

Research on Two-point Aggregated Algebraic Multigrid Preconditioning Methods

Jian-Ping Wu, Fu-Kang Yin, Jun Peng, Jin-hui Yang

Academy of Ocean Science and Engineering,
National University of Defense Technology,
Changsha, China

E-mail: wjp@nudt.edu.cn, yinfukang@nudt.edu.cn, pengjun@nudt.edu.cn, yangjinhui@nudt.edu.cn

Abstract-Among the methods to solve sparse linear systems, the aggregation based algebraic multigrid is widely focused on, due to the potential of approximately optimal convergence and the cheap cost to setup. And it is often used as the preconditioner to Krylov subspace methods. But the actual efficiency is restricted to the extent of compliment of two processes, the smoothing and the correction. In the aggregation based algebraic multigrid methods, it is one of the critical steps to aggregate the finer-grid points to construct coarser grids. In this paper, several two-point aggregations are given first, and then for model problems, three characteristics are compared and analyzed, including the spectral radius of the related iterative matrix, the similarity between the spectra of the coefficient matrix on the coarser grid and that on the finer grid, the number of preconditioned conjugate gradient iterations with algebraic multigrid as the preconditioner. The results show that the spectral radius and the similarity of spectra have potentials to direct the design of highly efficient algebraic multigrid methods. Meanwhile, during the aggregation, it is better to match the fine points with heavy edge weights than with weak ones. When Jacobi iteration is used as smoothing, it is difficult to determine the best strategy to aggregate. But if Gauss-Seidel iteration is used as smoothing, the number of iterations is significantly reduced, and the optimal aggregation strategy is more certain, that is, the optimal is always among the three ones with matching priority for edges with heavy weights, and the number of iterations with the other two is very close to that with the optimal.

Keywords-sparse linear algebraic equations; aggregation based algebraic multigrid; smoothing process; preconditioner; Krylov subspace method

I. INTRODUCTION

Multigrid has the potential optimal convergence, and is one of the efficient ways to solve sparse linear systems. It can be used solely, and can be used as the preconditioner of the so-called Krylov subspace iterations. The actual applications show that the latter is more robust, and then it is used more widely. When multigrid methods are used, a series of sparse linear systems with smaller and smaller scales are constructed from the original ones, and the residual vector related to the approximate solution on the finer grid is transferred to the coarser grid. The solution on the coarser grid is transferred back to the finer to correct the approximation [1].

The effectiveness of multigrid methods is originated from the extent of the complement of two processes, smoothing and correction. The smoothing process can reduce the

components of the error with higher frequencies, and the rest with lower frequencies, which cannot be effectively reduced by smoothing, are transferred to the coarser grid, where some components will have relatively higher frequencies, and again reduced significantly by smoothing on that level [2]. Therefore, the effectiveness of multigrid methods is influenced by two factors, the reduction of the error components with high frequencies by smoothing, and the approximation of the error components with lower frequencies by the transfer operator. The reduction of the error components with high frequencies are characterized by the convergence rate of the smoothing operator. And therefore, the faster the convergence of this operator is, the more effective the method will be. For approximation to the error components with lower frequencies, the accuracy is influenced by the accuracy of the correction on coarser grid, and the accuracy of the transfer operators. The more accurate the correction on the coarser grid is, and the more accurate the transfer operator is, the more effective the method will be.

For general sparse linear equations, discrete grid is unknown, irregular, or even does not exist. For these cases, algebraic multigrid method is the favorite choice. When this kind of multigrid is used, the grid is described with the adjacent graph of the coefficient matrix. The grid transfer operators and the coarser grid operators are defined from the elements of the coefficient matrices only [3]. Among these methods, the aggregation based ones are widely focused on, due to their cheap cost in the setup phase, and easiness to control the cost of computation and storage [4]. In these methods, a point on the coarser grid is corresponding to several points on the finer grid. The finer points related to a common coarser point is called an aggregation. The interpolation from the coarser points to the finer points in the corresponding aggregation is used to define the grid transfer operators, and the coarser grid operator is compounded from the grid transfer and the finer grid operator. Therefore, the selection of the aggregations is one of the core issues.

