Comput. Methods Appl. Mech. Engrg. 247–248 (2012) 216–225

Contents lists available at SciVerse ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Assessment of a vorticity based solver for the Navier–Stokes equations

Michele Benzi [a], Maxim A. Olshanskii [b,c,*,1], Leo G. Rebholz [d,2], Zhen Wang [e,3]

[a] Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA
[b] Department of Mathematics, University of Houston, Houston, TX 77204, USA
[c] Department of Mechanics and Mathematics, Moscow State University, Moscow 119899, Russia
[d] Department of Mathematical Sciences, Clemson University, Clemson, SC 29634, USA
[e] Scientific Computing Group, National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

ARTICLE INFO

Article history:
Received 20 March 2012
Accepted 30 July 2012
Available online 24 August 2012

Keywords:
Navier–Stokes equations
Vorticity
Helical density
Finite element method
Preconditioning
Augmented Lagrangian method

ABSTRACT

We investigate numerically a recently proposed vorticity based formulation of the incompressible Navier–Stokes equations. The formulation couples a velocity–pressure system with a vorticity–helicity system, and is intended to provide a numerical scheme with enhanced accuracy and superior conservation properties. For a few benchmark problems, we study the performance of a finite element method for this formulation and compare it with the commonly used velocity–pressure based finite element method. It is shown that both steady and unsteady discrete problems in the new formulation admit simple decoupling strategies followed by the application of iterative solves to auxiliary subproblems. Further, we compare several iterative strategies to solve the discrete problems and study the interplay between the choice of stabilization parameters in the finite element method and the efficiency of linear algebra solvers.

1. Introduction

We consider the system of the Navier–Stokes (NS) equation describing incompressible fluid dynamics in the velocity–vorticity–helicity (VVH) form, on a bounded domain $\Omega \subset \mathbb{R}^3$ with sufficiently smooth boundary and for time interval $t \in (0, T]$,

$$\mathbf{w}_t - \nu\Delta\mathbf{w} + 2D(\mathbf{w})\mathbf{u} - \nabla\eta = \nabla \times \mathbf{f}, \tag{1.1}$$

$$\mathbf{u}_t - \nu\Delta\mathbf{u} + \mathbf{w} \times \mathbf{u} + \nabla P = \mathbf{f}, \tag{1.2}$$

$$\nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{w} = 0, \tag{1.3}$$

where $\mathbf{u}$ denotes velocity, $\mathbf{w}$ vorticity, η and P denote the helical density and Bernoulli pressure, $D(\mathbf{w}) := \frac{1}{2}(\nabla\mathbf{w} + [\nabla\mathbf{w}]^T)$ is the symmetric part of the vorticity gradient, and ν is the kinematic viscosity. The system is equipped with the initial conditions

$$\mathbf{u} = \mathbf{u}_0, \quad \mathbf{w} = \nabla \times \mathbf{u}_0, \quad \text{for } t = 0, \tag{1.4}$$

and with the boundary conditions

$$\mathbf{u}|_{\partial\Omega} = \boldsymbol{\phi}, \quad \mathbf{w}|_{\partial\Omega} = \boldsymbol{\psi}, \tag{1.5}$$

where the natural choice of $\boldsymbol{\psi}$ is $\boldsymbol{\psi} = \nabla \times \mathbf{u}$ or $\boldsymbol{\psi} = 0$ for the far-field outflow boundaries. This formulation was derived in [31], and has since been studied numerically in the case of equilibrium NS equations [23], and for the Boussinesq system [27]. All three of these studies have shown promising results.

The VVH system is particularly interesting from the physical point of view. It solves directly for the vorticity, and it has been argued that methods that do so are more physically accurate, particularly near boundaries [8]. Using vorticity equations for fluid dynamics solvers has a long history and has been a subject of intensive studies, see, e.g., [16,18,24,25,34,35] for a sample of results. Furthermore, it was pointed out recently in [30], see also the discussion in [14], that the discrete vorticity $\mathbf{w}_h$ from the finite element vorticity equation is a more natural quantity than $\nabla \times \mathbf{u}_h$ for the discrete balance laws for vorticity, enstrophy and helicity when the forcing terms are conservative. Therefore, it may be beneficial to use this discrete vorticity $\mathbf{w}_h$ in the momentum equations for the velocity through the Lamb vector $\mathbf{w}_h \times \mathbf{u}_h$. Additionally, using the dynamic equation (1.2) for linking velocity and vorticity instead of the vector Poisson equation $\Delta\mathbf{u} = -\nabla \times \mathbf{w}$ immediately provides the discrete system with the 'correct' energy balance (or a desired alteration of it if a subgrid/stabilization model is used). This was exploited, in particular, in [23], where the first error analysis was done for vorticity based finite element formulations. VVH is also the first NS formulation to solve directly for the helical density (which is related to the helicity through $H = \int_\Omega \eta \, d\mathbf{x}$), a quantity known to be of fundamental physical importance in fluid

* Corresponding author at: Department of Mathematics, University of Houston, Houston, TX 77204, USA.
E-mail addresses: benzi@mathcs.emory.edu (M. Benzi), molshan@math.uh.edu (M.A. Olshanskii), rebholz@clemson.edu (L.G. Rebholz), wangz@ornl.gov (Z. Wang).

[1] Partially supported by the Russian Foundation for Basic Research through Grants 11-01-00767, 11-01-00971 and 12-01-00283.
[2] Partially supported by National Science Foundation Grant DMS 1112598.
[3] Partially supported by the Laney Graduate School of Arts and Science at Emory University and by the Mathematical, Information, and Computational Sciences Division, Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

0045-7825/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cma.2012.07.016

flow [26,6,15]. This formulation also explicitly enforces the vorticity to be incompressible by Eq. (1.3), with helical density η in (1.1) acting as a Lagrange multiplier corresponding to this constraint. Since $\nabla\cdot\nabla\times=0$, that the vorticity is solenoidal is important both for physical relevance and mathematical consistency. Although it is possible to couple this constraint to the usual vorticity equation by adding an artificial Lagrange multiplier, VVH enforces this constraint naturally. Thus, we deem the formulation worth of further study in the direction of cost-efficiency comparison to the more common velocity–pressure formulation and the development of fast algebraic solvers. This is the objective of the present paper.

For the purpose of benchmarking, we choose one problem with a known analytical solution from [13], and the unsteady flow over a 3D forward–backward facing step with $Re=200$, see [20]. Both problems feature essentially 3D solutions and are relevant for testing the accuracy, the stability, and the ability of an incompressible CFD solver to capture important flow properties on relatively coarse meshes. Another perceptible difficulty in computing with the fully coupled VVH system is solving the large sparse linear systems that arise in the discretizations. In this paper, we use explicit (for unsteady problem) or implicit (for steady solutions) decoupling strategies to reduce the computations and to solve more standard linear algebraic systems of lower dimension. The algebraic approach we consider herein is block preconditioned GMRES [33], where the block preconditioning is based on an augmented-Lagrangian (AL) approach developed earlier in [2,4,3] for velocity–pressure saddle point systems. Here we extend and study this approach for the dual-coupled saddle points systems resulting from a finite element discretization of (1.1)–(1.5).

This paper is arranged as follows. In Section 2, we present the finite element discretization for the VVH system (1.1)–(1.5). Results of numerical experiments for the Ethier–Steinman and the 3D step problems are discussed in Section 3. Preconditioning and the algebraic solvers are studied in Section 4. Finally, in Section 5, we draw conclusions and discuss future directions.

2. Finite element solver

We present in this section the finite element discretization for the system (1.1)–(1.5), along with a brief discussion. We assume homogeneous boundary conditions for the velocity to simplify the weak formulation and the subsequent analysis. Both of our numerical experiments are for inhomogeneous boundary conditions for the velocity. The Galerkin finite element method for the steady Eqs. (1.1)–(1.5) is as follows.

Let $(\mathbf{X}_h,Q_h)\subset(\mathbf{H}^1(\Omega),L^2(\Omega))$ be conforming finite element spaces on a regular mesh τ_h on a polyhedral domain Ω, satisfying the LBB condition, inverse inequality and the standard approximation properties, see, e.g., [17]:

$$\inf_{\mathbf{v}_h\in\mathbf{X}_h}(\|\boldsymbol{\phi}-\mathbf{v}_h\|_0+h\|\boldsymbol{\phi}-\mathbf{v}_h\|_1)\leqslant Ch^{\ell+1}|\boldsymbol{\phi}|_{\ell+1},$$
$$\inf_{q_h\in Q_h}\|r-q_h\|_0\leqslant Ch^{\ell}|r|_{\ell},\quad\text{for }\ell=0,\dots,k,$$

with some integer $k\geqslant 1$. Define the subspaces $\mathbf{X}_{h0}:=\mathbf{X}_h\cap\mathbf{H}_0^1(\Omega)$.

