grid corresponding to the frequency components of it. The multi-grid method make good use of this characteristic and exploit a lot of grids to aim at the convergence as rapid as possible in point of the iterative method.

These grids are leveled and numbered from the coarsest grid. The number is called *level number*. The basic element of the multi-grid method is *defect correction principle*, which the residual is reduced moving it from grid to grid. Defect correction scheme consists of three processes:

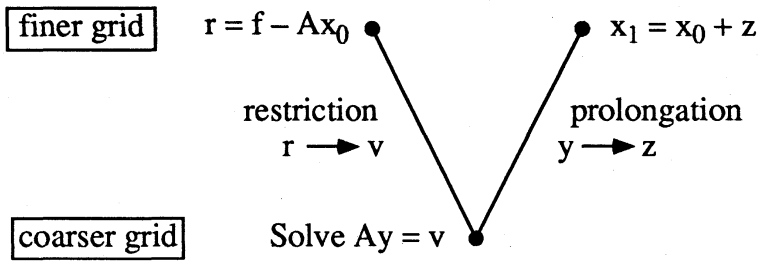

Figure 1: coarse grid correction

pre-smoothing process, *coarse grid correction* and *post-smoothing* process, and it uses only two grids. In the pre-smoothing process the residual reduces by relaxant calculation on a finer grid. In coarse grid correction(depicted in Fig. 1), the residual $\boldsymbol{r}$, which is not completely solved in the pre-smoothing process, is restricted to fit on a coarser grid, vector $\boldsymbol{v}$, and the equation $A\boldsymbol{y} = \boldsymbol{v}$ is solved on the coarser grid by the use of any method and the answer vector $\boldsymbol{y}$ is prolonged on the original finer grid, vector $\boldsymbol{z}$, then the vector $\boldsymbol{x}_0$ of the pre-smoothing process is corrected as $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{z}$. In post-smoothing process the residual also reduces by relaxant calculation on the finer grid.

It is the two-grid method that this scheme is used only once per iteration and in the coarse grid correction vector $\boldsymbol{y}$ is solved correctly enough. Besides, in the coarse grid correction if vector $\boldsymbol{y}$ is solved by the use of recursive call of the defect correction scheme, this method becomes the multi-grid method.

Many variations of how to call the defect correction procedure recursively is considerable. Fig. 2 shows V-cycle scheme and W-cycle scheme. Ω_i means the grid of level number i. V-cycle scheme is only one recursive call used in coarse grid correction and W-cycle scheme is two.

In the smoothing process, various methods, such as ILU, ADI and zebra relaxation, are proposed. One purpose of this research is, however, formation of the efficient method with high parallelism. Therefore iterative method with high parallelism, such as point Jacobi method and multi-color SSOR method, is used as smoothing method.

A operation of transfer a vector on a coarser grid to a vector on a finer grid calls *prolongation*, and opposite operator called *restriction*. And the matrix presented the operation of prolongation is wrote p in this paper, and restriction r.

In the following section, the equation of grid level i is described as

$$L_i\boldsymbol{x}_i = \boldsymbol{f}_i.$$

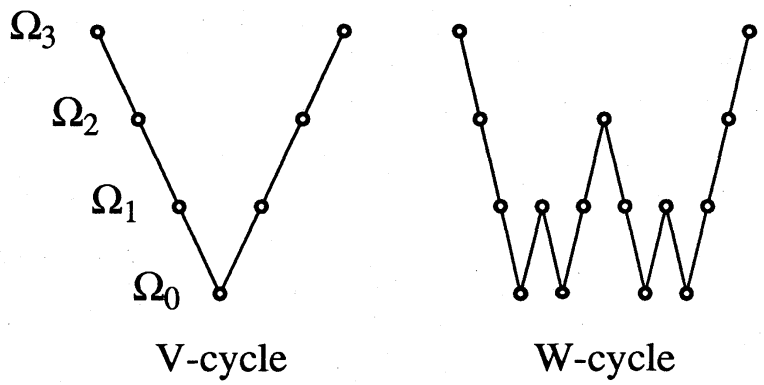

Figure 2: scheme of the multi-grid method

and between restriction and prolongation the following relation:

$$r = bp^T$$

should be satisfied, where b is a scalar constant.

