East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.080416.240516a

Vol. **6**, No. 4, pp. 353-366
November 2016

An AMG Preconditioner for Solving the Navier-Stokes Equations with a Moving Mesh Finite Element Method

Yirong Wu and Heyu Wang*

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

Received 8 April 2016; Accepted (in revised version) 24 May 2016.

Abstract. AMG preconditioners are typically designed for partial differential equation solvers and divergence-interpolation in a moving mesh strategy. Here we introduce an AMG preconditioner to solve the unsteady Navier-Stokes equations by a moving mesh finite element method. A $4P1-P1$ element pair is selected based on the data structure of the hierarchy geometry tree and two-layer nested meshes in the velocity and pressure. Numerical experiments show the efficiency of our approach.

AMS subject classifications: 65M60

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

1. Introduction

We consider the incompressible Navier-Stokes equations in primitive variables

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

subject to initial and boundary conditions on $\partial\Omega = \partial\Omega_D \bigcup \partial\Omega_N$ — viz.

$$\begin{aligned} &\boldsymbol{u} = \mathbf{w} \quad \text{on } \partial\Omega_D \times [0,T]\,, \\ &\nu\partial_{\boldsymbol{n}}\boldsymbol{u} - \boldsymbol{n}p = \mathbf{0} \quad \text{on } \partial\Omega_N \times [0,T]\,, \\ &\boldsymbol{u}|_{t=0} = \boldsymbol{u}_0 \quad \text{in } \Omega\,, \end{aligned} \tag{1.2}$$

where $\Omega \in \mathscr{R}^d$ $(d = 2,3)$ is the spatial domain, $[0,T]$ the time interval, $\boldsymbol{u}$ the velocity vector and p the scalar pressure, $\boldsymbol{n}$ denotes the outward normal to the boundary $\partial\Omega$ of Ω, $\nu > 0$ the constant kinematic viscosity coefficient, and $\boldsymbol{f}$ represents the total external force. In particular, here we discuss the solution of Eqs. (1.1) and (1.2) by moving mesh finite element methods.

*Corresponding author. *Email addresses:* 21106058@zju.edu.cn (Y. Wu), wangheyu@zju.edu.cn (H. Wang)

 ©2016 Global-Science Press

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

After Winslow [1] proposed using a moving mesh to solve elliptic partial differential equations, Dvinsky [2] pointed out harmonic function theory could be used to generate a mesh, which motivated Li *et al.* [3] to propose a moving mesh finite element strategy based upon harmonic mapping. Di *et al.* [4] extended the moving strategy to solve the incompressible Navier-Stokes equations in primitive variables. The present authors [5] then designed a divergence-free interpolation in a moving strategy for solving a system of linearised Navier-Stokes-type equations and the incompressible Navier-Stokes flow equations, by applying a $4P1-P1$ element pair with a moving mesh finite element method. This pair has the same mesh structure as a $P1isoP2P1$ element, which is naturally LBB stable [6]. Four velocity elements can be obtained by refining the pressure element once — cf. Fig. 1. In the $4P1-P1$ element, both velocity and pressure are all standard $P1$, but in the $P1isoP2P1$ element they are not — see Ref. [5] for details.

Spatial discretisation of the Navier-Stokes system via the LBB-stable $4P1-P1$ element pair leads to a saddle point problem. A two-grid method was introduced to solve the Navier-Stokes equations [7–9]. There are many articles on saddle point problems where preconditioners for the Krylov subspace method are developed, such as in block and multigrid preconditioning [10]. Many authors have introduced various block preconditioners, where the main issue is to find a good approximation of the Schur complement [11–14]. There are other preconditioned methods discussed in Refs. [15–17], involving an efficient AMG preconditioner for a Krylov solver for the Navier-Stokes equations. Efficient precondition methods for saddle point problems are mainly based on a uniform mesh, although a stretched mesh was considered in Ref. [16].

Here we apply an AMG preconditioner to a finite element moving mesh method for solving the Eqs. (1.1) and (1.2), based on the work of Ref. [18]. An AMG precondition stretegy is designed for divergence-free interpolation in our moving mesh method, and the efficiency of our AMG preconditioner is analysed through several numerical experiments. In Section 2, we use the $4P1-P1$ element to approximate the governing equations, and present our AMG preconditioner for the Navier-Stokes equations in Section 3. In Section 4, we briefly discuss our moving mesh strategy, and then present numerical experiments in Section 5. Our concluding remarks are in Section 6.

2. Data Structure and Weak Formulation

We divide the time interval $[0, T]$ into N steps with $\{t_i\}_{i=1}^{N}$, and let $\mathbf{u}^j$ and p^j denote the discrete approximation to $\boldsymbol{u}(\cdot, t_j)$ and $p(\cdot, t_j)$. For simplicity, we choose the backward Euler scheme in linearising the term $\boldsymbol{u}^{n+1} \cdot \boldsymbol{\nabla} \boldsymbol{u}^{n+1}$ as $\boldsymbol{u}^{n} \cdot \boldsymbol{\nabla} \boldsymbol{u}^{n+1}$. We adopt the finite element pair $4P1-P1$, in choosing two different triangular meshes and two different finite element spaces. By using the hierarchy geometry tree structure [19], the velocity mesh involving a global refining pressure mesh needs to be generated only once — cf. Fig. 2.

