

Pseudostress–velocity formulation for incompressible Navier–Stokes equations

Zhiqiang Cai¹ and Yanqiu Wang^{2,*}, †

¹*Department of Mathematics, Purdue University, West Lafayette, IN 47906, U.S.A.*

²*Department of Mathematics, Oklahoma State University, Stillwater, OK 74078, U.S.A.*

SUMMARY

This paper presents a numerical algorithm using the pseudostress–velocity formulation to solve incompressible Newtonian flows. The pseudostress–velocity formulation is a variation of the stress–velocity formulation, which does not require symmetric tensor spaces in the finite element discretization. Hence its discretization is greatly simplified. The discrete system is further decoupled into an $\mathbf{H}(\mathbf{div})$ problem for the pseudostress and a post-process resolving the velocity. This can be done conveniently by using the penalty method for steady-state flows or by using the time discretization for nonsteady-state flows. We apply this formulation to the 2D lid-driven cavity problem and study its grid convergence rate. Also, computational results of the time-dependent-driven cavity problem and the flow past rectangular problem are reported. Copyright © 2009 John Wiley & Sons, Ltd.

Received 22 July 2008; Revised 5 February 2009; Accepted 24 March 2009

KEY WORDS: pseudostress–velocity formulation; Navier–Stokes equations; finite element method; mixed finite element; driven cavity; flow past cylinder

1. INTRODUCTION

For decades, much research has been done on the stress–velocity/stress–velocity–pressure ($\boldsymbol{\sigma} - \mathbf{u}/\boldsymbol{\sigma} - \mathbf{u} - p$) formulations for Navier–Stokes equations [1–3]. These formulations attract many attentions because: (1) they come from the original physical laws and give a direct description of the stress, which in some applications is the most interesting variable and (2) formally they resemble the stress–displacement formulation of elasticity equations, which hopefully will give a better understanding of the coupled solid–fluid problem.

To further explain these advantages, we first give the stress–velocity–pressure formulation in below. Let Ω be a bounded domain in \mathbb{R}^n , where $n=2, 3$. Consider the behavior of a viscous, incompressible Newtonian fluid occupying Ω over a time period $t \in (0, T)$. The basic equations

*Correspondence to: Yanqiu Wang, Department of Mathematics, Oklahoma State University, Stillwater, OK 74078, U.S.A.

†E-mail: yqwang@math.okstate.edu

for the incompressible Newtonian flow can be stated as following: *Find the stress $\boldsymbol{\sigma}$, the velocity \mathbf{u} and the pressure p satisfying*

$$\begin{aligned} \partial \mathbf{u} / \partial t + \mathbf{u} \cdot \nabla \mathbf{u} - \operatorname{div} \boldsymbol{\sigma} &= \mathbf{f} && \text{(balance of momentum)} \\ \boldsymbol{\sigma} + p \mathbf{I} - \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^t) &= \mathbf{0} && \text{(constitutive law)} \\ \nabla \cdot \mathbf{u} &= 0 && \text{(conservation of mass)} \end{aligned} \quad (1)$$

where ν is the kinematic viscosity, \mathbf{I} is the identity tensor and \mathbf{f} is the external body force. Here, $\mathbf{u} = (u_i)_{1 \leq i \leq n}$ is an n -dimensional vector, $\boldsymbol{\sigma} = (\sigma_{ij})_{1 \leq i, j \leq n}$ is an $n \times n$ symmetric tensor and the divergence of $\boldsymbol{\sigma}$ is defined by $(\operatorname{div} \boldsymbol{\sigma})_i = \sum_{j=1}^n \partial_j \sigma_{ij}$, for $i = 1 \dots n$. The term $\mathbf{u} \cdot \nabla \mathbf{u}$ should be understood as a vector whose i th component is $\sum_{j=1}^n u_j \partial_j u_i$. To close the system, proper initial and boundary conditions need to be applied. In a mixed system like problem (1), either a Dirichlet boundary condition $\mathbf{u}|_{\Gamma_D} = \mathbf{g}$ or a Neumann boundary condition $\boldsymbol{\sigma} \mathbf{n}|_{\Gamma_N} = \mathbf{h}$, where \mathbf{n} is the unit outward normal, can be imposed. For simplicity, we only consider the formulation using pure Dirichlet boundary condition. An extension to a certain Neumann-type boundary condition and its implementation will be discussed later in Section 3.3. For now we consider the following initial and boundary conditions:

$$\mathbf{u}|_{t=0} = \mathbf{u}_0, \quad \mathbf{u}|_{\partial \Omega} = \mathbf{g}$$

System (1), which comes from the original physical laws, is usually called the stress–velocity–pressure formulation of incompressible Newtonian flows. Its formulation explains why it has the advantages mentioned in the beginning of this section.

However, one difficulty in solving problem (1) directly is that the discrete spaces for $\boldsymbol{\sigma}$ and \mathbf{u} have to satisfy the inf–sup condition (the LBB condition) [4], in order to prevent spurious modes to enter the approximation through the discrete gradient operator. We say that the discrete space is stable if it satisfies the inf–sup condition. It is extremely difficult to construct finite element spaces for the stress, which are symmetric and stable at the same time [4, 5]. The few currently available stable symmetric tensor spaces are very expensive, for example, 24 degrees of freedom (dofs) per triangle in 2D [5] and 162 dofs per tetrahedral in 3D [6]. This has been one of the most prohibitive aspects of the stress-based formulation. Many researchers have thus resorted to using augmented formulations [7–9] in which the symmetry is weakly imposed, or least-squares formulations [10–13], which do not require the LBB condition.

To avoid this difficulty while keeping advantages of the stress-based formulations, a new pseudostress–velocity formulation was proposed [14, 15] for Stokes equations. In this paper, we present algorithms based on the pseudostress–velocity formulation for solving Navier–Stokes equations. For steady-state problems, a Picard iteration is used to linearize the equations and the resulting linear system is decoupled by the penalty method. For nonsteady-state problems, the problem is linearized using a semi-implicit time discretization and the resulting linear system can be decoupled directly. In both cases, one ends up with solving a convective $\mathbf{H}(\operatorname{div})$ problem for the pseudostress.

A common criticism of stress-based formulation is that it seems to contain more variables than a velocity–pressure formulation, and hence will result in a larger problem. We will show in Section 2.4 that, by using the pseudostress–velocity formulation and decoupling as mentioned above, the resulting linear algebraic system is not necessarily larger than the system derived from a velocity–pressure formulation.

The paper is organized as follows. In Section 2, we describe the pseudostress–velocity formulation and its finite element discretization. Details about how to decouple the system for both steady-state and nonsteady-state problems are also explained. Section 3 is devoted to numerical experiments that demonstrate the accuracy and the convergence behavior of the pseudostress–velocity formulation. Finally, a conclusion is given in Section 4.

