Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **4**, No. 2, pp. 238-249

DOI: 10.4208/aamm.10-m11133
April 2012

A Two-Level Preconditioned Conjugate-Gradient Method in Distorted and Structured Grids

Qiaolin He*

Deparment of Mathematics, Sichuan University, Chengdu 610064, Sichuan, China

Received 23 August 2011; Accepted (in revised version) 26 December 2011

Available online 26 March 2012

Abstract. In this paper, we propose a new two-level preconditioned C-G method which uses the quadratic smoothing and the linear correction in distorted but topologically structured grid. The CPU time of this method is less than that of the multigrid preconditioned C-G method (MGCG) using the quadratic element, but their accuracy is almost the same. Numerical experiments and eigenvalue analysis are given and the results show that the proposed two-level preconditioned method is efficient.

AMS subject classifications: 65F08, 65N22, 65N30, 65N50

Key words: Precondition, conjugate gradient, multigrid, finite element.

1 Introduction

The multigrid scheme has been widely used to solve the partial differential equations. In the sixties Fedorenko [1, 2] developed the first multigrid scheme for approximating the solution of the Poisson equation in a unit square. Since then, other mathematicians extended his idea to general elliptic boundary value problems with variable coefficients; see, e.g., [3]. However, the full efficiency of the multigrid approach was realized after the works of Brandt [4, 5] and Hackbusch [6]. These authors also introduced multigrid methods for nonlinear problems such as the multigrid full approximation storage (FAS) scheme [5, 7]. Another achievement in the formulation of multigrid methods was the full multigrid (FMG) scheme [5, 7], based on the combination of nested iteration techniques and multigrid methods. Multigrid algorithms are now applied to a wide range of problems, primarily to solve linear and nonlinear boundary value problems. A multigrid preconditioned conjugate gradient (MGCG) method has been put forward by Tatebe in [11], which used the multigrid method as a preconditioner for CG method and has a good convergence rate even for the problems on which the standard multigrid method does not converge efficiently. On the

*Corresponding author.
Email: qlhejenny@gmail.com (Q. L. He)

other hand, Bank and Douglas [16] treated the conjugate gradient method as a relaxation method of the multigrid method. Braess [12] considered these two combinations and reported the conjugate gradient method with a multigrid preconditioning is effective for elasticity problems. Then Tatebe and Oyanagi considered a parallelization of the MGCG method and proposed an efficient parallel MGCG method on distributed memory machines [15]. A class of usefull of solvers based on the multigrid strategy are algebraic multigrid (AMG) methods [8] that resemble the geometric multigrid process utilizing only information contained in the algebraic system to be solved. It is noted that S. Shu et al proposed an algebraic multigrid method for higher order finite element discretizations [13], who also studied AMG method for finite element systems on criss-cross grids [14].

In this paper, we study an efficient multigrid method which can be used in distorted but topologically structured grid. We utilize iterative grid redistribution method, proposed by Ren and Wang in [10], to generate mesh which concentrates in the region where solution has large variation. A quadratic finite element method and a linear finite element method are both employed to discretize the equation and MGCG method is used to solve the discretized system $Au = b$. The result of using quadratic element is more accurate than that of using linear element, while the former costs more CPU time than the latter. We would like to obtain the accuracy of using quadratic element and cost less CPU time. We improve MGCG method and make it more efficient. Our two-level method in the preconditioning step has several crucial parts: (1) take pre-smoothing steps by Gauss-Seidel iteration in the quadratic finite element space; (2) calculate the residue and restrict it on the linear finite element space; (3) use the V-cycle multigrid scheme to solve $Au = r$ (r is the residue); (4) prolongate the solution from the linear element space to the quadratic element space; (5) take post-smoothing steps by Gauss-Seidel iteration in the quadratic element space. The above is different from current MGCG method and will be shown very useful.

