Mashadi, Syamsudhuha, MDH Gamal dan M. Imran, (Eds)
Proceedings of the International Seminar on Mathematics and Its Usage in Other Areas
November 11 – 12, 2010
ISBN. 978-979-1222-95-2

A LOCAL SMOOTING STRATEGY MULTIGRID METHOD FOR STABILISED DISCRETISATIONS OF CONVECTION-DIFFUSION PROBLEMS

Syamsudhuha

Department of Mathematics, Riau University, Pekanbaru, Indonesia.

ABSTRACT

In this paper, we discuss the development and practical application of multigrid methods with local smoothing to solve convection diffusion problems on adaptively refined meshes. The meshes are generated through adaptive refinement according to local error indicated by a residual-based a posteriori error estimation technique. We present a multigrid algorithm in which the smoothing is carried out only on the refined portion of the mesh. The smoothers, such as Gauss Seidel and ILU, give a good performance in both standard multigrid and local smoothing multigrid. Through several numerical examples, it seems that the rate of convergence of multigrid does not depend on the grid size and the viscosity parameter.

1. INTRODUCTION

Recently, there has been intensive research aimed at attaining a more theoretical understanding of multigrid techniques. In particular, for self–adjoint linear elliptic boundary value problems the convergence theory has been well developed and has reached a mature, if not final state. On the other hand, the quest for robust methods for convection-diffusion problems is ongoing. The application of a multigrid method becomes more problematic as the convection part becomes more dominant. This difficulty corresponds to finding a relaxation procedure that properly damps the high frequencies. In particular, even if the convection part is fairly modest with respect to the finest grid, it may still be difficult to find an appropriate smoother on the coarse grid level due to the fact that the mesh Peclet number is bigger on the coarser grids.

Adaptive refinement mesh can increase the efficiency of the finite element approximation. However, applying standard multigrid to the problems on locally refined meshes by smoothing at all points in each level is not efficient. It leads to a non-optimal growth of work and memory. In such a case it is convenient to be able to use a multigrid algorithm in which the smoothing is carried out only on the refined portion of the mesh (see, for example [1],[10],[13]). The first analysis of local smoothing has been given by Bramble et al. in 1991 [3]. They showed that the smoothing need only be done in the refined subdomain, and proved the convergence of a multigrid algorithm without regularity assumptions.

In this paper we address a local smoothing strategies as a solver for the linear equations arising on uniformly and adaptively refined meshes. We discuss the development and practical application of multigrid methods to solve PDE problems (1)–(2) on adaptively refined meshes. The approach followed here is the framework suggested by Christian Wieners in [13]. Then in this paper, we present numerical experiments associated with implementation of different strategies to address these issues.

2. LINEAR CONVECTION-DIFFUSION PROBLEMS

We consider a two-dimensional convection-diffusion problem of the following form

$$-\varepsilon \triangle u + \vec{\omega} \cdot \nabla u = f \quad \text{in } \Omega, \tag{1}$$

$$u = u_B \quad \text{on } \partial\Omega, \tag{2}$$

where Ω is an open region of $I\!R^2$, and f and u_B are given functions on Ω and $\partial\Omega$, respectively.

$0 < \varepsilon << 1$ is a diffusion or viscosity coefficient, and $\vec{\omega} = (\omega_x, \omega_y)$ is a given flow velocity field such that $\|\vec{\omega}\|_{L^\infty(\Omega)} = 1$; $\nabla \cdot \vec{\omega} = 0$.

Finite element discretisation of the problem can be done as follows: assume homogeneous boundary conditions $u_B = 0$ on all boundaries. Let $(\cdot,\cdot)$ denote the usual $L^2(\Omega)$ inner product. The weak formulation of (1) and (2) is then: find $u \in X := H_0^1(\Omega)$ such that

$$a(u,v) = (f,v) \quad \forall v \in X, \tag{3}$$

where the bilinear form is defined so that

$$a(u,v) := \varepsilon(\nabla u, \nabla v) + (\vec{\omega} \cdot \nabla u, v) \quad \forall u, v \in X. \tag{4}$$

Since Ω is bounded and ω is divergence-free, the bilinear form (4) has the following properties [5]:

$$\begin{aligned} a(u,u) &= \varepsilon\|\nabla u\|^2 && \forall u \in X && (5)\\ |a(u,v)| &\leq C\|\nabla u\|\,\|\nabla v\| && \forall u \in X, v \in X. && (6) \end{aligned}$$

