East Asian Journal on Applied Mathematics
doi: 10.4208/eajam

Vol. **xx**, No. x, pp. 1-14
xxx 200x

An AMG preconditioner for moving mesh finite element method

Yirong Wu, Heyu Wang*

School of Mathematical Science, ZheJiang University, HangZhou, 310027, China

Abstract. In this paper, we apply an AMG prconditioner to solve the unsteady Navier-Stokes equations with moving mesh finite element method. $4P1-P1$ element pair is selected, which based on the data structure of hierarchy geometry tree. We choose two-layer nested meshes that velocity mesh and pressure mesh. AMG preconditioners are designed for PDE solver and divergence-interpolation in moving mesh strategy. Numerical experiments show the efficiency of the AMG preconditoner for moving mesh finite element.

AMS subject classifications: 65M10, 78A48

Key words: Navier-Stokes, algbraic multigrid precondition, moving mesh.

1. Introduction

The incompressible Navier-Stokes equations in primitive variables are

$$\begin{aligned} \partial_t\vec{u} - \nu\nabla^2\vec{u} + (\vec{u}\cdot\nabla)\vec{u} + \nabla p &= \vec{f}, \\ \nabla\cdot\vec{u} &= 0, \end{aligned} \tag{1.1}$$

with initial and boundary conditions on $\partial\Omega = \partial\Omega_D \bigcup \partial\Omega_N$:

$$\begin{aligned} &\vec{u} = \vec{w}, && \text{on } \partial\Omega_D \times [0,T] \\ &\nu\frac{\partial\vec{u}}{\partial n} - p = \vec{0}, && \text{on } \partial\Omega_N \times [0,T], \\ &\vec{u}|_{t=0} = \vec{u_0}, && \text{in } \Omega. \end{aligned} \tag{1.2}$$

where $\Omega \in \mathscr{R}^d, (d=2,3)$ is computional domain, $[0,T]$ is the time interval, $\vec{u}$ is velocity and scalar p is pressure, $\vec{n}$ denotes outward normal direction of $\partial\Omega$, $\nu > 0$ is the constant kinematic viscosity.

We solve (1.1) and (1.2) by moving mesh finite element methods based on [1] and [2]. In the past, some moving mesh methods have been introduced. Winslow [3] proposed

*Corresponding author. *Email addresses:* `21106058@zju.edu.cn` (Yirong Wu), `wangheyu@zju.edu.cn` (Heyu Wang)

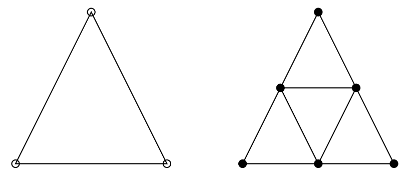


Figure 1: Left: pressure p element, $\circ$ for degrees of p; right: four velocity v elements, $\bullet$ for degrees of v.

solving elliptic PDEs using moving mesh. As an extension of Winslow' work, Dvinsky [4] pointed out that harmonic function theory could be used for generating mesh. Motivated by Dvinsky's work, L1, Tang and Zhang [1] proposed a moving mesh finite element strategy based upon harmonic mapping. The authors in [5] extended the moving strategy to solve the incompressible Navier-Stokes equations in primitive variables. The author designed a divergence-free interpolation in moving strategy by solving a linearized Navier-Stokes-type equations. In [6], $4P1-P1$ element pair is applied to solve incompressible Navier-Stokes flow with moving mesh finite element method based on the work of [5]. This pair has same mesh structure as $P1isoP2P1$ element, which is natrually LBB stable see [7]. Four velocity elements can be obtained by refining the pressure element one time see Figure 1. Linear velocity basis functions of $4P1-P1$ are all locally in the same velocity element, whereas $P1isoP2P1$ not, see [6] for detail.

As we known, spacial discretization of Navier-Stokes system with LBB-stable $4P1-P1$ element pair leads to a saddle point problem. There are a lot works on saddle point problems by developing preconditioners for Krylov subspace method, such as block preconditioner and multigrid strategy. Readers can refer to [8] for detail. Two-grid method was introduced to solve Navier-Stokes equations, see([9], [10] and [11]) for details. Many works([12], [13], [14], [15]) introduce a variety of block preconditioners, whose main issue are finding a good approximation of schur complement. Also there are other precondition methods, for instance ([16], [17]). The authors in ([18] [19]) propose an efficient AMG preconditioner for Krylov solver to solve Navier-Stokes equations. However, efficient precondition methods for saddle point problems are nearly based on uniform mesh (although the stretched mesh case is considered in [17]).

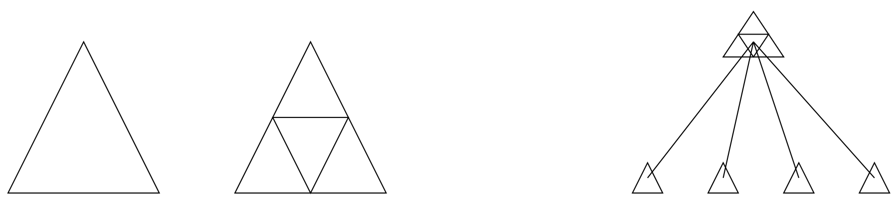
In this work, we apply an AMG preconditioner to moving mesh finite element for solving systems (1.1) and (1.2) based on the work of [20]. Also AMG precondition stretegy is designed for divergence-free interpolation in moving mesh method. Efficiency of the AMG preconditioner is analyzed througth several numerical experiments.