In the following, the multigrid preconditioned conjugate gradient method and our new two-level preconditioned C-G method are described in Sections 2 and 3. The efficiency of the two-level method are verified by numerical examples given in Section 4. In Section 5, eigenvalue analysis is presented, which explains why our two-level method is efficient. When the two-level method is used as a preconditioner of the conjugate gradient method, it becomes quite an effective and desirable preconditioner of the conjugate gradient method.

2 Multigrid preconditioned C-G method

The MGCG method is a PCG method that uses the multigrid method as a preconditioner. When a target linear equation is

$$L_l \mathbf{x} = \mathbf{f},$$

iteration of the MGCG method is described in Program 1. Let an initially approximate vector be $\mathbf{x}^0$ and an initial residual

$$\mathbf{r}^0 = \mathbf{f} - L_l\mathbf{x}^0.$$

Equation

$$L_l\widetilde{\mathbf{h}}^0 = \mathbf{r}^0,$$

is approximately solved by the multigrid method and we set an initial direction vector

$$\mathbf{p}^0 = \widetilde{\mathbf{h}}^0.$$

The loop of Program 1 is iterated until convergence.

In Program 1, (2.4) is the part of a multigrid preconditioning. Program 2 is the procedure of multigrid method implementation . The function shown in Program 2 is recursively defined, where a sequence of coefficient matrices is $\{L_i\}(0 \leq i \leq l)$, μ_1 and μ_2 are the numbers of pre- and post-smoothing iterations respectively. Multigrid cycle depends on γ, μ_1 and μ_2 in Program 2. When $\gamma = 1$ and $\mu_1 = \mu_2 \neq 0$, it is called V-cycle multigrid method. When $\gamma = 2$ and $\mu_1 = \mu_2 \neq 0$, it is called W-cycle multigrid method. When $\gamma = 1$ and $\mu_1 = 0$, $\mu_2 \neq 0$, it is called the sawtooth cycle. These cycles are depicted in Fig. 1. Popular multigrid cycles are V-cycle of $\gamma = 1$ and $\mu_1 = \mu_2 = 1$ and W-cycle of $\gamma = 2$ and $\mu_1 = \mu_2 = 2$. We use V-cycle in this paper.

Program 1. Iteration of the MGCG method.

$i = 0$;
while (! convergence)
{

$$\alpha_i = (\widetilde{\mathbf{h}}^i, \mathbf{r}^i)/(\mathbf{p}^i, L_l\mathbf{p}^i); \tag{2.1}$$

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha_i\mathbf{p}^i; \tag{2.2}$$

$$\mathbf{r}^{i+1} = \mathbf{r}^i - \alpha_i L_l\mathbf{p}^i; \tag{2.3}$$

convergence test;
Relax

$$L_l\widetilde{\mathbf{h}}^{i+1} = \mathbf{r}^{i+1}, \tag{2.4}$$

using the Multigrid method

$$\beta_i = (\widetilde{\mathbf{h}}^{i+1}, \mathbf{r}^{i+1})/(\widetilde{\mathbf{h}}^i, \mathbf{r}^i);$$
$$\mathbf{p}^{i+1} = \widetilde{\mathbf{h}}^{i+1} + \beta_i\mathbf{p}^i;$$
$$i++;$$

}

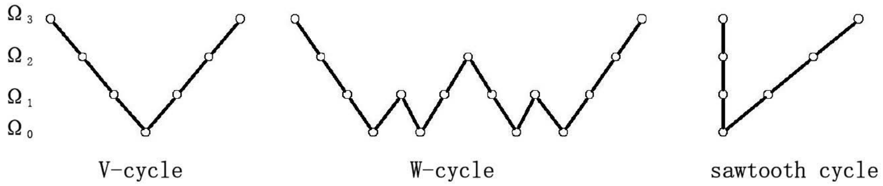

Figure 1: Multigrid cycle.

