Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **9**, No. 3, pp. 742-756

DOI: 10.4208/aamm.2016.m1457
June 2017

Moving Mesh Finite Element Method for Unsteady Navier-Stokes Flow

Yirong Wu and Heyu Wang*

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

Received 19 February 2016; Accepted (in revised version) 19 September 2016

Abstract. In this paper, we use moving mesh finite element method based upon $4P_1-P_1$ element to solve the time-dependent Navier-Stokes equations in 2D. Two-layer nested meshes are used including velocity mesh and pressure mesh, and velocity mesh can be obtained by globally refining pressure mesh. We use hierarchy geometry tree to store the nested meshes. This data structure make convienence for adaptive mesh method and the construction of multigrid preconditioning. Several numerical problems are used to show the effect of moving mesh.

AMS subject classifications: 65M60

Key words: Navier-Stokes system, $4P_1-P_1$, hierarchy geometry tree, moving mesh method.

1 Introduction

It is important that solving incompressible Navier-Stokes equations by mixed finite element method. In order to satisfy the LBB condition, one way is to enhance velocity space relative to pressure, for example Taylor-Hood element. The other is imposing constraint on pressure space, such as stablized P_1-P_1, P_1-P_0 (see [1, 2]). In practical computing, adaptive scheme is often used to decrease the computational cost for efficency. Meanwhile, it can improve the quality of solutions locally see [3]. There are some works using h-adaptive P_2-P_1 element because of simplicity, see (see [4–6]) for detail. However, if domain has some corners or solutions have some singularities, we tend to use lower order approximations in engineering computation. Adaptive method with stabilized P_1-P_1 and P_1-P_0 elements are proposed in [7]. In [8], dual element P_1-P_0 is used for moving mesh method. It has some technical difficulties in applying adaptive mesh based on unstable element pairs. So $P_1\text{iso}P_2P_1$ (see [9–11]) is considered. In [10], it is pointed out that $P_1\text{iso}P_2P_1$ satisfies the LBB condition. Four velocity elements can be obtained by refining

*Corresponding author.
Email: wuwuyiyirongrong@163.com (Y. R. Wu), wang.heyu@gmail.com (H. Y. Wang)

 ©2017 Global Science Press

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

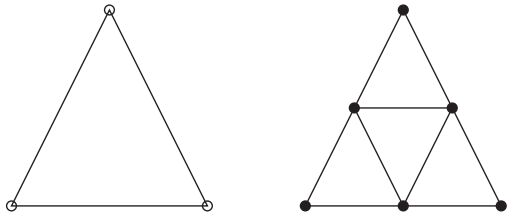


Figure 1: Left: pressure p element, $\circ$ for degrees of p; right: four velocity v elements, $\bullet$ for degrees of v.

the pressure element one time as Fig. 1 shows. Note that $P_1\text{iso}P_2P_1$ element is based on one set of mesh, so basis functions of pressure element are obtained by the interpolation of velocity basis functions. This will increase calculation.

In this paper, we choose $4P_1-P_1$ finite element pair. It has the same structure as $P_1\text{iso}P_2P_1$ pair, so the LBB condition is naturally satisfied. However, The basis functions of both velocity and pressure elements are all standard P_1 element without any extra interpolation. We use the hierarchy geometry tree which was proposed in [12] to store the mesh structure of $4P_1-P_1$ pair.

Moving mesh finite element methods have been developed by a lot of works such as [13–19]. In [16], a moving mesh finite element method based upon harmonic map was proposed. The authors in [8] extended the moving scheme to solve incompressible Navier-Stokes equations. However, the boundary conditions of numerical experiments in [8] are periodic. In this paper, we use moving mesh method based on $4P_1-P_1$ pair and the boundary conditions are general Dirichlet and Neumann boundary conditions.

The layout of the paper is arranged as follows. In Section 2, we introduce the Navier-Stokes governing equations. In Section 3, we illustrate data structure for finite element pair $4P_1-P_1$, In Section 4, $4P_1-P_1$ pair is used to approximate the governing equations. Next, the AMG preconditioner for steady stokes equations is shown. In Section 6, the moving mesh strategy is given. Then we present numerical experiments in Section 7. Finally, we give the conclusions in this section.

2 Governing equation

Following [20], we give non-dimensional incompressible Navier-Stokes equations as follows:

$$\partial_t \mathbf{u} - \frac{1}{Re}\nabla^2\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f}, \tag{2.1a}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{2.1b}$$

The variable $\mathbf{u}$ is velocity and the scalar variable p is pressure. The physical domain is Ω. Reynolds number $Re := \frac{UL}{\nu}$, where L represents a characteristic length scale for Ω, U is the mean velocity of the inflow and $\nu > 0$ is the constant kinematic viscosity. The

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

initial boundary value problem of the system (2.1), with initial and boundary conditions on $\partial\Omega=\partial\Omega_D\bigcup\partial\Omega_N$:

$$\mathbf{u}=\mathbf{w} \qquad \text{on } \partial\Omega_D\times[0,T], \tag{2.2a}$$

$$\frac{1}{Re}\frac{\partial\mathbf{u}}{\partial n}-\mathbf{n}p=\mathbf{0} \qquad \text{on } \partial\Omega_N\times[0,T], \tag{2.2b}$$

$$\mathbf{u}|_{t=0}=\mathbf{u_0} \qquad \text{in } \Omega, \tag{2.2c}$$

where $\mathbf{n}$ denotes the outward normal to the boundary.