The layout of the paper is arraged as follows. In section 2, we use $4P1-P1$ element to approximate the governing equations. Next, the AMG preconditioner for Navier-Stokes equations is shown. In Section 4, we give the moving mesh strategy briefly. Then we present numerical experiements in section 5. Finaly, we give the conclusions in this section.

2. Data structure and weak formulation

At time level discretization, we devide the time interval $[0, T]$ into N steps with $\{t_i\}_{i=1}^N$. Let $\vec{u}^j$ and p^j be the discrete approximation to $\vec{u}(\cdot, t_j)$ and $p(\cdot, t_j)$. For simplicity, we choose linear backward Euler scheme that linearizing the nonlinear term $(\vec{u}^{n+1} \cdot \nabla)\vec{u}^{n+1}$ with $(\vec{u}^n \cdot \nabla)\vec{u}^{n+1}$.

In this work, we adopt finite element pair $4P1 - P1$, which based on two different trianglar meshes and two different finite element spaces. By using the hierarchy geometry tree ([21]) structure, velocity mesh can be obtained via global refining pressure mesh one time see Figure 2.



(a) One time uniform refined mesh (b) its hierarchy tree

Figure 2: Hierachy tree structure

The $1 - 1$ index between velocity elements and pressure elements can be obtained without difficulties with the hierarchy geometry tree structure. Interested author see [6] for details. First some notification are denoted as fowllows. $\mathscr{T}_h$ is the grid triangulation division for velocity mesh with mesh size $h = max_{T \in \mathscr{T}_h} diam(T)$, while $\mathscr{T}_H(H = 2h)$ for pressure mesh. $X_h \subset (H_0^1(\Omega)^2)$ and $P_H \subset \mathscr{L}^2(\Omega)$ are finite-dimentional approximation spaces. Then the full discretization is the following: given$(\vec{u}_h^n, p_H^n)$ at time t_n, to compute $(\vec{u}_h^{n+1}, p_H^{n+1})$ via

$$
\begin{aligned}
\frac{1}{dt}(\vec{u}_h^{n+1}, \vec{v}_h) + \nu(\nabla \vec{u}_h^{n+1}, \nabla \vec{v}_h) + (\vec{u}_h^n \cdot \nabla \vec{u}_h^{n+1}, \vec{v}_h) - (p_H^{n+1}, \nabla \vec{v}_h) = & \quad \frac{1}{dt}(\vec{u}_h^n, \vec{v}) \\
(\nabla \cdot \vec{u}_h^{n+1}, q_H) = & \quad 0.
\end{aligned}
\tag{2.1}
$$

for all $(\vec{v}_h, q_H) \in \mathscr{X}_h \times P_H$.

3. Fast krylov solver with AMG precondition strategy

Let $\left(\{\phi_j\}_{j=1}^n, 0\right)^T$ and $\left(0, \{\phi_j\}_{j=1}^n\right)^T$ be linear basis functions for velocity space X_h. Meanwhile, $\{\psi_k\}_{k=1}^m$ denotes linear basis functions for pressure space P_H. Then components of velocity solutions $\vec{u}_h^{n+1} = (u_h^{x,n+1}, u_h^{y,n+1})^T$ and pressure solution $\vec{p}_H^{n+1}$ at $t = t_{n+1}$ can be written as

$$
u_h^{x,n+1} = \sum_{j=1}^{n_u} \alpha_j^{x,n+1} \phi_j, \qquad u_h^{y,n+1} = \sum_{j=1}^{n_u} \alpha_j^{y,n+1} \phi_j \qquad p_H^{n+1} = \sum_{k=1}^{n_p} \alpha_k^{p,n+1} \psi_k. \tag{3.1}
$$

substituting (3.1) inito weak form (2.1), a linear system can be obtained

$$\left[\begin{array}{lll} \frac{1}{dt}M+\nu A+N & 0 & B_x^T \\ 0 & \frac{1}{dt}M+\nu A+N & B_y^T \\ B_x & B_y & 0 \end{array}\right]\left[\begin{array}{c} \alpha^{x,n+1} \\ \alpha^{y,n+1} \\ \alpha^{p,n+1} \end{array}\right]=\left[\begin{array}{c} f_x \\ f_y \\ 0 \end{array}\right], \tag{3.2}$$

Notice that divergence matrix $B=[Bx,By]$ is

$$B_x := [B_x]_{kj} = -\left(\psi_k, \frac{\partial \phi_j}{\partial x}\right), k=1,\cdots,n_p, j=1,\cdots,n_u, \tag{3.3}$$

$$B_y := [B_y]_{kj} = -\left(\psi_k, \frac{\partial \phi_j}{\partial y}\right), k=1,\cdots,n_p, j=1,\cdots,n_u. \tag{3.4}$$

Assembling of matrix B is a non-trival process due to the basis functions of velocity elements and pressure elements are on different mesh. According to the $1-1$ index between velocity elements and pressure elements metioned above, we can just use local $P1$ element of both velocity and pressure elements to assemble B. B^T is in the same way.