2. THE PSEUDOSTRESS–VELOCITY FORMULATION AND ITS FINITE ELEMENT DISCRETIZATION

In this section, we discuss the pseudostress–velocity formulation for Navier–Stokes equations together with its discretization and decoupling. Denote \mathbb{M}_n , $n = 2, 3$, to be the field of $n \times n$ tensors. Let $\mathcal{A}: \mathbb{M}_n \rightarrow \mathbb{M}_n$ be a fourth-order tensor defined by $\mathcal{A}\boldsymbol{\tau} = (1/\nu)(\boldsymbol{\tau} - ((1/n)\text{tr}\boldsymbol{\tau})\mathbf{I})$ for all $\boldsymbol{\tau} \in \mathbb{M}_n$. Here, $\text{tr}\boldsymbol{\tau}$ is the trace of $\boldsymbol{\tau}$. Recall that the symmetric stress in system (1) is defined as

$$\boldsymbol{\sigma} = -p\mathbf{I} + \nu(\nabla\mathbf{u} + (\nabla\mathbf{u})^t)$$

We define the non-symmetric pseudostress as

$$\tilde{\boldsymbol{\sigma}} = -p\mathbf{I} + \nu\nabla\mathbf{u}$$

Since

$$\mathbf{div}\boldsymbol{\sigma} - \mathbf{div}\tilde{\boldsymbol{\sigma}} = \nu\mathbf{div}(\nabla\mathbf{u})^t = \nu\nabla(\nabla\cdot\mathbf{u}) = 0$$

clearly, System (1) can be rewritten as:

$$\begin{aligned} \partial\mathbf{u}/\partial t + \mathbf{u}\cdot\nabla\mathbf{u} - \mathbf{div}\tilde{\boldsymbol{\sigma}} &= \mathbf{f} \\ \mathcal{A}\tilde{\boldsymbol{\sigma}} - \nabla\mathbf{u} &= \mathbf{0} \end{aligned} \tag{2}$$

Notice that $\mathcal{A}\tilde{\boldsymbol{\sigma}}$ is trace free, hence the incompressible constraint $\nabla\cdot\mathbf{u} = 0$ is satisfied through $\nabla\cdot\mathbf{u} = \text{tr}(\nabla\mathbf{u}) = 0$. Also, the pressure

$$p = -\frac{1}{n}\text{tr}\tilde{\boldsymbol{\sigma}}$$

is unique only up to a constant. Other physical quantities such as the velocity gradient, stress and vorticity can be expressed algebraically in terms of the pseudostress:

$$\nabla\mathbf{u} = \mathcal{A}\tilde{\boldsymbol{\sigma}}, \quad \boldsymbol{\sigma} = \tilde{\boldsymbol{\sigma}} + \nu(\mathcal{A}\tilde{\boldsymbol{\sigma}})^t, \quad \boldsymbol{\omega} = \frac{1}{2}(\mathcal{A}\tilde{\boldsymbol{\sigma}} - (\mathcal{A}\tilde{\boldsymbol{\sigma}})^t)$$

Similar to the pressure, they can all be computed in a post-process in the same accuracy as the approximation of $\tilde{\boldsymbol{\sigma}}$. Here, we conveniently represent the vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ as the skew symmetric part of the velocity gradient.

2.1. The spatial discretization

The discretization will be applied in a variational form. Let $L^2(\Omega)$ be the space of all square-integrable functions and $\mathbf{H}(\mathbf{div}, \Omega, \mathbb{M}_n)$ be defined by

$$\{\boldsymbol{\tau} \in \mathbb{M}_n \mid \tau_{ij} \in L^2(\Omega) \text{ and } \mathbf{div}\boldsymbol{\tau} \in (L^2(\Omega))^n\}$$

For convenience, denote (\cdot, \cdot) to be the L^2 inner-product of scalar, vector and tensor functions on Ω , and $\langle \cdot, \cdot \rangle_{\partial\Omega}$ the L^2 inner-product on $\partial\Omega$. For example, let \mathbf{u}, \mathbf{v} be the n -dimensional vectors and $\boldsymbol{\sigma}, \boldsymbol{\tau}$ be the $n \times n$ tensors on Ω , then

$$(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} \, dx, \quad (\boldsymbol{\sigma}, \boldsymbol{\tau}) = \int_{\Omega} \sum_{i,j=1}^n \sigma_{ij} \tau_{ij} \, dx$$

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\partial\Omega} = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{v} \, ds$$

The finite element approximation for system (2) is much easier than the one for system (1). Finite element spaces for $\tilde{\boldsymbol{\sigma}}$ and \mathbf{u} still need to satisfy the discrete inf-sup condition, according to the mixed finite element theory. However, since the pseudostress is not necessarily symmetric, simple mixed elements like the Raviart–Thomas (RT [16]) or Brezzi–Douglas–Marini [17] elements have been proved stable for the discretization of system (2) [14]. In this paper, for simplicity, we consider the lowest-order RT element (RT₀). Since the RT element is designed to approximate coupled vector functions and scalar functions, n copies of the RT element are used to approximate the coupled pseudostress and velocity.

For the reader's convenience, we describe the two-dimensional RT₀ space for the pseudostress–velocity formulation in detail. Let \mathcal{T}_h be a quasi-uniform triangular mesh or a rectangular mesh in a polygonal domain Ω . Define discrete spaces for the pseudostress and the velocity, respectively, by

$$\Sigma_h = \left\{ \boldsymbol{\sigma} = \begin{pmatrix} a_1 + c_1x & b_1 + c_1y \\ a_2 + c_2x & b_2 + c_2y \end{pmatrix} \text{ on each triangle in } \mathcal{T}_h \right.$$

$$\left. \text{or } \boldsymbol{\sigma} = \begin{pmatrix} a_1 + c_1x & b_1 + d_1y \\ a_2 + c_2x & b_2 + d_2y \end{pmatrix} \text{ on each rectangle in } \mathcal{T}_h \right.$$

$$\left. \text{and } \boldsymbol{\sigma}\mathbf{n} \text{ is continuous across the internal edges of } \mathcal{T}_h \right\}$$

$$\mathbf{V}_h = \left\{ \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \text{ where } v_1, v_2 \text{ are piecewise constants on } \mathcal{T}_h \right\}$$

The dofs for the pseudostress are the values of $\boldsymbol{\sigma}\mathbf{n}$ at the center of each edge in \mathcal{T}_h . Hence, each edge has two basis functions attached to it. The total number of basis functions is two times the total number of edges. The dofs for the velocity are the values of \mathbf{v} , which are piecewise constants, in each triangle or rectangle. The total number of basis functions is two times the total number of triangles/rectangles. In Figure 1, the dofs are illustrated.

The spatial discretization for (2) can be written as: find $\tilde{\boldsymbol{\sigma}} \in \Sigma_h / \text{span}\{\mathbf{I}\}$ and $\mathbf{u} \in \mathbf{V}_h$ such that

$$(\mathcal{A}\tilde{\boldsymbol{\sigma}}, \boldsymbol{\tau}) + (\mathbf{div} \boldsymbol{\tau}, \mathbf{u}) = \langle \mathbf{g}, \boldsymbol{\tau}\mathbf{n} \rangle_{\partial\Omega} \quad \text{for } \boldsymbol{\tau} \in \Sigma_h / \text{span}\{\mathbf{I}\}$$

$$(\mathbf{div} \tilde{\boldsymbol{\sigma}}, \mathbf{v}) - (\mathbf{u} \cdot \mathcal{A}\tilde{\boldsymbol{\sigma}}, \mathbf{v}) - (\partial\mathbf{u}/\partial t, \mathbf{v}) = -(\mathbf{f}, \mathbf{v}) \quad \text{for } \mathbf{v} \in \mathbf{V}_h$$
(3)

PSEUDOSTRESS-VELOCITY FORMULATION

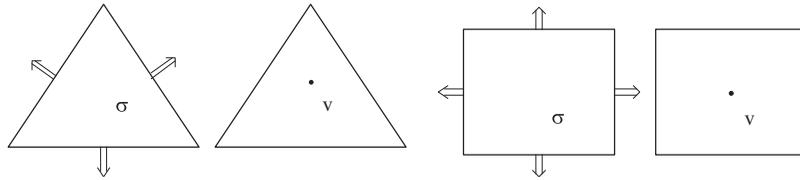


Figure 1. Degrees of freedom for Σ_h and V_h on each triangle and rectangle. Every double arrow and dot stand for two degrees of freedom corresponding to two components of σ or \mathbf{v} .

where $\text{span}\{\mathbf{I}\} = \{c\mathbf{I} \mid c \in \mathbb{R}\}$. One has to exclude $\text{span}\{\mathbf{I}\}$ from the trial and test spaces for the pseudostress, in order to make $p = -(1/n)\text{tr}\tilde{\sigma}$ unique.

