Journal of Computational Mathematics, Vol.21, No.4, 2003, 421–434.

GAUSS-SEIDEL-TYPE MULTIGRID METHODS[*1)]

Zhao-hui Huang Qian-shun Chang
(*Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China*)

Abstract

By making use of the Gauss-Seidel-type solution method, the procedure for computing the interpolation operator of multigrid methods is simplified. This leads to a saving of computational time. Three new kinds of interpolation formulae are obtained by adopting different approximate methods, to try to enhance the accuracy of the interpolatory operator. A theoretical study proves the two-level convergence of these Gauss-Seidel-type MG methods. A series of numerical experiments is presented to evaluate the relative performance of the methods with respect to the convergence factor, CPU-time(for one V-cycle and the setup phase) and computational complexity.

Key words: Multigrid methods, Gauss-Seidel solution, Interpolation formula, Convergence.

1. Introduction

Multigrid(MG) methods are very efficient methods for solving linear systems with a broad range of applications. The characteristic feature of multigrid iteration is its fast convergence. This convergence speed does not deteriorate when the discretization is refined, unlike for classical iterative methods which slow down for decreasing grid size. As a consequence an acceptable approximation of the discrete problem can be obtained at the expense of computational work proportional to the number of unknowns, which is also the number of equations in the system. It is not only the complexity which is optimal, but also the constant of proportionality is so small that other methods can rarely surpass multigrid efficiency [14] [2] [12].

Usual multigrid methods try to tailor the components to the problem at hand in order to obtain the highest possible efficiency for the solution process. However, the algebraic multigrid(AMG) method is to choose the components independently of the given problem, uniformly for as large a class of problem as possible. AMG provides a very robust solution method which can be applied directly to structured as well as unstructured grids. The strengths of AMG are exactly its robustness, its applicability in complex geometric situations and its applicability to even solve certain problems which are out of the reach of usual multigrid methods, in particular, problems with no geometric or continuous background at all as long as the given matrix satisfies certain conditions [1] [13].

There now exist various different algebraic approaches, all of which are hierarchical and close to the original AMG idea, but some of which focus on different coarsening and interpolation procedures. An efficient AMG algorithm for M-matrices is described in [13]. In [3] [4] [5] [6] [7] [8] [9] and [10], Chang et al. improve the interpolation operator and present different algorithms to construct the coarse grid equations. In this paper, we apply the Gauss-Seidel solution method to the computation of operator-dependent interpolation, and thus suggest three new Gauss-Seidel-type multigrid methods, whose convergence is also proved. Furthermore, many numerical examples are discussed to compare the efficiency of AMG methods. In section 2, the basic AMG algorithm by Chang is described. Then three different AMG methods are proposed

[*] Received January 15, 2001.
[1)] This work is supported in part by a grant (No. 19931030) from the National Natural Science Foundation of China.

in section 3. In section 4, a convergence analysis for each of the AMG algorithms is given. Numerical analysis and computational results are reported in section 5. Finally, conclusions are presented in section 6.

2. The Basic AMG Algorithm

Consider a (sparse) linear system of equations: $AU=F$ or $\sum_{j=1}^{n} a_{ij}u_j = f_i (i=1,\cdots,n)$. We first have to generate a sequence of smaller and smaller systems of equations: $A^mU^m=F^m$ or $\sum_{j=1}^{n_m} a_{ij}^m u_j^m = f_i^m (i=1,\cdots,n_m)$, where $A^m=(a_{ij}^m)_{n_m\times n_m}, U^m=(u_1^m,\cdots,u_{n_m}^m)^T, F^m=(f_1^m,\cdots,f_{n_m}^m)^T$, $m=1,\cdots,M, n=n_1>\cdots>n_M, A^1=A, U^1=U, F^1=F$.

A fictitious grid Ω^m can be regarded as a set of unknown $u_j^m (1\le j\le n_m)$. The coarser grid Ω^{m+1} is then chosen as a subset of Ω^m, which is denoted by C^m. The remainder subset $\Omega^m - C^m$ is denoted by F^m. A point i is said to be strongly connected to j, if

$$|a_{ij}^m| \ge \theta\cdot \max_{k\ne i}|a_{ik}^m|, 0<\theta\le 1.$$

Let S_i^m denote the set of all atrongly connected points of the point i and let $C_i^m = C^m\bigcap S_i^m, N_i^m=\{j\in\Omega^m, j\ne i, a_{ij}^m\ne 0\}, D_i^m=N_i^m-C_i^m, D_i^s=D_i^m\bigcap S_i^m, D_i^w=D_i^m-D_i^s$.

Each variable in C^m interpolates directly from the corresponding variable in Ω^{m+1} with a weighting of unity, and each variable $i\in F^m$ interpolates from the smaller set C_i^m.

Because the error e_i^m to be interpolated in an AMG method is obtained after a smoothing process, we have

$$a_{ii}^m e_i^m + \sum_{j\in N_i^m} a_{ij}^m e_j^m = d_i^m \approx 0, \forall i\in F^m, \tag{2.1}$$

which can be rewritten as

$$a_{ii}^m e_i^m + \sum_{k\in C_i^m} a_{ik}^m e_k^m + \sum_{j\in D_i^s} a_{ij}^m e_j^m + \sum_{j\in D_i^w} a_{ij}^m e_j^m \approx 0, \forall i\in F^m. \tag{2.2}$$

Let $g_{jk}^m = \frac{|a_{jk}^m|}{\sum_{k\in C_i^m}|a_{jk}^m|}, j\in D_i^m, k\in C_i^m$, for point $j\in D_i^m$, the following approximations are used in (2.2):

(1) For point $j\in D_i^w$,

$$e_j^m = \begin{cases} e_i^m, & \text{if } l_{ij}^m=0, a_{ij}^m<0,\\ -e_i^m, & \text{if } l_{ij}^m=0, a_{ij}^m>0,\\ 2\sum_{k\in C_i^m} g_{jk}^m e_k^m - e_i^m, & \text{if } l_{ij}^m>0, \xi_{ij}^m\ge 0.5, a_{ij}^m<0,\\ \sum_{k\in C_i^m} g_{jk}^m e_k^m, & otherwise. \end{cases} \tag{2.3}$$

(2) For points $j\in D_i^s$, more accurate approximations are used,

$$e_j^m = \begin{cases} 2\sum_{k\in C_i^m} g_{jk}^m e_k^m - e_i^m, & \text{if } \eta_{ij}^m<0.75, \xi_{ij}^m\ge 0.5, a_{ij}^m<0,\\ \frac{1}{2}(\sum_{k\in C_i^m} g_{jk}^m e_k^m + e_i^m), & \text{if } \eta_{ij}^m>2, \xi_{ij}^m\ge 0.5, a_{ij}^m<0,\\ \sum_{k\in C_i^m} g_{jk}^m e_k^m, & otherwise. \end{cases} \tag{2.4}$$

where

$$\xi_{ij}^m = \frac{\sum_{k\in C_i^m} a_{jk}^m}{\sum_{k\in C_i^m}|a_{jk}^m|}, \eta_{ij}^m = \frac{|a_{ji}^m| l_{ij}^m}{\sum_{k\in C_i^m}|a_{jk}^m|}, l_{ij}^m = |S_{ij}^m|, S_{ij}^m = \{k : k\in C_i^m, a_{jk}^m\ne 0\}.$$