A transfer operation of vectors on a grid level i to vectors on a grid level $i-1$ is called restriction, and an inverse operator is called prolongation. The matrices that represent the operations of restriction and prolongation are written as r and p respectively. The author of [11] proved that the multigrid method is a mathematically valid preconditioner of the PCG method if the following relation:

$$r = bp^T, \qquad \mu_1 = \mu_2 \neq 0,$$

where b is a scalar constant, is satisfied and if the pre- and post-smoothings are identical and symmetric methods. There are many smoothing methods. In this paper, we use symmetric Gauss-Seidel iteration method to do smoothing.

Program 2. The multigrid method.

Vector MG $(L_l, \mathbf{f}, \mathbf{x}, \gamma, \mu_1, \mu_2)$;
$\{$
if $(l ==$ coarsest-level$)$ Solve $L_l\mathbf{x} = \mathbf{f}$;
else $\{$

$\mathbf{x} =$ pre-smoothing$(L_l, \mathbf{f}, \mathbf{x}, \mu_1)$;
$\mathbf{d} =$ restrict$(\mathbf{f} - L_l\mathbf{x})$;
$\nu =$ initial $\mathbf{x}$;
repeat(γ) $\nu =$ MG$(L_{l-1}, \mathbf{d}, \nu, \gamma, \mu_1, \mu_2)$
$\mathbf{x} = \mathbf{x} +$ prolongate(ν);
$\mathbf{x} =$ post-smoothing$(L_l, \mathbf{f}, \mathbf{x}, \mu_2)$;
$\}$

return $\mathbf{x}$;
$\}$

3 Two-level preconditioned C-G method

Consider the finite element discretization of Poisson equation, the target linear equation is $Q\mathbf{x} = \mathbf{f}_q$, where Q is the coefficient matrix corresponding to quadratic finite element. Similarly for linear element discretization, the target equation is $L\mathbf{x} = \mathbf{f}_l$, where L is the coefficient matrix corresponding to linear finite element. MGCG method is used to solve such two linear equations. It is well known that numerical solution of quadratic element discretization is more accurate than that of linear element discretization, however using quadratic element costs more CPU time and more C-G iterations than using linear element. In order to obtain the accuracy of quadratic element discretization and cost less CPU time than that of quadratic element discretization, we introduce a two-level method which combines linear element and quadratic element together and makes the most of the advantage of these two discretization. In our paper, we use the triangle element for discretization.

3.1 Quadratic pre-smoothing

Consider the discretized equation $Q\mathbf{x} = \mathbf{f}_q$, suppose we obtain $\mathbf{x}^{i+1}$ and $\mathbf{r}^{i+1}$ by (2.2) and (2.3) in Program 1 after i iterations. Equation $Q\mathbf{h}^{i+1} = \mathbf{r}^{i+1}$ is solved by a few steps of Gauss-Seidel iterations and approximate result $\widetilde{\mathbf{h}}^{i+1}$ is obtained, therefore the high frequency components are suppressed efficiently. A version of the Gauss-Seidel iteration that can be used for preconditioning of the C-G method is following:

For j from 1 to n, **do**

$$x_j^{(i+\frac{1}{2})} = -\frac{1}{Q_{jj}}\Big[\sum_{l<j} Q_{jl} x_l^{(i+\frac{1}{2})} + \sum_{l>j} Q_{jl} x_l^{(i)}\Big] + \frac{1}{Q_{jj}} f_{q_j},$$

end do

For j from n to 1 by -1, **do**

$$x_j^{(i+1)} = -\frac{1}{Q_{jj}}\Big[\sum_{l<j} Q_{jl} x_l^{(i+\frac{1}{2})} + \sum_{l>j} Q_{jl} x_l^{(i+1)}\Big] + \frac{1}{Q_{jj}} f_{q_j}.$$

end do

3.2 Linear correction

We calculate the residual $\mathbf{w}_q = \mathbf{r}^{i+1} - Q\widetilde{\mathbf{h}}_{i+1}$ and restrict $\mathbf{w}_q$ on the linear finite element space by $\mathbf{w}_l = I_q^l \mathbf{w}_q$, where $I_q^l : V_q \to V_l$ is the restriction operator from quadratic element space V_q to linear element space V_l. We use ψ_l^j to denote the nodal basis function of point j corresponding to linear element and ψ_q^j to denote the nodal basis function of point j corresponding to quadratic element. Then

