Journal of Computational Physics **180,** 120–154 (2002)
doi:10.1006/jcph.2002.7079

Finite Difference Schemes for Incompressible Flow Based on Local Pressure Boundary Conditions

Hans Johnston* and Jian-Guo Liu†

**Department of Mathematics, University of Michigan, Ann Arbor, Michigan 48109; and †Institute for Physical Science and Technology and Department of Mathematics, University of Maryland, College Park, Maryland 20742*
E-mail: hansjohn@umich.edu

Received September 18, 2000; revised April 8, 2002

In this paper we discuss the derivation and use of local pressure boundary conditions for finite difference schemes for the unsteady incompressible Navier–Stokes equations in the velocity–pressure formulation. Their use is especially well suited for the computation of moderate to large Reynolds number flows. We explore the similarities between the implementation and use of local pressure boundary conditions and local vorticity boundary conditions in the design of numerical schemes for incompressible flow in 2D. In their respective formulations, when these local numerical boundary conditions are coupled with a fully explicit convectively stable time stepping procedure, the resulting methods are simple to implement and highly efficient. Unlike the vorticity formulation, the use of the local pressure boundary condition approach is readily applicable to 3D flows. The simplicity of the local pressure boundary condition approach and its easy adaptation to more general flow settings make the resulting scheme an attractive alternative to the more popular methods for solving the Navier–Stokes equations in the velocity–pressure formulation. We present numerical results of a second-order finite difference scheme on a nonstaggered grid using local pressure boundary conditions. Stability and accuracy of the scheme applied to Stokes flow is demonstrated using normal mode analysis. Also described is the extension of the method to variable density flows. © 2002 Elsevier Science (USA)

Key Words: incompressible flow; finite difference methods; pressure Poisson solver; local pressure boundary conditions.

1. INTRODUCTION AND OUTLINE OF THE SCHEME

The primitive variable formulation of the incompressible Navier–Stokes equations (NSE) on a domain $\Omega \subset \mathbb{R}^2$ (or $\mathbb{R}^3$) takes the form

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p = \nu \Delta \boldsymbol{u}, \\ \nabla \cdot \boldsymbol{u} = 0, \end{cases} \tag{1.1a–b}$$

where $\boldsymbol{u} = (u, v)^T$ (or $\boldsymbol{u} = (u, v, w)^T$), p, and ν are the velocity field, pressure, and kinematic viscosity, respectively. For now we consider the simplest physical boundary condition for $\boldsymbol{u}$, the no-penetration, no-slip condition

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

where $\Gamma = \partial\Omega$.

Numerical schemes for 2D computations of (1.1) and (1.2) have been quite successful, beginning with the pioneering MAC scheme [12] and projection methods [4, 21] in the late 60s. The current focus in computational incompressible fluid dynamics is (1) rapid 3D computation, (2) incorporation of more physics into the fluid equations, and (3) schemes for general geometries. The main objective of this paper is to investigate and adapt ideas used in a class of finite difference based on *local vorticity boundary conditions* for the vorticity–stream function formulation of the NSE to the primitive variables formulation (1.1). These schemes have been particularly successful in the computation of large Reynolds number 2D flows [5, 6, 8]. However, there is a major difference between 2D and 3D for vorticity-based numerical methods. Most apparent is the fact that both the vorticity and the stream function are vector fields in 3D (instead of scalar). Along with this comes the necessity of enforcing divergence-free conditions for the vorticity and the stream function. This turns out to be a major problem in the design of efficient numerical methods in 3D based on this formulation. Hence, for these reasons as well as others discussed below, it is desirable to design numerical schemes for the NSE in 3D based on the primitive variables formulation.

For a 2D, simply connected domain, (1.1) and (1.2) are equivalent to the vorticity–stream function formulation of the NSE given by

$$\begin{cases} \omega_t + (\boldsymbol{u} \cdot \nabla)\omega = \nu \Delta \omega, \\ \Delta \psi = \omega, \end{cases} \tag{1.3a–b}$$

where $\omega = \nabla \times \boldsymbol{u} = v_x - u_y$ is the vorticity, ψ is the stream function, and the velocity is recovered from $\boldsymbol{u} = \nabla^\perp \psi = (-\psi_y, \psi_x)^{\mathrm{T}}$. The velocity boundary conditions (1.2) are now written in terms of ψ as

$$\psi|_\Gamma = 0 \quad \text{and} \quad \left.\frac{\partial \psi}{\partial \boldsymbol{n}}\right|_\Gamma = 0. \tag{1.4}$$

This formulation is advantageous in a numerical setting over (1.1) and (1.2) for it eliminates the pressure variable, automatically enforces incompressibility at the discrete level, and requires only the scalar ω to be advanced in time compared to the two velocity components required by (1.1). The main numerical challenges now are the proper implementation of *two* boundary conditions for ψ, namely (1.4), and the lack of an explicit boundary condition for

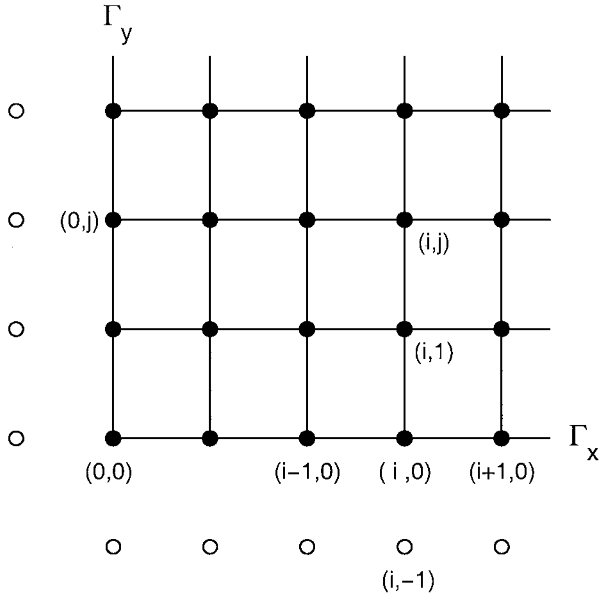


FIG. 1. A representative nonstaggered grid Ω_h.

ω. One approach which overcomes these difficulties is based on so-called *local vorticity boundary conditions*. To briefly explain, denote by Ω_h a nonstaggered finite difference grid for Ω, which without loss of generality has equal spacing h in each coordinate direction. Then along Γ, e.g., at the $(i, 0)$ grid point along Γ_x (see Fig. 1), a second-order centered discretization of (1.3b) is given by

$$\begin{aligned}\omega_{i,0} &= (\psi_{xx} + \psi_{yy})_{i,0}\\ &= (\psi_{i+1,0} + \psi_{i-1,0} + \psi_{i,1} + \psi_{i,-1} - 4\psi_{i,0})/h^2 + O(h^2)\\ &= (\psi_{i,1} + \psi_{i,-1})/h^2 + O(h^2),\end{aligned} \tag{1.5}$$

where we have used $\psi|_\Gamma = 0$. A discretization of the no-slip boundary condition, $0 = (\partial\psi/\partial\boldsymbol{n})_{i,0} = (\psi_{i,1} - \psi_{i,-1})/2h + O(h^2)$, along with (1.5) leads to

$$\omega_{i,0} = 2\psi_{i,1}/h^2 + O(h), \tag{1.6}$$

which is well known as Thom's formula [5 , 22]. While (1.6) is an $O(h)$ approximation of ω at the boundary, a careful truncation error analysis shows that when Thom's formula is coupled with a second-order discretization of (1.3), second-order accuracy is achieved [15] throughout the computational domain. With (1.6), an outline of the main time loop for (1.3) and (1.4) is as follows:

Step 1: Update ω in time at the *interior* grid points $\{(x_i, y_j) \mid i, j > 0\}$ via the momentum equation (1.3a).

Step 2: Solve the Poisson equation (1.3b) for ψ with the Dirichlet boundary condition $\psi|_\Gamma = 0$, and update the velocity via $\boldsymbol{u} = \nabla^\perp\psi$, with $\boldsymbol{u}|_\Gamma = 0$.

Step 3: Recover ω along Γ using (1.6) from the ψ computed in Step 2.

In the flow regime of moderate to large Reynolds numbers, the diffusive time step constraint is much less restrictive than the convective time step constraint. Thus, to efficiently realize Step 1 in this regime, the viscous term in (1.3a) should be treated *explicitly*. If the viscous term were to be treated implicitly then there would be no prescription for ω along Γ, which would result in a coupled system for (1.3). Iterative procedures have been proposed which use Thom's formula (1.6) in the solution of the coupled system; however, convergence is in general very slow and may even diverge at large Reynolds numbers. For a detailed discussion see [19]. We shall also point out that implicit treatment of the viscous term does not help the stability for large Reynolds numbers, which can be shown using linear stability analysis. In the case of explicit treatment of the viscous term some comments are in order. First, one should use a high-order time integration method, such as RK4, whose stability region includes a portion of imaginary axis. This avoids any severe stability constraints on the time step coming from the convective term. Upwind discretization of the convection term is not recommended in a resolved computation for it can add significant numerical dissipation which can dominate the physical dissipation and is hence not appropriate to achieve resolved large Reynolds number computations. For steady-state calculations much research has been done on the implicit or semi-implicit treatment of the convection term in order to take advantage of the large time steps permitted in these approaches to quickly reach the steady state. However, the resulting nonlinear or nonsymmetric systems that arise in such discretizations are generally computationally expensive to solve. Thus, for unsteady computations, with a time step satisfying the CFL constraint of the time stepping scheme, such implicit and semi-implicit schemes are not necessary from the viewpoint of both accuracy and stability. Secondly, the computation of (1.3a) decouples from that of (1.3b). The main computation then consists of solving a Poisson equation in Step 2, for which standard fast solvers can be used. Hence, the overall method is simple to implement and highly efficient. A detailed discussion of this approach as well as higher order spatial implementations and finite element implementations can be found in [5, 6, 8].

We now turn to the main focus of this paper: the application of a strategy similar to that of local vorticity boundary conditions to the computation of the NSE in the $(\boldsymbol{u}, p)$ formulation. To begin, in place of (1.1) and (1.2) we use an alternative primitive variable formulation of the NSE, the pressure Poisson equation (PPE) formulation.

$$\begin{cases}\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \nu\Delta\boldsymbol{u},\\ \Delta p = (\nabla\cdot\boldsymbol{u})^2 - (\nabla\boldsymbol{u}) : (\nabla\boldsymbol{u})^T,\end{cases} \tag{1.7a–b}$$

along with the boundary conditions

$$\boldsymbol{u}|_\Gamma = 0 \qquad \text{and} \qquad \nabla\cdot\boldsymbol{u}|_\Gamma = 0. \tag{1.8}$$

The divergence-free condition (1.1b) has been replaced by the PPE (1.7b) *and* the divergence-free velocity boundary condition in (1.8). A detailed discussion of the PPE formulation and related issues can be found in [10, 11]. For completeness, in Appendix A we show that formulation (1.7) and (1.8) is equivalent to the formulation (1.1) and (1.2) in the solution class $\boldsymbol{u} \in L^\infty([0, T], H^2)$.

Analogous to ψ and ω in the vorticity formulation, in (1.8) there are *two* boundary conditions for $\boldsymbol{u}$ and no *explicit* boundary condition for p. In fact, the lack of a proper boundary condition for p has traditionally been a stumbling block in the design of accurate and efficient numerical schemes based on the velocity–pressure formulation. A natural candidate is given by the normal component of the momentum equation (1.7) along Γ, a discretization of which, referring again to Fig. 1, at the $(i, 0)$ grid point along Γ_x is

$$\begin{aligned}\partial p/\partial \boldsymbol{n} &= \nu(v_{xx} + v_{yy})\\ &= \nu(v_{i+1,0} + v_{i-1,0} + v_{i,1} + v_{i,-1} - 4v_{i,0})/h^2 + O(h^2)\\ &= \nu(v_{i,1} + v_{i,-1})/h^2 + O(h^2),\end{aligned} \tag{1.9}$$

where we have used $\boldsymbol{u}|_\Gamma = 0$. Note that (1.9), like (1.5), requires the value of a flow variable outside the computational domain, namely $v_{i,-1}$. If we discretize the divergence-free boundary condition $\nabla \cdot \boldsymbol{u} = 0$ in (1.8), then

$$0 = \nabla \cdot \boldsymbol{u} = u_x + v_y = 0 + v_y = \frac{(v_{i,1} - v_{i,-1})}{2h} + O(h^2), \tag{1.10}$$

where we have used $\boldsymbol{u}|_\Gamma = 0$, giving $v_{i,-1} = v_{i,1} + O(h^3)$. This implies in (1.9) that one should take $v_{i,-1} = v_{i,1}$, resulting in the following approximation for the Neumann boundary condition for the PPE (1.7b):

$$\left.\frac{\partial p}{\partial \boldsymbol{n}}\right|_{(x_i,0)} = \left.\frac{\partial p}{\partial y}\right|_{(x_i,0)} = \frac{2\nu}{h^2} v_{i,1} + O(h). \tag{1.11}$$

As was the case for Thom's formula in the vorticity formulation, when coupled with a standard second-order finite difference discretization of (1.7), (1.11) results in second-order spatial accuracy of the overall scheme. This is proven in Section 2.1 via normal mode analysis of the 2D Stokes equation and further demonstrated numerically for the full NSE in Sections 3 and 4.1.

Clearly, the derivation of (1.11) for p follows the same spirit as the derivation of local vorticity boundary conditions. For this reason we refer to (1.11) as a *local pressure boundary condition*. We list, analogous to many high-order local vorticity boundary conditions, corresponding high-order local pressure boundary conditions in Appendix B. Moreover, one can easily extend this procedure to the situation in which the physical boundary slips with a prescribed velocity, which is described in Appendix C.

