Moving mesh finite element method for unsteady Navier-Stokes flow

Yirong Wu and Heyu Wang
Department of Mathematics, ZheJiang University, HangZhou, 310027, China
Department of Mathematics, ZheJiang University, HangZhou, 310027, China

January 30, 2016

Abstract

In this paper, we use a $4P1 - P1$ finite element pair to solve the time-dependent Navier Stokes equations in 2D. This $4P1 - P1$ element pair is based on the data structure of hierarchy geometry tree. Two-layer nested meshes are used, that velocity mesh can be obtained by globally refining pressure mesh. Based on this tree structure, we can make the index between elements of velocity and pressure. Thus the process to assemble divergence matrix is trival with the index. On account of this data structure, h-adaptive mesh method can be used. Meanwhile, multigrid precondition can be implemented by this data structure.

In this work, moving mesh method is applied to solve incompressible Navier Stokes equations using $4P1-P1$ finite element pair. Seveval numerical problems are used to test the moving mesh strategy.

Key words: Navier Stokes, $4P1 - P1$, multigrid precondition, hierarchy geometry tree, moving mesh.

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 the finite element space of velocity relative to pressure, for example Taylor-Hood [1]. The other method is imposing constraint on pressure space, such as stablized $P1P1$, $P1P0$([2],[3]). In practical computing, adaptive scheme is often used to decrease the computitional cost for efficency. There are some works using h-adaptive $P2P1$ element because of simplicity, see ([4],[5], [6]) for detail. However, if domain has some corner or the solution has some singularities, we tend to use lower order approximations in engeneering computation. Adaptive method with stablized $P1P1$ and $P1P0$ finite element is proposed in [7]. In [8], dual element $P1P0$ is used

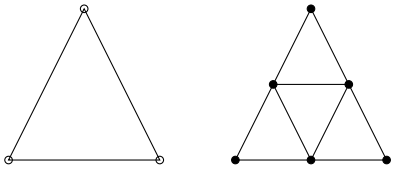


Figure 1: Left: pressure p element, $\circ$ for degrees of p; right: four velocity v elements, $\bullet$ for degrees of v.

for moving mesh method. Whereas the unstable finite element pairs applied to h-adaptive method have some technical difficulties. So the naturally stable $P1isoP2P1$ ([9],[10],[11]) is considered. In [11], it is pointed out that $P1isoP2P1$ satisfies the LBB condition. Four velocity elements can be obtained by refining the preesure element one time as Figure 1 shows. Noting that both elements of v and p are all linear elements. However, basis functions of $P1isoP2P1$ element are not in the same element. So assembling the divergent matrix between velocity and pressure is not a trival process.

In this paper, we choose a finite element pair called $4P1 - P1$ natrually satisfing the LBB condition. This finite element pair has the same mesh structure as $P1isoP2P1$ elements, but linear velocity basis functions are all locally in the same velocity element. So if we construct the index between four velocity elements and pressure element, divergent matrix can be assembled without difficulties. Note that mesh structure for $4P1 - P1$ elements is consistent with hierarchy geometry tree proposed by [12]. So the index can be built easily.

Moving mesh finite element methods have been considered by a lot of works such as [13],[14],[15], [16]. In [16], a moving finite element method based upon harmonic mapping was proposed, which is motivated by Dyinsky' work. The authors in[8] extend the moving scheme to solving incompressible Navier-Stokes equations in the primitive variables. A divergence-free interpolation is proposed in [8]. Our work is applying this moving strategy to $4P1 - P1$ finite element pair. As we known, adaptive mesh method can decrease computitioal cost as well as detect the fine flow structure, for example vorticity see [17].

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 $4P1 - P1$, In section 4, we use $4P1 - P1$ elements to approximate the governing equations. Next in section 5, the AMG preconditioner for steady stokes equations is shown. In Section 6, we give the moving mesh strategy. Then we present numerical experiements in section 7. Finaly, we give the conclusions in this section.

2 Governing Equation

Follow [18] and [17], we give nondimentional incompressible Navier-Stokes equations as follows:

$$\begin{aligned} \partial_t \vec{u} - \nu \nabla^2 \vec{u} + (\vec{u} \cdot \nabla)\vec{u} + \nabla p &= \vec{f}, \\ \nabla \cdot \vec{u} &= 0, \end{aligned} \tag{1}$$

The phisical domain is Ω in 2D or 3D. The initial boundary value problem of the system (1), with initial and boundary conditions on $\partial\Omega = \partial\Omega_D \bigcup \partial\Omega_N$:

$$\begin{aligned} &\vec{u} = \vec{w}, && \text{on } \partial\Omega_D \times [0,T] \\ &\nu \frac{\partial \vec{u}}{\partial n} - p = \vec{0}, && \text{on } \partial\Omega_N \times [0,T], \\ &\vec{u}|_{t=0} = \vec{u_0}, && \text{in } \Omega. \end{aligned} \tag{2}$$

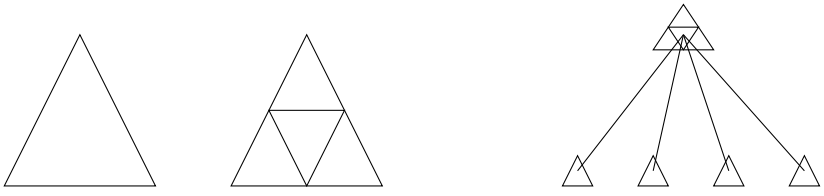
where $\vec{n}$ denotes the outward normal to the boundary. From equation (2), we only need to give the value of preesure on Ω_N. $\nu > 0$ is the constant kinematic viscosity. The vector $\vec{u}$ is velocity, and the scalar variable p is pressure.