5 The MGCG method

The MGCG method is the PCG method with a multi-grid preconditioner. The MGCG algorithm solving a linear equation $L_l\boldsymbol{x}=\boldsymbol{f}$ is described as the Program 2. Let $\boldsymbol{x}^0$ be an initial approximate vector, then an initial residual $\boldsymbol{r}^0$ is $\boldsymbol{r}^0 = \boldsymbol{f} - L_l\boldsymbol{x}^0$. Then $L_l\tilde{\boldsymbol{r}}^0=\boldsymbol{r}^0$ is relaxed by the multi-grid method, and an initial direction vector $\boldsymbol{p}^0$ is equal to $\tilde{\boldsymbol{r}}^0$.

$i = 0$;
while (!convergence) {
  $\alpha_i = (\tilde{\boldsymbol{r}}_i, \boldsymbol{r}_i)/(\boldsymbol{p}_i, L_l\boldsymbol{p}_i)$;
  $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i\boldsymbol{p}_i$;
  $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i L_l\boldsymbol{p}_i$;
  convergence test;
  Relax $L_l\tilde{\boldsymbol{r}}_{i+1} = \boldsymbol{r}_{i+1}$ using the Multi-grid method
  $\beta_i = (\tilde{\boldsymbol{r}}_{i+1}, \boldsymbol{r}_{i+1})/(\tilde{\boldsymbol{r}}_i, \boldsymbol{r}_i)$;
  $\boldsymbol{p}_{i+1} = \tilde{\boldsymbol{r}}_{i+1} + \beta_i\boldsymbol{p}_i$;
  i++;
}

Program 2: the MGCG method

There are several variations of the multi-grid preconditioner. What method is used as the multi-grid method? What method is used as a relaxant method? What cycle is used in the multi-grid method? If m is a cycle of the multi-grid method, l is a relaxant method, n is the number

of iterations of the relaxant method and g is the number of grids, MGCG method is shown MGCG(l, m, n, g). But g is an optional parameter. For example, MGCG$(RB, 1, 2)$ is the MGCG method of the V-cycle multi-grid preconditioner with two iterations of Red-Black SSOR method as a relaxant calculation.

6 Eigenvalue Analysis

In order to realize an efficiency of a multi-grid preconditioner, the eigenvalues' distribution of a coefficient matrix after preconditioning is examined. The number of iterations of the conjugate gradient method until convergence depends upon an initial vector, the distribution of the eigenvalues of a coefficient matrix and right-hand term. But in these factors the eigenvalues' distribution is particularly important.

Two-dimensional Poisson equation with Dirichlet boundary condition:

$$-\nabla (k \nabla u) = f \quad \text{in} \quad \Omega = [0,1] \times [0,1]$$

$$\text{with} \quad u = g \quad \text{on} \quad \partial\Omega$$

is considered. k is a diffusion constant, f is a source term and g is a boundary condition. The experiment is performed in the following condition.

The diffusion constants k is depicted by Fig. 3. This problem has the non-uniform diffusion

