

Fourth-Order Accurate Fractional-Step IMEX Schemes for the Incompressible Navier-Stokes Equations on Moving Overlapping Grids

F. Meng^{a,1}, J. W. Banks^{b,1,3}, W. D. Henshaw^{b,1,2}, D. W. Schwendeman^{b,1,2,*}

^aDepartment of Mechanical, Aerospace and Nuclear Engin., Rensselaer Polytechnic Institute, Troy, NY 12180, USA

^bDepartment of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Abstract

Two efficient fractional-step schemes for the incompressible Navier-Stokes equations in two and three dimensions are described. The schemes are fourth-order accurate in space and time, and are based on solving the velocity-pressure form of the equations. Both schemes employ predictor-corrector time-stepping approaches. The first is an explicit Adams-type scheme, while the second is an IMEX-BDF-type scheme in which the viscous/advective terms in the equations are treated implicitly/explicitly. The equations and boundary conditions are discretized in space using fourth-order accurate finite-difference approximations. The formulation of discrete boundary conditions for each stage of the fractional-step scheme is found to be critical to the accuracy and stability of the approach. A WENO-based scheme, called BWENO, provides upwind dissipation and ensures robustness of the schemes for problems where the solution is under-resolved on the grid (e.g. near boundary or shear layers). Complex, and possibly moving, domains are handled efficiently using composite overlapping grids. A variety of problems in two and three dimensions, some for which exact solutions are either known or manufactured, are used to verify the stability and accuracy of the new schemes.

Keywords: Incompressible Navier-Stokes; Velocity-pressure formulation; Fractional-step scheme; IMEX-BDF multistep method; WENO dissipation; Fourth-order accuracy.

1. Introduction

In this work we develop efficient and high-order accurate solvers for incompressible flow in complex, and possibly moving, domains. The schemes, which are based on the velocity-pressure form of the Navier-Stokes equations, are fourth-order accurate in space and time. Two fourth-order accurate predictor-corrector (PC) time-stepping schemes are developed. The first method is an explicit Adams-type scheme, while the second is a BDF-based semi-implicit (IMEX) scheme, where the viscous terms are treated implicitly. The schemes use a fractional-step approach whereby the update for the velocity is decoupled from the solution for the pressure. This is important for performance since it allows the use of efficient solvers for the decoupled pressure-Poisson equation as well as for the decoupled Poisson-like equations (implicit heat-operators) for implicit time-stepping of the velocity. The approximations of the spatial derivatives in the equations use fourth-order accurate finite differences, together with a new WENO-based upwind approximation for robust under-resolved simulations. Complex moving geometry is handled through the use of composite overlapping grids [1].

An important new feature of the work is the design of a fractional-step scheme that is fourth-order accurate in space and time for *both* the fluid velocity and pressure. Many fractional-step schemes used in the

*Department of Mathematical Sciences, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA.

Email addresses: mengf5@rpi.edu (F. Meng), banksj3@rpi.edu (J. W. Banks), henshw@rpi.edu

(W. D. Henshaw), schwend@rpi.edu (D. W. Schwendeman)

¹This work was performed under DOE contracts from the ASCR Applied Math Program.

²Research supported by the National Science Foundation under grants DMS-1519934 and DMS-1818926.

³Research supported by a U.S. Presidential Early Career Award for Scientists and Engineers.