Let L represent a characteristic length scale for the domain Ω. Let U be the maximum value of velocity on the inflow, then the Reynolds number denotes:

$$Re := \frac{UL}{\nu}. \tag{3}$$

3 Data Structure

In this paper finite element pair $4P1 - P1$ is adopted. $4P1 - P1$ is based on two different meshes and two different finite element spaces. Velocity mesh is obtained by globally refining pressure mesh. Mesh data structure is based on the hierarchy geometry tree in [12]. The four-fork tree is shown



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

Figure 2: Hierachy tree structure

in Figure 2. So a macro element(pressure) corresponds to four velocity elements. By traversing all pressure elements one time, we can acquire the $1-1$ index between elements of velocity and pressure based on the hierachy tree structure. See algorithm 1. After the index built, we can assemble

Algorithm 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
```

the element matrix for stiff matrix, mass matrix, and divergence matrix for velocity and pressure only using P1 element.

4 Finite Element Approximation

We define the solution and test spaces as

$$\mathbf{H}^1_E := \left\{ \vec{u} \in \mathcal{H}^1(\Omega)^d \middle| \vec{u} = \vec{w} \text{ on } \partial\Omega_D \right\}, \tag{4}$$

$$\mathbf{H}^1_{E_0} := \left\{ \vec{v} \in \mathcal{H}^1(\Omega)^d \middle| \vec{u} = \vec{0} \text{ on } \partial\Omega_D \right\}, \tag{5}$$

then the variational formulation is to find $(\vec{u}, p) \in (\mathbf{H}^1_E, L_2(\Omega))$ such that

$$\int_\Omega \frac{\partial \vec{u}}{\partial t} \cdot \vec{v} + \nu \int_\Omega \nabla\vec{u} : \nabla\vec{v} + \int_\Omega (\vec{u} \cdot \nabla\vec{u}) \cdot \vec{v} - \int p\,(\nabla \cdot \vec{v}) = \int_\Omega \vec{f} \cdot \vec{v}, \quad \forall \vec{v} \in \mathbf{H}^1_{E_0}, \tag{6}$$

$$\int_\Omega q\,(\nabla \cdot \vec{u}) = 0, \quad \forall q \in L_2(\Omega). \tag{7}$$

Here $\nabla\vec{u} : \nabla\vec{v}$ represents the componentwise scalar product, i. e., in 2D, $\nabla v_x \cdot \nabla u_x + \nabla v_y \cdot \nabla u_y$.

Assume τ_h is a triangulation of the domain Ω for pressure mesh with mesh parameter $h = max_{T \in \tau_h} diam(T)$. While $\tau_{\frac{1}{2}h}$ is the triangulation for velocity mesh. Two finite element spaces X^h_E and P_h are seperatly based on

$\tau_{\frac{1}{2}h}$ and τ_h such that

$$X_E^h \subset \mathcal{H}_E, \quad P_h \subset L_2(\Omega)$$

So (6) and (7) can be written as following : Find $(\vec{u}_h, p_h) \in X_E^h \times P_h$ such that

$$\begin{aligned} &\int_\Omega \frac{\partial \vec{u}_h}{\partial t} \cdot \vec{v}_h + \nu \int_\Omega \nabla \vec{u}_h : \nabla \vec{v}_h \\ &+ \int_\Omega (\vec{u}_h \cdot \nabla \vec{u}_h) \cdot \vec{v}_h - \int_\Omega p_h (\nabla \cdot \vec{v}_h) \quad = \int_\Omega \vec{f} \cdot \vec{v}_h, \qquad \forall \vec{v}_h \in \mathbf{X}_0^h; \\ &\int_\Omega q_h (\nabla \cdot \vec{u}_h) \qquad\qquad\qquad\qquad\quad =0, \qquad\qquad \forall q_h \in P^h. \end{aligned} \tag{8}$$