We now state the *streamline diffusion finite element method* (SDFEM) for (1) with homogenous Dirichlet boundary condition using the standard space of linear finite element functions $X_h \subset X$. Following Johnson [7, pp. 185–187], the approximation of (1), reads: find $u_h \in X_h$, such that

$$a_{SD}(u_h, v) = l(v), \tag{7}$$

where

$$a_{SD}(u_h, v) := a(u_h, v) + \sum_{T \in \mathcal{T}^h} (\vec{\omega} \cdot \nabla u_h, \delta_T \vec{\omega} \cdot \nabla v)_T \tag{8}$$

and

$$l(v) := (f, v) + \sum_{T \in \mathcal{T}^h} \delta_T (f, \vec{\omega} \cdot \nabla v)_T, \tag{9}$$

for all $v \in X_h$ and elements T in the triangulation $\mathcal{T}^h$, where δ_T is a non-negative constant to be specified. A good choice for the stabilisation parameter δ_T has been introduced by Fischer et. al. in [5], which is as follows

$$\delta_T = \frac{\delta_* h_T}{\|\vec{\omega}\|_{L^\infty(T)}}, \tag{10}$$

where

$$\delta_* = \begin{cases} \frac{1}{2}\left(1 - \frac{1}{P_e^T}\right) & \text{for} \quad P_e^T > 1 \\ 0 & \text{for} \quad P_e^T \leq 1 \end{cases}$$

(for details, see [5]) for all *mesh Peclet numbers*

$$P_e^T \equiv \frac{h_T \|\vec{\omega}\|_{L^\infty(T)}}{2\varepsilon},$$

with h_T is a measure of the element length in the direction of the wind. In the case (10) the form $a_{SD}(,\cdot,)$ satisfies the coercivity condition

$$a_{SD}(u,u) = \varepsilon\|\nabla u\|^2 + \sum_{T \in \mathcal{T}^h} \delta_T \|\vec{\omega} \cdot \nabla u\|_T^2 \qquad \forall u \in X_h, \tag{11}$$

and implies that (8) has better stability than (4) since there is additional coercivity in the local flow direction.

$0 < \varepsilon << 1$ is a diffusion or viscosity coefficient, and $\vec{\omega} = (\omega_x, \omega_y)$ is a given flow velocity field such that $\|\vec{\omega}\|_{L^\infty(\Omega)} = 1$; $\nabla \cdot \vec{\omega} = 0$.

Finite element discretisation of the problem can be done as follows: assume homogeneous boundary conditions $u_B = 0$ on all boundaries. Let $(\cdot,\cdot)$ denote the usual $L^2(\Omega)$ inner product. The weak formulation of (1) and (2) is then: find $u \in X := H_0^1(\Omega)$ such that

$$a(u,v) = (f,v) \quad \forall v \in X, \tag{3}$$

where the bilinear form is defined so that

$$a(u,v) := \varepsilon(\nabla u, \nabla v) + (\vec{\omega} \cdot \nabla u, v) \quad \forall u,v \in X. \tag{4}$$

Since Ω is bounded and ω is divergence-free, the bilinear form (4) has the following properties [5]:

$$a(u,u) = \varepsilon\|\nabla u\|^2 \qquad \forall u \in X \tag{5}$$
$$|a(u,v)| \leq C\|\nabla u\|\,\|\nabla v\| \quad \forall u \in X, v \in X. \tag{6}$$

We now state the *streamline diffusion finite element method* (SDFEM) for (1) with homogenous Dirichlet boundary condition using the standard space of linear finite element functions $X_h \subset X$. Following Johnson [7, pp. 185–187], the approximation of (1), reads: find $u_h \in X_h$, such that

$$a_{SD}(u_h, v) = l(v), \tag{7}$$

where

$$a_{SD}(u_h, v) := a(u_h, v) + \sum_{T\in\mathcal{T}^h} (\vec{\omega}\cdot\nabla u_h, \delta_T\vec{\omega}\cdot\nabla v)_T \tag{8}$$

and

$$l(v) := (f,v) + \sum_{T\in\mathcal{T}^h} \delta_T(f, \vec{\omega}\cdot\nabla v)_T, \tag{9}$$

