A fourth-order approximate projection method for the incompressible Navier-Stokes equations on locally-refined periodic domains

Qinghai Zhang

Department of Mathematics, University of Utah,
155 S. 1400 E, JWB 233, Salt Lake City, UT, 84112

Abstract

In this follow-up of our previous work [Zhang et. al., A fourth-order accurate finite-volume method with structured adaptive mesh refinement for solving the advection-diffusion equation, SIAM J. Sci. Comput. 34 (2012) B179-B201], the author proposes a high-order semi-implicit method for numerically solving the incompressible Navier-Stokes equations on locally-refined periodic domains. Fourth-order finite-volume stencils are employed for spatially discretizing various operators in the context of structured adaptive mesh refinement (AMR). Time integration adopts a fourth-order, semi-implicit, additive Runge-Kutta method to treat the non-stiff convection term explicitly and the stiff diffusion term implicitly. The divergence-free condition is fulfilled by an approximate projection operator. Altogether, these components yield a simple algorithm for simulating incompressible viscous flows on periodic domains with fourth-order accuracies both in time and in space. Results of numerical tests show that the proposed method is superior to previous second-order methods in terms of accuracy and efficiency. A major contribution of this work is the analysis of a fourth-order approximate projection operator.

Key words: Incompressible Navier-Stokes equations, Periodic domains, Approximate Projection, Adaptive mesh refinement, Additive Runge-Kutta method, Finite volume method.

Email address: qinghai@math.utah.edu; TEL: +1 (801)581-8174, FAX:+1 (801) 581-4148 (Qinghai Zhang).

Published in Appl. Numer. Math.77 (2014) 16-30 *Last updated : 21 August 2020*

1 Introduction

The Navier-Stokes equations are at the center of modern mathematical physics as they govern an enormous range of common phenomena such as ocean currents, blood flow, the atmosphere, and turbulence. For incompressible fluids, the non-dimensional form of these equations is

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{g} - \nabla p + \nu \Delta \mathbf{u}, \tag{1a}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{1b}$$

where $t \in [0, +\infty)$ is time, $\mathbf{x} \in \mathbb{R}^{\mathrm{D}}\,(\mathrm{D} = 2, 3)$ the spatial location, $\mathbf{g}$ the external forcing term, p the nondimensionalized pressure, $\mathbf{u}$ the velocity, $\nu = 1/\mathrm{Re}$, and Re the Reynolds number. (1) is also known as the incompressible Navier-Stokes equations (INSE).

Among the numerous computational methods for the INSE, the projection method introduced by Chorin [9] first computes an auxiliary velocity field $\mathbf{u}^*$ from the momentum equation by ignoring the pressure gradient term and then project $\mathbf{u}^*$ onto the divergence-free space to fulfill the incompressibility constraint. During the last 45 years, many variants of the original projection method have been proposed, some successful examples are those of Kim and Moin [19], Bell et al. [5], Orszag et al. [24], and E and Liu [12].

One attractive feature of the projection methods is its efficiency: during one time step it is only necessary to solve a sequence of elliptic equations either for the velocity or for the pressure. Alternatively, one can solve for the velocity and the pressure simultaneously by forming the linear system as a saddle point problem [6]. For this approach, Griffith [14] proposes to use the projection method as a preconditioner instead of a solver. Indeed, the linear system after the preconditioning has extremely clustered eigenvalues and can be solved by a Krylov subspace method within a few iterations. One advantage of this approach is the ease of imposing a variety of physical boundary conditions.

In comparison to second-order projection methods, the literature on fourth-order methods is much thinner; see, e.g., [17,16]. There also exist high-order finite difference/volume methods [15,25] that do not belong to the class of projection methods. However, most of these methods have explicit time integration, which incurs unnecessary restriction on the time-step size from the diffusion term. Shukla et. al. [26] coupled very high order spatial discretizations with a second-order semi-implicit time integrator, the overall accuracy for transient flows is still of second order. It is therefore desirable to have a high-order method of which the spatial and temporal accuracies are comparable and of which the time-step size is not constrained by the diffusion term.

In this paper the author proposes such a method for solving INSE with fourth-order accuracies both in time and in space, based on our previous work on the advection-diffusion equation [30]. In the proposed method, spatial discretization employs classical finite-volume stencils while the temporal integration adapts a semi-implicit additive Runge-Kutta (ARK) method [18], which admits explicit treatment of the non-stiff convection term and implicit treatment of the stiff diffusion term. As such, the time-step size is only constrained by the convection term. To fulfill the incompressibility constraint, an approximate projection operator is analyzed for its stability and accuracy. On periodic domains, combining these components yields a straightforward algorithm, thanks to the commutativity of the discrete Laplacian and projection operators. The flexibility and efficiency of the proposed method is further enhanced by implementing it in the context of structured adaptive mesh refinement (AMR). One significance of the AMR implementation is that the proposed method cannot be substituted by another high-order spectral method.

The rest of this paper is organized as follows. Section 2 contains fourth-order finite-volume discretization of various operators. In particular, the fourth-order approximate projection operator is analyzed for its stability and accuracy in Section 2.2 and single-level discrete operators are generalized to multiple-level AMR hierarchies in Section 2.3. In Section 3, the fourth-order method for solving the INSE on periodic domains is proposed based on the IMEX scheme, a particular type of the ARK method. In Section 4, two benchmark problems are numerically simulated using the proposed fourth-order method, showing that it is superior to a second-order method in terms of efficiency and accuracy. Section 5 concludes this paper with a prospect of future research.

2 Spatial discretization

The rectangular problem domain Ω is discretized into a collection of rectangular grid cells. Denote a grid cell by a multi-index $\mathbf{i} \in \mathbb{Z}^{\mathrm{D}}$, its region is represented by

$$\mathcal{C}_{\mathbf{i}} = \Big[\mathbf{x}_O + \mathbf{i}h,\ \mathbf{x}_O + (\mathbf{i} + \mathbb{1})\, h\Big], \tag{2}$$

and the region of the higher face of cell $\mathbf{i}$ in dimension d by

$$\mathcal{F}_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \Big[\mathbf{x}_O + \Big(\mathbf{i} + \mathbf{e}^d\Big)\, h,\ \mathbf{x}_O + (\mathbf{i} + \mathbb{1})\, h\Big], \tag{3}$$

where $\mathbf{x}_O \in \mathbb{R}^{\mathrm{D}}$ is some fixed origin of the coordinates, h the uniform mesh spacing, $\mathbb{1} \in \mathbb{Z}^{\mathrm{D}}$ the multi-index with all its components equal to one, and $\mathbf{e}^d \in \mathbb{Z}^{\mathrm{D}}$ a multi-index with 1 as its d-th component and 0 otherwise. Throughout this paper, regular lowercase letters denote scalars, boldface lowercase letters denote vectors, and boldface uppercase letters denote operators.

The author strictly distinguishes three different types of quantities, viz. point values, cell averages, and face averages. Point values are denoted by naked symbols with the subscripts indicating their locations, e.g. $\phi_{\mathbf{i}}$ and $\phi_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d}$ are the values of ϕ at the centers of the cell $\mathbf{i}$ and the face $\mathbf{i}+\frac{1}{2}\mathbf{e}^d$, respectively. A symbol within a pair of angle brackets is a cell/face average if the subscript is an integer/fraction. For example, the averaged ϕ over cell $\mathbf{i}$ is denoted by

$$\langle\phi\rangle_{\mathbf{i}} = \frac{1}{h^{\mathrm{D}}}\int_{\mathcal{C}_{\mathbf{i}}} \phi\left(\mathbf{x}\right)\,\mathrm{d}\mathbf{x}, \tag{4}$$

and the averaged ϕ over the face $\mathbf{i}+\frac{1}{2}\mathbf{e}^d$ by

$$\langle\phi\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \frac{1}{h^{\mathrm{D}-1}}\int_{\mathcal{F}_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d}} \phi\left(\mathbf{x}\right)\,\mathrm{d}\mathbf{x}. \tag{5}$$

Point values can be converted to face-averaged and cell-averaged quantities and vice versa by Taylor expansions, e.g., expanding the integrands in (4) and (5) yields

$$\langle\phi\rangle_{\mathbf{i}} = \phi_{\mathbf{i}} + \frac{h^2}{24}\sum_{d=1}^{\mathrm{D}} \left.\frac{\partial^2\phi(\mathbf{x})}{\partial x_d^2}\right|_{\mathbf{i}} + O(h^4); \tag{6a}$$

$$\langle\phi\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \phi_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} + \frac{h^2}{24}\sum_{d'\neq d} \left.\frac{\partial^2\phi(\mathbf{x})}{\partial x_{d'}^2}\right|_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} + O(h^4). \tag{6b}$$

The relation between cell-averaged quantities and face-averaged quantities can be derived via the first fundamental theorem of calculus; two formulas used throughout this work are

$$\langle\phi\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \frac{1}{12}\left(-\langle\phi\rangle_{\mathbf{i}+2\mathbf{e}^d} + 7\langle\phi\rangle_{\mathbf{i}+\mathbf{e}^d} + 7\langle\phi\rangle_{\mathbf{i}} - \langle\phi\rangle_{\mathbf{i}-\mathbf{e}^d}\right) + O(h^4), \tag{7a}$$

$$\left\langle\frac{\partial\phi}{\partial x_d}\right\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \frac{1}{12h}\left(-\langle\phi\rangle_{\mathbf{i}+2\mathbf{e}^d} + 15\langle\phi\rangle_{\mathbf{i}+\mathbf{e}^d} - 15\langle\phi\rangle_{\mathbf{i}} + \langle\phi\rangle_{\mathbf{i}-\mathbf{e}^d}\right) + O(h^4). \tag{7b}$$

See Appendix A of [30] for a derivation of (7).

Since the leading term of the difference between cell-centered and cell-averaged quantities is $O(h^2)$, second-order methods do not distinguish these two centerings. In contrast, fourth-order methods have to treat them differently for the targeted accuracy. Furthermore, finite-volume operators acting on cell- or face-averages are different from their finite-difference counterparts acting on point values. The difference between the three types of quantities account for much of the complexity of fourth-order finite-volume methods that is absent in second-order methods.