There are two things that need to be explained in the discrete system (3). First, notice that the velocity boundary condition $\mathbf{u}|_{\partial\Omega} = \mathbf{g}$ becomes naturally imposed as a boundary integral on the right-hand side. This is common in the mixed finite element methods [4], in which Dirichlet boundary conditions become natural and Neumann boundary conditions become essential. In the pure Dirichlet boundary case, $\tilde{\sigma}$ should not admit any type of boundary constraints, otherwise the system will be over-determined. Discussion of how to impose Neumann boundary conditions will be given in Section 3.3. The second is about the space $\Sigma_h/\text{span}\{\mathbf{I}\}$, which seems to make the discretization complicated. However, in the implementation one can proceed using discrete space Σ_h directly. This will result in a rank 1 deficiency in the discrete system, which will not cause any trouble if the system is solved by a Krylov subspace iterative solver such as the GMRES method. A carefully designed multigrid solver can also deal with this rank deficiency. Further discussion on this issue can be found in [14, 15].

The mixed problem (3) will result in a large linear algebraic system. To reduce the problem size, we consider decoupling the system. The idea is to use the penalty method for steady-state problems and the time discretization for nonsteady-state problems. Details will be given in the next two subsections. The advantage of decoupling the system will be discussed in Section 2.4.

2.2. Steady-state case

To solve the time-independent problem, our strategy is to first use the Picard iteration to linearize it, and then use the penalty method to decouple the system.

Consider Equation (3) but without the time derivative term $(\partial\mathbf{u}/\partial t, \mathbf{v})$. Let $\tilde{\sigma}_k$ and \mathbf{u}_k be the approximate solution from the k th step of the iteration. In the $(k+1)$ th step, one makes a small increment so that $\tilde{\sigma}_{k+1} = \tilde{\sigma}_k + \delta\tilde{\sigma}_k$ and $\mathbf{u}_{k+1} = \mathbf{u}_k + \delta\mathbf{u}_k$ will be the next approximation. By substituting $\tilde{\sigma}_{k+1}$ and \mathbf{u}_{k+1} into Equation (3) and dropping the higher-order term $\delta\mathbf{u}_k \cdot (\mathcal{A} \delta\tilde{\sigma}_k)$, we obtain

$$\begin{aligned}
 (\mathcal{A}\tilde{\sigma}_{k+1}, \boldsymbol{\tau}) + (\mathbf{div} \boldsymbol{\tau}, \mathbf{u}_{k+1}) &= \langle \mathbf{g}, \boldsymbol{\tau} \mathbf{n} \rangle_{\partial\Omega} \\
 (\mathbf{div} \tilde{\sigma}_{k+1}, \mathbf{v}) - (\mathbf{u}_k \cdot (\mathcal{A}\tilde{\sigma}_{k+1}), \mathbf{v}) - (\mathbf{u}_{k+1} \cdot (\mathcal{A}\tilde{\sigma}_k), \mathbf{v}) &= -(\mathbf{f}, \mathbf{v}) - (\mathbf{u}_k \cdot (\mathcal{A}\tilde{\sigma}_k), \mathbf{v})
 \end{aligned}
 \tag{4}$$

Problem (4) is the linearization of the steady-state Navier–Stokes equations by the well-known Newton’s iteration. In general, Newton’s iteration has quadratic convergence rate, if the initial guess lies in the basin of attraction. However, decoupling of system (4) turns out to be difficult. Hence, we will explore an inexact linearization using the Picard iteration. In system (4), by further assuming that $\delta\mathbf{u}_k \cdot \mathcal{A}\tilde{\sigma}_k$ is small, we have

$$(\mathbf{u}_{k+1} \cdot (\mathcal{A}\tilde{\sigma}_k), \mathbf{v}) \approx (\mathbf{u}_k \cdot (\mathcal{A}\tilde{\sigma}_k), \mathbf{v})$$

Then the equation for the $(k + 1)$ th step of approximation becomes

$$\begin{aligned} (\mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \boldsymbol{\tau}) + (\mathbf{div} \boldsymbol{\tau}, \mathbf{u}_{k+1}) &= \langle \mathbf{g}, \boldsymbol{\tau} \mathbf{n} \rangle_{\partial\Omega} \\ (\mathbf{div} \tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{v}) - (\mathbf{u}_k \cdot \mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{v}) &= -(\mathbf{f}, \mathbf{v}) \end{aligned} \quad (5)$$

Problem (5) is exactly the pseudostress–velocity formulation of the Oseen problem, whose convergence was established in [18]. Picard iteration usually has linear convergence but a larger basin of attraction compared with Newton’s iteration.

By using the penalty method, an extra term is added to the mixed system (5) so that it becomes

$$\begin{aligned} (\mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \boldsymbol{\tau}) + (\mathbf{div} \boldsymbol{\tau}, \mathbf{u}_{k+1}) &= \langle \mathbf{g}, \boldsymbol{\tau} \mathbf{n} \rangle_{\partial\Omega} \\ (\mathbf{div} \tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{v}) - (\mathbf{u}_k \cdot \mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{v}) - \varepsilon(\mathbf{u}_{k+1}, \mathbf{v}) &= -(\mathbf{f}, \mathbf{v}) \end{aligned} \quad (6)$$

where the penalty constant ε is small. It has been proved in [14] that for Stokes problems, the penalty method does not deteriorate the accuracy of approximation provided that ε is at least of the same order as the discretization error. Therefore, when the RT_0 element is used in the discretization, setting $\varepsilon = O(h)$ will guarantee optimal convergence in the Stokes case. For the Navier–Stokes equations, we will examine numerically whether the same result holds or not, later.

By setting $\mathbf{v} = \mathbf{div} \boldsymbol{\tau}$, the mixed problem (6) can be decoupled into

$$(\mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \boldsymbol{\tau}) + \frac{1}{\varepsilon}(\mathbf{div} \tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{div} \boldsymbol{\tau}) - \frac{1}{\varepsilon}(\mathbf{u}_k \cdot \mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{div} \boldsymbol{\tau}) = \langle \mathbf{g}, \boldsymbol{\tau} \mathbf{n} \rangle_{\partial\Omega} - \frac{1}{\varepsilon}(\mathbf{f}, \mathbf{div} \boldsymbol{\tau}) \quad (7)$$

$$\mathbf{u}_{k+1} = \frac{1}{\varepsilon}(\mathbf{div} \tilde{\boldsymbol{\sigma}}_{k+1} - \mathbf{u}_k \cdot \mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1} + \mathbf{f}) \quad (8)$$

The main work of solving the decoupled system is to solve Equation (7), which is an $\mathbf{H}(\mathbf{div})$ problem with a convection term. The velocity \mathbf{u}_{k+1} can then be explicitly computed from (8).

