Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 199 (2004) 221–259
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term

Hans Johnston [a,*], Jian-Guo Liu [b]

[a] Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, USA
[b] Institute for Physical Science and Technology and Department of Mathematics, University of Maryland, College Park, MD 20742, USA

Received 7 November 2003; received in revised form 5 February 2004; accepted 6 February 2004
Available online 19 March 2004

Abstract

We present numerical schemes for the incompressible Navier–Stokes equations based on a primitive variable formulation in which the incompressibility constraint has been replaced by a pressure Poisson equation. The pressure is treated explicitly in time, completely decoupling the computation of the momentum and kinematic equations. The result is a class of extremely efficient Navier–Stokes solvers. Full time accuracy is achieved for all flow variables. The key to the schemes is a Neumann boundary condition for the pressure Poisson equation which enforces the incompressibility condition for the velocity field. Irrespective of explicit or implicit time discretization of the viscous term in the momentum equation the explicit time discretization of the pressure term does not affect the time step constraint. Indeed, we prove unconditional stability of the new formulation for the Stokes equation with explicit treatment of the pressure term and first or second order implicit treatment of the viscous term. Systematic numerical experiments for the full Navier–Stokes equations indicate that a second order implicit time discretization of the viscous term, with the pressure and convective terms treated explicitly, is stable under the standard CFL condition. Additionally, various numerical examples are presented, including both implicit and explicit time discretizations, using spectral and finite difference spatial discretizations, demonstrating the accuracy, flexibility and efficiency of this class of schemes. In particular, a Galerkin formulation is presented requiring only C^0 elements to implement.

Keywords: Incompressible flow; Navier–Stokes; Pressure Poisson equation; Explicit pressure discretization; Unconditional stability; Galerkin formulation; Spectral method; Finite difference method

1. Introduction

The classical primitive variables formulation of the incompressible Navier–Stokes equations on a domain $\Omega \subset \mathbb{R}^3$ (or $\mathbb{R}^2$) are given by

* Corresponding author. Tel.: +1-734-764-0335; fax: +1-734-763-0937.
E-mail addresses: hansjohn@umich.edu (H. Johnston), jliu@math.umd.edu (J.-G. Liu).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2004.02.009

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \nu\Delta\boldsymbol{u} + \boldsymbol{f}\ , \tag{1.1a}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{1.1b}$$

$$\boldsymbol{u}|_{\Gamma} = 0, \tag{1.1c}$$

where $\boldsymbol{u} = (u, v, w)^{\mathrm{T}}$ (or $\boldsymbol{u} = (u, v)^{\mathrm{T}}$), p, ν, $\boldsymbol{f}$, and Γ represent the velocity field, pressure, kinematic viscosity, force term, and boundary of Ω, respectively. Note that, for simplicity of presentation, we have taken the no-slip boundary condition (1.1c) for $\boldsymbol{u}$.

Beginning with the pioneering MAC scheme [17] of Harlow and Welch in 1965, enforcement of the incompressibility constraint (1.1b) has always been, and rightly so, a central theme in the development of numerical methods for the Navier–Stokes equations. MAC scheme is a direct discretization of (1.1) using second order finite differences implemented on a staggered grid with explicit time treatment of the nonlinear convection and viscous terms, and implicit time treatment of the pressure term. In addition to enforcing incompressibility, the key success of the scheme is its efficiency, which results from the decoupling of the computation of the momentum and kinematic equations. This is accomplished by applying the incompressibility constraint to the discretized momentum equation, resulting in a discrete Poisson equation for the pressure. However, besides the restriction to a second order finite difference discretization for simple domains, there are three main drawbacks to the MAC scheme in its original form: (1) if the viscous term is treated implicitly the decoupling process is no longer possible, (2) parasitic pressure modes can arise if alternative spatial discretizations (e.g. a non-staggered grid) are used; the resulting discrete system has more unknowns than equations (cf. [20]), and (3) low order time discretization was used, resulting in a severe cell Reynolds number time step constraint.

Of course, in the intervening years much work has gone into addressing the shortcomings of the MAC scheme. In the late 60s Chorin [6] and Temam [32] introduced the projection method, which allows implicit treatment of the viscous term while retaining the efficiency of the MAC scheme. Projection methods are time splitting schemes in which an intermediate velocity is first computed and then projected onto the space of incompressible vector fields by solving a Poisson equation for pressure. Unfortunately, the time splitting introduces numerical boundary layers in the pressure and intermediate velocity field. It is possible to avoid them in the velocity field if an exact projection is used. However, this can only be achieved in simple geometries (e.g. one for which a staggered grid can be implemented), and even in this case the numerical boundary layers remain in the pressure. Greater control in reducing these numerical boundary layers was gained with the development of modern second order projection methods in the mid 80s [1,21,24,26,35]. They are by far the most popular schemes used in practice for Navier–Stokes, particularly for moderate to high Reynolds number flows since the strength of the numerical boundary layer decreases with increasing Reynolds number. However, the main advantage of implicit treatment of the viscous term in the projection method is for low Reynolds number flow, and in this regime the effects of the numerical boundary layers are quite severe [10]. Attempts to better understand and control these numerical artifacts remain an active area of research (cf. [10,33]). Indeed, a second order implicit projection method which achieves full temporal accuracy for both the velocity and pressure in rectangular domains was recently proposed by Brown et al. [3]. Unfortunately, due to the high order spatial derivatives required in their pressure update it is doubtful that full accuracy for the pressure can be achieved in general domains.

For the low Reynolds number (diffusion dominated) flow regime, fast Stokes solver schemes for fully coupled systems, particularly popular in the finite element community, have proven highly successful due to the maturity of multigrid, preconditioned conjugate-gradient and domain decomposition methods (cf. [12,13,36]). To avoid spurious pressure modes in the coupled system, the so-called inf–sup compatibility condition (or Babuska–Brezzi condition) between the pressure and velocity finite element approximation spaces has to be satisfied (cf. [13]). Unfortunately, this generally complicates the implementation of fast solvers for the resulting linear systems. Moreover, application of Stokes type solvers for the high Reynolds

number convection dominated regime is not recommended, for as the Reynolds number increases the convergence rate of the iterative linear solvers is significantly diminished.

Concerning the cell Reynolds number constraint, its cause is now well understood and is easily circumvented. Low order explicit time stepping schemes such as forward Euler and the Midpoint rule are unconditionally unstable for linear convection (transport) equations if center differencing is used for the spatial discretization; inclusion of a diffusion term stabilizes such schemes. However, for moderate to large Reynolds numbers the eigenvalues of the linearized system cluster near the imaginary axis but away from the real axis. Since the stability regions of low order explicit time stepping schemes do not encompass any portion of the imaginary axis the result is the cell Reynolds number constraint. The remedy is to simply use a higher order explicit scheme, such as classical fourth order Runge–Kutta, whose stability region contains an appreciable portion of the imaginary axis (cf. [7]).

In this paper we introduce a class of the numerical methods for Navier–Stokes equations based on the following equivalent pressure Poisson equation formulation of (1.1):

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \nu\Delta\boldsymbol{u} + \boldsymbol{f}\ , \tag{1.2a}$$

$$\Delta p = -\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) + \nabla\cdot\boldsymbol{f}\ , \tag{1.2b}$$

$$\left.\frac{\partial p}{\partial \boldsymbol{n}}\right|_{\Gamma} = [-\nu\boldsymbol{n}\cdot(\nabla\times\nabla\times\boldsymbol{u}) + \boldsymbol{n}\cdot\boldsymbol{f}\ ]|_{\Gamma}, \tag{1.2c}$$

$$\boldsymbol{u}|_{\Gamma} = 0 \tag{1.2d}$$

which is derived in Section 2. Here the incompressibility constraint $\nabla\cdot\boldsymbol{u} = 0$ has been replaced by a pressure Poisson equation (1.2b) and a Neumann boundary condition for the pressure (1.2c). This pressure boundary condition was first proposed by Orszag et al. [26] in the context of the projection methods for accuracy consideration, and it is referred to therein as an accurate pressure boundary condition. Their motivation for using it was as a means to both enforce a compatibility condition for the discrete linear system for pressure with a Neumann boundary condition, as well as a way of better enforcing incompressibility. Our primary motivation for using (1.2c), in addition to providing an implementable boundary condition for (1.2b), is for stability considerations. We show that irrespective of explicit or implicit time discretization of the viscous term in (1.2a), explicit time discretization of the pressure term does not affect the time step constraint. Indeed, *unconditional* stability for the Stokes equation with explicit treatment of the pressure term and implicit treatment of the viscous term is proven in Section 3.2. The proof relies on the observation that the pressure can be solved for in terms of the tangential component of the velocity, which in turn is controlled by the tangential component of the viscous term. Systematic numerical experiments are presented in Section 4.1 for the full NSE which indicate that a second order implicit time discretization of the viscous term that (1.2) is stable under the standard CFL condition.

The schemes based on (1.2) presented herein are also extremely efficient, for regardless of implicit or explicit treatment of the viscous term, explicit treatment of the pressure term decouples the computation of the momentum equation (1.2a) from that of the kinematic pressure Poisson equation (1.2b). The main computational cost at each time stage is then reduced to solving a standard Poisson equation, and two or three (2D or 3D) standard Helmholtz equations if the viscous term is treated implicitly. Moreover, full time accuracy is achieved for all flow variables.

Finally, the class of methods presented herein is quite flexible, for any standard spatial discretization such as finite difference, finite element or spectral methods can be used to approximate (1.2). In particular, a variational formulation for (1.2), derived in Section 2.1, is given by: Find $\boldsymbol{u}\in L^2(0,T;(H_0^1(\Omega)\cap H^2(\Omega))^d$ and $p\in L^2(0,T;H^1(\Omega)/C)$ such that $\forall\boldsymbol{v}\in(H_0^1(\Omega))^d$ and $\forall\psi\in H^1(\Omega)/C$

$$\langle \boldsymbol{u}_t, \boldsymbol{v}\rangle + \langle \boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{v}\rangle + \langle \nabla p, \boldsymbol{v}\rangle = -\nu\langle \nabla\boldsymbol{u}, \nabla\boldsymbol{v}\rangle + \langle \boldsymbol{f}\ , \boldsymbol{v}\rangle, \tag{1.3a}$$

$$\langle \nabla p, \nabla\psi\rangle = -\langle \boldsymbol{u}\cdot\nabla\boldsymbol{u}, \nabla\psi\rangle + \langle \boldsymbol{f}\ , \nabla\psi\rangle + \nu\langle \nabla\times\boldsymbol{u}, \boldsymbol{n}\times\nabla\psi\rangle_{\Gamma}. \tag{1.3b}$$

Note that only first order derivatives appear above, hence standard C^0 finite element spaces can be used to approximate the system, particularly desirable for a finite element implementation.

The remainder of this paper is organized as follows. In Section 2 equivalent pressure Poisson formulations of the Navier–Stokes equations are considered. We discuss why (1.2) is to be preferred for numerical implementation, and the Galerkin formulation (1.3) is derived. Time discretization issues are discussed in Section 3. In particular, we prove that both a first and second order semi-implicit discretization of (1.2) applied to the Stokes equations is unconditionally stable. A Chebyshev collocation implementation of (1.2) coupled with a second order implicit time discretization is presented in Section 4. The numerical results clearly indicate stability with respect to the standard CFL condition. A MATLAB code of this implementation can be found in Appendix A. A Legendre–Galerkin implementation of (1.3) coupled with explicit time discretization is presented in Section 5. Finally, in Section 6 a finite difference implementation of (1.2) is briefly discussed and the computation a flow past a cylinder in 2D presented.

2. Pressure Poisson formulations of the Navier–Stokes equations

In this section we investigate a number of Pressure Poisson equation (PPE) formulations of the Navier–Stokes equations (NSE). While each are equivalent to the NSE (1.1) at the continuous level the focus here is to understand why one formulation would be desirable over another as the basis for a numerical scheme. The objective is a PPE which can be solved accurately, and whose solution properly enforces incompressibility. The Galerkin formulation (1.3) in derived in Section 2.1.

To begin, recall the derivation of one possible form of the PPE. Taking the divergence of the momentum equation (1.1a) along with (1.1b) gives

$$\Delta p = -\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) + \nabla\cdot\boldsymbol{f}. \tag{2.1}$$

Enforcing the incompressibility constraint (1.1b) through the use of this PPE has been a main focus in the development of numerical methods for incompressible flow [14,15]. However, solely replacing (1.1b) by (2.1) does not result in a system equivalent to the NSE (1.1); additional conditions need to be enforced. Moreover, there is some choice in these conditions depending on the form of the PPE itself. For example:

(I) Enforce a divergence-free boundary condition for the velocity:

$$\nabla\cdot\boldsymbol{u}|_{\Gamma} = 0. \tag{2.2}$$

A system equivalent to (1.1) [14,20] consists of: (1.1a), (2.1), (1.1c), and (2.2). The necessity and role of the boundary condition (2.2) can be seen by deriving a evolution equation for $\phi = \nabla\cdot\boldsymbol{u}$. Take the divergence of (1.1a) and substitute the PPE (2.1) in the expression that results. Then assuming a divergence-free velocity field $\boldsymbol{u}$ at $t = 0$, ϕ is governed by the heat equation (PDE)

$$\begin{cases} \phi_t = \nu\Delta\phi \quad \text{for } t > 0, \\ \phi|_{\Gamma} = 0 \quad \text{for } t > 0, \\ \phi(0) = 0, \end{cases} \qquad \Rightarrow \qquad \phi = \nabla\cdot\boldsymbol{u} \equiv 0 \quad \text{for } t \geqslant 0.$$

Thus, incompressibility of $\boldsymbol{u}$ is enforced for $t > 0$ via (2.2) in the form of the homogeneous boundary condition for ϕ above. Note that for this PPE formulation there are two boundary conditions for $\boldsymbol{u}$, namely (1.1c) and (2.2), and none for p. The realization of (2.2) generally requires a global constraint, as in the spectral implementation of the capacitance matrix method [25]. In [19,20] the authors investigated a spatially second order finite difference method in which (2.2) is converted at the discrete level to a consistent local formula for the Neumann pressure boundary condition

$$(\partial p/\partial \boldsymbol{n})_{0,j} = (2\nu/\Delta x^2)(\boldsymbol{n}\cdot\boldsymbol{u})_{1,j},$$

where the subscript $(0,j)$ represents the value at grid point (x_0, y_j) on the the boundary (assume here a boundary parallel to the y-axis), and the subscript $(1,j)$ represents the value at the grid point (x_1, y_j), one stencil point inside the boundary. The derivation of this local pressure boundary condition is very similar to the derivation of a local vorticity boundary condition known as Thom's formula; see [7,20]. A higher order version of this approach was considered by Henshaw [18]. Unfortunately, when this local pressure boundary condition is used with explicit treatment of the pressure term a diffusive stability time constraint must be satisfied even in the case when the viscous term is treated implicitly. However, this drawback can be overcome by using the pressure boundary condition in (1.2), described in (IV).

(II) Retain the viscosity term in the PPE:

$$\Delta p = -\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) + \nu\Delta(\nabla\cdot\boldsymbol{u}) + \nabla\cdot\boldsymbol{f}. \tag{2.3}$$

An system equivalent to (1.1) [14] is given by: (1.1a), (2.3) and (1.1c). Taking the divergence of (1.1a) and applying (2.3) gives the following evolution equation for ϕ (ODE):

$$\begin{cases} \phi_t = 0 \quad \text{for } t > 0, \\ \phi(0) = 0, \end{cases} \quad \Rightarrow \quad \phi = \nabla\cdot\boldsymbol{u} \equiv 0 \quad \text{for } t \geqslant 0, \tag{2.4}$$

again ensuring incompressibility of $\boldsymbol{u}$ for $t > 0$. However, the third order derivative in the PPE would require at least C^1 elements be used in a finite element implementation of this formulation.

(III) Rewrite the viscous term in rotational form:

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = -\nu\nabla\times\nabla\times\boldsymbol{u} + \boldsymbol{f}. \tag{2.5}$$

An equivalent system is given by: (2.5), (2.1) and (1.1c). Again ϕ is governed by the ODE system (2.4), ensuring incompressibility. This formulation is well suited for numerical discretization only if we treat the viscous term in (2.5) explicitly. Implicit treatment would result in a coupled system for all the components of $\boldsymbol{u}$, and solving this non-standard elliptic system numerically, especially in 3D, is costly.

(IV) Enforce the following Neumann boundary condition for the pressure:

$$\frac{\partial p}{\partial \boldsymbol{n}} = -\nu\boldsymbol{n}\cdot(\nabla\times\nabla\times\boldsymbol{u}) + \boldsymbol{n}\cdot\boldsymbol{f}\ , \tag{2.6}$$

where $\boldsymbol{n}$ is a unit normal along Γ. A system equivalent to (1.1) is then given by

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \nu\Delta\boldsymbol{u} + \boldsymbol{f}\ , \tag{2.7a}$$

$$\Delta p = -\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) + \nabla\cdot\boldsymbol{f}\ , \tag{2.7b}$$

$$\left.\frac{\partial p}{\partial \boldsymbol{n}}\right|_{\Gamma} = [-\nu\boldsymbol{n}\cdot(\nabla\times\nabla\times\boldsymbol{u}) + \boldsymbol{n}\cdot\boldsymbol{f}\ ]|_{\Gamma}, \tag{2.7c}$$

$$\boldsymbol{u}|_{\Gamma} = 0, \tag{2.7d}$$

the same as (1.2) and repeated here for convenience. To see that incompressibility is enforced recall the vector identity

$$\Delta \boldsymbol{u} = -\nabla \times \nabla \times \boldsymbol{u} + \nabla(\nabla \cdot \boldsymbol{u}). \tag{2.8}$$

Taking the normal component of the momentum equation (2.7a) along Γ and using (2.8) gives

$$\left.\frac{\partial p}{\partial \boldsymbol{n}}\right|_{\Gamma} = \left[-\nu \boldsymbol{n} \cdot (\nabla \times \nabla \times \boldsymbol{u}) + \nu \frac{\partial(\nabla \cdot \boldsymbol{u})}{\partial \boldsymbol{n}} + \boldsymbol{n} \cdot \boldsymbol{f} \right]\Bigg|_{\Gamma}.$$

Comparing with (2.7c) we have

$$\left.\frac{\partial(\nabla \cdot \boldsymbol{u})}{\partial \boldsymbol{n}}\right|_{\Gamma} = 0. \tag{2.9}$$

Next, take the divergence of (2.7a) and apply the PPE (2.7b). The result is a heat equation (PDE) for ϕ with boundary condition (2.9) given by

$$\begin{cases} \phi_t = \nu \Delta \phi \quad \text{for } t > 0, \\ \left.\frac{\partial \phi}{\partial \boldsymbol{n}}\right|_{\Gamma} = 0 \quad \text{for } t > 0, \\ \phi(0) = 0, \end{cases} \quad \Rightarrow \quad \phi = \nabla \cdot \boldsymbol{u} \equiv 0 \quad \text{for } t \geqslant 0. \tag{2.10}$$

As noted in Section 1, the pressure boundary condition (2.7c) was first proposed for the projection method [26], and referred to therein as an accurate pressure boundary. The motivation for using (2.7c) in that work was as a means to both enforce a compatibility condition for the discrete linear system for pressure with a Neumann boundary condition, as well as to, in view of (2.8), better enforce incompressibility.