$$\psi_l^j = \psi_q^j + \frac{1}{2}\psi_{q1} + \frac{1}{2}\psi_{q2} + \frac{1}{2}\psi_{q3} + \frac{1}{2}\psi_{q4} + \frac{1}{2}\psi_{q5} + \frac{1}{2}\psi_{q6},$$

where ψ_{q1}, ψ_{q2}, ψ_{q3}, ψ_{q4}, ψ_{q5}, ψ_{q6} are respectively nodal basis of effective points $\{1,2,3,4,5,6\}$ in the Lagrangian quadratic element . Therefore

$$\mathbf{w}_l^j = \mathbf{w}_q^j + \frac{1}{2}\mathbf{w}_{q1} + \frac{1}{2}\mathbf{w}_{q2} + \frac{1}{2}\mathbf{w}_{q3} + \frac{1}{2}\mathbf{w}_{q4} + \frac{1}{2}\mathbf{w}_{q5} + \frac{1}{2}\mathbf{w}_{q6}.$$

The restriction operator I_q^l can be written in stencil notation as

$$\begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

Equation $L\mathbf{v}_l = \mathbf{w}_l$ is solved by V-cycle multigrid method and we get the approximate solution $\widetilde{\mathbf{v}}_l$ in the linear element space. Next we prolongate $\widetilde{\mathbf{v}}_l$ from linear element space to quadratic element space by $\widetilde{\mathbf{v}}_q = I_l^q \widetilde{\mathbf{v}}_l$, where $I_l^q : V_l \to V_q$ is prolongation operator. I_l^q can be obtained by $I_l^q = (I_q^l)^T$. Therefore the approximation $\widetilde{\mathbf{h}}^{i+1}$ can be updated:

$$\widetilde{\mathbf{h}}^{i+1} = \widetilde{\mathbf{h}}^{i+1} + \widetilde{\mathbf{v}}_q.$$

3.3 Quadratic post-smoothing

In the same way as in Section 3.1, we start from $\widetilde{\mathbf{h}}^{i+1}$, which is obtained from Section 3.2, and use a few steps of Gauss-Seidel iterations to update $\widetilde{\mathbf{h}}^{i+1}$. Go back to the MGCG program, we compute β_i, $\mathbf{p}^{i+1}$ and do the iteration until convergence. This two-level method is shown in Program 3.

Program 3. Two-level preconditioned C-G method.

$i = 0$;
while (! convergence){
$\alpha_i = (\widetilde{\mathbf{h}}^i, \mathbf{r}^i)/(\mathbf{p}^i, Q\mathbf{p}^i)$;
$\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha_i \mathbf{p}^i$;
$\mathbf{r}^{i+1} = \mathbf{r}^i - \alpha_i Q\mathbf{p}^i$;
convergence test;
$\widetilde{\mathbf{h}}^{i+1}$ = quadratic pre-smoothing$(Q, \mathbf{r}^{i+1}, \widetilde{\mathbf{h}}^{i+1}, \mu)$, μ is the number of Gauss-Seidel iterations
$\mathbf{w}_l$ = restrict$(\mathbf{r}^{i+1} - Q\widetilde{\mathbf{h}}^{i+1})$, from quadratic element to linear element
$\widetilde{\mathbf{v}}_l$ = multigrid V-cycle$(L, \mathbf{w}_l)$
$\widetilde{\mathbf{h}}^{i+1} = \widetilde{\mathbf{h}}^{i+1}$ + prolongate$(\widetilde{\mathbf{v}}_l)$, from linear element to quadratic element
$\widetilde{\mathbf{h}}^{i+1}$ = quadratic post-smoothing$(Q, \mathbf{r}^{i+1}, \widetilde{\mathbf{h}}^{i+1}, \mu)$
$\beta_i = (\widetilde{\mathbf{h}}^{i+1}, \mathbf{r}^{i+1})/(\widetilde{\mathbf{h}}^i, \mathbf{r}^i)$;
$\mathbf{p}^{i+1} = \widetilde{\mathbf{h}}^{i+1} + \beta_i \mathbf{p}^i$;
$i++$;
}