The finite element formulation reads: given forcing $\mathbf{f}\in\mathbf{L}^2(\Omega)$ and kinematic viscosity $\nu>0$, find $(\mathbf{u}_h,\mathbf{w}_h,P_h,\eta_h)\in\mathbf{X}_{h0}\times\mathbf{X}_h\times Q_h\times Q_h$ for any time $t\in[0,T]$ satisfying $\forall(\mathbf{v}_h,\boldsymbol{\chi}_h,q_h,r_h)\in\mathbf{X}_{h0}\times\mathbf{X}_{h0}\times Q_h\times Q_h$,

$$\begin{cases}((\mathbf{u}_h)_t,\mathbf{v}_h)+(\mathbf{w}_h\times\mathbf{u}_h,\mathbf{v}_h)-(P_h,\nabla\cdot\mathbf{v}_h)+\nu(\nabla\mathbf{u}_h,\nabla\mathbf{v}_h)=(\mathbf{f},\mathbf{v}_h),\\(\nabla\cdot\mathbf{u}_h,q_h)=0,\\((\mathbf{w}_h)_t,\mathbf{v}_h)+2(D(\mathbf{w}_h)\mathbf{u}_h,\boldsymbol{\chi}_h)+(\eta_h,\nabla\cdot\boldsymbol{\chi}_h)+\nu(\nabla\mathbf{w}_h,\nabla\boldsymbol{\chi}_h)=(\nabla\times\mathbf{f},\boldsymbol{\chi}_h),\\(\nabla\cdot\mathbf{w}_h,r_h)=0,\\\mathbf{w}_h-I_h(\nabla\times\mathbf{u}_h)=0\quad\text{on }\partial\Omega.\end{cases}\tag{2.6}$$

Here I_h denotes a generic interpolant such that $\int_{\partial\Omega}I_h(\nabla\times\mathbf{u}_h)\cdot\mathbf{n}=0$, where $\mathbf{n}$ is an outward normal vector to $\partial\Omega$, e.g., a Clement-type interpolant I_h^C based on local averaging.

This method was analyzed in [23] for the case of equilibrium solution, and was found to be stable and optimally convergent. More precisely, the following result is valid: let $(\mathbf{u},p)$ be the solution to the stationary incompressible Navier–Stokes equations in a bounded domain $\Omega\subset\mathbb{R}^3$ with a sufficiently regular boundary and homogeneous Dirichlet boundary conditions for $\mathbf{u}$. Assume $\mathbf{f}\in\mathbf{L}^2(\Omega),\mathbf{u}\in\mathbf{H}_0^1(\Omega)\cap\mathbf{H}^{k+1}(\Omega)$. If $(\mathbf{u}_h,P_h),(\mathbf{w}_h,\eta_h)$ are the solutions to (2.6), with $\mathbf{w}_h=I_h^C(\nabla\times\mathbf{u})$ on $\partial\Omega$ and with a small data assumption on $\|\mathbf{f}\|$, then the *a priori error estimate*

$$\|\nabla(\mathbf{u}-\mathbf{u}_h)\|^2+\|\mathbf{w}-\mathbf{w}_h\|^2\leqslant C\Big(h^{2k}+\|\Big(\nabla\times\mathbf{u}-I_h^C(\nabla\times\mathbf{u})\Big)\otimes\mathbf{n}\|^2_{-\frac{1}{2},\partial\Omega}+\|\Big(\nabla\times\mathbf{u}-I_h^C(\nabla\times\mathbf{u})\Big)\cdot\mathbf{n}\|^2_{-\frac{1}{2},\partial\Omega}\Big)$$

holds with $\mathbf{w}=\nabla\times\mathbf{u}$. Moreover, if additional regularity of the Navier–Stokes velocity is assumed, $\mathbf{u}\in\mathbf{H}_0^1(\Omega)\cap\mathbf{H}^{k+2}(\Omega)$, $P\in\mathbf{H}^k(\Omega)$ then it holds

$$\|\nabla(\mathbf{u}-\mathbf{u}_h)\|^2+\|\nabla(\mathbf{w}-\mathbf{w}_h)\|^2+\|P-P_h\|+\|\eta-\eta_h\|\leqslant Ch^{2k}.\tag{2.7}$$

The above convergence result assumes $\mathbf{w}_h=I_h^C(\nabla\times\mathbf{u})$ for the vorticity boundary condition instead of the more practical $\mathbf{w}_h=I_h^C(\nabla\times\mathbf{u}_h)$. For $\mathbf{w}_h=I_h^C(\nabla\times\mathbf{u}_h)$, numerical experiments from [23], using P2-P1 finite elements, show the 1 and 0.5 convergence order reduction for the vorticity in L_2 and H^1 norms, respectively, and less than 0.5 convergence order reduction for the helical density in L_2 norm, compared to those predicted by (2.7). On the other hand, velocity errors remain of optimal order.

2.1. Grad-div stabilization

In numerical experiments we use the LBB stable P2-P1 Taylor–Hood finite element on a regular mesh of tetrahedrons satisfying a uniform small angle condition. In practice, using an element pair that does not provide pointwise enforcement of the solenoidal constraints (such as Taylor–Hood) may lead to poor scaling of the velocity error with respect to the viscosity coefficient and the norm of the pressure gradient [22,32]. This effect is especially pronounced for the case of the rotation form of the momentum equation, since the Bernoulli pressure may share sharp internal or boundary layers with the velocity. One way to ameliorate much of this bad scaling of the velocity error with respect to the viscosity consists in introducing a simple grad-div stabilization [29,32]: one adds the least-squares type term

$$\gamma_1(\nabla\cdot\mathbf{u}_h,\nabla\cdot\mathbf{v}_h)$$

to the finite element momentum equation, with a parameter $\gamma_1=O(1)$. Since for 3D flows the helicity gradient can likewise affect the error in the vorticity for small viscosities, we add a similar term

$$\gamma_2(\nabla\cdot\mathbf{w}_h,\nabla\cdot\boldsymbol{\chi}_h)$$

to the finite element vorticity equation with $\gamma_2=O(1)$. It was shown in [29] (for the velocity–pressure form) that the accuracy of the finite element solution is not very sensitive to the variation in γ_1 up to $\gamma_1=O(1)$. The 'optimal' value was found for several flows to be around 0.2. Thus, we take $\gamma_1=0.2$ further in all numerical experiments. Here we also experiment with varying γ_2. The dependence of the error on γ_2 for the steady Ethier–Steinman problem (described in the next section) and the number of iterations in the augmented Lagrangian preconditioned Krylov subspace method (see details in Section 4) are shown in Fig. 1. Due to this, in the numerical experiments herein we take the value $\gamma_2=0.5$ as close to optimal.

Although the convergence results above were proved with $\gamma_1 = \gamma_2 = 0$, they can be easily extended to the case of $\gamma_1, \gamma_2 > 0$, with the constant C possibly dependent on the γ's. We shall see in Section 4 that introducing the stabilization is also favorable for building iterative solvers.

2.2. Numerical time integration

The Navier–Stokes equations written in the form (1.1)–(1.3) call for the natural splitting algorithm for time integration. Indeed, if the velocity $\mathbf{u}$ is frozen, then the vorticity equation (1.1) becomes linear; conversely, if the vorticity $\mathbf{w}$ is frozen, then the velocity equation (1.2) becomes linear. We exploit this property in the following second-order time integration splitting method (for the sake of notation we suppress the spacial discretization indices here). Denoting $\phi^n := \phi(t_n), t_n = t_0 + n(\Delta t), \phi^{n+1/2} := \frac{1}{2}(\phi^n + \phi^{n+1})$, we compute for $n = 0, 1, 2, \ldots$

Step 1:

$$\begin{cases} \frac{1}{\Delta t}(\mathbf{u}^{n+1} - \mathbf{u}^n) - \nu\Delta\mathbf{u}^{n+\frac{1}{2}} + \nabla P^{n+1} + \left(\frac{3}{2}\mathbf{w}^n - \frac{1}{2}\mathbf{w}^{n-1}\right) \times \mathbf{u}^{n+\frac{1}{2}} - \mathbf{f}^{n+\frac{1}{2}} = 0, \\ \nabla \cdot \mathbf{u}^{n+1} = 0, \\ \mathbf{u}^{n+1}|_{\partial\Omega} - \boldsymbol{\phi}^{n+1} = 0. \end{cases} \tag{2.8}$$