Let $\{\phi_j\}_{j=1}^n$ and $\{\psi_k\}_{k=1}^m$ be linear basis functions for velocity and pressure. In temporal direction, explicit Euler scheme is used. Then u_h, v_h, p_h can be written as follows.

$$u_h = \sum_{j=1}^n u_j \phi_j, \quad v_h = \sum_{j=1}^n v_j \phi_j, \quad p_h = \sum_{k=1}^m p_k \psi_k$$

Substituting the u_h, v_h, p_h into (8, this obtains a linear system

$$\begin{bmatrix} \frac{1}{dt}M + \nu A & 0 & B_x^T \\ 0 & \frac{1}{dt}M + \nu A & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ p \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ g \end{bmatrix}, \tag{9}$$

Where the $n \times n$ matrix M is the mass matrix and the Laplacian matrix is A, which are trivial.

$$A = [a_{ij}], \qquad a_{ij} = \int_\Omega \nabla \phi_i \cdot \nabla \phi_j \tag{10}$$

$$M = [m_{ij}], \qquad m_{ij} = \int_\Omega \phi_i \phi_j \tag{11}$$

$$\tag{12}$$

the right hand side terms f_x, f_y, g seperatly are

$$\begin{aligned} f_x = [f_i], \qquad & f_i = \int_\Omega (\frac{u^{(n)}}{dt} - (u^{(n)} \frac{\partial u^{(n)}}{\partial x} + v^{(n)} \frac{\partial u^{(n)}}{\partial y}) \phi_i \\ f_y = [f_i], \qquad & f_i = \int_\Omega (\frac{u^{(n)}}{dt} - (u^{(n)} \frac{\partial v^{(n)}}{\partial x} + v^{(n)} \frac{\partial v^{(n)}}{\partial y}) \phi_i \\ g = [g_k], \qquad & g_i = 0. \end{aligned}$$

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

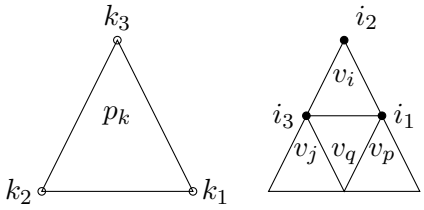


Figure 3: Left: p element; right: four velocity v elements corresponding to p element by the index built last section.

pressure element $\triangle_{p_k}$ (see Figure 3) by the index built last section. The local numbers 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 matrix of B_x^T is

$$\begin{bmatrix} \int_{\triangle_{v_i}} \psi_{k_1} \frac{\partial \phi_{i_1}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_2} \frac{\partial \phi_{i_1}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_3} \frac{\partial \phi_{i_1}}{\partial x} \\ \int_{\triangle_{v_i}} \psi_{k_1} \frac{\partial \phi_{i_2}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_2} \frac{\partial \phi_{i_2}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_3} \frac{\partial \phi_{i_2}}{\partial x} \\ \int_{\triangle_{v_i}} \psi_{k_1} \frac{\partial \phi_{i_3}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_2} \frac{\partial \phi_{i_3}}{\partial x} & \int_{\triangle_{v_i}} \psi_{k_3} \frac{\partial \phi_{i_3}}{\partial x} \end{bmatrix} \tag{13}$$