The $1-1$ index between the velocity elements and pressure elements can be obtained readily via the hierarchy geometry tree structure — cf. Ref. [5] for details. Let $\mathscr{T}_h$ denote the triangular subdivision for the velocity mesh with mesh size $h = \max_{T \in \mathscr{T}_h} diam(T)$, $\mathscr{T}_H (H = 2h)$ the pressure mesh, and $X^h \subset (H_0^1(\Omega)^2)$ and $P^H \subset L^2(\Omega)$ the respective

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

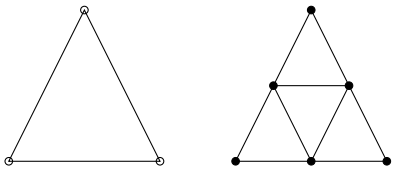


Figure 1: Pressure p element, where the $\circ$ denote the degree of freedom of p (left); four velocity v elements, where the $\bullet$ denote the degree of freedom of v (right).

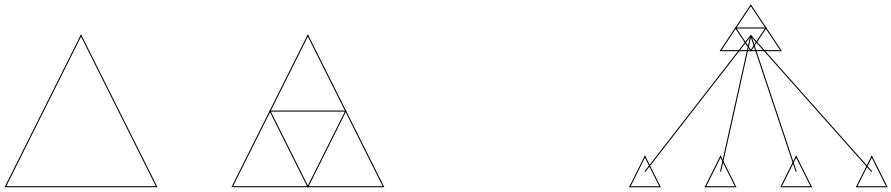


Figure 2: Hierarchy tree structure.

finite-dimensional approximation spaces. Thus given $(\mathbf{u}_h^n, p_H^n)$ at the time t_n, we compute $(\mathbf{u}_h^{n+1}, p_H^{n+1})$ via

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

for all $(\mathbf{v}_h, q_H) \in X^h \times P^H$.

3. Fast Krylov Solver in the AMG Preconditioner Strategy

Let $(\{\phi_j\}_{j=1}^n, 0)^T$ and $(0, \{\phi_j\}_{j=1}^n)^T$ be the linear basis functions for velocity space X^h, and $\{\psi_k\}_{k=1}^m$ denote linear basis functions for the pressure space P^H. Then the velocity components in $\mathbf{u}_h^{n+1} = (u_h^{x,n+1}, u_h^{y,n+1})^T$ and the pressure p_H^{n+1} at $t = t_{n+1}$ can be expressed as

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

On substituting (3.1) into the weak form (2.1), we obtain the linear system

$$\begin{bmatrix} \frac{1}{dt}M + \nu A + N & 0 & B_x^T \\ 0 & \frac{1}{dt}M + \nu A + N & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} \alpha^{x,n+1} \\ \alpha^{y,n+1} \\ \alpha^{p,n+1} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix}, \tag{3.2}$$

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

where the divergence matrix $B=[B_x,B_y]$ is such that

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

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

Assembling the matrix B is nontrivial, since the basis functions of the velocity and pressure elements are defined on different meshes. However, according to the $1-1$ index between the velocity and pressure elements mentioned above, we can use the local $P1$ element for both the velocity and pressure elements to assemble B and B^T. We write

$$F_v^{n+1} = \frac{1}{dt}M + \nu A + N\,,$$

where

$$A := [A]_{ij} = \left(\nabla\phi_i, \nabla\phi_j\right), \quad i,j=1,\cdots,n_u\,, \tag{3.5}$$

$$N := [N]_{ij} = \left(\boldsymbol{u}_h^n\cdot\nabla\phi_i, \phi_j\right), \quad i,j=1,\cdots,n_u\,, \tag{3.6}$$

$$M := [M]_{ij} = \left(\phi_i, \phi_j\right), \quad i,j=1,\cdots,n_u\,. \tag{3.7}$$

In order to solve the linear algebraic system (3.2) efficiently, we use a preconditioned GMRES as solver. The block triangular preconditioner discussed in Ref. [18] is

$$\mathscr{P} = \begin{pmatrix} F & 0 & B_x^T \\ 0 & F & B_y^T \\ 0 & 0 & S \end{pmatrix}, \tag{3.8}$$

where $F = F_v^{n+1}$ and $S = B_xF^{-1}B_x^T + B_yF^{-1}B_y^T$ is the Schur complement matrix. The action of $\mathscr{P}^{-1}$ is divided into two steps: — viz. first solve the Schur complement system, and then solve the two scalar systems associated with F. As it is costly to solve the Schur complement system, in practice a PCD preconditioner is used to approximate the Schur complement matrix S [18]. The PCD preconditioner is denoted by $S_* = A_pF_p^{-1}Q_p$, where the A_p, F_p and Q_p are defined on the pressure space, Q_p is the mass matrix, A_p the pressure diffusion matrix, and F_p the convection diffusion matrix