With (1.11), an outline of the main time loop for (1.7) and (1.8) is as follows:

Step 1: Update $\boldsymbol{u}$ in time at the *interior* grid points $\{(x_i, y_j) \mid i, j > 0\}$ via the momentum equation (1.7a), with $\boldsymbol{u}|_\Gamma = 0$.

Step 2: Compute the Neumann boundary condition, the local pressure boundary condition, along Γ using (1.11) from the $\boldsymbol{u}$ computed in Step 1.

Step 3: Solve the Poisson equation (1.7b) for p using the local pressure boundary condition from Step 2.

It is important in Step 1 that p in (1.7a) be treated *explicitly* in the time discretization to ensure that the computation of (1.7a) completely decouples from that of (1.7b). Also, for the moderate to large Reynolds number regime the viscous term should also be treated explicitly, for the same reasons outlined above in the discussion of the vorticity–stream function formulation. We will demonstrate with numerical experiments that the resulting scheme is indeed stable for large Reynolds numbers under the standard CFL constraint when a high-order time discretization is used, such as RK4.

Note that the above algorithm is efficient, with the primary computational cost per time step consisting of one Poisson solve, *irrespective* of a 2D or 3D computation. This is in fact the best that one can achieve for incompressible flows. There are additional advantages. In a finite difference setting, as explored in this paper, a nonstaggered grid can be used. However, in contrast to the popular use of a MAC grid for the primitive variables formulation, the divergence-free condition for the velocity field is no longer identically satisfied discretely. However, we will demonstrate that one can expect the divergence to be satisfied to the accuracy of the spatial discretization, which in our case is a second order. However, since a nonstaggered grid is used, generalization to more complicated systems (e.g., inclusion of temperature, density, and electromagnetic effects) is simplified. In fact, we will use the example of variable density flows in Section 4 as an illustration of the flexibility of the overall method.

We note that numerical methods have already been developed based on the PPE formulation. Kleiser and Schumann [17] first used the $\nabla \cdot \boldsymbol{u} = 0$ boundary condition to develop a class of spectral methods known as the capacity matrix method for incompressible flow. Henshaw, Kreiss, and Reyna [13] developed a fourth-order finite difference scheme based on this approach and also give a stability analysis. In addition, Henshaw [14] adapted the scheme to compute 3D flows on complex domains using overlapping grids. Gresho and Sani [10] gave an extensive review of pressure boundary conditions for the PPE formulation and identified two basic approaches: (1) enforce $\nabla \cdot \boldsymbol{u} = 0$ at the boundary or (2) include the viscous term in the source term of the PPE. The current work takes the former approach.

In [5, 9] it was shown that a standard second-order finite difference scheme in the vorticity–stream function formulation implemented with Thom's local vorticity boundary condition is equivalent to the classical MAC scheme. In Appendix D we show, when applied to a 1D model of the 2D unsteady Stokes equations in Section 2, that the local pressure boundary condition scheme outlined above is also equivalent to the MAC scheme when standard second-order finite differences are used on a staggered grid. Extension of this analysis to both a 2D and a 3D domain is straightforward, and we use the 1D model solely for clarity of presentation. Hence, we view the local pressure boundary condition approach described here as a generalization of the MAC scheme to a nonstaggered grid. We note that achieving high-order accuracy is also possible in this formulation, and we will present a spectral version of the method, as well as a finite element version, in a future paper.

We end this section with a complete outline of the use of the local pressure boundary condition (1.11) for the PPE formulation (1.7) and (1.8) in 2D. For simplicity of presentation we use forward Euler for the time stepping. Our computational examples use RK4, each stage of which can be written as a forward Euler step. The main point is that both the viscous term and the pressure are treated *explicitly*. Again refer to the representative grid shown in Fig. 1. To set notation, for a function $f(x, y)$ define the finite difference operators $\tilde{D}_x$ and D_x by

$$\tilde{D}_x f(x, y) = \frac{f(x+h, y) - f(x-h, y)}{2h},$$

$$D_x f(x, y) = \frac{f(x+h/2, y) - f(x-h/2, y)}{h},$$

with analogous definitions of $\tilde{D}_y$ and D_y. Second-order difference approximations of ∂_x and ∂_{xx} are then given by $\tilde{D}_x$ and D_x^2. The time stepping procedure is as follows:

Time Stepping: At the nth time step assume $\{\boldsymbol{u}^n_{i,j}\}$ and $\{p^n_{i,j}\}$ are given on all of Ω_h $(i, j \geq 0)$.

Step 1: Compute $\{\boldsymbol{u}^{n+1}_{i,j}\}$ at the interior grid points $(i, j \geq 1)$ using

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t} + (\boldsymbol{u}^n \cdot \nabla_h)\boldsymbol{u}^n + \nabla_h p^n = \nu \Delta_h \boldsymbol{u}^n, \tag{1.12}$$

setting $\boldsymbol{u}^{n+1}|_\Gamma = 0$. Here $\nabla_h = (\tilde{D}_x, \tilde{D}_y)^T$ and $\Delta_h = (D_x^2 + D_y^2)$ are the standard second-order centered difference approximations to ∇ and Δ, respectively.

Step 2: Recover $\{p^{n+1}_{i,j}\}$ on all of $\Omega_h(i, j \geq 0)$ by solving the PPE

$$\Delta_h p^{n+1} = 2(\tilde{D}_x u^{n+1} \tilde{D}_y v^{n+1} - \tilde{D}_y u^{n+1} \tilde{D}_x v^{n+1}), \tag{1.13}$$

using the local pressure boundary condition computed from $\{\boldsymbol{u}^{n+1}\}$, which, e.g., along $j = 0$ (say $y = 0$) is given by

$$\frac{\partial p^{n+1}}{\partial y}(x_i, 0) = \frac{2\nu}{h^2} v^{n+1}_{i,1}.$$

We note that for no-slip boundaries the right-hand side of (1.13) is identically zero along Γ, avoiding the use of any one-sided approximations.

As noted above, for moderate to large Reynolds number flows a convectively stable high-order explicit time stepping scheme (such as third-order or classical fourth-order Runge–Kutta) should be used in (1.12) in place of forward Euler to avoid any cell Reynolds number constraint; see [5]. In the case of classical fourth-order Runge–Kutta (RK4) (the choice of time stepping for all results presented in Sections 3 and 4), the overall scheme is stable, assuming a resolved smooth flow, as long as Δt satisfies (see [5])

$$\frac{\|\boldsymbol{u}\|_\infty \Delta t}{h} = \text{CFL} \leq C \qquad \text{and} \qquad \frac{\nu \Delta t}{h^2} \leq \left(\frac{1}{2}\right)^{\mathrm{D}}, \tag{1.14}$$

where D is the dimension (2 or 3), and C can be taken, e.g., as 1.5. In this case the eigenvalues of the linearized equation lie within the stability region of RK4.

Note that (1.12)–(1.14) easily extends with minor modification to the case of 3D flow, with $\boldsymbol{u} = (u, v, w)^T$, $\nabla_h = (\tilde{D}_x, \tilde{D}_y, \tilde{D}_z)^T$, and $\Delta_h = (D_x^2 + D_y^2 + D_z^2)$. Still only *one* Poisson solve per time step (or RK stage) is required, resulting in a highly efficient method for 3D computations.

2. ANALYSIS OF THE SCHEME FOR A 1D MODEL

In this section we investigate the local pressure boundary condition approach using a simplified 1D model of the unsteady 2D Stokes equations. Note that there is no difference required for the 3D Stokes equation in the following analysis. Our primary goal is to understand in this simplified setting the efficacy of discretizations of the NSE on nonstaggered grids, paying particular attention to the implementation of boundary conditions. We note that the topic of boundary conditions in numerical schemes for incompressible flow has received a great deal of attention in the literature, e.g., see [16, 19, 20].

In Appendix D we show that the local pressure boundary condition scheme outlined above, when applied to the 1D model, is exactly the classical MAC scheme when standard second-order finite differences are used on a staggered grid. Finally, in Section 2.1 we present normal mode analysis of the scheme applied to this 1D model, for in this setting the analysis becomes transparently clear.

Consider the unsteady 2D Stokes equations given by

$$\begin{cases} \boldsymbol{u}_t + \nabla p = \nu \Delta \boldsymbol{u}, \\ \nabla \cdot \boldsymbol{u} = 0, \end{cases} \tag{2.1}$$

on the domain $\Omega = [-1, 1] \times (0, 2\pi)$ with the no-slip boundary condition

$$\boldsymbol{u}|_\Gamma = 0, \tag{2.2}$$

applied at $x = -1, 1$ and periodic boundary conditions in y. Assume solutions of the form $\boldsymbol{u} = e^{iky}[u(x,t), v(x,t)]^T$ and $p = e^{iky}p(x,t)$. Then letting $\tilde{v} = iv$ and renaming (for simplicity of notation) $\tilde{v}$ as v, solutions of (2.1) and (2.2) are reduced to a family of 1D problems indexed by $k \in Z$ given by

$$\begin{cases} \partial_t u + \partial_x p = \nu\left(\partial_x^2 - k^2\right)u, \\ \partial_t v - kp = \nu\left(\partial_x^2 - k^2\right)v, \\ \partial_x u + kv = 0, \end{cases} \tag{2.3}$$

with the boundary conditions

$$u(\pm 1, t) = v(\pm 1, t) = 0. \tag{2.4}$$

The equivalent PPE formulation of (2.3) and (2.4) is given by

$$\begin{cases} \partial_t u + \partial_x p = \nu\left(\partial_x^2 - k^2\right)u, \\ \partial_t v - kp = \nu\left(\partial_x^2 - k^2\right)v, \\ \left(\partial_x^2 - k^2\right)p = 0, \end{cases} \tag{2.5}$$

along with the boundary conditions

$$u(\pm 1, t) = v(\pm 1, t) = \partial_x u(\pm 1, t) = 0. \tag{2.6}$$

The 1D linear models given in (2.3) and (2.4) and in (2.5) and (2.6) still embody the essential features of incompressibility and the viscous terms of the NSE while allowing us to analyze in a simplified setting possible finite difference discretizations. However, the main difficulty in the numerical treatment of the NSE remains, namely enforcing incompressibility and the lack of pressure boundary conditions. We first set some notation. Let a 1D nonstaggered grid Ω_h be defined by the points

$$x_j = -1 + jh, \quad h = 2/N, \quad j = 0, 1, \ldots, N,$$

and denote by u_j the approximation of u at the point x_j.

The most natural finite difference approximation of (2.3) and (2.4) is given by

$$\begin{cases} \partial_t u_j + \tilde{D}_x p_j = \nu\big(D_x^2 - k^2\big)u_j, & j = 1, 2, \ldots, N-1 \\ \partial_t v_j - k p_j = \nu\big(D_x^2 - k^2\big)v_j, & j = 1, 2, \ldots, N-1 \\ \tilde{D}_x u_j + k v_j = 0, & j = 1, 2, \ldots, N-1 \\ u_0 = v_0 = u_N = v_N = 0. \end{cases} \tag{2.7}$$

However, this discretized system is underdetermined, which is easily seen by simple counting. Indeed, the unknowns in (2.7) are given by

$$(u_j, v_j, p_j) \qquad j = 0, 1, \ldots, N,$$

totaling $3N + 3$, while the number of equations and boundary conditions in (2.7) is $3N + 1$. This naive discretization results in the well-known problem of parasitic modes in the solution. The trouble here lies solely in the particular discretization (2.7). To circumvent this problem one can use a staggered grid, which is the approach taken in the celebrated MAC scheme; see [12]. However, use of a staggered grid generally limits the application of the MAC scheme to simple geometries. We note that for finite element methods, analogous to the use of a staggered grid for finite differences, one has the Ladyzhenskaya–Babuska–Brezzi (LBB) compatibility conditions, which must be satisfied by the finite element spaces for velocity and pressure [18]. As discussed in the introduction, one may view the current approach as a generalization of the MAC scheme to a nonstaggered grid. See Appendix D.

Another way to overcome the possibility of parasitic modes is the consistent discretization of the PPE with local pressure boundary conditions. We discretize the PPE formulation (2.5) and (2.6) on the nonstaggered grid Ω_h as

$$\begin{cases} \partial_t u_j + \tilde{D}_x p_j = \nu\big(D_x^2 - k^2\big)u_j, & j = 0, 1, \ldots, N \\ \partial_t v_j - k p_j = \nu\big(D_x^2 - k^2\big)v_j, & j = 1, 2, \ldots, N-1 \\ \big(D_x^2 - k^2\big)p_j = 0, & j = 0, 1, \ldots, N \\ u_0 = v_0 = u_N = v_N = 0, & \\ \tilde{D}_x u_0 = \tilde{D}_x u_N = 0. & \end{cases} \tag{2.8a–e}$$

This gives a total of $3N + 7$ equations and boundary conditions. The unknowns here are

$$(u_j, p_j), \quad j = -1, 0, \ldots, N+1 \qquad \text{and} \qquad v_j, \quad j = 0, 1, \ldots, N,$$

also totaling $3N + 7$. This system is consistent, at least in the sense that there is the same number of unknowns as equations and boundary conditions combined. However, note that values for both u and p are required at the ghost points $j = -1, N + 1$. For u this can be resolved using (2.8e). However, just as for (1.7b) the true difficulty is that a consistent boundary condition for p is needed in a time stepping procedure for (2.8). Applying the boundary condition $u_0 = u_N = 0$, Eq. (2.8a) for $j = 0, N$ reads

$$\tilde{D}_x p_0 = \nu\big(D_x^2 - k^2\big)u_0, \quad \tilde{D}_x p_N = \nu\big(D_x^2 - k^2\big)u_N. \tag{2.9}$$

Using $\tilde{D}_x u_0 = \tilde{D}_x u_N = 0$ (compare to (1.10)), we eliminate the unknowns u_{-1} and u_{N+1} in (2.9), giving numerical boundary conditions for p

$$\tilde{D}_x p_0 = \frac{2\nu}{h^2} u_1, \quad \tilde{D}_x p_N = \frac{2\nu}{h^2} u_{N-1}. \tag{2.10}$$

Clearly, the derivation of (2.10) is the same as the local pressure boundary condition (1.11) for the 1D model (2.3).

