Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **xx**, No. x, pp. xx-xx

DOI: xxxx
xxx 2016

# Moving mesh finite element method for unsteady Navier-Stokes flow

Yirong Wu and Heyu Wang*

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

**Abstract.** In this paper, we use moving mesh fintie 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. Seveval numerical problems are used to show the effect of moving mesh.

**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 $P_1 - P_1$, $P_1 - P_0$( [1], [2]). In practical computing, adaptive scheme is often used to decrease the computitional 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 ( [4], [5], [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 stablized $P_1 - P_1$ 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 $P_1\text{iso}P_2P_1$ ( [9], [10], [11]) is considered. In [10], it is pointed out that $P_1 isoP_2P_1$ satisfies the LBB condition. Four velocity elements can be obtained by refining the pressure element one time as Figure 1 shows. Note that $P_1\text{iso}P_2P_1$ element is based on one set of mesh, so basis functions

*Corresponding author.
*Email:* wuwuyiyirongrong@163.com (Yirong Wu), wang.heyu@gmail.com (Heyu Wang)

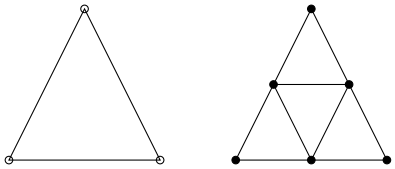


Figure 1: Left: pressure $p$ element, $\circ$ for degrees of $p$; right: four velocity $v$ elements, $\bullet$ for degrees of $v$.

of pressure element are obtained by the interpolation of velocity basis functions. This will increase caculation.

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 natrually 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], [14], [15], [16]. 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 arraged 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 experiements in section 7. Finaly, we give the conclusions in this section.

## 2 Governing Equation

Following [17], we give nondimentional incompressible Navier-Stokes equations as follows:

$$\begin{aligned} \partial_t \mathbf{u} - \tfrac{1}{Re}\nabla^2 \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f}, \\ \nabla \cdot \mathbf{u} &= 0. \end{aligned} \tag{2.1}$$

The variable $\mathbf{u}$ is velocity and the scalar variable $p$ is pressure. The phisical domain is $\Omega$. Reynolds number $Re := \frac{UL}{\nu}$, where $L$ represents a characteristic length scale for $\Omega$, $U$ is the mean velocity of the inflow and $\nu > 0$ is the constant kinematic viscosity. The initial boundary value problem of the system (2.1), with initial and boundary

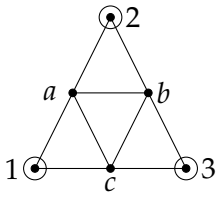



Figure 2: $P_1\text{iso}P_2P_1$ pair, pressure degrees of freedom are only stored in vertices marked by circles.

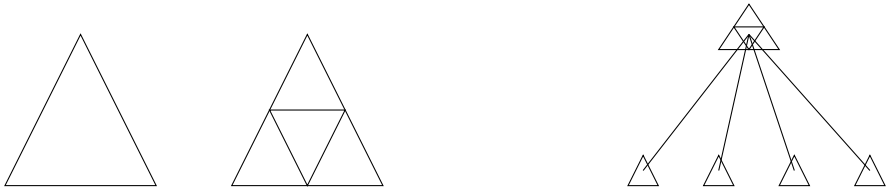
conditions on $\partial\Omega = \partial\Omega_D \bigcup \partial\Omega_N$:

$$\begin{aligned} &\mathbf{u} = \mathbf{w}, && \text{on}\partial\Omega_D \times [0,T] \\ &\tfrac{1}{Re}\frac{\partial \mathbf{u}}{\partial n} - \mathbf{n}p = \mathbf{0}, && \text{on } \partial\Omega_N \times [0,T], \\ &\mathbf{u}|_{t=0} = \mathbf{u_0}, && \text{in}\Omega, \end{aligned} \tag{2.2}$$

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 Figure 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 caculation complicated. Whereas $4P_1 - P_1$ element is based on two nested meshes including velocity and pressure mesh. So both velocity and pressure spaces are represented by standard $P_1$ element. This will simplify matrix assembly. We use the hierarchy geometry tree in [12] to store the nested meshes, as Figure 3 shows.