Step 2:

$$\begin{cases} \frac{1}{\Delta t}(\mathbf{w}^{n+1} - \mathbf{w}^n) - \nu\Delta\mathbf{w}^{n+\frac{1}{2}} - \nabla\eta^{n+1} + 2\mathbf{D}(\mathbf{w}^{n+\frac{1}{2}})\mathbf{u}^{n+\frac{1}{2}} - \nabla \times \mathbf{f}^{n+\frac{1}{2}} = 0, \\ \nabla \cdot \mathbf{w}^{n+1} = 0, \\ \mathbf{w}^{n+1}|_{\partial\Omega} - I_h(\nabla \times \mathbf{u}^{n+1})|_{\partial\Omega} = 0. \end{cases} \tag{2.9}$$

At every time step, two linear algebraic problems of saddle point type must be solved. These problems have the same structure as the discrete Oseen system resulting from the semi-explicit scheme for the Navier–Stokes equations in the velocity–pressure convection form (see, e.g., [21]): For $n = 0, 1, 2, \ldots$ compute

$$\begin{cases} \frac{1}{\Delta t}(\mathbf{u}^{n+1} - \mathbf{u}^n) - \nu\Delta\mathbf{u}^{n+\frac{1}{2}} + \nabla p^{n+1} + \left(\frac{3}{2}\mathbf{u}^n - \frac{1}{2}\mathbf{u}^{n-1}\right) \cdot \nabla\mathbf{u}^{n+\frac{1}{2}} - \mathbf{f}^{n+\frac{1}{2}} = 0, \\ \nabla \cdot \mathbf{u}^{n+1} = 0, \\ \mathbf{u}^{n+1}|_{\partial\Omega} - \boldsymbol{\phi}^{n+1} = 0. \end{cases} \tag{2.10}$$

We shall use the scheme (2.10) for the purpose of comparison of the vorticity–velocity solutions (2.8) and (2.9) to the more common velocity solutions. Obviously, one time step of (2.8) and (2.9) is nearly two times as expensive as one time step of (2.10). Thus, for a more fair comparison of the schemes we choose the time step for (2.10) half the time step for (2.8) and (2.9). An important observation is that by using splitting schemes to integrate (1.1) and (1.2), one largely avoids the increase of computer memory consumption due to having double the number of unknowns compared to the velocity–pressure formulation. Indeed, temporary data such as auxiliary vectors in Krylov subspace iterative methods or matrix factorizations for preconditioners account for a major part of total storage inputs. Thus, it is important to reduce the dimension of the auxiliary linear algebra problems to be solved in (2.8) and (2.9) to the same size as in (2.10).

It is possible to develop more explicit splitting (projection) schemes for (1.1) and (1.2) along the lines of Chorin–Temam–Yanenko type schemes for the velocity–pressure convection form of the Navier–Stokes equations. This would come with the well-known price of accepting numerical boundary layers and time step stability restrictions. We will explore such schemes elsewhere.

3. Numerical experiments

We now describe two numerical examples that illustrate the effectiveness of the proposed method. These tests obtain VVH approximations to the solution using the standard finite element approximation to (2.8) and (2.9), and P2-P1 Taylor–Hood elements for both the velocity–pressure and vorticity–helicity systems. For the vorticity boundary condition, the normal component can be determined from the Dirichlet velocity condition, and the tangential components come from a nodal averaging of $\nabla \times \mathbf{u}_h$ at the boundary.

The numerical tests in this section were performed in MATLAB on a 2 × 2.66 GHz Quad-Core Intel Xeon Mac 10.6.8 workstation with 32 GB 1066 MHz DDR3 memory.

3.1. Experiment 1: The Ethier–Steinman problem

The first numerical experiment we consider is to compute approximations to the Ethier–Steinman exact Navier–Stokes solution from [13] on $[-1, 1]^3$. For chosen parameters a, d and viscosity ν, this exact NSE solution is given by

$$\begin{cases} u_1 = -a(e^{ax}\sin(ay + dz) + e^{az}\cos(ax + dy))e^{-\nu d^2 t}, \\ u_2 = -a(e^{ay}\sin(az + dx) + e^{ax}\cos(ay + dz))e^{-\nu d^2 t}, \\ u_3 = -a(e^{az}\sin(ax + dy) + e^{ay}\cos(az + dx))e^{-\nu d^2 t}, \\ p = -\frac{a^2}{2}\left(e^{2ax} + e^{2ay} + e^{2az} + 2\sin(ax + dy)\cos(az + dx)e^{a(y+z)}\right. \\ \quad +2\sin(ay + dz)\cos(ax + dy)e^{a(z+x)} \\ \quad \left.+2\sin(az + dx)\cos(ay + dz)e^{a(x+y)}\right)e^{-\nu d^2 t}. \end{cases} \tag{3.11}$$

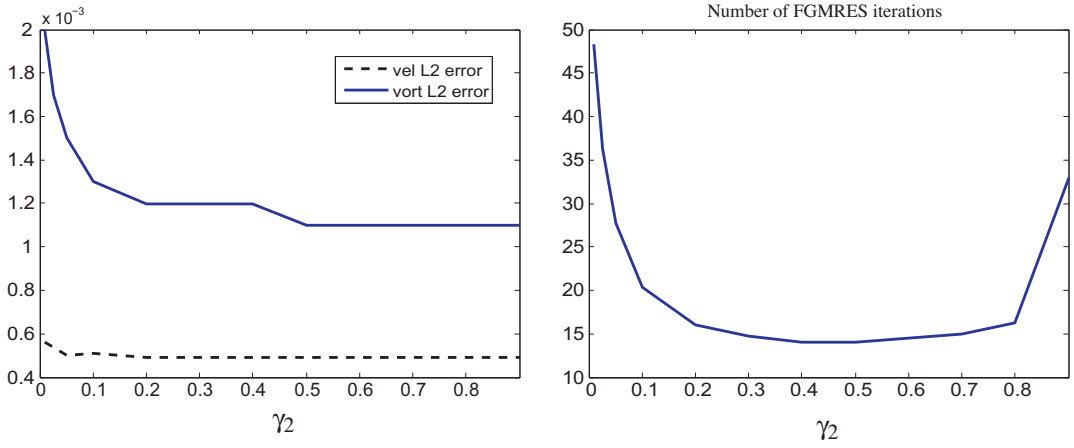

Fig. 1. Dependence of the L_2 velocity and vorticity error and the number of iterations for AL -preconditioned FGMRES on the variation of γ_2.

The first part of this test is to demonstrate the effectiveness of a linear algebra solver that works very well on the vorticity-helical density systems; effective methods for solving the velocity–pressure system in rotational form are already known [1,28], and so we do not discuss that linear solve further except to note that the solver discussed below for vorticity–helicity worked very well on the velocity–pressure system as well. The chosen solver was GMRES (50) with a block lower triangular preconditioner [10]. Diagonal blocks, which approximate the pressure (helical density) Schur complement matrices and velocity (vorticity) submatrices were built using an inexact Cahouet–Chabard preconditioner [5] and an incomplete LU factorization (drop tolerance 10^{-2}), respectively. To approximate the solution of the Poisson problem in the Cahouet–Chabard preconditioner, we use incomplete Cholesky factorization (again with drop tolerance 10^{-2}). Other drop tolerance values were tested, but we found 10^{-2} to be essentially optimal in terms of total solution times. We also tested an augmented Lagrangian preconditioner (as in [4]), and got nearly as good results.

We computed approximate solutions for several uniform tetrahedralizations of the unit cube (details of which are given in Table 1, using $\nu = 0.01, \Delta t = 0.01, a = 0.75, d = 0.5$ and $T = 0.05$. We used grad-div stabilization in both equations, taking $\gamma_1 = 0.2, \gamma_2 = 0.5$. Timings and iteration counts are also shown in Table 1. A slight growth in the number of total GMRES iterations with an increase in degrees of freedom is observed, but overall the iterations and timings are observed to be quite good.

In this numerical example, we also compare the accuracy of the splitting scheme (2.8), (2.9) to a commonly used Navier–Stokes discretization: the linear extrapolated Crank–Nicolson (CNLE) scheme (2.10). For the purpose of comparison, we also use the nonlinear Crank–Nicolson scheme (CN) as an ultimately implicit second-order scheme in the primitive variables: For $n = 0, 1, 2, \ldots$ compute

$$\begin{cases} \frac{1}{\Delta t}(\mathbf{u}^{n+1} - \mathbf{u}^n) - \nu\Delta\mathbf{u}^{n+\frac{1}{2}} + \nabla p^{n+1} + \mathbf{u}^{n+\frac{1}{2}} \cdot \nabla\mathbf{u}^{n+\frac{1}{2}} - \mathbf{f}^{n+\frac{1}{2}} = 0, \\ \nabla \cdot \mathbf{u}^{n+1} = 0, \\ \mathbf{u}^{n+1}|_{\partial\Omega} - \boldsymbol{\phi}^{n+1} = 0. \end{cases} \tag{3.12}$$

To compute with CNLE and CN we use P2-P1 Taylor–Hood finite elements, with the same mesh as for (2.8) and (2.9).