There are significant advantages of (2.7) in terms of numerical implementation when compared to the other PPE formulations. First, both the momentum equation and PPE are endowed with boundary conditions which are straight forward to implement; the no-slip boundary condition (2.7d) is used for the velocity update, and the Neumann boundary condition (2.7c) for the pressure update. Second, any standard spatial discretization such as finite difference, finite element or spectral methods can be applied to (2.7). These implementations are considered in Sections 4–6. In each case we treat the pressure term explicitly in time, decoupling the computation of (2.7a) and (2.7b), which results in a class of extremely efficient NSE solvers. Finally, systematic numerical experiments are presented in Section 4.1 showing that a second order implicit time discretization of the viscous term, with the pressure and convective terms treated explicitly, is stable under the standard CFL condition. Thus, the schemes presented herein are well suited for all Reynolds number flow regimes.

Remark. The PPE formulation (2.7) and the classical formulation of the NSE (1.1) are not equivalent in the case of steady state flows. This is clear, for in (2.10a) the time derivative ϕ_t is now absent, and hence non-zero constant solutions are possible due to the Neumann boundary condition (2.10b). In fact, of the four formulations discussed above it is easily verified that only for formulation (I) does one have equivalence with (1.1) for steady state flows.

2.1. Galerkin formulation of the PPE system

We derive the Galerkin formulation (1.3) by recasting (2.7) in a variational form. Denote an inner product on Ω by $\langle \cdot, \cdot \rangle$, and integration over the boundary Γ by $\langle \cdot, \cdot \rangle_{\Gamma}$. Our variational form of the momentum equation (2.7a) is standard. For all smooth test functions $\boldsymbol{v}$ with $\boldsymbol{v}|_{\Gamma} = 0$ find $\boldsymbol{u}$ such that

$$\langle \boldsymbol{u}_t, \boldsymbol{v}\rangle + \langle \boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{v}\rangle + \langle \nabla p, \boldsymbol{v}\rangle = -\nu\langle \nabla\boldsymbol{u}, \nabla\boldsymbol{v}\rangle + \langle \boldsymbol{f}\ , \boldsymbol{v}\rangle. \tag{2.11}$$

For the PPE (2.7b), first take the inner product of Δp and a smooth test function ψ. Then Green's formula and the boundary condition (2.7c) gives

$$\langle \Delta p, \psi\rangle = -\langle \nabla p, \nabla\psi\rangle - \nu\langle \boldsymbol{n}\cdot(\nabla\times\nabla\times\boldsymbol{u}), \psi\rangle_\Gamma + \langle \boldsymbol{n}\cdot\boldsymbol{f}\ , \psi\rangle_\Gamma.$$

Further applying the vector identity

$$\langle \boldsymbol{n}\cdot(\nabla\times\nabla\times\boldsymbol{u}), \psi\rangle_\Gamma = \langle \nabla\times\nabla\times\boldsymbol{u}, \nabla\psi\rangle = -\langle \nabla\times\boldsymbol{u}, \boldsymbol{n}\times\nabla\psi\rangle_\Gamma \tag{2.12}$$

above gives

$$\langle \Delta p, \psi\rangle = -\langle \nabla p, \nabla\psi\rangle + \nu\langle \nabla\times\boldsymbol{u}, \boldsymbol{n}\times\nabla\psi\rangle_\Gamma + \langle \boldsymbol{n}\cdot\boldsymbol{f}\ , \psi\rangle_\Gamma. \tag{2.13}$$

We note the importance of the identity (2.12) here, for it reduces the regularity requirements on $\boldsymbol{u}$ and allows the use of C^0 elements below in (2.15). Next, take the inner product of the right-hand side of the PPE (2.7b) with ψ and integrate by parts to arrive at (also using (2.7d))

$$\langle -\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) + \nabla\cdot\boldsymbol{f}\ , \psi\rangle = \langle \boldsymbol{u}\cdot\nabla\boldsymbol{u}, \nabla\psi\rangle - \langle \boldsymbol{f}\ , \nabla\psi\rangle + \langle \boldsymbol{n}\cdot\boldsymbol{f}\ , \psi\rangle_\Gamma, \tag{2.14}$$

Combining (2.13) and (2.14), along with (2.11), we have our full variational formulation for (2.7), namely

$$\langle \boldsymbol{u}_t, \boldsymbol{v}\rangle + \langle \boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{v}\rangle + \langle \nabla p, \boldsymbol{v}\rangle = -\nu\langle \nabla\boldsymbol{u}, \nabla\boldsymbol{v}\rangle + \langle \boldsymbol{f}\ , \boldsymbol{v}\rangle, \tag{2.15a}$$

$$\langle \nabla p, \nabla\psi\rangle = -\langle \boldsymbol{u}\cdot\nabla\boldsymbol{u}, \nabla\psi\rangle + \langle \boldsymbol{f}\ , \nabla\psi\rangle + \nu\langle \nabla\times\boldsymbol{u}, \boldsymbol{n}\times\nabla\psi\rangle_\Gamma. \tag{2.15b}$$

In order to make sense of the term $\langle \nabla\times\boldsymbol{u}, \boldsymbol{n}\times\nabla\psi\rangle_\Gamma$ there are two choices:

(i) Require $(\nabla\times\boldsymbol{u})|_\Gamma \in H^{1/2}(\Gamma)$ and $(\boldsymbol{n}\times\nabla\psi)|_\Gamma \in H^{-1/2}(\Gamma)$. This can be realized if $\boldsymbol{u}\in H^2$ and $\psi\in H^1$. With this choice our variational formulation is given by: Find $\boldsymbol{u}\in L^2(0,T;(H_0^1(\Omega)\cap H^2(\Omega))^d$ and $p\in L^2(0,T;H^1(\Omega)/C)$ such that (2.15) is satisfied $\forall\boldsymbol{v}\in(H_0^1(\Omega))^d$ and $\forall\psi\in H^1(\Omega)/C$.

(ii) Require $(\nabla\times\boldsymbol{u})|_\Gamma \in L^2(\Gamma)$ and $(\boldsymbol{n}\times\nabla\psi)|_\Gamma\in L^2(\Gamma)$, which can be realized by imposing these constraint on the velocity and pressure spaces. We thus take

$$X = \{\boldsymbol{u}\in(H_0^1(\Omega))^d, (\nabla\times\boldsymbol{u})|_\Gamma \in L^2(\Gamma)\} \quad \text{and} \quad Y = \{\psi\in H^1(\Omega)/C, (\boldsymbol{n}\times\nabla\psi)|_\Gamma\in L^2(\Gamma)\}.$$

With this choice our variational formulation is given by: Find $\boldsymbol{u}\in L^2(0,T;X)$ and $p\in L^2(0,T;Y)$ such that (2.15) is satisfied $\forall\boldsymbol{v}\in X$ and $\forall\psi\in Y$.

The variational formulation (2.15) is highly advantageous for, in the context of case (ii) above, it can be implemented using standard C^0 finite elements. To explain, assume Ω is a polygonal domain. Let X_h be a standard C^0 finite element space for the approximate velocity $\boldsymbol{u}_h$ and test functions $\boldsymbol{v}_h$ with zero boundary values, and Y_h a standard C^0 finite element space for the pressure p_h and test functions ψ_h. Then $\nabla\times\boldsymbol{u}_h$ and $\boldsymbol{n}\times\nabla\psi_h$ are piecewise polynomials on Γ and hence in $L^2(\Gamma)$. Moreover, $X_h\subset X$ and $Y_h\subset Y$. A finite element implementation is then given by: Find $\boldsymbol{u}_h\in L^2(0,T;X_h)$ and $p_h\in L^2(0,T;Y_h)$ such that $\forall\boldsymbol{v}_h\in X_h$ and $\forall\psi_h\in Y_h$

$$\langle \partial_t\boldsymbol{u}_h, \boldsymbol{v}_h\rangle + \langle \boldsymbol{u}_h\cdot\nabla\boldsymbol{u}_h, \boldsymbol{v}_h\rangle + \langle \nabla p_h, \boldsymbol{v}_h\rangle = -\nu\langle \nabla\boldsymbol{u}_h, \nabla\boldsymbol{v}_h\rangle + \langle \boldsymbol{f}\ , \boldsymbol{v}_h\rangle, \tag{2.16a}$$

$$\langle \nabla p_h, \nabla\psi_h\rangle = -\langle \boldsymbol{u}_h\cdot\nabla\boldsymbol{u}_h, \nabla\psi_h\rangle + \langle \boldsymbol{f}\ , \nabla\psi_h\rangle + \nu\langle \nabla\times\boldsymbol{u}_h, \boldsymbol{n}\times\nabla\psi_h\rangle_\Gamma. \tag{2.16b}$$

Numerical results of a Legendre–Galerkin implementation of (2.16) are presented in Section 5. We note that in [16] a related Galerkin formulation was proposed which can be implemented using C^0 elements after a projection of the discontinuous function $\nabla\cdot\boldsymbol{u}_h$ for the pressure variable update in their scheme.

3. Explicit treatment of the pressure term and stability results

The computational efficiency of the schemes presented here relies on the fact that the pressure term is *always* treated explicitly in time so that the velocity and pressure updates are completely decoupled. Of course, the nonlinear convection term is also treated explicitly in time to avoid the costly solution of a nonlinear system at each time stage. Thus, we treat the convection and pressure terms together in the time discretization. What remains to be decided is the temporal discretization of the viscous term, and in this section we explain why this choice is not affected by the explicit discretization of the pressure in our PPE formulation (2.7).

Whether one treats the viscous term explicitly or implicitly in time depends on the Reynolds number regime of the flow and the required spatial resolution. The choice is determined by (1) the diffusive stability constraint

$$\nu\frac{\Delta t}{\Delta x^2} \leqslant (1/2)^d, \tag{3.1}$$

where Δx is the smallest grid resolution and d is the dimension, (2) the convective stability constraint of the usual CFL type

$$\|\boldsymbol{u}\|_{L^\infty}\frac{\Delta t}{\Delta x} = \mathrm{CFL} \leqslant 1 \tag{3.2}$$

and (3) the stability region(s) of the time discretization(s) used for each term in the momentum equation. If ν is small (large Reynolds number regime) implicit treatment of the viscous term does nothing to stabilize the convection term. Hence, for the large Reynolds number regime an explicit time discretization should be used, but *only if* it is a convectively stable time stepping scheme, such as RK4. This is due to the fact that the stability region of RK4 encompasses a appreciable portion of the imaginary axis. In this case, unless the computation is over resolved, stability is governed by (3.2) rather than (3.1). In contrast, if a low order explicit scheme whose stability region did not encompass *any* portion of the imaginary axis, such as forward Euler or the Midpoint rule, a severe cell Reynolds number constraint would result (cf. [7]).

For large ν (small Reynolds number regime) the time constraint due to (3.1) dominates (3.2). In this case implicit treatment of the viscous term is clearly indicated to avoid (3.1). For example, one possible second order semi-implicit time discretization of (2.7a) (or (1.2)), with Crank–Nicholson applied to the viscous term and Adams–Bashforth discretization for both the convection pressure terms, is given by

$$\begin{aligned}&\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^n}{\Delta t}+\frac{3}{2}(\boldsymbol{u}^n\cdot\nabla\boldsymbol{u}^n+\nabla p^n)-\frac{1}{2}(\boldsymbol{u}^{n-1}\cdot\nabla\boldsymbol{u}^{n-1}+\nabla p^{n-1})=\frac{\nu}{2}\Delta(\boldsymbol{u}^{n+1}+\boldsymbol{u}^n)+\boldsymbol{f}^{n+1/2},\\&\boldsymbol{u}^{n+1}|_\Gamma=0.\end{aligned} \tag{3.3}$$

The PPE (2.7b) is then solved using

$$\begin{aligned}&\Delta p^{n+1}=-\nabla\cdot(\boldsymbol{u}^{n+1}\cdot\nabla\boldsymbol{u}^{n+1})+\nabla\cdot\boldsymbol{f}^{n+1},\\&\left.\frac{\partial p^{n+1}}{\partial\boldsymbol{n}}\right|_\Gamma=\left.\left[-\nu\boldsymbol{n}\cdot(\nabla\times\nabla\times\boldsymbol{u}^{n+1})+\boldsymbol{n}\cdot\boldsymbol{f}^{n+1}\right]\right|_\Gamma,\end{aligned} \tag{3.4}$$

As for the issue of whether or not treating the pressure explicitly imposes any stability constraint on the time step, if the viscous term is also treated explicitly then the stability is determined by both constraints (3.1) and (3.2). If the viscous term is treated implicitly, we show in Sections 3.1 and 3.2 that the standard CFL constraint (3.2) is sufficient for stability for a first or second order semi-implicit time discretization. In particular, we show unconditional stability for the Stokes equation in a flat domain. Numerical experiments

are then reported in Section 4 which show that (3.3) and (3.4) is stable under the standard CFL condition, and $\boldsymbol{u}$, p, $\nabla \cdot \boldsymbol{u} = 0$ and $\omega = \nabla \times \boldsymbol{u}$ all achieve full second order accuracy in time.

Remark. We note that the strategy of treating the convection and pressure terms together explicitly in time (such as in (3.3)) is a common approach found in spectral methods for simulating homogeneous turbulence in a periodic box. For such problems the viscous term is generally treated implicitly using an integrating factor. The analog of (1.2) for this case of periodic boundary conditions is given by

$$\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p = \nu \Delta \boldsymbol{u} + \boldsymbol{f}\ , \tag{3.5a}$$

$$\Delta p = -\nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u}) + \nabla \cdot \boldsymbol{f}. \tag{3.5b}$$

Solving the PPE, i.e. $p = -\Delta_P^{-1} \nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{f}\ )$ where the subscript P denotes the solution with periodic boundary conditions, and then substituting this expression for p into the momentum equation above gives

$$\boldsymbol{u}_t + P(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nu \Delta \boldsymbol{u} + P\boldsymbol{f}.$$

$P = (I - \nabla \Delta_P^{-1} \nabla \cdot)$ is the projection operator associated with the Helmholtz decomposition. Important here is that the incompressibility of the velocity field is accurately enforced [4,5,23,27].

3.1. Analysis of the tangential pressure gradient and tangential viscous forces for the Stokes equations

We consider the behavior of solutions to the Stokes equations in the new pressure Poisson formulation (2.7). Two important observations are: (i) the pressure can be solved for in terms of the tangential velocity, and (ii) the tangential pressure gradient is an *unstable* force, however, it is *dominated* by the tangential viscous term. As a consequence, we show in Section 3.2 that explicit treatment of the pressure is *unconditionally* stable when the viscous term is treated implicitly with a first or second order scheme. These observations shed light as to why handling of the pressure term is a very delicate issue. Indeed, of the pressure Poisson formulations that were discussed in Section 2 we found (2.7) the most advantageous in terms of stability. To explain why, consider the Stokes equation on a domain $\Omega = [-1, 1] \times (0, 2\pi)$ with periodic boundary conditions in y. The pressure Poisson formulation for the NSE (2.7) applied to the Stokes equation in this domain is given by

$$\boldsymbol{u}_t + \nabla p = \nu \Delta \boldsymbol{u}, \tag{3.6a}$$

$$\Delta p = 0, \tag{3.6b}$$

$$\boldsymbol{u}|_{x=-1,1} = 0, p_x + \nu \partial_y (v_x - u_y)|_{x=-1,1} = 0. \tag{3.6c}$$

Since $u|_{x=-1,1} = 0$ and $u_y|_{x=-1,1} = 0$, one has

$$p_x + \nu \partial_y (v_x - u_y)|_{x=-1,1} = \partial_x (p + \nu v_y)|_{x=-1,1} = 0. \tag{3.7}$$

Hence $p + \nu v_y$ is a solution of the homogeneous Neumann problem

$$\Delta(p + \nu v_y) = \nu \Delta \partial_y v, \quad \partial_x (p + \nu v_y)|_{x=-1,1} = 0.$$

Denoting its solution operator by Δ_N^{-1}, we can write the pressure as

$$p = \nu(\Delta_N^{-1} \Delta - I)\partial_y v.$$

Important here is that the solution only involves the tangential velocity v! Substituting this expression for p into the v component of the momentum equation (3.6a) gives

$$v_t - \nu\Delta v + \nu\partial_y(\Delta_N^{-1}\Delta - I)\partial_y v = 0. \tag{3.8}$$

Denoting the pressure operator as $\mathscr{B} = \partial_y(\Delta_N^{-1}\Delta - I)\partial_y$, and let $\langle\cdot,\cdot\rangle$ be the standard inner product. The key observations are:

(i) For any $u, v \in H^2 \cap H_0^1$, we have

$$\langle \Delta u, \mathscr{B} v\rangle = \langle \Delta v, \mathscr{B} u\rangle \tag{3.9a}$$

and

$$\langle \Delta v, \mathscr{B} v\rangle \geqslant 0. \tag{3.9b}$$

(ii) $\mathscr{B}$ is dominated by $-\Delta$ in the sense that

$$\langle \Delta v, \mathscr{B} v\rangle \leqslant \|\Delta v\|^2. \tag{3.10}$$

To verify (3.9a), take $u, v \in H^2 \cap H_0^1$ and let $\psi = \Delta_N^{-1}\Delta u$ and $\phi = \Delta_N^{-1}\Delta v$, i.e. ψ and ϕ solve

$$\Delta\psi = \Delta u, \quad \partial_n\psi|_{x=-1,1} = 0 \tag{3.11a}$$

and

$$\Delta\phi = \Delta v, \quad \partial_n\phi|_{x=-1,1} = 0, \tag{3.11b}$$

respectively. From (3.11b) we have

$$\mathscr{B} v = \partial_y(\Delta_N^{-1}\Delta - I)\partial_y v = \partial_{yy}\phi - \partial_{yy} v,$$

which along with (3.11a) gives

$$\langle \Delta u, \mathscr{B} v\rangle = \langle \Delta u, \partial_{yy}\phi\rangle - \langle \Delta u, \partial_{yy} v\rangle = \langle \Delta\psi, \partial_{yy}\phi\rangle - \langle \Delta u, \partial_{yy} v\rangle.$$

Integrating each term on the right by parts, along with the boundary conditions in (3.11a) and (3.11b), gives

$$\langle \Delta u, \mathscr{B} v\rangle = \langle \nabla\partial_y\psi, \nabla\partial_y\phi\rangle - \langle \nabla\partial_y u, \nabla\partial_y v\rangle. \tag{3.12}$$

Interchanging the arguments on the right-hand side above, along with the symmetry of the inner product, proves (3.9a). Next, from (3.12) one has

$$\langle \Delta v, \mathscr{B} v\rangle = \langle \nabla\partial_y\phi, \nabla\partial_y\phi\rangle - \langle \nabla\partial_y v, \nabla\partial_y v\rangle. \tag{3.13}$$

Integrating by parts and using (3.11b) gives

$$\|\nabla\partial_y v\|^2 = \langle \nabla\partial_y v, \nabla\partial_y v\rangle = -\langle \partial_y\Delta v, \partial_y v\rangle = -\langle \partial_y\Delta\phi, \partial_y v\rangle = \langle \nabla\partial_y\phi, \nabla\partial_y v\rangle.$$

Applying the Hölder inequality to the last term above and dividing by $\|\nabla\partial_y v\|$ shows that $\|\nabla\partial_y v\| \leqslant \|\nabla\partial_y\phi\|$. This together with (3.13) proves (3.9b). For (3.10), note that (3.13) implies

$$\langle \Delta v, \mathscr{B} v\rangle \leqslant \langle \nabla\partial_y\phi, \nabla\partial_y\phi\rangle = \langle \partial_{xy}\phi, \partial_{xy}\phi\rangle + \langle \partial_{yy}\phi, \partial_{yy}\phi\rangle, \tag{3.14}$$

while a direct computation gives

$$\langle \Delta\phi, \Delta\phi\rangle = \langle \partial_{xx}\phi, \partial_{xx}\phi\rangle + 2\langle \partial_{xx}\phi, \partial_{yy}\phi\rangle + \langle \partial_{yy}\phi, \partial_{yy}\phi\rangle = \langle \partial_{xx}\phi, \partial_{xx}\phi\rangle + 2\langle \partial_{xy}\phi, \partial_{xy}\phi\rangle + \langle \partial_{yy}\phi, \partial_{yy}\phi\rangle,$$

since integration by parts gives $\langle \partial_{xx}\phi, \partial_{yy}\phi\rangle = \langle \partial_{xy}\phi, \partial_{xy}\phi\rangle$. Hence, each of the terms above are non-negative, which along with (3.14) and (3.11b) gives

$$\langle \Delta v, \mathscr{B} v\rangle \leqslant \langle \Delta\phi, \Delta\phi\rangle = \langle \Delta v, \Delta v\rangle,$$

proving (3.10).