3 Illustration of the $4P_1-P_1$ element

From [10], we know that $P_1\text{iso}P_2P_1$ element satisfies the LBB condition. This element pair is based on only one set of mesh for instance velocity mesh as shown in Fig. 2. The velocity space is represented by standard P_1 basis functions, while pressure space is represented by piecewise linear on the larger P_2 element. So the basis function on node 1 of pressure element is $\tilde{N}_1=N_1+(N_a+N_b)/2$. This interpolation will make the assembly of divergent matrix slightly complicated, especially when combining the adaptive mesh method. Whereas $4P_1-P_1$ element is based on two-layer nested meshes including velocity and pressure mesh. So both velocity and pressure spaces are represented by standard P_1 element. Thus the assembly of divergent matrix can be completed only with P_1 element. Moreover, we only pay attention to one set of mesh (pressure mesh) in moving mesh algorithm, provided the index between two-layer nested meshes is built. In the following, we use the hierarchy geometry tree in [12] to store the nested meshes, as Fig. 3 shows.

Notice that two meshes are used, the $1-1$ index between them are needed. By traversing all pressure elements one time, we can acquire the $1-1$ index based on the hierarchy geometry tree as Algorithm 3.1 shown.

Algorithm 3.1. Make index between velocity elements and pressure elements.

1: achieveiterator ← irregualerMeshV.beginActiveElement()

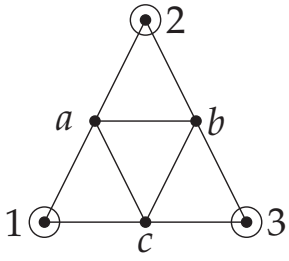
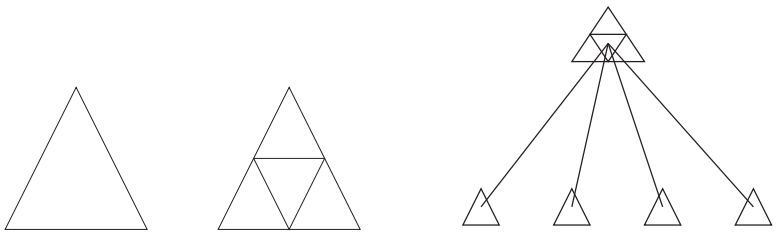


Figure 2: $P_1\text{iso}P_2P_1$ pair, pressure degrees of freedom are only stored in vertices marked by circles.

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press



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

Figure 3: Hierarchy tree structure.

```
2: enditerator ← irregualerMeshV.endActiveElement()
3: while achieveiterator ≠ enditerator do
4:     int index-v-element ← achieveiterator−⟩ index
5:     HElement ⟨ DIM, DIM⟩∗ parent ← activeiterator−⟩ parent
6:     int index-p-element ← parent−⟩ index
7:     int n_child ← parent−⟩ n_child
8:     index-p2v[index-p-element].resize(n_child)
9:     while i ≥ 0 and i < n_child do
10:        HElement⟨DIM,DIM⟩ ∗chi ← parent−⟩child[i]
11:        int index-v-element ← child−⟩index
12:        index-p2v[index-p-element][i] ← index-v-element
13:        index-v2p[index-v-element] ← index-p-element
14:    end while
15: end while
```

4 Finite element approximation

We define the solution and test spaces as

$$\mathbf{H}_E^1 := \left\{ \mathbf{u} \in \mathcal{H}^1(\Omega)^d \middle| \mathbf{u} = \mathbf{w} \text{ on } \partial\Omega_D \right\}, \quad \mathbf{H}_{E_0}^1 := \left\{ \mathbf{v} \in \mathcal{H}^1(\Omega)^d \middle| \mathbf{u} = \mathbf{0} \text{ on } \partial\Omega_D \right\}.$$

Variational formulation of (2.1) is to find $(\mathbf{u}, p) \in \mathbf{H}_E^1 \times L_2(\Omega)$ such that

$$\int_\Omega \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + \frac{1}{Re} \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} + \int_\Omega (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - \int_\Omega p(\nabla \cdot \mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v}, \tag{4.1a}$$

$$\int_\Omega q(\nabla \cdot \mathbf{u}) = 0, \tag{4.1b}$$

for $\forall (\mathbf{v}, q) \in \mathbf{H}_{E_0}^1 \times L_2(\Omega)$, where $\nabla \mathbf{u} : \nabla \mathbf{v} = \nabla v_x \cdot \nabla u_x + \nabla v_y \cdot \nabla u_y$ in 2D.

Assume $\mathcal{T}_H$ is a triangulation of domain Ω for pressure mesh with mesh parameter $H = \max_{T \in \mathcal{T}_H} diam(T)$, $\mathcal{T}_h$ $(h = 1/2H)$ is the triangulation for velocity mesh. Two finite element spaces $X_E^h \subset \mathcal{H}_E$ and $P^H \subset L_2(\Omega)$ are separately on $\mathcal{T}_h$ and $\mathcal{T}_H$.

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

So the discrete weak formulation of (4.1) is: find $(\mathbf{u}_h, p_H) \in X_E^h \times P^H$ such that

$$\int_\Omega \frac{\partial \mathbf{u}_h}{\partial t}\cdot\mathbf{v}_h + \frac{1}{Re}\int_\Omega \nabla\mathbf{u}_h : \nabla\mathbf{v}_h + \int_\Omega (\mathbf{u}_h\cdot\nabla\mathbf{u}_h)\cdot\mathbf{v}_h - \int_\Omega p_H(\nabla\cdot\mathbf{v}_h) = \int_\Omega \mathbf{f}\cdot\mathbf{v}_h, \qquad \forall \mathbf{v}_h \in \mathbf{X}_0^h, \tag{4.2a}$$

$$\int_\Omega q_h(\nabla\cdot\mathbf{u}_h) = 0, \qquad \forall q_h \in P^H. \tag{4.2b}$$

At the time discretization, we use linear backward Euler scheme, which linearizes the nonlinear term $\mathbf{u}_h^{n+1}\cdot\nabla\mathbf{u}_h^{n+1}$ with $\mathbf{u}_h^n\cdot\nabla\mathbf{u}_h^{n+1}$. Let $\{\phi_j\}_{j=1}^n$ and $\{\psi_k\}_{k=1}^m$ be linear basis functions for velocity and pressure. Then $u_{h,x}$, $u_{h,y}$, p_H can be written as follows:

$$u_{h,x} = \sum_{j=1}^n \alpha_{x,j}\phi_j, \quad v_{h,x} = \sum_{j=1}^n \alpha_{y,j}\phi_j, \quad p_H = \sum_{k=1}^m \alpha_{p,k}\psi_k. \tag{4.3}$$