We compare solutions to these schemes to that of VVH by computing each of them to $T = 1$ on the $h = 1/8$ uniform mesh, and comparing errors. Since the exact solution is given in (3.11) we use both exact vorticity values and nodal averaging of the curl of the computed finite element velocity as vorticity boundary conditions in (2.9). Note that CN is a nonlinear scheme, and we use Newton's method to resolve it. On average, CN needed three Newton iterations at each timestep. CNLE only needs one linear solve per timestep, while VVH needs two. The computational cost of each of these algorithms is proportional to the number of linear solves they need, and thus CNLE is about twice as fast as VVH for a single timestep, while CN is slower than VVH. Hence for a fair comparison, we use $\Delta t = 0.01$ for VVH and CN, but for CNLE we use $\Delta t = 0.005$. Plots of the velocity and vorticity errors are displayed in Fig. 2, and VVH is clearly more accurate, particularly when an exact vorticity boundary condition is known.

3.2. Experiment 2: Three-dimensional channel flow over a step

Our second test is for three-dimensional, time-dependent channel flow over a forward–backward facing step with $Re = 200$. Fig. 3 displays a diagram of the flow domain with a $40 \times 10 \times 10$ rectangular channel and a $10 \times 1 \times 1$ block step placed 5 units into the channel on the bottom.

This problem is an alteration of experiments of John and Liakos [20], but with a different treatment of inflow and outflow boundary conditions. John and Liakos use a constant inflow profile, which is likely not physical, and also not appropriate if solving directly for the vorticity since this inflow condition will create a blow-up of vorticity at the inflow edges. We use instead a quartic inflow profile, given below by (3.13), and for simplicity also enforce this condition at the outlet. The correct physical behavior for this flow problem, which was resolved by Cousins et al. [7], is that by $T = 10$, an eddy forms behind the step, detaches and moves into the flow, and another eddy forms.

For the velocity boundary conditions, we choose no-slip boundaries for the channel walls and step, and for the inflow and outflow we enforce the Dirichlet condition

$$\mathbf{u} = \begin{pmatrix} 0 \\ x(10-x)y(10-y)/625 \\ 0 \end{pmatrix}. \tag{3.13}$$

For the initial condition on the velocity we use the $Re = 50$ steady solution of this problem. For the vorticity boundary conditions, at the inflow and outflow we enforce the vorticity to be the curl of the inflow and outflow velocity (i.e., the curl of (3.13)), and on the walls and step we enforce the $\mathbf{w}_h \cdot \mathbf{n} = 0$ condition and for the tangential directions we enforce the vorticity at the nodes to be the average of the curl of the velocity. The initial vorticity is taken to be the L^2 projection of the initial velocity solution into the finite element velocity space, and satisfying the above vorticity boundary conditions. A timestep of $\Delta t = 0.04$ is used to advance the VVH algorithm (2.8) and (2.9) to $T = 10$, using P2-P1 Taylor–Hood elements on a tetrahedral mesh that provides 404,289 degrees of freedom both for velocity and for vorticity, and 18,045 degrees of freedom for both Bernoulli pressure and helical density (for a total of 844,668 total degrees of freedom). The mesh is built from a quasi-uniform mesh of tetrahedra, which are built from refinement of rectangular cubes that are refined near the step. The grad-div stabilization parameters were chosen as $\gamma_1 = 0.2$ and $\gamma_2 = 0.5$.

For the linear solves we used preconditioned GMRES for both the velocity–pressure and the vorticity–helicity linear systems.

Table 1
Degrees of freedom for the discretized Ethier–Steinman problem and iteration counts and timings for the vorticity–helicity solve for varying h using GMRES with block-triangular preconditioning. "# iter." denotes the average number of iterations, t_{iter} is the total times used by iterations and t_{setup} is the setup time.

Degrees of freedom for varying h					Solver performance		
h	# Tetrahedra	dim(X_h)	dim(Q_h)	Total VVH dof	# Iter.	t_{iter}	t_{setup}
$\frac{1}{4}$	3,072	14,739	729	30936	12	0.19	0.56
$\frac{1}{6}$	10,368	46,875	2197	98,144	14	0.73	3.57
$\frac{1}{8}$	24,576	107,811	4913	225,448	16	2.06	10.53
$\frac{1}{10}$	48,000	206,763	9261	432,048	18	5.20	21.38
$\frac{1}{12}$	82,944	352,947	15,625	737,144	20	10.26	49.30
$\frac{1}{14}$	131,712	555,579	24,389	1,159,936	22	22.15	98.23

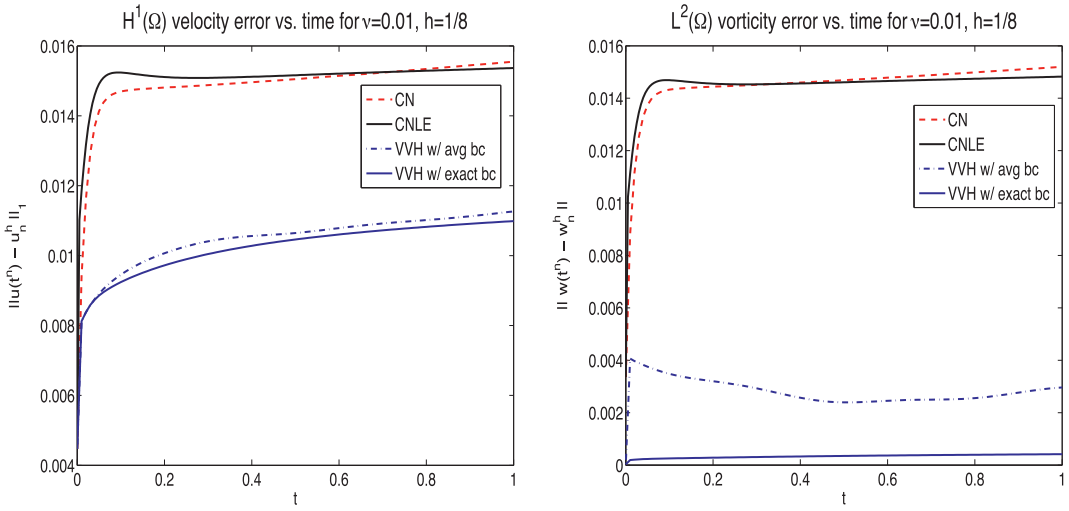

Fig. 2. The errors in discrete velocity and vorticity versus time, for CN, CNLE and VVH with averaging boundary condition, and VVH with exact boundary condition schemes for Ethier–Steinman problem with $\nu = 0.01$, using $h = 1/8, \Delta t = 0.01$ for VVH and CN, and $\Delta t = 0.005$ for CNLE.

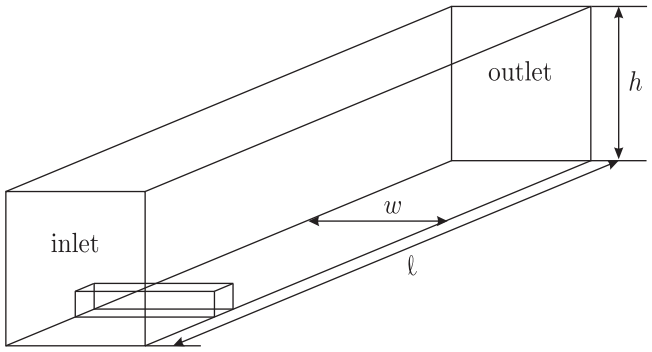

Fig. 3. Shown above the is domain for the 3D channel flow over a step problem.

The same block triangular preconditioner and the same inner and Poisson solvers as in the previous numerical example are used. We updated each preconditioner every 20 timesteps. The average number of iterations needed for each solve was 40 and 36, respectively, for the velocity–pressure and vorticity–helicity systems.

In Fig. 4, plots are provided for the velocity and vorticity solutions at $T = 10$, and these agree with the expected qualitative behavior; that is, it is clear from the plots that an eddy has detached and another has formed.