2.1 Fourth-order discrete operators on single-level grids

Following [30], the discrete gradient operator and the discrete divergence operator are defined as

$$\mathbf{G}_d \langle \phi \rangle_{\mathbf{i}} = \frac{1}{12h} \Big(- \langle \phi \rangle_{\mathbf{i}+2\mathbf{e}^d} + 8 \langle \phi \rangle_{\mathbf{i}+\mathbf{e}^d} - 8 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} + \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} \Big), \tag{8}$$

$$\mathbf{D} \langle \mathbf{u} \rangle_{\mathbf{i}} = \frac{1}{12h} \sum_d \Big(- \langle u_d \rangle_{\mathbf{i}+2\mathbf{e}^d} + 8 \langle u_d \rangle_{\mathbf{i}+\mathbf{e}^d} - 8 \langle u_d \rangle_{\mathbf{i}-\mathbf{e}^d} + \langle u_d \rangle_{\mathbf{i}-2\mathbf{e}^d} \Big). \tag{9}$$

The discrete divergence operator also acts on tensor averages,

$$\mathbf{D} \langle \mathbf{u}\mathbf{u} \rangle_{\mathbf{i}} = \frac{1}{h} \sum_d \left(\mathbf{F} \langle u_d, \mathbf{u} \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} - \mathbf{F} \langle u_d, \mathbf{u} \rangle_{\mathbf{i}-\frac{1}{2}\mathbf{e}^d} \right), \tag{10}$$

where the discrete face average of two scalar functions is

$$\mathbf{F} \langle \phi, \psi \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \langle \phi \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \langle \psi \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} + \frac{h^2}{12} \sum_{d' \neq d} \left(\mathbf{G}_{d'}^{\perp} \phi \right)_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} \left(\mathbf{G}_{d'}^{\perp} \psi \right)_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d}, \tag{11}$$

and $\mathbf{G}_{d'}^{\perp}$ is the discrete gradient operator in the transverse directions,

$$\left(\mathbf{G}_{d'}^{\perp} \phi \right)_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \frac{1}{2h} \left(\langle \phi \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d+\mathbf{e}^{d'}} - \langle \phi \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d-\mathbf{e}^{d'}} \right). \tag{12}$$

Second-order methods may completely ignore the transverse gradients $\mathbf{G}_{d'}^{\perp}$. In contrast, a discretization of $\mathbf{G}_{d'}^{\perp}$ with at least second-order accuracy is necessary to achieve a fourth-order discretization of $\langle \phi\psi \rangle$. These additional terms are another reason for the complexity of the fourth-order methods.

The discrete Laplacian is defined as

$$\mathbf{L} \langle \phi \rangle_{\mathbf{i}} = \frac{1}{12h^2} \sum_d \Big(- \langle \phi \rangle_{\mathbf{i}+2\mathbf{e}^d} + 16 \langle \phi \rangle_{\mathbf{i}+\mathbf{e}^d} - 30 \langle \phi \rangle_{\mathbf{i}} + 16 \langle \phi \rangle_{\mathbf{i}-\mathbf{e}^d} - \langle \phi \rangle_{\mathbf{i}-2\mathbf{e}^d} \Big). \tag{13}$$

It is clear that $\mathbf{L} \neq \mathbf{DG}$ since the stencil width of $\mathbf{L}$ is five while that of $\mathbf{DG}$ is nine.

It is emphasized here that the aforementioned operators act on *cell averages*, therefore the coefficients in the stencils (7), (8), (9), and (13) differ from their counterparts of the finite difference formulation.

$\mathbf{G} \langle \phi \rangle$, $\mathbf{D} \langle \mathbf{u} \rangle$, $\mathbf{L} \langle \mathbf{u} \rangle$, and $\mathbf{D} \langle \mathbf{u}\mathbf{u} \rangle$ approximate the cell-averaged gradient, velocity divergence, Laplacian, and convection, respectively. $\mathbf{L}$ applies componentwise to $\langle \mathbf{u} \rangle$ while computing a component of the convection term involves all

components of the velocity vector. They are identical to those in our previous work [30]; their fourth-order accuracies are summarized as follows.

Proposition 1 *The operators defined in (8), (9), (13), (10), and (11) are fourth-order accurate:*

$$\mathbf{G}_d \langle \phi \rangle_{\mathbf{i}} = \frac{1}{h^{\mathrm{D}}} \int_{\mathcal{C}_{\mathbf{i}}} \frac{\partial \phi}{\partial x_d} + O(h^4), \tag{14a}$$

$$\mathbf{D} \langle \mathbf{u} \rangle_{\mathbf{i}} = \frac{1}{h^{\mathrm{D}}} \int_{\mathcal{C}_{\mathbf{i}}} \nabla \cdot \mathbf{u} + O(h^4), \tag{14b}$$

$$\mathbf{L} \langle \phi \rangle_{\mathbf{i}} = \frac{1}{h^{\mathrm{D}}} \int_{\mathcal{C}_{\mathbf{i}}} \nabla^2 \phi + O(h^4), \tag{14c}$$

$$\mathbf{D} \langle \mathbf{u} \rangle_{\mathbf{i}} = O(h^4) \;\Rightarrow\; \mathbf{D} \langle \mathbf{u}\mathbf{u} \rangle_{\mathbf{i}} = \frac{1}{h^{\mathrm{D}}} \int_{\mathcal{C}_{\mathbf{i}}} (\mathbf{u} \cdot \nabla)\mathbf{u} + O(h^4), \tag{14d}$$

$$\mathbf{F} \langle \phi, \psi \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} = \frac{1}{h^{\mathrm{D}-1}} \int_{\mathcal{F}_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d}} \phi\psi + O(h^4) \tag{14e}$$

PROOF. Equation (14a) follows from (7a), the second fundamental theorem of calculus, and the fact that $\frac{\partial}{\partial x_d}$ commutes with $\langle \cdot \rangle$. Equation (14b) follows from the divergence theorem and (7a). Equations (14c) and (14e) are proved in Appendix B of [30]. Finally,

$$\begin{aligned} \frac{1}{h^{\mathrm{D}}} \int_{\mathcal{C}_{\mathbf{i}}} (\mathbf{u} \cdot \nabla)\mathbf{u} &= \frac{1}{h^{\mathrm{D}}} \int_{\mathcal{C}_{\mathbf{i}}} \nabla \cdot (\mathbf{u}\mathbf{u}) - \frac{1}{h^{\mathrm{D}}} \int_{\mathcal{C}_{\mathbf{i}}} (\nabla \cdot \mathbf{u})\mathbf{u} \\ &= \frac{1}{h} \sum_d \mathbf{e}^d \cdot \left(\langle \mathbf{u}\mathbf{u} \rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}^d} - \langle \mathbf{u}\mathbf{u} \rangle_{\mathbf{i}-\frac{1}{2}\mathbf{e}^d} \right) + O(h^4), \end{aligned}$$

where we have applied the chain rule, the condition $\mathbf{D}\langle \mathbf{u} \rangle_{\mathbf{i}} = O(h^4)$, (14b), and the divergence theorem. Equation (14d) then follows from (10) and (14e).

2.2 Fourth-order approximate projection on periodic domains

A projection operator $\bar{\mathbf{P}}$ is a linear transformation from a vector space to itself satisfying the idempotent condition $\bar{\mathbf{P}}^2 = \bar{\mathbf{P}}$. From $\mathbf{D}$ and $\mathbf{G}$ defined in the previous subsection, a projection operator can be constructed as

$$\mathbf{P}_E = \mathbf{I} - \mathbf{G}(\mathbf{D}\mathbf{G})^{-1}\mathbf{D}. \tag{15}$$

Unfortunately, the expanded stencil of $\mathbf{D}\mathbf{G}$ leads to two serious defects of $\mathbf{P}_E$. First, numerical evaluation of $\mathbf{P}_E$ requires four layers of ghost cells. More importantly, the coefficients in the stencil of $\mathbf{D}\mathbf{G}$ make the solutions susceptible to spurious oscillations. These drawbacks of $\mathbf{D}\mathbf{G}$ have motivated the

development of *approximate* projection. The main idea of this approach is to approximate $\mathbf{DG}$ by another operator so that the resulting projection operator is robust, compact, and accurate enough. As the trade-off, the idempotent condition is fulfilled only approximately; see [21], [2], and references therein for more discussions.

Most of the previous approximate projection operators are second-order accurate. In this work, $\mathbf{L}$ is used as an approximation of $\mathbf{DG}$ to define the following approximate projection operator,

$$\mathbf{P} = \mathbf{I} - \mathbf{G}\mathbf{L}^{-1}\mathbf{D}. \tag{16}$$

In the rest of this subsection, the author shows that, on periodic domains, $\mathbf{P}$ commutes with the discrete Laplacian $\mathbf{L}$ (Lemma 4), is stable and fourth-order accurate (Theorem 5).

To represent integrals over the partitioned computational domain, a scalar inner product and a vector inner product are respectively defined as

$$\begin{aligned} \langle \phi, \psi \rangle_S &= h^{\mathrm{D}} \textstyle\sum_{\mathbf{i}} \langle \phi \rangle_{\mathbf{i}} \langle \psi \rangle_{\mathbf{i}}, \\ \langle \mathbf{v}, \mathbf{w} \rangle_V &= h^{\mathrm{D}} \textstyle\sum_{\mathbf{i}} \langle \mathbf{v} \rangle_{\mathbf{i}} \cdot \langle \mathbf{w} \rangle_{\mathbf{i}}. \end{aligned} \tag{17}$$

Proposition 2 *On periodic domains,* $\mathbf{G}$ *and* $-\mathbf{D}$ *as defined in (8) and (9) are* adjoint *in the sense that*

$$\langle \mathbf{D}\mathbf{u}, \phi \rangle_S = - \langle \mathbf{u}, \mathbf{G}\phi \rangle_V . \tag{18}$$

The corresponding matrices satisfy $\mathbf{G} = -\mathbf{D}^T$.