$$\begin{aligned} F_p &:= [F_p]_{ij} = \nu(\boldsymbol{\nabla}\psi_i, \boldsymbol{\nabla}\psi_j) + (\boldsymbol{u}_h^n\cdot\boldsymbol{\nabla}\psi_i, \psi_j), \quad i,j=1,\cdots,n_p\,, \\ A_p &:= [A_p]_{ij} = (\boldsymbol{\nabla}\psi_i, \boldsymbol{\nabla}\psi_j), \quad i,j=1,\cdots,n_p\,, \\ Q_p &:= [Q_p]_{ij} = (\psi_i, \psi_j), \quad i,j=1,\cdots,n_p\,. \end{aligned} \tag{3.9}$$

Let $W_p^n := [W_p^n]_{ij} = (\boldsymbol{u}_h^n\cdot\boldsymbol{\nabla}\psi_i, \psi_j)$, $i,j=1,\cdots,n_p$ such that F_p can be rewritten as $F_p = \nu A_p + W_p^n$. We implement the PCD preconditioner by

$$S_*^{-1} \approx Q_p^{-1}F_pA_p^{-1}\,, \tag{3.10}$$

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

with the exact PCD preconditioning operator denoted by

$$\mathscr{M}^{-1} = \begin{pmatrix} F & 0 & B_x^T \\ 0 & F & B_y^T \\ 0 & 0 & S_* \end{pmatrix}^{-1} . \tag{3.11}$$

The $V^d = \mathscr{M}^{-1}V^s$, where $V^d = (V_x^d, V_y^d, V_p^d)^T, V^s = (V_x^s, V_y^s, V_p^s)^T$, is computed in two steps. Thus we first solve

$$V_p^d = S_*^{-1}V_p^s = Q_p^{-1}F_pA_p^{-1}V_p^s , \tag{3.12}$$

involving a Poisson solution A_p^{-1} and a mass solution Q_p^{-1}, so we can use an AMG solver for the Poisson equation and then to solve

$$\begin{aligned} FV_x^d &= (V_x^s - B_x^T V_p^d), \\ FV_y^d &= (V_y^s - B_y^T V_p^d) . \end{aligned} \tag{3.13}$$

In practice, we use a fixed number of AMG iterations (usually one or two) to solve the systems about the matrices F, F_p, Q_p and A_p in Eqs. (3.12) and (3.13) in iterated PCD preconditioning. We chose the AMG solver from the adaptive finite element package AFPack, which may be obtained from the website http://dsec.pku.edu.cn/∼rli. The efficiency of the PCD preconditioning is demonstrated in Ref. [18] (Section 10) and in Ref. [20] for the buoyancy driven flow problem. In our experiments, the F_p in (3.9) was not very efficient in dealing with the coefficient ν, and we used $F_p = A_p + W_p^n$ in comparing the efficiency of different choices of F_p.

The method in Ref. [14] for the matrices F_p and A_p improves the computational efficiency on a Neumann boundary. In the case of marix A_p, the discrete pressure p_h has to satisfy an homogeneous Dirichlet boundary condition along the outflow boundary $\partial\Omega_N$ and a Neumann condition $\partial p_h/\partial n = 0$ along $\partial\Omega_D$. The matrix F_p should satisfy the condition

$$\nu\frac{\partial p_h}{\partial n} + (\mathbf{w}_h \cdot \mathbf{n})p_h = 0. \tag{3.14}$$

on the boundary $\partial\Omega$, but in practice F_p is also treated on the outflow boundary $\partial\Omega_N$ in the same way as A_p — cf. Ref. [18], Section 9.2.2. For cavity flow, $\mathbf{w}_h \cdot \mathbf{n}$ must be zero everywhere on the boundary $\partial\Omega$, so (3.14) becomes $\partial p_h/\partial n = 0$ so we do nothing for F_p on $\partial\Omega$. We also design the preconditioning strategy for solution interpolation in moving mesh strategy as shown in the next section.

4. Moving Mesh Strategy

4.1. Moving mesh framework

In adopting the moving strategy of Ref. [4], we now briefly summarise our moving mesh method. At time $t = t_n$, we obtain the numerical solutions $\mathbf{u}_h^{(n)}$, $p_H^{(n)}$ on an old mesh $\mathscr{T}_h^n$ and then implement the divergence-free interpolation of the solutions on $\mathscr{T}_h^n$ to a new mesh $\mathscr{T}_h^{(n+1)}$, in a strategy mainly involving four steps.

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

Step 1 Obtain the monitor function, chosen appropriately for the adaptive scheme. Thus letting $m = 1/G$ where G denotes the monitor function, as illustrated in Ref. [4] there are some common choices of G. One based on vorticity is

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

where $\omega = \nabla \times \mathbf{u}$ and α, β are positive constants. In our work, $\beta = 2$ performs well, while α is user-defined according the problem.

