A COMPARISON OF GEOMETRIC AND ALGEBRAIC MULTIGRID FOR DISCRETE CONVECTION-DIFFUSION EQUATIONS

CHIN-TIEN WU AND HOWARD C. ELMAN

ABSTRACT. The discrete convection-diffusion equations obtained from streamline diffusion finite element discretization are solved on both uniform meshes and adaptive meshes. The performance of geometric multigrid method (GMG) and algebraic multigrid method (AMG), both as a solver and as a preconditioner of the generalized minimal residual method (GMRES), are evaluated. Our numerical results show that GMRES with AMG preconditioner appears to be a robust and reliable solver.

1. INTRODUCTION

The purpose of this paper is to evaluate different solution strategies for the linear systems obtained from discretization of the convection-diffusion equation

$$\begin{cases} -\epsilon\triangle u + b\cdot\nabla u = f, \\ u = g \quad \text{on } \partial\Omega, \end{cases} \tag{1}$$

where b and f are sufficiently smooth and the domain Ω is convex with Lipschitz boundary $\partial\Omega$. We are interested in convection-dominated case, i.e. $|b| \gg \epsilon$. In this setting, the solution typically has steep gradients in some parts of the domain Ω. These may take the form of boundary layers caused by Dirichlet conditions on the outflow boundaries or internal layers caused by discontinuities in the inflow boundaries.

Let $\Im_h$ be a given quasi-uniform mesh of triangles on Ω and V_h be the linear finite element space on $\Im_h$. We will discretize (1) on V_h by using the streamline diffusion finite element method (SDFEM) [3]. SDFEM is a variant of the standard Galerkin method with extra diffusion in the streamline direction. The formulation is to find $u_h \in V_h$ such that

$$B_{sd}(u_h, v_h) = f_{sd}(v_h), \text{ for all } v_h \in V_h, \tag{2}$$

where

$$B_{sd}(u_h, v_h) = \epsilon(\nabla u_h, \nabla v_h) + (b\cdot\nabla u_h, v_h) + \sum_{T\in\Im_h} \delta_T(b\cdot\nabla u_h, b\cdot\nabla v_h)_T,$$

$$f_{sd}(v_h) = (f_h, v_h) + \sum_{T\in\Im_h} (f, \delta_T b\cdot\nabla v_h),$$

and δ_T is the stablization parameter. It is well known that the standard Galerkin finite element discretization on uniform grids produce inaccurate oscillatory solutions to these problems. With carefully chosen stablization parameter δ_T, the streamline diffusion finite element discretization is able to eliminate most oscillations and produce accurate solutions in the regions where no layers are present [3]. A good choice of δ_T is [2]

$$\delta_T = \begin{cases} 0 & \text{if } \mathrm{P_{e_T}} < 1 \\ \frac{1}{2\|\mathrm{b}\|_T}\left(1 - \frac{1}{\mathrm{P_{e_T}}}\right) & \text{if } \mathrm{P_{e_T}} > 1, \end{cases}$$

where

$$P_{e_T} = \frac{\|b\|_T \, h_T}{2\epsilon}, \quad \text{for } T \in \Im_h \text{ with diameter } h_T,$$

is the mesh Peclét number. To obtain higher accuracy of the solutions in the regions where layers are present, an a posteriori error indicator is computed to pinpoint the locations of layers, and a local mesh refinement strategy is used to refine elements in which the values of error indicator are large. The local mesh refinement strategy can be applied recursively until a certain tolerance of the error between the finite element solution and the true solution is met. Such a process is called an adaptive mesh refinement process.