During the PCG procedure, we use the symmetric Gauss-Seidel method which is a superposition of a forward Gauss-Seidel iteration and a backward Gauss-Seidel iteration. A matrix form of symmetric Gauss-Seidel method corresponding to $Q\mathbf{x} = \mathbf{f}$ is

$$\mathbf{x}^{k+1} = B^{-1}(B-Q)\mathbf{x}^k + B^{-1}\mathbf{f},$$

where Q is symmetric and positive definite and B is also symmetric. Let

$$H = B^{-1}(B-Q), \qquad R = \sum_{i=0}^{m-1} H^i B^{-1}.$$

The precondition matrix M of the two-level preconditioned method can be written as

$$M = H^m R + R + H^m I_l^q \widetilde{M}_l I_q^l (H^T)^m. \tag{3.1}$$

The matrix $\widetilde{M}_l$ of the V-cycle multigrid method has been proved to be symmetric and positive definite (see [11]). It is easy to obtain that $M^T = M$ and M is positive definite by Theorem 2 in [11].

4 Numerical experiments

Consider the two dimensional Poisson equation with Dirichlet boundary condition:

$$\triangle u = f \quad \text{in } \Omega = [-1,1] \times [-1,1], \tag{4.1a}$$

$$u = g \quad \text{on } \partial\Omega. \tag{4.1b}$$

For some f, the solution may has large variation in some region. During our numerical tests, the iterative grid redistribution method proposed in [10] is used to generate adaptive grid, which is important for solving equation whose solution has large variation in a local region.

Example 4.1. The exact solution is taken as $u = 5.0e^{-250.0(x^2+y^2)}$. It is clear that mesh in Fig. 2 is concentrate at the origin. We divide each quadrilateral into two triangles and use triangle element for the discretization.

Example 4.2. The exact solution is $u = e^{-8.0(4.0x^2+9.0y^2-1)^2}$, the corresponding grid distribution is shown in Fig. 3.

Example 4.3. The exact solution is $u = e^{-100.0(y-x^2+0.5)^2}$ and grid distribution is shown in Fig. 4.

To compare MGCG method based on quadratic element and our two level preconditioned method, we use them to solve Poisson equation (4.1). The error distribution $e = |u - u_h|$ is displayed in Figs. 2 , 3 and 4 respectively for different examples, where u is the exact solution and u_h is the numerical solution. The results shown in these

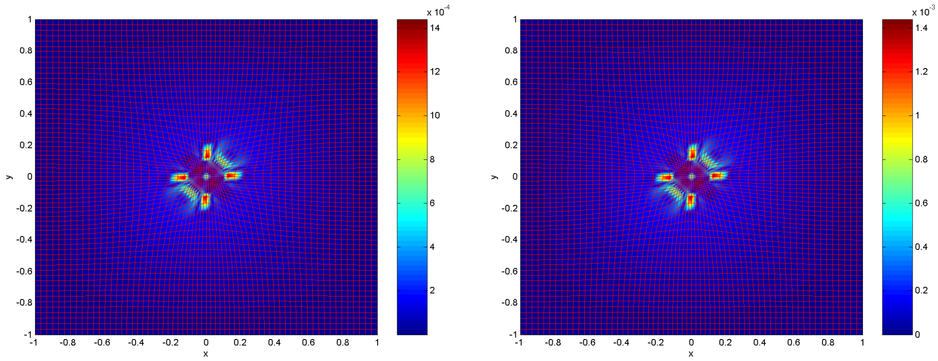

Figure 2: For Example 4.1 in distorted and structured grid, error distributions of two-level preconditioned method (left); MGCG based on quadratic element (right).