We denote $F_\nu^{n+1}=\frac{1}{dt}M+\nu A+N$, where

$$\begin{aligned} M := \quad [M]_{ij} = \quad & \left(\phi_i,\phi_j\right), \quad i,j=1,\cdots,n_u, & (3.5)\\ A := \quad [A]_{ij} = \quad & \left(\nabla\phi_i,\nabla\phi_j\right), i,j=1,\cdots,n_u, & (3.6)\\ N := \quad [N]_{ij} = \quad & \left(\vec{u}_h^n,\nabla\phi_i,\phi_j\right), i,j=1,\cdots,n_u. & (3.7) \end{aligned}$$

To solve linear system (3.2) efficiently, we use preconditioned GMRES as solver. The block triangular preconditioner $\mathscr{P}$ discussed in [20] which is defined as following

$$\mathscr{P}=\left(\begin{array}{ccc} F & 0 & B_x^T \\ 0 & F & B_y^T \\ 0 & 0 & S \end{array}\right) \tag{3.8}$$

where $S=B_xF^{-1}B_x^T+B_yF^{-1}B_y^T$ is the schur complement matrix. The action of $\mathscr{P}^{-1}$ is divided into two steps: first, solve schur complement system, second, solve two scalar systems associated with F. It is costly to directly solve schur complement system. So in practical computation, the PCD preconditioner discussed in [20] is used to approximate schur complement matrix S. PCD preconditioner is denoted as $S_*=A_pF_p^{-1}Q_p$, where A_p,F_p and Q_p are all on the pressure space. Q_p is mass matrix, A_p is pressure diffusion matrix and F_p is convection diffusion matrix denoted as

$$F_p := [F_p]_{ij} = \nu(\nabla\psi_i,\nabla\psi_j)+(\vec{u}_h^n\cdot\nabla\psi_i,\psi_j), \quad i,j=1,\cdots,n_p, \tag{3.9}$$

$$A_p := [A_p]_{ij} = (\nabla\psi_i,\nabla\psi_j) \quad i,j=1,\cdots,n_p. \tag{3.10}$$

Let $W_p^n := [W_p^n]_{ij} = (\vec{u}_h^n\cdot\nabla\psi_i,\psi_j), \quad i,j=1,\cdots,n_p$, then F_p can be rewitten as $F_p = \nu A_p + W_p^n$. We implement PCD preconditioning by

$$S_*^{-1} \approx Q_p^{-1}F_pA_p^{-1}. \tag{3.11}$$

Exact PCD preconditioning opearator is denoted as

$$\mathscr{M}^{-1} = \begin{pmatrix} F^{-1} & 0 & B_x^T \\ 0 & F^{-1} & B_y^T \\ 0 & 0 & S_*^{-1} \end{pmatrix} \tag{3.12}$$

We will explain the $V^d = \mathscr{M}^{-1}V^s$ in two steps, where $V^d = (V_x^d, V_y^d, V_p^d)^T, V^s = (V_x^s, V_y^s, V_p^s)^T$. Firstly, we solve

$$V_p^d = S_*^{-1}V_p^s = Q_p^{-1}F_pA_p^{-1}V_p^s. \tag{3.13}$$

It contains two possion problems Q_p^{-1} and A_p^{-1}, so we can use just an AMG solver for possion equation to solve them. Secondly, we use AMG solver to solve

$$\begin{aligned} V_x^d &= F^{-1}(V_x^s - B_x^TV_p^d), \\ V_y^d &= F^{-1}(V_y^s - B_y^TV_p^d). \end{aligned} \tag{3.14}$$

Combine (3.13) and (3.14), then V^d is obtained.

In practical computation, we use a fixed number of AMG iterations (usually one or two) for matrix F, F_p, Q_p, and A_p to replace accurate solving, which refers to iterated PCD preconditioning. The AMG solver is based on the AFPack(an adaptive finite element package), which can be obtained from `http://dsec.pku.edu.cn/~rli`. The efficiency of PCD preconditioning is shown in ([20], Section 10) and [22] for bouyancy driven flow problem. In our experiements, F_p in (3.10) is not so efficient as getting rid of ν in (3.10). If without explanation, we refer $F_p = A_p + W_p^n$ in this paper. We compare the efficiency of two choices of F_p in numerical test below. We adopt the method in [15] to deal with matrixes F_p and A_p on Neumann boundary for improving efficiency. F_p should satisfy condition

$$\nu\frac{\partial p_h}{\partial n} + (\vec{w}_h \cdot \vec{n})p_h = 0. \tag{3.15}$$

on boundary $\partial\Omega$. We know that for cavity flow, (3.15) will become $\frac{\partial p_h}{\partial n} = 0$, which means do-nothing for F_p on boundary Ω. In this work, we apply the PCD preconditioning strategy to moving mesh finite element method to efficiently solve system (3.2). The moving strategy will be shown in next section.