In this paper, we are concerned with the costs of solving the discrete system obtained when the effective discretization strategy SDFEM together with adaptive mesh refinement process are used. Let $A_h u_h = f_h$ denote the discrete system of equations derived from (2). Here we would like to explore the effectiveness of the geometric multigrid (GMG) and algebraic multigrid (AMG) methods for solving this system when SDFEM and adaptive mesh refinement are used to discretize. We start with a coarse mesh, compute the discrete system using SDFEM, solve it using GMG or AMG, and then refine the mesh using the information provided by an a posteriori error estimation. This adaptive mesh refinement process is stopped after a given number of refinement steps.

These experiments are performed using the following two benchmark problems.

Problem 1: Flow with characteristic and downstream layers: The equation (1) is given with the coefficient $b = (\sin(\phi), \cos(\phi))$ and the right hand side $f = 0$ on the domain $\Omega = [-1, 1] \times [-1, 1]$. The Dirichlet boundary condition is set as $g = 1$ on the segments $y = 0 \cap x > 0$ and $x = 1$, and $g = 0$ elsewhere.

Problem 2: Flow with closed characteristics: Here, the coefficient vector (b_1, b_2) is $(2y(1-x^2), 2x(1-y^2))$ and the righthand side $f = 0$ on the domain $\Omega = [-1, 1] \times [-1, 1]$. The Dirichlet boundary condition is $g = 1$ on the segments $y = 1$ and $g = 0$ elsewhere.

Details of the error estimator and refinement strategy are given in Section 2, and multigrid solvers are given in Section 3. A comparison of the solvers are shown in Section 4. This comparison examines both standalone versions of multigrid as well as versions in which multigrid is used as a preconditioner for the generalized minimal residual method (GMRES) [8]. We also compare these with unpreconditioned GMRES, and with a preconditioned GMRES algorithm in which the smoother used for multigrid is used as a preconditioner. In section 5, conclusions are drawn.

2. Adaptive Mesh Refinement by A Posteriori Error Estimation

One common technique to increase the accuracy of the finite element solution is mesh refinement. Unlike uniform mesh refinement, the adaptive mesh refinement process refines meshes only in the regions where errors between the weak solution of the partial differential equation and the corresponding finite element solution are large. In general, the adaptive mesh refinement process consists of the loops of the following form:

$$\underbrace{Solve}_{1} \rightarrow \underbrace{Compute\ error\ indicator}_{2} \rightarrow \underbrace{Refine\ mesh}_{3}$$

A reliable computable a posteriori error estimator in step 2 is the key for the adaptive mesh refinement process to succeed. In step 3, elements in which the value of the a posteriori error estimator is large are marked for refinement according to some element selection algorithm, the so-called marking strategy. On a given triangulation $\Im_h$, for any element $T \in \Im_h$, let η_T be the a posteriori error indicator of element T. A heuristic marking strategy is the *maximum marking strategy* where an element T^* is marked for refinement if

$$\eta_{T^*} > \theta \max_{T \in \Im_h} \eta_T, \tag{3}$$

with a prescribed threshold $0 < \theta < 1$. For other marking strategies, we refer to [5]. Once decisions on where to refine are made, commonly used mesh refinement schemes are the regular refinement where a triangle is divided into four triangles equally, and the longest side bisection [6] where a triangle is divided into two triangles by introducing a new node to midpoint of the longest edge.

In our numerical tests, we use the a posteriori error estimator developed by Kay and Silvester [4] together with the maximum marking strategy and the regular refinement scheme for adaptive mesh refinement. First, let us introduce the following abbreviations. The interior residual of element T and the flux jump of edge E are denoted as follows:

$$\begin{aligned}
R_T &= (f - b \cdot \nabla u_h)|_T. \\
R_T^0 &= \mathcal{P}_T^0(R_T), \text{ where } \mathcal{P}_T^0 \text{ is the } L^2(T)\text{-projection onto } P^0(T). \\
R_E &= \begin{cases} [\frac{\partial u_h}{\partial n_E}]_E & \text{if } E \in \Omega \\ -2(\frac{\partial u_h}{\partial n_E}) & \text{if } E \in \Gamma_N \\ 0 & \text{if } E \in \Gamma_D, \end{cases}
\end{aligned}$$

where Γ_N denotes the Neumann boundary and Γ_D denotes the Dirichlet boundary. Let Φ be the element affine mapping from the physical domain to the computational domain and χ_i be the nodal basis function of node i. The approximation space is denoted as $Q_T = Q_T \bigoplus B_T$, where

