Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 994–1006
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

# An alternative least-squares formulation of the Navier–Stokes equations with improved mass conservation

J.J. Heys [a,*], E. Lee [b], T.A. Manteuffel [b], S.F. McCormick [b]

[a] Chemical Engineering Department, Arizona State University, Tempe, AZ 85287-6006, United States
[b] Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309, United States

Received 21 December 2006; received in revised form 26 February 2007; accepted 1 May 2007
Available online 13 May 2007

**Abstract**

The focus of this paper is on incompressible flows in three dimensions modeled by least-squares finite element methods (LSFEM) and using a novel reformulation of the Navier–Stokes equations. LSFEM are attractive because the resulting discrete equations yield symmetric, positive definite systems of algebraic equations and the functional provides both a local and global error measure. On the other hand, it has been documented for existing reformulations that certain types of boundary conditions and high-aspect ratio domains can yield very poor mass conservation. It has also been documented that improved mass conservation with LSFEM can be achieved by strengthening the coupling between the pressure and velocity. The new reformulation presented here is demonstrated to provide both improved multigrid convergence rates because it is differentially diagonally dominant and improved mass conservation over existing methods because it increases the pressure–velocity coupling along the inflow and outflow boundaries.


*Keywords:* Navier–Stokes; Conservation; Finite elements; Least-squares; Multigrid

## 1. Introduction

Least-squares finite element methods (LSFEM) are attractive because they are based on a minimization principle and yield discrete linear systems that are symmetric and positive definite. These benefits hold even for problems like the Navier–Stokes equations, which yield a saddle-point problem and indefinite linear systems for mixed finite element methods [3,4,8]. The Navier–Stokes equations are second-order equations, so a least-squares method based on the original, primitive variables would yield an undesirable linear system with a condition number that is proportional to $h^{-4}$ [14]. To avoid this difficulty, the second-order equations are typically reformulated as a system of first-order equations through the introduction of new variables. Additional, equivalent equations can be added to the first-order system of equations because the problem is cast as a least-squares minimization problem. For example, if the original system of equations contains

* Corresponding author.
*E-mail addresses:* jheys@asu.edu (J.J. Heys), tmanteuf@colorado.edu (T.A. Manteuffel), stevem@colorado.edu (S.F. McCormick).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.05.005

the equation $\nabla \cdot \mathbf{v} = 0$, the equation $\nabla(\nabla \cdot \mathbf{v}) = 0$ could be included in the system of equations without contradiction, assuming that $\mathbf{v}$ is sufficiently smooth. There is typically some flexibility in how the second-order equations are rewritten as first-order equations and flexibility in which additional equations are added to the first-order system [5,11,21]. This flexibility is often exploited to achieve a second advantage of LSFEM: some reformulations of the original equation(s) allow the system variables to be essentially decoupled so that the homogenous form is elliptic and continuous, providing significant computational advantages [9]. A third advantage is that the functional provides a sharp *a priori* error estimator [10].

Every approximate solution contains error, and the goal of numerical approximation is to minimize that error, possibly under additional constraints. The Navier–Stokes equations represent conservation of momentum and conservation of mass for a region containing an incompressible, Newtonian fluid. For mixed finite element methods, conservation of mass is enforced by the divergence-free constraint on the velocity field, which essentially uses pressure as a Lagrange multiplier. As a result, most of the error in the numerical approximation is in the conservation of momentum. In contrast, least-squares finite element methods distribute the error between the different terms in the functional, so the approximate solution can emphasize conservation of mass or conservation of momentum by simply changing the functional. When looking at an approximate solution, error in the conservation of mass is typically more obvious than error in the conservation of momentum. As a result, one of the often cited drawbacks of least-squares methods for the Navier–Stokes equations is that some approximate solutions do not conserve mass at an acceptable level between the inflow surface, $\Gamma_i$, and the outflow surface, $\Gamma_o$. Of course, by changing the functional, conservation of mass can be improved at the expense of the other terms in the functional [25]. An additional expense of improving mass conservation is that the performance of standard multigrid solvers (a popular choice for these elliptic operators [9,20]) may degrade.

Several methods exist for improving the accuracy of least-squares methods, and some of these are summarized below.

- Refining the mesh (i.e. reducing $h$) reduces the total approximation error and, consequently, the mass conservation error in an optimal manner [1,4,6].
- Because the solution to the Navier–Stokes equations is smooth away from the boundaries, the use of higher-order finite elements (i.e. increasing $p$) can dramatically improve accuracy [24].
- The use of reduced integration can result in a collocation method and a zero residual for mass conservation [11,21], although this also results in the loss of positive definiteness of the operator, which has a strong negative impact on a standard multigrid solver [14].

These methods change the space from which an approximate solution is sought, instead of changing the functional. The goal in this paper is to examine how changing the first-order system of equations (i.e. changing the functional) affects both the error in mass conservation and the performance of the solver. Almost all the previous efforts to change the functional to improve mass conservation have focused on adding a weight to the $\nabla \cdot \mathbf{v}$ term (i.e. mass conservation term) within the functional. This method improves mass conservation, but it can also have a negative impact on the performance of linear system solvers, including standard multigrid, as we show in the following section. One notable alternative to weighting the divergence of velocity term was recently proposed by Pontanza [23]. He used a regularized divergence free constraint (i.e. $\nabla \cdot \mathbf{v} = -\epsilon \delta p$) to improve the coupling between the pressure and velocity and demonstrated improved mass conservation. Also, we recently developed a new first-order formulation of the Navier–Stokes equations that is related to the one developed in this paper and also demonstrates improved mass conservation [18]. Because of the boundary conditions used, however, the method was limited to problems where the velocity on the inflow and outflow boundaries was near the steady flow profile, a limitation that is not present for the method developed in this paper. Also, the operator associated with the first-order formulation was not differentially diagonally dominant, a weakness that is overcome with the formulation used here.

Our interest here is in three-dimensional problems involving laminar, incompressible fluid flow through a rigid conduit, such as air flow through the upper branches of the human airway. For all problems, we assume the inlet velocity is fully specified, the velocity along walls is set to zero, and, along the outlet, the tangential velocity and the normal gradient of the normal velocity are set to zero. These are the only boundary conditions

that are known, and any other boundary conditions specified on, for example, vorticity, are derived from these known conditions. The goal is to reformulate the Navier–Stokes equations as a first-order system of equations in a novel manner to improve global mass conservation. Global mass conservation is measured in terms of the fractional change of mass flow, defined as

$$\frac{\int_{\Gamma_i}(\mathbf{n}\cdot\mathbf{v})\,d\Gamma_i - \int_{\Gamma_o}(\mathbf{n}\cdot\mathbf{v})\,d\Gamma_o}{\int_{\Gamma_i}(\mathbf{n}\cdot\mathbf{v})\,d\Gamma_i}, \tag{1}$$

where $\mathbf{v}$ is the velocity of the incompressible fluid, $\Gamma_i$ is the inlet surface, and $\Gamma_o$ is the outlet surface. Global mass conservation is (or at least should!) be satisfied automatically for problems with the velocity specified on all boundaries. While this class of problems is not the focus here, it should be stated that mass conservation can still be a difficulty for LSFEM, but the methods presented can reduce (but not entirely prevent) the loss of mass. LSFEM are currently in use in part because the resulting linear systems can be effectively solved using standard multigrid solvers. Therefore, an additional goal is to improve (or at least maintain!) multigrid performance relative to existing reformulations. In the following section, existing reformulations are examined briefly and their performance on a representative test problem is measured, thus providing a baseline for comparison with the new least-squares formulation.