4. Moving mesh strategy

4.1. Moving mesh framework

We refer the moving strategy to [5]. In the following, we briefly introduce the moving mesh method. At time $t = t_n$, we obtain numerical solutions $\vec{u}_h^{(n)}, p_H^{(n)}$ on old mesh $\mathscr{T}_h^n$. We follow the framework in [5] to implement divergence-free interpolation of solutions on $\mathscr{T}_h^n$ to new mesh $\mathscr{T}_h^{(n+1)}$. Briefly speaking, the moving mesh strategy mainly contains four steps as follows.

step 1 Obtain monitor function. It is very important to choose an appropriate monitor function for adaptive scheme. Let $m = 1/G$, where G is the monitor function. As illustrated in [5], there are some common choices of G. One based on vorticity is

$$G = \sqrt{1 + \alpha|\omega|^{\beta}}. \tag{4.1}$$

where $\omega = \nabla \times \vec{u}$, α, β are positive constants. In this work, $\beta = 2$ performs well, while α is user defined according to different problems.

step 2 Get a new logical mesh. Solve elliptic equation

$$\begin{aligned} \nabla_{\vec{x}}(m\nabla_{\vec{x}}\vec{\xi}) = 0, \\ \vec{\xi}|_{\partial\Omega} = \vec{\xi}_b. \end{aligned} \tag{4.2}$$

where m is given in step 1. Then a new logical mesh $\mathscr{T}_c^*$ with $\mathscr{A}^*$ as nodes is obtained.

step 3 Achieve mesh move direction in physical domain. First, After Step 1 and Step 2, we obtain a new logical mesh $\mathscr{T}_c^*$ (with $\mathscr{A}^*$ as node). Then we can get the difference between $\mathscr{T}_c^*$ and initial logical mesh $\mathscr{T}_c^0(withnodes\mathscr{A}^0)$:

$$\delta\mathscr{A} = \mathscr{A}^0 - \mathscr{A}^*. \tag{4.3}$$

The displacement δX_i in physical domain cen be obtained with $\delta\mathscr{A}$. Moreover a positive parameter μ is multiplied to the displacement δX_i in updating old mesh in physical domain to a new one:

$$X_i^{(n+1)} = X_i^{(n)} + \mu\delta X_i. \tag{4.4}$$

Intrested readers can see [23] for mesh redistribution in detail.

step 4 Preserve divergence-free interpolation. It is necessary to keep divergence-free in the interpolation when solving incompressible flow with moving mesh finite element method. In [5], solution re-distribution on the new mesh $\mathscr{T}^{(n+1)}$ is achieved via solving a lineard inviscid Navier-Stokes-type system as following

$$\begin{aligned} \frac{\partial\vec{u}}{\partial\tau} - \nabla_{\vec{x}}\vec{u}\cdot\delta\vec{x} = & \quad -\nabla\hat{p}, \\ \nabla_{\vec{x}}\cdot\vec{u} = & \quad\quad 0. \end{aligned} \tag{4.5}$$

where $\delta\vec{x} := x^{\text{old}} - x^{\text{new}}$ and $x^{\text{old}}, x^{\text{new}}$ are two sets of coordinates in physical domain. τ is a virtual time variable and often choosen as 1.0, because of the convection speed $\delta\vec{x}$ is relatively small. Here $\hat{p}$ is a temporary variable distinguished from the pressure variable in (1.1).

Weak form of (4.5) is : find $(\vec{u}_h, \hat{p}_H) \in X_E^h \times P^H$ such that

$$\begin{aligned} \left(\partial_\tau \vec{u}_h - \nabla_{\vec{x}} \vec{u}_h \cdot \delta \vec{x}, \vec{v}_h\right) = & \quad \left(\hat{p}_H, \nabla \vec{v}_h\right), \quad \forall \vec{v}_h \in X_E^h, \\ \left(\nabla_{\vec{x}} \cdot \vec{u}, q_H\right) = & \quad\quad\quad\quad 0, \quad \forall q_H \in P^H. \end{aligned} \tag{4.6}$$

In this work, we use explicit scheme to (4.6) for time discretization:

$$\begin{aligned} \left(\frac{\vec{u}_{h,*}^{(n)} - \vec{u}_h^{(n)}}{\delta t}, \vec{v}_h\right) + \left(\delta \vec{x} \cdot \nabla \vec{u}_h^{(n)}, \vec{v}_h\right) = & \quad \left(\hat{p}_{H,*}^{(n)}, \nabla \vec{v}_h\right), \quad \forall \vec{v}_h \in X_E^h. \\ \left(\nabla \cdot \vec{u}_{h,*}^n, q_H\right) = & \quad\quad\quad\quad 0, \quad \forall q_H \in P^H. \end{aligned} \tag{4.7}$$

where $\vec{u}_h^{(n)}$ and $p_H^{(n)}$ are the numerical solutions of (1.1) at $t = t_n$ using the mesh at t_n. $\vec{u}_{h,*}^{(n)}$ and $p_{h,*}^{(n)}$ are the intermediate updated solutions at t_n on the new mesh $\mathcal{T}^{(n+1)}$.

4.2. AMG preconditioning strategy for (4.7) in solution re-distribution

(4.7) will bring out a linear system, whose coefficient matrix $\mathcal{M}^p$ can be denoted as

$$\mathcal{M}^p = \begin{pmatrix} \frac{1}{\delta t} Q_p & 0 & B_x^T \\ 0 & \frac{1}{\delta t} Q_p & B_y^T \\ B_x & B_y & 0 \end{pmatrix} \tag{4.8}$$