$Q_T = span\{\psi_E \circ \Phi^{-1} |\ \psi_E = 4\chi_i\chi_j$, i,j are the endpoints of E and $E \in \partial T \bigcap (\Omega \bigcup \Gamma_N)\}$

is the space spanned by the quadratic edge bubble functions and

$$B_T = span\{\psi_T \circ \Phi^{-1} |\ \psi_T = 27 \prod_{i=1}^{3} \chi_i\}$$

is the space spanned by cubic interior bubble function. For an element T, the error estimator is given by $\eta_{h,T} = \|\nabla e_T\|_{0,T}$, where $e_T \in Q_T$ satisfies

$$\epsilon(\nabla e_T, \nabla v)_T = (R_T^0, v)_T - \frac{1}{2}\epsilon \sum_{E \in \partial T} (R_E, v)_E. \tag{4}$$

Let $e_h = u - u_h$. The a posteriori error estimation is specified as follows:

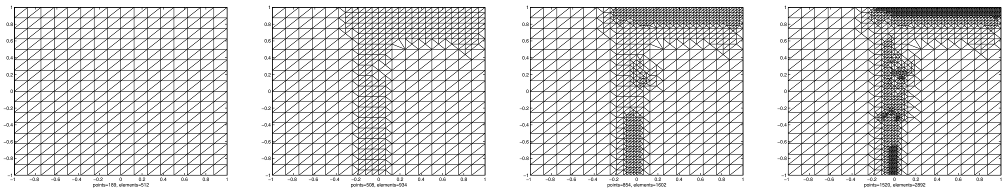
(**Global Upper Bound**):

$$\|\nabla(e_h)\|_{0,\Omega} \leq C\Big(\sum_{T\in\Im_h} \eta_{h,T}^2 + \sum_{T\in\Im_h} (\frac{h}{\epsilon})^2 \left\|R_T - R_T^0\right\|_{0,T}^2\Big)^{1/2} \tag{5}$$

(**Local Lower Bound**):

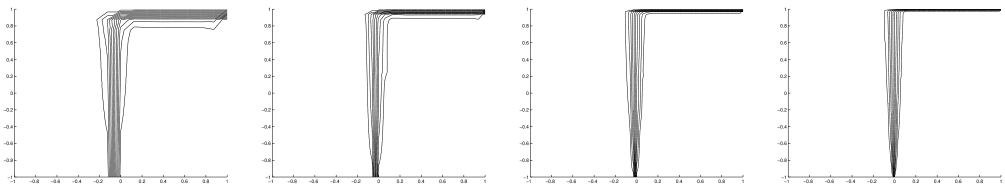
$$\eta_{h,T} \leq c\left(\|e_h\|_{0,\omega_T} + \sum_{T\subset\omega_T} \frac{h_T}{\epsilon} \|b\cdot\nabla e_h\|_{0,T} + \sum_{T\subset\omega_T} \frac{h_T}{\epsilon} \left\|R_T - R_T^0\right\|_{0,T}\right), \tag{6}$$

where constants C and c are independent of the diffusion parameter ϵ and mesh size h. Clearly, from the global upper bound, the error indicator provides an estimation of the true error on the whole domain. From the local lower bound, the error indicator can pinpoint where the error is large. In the following, we consider Problem 1 with the diffusion parameter $\epsilon = 10^{-3}$ and $\phi = 0$. By computing Kay and Silvester's error indicator, a sequence of adaptively refined meshes as shown in Figure 1 is obtained through the maximum marking strategy (3) with the threshold value $\theta = 0.01$. Figure 2 indicates that both the characteristic and downstream layers are resolved more accurately as the mesh is refined.