Add the entries of element matrix to relavent position (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) to B_x^T. After traversing all the velocity elements and adding element matrix (13)to B_x^T, 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 3. After seperatly assembling the element p_k with velocity elements v_i, v_j, v_p, v_p, one pressure element matrix is built and add it to B_x. Repeat above action until all the pressure elements are traversed. Therefor, B_x is completely built. Assembling of B_y is in the same way.

If we get rid of time term and nonlinear $\vec{u} \cdot \nabla \vec{u}$ in system (1), we can get steady Stokes equations. After the same discretization, the linear sestem equation is as follows

$$\begin{bmatrix} \nu A & 0 & B_x^T \\ 0 & \nu A & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ p \end{bmatrix} = \begin{bmatrix} \hat{f}_x \\ \hat{f}_y \\ \hat{g} \end{bmatrix}, \tag{14}$$

The right hand side of (14), we can set $\hat{f}_x = 0, \hat{f}_y = 0, \hat{g} = 0$ for simplicity.

We solve the time-dependent Navier-Stokes system (9) using GMRES method preconditioned with incomplete LU decomposition. In order to guarantee the numerical scheme stability, we choose the time adaptive step such that satisfing the CFL condition in each element. In order to illustrate our scheme, we demonstrate the flow-chart of the algorithm in (2).

Algorithm 2 Moving mesh FEM for Navier Stokes equation

1: **while** $t_n < T$ **do**
2: Solve Navier-Stokes system(9) in $t = t_n$ on the velocity mesh $\triangle_v^{(n)}$ and pressure mesh $\triangle_p^{(n)}$. Then obtain numerical solutions $\vec{u}_h^{(n)}, p_h^{(n)}$.
3: Caculate monitor function on $\triangle_p^{(n)}$ using $\vec{u}_h^{(n)}, p_h^{(n)}$. And obtain logical mesh $\vec{\xi}^*$ by solving (19).
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 (25) on $\triangle_v^{(n)}$ to get medium variable $\vec{u}_{h,*}^{(n)}, p_{h,*}^{(n)}$ on new mesh.
7: Update $\triangle_p^{(n)}$ to $\triangle_p^{(n+1)}$, Sync $\triangle_v^{(n)}$ by the hierachy geometry tree stucture to $\triangle_v^{(n+1)}$.
8: Go back to 3.
9: Solve Navier-Stokes system (9) to truely 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 Precondition strategy

5.1 AMG precondition for steady stokes equations

For simplicity, we choose the steady Stokes equations to show efficiency of AMG preconditioner. Recall linear system (14), which is indedinite, we choose a Krylov subspace method, MINRES as the solver. As we known, to make sure the efficency of MINRES, we should choose an efficient preconditioner. Motivated by the work in [18], pressure mass matrix Q can be an approximation of the schur complement of coefficient matrix in (14) and $\hat{Q} = diag(Q)$ works well in practical computation. Here, we choose matrix K

$$K = \begin{bmatrix} \nu A & 0 & 0 \\ 0 & \nu A & 0 \\ 0 & 0 & \hat{Q} \end{bmatrix} \tag{15}$$

as an approximation to the coefficient matrix M

$$M = \begin{bmatrix} \nu A & 0 & B_x^T \\ 0 & \nu A & B_y^T \\ B_x & B_y & 0 \end{bmatrix}, \tag{16}$$