With this, the system (2.8) is equivalent to

$$\begin{cases} \partial_t u_j + \tilde{D}_x p_j = \nu\big(D_x^2 - k^2\big)u_j, & j = 1, 2, \ldots, N-1 \\ \partial_t v_j - kp_j = \nu\big(D_x^2 - k^2\big)v_j, & j = 1, 2, \ldots, N-1 \\ u_0 = v_0 = u_N = v_N = 0, & \\ \big(D_x^2 - k^2\big)p_j = 0, & j = 0, 1, \ldots, N \\ \tilde{D}_x p_0 = \dfrac{2\nu}{h^2} u_1, \quad \tilde{D}_x p_N = \dfrac{2\nu}{h^2} u_{N-1}, & \end{cases} \tag{2.11}$$

a total of $3N + 5$ equations and $3N + 5$ unknowns. This is our second-order spatial discretization of the PPE formulation for the 1D model. The important point is that we now have a local pressure boundary condition computed from the velocity field.

2.1. *Normal Mode Analysis*

In this section we demonstrate the stability and second-order accuracy of the local pressure boundary condition approach applied to the PPE discretization of the 1D model (2.1) and (2.2) of the unsteady 2D Stokes equations using Godunov–Ryabenki (normal mode) analysis. The lemmas referred to in this section can be found in Appendix E. The normal mode solutions of (2.5) take the form

$$(u, v, p)(x, t) = e^{\sigma t}(\hat{u}, \hat{v}, \hat{p})(x), \tag{2.12}$$

and we take σ to be of the form $\sigma = -\nu(k^2 + \mu^2)$, with conditions on μ to be determined later. Plugging (2.12) into the continuous PPE formulation (2.5) gives

$$\begin{cases} \nu\big(\partial_x^2 + \mu^2\big)u = \partial_x p, \\ \nu\big(\partial_x^2 + \mu^2\big)v = -kp, \\ \big(\partial_x^2 - k^2\big)p = 0, \end{cases} \tag{2.13}$$

where for simplicity we have dropped the ˆ symbol. The boundary conditions for (2.13) are given by

$$u(\pm 1) = v(\pm 1) = \partial_x u(\pm 1) = 0. \tag{2.14}$$

There are two families of solutions, odd and even. The analysis of each is essentially the same and we focus on the former. The odd solutions of (2.13) take the form

$$\begin{cases} p(x) = \sinh(kx), \\ u(x) = A\left(\frac{\cos\mu x}{\cos\mu} - \frac{\cosh kx}{\cosh k}\right), \\ v(x) = B\left(\frac{\sin\mu x}{\sin\mu} - \frac{\sinh kx}{\sinh k}\right). \end{cases} \tag{2.15}$$

Clearly the first two boundary conditions in (2.14), namely

$$u(\pm 1) = v(\pm 1) = 0,$$

are satisfied by u and v in (2.15). By plugging (2.15) into (2.13) we easily see that

$$A = \frac{k\cosh k}{\sigma}, \quad B = -\frac{k\sinh k}{\sigma}. \tag{2.16}$$

Since

$$\partial_x u(x) = -A\left(\mu\frac{\sin\mu x}{\cos\mu} + k\frac{\sinh kx}{\cosh k}\right), \tag{2.17}$$

in order to satisfy the boundary condition

$$\partial_x u(\pm 1) = 0,$$

we must have that

$$\mu\tan\mu + k\tanh k = 0. \tag{2.18}$$

Equation (2.18) has a unique real solution μ in $(-\pi/2, \pi/2) + \ell\pi$, $\ell = 0, \pm 1, \pm 2, \ldots$. Hence $\sigma = -\nu(k^2 + \mu^2) < 0$, and note that these eigenmodes are complete. Finally, we show in the PPE formulation that the incompressibility condition is satisfied. From (2.15), (2.16), and (2.18) we have

$$\begin{aligned} \partial_x u + kv &= -\frac{k\cosh k}{\sigma}\left(\mu\frac{\sin\mu x}{\cos\mu} + k\frac{\sinh kx}{\cosh k}\right) - \frac{k^2\sinh k}{\sigma}\left(\frac{\sin\mu x}{\sin\mu} - \frac{\sinh kx}{\sinh k}\right) \\ &= -\frac{k}{\sigma}\frac{\cosh k}{\sin\mu}(\mu\tan\mu + k\tanh k)\sin\mu x = 0. \end{aligned} \tag{2.19}$$

We now carry out the analogous calculations for the finite difference discretization of the PPE formulation (2.8). Assuming normal mode solutions of the form

$$(u, v, p)(x, t) = e^{\tilde{\sigma}t}(\hat{u}, \hat{v}, \hat{p})(x), \tag{2.20}$$

and plugging (2.20) into (2.8) gives (where again we have dropped the ˆ symbol for simplicity of presentation)

$$\begin{cases} \left(D_x^2 - k^2 - \frac{\tilde{\sigma}}{\nu}\right)u = \frac{1}{\nu}\tilde{D}_x p, \\ \left(D_x^2 - k^2 - \frac{\tilde{\sigma}}{\nu}\right)v = -\frac{k}{\nu}p, \\ \left(D_x^2 - k^2\right)p = 0. \end{cases} \tag{2.21a–c}$$

The boundary conditions are now given by

$$u(\pm 1) = v(\pm 1) = \tilde{D}_x u(\pm 1) = 0. \tag{2.22}$$

The odd solutions of (2.21c) take the form

$$p(x) = \sinh(\tilde{k}x),$$

where $\tilde{k}$ is given implicitly by

$$\sinh\left(\frac{\tilde{k}h}{2}\right) = \frac{kh}{2}, \tag{2.23}$$

where, recall, h is the grid spacing. Following (2.15), we take the solutions u and v to be of the form

$$\begin{cases} u(x) = \tilde{A}\left(\frac{\cos\tilde{\mu}x}{\cos\tilde{\mu}} - \frac{\cosh\tilde{k}x}{\cosh\tilde{k}}\right), \\ v(x) = \tilde{B}\left(\frac{\sin\tilde{\mu}x}{\sin\tilde{\mu}} - \frac{\sinh\tilde{k}x}{\sinh\tilde{k}}\right). \end{cases} \tag{2.24}$$

As before, u and v in (2.24) satisfy the boundary conditions

$$u(\pm 1) = v(\pm 1) = 0.$$

A simple computation shows that the first terms of u and v in (2.24) solve, respectively, (2.21a–b) with homogeneous right-hand sides. From Lemma 1 we have

$$\left(D_x^2 - k^2 - \frac{\tilde{\sigma}}{\nu}\right)\cos\tilde{\mu}x = \left(-\frac{4}{h^2}\sin^2\left(\frac{\tilde{\mu}h}{2}\right) - k^2 - \frac{\tilde{\sigma}}{\nu}\right)\cos\tilde{\mu}x.$$

which leads to the following relation between $\tilde{\mu}$ and $\tilde{\sigma}$

$$\tilde{\sigma} = -\nu\left(k^2 + \frac{4}{h^2}\sin^2\left(\frac{\tilde{\mu}h}{2}\right)\right). \tag{2.25}$$

Hence,

$$\left(D_x^2 - k^2 - \frac{\tilde{\sigma}}{\nu}\right)\cos\tilde{\mu}x = 0. \tag{2.26}$$

Furthermore, the second terms of u and v in (2.24) solve, respectively, the inhomogeneous right-hand sides of (2.21a–b). Indeed, plugging (2.24) into (2.21) gives

$$\left(D_x^2 - k^2 - \frac{\tilde{\sigma}}{\nu}\right)u(x) = \frac{\tilde{A}\tilde{\sigma}}{\nu\cosh\tilde{k}}\cosh\tilde{k}x = \frac{\tilde{A}\tilde{\sigma}}{\nu\cosh\tilde{k}}\frac{h}{\sinh(\tilde{k}h)}\tilde{D}_x\sinh\tilde{k}x,$$

and thus setting

$$\tilde{A} = \frac{\sinh(\tilde{k}h)}{h}\frac{\cosh\tilde{k}}{\tilde{\sigma}}, \tag{2.27}$$

we then have

$$\left(D_x^2 - k^2 - \frac{\tilde{\sigma}}{\nu}\right)u(x) = \frac{1}{\nu}\tilde{D}_x p(x). \tag{2.28}$$

Similarly

$$\left(D_x^2 - k^2 - \frac{\tilde{\sigma}}{\nu}\right)v(x) = \frac{\tilde{B}\tilde{\sigma}}{\nu \sinh \tilde{k}} \sinh \tilde{k}x,$$

and setting

$$\tilde{B} = -\frac{k \sinh \tilde{k}}{\tilde{\sigma}}, \tag{2.29}$$

we have

$$\left(D_x^2 - k^2 - \frac{\tilde{\sigma}}{\nu}\right)v(x) = -\frac{k}{\nu}p(x). \tag{2.30}$$

Finally, plugging these back into (2.24) gives

$$\begin{cases} p(x) = \sinh(\tilde{k}x), \\ u(x) = \frac{\sinh(\tilde{k}h)}{h}\frac{\cosh \tilde{k}}{\tilde{\sigma}}\left(\frac{\cos \tilde{\mu}x}{\cos \tilde{\mu}} - \frac{\cosh \tilde{k}x}{\cosh \tilde{k}}\right), \\ v(x) = -\frac{k \sinh \tilde{k}}{\tilde{\sigma}}\left(\frac{\sin \tilde{\mu}x}{\sin \tilde{\mu}} - \frac{\sinh \tilde{k}x}{\sinh \tilde{k}}\right). \end{cases} \tag{2.31}$$

Similar to the continuous case, the boundary condition,

$$\tilde{D}_x u(\pm 1) = 0,$$

is used to determine the value of $\tilde{\mu}$. Since,

$$\tilde{D}_x u(x) = \tilde{A}\left(\frac{\sin \tilde{\mu}h}{h}\frac{\sin \tilde{\mu}x}{\cos \tilde{\mu}} + \frac{\sinh \tilde{k}h}{h}\frac{\sinh \tilde{k}x}{\cosh \tilde{k}}\right), \tag{2.32}$$

we need to set

$$\frac{\sin(\tilde{\mu}h)}{h}\tan \tilde{\mu} + \frac{\sinh(\tilde{k}h)}{h}\tanh \tilde{k} = 0. \tag{2.33}$$

As was the case for (2.18), there is a unique real solution $\tilde{\mu}$ of (2.33) in $(-\pi/2, \pi/2) + \ell\pi$, $\ell = 0, \pm 1, \pm 2, \ldots$. Hence by (2.25) all the eigenvalues $\tilde{\sigma}$ are real and negative, indicating stability for $\nu > 0$. See [7] for the convergence analysis of the projection method in this setting.

Next we show that the scheme is indeed spatially second-order accurate. Using Lemma 2 and (2.23) it follows that

$$\tilde{k} = k + O(h^2). \tag{2.34}$$

By Lemma 2 and (2.34), we have from (2.33) that

$$\tilde{\mu}\tan\tilde{\mu} + k\tanh k = O(h^2). \tag{2.35}$$

Comparing this with (2.18), we have

$$\tilde{\mu} = \mu + O(h^2).$$

Next, plugging $\tilde{\mu}$ into (2.25) gives

$$\tilde{\sigma} = k^2 + \tilde{\mu}^2 + O(h^2) = k^2 + \mu^2 + O(h^2) = \sigma + O(h^2).$$

Finally, we show that the incompressibility condition is satisfied to second-order accuracy. First note that

$$\begin{aligned}\tilde{D}_x u + kv = &-\frac{\sinh(\tilde{k}h)}{h}\frac{\cosh\tilde{k}}{\tilde{\sigma}}\left(\frac{\sin\tilde{\mu}h}{h}\frac{\sin\tilde{\mu}x}{\cos\tilde{\mu}} + \frac{\sinh\tilde{k}h}{h}\frac{\sinh\tilde{k}x}{\cosh\tilde{k}}\right)\\ &-\frac{k^2\sinh\tilde{k}}{\tilde{\sigma}}\left(\frac{\sin\tilde{\mu}x}{\sin\tilde{\mu}} - \frac{\sinh\tilde{k}x}{\sinh\tilde{k}}\right).\end{aligned} \tag{2.36}$$

Then using (2.33), we have

$$\begin{aligned}\tilde{D}_x u + kv = &-\frac{\sinh(\tilde{k}h)}{h}\frac{\cosh\tilde{k}}{\tilde{\sigma}}\left(-\frac{\sinh(\tilde{k}h)}{h}\tanh\tilde{k}\frac{\sin\tilde{\mu}h}{h}\frac{\sin\tilde{\mu}x}{\sin\tilde{\mu}} + \frac{\sinh\tilde{k}h}{h}\frac{\sinh\tilde{k}x}{\cosh\tilde{k}}\right)\\ &-\frac{k^2\sinh\tilde{k}}{\tilde{\sigma}}\left(\frac{\sin\tilde{\mu}x}{\sin\tilde{\mu}} - \frac{\sinh\tilde{k}x}{\sinh\tilde{k}}\right)\\ = &\ \frac{1}{\tilde{\sigma}}\left(\left(\frac{\sinh\tilde{k}h}{h}\right)^2 - k^2\right)\left(\sinh\tilde{k}\frac{\sin\tilde{\mu}x}{\sin\tilde{\mu}} - \sinh(\tilde{k}x)\right) = O(h^2).\end{aligned} \tag{2.37}$$

Summarizing, we have shown that

$$\begin{aligned}(\tilde{k}, \tilde{\mu}, \tilde{\sigma}) &= (k, \mu, \sigma) + O(h^2),\\ \tilde{D}_x\hat{u} + k\hat{v} &= O(h^2),\end{aligned}$$

which indicates that the solutions in (2.31) are second-order-accurate approximations of (2.15) and (2.16). Since it was shown in Section 2 that the discretized systems (2.8) and (2.11) are algebraically equivalent, all of the above conclusions hold for the local pressure boundary condition scheme (2.11).