for all $v \in X_h$ and elements T in the triangulation $\mathcal{T}^h$, where δ_T is a non-negative constant to be specified. A good choice for the stabilisation parameter δ_T has been introduced by Fischer et. al. in [5], which is as follows

$$\delta_T = \frac{\delta_* h_T}{\|\vec{\omega}\|_{L^\infty(T)}}, \tag{10}$$

where

$$\delta_* = \begin{cases} \frac{1}{2}\left(1 - \frac{1}{P_e^T}\right) & \text{for} \quad P_e^T > 1 \\ 0 & \text{for} \quad P_e^T \leq 1 \end{cases}$$

(for details, see [5]) for all *mesh Peclet numbers*

$$P_e^T \equiv \frac{h_T\|\vec{\omega}\|_{L^\infty(T)}}{2\varepsilon},$$

with h_T is a measure of the element length in the direction of the wind. In the case (10) the form $a_{SD}(,\cdot,)$ satisfies the coercivity condition

$$a_{SD}(u,u) = \varepsilon\|\nabla u\|^2 + \sum_{T\in\mathcal{T}^h} \delta_T\|\vec{\omega}\cdot\nabla u\|_T^2 \qquad \forall u \in X_h, \tag{11}$$

and implies that (8) has better stability than (4) since there is additional coercivity in the local flow direction.

3. A LOCAL SMOOTHING MULTIGRID STRATEGY

We consider the linear system arising from streamline diffusion finite element discretisation (7)

$$A\mathbf{u} = \mathbf{b}, \tag{12}$$

where A is a nonsymmetric matrix. It is well known that if ε is small, i.e. for a convection-dominated problem, the construction of appropriate fast solvers becomes more difficult. In this section we discuss a simple multigrid strategy, as a solver to solve such a linear system.

4. MULTIGRID WITH LOCAL SMOOTHING

Let $\Omega \subset \mathbb{R}^2$ be a domain and let $\{\mathcal{T}_k\}$, $k = 0, \cdots, k_{max}$ be the sequence of grids approximating Ω. The set of triangular elements is denoted by $\mathcal{T}^k$, $k = 0, \cdots, k_{max}$. Assume that there exists a father element $\mathcal{F}(E) \in \mathcal{T}^{k-1}$ for all $E \in \mathcal{T}^k$. The set of son elements is defined as follows

$$\mathcal{G}(F) = \{E \in \mathcal{T}^k \mid \mathcal{F}(E) = F\}.$$

The refinement of an element F to $\mathcal{G}(F)$ can be of type *regular, irregular* or *copy* (for detail see Wieners [13], Bastian [2]). Furthermore, let $\mathcal{P}(E)$ be the set of vertices that belong to an element E. Based on the types of elements above, the vertices can be classified as follows.

- Set $cl(P) = 3$, if there exists a refined element E such that $P \in \mathcal{P}(E)$;
- set $cl(P) = 2$, if $cl(P) \neq 3$ and if there exists an element E such that $P, Q \in \mathcal{P}(E)$ and $cl(Q) = 3$;
- set $cl(P) = 1$, if $cl(P) \neq 2$, $cl(P) \neq 3$ and if there exists an element E such that $P, Q \in \mathcal{P}(E)$ and $cl(Q) = 2$;
- set $cl(P) = 0$ for all remaining points $P \in \mathcal{P}_k$.

As an example, see Figures 1 and 2. Let $\mathcal{T}^0$ be the initial mesh. The mesh is then locally refined into $\mathcal{T}^1$ (see Figure 2). All vertices in base level $\mathcal{T}^0$ get class 3. While on the local grid $\mathcal{T}^1$, all vertices marked by '$\bigoplus$' get class 3, the vertices marked by '$*$' get class 2 and the vertices marked by '$\diamond$' get class 1. Finally the unmarked vertices get class 0.