In precondition, we often need to apply the action of K^{-1}. Instead of solving $V_{des} = K^{-1}V_{src}$, we use an AMG solver (for $P1$ element) of AFEPack(a finite element package) to solve block $(0,0)$ and $(1,1)$ of system $KV_{des} = V_{src}$, where V_{src} is given and V_{des} is unknown. AFEPack can be obtained from http://dsec.pku.edu.cn/~rli.

6 Moving mesh Stategy

At time $t = t_{n+1}$, by scheme mentioned above, we can get numerical solutions $\vec{u}_h^{(n+1)}, p_h^{(n+1)}$ on old mesh $\mathcal{T}_h^n$. We follow the framework in [8] to implement divergence-free interpolation of $(\vec{u}_h^{(n+1)}, p_h^{(n+1)})$ from $\mathcal{T}_h^n$ to new mesh $\mathcal{T}_h^{(n+1)}$. Noticing that common Dirichlet and Neumann boundary condition are considered instead of periodic boundary condition. Briefly speaking, the moving mesh strategy mainly contains four steps as follows.

6.1 Step 1 Obtain monitor function

Choices of an appropriate monitor function are very important for adaptive scheme. Let $m = 1/G$, where G is the monitor function. As illustrated in [8], there are some common choices of G. One based on vorticity is

$$G_0 = \sqrt{1 + \alpha|\omega|^\beta}. \tag{17}$$

where $\omega = \nabla \times \vec{u}$, α, β are positive constants. It is illustrated in [15] that $\beta = 4$ is a good choice for adaptation in flow past cylinder problem. In this work, $\alpha = 0.5, \beta = 2$ shows good result. Another selection of G is demonstrated as follows, which based on the gradient of solution.

$$G_1(\vec{u}) = \sqrt{1 + \alpha|\nabla\vec{u}|^\beta} \tag{18}$$

where $\vec{u}$ is velocity variable. This monitor function is used for problems which solutions have large gradient, such as colliding flow which are used for accuracy test for our algorithm.

6.2 Step 2 Get a new logical mesh

In [16], by solving elliptic equation

$$\nabla_{\vec{x}}(m\nabla_{\vec{x}}\vec{\xi}) = 0 \tag{19}$$

with Dirichlet boundary condition

$$\vec{\xi}|_{\partial\Omega} = \vec{\xi}_b \tag{20}$$

a new logical mesh $\mathcal{T}_c^*$ with its nodes $\mathcal{A}^*$ is obtained.

6.3 Step 3 Achieve mesh move direction in physical domain

Several notaions are introduced firstly. $\mathcal{T}_h$ is the triangulation of physical domain, and its i th node denotes X_i. Meantime, T_i is the set of elements containing X_i. Correspondingly, on the logical domain the notations are $\mathcal{T}_c, \mathcal{A}_i, T_{i,c}$. $(\mathcal{A}_i^1, \mathcal{A}_i^2)$ are denoted as the coordinates of i th node $\mathcal{A}_i$ in the

logical domain. According to Step 1 and Step 2, a new logical mesh $\mathcal{T}_c^*$ is got, meanwhile $\mathcal{A}_i^*$ is its i th node. Therefore we can attain the error of i th node:

$$\delta\mathcal{A}_i = \mathcal{A}_i^0 - \mathcal{A}_i^* \tag{21}$$