PROOF. It suffices to show

$$\langle \mathbf{D}\mathbf{u}, \phi \rangle_S + \langle \mathbf{u}, \mathbf{G}\phi \rangle_V = h^{\mathrm{D}} \textstyle\sum_{\mathbf{i}} \Big(\langle \phi \rangle_{\mathbf{i}} \, \mathbf{D} \, \langle \mathbf{u}_{\mathbf{i}} \rangle + \langle \mathbf{u} \rangle_{\mathbf{i}} \cdot \mathbf{G} \, \langle \phi \rangle_{\mathbf{i}} \Big) = 0.$$

Consider all possible terms containing $\langle \phi \rangle_{\mathbf{i}}$. For dimension d, $\sum_{\mathbf{j}} \Big(\langle \phi \rangle_{\mathbf{j}} \, \mathbf{D} \, \langle \mathbf{u} \rangle_{\mathbf{j}} \Big)$ expands to $\langle \phi \rangle_{\mathbf{i}} \Big(8 \, \langle u_d \rangle_{\mathbf{i}+\mathbf{e}^d} - \langle u_d \rangle_{\mathbf{i}+2\mathbf{e}^d} - 8 \, \langle u_d \rangle_{\mathbf{i}-\mathbf{e}^d} + \langle u_d \rangle_{\mathbf{i}-2\mathbf{e}^d} \Big)$. Similarly, $\sum_{\mathbf{j}} \Big(\langle \mathbf{u} \rangle_{\mathbf{j}} \cdot \mathbf{G} \, \langle \phi \rangle_{\mathbf{j}} \Big)$ expands to $\langle \phi \rangle_{\mathbf{i}} \Big(-8 \, \langle u_d \rangle_{\mathbf{i}+\mathbf{e}^d} + \langle u_d \rangle_{\mathbf{i}+2\mathbf{e}^d} + 8 \, \langle u_d \rangle_{\mathbf{i}-\mathbf{e}^d} - \langle u_d \rangle_{\mathbf{i}-2\mathbf{e}^d} \Big)$. In the former, all the terms are contributed by $\langle \phi \rangle_{\mathbf{i}} \, \mathbf{D} \, \langle \mathbf{u} \rangle_{\mathbf{i}}$; in the latter, no terms come from $\langle \mathbf{u} \rangle_{\mathbf{i}} \cdot \mathbf{G} \, \langle \phi \rangle_{\mathbf{i}}$, e.g., $-8 \langle \phi \rangle_{\mathbf{i}} \, \langle u_d \rangle_{\mathbf{i}+\mathbf{e}^d}$ is contributed by $\langle \mathbf{u} \rangle_{\mathbf{i}+\mathbf{e}^d} \cdot \mathbf{G} \, \langle \phi \rangle_{\mathbf{i}+\mathbf{e}^d}$. Because of periodicity, all five cell-indices are well-defined for the cells to remain inside the domain, hence these above terms cancel. The same argument can also be applied to the terms containing $\langle u_d \rangle_{\mathbf{i}}$ for all d.

Corollary 3 *On periodic domains, the corresponding matrix of the approximate projection operator* $\mathbf{P}$ *defined in (16) is symmetric, i.e.* $\mathbf{P}^T = \mathbf{P}$.

PROOF. The discrete Laplacian in (13) is clearly symmetric for periodic domains. The rest follows from PROPOSITION 2 and (16).

Lemma 4 *On period domains, the discrete gradient* $\mathbf{G}$ *in (8) and discrete Laplacian* $\mathbf{L}$ *in (13) commute,*

$$\mathbf{GL} = \mathbf{LG}. \tag{19}$$

Consequently, the discrete Laplacian and the approximate projection commute,

$$\mathbf{PL} = \mathbf{LP}. \tag{20}$$

PROOF. For a 1D periodic domain, let $\bar{\mathbf{G}}$ and $\bar{\mathbf{L}}$ denote the matrices of the discrete gradient operator and Laplacian operator scaled by $12h$ and $12h^2$, respectively. From (8) and (13), we have

$$\bar{\mathbf{G}}_{i,j} = \begin{cases} \pm 8, & j = \mathrm{mod}(i \pm 1, m) \\ \mp 1, & j = \mathrm{mod}(i \pm 2, m) \\ 0, & \text{otherwise} \end{cases}; \qquad \bar{\mathbf{L}}_{i,j} = \begin{cases} -30, & j = i \\ 16, & j = \mathrm{mod}(i \pm 1, m) \\ -1, & j = \mathrm{mod}(i \pm 2, m) \\ 0, & \text{otherwise} \end{cases},$$

where m is the number of cells. To avoid clustering of notation, I drop "mod" in the indices of matrix entries to use the cyclic shorthands "$i \pm \cdot$" for "$\mathrm{mod}(i \pm \cdot, m)$," "$k \pm \cdot$" for "$\mathrm{mod}(k \pm \cdot, m)$," and so on. It follows that

$$\begin{aligned} (\bar{\mathbf{G}}\bar{\mathbf{L}})_{k,\ell} &= \textstyle\sum_{j=k-2}^{k+2} \bar{\mathbf{G}}_{k,j}\bar{\mathbf{L}}_{j,\ell} = -\bar{\mathbf{L}}_{k+2,\ell} + 8\bar{\mathbf{L}}_{k+1,\ell} - 8\bar{\mathbf{L}}_{k-1,\ell} + \bar{\mathbf{L}}_{k-2,\ell}, \\ (\bar{\mathbf{L}}\bar{\mathbf{G}})_{k,\ell} &= \textstyle\sum_{j=k-2}^{k+2} \bar{\mathbf{L}}_{k,j}\bar{\mathbf{G}}_{j,\ell} = -\bar{\mathbf{L}}_{k,\ell-2} + 8\bar{\mathbf{L}}_{k,\ell-1} - 8\bar{\mathbf{L}}_{k,\ell+1} + \bar{\mathbf{L}}_{k,\ell+2}. \end{aligned}$$

$\bar{\mathbf{L}}$ is a Toeplitz matrix, hence $\bar{\mathbf{L}}_{k+2,\ell} = \bar{\mathbf{L}}_{k,\ell-2}$, $\bar{\mathbf{L}}_{k+1,\ell} = \bar{\mathbf{L}}_{k,\ell-1}$, etc. Thus we have

$$\bar{\mathbf{G}}\bar{\mathbf{L}} = \bar{\mathbf{L}}\bar{\mathbf{G}}. \tag{21}$$

For a 2D domain, the matrices of the discrete operators can be expressed as Kronecker products of their 1D counterparts and identity matrices [11, Chapter 6.3.3]:

$$12h^2\mathbf{L} = \bar{\mathbf{L}} \otimes \mathbf{I} + \mathbf{I} \otimes \bar{\mathbf{L}}, \qquad 12h\mathbf{G} = \left(\bar{\mathbf{G}} \otimes \mathbf{I},\ \mathbf{I} \otimes \bar{\mathbf{G}}\right)^T. \tag{22}$$

In order to prove (19), it suffices to show that $\mathbf{L}$ commutes with each subblock of $\mathbf{G}$, which follows from (21) and the mixed-product property of Kronecker products, i.e. $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$. For three and higher dimensions, (19) can be proved by a straightforward induction based on the 1D and 2D arguments.

Finally, (20) follows directly from (16), (19), and PROPOSITION 2.

Theorem 5 *On periodic domains, both the spectral radius and the Euclidean 2-norm of the approximate projection operator are one,*

$$\rho(\mathbf{P}) = \|\mathbf{P}\|_2 = 1. \tag{23}$$

Furthermore, $\mathbf{P}$ *is a fourth-order approximation to the continuous projection operator,*

$$\mathbf{P}\left\langle \mathbf{u} \right\rangle_{\mathbf{i}} - \frac{1}{h^{\mathrm{D}}} \int_{\mathcal{C}_{\mathbf{i}}} \left(\mathbf{I} - \nabla(\Delta^{-1})\nabla\cdot \right) \mathbf{u} = O(h^4). \tag{24}$$

PROOF. It follows from COROLLARY 3 that $\rho(\mathbf{P}) = 1$ implies $\|\mathbf{P}\|_2 = 1$, hence we only need to show the former. For the projection $\mathbf{P}_E$ in (15), we have $\lambda(\mathbf{P}_E) \in \{0, 1\}$ since its *minimal polynomial* is $\mathbf{P}_E^2 - \mathbf{P}_E = 0$. Let $\mathbf{P}_E = \mathbf{I} - \mathbf{Q}_E$, $\mathbf{P} = \mathbf{I} - \mathbf{Q}$. Clearly, $\lambda(\mathbf{Q}_E) \in \{0, 1\}$, and we only need to show $0 \leq \lambda(\mathbf{Q}) \leq 1$. By LEMMA 4, $\mathbf{G}$ commutes with both $\mathbf{L}^{-1}$ and $\mathbf{DG}$. Thus we have

$$\mathbf{Q} = \mathbf{G}\mathbf{L}^{-1}(\mathbf{DG})(\mathbf{DG})^{-1}\mathbf{D} = \mathbf{L}^{-1}(\mathbf{DG})\mathbf{G}(\mathbf{DG})^{-1}\mathbf{D} = \mathbf{L}^{-1}\mathbf{DG}\mathbf{Q}_E,$$

and it suffices to show

$$0 \leq \lambda\left(\mathbf{L}^{-1}\mathbf{DG}\right) \leq 1.$$

Using discrete Fourier analysis, we can define the shift operator as

$$s_d \left\langle \phi \right\rangle_{\mathbf{i}} = \left\langle \phi \right\rangle_{\mathbf{i}+\mathbf{e}^d}, \tag{25}$$

whose eigenvectors are the single Fourier modes with the eigenvalues $e^{\mathbf{i}\beta_d}$, where $\beta_d = \kappa_d \frac{\pi}{N}$, $\kappa_d = 1, \cdots, N-1$, and N the total number of points in dimension d. It follows from (8), (9), and (13) that for a given Fourier component,

$$\lambda\left(\mathbf{DG}\right) = -\tfrac{4}{h^2} \textstyle\sum_{d=1}^{\mathrm{D}} \sin^2 \tfrac{\beta_d}{2} \left(1 - \sin^2 \tfrac{\beta_d}{2}\right) \left(1 + \tfrac{2}{3}\sin^2 \tfrac{\beta_d}{2}\right)^2;$$
$$\mathbf{L} = \tfrac{1}{12h^2} \sum_{d=1}^{\mathrm{D}} \left(16 s_d + \tfrac{16}{s_d} - 30 - s_d^2 - \tfrac{1}{s_d^2}\right) \Rightarrow \ \lambda(\mathbf{L}) = -\tfrac{4}{h^2} \sum_{d=1}^{\mathrm{D}} \sin^2 \tfrac{\beta_d}{2} \left(1 + \tfrac{1}{3}\sin^2 \tfrac{\beta_d}{2}\right).$$

$\lambda\left(\mathbf{L}^{-1}\mathbf{DG}\right) \geq 0$ follows from the negative definiteness of $\lambda(\mathbf{L})$ and $\lambda(\mathbf{DG})$. $\lambda\left(\mathbf{L}^{-1}\mathbf{DG}\right) \leq 1$ holds because

$$\eta(1-\eta)\left(1 + \tfrac{2}{3}\eta\right)^2 - \eta\left(1 + \tfrac{1}{3}\eta\right) = -\tfrac{4}{9}\eta^2(2\eta + \eta^2) \leq 0,$$

for $\eta = \sin^2 \frac{\beta_d}{2} \in [0, 1]$.