(a) $\Im_1$ (b) $\Im_2$ (c) $\Im_3$ (d) $\Im_4$

FIGURE 1. Adaptively refined mesh with threshold value $\theta = 0.01$



(a) Solution on $\Im_1$ (b) Solution on $\Im_2$ (c) Solution on $\Im_3$ (d) Solution on $\Im_4$

FIGURE 2. Contour plots of the solutions of Problem 1 on adaptively refined meshes starting with a uniform 16×16 mesh

3. THE MULTIGRID ALGORITHMS

Given a sequence of meshes $\Im_0, \Im_1, \cdots, \Im_n$, let A_k be the matrix defined on $\Im_k$ for $k = 0, \cdots, n$, let w_0 and g_0 be the initial guess and the initial right hand size respectively on the finest grid $\Im_n$, and let MG_0 represents the direct solver on the coarsest grid $\Im_0$. The general form of the multigrid algorithm is shown in Algorithm 1.

Algorithm 1 Multigrid Algorithm

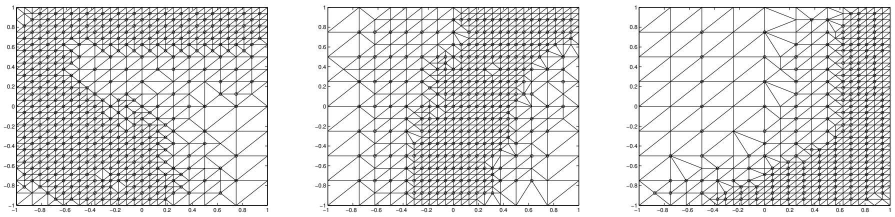
(1) set $x_k = w_k$,
(2) (pre-smoothing) $x_k = w_k + M_k^{-1}(g_k - A_k w_k)$,
(3) (restriction) $\bar{g}_k = I_k^{k-1}(g_k - A_k x_k)$,
(4) (correction) $q_i = MG_{k-1}(q_{i-1}, \bar{g}_k)$ for $1 \leq i \leq m$, $m = 1$ or 2 and $q_0 = 0$,
(5) (prolongation) $\bar{q}_m = I_{k-1}^k q_m$,
(6) set $x_k = x_k + \bar{q}_m$,
(7) (post-smoothing) $x_k = x_k + M_k^{-1}(g_k - A_k x_k)$,
(8) set $y_k = MG_k(w_k, g_k) = x_k$,

Here M_k^{-1} represents the smoothing operator and the operators I_k^{k-1} and I_{k-1}^k represent the grid transfers, restriction and prolongation, between $\Im_{k-1}$ and $\Im_k$. The multigrid V-cycle and W-cycle are defined by choosing $m = 1$ and $m = 2$ respectively. In this section, we describe the versions of GMG and AMG that we consider. We restrict our attention to the V-cycle. In the following, we describe each component of the multigrid V-cycle used.

First, we use the same smoother in both GMG and AMG. On uniform meshes, one step of the horizontal line Gauss-Seidel smoother (HGS) is used for Problem 1 and one step of a 4-directional line Gauss-Seidel smoother is used for Problem 4 in both pre-smoothing and post-smoothing. Here, a 4-directional line Gauss-Seidel smoother consists of four line Gauss-Seidel sweeps on the domain Ω: one sweeps from bottom to top, one sweeps from left to right, one sweeps from top to bottom, and one sweeps from right to left. On the adaptive meshes, since there is no natural horizonal line, one can number the nodes in lexicographical order where y-coordinate is the primary key and x-coordinate is the secondary key. The point Gauss-Seidel method associated with this node ordering is then called forward horizontal line Gauss-Seidel method (forward-HGS). Naturally, one can obtain another lexicographical order where the x-coordinate is the primary key and y-coordinate is the secondary key. The point Gauss-Seidel method associated with this node ordering is then called forward vertical line Gauss-Seidel method (forward-VGS). The backward-HGS and backward-VGS are obtained from forward-HGS and forward-VGS, respectively, by reversing the node ordering. In Problem 1, one step of forward-HGS is used in both pre-smoothing and post-smoothing on adaptive mesh. In Problem 4, the smoother consists of four point Gauss-Seidel sweeps (forward-HGS, forward-VGS, backward-HGS and backward-VGS) on the adaptive mesh. Hereafter, this smoother and the 4-directional line Gauss-Seidel are abbreviated as ADGS (namely alternative direction line Gauss-Seidel).