where $\mathcal{A}_i^0$ is the ith node of the initial logical mesh $\mathcal{T}_c^0$. To be noticed that once the initial logical mesh is obtained, it maintains unchange until the whole algorithm is over.

For a given element E in $\mathcal{T}_h$, its vertexes are denoted as $X_k, 0 \le k \le 2$. Piecewise linear map from $V_{T_c^*}(\Omega_c)$ to $V_T(\Omega)$ has constant gradient $\partial\vec{x}/\partial\xi$ on E, which can be obtained by solving following system

$$\begin{aligned}
&\begin{pmatrix} \mathcal{A}_{E_1}^{*,1} - \mathcal{A}_{E_0}^{*,1} & \mathcal{A}_{E_2}^{*,1} - \mathcal{A}_{E_0}^{*,1} \\ \mathcal{A}_{E_1}^{*,2} - \mathcal{A}_{E_0}^{*,2} & \mathcal{A}_{E_2}^{*,2} - \mathcal{A}_{E_0}^{*,2} \end{pmatrix} \begin{pmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^1}{\partial \xi^2} \\ \frac{\partial x^2}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^2} \end{pmatrix} \\
=&\begin{pmatrix} X_{E_1}^1 - X_{E_0}^1 & X_{E_2}^1 - X_{E_0}^1 \\ X_{E_1}^2 - X_{E_0}^2 & X_{E_2}^2 - X_{E_0}^2 \end{pmatrix}
\end{aligned}$$

If we let the volume of the element as the weight, the weighted average displacement of the ith node X_i is as follows:

$$\delta X_i = \frac{\sum\limits_{E\in T_i} |E| \frac{\partial\vec{x}}{\partial\xi}|_{\text{in}E} \delta\mathcal{A}_i}{\sum\limits_{E\in T_i} |E|}. \tag{22}$$

where $|E|$ is the volume of element E. We also choose a positive parameter μ to prevent mesh tangling. Assume that the new mesh on the physical domain is denoted as $\mathcal{T}^*$, and its nodes X_i^*

$$X_i^* = X_i + \mu\delta X_i \tag{23}$$

The method of selecting μ see [8] for detail.

6.4 Step 4 Eusure the incompressible constraint interpolation

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

$$\frac{\partial\vec{u}}{\partial\tau} - \nabla_{\vec{x}}\vec{u}\cdot\delta\vec{x} = -\nabla p. \tag{24}$$

$$\nabla_{\vec{x}}\cdot\vec{u} = 0 \tag{25}$$

where $\delta\vec{x} := x^{\text{old}} - x^{\text{new}}$, $x^{\text{old}}, x^{\text{new}}$ are two sets of coordinates in physical domain. τ is a virtual time variable and often choosen as 1. Here p is just an intermediate variable, different from pressure variable in (1).

Weak form of (25) is : find $(\vec{u}_h, p_h) \in V_h \times P_h$ such that

$$\begin{aligned} (\partial_\tau \vec{u}_h - \nabla_{\vec{x}} \vec{u}_h \cdot \delta\vec{x}, \vec{v}_h) = &\quad (p_h, \nabla \vec{v}_h)\,. \quad \forall \vec{v}_h \in V_h \\ (\nabla_{\vec{x}} \cdot \vec{u}, q_h) = &\quad 0. \quad \forall q_h \in P_h \end{aligned} \tag{26}$$

In this work, we apply full implicity scheme which is unconditional stable, instead of three-step Runge-Kutta in [8], to (26) for time discretization:

