A Local Pressure Boundary Condition Spectral Collocation Scheme for the Three-Dimensional Navier–Stokes Equations

Hans Johnston · Cheng Wang · Jian-Guo Liu

Abstract A spectral collocation scheme for the three-dimensional incompressible (u, p) formulation of the Navier–Stokes equations, in domains Ω with a non-periodic boundary condition, is described. The key feature is the high order approximation, by means of a local Hermite interpolant, of a Neumann boundary condition for use in the numerical solution of the pressure Poisson system. The time updates of the velocity u and pressure p are decoupled as a result of treating the pressure gradient in the momentum equation explicitly in time. The pressure update is computed from a pressure Poisson equation. Extension of the overall methodology to the Boussinesq system is also described. The uncoupling of the pressure and velocity time updates results in a highly efficient scheme that is simple to implement and well suited for simulating moderate to high Reynolds and Rayleigh number flows. Accuracy checks are presented, along with simulations of the lid-driven cavity flow and a differentially heated cavity flow, to demonstrate the scheme produces accurate three-dimensional results at a reasonable computational cost.

Keywords Three-dimensional viscous incompressible flow · pressure Poisson equation · Pressure boundary condition · Spectral collocation method · Lid driven cavity flow

H. Johnston (✉)
Department of Mathematics and Statistics, University of Massachusetts Amherst, Amherst, MA, USA
e-mail: johnston@math.umass.edu

C. Wang
Department of Mathematics, University of Massachusetts Dartmouth, Dartmouth, MA, USA
e-mail: cwang1@umassd.edu

J.-G. Liu
Departments of Physics and Mathematics, Duke University, Durham, NC, USA
e-mail: jian-guo.liu@duke.edu

1 Introduction

The incompressible Navier–Stokes equations (NSE) in non-dimensional form with a no-penetration, no-slip boundary condition for the velocity in a three-dimensional domain Ω with boundary $\Gamma = \partial\Omega$ are given by

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p = \nu \Delta \boldsymbol{u} + \boldsymbol{f}, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2}$$

$$\boldsymbol{u}|_\Gamma = 0. \tag{3}$$

Here $\boldsymbol{u} = (u, v, w)^T$ is the velocity field, p the pressure, $\boldsymbol{f} = (f_1, f_2, f_3)^T$ an external body force, and $\nu = 1/Re$ where Re is the Reynolds number.

A significant challenge in the design of numerical schemes for the $(\boldsymbol{u}, p)$ formulation (1)–(3) of the NSE is the nature of the coupling between the pressure gradient in the momentum equation (1) and the incompressibility constraint (2). The physical issue is: as the velocity field evolves in time the pressure changes *instantaneously* and *globally* to ensure that the velocity field remains incompressible. This can be directly enforced in a numerical scheme using a Stokes solver. However, this is computationally costly since the velocity and pressure time updates are coupled. Alternative approaches have been proposed in which the computation of the velocity and pressure updates are decoupled, most notably projection methods [2,5,16,22,28]. In these methods an intermediate velocity field is first computed, which may not satisfy the incompressibility constraint, and is then projected onto the space of divergence-free vector fields. Thus, projection method based schemes treat the pressure as a Lagrange multiplier to enforce the incompressibility condition. This methodology requires two Poisson solvers, one vector valued for $\boldsymbol{u}$ and one scalar for p, at each time step.

The starting point of the scheme presented herein is an alternate PDE formulation of the NSE, a pressure Poisson equation (PPE) formulation (c.f. [10,13,25]), given by

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p = \nu \Delta \boldsymbol{u} + \boldsymbol{f}, \tag{4}$$

$$\Delta p = -\nabla \cdot (\boldsymbol{u} \cdot \nabla \boldsymbol{u}) + \nabla \cdot \boldsymbol{f}, \tag{5}$$

$$\boldsymbol{u}|_\Gamma = 0, \quad (\nabla \cdot \boldsymbol{u})|_\Gamma = 0. \tag{6}$$

The incompressibility constraint (2) has been replaced by the PPE (5) and the divergence-free velocity boundary condition in (6). Taking the divergence of (4) and using (5), one sees that the divergence of the velocity field is governed by a heat equation,

$$\partial_t (\nabla \cdot \boldsymbol{u}) = \nu \Delta (\nabla \cdot \boldsymbol{u}). \tag{7}$$

Assuming the initial velocity field is divergence-free, i.e., $(\nabla \cdot \boldsymbol{u})|_{t=0} = 0$, applying the BCs in (6) shows that for $t > 0$, Eq. (7) has only the trivial solution

$$\nabla \cdot \boldsymbol{u} \equiv 0, \tag{8}$$

thus the velocity field remains divergence-free. A proof of the equivalence between the PPE formulation (4)–(6) and the classical formulation (1)–(3) within the framework of strong solutions can be found in [13].

We present a spectral collocation method here for both the natural clustering of grid points near physical boundaries, which aids in resolving fluid boundary layers. Spectral methods provide powerful tools for simulating fluid flows, particularly in three dimensions [3,4,6,8,15,24,29]. Following the advent of the FFT, their importance and influence were firmly established in the 1970s by pioneering simulations of inhomogeneous turbulence in periodic

domains. Since then, spectral methods have played a prominent role in simulating flows in domains with non-periodic boundary conditions [1,17,18,26,27].

As our interest is in simulating moderate to high Reynolds number flows, we employ the convectively stable classical fourth order Runge–Kutta (RK4) for the time discretization. When used in conjunction with the spectral collocation approximation in space described here, the resulting multi-stage explicit time stepping procedure is simple to implement and highly efficient. At each time step (stage), the velocity vector is dynamically updated by the momentum equation (4) at the interior collocation grid points with the pressure gradient computed from previous time stage. The pressure is then updated by solving the PPE (5) with a consistent BC derived at the numerical level for the pressure from the incompressibility condition in (6). The derivation of this pressure BC is described below in Sect. 2. The finite difference approximation analog of this BC was described in [11,13]. The key point in the derivation of our local pressure BC is a high order interpolation of the normal velocity with a local Hermite interpolant at the boundary. The order of accuracy depends on the number of collocation grid points involved, which we denote by K. To achieve higher order accuracy we can simply choose a larger K. The stability of this approach is demonstrated by numerical evidence.

It is straightforward to extend this methodology to the Boussinesq approximation for simulating moderate to high Rayleigh number flows. The momentum equation is treated in the same manner, with an addition of the gravity term. The temperature transport equation is the approximated by a similar spectral differentiation procedure as the momentum equation. The main computational cost at each time stage, for either the NSE or Boussinesq system, is a three-dimensional Poisson solver for the pressure. The resulting schemes are able to capture delicate and complicated structures of a flow with a high Reynolds and Rayleigh number at a reasonable computational cost.