3. NUMERICAL RESULTS

In this section we demonstrate the rate of convergence of the scheme, as outlined in Section 1, applied to both 2D and 3D test flows using accuracy checks. Additionally, we present results of the scheme applied to the computation of the flow past a cylinder. For this flow, a comparison of the computations is made with both second- and fourth-order vorticity–stream function based schemes. We note that a rectangular grid is used for all computations presented in this section, and hence we solve the discrete Poisson equations that arise using cosine transforms computed by FFT methods.

Example 1 (2D accuracy check). We first present an accuracy check of the scheme (1.12) and (1.13) of a computation on $\Omega = [0, \pi] \times [0, \pi]$. We have taken the exact solution of the 2D NSE as

$$\begin{cases} u(x, y, t) = -\sin t \sin^2 x \sin y \cos y, \\ v(x, y, t) = \sin t \sin x \cos x \sin^2 y, \\ p(x, y, t) = \sin t \cos x \sin y. \end{cases} \tag{3.1}$$

To ensure that (3.1) is an exact solution of (1.7) appropriate forcing functions are applied to the system. Note that $\boldsymbol{u}$ in (3.1) satisfies (1.8) for any t. We took $Re = \pi/\nu = 500$ and computed solutions for various values of $N = \pi/h$ until time $t = 3.0$. Fourth-order Runge–Kutta time stepping was used and CFL $= 1.0$, with Δt determined by (1.14). Table I shows the absolute errors between the numerical solutions and the exact solutions as well as the divergence of the computed velocity field. As the grid is refined the method achieves second-order accuracy for both $\boldsymbol{u}$ and p as well as the divergence.

Example 2 (3D accuracy check). Similar to the 2D case, we take the exact solution to be

$$\begin{cases} u(x, y, t) = \cos t \sin^2 x(\sin 2y \sin^2 z - \sin^2 y \sin 2z), \\ v(x, y, t) = \cos t \sin^2 y(\sin^2 x \sin 2z - \sin 2x \sin^2 z), \\ w(x, y, t) = \cos t \sin^2 z(\sin 2x \sin^2 y - \sin^2 x \sin 2y), \\ p(x, y, t) = \cos t \cos x \sin y \cos z, \end{cases} \tag{3.2}$$

on a domain $\Omega = [0, \pi]^3$ and again add forcing functions to ensure (3.2) is an exact solution of (1.7). We took $Re = \pi/\nu = 500$ and computed solutions for various values of $N = \pi/h$

TABLE I
Absolute Errors at Time t = 3.0 for 2D Accuracy Check (Re = 500)

	N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
div $\boldsymbol{u}$	32	2.80e-02		1.34e-02		1.61e-02	
	64	7.18e-03	1.96	3.48e-03	1.95	4.40e-03	1.88
	128	1.80e-03	1.99	8.80e-04	1.99	1.13e-03	1.96
	256	4.52e-04	2.00	2.20e-04	2.00	2.83e-04	1.99
u	32	1.30e-02		6.04e-03		6.43e-03	
	64	3.27e-03	1.99	1.53e-03	1.98	1.67e-03	1.94
	128	8.20e-04	2.00	3.85e-04	1.99	4.22e-04	1.99
	256	2.05e-04	2.00	9.62e-05	2.00	1.06e-04	2.00
v	32	1.09e-02		4.96e-03		5.18e-03	
	64	2.78e-03	1.97	1.27e-03	1.96	1.35e-03	1.94
	128	6.98e-04	1.99	3.20e-04	1.99	3.40e-04	1.99
	256	1.75e-04	2.00	8.02e-05	2.00	8.53e-05	2.00
p	32	5.07e-03		1.71e-03		8.71e-04	
	64	1.37e-03	1.89	4.60e-04	1.89	2.30e-04	1.92
	128	3.50e-04	1.97	1.17e-04	1.98	5.83e-05	1.98
	256	8.78e-05	1.99	2.93e-05	2.00	1.46e-05	2.00

TABLE II

Absolute Errors at Time $t = 2.0$ for 3D Accuracy Check ($Re = 500$)

	N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
div $\boldsymbol{u}$	16	1.82e-01		6.44e-02		7.90e-02	
	32	5.01e-02	1.86	1.83e-02	1.81	2.47e-02	1.68
	64	1.28e-02	1.97	4.73e-03	1.95	6.59e-03	1.90
	128	3.23e-03	1.99	1.19e-03	1.99	1.68e-03	1.97
u	16	5.08e-02		1.55e-02		1.40e-02	
	32	1.38e-02	1.88	4.18e-03	1.89	3.92e-03	1.84
	64	3.51e-03	1.97	1.07e-03	1.97	1.02e-03	1.95
	128	8.83e-04	1.99	2.68e-04	1.99	2.56e-04	1.99
v	16	5.10e-02		1.57e-02		1.38e-02	
	32	1.40e-02	1.87	4.25e-03	1.88	3.91e-03	1.82
	64	3.57e-03	1.97	1.09e-03	1.97	1.01e-03	1.95
	128	8.98e-04	1.99	2.73e-04	1.99	2.56e-04	1.98
w	16	5.13e-02		1.58e-02		1.45e-02	
	32	1.39e-02	1.88	4.27e-03	1.89	4.04e-03	1.84
	64	3.56e-03	1.97	1.09e-03	1.97	1.05e-03	1.94
	128	8.94e-04	1.99	2.74e-04	1.99	2.65e-04	1.99
p	16	2.19e-02		4.56e-03		1.68e-03	
	32	4.04e-03	2.44	8.54e-04	2.42	4.12e-04	2.03
	64	9.92e-04	2.03	2.23e-04	1.94	1.14e-04	1.85
	128	2.50e-04	1.99	5.68e-05	1.97	2.92e-05	1.96

until time $t = 2.0$. Fourth-order Runge–Kutta time stepping was used and CFL $= 1.0$, with Δt determined by (1.14). Table II shows the absolute errors between the numerical solutions and the exact solutions as well as the divergence of the computed velocity field. As the grid is refined the method achieves clean second-order accuracy for both $\boldsymbol{u}$ and p as well as the divergence. We also show in Fig. 2 the error in the pressure at $z = \pi/2$ for $N = 128$, indicating a smooth profile of order $O(10^{-5})$, with no oscillations at the boundary.

Example 3 (flow past a cylinder). We next applied the scheme to the computation of the flow around a unit ($R = 1$) circular cylinder in the plane. The geometry of the problem naturally dictates the use of the polar PPE formulation of the NSE, to which we further apply the transformation

$$z = \log r, \tag{3.3}$$

exponentially stretching the grid radially, thus automatically concentrating computational points in the viscous boundary layer. The governing equations, written using the radially scaled velocity components defined by

$$\boldsymbol{u} = (u, v)^{\mathrm{T}} = r(U, V)^{\mathrm{T}} = e^{z}(U, V)^{\mathrm{T}}, \tag{3.4}$$

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

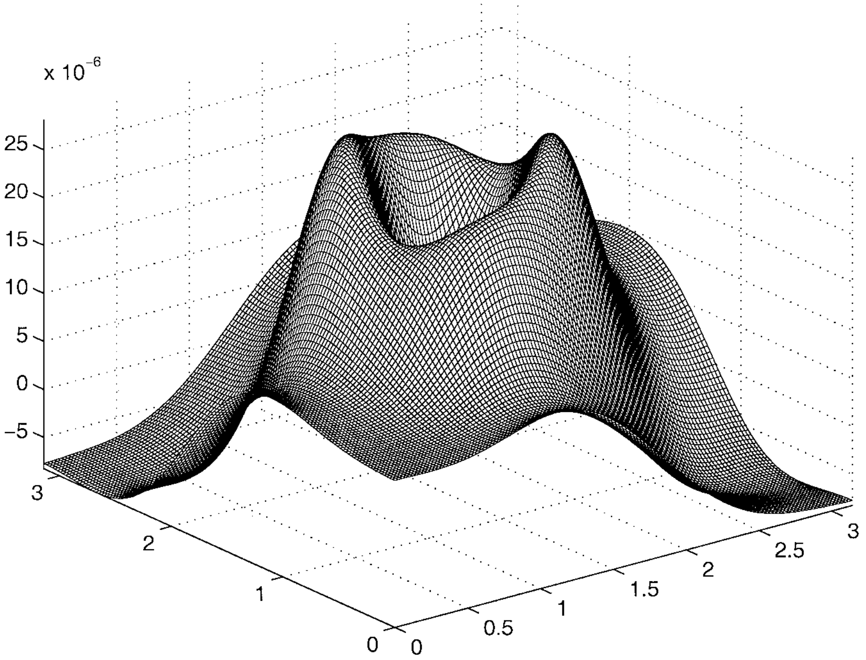


FIG. 2. Error in pressure along $z = \pi/2$ at time $t = 2.0$ for the 3D accuracy check. Grid size is $N = 128$.

$$\begin{bmatrix} u \\ v \end{bmatrix}_t + \frac{1}{e^{2z}} \begin{bmatrix} uu_z + vu_\theta - (u^2 + v^2) \\ uv_z + vv_\theta \end{bmatrix} + \begin{bmatrix} p_z \\ p_\theta \end{bmatrix} = \frac{\nu}{e^{2z}} \begin{bmatrix} \triangle_{(z.\theta)} u - 2(u_z + v_\theta) \\ \triangle_{(z.\theta)} v - 2(v_z + u_\theta) \end{bmatrix}, \qquad \text{(3.5a–b)}$$

and

$$\triangle_{(z,\theta)} p = -\frac{2}{e^{2z}} (u^2 + v^2 + uv_\theta + u_\theta v_z - u_z v_\theta - u_\theta v - vv_z - uu_z). \qquad (3.6)$$

Note that the divergence-free condition in the (z, θ) variables is given by $\nabla_{(z,\theta)} \cdot \boldsymbol{u} = (u_z + v_\theta) = 0$, and thus we have set the last term in (3.5a) equal to zero in the computations. Boundary conditions for the velocity field $\boldsymbol{u}$ at the cylinder surface $\Gamma (r = 1$ or $z = 0)$ are given by

$$\boldsymbol{u}|_\Gamma = 0 \qquad \text{and} \qquad \nabla_{(z,\theta)} \cdot \mathbf{u}|_\Gamma = (u_z + v_\theta) = 0. \qquad (3.7)$$

The no-slip condition is applied to the velocity field, while the divergence-free condition is used to derive the corresponding local pressure boundary condition for the PPE (3.6). In this setting we have

$$p_z = \nu u_{zz} \qquad \text{and} \qquad u_z = 0 \qquad \text{along } \Gamma, \qquad (3.8)$$

which follows from dotting the radial component of (3.5) with the unit normal along Γ and

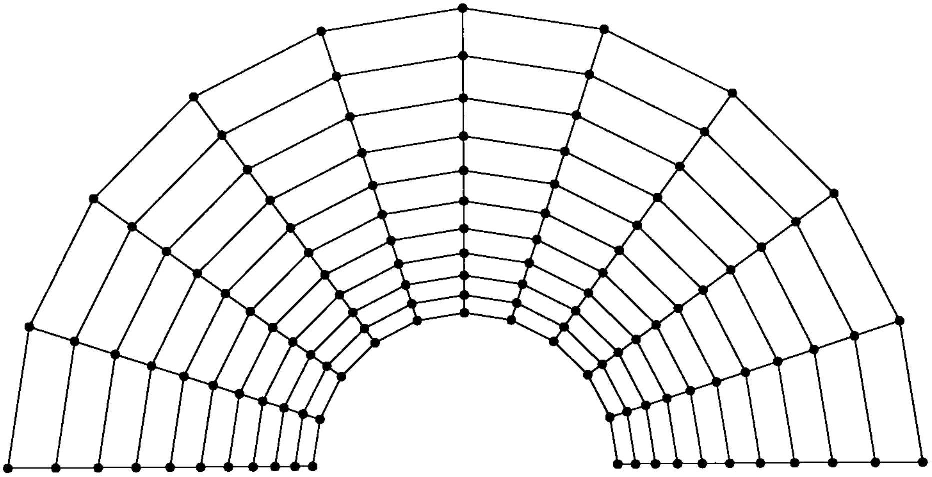


FIG. 3. A representative exponentially stretched (radially) polar grid Ω_h for the flow past a cylinder.

the boundary conditions in (3.7). Then mimicking the derivation of (1.11), (3.8) leads to

$$p_z|_\Gamma = \frac{2\nu}{(\Delta z)^2} u_{1,j}. \tag{3.9}$$

Here, $u_{1,j}$ denotes the value of u one grid point from the boundary and at the jth grid point in the angular direction.