$$\begin{aligned} \left(\frac{u_{h,*}^{(n)} - u_h^{(n)}}{\triangle \tau}, \vec{v}_h \right) + \left(\delta\vec{x} \cdot \nabla \vec{u}_{h,*}^{(n)}, \vec{v}_h \right) = &\quad \left(p_{h,*}^{(n)}, \nabla \vec{v}_h \right) \\ \left(\nabla \cdot u_{h,*}^{n}, q_h \right) = &\quad 0. \end{aligned} \tag{27}$$

where $\vec{u}_h^{(n)}$ 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.

7 Numerical Tests

There are three numerical tests here. In practical computing, we use solutions of Stokes equations as initial value of Navier-Stokes equations. We show moving mesh and numerical solutions in the following. Our codes are based on the AFEPack.

7.1 Jetting flow into a static field

This example models a thin flow jetting into a channel with static flow field. Our computional domain is $\Omega = [0, 12] \times [-3, 3]$ and viscosity is $\nu = 0.0005$. Natural outflow boundary is imposed on $x = 12$, while inflow boundary condition $u_x = 1 - 100y^2, u_y = 0$ on $x = 0, y \in [-0.1, 0.1]$. No-slip boundary condition is setted on other boundary of Ω.

We select (17) as monitor in our moving strategy. Parameter $\alpha = 2.0, \beta = 2.0$ performs well. As we known, when a thin flow jets into static field, two symmetric vortexes appears along the jetting flow. For improving the computional precision, it requires more grids around the flow. From Figure 4, mesh is concentrated around the vorticity contour with large velocity gradient.

7.2 Navier-Stokes flow over a step

We consider the test problem in [7] 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 $\vec{u} = (0, 0)^T$ is imposed. Outflow boundary is enforced natural boundary condition meanwhile at the inflow boundary $\vec{u} = (4y(1 - y), 0)$.

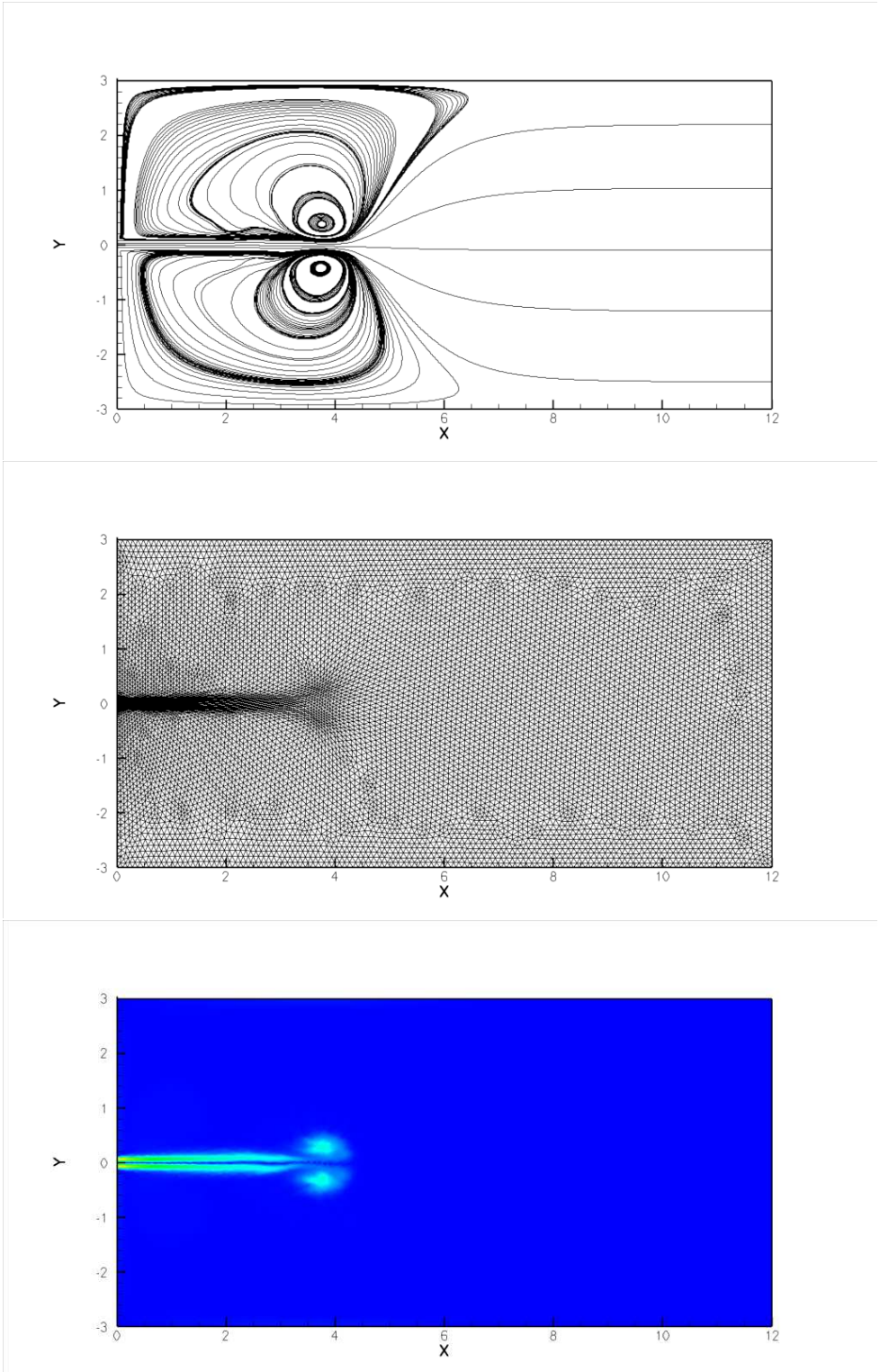


Figure 4: Jetting flow: top: velocity streamline, center: mesh, bottom: vorticity contour at t = 12s.

(17) is selected as monitor function based on vorticity with $\alpha = 0.4, \beta = 2.0$. As we known, singularities arise at the concave corner in this problem. Figure 5 shows the initial moving mesh clusters near the concave edge. Figure 6, 7 and 9 show moving mesh and contour of vorticity at $t = 0.5s, t = 1s$, and $t = 2s$. It is observed that our moving mesh is consistent with the structure of vorticity.

Pressure contour and velocity streamline at $t = 2s$ are shown in Figure 10.

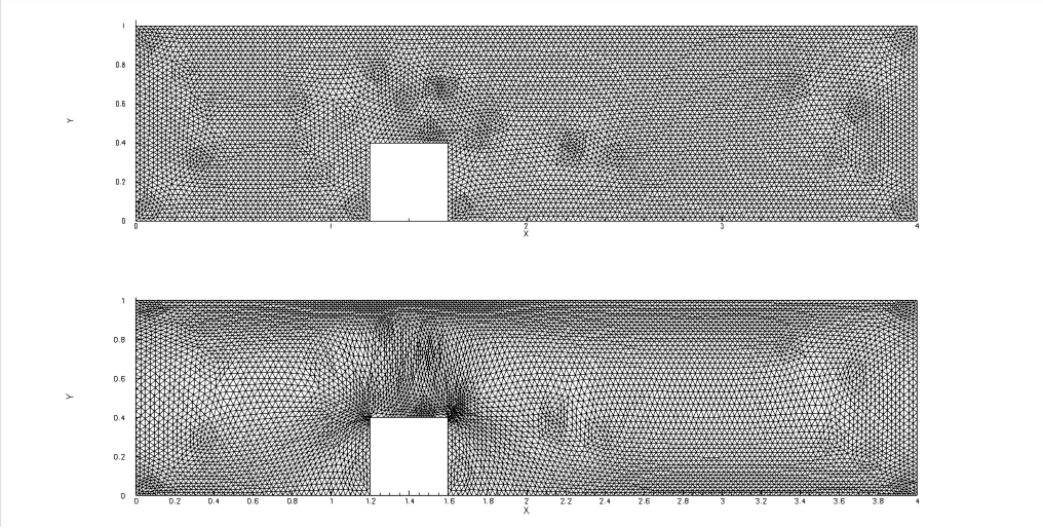


Figure 5: Up: Initial uniform mesh; bottom: initial moving mesh.

7.3 Navier-Stokes 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 viscosity be $\nu = 0.001$ 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, (17) is a good choice as the monitor in our moving strategy. The parameters α and β are 1.0, 2.0.

The evolution of mesh, pressure contour and velocity streamline are illustrated in Figure 12, 13, 14. It can be discoverd that moving mesh can efficiently capture the velocity structure and our mesh quality is better than

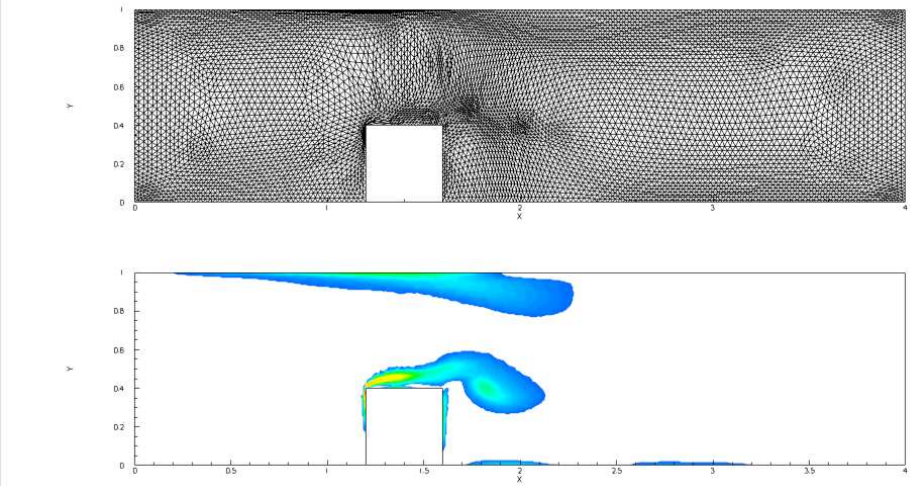