Substituting (2.3)-(2.4) into (2.2) is equivalent to modifying the coefficients in (2.2) by combining the following steps:

Step(1) add $-|a_{ij}^m|$ to $a_{ii}^m, \forall j\in D_i^{(1)}$, which is equivalent to e_j^m being replaced by e_i^m or $-e_i^m$;

Step(2) add $a_{ij}^m g_{jk}^m$ to $a_{ik}^m, \forall k\in C_i^m, \forall j\in D_i^{(2)}$, which is equivalent to e_j^m being approximated by $\sum_{k\in C_i^m} g_{jk}^m e_k^m$;

Step(3) add $2a_{ij}^m g_{jk}^m$ to $a_{ik}^m, \forall k \in C_i^m$, and subtract a_{ij}^m from $a_{ii}^m, \forall j \in D_i^{(3)}$, which is equivalent to e_j^m being approximated by $2\sum_{k\in C_i^m} g_{jk}^m e_k^m - e_i^m$;

Step(4) add $0.5a_{ij}^m g_{jk}^m$ to $a_{ik}^m, \forall k \in C_i^m$, and add $0.5a_{ij}^m$ to $a_{ii}^m, \forall j \in D_i^{(4)}$, which is equivalent to e_j^m being approximated by $0.5(\sum_{k\in C_i^m} g_{jk}^m e_k^m + e_i^m)$;

where $D_i^{(l)} = \{j : j \in D_i^m, e_j^m$ is eliminated by the corresponding step $(l)\}(l = 1,2,3,4)$.

Thus, an interpolation formula derived from (2.2) is given by

$$e_i^m = \sum_{j\in C_i^m} w_{ij}^m e_j^{m+1}, \forall i \in F^m, \tag{2.5}$$

where

$$\begin{aligned} w_{ik}^m &= -\frac{\bar{a}_{ik}^m}{\bar{a}_{ii}^m}, k \in C_i^m, \\ \bar{a}_{ii}^m &= a_{ii}^m - \sum_{j\in D_i^{(1)}} |a_{ij}^m| - \sum_{j\in D_i^{(3)}} a_{ij}^m + 0.5\sum_{j\in D_i^{(4)}} a_{ij}^m, \\ \bar{a}_{ik}^m &= a_{ik}^m + \sum_{j\in D_i^{(2)}} a_{ij}^m g_{jk}^m + 2\sum_{j\in D_i^{(3)}} a_{ij}^m g_{jk}^m + 0.5\sum_{j\in D_i^{(4)}} a_{ij}^m g_{jk}^m. \end{aligned} \tag{2.6}$$

We refer to the above algorithm as **Method I**.

3. Three Gauss-Seidel-type Algorithms

We can obtain three different new AMG methods by adopting different interpolation operators to be discussed in this section.

Equation (2.1) can be rewritten as

$$a_{ii}^m e_i^m + \sum_{k\in C_i^m} a_{ik}^m e_k^m + \sum_{j\in D_i^m} a_{ij}^m e_j^m \approx 0, \forall i \in F^m. \tag{3.1}$$

For each point $i \in F^m$, when computing the interpolation formula by using (3.1), we approximate for all points $j \in D_i^m$ by using (2.3)-(2.4), which is referred to as a Jacobi-type solution method. In fact, for points $i \in F^m$, only those points $j \in D_i^m$, and $j \geq i$ have to be approximated by using (2.3)-(2.4). This is because the interpolation formulae (2.5)-(2.6) for those points $j \in D_i^m$, and $j < i$ have been obtained by the previous computation. Thus, we derive the following three Gauss-Seidel-type solution methods to be given shortly.

More precisely, (3.1) can further be rewritten as

$$a_{ii}^m e_i^m + \sum_{k\in C_i^m} a_{ik}^m e_k^m + \sum_{j\in D_i^m, j<i} a_{ij}^m e_j^m + \sum_{j\in D_i^m, j\geq i} a_{ij}^m e_j^m \approx 0, \forall i \in F^m. \tag{3.2}$$

For points $j \in D_i^m$, and $j < i$, we use the interpolation formula (2.5)-(2.6); however, for points $j \in D_i^m$, and $j \geq i$, we still use the approximations (2.3)-(2.4), and thus $e_i^m = \sum_{j\in C_i^m} \bar{w}_{ik}^m e_k^{m+1}$. Then we approximately obtain

$$e_i^m = -\frac{1}{a_{ii}^m}\Big(\sum_{k\in C_i^m} a_{ik}^m e_k^{m+1} + \sum_{j\in D_i^m, j<i} \sum_{k\in C_j^m} a_{ij}^m w_{jk}^m e_k^{m+1} + \sum_{j\in D_i^m, j\geq i} \sum_{k\in C_i^m} a_{ij}^m \bar{w}_{jk}^m e_k^{m+1}\Big). \tag{3.3}$$

In the above formula, for points $k \in C_j^m, j \in D_i^m, j < i$, but $k \bar{\in} C_i^m$, we make the following three different approximations:

Method II $e_k^{m+1} \approx e_i^{m+1}$. We derive from (3.3)

$$e_i^m = -\frac{\sum\limits_{k\in C_i^m} a_{ik}^m e_k^{m+1} + \sum\limits_{j\in D_i^m, j<i} \sum\limits_{k\in C_j^m \bigcap C_i^m} a_{ij}^m w_{jk}^m e_k^{m+1} + \sum\limits_{j\in D_i^m, j\geq i} \sum\limits_{k\in C_i^m} a_{ij}^m \bar{w}_{jk}^m e_k^{m+1}}{a_{ii}^m + \sum\limits_{j\in D_i^m, j<i} \sum\limits_{k\in C_j^m\backslash C_i^m} a_{ij}^m w_{jk}^m}.$$

Let $w_{jk}^m = 0, k \in C_i^m \backslash C_j^m, j \in D_i^m, j < i$, and then we have

$$e_i^m = \sum_{k\in C_i^m} W_{ik}^m(1) e_k^{m+1}, \forall i \in F^m, \tag{3.4}$$

where

$$W_{ik}^m(1) = -\frac{a_{ik}^m + \sum\limits_{j\in D_i^m, j<i} a_{ij}^m w_{jk}^m + \sum\limits_{j\in D_i^m, j\geq i} a_{ij}^m \bar{w}_{jk}^m}{a_{ii}^m + \sum\limits_{j\in D_i^m, j<i} \sum\limits_{k\in C_j^m\backslash C_i^m} a_{ij}^m w_{jk}^m}. \tag{3.5}$$