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

Figure 3: Hierarchy tree structure

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. After the index built, we can just only use $P_1$ element to assemble divergence matrix for velocity and pressure.

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

```
achieveiterator ← irregualerMeshV.beginActiveElement()
enditerator ← irregualerMeshV.endActiveElement()
while achieveiterator ≠ enditerator do
    int index-v-element ← achieveiterator−⟩ index
    HElement ⟨ DIM, DIM⟩∗ parent ← activeiterator−⟩ parent
    int index-p-element ← parent−⟩ index
    int n_child ← parent−⟩ n_child
    index-p2v[index-p-element].resize(n_child)
    while i ≥ 0 and i < n_child do
        HElement⟨DIM,DIM⟩ ∗chi ← parent−⟩child[i]
        int index-v-element ← child−⟩index
        index-p2v[index-p-element][i] ← index-v-element
        index-v2p[index-v-element] ← index-p-element
    end while
end while
```

# 4 Finite Element Approximation

We define the solution and test spaces as

$$\mathbf{H}_E^1 := \left\{ \mathbf{u} \in \mathcal{H}^1(\Omega)^d \middle| \mathbf{u} = \mathbf{w} \text{ on } \partial\Omega_D \right\}, \tag{4.1}$$

$$\mathbf{H}_{E_0}^1 := \left\{ \mathbf{v} \in \mathcal{H}^1(\Omega)^d \middle| \mathbf{u} = \mathbf{0} \text{ on } \partial\Omega_D \right\}. \tag{4.2}$$

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

$$\int_\Omega \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + \frac{1}{Re} \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} + \int_\Omega (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - \int_\Omega p \, (\nabla \cdot \mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v}, \tag{4.3}$$

$$\int_\Omega q \, (\nabla \cdot \mathbf{u}) = 0, \tag{4.4}$$

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 $\Omega$ 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 seperatly on $\mathcal{T}_h$ and $\mathcal{T}_H$.

So the discreted weak formulation of (4.4) is: find $(\mathbf{u}_h, p_H) \in X_E^h \times P^H$ such that

$$\begin{aligned} &\int_\Omega \frac{\partial \mathbf{u}_h}{\partial t} \cdot \mathbf{v}_h + \frac{1}{Re} \int_\Omega \nabla \mathbf{u}_h : \nabla \mathbf{v}_h \\ &+ \int_\Omega (\mathbf{u}_h \cdot \nabla \mathbf{u}_h) \cdot \mathbf{v}_h - \int_\Omega p_H \, (\nabla \cdot \mathbf{v}_h) = \int_\Omega \mathbf{f} \cdot \mathbf{v}_h, \qquad \forall \mathbf{v}_h \in \mathbf{X}_0^h, \\ &\int_\Omega q_h \, (\nabla \cdot \mathbf{u}_h) = 0, \qquad \forall q_h \in P^H. \end{aligned} \tag{4.5}$$

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:

$$u_{h,x} = \sum_{j=1}^{n} \alpha_{x,j}\phi_j, \quad v_{h,x} = \sum_{j=1}^{n} \alpha_{y,j}\phi_j, \quad p_H = \sum_{k=1}^{m} \alpha_{p,k}\psi_k. \tag{4.6}$$

Substituting (4.6) into (4.5), then a linear system

$$\begin{bmatrix} \frac{1}{dt}M + \frac{1}{Re}A + W & 0 & B_x^T \\ 0 & \frac{1}{dt}M + \frac{1}{Re}A + W & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_p \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ g \end{bmatrix}, \tag{4.7}$$