In general, the selection of aggregation is based the adjacent graph [4]. Through the judgment of the relative scale of the edge weights between nodes, the specific nodes in each aggregation can be determined. For two-dimensional second order elliptic partial differential equation, Braess introduced two methods to construct aggregation [5]. The first is to directly fit one of the seven given structures with priorities. The second is to match nodes according to the edge weights first, and then to construct aggregations with similar ways as the first. Therefore, each aggregation

contains four or three nodes. The experiments show that the two methods have no significant differences in performance. These methods have been extended to three dimension problems further.

Okamoto et al. provided a global coarsening algorithm to construct aggregations [6]. In this algorithm, the edges are colored first, and those independent with each other are extracted to form a subset. Then based on the restriction of the ratio of volume to area, the control volumes related to each edge of the subset are combined, and aggregations with global sub-optimal shape are formed automatically.

For sparse linear equations derived from advection-diffusion equation, Kim et al. constructed aggregations based on matching of points pair by pair in the adjacent graph. Then the aggregation scheme is combined with the Gauss-Seidel smoothing, to construct aggregation based algebraic multigrid methods [7]. The two-point aggregation method often leads to too many levels. To relieve this problem, D'Ambra et al. apply this kind of method several times a time to construct the aggregations [8]. Notay apply the two-point aggregation twice once a time to construct the aggregations on the coarser levels [4][9], therefore the number of grid-points on the coarser level is reduced to about 1/4 of that on the finer level.

Kumar provided another scheme based on graph partitioning, with each sub-graph related to an aggregation [10]. He adopted ILU(0) as the smoother, and approximately solved the coarser grid problem with ILUT. The experiments show that, the robustness of the method is superior to the traditional aggregation-based algebraic multigrid.

In this paper, there provide several two-point aggregation methods first. Then, the methods are compared and analyzed when combined with Jacobi and Gauss-Seidel smoothing, from the spectral radius of the iterative matrix, and the similarity between the spectra of the coarser operator and that of the finer one. The results show that the spectral radius and the similarity of spectra are meaningful in designing efficient multigrids. When the more efficient smoothing process is used, the multigrid is more efficient and the multigrid has more certainty in optimal aggregation selection.

II. CONJUGATE GRADIENT WITH AGGREGATION-BASED ALGEBRAIC MULTIGRID AS THE PRECONDITIONER

Consider the solution of a symmetric positive definite sparse linear system of order n

$$A x = b, \quad (1)$$

where A is a known symmetric positive definite sparse matrix, b is a known right-hand vector, and x is the unknown solution vector. The preconditioned conjugate gradient (PCG) method to solve system (1) can be described as algorithm 9.1 in reference [11], where M^{-1} is the preconditioner. The more accurate M^{-1} approaches A^{-1} , the more rapid the algorithm is. In addition, the preconditioner M^{-1} should be symmetric, to ensure the high efficiency of the algorithm.

In this paper, the preconditioner M^{-1} is selected as the algebraic multigrid. The multigrid method (MGM) can be described as algorithm 2.3.2 in reference [12], which is defined recursively. We adopt a version with minor varies to the smoother. The pre- and the post-smoothing operator can be different and denoted as S_i^R and S_i^L respectively. In MGM here, the V cycle is adopted, and the pre- and the post-smoothing iterations is only 1. Then, the approximate solution is smoothed with the smoother S_i^R first. If the derived approximation with S_i^R is denoted as $x^{(l)}$, the residual vector $r^{(l)} = b^{(l)} - A_l x^{(l)}$ is mapped to the coarser grid by the operator $R_{l-1,l}$. On the $(l-1)$ -th grid, MGM is invoked again to derive the correction vector. And then it is mapped back to the l -th grid, to correct the approximation $x^{(l)}$. If the number of levels attains the threshold, or the sparse linear system is small enough, the recursive process is stopped and the correction on the coarsest level is solved with a certain method.