The remainder of this article is organized as follows: In Sect. 2 we derive a consistent and computable Neumann boundary condition for pressure. Section 3 describes the Chebyshev spectral collocation numerical scheme for the three-dimensional NSE based on the formulation (4)–(6). The extension of the scheme to the three-dimensional incompressible Boussinesq equations is then described in Sect. 4. In Sect. 5 we present numerical accuracy checks of the proposed scheme for both the NSE and Boussinesq equations. Finally, in Sect. 6 we present simulations of the three-dimensional lid-driven cavity flow at various Reynolds numbers. The results at the highest Reynolds number computed herein, $Re = 1,000$, are compared and contrasted with the benchmark results in [1].

2 Neumann Boundary Condition for Pressure

Key to the computational efficiency of our scheme for (4)–(6) is that, like projection methods, the time updates of the velocity and pressure are decoupled. However, doing so requires a boundary condition for the elliptic PPE (5), and finding a proper pressure boundary condition for the PPE has been the subject of some controversy (c.f. [10,25] and references therein), both from the view-point of the well-posedness of the resulting continuous formulation and with respect to an implementable boundary condition for use in a numerical scheme. A natural candidate is given by the normal component of the momentum equation at the boundary, which gives a Neumann boundary condition for p

$$\frac{\partial p}{\partial \boldsymbol{n}} = \nu \Delta \boldsymbol{u} \cdot \boldsymbol{n} + \boldsymbol{f} \cdot \boldsymbol{n} \quad \text{on} \quad \Gamma, \tag{9}$$

where $\boldsymbol{n}$ is the outward unit normal along Γ.

Based on the vector identity $\Delta \boldsymbol{u} = \nabla(\nabla \cdot \boldsymbol{u}) - \nabla \times \nabla \times \boldsymbol{u}$ and the incompressibility (2), the above (9) can be reduced to

$$\frac{\partial p}{\partial \boldsymbol{n}} = -\nu(\nabla \times \nabla \times \boldsymbol{u}) \cdot \boldsymbol{n} + \boldsymbol{f} \cdot \boldsymbol{n} \quad \text{on} \quad \Gamma, \tag{10}$$

which result in a well-posed system (3)–(5) and (10). Indeed, this boundary condition together with the derived boundary condition (9) implies that

$$\partial_n(\nabla \cdot \boldsymbol{u})|_\Gamma = 0. \tag{11}$$

Its combination with (7) and the initial condition implies that $\nabla \cdot \boldsymbol{u} \equiv 0$. Equivalence and well-posedness for the system (3)–(5) and (10) was proved in [20]. Recent analysis presented in [14,20,23] shows that such a reformulation could lead to an unconditional stable numerical scheme with an explicit treatment of the pressure gradient. When the flow is resolved, the system (3)–(5), together with the alternate Neumann boundary condition (10) gives an excellent enforcement of incompressibility. However, in the case of marginal resolution such an enforcement is much subtler. An additional projection or divergence suppression may be needed in the numerical implementation (see [23]), where numerical experiments showed that the enforcement of incompressibility will be enhanced with an implicit treatment of the pressure. However, such a coupled system of velocity and pressure requires complicated techniques, e.g. the influence matrix method [4], and makes the computational cost very expensive. For simplicity, we consider the reformulated system (3)–(5) together with the derived Neumann boundary condition (9).

3 Spectral Collocation Method for 3D NSE

In this section we present the complete details of our spatial and temporal discretization for the PPE formulation (4)–(6). First, a local formula for the pressure along the boundary Γ is derived from the momentum equation (4) evaluated at the boundary and the incompressibility constraint in (6). The spatial discretization, specifically the use of our derived local pressure boundary condition (LPBC) in conjunction with an explicit time stepping scheme, is then described.

Consider a flow in the canonical domain $\Omega = [-1, 1]^3$ and let N_x, N_y, and N_z be positive integers. We discretize in the x-, y-, and z-direction using the collocation points $\{x_i\}_{i=0}^{N_x}$, $\{y_j\}_{j=0}^{N_y}$, and $\{z_k\}_{k=0}^{N_z}$, respectively, forming discrete grids for $I = [-1, 1]$ in each coordinate, the Cartesian product of which gives a discrete grid $\Omega_h = I_h^{N_x} \times I_h^{N_y} \times I_h^{N_z}$ for Ω. Denote by $\Gamma_h = \Gamma \cap \Omega_h$ the grid points that lie along the boundary, and further distinguish the eight corner points by Γ_c.

We note that the method is easily adapted to many other simple domains. For example, in channel flow one usually enforces periodic conditions in the coordinate directions parallel to the channel. It is straightforward to modify the scheme described herein to handle such a flow scenario, and thus we omit the details.

In what follows we focus on the most common choice of collocation points for a bounded non-periodic domain, the Chebyshev Gauss–Lobatto points. Their popularity is of course due to the identification—after an appropriate change of variable—of Chebyshev series with Fourier Cosine series, making possible the efficient computation of discrete derivatives at the Chebyshev points using an FFT. We emphasize, however, that the methodology presented

below is quite general in that the collocation points can be chosen from any of the standard families of Gauss-type points.

Recall, for $N_x > 0$ the Chebyshev Gauss-Lobatto points are given by $x_i = -\cos(i\pi/N_x)$ for $i = 0, 1, \ldots, N_x$. Let $\mathcal{D}_x$ denote the $(N_x + 1) \times (N_x + 1)$ Chebyshev differentiation matrix (c.f. [3,12,29]). Then for a function $f(x, y, z)$, let the $(N_x + 1) \times (N_y + 1) \times (N_z + 1)$ array F be defined by $F_{i,j,k} = f(x_i, y_j, z_k)$. We then denote approximations to the partial derivatives $\partial_x f$ and $\partial_x^2 f$ as

$$\partial_x f\left(x_i, y_j, z_k\right) \approx \left(\mathcal{D}_x F\right)_{i,j,k}, \quad \partial_x^2 f\left(x_i, y_j, z_k\right) \approx \left(\mathcal{D}_x^2 F\right)_{i,j,k}, \tag{12}$$

which are spectrally accurate with increasing N_x provided that f is sufficiently smooth. With $\mathcal{D}_y$ and $\mathcal{D}_z$ analogously defined, we denote the discrete gradient and Laplacian operators by

$$\nabla_h = \left(\mathcal{D}_x, \mathcal{D}_y, \mathcal{D}_z\right)^T, \quad \Delta_h = \left(\mathcal{D}_x^2 + \mathcal{D}_y^2 + \mathcal{D}_z^2\right). \tag{13}$$

Even in the case of a periodic BC in one or two of the coordinate directions, where a uniform grid is used, we will still use this discrete operator notation to illustrate the scheme.