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:

$$\begin{aligned} A = [a_{ij}], \qquad & a_{ij} = \int_\Omega \nabla\phi_j \cdot \nabla\phi_i, \\ M = [m_{ij}], \qquad & m_{ij} = \int_\Omega \phi_j\phi_i, \\ W = [w_{ij}], \quad & w_{ij} = \int_\Omega (\mathbf{u}_h^n \cdot \nabla\phi_j)\phi_i, \\ B_x = [b_{ij}], \qquad & b_{ij} = \int_\Omega \frac{\partial\phi_j}{\partial x}\psi_i. \end{aligned} \tag{4.8}$$

The right hand side terms $f_x, f_y, g$ seperatly are

$$\begin{aligned} f_x = [f_{x,i}], \quad & f_{x,i} = \int_\Omega (\frac{u_{h,x}^{(n)}}{dt})\phi_i, \\ f_y = [f_{y,i}], \quad & f_{y,i} = \int_\Omega (\frac{u_{h,y}^{(n)}}{dt})\phi_i, \\ g = [g_k], \qquad & g_k = 0. \end{aligned} \tag{4.9}$$

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 Figure 4) by the index built in the last section. The local index of basis function is shown. $\phi_{i_1}, \phi_{i_2}, \phi_{i_3}$ are the standard linear triangle basis functions defined on $\triangle_{v_i}$. While $\psi_{k_1}, \psi_{k_2}, \psi_{k_3}$ are defined on $\triangle_{p_k}$. Therefor, the element contribution matrix of $B_x^T$ is

$$\begin{bmatrix} \int_{\triangle_{v_i}} \psi_{k_1}\frac{\partial\phi_{i_1}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_2}\frac{\partial\phi_{i_1}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_3}\frac{\partial\phi_{i_1}}{\partial x} \\ \int_{\triangle_{v_i}} \psi_{k_1}\frac{\partial\phi_{i_2}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_2}\frac{\partial\phi_{i_2}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_3}\frac{\partial\phi_{i_2}}{\partial x} \\ \int_{\triangle_{v_i}} \psi_{k_1}\frac{\partial\phi_{i_3}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_2}\frac{\partial\phi_{i_3}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_3}\frac{\partial\phi_{i_3}}{\partial x} \end{bmatrix} \tag{4.10}$$

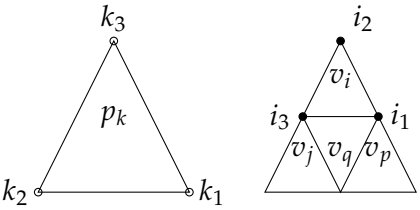



Figure 4: Left: the k-th pressure element $p_k$; right: four velocity elements corresponding to $p_k$.

Add the entries of (4.10) 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 $B_y^T$.

Conversly, 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 Figure 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

$$\begin{bmatrix} \frac{1}{Re}A & 0 & B_x^T \\ 0 & \frac{1}{Re}A & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ p \end{bmatrix} = \begin{bmatrix} \hat{f}_x \\ \hat{f}_y \\ \hat{g} \end{bmatrix}. \tag{4.11}$$

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 (4.11), we choose a MINRES as the solver. As we know, we should choose an efficient preconditioner to improve the efficency of MINRES. Let $\mathbf{A}$ be the coefficient matrix in (4.11):

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

Block diagonal preconditioner $\mathcal{K}$ discussed in [17] 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 $S = Re(B_x A^{-1} B_x^T + B_y A^{-1} B_y^T)$ is schur complement matrix of $\mathbf{A}$. A good approximation of $S$ will accelerate the speed of solving (4.11). According to [17], pressure mass matrix $Q$ can be an 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 $\frac{1}{Re}Au_d = u_s$ and $\frac{1}{Re}Av_d = v_s$ 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 (4.7) using GMRES method preconditioned with incomplete LU decomposition.

