An efficient algebraic multigrid method for second order elliptic equations on polygonal domains

Ming Li

Department of Mathematics and Statistics, Honghe University, Yunnan, 661199, China

http://orcid.org/0000-0003-4183-2404, mathlm@126.com

Abstract: Based on a coarsening strategy of adjacency matrix, a new algebraic prolongation operator is developed for standard V-cycle multigrid method to accelerate the whole process. An efficient algebraic multigrid (EAMG) method is proposed for solving large-scale linear systems, arising from finite element (FE) discretization of second order elliptic boundary value problems. Numerical experiments on polygonal domains are conducted to demonstrate the EAMG computation is more efficient than standard method.

Keywords: algebraic multigrid, prolongation, finite element, polygonal domains

MSC classification: 65N12; 65N30; 65N55;

1 Introduction

Multigrid (MG) methods [2] are among the most efficient numerical methods for solving the large-scale linear systems arising from FE or finite difference (FD) discretization of partial differential equations, such as elliptic equation [7, 11, 12, 18, 19], convection diffusion equation [4, 5, 10, 13, 16], semilinear Poisson equation [9] and parabolic equation [6]. The MG could be categorized into two groups: Geometry multigrid and the algebraic multigrid (AMG) methods. Compared to the Geometry multigrid method, the algebraic multigrid (AMG) method able to the case with complex domain where grid may be highly unstructured or irregular [1, 15, 17].

In AMG, prolongation operator plays an important role to interpolate the solutions from the coarse to the fine grids. By using some geometric assumptions and strongly connected component, Chang, Wong and Fu [3] developed a prolongation operator. Kickinger discussed two simple prolongation operators based on two simple coarsening with graph of matrix [8]. By combining the linear and higher order FE basis functions, Shu et al. constructed a prolongation operator for AMG to solve the higher order FE equations [14].

In this paper, we focus on constructing a new prolongation operator, based on a coarsening strategy of adjacency matrix. Embedding this operator with standard V-cycle multigrid method, an efficient algebraic multigrid (EAMG) method is established to solve second order elliptic equations in polygonal domains. Numerical results show that the proposed method is more efficient than the ordinary algebraic multigrid method.

This paper is organized as follows. Section 2 gives the model problem. A new EAMG method coupled and a new prolongation operator are developed in the Section 3. Section 3.2 introduces a brief overview of the coarsening algorithms with graph matrix. Section 3.3 study a new algebraic prolongation operator from the residual equation. Section 4 presents the numerical results to show the efficiency and robustness of the proposed algorithm. Some concluding remarks are given in final section.

2 Model problem

In this paper, we consider the following elliptic boundary value problem

$$\begin{cases} -\nabla \cdot (\alpha(x,y)\nabla u) = f, & in \;\; \Omega, \\ u = 0, & on \;\; \partial\Omega, \end{cases} \tag{2.1}$$

where $u \in H_0^1(\Omega)$ is the exact solution, f is a given source function, and Ω is a 2D bounded polygonal domain with Dirichlet boundary $\partial\Omega$.

The equivalent variational form of the problem (2.1) is as follows: find $u \in H_0^1(\Omega)$ such that

$$a(u,v) = (f,v), \quad \forall v \in H_0^1(\Omega), \tag{2.2}$$

where

$$a(u,v) = \int_\Omega \alpha(x,y)\nabla u \cdot \nabla v, \quad f(v) = \int_\Omega fv. \tag{2.3}$$

In order to approximate the weak solution u, we consider a unstructured triangulation T_h of Ω with mesh-size $h = \max_{T\in T_h} diam(T)$.

Given the triangulation T_h for Ω, the linear finite element space with respect to the triangulation T_h is defined as

$$V_h = \{u \in C(\bar{\Omega}) | \; u \in P_1(T), \;\; \forall T \in T_h, u|_{\partial\Omega} = 0\}, \tag{2.4}$$

where $P_1(T)$ is a linear function on the element T. The finite element approximation of the variational problem (2.2) is: find $u^h \in V_h$ such that

$$a(u^h, v^h) = (f, v^h), \quad \forall v^h \in V_h. \tag{2.5}$$

The linear finite element equation could be rewritten as

$$A^h u^h = F^h, \tag{2.6}$$

here A^h is known as the stiffness matrix with entries $a^h_{i,j}$, and F^h is the right-hand vector.

3 New algebraic multigrid method

In this section, we focus on develop an efficient algebraic multigrid method for solving (2.6).