3.1 Local Pressure Boundary Condition

The issue of a consistent, implementable boundary condition for pressure has been a challenging, even controversial, issue in the design of numerical schemes based on the velocity-pressure formulation of the NSE. The crux of the problem is that no explicit pressure boundary condition appears in either the classical formulation (1)–(3) or PPE formulation (4)–(6). While the Neumann pressure boundary condition (9) is a natural candidate, it adds no additional information to either system since it is derived by taking the dot product of the momentum equation at the boundary with the outward unit normal. Here we describe a discrete implementation of (9) derived from a local Hermite interpolant of the velocity field that enforces the divergence-free boundary condition in (6).

To derive a discrete boundary condition for the PPE formulation, we consider a particular segment of the boundary, say $\Gamma_{z=-1}$ on the bottom of Ω. Along this segment $\boldsymbol{n} = -\mathbf{k} = -(0, 0, 1)^T$ and the Neumann pressure boundary condition (9) reads

$$\frac{\partial p}{\partial \boldsymbol{n}} = \nu \Delta \boldsymbol{u} \cdot \boldsymbol{n} + \boldsymbol{f} \cdot \boldsymbol{n} = -\nu \Delta w - f_3 = -\nu \partial_z^2 w - f_3, \tag{14}$$

since $w|_{\Gamma_{z=-1}} = 0$ implies $\partial_x^2 w|_{\Gamma_{z=-1}} = \partial_y^2 w|_{\Gamma_{z=-1}} = 0$. Furthermore, at any point along $\Gamma_{z=-1}$ the normal velocity w satisfies the two boundary conditions

$$w = \partial_z w = 0. \tag{15}$$

The first of these follows from the no-penetration boundary condition, and the second from the divergence-free velocity boundary condition in (6)

$$0 = \nabla \cdot \boldsymbol{u} = \partial_x u + \partial_y v + \partial_z w = 0 + 0 + \partial_z w = \partial_z w, \tag{16}$$

since the no-slip velocity boundary condition implies that tangential derivatives of tangential velocities at the boundary are zero. A discrete approximation to (14) is found by differentiating a local polynomial interpolant of w which satisfies (15). To explain, fix a point $\boldsymbol{x} = (x_i, y_j, z_0)$ along $\Gamma_{z=-1}$ and a positive integer $K_z \leq N_z$. Let $W(z)$ be the unique polynomial of degree at most $K_z + 1$ determined by the interpolation conditions

$$W(z_k) = w_{i,j,k} \quad \text{for} \quad 0 \leq k \leq K_z, \quad W'(z_0) = C, \tag{17}$$

where C is a prescribed constant. Then $W(z)$ can be written as

$$W(z) = \sum_{k=0}^{K_z} w_{i,j,k} l_k(z) + \left[C - \sum_{k=0}^{K_z} w_{i,j,k} l_k'(z_0) \right] (z - z_0) l_0(z), \quad (18)$$

where $\{l_k\}_{k=0}^{K_z}$ are the classical Lagrange interpolating polynomial basis functions determined by the grid points $\{z_k\}_{k=0}^{K_z}$, i.e., $\deg(l_k) = K_z$ and $l_k(z_{k'}) = \delta_{kk'}$ for $0 \le k, k' \le K_z$.

Our LPBC is given by the second order derivative of (18) evaluated at $z = z_0 = -1$ to approximate the $\partial_z^2 w$ term in (14):

$$\begin{aligned}
\frac{\partial p}{\partial \boldsymbol{n}}(\boldsymbol{x}) &\approx -\nu W''(z_0) - (f_3)_{i,j,0} \\
&= -\nu \left(\sum_{k=0}^{K_z} w_{i,j,k} \left[l_k''(z_0) - 2l_k'(z_0) l_0'(z_0) \right] + 2C\, l_0'(z_0) \right) - (f_3)_{i,j,0},
\end{aligned}$$
$$(19)$$

with $C = 0$ to enforce $\partial_z w = 0$. Concise formulas for the derivatives of the Lagrange polynomials can be found in "Appendix".

The derivation of a discrete pressure boundary condition along the remaining boundaries can be easily generalized by following the steps in (14)–(19). While in theory (19) is spectrally accurate for $K_z = N_z$, in practice one can expect to resolve the kinematics of the boundary layer for significantly smaller values of K_z due to the clustering of the collocation points at the boundary. This is verified by the numerical experiments presented in Sect. 5.

Finally, derivation of our pressure Neumann boundary condition can be easily extended to inhomogeneous velocity boundary conditions. Indeed, in the case of a prescribed slip velocity boundary condition, from (16) we now have $\partial_z w = -(\partial_x u + \partial_y v)$; thus in (17)–(19) we set $C = -(\partial_x u(\boldsymbol{x}) + \partial_y v(\boldsymbol{x}))$.

Remark 1 The proposed numerical method is PPE based; the pressure gradient term is dynamically updated at each time step/stage, and the pressure boundary condition is enforced by a spectral approximation to (9) or (10), as given by (19). This discrete boundary condition can be very efficiently implemented through a fully explicit time stepping procedure, as described in later sections.

This numerical approach is very different from the projection method. In the projection method, an intermediate velocity field is first computed without a consideration of the pressure gradient term, so that the incompressibility condition is not satisfied; afterward, the intermediate velocity is projected into the space of divergence-free vector fields. In other words, projection based schemes treat the pressure as a Lagrange multiplier to enforce the incompressibility condition, while the PPE based schemes treat the pressure gradient as an intrinsic dynamic term.

There have been a few projection based methods that use a non-trivial pressure boundary condition in the projection step in order to reduce the numerical boundary layer. For example, the alternate boundary condition (10) was used in [22]. However, there are many subtle differences between the method reported in [22], a projection based scheme, and the proposed method in this article, a PPE based one. First, the right hand side of the alternate boundary condition (10) reported in [22] is computed by an application of $\nabla \times \nabla \times$ to the intermediate velocity field, while a similar operator is applied to the current time step velocity field $\boldsymbol{u}^n$ in the pressure boundary condition of our proposed scheme. Moreover, the projection method

assures the exact divergence-free property of the computed velocity vector (at a numerical level), while the PPE based scheme assures the exact no-penetration, no-slip boundary condition for the velocity field.