Method III $e_k^{m+1} \approx e_j^{m+1}$, which is possibly more accurate than Method II because of the geometric assumption that the error between points k and j is smaller than the one between points k and i. We derive from (3.3)

$$e_i^m = -\frac{\sum\limits_{k\in C_i^m} a_{ik}^m e_k^{m+1} + \sum\limits_{j\in D_i^m, j<i} \sum\limits_{k\in C_j^m \bigcap C_i^m} \frac{a_{ij}^m w_{jk}^m}{1 - \sum\limits_{k\in C_j^m\backslash C_i^m} w_{jk}^m} e_k^{m+1} + \sum\limits_{j\in D_i^m, j\geq i} \sum\limits_{k\in C_i^m} a_{ij}^m \bar{w}_{jk}^m e_k^{m+1}}{a_{ii}^m}.$$

Let $w_{jk}^m = 0, k \in C_i^m \backslash C_j^m, j \in D_i^m, j < i$, and then we have

$$e_i^m = \sum_{k\in C_i^m} W_{ik}^m(2) e_k^{m+1}, \forall i \in F^m, \tag{3.6}$$

where

$$W_{ik}^m(2) = -\frac{a_{ik}^m + \sum\limits_{j\in D_i^m, j<i} \frac{a_{ij}^m w_{jk}^m}{1 - \sum\limits_{k\in C_j^m\backslash C_i^m} w_{jk}^m} + \sum\limits_{j\in D_i^m, j\geq i} a_{ij}^m \bar{w}_{jk}^m}{a_{ii}^m}. \tag{3.7}$$

Method IV $e_k^{m+1} \approx 0$. We derive from (3.3)

$$e_i^m = -\frac{\sum\limits_{k\in C_i^m} a_{ik}^m e_k^{m+1} + \sum\limits_{j\in D_i^m, j<i} \sum\limits_{k\in C_j^m \bigcap C_i^m} a_{ij}^m w_{jk}^m e_k^{m+1} + \sum\limits_{j\in D_i^m, j\geq i} \sum\limits_{k\in C_i^m} a_{ij}^m \bar{w}_{jk}^m e_k^{m+1}}{a_{ii}^m}.$$

Let $w_{jk}^m = 0, k \in C_i^m \backslash C_j^m, j \in D_i^m, j < i$, and then we have

$$e_i^m = \sum_{k\in C_i^m} W_{ik}^m(3) e_k^{m+1}, \forall i \in F^m, \tag{3.8}$$

where

$$W_{ik}^m(3) = -\frac{a_{ik}^m + \sum\limits_{j\in D_i^m, j<i} a_{ij}^m w_{jk}^m + \sum\limits_{j\in D_i^m, j\geq i} a_{ij}^m \bar{w}_{jk}^m}{a_{ii}^m}. \tag{3.9}$$

4. Convergence Analysis

In this section, the basic convergence results for Gauss-Seidel-type algorithms are given.

Let $G^m : G(\Omega^m) \to G(\Omega^m)$, the smoothing operator, where $G(\Omega^m)$ denotes the linear space of grid functions on the grid Ω^m;

$I_{m+1}^m : G(\Omega^{m+1}) \to G(\Omega^m)$, the interpolation operators;
$I_m^{m+1} : G(\Omega^m) \to G(\Omega^{m+1})$, the restriction operators;
$A^{m+1} : G(\Omega^m) \to G(\Omega^m)$, the coarse-grid operators;
$T^m = I^m - I_{m+1}^m (A^{m+1})^{-1} I_m^{m+1} A^m$, (m, m+1) two-level correction operator, respectively.

We use the following three inner products besides the Euclidean inner product $(\cdot,\cdot)$: $(u,v)_0 = (Du,v), (u,v)_1 = (Au,v), (u,v)_2 = (D^{-1}Au, Av)$ together with their corresponding norms $\|\cdot\|_i (i = 0,1,2)$. Here, $D = diag(A), (\cdot,\cdot)_1$ is the so-called energy inner product and $\|\cdot\|_1$ the energy norm.

In [13], Ruge and Stueben prove the following theorems which will be used in our analysis:

Theorem 4.1. *Let $A > 0$. Assume that the interpolation operators I_{m+1}^m have full rank and the restriction and coarse-grid operators are defined by $I_m^{m+1} = I_{m+1}^m$ and $A^{m+1} = I_m^{m+1} A^m I_{m+1}^m$. Further, suppose that, for all $e^m, \|G^m e^m\|_1^2 \leq \|e^m\|_1^2 - \delta \|T^m e^m\|_1^2$ holds with some $\delta > 0$ independent of e^m and m. Then $\delta \leq 1$, and—provided that the coarsest grid equation is solved and at least one smoothing step is performed after each coarse grid correction step—the V-cycle to solve $AU = F$ has a convergence factor (with respect to the energy norm) bounded above by $\sqrt{1-\delta}$.*

We say that a relaxation operator G^m satisfies the smoothing property with respect to a matrix $A > 0$ if

$$\|G^m e^m\|_1^2 \leq \|e^m\|_1^2 - \alpha_m \|e^m\|_2^2 \quad (\alpha_m > 0) \tag{4.1}$$

holds with α_m being independent of e^m.

Lemma. *Let $A^m > 0$ and let the smoothing operator be of the form $G^m = I - Q^{-1}A$ with some non-singular matrix Q. Then the smoothing property is equivalent to*

$$\alpha_m Q^T D^{-1} Q \leq Q + Q^T - A.$$

Theorem 4.2. *Let $A^m > 0$ and define, with any vector*

$$w^m = (w_i^m) > 0, r_-^{(m)} = max_i \{ \frac{1}{w_i^m a_{ii}^m} \sum_{j<i} w_j^m |a_{ij}^m| \}, r_+^{(m)} = max_i \{ \frac{1}{w_i^m a_{ii}^m} \sum_{j>i} w_j^m |a_{ij}^m| \}.$$

Then the Gauss-Seidel relaxation satisfies (4.1) with $\alpha_m = 1/(1+\gamma_-^{(m)})(1+\gamma_+^{(m)})$.