Both the momentum equation (3.5) and the PPE (3.6) were discretized using second-order centered finite difference approximations. Symmetry of the flow was assumed, with the computational grid Ω_h(with $\Delta z = \log 16/N$ and $\Delta\theta = \pi/N$) extending to a radius of $z = \log 16(r = 16)$. A representative grid is shown in Fig. 3. 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

$$(u, v) = ((r_{\max} - 1/r_{\max})\cos\theta, -(r_{\max} + 1/r_{\max})\sin\theta). \tag{3.10}$$

with p_z determined at the far field by substitution of (3.10) into (3.5a). The Reynolds number was taken as $Re = 550 = (2R)/\nu = 2/\nu$ and solutions computed until $t = 3.0$ with four grid resolutions: $N = 128$, 256, 512, and 1024. Fourth-order Runge–Kutta time stepping was used and CFL = 1.0, with Δt determined by (1.14), with $h = \min\{\Delta z, \Delta\theta\}$. Table III shows the errors in $\boldsymbol{u}$ and p relative to the computation on the finest grid ($N = 1024$), indicating more or less second-order convergence. Table IV shows the numerical convergence for divergence of velocity field on each grid, which shows second-order convergence.

To further gauge the ability of the scheme to accurately compute this flow we compared the results above with computations of the polar form of the (ω, ψ) formulation (1.3) and (1.4). Again we applied the transformation (3.3), in which case the governing equations are given by (with $\boldsymbol{u}$ defined as in (3.4))

$$\begin{cases} e^{2z}\omega_t + (u\omega)_z + (v\omega)_\theta = \nu\Delta_{(z,\theta)}\omega, \\ \Delta\psi = -e^{2z}\omega, \end{cases} \tag{3.11}$$

TABLE III
Relative Errors at Time t = 3.0 for Cylinder Flow (Re = 550)

	n_x	L^1 error	Order	L^2 error	Order	L^∞ error	Order
u	128	3.06e-02		3.83e-02		1.43e-01	
	256	8.49e-03	1.85	1.05e-02	1.86	4.03e-02	1.83
	512	1.78e-03	2.25	2.21e-03	2.25	8.43e-03	2.26
v	128	1.24e-02		2.93e-02		1.58e-01	
	256	3.65e-03	1.76	8.97e-03	1.71	4.83e-02	1.71
	512	7.83e-04	2.22	1.94e-03	2.21	1.04e-02	2.21
p	128	1.04e-01		1.30e-01		2.26e-01	
	256	2.85e-02	1.86	3.56e-02	1.87	6.22e-02	1.86
	512	5.96e-03	2.26	7.45e-03	2.26	1.30e-02	2.26

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

$$\psi|_\Gamma = 0 \qquad \text{and} \qquad \left.\frac{\partial\psi}{\partial\boldsymbol{n}}\right|_\Gamma = 0. \tag{3.12}$$

Both second- and fourth-order spatial discretizations of (3.11) and (3.12) implemented using local vorticity boundary conditions were used to compute the flow. The same parameters and computational grids were used as in the $(\boldsymbol{u}, p)$ computations above. In the (ω, ψ) formulation the potential boundary condition (3.10) is enforced by the appropriate prescription of ψ at the far-field computational boundary. Figure 4 shows the vorticity contours (levels $-12:1:12$) for $N = 1024$, and all three computations are seen to be in excellent agreement. However, a much more sensitive comparison can be made using the computed measurement of the coefficient of total drag (C_D). In the vorticity–stream formulation, C_D may be computed using

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

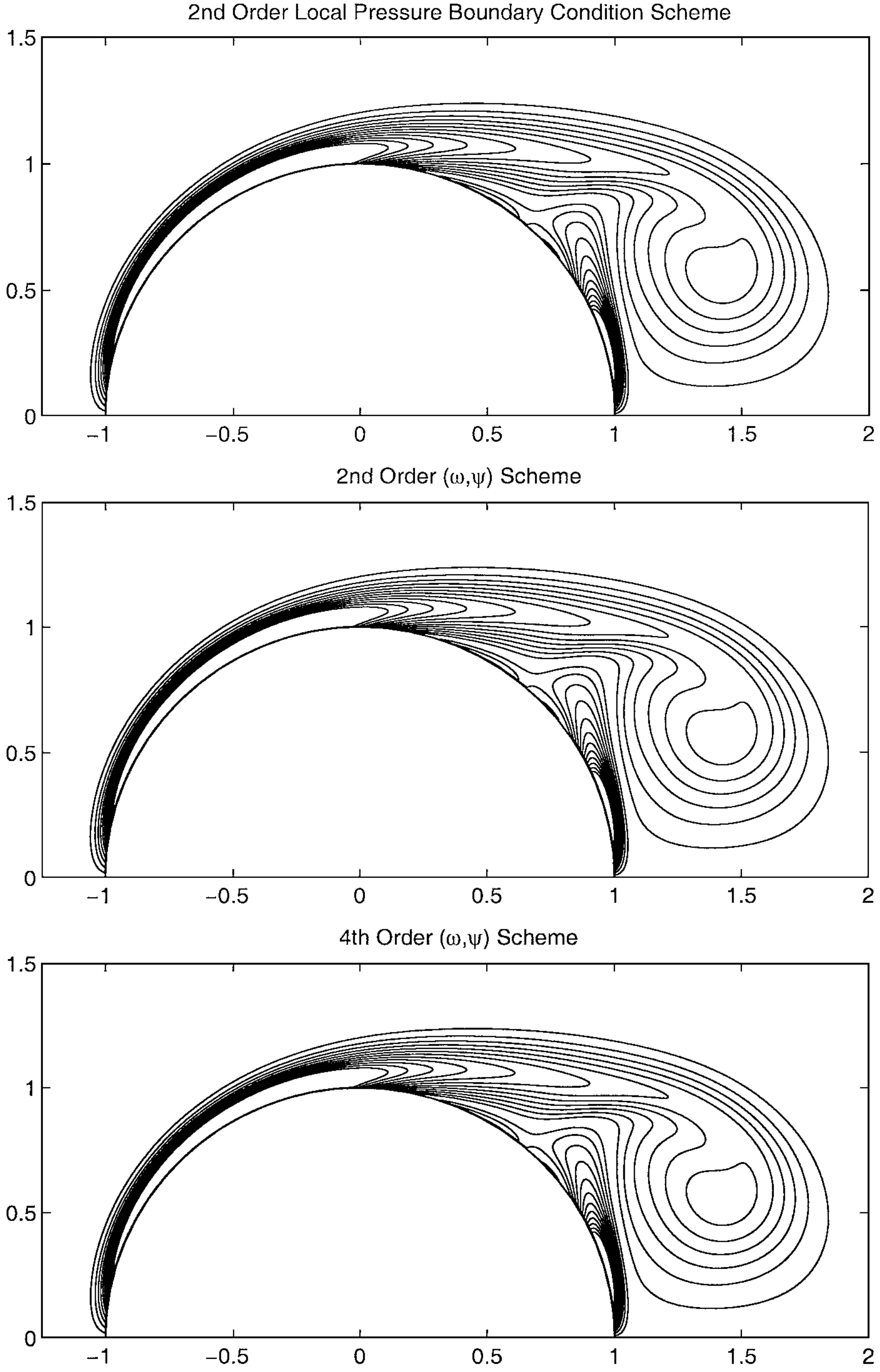
and in the velocity–pressure formulation by

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

The first image in Fig. 5 shows the computation of C_D from the local pressure boundary condition scheme. The initial singular nature of the flow at $t = 0$ in clearly indicated,

TABLE IV
Divergence Errors for Cylinder Flow at Time t = 3.0 (Re = 550)

n_x	L^1 error	Order	L^2 error	Order	L^∞ error	Order
128	4.07e-01		2.42e-01		7.03e-01	
256	1.14e-01	1.84	6.27e-02	1.95	1.24e-01	2.50
512	2.95e-02	1.95	1.59e-02	1.98	2.88e-02	2.11
1024	7.53e-03	1.97	4.02e-03	1.99	7.14e-03	2.01



2nd Order Local Pressure Boundary Condition Scheme

2nd Order (ω,ψ) Scheme

4th Order (ω,ψ) Scheme

FIG. 4. Vorticity contours of the flow past a cylinder for $Re = 550$ at time $t = 3.0$. Contour levels are $(-12:1:12)$, excluding 0.

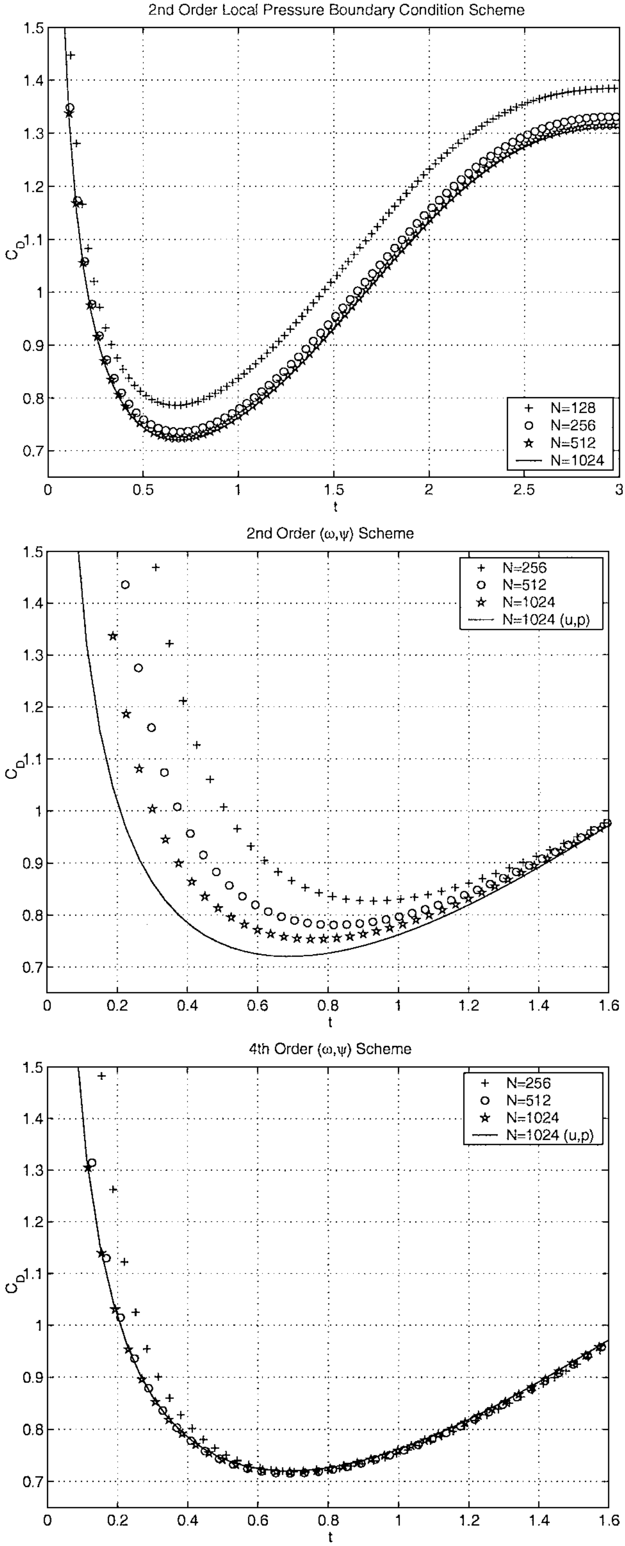


FIG. 5. Comparison of total drag for flow past a cylinder.

with the results of the computations on the three finest grids in very good agreement. Of particular interest in Fig. 5 is the comparison between the second- and fourth-order (ω, ψ) computations of C_D with the $(\boldsymbol{u}, p)$ scheme at the finest grid resolution. The comparison is only plotted up to $t \approx 1.6$ since all results are in good agreement after this time. Note that even for $N = 1024$ the second-order (ω, ψ) computation does not resolve the initial flow as well as the $(\boldsymbol{u}, p)$ scheme. We believe the $(\boldsymbol{u}, p)$ scheme is accurate because of the excellent agreement with the fourth-order (ω, ψ) computation. 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 the local pressure boundary condition, as well as the overall scheme, in recovering the pressure when solving the PPE.

4. EXTENSION OF LOCAL PRESSURE BOUNDARY CONDITION SCHEME TO VARIABLE DENSITY FLOWS

In this section we outline the extension of the local pressure boundary condition scheme for flows governed by the incompressible NSE with finite-amplitude density variations. Once again, the key is an accurate approximation of the Neumann boundary condition for the derived PPE. However, as will become clear below, variable density is easily synthesized into the current framework.

For simplicity, we focus on 2D flows ($\Omega \subset \mathbb{R}^2$) but note that extension to 3D is straightforward. The governing flow equations now take the form

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \frac{1}{\rho}\nabla p = \nu(\rho)\Delta \boldsymbol{u} + \mathbf{F}, \\ \nabla \cdot \boldsymbol{u} = 0, \\ \rho_t + \boldsymbol{u} \cdot \nabla \rho = 0, \end{cases} \tag{4.1a–c}$$

where ρ is the density and $\mathbf{F}$ represents any external forces.