In MGM, if the smoother S is defined as

$$x := S(x, b), \quad x := (I - SA)x + Sb, \quad (2)$$

and $m=2$, that is, two-levels are used, MGM is equivalent to the sequence of the following iterations

$$\begin{aligned} e^{(l)} &:= (I - S_i^R A_i) e^{(l)}, \\ d^{(l-1)} &:= -R_{l-1,l} A_i e^{(l)}, \\ v^{(l-1)} &:= (R_{l-1,l} A_i P_{l-1,l})^{-1} d^{(l-1)}, \\ e^{(l)} &:= e^{(l)} + P_{l-1,l} v^{(l-1)}, \\ e^{(l)} &:= (I - S_i^L A_i) e^{(l)}, \end{aligned}$$

where $e^{(l)} = x^{(l)} - x$ is the error. Therefore, it is equivalent to

$$e^{(l)} := B_l e^{(l)}, \quad (3)$$

$$B_l = (I - S_i^L A_i) \{I - P_{l-1,l} (R_{l-1,l} A_i P_{l-1,l})^{-1} R_{l-1,l} A_i\} (I - S_i^R A_i). \quad (4)$$

Obviously, to ensure that MGM can be used as a preconditioner of PCG, the iterative matrix B_l should be symmetric. Therefore, we can select $R_{l-1,l} = (P_{l-1,l})^T$, and $S_i^L = (S_i^R)^T$. Then, to apply MGM, the remaining work is to determine the coarser grids, the prolong operator $P_{l-1,l}$ from coarser grid to finer, the coefficient matrix of the linear system on the coarser level A_{l-1} , and the smoother S_i^R on the finer grid.

In aggregation based algebraic multigrid methods, the coarser grids are constructed from aggregating the points on the finer grid. Here, two-point aggregation algorithms are focused on, and details will be given in section 3. The basic scheme to construct prolong operators is considered here only. If there are n_l points on the l -th grid, the prolong operator from the $(l-1)$ -th grid to the l -th grid is an $n_l \times n_{l-1}$ matrix. If some point i in the l -th grid belongs to the j -th aggregation, it is mapped to the j -th point in the $(l-1)$ -th level, and the element of $P_{l-1,l}$ on the i -th row and the j -th column is 1. All elements not equal to 1 are zeros.

The smoother can be derived from any iteration with cheaper cost. For Jacobi iteration, there have

$$x^{(k+1)} := (I - D^{-1}A)x^{(k)} + D^{-1}b, \quad (5)$$

where D is a diagonal matrix with each diagonal element equal to the corresponding element of A . Compare (5) with (2), it is known that (5) is equivalent to (2) with $S = D^{-1}$. When A is symmetric, S is symmetric too. Therefore, we can select $S_l^R = D_l^{-1}$ and $S_l^L = D_l^{-1}$.

For Gauss-Seidel, there have

$$x^{(k+1)} := (I - (D-L)^{-1}A)x^{(k)} + (D-L)^{-1}b, \quad (6)$$

where L is a matrix, where the strictly lower triangular part is the same as that of A , and other elements are zeros. Thus, when applied as the smoother of MGM, we can select $S_l^R = (D_l - L_l)^{-1}$ and $S_l^L = (D_l - L_l)^T$.

When performing multigrid preconditioned iterations with PCG to solve linear systems, on each iteration, the grid hierarchy, A_i , $P_{i,l-1}$, and S_i^R are all not changed. Therefore, the information for constructing them can be done during the setup, and be used directly in the latter iterations.

III. TWO-POINT AGGREGATION ALGORITHMS

From the point of graph theory, even if only two points are selected each time, the strategies are very fruit. In this section, we only describe seven of them. These strategies are analyzed and studied in the next section by comparison, based on the performance of the corresponding aggregation based algebraic multigrid methods.