3.1 AMG setup phase

To start the process of AMG method, we need the setup phase, which could be described as follows Here, subscript j indicate level number, L denotes the number of grid levels, and 1 is the finest grid level.

From the Algorithm , all these components (include prolongation P^j_{j+1}, restriction R^j_{j+1} and hierarchy matrix A_j) could be constructed.

Algorithm 1 AMG setup phase

1: Input matrix A^h, set $A_1 \leftarrow A^h$
2: **for** $j \leftarrow 1, L-1$ **do**
3: Coarsening: partition Ω_j into disjoint sets C and M
4: Constructing prolongation matrix P^j_{j+1}
5: Taking restriction matrix: $R^{j+1}_j \Leftarrow (P^{j+1}_j)^T$
6: Computing coarse matrix $A_{j+1} \Leftarrow R^{j+1}_j A_j P^j_{j+1}$
7: $\Omega_{j+1} \leftarrow C$
8: **end for**

3.2 Coarsening strategy

In this subsection, we introduce a simple coarsening strategy based on the graph associated with the matrix, which will be employed in EAMG method, to obtain a set of coarse nodes. For convenience to description, we need the following definition.

Definition 1 *[8] Let $A = [a_{i,j}]_{n\times n}$ be a n-order square matrix. By the Graph $G(A)$ of the matrix A, we denote the following :*

$$G(A) = \{(i, j)|\ a_{i,j} \neq 0\}, \tag{3.7}$$

where i and j are called vertexes, and $(i, j) \in G(A)$ are called edges, which going from i to j.

Based on the Definition 1 and matrix graph $G(A)$, we can pick up coarse nodes from the following coarsening strategy [8] (Algorithm 2).

Algorithm 2 Coarsening algorithm

1: Input graph $G(A)$ corresponding to matrix A and build up two empty sets C and I.
2: Go through the vertices in $G(A)$ in ascending order.
3: (1) Let i be the actual vertex.
4: (2) If $i \in G(A)$ is not visited, put i into C and mark this node in $G(A)$ as **visited**.
5: (3) Go through the connected vertices of i and let j be the actual vertex with $(i, j) \in G(A)$.
6: If this vertex of $G(A)$ is not marked as **visited**, put (i, j) into I, and mark j in $G(A)$ as **visited**.
7: Return C and I.

If the set C has be obtained, call it as *coarse set*, then the other vertexes in graph $G(A)$ can be organized into *fine set* M in ascending order.

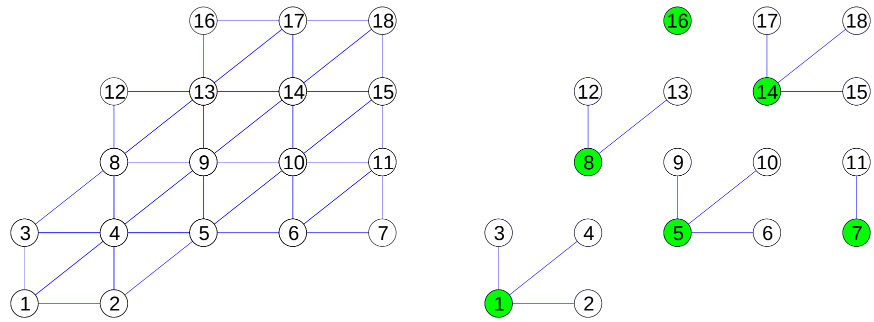
To illustrate how to select coarse vertexes in graph $G(A)$, we take a simple graph in Figure 1, as well as in Ref. [8]. It is shown that all vertexes $\{1, 2, \ldots, 18\}$ are split into two disjoint subsets C and M, correspond to the coarse points and fine points, respectively. Namely, $C = \{1, 5, 7, 8, 14, 16\}$ and $M = \{2, 3, 4, 6, 9, 10, 11, 12, 13, 15, 17, 18\}$.

3.3 Prolongation operator

Assume that the coarse set C and fine set M have been obtained. We introduce the following symbols

$$N_i = \{j|\ a_{ij} \neq 0, i \neq j\}, \qquad N_i^C = C \cap N_i,$$

$$N_i^M = M \cap N_i, \qquad s = ind(i), \quad i \in C,$$



(a) Graph $G(A)$ (b) Coarsening process

Figure 1: Generation of coarse set C

where N_i is the direct neighborhood of a point i, s is the renumbered i in C in ascending order. Let N denote the number of objects in the *coarse set* C, we now focus on developing an algebraic prolongation operator P_H^h (although physical meshes may not be given, we continue to denote fine-grid and coarse-grid quantities by h and H, respectively).