## 2. Background

The Navier–Stokes equations for an incompressible, Newtonian fluid are

$$-\sqrt{Re}(\mathbf{v}\cdot\nabla\mathbf{v}) - \nabla p + \frac{1}{\sqrt{Re}}\Delta\mathbf{v} = 0 \quad \text{in } \Omega, \tag{2}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega, \tag{3}$$

where $p$ is the non-dimensional pressure, $Re$ is the Reynolds number, and $\mathbf{v} = (v_x, v_y, v_z)$ is the dimensionless velocity. This is an unusual scaling of the conservation of momentum equation in that the Reynolds number appears on both the convective and viscous terms. The most common way to reformulate (2) and (3) as a first-order system is to define the vorticity, $\omega$, which is equal to the curl of the velocity and leads to the first-order system

$$\begin{aligned} &\nabla\times\mathbf{v} - \omega = 0 \quad \text{in } \Omega,\\ &\sqrt{Re}(\mathbf{v}\cdot\nabla)\mathbf{v} + \nabla p + \frac{1}{\sqrt{Re}}\nabla\times\omega = 0 \quad \text{in } \Omega,\\ &\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega. \end{aligned} \tag{4}$$

This first-order system can be augmented by the following consistent equation:

$$\nabla\cdot\omega = 0 \quad \text{in } \Omega. \tag{5}$$

Eqs. (4) and (5) may be combined in a least-squares sense into a single functional given by

$$G_\omega(\omega,\mathbf{v},p) := \|\nabla\times\mathbf{v} - \omega\|_{0,\Omega}^2 + \|\sqrt{Re}(\mathbf{v}\cdot\nabla)\mathbf{v} + \nabla p + \frac{1}{\sqrt{Re}}\nabla\times\omega\|_{0,\Omega}^2 + \|\nabla\cdot\mathbf{v}\|_{0,\Omega}^2 + \|\nabla\cdot\omega\|_{0,\Omega}^2. \tag{6}$$

The geometry, boundary conditions, and parameters for the first test problem are given in Table 1. The problem shares many of the characteristics of the real world problems that we are interested in solving. These characteristics include a moderately high aspect ratio, a fully specified inlet velocity, and an outlet velocity that is only partly specified. The Reynolds number is also sufficiently low to ensure laminar flow, but not so low that it is in the Stokes regime. Before the functional (6) can be minimized, the domain is subdivided into a finite element mesh and a basis for the approximation is chosen. The brick-shaped domain is divided into $32\times 8\times 8$ hexahedral elements, and a triquadratic basis is used for all variables. The algorithmic details for minimizing the functional are given in the following section.

Fig. 1 shows the central cross-section of the first test problem when the vorticity functional, $G_\omega$, is minimized. Of the mass that enters the domain, 99.3% is lost before it leaves the domain, which is clearly unacceptable. The

Table 1
Geometry, parameters, discretization, and boundary conditions used for the first test problem

| | |
|---|---|
| Geometry | Brick: $0 \leqslant x \leqslant 5$, $0 \leqslant y \leqslant 1$, $0 \leqslant z \leqslant 0$ |
| Mesh | $32 \times 8 \times 8$ hexahedral elements |
| Basis | Triquadratics (all variables) |
| Inlet | $\mathbf{n} \cdot \mathbf{v} = -y(1-y)z(1-z)$ |
| Boundary | $\mathbf{n} \times \mathbf{v} = 0$ |
| Conditions | $\mathbf{n} \cdot \omega = 0$ |
| Outlet | $\mathbf{n} \cdot \nabla(\mathbf{v} \cdot \mathbf{n}) = 0$ |
| Boundary | $\mathbf{n} \times \mathbf{v} = 0$ |
| Conditions | $\mathbf{n} \cdot \omega = 0$ |
| Wall | $\mathbf{n} \cdot \mathbf{v} = 0$ |
| Boundary | $\mathbf{n} \times \mathbf{v} = 0$ |
| Conditions | $\mathbf{n} \cdot \omega = 0$ |
| *Re* | 100 |

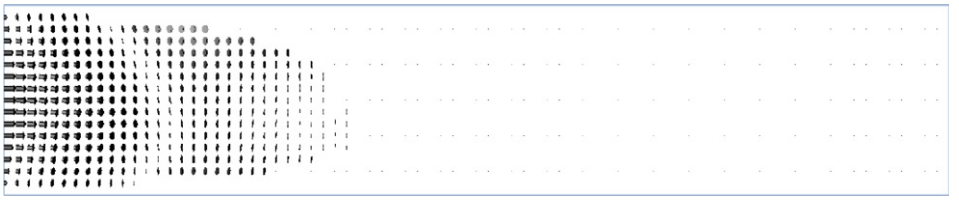

Fig. 1. Central cross-section of flow through the first test problem using the standard vorticity formulation of the Navier–Stokes equations.

standard method for improving mass conservation is to multiply the $\nabla \cdot \mathbf{v}$ term in the functional by a constant, $\alpha$, greater than 1.0. For example, replacing Eq. (4) by $10 \cdot \nabla \cdot \mathbf{v} = 0$ results in the mass loss being reduced to 71%, and using $100 \cdot \nabla \cdot \mathbf{v} = 0$ results in the mass loss being only 0.2%.

To understand the effects of different first-order formulations on the rate of convergence of a standard multigrid solver, it is useful to examine the weak form being used. Defining the following operators:

$$\mathscr{L}_\omega = \begin{bmatrix} \nabla\times & 0 & -I \\ \alpha\nabla\cdot & 0 & 0 \\ \sqrt{Re}(\mathbf{v}\cdot\nabla) & \nabla & \frac{1}{\sqrt{Re}}\nabla\times \\ 0 & 0 & \nabla\cdot \end{bmatrix} \quad \text{and} \quad u = \begin{bmatrix} \mathbf{v} \\ p \\ \omega \end{bmatrix}$$

allows the system of Eqs. (4),(5) to be rewritten as

$$\mathscr{L}_\omega u = f.$$

The weak form, obtained using standard variational calculus techniques [1,4], is

$$\langle \mathscr{L}_\omega u, \mathscr{L}_\omega v \rangle = \langle f, \mathscr{L}_\omega v \rangle \quad \forall v \in \mathscr{V}. \tag{7}$$