3.2. Unconditional stability analysis

With the above key observations, we first prove that the following first order semi-implicit scheme for our PPE formulation applied to the Stokes equation, namely

$$\frac{v^{n+1}-v^n}{\Delta t} - \nu\Delta v^{n+1} + \nu\mathscr{B} v^n = 0, \tag{3.15}$$

is unconditionally stable. Indeed, taking the inner product of $-\Delta(v^{n+1}+v^n)$ with each term in (3.15) gives

$$\left\langle -\Delta(v^{n+1}+v^n), \frac{v^{n+1}-v^n}{\Delta t}\right\rangle = \frac{\|\nabla v^{n+1}\|^2 - \|\nabla v^n\|^2}{\Delta t},$$

$$\langle -\Delta(v^{n+1}+v^n), -\Delta v^{n+1}\rangle = \frac{1}{2}(\|\Delta v^{n+1}\|^2 - \|\Delta v^n\|^2) + \frac{1}{2}\|\Delta(v^{n+1}+v^n)\|^2,$$

$$\langle -\Delta(v^{n+1}+v^n), \mathscr{B} v^n\rangle = \frac{1}{2}\langle \Delta(v^{n+1}+v^n), \mathscr{B}(v^{n+1}-v^n)\rangle - \frac{1}{2}\langle \Delta(v^{n+1}+v^n), \mathscr{B}(v^{n+1}+v^n)\rangle.$$

Using (3.9a) one has

$$\langle \Delta(v^{n+1}+v^n), \mathscr{B}(v^{n+1}-v^n)\rangle = \langle \Delta v^{n+1}, \mathscr{B} v^{n+1}\rangle - \langle \Delta v^n, \mathscr{B} v^n\rangle,$$

and from (3.10) we see that

$$\langle \Delta(v^{n+1}+v^n), \mathscr{B}(v^{n+1}+v^n)\rangle \leqslant \|\Delta(v^{n+1}+v^n)\|^2.$$

Thus, we have the energy estimate

$$\frac{\|\nabla v^{n+1}\|^2 - \|\nabla v^n\|^2}{\Delta t} + \frac{\nu}{2}(\|\Delta v^{n+1}\|^2 - \|\Delta v^n\|^2) + \frac{\nu}{2}(\langle \Delta v^{n+1}, \mathscr{B} v^{n+1}\rangle - \langle \Delta v^n, \mathscr{B} v^n\rangle) \leqslant 0,$$

or

$$\|\nabla v^{n+1}\|^2 + \frac{\nu\Delta t}{2}\|\Delta v^{n+1}\|^2 + \frac{\nu\Delta t}{2}\langle \Delta v^{n+1}, \mathscr{B} v^{n+1}\rangle \leqslant \|\nabla v^n\|^2 + \frac{\nu\Delta t}{2}\|\Delta v^n\|^2 + \frac{\nu\Delta t}{2}\langle \Delta v^n, \mathscr{B} v^n\rangle. \tag{3.16}$$

From (3.9b) we know that the last term on each side above is non-negative, which shows that (3.15) is unconditionally stable. For a second order semi-implicit scheme we discretize the pressure term using second order Adams–Bashforth and the viscous term with Crank–Nicholson, which reads

$$\frac{v^{n+1}-v^n}{\Delta t} - \nu\Delta\frac{v^{n+1}+v^n}{2} + \nu\mathscr{B}\frac{3v^n - v^{n-1}}{2} = 0. \tag{3.17}$$

Unfortunately we cannot show stability of (3.17) using energy estimates. We thus resort to normal modes analysis. Setting $v^n = \kappa^n \widehat{v}$, (3.17) becomes

$$2(\kappa^2 - \kappa)\widehat{v} - \nu(\kappa^2 + \kappa)\Delta t \Delta \widehat{v} + \nu(3\kappa - 1)\Delta t \mathscr{B}\widehat{v} = 0.$$

Taking the inner product of the above equation with $-\Delta\widehat{v}$ gives

$$2(\kappa^2 - \kappa)\|\nabla\widehat{v}\|^2 + \nu(\kappa^2 + \kappa)\Delta t\|\Delta\widehat{v}\|^2 - \nu(3\kappa - 1)\Delta t\langle\Delta\widehat{v}, \mathscr{B}\widehat{v}\rangle = 0 \tag{3.18}$$

Now write (3.18) as $a\kappa^2 - b\kappa + c = 0$, with $a = 2\|\nabla\widehat{v}\|^2 + \nu\Delta t\|\Delta\widehat{v}\|^2$, $b = 2\|\nabla\widehat{v}\|^2 - \nu\Delta t\|\Delta\widehat{v}\|^2 + 3\nu\Delta t\langle\Delta\widehat{v}, \mathscr{B}\widehat{v}\rangle$, and $c = \nu\Delta t\langle\Delta\widehat{v}, \mathscr{B}\widehat{v}\rangle$. Using (3.9b) and (3.10) one can easily verify that $0 \leqslant c < a$, $|b| < a + c$. Hence all the eigenvalues κ of the normal modes are strictly in the unit disk [31], indicating that (3.16) is unconditionally stable. Since we have obtained bounds on v^n, and hence on p^n, p^n can be treated as a force term in the u component of the momentum equation (3.6a). We omit the details.

We now have a second order unconditionally stable scheme for Stokes equation (3.6), namely

$$\frac{u^{n+1} - u^n}{\Delta t} + \partial_x \frac{3p^n - p^{n-1}}{2} = \nu\Delta\frac{u^{n+1} + u^n}{2}, \quad u^{n+1}|_{x=-1,1} = 0, \tag{3.19a}$$

$$\frac{v^{n+1} - v^n}{\Delta t} + \partial_y \frac{3p^n - p^{n-1}}{2} = \nu\Delta\frac{v^{n+1} + v^n}{2}, \quad v^{n+1}|_{x=-1,1} = 0, \tag{3.19b}$$

$$\Delta p^{n+1} = 0, \quad \partial_x p^{n+1}|_{x=-1,1} = \nu\partial_y(\partial_x v^{n+1} - \partial_y u^{n+1})|_{x=-1,1}. \tag{3.19c}$$

Normal mode of analysis of (3.19) is presented in Appendix C which shows that second order time accuracy is achieved for $\boldsymbol{u}$, p and $\nabla \cdot \boldsymbol{u}$. Moreover, the solutions are free of any numerical boundary layers.

This second order scheme applied to the pressure Poisson formulation of the full Navier–Stokes equation (2.7) gives (3.3) and (3.4). To reiterate, since the pressure term is treated explicitly in the momentum equation, the computation of (3.3) is decoupled from that of (3.4). The cost then consists of solving a standard Poisson equation, and two or three (2D or 3D) standard Helmholtz equations. Higher order stiffly stable schemes such as explicit–implicit Runge–Kutta methods [29], multi-step Adams–Bashforth/Adams–Moulton methods [21], or exponential time differencing fourth order Runge–Kutta methods [22] could also be used for the time discretization.

4. Chebyshev spectral collocation: implementation and numerical results

We present implementation details of our PPE formulation using Chebyshev spectral collocation coupled with the second order semi-implicit time discretization (3.3) and (3.4). Computational results of a 2D problem, in the form of accuracy checks, are then presented. Moreover, a Δt convergence study clearly indicates full second order accuracy in time, and also that the scheme is stable under the standard CFL constraint (3.2). The MATLAB code used to produce these results can be found in Appendix A. First, in place of the PPE system (1.2) (or (2.7)) we use the equivalent PPE formulation given by

$$\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p = \nu\Delta\boldsymbol{u} + \boldsymbol{f}\ , \tag{4.1a}$$

$$\Delta p = \frac{1}{2}(\nabla \cdot \boldsymbol{u})^2 - (\nabla\boldsymbol{u}) : (\nabla\boldsymbol{u})^{\mathrm{T}} + \nabla \cdot \boldsymbol{f}\ , \tag{4.1b}$$

$$\left.\frac{\partial p}{\partial \boldsymbol{n}}\right|_{\Gamma} = [-\nu\boldsymbol{n} \cdot (\nabla \times \nabla \times \boldsymbol{u}) + \boldsymbol{n} \cdot \boldsymbol{f}]|_{\Gamma}, \tag{4.1c}$$

$$\boldsymbol{u}|_{\Gamma} = 0. \tag{4.1d}$$

The proof of the equivalence of (4.1) and (1.1) is presented in Appendix B. The sole difference between (4.1) and (1.2) is the right-hand side of the PPE (4.1b), where now only first derivatives appear. For the C^∞ functions used in the accuracy checks presented below, no significant differences in the computed results were found between formulations (4.1) and (1.2). This is not to say that (4.1) would not prove useful in computing a flow with less regularity, hence its inclusion here.

Since we are considering here 2D, for simplicity take $\Omega = [-1,1]^2 \subset R^2$. Recall for $N > 0$ the Chebyshev points are given by

$$x_i = \cos((i-1)\pi/N), \quad i = 1,2,\ldots,N+1, \tag{4.2}$$

which defines a discrete grid I_h for $I = [-1,1]$, and let $\mathscr{D}_N$ denote the $(N+1)\times(N+1)$ Chebyshev differentiation matrix(cf. [2,28,34]). A discrete grid for $\Omega = [-1,1]^2 = I \times I$ is given by $\Omega_h = I_h \times I_h$. Then for a function $a(x,y)$ and the $(N+1)\times(N+1)$ matrix A with $A_{i,j} = a(x_i,y_j)$, we have

$$\frac{\partial a}{\partial x}(x_i,y_j) \approx (\mathscr{D}_N A)_{i,j}, \quad \frac{\partial a}{\partial y}(x_i,y_j) \approx (A\mathscr{D}_N^{\mathrm{T}})_{i,j}, \quad \frac{\partial^2 a}{\partial x^2}(x_i,y_j) \approx (\mathscr{D}_N^2 A)_{i,j}, \quad \text{for } i,j = 1,2,\ldots,N+1,$$

and the approximations achieve spectral accuracy with increasing N if $a(x,y)$ is a sufficiently smooth function.

To facilitate the discussion below we set some notation. Unless otherwise noted, all matrices are assumed to be $(N+1)\times(N+1)$. $\mathscr{I}$ is the identity matrix, the discrete Laplacian is given by $\Delta_N A = \mathscr{D}_N^2 A + A(\mathscr{D}_N^2)^{\mathrm{T}}$, and $A^{(n+\frac{1}{2})} = \frac{3}{2}A^n - \frac{1}{2}A^{n-1}$, where the superscripts on the left refer to the n and $n-1$ time step, thus $A^{(n+\frac{1}{2})}$ is distinguished from $A^{n+\frac{1}{2}}$. Furthermore, motivated by MATLAB matrix operations syntax, $A.^*B$ denotes element-wise multiplication, i.e., $(A.^*B)_{i,j} = A_{i,j}\cdot B_{i,j}$, and $A.\hat{}\,2 = A.^*A$. Finally, the boundary points of Ω_h are denoted by $\Gamma_N = \{(i,j) : i \text{ or } j \text{ equal 1 or } N+1\}$, with $\Gamma_N^C \subset \Gamma_N$ representing the four corner points.

Given a Δt, let (U^n, V^n, P^n) and $(U^{n-1}, V^{n-1}, P^{n-1})$ denote discrete solutions on Ω_h at times $n\Delta t$ and $(n-1)\Delta t$, respectively. Discretizing (4.1a) using (3.3) results in two Helmholtz equations for the velocity components U^{n+1} and V^{n+1}, which for U^{n+1} reads

$$\frac{U^{n+1}-U^n}{\Delta t} + \left(U^n.^*(\mathscr{D}_N U^n) + V^n.^*(U^n\mathscr{D}_N^{\mathrm{T}}) + \mathscr{D}_N P\right)^{(n+\frac{1}{2})} = \frac{\nu}{2}\Delta_N(U^{n+1}+U^n) + F_1^{n+\frac{1}{2}},$$

where $F = (F_1,F_2)^{\mathrm{T}}$ represents the force term $\boldsymbol{f} = (f_1,f_2)^{\mathrm{T}}$ evaluated on Ω_h. Along with (4.1d), the linear system for U^{n+1} is given by

$$\left(\mathscr{I} - \frac{\nu\Delta t}{2}\Delta_N\right)U^{n+1} = U^n - \Delta t\left[\left(U.^*(\mathscr{D}_N U) + V.^*(U\mathscr{D}_N^{\mathrm{T}}) + \mathscr{D}_N P\right)^{(n+\frac{1}{2})} - \frac{\nu}{2}\Delta_N U^n - F_1^{n+\frac{1}{2}}\right], \tag{4.3a}$$

$$U^{n+1}\big|_{\Gamma_N} = 0. \tag{4.3b}$$

As is, system (4.3) is overdetermined since (4.3a) and (4.3b) represent $(N+1)^2$ and $4N$ equations, respectively, for the $(N+1)^2$ unknowns U^{n+1}. However, note that since collocation is based on representation in physical space we use the discrete homogeneous boundary condition (4.3b) to define for U^{n+1}on Γ_N, with (4.3a) used to determine U^{n+1} at the $(N-1)^2$ interior grid points of $\widetilde{\Omega}_h = \Omega_h/\Gamma_N$. Denote by G the right-hand side of (4.3a), and let $\widetilde{U} = U^{n+1}(2:N,2:N)$, with the analogous definitions for $\widetilde{G}$ and $\widetilde{\mathscr{D}}_N^2$ (each of which are $(N-1)\times(N-1)$). Then (4.3a) for $\widetilde{\Omega}_h$ reads

$$\widetilde{U} - \frac{\nu\Delta t}{2}\left(\widetilde{\mathscr{D}}_N^2\widetilde{U} + \widetilde{U}(\widetilde{\mathscr{D}}_N^2)^{\mathrm{T}}\right) = \widetilde{G}, \tag{4.4}$$

which can be solved efficiently using the matrix-diagonalization procedure [11,28]. To briefly explain, $\widetilde{\mathscr{D}}_N^2$ is diagonalizable hence $\widetilde{\mathscr{D}}_N^2 = \mathscr{Q}\Lambda\mathscr{Q}^{-1}$, where $\Lambda = \mathrm{diag}\{\lambda_1, \ldots, \lambda_{N-1}\}$, the λ_i are the eigenvalues of $\widetilde{\mathscr{D}}_N^2$, and the columns of $\mathscr{Q}$ the corresponding eigenvectors. Defining $\widehat{U} = \mathscr{Q}^{-1}\widetilde{U}(\mathscr{Q}^{\mathrm{T}})^{-1}$ and $\widehat{G} = \mathscr{Q}^{-1}\widetilde{G}(\mathscr{Q}^{\mathrm{T}})^{-1}$, (4.4) is equivalent to

$$\widehat{U} - \frac{v\Delta t}{2}\left(\Lambda\widehat{U} + \widehat{U}\Lambda\right) = \widehat{G} \quad \Rightarrow \quad \widehat{U}_{i,j} = \frac{\widehat{G}_{i,j}}{1 - (v\Delta t/2)(\lambda_i + \lambda_j)}, \quad i,j = 1,2,\ldots,N-1. \tag{4.5}$$

Then $\widetilde{U} = \mathscr{Q}\widehat{U}\mathscr{Q}^{\mathrm{T}}$ is recovered from the computation of $\widehat{U}$ above, and along Γ_N (4.3b) is applied. This completes the computation of U^{n+1}. A system analogous to (4.3) is similarly derived for V^{n+1} (we omit the details) and solved using the same procedure.

To compute P^{n+1}, first discretize the PPE (4.1b) by

$$\begin{aligned}\Delta_N P^{n+1} = {} & \frac{1}{2}(\mathscr{D}_N U^{n+1} + V^{n+1}\mathscr{D}_N^{\mathrm{T}}).\hat{}\,2 + \mathscr{D}_N F_1 + F_2\mathscr{D}_N^{\mathrm{T}} - \left[(\mathscr{D}_N U^{n+1}).\hat{}\,2 + 2(\mathscr{D}_N V^{n+1}).^*(U^{n+1}\mathscr{D}_N^{\mathrm{T}})\right. \\ & \left. + (V^{n+1}\mathscr{D}_N^{\mathrm{T}}).\hat{}\,2\right].\end{aligned} \tag{4.6}$$

Note that (4.6) represents $(N+1)^2$ equations for the $(N+1)^2$ unknowns P^{n+1}, and we have yet to enforce the PPE boundary condition (4.1c). To do so, analogous to our approach for (4.3), first apply (4.6) only at the $(N-1)^2$ interior grid points $\widetilde{\Omega}_h = \Omega_h/\Gamma_N$. The result is $(N-1)^2$ linear equations. However, due to the global stencil of the differentiation matrices of Δ_N these equations involve the $4(N-1)$ boundary points in Γ_N/Γ_N^C. We remark that this was also the case when computing U^{n+1} on $\widetilde{\Omega}_h$, but since (4.3b) is a homogeneous Dirichlet boundary condition it was transparent in going from (4.3a) to (4.4).