2.3. Nonsteady-state case

To solve the time-dependent problem, a semi-implicit Euler scheme is used for time discretization. Given the solution of the k th time step, the solution of the $(k + 1)$ th time step is calculated by

$$\begin{aligned} (\mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \boldsymbol{\tau}) + (\mathbf{div} \boldsymbol{\tau}, \mathbf{u}_{k+1}) &= \langle \mathbf{g}, \boldsymbol{\tau} \mathbf{n} \rangle_{\partial\Omega} \\ (\mathbf{div} \tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{v}) - (\mathbf{u}_k \cdot \mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{v}) - \frac{1}{\Delta t}(\mathbf{u}_{k+1}, \mathbf{v}) &= -(\mathbf{f}_{k+1}, \mathbf{v}) - \frac{1}{\Delta t}(\mathbf{u}_k, \mathbf{v}) \end{aligned} \quad (9)$$

Again by setting $\mathbf{v} = \mathbf{div} \boldsymbol{\tau}$, system (9) can easily be decoupled into

$$(\mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \boldsymbol{\tau}) + \Delta t(\mathbf{div} \tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{div} \boldsymbol{\tau}) - \Delta t(\mathbf{u}_k \cdot \mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1}, \mathbf{div} \boldsymbol{\tau}) = \langle \mathbf{g}, \boldsymbol{\tau} \mathbf{n} \rangle_{\partial\Omega} - (\Delta t \mathbf{f}_{k+1} + \mathbf{u}_k, \mathbf{div} \boldsymbol{\tau}) \quad (10)$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta t(\mathbf{div} \tilde{\boldsymbol{\sigma}}_{k+1} - \mathbf{u}_k \cdot \mathcal{A}\tilde{\boldsymbol{\sigma}}_{k+1} + \mathbf{f}_{k+1}) \quad (11)$$

The majority work in the solution process is to solve Equation (10).

In practical simulation, one may want to use higher-order schemes in order to get better convergence rates. Again, it is essential to choose schemes that allow the decoupling as well.

2.4. The convective $\mathbf{H}(\mathbf{div})$ problem

Equations (7) and (10) have the common form

$$(\Lambda \tilde{\boldsymbol{\sigma}}, \boldsymbol{\tau}) \triangleq (\mathcal{A} \tilde{\boldsymbol{\sigma}}, \boldsymbol{\tau}) + \gamma(\mathbf{div} \tilde{\boldsymbol{\sigma}}, \mathbf{div} \boldsymbol{\tau}) - \gamma(\mathbf{u}_k \cdot \mathcal{A} \tilde{\boldsymbol{\sigma}}, \mathbf{div} \boldsymbol{\tau}) = F(\boldsymbol{\tau}) \quad (12)$$

where $\gamma = 1/\varepsilon \gg 1$ for the steady-state problem and $\gamma = \Delta t \ll 1$ for the nonsteady-state problem. Again, we mention that the boundary condition is imposed by $\langle \mathbf{g}, \boldsymbol{\tau} \mathbf{n} \rangle_{\partial\Omega}$ in the right-hand side. For pure Dirichlet problem, there should be no boundary condition for $\tilde{\boldsymbol{\sigma}}$. Otherwise, the problem will become ill-posed.

The $\mathbf{H}(\mathbf{div})$ problem (12), in its Stokes limit, has been studied in [14, 15], where optimal error estimates and multigrid convergence rates were proved. However, for the Navier–Stokes case and especially when the Reynolds number is large, the equation becomes convection dominated and its numerical simulation becomes difficult. To our knowledge, currently there is no theoretical or numerical study of such convective $\mathbf{H}(\mathbf{div})$ -type problems yet.

Decoupling of the system greatly reduces the size of the discrete problem. It is worth comparing the size of pseudostress–velocity formulation discretized using the RT_0 element with the size of velocity–pressure formulation using Crouzeix–Raviart elements [19, 20], since both approaches have the same accuracy. In n dimension where $n = 2, 3$, the total number of dofs for system (3) using the RT_0 element is $nN_f + nN_t$, and the total number is $nN_f + N_t$ for the velocity–pressure, where N_f, N_t are the number of faces/edges and elements, respectively. However, after decoupling, the $\mathbf{H}(\mathbf{div})$ equation (12) has a total size nN_f . In this case, the discrete pseudostress system is even smaller than the discrete velocity–pressure system. We also note that the velocity–pressure system, especially in the time-dependent case, cannot be naturally decoupled as in the case of pseudostress–velocity formulation.

3. NUMERICAL RESULTS

Our numerical experiments are done for two-dimensional problems only, although the formulation can be applied to three-dimensional cases. For now we would like to avoid the complications that arise in three dimensions, and focus the numerical study on the convergence and accuracy in 2D. Two types of problems, the lid-driven cavity and the flow past a rectangular cylinder, are considered.

3.1. Steady-state-driven cavity flow

The 2D lid-driven cavity problem, which describes the flow in a rectangular container driven by the uniform motion of one lid, is one of the most popular benchmark problems. Consider the Navier–Stokes equations in $\Omega = (0, 1) \times (0, 1)$, with boundary data $\mathbf{u} = (1, 0)^t$ on the top lid and $\mathbf{u} = (0, 0)^t$ everywhere else. One difficulty of this problem is that the velocity boundary data is discontinuous at the two top corners. Thus, in the velocity-based approximation, one needs to choose whether to use velocity $(1, 0)^t$, leaky, or $(0, 0)^t$, non-leaky at the two top corners. However, the pseudostress–velocity formulation handles this discontinuous boundary data naturally. The boundary term in Equation (12) is posed weakly using $\langle \mathbf{g}, \boldsymbol{\tau} \mathbf{n} \rangle_{\partial\Omega}$, which can be calculated on each boundary segment separately. There is no need to choose the velocity at the two top corners.

We solve this problem using the process described in Section 2. Uniform rectangular meshes and the RT_0 element are used in the discretization. Since the exact solution is not available,

Table I. Grid convergence rate for driven cavity.

h	$\ \mathbf{u}_h - \mathbf{u}_{h/2}\ _{L^2}$			
	Stokes	$Re=100$	$Re=100$	$Re=200$
	$\varepsilon=1e-4$	$\varepsilon=h$	$\varepsilon=1e-4$	$\varepsilon=h$
$\frac{1}{16}$	0.0513	0.0550	0.0540	0.0669
$\frac{1}{32}$	0.0290	0.0306	0.0302	0.0347
$\frac{1}{64}$	0.0160	0.0167	0.0165	0.0184
$\frac{1}{128}$	0.0087	0.0090	0.0089	0.0093
Asym. Order $O(h^k)$, $k=$	0.85	0.87	0.87	0.94

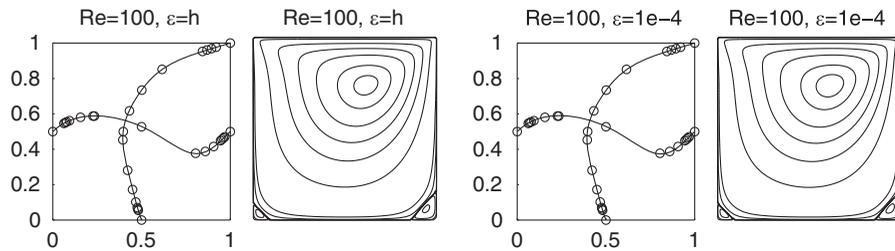


Figure 2. Velocity profiles and streamline portraits for $Re=100$ with $h = \frac{1}{128}$ and different ε . Circles in velocity profiles are solutions from [23].