## 6 Moving mesh Stategy

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 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 $\omega = \nabla \times \mathbf{u}$, $\alpha, \beta$ 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

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

$$\vec{\xi}|_{\partial\Omega} = \vec{\xi}_b, \tag{6.4}$$

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):

$$\delta\mathcal{A} = \mathcal{A}^0 - \mathcal{A}^*. \tag{6.5}$$

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

$$X_i^* = X_i + \mu\delta X_i, \tag{6.6}$$

where $\mu$ 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 lineard inviscid Navier-Stokes-type equations:

$$\frac{\partial \mathbf{u}}{\partial \tau} - \nabla_{\mathbf{x}}\mathbf{u}\cdot\delta\mathbf{x} = -\nabla p, \tag{6.7}$$

$$\nabla_{\mathbf{x}}\cdot\mathbf{u} = 0, \tag{6.8}$$

where $\delta\mathbf{x} := x^{\text{old}} - x^{\text{new}}$, $x^{\text{old}}, x^{\text{new}}$ are two sets of coordinates in physical domain. $\tau$ 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 (6.8) is: find $(\mathbf{u}_h, p_H) \in X_E^h \times P^H$ such that

$$\begin{aligned} (\partial_\tau \mathbf{u}_h - \nabla_{\mathbf{x}}\mathbf{u}_h\cdot\delta\mathbf{x}, \mathbf{v}_h) &= (p_H, \nabla\mathbf{v}_h), \qquad \forall \mathbf{v}_h \in X_E^h, \\ (\nabla_{\mathbf{x}}\cdot\mathbf{u}, q_H) &= 0, \qquad \forall q_H \in P^H, \end{aligned} \tag{6.9}$$

In this work, we apply full implicity scheme which is unconditional stable, instead of three-step Runge-Kutta in [8], to (6.9) for time discretization:

$$\begin{aligned} \left(\frac{u_{h,*}^{(n)} - u_h^{(n)}}{\triangle\tau}, \mathbf{v}_h\right) + \left(\delta\mathbf{x}\cdot\nabla\mathbf{u}_{h,*}^{(n)}, \mathbf{v}_h\right) &= \left(p_{H,*}^{(n)}, \nabla\mathbf{v}_h\right), \\ \left(\nabla\cdot u_{h,*}^{n}, q_H\right) &= 0. \end{aligned} \tag{6.10}$$

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(4.11) 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 initital 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 (6.4).
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 equations (6.8) 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 (4.7) to truely 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**

# 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):

$$\begin{aligned} u_x &= e^{-2\pi^2 t/Re} sin(\pi x) cos(\pi y), \\ u_y &= e^{-2\pi^2 t/Re} cos(\pi x) sin(\pi y), \\ p &= e^{-4\pi^2 t/Re} \frac{1}{4} \left[ cos(2\pi x) + cos(2\pi y) \right], \end{aligned} \tag{7.1}$$

where Reynolds number $Re = 10$. Computational domain is $\Omega = [0,1] \times [0,1]$ and the velocity in (7.2) is interpolated on all boundary of $\Omega$.

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 $N_p$ seperatly 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 Figure 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 Figure 6.