Expressions for these $4(N-1)$ boundary values are derived using (4.1c), thus enforcing the Neumann boundary condition for the PPE. To explain, first consider only $2(N-1)$ of them, namely those along $x = \pm 1$ excluding the corners, i.e., (x_i, y_j) with $i = 1$ or $i = N+1$ and $j = 2,3,\ldots,N$. Next, a discretization of the x-component of the right-hand side of (4.1c) is given by $BC = -v(\mathscr{D}_N V^{n+1} - U^{n+1}\mathscr{D}_N^{\mathrm{T}})\mathscr{D}_N^{\mathrm{T}} + F_1$. Then a discrete application of (4.1c) along $x = \pm 1$ reads

$$\sum_{i=1}^{N+1} d_{1,i}P_{i,j}^{n+1} = BC_{1,j} \quad \text{and} \quad \sum_{i=1}^{N+1} d_{N+1,i}P_{i,j}^{n+1} = BC_{N+1,j}, \quad \text{for} \quad j = 2,3\ldots,N, \tag{4.7}$$

where $d_{i,j} = (\mathscr{D}_N)_{i,j}$. For each j (4.7) is a nonsingular 2×2 linear system for $P_{1,j}^{n+1}$ and $P_{N+1,j}^{n+1}$ in terms of $BC_{1,j}$, $BC_{N+1,j}$ and $\{P_{i,j}^{n+1}\}_{i=2}^{N}$, from which the $2(N-1)$ required expressions are derived. This procedure is then repeated by applying (4.1c) along $y = \pm 1$ (again excluding the corners), and with the obvious modifications, for the remaining $2(N-1)$ expressions for $P_{i,1}^{n+1}$ and $P_{i,N+1}^{n+1}$ for $i = 2,3,\ldots,N$.

A linear system for $\widetilde{P} = P^{n+1}(2:N, 2:N)$ is now given by

$$\overline{\mathscr{D}}_N^2\widetilde{P} + \widetilde{P}(\overline{\mathscr{D}}_N^2)^{\mathrm{T}} = \overline{H}, \tag{4.8}$$

where $\overline{\mathscr{D}}_N^2$ and $\overline{H}$ are, respectively, the interior elements $(i,j = 2,3,\ldots,N)$ of $\mathscr{D}_N^2$ and the right-hand side of (4.6), with each modified to account for the expressions for P^{n+1} on Γ_N/Γ_N^C derived above(see [28] for details). $\overline{\mathscr{D}}_N^2$ is diagonalizable, and hence (4.8) can also be solved using the matrix-diagonalization procedure. Thus, let $\overline{\mathscr{D}}_N^2 = \mathscr{R}\Sigma\mathscr{R}^{-1}$ where $\Sigma = \mathrm{diag}\{\sigma_1,\ldots,\sigma_{N-1}\}$, and let $\widehat{P} = \mathscr{R}^{-1}\widetilde{P}(\mathscr{R}^{\mathrm{T}})^{-1}$ and $\widehat{H} = \mathscr{R}^{-1}\overline{H}(\mathscr{R}^{\mathrm{T}})^{-1}$. Note that $\overline{\mathscr{D}}_N^2$ is singular with a one-dimensional null space [28], thus we assume $\sigma_1 = 0$.

This simply reflects the fact that solutions of the Neumann problem are unique up to a constant. Then a particular solution of (4.8) is given by

$$\Sigma\widehat{P} + \widehat{P}\Sigma = \widehat{H} \quad \Rightarrow \quad \widehat{P}_{i,j} = \begin{cases} 0 & i = j = 1, \\ \widehat{H}_{i,j}/(\sigma_i + \sigma_j) & \text{otherwise}, \end{cases} \tag{4.9}$$

where the constant 0 is arbitrary, and we have $P^{n+1}(2:N, 2:N) = \widetilde{P} = \mathscr{R}\widehat{P}\mathscr{R}^{\mathrm{T}}$. The values of P^{n+1} on Γ_N/Γ_N^C is recovered from the expressions for them used in deriving (4.8). Finally, note that (4.7) also holds for $j = 1, N+1$, and hence P^{n+1} can be recovered at the four corner points Γ_N^C by solving two 2×2 linear systems derived from these expressions.

Remark. In the algorithm above the values of the velocity field and pressure at the corner points Γ_N^C of Ω_h never enter the computation. Nonetheless, we recover the pressure at these points since it is often a quantity of interest in applications.

4.1. Numerical results: Chebyshev collocation

In this section we report on numerical results of the Chebyshev collocation method outlined above. A spatial accuracy check and temporal convergence study are presented. Assume exact solutions of (4.1) given by

$$\begin{aligned} u(x,y,t) &= \cos t \cos^2(\pi x/2)\cos(\pi y/2)\sin(\pi y/2), \\ v(x,y,t) &= -\cos t \cos(\pi x/2)\sin(\pi x/2)\cos^2(\pi y/2), \\ p(x,y,t) &= \cos t \cos(\pi x/2)\sin(\pi y/2) \end{aligned} \tag{4.10}$$

with the force term $\boldsymbol{f}$ (time dependent) chosen to ensure that (4.10) is a solution of (4.1). For $\nu = 1$ we computed solutions with grid sizes $N = 8{:}4{:}36$ until a final time of $T = 2$. The time step Δt was determined from (3.2) with CFL $= 0.75$, $\|\boldsymbol{u}\|_{L^\infty} = 1$, and $\Delta x = x_1 - x_2 = 1 - \cos(\pi/N)$.

In Fig. 1 is shown the L^2 and L^∞ norm of the errors between the computed solutions and the exact solution (4.10) evaluated on Ω_h. First, the results provide clear evidence that the scheme is stable with respect to the CFL condition (3.2). In particular, the convective time constraint (3.2) with $N = 36$ gives $\Delta t \approx 2.85 \times 10^{-3}$, while the diffusive time constraint (3.1), which due to the excellent stability properties of the scheme we do not have to satisfy, would have required $\Delta t \approx 3.62 \times 10^{-6}$. Second, while spectral spatial accuracy is not immediately apparent as N is increased, recall that the time discretization is only second order accurate, and is thus the limiting factor in these results. To verify this we recomputed the results for each N with a fixed $\Delta t = 5 \times 10^{-6}$. The results are shown in Fig. 2, where the spatial errors are seen to quickly saturate at the level of the time discretization error for the chosen Δt.

Next, in order to verify the temporal accuracy a Δt convergence study with fixed spatial resolution, $N = 36$, was performed. This value of N is large enough to ensure that the spatial errors are on the order of roundoff error in MATLAB. The results, shown in Fig. 3, clearly indicate that full second order time accuracy is achieved for $\boldsymbol{u}$ and p, as well as the vorticity and divergence, in both the L^2 and L^∞ norms. Additionally, the time history of the divergence error in the L^∞ norm with $\Delta t = 10^{-3}$ and $N = 36$ is shown in Fig. 4. The final time is $T = 50$, demonstrating the ability of the method to accurately enforce $\nabla \cdot \boldsymbol{u} = 0$ in long time simulations.

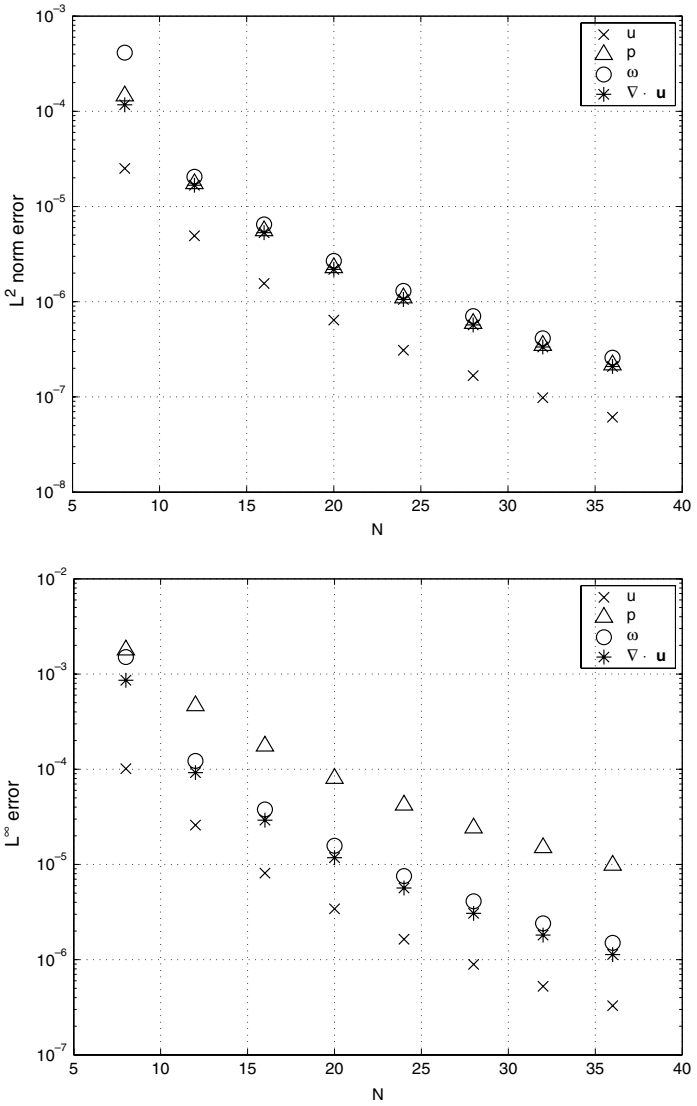

Fig. 1. L^2 and L^∞ norm errors at $T = 2$ of collocation method applied to (4.10). Parameters: $\nu = 1$, CFL $= 0.75$.

Finally, a plot of the pressure error with $N = 36$ and with Δt determined from (3.2) with CFL $= 0.75$ is shown in Fig. 5. Note the absence of any numerical boundary layers.

5. Galerkin formulation: spectral implementation and numerical results

To validate the variational formulation (2.15) (or (1.3)) we present accuracy checks for a 2D test problem using a Legendre–Galerkin spectral method. Since implicit time discretization was investigated for

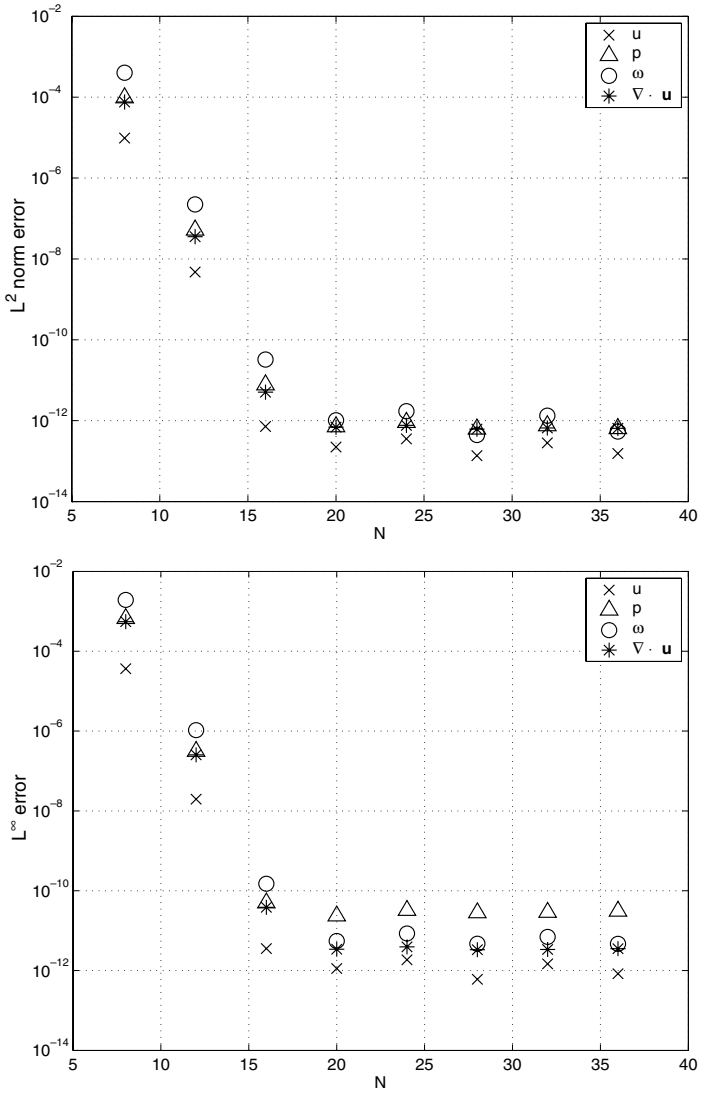

Fig. 2. L^2 and L^∞ norm of error at $T = 2$ of collocation method applied to (4.10) with $\Delta t = 5 \times 10^{-6}$.

the collocation method in Section 4, here we focus on an explicit time discretization of (2.15) and consider the moderate to large Reynolds number flow regime. As discussed in Section 3, in order to avoid a cell Reynolds number constraint we use the convectively stable classical fourth order Runge–Kutta (RK4) method. We emphasize that the intended application of the variational formulation (2.15) is for finite element methods for flows in general domains, which we leave to future works. Since our purpose here is solely to validate the variational formulation we only briefly describe the numerical implementation. In particular, the linear systems which arise are solved using standard methods. The interested reader is referred to [30] where efficient methods for solving linear systems arising from Legendre–Galerkin formulations are discussed.

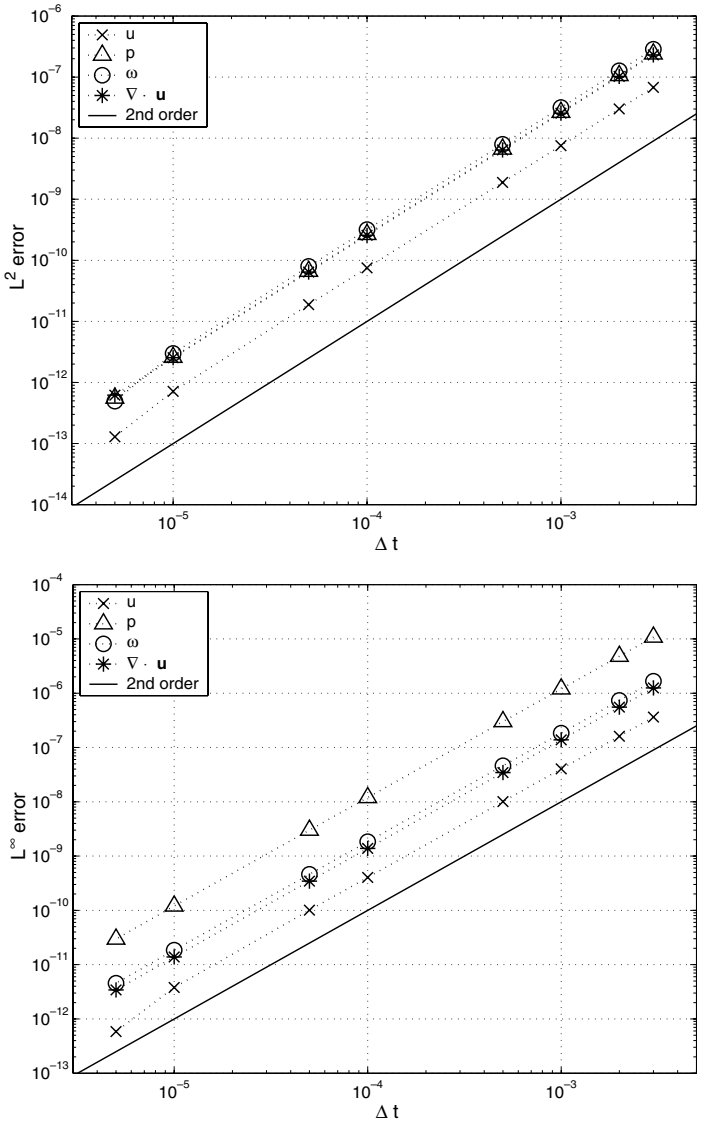

Fig. 3. Time accuracy in both the L^2 and L^∞ norms of collocation method applied to (4.10) with $N = 36$ and $v = 1$. Final time is $T = 2$. The solid line represents second order accuracy.

Again for simplicity take $\Omega = [-1, 1]^2 \subset R^2$. For $n \geqslant 0$ denote by $L_n(x)$ the nth degree Legendre polynomial defined on $[-1, 1]$ and recall the orthogonality relation

$$(L_i(x), L_j(x)) = \frac{2}{2i+1}\delta_{i,j} \quad \forall i, j \geqslant 0, \tag{5.1}$$

where $(f, g) = \int_{-1}^{1} f(x)g(x)\,dx$. Also recall that $L_n'(x)$ satisfies the recurrence relation

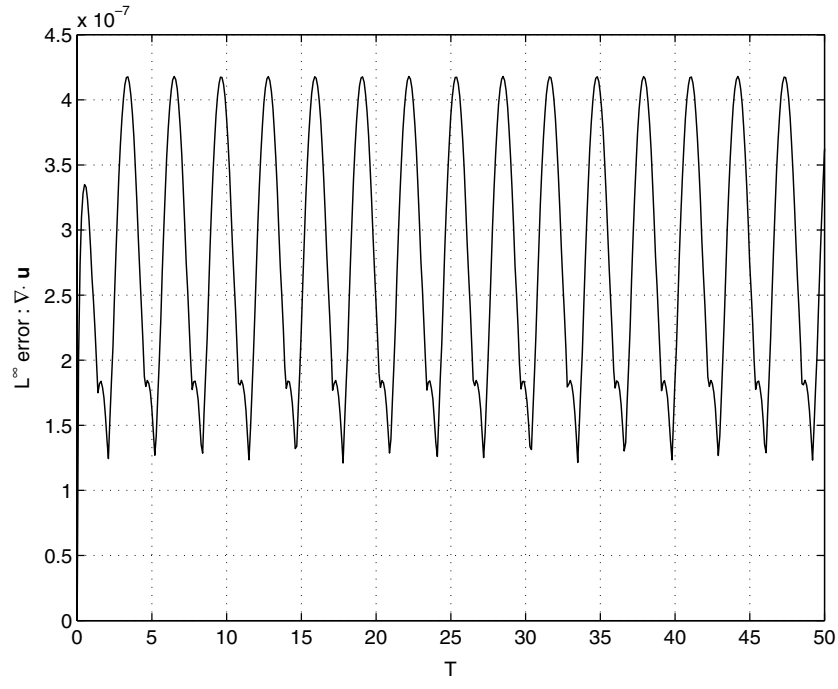

Fig. 4. Divergence error in the L^{∞} norm for the collocation method: $N = 36$, $v = 1$ and $\Delta t = 10^{-3}$.