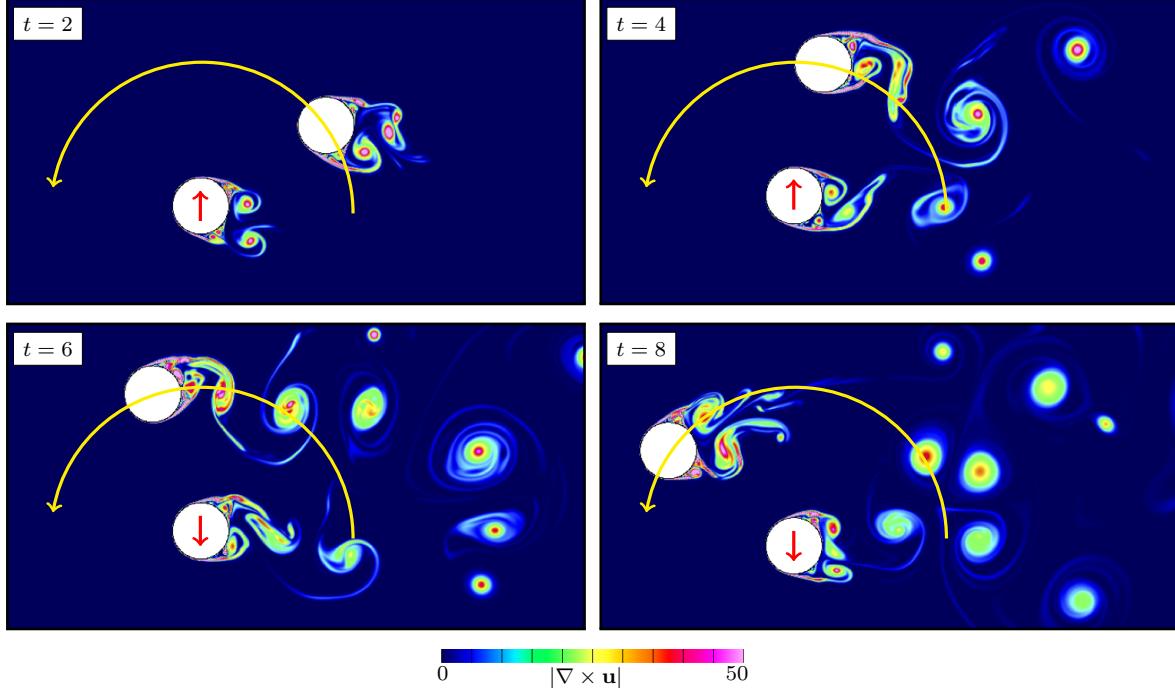


Figure 1: Contour plots of the enstrophy $|\nabla \times \mathbf{u}|$ for flow past two disks in a channel computed with the new fourth-order accurate semi-implicit predictor-corrector fractional-step scheme and BWENO upwind dissipation. The lower disk oscillates up and down while the upper disk rotates about the center of the lower disk, as indicated by the red arrows and yellow curves, respectively.

literature, such as the traditional projection scheme [2], do not achieve second-order accuracy in time due to splitting errors in either the interior equations or in the boundary conditions. Here, we extend the spatially fourth-order accurate scheme developed by Henshaw et al. [3–5] to fourth-order accuracy in time. This is achieved through the use of novel compatibility conditions to construct fourth-order accurate numerical boundary conditions. Some of these numerical boundary conditions, however, couple the velocity and the pressure, and are thus not immediately applicable to a fractional-step scheme that requires the boundary conditions to be split between the velocity solve and the pressure solve; in this paper we show how to split the boundary conditions while retaining fourth-order accuracy and stability.

To stabilize under-resolved simulations, upwind-schemes or artificial dissipation are often used. However, an upwind approximation typically uses a wider spatial-stencil than the corresponding centered approximation, while artificial dissipation often relies on tunable coefficients. To overcome these disadvantages, we adopt the BWENO scheme, originally designed for Vlasov simulations [6], and apply it for the case of incompressible flow. Similar to WENO [7–9], the BWENO scheme blends one-sided approximations with weights based on the smoothness of the solution. Unlike the WENO scheme, however, BWENO uses the same five-point stencil as the centered scheme. The BWENO approximation approaches the fourth-order accurate centered scheme for smooth flows, while reducing to a third-order upwind scheme in regions of the domain where the flow is not well-resolved on the grid. Further modifications to the original BWENO scheme for incompressible flow are developed to improve the treatment of the upwind dissipation near stagnation points. Figure 1 shows a sample computation of flow past two moving cylinders in a channel using the new fourth-order accurate scheme developed in this work.

There is a vast literature on numerical methods for incompressible flows and we do not attempt to review this past work here. Instead we note two recent developments on high-order accurate fractional-step schemes that are most relevant to the current work. In the first, Liu et al. [10, 11] have developed various fractional-step schemes including a third-order accurate time-discrete projection method that extends a slip-correction idea behind the well-known finite-difference scheme of Kim and Moin [12]. In the second, Colomés and Badia [13] have developed IMEX-SRK (Segregated Runge–Kutta) schemes, with orders of accuracy up to

three, that decouple the velocity and pressure solves. After discretizing with a mixed FEM approach, they algebraically form a discrete pressure equation and then eliminate the pressure from the discrete momentum equation.