a grid convergence study will be employed. In particular, the problem is solved on a sequence of meshes generated by a refining process. The error between solutions on consequent meshes is examined. In general, this should give a good indicator of the actual convergence rate. The calculated L^2 norm of the velocity, under different Reynolds number Re and penalty constant ε , is reported in Table I. We observe a convergence rate slightly less than $O(h)$. This is because the velocity is approximated by piecewise constants in the RT_0 element. The best possible convergence rate of piecewise constants is $O(h)$. Also, since the exact solution to the driven cavity problem has a corner singularity due to the discontinuous boundary data, it has been proved that the best possible convergence rate for this problem is $O(h(-\ln h))$ [21]. The results in Table I for $Re=100$ with different choices of ε also indicate that $\varepsilon=h$ is enough to guarantee convergence of the scheme. In all cases for $Re=100$ and $h \geq \frac{1}{128}$, the Picard iteration starts with the initial guess $\mathbf{u}_0 = \mathbf{0}$ and reaches $\|\mathbf{u}_{k+1} - \mathbf{u}_k\| \leq 10^{-6}$ within 20 steps. For $Re=100$ on a 256×256 mesh and $Re=200$, the Picard iteration starts with solutions computed from a coarse grid.

To check the accuracy of the solution with different penalty constants, we compare the velocity with published results in [22, 23]. Using a 128×128 mesh, velocity profiles on the vertical centerline for u_1 and on the horizontal centerline for u_2 , together with the streamline portraits, are plotted in Figure 2. Circles in velocity profiles are solutions from [23]. From the graph, the differences between solutions from different ε settings are not visible by eye. Thus, we also compare the

Table II. Extremes of u_1 on the vertical centerline and u_2 on the horizontal centerline for the time-independent Navier–Stokes-driven cavity flow with $Re = 100$.

h	$\varepsilon = 1e-4$			$\varepsilon = h$		
	$\min(u_1)$	$\max(u_2)$	$\min(u_2)$	$\min(u_1)$	$\max(u_2)$	$\min(u_2)$
$\frac{1}{16}$	-0.2059	0.1671	-0.2333	-0.1887	0.1493	-0.2077
$\frac{1}{32}$	-0.2121	0.1770	-0.2491	-0.2024	0.1665	-0.2343
$\frac{1}{64}$	-0.2136	0.1789	-0.2527	-0.2087	0.1734	-0.2450
$\frac{1}{128}$	-0.2139	0.1794	-0.2535	-0.2114	0.1766	-0.2496
$\frac{1}{256}$	-0.2140	0.1795	-0.2537	-0.2127	0.1781	-0.2518
[23]	-0.2109	0.17527	-0.24533			
[22]	-0.2140424	0.1795728	-0.253830			

extreme values of u_1 on the vertical centerline and u_2 on the horizontal centerline in Table II. Smaller ε gives better approximations.

Away from the two top corners, we also observe super-convergence of the pseudostress–velocity formulation. The pseudostress is approximated using the RT_0 element, which has jumps on tangential components across internal edges. A post-process is applied, which takes the average over jumps, to derive a continuous pseudostress. A similar post-process is applied to the discrete velocity and a continuous piecewise linear velocity is computed. The value of the continuous velocity at each node is the average of the piecewise constant velocity on the surrounding rectangles. Define $\Omega_1 = [(0, 1) \times (0, \frac{1}{2})] \cup [(\frac{1}{8}, \frac{7}{8}) \times (\frac{1}{2}, \frac{7}{8})]$. Ω_1 is a subset of Ω , which excludes a strip of width $\frac{1}{8}$ near each top corner. When $Re = 100$ and $\varepsilon = 1e-4$, the maximum norm

$$\|\phi_h\|_{\max} = \max_{x \in \Omega_1} |\phi_h(x) - \phi_{h/2}(x)|$$

for the post-processed pseudostress and velocity components are reported in Table III. The convergence rates of \mathbf{u} and the off-diagonal components of $\tilde{\boldsymbol{\sigma}}$ are near $O(h^2)$. The convergence rates of $\tilde{\sigma}_{11}$ and $\tilde{\sigma}_{22}$ suggest that the pressure $p = -\frac{1}{2}\text{tr}\tilde{\boldsymbol{\sigma}}$ does not have super-convergence. Similar super-convergence results are observed in the Stokes case with $\varepsilon = 1e-4$, but not in the $Re = 100$ case with $\varepsilon = h$. This is understandable, because the penalized system with $\varepsilon = h$ has an $O(h)$ dependence on the mesh size and hence super-convergence is not possible. When ε is small, but constant on all grids, the results given in Table III suggest certain level of super-convergence.

We have examined the convergence behavior and accuracy of the pseudostress–velocity formulation for the steady-state lid-driven cavity problem. A smaller ε is preferred, although $\varepsilon = h$ gives a reasonable solution. However, one problem with using small ε is that the system becomes ill-conditioned and hence it takes more time to solve. For all examples presented in this paper, the convective $\mathbf{H}(\mathbf{div})$ equation (12), in each Picard iteration or each time step, is solved with a multigrid algorithm as defined in [15]. This multigrid solver was originally designed for symmetric $\mathbf{H}(\mathbf{div})$ problems. Its performance deteriorates for the non-symmetric $\mathbf{H}(\mathbf{div})$ problem (12), especially when γ or the Reynolds number become large. The size of the coarsest mesh has to be ‘fine’ enough to ensure multigrid convergence [24]. In our experiments, when Reynolds number

Table III. Super-convergence in Ω_1 for $Re=100$ and $\varepsilon=1e-4$.

h	$\ \cdot\ _{\max}$					
	u_1	u_2	$\tilde{\sigma}_{11}$	$\tilde{\sigma}_{12}$	$\tilde{\sigma}_{21}$	$\tilde{\sigma}_{22}$
$\frac{1}{16}$	1.25e-2	2.57e-2	3.47e-3	7.74e-4	2.08e-3	3.89e-3
$\frac{1}{32}$	3.80e-3	6.94e-3	1.20e-3	3.06e-4	6.54e-4	1.30e-3
$\frac{1}{64}$	1.38e-3	2.02e-3	4.46e-4	6.57e-5	1.88e-4	4.96e-4
$\frac{1}{128}$	2.74e-4	6.03e-4	1.90e-4	1.21e-5	5.43e-5	1.96e-4
Asym. order $O(h^k)$, $k =$	1.80	1.80	1.39	2.02	1.75	1.43

is 100 and the penalty constant is h , the multigrid algorithm converges with a 16×16 coarsest mesh. However, when the penalty constant is set to be $1e-4$, the multigrid only converges with a 32×32 coarsest mesh (or 64×64 coarsest mesh when the finest mesh is 256×256). For larger Reynolds numbers, one needs an even larger coarsest mesh to ensure the multigrid convergence. Our experiments show that multigrid completely fails to converge when $Re=400$. Although a plain GMRES solver with Picard iteration still converges, it is very slow on large meshes. Because of the lack of an efficient solver, we did not perform experiments on steady-state lid-driven cavity with higher Reynolds numbers.

There are many possible ways to improve the efficiency of the multigrid algorithm for such kind of convection dominant problem. One can try special convection-related restriction, prolongation operators and smoothers in the multigrid algorithm. Another choice is to use an upwinding or streamline-diffusion finite element scheme. It is a good topic for future research.