3.2 Solution of the Discrete PPE

With the LPBC in-hand we turn to the numerical solution of PPE (5), which is required at each stage of the time stepping procedure to be described below in Sect. 3.3. For a velocity field $\boldsymbol{u}$ defined on Ω_h/Γ_c we need to solve

$$\Delta_h p = -\nabla_h \cdot (\boldsymbol{u} \cdot \nabla_h \boldsymbol{u}) \quad \text{in} \quad \Omega_h/\Gamma_c, \tag{20}$$

$$\frac{\partial p}{\partial \boldsymbol{n}} = g \qquad\qquad \text{on} \quad \Gamma_h/\Gamma_c, \tag{21}$$

where g denotes the evaluation of the LPBC (19). This discrete elliptic system can be solved efficiently by the matrix-diagonalization method [7,21,24]. The Neumann boundary condition (21) is first used to reduce (20) to a system involving only unknowns in the interior of the domain Ω_h/Γ_h. Once solved, Eq. (21) is again used to recover the pressure on Γ_h/Γ_c. To briefly explain, fix a point (x_i, y_j, z_0) along $\Gamma_{z=-1}$ and consider also the point (x_i, y_j, z_{N_z}) on the opposite boundary $\Gamma_{z=1}$. Denote by $\widehat{p}$ the column vector comprised of the unknown pressure variable for these fixed values of i and j, i.e., $\widehat{p}_k = p_{i,j,k}$ for $0 \le k \le N_z$. Then the discrete form (21) reads $(D_z\widehat{p})_0 = g_{i,j,0}$ and $(D_z\widehat{p})_{N_z} = g_{i,j,N_z}$, and from these two expressions one can uniquely solve for $\widehat{p}_0$ and $\widehat{p}_{N_z}$ in terms of $\widehat{p}_k$ for $1 \le k \le N_z - 1$. Repeating this process for all remaining pairs of opposing boundary points in Γ_h/Γ_c leads to the elimination of the equations in the discrete PPE (20) for the pressure at these points. The reduced system is given by

$$\widetilde{\Delta}_h p = \left(\widetilde{D}_x^2 + \widetilde{D}_y^2 + \widetilde{D}_z^2 \right) p = h \quad \text{in} \quad \Omega_h/\Gamma_h, \tag{22}$$

where h is the interior of the right-hand side of (20) appropriately modified by the elimination of the pressure on the boundary. $\widetilde{D}_x^2$, $\widetilde{D}_y^2$, and $\widetilde{D}_z^2$ are, respectively, the interiors of D_x^2, D_y^2, and D_z^2 modified by the elimination of the pressure variables on Γ_h/Γ_c. The eigenvalues of each matrix are real, non-positive, and distinct, hence diagonalizable [9]. This fact leads to an efficient inversion of the discrete operator $\widetilde{\Delta}_h$. Note that, as expected, this operator has a one-dimensional kernel, consistently reflecting the fact that pressure in the NSE is only determined up to an additive constant. Once (22) is solved, the pressure on Γ_h/Γ_c is recovered from (21).

3.3 Time Discretization-Method of Lines

In this section we describe the time stepping procedure for the PPE formulation (4)–(6). As our goal is a scheme capable of efficiently computing moderate to high Reynolds (or Rayleigh; see Sect. 4) number flows, a convectively stable explicit time discretization of the momentum equation should be used. An ideal candidate is RK4, used for all the results presented herein, since its stability region encompasses an appreciable portion of the imaginary axis. In addition, computation of the velocity and pressure time updates are decoupled by explicitly discretizing in time the pressure term in the momentum equation, which is equally critical to the numerical efficiency. In other words, the pressure term is dynamically updated as a force term, in stark contrast to being cast as a Lagrange multiplier in the projection

method. Of course, the nonlinear term is also discretized explicitly in time to avoid a costly nonlinear solver.

Given a time step Δt let $t^n = n\Delta t$. For simplicity, we use forward Euler to illustrate the time discretization, noting that the extension to a higher order time stepping, such as Runge–Kutta methods, is straightforward.

Time-stepping: Given $(\boldsymbol{u}^n, p^n)$ compute $(\boldsymbol{u}^{n+1}, p^{n+1})$ by

Step 1. Update the velocity at the interior points Ω_h/Γ_h via

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t} + \boldsymbol{u}^n \cdot \nabla_h \boldsymbol{u}^n + \nabla_h p^n = \nu \Delta_h \boldsymbol{u}^n + \boldsymbol{f}^n, \tag{23}$$

and impose the boundary condition $\boldsymbol{u}^{n+1}|_{\Gamma_h} = 0$.
Step 2. Compute the LPBC (19) for p^{n+1} from $\boldsymbol{u}^{n+1}$.
Step 3. Solve for the PPE for p^{n+1}:

$$\Delta_h p^{n+1} = -\nabla_h \cdot (\boldsymbol{u}^{n+1} \cdot \nabla_h \boldsymbol{u}^{n+1}) + \nabla \cdot \boldsymbol{f}^{n+1}. \tag{24}$$

The numerical experiments indicate that the overall scheme is stable under the condition

$$\frac{\|\boldsymbol{u}\|_\infty \Delta t}{h_{min}} \leq \text{CFL} < 1, \quad \text{and} \quad \frac{\nu \Delta t}{h_{min}^2} \leq \frac{1}{10}, \tag{25}$$

in which h_{min} denotes the smallest mesh size in Chebyshev collocation grid.

The classical fourth order Runge–Kutta method, a multi-stage explicit time stepping procedure, is used as the time discretization in the momentum equations. The explicit time stepping of convection and diffusion terms makes the scheme easy to implement. Such an explicit treatment can avoid any stability concern caused by the cell-Reynolds number constraint if a high order Runge–Kutta method, such as classical RK4, is applied. Note that only one Poisson solver, above in Step 2, is required at each stage in Runge–Kutta time stepping.

In more detail, the classical RK4 discretization is given by

$$\xi_1 = \boldsymbol{u}^n, \quad \xi_2 = \boldsymbol{u}^n + \frac{1}{2}\Delta t\, F(t^n, \xi_1), \quad \xi_3 = \boldsymbol{u}^n + \frac{1}{2}\Delta t\, F(t^n + \frac{1}{2}\Delta t, \xi_2),$$

$$\xi_4 = \boldsymbol{u}^n + \Delta t\, F(t^n + \frac{1}{2}\Delta t, \xi_3),$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \frac{1}{6}\Delta t\, F(t^n, \xi_1) + \frac{1}{3}\Delta t\, F(t^n + \frac{1}{2}\Delta t, \xi_2)$$

$$+ \frac{1}{3}\Delta t\, F(t^n + \frac{1}{2}\Delta t, \xi_3) + \frac{1}{6}\Delta t\, F(t^n + \Delta t, \xi_4), \tag{26}$$

in which F is formulated as $\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t F(t^n, \boldsymbol{u}^n)$, corresponding to the forward Euler time update (23), combined with the PPE (24) and the boundary condition (19).