Figure 1: An initial mesh $\mathcal{T}^0$

4.1. Local smoothing

In the implementation of multigrid on locally refined meshes, the relaxation is carried out only for nodes that belong to the refined elements and some copy elements. This implies that local smoothing is performed to solve the system of equations

$$A^k(\mathcal{T}^k_{loc})\mathbf{e}^k(\mathcal{T}^k_{loc}) = \mathbf{r}^k(\mathcal{T}^k_{loc}),$$

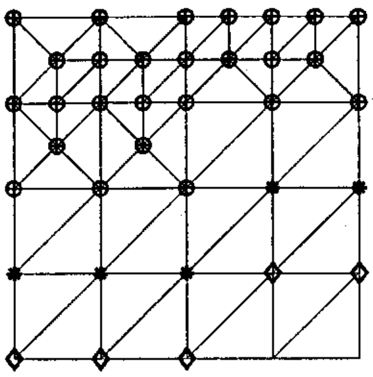
Figure 2: The mesh $\mathcal{T}^1$

where $\mathcal{T}^k_{loc}$ contains all nodes $P \in \mathcal{P}_k$ for $cl(P) \geq 2$. The smoothing is given by the following algorithm.

Algorithm 0.1 *(Local smoothing)*
$[\mathbf{e}^k, \mathbf{r}^k] = \mathcal{S}(\mathbf{e}^k, A^k, \mathbf{r}^k)$

```
1. Smooth $B\mathbf{v}^k = \mathbf{r}^k$, where $B \approx (A^k(\mathcal{T}^k_{loc}))$;
2. $\mathbf{r}^k := \mathbf{r}^k - A^k\mathbf{v}^k$, for nodes in $\mathcal{T}^k_{loc}$
3. $\mathbf{e}^k := \mathbf{e}^k + \mathbf{v}^k$
```

The computation in step 1 in Algorithm 0.1 should be local, e.g. $B = LU$ (incomplete LU decomposition of $A(\mathcal{T}_{loc})$). The defect $\mathbf{r}^k$ in step 2 is computed for all nodes of class 2 and 3. The correction step of the algorithm ensures smoothing on the refined region only i.e. all nodes of class 3.

4.2. Intergrid transfer

Since we deal with the meshes generated adaptively, the prolongation and restriction can be constructed locally. To this end, let a vector $\mathbf{v}^k$ on each level k be partitioned as

$$\mathbf{v}^k = \begin{pmatrix} \mathbf{v}^k_c \\ \mathbf{v}^k_f \end{pmatrix}, \tag{13}$$

where $\mathbf{v}^k_c$ is associated with the nodes of $\mathcal{T}^{k-1} \subset \mathcal{T}^k$ and $\mathbf{v}^k_f$ is associated with the nodes that are in $\mathcal{T}^k$ but not in $\mathcal{T}^{k-1}$. Following the partition in (13), the prolongation and restriction is given by

$$\mathcal{I}^k_{k-1} = \begin{pmatrix} I \\ \tilde{\mathcal{I}}^k_{k-1} \end{pmatrix}, \tag{14}$$

where $\tilde{\mathcal{I}}^k_{k-1}$ is the operator forming the arithmetic average of two values at nodes connected by an edge of the coarse triangulation. This implies that $\tilde{\mathcal{I}}^k_{k-1}(P, Q) \neq 0$ only if there is an element $E \in \mathcal{T}^k$ such that $P, Q \in \mathcal{P}(E)$ and $Q \in \mathcal{P}(\mathcal{F}(E))$. Furthermore, using (13) and (14), the interpolation from level $k-1$ to level k can be written as

$$\mathbf{v}^k = \mathcal{I}^k_{k-1}\mathbf{v}^{k-1} = \begin{pmatrix} \mathbf{v}^{k-1} \\ \tilde{\mathcal{I}}^k_{k-1}\mathbf{v}^{k-1} \end{pmatrix}, \tag{15}$$

which give values to new nodes in $\mathcal{P}_k \setminus \mathcal{P}_{k-1}$. Similarly, the restriction is given by

$$\mathbf{v}^{k-1} = \mathcal{I}^{k-1}_k \mathbf{v}^k = \mathbf{v}^k_c + \tilde{\mathcal{I}}^{k-1}_k \mathbf{v}^k_f. \tag{16}$$

This restriction will changes the value of $\mathbf{v}(P)$, if $\tilde{\mathcal{I}}_k^{k-1}(P,Q) \neq 0$ for $P \in \mathcal{P}_{k-1}$ and $Q \in \mathcal{P}_k \setminus \mathcal{P}_{k-1}$.