Substituting (4.3) into (4.2), then a linear system

$$\begin{bmatrix} \frac{1}{dt}M + \frac{1}{Re}A + W & 0 & B_x^T \\ 0 & \frac{1}{dt}M + \frac{1}{Re}A + W & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_p \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ g \end{bmatrix}, \tag{4.4}$$

is obtained, where $\alpha_x = (\alpha_{x,1},\cdots,\alpha_{x,n_u})^T$, $\alpha_y = (\alpha_{y,1},\cdots,\alpha_{y,n_u})^T$, $\alpha_p = (\alpha_{p,1},\cdots,\alpha_{p,n_p})^T$, dt is the time step. Mass matrix M, Laplacian matrix A, convection-diffusion matrix W and divergence matrix B_x are defined as following:

$$A = [a_{ij}], \qquad a_{ij} = \int_\Omega \nabla\phi_j\cdot\nabla\phi_i, \tag{4.5a}$$

$$M = [m_{ij}], \qquad m_{ij} = \int_\Omega \phi_j\phi_i, \tag{4.5b}$$

$$W = [w_{ij}], \qquad w_{ij} = \int_\Omega (\mathbf{u}_h^n\cdot\nabla\phi_j)\phi_i, \tag{4.5c}$$

$$B_x = [b_{ij}], \qquad b_{ij} = \int_\Omega \frac{\partial\phi_j}{\partial x}\psi_i. \tag{4.5d}$$

The right hand side terms f_x, f_y, g, separately are

$$f_x = [f_{x,i}], \qquad f_{x,i} = \int_\Omega \left(\frac{u_{h,x}^{(n)}}{dt}\right)\phi_i, \tag{4.6a}$$

$$f_y = [f_{y,i}], \qquad f_{y,i} = \int_\Omega \left(\frac{u_{h,y}^{(n)}}{dt}\right)\phi_i, \tag{4.6b}$$

$$g = [g_k], \qquad g_k = 0. \tag{4.6c}$$

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

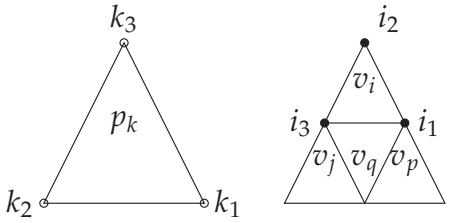


Figure 4: Left: the k-th pressure element p_k; right: four velocity elements corresponding to p_k.

Here, we concentrate on the assembling of divergent matrix. Take B_x^T for instance. Let $\triangle_{v_i}$ be a velocity element, we can find the corresponding pressure element $\triangle_{p_k}$ (see Fig. 4) by the index built in the last section. The local index of basis function is shown. ϕ_{i_1}, ϕ_{i_2}, ϕ_{i_3} are the standard linear triangle basis functions defined on $\triangle_{v_i}$. While ψ_{k_1}, ψ_{k_2}, ψ_{k_3} are defined on $\triangle_{p_k}$. Therefor, the element contribution matrix of B_x^T is

$$\begin{bmatrix} \int_{\triangle_{v_i}} \psi_{k_1} \dfrac{\partial \phi_{i_1}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_2} \dfrac{\partial \phi_{i_1}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_3} \dfrac{\partial \phi_{i_1}}{\partial x} \\ \int_{\triangle_{v_i}} \psi_{k_1} \dfrac{\partial \phi_{i_2}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_2} \dfrac{\partial \phi_{i_2}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_3} \dfrac{\partial \phi_{i_2}}{\partial x} \\ \int_{\triangle_{v_i}} \psi_{k_1} \dfrac{\partial \phi_{i_3}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_2} \dfrac{\partial \phi_{i_3}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_3} \dfrac{\partial \phi_{i_3}}{\partial x} \end{bmatrix}. \tag{4.7}$$

Add the entries of (4.7) to correspondent positions (i_1,k_1), (i_1,k_2), (i_1,k_3), (i_2,k_1), (i_2,k_2), (i_2,k_3), (i_3,k_1), (i_3,k_2), (i_3,k_3) in B_x^T. After traversing all the velocity elements and repeat above action, B_x^T is completely assembled. Similarly, we can assemble matrix B_y^T.

Conversely, by the index of pressure element p_k, we can find the corresponding four small velocity elements v_i, v_j, v_p, v_q, see Fig. 4. B_x can be assembled by the similar process of assembling of B_x^T.

If we get rid of time term and nonlinear $\mathbf{u}\cdot\nabla\mathbf{u}$ in system (2.1), we can get steady Stokes equations. After the same discretization, the linear system equation is as follows

$$\begin{bmatrix} \dfrac{1}{Re}A & 0 & B_x^T \\ 0 & \dfrac{1}{Re}A & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ p \end{bmatrix} = \begin{bmatrix} \hat{f}_x \\ \hat{f}_y \\ \hat{g} \end{bmatrix}. \tag{4.8}$$

Without loss of generality, we set $\hat{f}_x = 0$, $\hat{f}_y = 0$, $\hat{g} = 0$.