Consider the residual equation with the approximation v^0

$$A^h E^h = R^h, \tag{3.8}$$

where $E^h = u^h - v^0 = [e_1^h, e_2^h, \ldots, e_n^h]^T$, $R^h = F^h - A^h v^0$ are error and residual vector, respectively.

If approximation v^0 approach to the solution u^h, then the right-hand side $R^h \approx 0$. Therefore, we have

$$A^h E^h \approx 0. \tag{3.9}$$

The $i-$th component of (3.9) could be written in the following form

$$a_{ii} e_i^h + \sum_{j \in N_i} a_{ij} e_j^h \approx 0. \tag{3.10}$$

Assume that the coarse grid values $E^H = [e_1^H, e_2^H, \ldots, e_N^H]^T$ have been obtained. Some fine grid values could be constructed as below

$$e_i^h \Leftarrow e_s^H, \quad s = ind(i), \quad \forall\, i \in C. \tag{3.11}$$

Applying $N_i = N_i^C \cup N_i^M$ in the Eq. (3.10), we have

$$a_{ii} e_i^h + \sum_{t \in N_i^M} a_{it} e_t^h \approx - \sum_{j \in N_i^C} a_{ij} e_{ind(j)}^H. \tag{3.12}$$

Set

$$e_t^h :\approx e_i^h, \quad \forall\, t \in N_i^M, \tag{3.13}$$

then the (3.12) can be written as

$$(a_{ii} + \sum_{t \in N_i^M} a_{it}) e_i^h \approx - \sum_{j \in N_i^C} a_{ij} e_{ind(j)}^H. \tag{3.14}$$

Since A is a positive definite matrix, we get

$$(a_{ii} + \sum_{t\in N_i^M} a_{it}) \neq 0.$$

Therefore,

$$e_i^h \approx \frac{1}{a_{ii} + \sum\limits_{t\in N_i^M} a_{it}} \left(- \sum_{j\in N_i^C} a_{ij} e^H_{ind(j)} \right), \quad \forall\, i \in M. \tag{3.15}$$

we denote this procedure from E^H to e^h as an algebraic prolongation operator, and call it as P_H^h, i.e.

$$e^h \Leftarrow P_H^h e^H, \tag{3.16}$$

where

$$P_H^h = (p_{ij})_{n\times N} = \begin{cases} 1, & \text{if } \ i \in C,\, j = ind(i), \\ -a_{ik}/(a_{ii} + \sum\limits_{t\in N_i^M} a_{it}), & \text{if } \ i \in M, k \in N_i^C,\, j = ind(k), \\ 0, & \text{otherwise}, \end{cases} \tag{3.17}$$

Once the prolongation operator is specified, the restriction operator could be constructed as below

$$R_h^H \Leftarrow (P_H^h)^T. \tag{3.18}$$

3.4 Description of the EAMG algorithm

Applying the new prolongation operator for original algebraic multigrid, we shall establish an efficient algebraic multigrid (EAMG) method as below

Algorithm 3 EAMG

Input u_1^0, set $\bar{F}_1 \leftarrow F^h$
for $j \leftarrow 1, L-1$ **do**
    Pre-smoothing v_1 times: $u_j^{v_1} \Leftarrow \mathrm{CG}^{v_1}(A_j, u_j^0, \bar{F}_j)$ ▷ CG denotes the conjugate gradient method
    Restricting: $\bar{F}_{j+1} \Leftarrow R_{j+1}^j(\bar{F}_j - A_j u_j^{v_1})$
    if $j > 1$ **then**
        $u_j^0 \leftarrow 0$
    end if
end for
Solve the coarsest equation $A_L u_L = \bar{F}_L$ to get u_L^*
for $j \leftarrow L-1, 1$ **do**
    Correcting: $\bar{u}_j \Leftarrow u_j^{v_1} + P_{j+1}^j u_{j+1}^*$, ▷ P_{j+1}^j denotes the proposed prolongator
    Post-smoothing v_2 times: $u_j^* \Leftarrow \mathrm{CG}^{v_2}(A_j, \bar{u}_j, \bar{F}_j)$
end for
Return u_1^*