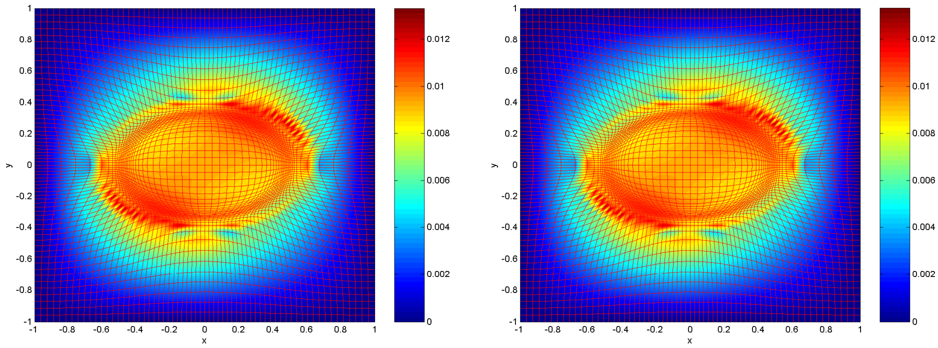

Figure 3: For Example 4.2 in distorted and structured grid, error distributions of two-level preconditioned method (left); MGCG based on quadratic element (right).

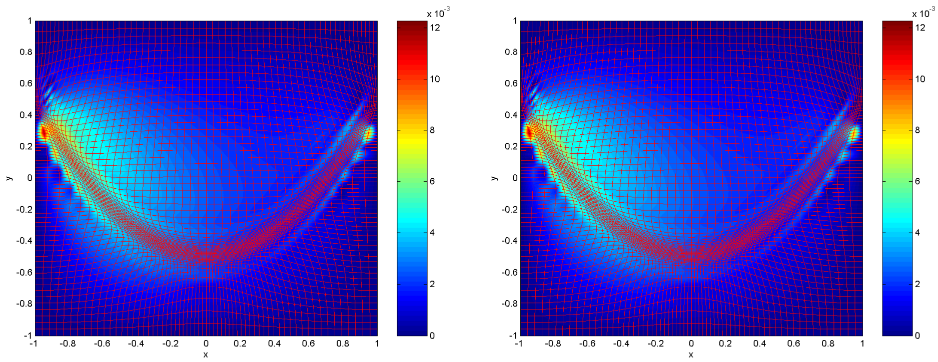

Figure 4: For Example 4.3 in distorted and structured grid, error distributions of two-level preconditioned method (left); MGCG based on quadratic element (right).

figures suggest that the accuracy of the two-level preconditioned method is almost the same as that of MGCG based on quadratic element. We report both C-G iteration numbers and CPU time of these two methods in Table 1 and Table 2. The convergence stopping criterion is $\| r \|/\| r_0 \| < 1.0^{-6}$. We implement 5 Gauss-Seidel iterations

Table 1: The number of C-G iteration comparison.

	Method	32^2 grid	64^2	128^2	256^2
Example 4.1	MGCG based on quadratic element	9	12	16	19
	Two-level preconditioned method	5	7	8	10
Example 4.2	MGCG based on quadratic element	11	16	19	21
	Two-level preconditioned method	6	8	9	10
Example 4.3	MGCG based on quadratic element	15	20	23	23
	Two-level preconditioned method	10	13	15	15

Table 2: The CPU time comparison.

	Method	32^2 grid	64^2	128^2	256^2
Example 4.1	MGCG based on quadratic element	0.06	0.33	1.77	8.33
	Two-level preconditioned method	0.03	0.21	1.02	4.88
Example 4.2	MGCG based on quadratic element	0.07	0.40	1.84	8.16
	Two-level preconditioned method	0.03	0.21	1.05	4.52
Example 4.3	MGCG based on quadratic element	0.08	0.52	2.39	9.79
	Two-level preconditioned method	0.06	0.34	1.68	6.59

during the process of quadratic pre-smoothing and post-smoothing in the two-level preconditioned method. Table 1 and Table 2 suggest that the two-level preconditioned method is more efficient than the other one. Firstly, the number of the C-G iterations is much reduced. Secondly, the CPU time is much less than that of MGCG using the quadratic finite element.