5 Precondition strategy

For simplicity, we apply algbraic multigrid preconditioner to the steady Stokes equations. Recall linear system (4.8), we choose a MINRES as the solver. As we know, we should

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

choose an efficient pre-conditioner to improve the efficiency of MINRES. Let $\mathbf{A}$ be the coefficient matrix in (4.8):

$$\mathbf{A}=\begin{bmatrix} \frac{1}{Re}A & 0 & B_x^T \\ 0 & \frac{1}{Re}A & B_y^T \\ B_x & B_y & 0 \end{bmatrix}. \tag{5.1}$$

Block diagonal pre-conditioner $\mathcal{K}$ discussed in [20] is defined as following:

$$\mathcal{K}=\begin{bmatrix} \frac{1}{Re}A & 0 & 0 \\ 0 & \frac{1}{Re}A & 0 \\ 0 & 0 & S \end{bmatrix}, \tag{5.2}$$

where $S=Re(B_xA^{-1}B_x^T+B_yA^{-1}B_y^T)$ is schur complement matrix of $\mathbf{A}$. A good approximation of S will accelerate the speed of solving (4.8). According to [20], pressure mass matrix Q can be a good approximation of S and $\hat{Q}=diag(Q)$ works well in practical computation. Here, we briefly illustrate the implementation of preconditioning. We use a matrix K

$$K=\begin{bmatrix} \frac{1}{Re}A & 0 & 0 \\ 0 & \frac{1}{Re}A & 0 \\ 0 & 0 & \hat{Q} \end{bmatrix}, \tag{5.3}$$

as an approximation of $\mathbf{A}$. In preconditioning, we need to solve $V_{des}=K^{-1}V_{src}$, where the unknown is $V_{des}=(u_d,v_d,p_d)^T$, $V_{src}=(u_s,v_s,p_s)^T$ is given. Firstly, obtain p_d by solving $\hat{Q}p_d=p_s$. Secondly, we have to solve

$$\frac{1}{Re}Au_d=u_s \quad \text{and} \quad \frac{1}{Re}Av_d=v_s$$

to obtain u_d and v_d. Since matrix A is positive-definite, we can use an algbraic multigrid solver for Possion problem to solve $\frac{1}{Re}Au_d=u_s$ and $\frac{1}{Re}Av_d=v_s$.

We solve the time-dependent Navier-Stokes system (4.4) using GMRES method preconditioned with incomplete LU decomposition.

6 Moving mesh stategy

In our moving strategy, we only move pressure mesh and maintain that the velocity mesh is once refined of pressure mesh throughout. At time $t=t_{n+1}$, we can obtain numerical solutions $\mathbf{u}_h^{(n)},p_H^{(n)}$ on old mesh $\mathcal{T}_h^{(n)}$ by scheme mentioned above. We follow the

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

framework in [8] to implement divergence-free interpolation of $(\mathbf{u}_h^{(n)}, p_H^{(n)})$ from $\mathcal{T}_h^n$ to new mesh $\mathcal{T}_h^{(n+1)}$. However common Dirichlet and Neumann boundary conditions are considered instead of periodic boundary conditions. Briefly speaking, our moving mesh strategy mainly contains four steps as follows.

Step 1 Obtain monitor function. Choices of an appropriate monitor function are very important for adaptive scheme. There are some common choices of monitor function G, such as vorticity-based monitor:

$$G_0 = \sqrt{1+\alpha|\omega|^\beta}, \tag{6.1}$$

where $\omega = \nabla \times \mathbf{u}$, α, β are positive constants. In this work, $\beta = 2$ obtains good result. Another selection of G is gradient-based monitor:

$$G_1(\mathbf{u}) = \sqrt{1+\alpha|\nabla \mathbf{u}|^\beta}, \tag{6.2}$$

where $\mathbf{u}$ denotes velocity.

Step 2 Get a new logical mesh. Solve elliptic equation

$$\nabla_{\mathbf{x}}(m\nabla_{\mathbf{x}}\vec{\xi}) = 0, \tag{6.3a}$$

$$\vec{\xi}|_{\partial\Omega} = \vec{\xi}_b, \tag{6.3b}$$

where $m = 1/G$, $\vec{\xi}$ is the coordinates of the logical domain, see [16] for details. Then a new logical mesh is obtained with triangulation $\mathcal{T}_c^*$ and $\mathcal{A}^*$ as its nodes.

Step 3 Mesh-motion in physical domain. Compute the error between $\mathcal{A}^*$ and nodes $\mathcal{A}^0$ of initial logical mesh$\mathcal{T}_c^0$(fixed uniform mesh):

$$\delta\mathcal{A} = \mathcal{A}^0 - \mathcal{A}^*. \tag{6.4}$$

Then the displacement of physical mesh δX is obtained according to $\delta\mathcal{A}$. Finally, a new physical mesh is achieved:

$$X_i^* = X_i + \mu\delta X_i, \tag{6.5}$$

where μ is a constant, see [8] for details.

Step 4 Interpolation of solutions. It is required to maintain divergence-free in the interpolation when implementing moving mesh method to solve incompressible flow. In [8], solution re-distribution on the new mesh $\mathcal{T}^*$ is achieved by solving linear inviscid Navier-Stokes-type equations:

$$\frac{\partial \mathbf{u}}{\partial \tau} - \nabla_{\mathbf{x}}\mathbf{u}\cdot\delta\mathbf{x} = -\nabla p, \tag{6.6a}$$

$$\nabla_{\mathbf{x}}\cdot\mathbf{u} = 0, \tag{6.6b}$$

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

where $\delta\mathbf{x}:=x^{\text{old}}-x^{\text{new}}$, x^{old},x^{new} are two sets of coordinates in physical domain. τ is a virtual time variable and often chosen as 1.0. Here p is just an intermediate variable, different from pressure variable in (2.1).