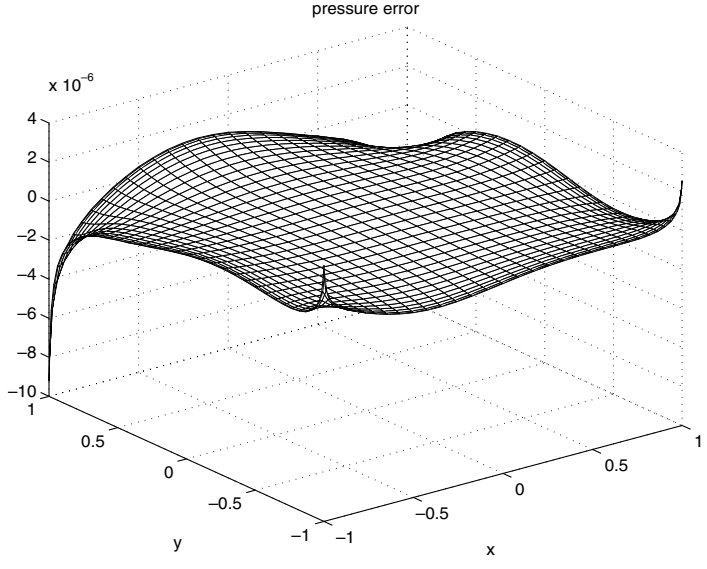

Fig. 5. Pressure error with $N = 36$ for the collocation method at $T = 2$.

$$L_n'(x) = \sum_{\substack{k=0 \\ k+n \text{ odd}}}^{n-1} (2k+1)L_k(x). \tag{5.2}$$

Next, for $n \geqslant 0$ define

$$\Phi_n(x) = \begin{cases} L_{n+2}(x) - L_0(x), & n \text{ even}, \\ L_{n+2}(x) - L_1(x), & n \text{ odd}. \end{cases} \tag{5.3}$$

Each $\Phi_n(x)$ is a polynomial of degree n and $\Phi_n(\pm 1) = 0$.

Let $N \geqslant 2$. The approximation spaces X_h and Y_h for the Galerkin formulation (2.15) (or (2.16)) are taken to be

$$X_h = \operatorname{span}\{\Phi_i(x)\Phi_j(y) : i,j = 0,1,\ldots,N-2\}, \quad Y_h = \operatorname{span}\{L_i(x)L_j(y) : i,j = 0,1,\ldots,N\}, \tag{5.4}$$

and note that $X_h \subset Y_h$, $\dim(X_h) = (N-1)^2$ and $\dim(Y_h) = (N+1)^2$.

Then given Δt, let $(U^n, V^n, P^n) \in X_h \times X_h \times Y_h$ be a discrete solution at time $t^n = n\Delta t$. Thus

$$U^n = \sum_{i,j=0}^{N-2} \widetilde{u}_{ij}^n \Phi_i(x)\Phi_j(y), \quad V^n = \sum_{i,j=0}^{N-2} \widetilde{v}_{ij}^n \Phi_i(x)\Phi_j(y), \quad P^n = \sum_{i,j=0}^{N} \widetilde{p}_{ij}^n L_i(x)L_j(y), \tag{5.5}$$

where $\widetilde{u}_{ij}^n$, $\widetilde{v}_{ij}^n$ and $\widetilde{p}_{ij}^n$ are uniquely determined by (5.1) and (5.3) applied in each coordinate direction. Denote the spectral coefficient matrices for U^n, V^n and P^n by, respectively, $(\widetilde{U}^n)_{ij} = \widetilde{u}_{ij}^n$, $(\widetilde{V}^n)_{ij} = \widetilde{v}_{ij}^n$ and $(\widetilde{P}^n)_{ij} = \widetilde{p}_{ij}^n$.

As noted above, we treat the viscous term explicitly and use RK4 for the time discretization. Since each time stage of RK4 is essentially a forward Euler step we use the latter for illustration. Recall that $\langle \cdot, \cdot \rangle$ denotes integration over Ω and $\langle \cdot, \cdot \rangle_\Gamma$ integration along the boundary Γ. Then (2.15a) for U^{n+1} reads: $\forall \phi \in X_h$ find U^{n+1} such that

$$\left\langle \frac{U^{n+1} - U^n}{\Delta t}, \phi \right\rangle + \langle U^n \partial_x U^n + V^n \partial_y U^n, \phi \rangle + \langle \partial_x P^n, \phi \rangle = -\nu \langle \nabla U^n, \nabla \phi \rangle + \langle F_1^n, \phi \rangle,$$

where $(F_1^n, F_2^n)^{\mathrm{T}}$ represents the force vector $\boldsymbol{f}$ at time t^n. Solving for U^{n+1} gives: $\forall \phi \in X_h$ find U^{n+1} such that

$$\langle U^{n+1}, \phi \rangle = \langle U^n, \phi \rangle - \Delta t \big[\langle U^n \partial_x U^n + V^n \partial_y U^n + \partial_x P^n - F_1^n, \phi \rangle + \nu \langle \nabla U^n, \nabla \phi \rangle \big]. \tag{5.6}$$

The right-hand side above can be evaluated using (5.1)–(5.3). Hence (5.6) is a linear system for the $(N-1)^2$ spectral coefficients $\widetilde{U}^{n+1}$ of U^{n+1} whose coefficient matrix (mass matrix) is invertible. A system analogous to (5.6) for V^{n+1} is similarly derived and solved for $\widetilde{V}^{n+1}$.

Next, (2.15b) for P^{n+1} reads: $\forall \psi \in Y_h$ find P^{n+1} such that

$$\begin{aligned} \langle \nabla P^{n+1}, \nabla \psi \rangle = &- \big[\langle U^{n+1} \partial_x U^{n+1} + V^{n+1} \partial_y U^{n+1}, \partial_x \psi \rangle + \langle U^{n+1} \partial_x V^{n+1} + V^{n+1} \partial_y V^{n+1}, \partial_y \psi \rangle \big] \\ &+ \langle F_1^{n+1}, \partial_x \psi \rangle + \langle F_2^{n+1}, \partial_y \psi \rangle + \nu \langle \partial_x V^{n+1} - \partial_y U^{n+1}, \partial \psi / \partial \boldsymbol{\tau} \rangle_\Gamma, \end{aligned} \tag{5.7}$$

where $\boldsymbol{\tau} = \boldsymbol{n}^\perp$ and $\boldsymbol{n}$ is the unit outward normal along Γ. Since U^{n+1} and V^{n+1} are known, (5.7) is a linear system for the $(N+1)^2$ spectral coefficients $\widetilde{P}^{n+1}$ of P^{n+1}. However, the coefficient matrix (stiffness matrix) is singular with a one dimensional kernel, reflecting the fact that the pressure, being the solution of a Neumann problem, is unique up to an additive constant. Thus we choose a particular solution by setting $\widetilde{p}_{00}^{n+1} = 0$.

Of course for small Reynolds number flow we would treat the viscous term implicitly. For example a second order scheme, following (3.3) and (3.4), is given by: $\forall \phi \in X_h$ find $U^{n+1} \in X_h$ such that

$$\left\langle \frac{U^{n+1}-U^n}{\Delta t}, \phi \right\rangle + \frac{3}{2}\left[\langle U^n \partial_x U^n + V^n \partial_y U^n, \phi\rangle + \langle \partial_x P^n, \phi\rangle\right] - \frac{1}{2}\left[\langle U^{n-1}\partial_x U^{n-1} + V^{n-1}\partial_y U^{n-1}, \phi\rangle + \langle \partial_x P^{n-1}, \phi\rangle\right] = -\frac{v}{2}\langle \nabla(U^{n+1}+U^n), \nabla\phi\rangle + \langle F_1^{n+1/2}, \phi\rangle,$$

along with an analogous formula for V^{n+1} and (5.7).

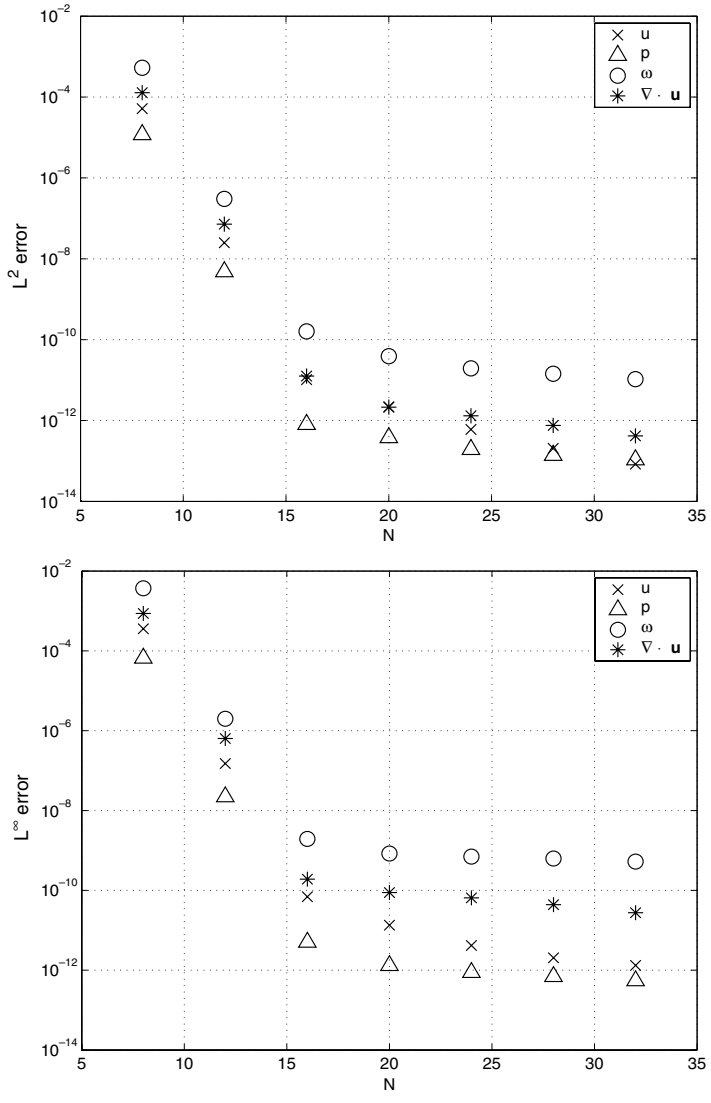

Fig. 6. L^2 and L^∞ norm errors at $T = 2$ of Legendre–Galerkin error applied to (4.10). Parameters: $v = 0.001$, CFL $= 0.75$.

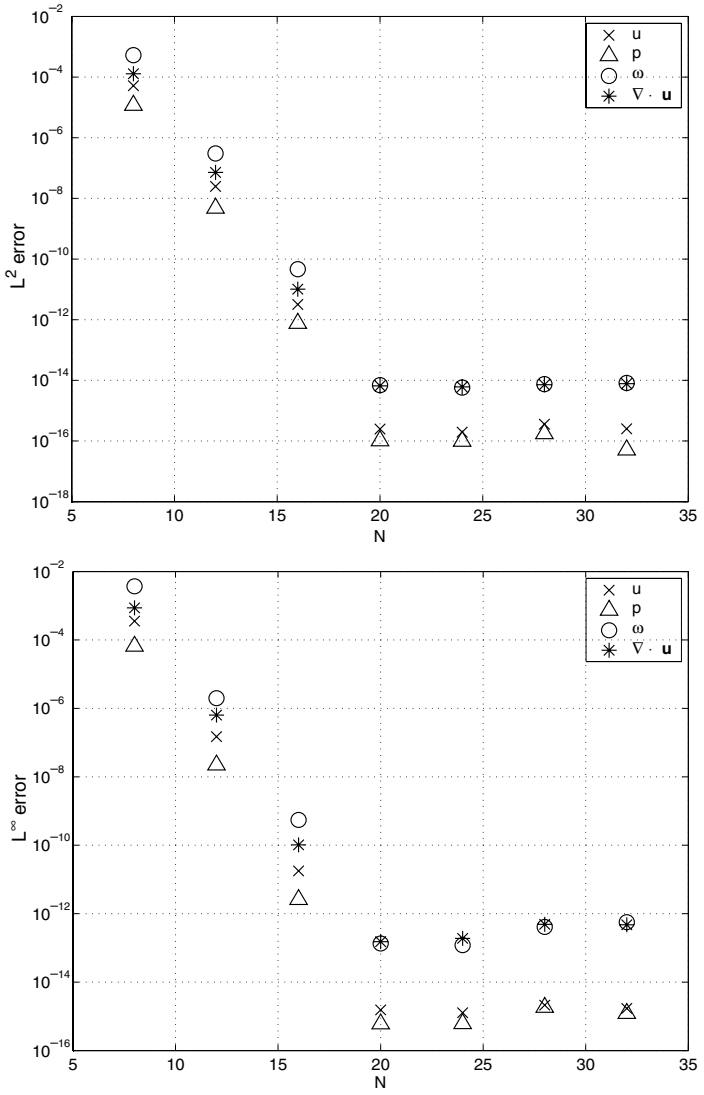

Fig. 7. L^2 and L^∞ norm of error at $T = 2$ of Legendre–Galerkin method applied to (4.10) with $\Delta t = 5 \times 10^{-4}$.

5.1. Numerical Results: Legendre–Galerkin

We report numerical results of the Legendre–Galerkin method described above with explicit fourth order Runge–Kutta for the time discretization. Both a spatial accuracy check and temporal convergence study are presented. Again assume an exact solutions of the NSE given by (4.10). With $v = 0.001$ we computed solutions for grid sizes $N = 8{:}4{:}32$ until a final time of $T = 2$. The time step Δt was determined from the minimum of (3.1) and (3.2) with CFL $= 0.75$, $\|\boldsymbol{u}\|_{L^\infty} = 1$, and Δx the distance between the two closest Gauss–Lobatto points for the given N.

In Fig. 6 is shown the L^2 and L^∞ norm of the errors between the computed solutions and the exact solution (4.10). Note that spectral spatial accuracy is more apparent here as N is increased then was the

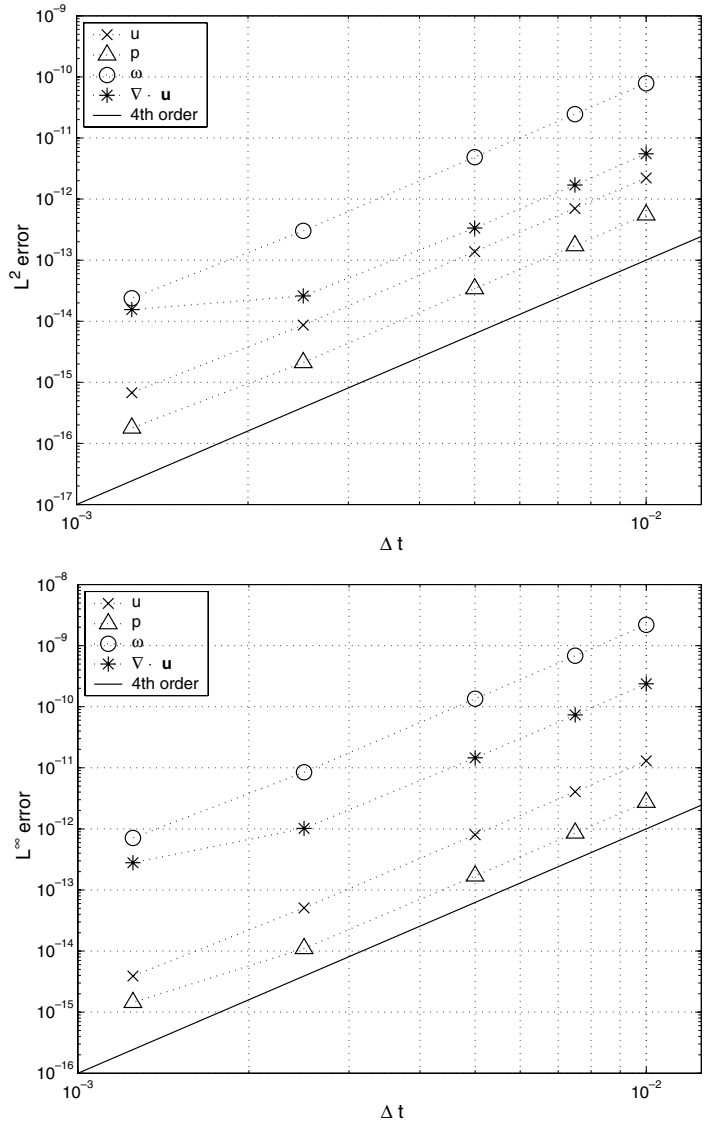

Fig. 8. Time accuracy in both the L^2 and L^∞ norms of Legendre–Galerkin method applied to (4.10) with $N = 20$ and $v = 0.001$. Final time is $T = 2$. The solid line represents fourth order accuracy.

case for the collocation results in Section 4.1 since now the time discretization is fourth order accurate. Moreover, for large N the timestep Δt is dictated by the diffusive time constraint (3.2). To verify this conclusion we recomputed the results for each N with a fixed $\Delta t = 5 \times 10^{-4}$. The results are shown in Fig. 7, where the spatial errors are seen to quickly saturate to the accuracy of double precision roundoff.