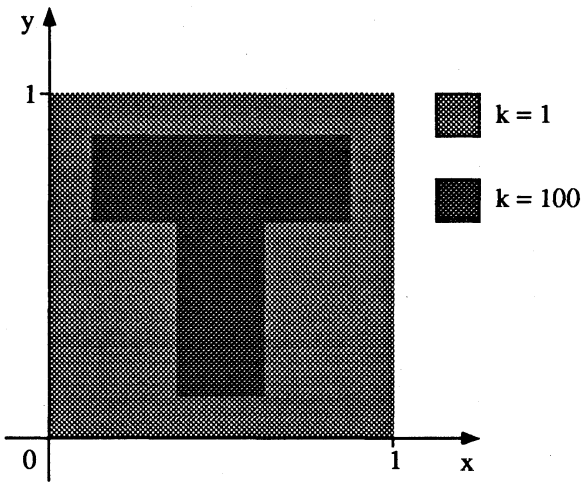

Figure 3: diffusion constant of the problem

constants and the shape of the large diffusion constants is 'T', therefore it has rich distribution of the eigenvalues of the problem matrix. This problem is discretized to the mesh of 16×16 by the control volume method. The condition number of this coefficient matrix is 5282.6. The eigenvalues' distribution of the problem matrix and the eigenvalues' distribution of the matrix after preconditioning is shown at Fig. 4. In order to compare, the preconditioning is occurred both the multi-grid method and the incomplete Cholesky decomposition.

The matrix after the multi-grid preconditioning is calculated as follows. Let the matrix M represent operations of the multi-grid method, and this matrix M is Cholesky decomposed

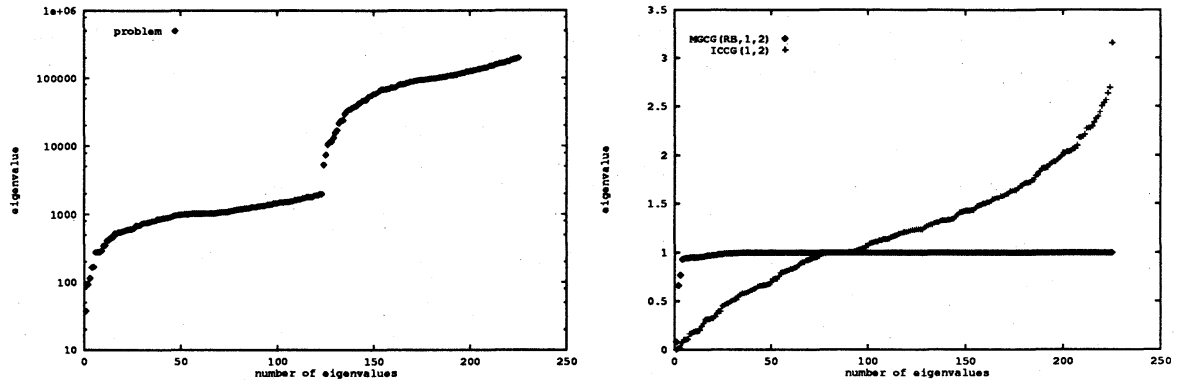

Figure 4: eigenvalues' distribution

as $M=U^TU$. Then eigenvalues of the matrix UL_lU^T is investigated. On the other hand the matrix using the ICCG method is calculated as follows. The matrix L_l is incomplete Cholesky decomposed as $L_l=S^TS-T$, and the general eigenvalue problem $L_l\boldsymbol{x}=\lambda S^TS\boldsymbol{x}$ is solved.

The eigenvalues' distribution of the multi-grid preconditioner is effective for the conjugate gradient method as the following points:

1. eigenvalues are crowded almost around 1.
2. the least eigenvalue is increased.
3. the condition number is decreased.

These characteristics are all desirable to the conjugate gradient method.

7 Parallelization

In this section, the parallelization of the MGCG method on a multicomputer is discussed. A Multicomputer is a computer that has multiple processors and lacks a shared memory. All communication and synchronization between processors must take place through message passing.

7.1 Parallelization of the MGCG method

The MGCG method is the PCG method with a multi-grid preconditioner, so both the CG method and the multi-grid method have to be parallelized. The CG method is easily parallelized but the multi-grid method has the following problems.

1. The smoothing method must have high parallelism.
2. When the grid size is coarser, the overhead of communication is obvious.
3. When the grid size is coarser, the utilization of processors decreases. Idle processors appear and a load imbalance is occurred.