Finally, (24) follows directly from (14a), (14b), and (14c) in PROPOSITION 1; it can also be proved via considering the Taylor expansions of the symbols of individual operators for any *fixed* Fourier mode [21, Chapter 3].

The spectral radius of $\mathbf{P}$ being not bigger than unity is both necessary and sufficient for numerical stability of $\mathbf{P}$ if the discretized system is linear. However, in the context of INSE, $\rho(\mathbf{P}) = 1$ alone is not sufficient for numerical stability because the nonlinear convection term changes the spectrum of the velocity at every time step. Consequently, the total kinetic energy of the velocity field might increase over time if the 2-norm of $\mathbf{P}$ is larger than unity; more precisely, $\|\mathbf{P}\|_2 > 1$ implies $\|\mathbf{P}\langle\mathbf{u}\rangle\|_2 > \|\langle\mathbf{u}\rangle\|_2$ for some $\langle\mathbf{u}\rangle$. Therefore both $\rho(\mathbf{P}) \leq 1$ and $\|\mathbf{P}\|_2 \leq 1$ are needed to guarantee the stability of the projection operator in the context of INSE.

2.3 Locally-refined grids

This subsection generalizes discrete spatial operators to AMR hierarchies with multiple levels. Most of the AMR techniques are already described in [30] as well as earlier works such as [23], they are briefly summarized here in the context of periodic domains.

The rectangular problem domain Ω is covered simultaneously by a number of discretizations, each of which is a multi-index set representing all the control volumes at the corresponding discretization:

$$\Upsilon = \left\{ \Upsilon^\ell : \Upsilon^\ell \subset \mathbb{Z}^D,\ \Upsilon^{\ell+1} = \mathscr{C}_r^{-1}\Upsilon^\ell,\ \ell = 0, 1, \cdots, \ell^{\max} \right\}. \tag{26}$$

Inductively, Υ^0 is the coarsest discretization and a fine discretization results from refining its coarser. The coarsening operator $\mathscr{C}_r : \mathbb{Z}^{\mathrm{D}} \to \mathbb{Z}^{\mathrm{D}}$ relates geometric regions and variables between two adjacent levels via $\mathscr{C}_r(\mathbf{i}) = \lfloor \mathbf{i}/r \rfloor$, where the floor operator applies component-wise to $\mathbf{i}$, and r is the *refinement ratio* between two successive discretizations. For simplicity r is assumed to be a constant across all levels.

The AMR hierarchy Ω_Υ consists of a number of *level*s, each of which is a subset of its corresponding discretization, and can be decomposed into a disjoint union of *patch*es, or *box*es:

$$\begin{aligned} \Omega_\Upsilon &= \left\{ \Omega^\ell :\ \Omega^0 = \Upsilon^0;\ \forall \ell > 0,\ \Omega^\ell \subset \Upsilon^\ell \right\}, \\ \Omega^\ell &= \left\{ \Omega_k^\ell :\ k \neq k' \Leftrightarrow \Omega_k^\ell \cap \Omega_{k'}^\ell = \emptyset \right\}, \\ \Omega_k^\ell(\mathbf{j}, \bar{\mathbf{j}}) &= \left\{ \mathbf{i} \in \mathbb{Z}^{\mathrm{D}} :\ \min(\mathbf{j}, \bar{\mathbf{j}}) \leq \mathbf{i} \leq \max(\mathbf{j}, \bar{\mathbf{j}}) \right\}, \end{aligned} \tag{27}$$

where each box Ω_k^ℓ is a rectangular region uniquely determined by two multi-indices. Following the usual multi-index notation, min and max apply component-wise. On periodic domains, the discretization Υ is shifted to tile the entire Euclidean space, along with the relative position of AMR levels Ω_Υ fixed within it. As a practical note, any box might be divided into smaller boxes for load

balancing in parallel computing, hence the size of Υ^0 is almost always chosen to be a power of two, so is the refinement ratio r.

To facilitate discrete operator evaluation, each patch is augmented by wrapping it with n_{ghost} layers of ghost cells [1]. The width of the discrete operators in Section 2.1 dictates that n_{ghost} be no less than two. To minimize the communication in parallel computing, $n_{\text{ghost}} = 2$ is used in this work.

Due to the absence of physical boundaries on periodic domains, there are only two types of ghost cells: those covered by patches of the same level and those close to a coarse-fine interface, denoted by $\mathcal{G}^\ell_{\text{exchange}}$ and $\mathcal{G}^\ell_{\text{CFI}}$, respectively. The former type of ghost cells are easily filled by inter-patch exchange, i.e., copying values from adjacent patches. In comparison, those in $\mathcal{G}^\ell_{\text{CFI}}$ must be filled by coarse-fine interpolation (CFI).

Different from the least-square-based CFI algorithm used in [30], AMRCFI, [29,28], an efficient, generic, and conservative interpolation algorithm based on multi-dimensional polynomials, is employed in this work. Fifth-order [2] formulas are used to fill the ghost cells for evaluating the discrete Laplacian, while fourth-order formulas for the convection operator. The details will not be repeated here as both the algorithms [29] and the source code [28] are readily available.

Across different levels of the AMR hierarchy, a physical variable might have multiple values at the same location. This ambiguity is resolved by defining the computed solution only on the *valid region*s,

$$\Omega^\ell_{\text{valid}} = \Omega^\ell \setminus \mathscr{C}_r(\Omega^{\ell+1}). \tag{28}$$

The evaluation of an operator $\mathcal{L}$ on a *periodic* AMR hierarchy can now be summarized as follows:

(OpAMR.1) for each level $\ell = 0, 1, \ldots, \ell^{\max}$, fill the ghost cells $\mathcal{G}^\ell_{\text{exchange}}$ and $\mathcal{G}^\ell_{\text{CFI}}$ by inter-patch exchange and AMRCFI [29,28],
(OpAMR.2) evaluate $\mathcal{L}$ for each box Ω^ℓ_k,
(OpAMR.3) if the operator can be expressed as the divergence of a vector, perform refluxing [30] for each level $\ell = \ell^{\max}, \ell^{\max} - 1, \ldots, 1$,
(OpAMR.4) for each level $\ell = \ell^{\max}, \ell^{\max} - 1, \ldots, 1$, average the data on Ω^ℓ to those on $\Omega^{\ell-1} \setminus \Omega^{\ell-1}_{\text{valid}}$.

Thanks to the above algorithm, the discrete spatial operators defined on single-

[1] For non-periodic domains, ghost cells also aid the evaluation of discrete operators. In that scenario, ghost-cell values are usually determined by fitting local polynomials using values of adjacent non-ghost cells and the boundary conditions.
[2] p, the degree of the fitting polynomial, is four.

level grids are now applicable to multi-level AMR hierarchies. The symbols representing single-level operators also carry over naturally to a multi-level AMR hierarchy, since the former may be regarded as a special case of the latter.

3 Temporal integration

By Proposition 1, the following system of ordinary differential equations (ODEs),

$$\frac{\mathrm{d}\langle \mathbf{u} \rangle}{\mathrm{d}t} = -\mathbf{D}\langle \mathbf{u}\mathbf{u} \rangle + \langle \mathbf{g} \rangle - \mathbf{G}\langle p \rangle + \nu \mathbf{L}\langle \mathbf{u} \rangle ; \tag{29a}$$

$$\mathbf{D}\langle \mathbf{u} \rangle = 0, \tag{29b}$$

is a fourth-order approximation of the spatially integrated INSE over the control volumes. It is desirable to treat the non-stiff convection term explicitly and the stiff diffusion term implicitly so that the time-step size is only constrained by the Courant number; this is achieved by employing an ARK method.

3.1 Additive Runge-Kutta methods

ARK methods [10,3] decompose the right-hand side of an ODE into multiple terms and assemble the solution in an additive fashion from the "sub-solutions" to the individual terms; hence the name "additive" follows. One particular type of ARK methods, the implicit-explicit (IMEX) scheme [4], treats the non-stiff terms by an explicit Runge-Kutta (ERK) method and the stiff terms by an implicit Runge-Kutta method. This IMEX partition is natural because treating the stiff terms explicitly incurs very small time steps and treating the non-stiff terms implicitly leads to a linear system challenging for iterative solvers. It is also amenable to software reusability since the explicit module and the implicit module are loosely coupled.