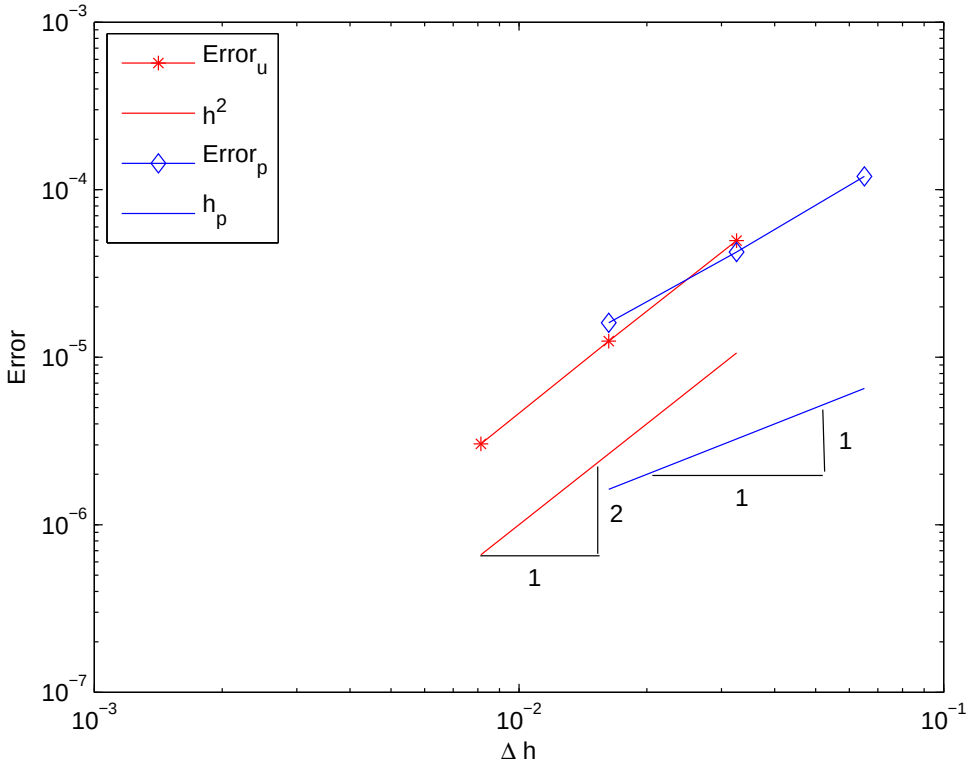



Figure 5: Accuracy test: The convergence order of $Error_u$ and $Error_p$ with mesh refinement.

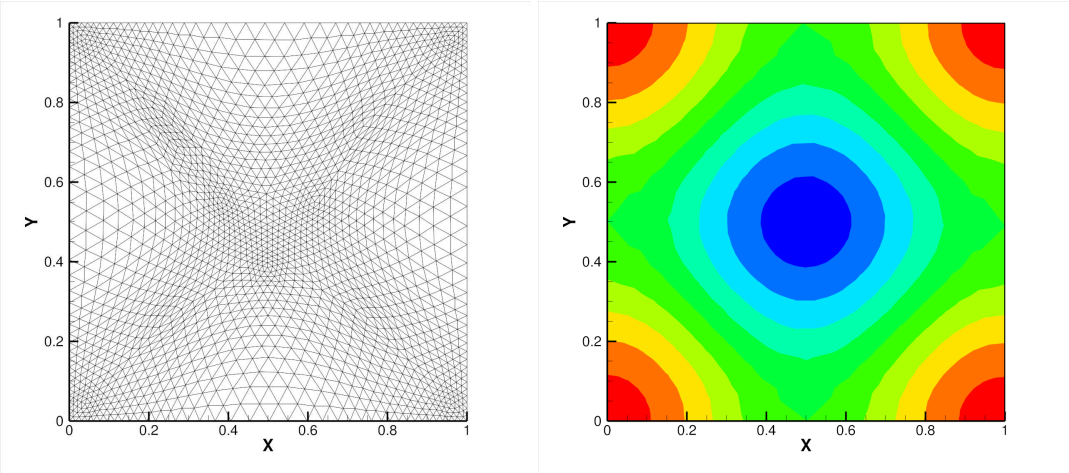


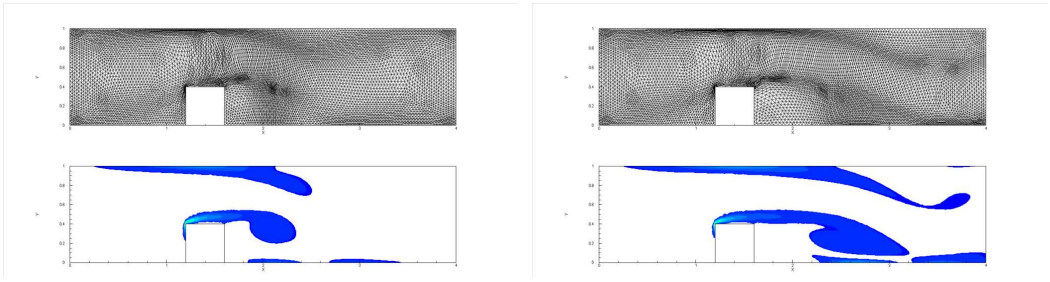

Figure 6: Accuracy test: Moving mesh (left), pressure contour (right).