Weak formulation of (6.6) is: find $(\mathbf{u}_h,p_H)\in X_E^h\times P^H$ such that

$$(\partial_\tau\mathbf{u}_h-\nabla_{\mathbf{x}}\mathbf{u}_h\cdot\delta\mathbf{x},\mathbf{v}_h)=(p_H,\nabla\mathbf{v}_h), \qquad \forall\mathbf{v}_h\in X_E^h, \tag{6.7a}$$

$$(\nabla_{\mathbf{x}}\cdot\mathbf{u},q_H)=0, \qquad \forall q_H\in P^H. \tag{6.7b}$$

In this work, we apply full implicit scheme which is unconditional stable, instead of three-step Runge-Kutta in [8], to (6.7) for time discretization:

$$\left(\frac{u_{h,*}^{(n)}-u_h^{(n)}}{\Delta\tau},\mathbf{v}_h\right)+\left(\delta\mathbf{x}\cdot\nabla\mathbf{u}_{h,*}^{(n)},\mathbf{v}_h\right)=\left(p_{H,*}^{(n)},\nabla\mathbf{v}_h\right), \tag{6.8a}$$

$$\left(\nabla\cdot u_{h,*}^{n},q_H\right)=0, \tag{6.8b}$$

where $\mathbf{u}_h^{(n)}$ and $p_H^{(n)}$ are the numerical solutions of Navier-Stokes equations at $t=t_{n+1}$ using the mesh at $t=t_n$. $u_{h,*}^{(n)}$ and $p_{H,*}^{(n)}$ are the intermediate updated solutions at $t=t_{n+1}$ on the new mesh.

In order to illustrate our scheme, we demonstrate the flow-chart of the Algorithm 6.1.

Algorithm 6.1. Moving mesh FEM for Navier-Stokes equations.

1: Solve steady Stokes system (4.8) on initial velocity mesh $\triangle_v^{(0)}$ and initial pressure mesh $\triangle_p^{(0)}$. Then $\mathbf{u}_h^{(0)}$,$p_H^{(0)}$ are obtained and chosen as the initial values for Navier-Stokes equations.
2: **while** $t_n<T$ **do**
3: Caculate monitor function on $\triangle_p^{(n)}$ using $\mathbf{u}_h^{(n)}$,$p_H^{(n)}$. And obtain logical mesh $\vec{\xi}^*$ by solving (6.3).
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 $\delta\mathbf{x}$ of $\triangle_p^{(n)}$ using the difference of $\vec{\xi}^*-\vec{\xi}^{(0)}$.
6: Solve Eqs. (6.6) on $\triangle_v^{(n)}$ to get intermediate variables $\mathbf{u}_{h,*}^{(n)}$, $p_{h,*}^{(n)}$ on new mesh.
7: Update $\triangle_p^{(n)}$ to $\triangle_p^{(n+1)}$, synchronize $\triangle_v^{(n)}$ to $\triangle_v^{(n+1)}$ by the hierarchy geometry tree.
8: Go back to 3.
9: Solve Navier-Stokes system (4.4) to truly obtain numerical solutions $\mathbf{u}_h^{(n+1)}$, $p_H^{(n+1)}$ on mesh $\triangle_v^{(n+1)}$ and $\triangle_p^{(n+1)}$.
10: **end while**

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

7 Numerical tests

There are three numerical tests here. In practical computing, we use solutions of Stokes equations as the initial values of Navier-Stokes equations. We show our moving mesh and numerical solutions in the following. Our codes are based on the AFEPack (an adaptive finite element package), which can be obtained from `http://dsec.pku.edu.cn/~rli`.

7.1 Accuracy test

We consider the analytic solutions of (2.1):

$$u_x = e^{-2\pi^2 t/Re}\sin(\pi x)\cos(\pi y), \tag{7.1a}$$

$$u_y = e^{-2\pi^2 t/Re}\cos(\pi x)\sin(\pi y), \tag{7.1b}$$

$$p = e^{-4\pi^2 t/Re}\frac{1}{4}[\cos(2\pi x)+\cos(2\pi y)], \tag{7.1c}$$

where Reynolds number $Re=10$. Computational domain is $\Omega=[0,1]\times[0,1]$ and the velocity in (7.1) is interpolated on all boundary of Ω.

We use this example to check the accuracy of our moving mesh method. According to [10], the convergence rate of velocity is second-order and pressure is first order in the uniform mesh computation. Moving mesh algorithm is applied to this example. We choose gradient-based monitor (6.2), where user-defined parameters $\alpha=5.0$, $\beta=2.0$. The computations are started at $t=0$ and evolving forward until $t=0.4$, meanwhile we compute the numerical error:

$$Error_u = \frac{\|u-u_h\|_{L^2}}{\sqrt{N_u}}, \quad Error_p = \frac{\|p-p_h\|_{L^0}}{\sqrt{N_p}}, \tag{7.2}$$

where N_u and N_p separately denote the number of edges of velocity mesh and pressure mesh. The velocity mesh space h on velocity mesh is defined as

$$h = \frac{1}{N_e}, \tag{7.3}$$

where N_e denotes the number of elements of velocity mesh. Note that the pressure mesh space is $2h$. From Fig. 5, it is discoverd that velocity has a second-order accuracy and pressure has at least first-order. Moving mesh and pressure contour are shown in Fig. 6.

7.2 Navier-Stokes flow over a step

We consider the test problem in [21] that the Navier-Stokes flow over a step with $\mathrm{Re}=1000$. The domain is $\Omega=(0,4)\times(0,1)/(1.2,1.6)\times(0,0.4)$, and on the upper and bottom boudaries, Dirichlet condition $\mathbf{u}=(0,0)^T$ is imposed. Outflow boundary is enforced natural boundary condition meanwhile at the inflow boundary $\mathbf{u}=(4y(1-y),0)$.

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

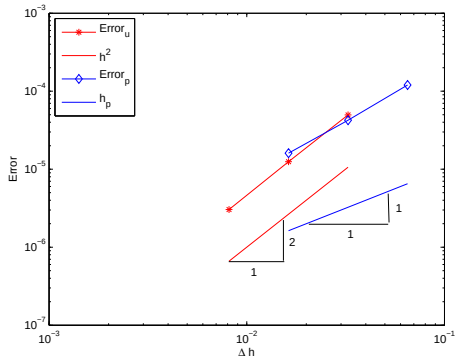


Figure 5: Accuracy test: The convergence order of $Error_u$ and $Error_p$ with mesh refinement.