As we known, the schur complement of matrix $\mathcal{M}^p$ is $M_S = B_x Q_p^{-1} B_x^T + B_y Q_p^{-1} B_y^T$. Refering to ([20], section 5), for LBB stable mixed approximations with enclosed flow boundary conditions, M_S is spectral equivalent with pressure Laplacian matrix A_p. So we use A_p to appropriate schur complement M_S. Then we choose the block trianglar preconditioner

$$\mathcal{K} = \begin{pmatrix} Q_p & 0 & B_x^T \\ 0 & Q_p & B_y^T \\ 0 & 0 & M_S^* \end{pmatrix} \tag{4.9}$$

for (4.8), where $M_S^* = A_p$ or $M_S^* = \frac{1}{\nu} A_p$. We will contrast the efficiency of the preconditioner by different choice of M_S^*. In our practical computation, $\frac{1}{\nu} A_p$ performs more efficiently than A_p. For inflow/outflow problem, some modifications should be given for A_p on Neumann boundary to improve efficiency. Along the outflow boundary $\partial \Omega_N$, the discrete pressure p_h has to satisfy a homogeneous Dirichlet boundary condition. While a Neumann condition $\frac{\partial p_h}{\partial n} = 0$ is needed along $\partial \Omega_D$, that means we do nothing on p_h along Dirichlet boundary. see [15] for detail. Note that all the matrixes $M, B_x^T, B_y^T, B_x, B_y, A_p$ have to rebuilt once the meshes move.

In our algorithm, PCD preconditioned GMRES is selected as a solver solving linear system (3.2). We denote the stop criterion for GMRES convergence is

$$||r^{(k)}|| \leq 10^{-6}||r^{(0)}|| \tag{4.10}$$

where $r^{(k)}$ is the residual of the linear system (3.2) and r^0 is right hand side of (3.2). Finally, to illustrate our algorithm clearly, we give the flow-chart as the following algorithm 4.1:

Algorithm 4.1 Moving mesh FEM for Navier Stokes equation

1: Solve steady Stokes flow to give the initial value $\vec{u}_h^{(0)}, p_H^{(0)}$.
2: **while** $t_n < T$ **do**
3: Caculate monitor function on mesh $\triangle_p^{(n)}$ using $\vec{u}_h^{(n)}, p_H^{(n)}$ and obtain logical mesh $\vec{\xi}^*$ by solving (4.2).
4: Judge if L_2 norm of $\vec{\xi}^* - \vec{\xi}^{(0)}$ is less than tolerance. If yes, the iterator is over, else continue 5 - 8.
5: Caculate move direction $\delta\vec{x}$ of $\triangle_p^{(n)}$ using the difference of $\vec{\xi}^* - \vec{\xi}^{(0)}$.
6: Solve equation (4.7) on $\triangle_v^{(n)}$ to get medium variable $\vec{u}_{h,*}^{(n)}, p_{H,*}^{(n)}$.
7: Update mesh $\triangle_p^{(n)}$ to $\triangle_p^{(n+1)}$ and synchronize $\triangle_v^{(n)}$ to $\triangle_v^{(n+1)}$ by the hierachy geometry tree stucture.
8: Go back to 3.
9: Solve Navier-Stokes system (3.2) to obtain numerical solutions $\vec{u}_h^{(n+1)}, p_H^{(n+1)}$ on mesh $\triangle_v^{(n+1)}$ and $\triangle_p^{(n+1)}$.
10: **end while**

5. Numerical tests

We use three numerical tests to show our strategy. In practical computation, we choose the solutions of steady Stokes equations as the initial value of Navier-Stokes equations. The initial physical domain and logical domain in moving algorithm are the same. Moving mesh and numerical solutions are shown in below. Our codes are all based on the finite element package AFEPack.

5.1. Driven cavity flow

We consider the benchmark problem: regularized cavity flow. Our computional domain is $\Omega = [-1,1]\times[-1,1]$ and viscosity is $\nu = 0.001$. Dirichlet boundary condition is imposed on $\partial\Omega$. At the top boundary, $\vec{u} = (1-x^4, 0)^T$ while no-slip boundary condition is setted on other parts of $\partial\Omega$.

In our moving strategy, (4.1) is selected as monitor function. Parameters $\alpha = 0.5, \beta = 2.0$ perform well. The moving mesh and vorticity contour evoluting to steady state are shown in Figure 3. It can be seen that mesh clusters at top boundary and right boundary where the magnitude of vorticity is large. Velocity streamline is shown in Figure 5. We contrast the magnitude of velocity divergence between uniform mesh and moving mesh in Figure 4. It is found that the magnitude of velocity divergence in moving mesh is one harf of uniform case. From Figure **??**, it requires less GMRES iteration steps by choosing $F_p = A_p + W_p^n$ than $F_p = \nu A_p + W_p^n$ in PCD preconditioning. We compare the number of GMRES iteration counts in solving linear system (3.2) with AMG preconditioner and ILU preconditioner in Table 1. It is discoverd that AMG in more efficient than ILU. We also compare the number of GMRES iteration steps in solving (4.7) with different preconditioning matrix M_S^*. It turns out to be that $M_S^* = \frac{1}{\nu}$ is more efficient than $M_S^* = A_p$ in solving (4.7).