For the implicit part of IMEX, a fully implicit method is daunting since all stages are coupled together as a big implicit system. Semi-implicit, or diagonally implicit RK (DIRK) methods have the Butcher matrix $A^{[I]}$ as lower-triangular so that the unknowns at each stage are only coupled to those in previous stages. If all the diagonal elements of $A^{[I]}$ are further made equal (i.e. $a^{[I]}_{j,j} = \gamma$), we arrive at the singly diagonally implicit Runge-Kutta (SDIRK) methods [1,7], which have been popular for their efficiency, easy implementation, and excellent stability properties. As minor drawbacks, they tend to be restricted to relatively low order and suffer from order reduction for very stiff ODEs. One simple way to alleviate these problems is to use the ESDIRK

methods [8,20], which differ from the SDIRK methods in having an explicit first stage by setting $a^{[I]}_{1,j} = 0$. For an ODE of the form

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \mathbf{X}^{[\mathrm{E}]}(\phi, t) + \mathbf{X}^{[\mathrm{I}]}(\phi), \tag{30}$$

the steps of an ERK-ESDIRK IMEX scheme are as follows:

$$\phi^{(1)} = \phi^n \approx \phi(t^n), \tag{31a}$$

for $s = 2, 3, \ldots, n_{\mathbf{s}}$,

$$\left(I - \Delta t \gamma \mathbf{X}^{[\mathrm{I}]}\right)\phi^{(s)} = \phi^n + \Delta t \sum_{j=1}^{s-1} a^{[E]}_{s,j} \mathbf{X}^{[\mathrm{E}]}\left(\phi^{(j)}, t^{(j)}\right) + \Delta t \sum_{j=1}^{s-1} a^{[I]}_{s,j} \mathbf{X}^{[\mathrm{I}]}\phi^{(j)}, \tag{31b}$$

$$\phi^{n+1} = \phi^n + \Delta t \sum_{s=1}^{n_{\mathbf{s}}} b^{[E]}_s \mathbf{X}^{[\mathrm{E}]}\left(\phi^{(s)}, t^{(s)}\right) + \Delta t \sum_{s=1}^{n_{\mathbf{s}}} b^{[I]}_s \mathbf{X}^{[\mathrm{I}]}\phi^{(s)}, \tag{31c}$$

where the superscript $^{(s)}$ denotes an intermediate stage, $t^{(s)} = t^n + c_s \Delta t$ the time of that stage, $n_{\mathbf{s}}$ the number of stages, and $A, \mathbf{b}, \mathbf{c}$ the standard coefficients of the *Butcher tableau* [22,7]. For each intermediate stage, $\phi^{(s)}$ is determined by solving the linear system (31b) with its RHS calculated from values of previous stages. Finally in (31c), ϕ^{n+1} is computed from all the intermediate values. Both the matrices $A^{[\mathrm{E}]}$ and $A^{[\mathrm{I}]}$ are lower-triangular, and the diagonal entries of $A^{[\mathrm{E}]}$ are all zero's, thus algorithm (31) is indeed semi-implicit.

3.2 The time-stepping algorithm

The pressure gradient in (29a) can be removed by applying $\mathbf{P}_E$ in (15) to both sides of the equation because $\mathbf{G}\langle p \rangle$ is in the null space of $\mathbf{P}_E$. Further applying (29b) yields a single evolution equation on the velocity,

$$\frac{\mathrm{d}\langle \mathbf{u} \rangle}{\mathrm{d}t} = \mathbf{P}_E \Big(-\mathbf{D}\langle \mathbf{u}\mathbf{u} \rangle + \langle \mathbf{g} \rangle + \nu \mathbf{L}\langle \mathbf{u} \rangle \Big). \tag{32}$$

As discussed in Section 2.2, $\mathbf{P}_E$ has a very wide stencil that invites numerical oscillations. In contrast, the approximate projection operator $\mathbf{P}$ defined in (16) has a compact stencil, and is stable and fourth-order accurate. Consequently, the following ODE is a fourth-order approximation to (29) on periodic domains:

$$\frac{\mathrm{d}\langle \mathbf{u} \rangle}{\mathrm{d}t} = \mathbf{P}\left(\mathbf{X}^{[\mathrm{E}]}\langle \mathbf{u} \rangle + \mathbf{X}^{[\mathrm{I}]}\langle \mathbf{u} \rangle \right), \tag{33}$$

where the averaged Eulerian acceleration has been split into two parts,

$$\mathbf{X}^{[\mathrm{E}]}\langle\mathbf{u}\rangle = \langle\mathbf{g}\rangle - \mathbf{D}\langle\mathbf{u}\mathbf{u}\rangle\,, \qquad \mathbf{X}^{[\mathrm{I}]}\langle\mathbf{u}\rangle = \nu\mathbf{L}\langle\mathbf{u}\rangle\,. \tag{34}$$

A direct application of the ERK-ESDIRK IMEX algorithm (31) to the ODE system (33) leads to the following algorithmic steps for solving the INSE on periodic domains.

$$\langle\mathbf{u}\rangle^{(1)} = \langle\mathbf{u}\rangle^{n} \approx \langle\mathbf{u}(t^n)\rangle\,, \tag{35a}$$

for $s = 2, 3, \ldots, n_{\mathsf{s}}$,

$$(\mathbf{I} - \Delta t\nu\gamma\mathbf{L})\langle\mathbf{u}\rangle^{(s)} = \langle\mathbf{u}\rangle^{n} + \Delta t\sum_{j=1}^{s-1} a_{s,j}^{[\mathrm{E}]}\mathbf{P}\mathbf{X}^{[\mathrm{E}]}\langle\mathbf{u}\rangle^{(j)} + \Delta t\nu\sum_{j=1}^{s-1} a_{s,j}^{[\mathrm{I}]}\mathbf{L}\langle\mathbf{u}\rangle^{(j)}\,, \tag{35b}$$

$$\langle\mathbf{u}^*\rangle^{n+1} = \langle\mathbf{u}\rangle^{n} + \Delta t\sum_{s=1}^{n_{\mathsf{s}}} b_{s}^{[\mathrm{E}]}\mathbf{X}^{[\mathrm{E}]}\langle\mathbf{u}\rangle^{(s)} + \Delta t\nu\sum_{s=1}^{n_{\mathsf{s}}} b_{s}^{[\mathrm{I}]}\mathbf{L}\langle\mathbf{u}\rangle^{(s)}\,, \tag{35c}$$

$$\langle\mathbf{u}\rangle^{n+1} = \mathbf{P}\langle\mathbf{u}^*\rangle^{n+1} \tag{35d}$$

where the Butcher coefficients are identical to those of "ARK4(3)6L[2]SA", a fourth-order method in [18]; they are also enclosed in Appendix C of [30]. Note that $\mathbf{P}\mathbf{L}\langle\mathbf{u}\rangle = \mathbf{L}\langle\mathbf{u}\rangle + O(h^4)$ because of (24) and the commutativity of $\mathbf{P}$ and $\mathbf{L}$, hence $\mathbf{P}\mathbf{L}\langle\mathbf{u}\rangle$ can be replaced by $\mathbf{L}\langle\mathbf{u}\rangle$ without affecting the fourth-order accuracy.

The design of algorithm (35) ensures that the velocity evolves in the vector space that is *approximately* solenoidal, i.e. $\mathbf{D}\langle\mathbf{u}\rangle = O(h^4)$. To show this, it suffices to point out that the velocity at each intermediate stage satisfies $\mathbf{D}\langle\mathbf{u}\rangle^{(s)} = O(h^4)$ due to the following,

(a) the initial velocity $\langle\mathbf{u}\rangle^{n}$ satisfies $\mathbf{D}\langle\mathbf{u}\rangle^{n} = O(h^4)$,
(b) $\mathbf{D}\mathbf{P}\mathbf{X}^{[\mathrm{E}]}\langle\mathbf{u}\rangle = O(h^4)$,
(c) $\mathbf{L}$ is linear and commutes with $\mathbf{P}$,
(d) when evaluating $\mathbf{L}$ on an AMR hierarchy, the value of $\langle\mathbf{u}\rangle$ at the ghost cells near the coarse-fine interface are interpolated to fifth-order accuracy, hence $\mathbf{D}\mathbf{L}\langle\mathbf{u}\rangle = O(h^4)$.

To emphasize that $\mathbf{P}$ is an approximate projection, the phrase "divergence-free" is not used in the above arguments to avoid potential confusion of $\mathbf{D}\langle\mathbf{u}\rangle = O(h^4)$ with $\mathbf{D}\langle\mathbf{u}\rangle = 0$. In contrast, refluxing and AMRCFI enforce mass conservation to machine precision:

- the refluxing procedure [30] is applied to $\mathbf{L}$ so that the fluxes across any closed coarse-fine boundary always sum up to zero *exactly*,
- the coarse-fine interpolation algorithm AMRCFI enforces *exact* mass conservation [29] when filling the fine ghost cells.

The proposed algorithm (35) is pressure-free in that no information of pressure is needed to evolve velocity. If needed, pressure at any time instant can be easily extracted from the velocity field by solving the discrete pressure Poisson equation,

$$\mathbf{L}\langle p\rangle = \mathbf{D}\Big(\langle \mathbf{g}\rangle - \mathbf{D}\langle \mathbf{uu}\rangle + \nu \mathbf{L}\langle \mathbf{u}\rangle\Big). \tag{36}$$

A standard multigrid method identical to that in [30] is used to solve the linear systems in (35) and (36); see [30, p. B185] for a brief discussion and [30, Fig. 5.2] for a demonstration of its efficiency.

3.3 The range of stable Courant numbers

To derive the range of stable Courant numbers, the convection operator is linearized in the ODE system (33). Theorem 5 states that the spectral radius of $\mathbf{P}$ is one, hence the stability region of the linearized ODE is the same as that of the corresponding advection-diffusion equation as in [30],

$$\mathrm{Cr} \leq \frac{2.91}{\mathrm{D}}. \tag{37}$$

Note that the above condition is the stability condition for pure advection and has been verified by numerical experiments. In the presence of diffusion, the maximum stable Courant number becomes larger and larger as diffusion gets stronger and stronger. As an additional advantage of using a one-step integration method, (37) might be removed as superfluous in the scenario that the time-step size is changed adaptively for error-control, which is not unusual for Runge-Kutta methods.

4 Numerical tests

In this section, a Taylor vortex test problem is first simulated for a wide range of Reynolds numbers to show the fourth-order convergence of the proposed method. The results are then compared to those of a second-order method to demonstrate the superior accuracy and efficiency of the proposed fourth-order method. Finally, another decaying sinusoidal test is performed on locally-refined grids both in 2D and 3D to illustrate the flexibility of local refinement.