However the multi-grid method is very effective, if the following restrictions is satisfied.

1. A strong smoothing method that have poor parallelism is used in order to solve complex and ill-conditioned problems.

2. The multi-grid method solves exactly on the finest grid in order to get fast convergence.

When the multi-grid method is parallelized, the former three problems are not settled and the parallelization of the multi-grid method is severe. Therefore it has been invented the parallel multi-grid method[3] not being equal to the stock serial multi-grid algorithm.

The MGCG method, however, uses the multi-grid method as only a preconditioner of the PCG method, so it is dispensable that the multi-grid method must use a strong smoothing method and must exactly solve on the finest grid. In other words, these two restrictions can be loosened within the scope that a good performance of the target machine is gain. Of course, the looser these restrictions are, the worse a speed of the convergence is. Therefore there is a trade-off between the performance of the target machine, both communication and computation, and a speed of the convergence.

A smoothing method of the MGCG method should be a fine iterative method with high parallelism, therefore the Red-Black symmetric SOR method(RB-SSOR method) is selected. Then there is a trade-off between the number of grids used, a scheme of the multi-grid method and the number of iterations of the RB-SSOR method, and the performance of the target machine.

7.2 Eigenvalue analysis

This section considers a multi-grid preconditioner that utilizes fewer grids and does not exactly solve on the finest grid. The problem is the same problem of the section 6, and the multi-grid preconditioner uses only two-grid and on the coarsest grid the equation is not exactly solved but relaxed by the smoothing method. The eigenvalues' distribution of the matrix after preconditioning is shown at Fig. 5. In order to compare usual method, the preconditioning is occurred both the multi-grid method and the incomplete Cholesky decomposition. The eigenvalues' distribu-

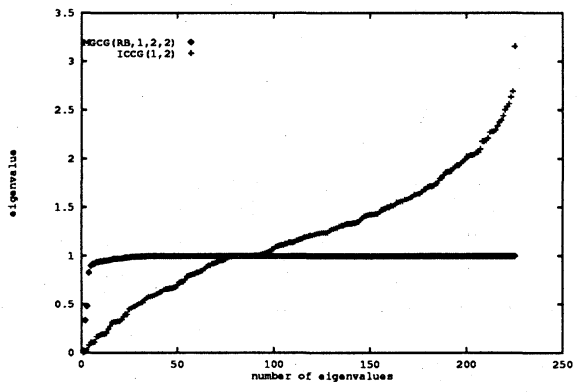

Figure 5: eigenvalues' distribution after preconditioning

tion of the multi-grid preconditioning is also effective for the conjugate gradient method as the

same reason of the section 6 and in comparison with the incomplete Cholesky decomposition, the multi-grid method is very effective preconditioner. Therefore as long as the multi-grid method is used for a preconditioner, it is realized that the two restrictions of the section 7.1 do not have to be required for the MGCG method.

7.3 Domain Decomposition

This section presents the domain decomposition for the implementation of the MGCG method on multicomputer. A product of the matrix and the vector, a restriction and a prolongation of the multi-grid method, and a smoothing method require nearest-neighbor communications. Therefore it is desired that data in one processor are allocated continuously if possible. If M processors are connected by a torus network, the following two methods are considerable.

(A) The domain of the differential equation is splitted into subdomains $M \times 1$ which are assigned to the processors of the system.

(B) The domain of the differential equation is splitted into subdomains $\sqrt{M} \times \sqrt{M}$ which are assigned to the processors of the system.

Let problem size be N. The computation complexity and the communication complexity of these two methods are discussed. The computation complexity is proportional to the number of inner grids, so the computation complexity of both methods is $O(N/M)$. The communication complexity is proportional to the number of grids of subdomain's edge, so the communication complexity of the method (A) is $O(\sqrt{N})$ and the method (B) $O(\sqrt{N}/\sqrt{M})$. However the number of communications of the method (B) is larger than it of the method (A). Therefore the larger problem size and processors are, the more profitable the method (B) is.