Second, in GMG, (1) the coarse grids are obtained directly from the mesh refinement process, (2) the grid transfer is via linear interpolation for prolongation and the restriction operator I_k^{k-1} is then taken to be the transport of the prolongation operator I_{k-1}^k, and (3) the matrix A_k is obtained directly from SDFEM discretization in the refinement process. On the other hand, in the algebraic multigrid method proposed by Ruge and Stüben [7] (AMG), (1) a coarse grid at level k-1 is generated by coarsening the fine grid along the so-called strong connection direction of matrix graph $\mathcal{G}$ of $A_k = (a_{i,j})$ where a directive edge $\overrightarrow{e}_{i,j} \in \mathcal{G}$ is defined to be strongly connected if $\mu \leq \frac{-a_{i,j}}{\max_{m \neq i}(-a_{i,m})}$ for a given parameter $0 < \mu < 1$; (2) the interpolation operator I_{k-1}^k is defined dynamically by a sophisticated algebraic formula during the AMG coarsening process, and (3) the matrix A_{k-1} in coarse grid is computed by $I_k^{k-1} A_k I_{k-1}^k$. The above process is repeated until all coarse grids are generated.

For details of the theoretical analysis on AMG and AMG coarsening process, we refer to [7]. In the following, we show the AMG coarse grids for the Problem 1 with $\epsilon = 10^{-3}$ and $\phi = -45^o$, 15^o and 75^o where the parameter $\mu = 0.25$. Each fine grid is generated by two adaptive mesh refinement steps from an initial 8x8 mesh and the threshold value θ in (3) is set to 0.01. In Figure 3, the fine grid is plotted and the coarse grid points selected by AMG coarsening are marked by $\circ$. It is clear that the AMG coarsening process is sensitive to the flow direction and generates meshes in a semi-coarsening way. Although, comparing to the GMG on adaptive meshes where interpolation and coarse-grid correction operators need not to be computed, extra computation cost of AMG is expected, AMG may converge much faster than GMG when the smooth errors are well in the range of the interpolation operator obtained from the AMG coarsening process.



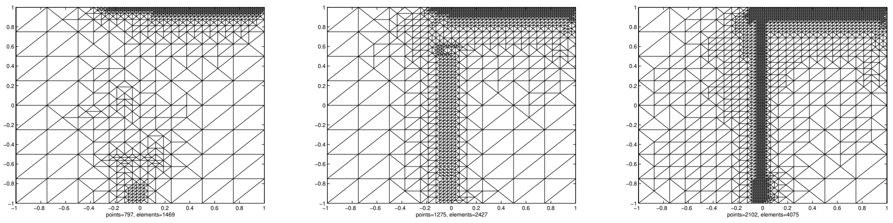
(a) $\phi = -45^o$ (b) $\phi = 15^o$ (c) $\phi = 75^o$