4.1 Single level grids: Taylor vortex

This test is taken from [14], with the exact solutions of the INSE as

Table 1
Errors and convergence rates for the Taylor vortex test with Re = 30. $\nu = 0.1$, $t_0 = 0.0$, $t_e = 0.5$, Cr = 0.75.

results of the second-order method by Griffith [14]

h	1/128	Rate	1/256	Rate	1/512	Rate	1/1024
u, v L_∞	3.89e-05	2.00	9.74e-06	2.00	2.44e-06	2.00	**6.10e-07**
u, v L_1	3.44e-05	2.00	8.63e-06	2.00	2.16e-06	2.00	**5.41e-07**
p L_∞	1.66e-06	2.01	4.13e-07	2.00	1.03e-07	2.00	**2.57e-08**
p L_1	6.72e-07	2.01	1.67e-07	2.00	4.17e-08	2.00	**1.04e-08**

results of the proposed fourth-order method: single-level grids

h	1/32	Rate	1/64	Rate	1/128	Rate	1/256
u, v L_∞	6.47e-06	3.89	**4.36e-07**	3.95	2.82e-08	3.98	1.79e-09
u, v L_1	3.64e-06	3.93	**2.39e-07**	3.97	1.53e-08	3.98	9.64e-10
p L_∞	4.80e-07	4.00	**3.00e-08**	4.01	1.85e-09	4.01	1.15e-10
p L_1	1.96e-07	4.01	**1.21e-08**	4.01	7.52e-10	4.01	4.67e-11

Table 2
Errors and convergence rates for the Taylor vortex test with Re = 300. $\nu = 0.01$, $t_0 = 0.0$, $t_e = 0.5$, Cr = 0.75.

results of the second-order method by Griffith [14]

h	1/128	Rate	1/256	Rate	1/512	Rate	1/1024
u, v L_∞	2.80e-04	1.99	7.04e-05	2.00	1.77e-05	2.00	**4.42e-06**
p L_∞	4.18e-04	1.98	1.06e-04	1.99	2.66e-05	2.00	**6.66e-06**

results of the proposed fourth-order method: single-level grids

h	1/32	Rate	1/64	Rate	1/128	Rate	1/256
u, v L_∞	3.88e-04	4.50	1.71e-05	4.22	**9.17e-07**	4.10	5.35e-08
p L_∞	1.18e-03	4.39	5.64e-05	4.23	**3.01e-06**	4.11	1.74e-07

$$\begin{cases} \mathbf{u}(x,y,t) = 1 + 2\exp(-8\pi^2\nu t)\begin{pmatrix} -\cos(2\pi(x-t))\sin(2\pi(y-t)) \\ \sin(2\pi(x-t))\cos(2\pi(y-t)) \end{pmatrix}, \\ p(x,y,t) = -\exp(-16\pi^2\nu t)\Big(\cos(4\pi(x-t)) + \cos(4\pi(y-t))\Big). \end{cases} \tag{38}$$

On a unit domain $[0,1]^2$, (38) is advanced from $t_0 = 0$ to $t_e = 0.5$ on four successively refined single-level grids with Cr = 0.75. The initial cell-averaged velocity is calculated by (38) and a sixth-order Newton-Cotes formula. The same test is repeated for Reynolds numbers Re$=\frac{U_0 L_0}{\nu} = 30, 300, 3000, 30000$ with $U_0 = 3$ and $L_0 = 1$. The errors in the horizontal velocity are shown in

Table 3
Errors and convergence rates for the Taylor vortex test with Re = 3000. $\nu = 0.001$, $t_0 = 0.0$, $t_e = 0.5$, Cr = 0.75.

results of the second-order method by Griffith [14]

h	1/128	Rate	1/256	Rate	1/512	Rate	1/1024
u, v L_∞	3.11e-04	2.00	7.77e-05	2.00	1.94e-05	2.00	**4.86e-06**
p L_∞	6.31e-04	1.98	1.60e-04	1.99	4.01e-05	2.00	**1.01e-05**

results of the proposed fourth-order method: single-level grids

h	1/32	Rate	1/64	Rate	1/128	Rate	1/256
u, v L_∞	1.33e-03	5.03	4.07e-05	4.50	**1.79e-06**	4.31	9.05e-08
p L_∞	3.79e-03	4.56	1.61e-04	4.42	**7.54e-06**	4.24	3.98e-07

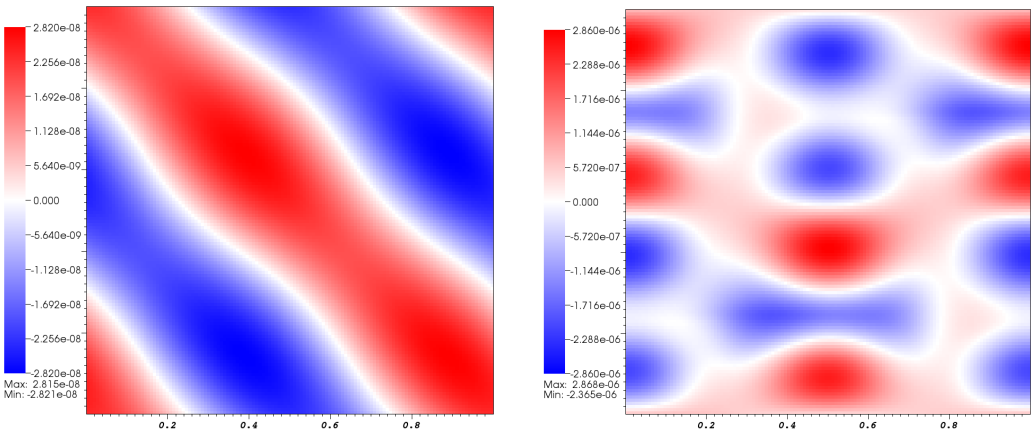
Table 4
Errors and convergence rates for the Taylor vortex test with Re = 30000. $\nu = 0.0001$, $t_0 = 0.0$, $t_e = 0.5$, Cr = 0.75.

results of the second-order method by Griffith [14]

h	1/128	Rate	1/256	Rate	1/512	Rate	1/1024
u, v L_∞	3.15e-04	2.00	7.87e-05	2.00	1.97e-05	2.00	**4.92e-06**
p L_∞	6.62e-04	1.99	1.67e-04	1.99	4.20e-05	2.00	**1.05e-05**

results of the proposed fourth-order method: single-level grids

h	1/32	Rate	1/64	Rate	1/128	Rate	1/256
u, v L_∞	1.50e-03	5.06	4.51e-05	4.53	**1.95e-06**	4.34	9.65e-08
p L_∞	4.31e-03	4.57	1.81e-04	4.44	**8.36e-06**	4.26	4.35e-07



(a) Re= 30 (b) Re= 30000

Fig. 1. The errors of the horizontal velocity in the Taylor vortex test. $h = \frac{1}{128}$.

Table 5
Errors and convergence rates of the proposed fourth-order method for the Taylor vortex test with Re = 30000. $t_0 = 0.0$, $t_e = 0.5$, Cr = 1.5.

h	1/32	Rate	1/64	Rate	1/128	Rate	1/256
u, v L_∞	3.24e-03	5.06	9.70e-05	5.08	**2.87e-06**	4.53	1.24e-07
p L_∞	7.33e-03	4.71	2.79e-04	4.62	**1.14e-05**	4.44	5.24e-07
Du L_∞	2.36e-04	7.46	1.34e-06	7.51	7.37e-09	9.22	1.24e-11

Figure 1 for the lowest and highest Reynolds numbers. Quantitative results are shown in Tables 1, 2, 3, and 4, together with those from the second-order method by Griffith [14]. The fourth-order convergence of the proposed method is confirmed for both velocity and pressure in max-norm and in 1-norm. Another test with Re= 30000 and Cr = 1.5 is also performed to verify (37), the super-convergence shown in Table 5 might indicate that the time-step size is not in the asymptotic range yet, especially in the case of velocity divergence.

4.2 Superior efficiency of the proposed fourth-order method

Let ϵ denote the desired accuracy: when the exact solution is available, ϵ is the relative difference between the exact and computed solutions; otherwise ϵ is the relative difference between the computed solutions on two successively refined grids. Let h_2 and h_4 denote the grid sizes necessary to obtain the same accuracy ϵ for a second-order method and a fourth-order method, respectively, then

$$h_2 = O(\epsilon^{\frac{1}{2}}),\ h_4 = O(\epsilon^{\frac{1}{4}}) \ \Rightarrow\ h_4/h_2 = c\,\epsilon^{-1/4}. \tag{39}$$

Clearly the scaling factor c does not depend on ϵ for a given test problem.

Suppose a given problem costs one unit memory and one unit CPU time for the proposed fourth-order method, and it costs r_M unit memory and r_C unit CPU time for a second-order method, then r_M and r_C can be considered as the saving factors of memory and CPU time of the proposed method over a second-order method. (39) yields

$$\begin{aligned} r_M &= \tfrac{1}{3}\left(\tfrac{h_4}{h_2}\right)^{\mathrm{D}} = \tfrac{1}{3}c^{\mathrm{D}}\epsilon^{-\frac{\mathrm{D}}{4}}, \\ r_C &= \tfrac{1}{5}\left(\tfrac{h_4}{h_2}\right)^{\mathrm{D}+1} = \tfrac{1}{5}c^{\mathrm{D}+1}\epsilon^{-\frac{\mathrm{D}+1}{4}}, \end{aligned} \tag{40}$$

where the factor $\frac{1}{3}$ in the formula for r_M comes from the fact that the proposed fourth-order method has to store intermediate variables for all six stages while a second-order method only has two stages. The factor $\frac{1}{5}$ in the formula for r_C

Table 6
The saving factors of the proposed fourth-order method over the second-order method by Griffith [14] for the Taylor vortex test. From (39) and Tables 1, 2, 3 & 4, the values of c for Re=30, 300, 3000, 30000 are calculated as 0.50, 0.52, 0.44, 0.43, respectively. Low-Reynolds-number flows have fewer fine structures than high-Reynolds-number flows; see Figure 1. Hence $c = 0.50$ is used for $\epsilon = 10^{-2}, 10^{-4}$ while $c = 0.43$ for $\epsilon = 10^{-6}, 10^{-8}, 10^{-10}$.