Traditionally, when aggregation is performed to the adjacent graph, the process begins from the node labeled 1 in general, and then the nodes are considered in increasing order. If an un-aggregated node is found, we can search other un-aggregated nodes among its neighbors. If there are such nodes, we can select one to aggregate. If there are no such nodes, it is itself an aggregation alone. This process is continued, until all nodes are aggregated. And then, each node is mapped to a certain aggregation. If the weight of an edge is defined as the magnitude of the element with row and column labels related to its end points, we can match the nodes accordingly. When the edge with maximum weights is select in the matching, the aggregation is denoted as *m1_strong* here. When the edge with minimum weights is select, it is denoted as *m1_weak*. And when the node with minimal label is selected to match, it is denoted as *m1_serial*.

Four other strategies for aggregation are considered here. The first two are based the relative magnitude of the edge weights. The diagonal elements of the coefficient matrix are scaled to ones in advance, and then the off-diagonals are sorted in magnitude. The off-diagonal elements with end points un-aggregated are selected one by one to aggregate. Each of the remaining un-aggregated nodes is mapped to an aggregation alone. In this kind of strategy, when the off-diagonals are ordered in decreasing order, the strategy is denoted as *m2_desend*. And when the off-diagonals are

ordered in ascending order, the strategy is denoted as *m2_asend*.

In the other two strategies, the diagonal elements of the coefficient matrix are scaled to ones in advance too. But then the off-diagonal with extreme magnitude is selected and the end points of the related edge are aggregated. After that, all un-aggregated nodes adjacent to aggregated ones are recorded and form a set I . Then the edges from each of I to any other un-aggregated nodes are find out, and that with extreme magnitude is selected. The end points of the selected edge are aggregated. Each of the remaining un-aggregated nodes is mapped to an aggregation alone. If the extreme value is selected as the maximum, the strategy is denoted as *m3_max*. If selected as the minimum, the strategy is denoted as *m3_min*.

IV. NUMERICAL EXPERIMENTS

In this paper, all the experiments are done on a processor of Intel(R) Xeon(R) CPU E5-2692 v2@2.20 GHz(cache 30720KB). The operating system is Red Hat Linux 2.6.32-220-TH and the compiler is Intel FORTRAN Version 11.1. For the coefficient matrices of the derived linear systems are symmetric positive definite in all experiments, PCG is used. The initial approximation solution is selected as the zero vector and the stop criterion is selected as $\varepsilon = 1E-10$. When the multigrid method is used as the preconditioner, the linear system on the coarsest level is solved with ILU(0) preconditioned conjugate gradient and the stop criterion is selected as $\varepsilon = 1E-10$ too.

When the aggregation strategies described in section 3 are used, it can be expected that the time elapsed for each iteration is linearly dependent on the non-zeros of the coefficient matrix on each level, and the number of iterations. Thus, these parameters are used to characterize the efficiency of the algorithm in all experiments. In all the tables and figures, Nl denotes the number of levels of the multigrid, Cg denote the complexity of the grids, that is, the ratio of the total number of grid points on each level to that on the finest level. Co denotes the complexity of the operators, that is, the ratio of the total number of non-zeros in the coefficient matrices on each level to that on the finest level. It_JA and it_GS denote the number of iterations used with Jacobi and Gauss-Seidel smoothing respectively, and DOF denotes the degree of freedom, that is, the order of the original sparse linear system.

Experiments are done for four sparse linear systems, namely matrix51, matrix52, matrix53 and matrix92, which are derived from two dimensional PDE problem with Dirichlet boundaries

$$-\frac{\partial}{\partial x} \left(\rho \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left(\rho \frac{\partial u}{\partial y} \right) + \delta u = f. \quad (7)$$

The definition region is $(0,c) \times (0,c)$ and the function f and the boundary values are given from a known true solution $u=1$. All the linear systems are derived with finite difference method. There are $n+2$ points in each dimension and the

value $z(x_i, y_j)$ is defined as z_{ij} for any continuous function z , where

$$x_i = ih, x_j = jh, i, j = 0, 1, \dots, n+1,$$

and $h=c/(n+1)$. For the first three linear systems, the following discrete form is used:

$$-\rho_{i,j+1/2}u_{i,j+1} - \rho_{i+1/2,j}u_{i+1,j} - \rho_{i,j-1/2}u_{i,j-1} - \rho_{i-1/2,j}u_{i-1,j} + \lambda_{i,j}u_{i,j} = h^2 f_{i,j},$$

$$\lambda_{i,j} = \delta_{i,j}h^2 + \rho_{i,j+1/2} + \rho_{i+1/2,j} + \rho_{i,j-1/2} + \rho_{i-1/2,j}.$$