8 Implementation on the AP1000

Two methods (A), (B) of the previous section is implemented on the Fujitsu multicomputer AP1000. The numerical experiments is performed by MGCG(RB, 1, 2, 2) which exploits only two grids. The problem is two-dimensional Poisson equation with Dirichlet boundary condition:

$$-\nabla (k \nabla u) = f \quad \text{in} \quad \Omega = [0,1] \times [0,1]$$

$$\text{with} \quad u = g \quad \text{on} \quad \partial\Omega,$$

where k is a constant. The problem is discretized to the three kind of meshes: 64×64, 128×128 and 256×256, by the control volume method. The result is tables 1, 2 and a speedup of these methods is shown at Fig. 6. These tables and figures indicate the method (B) is more advantageous than the method (A) when a problem size is larger and the number of processors is larger. Both graphs are superlinear. This is because the problem size is fixed and the number of processors increases. Therefore the number of data in one processor decreases, and the cache hit ratio and the hit ratio of line-send increase. Line-send is a special function of AP1000 and data is directly transferred from a cache memory if this data exists in the cache. Cache hit ratio is shown at Fig. 7

size	# of iter.	1×1 procs	2×1	4×1	8×1	16×1	32×1	64×1
63^2	19	6.72	2.93	1.65	0.85	0.55	-	-
127^2	38	62.48	29.46	14.41	6.73	3.94	2.21	-
255^2	71	476.99	236.95	114.84	58.20	29.24	15.43	9.40

(sec)

Table 1: the method (A) {MGCG(RB, 1, 2, 2)}

size	# of iter.	1×1 procs	2×2	4×4	8×8
63^2	19	7.09	1.66	0.50	0.23
127^2	38	60.67	14.26	3.46	1.01
255^2	71	455.4	113.8	27.3	6.49

(sec)

Table 2: the method (B) {MGCG(RB, 1, 2, 2)}

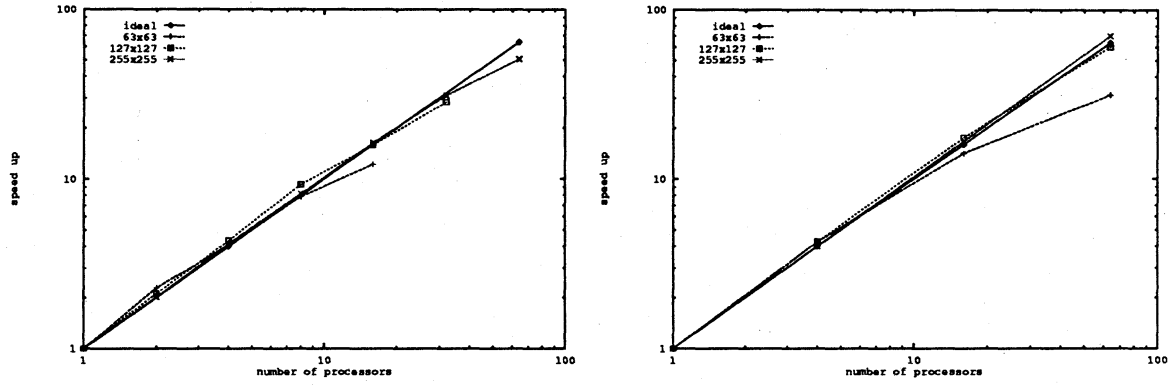

Figure 6: speedup of the methods (A) and (B)