Theorem 4.3. *Let $A^m > 0$ and $\eta_m \geq \rho((D^m)^{-1}A^m)$. Then Jacobi relaxation with relaxation parameter $0 < \omega^m < 2/\eta_m$ satisfies (4.1) with $\alpha_m = \omega^m(2 - \omega^m \eta_m)$. In terms of η_m, the optimal parameter (which gives the largest value of α_m) is $\omega^* = 1/\eta_m$. For this optimal parameter, the smoothing property is satisfied with $\alpha_m = 1/\eta_m$.*

Theorem 4.4. *Let $A^m > 0$ and let $G^m > 0$ satisfy (4.1). Suppose that the interpolation operator I_{m+1}^m has a full rank and that, for each e^m,*

$$min_{e^{m+1}} \|e^m - I_{m+1}^m e^{m+1}\|_0^2 \leq \beta_m \|e^m\|_1^2 \tag{4.2}$$

with $\beta_m > 0$ independent of e^m. Then $\beta_m \geq \alpha_m$, and the (m, m+1) two-level convergence factor satisfies: $\|G^m T^m\|_1 \leq \sqrt{1 - \alpha_m/\beta_m}$.

In [11], Huang extends the results of Ruge and Stueben and gives another sufficient condition for (4.2).

In [8], Chang presents the proof of convergence for the AMG method with the interpolation formula (2.5)-(2.6).

Here, concerning the interpolation formulae (3.4)-(3.9), we have the following theorem:

Theorem 4.5. *Let $A^m > 0$, and assume A^m is a weakly diagonally dominant matrix. suppose the C-points are picked in such a way that, for each $i \in F^m$, and omitting the indices of levels, we have*

$$\sum_{k \in C_i} a_{ik}^+ \geq \sum_{j \in F \bigcap N_i} a_{ij}^+, \text{ and } \sum_{k \in C_i} a_{ik}^- \leq \sum_{j \in F \bigcap N_i} a_{ij}^-,$$

where

$$a_{ij}^- = \begin{cases} a_{ij} & (\text{if} \quad a_{ij} < 0) \\ 0 & (\text{if} \quad a_{ij} \geq 0), \end{cases} \quad \text{and } a_{ij}^+ = \begin{cases} 0 & (\text{if} \quad a_{ij} \leq 0) \\ a_{ij} & (\text{if} \quad a_{ij} > 0). \end{cases}$$

Then the interpolation formulae (3.4)-(3.9) satisfy estimate (4.2).

Proof. For simplicity, we only prove (4.2) with respect to the interpolation formula (3.8)-(3.9) as follows:

$$\begin{aligned}
& e_i - \textstyle\sum_{k\in C_i} W_{ik}(3)e_k \\
= \;& e_i + \tfrac{1}{a_{ii}} \textstyle\sum_{k\in C_i} a_{ik}e_k - \tfrac{1}{a_{ii}} \sum_{k\in C_i}[\sum_{j\in F\bigcap N_i} \tfrac{a_{ij}}{\bar{a}_{jj}}(a_{jk} + \sum_{l\in D_j^{(2)}} a_{jl}g_{lk} \\
& +2\textstyle\sum_{l\in D_j^{(3)}} a_{il}g_{lk} + 0.5\sum_{l\in D_j^{(4)}} a_{il}g_{lk})]e_k \\
= \;& e_i + \tfrac{1}{a_{ii}} \textstyle\sum_{k\in C_i} a_{ik}^+e_k - \tfrac{1}{a_{ii}} \sum_{k\in C_i}[\sum_{j\in F\bigcap N_i} \tfrac{a_{ij}^+}{\bar{a}_{jj}}(a_{jk}^+ + \sum_{l\in D_j^{(2)}} a_{jl}^+g_{lk})]e_k \\
& -\tfrac{1}{a_{ii}} \textstyle\sum_{k\in C_i}[\sum_{j\in F\bigcap N_i} \tfrac{a_{ij}^-}{\bar{a}_{jj}}(a_{jk}^- + \sum_{l\in D_j^{(2)}} a_{jl}^-g_{lk} + 2\sum_{l\in D_j^{(3)}} a_{il}g_{lk} + 0.5 \\
& \textstyle\sum_{l\in D_j^{(4)}} a_{il}g_{lk})]e_k + \tfrac{1}{a_{ii}} \sum_{k\in C_i} a_{ik}^-e_k - \tfrac{1}{a_{ii}} \sum_{k\in C_i}[\sum_{j\in F\bigcap N_i} \tfrac{a_{ij}^+}{\bar{a}_{jj}}(a_{jk}^- + \\
& \textstyle\sum_{l\in D_j^{(2)}} a_{jl}^-g_{lk} + 2\sum_{l\in D_j^{(3)}} a_{il}g_{lk} + 0.5\sum_{l\in D_j^{(4)}} a_{il}g_{lk})]e_k - \tfrac{1}{a_{ii}} \sum_{k\in C_i} \\
& [\textstyle\sum_{j\in F\bigcap N_i} \tfrac{a_{ij}^-}{\bar{a}_{jj}}(a_{jk}^+ + \sum_{l\in D_j^{(2)}} a_{jl}^+g_{lk})]e_k \\
= \;& E_{ik}^-(e_i - e_k) + E_{ik}^+(e_i + e_k) + E_{ii}e_i,
\end{aligned}$$

where

$$\begin{aligned}
E_{ik}^- &= -\tfrac{1}{a_{ii}} \textstyle\sum_{k\in C_i}\{a_{ik}^- - \sum_{j\in F\bigcap N_i}[\tfrac{a_{ij}^-}{\bar{a}_{jj}}(a_{jk}^+ + \sum_{l\in D_j^{(2)}} a_{jl}^+g_{lk}) \\
&\quad +\tfrac{a_{ij}^+}{\bar{a}_{jj}}(a_{jk}^- + \textstyle\sum_{l\in D_j^{(2)}} a_{jl}^-g_{lk} + 2\sum_{l\in D_j^{(3)}} a_{il}g_{lk} + 0.5\sum_{l\in D_j^{(4)}} a_{il}g_{lk})]\}, \\
E_{ik}^+ &= \tfrac{1}{a_{ii}} \textstyle\sum_{k\in C_i}\{a_{ik}^+ - \sum_{j\in F\bigcap N_i}[\tfrac{a_{ij}^+}{\bar{a}_{jj}}(a_{jk}^+ + \sum_{l\in D_j^{(2)}} a_{jl}^+g_{lk}) \\
&\quad +\tfrac{a_{ij}^+}{\bar{a}_{jj}}(a_{jk}^- + \textstyle\sum_{l\in D_j^{(2)}} a_{jl}^-g_{lk} + 2\sum_{l\in D_j^{(3)}} a_{il}g_{lk} + 0.5\sum_{l\in D_j^{(4)}} a_{il}g_{lk})]\}, \\
E_{ii} &= \tfrac{1}{a_{ii}}(a_{ii} + \textstyle\sum_{k\in C_i}\{a_{ik}^- - \sum_{j\in F\bigcap N_i}[\tfrac{a_{ij}^-}{\bar{a}_{jj}}(a_{jk}^+ + \sum_{l\in D_j^{(2)}} a_{jl}^+g_{lk}) \\
&\quad +\tfrac{a_{ij}^-}{\bar{a}_{jj}}(a_{jk}^- + \textstyle\sum_{l\in D_j^{(2)}} a_{jl}^-g_{lk} + 2\sum_{l\in D_j^{(3)}} a_{il}g_{lk} + 0.5\sum_{l\in D_j^{(4)}} a_{il}g_{lk})]\} \\
&\quad -\textstyle\sum_{k\in C_i}\{a_{ik}^+ - \sum_{j\in F\bigcap N_i}[\tfrac{a_{ij}^+}{\bar{a}_{jj}}(a_{jk}^+ + \sum_{l\in D_j^{(2)}} a_{jl}^+g_{lk}) \\
&\quad +\tfrac{a_{ij}^+}{\bar{a}_{jj}}(a_{jk}^- + \textstyle\sum_{l\in D_j^{(2)}} a_{jl}^-g_{lk} + 2\sum_{l\in D_j^{(3)}} a_{il}g_{lk} + 0.5\sum_{l\in D_j^{(4)}} a_{il}g_{lk})]\}).
\end{aligned}$$