FIGURE 3. Coarse grid points selected by AMG coarsening

4. NUMERICAL RESULTS

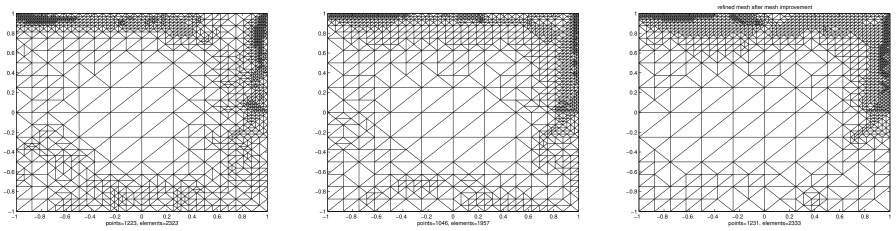
In this section, the performances of different linear solvers, including GMG, AMG and preconditioned GMRES, for the discrete convection-diffusion equation of Problem 1 with $\phi = 0$ and Problem 2, are compared on both adaptively refined meshes and uniform meshes. In both problems, the equation (1) is discretized on both uniform 32x32 mesh and an adaptively refined mesh for $\epsilon = 10^{-2}, 10^{-3}$ and 10^{-4}. Here, a 32×32 mesh is generated from 3 uniform refinement starting with a 4x4 initial mesh and the adaptively refined mesh is generated by refining an initial 8x8 mesh 4 times. The threshold value θ in the maximum marking strategy is chosen such that elements in the layer regions can be refined for both problems. For Problem 1, $\theta = 0.1$, 0.01 and 0.001 for $\epsilon = 10^{-2}, 10^{-3}$ and 10^{-4}, respectively. The adaptive meshes are shown in Figure 4. For Problem 2, $\theta = 0.1$ for all ϵ and the adaptive meshes are shown in Figure 5.

Both problems are solved by GMG, AMG, GMRES and preconditioned GMRES with zero initial guess. The preconditioned GMRES methods with HGS, ADGS, GMG and AMG preconditioners are denoted as GMRES-HGS, GMRES-ADGS, GMRES-GMG and GMRES-AMG respectively. The stopping tolerance of the iterative solvers is set to be $\|r_m\| \leq 10^{-6} \|r_0\|$, where r_0 is the initial residual and r_m is the residual at m-th iteration. In the following tables, level number is increased as mesh being refined and level 1 represents the coarsest mesh in multigrid solver. The notation "-" represents that the number of iterations is greater than 400.



(a) Mesh: $\epsilon = 10^{-2}$ (b) Mesh: $\epsilon = 10^{-3}$ (c) Mesh: $\epsilon = 10^{-4}$

FIGURE 4. Problem 1: Adaptively refined meshes after 4 refinements starting from a 8×8 grid for various ϵ



(a) Mesh: $\epsilon = 10^{-2}$ (b) Mesh: $\epsilon = 10^{-3}$ (c) Mesh: $\epsilon = 10^{-4}$

FIGURE 5. Problem 2: Adaptively refined meshes for various ϵ

First, Table 1 and 2, show some details about the meshes used by GMG and AMG. For GMG, meshes generated during the refinement process are considered as coarse grids and the numbers of grid points of these meshes are listed on the left of Table 1 and 2. For AMG, AMG coarsening, starting from the same finest mesh as in GMG , produces a sequence of coarse grids and the numbers of grid points of these grids are listed on the right of Table 1 and 2. Clearly, the AMG coarsening on the uniform meshes generates more coarse grid points but the AMG coarsening on adaptive meshes generates fewer coarse grid points compared to the corresponding coarse grids in the refinement process.