Multiplying the left side by $\mathscr{L}_\omega^*$, the formal adjoint of $\mathscr{L}_\omega$ (formal in the sense that full Dirichlet boundary conditions are assumed), allows the two operators on the left to be combined together into the formal normal, $\mathscr{L}_\omega^* \mathscr{L}_\omega$, written with $\sqrt{Re}$ replaced with 0 and $\frac{1}{\sqrt{Re}}$ replaced with 1 for simplicity as

$$\begin{bmatrix} \nabla\times & -\alpha\nabla & 0 & 0 \\ 0 & 0 & -\nabla\cdot & 0 \\ -I & 0 & \nabla\times & -\nabla \end{bmatrix} \begin{bmatrix} \nabla\times & 0 & -I \\ \alpha\nabla\cdot & 0 & 0 \\ 0 & \nabla & \nabla\times \\ 0 & 0 & \nabla\cdot \end{bmatrix} = \begin{bmatrix} (\nabla\times\nabla\times - \alpha^2\nabla\nabla\cdot) & 0 & -\nabla\times \\ 0 & -\nabla\cdot\nabla & 0 \\ -\nabla\times & 0 & (I + \nabla\times\nabla\times - \nabla\nabla\cdot) \end{bmatrix}. \tag{8}$$

The goal is to obtain a formal normal that has differential diagonal dominance, by which we mean that $\mathscr{L}^*_\omega \mathscr{L}_\omega$ is diagonally dominant and the entries of the off-diagonal are of lower differential order than those on the diagonal. This differential diagonal dominance holds for the vorticity formulation, and, as a result, the performance of the algebraic multigrid (AMG) preconditioned conjugate gradient (CG) solver, which is summarized in Table 2, is acceptable considering that a triquadratic nodal basis was used [7,19]. Solver performance is measure using the average convergence factor, $\rho$, defined as

$$\rho \approx \sqrt[m]{\frac{\|\mathrm{res}^{(m)}\|_2}{\|\mathrm{res}^{(0)}\|_2}}, \tag{9}$$

where $\mathrm{res}^{(m)}$ is the residual (or defect) after the $m$th iteration. In other words, if $\rho = 0.1$, then the residual is reduced on average by a factor of 10 each preconditioned CG iteration.

A second reformulation of the Navier–Stokes equations that has been previously published is the velocity-flux formulation, obtained by defining a new variable, $\mathbf{V}$, that is equal to the gradient of the velocity:

$$\begin{aligned} &\mathbf{V} - \nabla \mathbf{v} = 0 \quad \text{in } \Omega, \\ &-\sqrt{Re}(\mathbf{v}\cdot\mathbf{V}) - \nabla p + \frac{1}{\sqrt{Re}}\nabla\cdot\mathbf{V} = 0 \quad \text{in } \Omega, \\ &\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega. \end{aligned} \tag{10}$$

The first-order system of equations is typically augmented by the following consistent equations:

$$\begin{aligned} &\nabla(\mathrm{tr}(\mathbf{V})) = 0 \quad \text{in } \Omega, \\ &\nabla\times\mathbf{V} = 0 \quad \text{in } \Omega, \end{aligned} \tag{11}$$

where $\mathrm{tr}(\mathbf{V}) = V_{11} + V_{22} + V_{33}$. This formulation has a formal normal that is diagonally dominant, though not differentially so. This "algebraic" diagonal dominance does, however, yield a number of theoretical and practical advantages, including optimal error estimates in the $H^1$ norm for each variable and optimal multigrid convergence estimates due to the fact that it is coercive in $H^1$ [2,6,9]. The corresponding functional for the velocity-flux formulation is

$$G_{v.f.}(\mathbf{V},\mathbf{v},p) := \|\mathbf{V}-\nabla\mathbf{v}\|^2_{0,\Omega} + \|-\sqrt{Re}(\mathbf{v}\cdot\mathbf{V}) - \nabla p + \frac{1}{\sqrt{Re}}\nabla\cdot\mathbf{V}\|^2_{0,\Omega} + \|\nabla\cdot\mathbf{v}\|^2_{0,\Omega} + \|\nabla(\mathrm{tr}(\mathbf{V}))\|^2_{0,\Omega} + \|\nabla\times\mathbf{V}\|^2_{0,\Omega}. \tag{12}$$

The use of the velocity-flux formulation on the first test problem is summarized in Table 2. The only change to the boundary conditions that is required is that the $\mathbf{n}\cdot\omega$ conditioners are dropped in favor of tangential boundary conditions on $\mathbf{V}$. Multigrid convergence factors for the velocity-flux formulation show improvement over those for the vorticity formulation. Further, mass loss is also slightly improved, but still not acceptable without some weighting of the $\nabla\cdot\mathbf{v}$ term. The one disadvantage of the velocity-flux formulation relative to the vorticity formulation is that it requires 13 unknowns per node, compared to 7 unknowns per node for the vorticity formulation. Therefore, the results shown in Table 2 required 20 processors for the velocity-flux

Table 2
Performance of existing first-order formulations of the Navier–Stokes equation on the first test problem

| Formulation | $\alpha$ | Mass loss (%) | $\rho$ |
|---|---|---|---|
| Vorticity | 1.0 | 99.3 | 0.86 |
| | 10.0 | 71 | 0.91 |
| | 100.0 | 0.2 | 0.91 |
| Velocity-flux | 1.0 | 76 | 0.80 |
| | 10.0 | 3.9 | 0.82 |
| | 100.0 | 0.04 | 0.88 |

Here, $\alpha$ is the weighting on the $\nabla\cdot\mathbf{v}$ term in the functional and $\rho$ is the average convergence factor for the AMG preconditioned CG solver.

formulation relative to 10 processors for the vorticity formulation. Of course, it could be argued that the velocity-flux formulation would still produce a more accurate answer on a coarser grid (and require fewer processors). In summary, neither of the existing LSFEM for the Navier–Stokes equations provide an acceptable level of mass conservation for the test problem used here unless a significant weight ($\alpha$) is used on the $\nabla \cdot \mathbf{v}$ term. Of course, both methods would provide acceptable mass conservation with a smaller weighting if a finer mesh (or a higher-order basis) had been used. Our goal in the next section is to develop a new functional that provides a reduction in mass loss on these coarser grids without a degradation in solver performance for this class of flow problems.