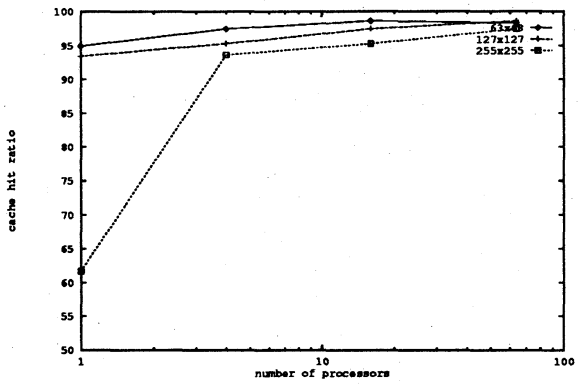

Figure 7: cache hit ratio

9 Conclusion

This paper firstly considers eigenvalue analysis in order to verify the effect of the MGCG method which is a promising method as the solution of a large-scaled sparse, symmetric and positive definite matrix. The eigenvalues' distribution of the multi-grid preconditioner is effective for the conjugate gradient method as the following points:

1. eigenvalues are crowded almost around 1.
2. the least eigenvalue is increased.
3. the condition number is decreased.

These characteristics are all desirable to the conjugate gradient method. Secondly parallelization of the MGCG method on multicomputer lacking a shared memory is discussed. The multi-grid method has some problems when it is effectively parallelized, but the multi-grid preconditioner of the CG method is sufficiently efficient even if two restrictions of section 7.1 is not satisfied and the performance of multicomputer is exploited.

References

[1] Adams, L. M., *Iterative Algorithms for Large Sparse Linear System on Parallel Computers.* PhD thesis, University of Virginia, November 1982.

[2] Chan, T. F. and R. S. Tuminaro, "Analysis of a Parallel Multigrid Algorithm," in *Proceedings of the fourth Copper Mountain Conference on Multigrid Method* (J. M. et al, ed.), pp. 66–86, April 1989.

[3] Frederickson, P. O. and O. A. McBryan, "Recent Developments for the PSMG Multiscale Method," in *Multigrid Methods III* (W. Hackbusch and U. Trottenberg, eds.), vol. 98 of *ISNM*, 1991.

[4] Hackbusch, W., "Convergence of Multi-Grid Iterations Applied to Difference Equations," *Mathematics of Computation*, vol. 34, no. 150, pp. 425–440, 1980.

[5] Hackbusch, W., "Mult-grid Convergence Theory," in *Multigrid Methods* (W. Hackbusch and U. Trottenberg, eds.), vol. 960 of *Lecture Notes in Mathematics*, pp. 177–219, Springer-Verlag, 1982.

[6] Hackbusch, W., *"Multi-Grid Methods and Applications"*. Springer-Verlag, 1985.

[7] Maitre, J. F. and F. Musy, "Multigrid Methods: Convergence Theory in a Variational Framework," *SIAM J. Numer. Anal.*, vol. 21, pp. 657–671, 1984.

[8] Moji, J. J., *"Parallel Algorithms and Matrix Computation"*. Oxford University Press, 1988.

[9] Ortega, J. M., *"Introduction to Parallel and Vector Solution of Linear Systems"*. Plenum Press, 1988.

[10] Sbosny, H., "Domain Decomposition Method and Parallel Multigrid Algorithms," in *Multigrid Methods: Special Topics and Applications II* (W. Hackbusch and U. Trottenberg, eds.), no. 189 in GMD-Studien, pp. 297–308, May 1991.

[11] Stüben, K. and U. Trottenberg, "Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications," in *Multigrid Methods* (W. Hackbusch and U. Trottenberg, eds.), vol. 960 of *Lecture Notes in Mathematics*, pp. 1–176, Springer-Verlag, 1982.

[12] Tatebe, O. and Y. Oyanagi, "The Multi-grid Preconditioned Conjugate Gradient Method," in *IPSJ SIM Notes, 92-NA-42*, pp. 9–16, August 1992. SWoPP'92.

[13] Wesseling, P., "Theoretical and practical aspects of a muligrid method," *SIAM Journal on Scientific and Statistical Computing*, vol. 3, pp. 387–407, 1982.