Again consider the simplest physical boundary condition for $\boldsymbol{u}$, the no-slip condition

$$\boldsymbol{u}|_\Gamma = 0, \tag{4.2}$$

where $\Gamma = \partial\Omega$. Note that in (4.1a) the viscosity ν has been taken as a function of ρ. The equivalent PPE formulation of (4.1) and (4.2) on which our finite difference scheme is based is given by

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \frac{1}{\rho}\nabla p = \nu(\rho)\Delta \boldsymbol{u} + \mathbf{F}, \\ \nabla \cdot \left(\frac{1}{\rho}\nabla p\right) = \nabla \cdot (-(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nu(\rho)\Delta \boldsymbol{u} + \mathbf{F}), \\ \rho_t + \boldsymbol{u} \cdot \nabla \rho = 0, \end{cases} \tag{4.3a–c}$$

with boundary conditions

$$\boldsymbol{u}|_\Gamma = 0 \qquad \text{and} \qquad \nabla \cdot \boldsymbol{u}|_\Gamma = 0. \tag{4.4}$$

As was the case for (1.7) and (1.8) (essentially (4.3) and (4.4) with constant density $\rho = 1$) the main difficulty in solving (4.3) and (4.4) numerically is the lack of an accurate computable boundary condition for the PPE (4.3b). Furthermore, the variable density results in a variable coefficient PPE, and additional care must be taken when solving the hyperbolic continuity equation (4.3c).

As before, denote by Ω_h a finite difference grid for Ω, which without loss of generality, has equal spacing h in each coordinate direction. See Fig. 1. Also, the additional finite difference operators D_x^+ and D_x^-, defined by

$$(D_x^+)f(x,y) = \frac{f(x+h,y)-f(x,y)}{h}, \qquad (D_x^-)f(x,y) = \frac{f(x,y)-f(x-h,y)}{h},$$

are used below, with D_y^+ and D_y^- defined in the obvious manner. We refer the reader to Section 1 for the definitions of the previously defined difference operators $\tilde{D}$ and D.

Recalling the derivation of the local approximation (1.11) for the Neumann boundary condition, we see that (1.9) and (1.10) are essentially unchanged except that now we must account for the force term $\mathbf{F}$ and also that in (4.3a) ν is taken as a function of ρ. Hence, in the variable density setting the local pressure boundary condition for the PPE (4.3b) is given by (along Γ_x)

$$\left.\frac{\partial p}{\partial y}\right|_{(x_i,0)} = \rho_{i,0}\left(\frac{2\nu(\rho_{i,0})v_{i,1}}{h^2} + \mathbf{F}_{i,0}\cdot\begin{bmatrix}0\\1\end{bmatrix}\right), \tag{4.5}$$

where the premultiplication by $\rho_{i.0}$ is due to the $1/\rho$ term multiplying the pressure gradient in the momentum equation.

Remark. We can similarly carry out the derivation of our numerical scheme for the case in which the viscosity term is written in conservative form as

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \frac{1}{\rho}\nabla p = \frac{1}{\rho}\nabla\cdot(\mu(\rho)\nabla\boldsymbol{u}) + \mathbf{F}, \\ \nabla\cdot\boldsymbol{u} = 0, \\ \rho_t + \boldsymbol{u}\cdot\nabla\rho = 0, \end{cases} \tag{4.1a$'$–c$'$}$$

where μ is the dynamical viscosity, which can also depend on the temperature when thermal effects are considered. Since the viscosity term is treated explicitly, there is no real difference in the two implementations. The corresponding local pressure formula becomes

$$\left.\frac{\partial p}{\partial y}\right|_{(x_i,0)} = \left(\frac{\mu(\rho_{i,1/2})+\mu(\rho_{i,-1/2})}{h^2}v_{i,1} + \rho_{i,0}\mathbf{F}_{i,0}\cdot\begin{bmatrix}0\\1\end{bmatrix}\right). \tag{4.5$'$}$$

We choose to give a detailed derivation of the method based on (4.1) simply because in the numerical computation of an air bubble below we use a linear interpolant for the kinetic viscosity $\nu(\rho)$ between air and water, in which the jump is far less than the jump in the dynamic viscosity $\mu(\rho)$. The numerical computation using (4.1$'$) would not result in any significant difference.

With (4.5) in hand, we outline the time stepping procedure, again using forward Euler for simplicity of presentation, the key point being that p is treated *explicitly* in time in the momentum equation. Let $\boldsymbol{u}^n$, ρ^n, and p^n denote the flow variables on Ω_h, where the superscript denotes the nth time step. We first compute $\boldsymbol{u}^{n+1}$ at the *interior* grid points by discretizing the momentum equation (4.3a) as

$$\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n}}{\Delta t}+(\boldsymbol{u}^{n}\cdot\nabla_h)\boldsymbol{u}^{n}+\frac{1}{\rho^{n}}\nabla_h p^{n}=\nu(\rho^{n})\Delta_h\boldsymbol{u}^{n}+F^{n}, \tag{4.6}$$

and apply the no-slip boundary condition, setting $\boldsymbol{u}^{n+1}|_\Gamma=0$. Recall that $\nabla_h=(\tilde{D}_x,\tilde{D}_y)^T$ and $\Delta_h=(D_x^2+D_y^2)$ are second-order centered difference approximations to ∇ and Δ, respectively.

We next update ρ using the hyperbolic continuity equation (4.3c), noting that care must be taken when computing ρ^{n+1} to prevent spurious oscillations in regions where large density gradients are present. We treat this difficulty by discretizing (4.3c) using a shock-capturing scheme based on slope limiters and describe our approach by considering a 1D analog of (4.3c), namely

$$\rho_t+u\rho_x=0, \tag{4.7}$$

where u is now a scalar velocity, and the data $\{\rho_i^n\}$ is given on a 1D grid with mesh spacing h. We compute $\{\rho_i^{n+1}\}$ using

$$\begin{aligned}\frac{\rho_i^{n+1}-\rho_i^{n}}{\Delta t}&+\frac{\left(u_i^n\right)^+}{h}\left[\left(\rho_i^n+\frac{h}{2}s_i\right)-\left(\rho_{i-1}^n+\frac{h}{2}s_{i-1}\right)\right]\\&+\frac{\left(u_i^n\right)^-}{h}\left[\left(\rho_{i+1}^n-\frac{h}{2}s_{i+1}\right)-\left(\rho_i^n-\frac{h}{2}s_i\right)\right]=0,\end{aligned} \tag{4.8}$$

where $(u_i^n)^\pm=(u_i^n\pm|u_i^n|)/2$, and the slopes $\{s_{i-1},s_i,s_{i+1}\}$ are determined by a limiter applied to the data $\{\rho_i^n\}$. In the present scheme the slopes are chosen using a biased averaging procedure (BAP) [2, 3], given by

$$s_j=\mathcal{B}^{-1}\left[\frac{1}{2}\left(\mathcal{B}\left(D_x^+\rho_j^n\right)+\mathcal{B}\left(D_x^-\rho_j^n\right)\right)\right], \tag{4.9}$$

where $\mathcal{B}(x)=\arctan(x)$, which is referred to as the biased function.

Simply put, (4.8) and (4.9) update ρ via a parameter-free reconstruction of the density flux over the cell $(x_i-h/2,x_i+h/2)$ using upwind fluxes with the slope determined by (4.9), which in the case of smooth data gives a second-order approximation, even at extrema points [2, 3]. This is in contrast to standard second-order TVD limiters, such as the van Leer limiter, which gives only first-order accuracy at extrema points. At a physical boundary Γ (e.g., $i=0$) the no-slip condition $u|_\Gamma=0$ implies that $\rho^{n+1}|_\Gamma=\rho^n|_\Gamma$. However, note that the application of (4.9) requires the data $\{\rho_{i-2}^n,\rho_{i-1}^n,\rho_i^n,\rho_{i+1}^n,\rho_{i+2}^n\}$, and at the first point just inside the boundary, e.g., when $i=1$, the value of ρ_{-1}^n is required which lies outside of the computational domain. Either extrapolation is used or the value may be determined from a problem-dependent condition such as $(\partial\rho/\partial\mathrm{n})|_\Gamma=0$, in which case we take $\rho_{-1}^n=\rho_1^n$.

Having computed $\boldsymbol{u}^{n+1}$ and ρ^{n+1} we now update the pressure by solving the PPE (4.3b) for p^{n+1}. We begin by noting that by applying $\nabla\cdot\boldsymbol{u}=0$ we can rewrite the right-hand side of the PPE as

$$\nabla\cdot(-(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}+\nu(\rho)\Delta\boldsymbol{u}+\mathbf{F})=2\nabla u\cdot\nabla^{\perp}v+\nabla(\nu(\rho))\cdot\Delta\boldsymbol{u}+\nabla\cdot\mathbf{F},$$

where $\nabla^{\perp} = (-\partial_y, \partial_x)^T$. With this, the PPE (4.3b) is discretized as

$$\begin{aligned} &D_x\left(\frac{1}{\rho^{n+1}} D_x p^{n+1}\right) + D_y\left(\frac{1}{\rho^{n+1}} D_y p^{n+1}\right) \\ &\quad = 2\left(\nabla_h u^{n+1} \cdot \nabla_h^{\perp} v^{n+1}\right) + \nabla_h(\nu(\rho^{n+1})) \cdot \triangle_h \boldsymbol{u}^{n+1} + \nabla_h \mathbf{F}^{n+1}, \end{aligned} \tag{4.10}$$

where $\nabla_h^{\perp} = (-\tilde{D}_y, \tilde{D}_x)^T$, with the interpretation that

$$\frac{1}{\rho_{i+1/2}} = \frac{1}{2}\left(\frac{1}{\rho_{i+1}} + \frac{1}{\rho_i}\right).$$

As discussed in Section 1, treating p explicitly in time in (4.6) decouples the computation of (4.6) from the PPE (4.10). However, now the PPE has variable coefficients, and we resort to an iterative scheme for its solution. With this in mind, in order to demonstrate our scheme numerically, in the examples presented in Section 4.1 we have focused on flows for which $(\partial\rho/\partial\mathbf{n})|_{\Gamma} = 0$. In this case the coefficient matrix in (4.10) is symmetrizable, resulting in a linear system that is negative semidefinite (with a one-dimensional kernel). We solve the system using a preconditioned conjugate gradient iterative method with the discrete Laplacian ($\triangle_h$, implemented using FFTs) as the preconditioner. We emphasize that more general boundary conditions for ρ can be handled; our intention is *not* to investigate possible iterative solvers but rather to focus on the use of local pressure boundary conditions for approximating solutions of (4.3) and (4.4). Also, as before, since the viscous term is treated explicitly, a convectively stable high-order time stepping scheme should be used in (4.6) (and hence (4.8)) in place of forward Euler to avoid any cell Reynolds number constraint (see Section 1).

4.1. *Numerical Results for Variable Density Flows*

We present numerical results, in the form of an accuracy check and computation of an air bubble rising in water, of the variable density scheme outlined in Section 4.

Example 4 (2D accuracy check for variable density problem). We present an accuracy check of the variable density scheme of a computation on $\Omega = [0, \pi] \times [0, \pi]$. We have taken the exact solution of the 2D NSE as

$$\begin{cases} u(x, y, t) = ((3/4) + (1/4)\sin t)(-\sin^2 x \sin y \cos y), \\ v(x, y, t) = ((3/4) + (1/4)\sin t)(\sin x \cos x \sin^2 y), \\ p(x, y, t) = ((3/4) + (1/4)\sin t)(\cos x \sin y), \\ \rho(x, y, t) = ((3/4) + (1/4)\sin t)(2 + \cos x \cos y). \end{cases} \tag{4.11}$$

To ensure that (4.11) is an exact solution of (4.3) appropriate forcing functions are applied to the system. We took $Re = \pi/\nu = 500$, and computed solutions for various values of $N = \pi/h$ until time $t = 3.0$. Fourth-order Runge–Kutta time stepping was used and CFL $= 1.0$. Table V shows the absolute errors between the numerical solutions and the exact solutions, as well as the divergence of the computed velocity field. As the grid is refined the method achieves clean second-order accuracy for all flow variables as well as the divergence.

TABLE V
Absolute Errors at Time t = 3.0 for Variable Density Flow Accuracy Check (Re = 500)

	N	L^1 error	Order	L^2 error	Order	L^∞ error	Order
div $\boldsymbol{u}$	32	4.38e-02		2.09e-02		2.44e-02	
	64	1.14e-02	1.95	5.46e-03	1.94	6.95e-03	1.81
	128	2.86e-03	1.99	1.38e-03	1.98	1.80e-03	1.95
	256	7.18e-04	2.00	3.47e-04	2.00	4.55e-04	1.99
u	32	2.74e-02		1.28e-02		1.53e-02	
	64	7.07e-03	1.95	3.30e-03	1.96	3.99e-03	1.94
	128	1.78e-03	1.99	8.31e-04	1.99	1.01e-03	1.98
	256	4.47e-04	2.00	2.08e-04	2.00	2.54e-04	2.00
v	32	2.51e-02		1.11e-02		1.03e-02	
	64	6.21e-03	2.02	2.79e-03	1.98	2.82e-03	1.87
	128	1.55e-03	2.00	7.02e-04	1.99	7.19e-04	1.97
	256	3.87e-04	2.00	1.76e-04	2.00	1.81e-04	1.99
p	32	1.20e-01		3.99e-02		1.82e-02	
	64	3.23e-02	1.90	1.07e-02	1.90	4.86e-03	1.91
	128	8.22e-03	1.97	2.73e-03	1.97	1.24e-03	1.98
	256	2.07e-03	1.99	6.86e-04	1.99	3.10e-04	1.99
ρ	32	1.23e-01		5.91e-02		7.27e-02	
	64	3.19e-02	1.95	1.54e-02	1.94	2.00e-02	1.86
	128	8.03e-03	1.99	3.90e-03	1.98	5.13e-03	1.96
	256	2.01e-03	2.00	9.79e-04	2.00	1.29e-03	1.99

Example 5 (axisymmetric "air bubble" rising in water). We model an 0.0025 m (1/4 cm) radius air bubble rising in water using true physical parameters.