ϵ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
2D r_M	1.16	11.6	61.6	616	6.16×10^3
3D r_M	2.15	137	838	2.65×10^4	8.38×10^5
2D r_C	0.79	25.0	503	1.59×10^4	5.03×10^5
3D r_C	1.25	125	6.84×10^3	6.84×10^5	6.84×10^7

comes from the fact that ten [3] /two implicit solves per time step are necessary for the fourth-order/second-order methods. It is also assumed that for both methods efficient multigrid algorithms are employed to solve the linear systems so that the CPU time is proportional to the total number of control volumes [4] . Note that the power of $\frac{h_4}{h_2}$ in the expression of r_M is D instead of D+1 because a one-step method only needs to store the data of a single time step to advance the solution.

Table 6 shows the values of r_M and r_C for the Taylor vortex test. For relatively low accuracies such as $\epsilon = 0.01$, the fourth-order method does not save much from the second-order method. However, as ϵ gets smaller and smaller, the saving factors of the fourth-order method grow exponentially fast. In fact, even for moderate accuracies $\epsilon = 10^{-4}, 10^{-6}$, the benefit of the fourth-order method is still very appealing. Certainly the numbers in Table 6 come from a single test, and the values of r_M and r_C will vary for different problems. Nonetheless, the pattern of the power growth of r_M and r_C should stay the same; this is evident from (40).

Sometimes it is impossible to have numerical solutions with accuracy higher than first order, one example being the shock waves and the famous Godunov theorem [13]. For such problems, first-order and second-order methods might be more efficient than a higher-order method. Nonetheless, in solving challenging problems for which high-order numerical solutions are indeed possible,

[3] At each intermediate stage $s = 2, 3, 4, 5, 6$ of the proposed fourth-order method, one has to solve a linear system associated with the approximate projection operator and another associated with (35b). The implicit solve in (35d) can be avoided by [30, Eq. (3.3)] and the fact that each intermediate velocity is approximately solenoidal.
[4] Second-order methods typically have $n_{\text{ghost}} = 1$ while the proposed fourth-order method has $n_{\text{ghost}} = 2$. However, the negative impact of the extra ghost cells on the performance of parallel computing can be reduced to a minimum by communication aggregation [27].

high-order methods should be preferred to fully exploit the power of supercomputers. After all, as shown in (40), the savings on memory and time by high-order methods are power functions of the relative accuracy whereas the augmentation of computational resources from parallel computing is at best linear.

4.3 Locally-refined grids: decaying sinusoidal

This test uses the following exact solution of the INSE:

$$\mathbf{u} = \exp(-2\pi^2\nu t)\begin{pmatrix} -\cos(\pi x)\sin(\pi y) \\ \sin(\pi x)\cos(\pi y) \end{pmatrix}, \tag{41a}$$

$$p = -\frac{1}{4}\exp(-4\pi^2\nu t)\Big(\cos(2\pi x) + \cos(2\pi y)\Big), \tag{41b}$$

with its 3D counterpart as

$$\begin{cases} \mathbf{u}(x,y,z,t) = \exp(-3\pi^2\nu t)\begin{pmatrix} -2\cos(\pi x)\sin(\pi y)\sin(\pi z) \\ \sin(\pi x)\cos(\pi y)\sin(\pi z) \\ \sin(\pi x)\sin(\pi y)\cos(\pi z) \end{pmatrix}, \\ p(x,y,z,t) = -\frac{1}{4}\exp(-6\pi^2\nu t)\Big(\cos(2\pi x) + \cos(2\pi y) + \cos(2\pi z)\Big). \end{cases} \tag{42}$$

In the 2D case, the time derivative of velocity cancels the diffusion term and pressure gradient cancels the convection term, resulting in a zero forcing term. In comparison, the forcing term in the 3D case is no longer zero, but can be readily obtained from (1a).

(41) and (42) are advanced from $t_0 = 0$ to $t_e = 1$ with Cr = 1.0 on three successively refined hierarchies shown in Figure 2. Note that the layout of boxes on level 1 is chosen on purpose to test the two different scenarios of filling ghost cells.

Table 7 and Table 8 clearly show fourth-order convergence both in 2D and in 3D for all variables in terms of max-norm, 1-norm, and 2-norm. The computed solution of 2D pressure at the final time is also shown in Figure 3. The mosaic patterns are greatly reduced from the coarsest level to the intermediate level, and become completely indiscernible on the finest level. Since each colored square represents a control volume in its actual sizes, this plot serves as a visual illustration of the flexibility of local refinement in directing computational resources to regions of primary interests. The 3D solutions of velocity and pressure are shown in Figure 4, once again illustrating the flexibility of local mesh refinement.

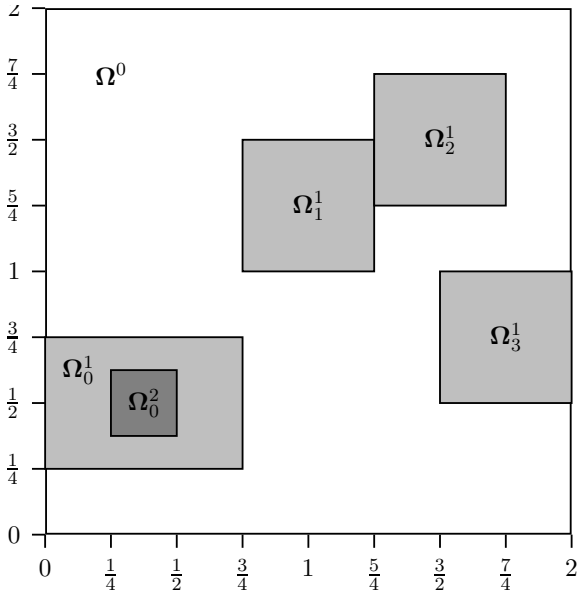


Fig. 2. The locally-refined grids with three levels for the decaying sinusoidal test. The white, light-gray, and gray boxes represent for the coarsest, intermediate, and the finest levels, respectively. Their ranges of the third dimension are $[0, 2]$, $[1/4, 3/4]$, and $[3/8, 5/8]$, respectively.

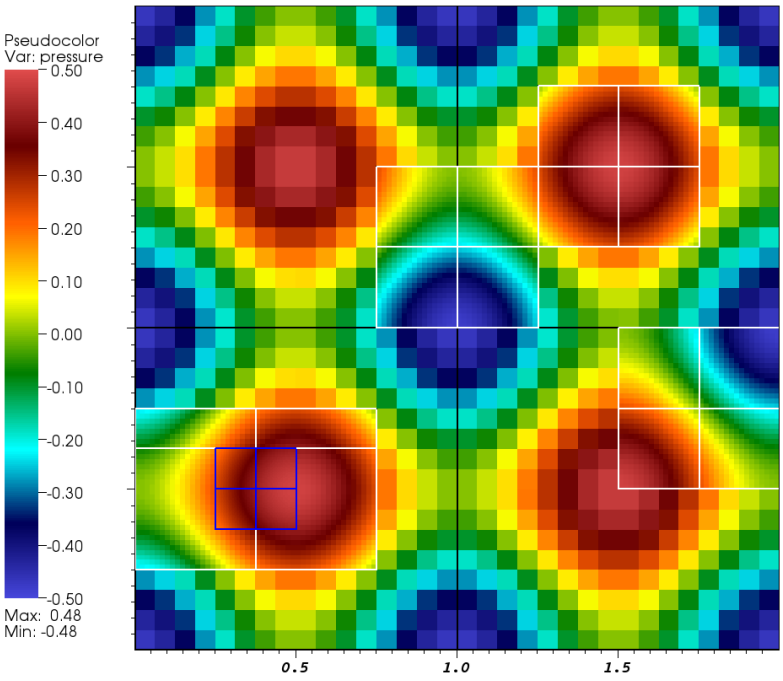


Fig. 3. Pressure at $t_e = 1.0$ for the 2D decaying sinusoidal test ($\nu = 0.001$) on a three-level hierarchy with base grid size $h = 1/32$ and refinement ratio $r = 4$. The black, white, and blue boxes represent the patches of the coarsest, the intermediate, and the finest levels, respectively. Each square shaded by a single solid color represents the extent of one control volume. No smoothing is applied to the plot.

Table 7
Errors and convergence rates of the 2D decaying sinusoidal test with Re = 1000. $\nu = 0.001$, $t_0 = 0.0$, $t_e = 1.0$, Cr = 1.0. The refinement ratio is four.

Base grid h	1/32	Rate	1/64	Rate	1/128
u L_∞	5.96e-04	4.06	3.58e-05	4.01	2.23e-06
u L_1	1.25e-04	3.99	7.87e-06	4.00	4.92e-07
u L_2	1.58e-04	4.00	9.90e-06	4.00	6.18e-07
v L_∞	5.85e-04	4.02	3.59e-05	4.01	2.24e-06
v L_1	1.37e-04	3.99	8.63e-06	4.00	5.39e-07
v L_2	1.91e-04	3.98	1.21e-05	4.00	7.55e-07
p L_∞	5.34e-04	4.06	3.21e-05	4.05	1.94e-06
p L_1	1.53e-04	4.06	9.18e-06	4.04	5.57e-07
p L_2	1.96e-04	4.07	1.17e-05	4.05	7.06e-07

Table 8
Errors and convergence rates of the 3D decaying sinusoidal test with Re = 1000. $\nu = 0.001$, $t_0 = 0.0$, $t_e = 1.0$, Cr = 1.0. The refinement ratio is two.