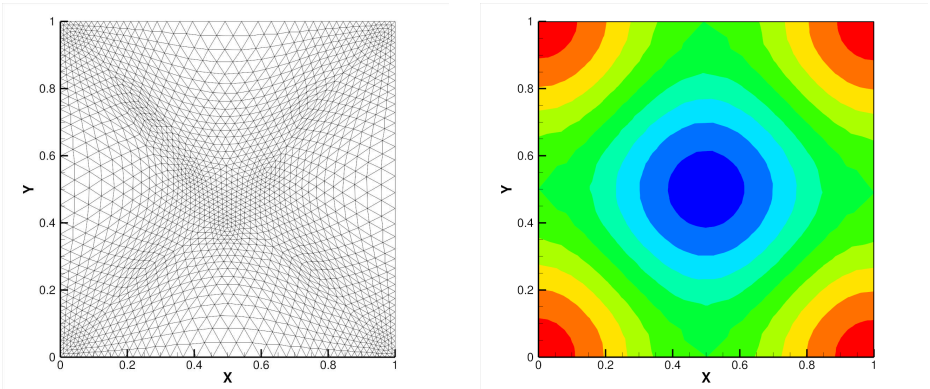


Figure 6: Accuracy test: Moving mesh (a), pressure contour (b).

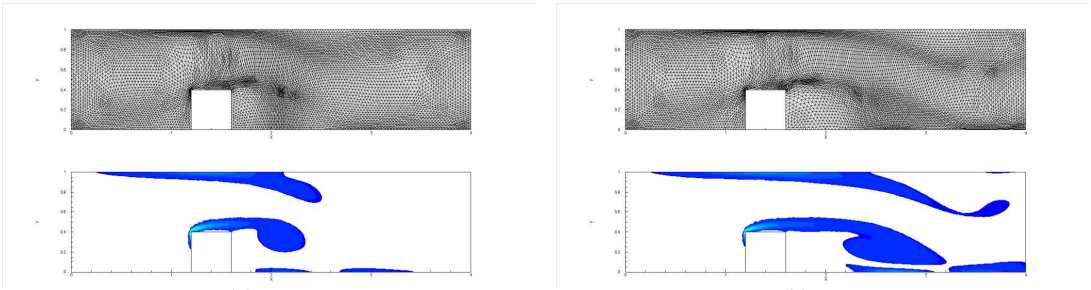
Vorticity-based monitor (6.1) is selected as monitor with α=0.4, β=2.0. As we known, singularities arise due to the concave corner in this problem. The development of moving mesh and vorticity contour as time evolving are shown in Figs. 7 and 8. It can be observed that mesh clusters near the top concave edge and moving mesh is consistent with the structure of vorticity.

In Fig. 9, the comparation of pressure contour with uniform mesh and moving mesh at t=20 are shown. In the uniform case, numerical dissipation is introduced. While better representation of pressure contour with less dissipation is obtained in the moving case.

7.3 Flow over cylinder

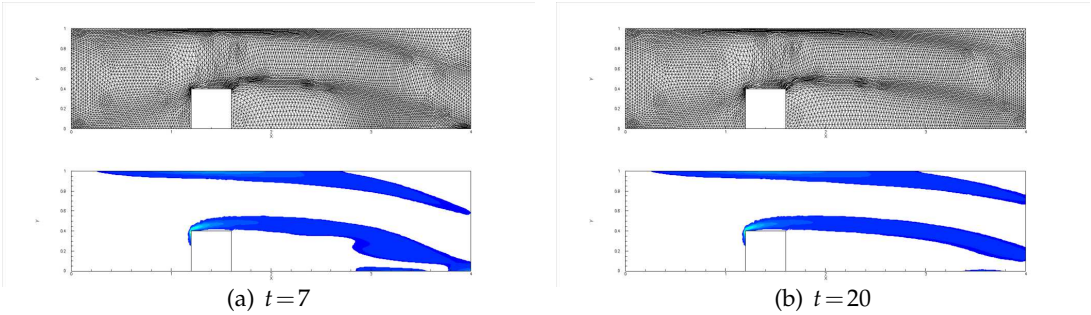
This example models the development of flow over an cylinder along a retangular channel. In [15], the authors apply moving finite element method to this problem. The center of cylinder is $(0,0)$ and radius is $r=0.3$. Let Reynolds number $Re=300$ and domain

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press



(a) $t=1$ (b) $t=3$

Figure 7: Top: moving mesh; bottom: contour of vorticity.



(a) $t=7$ (b) $t=20$

Figure 8: Top: moving mesh; bottom: contour of vorticity.

$\Omega=[-1,5]\times[-1,1]$. Poiseuille flow $u=1-y^2$, $v=0$ is imposed on inflow boundary $x=-1$. $u=v=0$ is imposed on the top and bottom of the channel. Outflow boundary $x=5$ is natrual condition.

If we concern the fine flow structure, it is required high resolution for small scale structure. For detecting the vorticity, (6.1) is a good choice as the monitor in our moving strategy. The parameters α and β are 1.0, 2.0.

In Fig. 10, initial moving mesh is shown clustering near the circular cylinder where need high resolution. The evolution of moving mesh and vorticity contour are illustrated in Fig. 11. It can be discovered that our moving mesh can efficiently capture the vorticity structure and our mesh is more concentrated than [15]. Velocity streamline is shown in Fig. 12.

8 Remarks

In this paper, unsteady Navier-Stokes equations have been numerically solved using moving mesh method on $4P_1-P_1$ element. Hierarchy geometry tree is used to store the mesh structure of $4P_1-P_1$ element. This pair naturally satisfies LBB condition and it's reference element has only P_1 element.