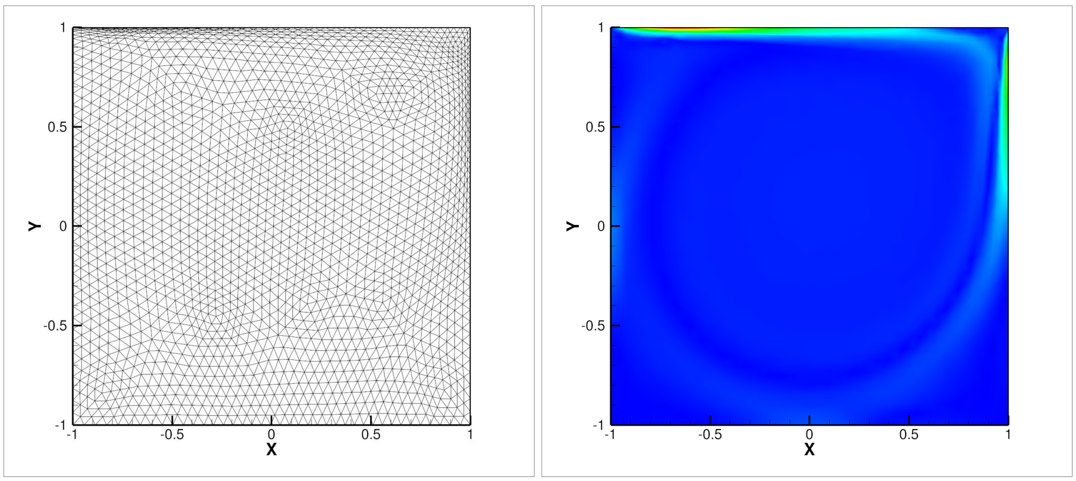


Figure 3: Cavity flow, left: mesh, right: vorticity contour, pressure mesh 20×20, $Re = 2000$.

pressure mesh	mean GMRES step number	
	AMG	ILU
20×20	10.8	107.5
40×40	16.01	307.33
80×80	27.18	> 500

Table 1: Cavity flow: comparation of mean GMRES step counts in solving linear system (3.2) with AMG and ILU precondtioning, $Re = 2000$.

5.2. Flow over cylinder

This example models the development of flow over an cylinder along a retangular channel. This problem has been considered in [24]. The center of cylinder is $(0,0)$ and the radius is $r = 0.3$. Let viscosity $\nu = 1/300$, thus the Reynolds number $Re = \frac{2rU_m}{\nu} = 240$

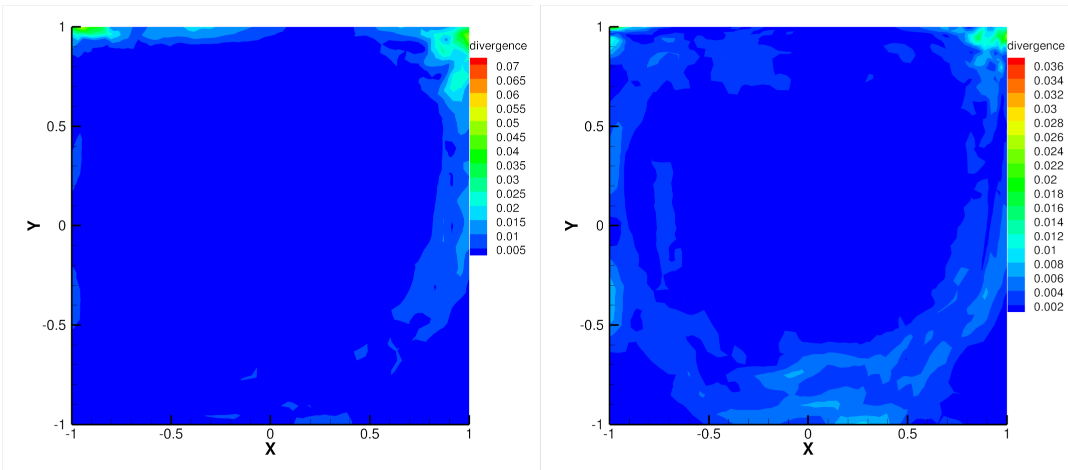


Figure 4: Cavity flow, divergence of velocity, left: uniform mesh, right: moving mesh, pressure mesh 20×20, $Re = 2000$.

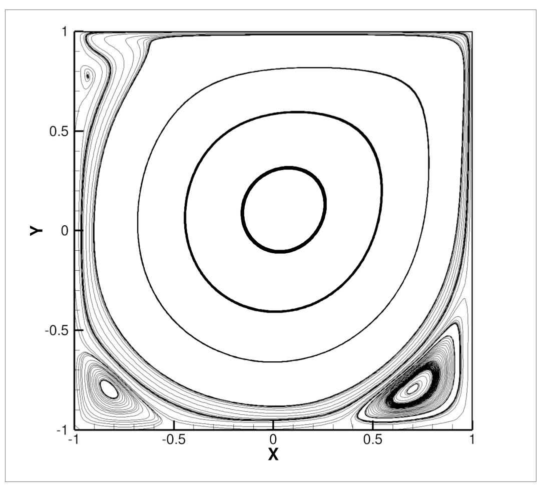


Figure 5: Cavity flow: velocity streamline, pressure mesh 20×20, $Re = 2000$.

pressure mesh	GMRES step number	
	$M_S^* = \frac{1}{\nu} A_p$	$M_S^* = A_p$
20×20	4.96	13.19
40×40	10.99	42.36

Table 2: Cavity flow: the number of GMRES steps in solving (4.7) in solution redistribution with different precondition matrix M_S^*, $Re = 2000$.

by direct computation, where U_m is the mean velocity of inflow. The domain is $\Omega = [-1,5]\times[-1,1]$. At the inflow boundary $x=-1$, $\vec{u}=(1-y^2,0)^T$ with poiseuille profile is imposed. On the top and bottom boundary of the channel, condition $\vec{u}=(0,0)^T$ is settled. Natural condition is imposed on $x=5$.