4 Numerical examples

In the section, we test the efficiency of the proposed EAMG method for solving elliptic equation of second order on some typical triangle meshes (see Figure 2 and 3). The AMG represent the ordinary AMG method (AMG) which embedded the coarsening strategy (namely Algorithm 2), conjugate gradient method and the prolongator [8]

$$\bar{P} = (\bar{p}_{ij})_{n\times N} = \begin{cases} 1, & \text{if } i \in C, \quad j = ind(i), \\ 1/m, & \text{if } i \in M, \; k \in N_i^C, \; j = ind(k), \; m = |N_i^C|, \\ 0, & \text{otherwise}. \end{cases} \tag{4.19}$$

Let the number of grids level $L = 4$, and set the number of pre-smoothing and post-smoothing steps as $v_1 = v_2 = 3$ in the two solvers. The zero vector is taken as the initial guess for AMG and EAMG solvers. The process of two computations are both start with zero initial guesses and are terminated when the 2-norm of residual vector on the finest grids is reduced by 10^{-5}. Our codes of the above algorithms are written in MATLAB and run on a desktop computer with 4GB RAM and Inter (R) Core (TM) i5-6200U CPU.

The $\|u - u_1^*\|_\infty$ and $\|u - u_1^*\|_\infty$ reported are the maximum absolute errors and residual errors over the entire discretized grid points on the finest grid, respectively. The $Time$ express the computational time (unit:second) of solvers.

Example 4.1 *The first test problem is*

$$-\Delta u(x, y) = 6xy(y^4 - 1) - 20x^3y^3(x^2 - 1) - 20x^3y(y^4 - 1), \quad in\ \Omega = [0, 1]^2, \tag{4.20}$$

with the Dirichlet boundary condition

$$u(0, y) = u(x, 0) = u(1, y) = u(x, 1) = 0. \tag{4.21}$$

The exact solution of Eq. (4.20) is

$$u(x, y) = x^3y(x^2 - 1)(y^4 - 1). \tag{4.22}$$

Example 4.2 *We consider the following second-order elliptic problem*

$$-\nabla \cdot (\alpha(x, y)\nabla u(x, y)) = f(x, y)$$

with variable coefficient $\alpha(x, y) = 2 + \sin(xy)$ on the square domain $\Omega = [0, 1]^2$. The function $f(x, y)$ and Dirichlet boundary conditions on $\partial\Omega$ are obtained from the analytic solution,

$$u(x, y) = (1 - \exp(\sin(\pi x)))(1 - \exp(\sin(\pi y))). \tag{4.23}$$

Example 4.3 *(L - shaped domain)We consider the model problem (2.1) with variable coefficient function*

$$\alpha(x, y) = x^2y^3 + 2,$$

on the L - shaped domain $\Omega = \Omega_1 \cup \Omega_2$, where $\Omega_1 = [0, 1] \times [-1, 1]$ and $\Omega_2 = [-1, 0] \times [-1, 0]$, which has analytic solution

$$u(x, y) = (\exp(\sin(xy)) - 1)(1 + \cos(\pi x))(x + 1)(\sin(\frac{\pi}{2}y) + 1)(1 - y). \tag{4.24}$$

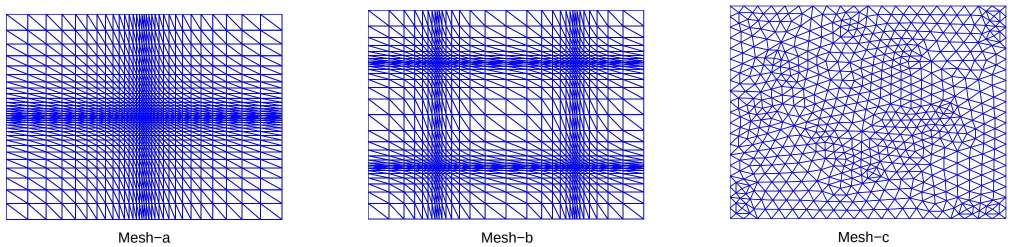


Figure 2: Some typical triangle meshes on square domain (Examples 4.1 and 4.2)

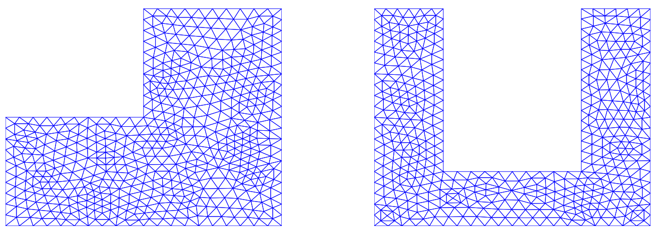


Figure 3: Unstructured triangle meshes on reentrant domain (Examples 4.3 and 4.4)

The corresponding right hand side function $f(x, y)$ and Dirichlet boundary conditions on $\partial\Omega$ are set to satisfy the analytic solution.