For a comparison of results, we also ran the CNLE algorithm (2.10), with the same mesh and initial and boundary conditions for velocity, but with $\Delta t = 0.02$ (so the overall cost is approximately the same as for VVH). The results of this simulation at $T = 10$ are shown in Fig. 5, and are visibly less accurate than for VVH, in that we do not see eddy detachment and reformation. Based on these results, VVH is more accurate with CNLE for this test problem.

One well known benefit of the vorticity based numerical method is that it gives direct access to the discrete vorticity, instead of computing the discrete vorticity by postprocessing as $\mathbf{w}_h := \nabla \times \mathbf{u}_h$. In the present formulation (1.1), the computation of vorticity and velocity are even one step further decoupled in the sense that instead of solving $\Delta \mathbf{u} = -\nabla \times \mathbf{w}$ (as many vorticity formulations do), the discrete velocity directly solves the momentum equation, where the vorticity enters the nonlinearity. This observation suggests that the difference $|\mathbf{w}_h - \nabla \times \mathbf{u}_h|$ can be a reasonable measure of the discrete solution accuracy and thus to serve as a simple and easily computable error indicator for a mesh adaptation. The same is true for the difference $|\eta_h - \mathbf{w}_h \cdot \mathbf{u}_h|$. We will study such adaptive strategies elsewhere. Here we illustrate our hypothesis by plotting the difference $|\mathbf{w}_h - \nabla \times \mathbf{u}_h|$ in Fig. 6 (top plot). Note that the difference is large precisely near the step corners where the solution is known to be non-smooth, but not necessarily in those regions where the vorticity magnitude is large (see the bottom plot in Fig. 6).

4. Preconditioners and solvers for steady problems

We now turn to the solution of a velocity based system in the steady case. Use of a vorticity based formulation for steady-state computations can lead to more accurate computed velocities near the boundary, and is a natural approach when the vorticity is required.

Solution of the discrete VVH system in the steady case poses considerable challenges from the linear algebra point of view. As already discussed, in the unsteady case, decoupling of the velocity and vorticity fields results in two fairly standard saddle-point problems which can be effectively solved by GMRES with block triangular preconditioning. For steady problems, on the other hand, Newton (or Picard) linearization leads to a sequence of coupled block 4×4 linear systems for the velocity, Bernoulli pressure, vorticity and helical density. Here we propose tackling this challenging system by GMRES with block triangular preconditioning, so that decoupling of the unknowns now takes place when applying the preconditioner within a GMRES step. Our approach should be considered as a first attempt only, and more work is necessary to make this approach competitive.

We use Newton's method to converge to the solution of the nonlinear problem (2.6). For higher Reynolds numbers, Newton's method should be combined with a continuation technique with respect to ν. Suppressing the spatial discretization notation, the Newton linearization of the system (2.6) reads: Given the velocity and vorticity approximations U and W solve for the updates $\mathbf{u}, \mathbf{w}, P, \eta$ the system

$$\begin{cases} -\nu\Delta\mathbf{u} - \gamma_1 \nabla\nabla\cdot\mathbf{u} + W\times\mathbf{u} + \nabla P + \mathbf{w}\times U = \mathbf{f}_u, \\ \nabla\cdot\mathbf{u} = g_u, \\ 2D(W)\mathbf{u} - \nu\Delta\mathbf{w} - \gamma_2\nabla\nabla\cdot\mathbf{w} + 2D(\mathbf{w})U - \nabla\eta = \mathbf{f}_v, \\ \nabla\cdot\mathbf{w} = g_v, \\ \mathbf{u}|_{\partial\Omega} = \mathbf{0}, \quad \mathbf{w}|_{\partial\Omega} - \nabla\times\mathbf{u}|_{\partial\Omega} = g_{bc}. \end{cases} \tag{4.14}$$

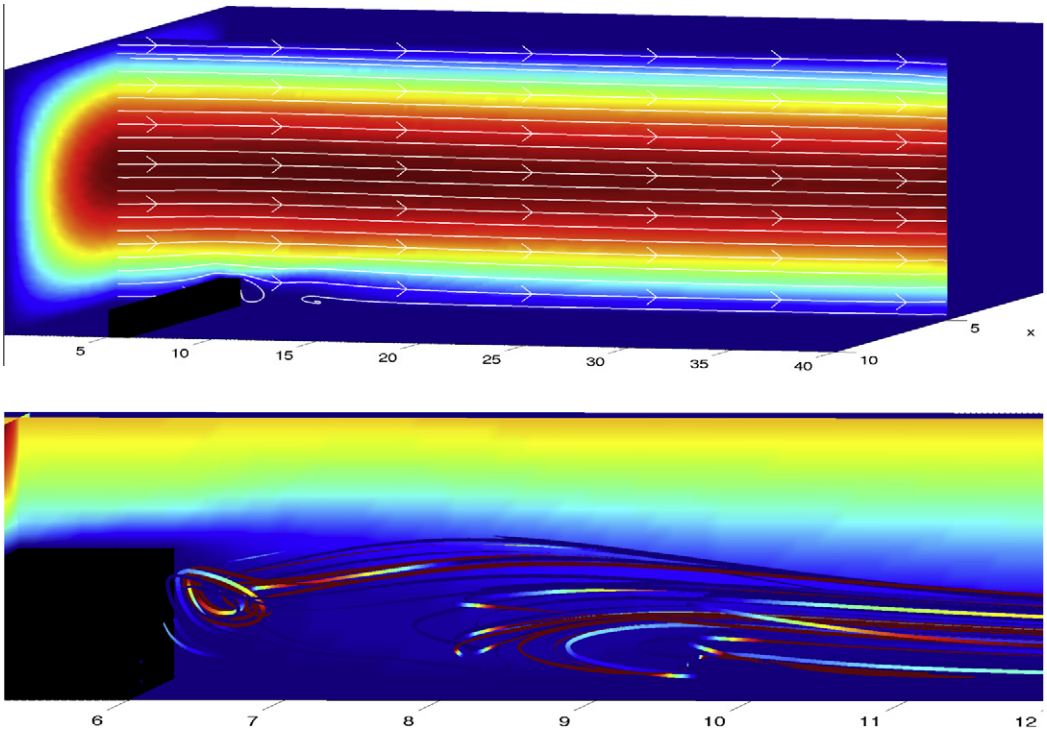

Fig. 4. Shown above is the VVH $T = 10$ solution (top) velocity streamlines over speed contours over the entire channel, (bottom) velocity streamribbons zoomed in at the step.

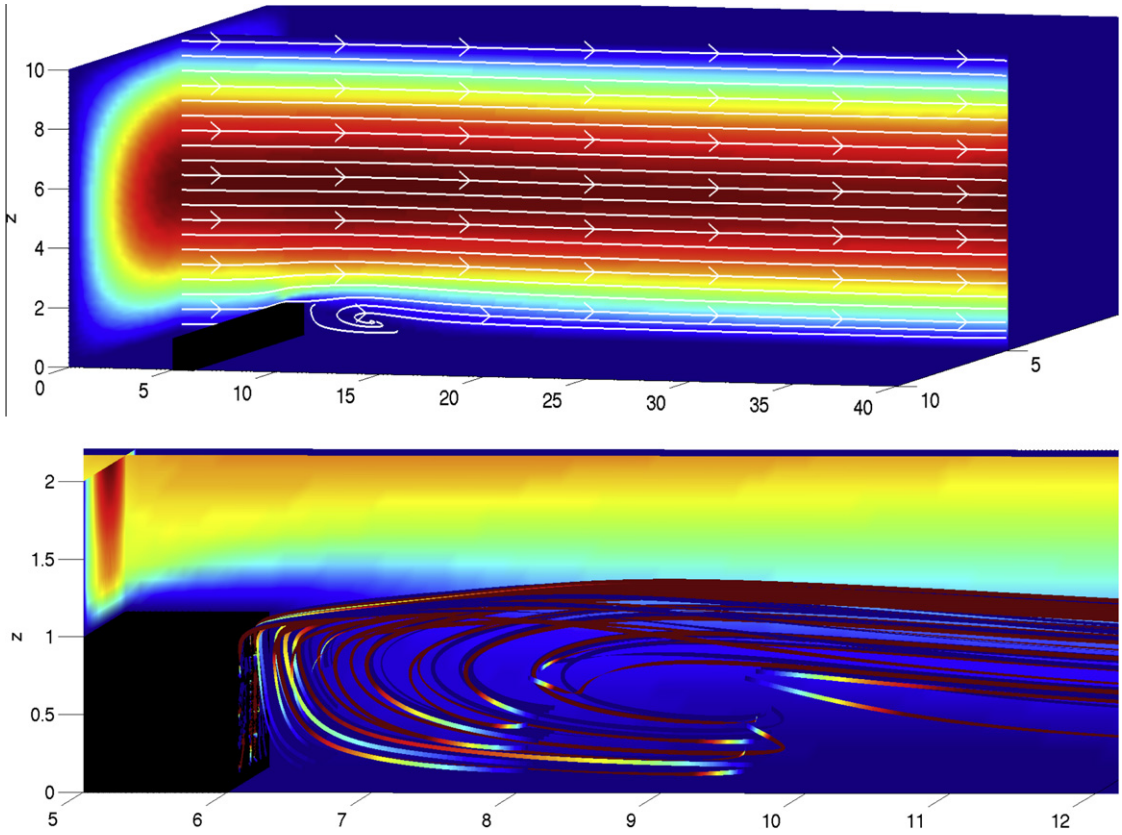

Fig. 5. Shown above is the $T = 10$ CNLE solution (top) velocity streamlines over speed contours over the entire channel, (bottom) velocity streamribbons zoomed in at the step.