Step 2 Denote the initial logical fixed uniform mesh as $\mathscr{T}_c^0$ (with nodes $\mathscr{A}^0$). Then obtain a new logical mesh $\mathscr{T}_c^*$ (with nodes $\mathscr{A}^*$), by solving elliptic equation

$$\begin{aligned} &\nabla_{\mathbf{x}}(m\nabla_{\mathbf{x}}\xi) = 0\,, \\ &\xi|_{\partial\Omega} = \xi_b\,, \end{aligned} \tag{4.2}$$

where m is given in Step 1.

Step 3 Achieve the mesh move direction in the physical domain from the difference between $\mathscr{T}_c^*$ and the initial logical mesh $\mathscr{T}_c^0$:

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

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

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

Ref. [21] provides more detail on this mesh redistribution.

Step 4 Preserve divergence-free interpolation. It is necessary to keep the divergence-free feature in the interpolation, when solving incompressible flow with the moving mesh finite element method. As in Ref. [4], the solution re-distribution on the new mesh $\mathscr{T}^{(n+1)}$ can be achieved by solving a linearised inviscid Navier-Stokes-type system as follows:

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

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

The weak form of Eqs. (4.5) is : find $(\mathbf{u}_h, \hat{p}_H) \in X^h \times P^H$ such that

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

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

We use an explicit scheme in Eqs. (4.6) for the time discretisation, to obtain

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

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

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

Eqs. (4.7) lead to a linear system with the coefficient matrix

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

where M is the velocity mass matrix given in Eq. (3.7). As is well known, the Schur complement of the matrix $\mathscr{M}^p$ is $M_S=B_xM^{-1}B_x^T+B_yM^{-1}B_y^T$. From Ref. [18] Section 5, for LBB stable mixed approximations with enclosed flow boundary conditions and noting that M_S is the spectral equivalent with the pressure Laplacian matrix A_p, we use A_p for the appropriate Schur complement M_S. We then choose the block triangular preconditioner

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

for the coefficient matrix (4.8) where $M_S^*=A_p$ or $M_S^*=A_p/\nu$. In our practical computation, we compare the efficiency of A_p and A_p/ν and found that A_p/ν performs more efficiently than A_p. For the inflow/outflow problem, some modifications should be made for A_p on the Neumann boundary to improve efficiency. Thus along the outflow boundary $\partial\Omega_N$ the discrete pressure p_h has to satisfy a homogeneous Dirichlet boundary condition, while a Neumann condition $\partial p_h/\partial n=0$ is imposed on $\partial\Omega_D$ — cf. Section 3. It is notable that all of the matrices M, B_x^T, B_y^T, B_x, B_y and A_p have to be rebuilt after the meshes are moved.

In our algorithm, a PCD preconditioned GMRES is chosen to solve the linear system (3.2). The stopping criterion for GMRES convergence is

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

where $r^{(k)}$ is the residual of the linear system (3.2) and $r^{(0)}$ is its right-hand side. Our algorithm may thus be summarised as follows.

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

Algorithm 4.1 Moving mesh FEM for Navier Stokes equation

1: Solve steady Stokes flow to give the initial value $\mathbf{u}_h^{(0)}, p_H^{(0)}$.
2: **while** $t_n < T$ **do**
3: Calculate the monitor function on the mesh $\triangle_p^{(n)}$ using $\mathbf{u}_h^{(n)}$, $p_H^{(n)}$ and obtain the logical mesh ξ^* by solving (4.2).
4: Judge if the L_2 norm of $\xi^* - \xi^{(0)}$ is less than the tolerance. If yes, the iteration is complete, or else continue with steps 5 - 8.
5: Calculate the move direction $\delta\mathbf{x}$ of $\triangle_p^{(n)}$ using the difference of $\xi^* - \xi^{(0)}$.
6: Solve Eq. (4.7) on $\triangle_v^{(n)}$ to get the intermediate variable $\mathbf{u}_{h,*}^{(n)}, p_{H,*}^{(n)}$.
7: Update the mesh $\triangle_p^{(n)}$ to $\triangle_p^{(n+1)}$ and synchronise $\triangle_v^{(n)}$ to the $\triangle_v^{(n+1)}$ by the hierarchy geometry tree structure.
8: Return to step 3.
9: Solve the Navier-Stokes system (3.2) with the AMG preconditioner to obtain the 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**

5. Numerical Tests

We considered two numerical test problems to demonstrate our strategy where steady Stokes equations were invoked initially in solving the Navier-Stokes equations, with the logical domain the same as the physical domain in the moving algorithm. The moving mesh and numerical solutions are shown below. Our codes are based on the finite element package AFEPack.