## 7.2 Navier-Stokes flow over a step

We consider the test problem in [18] that the Navier-Stokes flow over a step with 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 $\mathbf{u} = (4y(1-y),0)$.

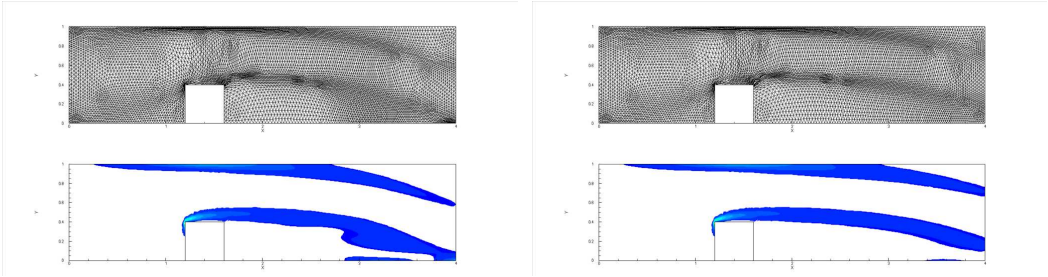
Vorticity-based monitor (6.1) is selected as monitor with $\alpha = 0.4, \beta = 2.0$. As we known, singularities arise due to the concave corner in this problem. The development of moving mesh and vorticity contour as time evolving are show in Figure 7 and 8. It can be observed that mesh clusters near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

In Figure 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.



(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.

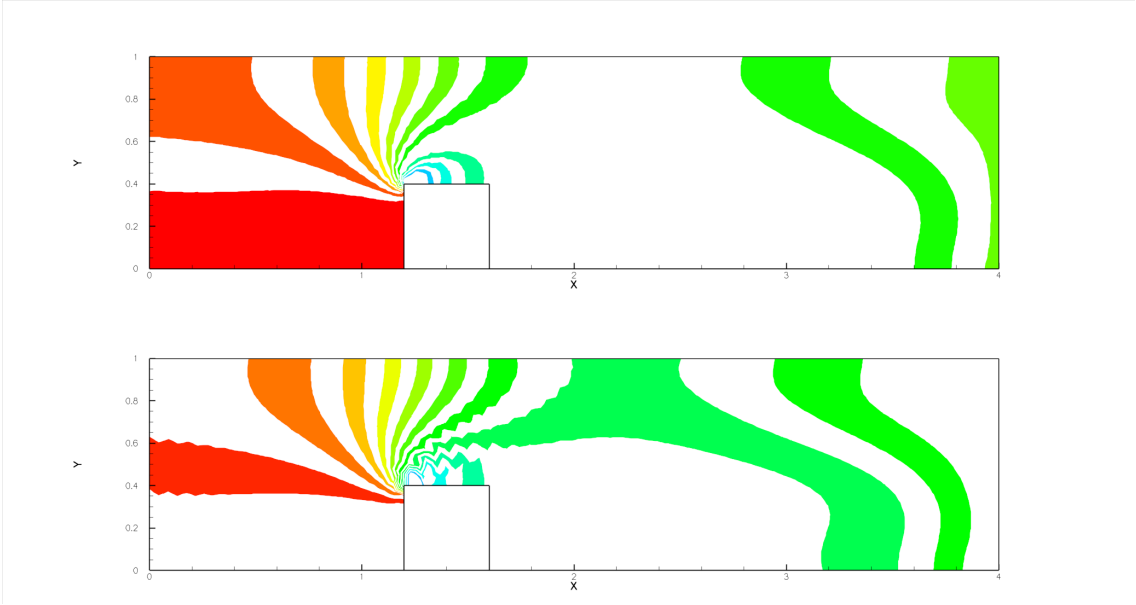


Figure 9: Pressure contour: moving mesh (top); uniform mesh (bottom) at t = 20.

### 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 $\Omega = [-1,5] \times [-1,1]$. 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 $\alpha$ and $\beta$ are 1.0, 2.0.