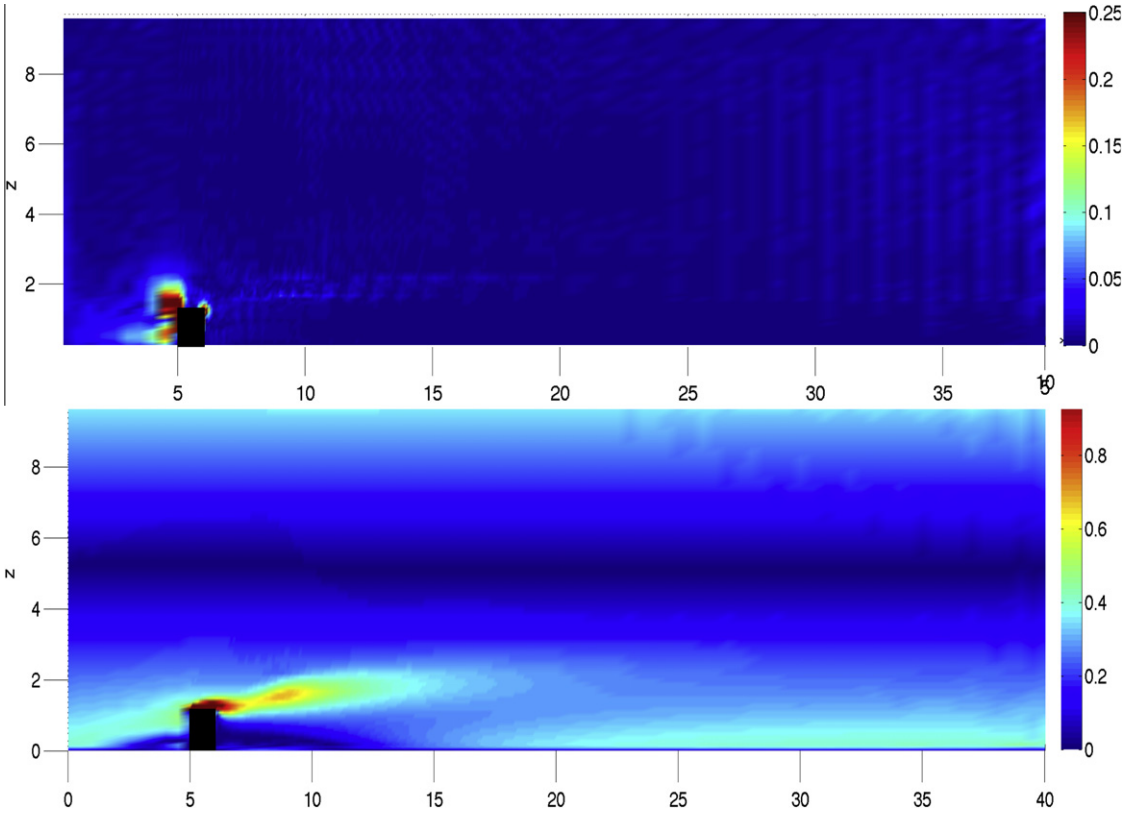

Fig. 6. Shown above are the $x = 5$ sliceplanes at $T = 10$, zoomed in near the step, of the magnitude of the difference $\mathbf{w}_h - \nabla \times \mathbf{u}_h$ (top) and vorticity magnitude (bottom) computed by the VVH method.

with $\{\mathbf{f}_u, g_u, \mathbf{f}_v, g_v, g_{bc}\}$ standing for a (nonlinear) residual. We remark that the last equation in (4.14), representing the boundary coupling of the vorticity and velocity, requires a special treatment while solving the discrete linear system iteratively. In particular, we enforce in the iteration that $\mathbf{w}|_{\partial\Omega}$ be equal to the nodal average of $\nabla \times \mathbf{u}$, on the boundary, from the previous iteration.

For the sake of clarity, assume that the vorticity boundary conditions are decoupled from the velocity, say $\mathbf{w}|_{\partial\Omega} = 0$, and do not contribute to the vorticity d.o.f. Given the structure of the system in (4.14), the algebraic form of the finite element linearized equations in our case is the following coupled system:

$$\begin{pmatrix} A_u & -B^T & M & 0 \\ -B & 0 & 0 & 0 \\ N & 0 & A_v & B^T \\ 0 & 0 & B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ P \\ \mathbf{w} \\ \eta \end{pmatrix} = \begin{pmatrix} f_u \\ g_u \\ f_v \\ g_v \end{pmatrix}. \tag{4.15}$$

More specifically, the four blocks in the upper left corner

$$\begin{pmatrix} A_u & -B^T \\ -B & 0 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & -B_1^T \\ A_{21} & A_{22} & A_{23} & -B_2^T \\ A_{31} & A_{32} & A_{33} & -B_3^T \\ -B_1 & -B_2 & -B_3 & 0 \end{pmatrix} \tag{4.16}$$

correspond to the rotation form of the linearized Navier–Stokes equations. The diffusive term multiplied by the viscosity ν is contained in the diagonal blocks of A_u, and the cross-product terms are included in off-diagonal blocks $A_{ij}, i \neq j$. The grad-div stabilization terms with parameter γ_1 are in all nine blocks of A_u. The four blocks in the lower right corner of (4.15), $\begin{pmatrix} A_v & B^T \\ B & 0 \end{pmatrix}$, which arise from the vorticity–helicity saddle point system, are similar in form to the convection form of the linearized Navier–Stokes equation, but the convection term is distributed in all nine blocks of A_v due to the definition of $D(\mathbf{w})$.

Observe that the coupled VVH system (4.15) is singular for the Ethier–Steinman and step problem we consider in this paper. In both problems, the Bernoulli pressure P and helical density η are unique up to an additive constant, making the linear system in (4.15) rank deficient by 2 (because B is rank deficient by one). One may either remove these singularities by setting a single Dirichlet degree of freedom for both P and η, but as is the case for velocity–pressure systems as well, when using Krylov solvers these singularities need not be removed provided the iterations take place in an appropriate subspace [10]. Preconditioning techniques for saddle point problems have been studied intensively in recent years see, e.g., [1,2,4,9,11] as well as the systematic treatment in [10]. Here we focus on augmented Lagrangian preconditioning [2,4], which is especially well-suited when grad-div stabilization is applied to the velocity and vorticity equations.

To build a preconditioner for (4.15), assume we are given a generalized saddle point system of the form

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \quad \text{or } \mathcal{A}\mathbf{x} = \mathbf{b}. \tag{4.17}$$

The augmented Lagrangian (AL) approach from [2] consists first of replacing the original system (4.17) with the equivalent one

$$\begin{pmatrix} A + \gamma B^T W^{-1} B & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \quad \text{or } \mathcal{A}_\gamma \mathbf{x} = \mathbf{b}, \tag{4.18}$$

followed by preconditioning (4.18) with a block triangular preconditioner of the form

$$\mathcal{P}_\gamma = \begin{pmatrix} \widehat{A}_\gamma & B^T \\ 0 & -\widehat{S}_\gamma \end{pmatrix}. \tag{4.19}$$

Here and in the following $\widehat{A}_\gamma$ denotes a preconditioner for the velocity block $A_\gamma = A + \gamma B^T W^{-1} B$ and $\widehat{S}_\gamma$ is a preconditioner for the Schur complement of the augmented system $S_\gamma = B(A + \gamma B^T W^{-1} B)^{-1} B^T$. Based on the identity

$$S_\gamma^{-1} = S_0^{-1} + \gamma W^{-1},$$

a reasonable choice of $\widehat{S}_\gamma$ is the scaled W matrix, e.g., $\widehat{S}_\gamma = \gamma^{-1} W$, where W is typically a diagonal matrix, for example, an approximation of the pressure mass matrix in the case of a linearized Navier–Stokes problem. Eigenvalue bounds for $\mathcal{P}_\gamma^{-1}\mathcal{A}_\gamma$ have been established in [2,4], and field of values type bounds for $\mathcal{P}_\gamma^{-1}\mathcal{A}_\gamma$, which lead to rigorous convergence estimates for GMRES, have been proved in [3].