5 Eigenvalue analysis

In order to further investigate the efficiency of the two-level preconditioned method, we compare the eigenvalue distribution of coefficient matrix before and after preconditioning. The iteration number of C-G method depends upon the initial vector, the distribution of eigenvalues of coefficient matrix and the right-hand term. Let us consider Example 4.1 in Section 4, with grids points of 16×16 in computational domain. The condition number of the coefficient matrix corresponding to quadratic element discretization is 5020.3.

A matrix after the preconditioning is calculated as follows. The precondition matrix M of the two-level method is defined in Eq. (3.1), then eigenvalues of the matrix MQ is investigated, where Q is the coefficient matrix corresponding to quadratic element discretization. On the other hand the precondition matrix using MGCG method is calculated in [11] as follows:

$$
\begin{aligned}
&M_0 = L_0^{-1} \quad \text{or} \quad R_0, \\
&M_i = H^m R_i + R_i + H^m p M_{i-1} r (H^T)^m, \qquad i \geq 1.
\end{aligned}
$$

This preconditioner is called V-cycle multigrid preconditioner, where i is the level number of multigrid method, H and R_i are the similar matrices as in (3.1), r and p are

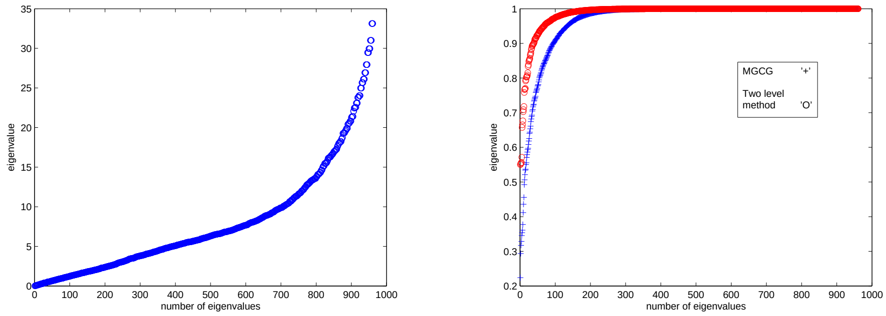

Figure 5: For Example 4.1, eigenvalue distribution before preconditioning (left); eigenvalue distribution after preconditioning (right).

restriction and prolongation matrix respectively.

The eigenvalue distribution of the coefficient matrix before preconditioning of Example 4.1 is shown in Fig. 5(left). The horizontal x axis represents the order of the eigenvalues and the vertical y axis represents the eigenvalues. This y axis behaves in an exp scale. The eigenvalue distribution of the coefficient matrix after preconditioning is shown in Fig. 5(right). This y axis is in a linear scale. In order to do comparison, preconditioning is carried out by the two-level preconditioned method and MGCG method using quadratic element discretization.

The eigenvalue distribution of the two-level preconditioned matrix is effective for the C-G method as the following observations:

1. Almost all eigenvalues are clustered around 1 and a few eigenvalues are scattered between 1 and 0.
2. The smallest eigenvalue is larger than that of the MGCG method with quadratic element discretization.
3. Condition number is decreased.

The first item is no problem for the conjugate gradient method since each eigenvector included in residual vector corresponding to these scattered eigenvalues is vanished in each C-G iteration. All these characteristics are desirable to accelerate the convergence of the conjugate gradient method. From Fig. 5, it is clear that the multigrid preconditioner is very useful for the C-G method. But if we compare two-level preconditioned method and MGCG method, the condition number of the former is smaller than that of the latter. Therefore our two-level preconditioner for the conjugate gradient method is more efficient, especially for cases when grid distribution is not uniform and has a large concentration.