Physical parameters	Air	Water	Units (MKS)
Density(ρ)	1.161	995.65	kg/m^3
Coeff. of viscosity (μ)	0.0000186	0.0007977	kg-s/m

We set $\nu = \mu/\rho$ and use linear interpolation to determine ν for intermediate values of ρ. Gravity is accounted for via the force term $\mathbf{F} = [0, -9.80665]^T$. We assume symmetry of the flow along the vertical centerline of the bubble. The bubble is initially at rest, with the density interface between the air and water desingularized using the initial density profile,

$$\rho(x, y, 0) = \rho_{\text{air}} + \left(\frac{\rho_{\text{water}} - \rho_{\text{air}}}{2}\right) * \left(1 + \tanh\left(\frac{d - 0.0025}{0.00025}\right)\right).$$

Here d is the distance (in meters) from the center of the bubble to the point (x, y). The computational domain is taken to be $[0, 0.01] \times [0, 0.03]$ (in meters) with grid sizes $(N_x, N_y) = (256, 768)$ and we apply the boundary condition $(\partial\rho/\partial\mathbf{n})|_\Gamma = 0$.

We choose this test problem in order to gauge the ability of the variable density scheme to handle large density gradients, noting that for our simulation the density of the surrounding

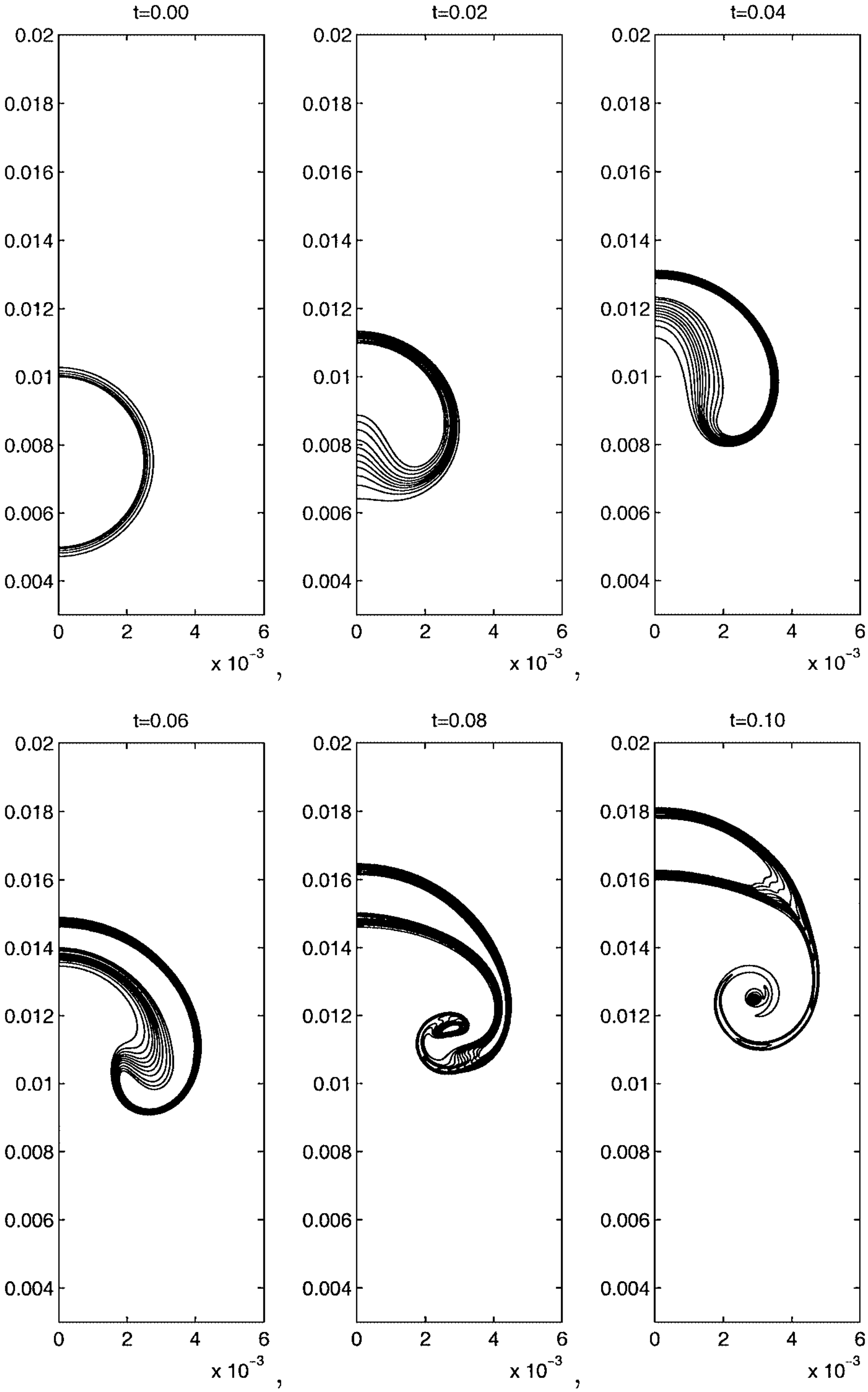


FIG. 6. Air bubble rising in water: Density contours at times $t = 0.00, 0.02, 0.04, 0.06, 0.08, 0.10$. Contour levels are (1 25 50 100 : 100 : 1000).

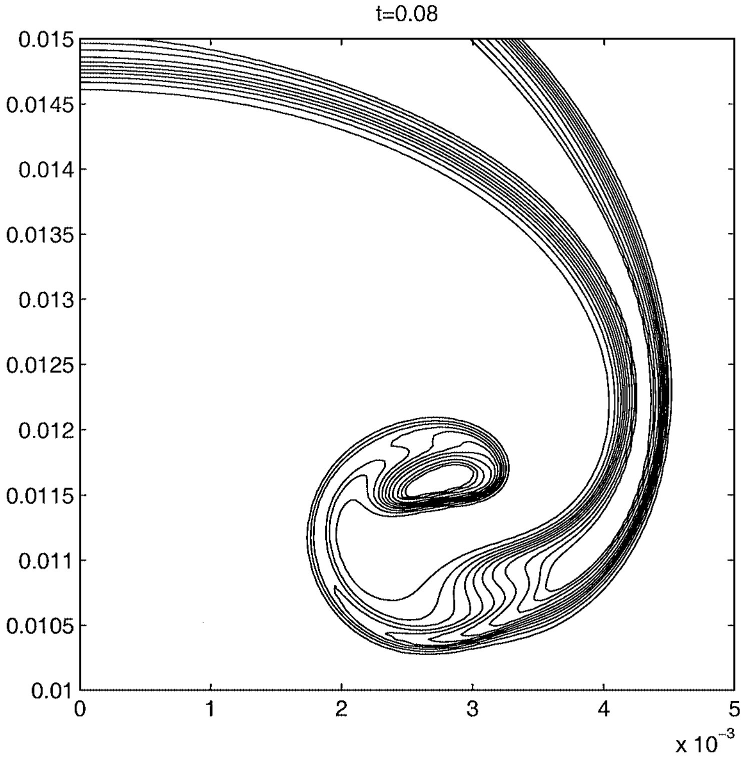


FIG. 7. Air bubble rising in water: Close-up of density contours at time $t = 0.08$.

fluid is approximately 850 times that of the air bubble. Similar simulations can be found in [1], where an adaptive projection method was used. Surface tension effects have been ignored in our model and leave this extension to future work. Figure 6 contains contour plots of the density at times $t = 0.00, 0.02, 0.04, 0.06, 0.08$, and 0.10, where we have only shown the bottom two-thirds of the flow domain. Figure 7 is a closeup of the density at $t = 0.08$, clearly indicating the ability of the scheme to resolve large density gradients with no oscillations.

5. CONCLUSIONS

We have presented a second-order numerical scheme implemented on a nonstaggered grid for time-dependent viscous incompressible flow. It is based on the velocity–pressure formulation and is well suited for moderate to large Reynolds number simulations. The key to the scheme is a consistent and accurate approximation of the Neumann pressure boundary condition for the associated pressure Poisson equation. Stability and accuracy of the method was demonstrated using normal mode analysis of the unsteady 2D Stokes flow. Extension of the scheme to the computation of both 3D and variable density flows was discussed. In all cases, the scheme is simple to implement and highly efficient. Numerical evidence, in the form of accuracy checks and results of computations of realistic physical flows, was presented to demonstrate the feasibility and effectiveness of the method.

APPENDIX A: EQUIVALENCE OF FORMULATIONS

THEOREM 1. *For $u \in L^\infty([0, T], H^2)$ the formulation* (1.1) *and* (1.2) *of the Navier–Stokes equations is equivalent to the PPE formulation* (1.7) *and* (1.8).

Proof: Assume $(\boldsymbol{u}, p)$ is a solution of (1.1) and (1.2). We need to show that $\boldsymbol{u}$ satisfies the boundary condition

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

This can be obtained directly by taking the trace of $\nabla \cdot \boldsymbol{u}$ on the boundary and using the fact that $\nabla \cdot \boldsymbol{u} = 0$ and $\boldsymbol{u}$ have enough regularity for $\boldsymbol{u} \in L^\infty$ $([0, T], H^2)$.

We also need to show that $(\boldsymbol{u}, p)$ satisfy the PPE (1.7b). This can again be obtained directly by taking the divergence of the momentum equation (1.1) and using the fact that $\nabla \cdot \boldsymbol{u} = 0$ and $\boldsymbol{u} \in L^\infty$ $([0, T], H^2)$, giving

$$\triangle p = -\nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u}).$$

Simple algebra and the fact that $\nabla \cdot \boldsymbol{u} = 0$ gives

$$-\nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u}) = (\nabla \cdot \boldsymbol{u})^2 - (\nabla \boldsymbol{u}) : (\nabla \boldsymbol{u})^T.$$

This shows that $(\boldsymbol{u}, p)$ is also a solution to (1.7) and (1.8).

Now, assume $(\boldsymbol{u}, p)$ is a solution of (1.7) and (1.8). All we need to show is that $\nabla \cdot \boldsymbol{u} = 0$. Take the divergence of (1.7a) to obtain

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

Now replace $\triangle p$ above with the right-hand side of (1.7b), and using the identity

$$\nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u}) - (\nabla \boldsymbol{u}) : (\nabla \boldsymbol{u})^T = \boldsymbol{u} \cdot \nabla (\nabla \cdot \boldsymbol{u}),$$

we obtain

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

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

$$\partial_t \phi + \boldsymbol{u} \cdot \nabla \phi + \phi^2 = \nu \triangle \phi. \tag{A1}$$

with boundary condition

$$\phi|_\Gamma = 0 \tag{A2}$$

and initial data

$$\phi|_{t=0} = 0.$$

We now show that ϕ equals zero almost everywhere, which can be proved by an energy estimate. Multiplying (A1) by 2ϕ and integrating over the domain Ω, we obtain

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

To obtain the last term of the above equation we used the integration by parts and the boundary condition (A2). Integration by parts also gives

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

Hence, we have

$$\frac{d}{dt}\int_\Omega \phi^2\,dx + \int_\Omega \phi^3\,dx = -2\nu\int_\Omega |\nabla\phi|^2\,dx. \tag{A3}$$

By Hölder's inequality

$$\int_\Omega |\phi|^3\,dx \le \left(\int_\Omega \phi^2\,dx\right)^{3/4}\left(\int_\Omega \phi^6\,dx\right)^{1/4}.$$

Furthermore, the Sobolev embedding theorem and Poincare's inequality give

$$\|\phi\|_{L^6} \le C\|\phi\|_{H^1} \le C\|\nabla\phi\|_{L^2}.$$

Hence,

$$\int_\Omega |\phi|^3\,dx \le C\|\phi\|_{L^2}^{3/2}\|\nabla\phi\|_{L^2}^{3/2},$$

and from Young's inequality we also have

$$\int_\Omega |\phi|^3\,dx \le \nu\|\nabla\phi\|_{L^2}^2 + C_\nu\|\phi\|_{L^2}^6.$$

Plugging the above into (A3) we arrive at

$$\frac{d}{dt}\int_\Omega \phi^2\,dx \le C\left(\int_\Omega \phi^2\,dx\right)^3,$$

with the initial conditions

$$\int_\Omega \phi^2\,dx = 0 \qquad \text{for } t = 0.$$

Hence, we have

$$\int_\Omega \phi^2\,dx = 0,$$

for all $t > 0$. Therefore, we have proved that

$$\phi = 0 \quad a.e.,$$

or

$$\nabla \cdot \boldsymbol{u} = 0 \quad a.e.$$

This proves that $(\boldsymbol{u}, p)$ is also the solution to (1.1) and (1.2) and completes the proof of the theorem.

APPENDIX B: HIGHER ORDER LOCAL PRESSURE BOUNDARY CONDITIONS

Analogous to the derivation of local vorticity boundary conditions (see [5] for a review), there are also many alternative and higher order local pressure boundary conditions. We list here a few of them along with their accuracy. For simplicity they are derived in the context of the 1D model discussed in Section 2 with the no-slip ($\boldsymbol{u}|_\Gamma = 0$) boundary condition. Extension to prescribed slip boundary conditions (see Appendix C) and 2D and 3D is straightforward.