In our moving strategy, parameters α and β in (4.1) are user defined. The value of α is greater, the degree of mesh clustering is larger. From Figure 6, it can be shown that the number of GMRES iteration step with $\alpha=5$ is larger than $\alpha=1.0$. Comparation of GMRES step counts between different choices of F_p are shown in Figure 6, it is found that the number of GMRES iteration steps will decrease more than 20 by using $F_p = A + W_p^n$.

We show the moving mesh at $t=2s$ in Figure 7. It can be seen that the mesh obviously clusters around the cylinder. As we known, wall street phenomena will occur as time evoluting when the flow has an appropriate viscosity according to [25], just as the mesh shown in Figure 8.

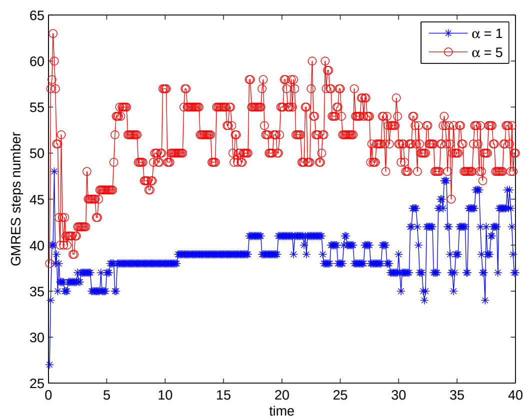


Figure 6: Flow over cylinder: GMRES iteration counts of solving(3.2) using PCD preconditioning with different α, $Re=240$.

linear system	number of GMRES steps			
	$F_p = A + W_p^n$		$F_p = \nu A + W_p^n$	
	$M_S^* = \frac{1}{\nu}A_p$	$M_S^* = A_p$	$M_S^* = \frac{1}{\nu}A_p$	$M_S^* = A_p$
(3.2)	51.06	45.13	0	61.08
(4.7)	22.25	40.35	0	40.78

Table 3: Flow over cylinder: the number of GMRES steps in solving (3.2) and (4.7) with different precondition matrix M_S^* and different F_p in PCD preconditioning, $\alpha=5, Re=240$.

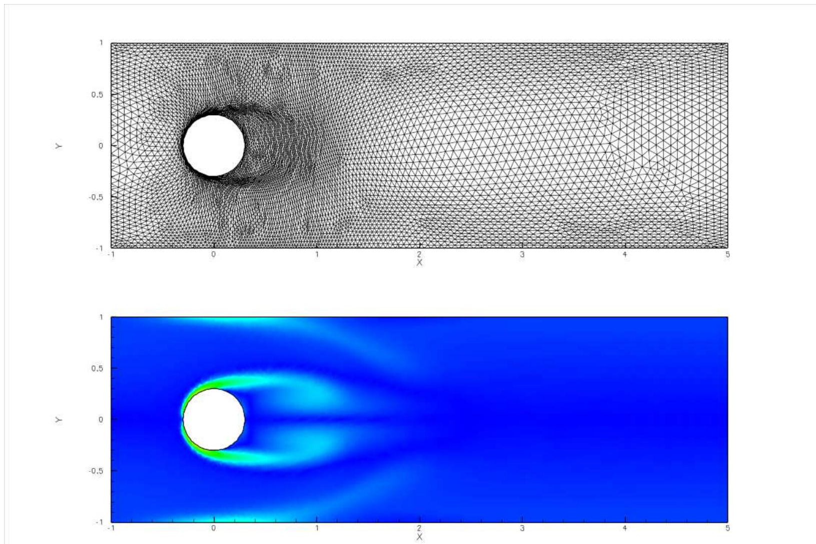


Figure 7: Flow over cylinder: moving mesh at $t = 2s$, $Re = 240$.

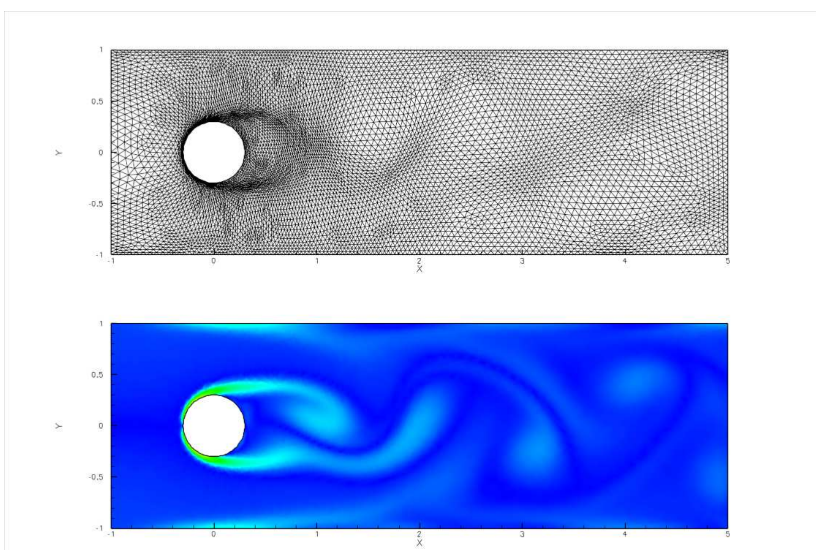


Figure 8: Flow over cylinder: moving mesh at $t = 40s$, $Re = 240$.