Example 4.4 *(U- shaped domain)We consider the model problem (2.1) with variable coefficient function*

$$\alpha(x, y) = exp(xy) + 1$$

on the U - shaped domain $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$, where $\Omega_1 = [-1, -0.5] \times [-0.5, 1]$, $\Omega_2 = [-1, 1] \times [-1, -0.5]$ and $\Omega_3 = [0.5, 1] \times [-0.5, 1]$, which has analytic solution

$$u(x, y) = \cos(\frac{\pi}{2}y^2)\sin(\pi x^3)(\exp(y + 0.5) - 1)(x^2 - 0.25). \tag{4.25}$$

The corresponding right hand side function $f(x, y)$ and Dirichlet boundary conditions on $\partial\Omega$ are set to satisfy the analytic solution.

Table 1: Hierarchy information of EAMG method applied to Example 4.2 with Mesh-c

A_j	Number of rows	Number of nonzeros	Density (%,full)	Minimum entries per row	Maximum entries per row	Average entries per row	Condition number
A_1	159105	1111151	0.004%	4	9	7.0	1.46(+5)
A_2	39617	276015	0.02%	4	9	7.0	3.64(+4)
A_3	9825	68111	0.07%	4	9	6.9	9.03(+3)
A_4	2417	16575	0.28%	4	9	6.9	2.22(+3)
A_5	585	3911	1.14%	4	9	6.7	4.92(+2)

Table 2: Hierarchy information of EAMG method applied to Example 4.3

A_j	Number of rows	Number of nonzeros	Density (%,full)	Minimum entries per row	Maximum entries per row	Average entries per row	Condition number
A_1	131457	917621	0.01%	4	9	7.0	7.62(+4)
A_2	32705	227637	0.02%	4	9	7.0	1.86(+4)
A_3	8097	56021	0.09%	4	9	6.9	4.47(+3)
A_4	1985	13557	0.34%	4	9	6.8	1.05(+3)
A_5	477	3161	1.39%	4	9	6.6	2.37(+2)

Table 3: Properties of hierarchy matrix A_j (Hierarchy information of EAMG method applied to Example 4.4

A_j	Number of rows	Number of nonzeros	Density (%,full)	Minimum entries per row	Maximum entries per row	Average entries per row	Condition number
A_1	111217	774849	0.01%	4	9	7.0	3.68(+4)
A_2	27577	191193	0.03%	4	9	6.9	9.04(+3)
A_3	6781	46533	0.10%	4	9	6.9	2.19(+3)
A_4	1639	10995	0.41%	4	9	6.7	5.16(+2)
A_5	382	2424	1.66%	4	9	6.3	1.10(+2)

Table 4: Numerical results of for Example 4.1 with Mesh-a

Cells	*Methods*	$\|u-u_1^*\|_\infty$	$\|F^h-A^hu_1^*\|_2$	*Iterations*	*Time*
8192	AMG	2.50(-4)	7.06(-6)	18	0.13
	EAMG	2.51(-4)	9.46(-6)	4	0.05
32768	AMG	6.26(-5)	8.81(-6)	20	0.47
	EAMG	6.23(-5)	9.19(-7)	5	0.13
131072	AMG	1.56(-5)	8.45(-6)	22	3.81
	EAMG	1.59(-5)	9.56(-6)	4	0.59
524288	AMG	8.45(-6)	8.61(-6)	23	15.8
	EAMG	4.11(-6)	7.15(-6)	4	2.88

Table 5: Numerical results of Example 4.1 with Mesh-b

Cells	*Methods*	$\|u-u_1^*\|_\infty$	$\|F^h-A^hu_1^*\|_2$	*Iterations*	*Time*
8192	AMG	3.23(-4)	9.23(-6)	14	0.08
	EAMG	3.22(-4)	7.19(-6)	5	0.05
32768	AMG	7.93(-5)	6.40(-6)	18	0.47
	EAMG	7.87(-5)	1.54(-6)	5	0.09
131072	AMG	2.01(-5)	9.77(-6)	19	3.36
	EAMG	1.93(-5)	1.62(-6)	5	0.83
524288	AMG	7.13(-6)	8.46(-6)	21	13.9
	EAMG	4.77(-6)	7.50(-7)	5	3.55