	GMG	AMG		
$\log_{10}\frac{1}{\epsilon}$	$2,3,4$	2	3	4
level=4	1089	1089	1089	1089
level=3	289	480	479	479
level=2	81	307	331	231
level=1	25	157	108	108

(a) On the uniform mesh

	GMG			AMG		
$\log_{10}\frac{1}{\epsilon}$	2	3	4	2	3	4
level=4	797	1275	2102	797	1275	2102
level=3	410	649	1047	348	580	996
level=2	215	320	528	159	304	523
level=1	122	176	239	88	166	281

(b) On the adaptive mesh

TABLE 1. Problem 1: Number of coarse grid points from adaptive refinement and AMG coarsening

	GMG	AMG		
$\log_{10}\frac{1}{\epsilon}$	$2,3,4$	2	3	4
level=4	1089	1089	1089	1089
level=3	289	502	500	498
level=2	81	289	288	280
level=1	25	168	146	151

(a) On the uniform mesh

	GMG			AMG		
$\log_{10}\frac{1}{\epsilon}$	2	3	4	2	3	4
level=4	1223	1046	1231	1223	1046	1231
level=3	629	645	824	573	461	565
level=2	315	381	390	311	254	323
level=1	161	203	202	171	127	179

(b) On the adaptive mesh

TABLE 2. Problem 2: Number of coarse grid points from adaptive refinement and AMG coarsening

Table 3-6 examine the iteration counts for both MG strategies as a function of the levels at which the solver is used. Table 3 and 4 show that both solvers display "textbook" performance on Problem 1 for $\epsilon = 10^{-2}$. As ϵ is reduced, however, performance of GMG is largely decreased and becomes dependent on mesh size. On the contrary, the performance of AMG is robust and shows weaker dependency on ϵ and mesh size. We contribute this to the superior choice of coarse mesh identified by AMG coarsening strategy, for which the smoother is better able to mimic the flow characteristics of the problem. Problem 2, with a recirculating flow, is more difficult for both solvers. In this case, although textbook performance is achieved on adaptively refined meshes, AMG is considerably slower then GMG. Our speculation is that this is caused by inadequacy of the grid transfer operator used here and this can be improved by the recently developed AMGe methods [1]. As we will show below, it is also remediable by using a Krylov Subspace acceleration.

level	GMG	AMG
3	13	7
2	13	6
1	12	6

(a) $\epsilon = 10^{-2}$

level	GMG	AMG
3	27	7
2	26	7
1	16	6

(b) $\epsilon = 10^{-3}$

level	GMG	AMG
3	51	9
2	35	8
1	17	6

(c) $\epsilon = 10^{-4}$

TABLE 3. Problem 1: Iteration counts for GMG and AMG on uniform meshes

level	GMG	AMG
4	9	6
3	8	8
2	7	6
1	7	5

(a) $\epsilon = 10^{-2}$

level	GMG	AMG
4	22	8
3	24	9
2	18	8
1	17	7

(b) $\epsilon = 10^{-3}$

level	GMG	AMG
4	59	14
3	57	10
2	47	8
1	34	7

(c) $\epsilon = 10^{-4}$

TABLE 4. Problem 1: Iteration counts for GMG and AMG on adaptive meshes

level	GMG	AMG
3	26	29
2	26	21
1	20	21

(a) $\epsilon = 10^{-2}$

level	GMG	AMG
3	187	256
2	142	106
1	50	25

(b) $\epsilon = 10^{-3}$

level	GMG	AMG
3	-	-
2	356	256
1	61	29

(c) $\epsilon = 10^{-4}$

TABLE 5. Problem 2: Iteration counts for GMG and AMG on uniform meshes

level	GMG	AMG
4	8	23
3	9	25
2	8	21
1	6	19

(a) $\epsilon = 10^{-2}$

level	GMG	AMG
4	19	142
3	22	172
2	25	127
1	21	110

(b) $\epsilon = 10^{-3}$

level	GMG	AMG
4	13	242
3	35	243
2	43	275
1	33	210

(c) $\epsilon = 10^{-4}$

TABLE 6. Problem 2: Iteration counts for GMG and AMG on adaptive meshes

Next, we compare the performance of preconditioned GMRES methods on both the finest uniform mesh and the finest adaptive mesh. Table 7 and 8 examine the iteration counts for varies versions of preconditioned GMRES. The numerical results indicate that the AMG-preconditioned GMRES requires fewest iterative steps to converge, compared to GMG, AMG and other versions of GMRES, except for solving Problem 2 on the adaptive mesh. However, even in this case, GMRES-AMG performs as competitive as the GMG-preconditioned GMRES (GMRES-GMG). The improvement of the convergence of GMG and AMG from Krylov-subspace acceleration is observed in most cases, particularly in Problem 2.