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

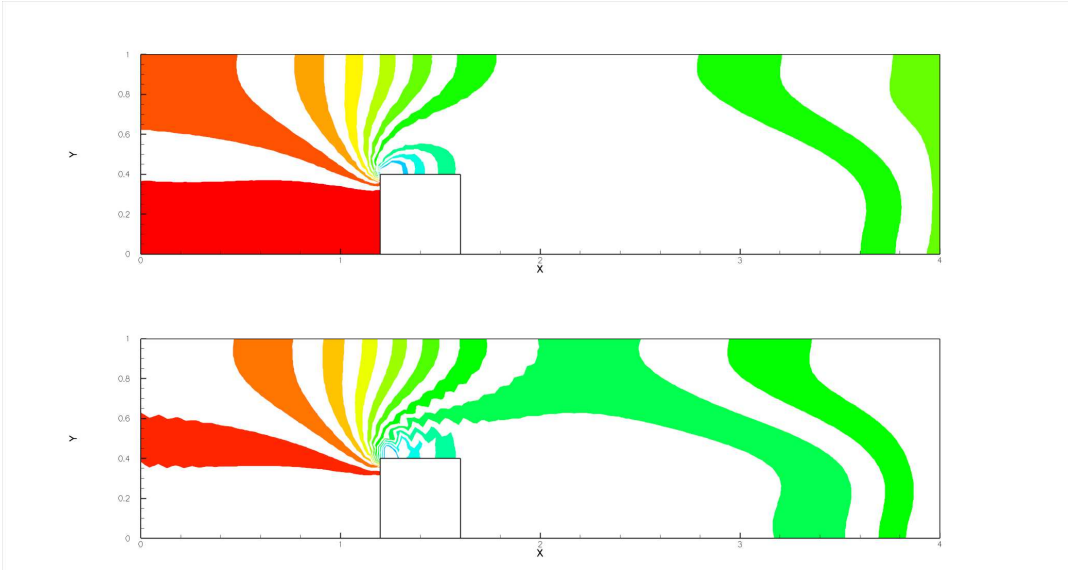


Figure 9: Pressure contour: moving mesh (top); uniform mesh (bottom) at $t=20$.

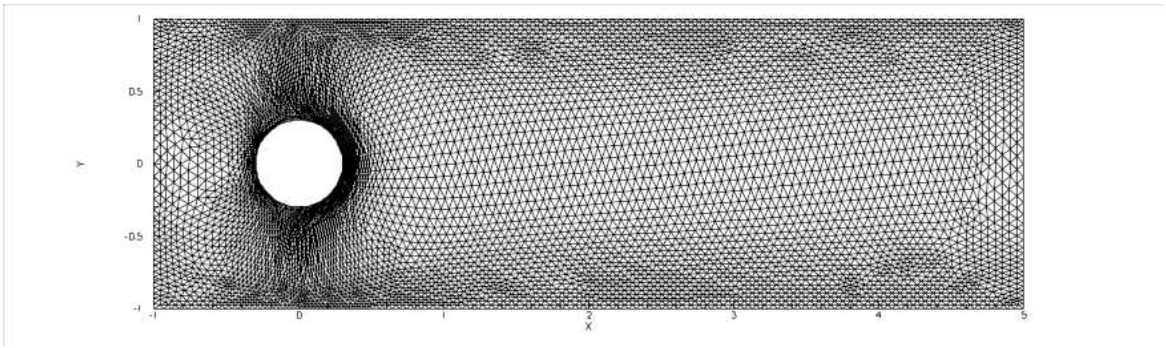


Figure 10: Initial moving mesh, $Re=300$.

Hierarchy geometry tree structure makes convienence for multigrid preconditioning. However, the preconditioning strategy isn't implemented for Navier-Stokes system in this work. But it will appear in future work. Meanwhile, this tree structure is originally used for h-adaptive mesh. So the combining moving mesh with h-adaptive mesh by making use of this tree structure will be also considered.

Acknowledgments

The authors' work was supported in part by the National Basic Research Program of China (No. 2011CB309704) and the National Natural Science Foundation of China (Nos. 11271358 and 91230108).

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

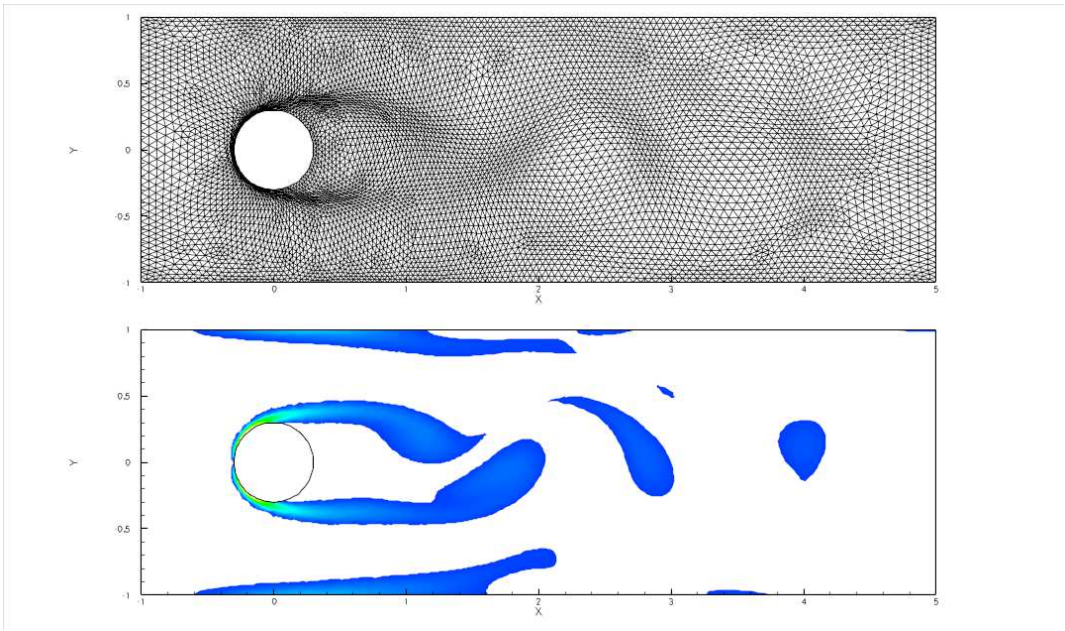


Figure 11: Top: moving mesh; bottom: vorticity contour at $t=24.5$, $Re=300$.