For linear systems matrix51 and matrix92, $\rho=1$, $\delta=0$, $c=1$ and for matrix92, the discrete form is

$$-u_{i-1,j-1} - 4u_{i,j-1} - u_{i+1,j-1} - 4u_{i,j+1} + 20u_{i,j} - 4u_{i+1,j} - u_{i-1,j+1} - 4u_{i,j+1} - u_{i+1,j+1} = h^2 f_{i,j}.$$

In linear systems matrix52 and matrix53,

$$\rho = \begin{cases} \rho_1, & x \in (2, 2.1] \ \& \ y \in (1, 2.1], \\ \rho_1, & y \in (2, 2.1] \ \& \ x \in (1, 2.1], \\ \rho_2, & x \in (1, 2) \ \& \ y \in (1, 2), \\ \rho_3, & x \in [0, 1) \ \& \ y \in [0, 2.1], \\ \rho_3, & y \in [0, 1) \ \& \ x \in [0, 2.1], \end{cases}$$

$$\delta = \begin{cases} \delta_1, & x \in (2, 2.1] \ \& \ y \in (1, 2.1], \\ \delta_1, & y \in (2, 2.1] \ \& \ x \in (1, 2.1], \\ \delta_2, & x \in (1, 2) \ \& \ y \in (1, 2), \\ \delta_3, & x \in [0, 1) \ \& \ y \in [0, 2.1], \\ \delta_3, & y \in [0, 1) \ \& \ x \in [0, 2.1]. \end{cases}$$

In linear system matrix52,

$$\rho_1=1, \rho_2=2 \times 10^3, \rho_3=3 \times 10^5, \delta_1=0.02, \delta_2=3, \delta_3=500.$$

In linear system matrix53,

$$\rho_1=x, \rho_2=2 \times 10^3 x, \rho_3=3 \times 10^5 x, \delta_1=0.02x, \delta_2=3x, \delta_3=500x.$$

To compare the effectiveness of the seven two-point aggregation strategies, three characteristics are considered here, including the spectral radius of the iterative matrix, the similarity between the spectra of the coefficient matrix on the coarser level and that on the finer level, and the actual number of iterations used, where the first two are evaluated for two-level grids.

For the four linear systems with $n=32$, the spectral radius of the iterative matrix B for two-level grids are given in table 1 and table 2, where Jacobi and Gauss-Seidel iteration is used as the smoother in table 1 and table 2 respectively. From table 1 and table 2, it is clear that for m1_serial and m1_strong, the spectral radius are nearly the same, while the spectral radius is relatively larger in general when m1_weak is used. The spectral radius with m2_desend is smaller than that with m2_asend in general. And for m3_max and m3_min, the spectral radius with the former is smaller in general. Meanwhile, when Jacobi smoothing is replaced by Gauss-Seidel, for matrix51, matrix52 and matrix53, the spectral radius is significantly reduced. For matrix92, although the spectral radius is reduced too, the ratio is much moderate.

From the point of spectral radius, among the seven strategies, when Jacobi smoothing is used, m1_serial, m1_strong and m3_max are better in general. But for

matrix53, m2_asend is the best. When Gauss-Seidel smoothing is used, the results are similar, but the spectral radius with m2_desend is the smallest, and for matrix53, the best choice is m3_max.