In this paper we study the augmented Lagrangian preconditioning, when the augmentation is introduced on the differential level, the so called "first augment, then discretize" approach. This approach allows us both to improve accuracy of the finite-element solution (see Section 2.1) and to build an efficient preconditioner. Indeed, the matrix A_u can be decomposed as $A_u = A + \gamma_1 G$, where A corresponds to the discretization of $-\nu\Delta + \mathbf{w}\times$ operator, while G discretizes $-\nabla\nabla\cdot$. Thus adding $\gamma_1 G$ is similar from an algebraic point of view to the addition of $\gamma_1 B^T W^{-1} B$ with W given by the pressure mass matrix. The same observation is valid for the matrix A_v. Since (4.16) can be regarded as the augmented Lagrangian linear system, we consider the modified variant of the AL preconditioner:

$$\begin{pmatrix} \widehat{A}_u & -B^T \\ 0 & -\widehat{S}_u \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & -B_1^T \\ 0 & A_{22} & A_{23} & -B_2^T \\ 0 & 0 & A_{33} & -B_3^T \\ 0 & 0 & 0 & -\widehat{S}_u \end{pmatrix} \tag{4.20}$$

where $\widehat{S}_u^{-1} = (\gamma_1 + \nu)\widehat{M}_p^{-1}$ and $\widehat{M}_p$ is the main diagonal of the pressure mass matrix M_p. The presence of the grad-div stabilization terms in A_{11}, A_{22} and A_{33} makes the preconditioner (4.20) of augmented Lagrangian type. For the vorticity–helicity system, a similar block upper-triangular preconditioner is used except that the grad-div stabilization parameter is γ_2 instead of γ_1. To solve subproblems with A_{11}, A_{22} and A_{33} in the velocity and vorticity blocks, we consider different inexact solvers.

For the coupled system (4.15), we define the following block lower triangular approximation

$$\begin{pmatrix} \widehat{A}_u & -B^T & 0 & 0 \\ 0 & -\widehat{S}_u & 0 & 0 \\ N & 0 & \widehat{A}_v & B^T \\ 0 & 0 & 0 & -\widehat{S}_v \end{pmatrix}, \tag{4.21}$$

as the *global preconditioner*, where $\widehat{A}_u$ and $\widehat{A}_v$ are corresponding block upper triangular approximations of A_u and A_v. The reason for using a block lower triangular matrix is that keeping N, a discrete analogue of the $2D(W)\mathbf{u}$ operator, appeared to be superior to including M, a discrete analogue of the $\mathbf{w} \times U$ operator.

In (4.21), since S_u and S_v are both diagonal, the major computation lies in solving linear systems with the diagonal blocks of A_u and A_v. For these inner solves, we can use sparse direct methods, but these become quickly prohibitive for the 3D problems of interest here. Here we compare the incomplete LU factorization and the algebraic multigrid method (AMG) implemented in IFISS 3.1 [12]. Note that the incomplete LU factorization has been optimized and built into MATLAB (`ilu` function), so it is very efficient, while AMG is written in MATLAB, and therefore it is slower than incomplete LU factorization in terms of execution time.

We also investigate an inner–outer Flexible GMRES (FGMRES) scheme. For the latter we use the implementation based on the simpler GMRES algorithm described in [19]. Here, to solve the linear systems with the velocity–pressure equation and the vorticity–helicity equation, instead of applying one action of the AL-type preconditioners, a few inner GMRES iterations with corresponding preconditioners are used. This inevitably increases the cost, but we find it significantly reduces the outer FGMRES iterations and thus total iteration time. This method is found to be, by far, the most efficient of those tested.

4.1. Numerical experiment: steady Ethier–Steinman flow

We now test the methods described above on two test problems, a steady analog of the Ethier–Steinman problem, and the steady channel flow over a step problem with Re = 50. Here, we suppose the solution is time-independent, which is done by simply using the Ethier–Stienman solution with the $e^{-\nu d^2 t}$'s in (3.11) removed. This leads to a nonzero right hand side function, and we compute using the solution for the inhomogeneous Dirichlet boundary conditions. We take the Ethier–Steinman parameters as $a = d = 1$, kinematic viscosity $\nu = 0.02$, and $\gamma_1 = 0.2, \gamma_2 = 0.5$.

The results using these various solvers are given in Tables 2,3. First, we show L^2 norms of the velocity and vorticity errors. Further Table 2 gives iteration counts and timings for FGMRES, with a few ILU-preconditioned GMRES iterations for computing (4.16) and the vorticity–helicity block in (4.15). Note that this results in a variable (hence, non-linear) block triangular preconditioner. This table also shows results for GMRES preconditioned by (4.21); no inner iterations where executed. Finally, we repeat the same experiments with ILU replaced by the AMG preconditioner. These results are shown in Table 3. In all the tables, the first number in the "Iterations" column is the number of Newton iterations, and the second is the average (F)GMRES iterations. For the outer FGMRES, restarts were done every 50 iterations, the maximum number of iterations was set to 500, and the convergence tolerance to 1 e−5. For the inner GMRES, the maximum number of iterations was set to 10, and the tolerance to 1 e−3 (although this tolerance was never reached). For the global preconditioner, GMRES with restarts every 50 iterations, maximum number of iterations was set to 500, and the tolerance 1 e−5 was used; for the associated inner solvers, ILU used a drop tolerance of 1 e−3. The AMG used is the IFISS 3.1 implementation with ILU smoother (this was found to be more effective than damped point Jacobi and Gauss–Seidel), and the levels are automatically created by the algorithm. In our problems, 13–19 levels are generated.

Note that for this analytical example the L^2 norms of the velocity and vorticity errors scale approximately as $O(\text{dof}^{-1})$, which is optimal for piecewise quadratic finite elements in 3D. The modified augmented Lagrangian preconditioner results in convergent Krylov subspace iterations when ILU is used for approximating the block solves. FGMRES with inner iterations appears to be somewhat more efficient in terms of timings than the plain GMRES with block triangular linear (i.e., constant) preconditioner. It is interesting to observe that the AMG method, known to be quite useful as an inner auxiliary solver for the Oseen problem in convection form (at least in 2D and for ν not too small [10]), generally fails for the vorticity systems.

4.2. Numerical experiment: steady channel flow over a step

We observed in the previous test that the inner–outer FGMRES with ILU solver performed the best on the steady Ethier–Steinman problem. We now test this solver on the physically motivated,

Table 2
Finite element errors; timings and iterations of inner–outer FGMRES with block triangular non-linear preconditioner (using ILU) and with block triangular linear preconditioner (using ILU).

DOF	FE error		Non-linear preconditioner			Linear preconditioner		
	$\|\mathbf{u}-\mathbf{u}_h\|_{L^2}$	$\|\mathbf{w}-\mathbf{w}_h\|_{L^2}$	t_{setup}	t_{iter}	# Iter.	t_{setup}	t_{iter}	# Iter.
30,936	1.1×10^{-2}	2.6×10^{-2}	1.5	9.5	3, 13.3	1.5	20.2	3, 176.3
225,448	1.1×10^{-3}	2.5×10^{-3}	62.1	78.9	3, 13.7	62.0	85.4	3, 96.3
432,048	4.9×10^{-4}	1.1×10^{-3}	231.1	156.6	3, 14.0	231.0	193.6	3, 108.3
737,144	2.5×10^{-4}	5.8×10^{-4}	624.4	293.1	3, 15.3	623.7	414.2	3, 127.3

Table 3
Timings and iterations of inner–outer FGMRES with block triangular non-linear preconditioner (using AMG) and with block triangular linear preconditioner (using AMG).

DOF	Non-linear preconditioner			Linear preconditioner		
	t_{setup}	t_{iter}	# Iter.	t_{setup}	t_{iter}	# Iter.
30,936	39.4	1520.6	5, 500	Stagnates		
225,448	Stagnates					

more complex test problem of 3D channel flow over a forward–backward step. The problem we study is the steady analog of the problem studied in Section 3, using $\nu = 0.05$ and 0.02. Again we choose grad-div stabilization parameters to be $\gamma_1 = 0.2$ and $\gamma_2 = 0.5$. We test the solver on several meshes, and show the results in Table 4, and observe that the solver is quite effective on this problem as well. To test the method, we compute solutions on five different mesh levels. For the purpose of comparison we also give timing of the direct sparse solver (Matlab's 'backslash') applied to the same problem.