In Figure 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 Figure 11. It can be discoverd that our moving mesh can efficiently capture the vorticity structure and our mesh is more concentrated than [15]. Velocity streamline is shown in Figure 12.

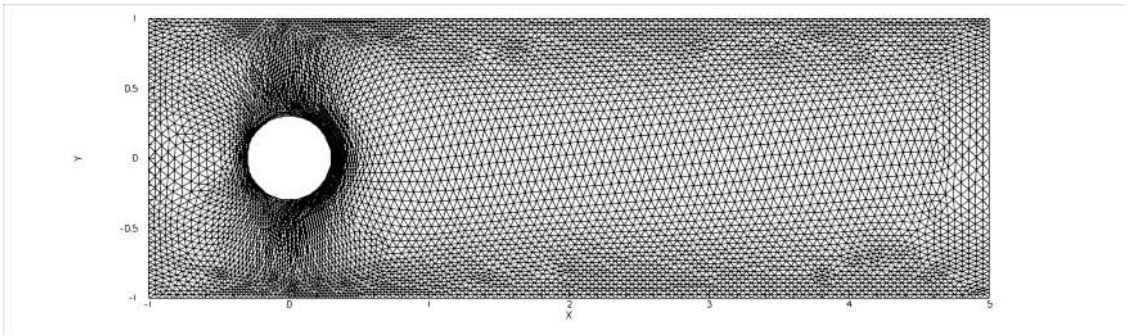


Figure 10: Initial moving mesh, $Re = 300$.

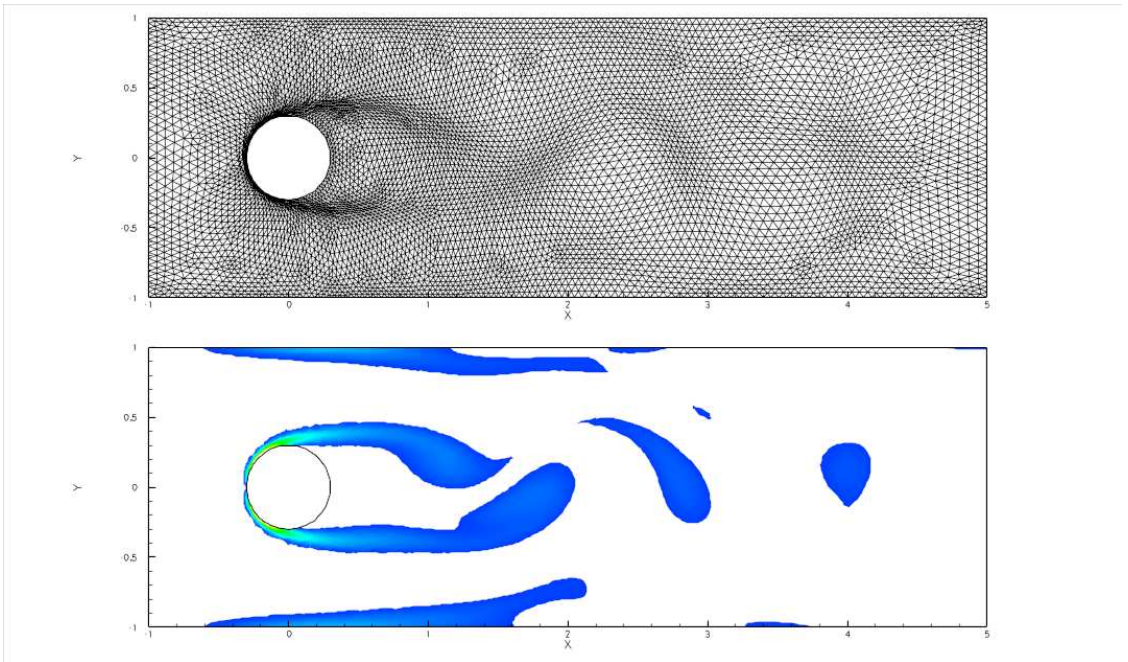


Figure 11: Top: moving mesh; bottom: vorticity contour at $t = 24.5$, $Re = 300$.