The current work is motivated, in part, by our recent development of algorithms for fluid-structure interactions (FSI) involving incompressible flows and rigid bodies [14–17], elastic beams [18, 19] and elastic bulk solids [20–22]. In these FSI simulations, accurate velocities and tractions are needed at fluid-solid interfaces and thus we need schemes that can provide accuracy up to the boundary (some FEM approaches loose accuracy at boundaries). In addition, efficient schemes for complex moving geometry are required. For this latter purpose we use composite overlapping grids (also known as overset, or Chimera, grids) which are well known to be very effective for moving grid problems by providing high-quality grids even under large solid motions, see for example [23–26].

The remainder of the manuscript is organized as follows. Section 2 discusses the governing equations for the velocity-pressure formulation of the incompressible Navier-Stokes equations including the important specification of boundary conditions. Section 3 then introduces the explicit and semi-implicit predictor-corrector time-stepping methods and their stability properties. The spatial discretization is described in Section 4, including a discussion of overlapping grids and the BWENO upwind approximation, followed by a detailed discussion of the fractional-step algorithms. Numerical results are presented in Section 5 for both two and three-dimensional problems. Concluding remarks are given in Section 6.

2. Governing equations

Let $\Omega \subset \mathbb{R}^d$ denote a bounded domain in two ($d = 2$) or three ($d = 3$) dimensions, and let $\mathbf{x} = (x_1, \dots, x_d)$ and t denote position and time, respectively. The *velocity-pressure* form of the INS equations is given by

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{F}(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (1a)$$

$$\Delta p = -\rho \nabla \mathbf{u} : (\nabla \mathbf{u})^T + \nabla \cdot \mathbf{F}(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (1b)$$

where

$$\nabla \mathbf{u} : (\nabla \mathbf{u})^T \stackrel{\text{def}}{=} \sum_{i=1}^d \sum_{j=1}^d \frac{\partial u_i}{\partial x_j} \frac{\partial u_j}{\partial x_i}.$$

Here, $\mathbf{u}(\mathbf{x}, t) = (u_1(\mathbf{x}, t), \dots, u_d(\mathbf{x}, t))$ and $p(\mathbf{x}, t)$ are the velocity and pressure of the fluid, respectively, while ρ is its density, μ is its dynamic viscosity, and $\mathbf{F}(\mathbf{x}, t) = (F_1(\mathbf{x}, t), \dots, F_d(\mathbf{x}, t))$ is an external forcing function. Note that ρ and μ are assumed to be constant throughout the domain. The boundary conditions for (1) are given by

$$\mathcal{B}(\mathbf{u}, p) = \mathbf{g}(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega, \quad t > 0, \quad (2a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \partial\Omega, \quad t > 0. \quad (2b)$$

where \mathcal{B} is a boundary operator and $\mathbf{g}(\mathbf{x}, t)$ is a forcing function. For example, for no-slip boundary conditions, (2a) becomes

$$\mathcal{B}(\mathbf{u}, p) = \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega, \quad t > 0,$$

where \mathbf{g} is interpreted as the (given) velocity of the wall. We note that an extra boundary condition is needed for the velocity-pressure form of the equations and the continuity equation in (2b), applied on the boundary, is the correct choice. We also include the initial condition

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

where $\mathbf{f}(\mathbf{x})$ is a given initial profile of velocity (assumed to be divergence free).

For the fractional-step scheme discussed later, we consider the equations in a moving frame with a given velocity $\mathbf{w}(\mathbf{x}, t) = (w_1(\mathbf{x}, t), \dots, w_d(\mathbf{x}, t))$, and we also add a divergence-damping term to the right-hand side

of (1b). The resulting equations are

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} - \mathbf{w}) \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{F}(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (3a)$$

$$\Delta p = -\rho \nabla \mathbf{u} : (\nabla \mathbf{u})^T + \nabla \cdot \mathbf{F}(\mathbf{x}, t) + \alpha(\mathbf{x}) \nabla \cdot \mathbf{u}, \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (3b)$$