Remark 2 Note in (23) that for neither the nonlinear convection, the pressure gradient or diffusion term does time appear as a variable explicitly. As a result, the time evaluations at different time stages for F are not needed in (26), unless a time-dependent external force term $\boldsymbol{f}$ is involved in the equation.

In particular, for the PPE (24) and the pressure boundary condition (19), the pressure variable always keeps the same time instant as the velocity field $\boldsymbol{u}$, after $\boldsymbol{u}$ is updated by the last time stage. Therefore, a time evaluation at different stages is not needed in the pressure Poisson solver.

Remark 3 In addition to the classical RK4, some other multi-stage Runge–Kutta discretizations can also be used. Meanwhile, for the computation of moderate to high Reynolds number flows (with small ν), the corresponding RK scheme has to have a stability region that encompasses an appreciable portion of the imaginary axis. From this point of view, both RK3 and RK4 are good choices, while RK2 is insufficient for moderate to high Reynolds number flows since its stability region does not contain any portion of the imaginary axis. This phenomenon has also been verified by various numerical experiments.

Regarding this linearized stability property, all versions of RK2 are equivalent, since they share the same characteristic polynomial and stability domain. Similarly, all versions of RK3 scheme share the same linear stability property.

4 Extension to the Boussinesq Equations

For both the NSE formulations (1)–(3) and (4)–(6), the fluid was assumed to have constant density and temperature. Meanwhile, for flows that have slight temperature variations, the change in density can be neglected. Under this assumption the flow can be modeled by the Boussinesq approximation for a unit density incompressible fluid in a domain Ω of height h with no-slip velocity boundary conditions.

By introducing the temperature variable T, and when non-dimensionalized by space, time, and temperature scales, which are dependent on the particular temperature boundary conditions at $z = \pm 1$, the PPE formulation of the Boussinesq equations for a fluid in a domain Ω of height h are given by

$$\partial_t \mathrm{T} + (\boldsymbol{u} \cdot \nabla)\mathrm{T} = (\mathrm{Pr}R)^{-1/2} \Delta \mathrm{T} + \boldsymbol{f}_T, \tag{27}$$

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p = (\mathrm{Pr}/\mathrm{R})^{1/2} \Delta \boldsymbol{u} + \mathrm{T}\mathbf{k} + \boldsymbol{f}, \tag{28}$$

$$\Delta p = -\nabla \cdot (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \partial_z \mathrm{T} + \nabla \cdot \boldsymbol{f}, \tag{29}$$

$$\boldsymbol{u}|_\Gamma = 0, \quad (\nabla \cdot \boldsymbol{u})|_\Gamma = 0, \tag{30}$$

where $\boldsymbol{f}_T$ is a force term. The parameter R is the measure of the imposed thermal forcing, with its definition dependent on the choice of temperature boundary conditions at $z = \pm 1$, and $\mathrm{Pr} = \nu/\kappa$ the Prandtl number, where ν is the fluid's kinematic viscosity and κ its thermal diffusivity.

We consider here two boundary condition scenarios for T at the top and bottom boundaries $z = \pm 1$, fixed temperature (Dirichlet) and inhomogeneous fixed heat flux (Neumann). We leave the boundary conditions for T at the sidewalls $x, y = \pm 1$ unspecified since they do not enter into the determination of R.

In the case of fixed temperature boundary conditions, the space, time, and temperature scales are, respectively, h, $(h/\alpha g \delta \mathrm{T}^*)^{1/2}$, and $\delta \mathrm{T}^* = \mathrm{T}|_{z=-1} - \mathrm{T}|_{z=1}$, the dimensional temperature drop, where g is the acceleration due to gravity and α is the thermal expansion coefficient of the fluid. In this case, the thermal forcing parameter R is the usual Rayleigh number

$$\mathrm{R} = \mathrm{Ra} = \alpha g \delta \mathrm{T}^* h^3 / \nu \kappa, \tag{31}$$

and the non-dimensional temperature boundary conditions are $\mathrm{T}|_{z=-1} = 1$ and $\mathrm{T}|_{z=1} = 0$. For the case of an imposed vertical fixed heat flux $\sim \beta \neq 0$, the space, time, and temperature scales are, respectively, h, $(\alpha g \beta)^{-1/2}$, and βh. Then

$$\mathrm{R} = \widehat{\mathrm{R}} = \alpha g \beta h^4 / \nu \kappa, \tag{32}$$

with $\partial T/\partial z|_{z=\pm 1} = -1$.

For the space and time discretization of the Boussinesq system (27)–(30), we again use forward Euler to illustrate. Given $(\boldsymbol{u}^n, p^n, T^n)$, first compute T^{n+1} on Ω_h by

$$\frac{T^{n+1} - T^n}{\Delta t} + \boldsymbol{u}^n \cdot \nabla_h T^n = (\text{Pr}R)^{-1/2} \Delta_h T^n + f_T^n. \tag{33}$$

To enforce a prescribed boundary condition for the temperature we overwrite the values of T^{n+1} on Γ_h/Γ_c. For a given Dirichlet condition T_{BC}^D, we set $T^{n+1} = T_{BC}^D$ on Γ_h/Γ_c. In the case of a Neumann boundary condition, again fix a point (x_i, y_j, z_0) along the bottom boundary at $z = -1$ at which we need to enforce $\partial T^{n+1}/\partial z = T_{BC}^N$. Choose a positive integer $K_{BC} < N_z$, and let $\mathcal{T}(z) = \sum_{k=0}^{K_{BC}} T_{i,j,k}^{n+1} l_k(z)$ be a local polynomial interpolant of T^{n+1}, where $\{l_k\}_{k=0}^{K_{BC}}$ are the Lagrange polynomial basis functions defined by the points $\{z_k\}_{k=0}^{K_{BC}}$. Then evaluating the derivative of $\mathcal{T}(z)$ at the boundary and solving for T_{i,j,z_0}^{n+1} gives

$$T_{i,j,z_0}^{n+1} = \left(T_{BC}^N - \sum_{k=1}^{K_{BC}} T_{i,j,k}^{n+1} l_k'(z_0) \right) \Big/ l_0'(z_0). \tag{34}$$

While in theory (34) is spectrally accurate for $K_{BC} = N_z$, as was the case for the LPBC, in practice one can expect (34) to be highly accurate for significantly smaller values of K_{BC} since the collocation points cluster at the boundary. This is verified by the numerical results presented in Sect. 5.2. Note that this procedure is easily generalized to any of the other boundary segments. Concise formulas for the Lagrange polynomial derivatives that appear above can be found in "Appendix".