## 3. New formulation

It has recently been demonstrated that improving the coupling between the pressure and velocity, especially through the use of boundary conditions, improves mass conservation with LSFEM [18,23]. To achieve stronger coupling between pressure and velocity, the new formulation uses the following identity:

$$(\mathbf{v} \cdot \nabla)\mathbf{v} = \frac{1}{2}\nabla|\mathbf{v}|^2 - \mathbf{v} \times (\nabla \times \mathbf{v}), \tag{13}$$

and we define a new variable, $\mathbf{r}$, by

$$\mathbf{r} = \nabla p + \frac{\sqrt{Re}}{2}\nabla|\mathbf{v}|^2 = \nabla\left(\frac{\sqrt{Re}}{2}|v|^2 + p\right), \tag{14}$$

which is the gradient of the the total head (commonly referred to as the gradient of "pressure" [1]) and includes both pressure and velocity. Our new first-order system reformulation of the Navier–Stokes equations is as follows:

$$\begin{aligned}
&\nabla \times \mathbf{v} + \omega = 0 \quad \text{in } \Omega,\\
&\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega,\\
&\frac{1}{\sqrt{Re}}\nabla \times \omega - \mathbf{r} + \sqrt{Re}(\mathbf{v} \times \omega) = 0 \quad \text{in } \Omega,\\
&\nabla \cdot \omega = 0 \quad \text{in } \Omega,\\
&\nabla \times \mathbf{r} = 0 \quad \text{in } \Omega,\\
&\nabla \cdot \mathbf{r} + \sqrt{Re}(\omega \cdot \omega) + Re(\mathbf{v} \cdot \mathbf{r}) = 0 \quad \text{in } \Omega.
\end{aligned} \tag{15}$$

The first two equations in system (15) are the same as those previously used in the vorticity formulation except that the vorticity is now defined as the *negative* curl of velocity. The third equation is the momentum balance, rewritten in terms of $\mathbf{r}$ and the curl of vorticity. The fourth equation is derived by taking the divergence of the first equation. The fifth equation comes from taking the curl of Eq. (14), and the final equation is the result of taking the divergence of the momentum balance, and simplifying as follows:

$$\begin{aligned}
\nabla \cdot \mathbf{r} &= \sqrt{Re}(\nabla \cdot (\mathbf{v} \times \omega))\\
&= \sqrt{Re}(\omega \cdot \nabla \times \mathbf{v} - \mathbf{v} \cdot \nabla \times \omega)\\
&= \sqrt{Re}\left(-\omega \cdot \omega - \mathbf{v} \cdot \left(\sqrt{Re}\,\mathbf{r} - Re(\mathbf{v} \times \omega)\right)\right)\\
&= -\sqrt{Re}(\omega \cdot \omega) - Re(\mathbf{v} \cdot \mathbf{r}) - Re\sqrt{Re}(\mathbf{v} \cdot (\mathbf{v} \times \omega))\\
&= -\sqrt{Re}(\omega \cdot \omega) - Re(\mathbf{v} \cdot \mathbf{r}).
\end{aligned} \tag{16}$$

The corresponding functional for the new formulation is

$$\begin{aligned}
G_{\text{new}}(\omega, \mathbf{v}, \mathbf{r}) := {} & \|\omega + \nabla \times \mathbf{v}\|^2_{0,\Omega} + \|\nabla \cdot \mathbf{v}\|^2_{0,\Omega} + \| - \mathbf{r} + \frac{1}{\sqrt{Re}}\nabla \times \omega + \sqrt{Re}(\mathbf{v} \times \omega)\|^2_{0,\Omega} + \|\nabla \cdot \omega\|^2_{0,\Omega}\\
&+ \|\nabla \times \mathbf{r}\|^2_{0,\Omega} + \|\nabla \cdot \mathbf{r} + \sqrt{Re}(\omega \cdot \omega) + Re(\mathbf{v} \cdot \mathbf{r})\|^2_{0,\Omega}.
\end{aligned} \tag{17}$$

To further analyze this system of equations, it is useful to rewrite it using slack variables, $\alpha$, $\beta$, and $\delta$, and moving subprincipal terms to the right-hand side:

$$\begin{aligned}
&\nabla \times \mathbf{v} - \nabla\alpha = -\omega \\
&\nabla \cdot \mathbf{v} = 0 \\
&\frac{1}{\sqrt{Re}}\nabla \times \omega - \nabla\beta = \mathbf{r} - \sqrt{Re}(\mathbf{v} \times \omega) \\
&\nabla \cdot \omega = 0 \\
&\nabla \times \mathbf{r} - \nabla\delta = 0 \\
&\nabla \cdot \mathbf{r} = -\sqrt{Re}(\omega \cdot \omega) - Re(\mathbf{v} \cdot \mathbf{r}).
\end{aligned} \tag{18}$$

Here, the slack variables are included to help analyze the new formulation, but the boundary conditions are set so that they equal zero everywhere and, hence, are not necessary in the computation. The system of equations given by (18) contains a block diagonal operator with three blocks. Each block consists of a curl and divergence of a variable we are interested in approximating. With the proper boundary conditions on each block (specifically, one condition on the slack variable and one on the original variable in the normal direction or, alternatively, conditions on the original variable in the two tangential directions), it is easy to show that the functional associated with (18) and $Re = 0$ is continuous and coercive in the $(H(\mathrm{div}) \cap H(\mathrm{curl})) \times H^1$ norm for each block of variables. This result is obtained by using a standard compactness argument.

The formal normal for the principle part (neglecting nonlinear terms) of the new formulation is simply

$$\begin{bmatrix} \nabla\times -\nabla & & \\ & \nabla\times -\nabla & \\ & & \nabla\times -\nabla \end{bmatrix} \begin{bmatrix} \nabla\times & & \\ \nabla\cdot & & \\ & \nabla\times & \\ & \nabla\cdot & \\ & & \nabla\times \\ & & \nabla\cdot \end{bmatrix} = \begin{bmatrix} -\Delta & & \\ & -\Delta & \\ & & -\Delta \end{bmatrix}, \tag{19}$$

which has Laplacian operators, $\Delta \equiv \nabla \cdot \nabla$, along the diagonal. The new formulation thus has a differentially diagonally dominant formal normal, even with the nonlinear terms included, as desired. We can see here where the previous efforts to remove derivatives from the $\nabla \cdot \mathbf{r}$ term, described in (16), allow for this property. The structure of the operator for this new formulation also creates the possibility of breaking the problem up into smaller subproblems. If the Reynolds number is zero and normal boundary conditions on $\mathbf{r}$ are available, the lower right part of the operator can be solved independent of $\mathbf{v}$ and $\omega$. Then, once $\mathbf{r}$ is known, the central block can be solved for $\omega$, and finally the upper left block can be solved for $\mathbf{v}$. There are two potential difficulties with the approach. First, the boundary conditions that we are proposing to use on $\mathbf{r}$ in the following section depend on the value of $\mathbf{v}$ or $\omega$. Second, in the case of a nonzero Reynolds number, the block strategy must be repeated multiple times until convergence. However, this method, which we explore later, has the advantage of a significant memory savings.

To solve the least-squares problem, the equations that are used in the functional ($G_{\mathrm{new}}$) are first linearized so that the solution can be found using a Gauss–Newton approach. The value of the nonlinear functional is calculated after each Gauss–Newton step to ensure that the nonlinear functional is decreasing to a minimum. The least-squares weak form, Eq. (7), is converted into a linear system of equations by choosing a finite element basis. An alternative approach is to linearize the weak form instead of linearizing the original equations [12,13], but we found better convergence by first linearizing the equations. All of the simulations presented here utilized a triquadratic finite element basis for all of the variables. The LSFEM allows the solution spaces for the variables to be chosen independently, and there is no restrictive stability condition (i.e. inf-sup condition) to satisfy. As a result, all variables in the reformulation of the Navier–Stokes equations can be approximated with the same basis.