where $\alpha(\mathbf{x})$ is a chosen coefficient function of the damping term. Note that the damping term has no effect on the solution at the continuous level since $\nabla \cdot \mathbf{u} = 0$ everywhere. The motivation for including the damping term is to suppress the growth of the divergence of the velocity in the numerical solution that may occur due to discretization errors. The fractional-step method advances the velocity in time using a numerical integration of (3a) in one step followed by an update of the pressure from a numerical solution of a Poisson problem involving (3b) in a subsequent step. The Poisson problem requires a suitable boundary condition for pressure, and this can be derived based on a *compatibility condition* involving (3a) and the boundary condition in (2b). Taking the normal component of (3a) and assuming that $\mathbf{w} = \mathbf{u}$ on the boundary (e.g. a no-slip wall) gives

$$\frac{\partial p}{\partial n} = \mathbf{n} \cdot \left(-\rho \frac{\partial \mathbf{u}}{\partial t} + \mu \Delta \mathbf{u} + \mathbf{F}(\mathbf{x}, t) \right), \quad \mathbf{x} \in \partial \Omega, \quad t > 0, \quad (4)$$

where $\mathbf{n}(\mathbf{x}, t)$ is the unit normal on the boundary and $\partial_n p$ is the normal derivative of $p(\mathbf{x}, t)$. Following the discussion in [27], we replace the Laplacian of the velocity in (4) using the vector identity

$$\Delta \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times \nabla \times \mathbf{u},$$

together with the divergence-free condition in (2b) to obtain the so-called curl-curl form of the pressure boundary condition

$$\frac{\partial p}{\partial n} = \mathbf{n} \cdot \left(-\rho \frac{\partial \mathbf{u}}{\partial t} - \mu \nabla \times \nabla \times \mathbf{u} + \mathbf{F}(\mathbf{x}, t) \right), \quad \mathbf{x} \in \partial \Omega, \quad t > 0. \quad (5)$$

The pressure equation in (3b) along with the curl-curl boundary condition in (5) forms the Poisson problem which is solved to update the pressure in the fractional-step scheme.

3. Predictor-corrector methods

The governing equations are discretized using a method-of-lines approach. The integration of the equations in time is described in this section, while the spatial discretization is discussed in the next section. For notational convenience, we consider the momentum equations (3a) and the pressure-Poisson equation (3b) in the form

$$\frac{\partial \mathbf{u}}{\partial t} = \mathcal{L}(\mathbf{u}, p), \quad (6a)$$

$$0 = \mathcal{P}(\mathbf{u}, p), \quad (6b)$$

where

$$\mathcal{L}(\mathbf{u}, p) = -(\mathbf{u} - \mathbf{w}) \cdot \nabla \mathbf{u} - \frac{1}{\rho} \nabla p + \nu \Delta \mathbf{u} + \frac{1}{\rho} \mathbf{F},$$

$$\mathcal{P}(\mathbf{u}, p) = -\Delta p - \rho \nabla \mathbf{u} : (\nabla \mathbf{u})^T + \nabla \cdot \mathbf{F} + \alpha(x) \nabla \cdot \mathbf{u},$$

and $\nu = \mu/\rho$ is the kinematic viscosity. We consider two different linear, multi-step, predictor-corrector (PC) time-integration schemes. The first scheme is explicit and uses a Adams-Bashforth predictor step followed by an Adams-Moulton corrector step. The explicit scheme is suitable for advection dominated problems in which the time-step restriction is effectively determined by the advection terms so that $\Delta t \propto h$, where Δt is the time step and h is a grid spacing. For such a scheme, it is desirable for its stability region to include a portion of the imaginary axis so that it can be used for inviscid problems ($\nu = 0$). We also consider an IMEX-PC scheme in which the viscous terms in the momentum equations are treated implicitly, while the pressure gradient and nonlinear advection terms are handled explicitly. This second PC scheme is suitable

for problems where the viscous terms are stiff (e.g., due to fine grid-spacings in boundary layers), and implicit treatment of the viscous terms ensures that the time-step restriction remains $\Delta t \propto h$. For the description of the IMEX-PC scheme, it is helpful to introduce the further notation

$$\mathcal{L}(\mathbf{u}, p) = \mathcal{L}_E(\mathbf{u}, p) + \mathcal{L}_I(\mathbf{u}), \quad (7)$$

where $\mathcal{L}_E(\mathbf{u}, p)$ and $\mathcal{L}_I(\mathbf{u})$ represent the terms in the momentum equations treated explicitly and implicitly, respectively. These terms are given by