4.3. Local multigrid algorithm

We consider a linear system of equations $A^k\mathbf{u}^k = \mathbf{b}^k$ on each level k. Note that A^k arises from discretisation on a triangulation $\mathcal{T}^k$ for $k = 0, 1, \cdots, k_{max}$. Whilst $\mathbf{b}^k$ comes from discretisation (9) for $k = k_{max}$ (the finest level), and $\mathbf{b}^k = \mathbf{r}^k$ for coarser levels ($k < k_{max}$). Having defined the local smoothing and local grid transfer, a multigrid algorithm for local mesh refinement can be written as follows:

Algorithm 0.2 *(Local multigrid)*

```
$[\mathbf{e}^k, \mathbf{r}^k] = LMG(A^k, \mathbf{r}^k)$
  1. $\mathbf{e}^k = 0$
  2. if $k = 0$ the coarsest level
     $\mathbf{e}^0 = (A^0)^{-1}\mathbf{r}^0$
  3. else
     (a) for $0 \le i \le m_1$,
         $[\mathbf{e}^k, \mathbf{r}^k] := \mathcal{S}(\mathbf{e}^k, A^k, \mathbf{r}^k)$;
     (b) $\mathbf{r}^{k-1} = \mathcal{I}_k^{k-1}\mathbf{r}^k$;
     (c) $[\mathbf{e}^{k-1}, \mathbf{r}^{k-1}] = LMG(A^{k-1}, \mathbf{r}^{k-1})$
     (d) $\mathbf{v}^k = \mathcal{I}_{k-1}^k\mathbf{e}^{k-1}$;
     (e) $\mathbf{e}^k := \mathbf{e}^k + \mathbf{v}^k$;
     (f) $\mathbf{r}^k := \mathbf{r}^k - A^k\mathbf{v}^k$;
     (g) for $0 \le i \le m_2$,
         $[\mathbf{e}^k, \mathbf{r}^k] := \mathcal{S}(\mathbf{e}^k, A^k, \mathbf{r}^k)$.
```

The restriction from level k to $k-1$ in step 3b changes only the nodal values $P \in \mathcal{P}^{k-1}$ such that $cl(\mathcal{G}(P)) > 2$ in $\mathcal{P}^k$. The linear interpolation $\mathcal{I}_{k-1}^k$ in step 3d is used to prolongate the correction at all the nodes. This interpolation of $\mathbf{e}^{k-1}(P)$ will not change $\mathbf{e}^k(P)$ for all $P \in \mathcal{P}_{k-1}$. Hence, the smoothing and error correction process (relaxation, restriction and interpolation) are performed only on local grids. The complete multigrid solver for locally refined grids is finally defined as follows.

Algorithm 0.3 *(Local multigrid solver (LMG))*

```
1. $\mathbf{r}_0^k = b^k - A^k\mathbf{u}_0^k$, for a given initial guess $\mathbf{u}_0^k$
2. repeat
   • $[\mathbf{e}^k, \mathbf{r}^k] := LMG(A^k, \mathbf{r}^k)$;
   • $\mathbf{u}^k := \mathbf{u}^k + \mathbf{e}^k$;
3. until $\|\mathbf{r}^k\| \le TOL \times \|\mathbf{r}_0^k\|$
```

Here, it is not necessary to update the defect $\mathbf{r}^k$, since after every multigrid cycle the equality $\mathbf{r}^k = \mathbf{b}^k - A^k(\mathbf{u}^k + \mathbf{e}^k)$ holds.

5. NUMERICAL RESULTS

Now we apply the local multigrid technique (LMG) as a solver and a preconditioner (LMG-GMRES) to solve the test problems given in (1)-(2) on a locally refined mesh. As test problems we consider the equation,

$$-\varepsilon \triangle u + \vec{\omega} \cdot \nabla u = 0 \qquad \text{in } \vec{\Omega} \equiv (0,1) \times (0,1), \tag{17}$$

with (Dirichlet) boundary conditions as described in Figure 3. We take the following cases.