As discussed in Section 3, for the case of explicit time discretization one expects the computed solutions to achieve the full temporal accuracy of the scheme, which here is fourth order. To verify this a Δt convergence study with fixed spatial resolution, $N = 20$, was performed. This value of N is large enough to

ensure that the spatial errors are on the order of roundoff error in MATLAB while at the same time allow for a variation in Δt before roundoff is reached. The results, shown in Fig. 8, clearly indicate that full fourth order time accuracy is achieved for $\boldsymbol{u}$ and p, as well as the vorticity and divergence, in both the L^2 and L^∞ norms. It is only at the smallest Δt that these results deviate from fourth order accuracy. However, this can be attributed to saturation of the errors at the level of roundoff. Additionally, the time history of the di-

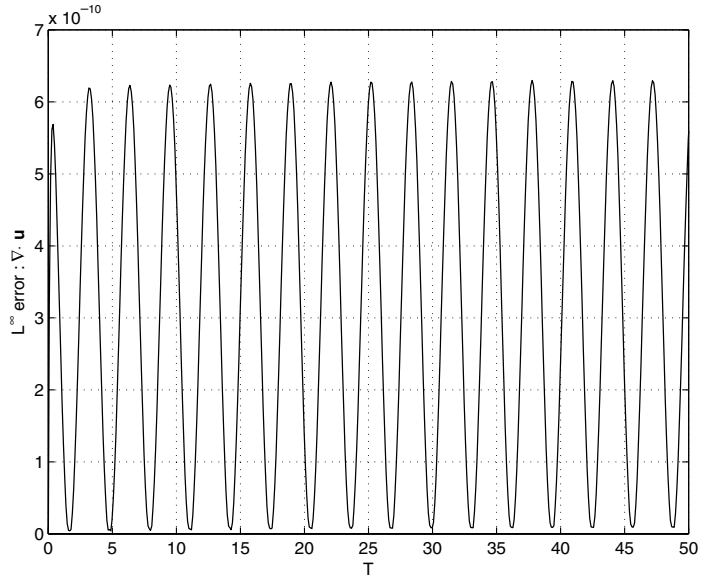

Fig. 9. Divergence error in the L^∞ norm for the Legendre–Galerkin method: $N = 20$, $\nu = 0.001$ and $\Delta t = 10^{-2}$.

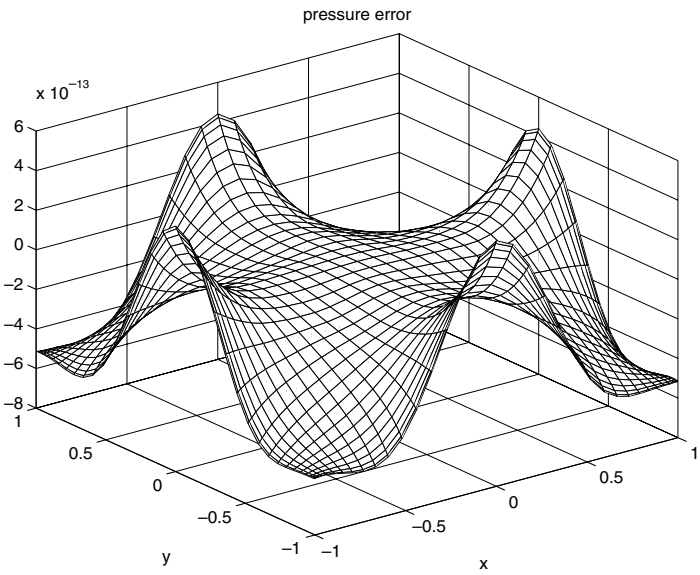

Fig. 10. Pressure error with $N = 32$ for the Legendre–Galerkin method at $T = 2$.

vergence error in the L^∞ norm with $\Delta t = 10^{-2}$ and $N = 20$ is shown in Fig. 9. The final time is $T = 50$, demonstrating the ability of the method to accurately enforce $\nabla \cdot \boldsymbol{u} = 0$ in long time simulations.

Finally, a plot of the pressure error with $N = 32$ is shown in Fig. 10. Note the clear absence of any numerical boundary layers.

6. Finite difference implementation for flow past a cylinder

In this section we revisit an example from [20], and present results of a spatially second order finite difference implementation of (1.2) applied to the computation of the impulsively started flow past a unit ($R = 1$) circular cylinder in the infinite plane. The geometry of the problem naturally dictates the use of the polar PPE formulation of the NSE, to which the transformation $z = \log r$ is applied resulting in an exponentially stretched grid radially. The governing equations, written using radially scaled velocity components defined by

$$\boldsymbol{u} = (u, v) = r(U, V) = e^z(U, V), \tag{6.1}$$

where U and V are the radial and tangential polar form velocities, respectively, are given by

$$\begin{bmatrix} u \\ v \end{bmatrix}_t + \frac{1}{e^{2z}} \begin{bmatrix} uu_z + vu_\theta - (u^2 + v^2) \\ uv_z + vv_\theta \end{bmatrix} + \begin{bmatrix} p_z \\ p_\theta \end{bmatrix} = \frac{\nu}{e^{2z}} \begin{bmatrix} \Delta_{(z,\theta)} u - 2(u_z + v_\theta) \\ \Delta_{(z,\theta)} v - 2(v_z - u_\theta) \end{bmatrix} \tag{6.2a–b}$$

and

$$\Delta_{(z,\theta)} p = -\frac{2}{e^{2z}} \left(u^2 + v^2 + uv_\theta + u_\theta v_z - u_z v_\theta - u_\theta v - v v_z - u u_z \right). \tag{6.3}$$

The incompressibility condition in the (z, θ) variables is given by $\nabla_{(z,\theta)} \cdot \boldsymbol{u} = (u_z + v_\theta) = 0$, and thus we set the last term in (6.2a) equals to zero. At the cylinder surface Γ the no-slip boundary condition $\boldsymbol{u}|_\Gamma = 0$ is enforced for the velocity field. In this geometry the Neumann pressure boundary condition (1.2c) is given by

$$p_z|_\Gamma = -\nu(v_{\theta z} - u_{\theta\theta})|_\Gamma = -\nu v_{\theta z}|_\Gamma, \tag{6.4}$$

where we have used the no-slip condition $\boldsymbol{u}|_\Gamma = 0$. A one-sided second order discretization of this boundary condition reads

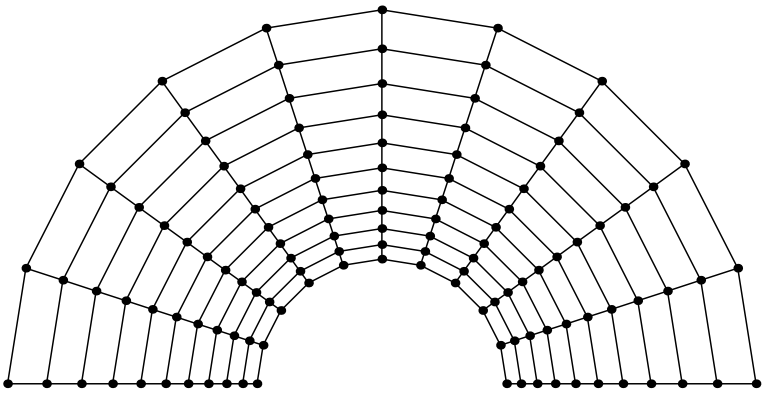

Fig. 11. A representative exponentially stretched (radially) polar grid Ω_h for the flow past a cylinder.

$$p_z|_\Gamma = -\nu \frac{4v_{1,j+1} - v_{2,j+1} - 4v_{1,j-1} + v_{2,j-1}}{4\Delta z \Delta\theta}. \tag{6.5}$$

Here, $v_{1,j}$ ($v_{2,j}$) denotes the value of v one (two) grid point from the boundary, and at the jth grid point in the angular direction.

The momentum equation (6.2) and PPE (6.3) were discretized using second order centered finite difference approximations. Symmetry of the flow was assumed along the horizontal centerline, and the computational grid Ω_h (with $\Delta z = \log 16/N$ and $\Delta\theta = \pi/N$) extended to a radius of $z = \log 16$ ($r = 16$). A representative grid is shown in Fig. 11. At the far-field computational boundary $r_{\max} = 16$, conditions for both $\boldsymbol{u}$ and p corresponding to a potential flow with unit free-stream velocity at infinity were applied, which for $\boldsymbol{u}$ is given by

$$(u, v) = ((r_{\max} - 1/r_{\max})\cos\theta, -(r_{\max} + 1/r_{\max})\sin\theta) \tag{6.6}$$

with p_z determined at the far-field by substitution of (6.6) into (6.2a). The Reynolds number was taken as $Re = 550 = (2R)/\nu = 2/\nu$ and solutions computed until $t = 3.0$ with three grid resolutions: $N = 256$, 512, and 1024. Fourth order Runge–Kutta time stepping was used, with CFL = 0.75 and Δt determined by the minimum of (3.1) and (3.2), and $\Delta x = \min\{\Delta z, \Delta\theta\}$.

Remark. We point out that explicit treatment of the pressure term in conjunction with the local pressure boundary condition discussed in (I) of Section 2 was implemented in the authors previous work for the cylinder flow [20]. However, as we stated in Section 2, using (2.2) to derive a local pressure boundary

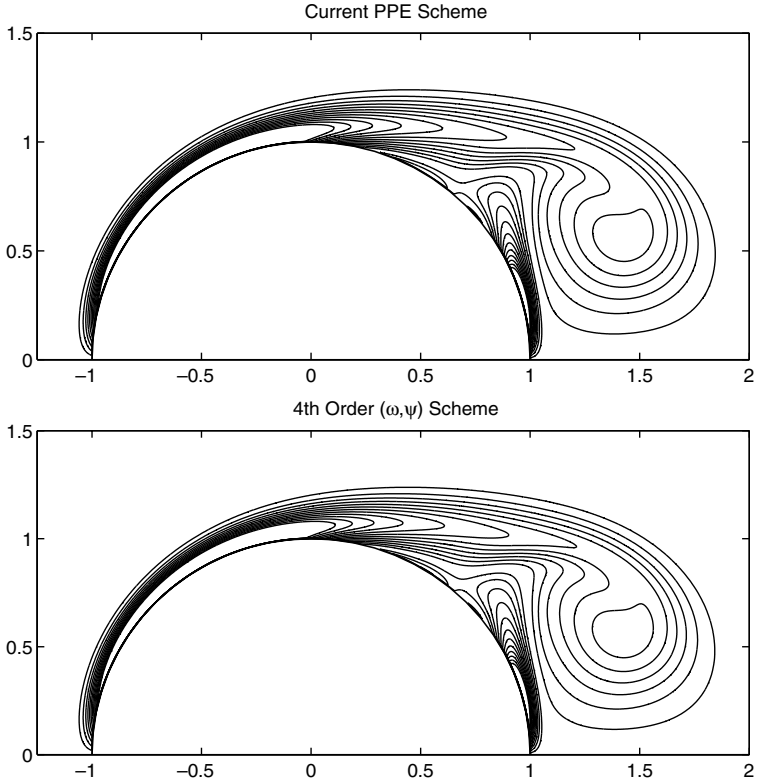

Fig. 12. Vorticity contours of the flow past a cylinder for $Re = 550$ at time $t = 3.0$. Contour levels are (−12:1:12), excluding 0.

condition requires that a diffusive stability time constraint must be satisfied, even if the viscous term is treated implicitly. Fortunately, if it were the case that implicit treatment of the viscous term was warranted (low Reynolds number flow) this restriction is removed completely by instead using the pressure boundary condition (1.2c) (here in the form of (6.4)).

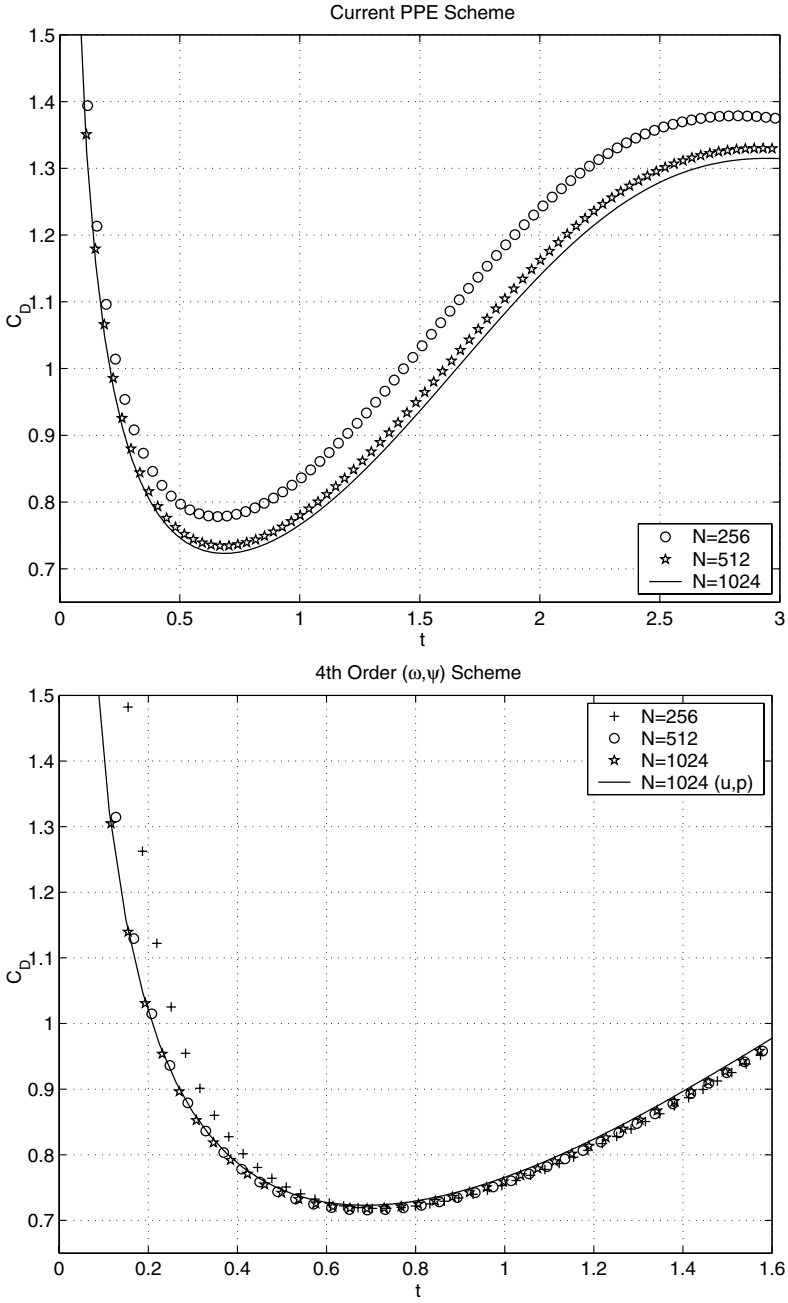

Fig. 13. Comparison of total drag for flow past a cylinder.

We compare the results above with computations of the polar form of a (ω,ψ) formulation of the NSE, to which the transformation $z=\log r$ has been applied. The governing equations are given by (with $\boldsymbol{u}$ defined as in (6.1))

$$\begin{aligned} &e^{2z}\omega_t+(u\omega)_z+(v\omega)_\theta=\nu\Delta_{(z,\theta)}\omega,\\ &\Delta\psi=-e^{2z}\omega, \end{aligned} \tag{6.7}$$

where $\boldsymbol{u}=(-\psi_\theta,\psi_z)$, with boundary conditions

$$\psi|_\Gamma=0 \quad \text{and} \quad \left.\frac{\partial\psi}{\partial\boldsymbol{n}}\right|_\Gamma=0. \tag{6.8}$$

A fourth order spatial discretization of (6.7) and (6.8) was implemented using local vorticity boundary conditions [20,19,8] and RK4 for the time discretization using the same parameters above for the $(\boldsymbol{u},p)$ computation.

In Fig. 12 is shown the vorticity contours (levels $-12{:}1{:}12$) for $N=1024$. Excellent agreement is seen between the two methods. A more sensitive comparison can be made using the computed coefficient of the total drag (C_{D}). In the vorticity-stream function, C_{D} may be computed using

$$C_{\mathrm{D}}=-2\nu\int_0^\pi\frac{\partial\omega}{\partial z}(0,\theta)\sin\theta\,\mathrm{d}\theta+2\nu\int_0^\pi\omega(0,\theta)\sin\theta\,\mathrm{d}\theta,$$

while in the velocity–pressure formulation by

$$C_{\mathrm{D}}=-2\int_0^\pi p(0,\theta)\cos\theta\,\mathrm{d}\theta-2\nu\int_0^\pi\frac{\partial v}{\partial z}(0,\theta)\sin\theta\,\mathrm{d}\theta.$$

The first image in Fig. 13 shows the computation of C_{D} using the current PPE scheme. The initial singular nature of the flow (impulsive start) at $t=0$ in clearly indicated, with the results of the computations on the two finest grids in very good agreement. Of particular interest in Fig. 13 is the comparison between the fourth order (ω,ψ) computations of C_{D} along with the $(\boldsymbol{u},p)$ scheme at the finest grid resolution (all computations are in good agreement after $t\approx 1.6$). This is an important point, for the dominant contribution to C_{D} in the $(\boldsymbol{u},p)$ scheme is computed from the pressure, which is integrated along the surface of the cylinder. This clearly indicates the accuracy of Neumann boundary condition (6.4) (or (1.2c)) in recovering the pressure when solving the PPE.

7. Conclusion

We have presented a pressure Poisson formulation of the Navier–Stokes equations, along with a general framework based on this formulation, for the development of accurate, stable and efficient numerical methods for the simulation of incompressible flow. The key to these methods is a particular choice for the Neumann pressure boundary condition for the pressure Poisson equation which allows explicit time treatment of the pressure, which does not affect the time step stability constraint. This is proven rigorously for the Stokes equation, and verified by systematic numerical experiments for the full nonlinear Navier–Stokes equations. The computation of the velocity and pressure are completely decoupled, resulting in a

class of extremely efficient methods. The methods are robust in that they can be applied to problems in both the small or large Reynolds number regimes, i.e., the viscous term can be treated either implicitly or explicitly in time. Moreover, the method is well suited for the computation of flows in general two- and three-dimensional domains since standard C^0 finite elements can be used for the spatial discretization. Thus, no matter the flow regime of interest the computational cost is reduced to the solution of a standard Poisson equation, and two or three (2D or 3D) Helmholtz equations in the case of implicit time discretization of the viscous term.

Acknowledgements

The authors thank Bob Pego for some very stimulating discussions. We also thank John Boyd for referring us to the detailed exposition in [28] on solving the discrete Neumann problem using collocation, and Nick Trefethen for allowing the inclusion of his MATLAB M-file cheb.m in our collocation code. The work of J.-G. Liu was supported by NSF Grant DMS-0107218. The work of H. Johnston was supported by a 2002 University of Michigan Rackham Research Fellowship.