6. Remarks

In this work, we apply an efficient AMG preconditioning strategy to moving mesh finite elment method based on $4P1-P1$ pair. The $4P1-P1$ element pair naturally satisfies the inf-sup condition and is linear-order. Linear element is more prefered than high order element in practical engeneering computation, according to its simplicity and complexities of problems. In our moving strategy, we use the monitor function based on vorticity to capture the fine flow structure. The structure of mesh is consistent with vorticity structure. We compare the number of GMRES iteration steps in solving Navier-Stokes problems with different F_p in PCD preconditioning. It is verified that choosing $F_p = A_p + W_p^n$ in PCD preconditioning is more efficient by three numerical tests. We also constrast the number of GMRES step counts of solving linear system in solution re-distribution with different preconditioning matrix M_S^*. It turns out to be that $M_S^* = \frac{1}{\nu}A_p$ is more efficient. We find that the number of GMRES iteration steps will be larger as the mesh becomes clustering.

We will extend the efficient preconditioning to some intrested problems such as free boundary problem. Also three dimention problems of solving incompressible flow with moving mesh finite element based on $4P1-P1$ pair will be considered in future work.

Acknowledgments

The authors' work was supported in part by the National Basic Research Program of China(2011CB309704) and the National Natrual Science Foundation of China(11271358 and 91230108).

References

[1] R. Li, W. Liu, T. Tang, and P. Zhang. Moving mesh finite element methods based on harmonic maps, 2001.

[2] Yana Di, Ruo Li, Tao Tang, and Pingwen Zhang. Moving mesh finite element methods for the incompressible Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 26(3):1036–1056, 2005.

[3] Alan M Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics*, 135(2):128–138, 1966.

[4] Arkady S. Dvinsky. Adaptive grid generation from harmonic maps on Riemannian manifolds. *Journal of Computational Physics*, 95(2):450–476, 1991.

[5] Yana Di, Ruo Li, Tao Tang, and Pingwen Zhang. Moving mesh finite element methods for the incompressible navier–stokes equations. *SIAM Journal on Scientific Computing*, 26(3):1036–1056, 2005.

[6] Wu Yirong and Heyu Wang. Moving mesh finite element method for unsteady navier–stokes flow. *submitted*.

[7] Michel Bercovier and Olivier Pironneau. Error estimates for finite element method solution of the stokes problem in the primitive variables. *Numerische Mathematik*, 33(2):211–224, 1979.

[8] Michele Benzi, Gene H Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta numerica*, 14:1–137, 2005.

[9] Ling Shen and Jinchao Xu. On a schur complement operator arisen from navier-stokes equations and its preconditioning. *LECTURE NOTES IN PURE AND APPLIED MATHEMATICS*, pages 481–490, 1999.

[10] Jinchao Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613, 1992.

[11] Yinnian He. Two-level method based on finite element and crank-nicolson extrapolation for the time-dependent navier-stokes equations. *SIAM Journal on Numerical Analysis*, 41(4):1263–1285, 2003.

[12] Zhong-Zhi Bai and Michael K Ng. On inexact preconditioners for nonsymmetric matrices. *SIAM Journal on Scientific Computing*, 26(5):1710–1724, 2005.

[13] Zhong-Zhi Bai. Structured preconditioners for nonsingular matrices of block two-by-two structures. *Mathematics of Computation*, 75(254):791–815, 2006.

[14] Howard Elman, Victoria E Howle, John Shadid, David Silvester, and Ray Tuminaro. Least squares preconditioners for stabilized discretizations of the navier-stokes equations. *SIAM Journal on Scientific Computing*, 30(1):290–311, 2007.

[15] Howard C Elman and Ray S Tuminaro. Boundary conditions in approximate commutator preconditioners for the navier-stokes equations. *Electronic Transactions on Numerical Analysis*, 35:257–280, 2009.

[16] Michele Benzi and Maxim A Olshanskii. An augmented lagrangian-based approach to the oseen problem. *SIAM Journal on Scientific Computing*, 28(6):2095–2113, 2006.

[17] Michele Benzi, Michael Ng, Qiang Niu, and Zhen Wang. A relaxed dimensional factorization preconditioner for the incompressible navier–stokes equations. *Journal of Computational Physics*, 230(16):6185–6202, 2011.

[18] J Boyle, MD Mihajlovic, and JA Scott. Hsl mi20: an efficient amg preconditioner. Technical report, Citeseer, 2007.

[19] Jonathan Boyle, Milan Mihajlović, and Jennifer Scott. Hsl_mi20: an efficient amg preconditioner for finite element problems in 3d. *International journal for numerical methods in engineering*, 82(1):64–98, 2010.

[20] Howard C Elman, David J Silvester, and Andrew J Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2005.

[21] Ruo Li. On multi-mesh h-adaptive methods. *Journal of Scientific Computing*, 24(3):321–341, 2005.

[22] Howard Elman, Milan Mihajlović, and David Silvester. Fast iterative solvers for buoyancy driven flow problems. *Journal of Computational Physics*, 230(10):3900–3914, 2011.

[23] Ruo Li, Tao Tang, and Pingwen Zhang. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *Journal of Computational Physics*, 177(2):365–393, 2002.

[24] Weiming Cao, Weizhang Huang, and Robert D Russell. Anr-adaptive finite element method based upon moving mesh pdes. *Journal of Computational Physics*, 149(2):221–244, 1999.

[25] Milton Van Dyke. An album of fluid motion. 1982.