5.1. Driven cavity flow

We considered a benchmark problem for regularised cavity flow on the domain $\Omega = [-1,1]\times[-1,1]$ for the Reynolds number $Re = 2000$. The Dirichlet boundary condition was imposed on $\partial\Omega$, with $\mathbf{u} = (1-x^4, 0)^T$ at the top boundary and the no-slip boundary condition elsewhere.

Eq. (4.1) was selected for the monitor function in our moving mesh strategy. The parameters $\alpha = 0.5$ and $\beta = 2.0$ performed well, with the moving mesh and vorticity contour evolution to the steady state shown in Fig. 3. It can be seen that the mesh clusters at the top and right boundaries, where the vorticity is large. Velocity streamlines are shown in Fig. 5. As shown in Fig. 4, the magnitude of the velocity divergence in the moving mesh is notably less than with the uniform mesh.

1. Comparison of GMRES iteration counts in solving Eq. (3.2).

   In Fig. 6, we compare the mean GMRES iteration steps for six groups of $F_p = \gamma A_p + W_p^n$ with different γ in the PCD preconditioning. Less iteration steps were required with $\gamma = 1$ than in the other five cases, with a satisfactory result obtained by choosing $F_p = A_p + W_p^n(\gamma = 1)$. For simplicity, we therefore omitted ν in $F_p = \nu A_p + W_p^n$,

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

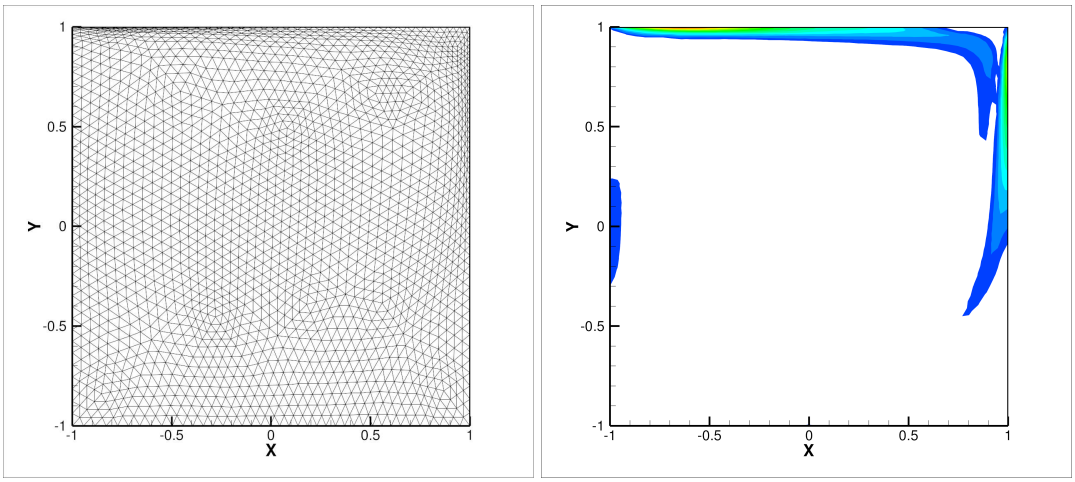


Figure 3: Cavity flow: mesh (left), vorticity contour (right), pressure mesh 20×20 and $Re = 2000$.

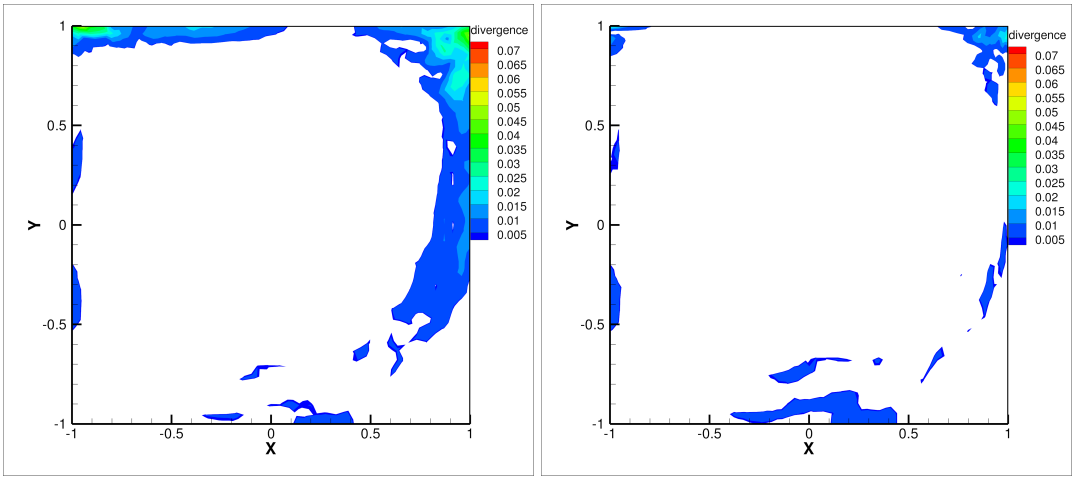


Figure 4: Cavity flow: divergence of velocity: uniform mesh (left), moving mesh (right), pressure mesh 20×20 and $Re = 2000$.

and contrasted the number of GMRES iteration counts in solving the linear system (3.2) with AMG and ILU preconditioners, as shown in Table 1. Evidently, the AMG preconditioner proved more efficient than the ILU preconditioner.