All simulations, including those in the previous sections, were performed using the ParaFOS code, written by the authors. The code imports hexahedral meshes from the Cubit mesh generation package (Sandia National Laboratory). The finite element meshes are then partitioned using the Metis graph partitioning

library [22]. The software is designed to run on distributed memory clusters using the MPI library for communication. The linear matrix problem generated during each Gauss–Newton step is solved using the *hypre* library of solvers (from the Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, see [15]). Specifically, the BoomerAMG parallel algebraic multigrid solver is used as a preconditioner for a conjugate gradient iteration. Most of the solver parameters used were the default settings except for the strong threshold, which was set to zero, and the truncation factor, which was set to 0.3.

### 3.1. Boundary conditions

Three basic types of boundaries are present in the class of problems we are targeting with this new formulation. First are inflow boundaries where the velocity is fully specified, with the normal velocity generally being nonzero. Second are outflow boundaries where only the tangential velocity and the normal derivative of the normal velocity are specified. Finally, a no-slip condition is set along all the walls. The boundary conditions on velocity and vorticity in the new formulation are identical to those set for the original vorticity formulation. For a more complete discussion of possible boundary conditions on velocity and vorticity, see [1]. When enforcing boundary conditions with LSFEM, two mechanisms are available. First, boundary conditions can be enforced strongly by requiring the finite element space to satisfy the conditions. This is analogous to enforcing essential or Dirichlet boundary conditions with the Galerkin finite element method. A second option is to include boundary conditions in the functional so that the condition is satisfied weakly in a least-squares sense. For all the results shown here, boundary conditions involving a derivative are set weakly, and all other boundary conditions (i.e. Dirichlet conditions on velocity and vorticity) are set strongly.

The major change between the new formulation and the original vorticity formulation is the introduction of a new variable, $\mathbf{r}$, equal to the gradient of the total pressure. A number of options are available with regard to boundary conditions on $\mathbf{r}$. First, the simplest solution is to not enforce any boundary conditions on $\mathbf{r}$ because the problem is well-posed with only the velocity fully specified on all boundaries. Unfortunately, the convergence rate of a multigrid solver can deteriorate significantly if boundary conditions are not present on this lower block of the operator. A second option, available only for wall boundaries, is to set $\mathbf{n}\cdot\mathbf{r}=0$. This is almost never physically accurate, but the error is small enough that this is a common practice when solving the pressure Poisson equation [17]. The remaining two options involve taking the inner product of the momentum equation with the normal vector as follows:

$$\begin{aligned}\mathbf{n}\cdot\mathbf{r} &= \frac{1}{\sqrt{Re}}(\mathbf{n}\cdot\nabla\times\omega)+\sqrt{Re}(\mathbf{n}\cdot(\mathbf{v}\times\omega)) = \frac{1}{\sqrt{Re}}(\mathbf{n}\cdot\nabla\times\omega)+\sqrt{Re}((\mathbf{n}\times\mathbf{v})\cdot\omega)\\ &= \frac{1}{\sqrt{Re}}(\mathbf{n}\cdot\nabla\times\omega) = \frac{1}{\sqrt{Re}}(\mathbf{n}\cdot\nabla\cdot\nabla\mathbf{v}).\end{aligned} \tag{20}$$

Table 3
Boundary conditions used for the new first-order formulation of the Navier–Stokes equations

| | |
|---|---|
| Inlet | $\mathbf{n}\cdot\mathbf{v}=g$ |
| Boundary | $\mathbf{n}\times\mathbf{v}=0$ |
| Conditions | $\mathbf{n}\cdot\omega=0$ |
| | $\mathbf{n}\cdot\mathbf{r}-\mathbf{n}\cdot\frac{1}{\sqrt{Re}}\nabla\times\omega=0$ or |
| | $\mathbf{n}\cdot\mathbf{r}-\mathbf{n}\cdot\frac{1}{\sqrt{Re}}\nabla\cdot\nabla\mathbf{v}=0$ |
| Outlet | $\mathbf{n}\cdot\nabla(\mathbf{v}\cdot\mathbf{n})=0$ |
| Boundary | $\mathbf{n}\times\mathbf{v}=0$ |
| Conditions | $\mathbf{n}\cdot\omega=0$ |
| | $\mathbf{n}\cdot\mathbf{r}-\mathbf{n}\cdot\frac{1}{\sqrt{Re}}\nabla\times\omega=0$ or |
| | $\mathbf{n}\cdot\mathbf{r}-\mathbf{n}\cdot\frac{1}{\sqrt{Re}}\nabla\cdot\nabla\mathbf{v}=0$ |
| Wall | $\mathbf{n}\cdot\mathbf{v}=0$ |
| Boundary | $\mathbf{n}\times\mathbf{v}=0$ |
| Conditions | $\mathbf{n}\cdot\omega=0$ |
| | $\mathbf{n}\cdot\mathbf{r}-\mathbf{n}\cdot\frac{1}{\sqrt{Re}}\nabla\times\omega=0$ or |
| | $\mathbf{n}\cdot\mathbf{r}-\mathbf{n}\cdot\frac{1}{\sqrt{Re}}\nabla\cdot\nabla\mathbf{v}=0$ |

The final two relationships in (20) can be used as boundary conditions on $\mathbf{r}$ for surfaces where the tangential velocity is zero, as summarized in Table 3. A triquadratic or higher-order finite element basis is required to use the boundary condition involving the second derivative of velocity. If the tangential velocity is not zero, an additional term is required in the boundary condition. It is important to note that neither boundary condition is sufficient to make the operator $H^1$-elliptic, but, as we show in the next section, it is enough provide good multigrid performance and more accurate (but not perfect) mass conservation.

## 4. Results

The same test problem used in Section 2 to examine the performance of the vorticity and velocity-flux formulations can also be used to test the new formulation. As discussed in the previous section, however, there are two possible boundary conditions that can be used on the variable $\mathbf{r}$. Table 4 summarizes the performance of the new formulation using the two different boundary conditions. The first section of the table shows the results when using $\mathbf{n}\cdot\mathbf{r} = \mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\times\omega\right)$ on both the wall and inlet/outlet surfaces. The results show acceptable mass conservation (98%) when the weighting on $\nabla\cdot\mathbf{v}$ term, $\alpha$, is equal to 10.0. Recall that, with the original vorticity formulation, setting $\alpha$ equal to 10.0 still resulted in only 29% of the mass being conserved. Also, the average convergence factor of the linear solver, AMG preconditioned CG, is slightly better than either of the previous formulations. Considering that a triquadratic basis is being used here, the convergence rate is surprisingly good [19]. When the inlet and outlet boundary conditions on $\mathbf{r}$ are changed to $\mathbf{n}\cdot\mathbf{r} = \mathbf{n}\cdot(\frac{1}{\sqrt{Re}}\nabla\cdot\nabla\mathbf{v})$, there is only a small impact on the performance of the new formulation. Similarly, when the $\mathbf{n}\cdot(\frac{1}{\sqrt{Re}}\nabla\cdot\nabla\mathbf{v})$ form of the boundary condition is used on all boundaries, both the convergence factors and mass conservation are only slightly changed. It should be noted that normally, weak Dirichlet boundary conditions are scaled by $1/h$ so that the $L_2$-norm resembles an $H^{1/2}$-norm. For this second-order derivative term on velocity, a scaling of $h^3$ was used on the $L_2$-norm so that it resembles an $H^{1/2}$-norm on velocity, but a scaling of $h^1$ also led to convergence.