TABLE I. TABLE THE SPECTRAL RADIUS OF B WITH N=32 AND JACOBI SMOOTHING

	matrix51	matrix52	matrix53	matrix92
m1_serial	0.9870	0.9757	0.9731	0.4554
m1_strong	0.9870	0.9757	0.9731	0.4554
m1_weak	0.9870	0.9757	0.9789	0.5062
m2_asend	0.9871	0.9745	0.9340	0.5965
m2_desend	0.9871	0.9830	0.9800	0.5249
m3_max	0.9870	0.9753	0.9804	0.4554
m3_min	0.9870	0.9760	0.9789	0.5062

TABLE II. TABLE THE SPECTRAL RADIUS OF B WITH N=32 AND GAUSS-SEIDEL SMOOTHING

	matrix51	matrix52	matrix53	matrix92
m1_serial	0.4274	0.5280	0.5271	0.4203
m1_strong	0.4274	0.5280	0.5271	0.4203
m1_weak	0.4274	0.5280	0.5308	0.4821
m2_asend	0.4220	0.5418	0.5532	0.5523
m2_desend	0.4220	0.4206	0.4754	0.4168
m3_max	0.4274	0.4671	0.4718	0.4203
m3_min	0.4274	0.5366	0.5370	0.4821

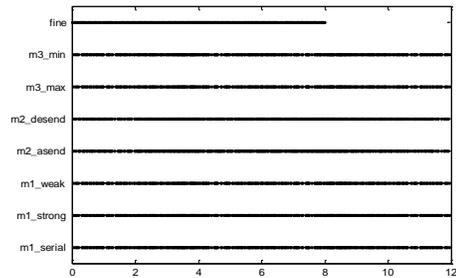


Figure 1. The spectra of matrix51 when $n=32$.

From figure 1 to figure 4, it can be seen that, the spectra of the coefficient matrix on the coarser grid is wider than that on the finer, that is, the spectra differs significantly for coarser and finer grids, which may be one of the main reasons for the low efficiency of this kind of multigrid methods. Secondly, as for the spectra of the coefficient matrices, for any one of the four linear systems, the results with m1_strong, m2_desend and m3_max are very close. Meanwhile, for matrix51, the results with the seven strategies have no significant difference. For matrix52, the other four strategies have similar results to each other, but are inferior to m1_strong, m2_desend or m3_max. For matrix53, except the best three strategies, m2_asend and

m1_serial is the second. For matrix92, m1_serial is comparable to the three best ones.

The effectiveness of multigrid methods originates from the complement of the smoothing process and the correction. The effectiveness of the correction is dependent on the similarity between the spectra of the coarser operator and that of the finer. We consider the two-level grids here. The spectra of the coefficient matrices on the finer and the coarser grid are given in figure 1 for the four linear systems, with one sub-graph for each. In each sub-graph, the uppermost line describes the spectra of the coefficient matrix on the finer level, and those on the coarser level with different aggregation strategies are given below.

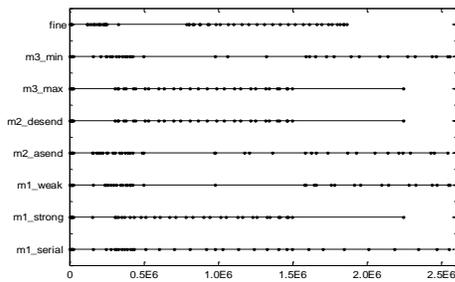


Figure 2. The spectra of matrix52 when n=32.

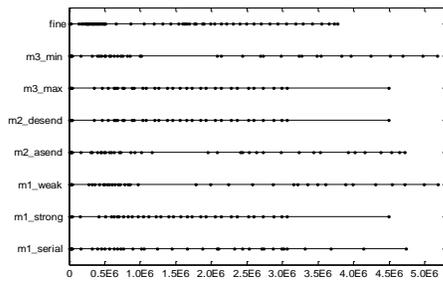


Figure 3. The spectra of matrix53 when n=32.

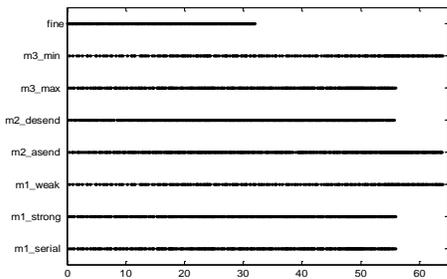


Figure 4. The spectra of matrix92 when n=32.