The time discretization of the Boussinesq system requires only minor modifications to the scheme for the NSE in Sect. 3.3 to account for the temperature terms in (28)–(29). Treating the gravity term explicitly in time in the momentum equation, we compute $\boldsymbol{u}^{n+1}$ using

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t} + \boldsymbol{u}^n \cdot \nabla_h \boldsymbol{u}^n + \nabla_h p^n = (\text{Pr}/R)^{1/2} \Delta_h \boldsymbol{u}^n + T^n \mathbf{k} + f_T^n, \tag{35}$$

and set $\boldsymbol{u} = 0$ on Γ_h/Γ_c. For the LPBC, the dot product of (28) and $\boldsymbol{n}$ gives

$$\frac{\partial p}{\partial \boldsymbol{n}} = \left((\text{Pr}/R)^{1/2} \Delta \boldsymbol{u} \right) \cdot \boldsymbol{n} + T(\mathbf{k} \cdot \boldsymbol{n}),$$

Thus, only for $\boldsymbol{n} = \pm\mathbf{k} = \pm(0, 0, 1)^T$ (i.e., along the boundary), there is a contribution from the temperature to the derived Neumann pressure boundary condition. Since T^{n+1} has already been computed, it is straightforward to incorporate this term into (19) to compute the LPBC at time t^{n+1}. Finally, from (29) the discrete PPE reads

$$\Delta_h p^{n+1} = -\nabla_h \cdot \left(\boldsymbol{u}^{n+1} \cdot \nabla_h \boldsymbol{u}^{n+1} \right) + \mathcal{D}_z T^{n+1}, \tag{36}$$

for which the matrix-diagonalization method described in Sect. 3.2 is used to obtain the solution.

As was the case for the NSE in Sect. 3.2, the classical RK4 method is chosen as the time discretization in both the momentum equation and the temperature transport equation. The explicit treatment of convection, diffusion and gravity terms in both equations avoids a stability constraint caused by the cell-Reynolds number, if the stability condition (25) is satisfied. The overall scheme is efficient and highly accurate. Only one three-dimensional Poisson solver for the pressure is required at each stage in Runge–Kutta time stepping.

5 Numerical Accuracy Checks

In this section we present accuracy checks to demonstrate the numerical convergence properties of the schemes for the NSE in Sect. 3 and the Boussinesq system in Sect. 4. In each case the force term f is chosen to ensure that our prescribed exact solutions satisfy the system at-hand, and as the velocity field in the latter must satisfy a no-slip boundary condition. We take the computational domain to be $\Omega = [0, 1]^3$ to allow for simpler analytic expressions of these solutions. The error is measured in the discrete L^∞ and L^2 norms,

$$\|f\|_{L^\infty} = \max_{x \in \Omega_h} |f(x)|, \quad \|f\|_{L^2} = \left(\frac{1}{|\Omega_h|} \int_{\Omega_h} |f(x)|^2 dx \right)^{\frac{1}{2}}, \tag{37}$$

where a spectrally accurate quadrature rule, based on weights from entries of $\mathcal{D}^{-1}$ (c.f. [29]), is used to compute the L^2 integral over Ω_h.

5.1 NSE Accuracy Checks

We choose the exact velocity $u_e = (u_e, v_e, w_e)^T$ on $\Omega = [0, 1]^3$ to be

$$u_e = \cos(t) \begin{bmatrix} -\dfrac{6}{\pi^2} \sin^2(\pi x) \sin(2\pi y)(z - z^2) \\[2mm] \dfrac{1}{2\pi} \sin(2\pi x) \sin^2(\pi y) \sin(\pi z) \\[2mm] \sin(2\pi x) \sin(2\pi y) \left(\dfrac{3}{\pi} z^2 - \dfrac{2}{\pi} z^3 + \dfrac{1}{2\pi} \cos(\pi z) - \dfrac{1}{2\pi} \right) \end{bmatrix}, \tag{38}$$

which satisfies $\nabla \cdot u_e = 0$ and $u_e|_\Gamma = 0$ for all t. The exact pressure is taken as

$$p_e(x, y, z, t) = \cos(t) \left[\frac{1}{2\pi} \cos(2\pi x) \cos(2\pi y) \cos(\pi z) \right], \tag{39}$$

with the time dependent force term f in (1) set to

$$f = \nu \Delta u_e - \left(\partial_t u_e + u_e \cdot \nabla u_e + \nabla p_e \right) \tag{40}$$

to ensure that (u_e, p_e) is indeed a solution of (1)–(3).

Solutions with $\nu = 10^{-4}$ were computed until a final time $T = 5$ for equal grid sizes $N = 16 : 4 : 48$ in each coordinate, and the LPBC was computed in each coordinate with $K = N/2 - 1$. The time stepping is classical RK4 with Δt determined by (25) with CFL = 0.5 and $h_{min} = (1 - \cos(\pi/N))/2$.

Figure 1 shows the L^2 and L^∞ norm of the errors between the numerical and exact solutions for velocity and pressure. The numerical results provide clear evidence that the scheme is stable and the spatial spectral accuracy is observed for all physical variables. In particular, the spectral accuracy of the pressure variable shows that the numerical approximation to the pressure BC is stable and does not cause any artificial numerical boundary layer.
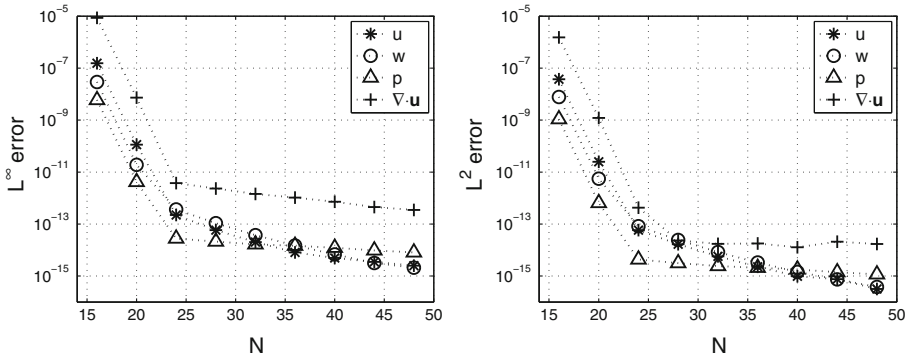
Fig. 1 L^∞ and L^2 and errors at $T = 5$ of the Chebyshev collocation scheme for the NSE with $N = 16:4:48$ and the local pressure boundary condition computed with $K = N/2 - 1$
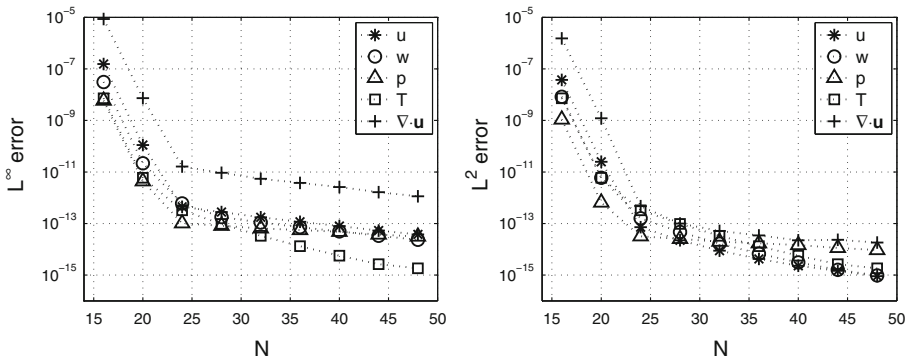