The results of these test indicate that setting a boundary condition of $\mathbf{n}\cdot\mathbf{r} = \mathbf{n}\cdot(\frac{1}{\sqrt{Re}}\nabla\times\omega)$ or $\mathbf{n}\cdot\mathbf{r} = \mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\cdot\nabla\mathbf{v}\right)$ on the inlet/outlet and walls as well as weighting the $\nabla\cdot\mathbf{v}$ term by 10 leads to an efficient algorithm. Fig. 2 shows the velocity for the central cross-section of the test problem using $\alpha = 10.0$ and $\mathbf{n}\cdot\mathbf{r} = \mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\times\omega\right)$. This method is used for all the additional test problems, unless otherwise noted.

The second problem is similar to the first except that a cylindrical obstruction with a diameter of 0.4 is placed in the path of the flow. A cross-section of the three-dimensional, all-hexahedral mesh is shown in Fig. 3a. The mesh is regular in the dimension normal to the figure, and it consists of 1960 elements, about three quarters of that of the first test problem. The boundary conditions are identical to those of the first test problem, with the obstruction being treated as a wall or no-slip boundary and $\mathbf{n}\cdot\mathbf{r} = \mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\times\omega\right)$ set as the $\mathbf{r}$ boundary condition on all boundaries. The Reynolds number is also unchanged at 100. Using a triquadratic

Table 4
Performance of the new formulation of the Navier–Stokes equation on the first test problem with different boundary conditions on $\mathbf{n}\cdot\mathbf{r}$

| Inlet/outlet | Wall | $\alpha$ | Mass loss (%) | $\rho$ |
|---|---|---|---|---|
| $\mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\times\omega\right)$ | $\mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\times\omega\right)$ | 1.0 | 31 | 0.75 |
| | | 10.0 | 2 | 0.73 |
| | | 100.0 | 0.05 | 0.84 |
| $\mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\cdot\nabla\mathbf{v}\right)$ | $\mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\times\omega\right)$ | 1.0 | 26 | 0.75 |
| | | 10.0 | 2 | 0.74 |
| | | 100.0 | 0.03 | 0.86 |
| $\mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\cdot\nabla\mathbf{v}\right)$ | $\mathbf{n}\cdot\left(\frac{1}{\sqrt{Re}}\nabla\cdot\nabla\mathbf{v}\right)$ | 1.0 | 26 | 0.77 |
| | | 10.0 | 2 | 0.77 |
| | | 100.0 | 0.03 | 0.89 |

Here, $\alpha$ is the weighting on the $\nabla\cdot\mathbf{v}$ term in the functional and $\rho$ is the convergence factor of the AMG preconditioned CG solver.

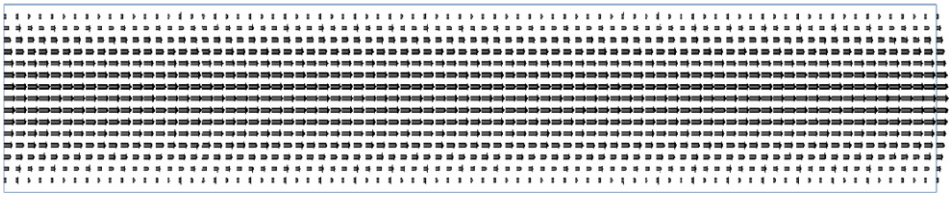

Fig. 2. Central cross-section of flow through the first test problem using the new formulation of the Navier–Stokes equations with $\alpha = 10$ and $\mathbf{n} \cdot \mathbf{r} = \mathbf{n} \cdot \left(\frac{1}{\sqrt{Re}} \nabla \times \omega\right)$ along all boundaries.

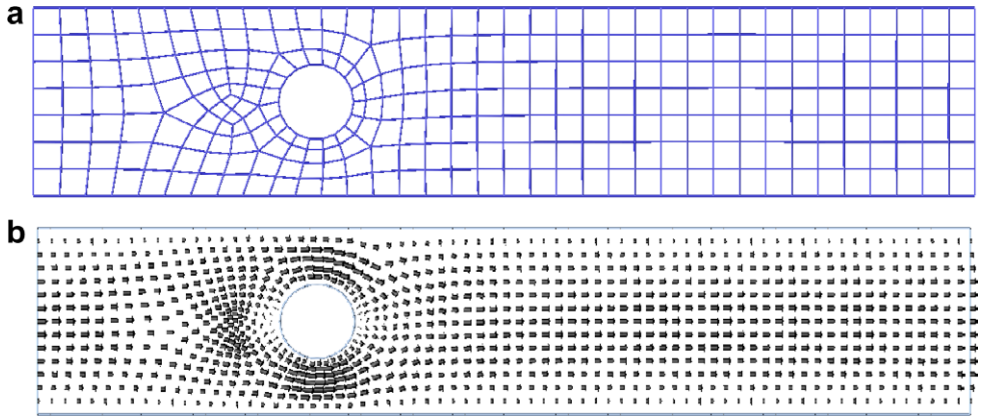


Fig. 3. (a) A cross-section of the finite element mesh for the second test problem. The domain is $5 \times 1 \times 1$ and the cylindrical obstruction has a diameter of 0.4. (b) Velocity vectors for the central cross-section using a triquadratic basis and $\alpha = 10$.

finite element basis and $\alpha = 10$, the global mass loss was 2%. Using the same basis and value for $\alpha$, the traditional vorticity formulation had a mass loss of 45% and the velocity-flux formulation had a mass loss of 29%. The velocity vectors along the central cross-section are shown in Fig. 3b for the solution using the new formulation. The unstructured mesh and curved boundary did have an impact on the performance of the linear solver for this problem. Solving on eight distributed processors resulted in an average convergence factor of 0.83 for the iterations of the AMG preconditioned CG solver. The average convergence factor can be decreased slightly to 0.81 by solving the problem on only six distributed processors, but this required 1.2 GB of memory per processor, and we are normally limited to 1 GB per processor. The reason for the slight improvement in convergence factor is that the BoomerAMG preconditioner uses a hybrid Gauss–Seidel/Jacobi smoother. Basically, Gauss–Seidel is used on each independent processor and Jacobi is used for the boundary between processors. Also, the special coarse-grid selection algorithms used along processor boundaries can impact the convergence factors [16].