In order to use the inequality

$$(\sum_j a_ju_j)^2 \leq \sum_j a_ju_j^2, a_j \geq 0, \sum_j a_j \leq 1,$$

we have to verify the corresponding conditions:

The fact that $E_{ik}^- + E_{ik}^+ + E_{ii} = 1$, and $E_{ii} \geq 0$ is evident; and we easily deduce $E_{ik}^+ > 0$ if $\sum_{k\in C_i} a_{ik}^+ \geq \sum_{j\in F\bigcap N_i} a_{ij}^+$, and $E_{ik}^- > 0$ if $\sum_{k\in C_i} a_{ik}^- \leq \sum_{j\in F\bigcap N_i} a_{ij}^-$.

Then we have the following estimate

$$\min_{e^{m+1}} \|e^m - I_{m+1}^m e^{m+1}\|_0^2 \leq \sum_{i\in F} a_{ii}(E_{ii}e_i^2 + E_{ik}^+(e_i + e_k)^2 + E_{ik}^-(e_i - e_k)^2).$$

Hence, from the equality

$$
\begin{aligned}
\|e^m\|_1^2 &= (Ae,e)_E = \sum_{i,j} a_{ij}e_ie_j \\
&= 1/2\sum_{i,j}(-a_{ij})(e_i-e_j)^2 + \sum_i\sum_j a_{ij}e_i^2 \\
&= 1/2\sum_{i,j}(-a_{ij}^-)(e_i-e_j)^2 - 1/2\sum_{i,j} a_{ij}^+(e_i-e_j)^2 + \sum_i\sum_j a_{ij}e_i^2 \\
&= 1/2\sum_{i,j}(-a_{ij}^-)(e_i-e_j)^2 + \sum_i\sum_{j\neq i} a_{ij}^+(2e_i^2-(e_i-e_j)^2/2) \\
&\quad + \sum_i(a_{ii}-\sum_{j\neq i}|a_{ij}|)e_i^2 \\
&= 1/2\sum_i(\sum_{j\neq i}(-a_{ij}^-)(e_i-e_j)^2 + \sum_{j\neq i} a_{ij}^+(e_i+e_j)^2) + \sum_i(a_{ii}-\sum_{j\neq i}|a_{ij}|)e_i^2,
\end{aligned}
$$

it follows that (4.2) holds with $\beta_m \geq 2$.

The proof of two-level convergence for the AMG method with the interpolation formulae (3.4)-(3.5) or (3.6)-(3.7) is similar.

Theorem 4.6. *Assume A^m is a symmetric positive definite matrix with weakly diagonal dominance. Let the C-points be picked in such a way that, for each $i \in F^m$, the inequalities $\sum_{k\in C_i^m} a_{ik}^+ \geq \sum_{j\in F\bigcap N_i} a_{ij}^+, \sum_{k\in C_i^m} a_{ik}^- \leq \sum_{j\in F\bigcap N_i} a_{ij}^-$ are satisfied. Suppose that the interpolation formulae (3.4)-(3.5), (3.6)-(3.7) or (3.8)-(3.9) and Gauss-Seidel relaxation (or Jacobi relaxation with parameter $0 < \omega^m < 2/\gamma_0^m, \gamma_0^m \geq \rho((D^m)^{-1}A^m)$) are used in the AMG method. Then the (m, m+1)-two-level AMG algorithm is convergent with factor $\|G^mT^m\|_1 \leq \sqrt{1-\alpha_m/\beta_m}$, where $\beta_m \geq \alpha_m, \alpha_m$ is given by Theorem 4.2 or Theorem 4.3.*

Theorem 4.7. *Suppose the matrix A^m is symmetric positive definite and weakly diagonally dominant. If the interpolation weights satisfy $\frac{w_{ij}^m}{|w_{ij}^m|} = -\frac{a_{ij}^m}{|a_{ij}^m|}, |a_{ij}^m| \geq |w_{ij}^m|a_{ii}^m, \forall i \in F^m, j \in C_i^m$, then the coarse grid operator A^{m+1} is also symmetric positive definite and weakly diagonally dominant.*

5. Numerical Analysis and Computational Results

In this section, we present some numerical results for various problems by applying the Gauss-Seidel-type multigrid methods given in this paper to evaluate their performance. Numerical results are compared with those obtained using the standard AMG algorithm (Method I). Particular attention is focused on the convergence factor and the CPU-time consumed.

The following notations are used for the results reported in all tables:

ρ : asymptotic convergence factor,

t_I: computing time in seconds for one V-cycle,

t_p: computing time for the setup phase,

N: number of iterations for convergence defined by $\|r^N\|/\|r^0\| \leq 10^{-6}$, where r^N is the residual vector at the N-th iteration,

EQ: total number of matrix equations,

σ^A: ratio of the space occupied by all operators to the space at the finest grid,

σ^Ω: ratio of the total number of points on all grids to that on the finest grid.

In all computations, the initial iteration u^0 is taken to be random numbers uniformly distributed in [0,1], and Gauss-Seidel relaxation is used as the smoothing operator and $\theta = 0.25$.

Problem 1. Poisson problem on a unit square with Dirichlet boundary conditions(5-point discretization).

First, we consider the following standard 5-point difference stencil

$$
L_h^{sd} = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h .
$$

Subsequently, the following skew 5-point stencil

$$L_h^{sw} = \frac{1}{2h^2}\begin{bmatrix} -1 & & -1 \\ & 4 & \\ -1 & & -1 \end{bmatrix}_h,$$

is considered. Applying the Local Fourier Analysis philosophy, we obtain the h-ellipticity measure for this operator

$$E_h(L_h^{sw}) := \frac{\min\{|\tilde{L}_h^{sw}(\theta)| : \frac{\pi}{2} \le |\theta| < \pi\}}{\max\{|\tilde{L}_h^{sw}(\theta)| : 0 \le |\theta| \le \pi\}} = 0,$$

where $\tilde{L}_h^{sw}(\theta)$ represents the Fourier symbol of L_h^{sw}. However Method IV is able to solve this discrete operator efficiently.