Appendix A. MATLAB code for collocation implementation

In this section is listed the MATLAB code for the collocation method presented in Section 4. Note that we have followed the naming convention used in [28] for the auxiliary variables required for the PPE solver. The M-file cheb.m is taken from [34] with permission.

```
  function [uc,vc,pc,wc,divc,ut,vt,pt,wt,xx,yy,TFINAL]=coll2dnse(N,nu,cfl,tfinal);
%
% SPECTRAL COLLOCATION CODE FOR 2D NAVIER-STOKES
%
% Inputs:  N      - # of Chebyshev grid points
%          nu     - viscosity
%          cfl    - CFL number
%          tfinal - final time
%
% Outputs: uc, vc, pc, wc, and divc : Computed velocities, pressure, vorticity and
%                                     divergence of the velocity field at time TFINAL
%          ut, vt, pt, wt           : Exact values of flow quantities at time TFINAL
%          xx,yy                    : x and y grids points for plotting
%          TFINAL                   : Largest time <= tfinal which is a multiple of
%                                     the dt computed by (3.1).
%
%  To visualize the errors use (e.g. error in vorticity)  mesh(xx,yy,wt-wc)
%
%  NOTE: we index from 1:N+1 in MATLAB, not 0:N
%
  [D1,Dx,Dy,D2,D2T,xx,yy,dt,nts,TFINAL,c0_m,c0_p,cN_m,cN_p,b0,bN,e, ...
   Q,QT,invQ,invQT,Qevs,R,RT,invR,invRT,Revs,CM] = setup(N,nu,cfl,tfinal);

  [tu,tv,tp,tw,fu1,fu2,fu3,fv1,fv2,fv3,fu1_x,fu2_x,fu3_x,fv1_y,fv2_y,fv3_y]=accfuncs(xx,yy,nu);
```

```
  u1=tf1(-dt)*tu; v1=tf1(-dt)*tv; p1=tf1(-dt)*tp;   u2=tu; v2=tv; p2=tp;  % two time steps of

  NP1 = N+1; NM1 = N-1;  disp('Time Stepping :'); disp(' ');

  i=2:N; j=2:N;  % set vector indices for simpler code

  for k=1:nts      % TIME STEPPING

    time=(k-1)*dt;
    if (mod(k,25) == 0) disp(['step : ',num2str(k),'  time : ',num2str(time), ...
       '  div_inf : ',num2str(max(max(abs(Dx*u2+v2*Dy))))]);  end

%   Compute new velocities un and vn

    f_time = time+dt/2; ft1=tf1(f_time); ft2=tf2(f_time); ft3=tf3(f_time);
    FU = ft1*fu1+ft2*fu2+ft3*fu3; FV = ft1*fv1+ft2*fv2+ft3*fv3;
    FU_x =ft1*fu1_x+ft2*fu2_x+ft3*fu3_x; FV_y = ft1*fv1_y+ft2*fv2_y+ft3*fv3_y;
    NLT_u1 = u1.*(Dx*u1)+v1.*(u1*Dy); NLT_v1 = u1.*(Dx*v1)+v1.*(v1*Dy);
    NLT_u2 = u2.*(Dx*u2)+v2.*(u2*Dy); NLT_v2 = u2.*(Dx*v2)+v2.*(v2*Dy);
    lap_u2 = D2*u2+u2*D2T; lap_v2 = D2*v2+v2*D2T;

    un = u2+dt*(0.5*nu*lap_u2-(3*NLT_u2-NLT_u1)/2-Dx*(3*p2-p1)/2+FU);   % un rhs
    vn = v2+dt*(0.5*nu*lap_v2-(3*NLT_v2-NLT_v1)/2-(3*p2-p1)/2*Dy+FV);   % vn rhs

    un(i,j) = Q*((invQ*un(i,j)*invQT)./Qevs)*QT;                        % recover un (interior)
    vn(i,j) = Q*((invQ*vn(i,j)*invQT)./Qevs)*QT;                        % recover vn (interior)
    un(1,:)=0;un(NP1,:)=0;un(:,1)=0;un(:,NP1)=0;                        % set homogeneous
    vn(1,:)=0;vn(NP1,:)=0;vn(:,1)=0;vn(:,NP1)=0;                        % BC for un and vn

% pressure: 1st compute force terms and nonlinear terms

    f_time = time+dt; ft1=tf1(f_time); ft2=tf2(f_time); ft3=tf3(f_time);
    FU = ft1*fu1+ft2*fu2+ft3*fu3; FV = ft1*fv1+ft2*fv2+ft3*fv3;
    FU_x =ft1*fu1_x+ft2*fu2_x+ft3*fu3_x; FV_y = ft1*fv1_y+ft2*fv2_y+ft3*fv3_y;
    un_x = Dx*un; un_y = un*Dy; vn_x = Dx*vn; vn_y = vn*Dy;
    NLT_u = un.*un_x+vn.*un_y; NLT_v = un.*vn_x+vn.*vn_y;

    pn = 0.5*(un_x+vn_y).^2+FU_x+FV_y-(un_x.^2+2*vn_x.*un_y+vn_y.^2);  % PPE rhs

% compute Neumann BC for PPE: _m -> -1  and _p -> +1

    w = vn_x-un_y; w_x = Dx*w; w_y = w*Dy;
    px_m = FU(NP1,:)-nu*w_y(NP1,:); px_p = FU(1,:)-nu*w_y(1,:);
    py_m = FV(:,NP1)+nu*w_x(:,NP1); py_p = FV(:,1)+nu*w_x(:,1);

% adjust  RHS accounting for BC, and solve PPE for pn in the interior

    pn(i,j) = pn(i,j)-(1/e)*(D2(i,1)*(c0_m*px_m(j)+c0_p*px_p(j))+ ...
              D2(i,NP1)*(cN_m*px_m(j)+cN_p*px_p(j))+ ...
              (c0_m*py_m(i)+c0_p*py_p(i))*D2(j,1)'+(cN_m*py_m(i)+cN_p*py_p(i))*D2(j,NP1)');
```

```
    pn(i,j) = invR*pn(i,j)*invRT; pn(2,2)=0; pn(i,j) = R*(pn(i,j)./Revs)*RT; % solve for pn

    pn(  1,j)=(b0*pn(i,j)+c0_m*px_m(j)+c0_p*px_p(j))/e;         % recover pressure
    pn(NP1,j)=(bN*pn(i,j)+cN_m*px_m(j)+cN_p*px_p(j))/e;         % along the four
    pn(i,  1)=(pn(i,j)*b0'+c0_m*py_m(i)+c0_p*py_p(i))/e;        % sides, excluding
    pn(i,NP1)=(pn(i,j)*bN'+cN_m*py_m(i)+cN_p*py_p(i))/e;        % the corner points

% recover pressure at corner points

    cp(1,1) = px_p(1)-D1(1,j)*pn(i,1);  cp(2,1) = px_m(1)-D1(NP1,j)*pn(i,1);
    cp(1,2) = px_p(NP1)-D1(1,j)*pn(i,NP1); cp(2,2) = px_m(NP1)-D1(NP1,j)*pn(i,NP1);
    cp = CM\cp; pn(1,1)=cp(1,1); pn(NP1,1)=cp(2,1); pn(1,NP1)=cp(1,2); pn(NP1,NP1)=cp(2,2);

%  overwrite variables at new time step

    u1=u2; u2=un; v1=v2; v2=vn; p1=p2; p2=pn;

  end  % TIME STEPPING

% compute true solution at time TFINAL

  tfac=tf1(TFINAL); ut = tfac*tu; vt = tfac*tv; pt = tfac*tp; wt = tfac*tw;

  pdiff=pt(floor(N/2),floor(N/2))-p2(floor(N/2),floor(N/2));  % pressure adjustment
  uc = u2; vc = v2; pc = p2+pdiff; wc = Dx*vc-uc*Dy;  divc = Dx*uc+vc*Dy;

%%%%%%%%%%%%%%%%%% Setup computational parameters and auxillary variables.

  function [D1,Dx,Dy,D2,D2T,xx,yy,dt,nts,TFINAL,c0_m,c0_p,cN_m,cN_p,b0,bN,e, ...
            Q,QT,invQ,invQT,Qevs,R,RT,invR,invRT,Revs,CM] = setup(N,nu,cfl,tfinal)

  NP1=N+1; NM1=N-1;
  [D1,x]=cheb(N); y=x; Dx = D1; Dy = D1'; D2 = D1*D1; D2T = D2'; % differentiation matrices
  [xx,yy]=meshgrid(x,y); xx=xx'; yy=yy';

% compute stable dt and number of time steps

  dx=x(1)-x(2);  dt=cfl*dx;  nts=floor(tfinal/dt);  TFINAL=nts*dt;  % TFINAL <= tfinal

% Setup collocation matrices; see reference [22].

  [Q,evals]=eig(D2(2:N,2:N)); evals=diag(evals); QT=Q'; invQ =inv(Q); invQT = invQ';
  Qevs = 1-(nu*dt/2)*(repmat(evals,1,NM1)+repmat(evals',NM1,1));

  c0_m = -D1(1,NP1); c0_p = D1(NP1,NP1); cN_p = -D1(NP1,1); cN_m = D1(1,1);
  b0 = -c0_p*D1(1,2:N)-c0_m*D1(NP1,2:N); bN = -cN_m*D1(NP1,2:N)-cN_p*D1(1,2:N);
  e=c0_p*cN_m - c0_m*cN_p;

  [R,evals] = eig(D2(2:N,2:N)+(1/e)*(D2(2:N,1)*b0+D2(2:N,NP1)*bN));
  evals = diag(evals); [evals,ind] = sort(evals); evals = flipud(evals);
```

```
  evals(1)=0;  R=R(:,flipud(ind)'); RT = R'; invR =inv(R); invRT = invR';
  Revs = repmat(evals,1,NM1)+repmat(evals',NM1,1); Revs(1,1)=1;

  CM = [D1(1,1) D1(1,NP1);D1(NP1,1) D1(NP1,NP1)];  % needed to solve for pressure at corner pts

%%%%%%%%%%%%%%%%%% Forcing functions assuming exact solutions (4.10)

  function [tu,tv,tp,tw,fu1,fu2,fu3,fv1,fv2,fv3, ...
            fu1_x,fu2_x,fu3_x,fv1_y,fv2_y,fv3_y] = accfuncs(xx,yy,nu);

% tu = timefunc * (  (cos(pi*x/2))^2*cos(pi*y/2)*sin(pi*y/2)  )
% tv = timefunc * (  -cos(pi*x/2)*sin(pi*x/2)*(cos(pi*y/2))^2
% tp = timefunc * (  cos(pi*x/2)*sin(pi*y/2) )

  pxd2 = pi*xx/2; pyd2=pi*yy/2;
  cpxd2 = cos(pxd2); cpyd2 = cos(pyd2); spxd2 = sin(pxd2); spyd2 = sin(pyd2);
% u
  tu = cpxd2.^2.*cpyd2.*spyd2; tux = -pi*cpxd2.*cpyd2.*spyd2.*spxd2;
  tuy = (pi/2)*(cpxd2.^2.*cpyd2.^2-cpxd2.^2.*spyd2.^2);
  tuxx = (pi^2/2)*(spxd2.^2.*cpyd2.*spyd2-cpxd2.^2.*cpyd2.*spyd2);
  tuyx = (pi^2/2)*(cpxd2.*spyd2.^2.*spxd2-spxd2.*cpyd2.^2.*cpxd2);
  tuxy = tuyx;  tuyy = -(pi^2)*(cpxd2.^2.*cpyd2.*spyd2);
  tuyyx = (pi^3)*(spxd2.*cpyd2.*spyd2.*cpxd2);  laptu = tuxx+tuyy;
% v
  tv =  -cpxd2.*spxd2.*cpyd2.^2; tvx = (pi/2)*(spxd2.^2.*cpyd2.^2-cpxd2.^2.*cpyd2.^2);
  tvy = pi*cpxd2.*cpyd2.*spyd2.*spxd2; tvxx = (pi^2)*spxd2.*cpyd2.^2.*cpxd2;
  tvxy = (pi^2/2)*(cpxd2.^2.*cpyd2.*spyd2-spxd2.^2.*cpyd2.*spyd2);
  tvxyx = -(pi^3)*spxd2.*cpyd2.*spyd2.*cpxd2;
  tvyy = (pi^2/2)*(spxd2.*cpyd2.^2.*cpxd2-cpxd2.*spyd2.^2.*spxd2);
  laptv = tvxx+tvyy;
% p
  tp =  cpxd2.*spyd2; tpx = -(pi/2)*spxd2.*spyd2; tpxx = -(pi^2/4)*cpxd2.*spyd2;
  tpy =  (pi/2)*cpxd2.*cpyd2; tpyy = -(pi^2/4)*cpxd2.*spyd2; laptp = tpxx+tpyy;
% w - vorticity
  tw=tvx-tuy; twx=tvxx-tuyx; twy=tvxy-tuyy; twyx=tvxyx-tuyyx; twxy=twyx;
% set force terms
  fu1 = tpx+nu*twy; fu2 = tu; fu3 = tu.*tux+tv.*tuy; fv1 = tpy+nu*(-twx);
  fv2 = tv; fv3 = tu.*tvx+tv.*tvy; fu1_x = tpxx+nu*twyx; fu2_x = tux;
  fu3_x = tux.^2+tu.*tuxx+tvx.*tuy + tv.*tuxy; fv1_y = tpyy + nu*(-twxy); fv2_y = tvy;
  fv3_y = tu.*tvxy + tuy.*tvx + tv.*tvyy + tvy.^2;

%%%%%%%%%%%%%%%%%% TIME FUNCTIONS

  function t=tf1(ctime); t=cos(ctime);
  function t=tf2(ctime); t=-sin(ctime);
  function t=tf3(ctime); t=cos(ctime)*cos(ctime);

%%%%%%%%%%%%%%%%%%
%
% The following is taken with permission from reference [27].
%
```

```
% CHEB  compute D = differentiation matrix, x = Chebyshev grid

  function [D,x] = cheb(N)
  if N==0, D=0; x=1; return, end
  x = cos(pi*(0:N)/N)';
  c = [2; ones(N-1,1); 2].*(-1).^(0:N)';
  X = repmat(x,1,N+1);
  dX = X-X';
  D  = (c*(1./c)')./(dX+(eye(N+1)));      % off-diagonal entries
  D  = D - diag(sum(D'));                 % diagonal entries
```

Appendix B. Equivalence proof of the PPE formulation (4.1) and (1.1)

Theorem B.1. *Assume* $\boldsymbol{f} \in L^\infty([0,T],H^s)$, $s > 1/2$. *For* $\boldsymbol{u} \in L^\infty([0,T],H^{2+s}) \cap Lip\,([0,T],H^s)$ *the formulation* (1.1) *of the Navier–Stokes equations is equivalent to the PPE formulation* (4.1).

Proof. Assume $(\boldsymbol{u},p)$ is a solution of (1.1). Using $\nabla\cdot\boldsymbol{u}=0$ we can rewrite the momentum equation (1.1a) in rotational form as

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = -\nu\nabla\times\nabla\times\boldsymbol{u} + \boldsymbol{f}.$$

From this expression and the regularity assumptions of $\boldsymbol{u}$ and $\boldsymbol{f}$, ∇p is in $H^s(\Omega)$. Taking the normal component of the trace of the above equation along with $\boldsymbol{u}|_\Gamma = 0$ gives

$$\boldsymbol{n}\cdot\nabla p = -\nu\boldsymbol{n}\cdot\nabla\times\nabla\times\boldsymbol{u} + \boldsymbol{n}\cdot\boldsymbol{f}$$

on Γ, thus p satisfies (4.1c). Next, take the divergence of (1.1a), and using $\nabla\cdot\boldsymbol{u}=0$ and $\boldsymbol{u}\in L^\infty([0,T],H^2)$ gives

$$\Delta p = -\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) + \nabla\cdot\boldsymbol{f}.$$

Simple algebra and $\nabla\cdot\boldsymbol{u}=0$ gives $-\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) = \frac{1}{2}(\nabla\cdot\boldsymbol{u})^2 - (\nabla\boldsymbol{u}):(\nabla\boldsymbol{u})^{\mathrm{T}}$. Thus, $(\boldsymbol{u},p)$ satisfy the the PPE (4.1b). This proves that $(\boldsymbol{u},p)$ is also a solution to (4.1).

Now, assume $(\boldsymbol{u},p)$ is a solution of (4.1). All we need to show is that $\nabla\cdot\boldsymbol{u}=0$. Take the divergence of (4.1a) to obtain

$$\partial_t(\nabla\cdot\boldsymbol{u}) + \nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) + \Delta p = \nu\Delta(\nabla\cdot\boldsymbol{u}) + \nabla\cdot\boldsymbol{f}.$$

Now replace Δp above with the right-hand side of (4.1b). Using the identity

$$\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) = (\nabla\boldsymbol{u}):(\nabla\boldsymbol{u})^{\mathrm{T}} + \boldsymbol{u}\cdot\nabla(\nabla\cdot\boldsymbol{u})$$

we obtain

$$\partial_t(\nabla\cdot\boldsymbol{u}) + \boldsymbol{u}\cdot\nabla(\nabla\cdot\boldsymbol{u}) + \frac{1}{2}(\nabla\cdot\boldsymbol{u})^2 = \nu\Delta(\nabla\cdot\boldsymbol{u}).$$

Denoting $\phi = \nabla\cdot\boldsymbol{u}$, the above equation becomes

$$\partial_t\phi + \boldsymbol{u}\cdot\nabla\phi + \frac{1}{2}\phi^2 = \nu\Delta\phi \tag{B.1}$$

with initial data $\phi|_{t=0}=0$. Rewrite (4.1a) as

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = -\nu\nabla\times\nabla\times\boldsymbol{u} + \nu\nabla(\nabla\cdot\boldsymbol{u}) + \boldsymbol{f}.$$

Again, the regularity assumptions for $\boldsymbol{u}$ and $\boldsymbol{f}$ imply ∇p is also in $H^s(\Omega)$. Taking the normal component of the trace in the above equation and using $\boldsymbol{u}|_\Gamma = 0$ gives

$$\boldsymbol{n} \cdot \nabla p = -\nu \boldsymbol{n} \cdot \nabla \times \nabla \times \boldsymbol{u} + \boldsymbol{n} \cdot \nabla(\nabla \cdot \boldsymbol{u}) + \boldsymbol{n} \cdot \boldsymbol{f}$$

on Γ. Comparing with (4.1c), we have

$$\frac{\partial \phi}{\partial \boldsymbol{n}} = \boldsymbol{n} \cdot \nabla(\nabla \cdot \boldsymbol{u}) = 0.$$

We now show using energy estimates that ϕ equals zero almost everywhere. Multiply (B.1) by 2ϕ and integrate over the domain Ω to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \phi^2 \, \mathrm{d}x + \int_\Omega \boldsymbol{u} \cdot \nabla(\phi^2) \, \mathrm{d}x + \int_\Omega \phi^3 \, \mathrm{d}x = -2\nu \int_\Omega |\nabla \phi|^2 \, \mathrm{d}x.$$

Integration by parts also gives

$$\int_\Omega \boldsymbol{u} \cdot \nabla(\phi^2) \, \mathrm{d}x = -\int_\Omega (\nabla \cdot \boldsymbol{u}) \phi^2 \, \mathrm{d}x = -\int_\Omega \phi^3 \, \mathrm{d}x.$$

Hence, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \phi^2 \, \mathrm{d}x + 2\nu \int_\Omega |\nabla \phi|^2 \, \mathrm{d}x = 0 \tag{B.2}$$

with the initial conditions

$$\int_\Omega \phi^2 \, \mathrm{d}x = 0 \quad \text{for } t = 0.$$

Hence, we have

$$\int_\Omega \phi^2 \, \mathrm{d}x = 0$$

for all $t > 0$. Therefore, we have proved that $\phi = 0$ a.e., or $\nabla \cdot \boldsymbol{u} = 0$ a.e., which proves that $(\boldsymbol{u}, p)$ is also the solution to (1.1), and completes the proof of the theorem. □

Appendix C. Normal mode analysis for the Stokes equations

Normal mode analysis is a commonly employed tool for investigating numerical schemes for the NSE. It has proven invaluable in revealing the complicated and subtle behavior of numerical boundary layers in projection methods [9,26]. Here we investigate the second order implicit time discretization (3.3) and (3.4) of our PPE formulation (1.2) (or (2.7)) by applying normal mode analysis to a 1D model of the unsteady 2D Stokes equations. Full second order accuracy is shown for the velocity field and pressure, as well as the divergence of the velocity field. Moreover, unlike projection methods, the flow variables are completely free of any numerical boundary layers. We note that the same analysis, while a bit more cumbersome, can be applied to the 3D Stokes equations.