2. Comparison of GMRES iteration counts in solving Eqs. (4.7).

   The mean number of GMRES iteration steps needed to solve Eqs. (4.7) with different preconditioning matrices M_S^* is shown in Table 2, where the GMRES step counts are seen to decrease dramatically on using $M_S^* = A_p/\nu$ rather than $M_S^* = A_p$.

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

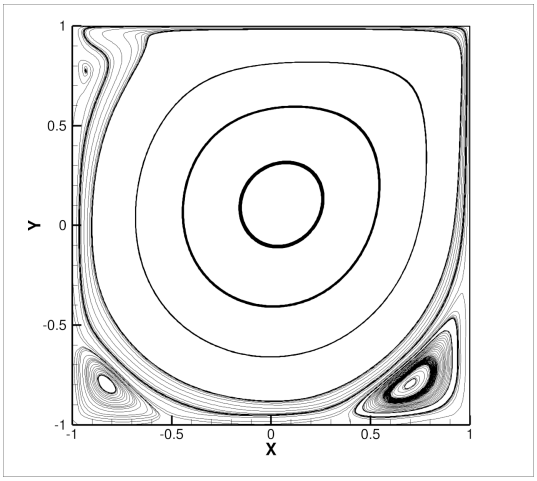


Figure 5: Cavity flow: velocity streamline, pressure mesh 20×20 and $Re = 2000$.

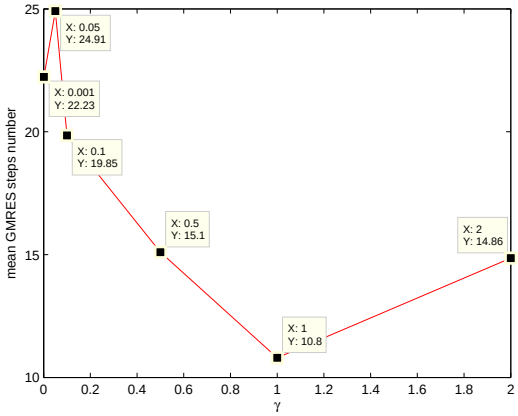


Figure 6: Cavity flow: comparison of mean GMRES step counts in solving the linear system (3.2) with different γ in $F_p = \gamma A_p + W_p^n$ for our AMG preconditioner, pressure mesh 20×20 and $Re = 2000$.

Table 1: Cavity flow: comparison of mean GMRES step counts in solving the linear system (3.2) with our preconditioning strategy and ILU precondtioning, $Re = 2000$.

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

5.2. Flow over cylinder

We considered the flow development over a cylinder along a rectangular channel — cf. Ref. [22]. The centre of the cylinder is at the origin $(0,0)$ and the cylindrical radius is $r =$

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

Table 2: Cavity flow: mean GMRES step counts in solving Eqs. (4.7) with different preconditioning matrices M_S^* and $Re = 2000$.

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

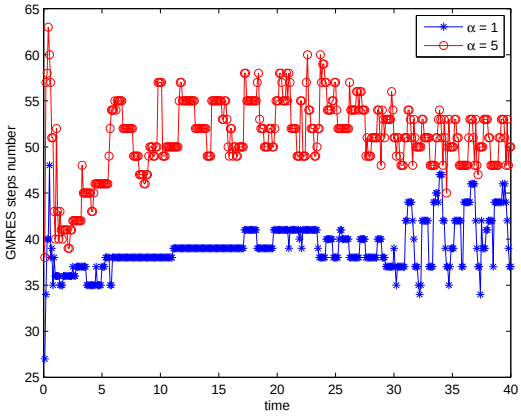


Figure 7: Flow over cylinder: GMRES iteration counts in solving (3.2) using our preconditioning strategy with different α and $Re = 240$.

0.3. For the viscosity coefficient $\nu = 1/300$, the Reynolds number is $Re = 2rU_m/\nu = 240$ where U_m is the mean inflow velocity, and the flow domain is $\Omega = [-1,5] \times [-1,1]$. We have $\mathbf{u} = (1-y^2, 0)^T$ (a Poiseuille profile) at the inflow boundary $x = -1$, $\mathbf{u} = (0,0)^T$ (no flow) at the top and bottom of the channel, and a natural condition imposed on $x = 5$.

In our moving mesh strategy, we recall that the parameters α and β in Eq. (4.1) are user-defined, and the degree of mesh clustering is larger when α is greater. From Fig. 7, it can be seen that the number of GMRES iteration steps with $\alpha = 5$ is larger than when $\alpha = 1.0$. Table 3 shows the mean GMRES step counts in solving the Eq. (3.2) from Eq. (4.7) with different choices of F_p in the PCD preconditioner and different M_S^*. It is seen that the mean number of GMRES iteration steps decreased by about 10 in using $F_p = A + W_p^n$, and it is more efficient to choose $M_S^* = A_p/\nu$ rather than $M_S^* = A_p$ as the preconditioning matrix.