3.2. Time-dependent solution of the driven cavity flow

Next we look at numerical results for the time-dependent-driven cavity problem, discretized and decoupled using the semi-implicit Euler method. For moderate values of the Reynolds number, it is known that the solution to the time-dependent-driven cavity problem converges to the steady-state solution, as time goes to infinity [22, 25]. Therefore, in our experiments, the time marching is stopped when $\|\mathbf{u}_h(t) - \mathbf{u}_h(t - \Delta t)\|_{L^2} \leq 10^{-6}$. Our numerical experiments show that multigrid algorithm works well in this case, if the time step Δt is sufficiently small. All following computations are done with a multigrid solver using a 2×2 coarsest mesh.

We first compute the time-dependent problem for $Re=100$ and $Re=200$, and compare the solutions at the final time-step with corresponding steady-state solutions computed by the penalty method, as shown in Section 3.1. A time-step size $\Delta t=0.02$ is used in the computation. The results given in Table IV verify that the time-stepping solution converges to the steady-state solution. Both the penalty constant and Reynolds number slightly affect the asymptotic behavior of the comparison.

We next examine the startup flows for $Re=400$ and $Re=1000$. Both problems are computed on a 128×128 mesh, with a seven-level multigrid solver. A time-step size $\Delta t=0.01$ is used in the computation. Such a small time step is needed to guarantee the multigrid convergence. We suspect that if plain GMRES is used in each time step, larger time-step size can be used in

Table IV. Difference between the time-stepping solution and the steady-state solution.

h	$\ \mathbf{u}_h(t_N) - \mathbf{u}_{h,\varepsilon}\ _{L^2} (Re=100)$			$\ \mathbf{u}_h(t_N) - \mathbf{u}_{h,\varepsilon}\ _{L^2} (Re=200)$	
	t_N	$\varepsilon=h$	$\varepsilon=1e-4$	t_N	$\varepsilon=h$
$\frac{1}{16}$	6	0.0106	0.0059	10.2	0.0223
$\frac{1}{32}$	7.4	0.0060	0.0031	12.8	0.0149
$\frac{1}{64}$	8.8	0.0032	0.0015	15	0.0081
$\frac{1}{128}$	10	0.0016	0.00075	17	0.0042
Asym. Order $O(h^k)$, $k =$		0.9091	0.9961		0.8105

Here, t_N is the time when the stopping criteria $\|\mathbf{u}_h(t) - \mathbf{u}_h(t - \Delta t)\|_{L^2} \leq 10^{-6}$ is reached, and $\mathbf{u}_{h,\varepsilon}$ is the steady-state solution computed using the penalty method with the penalty constant ε , as shown in Section 3.1.

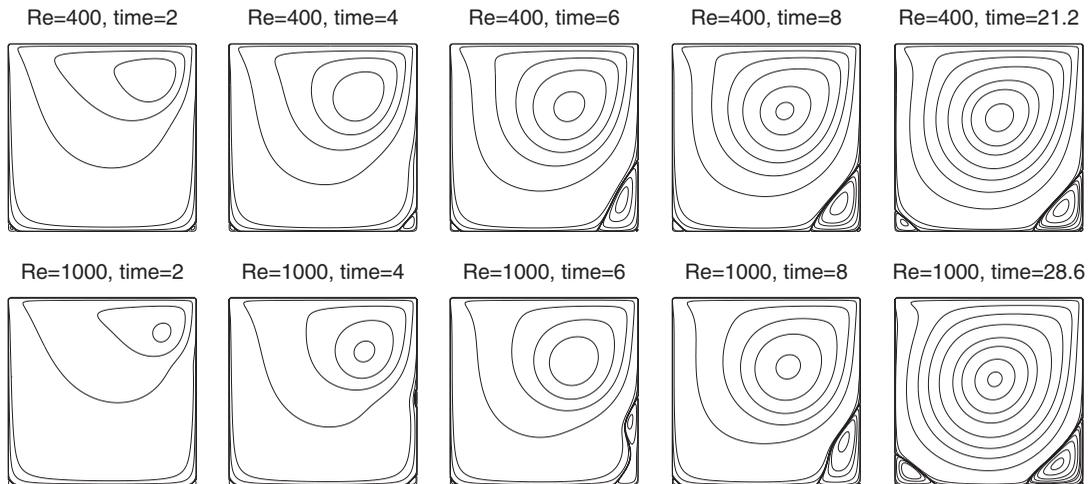


Figure 3. Startup flow for $Re=400$ and $Re=1000$ at different times.

the computation. However, since multigrid converges much faster than plain GMRES, we decided to sacrifice the time-step size for being able to use multigrid. Again, developing a more robust and efficient solver for the convective $\mathbf{H}(\mathbf{div})$ problem will be very helpful here.

The streamline portraits at several different times are given in Figure 3. The time marching terminates at $t=21.2$ for $Re=400$, and at $t=28.6$ for $Re=1000$. It is interesting to observe the obviously different behavior of startup flows at $Re=400$ and $Re=1000$. We also computed the solution for $Re=3200$ with mesh size 128×128 . Although the solution converges, numerical oscillations can be observed near two upper corners in the vorticity contour (see Figure 4). The oscillation disappears when using a 256×256 mesh and time step $\Delta t = 0.005$, as shown in Figure 4. We especially point this out because it has been reported in many circumstances [22, 23] that a 128×128 mesh is fine enough for the simulation of $Re=3200$. However, since the velocity in the

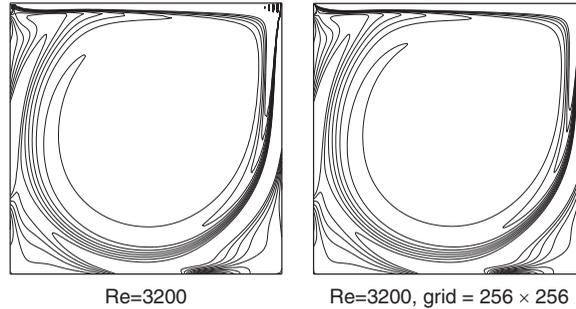


Figure 4. Vorticity portraits on a 128×128 mesh (left) and on a 256×256 mesh (right).

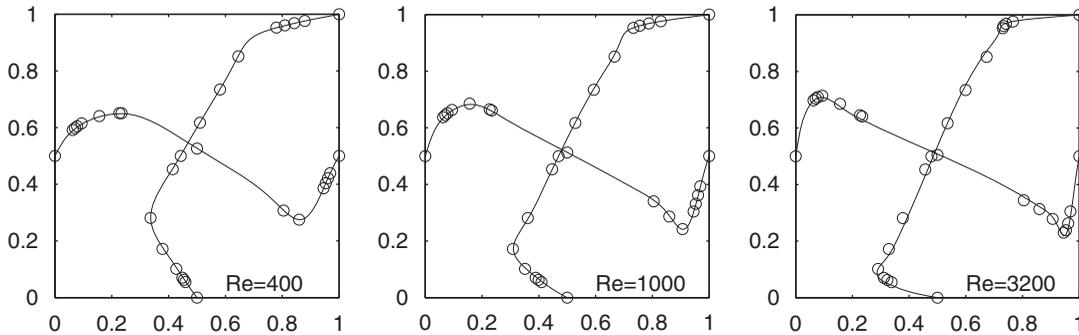


Figure 5. Velocity profiles on the centerlines on a 128×128 mesh. Circles denote results from [23].

pseudostress–velocity formulation is only approximated by piecewise constants, whose accuracy is an order lower than most other numerical methods, our experiments show that a mesh finer than 128×128 is needed to resolve all the boundary layers in this case. To get better approximation results, one can also use the RT_1 element that has a first-order approximation for the velocity, or an upwinding scheme.