Consider the unsteady 2D Stokes equations given by

$$\boldsymbol{u}_t + \nabla p = \nu \Delta \boldsymbol{u}, \tag{C.1a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{C.1b}$$

$$\boldsymbol{u}|_{x=-1,1} = 0 \tag{C.1c}$$

on the domain $\Omega = [-1,1] \times (0, 2\pi)$ with periodic boundary conditions in y. Assume solutions of the form $\boldsymbol{u} = \mathrm{e}^{\mathrm{i}ky}[u(x,t), v(x,t)]^{\mathrm{T}}$ and $p = \mathrm{e}^{\mathrm{i}ky} p(x,t)$, let $\widetilde{v} = \mathrm{i}v$, and rename (for simplicity of notation) $\widetilde{v}$ by v. Then solutions of (C.1) reduce to a family of 1D problems indexed by $k \in Z$ given by

$$\partial_t u + \partial_x p = \nu(\partial_x^2 - k^2)u, \quad \partial_t v - kp = \nu(\partial_x^2 - k^2)v, \tag{C.2a}$$

$$\partial_x u + kv = 0, \tag{C.2b}$$

$$u(\pm 1, t) = v(\pm 1, t) = 0. \tag{C.2c}$$

Our equivalent PPE formulation (1.2) of (C.2) is given by

$$\partial_t u + \partial_x p = \nu(\partial_x^2 - k^2)u, \quad \partial_t v - kp = \nu(\partial_x^2 - k^2)v, \tag{C.3a}$$

$$(\partial_x^2 - k^2)p = 0, \tag{C.3b}$$

$$u(\pm 1, t) = v(\pm 1, t) = \partial_x(p + \nu k v)(\pm 1, t) = 0. \tag{C.3c}$$

The 1D linear models (C.2) and (C.3) still embody the essential features of the NSE, i.e., an incompressible velocity field and the presence of a viscous term, while allowing us to analyze in a simplified setting the effects of discrete time stepping.

Assuming normal mode solutions of (C.3) given by

$$(u, v, p)(x, t) = e^{\sigma t}\ (\widehat{u}, \widehat{v}, \widehat{p})(x), \tag{C.4}$$

where we take σ to be of the form $\sigma = -\nu(k^2 + \mu^2)$, with conditions on μ to be determined later. Plugging (C.4) into the continuous PPE formulation (C.3) gives

$$\nu(\partial_x^2 + \mu^2)u = \partial_x p, \quad \nu(\partial_x^2 + \mu^2)v = -kp, \tag{C.5a}$$

$$(\partial_x^2 - k^2)p = 0, \tag{C.5b}$$

$$u(\pm 1) = v(\pm 1) = \partial_x(p + \nu k v)(\pm 1) = 0, \tag{C.5c}$$

where for simplicity we have dropped the $\widehat{\ }$ symbol.

The system (C.5) admits two distinct families of solutions, one odd and one even (determined by the choice of $\sinh(kx)$ or $\cosh(kx)$ for $p(x)$). Since the analysis of each is similar, without loss of generality we consider the odd solutions, given by

$$u(x) = A\left(\frac{\cos \mu x}{\cos \mu} - \frac{\cosh kx}{\cosh k}\right), \quad v(x) = B\left(\frac{\sin \mu x}{\sin \mu} - \frac{\sinh kx}{\sinh k}\right),$$
$$p(x) = \sinh(kx). \tag{C.6}$$

The velocity boundary conditions in (C.5c) are clearly satisfied by u and v in (C.6). Moreover, plugging (C.6) into (C.5a) it follows that

$$A = -\frac{k \cosh k}{\nu(k^2 + \mu^2)}, \quad B = \frac{k \sinh k}{\nu(k^2 + \mu^2)}. \tag{C.7}$$

A direct calculation applied to $p(x)$ in (C.6) gives

$$\partial_x(p + \nu k v)(x) = \frac{\nu k B}{\sin \mu}\, \mu \cos \mu x + \left(1 - \frac{\nu k B}{\sinh k}\right) k \cosh kx. \tag{C.8}$$

Thus in order to satisfy the boundary condition for p in (C.5c) we must have that

$$\nu kB\mu\cot\mu + (\sinh k - \nu kB)k\coth k = 0 \quad \text{or} \quad k\tanh k + \frac{1}{\mu}\left(\frac{k\sinh k}{\nu B} - k^2\right)\tan\mu = 0,$$

which after substitution of the expression for B in (C.7) gives an equation for μ, namely

$$k\tanh k + \mu\tan\mu = 0. \tag{C.9}$$

Rewriting (C.9) gives $\mu = -k\tanh k\cot\mu$, thus for each fixed k a solution μ is given by the intersection of the linear function μ and the curve $-k\tanh k\cot\mu$, the latter of which has a range of $-\infty$ to ∞ over each interval $(\ell, (\ell+1)\pi)$ for $\ell \in Z$. Hence for every k, $\ell = 0, \pm1, \pm2, \ldots$, Eq. (C.9) has a unique real solution μ in $(\ell\pi, (\ell+1)\pi)$. Furthermore, for $\nu > 0$ it then follows that $\sigma < 0$ in (C.4), and as expected solutions decay in time.

Finally, from (C.6) and (C.7) and (C.9) we have

$$\begin{aligned}\partial_x u + kv &= \frac{k\cosh k}{\nu(k^2+\mu^2)}\left(\mu\frac{\sin\mu x}{\cos\mu} + k\frac{\sinh kx}{\cosh k}\right) + \frac{k^2\sinh k}{\nu(k^2+\mu^2)}\left(\frac{\sin\mu x}{\sin\mu} - \frac{\sinh kx}{\sinh k}\right)\\ &= \frac{k}{\nu(k^2+\mu^2)}\frac{\cosh k}{\sin\mu}(\mu\tan\mu + k\tanh k)\sin\mu x = 0,\end{aligned} \tag{C.10}$$

showing that incompressibility is indeed satisfied in the PPE formulation.

With the groundwork now laid we examine a discrete time discretization of the PPE system. Following (3.3) and (3.4), a second order semi-implicit time discretization of (C.1) is given by

$$\begin{aligned}&\frac{u^{n+1}-u^n}{\Delta t} + \partial_x\left(\frac{3}{2}p^n - \frac{1}{2}p^{n-1}\right) = \nu(\partial_x^2 - k^2)\frac{(u^{n+1}+u^n)}{2},\\ &\frac{v^{n+1}-v^n}{\Delta t} - k\left(\frac{3}{2}p^n - \frac{1}{2}p^{n-1}\right) = \nu(\partial_x^2 - k^2)\frac{(v^{n+1}+v^n)}{2},\\ &(\partial_x^2 - k^2)p^{n+1} = 0,\\ &u^{n+1}(\pm1) = v^{n+1}(\pm1) = \partial_x(p+\nu kv)^{n+1}(\pm1) = 0.\end{aligned} \tag{C.11}$$

Keeping space continuous, assume normal mode solutions for (C.11) of the form

$$(u,v,p)^n(x) = \kappa^n\ (\widehat{u},\widehat{v},\widehat{p})(x). \tag{C.12}$$

Plugging (C.12) into (C.11) gives (and for simplicity of presentation dropping the $\widehat{\ }$ symbol)

$$\begin{aligned}&\frac{\kappa-1}{\Delta t}u + \frac{3\kappa-1}{2\kappa}\partial_x p = \nu\frac{\kappa+1}{2}(\partial_x^2 - k^2)u, \quad \frac{\kappa-1}{\Delta t}v - \frac{3\kappa-1}{2\kappa}kp = \nu\frac{\kappa+1}{2}(\partial_x^2-k^2)v,\\ &(\partial_x^2 - k^2)p = 0,\\ &u(\pm1) = v(\pm1) = \partial_x(p+\nu kv)(\pm1) = 0.\end{aligned} \tag{C.13}$$

Introducing $\widetilde{\mu}$, let

$$\frac{2(\kappa-1)}{(\kappa+1)\Delta t} = -\nu(k^2+\widetilde{\mu}^2) \quad \Rightarrow \quad \kappa = \frac{2-\nu(k^2+\widetilde{\mu}^2)\Delta t}{2+\nu(k^2+\widetilde{\mu}^2)\Delta t}. \tag{C.14}$$

Substitution of the first expression of (C.14) into (C.13) gives

$$v(\partial_x^2+\widetilde{\mu}^2)u=\frac{3\kappa-1}{\kappa(\kappa+1)}\partial_x p,\quad v(\partial_x^2+\widetilde{\mu}^2)v=-\frac{3\kappa-1}{\kappa(\kappa+1)}kp, \tag{C.15a}$$

$$(\partial_x^2-k^2)p=0, \tag{C.15b}$$

$$u(\pm1)=v(\pm1)=\partial_x(p+vkv)(\pm1)=0. \tag{C.15c}$$

The odd solutions of (C.15) take the form

$$u(x)=\widetilde{A}\left(\frac{\cos\widetilde{\mu}x}{\cos\widetilde{\mu}}-\frac{\cosh kx}{\cosh k}\right),\quad v(x)=\widetilde{B}\left(\frac{\sin\widetilde{\mu}x}{\sin\widetilde{\mu}}-\frac{\sinh kx}{\sinh k}\right), \tag{C.16a}$$

$$p(x)=\sinh(kx). \tag{C.16b}$$

Note that u and v in (C.16a) and (C.16b) satisfy the boundary conditions $u(\pm1)=v(\pm1)=0$, and by substitution of (C.16a), (C.16b) into (C.15a) we see that

$$\widetilde{A}=-\frac{3\kappa-1}{\kappa(\kappa+1)}\frac{k\cosh k}{v(k^2+\widetilde{\mu}^2)},\quad \widetilde{B}=\frac{3\kappa-1}{\kappa(\kappa+1)}\frac{k\sinh k}{v(k^2+\widetilde{\mu}^2)}. \tag{C.17}$$

Since

$$\partial_x(p+vkv)(x)=\frac{vk\widetilde{B}}{\sin\widetilde{\mu}}\widetilde{\mu}\cos\widetilde{\mu}x+\left(1-\frac{vk\widetilde{B}}{\sinh k}\right)k\cosh kx,$$

in order to satisfy the pressure boundary condition in (C.15c) we must have that

$$vk\widetilde{B}\widetilde{\mu}\cot\widetilde{\mu}+(\sinh k-vk\widetilde{B})k\coth k=0\quad\Rightarrow\quad k\tanh k+\frac{1}{\widetilde{\mu}}\left(\frac{k\sinh k}{v\widetilde{B}}-k^2\right)\tan\widetilde{\mu}=0. \tag{C.18}$$

A direct computation using $\widetilde{B}$ from (C.17) and κ from (C.14) gives

$$\frac{1}{\widetilde{\mu}}\left(\frac{k\sinh k}{v\widetilde{B}}-k^2\right)=\widetilde{\mu}+\frac{(\kappa-1)^2}{3\kappa-1}\frac{k^2+\widetilde{\mu}^2}{\widetilde{\mu}}=\widetilde{\mu}+\frac{v^2(k^2+\widetilde{\mu}^2)^3}{\widetilde{\mu}(1-v(k^2+\widetilde{\mu}^2)\Delta t)}\Delta t^2,$$

which upon substitution into the last expression of (C.18) gives

$$k\tanh k+\left(\widetilde{\mu}+\frac{v^2(k^2+\widetilde{\mu}^2)^3}{\widetilde{\mu}(1-v(k^2+\widetilde{\mu}^2)\Delta t)}\Delta t^2\right)\tan\widetilde{\mu}=0. \tag{C.19}$$

Comparing the above with (C.9), for each fixed k and μ we have $\widetilde{\mu}=\mu+\mathrm{O}(\Delta t^2)$, and a direct computation gives

$$\kappa-\mathrm{e}^{\sigma\Delta t}=\frac{2-v(k^2+\widetilde{\mu}^2)\Delta t}{2+v(k^2+\widetilde{\mu}^2)\Delta t}-1+v(k^2+\mu^2)\Delta t-\frac{v^2}{2}(k^2+\mu^2)^2\Delta t^2+\mathrm{O}(\Delta t^3)=\mathrm{O}(\Delta t^3). \tag{C.20}$$

Comparing (C.4) and (C.12), (C.20) shows that the scheme (C.12) is fully second order accurate in time. Moreover, comparing (C.16) and (C.17) with (C.6) and (C.7) clearly indicates an absence of numerical boundary layers in any of the flow variables. Finally, from (C.16), (C.17) and (C.19) we have

$$\partial_x u + kv = \frac{3\kappa - 1}{\kappa(\kappa+1)} \frac{k}{v(k^2+\widetilde{\mu}^2)} \frac{\cosh k}{\sin \mu} (\mu \tan \mu + k \tanh k) \sin \mu x$$
$$= -\left(\frac{3\kappa - 1}{\kappa(\kappa+1)} \frac{kv(k^2+\widetilde{\mu}^2)^2}{\widetilde{\mu}(1 - v(k^2+\widetilde{\mu}^2)\Delta t)} \frac{\cosh k}{\cos \mu} \sin \mu x \right) \Delta t^2, \tag{C.21}$$

which shows that incompressibility is satisfied to second order accuracy.

References

[1] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (1989) 257–283.
[2] J.P. Boyd, Chebyshev and Fourier Spectral Methods, second ed. (revised), Dover, New York, 2001.
[3] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.
[4] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer, New York, 1988.
[5] S. Chen, G.D. Doolen, R.H. Kraichnan, Z. She, On statistical correlation between velocity increments and locally averaged dissipation in homogeneous turbulence, Phys. Fluids 5 (1993) 458–463.
[6] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[7] W. E, J.-G. Liu, Vorticity boundary conditions and related issues for finite difference schemes, J. Comput. Phys. 124 (1996) 368–382.
[8] W. E, J.-G. Liu, Essentially compact schemes for unsteady viscous incompressible flows, J. Comput. Phys. 126 (1996) 122–138.
[9] W. E, J.-G. Liu, Projection Method I: convergence and numerical boundary layers, SIAM J. Numer. Anal. 33 (1996) 1597–1621.
[10] W. E, J.-G. Liu, Gauge method for viscous incompressible flows, Commun. Math. Sci. 1 (2003) 317–332.
[11] U. Ehrenstein, R. Peyret, A Chebyshev collocation method for the Navier–Stokes equations with application to double-diffusive convection, Int. J. Numer. Meth. Fluids 9 (1987) 427–452.
[12] M.O. Deville, P.F. Fischer, E.H. Mund, High-order Methods for Incompressible Fluid Flow, Cambridge University Press, Cambridge, 2002.
[13] V. Girault, P.A. Raviart, Finite Element Methods for Navier–Stokes Equations, Theory and Algorithms, Springer, Berlin, 1986.
[14] P.M. Gresho, R.L. Sani, On pressure boundary conditions for the incompressible Navier–Stokes equations, Int. J. Numer. Meth. Fluids 7 (1987) 1111–1145.
[15] P.M. Gresho, Some current CFD issues relevant to the incompressible Navier–Stokes equations, Comput. Meth. Appl. Mech. Eng. 87 (1991) 201–252.
[16] J.-L. Guermond, J. Shen, A new class of truly consistent splitting schemes for incompressible flows, J. Comput. Phys. 192 (2003) 262–276.
[17] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.
[18] W.D. Henshaw, A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids, J. Comput. Phys. 113 (1994) 13–25.
[19] H. Johnston, Efficient computation of viscous incompressible flow, Ph.D. Thesis, Temple University, Philadelphia, PA, 1999.
[20] H. Johnston, J.-G. Liu, A finite difference method for incompressible flow based on local pressure boundary conditions, J. Comput. Phys. 180 (2002) 120–154.
[21] G. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 97 (1991) 414–443.
[22] A.-K. Kassam, L.N. Trefethen, Fourth-order time-stepping for stiff PDEs, SIAM J. Sci. Comput. (to appear).
[23] R.M. Kerr, Evidence for a singularity of the three-dimensional, incompressible Euler equations, Phys. Fluids A 5 (1993) 1725–1746.
[24] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[25] L. Kleiser, U. Schumann, Treatment of the incompressibility and boundary conditions in 3-D numerical spectral simulation of plane channel flows, in: E.H. Hirschel (Ed.), Notes on Numerical Fluid Mechanics, Vieweg, Braunschweig, 1980, pp. 165–173.
[26] S.A. Orszag, M. Israeli, M. Deville, Boundary conditions for incompressible flows, J. Sci. Comput. 1 (1986) 75–111.
[27] G.S. Patterson, S.A. Orszag, Spectral calculation of isotropic turbulence: efficient removal of aliasing interactions, Phys. Fluids 14 (1971) 2538–2541.
[28] R. Peyret, Spectral Methods for Incompressible Viscous Flow, Springer, New York, 2002.

[29] S.J. Ruuth, Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (1997) 151–167.
[30] J. Shen, Efficient Spectral-Galerkin Method I. Direct solvers for the second and fourth order equations using Legendre polynomials, SIAM J. Sci. Comput. 15 (1994) 1489–1505.
[31] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Wadsworth and Brooks/Cole, 1989.
[32] R. Temam, Sur l'approximation de la solution des equations Navier–Stokes par la méthode des Fractionnarires II, Arch. Rational Mech. Anal. 33 (1969) 377–385.
[33] L.J.P. Timmermans, P.D. Minev, F.N. Van De Vosse, An approximate projection scheme for incompressible flow using spectral elements, Int. J. Numer. Meth. Fluids 22 (1996) 673–688.
[34] L.N. Trefethen, Spectral Methods in MATLAB, SIAM, Philadelphia, PA, 2000.
[35] J. van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Sci. Stat. Comput. 7 (1986) 870–891.
[36] P. Wesseling, An Introduction to Multigrid Methods, Wiley, New York, 1992.