Base grid h	1/16	Rate	1/32	Rate	1/64
u L_∞	2.83e-02	4.92	9.37e-04	4.63	3.78e-05
u L_1	2.41e-03	4.66	9.54e-05	4.36	4.66e-06
u L_2	3.14e-03	4.61	1.29e-04	4.29	6.58e-06
v L_∞	2.09e-02	4.58	8.71e-04	4.11	5.04e-05
v L_1	1.69e-03	4.58	7.07e-05	4.33	3.52e-06
v L_2	2.34e-03	4.48	1.05e-04	4.22	5.64e-06
w L_∞	1.33e-02	4.78	4.85e-04	3.94	3.16e-05
w L_1	1.54e-03	4.68	6.01e-05	4.47	2.72e-06
w L_2	2.12e-03	4.60	8.71e-05	4.35	4.29e-06
p L_∞	1.13e-02	4.12	6.51e-04	3.98	4.13e-05
p L_1	3.09e-03	4.09	1.82e-04	3.96	1.17e-05
p L_2	3.76e-03	4.02	2.32e-04	3.96	1.49e-05

5 Concluding remarks

The author has proposed a fourth-order approximate projection method for numerically solving the incompressible Navier-Stokes equations on locally-refined periodic domains. The main contribution of this work is the analysis

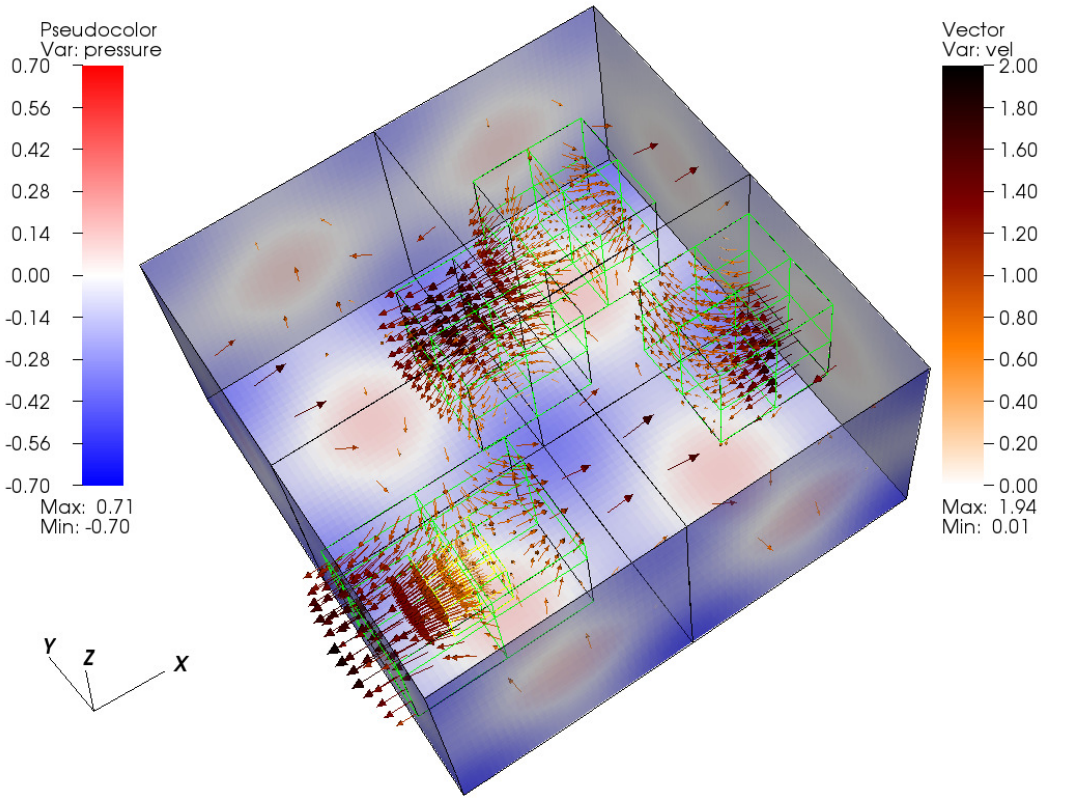


Fig. 4. Velocity and pressure at $t_e = 1.0$ for the 3D decaying sinusoidal test ($\nu = 0.001$) on a three-level hierarchy with base grid size $h = 1/64$ and refinement ratio $r = 2$. The black, green, and yellow boxes represent the patches of the coarsest, the intermediate, and the finest levels, respectively. Only the lower half of the domain ($\mathbf{x} \in [0, 2] \times [0, 2] \times [0, 1]$) is shown. The velocity field is represented by the vectors, whose density is proportional to the inverse of local grid size. The legend on the right indexes the magnitudes of the velocity vectors and the one on the left the value of pressure.

of the fourth-order approximate projection operator on its stability and accuracy. Another relatively minor contribution is the coupling of various modules to solving the INSE. Due to the simplicity of the proposed method, it can be generalized to even higher-order accuracies in a straightforward way.

The next step along this research is to numerically solve the INSE on no-slip domains; after all, many interesting phenomena of incompressible flows occur near no-slip boundaries. Since the Laplacian operator and diffusion operator do not commute on such domains, the analysis of the approximate projection operator on such domains is highly difficult. Another consequence of the non-commutativity is that the time-integration algorithm proposed in this work does not have a straightforward extension to no-slip domains. Lastly, the physical boundaries give rise to a new type of ghost cells, and it is not yet clear how to set the ghost values of the velocity for the intermediate stages without deteriorating the fourth-order accuracy. So far we have not been able to obtain

an approximate projection projector of which both the max eigenvalue and the 2-norm are less or equal to one for no-slip domains. Nonetheless, many progresses have been made and a fourth-order adaptive projection method for solving the INSE on no-slip domains will be reported in a future paper.

Acknowledgements

The author would like to thank an anonymous referee, whose comments lead to an improvement of the exposition.

References

[1] R. Alexander, Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s, SIAM J. Numer. Anal. 14 (6) (1977) 1006–1021.

[2] A. S. Almgren, J. B. Bell, W. Y. Crutchfield, Approximate projection methods: Part I. Inviscid analysis, SIAM J. Sci. Comput. 22 (4) (2000) 1139–1159.

[3] A. L. Araújo, A. Murua, J. M. Sanz-Serna, Symplectic methods based on decompositions, SIAM J. Numer. Anal. 34 (5) (1997) 1926–1947.

[4] U. M. Ascher, S. J. Ruuth, R. J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (1997) 151–167.

[5] J. B. Bell, P. Colella, H. M. Glaz, A second-order projection method for the incompressible Navier-Stokes equations, J. Comput. Phys. 85 (1989) 257–283.

[6] M. Benzi, G. H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137, doi:10.1017/S0962492904000212.

[7] J. C. Butcher, Numerical Methods for Ordinary Differential Equations, 2nd ed., John Wiley & Sons, 2008, ISBN: 978-0-470-72335-7.

[8] J. C. Butcher, D. J. L. Chen, A new type of singly-implicit Runge-Kutta method, Appl. Numer. Math. 34 (2000) 179–188.

[9] A. J. Chorin, Numerical solution of the Navier-Stokes equations, Math. Comput. 22 (104) (1968) 745–762.

[10] G. J. Cooper, A. Sayfy, Additive methods for the numerical solution of ordinary differential equations, Math. Comput. 35 (152) (1980) 1159–1172.

[11] A. W. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, 1997, ISBN:978-0-898713-89-3.

[12] W. E., J.-G. Liu, Gauge method for viscous incompressible flows, Comm. Math. Sci. 1 (2) (2003) 317.

[13] S. K. Godunov, A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations, Math. Sbornik 47 (1959) 271–306, translated by US Joint Publ. Res. Service, JPRS 7226, 1969.

[14] B. E. Griffith, An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as a preconditioner, J. Comput. Phys. 228 (20) (2009) 7565–7595, doi:10.1016/j.jcp.2009.07.001.

[15] W. D. Henshaw, A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids, J. Comput. Phys. 113 (1994) 13–25.

[16] S. Y. Kadioglu, R. Klein, M. L. Minion, A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics, J. Comput. Phys. 227 (3) (2008) 2012–2043, doi:10.1016/j.jcp.2007.10.008.

[17] N. A. Kampanis, J. A. Ekaterinaris, A staggered grid, high-order accurate method for the incompressible Navier-Stokes equations, J. Comput. Phys. 215 (2006) 589–613, doi:10.1016/j.jcp.2005.11.014.

[18] C. A. Kennedy, M. H. Carpenter, Additive Runge-Kutta schemes for convection-diffusion-reaction equations, Appl. Numer. Math. 44 (2003) 139–181.

[19] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, J. Comput. Phys. 59 (2) (1985) 308–323.

[20] A. Kværnø, Singly diagonally implicit Runge-Kutta methods with an explicit first stage, BIT 44 (2004) 489–502.

[21] M. F. Lai, A projection method for reacting flow in the zero Mach number limit, Ph.D. thesis, University of California at Berkeley (FEB 1994).

[22] J. D. Lambert, Numerical Methods for Ordinary Differential Systems, John Wiley & Sons, New York, 1991, ISBN:0-471-92990-5.

[23] D. F. Martin, P. Colella, D. Graves, A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions, J. Comput. Phys. 227 (2008) 1863–1886, doi:10/1016/j.jcp.2007.09.032.

[24] S. A. Orszag, M. Israeli, M. O. Deville, Boundary conditions for incompressible flows, J. Sci. Comput. 1 (1) (1986) 75–111.

[25] J. M. C. Pereira, M. H. Kobayashi, J. C. F. Pereira, A fourth-order-accurate finite volume compact method for the incompressible Navier-Stokes solutions, J. Comput. Phys. 167 (1) (2001) 217–243, doi:10.1006/jcph.2000.6673.

[26] R. K. ShukLa, M. Tatineni, X. Zhong, Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier-stokes equations., J. Comput. Phys. 224 (2007) 1064–1094, doi:10.1016/j.jcp.2006.11.007.

[27] S. Williams, D. D. Kalamkar, A. Singh, A. M. Deshpande, B. V. Straalen, M. Smelyanskiy, A. Almgren, P. Dubey, J. Shalf, L. Oliker, Optimization of geometric multigrid for emerging multi- and manycore processors, in: Proceedings of SC12 - The International Conference for High Performance Computing, Networking, Storage and Analysis, Salt lake city, UT, 2012, http://conferences.computer.org/sc/2012/papers/1000a095.pdf.

[28] Q. Zhang, AMRCFI: A matlab package for coarse-fine interpolation in adaptive mesh refinement, http://sourceforge.net/projects/amrcfi/ (2011).

[29] Q. Zhang, High-order, multidimensional, and conservative coarse-fine interpolation for adaptive mesh refinement, Comput. Methods Appl. Mech. Engrg. 200 (45-46) (2011) 3159–3168, doi:10.1016/j.cma.2011.07.009.

[30] Q. Zhang, H. Johansen, P. Colella, A fourth-order accurate finite-volume method with structured adaptive mesh refinement for solving the advection-diffusion equation, SIAM J. Sci. Comput. 34 (2) (2012) B179–B201, doi:10.1137/110820105.