Finally, to verify the accuracy of our numerical solution, in Figure 5 we report the velocity profile at a time when steady state is reached and compare it with results from [23].

3.3. Flow past a rectangular cylinder

The flow past a rectangular cylinder is known to generate dynamic patterns for low to moderate Reynolds numbers. To apply the pseudostress–velocity formulation to this problem, one has to carefully set the outflow boundary condition. In this paper, we set the problem as in Figure 6. The computational domain has dimension $L \times H = 22 \times 11$. A square cylinder with edge length $B = 1$ is positioned in the horizontal center and with the upstream extent $d = 5$. The boundary condition on the surface of blockage is set to be $\mathbf{u} = \mathbf{0}$. The boundary condition on the outer boundary is

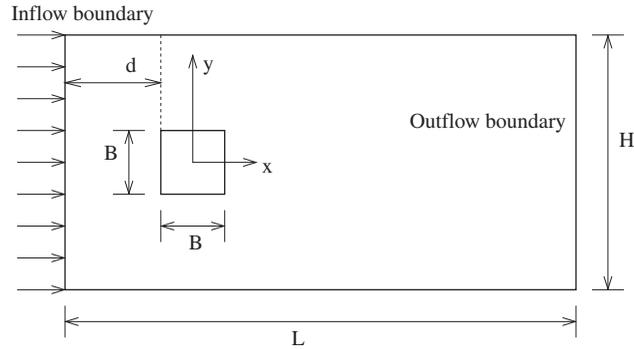


Figure 6. Flow past a rectangular cylinder.

set as

$$\text{Left wall: } \mathbf{u} = (1, 0)^t$$

$$\text{Top and bottom walls: } (\tilde{\boldsymbol{\sigma}}\mathbf{n})_1 = 0, \quad u_2 = 0$$

$$\text{Right wall: } \tilde{\boldsymbol{\sigma}}\mathbf{n} = -v \left(\frac{\partial \mathbf{u}}{\partial t} \right)_{\text{prev}}$$

Here, $(\partial \mathbf{u} / \partial t)_{\text{prev}}$ is the time derivative of \mathbf{u} calculated from previous time steps. Notice that on the top and bottom walls, the first component of $\tilde{\boldsymbol{\sigma}}\mathbf{n}$ is actually $(\tilde{\boldsymbol{\sigma}}\mathbf{n})_1 = \sigma_{12} = v \partial u_1 / \partial y$. Hence, this boundary condition is equivalent to $\partial u_1 / \partial y = 0$ and $u_2 = 0$. On the right wall (the outflow boundary), the boundary condition can be understood as an approximation to

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{v} \tilde{\boldsymbol{\sigma}}\mathbf{n} = 0$$

which is derived from the nonreflexive advective condition in [26] with the average convective velocity on the outflow boundary set to 1.

This boundary condition is of a mixed type, which contains both Dirichlet and Neumann parts. The Dirichlet part, or the velocity boundary data, is imposed as a boundary integral on the right-hand side. However, the Neumann part, or the pseudostress boundary data, should be set explicitly on $\tilde{\boldsymbol{\sigma}}$. Take the right wall as an example. Let $\mathbf{h} = -v(\partial \mathbf{u} / \partial t)_{\text{prev}}$, which is a vector of piecewise constants computed from the previous time step. Since the dofs of the RT_0 element are $\tilde{\boldsymbol{\sigma}}\mathbf{n}$ on each edge, one can simply set its value to be equal to \mathbf{h} on each boundary edge in the mesh \mathcal{T}_h .

We use the semi-implicit Euler scheme with time-step size 0.02 for the time discretization, and a five-level multigrid solver for the linear system in each time step. The problem is solved on uniform rectangular meshes and the finest level in multigrid is of size 352×176 . Flows with Reynolds numbers 100 and 200 are tested. In Figure 7, the drag and lift history are given. It can be seen that the drag coefficient oscillates twice as fast as the lift coefficient.

The calculated drag coefficients and Strouhal numbers are listed in Table V, and are compared with other published results [28–30, 27]. Owing to the large blockage rate limited by the size of the computational domain and the fact that the real flow is by nature a 3D phenomenon while

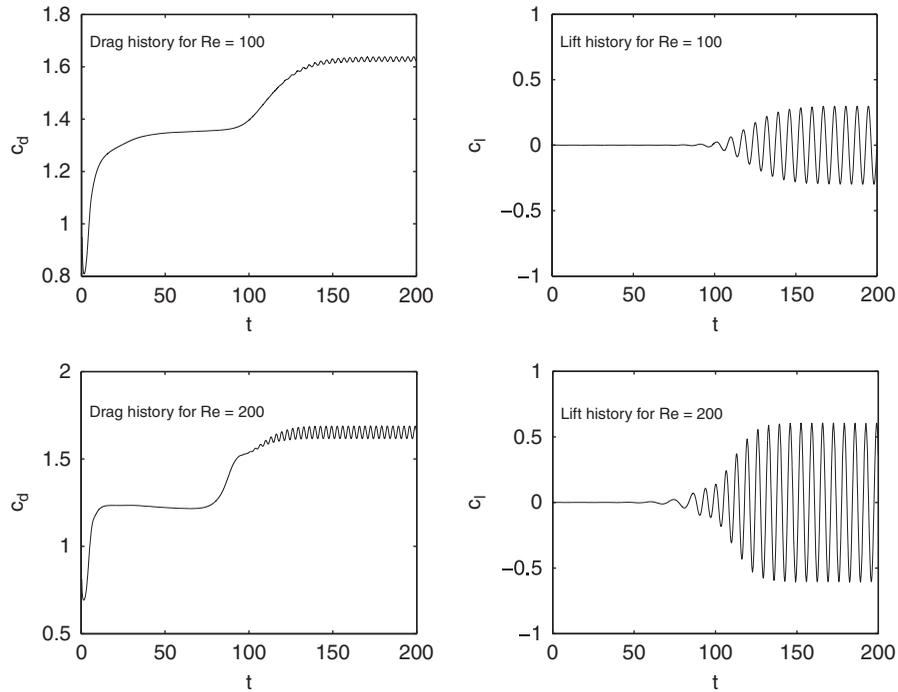


Figure 7. Drag coefficient and lift coefficient history for $Re=100$ and $Re=200$.

Table V. Comparison of drag coefficient (c_d) and Strouhal number (St).

Re	Source	B/H (%)	d	c_d	St
100	Present	9	5	1.6291	0.1442
100	Franke <i>et al.</i> [27]	7	11.1	1.483	0.149
100	Sohankar <i>et al.</i> [28]	7	18.3	1.466	0.142
100	Saha <i>et al.</i> [29]	10	6.5	1.51	0.159
100	Okajima (experimental) [30]	0	—	—	0.141–0.145
200	Present	9	5	1.6528	0.1507
200	Franke <i>et al.</i> [27]	8.3	4.5	1.65	0.157
200	Sohankar <i>et al.</i> [28]	5	11.1	1.424	0.165
200	Saha <i>et al.</i> [29]	10	6.5	1.67	0.163
200	Okajima (experimental) [30]	0	—	—	0.138–0.145

the numerical simulation here is done in 2D, it is known that the numerical Strouhal number is usually slightly larger than the actual number.