Table 6: Numerical results of Example 4.1 with Mesh-c

Cells	*Methods*	$\|u-u_1^*\|_\infty$	$\|F^h-A^h u_1^*\|_2$	*Iterations*	*Time*
19968	AMG	7.28(-5)	8.95(-6)	11	0.14
	EAMG	6.92(-5)	1.51(-6)	5	0.08
79872	AMG	2.36(-5)	5.24(-6)	12	0.64
	EAMG	2.13(-5)	3.13(-6)	5	0.25
319488	AMG	9.55(-6)	9.82(-6)	11	4.42
	EAMG	6.31(-6)	3.05(-6)	5	1.63

Table 7: Numerical results of Example 4.2 with Mesh-a

Cells	*Methods*	$\|u-u_1^*\|_\infty$	$\|F^h-A^h u_1^*\|_2$	*Iterations*	*Time*
8192	AMG	5.51(-3)	9.91(-6)	27	0.17
	EAMG	5.51(-3)	6.86(-6)	6	0.02
32768	AMG	1.35(-3)	8.72(-6)	31	0.67
	EAMG	1.35(-3)	8.15(-6)	6	0.13
131072	AMG	3.35(-4)	9.31(-6)	33	5.95
	EAMG	3.34(-4)	1.39(-6)	7	1.23
524288	AMG	8.34(-5)	8.25(-6)	36	23.6
	EAMG	8.32(-5)	1.35(-6)	7	4.48

Table 8: Numerical results of Example 4.2 with Mesh-b

Cells	*Methods*	$\|u-u_1^*\|_\infty$	$\|F^h-A^h u_1^*\|_2$	*Iterations*	*Time*
8192	AMG	2.99(-3)	6.07(-6)	22	0.14
	EAMG	2.99(-3)	4.70(-6)	7	0.03
32768	AMG	7.24(-4)	7.35(-6)	27	0.70
	EAMG	7.24(-4)	1.14(-6)	7	0.19
131072	AMG	1.78(-4)	9.30(-6)	30	5.23
	EAMG	1.78(-4)	7.07(-6)	6	1.14
524288	AMG	4.42(-5)	8.32(-6)	33	21.9
	EAMG	4.41(-5)	8.81(-7)	7	4.19

Table 9: Numerical results of Example 4.3

Cells	*Methods*	$\|u-u_1^*\|_\infty$	$\|F^h-A^h u_1^*\|_2$	*Iterations*	*Time*
16512	AMG	3.63(-4)	6.33(-6)	13	0.13
	EAMG	3.61(-4)	6.13(-6)	5	0.05
66048	AMG	1.00(-4)	4.94(-6)	14	0.58
	EAMG	9.97(-5)	4.65(-6)	5	0.16
264192	AMG	2.79(-5)	4.96(-6)	14	2.73
	EAMG	2.74(-5)	4.02(-6)	5	1.05

Table 10: Numerical results of Example 4.4

Cells	*Methods*	$\|u-u_1^*\|_\infty$	$\|F^h - A^h u_1^*\|_2$	*Iterations*	*Time*
14016	AMG	1.76(-3)	5.16(-6)	18	0.16
	EAMG	1.76(-3)	2.63(-6)	8	0.06
56064	AMG	5.20(-4)	8.83(-6)	19	0.59
	EAMG	5.20(-4)	4.38(-6)	8	0.25
224256	AMG	1.50(-4)	8.27(-6)	19	2.72
	EAMG	1.50(-4)	7.60(-6)	8	1.14

From Tables 1, 2 and 3, the number of rows and the number of nonzero entries both decreases with the rate $\frac{1}{4}$ (see the second and the third columns). The minimum, maximum and average entries per row keep stability, the condition number of hierarchy matrix reduced. Therefore, those equations on auxiliary coarse grids are convenient to compute and store.

The computational results of AMG and EAMG methods with some nonuniform meshes (see Figure 2 and 3) for the four examples are listed in Tables 4-10. Compared with computational cost of the existing AMG method, fewer number of iterations of the proposed method are required on the finest grid. Therefore, the EAMG method is faster than classical AMG method. When a larger scale grid is required, the superiority of the proposed method is more obvious. In the respect of precision, the two methods are able to obtain a desired accuracy approximation solutions. Particularly, the residual norm $\|F^h - A^h u_1^*\|_2$ of the EAMG method is a little bit more accurate than that of the compared method.