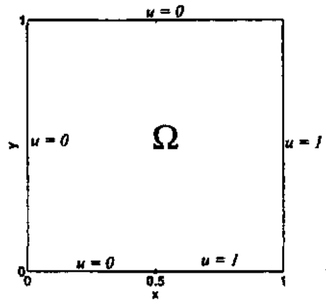

Figure 3: Boundary conditions

1. Constant wind $\vec{\omega} = (\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$;
2. Recirculating wind $\vec{\omega} = \big(2(2y-1)(1-(2x-1)^2), 2(2x-1)(1-(2y-1)^2)\big)$;

The equations are discretised using the SDFEM (see (7)) on triangular meshes. The meshes are refined (uniformly and adaptively) by connecting the mid-edges of triangles. We use the most basic multigrid method, namely a single V-Cycle on a series of nested grids. We begin with a finest grid of a certain size, then produce a series of coarser grids down to the coarsest level 0. On the coarsest grid level, we perform direct solution. Information is then passed straight back up through the same sequence in reverse to the finest level. The smoothers used are a standard point Gauss Seidel (GS) iteration and an Incomplete LU factorisation (ILU) method where the nodes are ordered by Matlab. One pre- and post-smoothing is performed at each level (MG-V(1,1)) for case 1, and we use MG-V(2,2) for case 2. The ILU method in this experiment is the Matlab built in function `luinc`. For grid transfer we use a standard linear interpolation which is easily applied in both uniform and non-uniform grid cases. The iteration is said to be converged when the error is less than the given tolerance, that is

$$\frac{||\mathbf{r}_i||}{||\mathbf{r}_0||} < 10^{-6}, \text{ where } \mathbf{r} = \mathbf{f} - A\mathbf{u},$$

with initial guess $\mathbf{u}_0 = \mathbf{0}$, a zero vector. After extensive testing, we determined an "optimal" strategy – that was choosing one local relaxation (i.e. $m_l = 1$ in Algorithm 0.2) at the pre-smoothing step. Whilst at the post-smoothing step we use one local and one standard relaxation. The results are presented in term of tables. The tables display; number of iterations (NI), and the number of floating point operations in millions (Mflops). We use the standard multigrid method (MG) and multigrid with Local smoothing (LMG).

Table 1 and Tabel 2 present results using MG and LMG respectively to solve Problem 1. It seems that LMG as a solver is fairly sensitive to any variation of the parameter ε. Using LMG as a preconditioner however, seems to be less sensitive to the change of ε. Grid-independent convergence is observed for both LMG and LMG-GMRES methods. The convergence rate of LMG-GMRES is slightly faster than LMG method. LMG-GMRES however, is more expensive than LMG, especially for linear systems of equations with high dimension. Moreover, from the table we see that LMG with ILU smoother is the most efficient solver in this case.

ε	d.o.f	MG		MG-GMRES	
		NI	Mflops	NI	Mflops
10^{-3}	1098	12	1.7	8	1.8
	2144	14	3.9	9	4.1
	4188	12	6.7	9	8.1
	8490	11	17.2	9	20.3
	16363	7	31.0	6	33.3
10^{-6}	1101	14	2.04	9	2.1
	2175	18	5.37	11	5.4
	4290	17	10.23	12	12.1
	8744	21	35.89	13	33.3
	16563	18	69.19	13	71.0
	32390	21	128.80	15	143.4

Table 1: *Multigrid convergence history for GS smoother: constant wind*

d.o.f	$\varepsilon = 10^{-3}$							
	LMG				LMG-GMRES			
	GS		ILU		GS		ILU	
	NI	Mflops	NI	Mflops	NI	Mflops	NI	Mflops
1098	9	1.4	5	1.3	6	1.4	5	1.7
2144	10	3.1	5	2.7	7	3.3	5	3.3
4188	9	5.5	5	5.2	7	6.4	5	6.5
8490	9	13.5	4	10.3	7	14.7	4	12.3
16363	7	26.1	4	25.7	6	28.9	4	29.5
	$\varepsilon = 10^{-6}$							
1101	10	1.5	5	1.3	7	1.6	5	1.7
2175	13	4.1	7	3.8	8	3.8	5	3.4
4290	13	8.1	6	6.5	9	8.7	6	8.2
8744	16	24.8	8	21.6	11	25.6	7	23.3
16563	16	52.6	8	45.8	12	57.9	7	48.4
32390	18	101.0	9	87.8	13	113.8	8	97.7