The computational results for the problem are given in Table 1 and 2.

Table 1 Numerical results for L_h^{sd}

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	64×64	0.021	4	0.17	0.040	2.16	1.66
	128×128	0.022	4	0.61	0.123	2.18	1.67
II	64×64	0.062	5	0.17	0.056	2.16	1.66
	128×128	0.079	6	0.50	0.128	2.18	1.67
III	64×64	0.078	6	0.22	0.055	2.16	1.66
	128×128	0.087	6	0.44	0.137	2.18	1.67
IV	64×64	0.020	4	0.11	0.043	2.16	1.66
	128×128	0.020	4	0.44	0.123	2.18	1.67

Table 2 Numerical results for L_h^{sw}

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	64×64	0.042	5	0.11	0.022	2.16	1.67
	128×128	0.045	5	0.55	0.120	2.18	1.67
II	64×64	0.061	5	0.17	0.032	2.15	1.66
	128×128	0.061	5	0.44	0.125	2.18	1.67
III	64×64	0.056	5	0.10	0.044	2.16	1.66
	128×128	0.060	5	0.61	0.110	2.18	1.67
IV	64×64	0.029	4	0.11	0.028	2.17	1.67
	128×128	0.029	4	0.44	0.138	2.18	1.67

Problem 2. Poisson problem on a unit square with Dirichlet boundary condition(9-point discretization).

First, we consider the following 9-point difference stencil

$$L_h^{(9)} = \frac{1}{6h^2}\begin{bmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{bmatrix}_h,$$

which can be obtained by $\frac{2}{3}L_h^{sd} + \frac{1}{3}L_h^{sw}$.

Secondly, if I_m^{m+1} and $I_{m+1}^m (m = 1, 2, \cdots)$ are defined as full weight for the restriction and bilinear interpolation as the prolongation, we obtain the following coarse grid Galerkin discretization from the standard 5-point discrete Laplace operator $A^m, A^{m+1} = I_m^{m+1}A^m I_{m+1}^m$. For $m \to \infty, A^m$ converges to a difference operator which is characterized by the 9-point difference stencil

$$L_h^{(9-limit)} = \frac{1}{3h^2}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}_h,$$

for which improved results are obtained by using Method IV.

The computational results for this problem are given in Table 3 and 4.

Table 3 Numerical results for $L_h^{(9)}$

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	32×32	0.075	6	0.11	0.028	1.31	1.33
	48×48	0.075	6	0.11	0.018	1.31	1.33
II	32×32	0.097	6	0.11	0.028	1.31	1.33
	48×48	0.098	6	0.11	0.010	1.32	1.33
III	32×32	0.083	6	0.06	0.028	1.31	1.33
	48×48	0.086	6	0.11	0.018	1.32	1.33
IV	32×32	0.060	5	0.06	cta0	1.31	1.33
	48×48	0.060	5	0.11	0.012	1.32	1.33

cta0: CPU-time approximates 0.

Table 4 Numerical results for $L_h^{(9-limit)}$

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	48×48	0.083	6	0.05	0.008	1.32	1.33
	64×64	0.083	6	0.16	0.037	1.32	1.33
II	48×48	0.116	7	0.05	0.024	1.32	1.33
	64×64	0.115	7	0.16	0.031	1.32	1.33
III	48×48	0.116	7	0.06	0.016	1.32	1.33
	64×64	0.118	7	0.11	0.040	1.32	1.33
IV	48×48	0.059	5	0.06	0.022	1.32	1.33
	64×64	0.058	5	0.11	0.044	1.32	1.33

Problem 3. To compare with Problem 1 and 2, we consider another 5-point difference stencil

$$L_h^{hs} = \frac{1}{h^2}\begin{bmatrix} & 1 & \\ -1 & 4 & -1 \\ & 1 & \end{bmatrix}_h ,$$

whose corresponding matrix has the very special property that algebraically smooth error is geometrically smooth only in the x-direction but strongly oscillatory in the y-direction. Thus the error between any two horizontal gridlines is strongly related.

As a result, this is a difficult problem for standard AMG or even Gauss-Seidel-type MG methods. Its convergence factors are not satisfactory, which can be observed from the computational results in Table 5.

Consequently, we have to further improve the AMG algorithm based on Gauss-Seidel-type MG methods. Up to now, we know that an additional fully relaxed Jacobi-interpolation step can solve this problem successfully, and will give the results in future paper [10].

The computational results for this problem are given in Table 5.

Table 5 Numerical results for L_h^{hs}

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	64×64	0.917	159	0.11	0.022	2.20	1.72
	128×128	0.938	216	0.50	0.090	2.22	1.73
II	64×64	0.906	140	0.11	0.022	2.28	1.69
	128×128	0.929	189	0.44	0.091	2.32	1.69
III	64×64	0.905	138	0.06	0.023	2.28	1.69
	128×128	0.928	186	0.44	0.092	2.38	1.70
IV	64×64	0.858	90	0.11	0.025	2.29	1.69
	128×128	0.883	122	0.39	0.093	2.33	1.69

Problem 4. General matrix equations with large off-diagonal positive entries.

We use the following stencils:

$$L_h^{N1} = \frac{1}{h^2}\begin{bmatrix} & 1 & \\ 0 & 4 & 1 \\ & -1 & \end{bmatrix}_h,$$

and

$$L_h^{N2} = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & 3 \\ & -1 & \end{bmatrix}_h,$$

whose corresponding matrices are non-symmetric and non-diagonally dominant. Thus these are very difficult problems.

The computational results for this problem are given in Table 6 and 7.