A first-order approximation of the boundary condition $\partial_x u = 0$ is given by

$$u_{-1} = u_0,$$

which leads to

$$D_x p_0 = \frac{\nu}{h^2} u_1.$$

This is analogous to Fromm's vorticity boundary condition.

The second-order centered approximation of the boundary condition $\partial_x u = 0$ is given by

$$u_{-1} = u_1,$$

and the corresponding local pressure boundary condition is given by

$$D_x p_0 = \frac{2\nu}{h^2} u_1.$$

This is analogous to Thom's vorticity boundary condition.

A third-order approximation of $\partial_x u = 0$ is given by

$$u_{-1} = -\frac{3}{2} u_0 + 3u_1 - \frac{1}{2} u_2,$$

which gives

$$D_x p_0 = \frac{\nu}{2h^2} (8u_1 - u_2),$$

analogous to the Wilkes' vorticity boundary condition.

Similarly, corresponding to the Orszag–Israeli's [19] vorticity boundary conditions are the boundary conditions given by

$$\begin{aligned} D_x p_0 &= \frac{\nu}{3h^2} (10u_1 - u_2), \\ D_x p_0 &= \frac{\nu}{13h^2} (35u_1 - u_2). \end{aligned}$$

APPENDIX C: THE LOCAL PRESSURE BOUNDARY CONDITION FOR SLIP BOUNDARIES

The approximation (1.11) of the Neumann boundary condition for the PPE is easily extended to the situation where the physical boundary, say Γ_x, slips with a given velocity $u_b(x)$. In this case the divergence-free boundary condition in (1.10) reads

$$0 = \nabla \cdot \boldsymbol{u} = (u_b)_x + v_y = u_b'(x_i, 0) + \frac{(v_{i,1} - v_{i,-1})}{2h} + O(h^2),$$

and in (1.9) we now take

$$v_{i,-1} = v_{i,1} + 2h \cdot u_b'(x_i, 0).$$

The resulting boundary condition for p is then given by (along Γ_x)

$$\left.\frac{\partial p}{\partial y}\right|_{(x_i,0)} = 2\nu\left(\frac{v_{i,1}}{h^2} + \frac{u_b'(x_i, 0)}{h}\right).$$

APPENDIX D: EQUIVALENCE OF MAC SCHEME AND THE USE OF LOCAL PRESSURE BOUNDARY CONDITIONS FOR THE 1D MODEL ON A STAGGERED GRID

Again, consider the 1D model (2.3) and (2.4) on $\Omega = [-1, 1]$ of the unsteady 2D Stokes equations in Section 2. Recall, we denote a 1D nonstaggered grid Ω_h, defined by the points

$$x_j = -1 + jh, \quad h = 2/N, \quad j = 0, 1, \ldots, N,$$

and introduce staggered grid points

$$x_{j-1/2} = x_j - h/2, \quad j = 1, 2, \ldots, N.$$

Denote by u_j the approximation of u at the x_j point and by $u_{j-1/2}$ the approximation of u at the $x_{j-1/2}$ point. We require here an additional finite difference operator, the average operator, defined by

$$\mathcal{A}u(x) = \frac{u(x + h/2) + u(x - h/2)}{2}.$$

In the classic MAC scheme, approximations are computed with the use of a staggered grid, and the unknowns are given by

$$u_j, \quad j = 0, 1, \ldots, N; \quad v_{j-1/2}, \quad j = 0, 1, \ldots N + 1; \quad p_{j-1/2}, \quad j = 1, 2, \ldots N,$$

totaling $3N + 3$. The MAC discretization of the 1D model (2.3) is given by

$$\begin{cases} \partial_t u + D_x p = \nu\left(D_x^2 - k^2\right)u, & \text{at } x_j \text{ for } j = 1, 2, \ldots, N-1 \\ \partial_t v - kp = \nu\left(D_x^2 - k^2\right)v, & \text{at } x_{j-1/2} \text{ for } j = 1, 2, \ldots, N \\ D_x u + kv = 0, & \text{at } x_{j-1/2} \text{ for } j = 1, 2, \ldots, N \\ u_0 = u_N = \mathcal{A}v_0 = \mathcal{A}v_N = 0, \end{cases} \tag{D1a–d}$$

also totaling $3N + 3$ equations and boundary conditions, and the system is consistent. $\mathcal{A}v_0 = \mathcal{A}v_N = 0$ is known as the reflection boundary condition. We now increase by 4, in a consistent manner, both the number of unknowns and equations in this scheme. We define u_{-1} and u_{N+1} so that

$$(D_x u + kv)_{-1/2} = (D_x u + kv)_{N+1/2} = 0, \tag{D2}$$

and also define $p_{-1/2}$ and $p_{N+1/2}$ by

$$D_x p_0 = \left(D_x^2 - k^2\right)u_0, \quad D_x p_N = \left(D_x^2 - k^2\right)u_N. \tag{D3}$$

Appending these to the MAC scheme discretization (D1) we have

$$\begin{cases} \partial_t u + D_x p = \nu\left(D_x^2 - k^2\right)u, & \text{at } x_j \text{ for } j = 0, 1, \ldots, N \\ \partial_t v - kp = \nu\left(D_x^2 - k^2\right)v, & \text{at } x_{j-1/2} \text{ for } j = 1, 2, \ldots, N \\ D_x u + kv = 0, & \text{at } x_{j-1/2} \text{ for } j = 0, 1, \ldots, N+1 \\ u_0 = u_N = \mathcal{A}v_0 = \mathcal{A}v_N = 0, & \end{cases} \tag{D4a–d}$$

for a total of $3N + 7$ equations and boundary conditions and $3N + 7$ unknowns.

Now apply D_x to (D4a) and add to this (D4b) multiplied by k. Then using (D4c) we have

$$\left(D_x^2 - k^2\right)p = 0, \quad \text{at } x_{j-1/2} \text{ point for } j = 1, 2, \ldots, N.$$

From (D4c), we have

$$\mathcal{A}(D_x u + kv)_0 = \mathcal{A}(D_x u + kv)_N = 0,$$

and, using the boundary condition (D4d), we have

$$\tilde{D}_x u_0 = \tilde{D}_x u_N = 0. \tag{D5}$$

This gives the equivalent system

$$\begin{cases} \partial_t u + D_x p = \nu\left(D_x^2 - k^2\right)u, & \text{at } x_j \text{ for } j = 0, 1, \ldots, N \\ \partial_t v - kp = \nu\left(D_x^2 - k^2\right)v, & \text{at } x_{j-1/2} \text{ for } j = 1, 2, \ldots, N \\ \left(D_x^2 - k^2\right)p = 0, & \text{at } x_{j-1/2} \text{ for } j = 1, 2, \ldots, N \\ u_0 = u_N = \mathcal{A}v_0 = \mathcal{A}v_N = 0, & \\ \tilde{D}_x u_0 = \tilde{D}_x u_N = 0, & \end{cases} \tag{D6a–d}$$

for a total of $3N + 7$ equations and boundary conditions and $3N + 7$ unknowns. Since $u = 0$ on the boundary, as in the nonstaggered grid case the equation in (D6a) for $j = 0, N$ reads

$$D_x p_0 = \nu\left(D_x^2 - k^2\right)u_0, \quad D_x p_N = \nu\left(D_x^2 - k^2\right)u_N. \tag{D7}$$

Using the boundary condition (D5) to eliminate u_{-1} and u_{n+1} from (D7), we have

$$D_x p_0 = \frac{2\nu}{h^2} u_1, \quad D_x p_N = \frac{2\nu}{h^2} u_{N-1}.$$

Hence the system (D6) is equivalent to

$$\begin{cases} \partial_t u + D_x p = \nu\left(D_x^2 - k^2\right)u, & \text{at } x_j \text{ for } j = 1, 2, \ldots, N-1 \\ \partial_t v - kp = \nu\left(D_x^2 - k^2\right)v, & \text{at } x_{j-1/2} \text{ for } j = 1, 2, \ldots, N \\ u_0 = u_N = \mathcal{A}v_0 = \mathcal{A}v_N = 0 & \\ \left(D_x^2 - k^2\right)p = 0, & \text{at } x_{j-1/2} \text{ for } j = 1, 2, \ldots, N \\ D_x p_0 = \frac{2\nu}{h^2}u_1, \quad D_x p_N = \frac{2\nu}{h^2}u_{N-1}, & \end{cases} \tag{D8a–e}$$

for a total of $3N + 5$ equations and boundary conditions and $3N + 5$ unknowns. Comparing this with (2.11), we see that (D8) is precisely the local pressure boundary condition approach implemented on a staggered grid.

APPENDIX E: LEMMAS USED IN THE NORMAL MODE ANALYSIS

This appendix contains some formulas in the form of two lemmas used in the normal mode analysis in Section 2.1.

LEMMA 1. *We have*

$$\begin{aligned} \tilde{D}_x \sinh(\alpha x) &= \frac{\sinh(\alpha h)}{h} \cosh(\alpha x), \\ \tilde{D}_x \cosh(\alpha x) &= \frac{\sinh(\alpha h)}{h} \sinh(\alpha x), \\ \tilde{D}_x \sin(\gamma x) &= \frac{\sin(\gamma h)}{h} \cos(\gamma x), \\ \tilde{D}_x \cos(\gamma x) &= -\frac{\sin(\gamma h)}{h} \sin(\gamma x), \\ D_x^2 \sinh(\alpha x) &= \frac{4}{h^2} \sinh^2\left(\frac{\alpha h}{2}\right) \sinh(\alpha x), \\ D_x^2 \cosh(\alpha x) &= \frac{4}{h^2} \sinh^2\left(\frac{\alpha h}{2}\right) \cosh(\alpha x), \\ D_x^2 \sin(\gamma x) &= -\frac{4}{h^2} \sin^2\left(\frac{\gamma h}{2}\right) \sin(\gamma x), \\ D_x^2 \cos(\gamma x) &= -\frac{4}{h^2} \sin^2\left(\frac{\gamma h}{2}\right) \cos(\gamma x). \end{aligned}$$

LEMMA 2. *We have*

$$\frac{\sinh(\alpha h)}{h} = \alpha + O(h^2), \quad \frac{\sin(\alpha h)}{h} = \alpha + O(h^2).$$

ACKNOWLEDGMENT

The work of J.-G. Liu was supported by NSF grant DMS-0107218.

REFERENCES

1. A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, *J. Comput. Phys.* **142**, 1 (1998).
2. H. Choi and J. G. Liu, The reconstruction of upwind fluxes for conservation laws, *J. Comput. Phys.* **144**, 237 (1998).
3. H. Choi and J. G. Liu, Shock capturing with a simple biased averaging procedure, submitted for publication.
4. A. J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comput.* **22**, 745 (1968).
5. E. Weinan and J.-G. Liu, Vorticity boundary condition and related issues for finite difference schemes, *J. Comput. Phys.* **124**, 368 1996.
6. E. Weinan and J.-G. Liu, Essentially compact schemes for unsteady viscous incompressible flows, *J. Comput. Phys.* **126**, 122 1996.
7. E. Weinan and J.-G. Liu, Projection method II: Convergence and numerical boundary layers, *SIAM J. Numer. Anal.* **33**, 1597 1996.
8. J.-G. Liu and E. Weinan, Simple finite element method in vorticity formulation for incompressible flow, *Math. Comput.* **70**, 579 (2001).
9. J.-G. Liu and W. C. Wang, An energy preserving MAC–Yee scheme for the incompressible MHD equation, *J. Comput. Phys.* **174**, 12, (2001).
10. P. M. Gresho and R. L. Sani, On pressure boundary conditions for the incompressible Navier–Stokes equations, *Int. J. Numer. Methods Fluids* **7**, 1111 (1987).
11. P. M. Gresho, Some current CFD issues relevant to the incompressible Navier–Stokes equations, *Comput. Meth. Appl. Mech. Eng.* **87**, 201 (1991).
12. F. H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* **8**, 2182 (1965).
13. W. D. Henshaw, H. O. Kreiss, and L. G. M. Reyna, A fourth-order-accurate difference approximation for the incompressible Navier–Stokes equations, *Comput. Fluids* **23**, 575 (1994).
14. W. D. Henshaw, A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids, *J. Comput. Phys.* **113**, 13 (1994).
15. T. Y. Hou and B. T. R. Wetton, Convergence of a finite difference scheme for the Navier–Stokes equations using vorticity boundary conditions, *SIAM J. Numer. Anal.* **29**, 615 (1992).
16. K. E. Karniadakis, M. Israeli, and S. A. Orsag, High-order splitting methods for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **97**, 414 (1991).
17. L. Kleiser and U. Schumann, Treatment of the incompressibility and boundary conditions in 3-D numerical spectral simulation of plane channel flows, in *Notes on Numerical Fluid Mechanics*, edited by E. H. Hirschel (Vieweg, Braunschweig, 1980), pp. 165–173.
18. V. Girault and P. A. Raviart, *Finite Element Methods for Navier–Stokes Equations, Theory and Algorithms* (Springer-Verlag, Berlin, 1986).
19. S. A. Orszag and M. Israeli, Numerical simulation of viscous incompressible flow, *Ann. Rev. Fluid Mech.* **6**, 281 (1974).
20. S. A. Orsag, M. Israeli, and M. Deville, Boundary conditions for incompressible flows, *J. Sci. Comput.* **1**, 75 (1986).
21. R. Temam, Sur l'approximation de la solution des equations Navier–Stokes par la méthode des fractionnarires II, *Arch. Rational Mech. Anal.* **33**, 377 (1969).
22. A. Thom, The flow past circular cylinders at low speeds, *Proc. Royal Soc. A* **141**, 651 (1933).