Now we investigate the performance of the actual preconditioned conjugate gradient iterations, with results given in table 3 to table 10. It is clear from the tables that, in general, the number of iterations is relatively less when m1_strong, m2_desend or m3_max is used. As for the number of iterations, among m1_serial, m1_strong and m1_weak, m1_strong is the best. Among m2_asend and

m2_desend, m2_desend is better. Among m3_max and m3_min, m3_max is better in general. These conclusions are fit with those of spectral radius of the iterative matrix and the similarity of spectra.

TABLE III. TABLE RESULTS FROM PCG WITH N=512 FOR MATRIX51

parameter	Nl	Cg	Co	It_JA	It_GS
m1_serial	14	2.00	1.99	319	132
m1_strong	14	2.00	2.00	322	62
m1_weak	14	2.00	1.99	319	132
m2_asend	21	2.04	2.10	684	159
m2_desend	16	2.10	2.24	651	73
m3_max	14	2.00	2.00	322	62
m3_min	14	2.00	1.99	319	132

TABLE IV. TABLE RESULTS FROM PCG WITH N=512 FOR MATRIX52

parameter	Nl	Cg	Co	It_JA	It_GS
m1_serial	14	2.00	1.99	415	148
m1_strong	14	2.00	2.03	283	64
m1_weak	15	2.00	2.00	365	138
m2_asend	48	2.14	2.28	421	137
m2_desend	15	2.06	2.17	343	63
m3_max	15	2.01	2.15	368	66
m3_min	48	2.03	2.10	488	136

But there are exceptions too. For example, as in table 3, for matrix51, when Jacobi smoothing is used, the number of iterations with m2_desend is significantly larger. For matrix53, as in table 7, when Jacobi smoothing is used, the number of iterations with m2_asend is the least instead, which is opposite to the other cases. In addition, we can see that, for matrix51, matrix52 and matrix53, when the aggregation strategy is given, the number of iteration with Gauss-Seidel smoothing is significantly less than that with Jacobi. For matrix92, though the results are similar, the extent of improvement is weakened. Meanwhile, it can be seen that m1_strong is superior to m1_serial and m1_weak, m2_desend is superior to m2_asend, and m3_max is superior to m3_min, which shows again that, it is superior to select the edge with heaviest weight.

As for the number of iterations, among the seven aggregation strategies, when Jacobi smoothing is used, the best strategy varies significantly for different problem. For matrix51, the best are m1_serial, m1_weak and m3_min. For matrix52 and matrix92, the best is m1_strong. For matrix53, the best is m2_asend. When Gauss-Seidel smoothing is used, the best strategy is more certain, which is m1_strong, or m2_desend, or m3_max. For matrix51, the best are m1_strong and m3_max. For matrix52, the best is

m2_desend. For matrix53, the best is m3_max. For matrix92, the best is m1_strong. Meanwhile, when Gauss-Seidel smoothing is used, the best aggregation strategy may vary between m1_strong, m2_desend, and m3_max, but the number of iterations with any one of them is not very larger than that with the best.

TABLE V. TABLE RESULTS FROM PCG WITH N=512 FOR MATRIX53

parameter	NI	Cg	Co	It_JA	It_GS
m1_serial	14	2.00	1.99	417	152
m1_strong	14	2.00	2.03	282	64
m1_weak	22	2.00	2.00	404	140
m2_asend	43	2.28	2.55	230	137
m2_desend	16	2.10	2.26	284	67
m3_max	14	2.00	2.01	314	58
m3_min	114	2.11	2.25	511	158

TABLE VI. TABLE RESULTS FROM PCG WITH N=512 FOR MATRIX92

parameter	NI	Cg	Co	It_JA	It_GS
m1_serial	14	2.00	1.99	142	128
m1_strong	14	2.00	2.00	62	62
m1_weak	14	2.00	2.32	162	147
m2_asend	35	2.05	2.39	245	192
m2_desend	15	2.03	1.98	79	71
m3_max	15	2.01	1.98	85	75
m3_min	16	2.02	2.55	245	190