Figure 6: Up: moving mesh; bottom: contour of vorticity at t = 0.5s

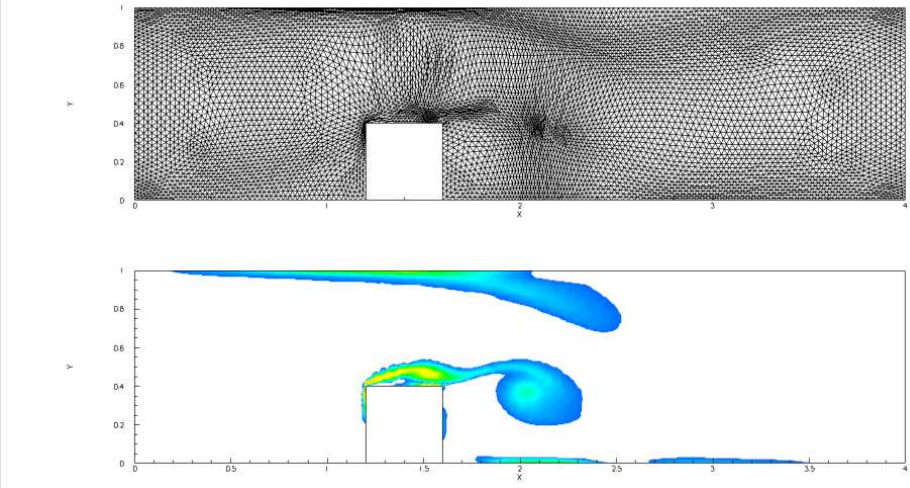


Figure 7: Up: moving mesh; bottom: contour of vorticity at t = 1s

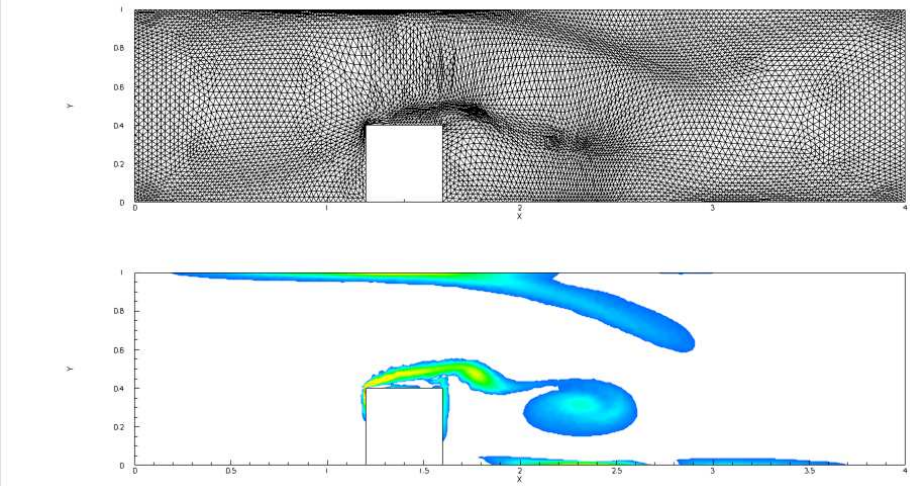


Figure 8: Up: moving mesh; bottom: contour of vorticity at t = 1.5s

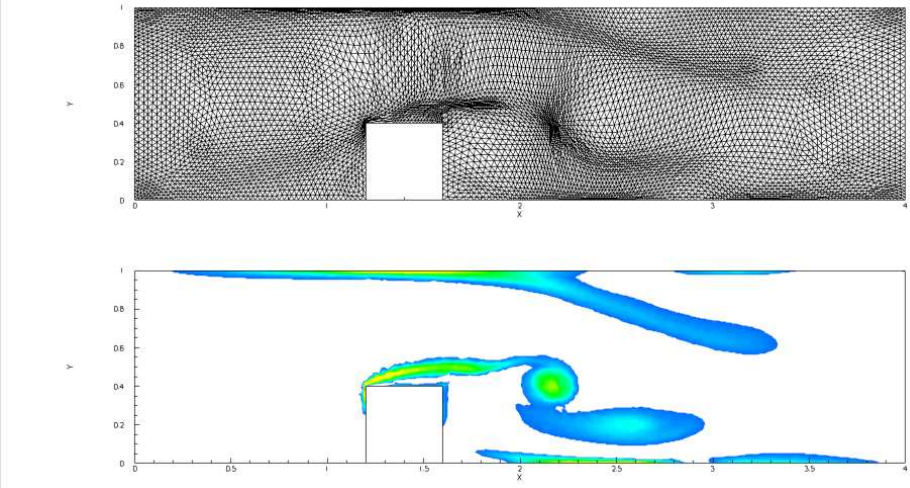


Figure 9: Up: moving mesh; bottom: contour of vorticity at t = 2s

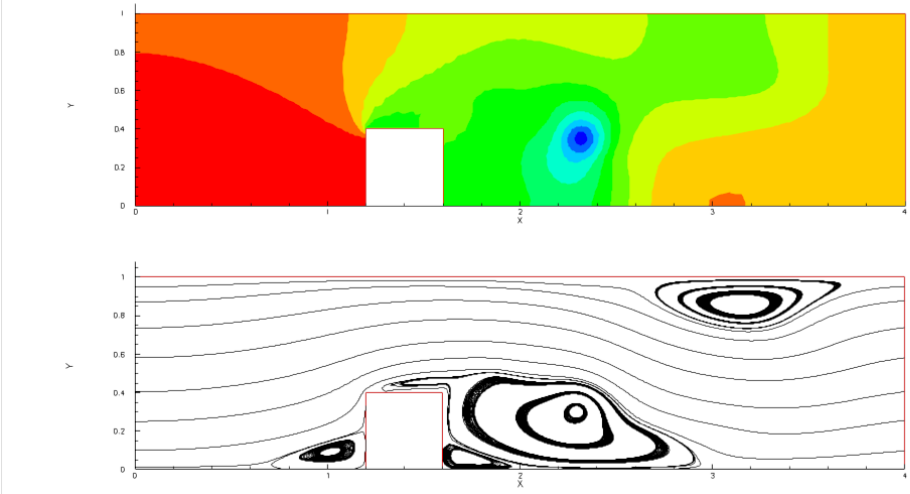


Figure 10: Up: Pressure contour; bottom: velocity streamline at t = 2s

[15].

Figure 15 shows the velocity contour and mesh when t = 23s.

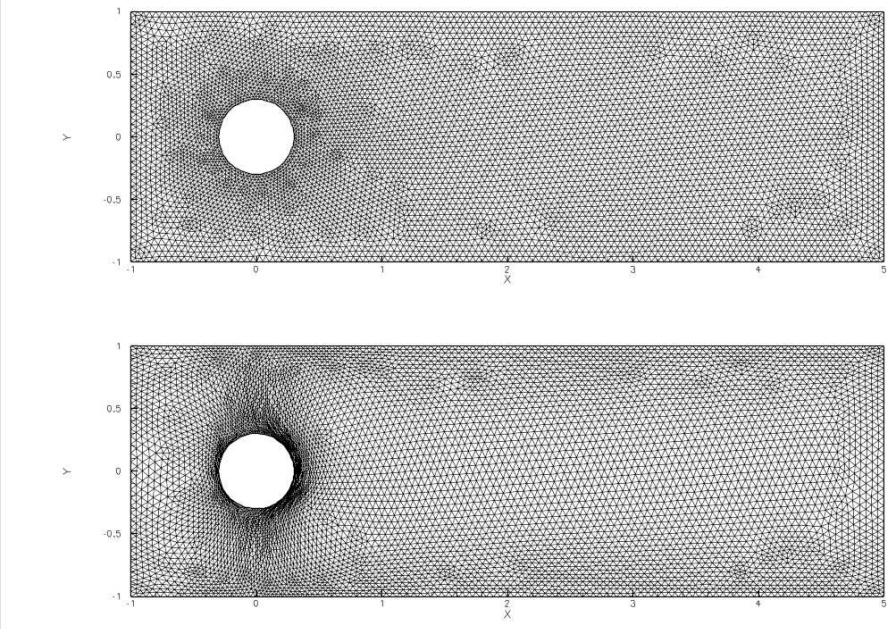


Figure 11: Up: uniform mesh; bottom: initial moving mesh.

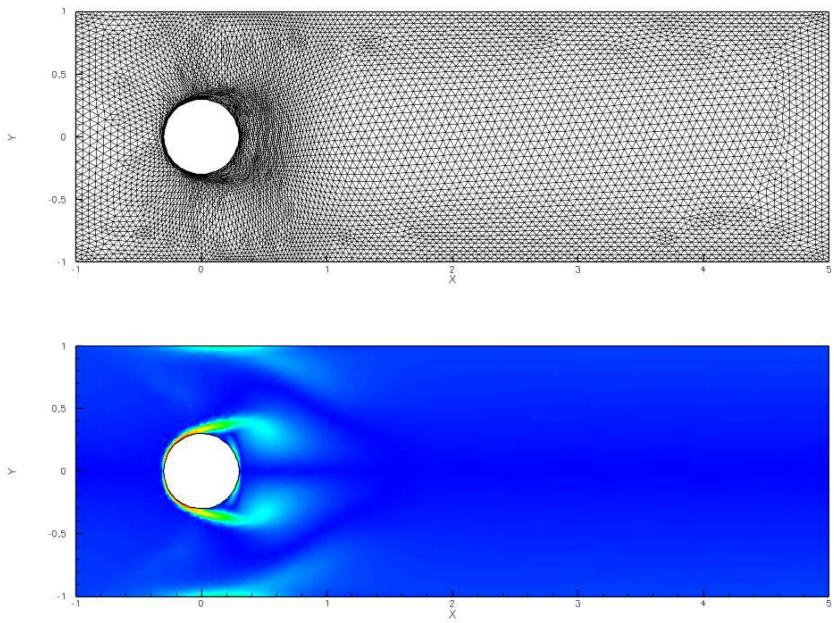


Figure 12: Up: moving mesh; bottom: vorticity contour at $t = 1s$.

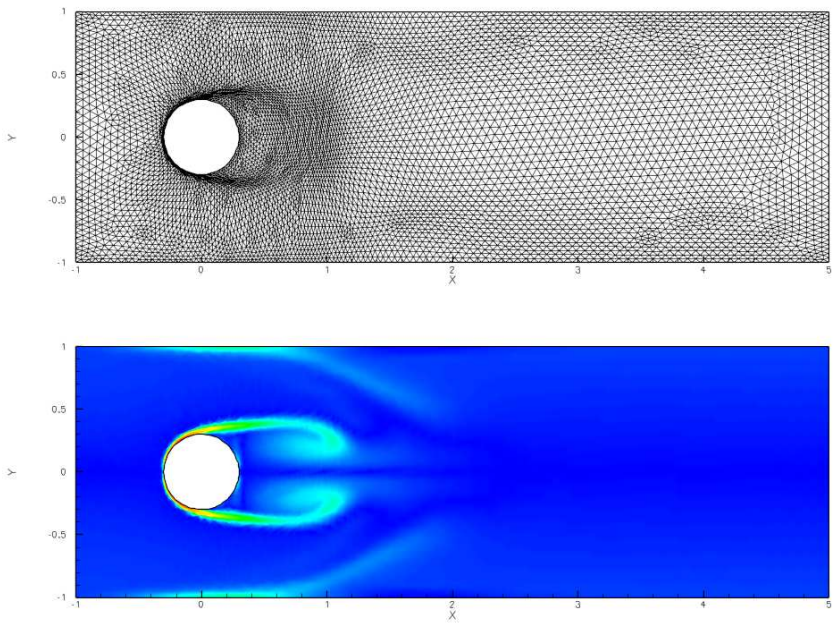


Figure 13: Up: moving mesh; bottom: vorticity contour at $t = 2s$.

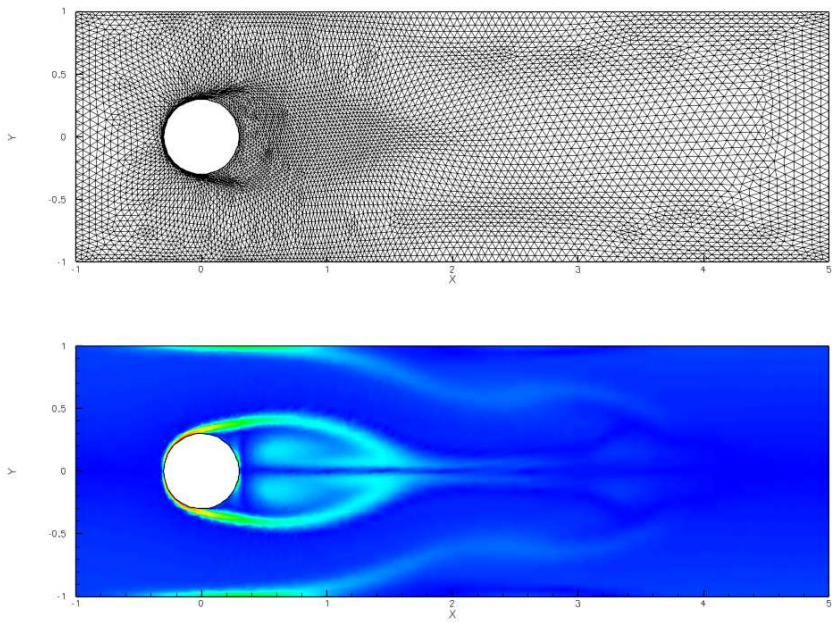


Figure 14: Up: moving mesh; bottom: velocity contour at $t = 4s$.

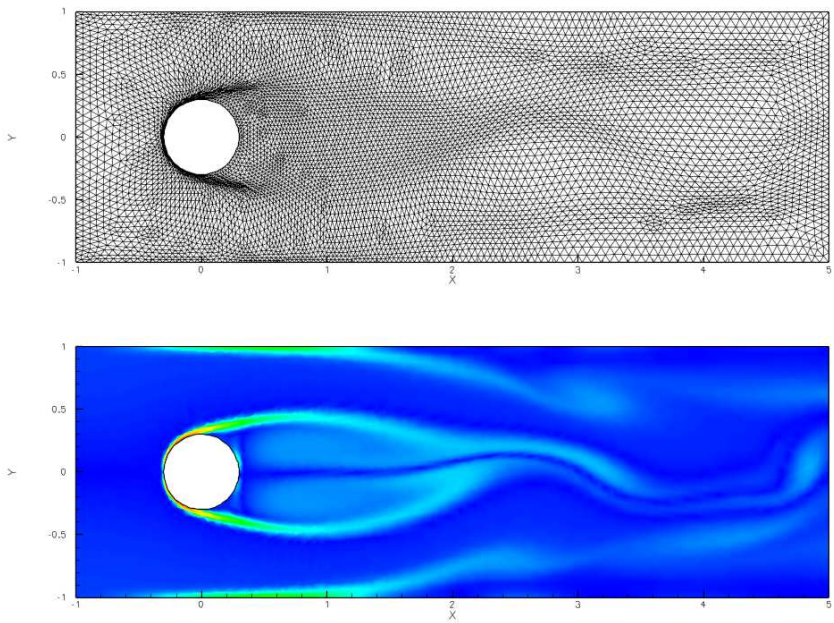


Figure 15: Up: moving mesh; bottom: velocity contour at $t = 23s$.

8 Remarks

In this paper, unsteady Navier-Stokes equations have been numerically solved using moving mesh method on $4P1-P1$ finite element pair. The main work is using the hierachy geometry tree structure to simply build system matrix by $4P1-P1$ element pair. And we apply moving mesh method to this pair.

$4P1-P1$ element pair natrually satisfies the LBB confiditon and is linear-order element. In practical engeneering computation, linear-order element is more popular than high order because of its simplicity and complexity of problems. For capturing the fine flow structure, we apply moving mesh method to this pair using vorticity as monitor. The moving mesh is consistent with the structure of vorticity. Boundary conditions in our testing problem are all non-periodic, so it has more extensive application.

Our mesh structure is based on the hierachy geometry tree structure, multi-mesh adaptive method will be applied to $4P1-P1$ pair. Also moving mesh method for incompressible flow will be extended to three-dimention in future work.

References

[1] C Taylor and P Hood. Navier-stokes equation using mixed interpolation. finite element method in flow problems oden editor, 1974.

[2] 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.

[3] 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.

[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] 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.

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

[11] 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.

[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] 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.

[18] 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.