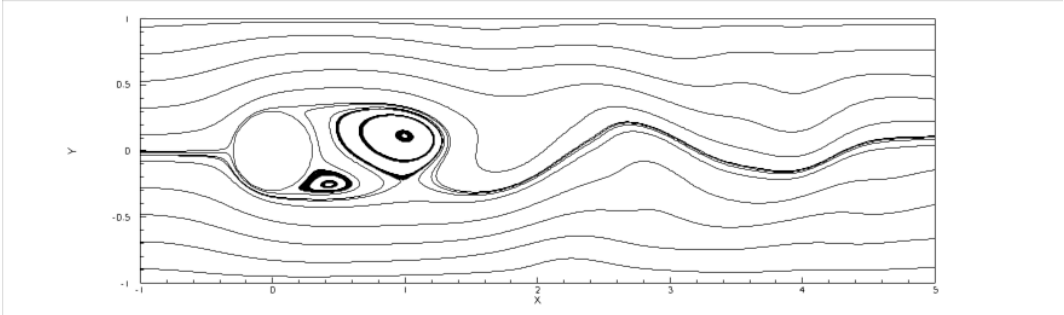


Figure 12: Top: vorticity streamline at $t=24.5$, $Re=300$.

References

[1] J. Li, Y. He and Z. Chen, *Performance of several stabilized finite element methods for the Stokes equations based on the lowest equal-order pairs,* Computing, 86 (2009), pp. 37–51.

[2] P. B. Bochev, C. R. Dohrmann and M. D. Gunzburger, *Stabilization of low-order mixed finite elements for the stokes equations,* SIAM J. Numer. Anal., 44 (2006), pp. 82–101.

[3] T. L. Popiolek and A. M. Awruch, *Numerical simulation of incompressible flows using adaptive unstructured meshes and the pseudo-compressibility hypothesis,* Adv. Eng. Software, 37 (2006), pp. 260–274.

[4] I. Danaila, F. Hecht and S. L. Masson, *A newton method with adaptive finite elements for solving phase-change problems with natural convection,* J. Comput. Phys., 274 (2014), pp. 826–

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press

840.

[5] M. S. EBEDIA, R. L. DAVIS AND R. W. FREUND, *Unsteady incompressible flow simulation using galerkin finite elements with spatial/temporal adaptation,* in 47th AIAA Aerospace Sciences Meeting, Orlando, FL, 2009.

[6] S. BERRONE AND M. MARRO, *Space-time adaptive simulations for unsteady navier-stokes problems,* Comput. Fluids, 38 (2009), pp. 1132–1144.

[7] H. ZHENG, Y. HOU AND F. SHI, *A posteriori error estimates of stabilization of low-order mixed finite elements for incompressible flow,* SIAM J. Sci. Comput., 32 (2010), pp. 1346–1360.

[8] Y. DI, R. LI, T. TANG AND P. ZHANG, *Moving mesh finite element methods for the incompressible navier-stokes equations,* SIAM J. Sci. Comput., 26 (2005), pp. 1036–1056.

[9] S. FUJIMA, *Iso-p2p1/p1/p1 domain-decomposition/finite-element method for the navier-stokes equations,* Contemp. Math., 218 (1998), pp. 246–253.

[10] M. BERCOVIER AND O. PIRONNEAU, *Error estimates for finite element method solution of the stokes problem in the primitive variables,* Numer. Math., 33 (1979), pp. 211–224.

[11] D. VANDEN-ABEELE, D. SNYDER, Y. DETANDT AND G. DEGREZ, *A kinetic energy-preserving p1isop2/p1 finite-element method for computing unsteady incompressible flows,* Comput. Fluid Dyn., (2006), pp. 267–272.

[12] R. LI, *On multi-mesh h-adaptive methods,* J. Sci. Comput., 24 (2005), pp. 321–341.

[13] A. M. WINSLOW, *Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh,* J. Comput. Phys., 135 (1966), pp. 128–138.

[14] A. S. DVINSKY, *Adaptive grid generation from harmonic maps on Riemannian manifolds,* J. Comput. Phys., 95 (1991), pp. 450–476.

[15] W. CAO, W. HUANG AND R. D. RUSSELL, *An r-adaptive finite element method based upon moving mesh PDEs,* J. Comput. Phys., 149 (1999), pp. 221–244.

[16] R. LI, W. LIU, T. TANG AND P. ZHANG, *Moving mesh finite element methods based on harmonic maps,* J. Comput. Phys., 170 (2001), pp. 562–588.

[17] J. LANG, W. CAO, W. HUANG AND R. D. RUSSELL, *A two-dimensional moving finite element method with local refinement based on a posteriori error estimates,* Appl. Numer. Math., 46 (2003), pp. 75–94.

[18] W. HUANG AND R. D. RUSSEL, Adaptive Moving Mesh Methods, Springer Science & Business Media, 2010.

[19] R. D. HAYNES, W. HUANG AND P. A. ZEGELING, *A numerical study of blowup in the harmonic map heat flow using the mmpde moving mesh method,* Numer. Math. Theory Methods Appl., 6 (2013), pp. 364–383.

[20] H. C. ELMAN, D. J. SILVESTER AND A. 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] H. ZHENG, Y. HOU AND F. SHI, *Adaptive variational multiscale methods for incompressible flow based on two local gauss integrations,* J. Sci. Comput., 229 (2010), pp. 7030–7041.

https://doi.org/10.4208/aamm.2016.m1457 Published online by Cambridge University Press