Table 2: *Local multigrid convergence history: constant wind*

We now study the results in Table 2 and Table 1. Comparison of these tables shows that (as expected) the operational cost of LMG scheme is less than that of standard multigrid (MG). It converges faster than MG. Interestingly, its cost is also (mostly) less than that of MG. For instance, we consider the problem with $\varepsilon = 10^{-6}$, and dof = 16563. MG in Table 1 converge after 18 iterations with 69.16 Mflops, respectively. On the other hand LMG needs 16 iterations and 52.6 Mflops.

d.o.f	$\varepsilon = 10^{-3}$							
	LMG				LMG-GMRES			
	GS		ILU		GS		ILU	
	NI	Mflops	NI	Mflops	NI	Mflops	NI	Mflops
1007	56	9.4	34	10.2	18	5.2	12	4.8
2210	47	20.3	29	22.6	18	112.6	12	11.9
3180	42	32.5	27	38.3	17	19.5	11	18.9
3560	38	41.5	25	50.1	15	22.2	10	23.2
3693	36	49.9	24	61.1	15	26.8	9	25.7
	$\varepsilon = 10^{-4}$							
1073	196	34.8	164	51.9	28	9.8	16	7.0
2855	196	120.3	123	137.2	39	46.8	24	36.8
4108	173	177.5	106	198.7	39	55.4	25	62.2
5120	104	238.6	97	270.9	40	103.8	26	93.0
7767	138	288.6	87	334.1	41	153.6	25	125.1

Table 3: Local multigrid convergence history: recirculating wind

Table 3 presents the local multigrid performance for the problem with recirculating wind. As in Problem 1, LMG as a solver and a preconditioner converges independent of the grid size. The convergence rate of LMG is on the other hand, both slower and more sensitive to the viscosity parameter than other methods. Using LMG as a preconditioner in this case can improve the convergence by factor of three.

6. CONCLUSION

Two test problems have been solved using two different multigrid techniques (MG and LMG). We have shown that the local relaxation affects the convergence rate appreciably. For problems with dominant diffusion, the standard multigrid may be used instead of LMG to obtain efficient convergence. LMG is shown to have less cost than the standard MG methods. In particular, for constant wind (Problem 1), this method (with ILU smoother) is the most efficient method. In the recirculating wind case, LMG-GMRES (with ILU smoother) performs better than LMG.

Acknowledgement: We thank DUE-Project Universitas Riau for funding this research.

REFERENCES

[1] Bastian, P., Wittum, G., *On robust and adaptive multigrid methods*, in Proc. of the 4th European Multigrid Conference, P. W. P. Hemker, ed., Birkhäuser, 1994.

[2] Bastian, P., *Load balancing for adaptive multigrid methods*, SIAM J. Sci. Comput., **19**, 1303–1321, 1998.

[3] Bramble, J.H., Pasciak, J. E., Wang, J., and Xu, J., *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comp., **57**, 23–45, 1991.

[4] Briggs W.L., *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

[5] Fisher, B., Ramage, A., Silvester, D.J., and Wathen, A.J., *On parameter choice and iterative convergence for stabilised discretisations of advection-diffusion problems*, Comput. Methods Appl. Mech. Engrg. **179**, 179 -195, 1999.

[6] Hackbusch, W. *Multigrid Methods and Applications*, Springer-Verlag, Berlin, 1985.

[7] Johnson, C., *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, 1992 .

[8] Kay, D., and Silvester, D. J., *The reliability of local error estimators for convection-diffusion equations*, IMA J. Numer. Anal., **21**, 107–122, 2001.

[9] Morton, K.W., *Numerical Solution of Convection-Diffusion Problems*, Chapman and Hall, 1996.

[10] Storti, M., Nigro, N., Idleson, S., *Multigrid methods and adaptive refinement techniques in elliptic problems by finite element*, Comput. Methods Appl. Mech. Engrg., **93**, 13-30, 1991.

[11] Syamsudhuha, Silvester, D. J., *Efficient solution of the steady-state Navier–Stokes equations using a multigrid preconditioned Newton–Krylov method*, International Journal for Numerical Method in Fluids, **43**; 1407-1427, 2003 .

[12] Wesseling, P., *Theoretical and practical aspects of a multigrid method*, SIAM J. Sci. Stat. Comput., **3**, 4, 387-407, 1982.

[13] Wieners, C., *The Implementation of adaptive multigrid methods for finite elements*, Institut für Computeranwendungen III, Universität Stuttgart Pfaffenwaldring 27, 70569 Stutgart, Germany.