To sum up the above arguments, the proposed EAMG method keeps the scalability and efficiency.

5 Conclusion

We presented an efficient algebraic multigrid (EAMG) method embedding with a new algebraic prolongation operator based on coarsening strategy of adjacency matrix. Numerical results show that our EAMG method could achieve desired approximation solutions fast than the original AMG method. The present method is more suitable to computationally second order elliptic equations on polygonal domains.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (51974377), the Major Science and Technology Project of Precious Metal Materials Genetic Engineering in Yunnan Province (2019ZE001-1, 202002AB080001), and the Natural Science Foundation of Yunnan Province of China (2017FH001-012).

References

[1] Brandt A. Algebraic Multigrid Theory: the Symmetric Case. *Appl Math Comput.* 1986; (19): 23-56.

[2] Briggs WL, Henson VE, McCormick SF. A multigrid tutorial (second edition). SIAM. Philadelphia, Pennsylvania, PA, 2000.

[3] Chang QS, Wong YS, Fu HQ. On the algebraic multigrid method. *J. Comput Phys.* 1996; 125: 279-292.

[4] Ge YB. Multigrid method based on the transformation-free HOC scheme on nonuniform grids for 2D convection diffusion problems.*J Comput Phys.* 2011; 230(10): 4051-4070.

[5] Gupta MM, Zhang J. High accuracy multigrid solution of the 3D convection-diffusion euqaiton. *Appl Math Comput.* 2000; 113(2): 249-274.

[6] Hu HL, Chen CM, Pan KJ. Time-extrapolation algorithm (TEA) for linear parabolic problems. *J Comput Math.* 2014; 32: 183-194.

[7] Hu HL, Ren ZY, He DD, Pan KJ. On the convergence of an extrapolation cascadic multigrid method for elliptic problems. *Comput Math Appl.* 2017; 74: 759-771.

[8] Kickinger F. Algebraic Multigrid for Discrete Elliptic Second-Order problems, multigrid Methods V.Proceedings of the 5th European Multigrid conference ,Hackbusch W(ed).Springer Lecture Notes in computational Science and Engineering, vol.3.Springer:Berlin, 1998; 157-172.

[9] Li M, Zheng ZS, Pan KJ, Yue XQ. An efficient Newton multiscale multigrid method for 2D semilinear Poisson equations. *East Asian J Appl Math.* 2020; 10: 620-634.

[10] Li M, Zheng ZS. An efficient multiscale-like multigrid computation for 2D convection-diffusion equations on nonuniform grids. *Math Meth Appl Sci.* 2020; 1-11. https://doi.org/10.1002/mma.6895

[11] Pan KJ, He DD, Hu HL, Ren ZY. A new extrapolation cascadic multigrid method for three dimensional elliptic boundary value problems. *J Comput Phys.* 2017; 344: 499-515.

[12] Pan KJ, He DD, Hu HL. An extrapolation cascadic multigrid method combined with a fourth-order compact scheme for 3D Poisson equation. *J Sci Comput.* 2016; 70: 1180-1203.

[13] Pan KJ, Sun HW, Xu Y, Xu YF. An efficient multigrid solver for two-dimensional spatial fractional diffusion equations with variable coefficients. *Appl Math Comput.* 2021; 402: 126091.

[14] Shu S, Sun DD, Xu JC. An Algebraic Multigrid Method for Higher-order Finite Element Discretizations. *Computing*. 2006; 77: 347-377.

[15] Stüben K. A review of algebraic multigrid. *Comput Math.* 2001; 128(12): 281-309.

[16] Treister E, Yavneh I. Non-Galerkin multigrid based on sparsified smoothed aggregation. *SIAM J Sci Comput.* 2015; 37(1): A30-A54.

[17] Treister E, Yavneh I. On-the-fly adaptive smoothed aggregation multigrid for Markov chains. *SIAM J Sci Comput.* 2011; 33: 2927-2949.

[18] Wang Y, Zhang J. Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation. *J Comput Phy.* 2009; 228: 137-146.

[19] Zhang J. Multigrid method and fourth order compact difference scheme for 2D Poisson equation with unequal meshsize discretization. *J Comput Phys.* 2002; 179: 170-179.