Similar to the analytical test the inner–outer FGMRES with ILU as the preconditioner for the inner iterations was the best of all the methods we tried. Using AMG instead of ILU results in stagnation for DOF = 18,922 and $\nu = 0.05$. The global preconditioner with linear block triangular preconditioner stagnates for DOF = 58,656 and $\nu = 0.05$. In this later case, the reason is that FGMRES does not reach the tolerance 10^{-3}.

5. Conclusions and future directions

We studied a recently introduced vorticity based solver for the incompressible Navier–Stokes equations. For two non-steady problems the solver was found to provide more accurate solutions than the more common primitive variables formulation. The complexity of both approaches is, however, comparable if one uses the natural and stable splitting scheme to decouple time advances in velocity and vorticity. Linear algebraic solvers for non-steady problems were found to perform equally well for the vorticity based and primitive variable formulations. In the steady case, however, the coupled VVH problem appears to pose a serious challenge from the viewpoint of algebraic solvers. In this case, we found the approach based on augmented Lagrangian preconditioner and inner–outer iterations to be the best among those we tried.

Many important questions remain open. Among them are finding alternative simple (e.g., weak) vorticity boundary conditions, looking for multiscale/stabilized formulations in **w**—**u** variables, error analysis for unsteady problem, the study of error indicators (e.g., based on $\mathbf{w}_h - \nabla\times\mathbf{u}_h$ and $\eta_h - \mathbf{u}_h\cdot\mathbf{w}_h$ quantities) and adaptive methods. We plan to address these questions in the future.

Table 4
Timings and iteration counts of inner–outer FGMRES (ILU) and direct solve.

DOF	Newton	t_{setup}	t_{iter}	# Iter.	t_{direct}
$\nu = 0.05$					
2394	8	0.01	10.6	123.4	0.1
18,922	5	0.4	6.6	18.8	3.8
58,656	4	3.1	23.4	20.8	28.0
286,360	4	55.8	148.1	21.8	722.8
490,240	4	162.0	303.7	23.5	2328.55
$\nu = 0.02$					
2394	8	0.01	10.6	123.4	0.1
18,922	5	0.4	6.4	18.8	3.7
58,656	4	3.2	23.4	20.8	29.1
286,360	4	54.8	148.3	21.8	742.4
490,240	4	167.0	295.8	23.5	2337.1

References

[1] M. Benzi, J. Liu, An efficient solver for the Navier–Stokes equations in rotation form, SIAM J. Sci. Comput. 29 (2007) 1959–1981.
[2] M. Benzi, M.A. Olshanksii, An augmented Lagrangian-based approach to the Oseen problem, SIAM J. Sci. Comput. 28 (6) (2006) 2095–2113.
[3] M. Benzi, M.A. Olshanskii, Field-of-values convergence analysis of augmented Lagrangian preconditioners for the linearized Navier–Stokes problem, SIAM J. Numer. Anal. 49 (2011) 770–788.
[4] M. Benzi, M.A. Olshanskii, Z. Wang, Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 66 (2011) 486–508.
[5] J. Cahouet, J.-P. Chabard, Some fast 3D finite element solvers for the generalized Stokes problem, Int. J. Numer. Methods Fluids 8 (1988) 869–895.
[6] Q. Chen, S. Chen, G. Eyink, The joint cascade of energy and helicity in three dimensional turbulence, Phys. Fluids 15 (2) (2003) 361–374.
[7] B. Cousins, L.G. Rebholz, N. Wilson, Enforcing energy, helicity and strong mass conservation in finite element computations for incompressible Navier–Stokes simulations, Appl. Math. Comput. 281 (2011) 1208–1221.
[8] C. Davies, P.W. Carpenter, A novel velocity–vorticity formulation of the Navier–Stokes equations with applications to boundary layer disturbance evolution, J. Comput. Phys. 172 (2001) 119–165.
[9] H. Elman, D. Silvester, Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations, SIAM J. Sci. Comput. 17 (1) (1996) 33–46.
[10] H. Elman, D. Silvester, A. Wathen, Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2005.
[11] H.C. Elman, D.J. Silvester, A.J. Wathen, Performance and analysis of saddle point preconditioners for the discrete steady-state Navier–Stokes equations, Numer. Math. 90 (2002) 665–688.
[12] H. Elman, A. Ramage, D. Silvester, Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow, ACM Trans. Math. Softw. 33 (2007) 2–14.
[13] C. Ethier, D. Steinman, Exact fully 3D Navier–Stokes solutions for benchmarking, Int. J. Numer. Methods Fluids 19 (5) (1994) 369–375.
[14] J.A. Evans, Divergence-free B-spline Discretizations for Viscous Incompressible Flows, Ph.D. thesis, University of Texas at Austin, 2012.
[15] C. Foias, L. Hoang, B. Nicolaenko, On the helicity in 3D-periodic Navier–Stokes equations i: the non-statistical case, Proc. Lond. Math. Soc. 94 (2007) 53–90.
[16] T.B. Gatski, Review of incompressible fluid flow computations using the vorticity–velocity formulation, Appl. Numer. Math. 7 (1991) 227–239.
[17] V. Girault, P.-A. Raviart, Finite Element Methods for Navier–Stokes equations: Theory and Algorithms, Springer-Verlag, 1986.
[18] G. Guevremont, W.G. Habashi, M.M. Hafez, Finite element solution of the Navier–Stokes equations by a velocity–vorticity method, Int. J. Numer. Methods Fluids 10 (1990) 461–475.

[19] P. Jiranek, M. Rozloznik, Adaptive version of simpler GMRES, Numer. Algorithms 53 (2010) 93–112.
[20] V. John, A. Liakos, Time dependent flow across a step: the slip with friction boundary condition, Int. J. Numer. Methods Fluids 50 (2006) 713–731.
[21] W. Layton, An Introduction to the Numerical Analysis of Viscous Incompressible Flows, SIAM, 2008.
[22] W. Layton, C. Manica, M. Neda, M.A. Olshanskii, L. Rebholz, On the accuracy of the rotation form in simulations of the Navier–Stokes equations, J. Comput. Phys. 228 (5) (2009) 3433–3447.
[23] H.K. Lee, M.A. Olshanskii, L. Rebholz, On error analysis for the 3D Navier–Stokes equations in Velocity–vorticity–helicity form, SIAM J. Numer. Anal. 49 (2011) 711–732.
[24] D.C. Lo, D.L. Young, K. Murugesan, An accurate numerical solution algorithm for 3D velocity–vorticity Navier–Stokes equations by the DQ method, Commun. Numer. Methods Engrg 22 (2006) 235–250.
[25] H.L. Meitz, H.F. Fasel, A compact-difference scheme for the Navier–Stokes equations in vorticity–velocity formulation, J. Comput. Phys. 157 (2000) 371–403.
[26] H. Moffatt, A. Tsoniber, Helicity in laminar and turbulent flow, Annu. Rev. Fluid Mech. 24 (1992) 281–312.
[27] M.A. Olshanskii, A fluid solver based on vorticity–helical density equations with application to a natural convection in a cubic cavity, Int. J. Numer. Methods Fluids. http://dx.doi.org/10.1002/fld.2622.
[28] M.A. Olshanskii, Iterative solver for Oseen problem and numerical solution of incompressible Navier–Stokes equations, Numer. Linear Algebra Appl. 6 (1999) 353–378.
[29] M.A. Olshanskii, G. Lube, T. Heister, J. Löwe, Grad-div stabilization and subgrid pressure models for the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 198 (2009) 3975–3988.
[30] M.A. Olshanskii, L. Rebholz, A note on helicity balance of the Galerkin method for the 3D Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1032–1035.
[31] M.A. Olshanskii, L. Rebholz, Velocity–vorticity–helicity formulation and a solver for the Navier–Stokes equations, J. Comput. Phys. 229 (2010) 4291–4303.
[32] M.A. Olshanskii, A. Reusken, Grad-Div stabilization for the Stokes equations, Math. Comput. 73 (2004) 1699–1718.
[33] Y. Saad, M.H. Schultz, GMRES: a generalized minimum residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[34] S. Saitoh, S. Wakashima, Benchmark solutions for natural convection in a cubic cavity using the high-order time-space method, Int. J. Heat Mass Transfer 47 (2004) 853–864.
[35] K.L. Wong, A.J. Baker, A 3D incompressible Navier–Stokes velocity–vorticity weak form finite element algorithm, Int. J. Numer. Methods Fluids 38 (2002) 99–123.