Fig. 2 L^∞ and L^2 and errors at $T = 5$ of the Chebyshev collocation scheme for the Boussinesq system with $N = 16:4:48$, the local pressure boundary condition computed with $K = N/2 - 1$, and the temperature boundary formula (34) with $K_{BC} = N/2 - 1$

5.2 Boussinesq Equations Accuracy Checks

An accuracy check can also be performed for the Chebyshev collocation method applied to the incompressible Boussinesq equations outlined in Sect. 4. Again, the exact profile for the velocity (38) and pressure (39), and the exact temperature profile is chosen as

$$T_e(x, y, z, t) = \frac{1}{2\pi} \cos(\pi x) \cos(\pi y) \cos(\pi z) \cos t, \tag{41}$$

which satisfies the no-flux boundary conditions on all six boundary sections.

With viscosity parameters $\nu = \kappa = 0.0001$ we compute solutions with grid sizes $16:4:48$ until a final time $T = 5$. Formulas are used to determine the pressure boundary condition and boundary value for temperature, respectively, with $K = N/2 - 1$, $K_{BC} = N/2 - 1$. Again, the time step Δt is determined from (25) with CFL $= 0.5$ and $h_{min} = (1 - \cos(\pi/N))/2$.

Figure 2 shows the L^2 and L^∞ norm of the errors between the numerical and exact solutions for velocity, pressure and temperature. As was the case for the NSE above, the spatial spectral accuracy is observed for all physical variables. In particular, the spectral accuracy in the L^∞ norm of the temperature shows that the temperature boundary formula (34) does not cause any instability of loss of accuracy, fully resolving the thermal boundary layer.

6 Simulation of the Three-Dimensional Lid-Driven Cavity Flow

In this section we present benchmark simulations of a three-dimensional lid-driven cavity flow, a detailed description of which can be found in [1,18,19].

The physical set-up is a viscous incompressible fluid in a cubic box $\Omega = [0, 1]^3$. The no-penetration, no-slip boundary condition is imposed on all six boundary surfaces, except for the left side,

$$\boldsymbol{u} = \left(0, 1, 0\right)^T, \quad \text{at} \quad x = 0.$$

The proposed collocation spectral method is applied to compute the lid-driven cavity flow for a sequence of Reynolds numbers: $Re = 100$ with a 24^3 grid, $Re = 400$ with a 32^3 grid, and $Re = 1,000$ with a 64^3 grid. In each case the LPBC was computed in each coordinate with $K = N/2 - 1$, where N is the respective grid size. The time stepping is classical RK4 with the time step Δt determined by (25) with CFL = 0.5 and $h_{min} = (1 - \cos(\pi/N))/2$. The computation was terminated in each case when a steady state was reached.

A plot of normal velocities u and v at centerlines $(x, 0.5, 0.5)$ and $(0.5, y, 0.5)$, respectively, is presented in Fig. 3. Furthermore, in Fig. 4 we present a comparison of the computed normal velocity profiles $Re = 1,000$ with a 64^3 grid at the centerlines $(x, 0.5, 0.5)$ and $(0.5, y, 0.5)$, showing excellent agreement with the benchmark data provided in [1] computed using a $96 \times 96 \times 64$ grid, further demonstrating the accuracy of our spectral method presented herein.

Appendix: A Derivative Formulas for Lagrange Interpolation Basis Function

Given K, let $\{l_k(y)\}_{k=0}^K$ denote the $K+1$ Lagrange interpolation basis functions based on the $K+1$ distinct points $\{y_j\}_{j=0}^K$. The functions are defined by $l_k(y) = \prod_{\substack{i=0 \\ i \neq k}}^K \dfrac{y - y_i}{y_k - y_i}$, and satisfy
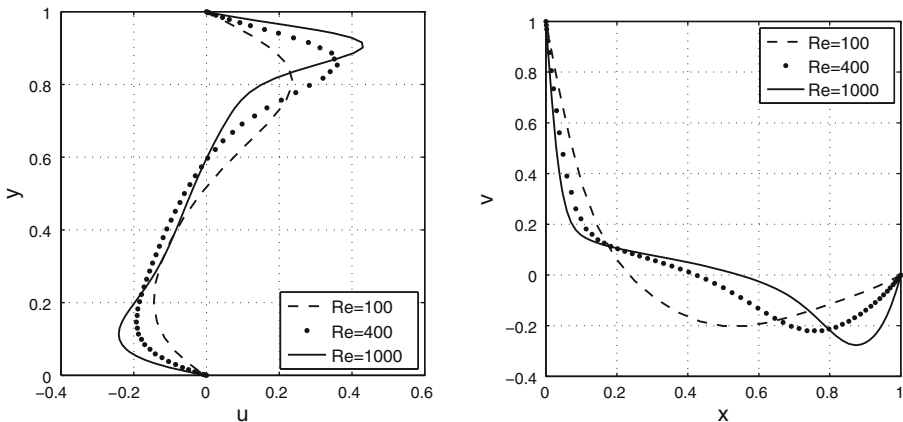

Fig. 3 Normal velocities on the center line $(x, 0.5, 0.5)$ and $(0.5, y, 0.5)$ in the cubic driven cavity in $[0, 1]^3$
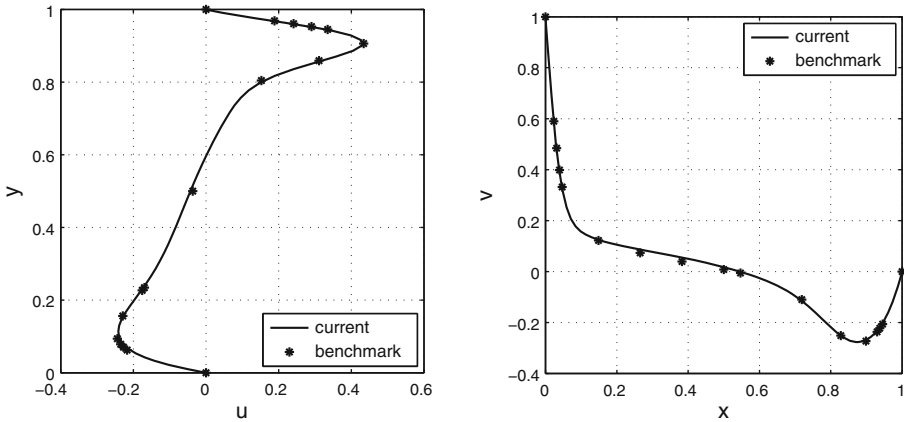
Fig. 4 Comparison between normal velocities as computed by the proposed scheme and benchmark data [1] for $Re = 1,000$