Numerical oscillations can be observed in the vorticity portrait for $Re=200$ solved on a 352×176 mesh. However, the oscillation disappears when the problem is solved on a 704×352 mesh. In Figure 8, vorticity profiles on these two meshes are given, both at time when the lift force is 0 and the drag force is at its minimum. The difference can be seen clearly. The entire computational

PSEUDOSTRESS–VELOCITY FORMULATION

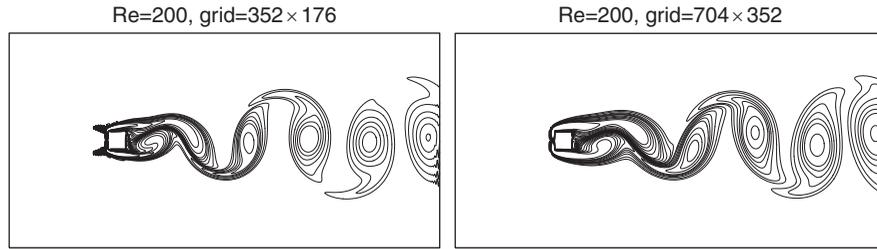


Figure 8. Vorticity contours for $Re=200$ on different meshes. Both at time when the lift force is 0 and the drag force is at its minimum.

domain is plotted in Figure 8. Notice that there seems to be little or no reflection on the outflow boundary.

4. CONCLUSION

We have presented a numerical algorithm using the pseudostress–velocity formulation for solving Navier–Stokes equations. By introducing a new variable, the non-symmetric pseudostress, this formulation avoids the common difficulty of discretizing the symmetric stress in other stress-based formulations, while it maintains the nice property of the direct calculation of stress. The resulting discrete system can be easily decoupled in both the steady-state and the nonsteady-state cases. A detailed comparison shows that the decoupled problem, discretized with the lowest-order RT element, has less dofs than the discretization of the velocity–pressure formulation, using the Crouzeix–Raviart element.

Numerical examples for steady-state and nonsteady-state problems are reported. The grid convergence analysis and the comparison to published results validate the accuracy of the pseudostress–velocity formulation. The pseudostress–velocity formulation copes with the discontinuous velocity boundary data naturally, as explained in the case of the driven cavity flow.

The problem we are facing now is to develop an efficient solver for the convective $\mathbf{H}(\mathbf{div})$ problem. The current multigrid solver has many limits, for example, the penalty constant cannot be too small or the time-step size cannot be large. Also, in some cases it requires the coarsest mesh to be ‘fine’ enough. All these needs to be improved, which provides a good topic for future research. An upwinding scheme might be able to boost the multigrid performance.

ACKNOWLEDGEMENTS

The authors would like to thank an anonymous referee for valuable comments and suggestions.

REFERENCES

1. Behr M, France L, Tezduyar T. Stabilized finite element methods for the velocity–pressure–stress formulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 1993; **104**:31–48.
2. Buscaglia G. The behaviour of several stress–velocity–pressure mixed finite elements for Newtonian flows. *International Journal for Numerical Methods in Fluids* 1993; **17**:99–113.

3. Gerritsma M, Phillips T. Compatible spectral approximations for the velocity–stress–pressure formulation of the Stokes problem. *SIAM Journal on Scientific Computing* 1999; **20**:1530–1550.
4. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer: New York, 1991.
5. Arnold DN, Winther R. Mixed finite element for elasticity. *Numerische Mathematik* 2002; **92**:401–419.
6. Adams S, Cockburn B. A mixed finite element method for elasticity in three dimensions. *Journal on Scientific Computing* 2005; **25**:515–521.
7. Arnold D, Brezzi F, Douglas JJ, Peers: a new mixed finite element for plane elasticity. *Japan Journal of Applied Mathematics* 1984; **1**:347–367.
8. Stenberg R. On the construction of optimal mixed finite element methods for the linear elasticity problem. *Numerische Mathematik* 1986; **48**:447–462.
9. Stenberg R. A family of mixed finite elements for the elasticity problem. *Numerische Mathematik* 1988; **53**:513–538.
10. Cai Z, Lee B, Wang P. Least-squares methods for incompressible Newtonian fluid flow: linear stationary problems. *SIAM Journal on Numerical Analysis* 2004; **42**:843–859.
11. Cai Z, Starke G. First-order system least squares for the stress–displacement formulation: linear elasticity. *SIAM Journal on Numerical Analysis* 2003; **41**:715–730.
12. Pontaza J, Reddy J. Spectral/*hp* least-squares finite element formulation for the Navier–Stokes equations. *Journal of Computational Physics* 2003; **190**:523–549.
13. Pontaza J, Reddy J. Space–time coupled spectral/*hp* least-squares finite element formulation for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 2004; **197**:418–459.
14. Cai Z, Tong C, Vassilevski P, Wang C. Mixed finite element methods for incompressible flow: stationary Stokes equations. *Numerical Methods for Partial Differential Equations*, to appear.
15. Cai Z, Wang Y. A multigrid method for the pseudostress formulation of Stokes problems. *SIAM Journal on Scientific Computing* 2007; **29**:2078–2095.
16. Raviart P, Thomas J. A mixed finite element method for second order elliptic problems. In *Mathematical Aspects of the Finite Element Method*, Galligani I, Magenes E (eds). Lecture Notes in Mathematics, vol. 606. Springer: Berlin, 1977.
17. Brezzi F, Douglas J, Marini L. Two families of mixed finite elements for second order elliptic problem. *Numerische Mathematik* 1985; **47**:217–235.
18. Karakashian OA. On a Galerkin–Lagrange multiplier method for the stationary Navier–Stokes equations. *SIAM Journal on Numerical Analysis* 1982; **19**:909–923.
19. Crouzeix M, Raviart P. Conforming and nonconforming finite element methods for solving the stationary Stokes equations. *RAIRO: Numerical Analysis* 1973; **R3**:33–76.
20. Rannacher R, Turek S. Simple nonconforming quadrilateral Stokes element. *Numerical Methods for Partial Differential Equations* 1992; **8**:97–111.
21. Cai Z, Wang Y. An error estimate for two dimensional stokes driven cavity flow. *Mathematics of Computation* 2009; **78**:771–787.
22. Botella O, Peyret R. Benchmark spectral results on the lid-driven cavity flow. *Computers and Fluids* 1998; **27**:421–433.
23. Ghia U, Ghia K, Shin C. High-Re solution for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
24. Bramble J. *Multigrid Methods*. Pitman Research Notes in Mathematics Series. Pitman: London, 1993.
25. Shankar P, Deshpande M. Fluid mechanics in the driven cavity. *Annual Review of Fluid Mechanics* 2000; **32**:93–136.
26. Orlandi I. A simple boundary condition for unbounded flows. *Journal of Computational Physics* 1976; **21**:251–269.
27. Franke R, Rodi W, Schonung B. Numerical calculation of laminar vortex-shedding flow past cylinders. *Journal of Wind Engineering and Industrial Aerodynamics* 1990; **35**:237–257.
28. Sohankar A, Davidson L, Norberg C. Vortex shedding: numerical simulation of unsteady flow around a square two-dimensional cylinder. *The 12th Australian Fluid Mechanics Conference*, Sydney, Australia, 1995.
29. Saha A, Muralidhar K, Biswas G. Transition and chaos in two-dimensional flow past a square cylinder. *Journal of Engineering Mechanics (ASCE)* 2000; **126**:523–532.
30. Okajima A. Strouhal numbers of rectangular cylinders. *Journal of Fluid Mechanics* 1982; **123**:379–398.