Table 6 Numerical results for L_h^{N1}

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	64×64	0.009	3	0.11	0.037	2.53	2.02
	128×128	0.009	3	0.50	0.180	2.55	2.02
II	64×64	0.003	3	0.17	0.037	3.78	2.02
	128×128	0.003	3	0.71	0.147	3.84	2.02
III	64×64	0.003	3	0.11	0.053	3.78	2.02
	128×128	0.003	3	0.71	0.147	3.84	2.02
IV	64×64	0.003	3	0.11	0.057	3.79	2.02
	128×128	0.003	3	0.66	0.183	3.87	2.02

Table 7 Numerical results for L_h^{N2}

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	48×48	0.067	6	0.05	0.028	3.76	2.12
	64×64	0.090	6	0.22	0.047	3.95	2.16
II	48×48	0.065	6	0.11	0.027	4.14	2.13
	64×64	0.084	6	0.17	0.053	4.37	2.16
III	48×48	0.065	6	0.05	0.028	4.14	2.12
	64×64	0.084	6	0.22	0.063	4.36	2.16
IV	48×48	0.066	6	0.16	0.018	4.20	2.14
	64×64	0.086	6	0.22	0.055	4.39	2.17

Problem 5. The equation with cross-derivative terms

$$L^\varepsilon u = -\Delta u + \varepsilon u_{xy} = f.$$

This problem is elliptic for $|\varepsilon| < 2$, parabolic for $|\varepsilon| = 2$, and hyperbolic for $|\varepsilon| > 2$. Hence standard grid coarsening is problematic for certain values of ε and more robust smoothers, like modified ILU, have to be used to handle the problems on a fine grid in GMG method. However, AMG changes the grid coarsening process and directs it towards the problem at hand, while keeping a cheap point smoother. The differential operator L^ε can not be discretized consistently by a difference stencil which is axially symmetric for $\varepsilon \neq 0$, but can be done by the following second order and non-symmetric 7-point differential stencil

$$L_h^{(7)}(\varepsilon) = \frac{1}{h^2}\begin{bmatrix} -\frac{\varepsilon}{2} & -1+\frac{\varepsilon}{2} & 0 \\ -1+\frac{\varepsilon}{2} & 4-\varepsilon & -1+\frac{\varepsilon}{2} \\ 0 & -1+\frac{\varepsilon}{2} & -\frac{\varepsilon}{2} \end{bmatrix}_h,$$

where ε =1.5 is used.

The computational results for the problem are given in Table 8.

Table 8 Numerical results for $L_h^{(7)}(\varepsilon)$

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	64×64	0.053	5	0.11	0.022	1.81	1.44
	128×128	0.058	5	0.17	0.022	1.83	1.44
II	64×64	0.083	6	0.06	0.018	1.82	1.44
	128×128	0.084	6	0.11	0.037	1.83	1.44
III	64×64	0.080	6	0.05	0.018	1.81	1.44
	128×128	0.087	6	0.11	0.037	1.83	1.44
IV	64×64	0.053	5	0.06	0.022	1.81	1.44
	128×128	0.052	5	0.11	0.032	1.83	1.44

Problem 6. First, we consider the Poisson problem on a unit square with Neumann boundary conditions. The Poisson equation is approximated by the following standard 5-point difference star

$$L_h^{sd} = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}_h ,$$

whose coefficient matrix is symmetric nonnegative definite with $|a_{ii}^1| \geq \sum_{j\neq i}|a_{ij}^1|$. For such a singular system [3], we obtain as good convergence as those corresponding Dirichlet problems in which the coefficient matrix is symmetric positive definite and diagonally dominant.

Next, as an extreme case, we consider the difference stencil

$$L_h^{e} = \frac{1}{h^2}\begin{bmatrix} & 1 & \\ 1 & 4 & 1 \\ & 1 & \end{bmatrix}_h$$

on a unit square with Neumann boundary conditions. The coefficient matrix is similar to that for the standard 5-point difference scheme for the Poisson equation except that some diagonal entries are larger and the sign of all off-diagonal entries has changes from negative to positive. As a consequence, compared to the Poisson case, the role of geometrically smooth and non-smooth error is completely interchanged: algebraically smooth error is actually highly oscillatory geometrically and algebraically non-smooth error is very smooth geometrically.

The computational results for this problem are given in Table 9, 10.

Table 9 Numerical results for L_h^{sd}

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	64×64	0.098	6	0.11	0.047	2.17	1.67
	128×128	0.090	6	0.60	0.118	2.18	1.67
II	64×64	0.097	6	0.11	0.063	2.17	1.67
	128×128	0.087	6	0.61	0.110	2.19	1.67
III	64×64	0.209	9	0.22	0.054	2.16	1.66
	128×128	0.343	13	0.60	0.102	2.18	1.67
IV	64×64	0.208	9	0.11	0.042	2.16	1.66
	128×128	0.306	14	0.49	0.102	2.18	1.67

Table 10 Numerical results for L_h^e

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	64×64	0.0098	3	0.06	0.037	2.16	1.66
	128×128	0.0097	3	0.60	0.167	2.18	1.67
II	64×64	0.0479	5	0.11	0.032	2.16	1.66
	128×128	0.0511	5	0.49	0.154	2.18	1.67
III	64×64	0.0530	5	0.11	0.032	2.16	1.66
	128×128	0.0568	5	0.44	0.166	2.18	1.67
IV	64×64	0.0096	3	0.11	0.057	2.16	1.66
	128×128	0.0093	3	0.44	0.167	2.18	1.67

Problem 7. Anisotropic problem on a unit square with Dirichlet boundary conditions.

The anisotropic equation: $-\varepsilon u_{xx} - u_{yy} = f$ plays an important role in practice, as many physical problems are highly anisotropic by nature. If we discretize using the standard 5-point different operator, we obtain the following difference stencil

$$L_h^{(5)}(\varepsilon) = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -\varepsilon & 2(1+\varepsilon) & -\varepsilon \\ & -1 & \end{bmatrix}_h,$$

where ε =0.1 and 0.01 is used. The same discrete operator is obtained if we discretize the pure Poisson operator by the standard 5-point difference operator on a stretched grid with mesh sizes $h_x = h_y/\sqrt{\varepsilon}$.

For any $\varepsilon > 0, L_h^{(5)}(\varepsilon)$ is elliptic, but not uniformly elliptic with respect to ε. Furthermore, its h-ellipticity measure $E_h(L_h^{(5)}(\varepsilon)) = \frac{\varepsilon}{2+\varepsilon} \to 0$ as $\varepsilon \to 0$. However, AMG methods apply the operator-based interpolation and coarsening which can adjust its coarsening process to the direction of only strong connectivity, that is, the direction of smoothness, and obtain better results.

The computational results for this problem are given in Table 11.

Table 11(1) Numerical results for $L_h^{(5)}(0.1)$

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	32×32	0.036	5	0.05	0.012	3.15	1.87
	48×48	0.036	5	0.11	0.032	3.47	1.87
II	32×32	0.036	5	0.00	0.012	3.21	1.87
	48×48	0.036	5	0.06	0.022	3.28	1.87
III	32×32	0.036	5	0.05	0.012	3.23	1.88
	48×48	0.036	5	0.06	0.010	3.28	1.87
IV	32×32	0.036	5	0.05	0.006	3.22	1.88
	48×48	0.036	5	0.11	0.022	3.27	1.87

Table 11(2) Numerical results for $L_h^{(5)}(0.01)$

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	48×48	0.024	4	0.05	0.028	2.75	1.96
	64×64	0.025	4	0.16	0.043	2.80	1.96
II	48×48	0.024	4	0.05	0.015	2.75	1.96
	64×64	0.025	4	0.11	0.043	2.80	1.96
III	48×48	0.024	4	0.06	0.028	2.75	1.96
	64×64	0.025	4	0.11	0.043	2.80	1.96
IV	48×48	0.024	4	0.01	0.015	2.75	1.96
	64×64	0.025	4	0.11	0.043	2.80	1.96

Problem 8. Biharmonic problem on a unit square.