The final test problem starts as a cylindrical tube followed by a series of downstream bifurcations. An all hexahedral mesh of the domain, which consists of 420 elements, is shown in Fig. 4a. Incompressible flow at a Reynolds number of 100 was modeled in this domain using boundary conditions identical to those of the previous test problems except that the normal velocity along the inlet was set to be a paraboloid with a maximum value of 1.0. This is the steady flow profile in a cylindrical tube, so the profile remains unchanged until the flow approaches the bifurcation. Fig. 4b shows the velocity vectors using the new formulation and a weighting of 10.0 on the $\nabla \cdot \mathbf{v}$ term in the functional. Considering the coarse mesh used and irregular geometry, it is somewhat surprising that the same weight as previous test problems was required and not a larger weight. Quantitatively, the global mass loss in the solution shown in Fig. 4b, between the inlet and outlet is 3%. This is much better that the traditional vorticity functional (Eq. (6)), which had a global mass loss of 81%, and the velocity-flux function (Eq. (12)), which had a global mass loss of 42%, even though both had a weighting of 10 on the $\nabla \cdot \mathbf{v}$ term. For the previous test problems, the convergence factor of the AMG preconditioned CG solver varied little between the three or four Gauss–Newton steps required to converge to a solution. For this problem,

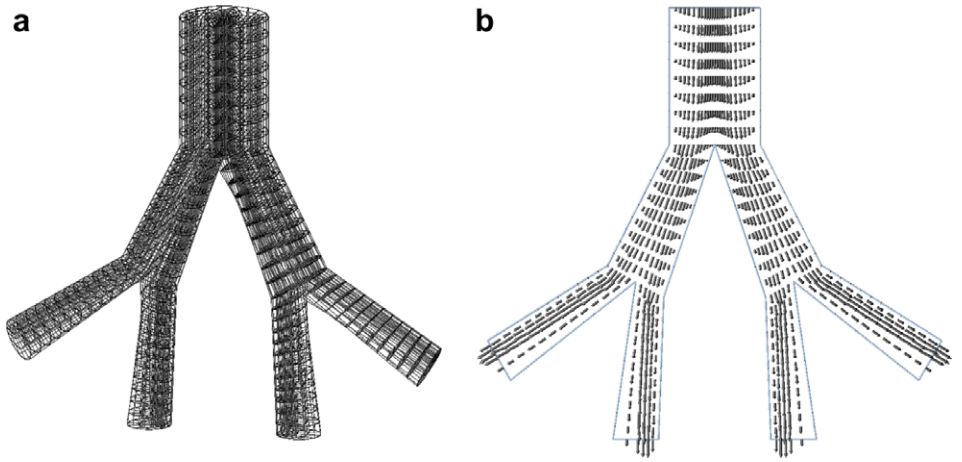


Fig. 4. (a) The finite element mesh for the third test problem composed of 420 hexahedral elements. (b) Velocity vectors for the central cross-section using a triquadratic basis and $\alpha = 10$.

however, the convergence factor varied between 0.7 and 0.9 for the four iterations. Further, the first iteration was the second fastest at 0.73, and the final iteration was the fastest at 0.7. The second and third steps were slower at 0.84 and 0.9, respectively. This is uncharacteristic since we normally observe only small changes in the average convergence factor, and, even if we do observe changes, they are typically a decrease in the converge factor and an acceleration of the solution process. We do not have an explanation for this unusual behavior.

### 4.1. Block solver

The differentially diagonally dominate structure of the operator shown in Eq. (18) suggests the possibility of dividing the problem into three parts that can be solved separately. Given an initial guess for the unknowns ($\mathbf{v}$, $\omega$, and $\mathbf{r}$) and appropriate boundary conditions (e.g. Table 3), the lower left block may be solved by minimizing the functional

$$G_1(\mathbf{r}) := \|\nabla \times \mathbf{r}\|^2_{0,\Omega} + \|\nabla \cdot \mathbf{r} + \sqrt{Re}(\mathbf{r} \cdot \mathbf{r}) + Re(\mathbf{v} \cdot \mathbf{r})\|^2_{0,\Omega}. \tag{21}$$

The second step is to solve the middle block (using the new value for $\mathbf{r}$) by minimizing

$$G_2(\omega) := \|-\mathbf{r} + \frac{1}{\sqrt{Re}}\nabla \times \omega + \sqrt{Re}(\mathbf{v} \times \omega)\|^2_{0,\Omega} + \|\nabla \cdot \omega\|^2_{0,\Omega}, \tag{22}$$

and the final step is solving for the velocity in the upper block (using the new values for $\mathbf{r}$ and $\omega$) by minimizing

$$G_3(\mathbf{v}) := \|\omega + \nabla \times \mathbf{v}\|^2_{0,\Omega} + \|\nabla \cdot \mathbf{v}\|^2_{0,\Omega}. \tag{23}$$

For the $Re = 0$ case, the functionals, $G_1$–$G_3$, are uncoupled in a way that the problem may be solved by minimizing each functional just once in the order described. If $Re \neq 0$, an iterative solution strategy is required, and the convergence rate to the overall solution determines whether or not this is a practical method.

We tested the block solver approach on the first test problem (see Table 1) with an additional boundary condition of $\mathbf{n} \cdot \mathbf{r} = \frac{1}{\sqrt{Re}}(\mathbf{n} \cdot \nabla \times \omega)$ on all boundaries. Convergence to the approximate solution was defined as each functional changing by less than $10^{-3}$. However, with $Re = 100$, the block method failed to converge to an approximate solution. Convergence, instead, was achieved through the use of a Reynolds number continuation strategy. First, the problem was solved with $Re = 1$, then $Re = 10$, and finally $Re = 100$. A total of three iterations through the functionals, $G_1$–$G_3$, were used for the $Re = 1$ and $Re = 10$ cases, and four iterations were required for the $Re = 100$ case. The overall efficiency was helped by the fact that the convergence factor on each subproblem was 0.3–0.5, and the memory requirements were 1/3 that of solving the full problem. The global mass conservation, however, was not very good because 30% of the mass was lost with a weighting of 10 on the $\nabla \cdot \mathbf{v}$ term. When the boundary condition on $\mathbf{r}$ was changed to $\mathbf{n} \cdot \mathbf{r} = \frac{1}{\sqrt{Re}}(\mathbf{n} \cdot \nabla \cdot \nabla \mathbf{v})$

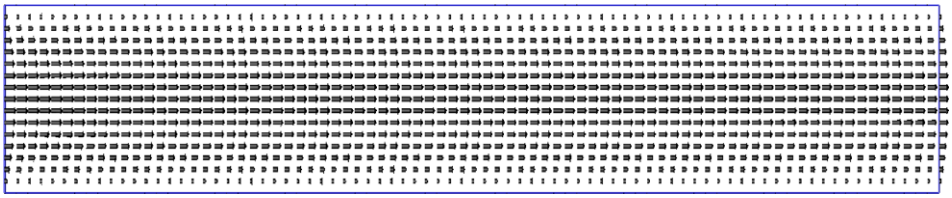

Fig. 5. Velocity vectors for the central cross-section using a triquadratic basis and the block solver approach on the first test problem.

on all boundaries, the mass loss decreased to only 5%, as shown in Fig. 5. It is not clear why the mass loss is higher when using the block solver approach, but the solution when minimizing three separate functionals is different than minimizing a single combined functional. However, the two methods must give the same solution as $h \to 0$. In summary, the poor robustness of this solution method prevents it from competing with the full functional method at this time. On the other hand, if the convergence and robustness could be improved by some other means (e.g. grid continuation), it could become a very efficient method. The block solver could also be used as a preconditioner for the full functional, but we have not explored this possibility.