6 Conclusions

We have proposed a two-level preconditioned conjugate gradient method in distorted and structured grid. It is an improvement of C-G method with multigrid preconditioner. The main idea of this two-level preconditioned method is using quadratic smoothing and linear correction. It converges after very few iterations and saves much CPU time. The accuracy is almost the same as that of MGCG based on quadratic finite element. Numerical experiments are presented to show that the two-level preconditioned method is an efficient method. Finally eigenvalue analysis is given in order to further investigate the effect of this method. It concludes that the two-level preconditioner is an excellent preconditioner and it improves the number of the C-G iterations remarkably.

Acknowledgments

The author thanks Prof. Xiaoping Wang for a very stimulating discussion, and Prof. Jinchao Xu for bringing this problem to our attention. The work is partially supported by Youth Foundation of Sichuan University No 2010SCU11072 and Doctoral Fund of Ministry of Education of China No 20110181120090.

References

[1] R. P. FEDORENKO, *A relaxation method for solving elliptic difference equations*, USSR Comput. Math. Math. Phys., 1 (1962), pp. 1092–1096.

[2] R. P. FEDORENKO, *The speed of convergence of one iterative process*, USSR Comput. Math. Math. Phys., 4 (1964), pp. 227–235.

[3] N. S. BAKHVALOV, *On the convergence of a relaxation method with natrural constraints on the elliptic operator*, USSR Comput. Math. Math. Phys., 6 (1966), pp. 101–113.

[4] A. BRANDT, *Multi-level Adaptive technique (MLAT) for fast numerical solution to boundary-value problems*, in: H. Cabannes and R. Temam (eds.), Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, Paris 1972, Lecture Notes in Physics 18, Springer-Verlag, Berlin, 1973.

[5] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comput., 31 (1977), pp. 333–390.

[6] W. HACKBUSCH, *A multi-grid method applied to a boundary problem with variable coefficients in a rectangle*, Report 77-17, Institut für Angewandte Mathematik, Universität Köln, 1977.

[7] W. HACKBUSCH, Multi-Grid Methods and Applications, Springer-Verlag, Heidelberg, 1985.

[8] K. STÜBEN AND U. TROTTENBERG, Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications, in: Hackbusch-Trottenberg [9].

[9] W. Hackbusch and U. Trottenberg(eds.) Multi-Grid Methods, Proceedings, Köln-Porz, Nov. 1981, Lecture Notes in Mathematics 960, Springer-Verlag, Berlin, 1982.

[10] W.Q. REN AND X.P. WANG, *An iterative grid redistribution method for singular problems in multiple dimensions*, J. Comput. Phys., 159 (2000), pp. 246–273.

[11] O. TATEBE, *The multigrid preconditioned conjugate gradient method*, in Proceedings of Sixth Copper Mountain Conference on Multigrid Methods, NASA Conference Publication 3224, April 1993, pp. 621–624.

[12] D. BRAESS, *On the combination of the multigrid method and conjugate gradients*, in Multigrid Methods II (W. Hackbusch and U. Trottenberg, eds.), Vol. 1228 of Lecture Notes in Mathematics, (1986), pp. 52–64, Springer-Verlag.

[13] S. SHU, D. SUN AND J. XU, *An algebraic multigrid method for higher order finite element discretizations*, Comput., 77 (2006), pp. 347–377.

[14] S. SHU, J. XU, Y. YANG AND H. YU, *An algebraic multigrid method for finite element systems on criss-cross grids*, Adv. Comput. Math., 25 (2006), pp. 287–304.

[15] O. TATEBE AND Y. OYANAGI, *Efficient implementation of the multigrid preconditioned conjugate gradient method on distributed memory machines*, in Proceedings of Supercomputing'94, pp. 194–203, IEEE Computer Society, November 1994.

[16] R. E. BANK AND C. C. DOUGLAS, *Sharp estimates for multigrid rates of convergence with general smoothing and acceleration*, SIAM J. Numer. Anal., 22 (1985), pp. 617–633.