Let

$$\Delta^2 u = 0, \text{ in } \Omega,$$
$$u = 0, \text{ on } \partial\Omega,$$
$$\frac{\partial u}{\partial n} = 0, \text{ on } \partial\Omega,$$

with the following 13-point finite difference stencil

$$\begin{bmatrix} & & 1 & & \\ & 2 & -8 & 2 & \\ 1 & -8 & 20 & -8 & 1 \\ & 2 & -8 & 2 & \\ & & 1 & & \end{bmatrix}.$$

The resulting coefficient matrix is symmetric and positive definite, but not weakly diagonally dominant. Furthermore, the matrix is very ill-conditioned with a condition number of $O(h^{-4})$. Therefore, this problem provides a good test case of the robustness and efficiency for various algorithms.

The computational results for this problem are given in Table 12.

Table 12 Numerical results for Biharmonic problem

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	32×32	0.788	58	0.11	0.013	2.13	1.65
	48×48	0.816	68	0.17	0.028	2.17	1.65
II	32×32	0.729	44	0.16	0.015	2.16	1.67
	48×48	0.701	39	0.22	0.032	2.19	1.66
III	32×32	0.787	58	0.17	0.011	2.15	1.66
	48×48	0.866	96	0.17	0.026	2.20	1.66
IV	32×32	0.656	33	0.11	0.013	2.15	1.66
	48×48	0.714	41	0.22	0.032	2.19	1.66

Problem 9. Poisson problem on a unit cube with Dirichlet boundary conditions.

For the three-dimensional problem, the 7-point difference approximation

$$\frac{1}{h^2}(6u_{i,j,k} - u_{i+1,j,k} - u_{i-1,j,k} - u_{i,j+1,k} - u_{i,j-1,k} - u_{i,j,k+1} - u_{i,j,k-1}) = f_{i,j,k}$$

is applied.

The computational results for the problem are given in Table 13.

Table 13 Numerical results for 3D problem

method	EQ	ρ	N	t_p	t_I	σ^A	σ^Ω
I	$8 \times 8 \times 8$	0.013	4	0.06	cta0	2.46	1.61
	$16 \times 16 \times 16$	0.016	4	0.33	0.070	2.63	1.60
II	$8 \times 8 \times 8$	0.039	5	0.05	0.022	2.40	1.59
	$16 \times 16 \times 16$	0.059	5	0.27	0.044	2.64	1.60
III	$8 \times 8 \times 8$	0.044	5	0.06	0.022	2.40	1.59
	$16 \times 16 \times 16$	0.061	5	0.11	0.010	2.64	1.60
IV	$8 \times 8 \times 8$	0.009	3	0.05	0.037	2.46	1.61
	$16 \times 16 \times 16$	0.017	4	0.22	0.043	2.64	1.60

cta0: CPU-time approximates 0.

6. Conclusions

In this paper we give three new Gauss-Seidel-type interpolation formulae (3.4)-(3.9). We have shown that the new interpolation operators have their own advantages compared with the basic interpolation operator (2.5)-(2.6) proposed in [8]. For each variable $i \in F^m$, our interpolation formulae (3.4)-(3.9) still choose C_i^m as the basic interpolatory set, and thus the resulting Galerkin operators do not substantially increase towards coarser levels.

Various numerical experiments are reported including the Poisson equation, an anisotropic equation, the biharmonic equation, and even a 3D problem. Numerical results demonstrate that Gauss-Seidel-type multigrid methods not only can accelerate convergence, but they can also reduce computing time, in particular, for the setup phase. Furthermore, the number of iterations needed to achieve a fixed accuracy is less than for Method I. This contributes to decreasing the overall computing time of the AMG method.

It should especially be pointed out that Method IV can be efficiently applied to all the above problems and thus is a very robust and powerful MG method.

Acknowledgment. Correction of the English text by Prof. Peter Monk is gratefully acknowledged.

References

[1] A. Brandt, S. F. McCormick, and J. Ruge, Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations, Institute for Computational studies: Technical Report, Colorado State University, Fort Collins, Colorado, 1982.

[2] A. Brandt, Multigrid techniques: 1984 Guide with applications to fluid dynamics, Arbeitspapiere der GMD85, 1984.

[3] R. H. Chan, Q. S. Chang and H. W. Sun, Multigrid method for ill-conditioned symmetric toeplitz systems, *SIAM J. Sci. Comput.*, **19**:2 (1998), 516-529.

[4] Q. S. Chang and Z. H. Huang, Efficient algebraic multigrid algorithms and their convergence, *SIAM J. Sci. Comput.,* **24**:2 (2002), 597-618.

[5] Q. S. Chang, S. Q. Ma and G. Y. Lei, Algebraic multigrid for queuing networks, *Intern. J. Computer Math.*, **70** (1999), 539-552.

[6] Q. S. Chang and Y. S. Wong, Recent Developments in Algebraic Multigrid Methods, Proceedings of Copper Mountain Conference on Iterative Methods, Colorado, 1(1992).

[7] Q. S. Chang, Y. S. Wong and H. Fu, Algebraic Multigrid and Its Application to Euler Equations, Proceedings of Second International Conference on Computational Physics, World Scientific, Singapore, 1993.

[8] Q. S. Chang, Y. S. Wong and H. Fu, On the algebraic multigrid methods, *Journal of Computational Physics*, **125** (1996), 279-292.

[9] Z. H. Huang and Q. S. Chang, Multigrid solver based on defect equation, *Chinese J. Comput. Phys.*, **18** (2001), 423-428.

[10] Z. H. Huang and Q. S. Chang, The improved algorithm of multigrid interpolation operator, Acta Mathematicae Applicatae Sinica, to appear.

[11] W. Z. Huang, Convergence of algebraic multigrid methods for symmetric positive definite matrices with weak diagonal dominance, *Appl. Math. Comp.*, **46** (1991), 145-164.

[12] J. Ruge and K. Stuben, Efficient solution of finite difference and finite element equations, in: D.J.Paddon and H.Holstein(eds.), Multigrid methods for Integral and Differential Equations, 1985.

[13] J. Ruge and K. Stuben, Algebraic Multigrid, in: Multigrid Methods (S.F.McCormick, Ed.), SIAM, Philadelphia, 4(1987).

[14] K. Stueben and U. Trottenberg, Multigrid methods: Fundamental Algorithms, Model Problem Analysis and Application, Lecture Note in Mathematics, Springer-Verlag, New York/Berlin, 960(1982).