We show the moving mesh at $t = 2s$ in Fig. 8, where the mesh obviously clusters around the cylinder. The wall street phenomenon is known to occur when the flow has an appropriate viscosity [23] — cf. also the mesh shown in Fig. 9.

6. Conclusions

We have applied an efficient AMG preconditioning strategy to a moving mesh finite element method based on a $4P1 - P1$ element pair, which naturally satisfies the inf-sup

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

Table 3: Flow over cylinder: the mean GMRES iteration steps in solving Eqs. (3.2) and (4.7) with different M_S^* and F_p in our preconditioning method, for $\alpha = 5$ and $Re = 240$.

linear system	mean number of GMRES steps			
	$F_p = A + W_p^n$		$F_p = \nu A + W_p^n$	
	$M_S^* = \frac{1}{\nu} A_p$	$M_S^* = A_p$	$M_S^* = \frac{1}{\nu} A_p$	$M_S^* = A_p$
(3.2)	51.06	45.13	61.06	61.08
(4.7)	22.25	40.35	23.77	40.78

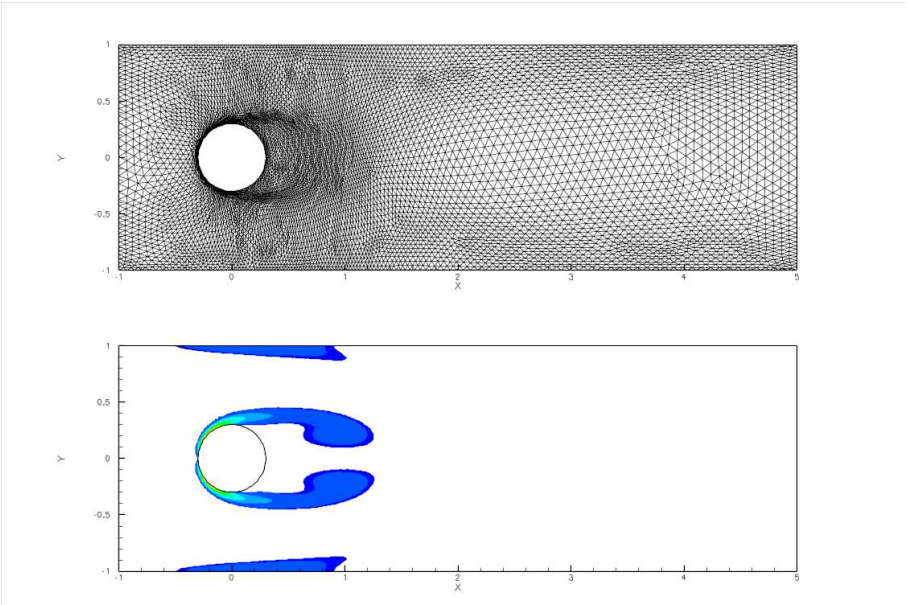


Figure 8: Flow over cylinder: moving mesh at $t = 2s$ for $Re = 240$.

condition and is linear-order. A linear element is preferred to a high-order element in practical engineering computation, on simplicity and problem complexity grounds. In our moving mesh strategy, we use a monitor function based on vorticity to capture the fine flow structure, and the mesh is consistent with the vorticity structure. We compared the number of GMRES iteration steps in solving Navier-Stokes problems with different F_p in PCD preconditioning, and demonstrated that choosing $F_p = A_p + W_p^n$ is more efficient via numerical tests. We also compared the number of GMRES step counts to solve the linear system in the solution re-distribution with different preconditioning matrices M_S^*, and found that $M_S^* = A_p/\nu$ is more efficient. The number of GMRES iteration steps became larger as the mesh clustering increased.

In future we intend to extend efficient preconditioning to some other applications such as free boundary and three-dimensional problems of incompressible flow, using the moving mesh finite element approach based on $4P1 - P1$ pairs.