$l_k(y_j) = \delta_{k,j}$ for $0 \le k, j \le K$. A direct calculation gives

$$
l'_k(y_0) = \begin{cases} \sum_{m=1}^{K} (y_0 - y_m)^{-1}, & k = 0 \\ \prod_{\substack{i=0 \\ i \neq k}}^{K} (y_k - y_i)^{-1} \prod_{\substack{i=0 \\ i \neq k}}^{K} (y_0 - y_i), & k \neq 0 \end{cases}
\tag{42}
$$

and

$$
l''_k(y_0) = \begin{cases} \sum_{m=1}^{K} \sum_{\substack{n=1 \\ n \neq k}}^{K} [(y_0 - y_m)(y_0 - y_n)]^{-1}, & k = 0 \\ 2 \prod_{\substack{i=0 \\ i \neq k}}^{K} (y_k - y_i)^{-1} \sum_{\substack{n=1 \\ n \neq k}}^{K} \prod_{\substack{i=0 \\ i \neq k}}^{K} (y_0 - y_i), & k \neq 0. \end{cases}
\tag{43}
$$

References

1. Albensoeder, S., Kuhlmann, H.C.: Accurate three-dimensional lid-driven cavity flow. J. Comput. Phys. **206**, 536–558 (2005)
2. Bell, J., Colella, P., Glaz, H.: A second order projection method for the incompressible Navier–Stokes equations. J. Comput. Phys. **85**, 257–283 (1989)
3. Boyd, J.P.: Chebyshev and Fourier Spectral Methods, 2nd edn. Dover, New York (2001)
4. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: Spectral Methods in Fluid Dynamics. Springer, New York (1988)
5. Chorin, A.: Numerical solution of Navier–Stokes equations. Math. Comput. **22**, 745–762 (1968)
6. Deville, M., Fischer, P., Mund, E.: High-order Methods for Incompressible Fluid Flow. Cambridge University Press, Cambridge (2002)
7. Ehrenstein, U., Peyret, R.: A Chebyshev collocation method for the Navier–Stokes equations with application to double-diffusive convection. Int. J. Numer. Meth. Fluids **9**, 427–452 (1989)
8. Fornberg, B.: A Practical Guide to Pseudospectral Methods. Cambridge University Press, Cambridge (1996)
9. Gottlieb, D., Lutsman, L.: The spectrum of the Chebyshev collocation operator for the heat equation. SIAM J. Numer. Anal. **20**, 909–921 (1983)
10. Gresho, P.M., Sani, R.L.: On pressure boundary conditions for the incompressible Navier–Stokes equations. Intern. J. Numer. Methods Fluids **7**, 1111–1145 (1987)

11. Henshaw, W.D., Kreiss, H.-O., Reyna, L.G.: A fourth-order accurate difference approximation for the incompressible Navier–Stokes equations. Comput. Fluids **23**, 575–593 (1994)
12. Hesthaven, J., Gottlieb, S., Gottlieb, D.: Spectral Methods for Time-Dependent Problems. Cambridge University Press, Cambridge (2007)
13. Johnston, H., Liu, J.-G.: A finite difference scheme for incompressible flow based on local pressure boundary conditions. J. Comput. Phys. **180**, 120–154 (2002)
14. Johnston, H., Liu, J.-G.: Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term. J. Comput. Phys. **199**, 221–259 (2004)
15. Karniadakis, G.E., Sherwin, S.: Spectral/hp Element Methods for Computational Fluid Dynamics, 2nd edn. Oxford University Press, New York (2005)
16. Kim, J., Moin, P.: Application of a fractional-step method to incompressible Navier–Stokes equations. J. Comput. Phys. **59**, 308–323 (1985)
17. Kim, J., Moin, P., Moser, R.: Turbulence statistics in fully developed channel flow at low Reynolds number. J. Fluid Mech. **177**, 133–166 (1987)
18. Ku, H.C., Hirsh, R.S., Taylor, H.D.: A pseudo-spectral method for solution of three-dimensional incompressible Navier–Stokes equations. J. Comput. Phys. **70**, 439–462 (1987)
19. Leiriche, E.: Direct numerical simulation in a lid-driven cavity at high Reynolds number. J. Sci. Comput. **27**, 335–345 (2006)
20. Liu, J.-G., Liu, J., Pego, R.: Stability and convergence of efficient Navier–Stokes solver via a commutator estimate. Commun. Pure Appl. Math. **60**, 1443–1487 (2007)
21. Lynch, R.E., Rice, J.R., Thomas, D.H.: Direct solution of partial difference equations by tensor product methods. Numer. Math. **6**, 185–199 (1964)
22. Orszag, S.A., Israeli, M., Deville, M.: Boundary conditions for incompressible flows. J. Sci. Comput. **1**, 75–111 (1986)
23. Petersson, N.A.: Stability of pressure boundary conditions for Stokes and Navier–Stokes equations. J. Comput. Phys. **224**, 1064–1094 (2001)
24. Peyret, R.: Spectral Methods for Incompressible Viscous Flow. Springer, New York (2002)
25. Sani, R.L., Shen, J., Pironneau, O., Gresho, P.M.: Pressure boundary condition for the time-dependent incompressible Navier–Stokes equations. Intern. J. Numer. Methods Fluids **50**, 673–682 (2006)
26. Spalart, P.: Direct simulation of a turbulent boundary layer up to $R_\theta = 1410$. J. Fluid Mech. **187**, 61–98 (1988)
27. Spalart, P.: Theoretical and numerical study of a three-dimensional turbulent boundary layer. J. Fluid Mech. **205**, 319–340 (1989)
28. Temam, R.: Sur l'approximation de la solution des equation de Navier–Stokes Par la Méthode Des Fractionnarires II. Arch. Ration. Mech. Anal. **33**, 377–385 (1969)
29. Trefethen, L.N.: Spectral Methods in MATLAB. SIAM, Philadelphia (2001)