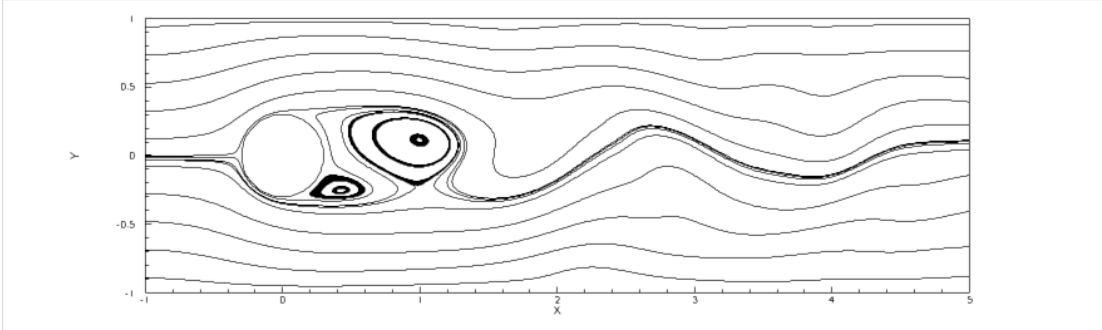



Figure 12: Top: vorticity streamline at $t = 24.5$, $Re = 300$.

## 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 natrually satisfies LBB condition and it's reference element has only $P_1$ element.

Hierarchy geometry tree structure make 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(2011CB309704) and the National Natrual Science Foundation of China(11271358 and 91230108).

## References

[1] Jian Li, Yinnian He, and Zhangxin Chen. Performance of several stabilized finite element methods for the stokes equations based on the lowest equal-order pairs. *Computing*, 86(1):37–51, 2009.

[2] Pavel B Bochev, Clark R Dohrmann, and Max D Gunzburger. Stabilization of low-order mixed finite elements for the stokes equations. *SIAM Journal on Numerical Analysis*, 44(1):82–101, 2006.

[3] TL Popiolek and AM Awruch. Numerical simulation of incompressible flows using adaptive unstructured meshes and the pseudo-compressibility hypothesis. *Advances in Engineering Software*, 37(4):260–274, 2006.

[4] Ionut Danaila, Raluca Moglan, Frédéric Hecht, and Stéphane Le Masson. A newton method with adaptive finite elements for solving phase-change problems with natural convection. *Journal of Computational Physics*, 274:826–840, 2014.

[5] Mohamed S Ebeida, Roger L Davis, and Roland W Freund. Unsteady incompressible flow simulation using galerkin finite elements with spatial/temporal adaptation. In *47th AIAA Aerospace Sciences Meeting, Orlando, FL*, 2009.

[6] Stefano Berrone and Massimo Marro. Space–time adaptive simulations for unsteady navier–stokes problems. *Computers & Fluids*, 38(6):1132–1144, 2009.

[7] Haibiao Zheng, Yanren Hou, and Feng Shi. A posteriori error estimates of stabilization of low-order mixed finite elements for incompressible flow. *SIAM Journal on Scientific Computing*, 32(3):1346–1360, 2010.

[8] 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.

[9] Shoichi Fujima. Iso-p2 p1/p1/p1 domain-decomposition/finite-element method for the navier-stokes equations. *Contemporary Mathematics*, 218:246–253, 1998.

[10] 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.

[11] David Vanden-Abeele, Deryl Snyder, Yves Detandt, and Gérard Degrez. A kinetic energy-preserving p1 iso p2/p1 finite-element method for computing unsteady incompressible flows. In *Computational Fluid Dynamics 2006*, pages 267–272. Springer, 2009.

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

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

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

[15] 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.

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

[17] 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.

[18] Haibiao Zheng, Yanren Hou, and Feng Shi. Adaptive variational multiscale methods for incompressible flow based on two local gauss integrations. *Journal of Computational Physics*, 229(19):7030–7041, 2010.