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

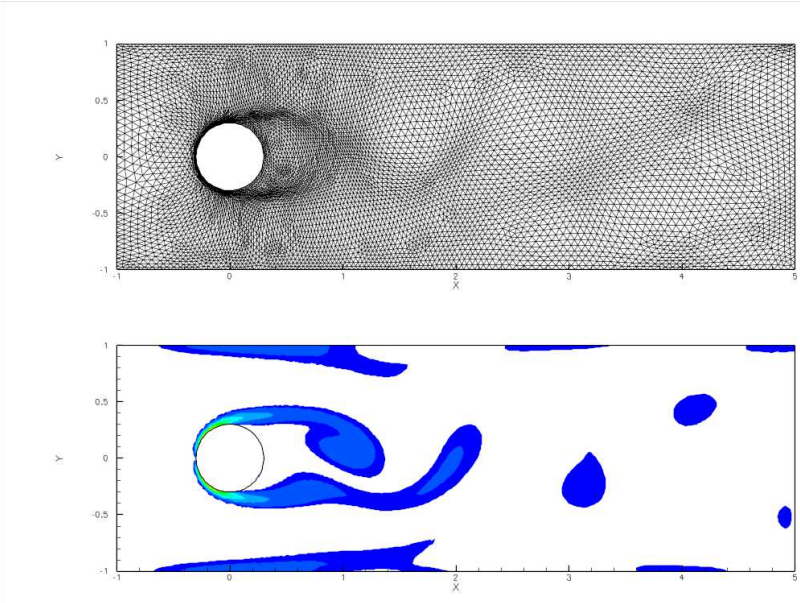


Figure 9: Flow over cylinder: moving mesh at $t = 40s$ for $Re = 240$.

Acknowledgments

This work was supported in part by the National Basic Research Program of China (Grant 2011CB309704) and the National Natural Science Foundation of China (Grants 11271358 and 91230108).

References

[1] A.M. Winslow, *Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh*, J. Comput. Phys. **135**, 128-138 (1966).

[2] A.S. Dvinsky, *Adaptive grid generation from harmonic maps on Riemannian manifolds*, J. Comput. Phys. **95**, 450-476 (1991).

[3] R. Li, T. Tang, P.W. Zhang, *Moving mesh methods in multiple dimensions based on harmonic maps*, J. Comput. Phys. **170**, 562-588 (2001).

[4] 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**, 1036-1056 (2005).

[5] Y.R. Wu and H.Y. Wang, *Moving mesh finite element method for unsteady Navier-Stokes flow*, East Asian J. Appl. Math. to appear.

[6] M. Bercovier and O. Pironneau, *Error estimates for finite element method solution of the Stokes problem in the primitive variables*, Numer. Math. **33**, 211-224 (1979).

[7] L. Shen and J.C. Xu, *On a Schur complement operator arisen from Navier-Stokes equations and its preconditioning*, Lecture Notes in Pure and Appl. Math. **202**, 481-490 (1999).

[8] J.C. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Rev. **34**, 581-613 (1992).

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press

[9] Y.N. He, *Two-level method based on finite element and Crank-Nicolson extrapolation for the time-dependent Navier-Stokes equations*, SIAM J. Numer. Anal. **41**, 1263-1285 (2003).
[10] M. Benzi, G.H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer. **14**, 1-137 (2005).
[11] Z.Z. Bai and M.K. Ng, *On inexact preconditioners for nonsymmetric matrices*, SIAM J. Sci. Comput. **26**, 1710-1724 (2005).
[12] Z.Z. Bai, *Structured preconditioners for nonsingular matrices of block two-by-two structures*, Math. Comp. **75**, 791-815 (2006).
[13] H. Elman, V.E. Howle, J. Shadid, D. Silvester, and R. Tuminaro, *Least squares preconditioners for stabilised discretisations of the Navier-Stokes equations*, SIAM J. Sci. Comput. **30**, 290-311 (2007).
[14] H.C. Elman and R. Tuminaro, *Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations*, Electron. Trans. Numer. Anal. **35**, 257-280 (2009).
[15] M. Benzi and M.A. Olshanskii, *An augmented Lagrangian-based approach to the Oseen problem*, SIAM J. Sci. Comput. **28**, 2095-2113 (2006).
[16] M. Benzi, M.K. Ng, Q. Niu, and Z. Wang, *A relaxed dimensional factorisation preconditioner for the incompressible Navier-Stokes equations*, J. Comput. Phys. **230**, 6185-6202 (2011).
[17] J. Boyle, M.D. Mihajlovic, and J.A. Scott, *Hsl_mi20: An efficient amg preconditioner for finite element problems in 3d*, Internat. J. Numer. Methods Engrg. **82**, 64-98 (2010).
[18] H.C. Elman, D.J. Silvester, and A.J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford (2014).
[19] R. Li, *On multi-mesh h-adaptive methods*, J. Sci. Comput. **24**, 321-341 (2005).
[20] H. Elman, M. Mihajlovi, and D. Silvester, *Fast iterative solvers for buoyancy driven flow problems*, J. Comput. Phys. **230**, 3900-3914 (2011).
[21] R. Li, T. Tang, and P.W. Zhang. *A moving mesh finite element algorithm for singular problems in two and three space dimensions*, J. Comput. Phys. **177**, 365-393 (2002).
[22] W.M. Cao, W.Z. Huang, and R.D. Russell, *An r-adaptive finite element method based upon moving mesh pdes*, J. Comput. Phys. **149**, 221-244 (1999).
[23] M.V. Dyke, *An Album of Fluid Motion*, Parabolic Press, Stanford (1982).

https://doi.org/10.4208/eajam.080416.240516a Published online by Cambridge University Press