In addition, from the comparison of the number of iterations with the results of spectral radius, it can be seen the strategy which leads to smaller radius can leads to less iterations in general too. But this is not true for all cases. For example, the radius with m1_serial and m1_strong are almost the same, but in most cases, the number of iterations with m1_strong is less than that with m1_serial. Similarly, from the comparison of the number of iterations with the spectra differences between coarser and finer grids, it can be seen that the less the difference is, the less the number of iterations is in general too. But it is not true for all cases too. For example, for matrix92, the spectra with m1_serial, m1_strong, m2_desend and m3_max are similar, but the number of iterations with m1_strong is significantly less than that with the other three, whenever Jacobi or Gauss-Seidel smoothing is used as the smoother.

V. CONCLUSION

In this paper, we provide four new two-point aggregation algorithm besides three known ones, and for sparse linear systems derived from two-dimensional partial differential

equation with Dirichlet boundaries, they are compared and analyzed systematically from three aspects, including the spectral radius of the iterative matrix, the similarity between the spectra of the coefficient matrix on the coarser level and that on the finer, and the actual number of iterations. The results show that the aggregation strategy which leads to less number of iterations varies in general among those with smaller spectral radius, and with maximum similarity between the spectra. At the same time, during the aggregation process, the strategy to select the edge with heaviest weight is superior to that with weakest. When Jacobi smoothing is used, it is hard to say which strategy is the best. But when Gauss-Seidel smoothing is used, the number of iterations is significantly less than that when Jacobi used, and the best aggregation strategy is more certain, which is always one of the three with the heaviest edge selected first. At this situation, the number of iterations with any one of these three strategies is not significantly different from each other.

ACKNOWLEDGMENT

This work is funded by NSFC(61379022).

REFERENCES

- [1] J. W. Ruge and K. Stuben, Algebraic multigrid, Multigrid Methods, Frontiers of Applied Mathematics, vol.3. SIAM Philadelphia, PA(1987), pp.73-130.
- [2] R. Wienands, W. Joppich, Practical Fourier analysis for multigrid methods, Taylor and Francis Inc., 2004
- [3] K. Stuben, "A review of algebraic multigrid", J. Comput. Appl. Math. And applied mathematics, 128(1-2), 2001, pp.281-309
- [4] Y. Notay, "Aggregation-based algebraic multilevel preconditioning", SIAM Journal On Matrix Analysis And Applications, 27:4(2006), pp.998-1018
- [5] D. Braess, "Towards algebraic multigrid for elliptic problems of second order", Computing, 55(1995), pp.379-393
- [6] N. Okamoto, K. Nakahashi, S. Obayashi, "A coarse grid generation algorithm for agglomeration multigrid method on unstructured grids", In Proceedings of 36th Aerospace Sciences Meeting and Exhibit , volume 98-0615, AIAA, Reno, 1998
- [7] H. Kim, J. Xu, and L. Zikatanov, "A multigrid method based on graph matching for convection-diffusion equations", Numer. Linear Algebra Appl., 10(2003), pp.181-195
- [8] P. D'Ambra, A. Buttari, D. di Serafino, S. Filippone, S. Gentile, B. Ucar, "A novel aggregation method based on graph matching for algebraic multigrid preconditioning of sparse linear systems", in: International Conference on Preconditioning Techniques for Scientific & Industrial Applications 2011, May 2011, Bordeaux, France
- [9] Y. Notay, "Aggregation-Based Algebraic Multigrid for Convection-Diffusion Equations", SIAM J. Sci. Comput., 2012, 34(4), pp.A2288-A2316
- [10] P. Kumar, "Aggregation based on graph matching and inexact coarse grid solve for algebraic two grid", International Journal of Computer Mathematics, 2011, 91(5):1061-1081
- [11] Y. Saad, Iterative methods for sparse linear systems, Boston: PWS Pub. Co., 1996
- [12] C. Wagner, Introduction to algebraic multigrid, Course Notes, University of Heidelberg, 1998/1999; available at: <http://www.iwr.uni-heidelberg.de/~Christian.Wagner/>