ϵ	10^{-2}	10^{-3}	10^{-4}
GMG	13	27	51
AMG	7	7	9
GMRES	58	75	94
GMRES-HGS	26	32	43
GMRES-GMG	14	20	28
GMRES-AMG	8	9	12

(a) Iterative steps on uniform mesh

ϵ	10^{-2}	10^{-3}	10^{-4}
GMG	9	22	59
AMG	6	8	14
GMRES	65	146	-
GMRES-HGS	11	31	59
GMRES-GMG	5	16	36
GMRES-AMG	4	8	14

(b) Iterative steps on adaptive mesh

TABLE 7. Problem 1: Iteration counts for various GMRES methods on finest grids

ϵ	10^{-2}	10^{-3}	10^{-4}
GMG	26	187	-
AMG	29	256	-
GMRES	-	-	-
GMRES-ADGS	37	59	77
GMRES-GMG	13	32	45
GMRES-AMG	11	24	33

(a) Iterative steps on uniform mesh

ϵ	10^{-2}	10^{-3}	10^{-4}
GMG	8	19	13
AMG	23	142	242
GMRES	-	-	-
GMRES-ADGS	34	42	40
GMRES-GMG	8	12	14
GMRES-AMG	10	16	16

(b) Iterative steps on adaptive mesh

TABLE 8. Problem 2: Iteration counts for various GMRES methods on finest grids

5. CONCLUSION

Accurate approximate solution for the convection-diffusion equation can be obtained from SDFEM discretization together with adaptive mesh refinement process. Although, with the nested meshes generated from adaptive refinement process, GMG seems to be a natural choice for solving this linear system, GMG converges much slower than AMG for flows with both characteristic and downstream layers (Problem 1). On the other hand, AMG converges much slower than GMG for flows with recirculation (Problem 2). In this work, we have found that when the Krylov-subspace acceleration is applied to both GMG and AMG, the convergence of GMG and AMG is improved for both types of flows. Moreover, GMRES-AMG performs either significantly better than GMRES-GMG or as competitive as GMRES-GMG. Despite of extra computation costs in AMG, the most important strength of AMG is that it is applicable to a wide range of applications as a black-box solver. Our studies have shown that the Krylov-subspace acceleration can make AMG an attractive solver for the convection-diffusion problems.

REFERENCES

[1] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S.F. Mccormick, and J. W. Ruge. Algebraic multigrid based on element interpolation (amge). *SIAM J. SCI. Comput.*, 22(5):1570–1592, 2000.

[2] B. Fischer, A. Ramage, D. Silvester, and A. J. Wathen. On parameter choice and iterative convergence for stabilised discretisations of advection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 179:185–202, 1999.

[3] T. J. R. Hughes, M Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Comput. Methods Appl. Mech. Engrg.*, 54:485–501, 1986.

[4] D. Kay and D. Silvester. The reliability of local error estimators for convection-diffusion equations. *IMA. Journal of Num. Anal.*, 21:107–122, 2001.

[5] A. Papastavrou and R. Verfürth. A posteriori error estimators for stationary convection-diffusion problems: a computational comparison. *Comput. Methods Appl. Mech. Engrg.*, 189:449–462, 2000.

[6] M. C. Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *Int. J. Numer. Methods Eng.*, 20:745–756, 1984.

[7] J. W. Ruge and K. Stüben. Algebraic Multigrid (AMG). Multigrid Methods. *SIAM, Frontiers in Appl. Math. Philadelphia.*, 5:73–130, 1987.

[8] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.