## 5. Discussion

The new formulation results in an improvement in both convergence rate and mass conservation accuracy relative to the original vorticity formulation. Of course, the new formulation also has nine unknowns per node compared to the vorticity formulation, which has seven unknowns per node. This difference translates into approximately 65% more nonzero entries in the matrix, but the new formulation is still more computationally efficient because it requires only half the number of AMG preconditioned CG iterations and gives better mass conservation with the same weight on the $\nabla \cdot \mathbf{v}$ term. Compared to the velocity-flux formulation, the new method gives similar mass conservation and similar convergence rates for the linear problem, but it has significantly fewer unknowns per node (9 vs. 13), resulting in less than half as many non-zeros in the matrix.

The improved convergence rate of the new formulation is a result of the operator being differentially diagonally dominate and consisting of basically three independent Laplacian operators. However, the reason for the improved mass conservation is less obvious. One important clue comes from the observation that with any of the formulations shown here, excellent mass conservation ($\ll$1% error) can be achieved by enforcing inflow and outflow boundary conditions on the pressure instead of on the normal velocity. Without a boundary condition on the pressure, the coupling between the pressure and the velocity is insufficient on coarse grids and the pressure tends to be relatively constant. The importance of strong coupling between the pressure and velocity in LSFEM formulations has also been stressed by others [23]. With the new formulation, the variable $\mathbf{r}$ represents the 'pressure' gradient, and we are able to achieve stronger coupling between the pressure and velocity through this variable by essentially adding a pressure Poisson equation and consistent Neumann boundary conditions. The pressure Poisson equation is essentially a restatement of conservation of mass in terms of pressure, and, as a result, the new formulation yields improved mass conservation relative to existing formulations for a given scaling on the $\nabla \cdot \mathbf{v}$ term.

## Acknowledgments

This work was sponsored by the Department of Energy under Grant Nos. DE-FC02-01ER25479 and DE-FG02-03ER25574, Lawrence Livermore National Laboratory under Contract No. B541045, Sandia National Laboratory under Contract No. 15268, the National Science Foundation under Grant Nos. DMS-0410318, and the Flight Attendant Medical Research Institute.

## References

[1] P. Bochev, Analysis of least-squares finite element methods for the Navier–Stokes equations, SIAM J. Numer. Anal. 34 (5) (1997) 1817–1844.

[2] P. Bochev, Z. Cai, T. Manteuffel, S. McCormick, Analysis of velocity-flux first-order system least-squares principles for the Navier–Stokes equations: part 1, SIAM J. Numer. Anal. 35 (1998) 990–1009.
[3] P. Bochev, Z. Cai, T. Manteuffel, S. McCormick, Analysis of velocity-flux least-squares principles for the Navier–Stokes equations: Part i, SIAM J. Numer. Anal. 35 (3) (1998) 990–1009.
[4] P. Bochev, M. Gunzburger, Accuracy of least-squares methods for the Navier–Stokes equations, Comput. Fluids 22 (4/5) (1993) 549–563.
[5] P. Bochev, M. Gunzburger, Finite element methods of least-squares type, SIAM Rev. 40 (4) (1998) 789–837.
[6] P. Bochev, T. Manteuffel, S. McCormick, Analysis of velocity-flux least-squares principles for the Navier–Stokes equations: Part ii, SIAM J. Numer. Anal. 36 (4) (1999) 1125–1144.
[7] A. Brandt, S. McCormick, J. Ruge, Algebraic Multigrid (AMG) for Sparse Matrix Equations, Cambridge University Press, Cambridge, UK, 1984, pp. 257–284.
[8] S. Brenner, L. Scott, The Mathematical Theory of Finite Element Methods, second ed., Springer, New York, 2000.
[9] Z. Cai, T. Manteuffel, S. McCormick, First-order system least squares for the Stokes equations, with application to linear elasticity, SIAM J. Numer. Anal. 34 (5) (1997) 1727–1741.
[10] Z. Cai, T.A. Mantueffel, S.F. McCormick, First-order system least squares for second-order partial differential equations: Part ii, SIAM J. Numer. Anal. 34 (2) (1997) 425–545.
[11] C. Chang, J. Nelson, Least-squares finite element method for the Stokes problem with zero residual of mass conservation, SIAM J. Numer. Anal. 34 (2) (1997) 480–489.
[12] A. Codd, T. Manteuffel, S. McCormick, Multilevel first-order system least squares for nonlinear partial differential equations, SIAM J. Numer. Anal. 41 (6) (2003) 2197–2209.
[13] A. Codd, T. Manteuffel, S. McCormick, J.W. Ruge, Multilevel first-order system least squares for elliptic grid generation, SIAM J. Numer. Anal. 41 (6) (2003) 2210–2232.
[14] J. Deang, M. Gunzburger, Issues related to least-squares finite element methods for the stokes equations, SIAM J. Sci. Comp. 20 (3) (1998) 878–906.
[15] R. Falgout, U. Yang, hypre: A library of high performance preconditioners, Lecture Notes Comput. Sci. 2331 (2002) 632–641.
[16] R. Falgout, U. Yang, Pursuing scalability for hypre's conceptual interfaces, ACM Trans. Math. Software 31 (3) (2005) 326–350.
[17] P. Gresho, R. Sani, Incompressible Flow and the Finite Element Method, Advection–Diffusion and Isothermal Laminar Flow, John Wiley and Sons, New York, 1998.
[18] J. Heys, E. Lee, T. Manteuffel, S. McCormick, On mass-conserving least-squares methods, SIAM J. Sci. Comp. 28 (2006) 1675–1693.
[19] J. Heys, T. Manteuffel, S. McCormick, L. Olson, Algebraic multigrid for higher-order finite elements, J. Comput. Phys. 204 (2) (2005) 520–532.
[20] J. Heys, T. Manteuffel, S. McCormick, J. Ruge, First-order systems least squares (FOSLS) for coupled fluid-elastic problems, J. Comput. Phys. 195 (2) (2004) 560–575.
[21] B. Jiang, The Least-Squares Finite Element Method, Springer, Berlin, 1998.
[22] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1) (1998) 359–392.
[23] J. Pontaza, A least-squares finite element formulation for unsteady incompressible flows with improved velocity-pressure coupling, J. Comput. Phys. 217 (2) (2006) 563–588.
[24] J. Pontaza, J. Reddy, Space-time coupled spectral/hp least-squares finite element formulations for the incompressible Navier–Stokes equations, J. Comput. Phys. 197 (2004) 418–459.
[25] M. Proot, M. Gerritsma, Mass- and momentum conservation of the least-squares spectral element method for the stokes problem, J. Sci. Comput. 27 (1–3) (2006) 389–401.