$$\mathcal{L}_E(\mathbf{u}, p) = -(\mathbf{u} - \mathbf{w}) \cdot \nabla \mathbf{u} - \frac{1}{\rho} \nabla p + \frac{1}{\rho} \mathbf{F}, \quad (8a)$$

$$\mathcal{L}_I(\mathbf{u}) = \nu \Delta \mathbf{u}. \quad (8b)$$

For the subsequent description of the two PC schemes, let $\mathbf{U}^n(\mathbf{x})$ and $P^n(\mathbf{x})$ denote approximations of $\mathbf{u}(\mathbf{x}, t^n)$ and $p(\mathbf{x}, t^n)$, respectively, at a time $t^n = n\Delta t$, and let $\mathbf{U}^{(p)}(\mathbf{x})$ and $P^{(p)}(\mathbf{x})$ denote the corresponding predicted quantities at t^{n+1} . The time-stepping scheme assumes that the velocity and pressure are known at four previous time levels, t^k , $k = n-3, \dots, n$, and that the solution is obtained at t^{n+1} after one corrector step. For both PC schemes, the velocity and pressure are advanced in a fractional-step fashion in which the velocity is advanced in time in the first step followed by an update of the pressure. In the two subsections below, we describe the explicit and IMEX time-stepping schemes with a focus on the time-integration of the momentum equations. For clarity, we will refer to a specific predictor-corrector scheme as PC[M_p, M_c], where M_p and M_c denote the methods used for the predictor and corrector steps, respectively.

3.1. Explicit predictor-corrector method

For the explicit-PC method, we use a third-order accurate Adams-Bashforth (AB3) method given by

$$\mathbf{U}^{(p)} = \mathbf{U}^n + \frac{\Delta t}{12} [23\mathcal{L}(\mathbf{U}^n, P^n) - 16\mathcal{L}(\mathbf{U}^{n-1}, P^{n-1}) + 5\mathcal{L}(\mathbf{U}^{n-2}, P^{n-2})], \quad (9a)$$

$$0 = \mathcal{P}(\mathbf{U}^{(p)}, P^{(p)}), \quad (9b)$$

to obtain $(\mathbf{U}^{(p)}, P^{(p)})$ in the predictor step, followed by a fourth-order accurate Adams-Moulton (AM4) method, given by

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \frac{\Delta t}{24} [9\mathcal{L}(\mathbf{U}^{(p)}, P^{(p)}) + 19\mathcal{L}(\mathbf{U}^n, P^n) - 5\mathcal{L}(\mathbf{U}^{n-1}, P^{n-1}) + \mathcal{L}(\mathbf{U}^{n-2}, P^{n-2})], \quad (10a)$$

$$0 = \mathcal{P}(\mathbf{U}^{n+1}, P^{n+1}), \quad (10b)$$

to obtain $(\mathbf{U}^{n+1}, P^{n+1})$ in the corrector step. Note that the PC[AB3,AM4] method is fourth-order accurate even though the method used for the predictor step is only third-order accurate in isolation. Another choice would be to use AB4 in the predictor step, but the region of absolute stability for PC[AB4,AM4] is smaller than that for PC[AB3,AM4], see Figure 2. Also, the region of absolute stability for PC[AB3,AM4] includes a finite interval of the imaginary axis about the origin, so that the scheme can be used for inviscid problems. In contrast, the corresponding region for PC[AB4,AM4] does not include the imaginary axis.

As a standard model, it is helpful to consider the one-dimensional advection-diffusion equation

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad (11)$$

where the velocity a and diffusivity $\nu \geq 0$ are real constants. Assuming Fourier-mode solutions of the form $u(x, t) = \eta(t)e^{ikx}$, where k is a wave number, leads to the test equation

$$\eta' = \lambda \eta, \quad (12)$$

where $\eta(t)$ is the amplitude of the Fourier mode and $\lambda = -ika - \nu k^2$ is the time-stepping eigenvalue for the equation. A stability analysis of the two explicit PC methods applied to (12) leads to the characteristic polynomials

$$\mathcal{C}_3(\zeta) = \zeta^3 - \left(\frac{23}{32}z^2 + \frac{7}{6}z + 1\right)\zeta^2 + \left(\frac{1}{2}z^2 + \frac{5}{24}z\right)\zeta - \left(\frac{5}{32}z^2 + \frac{1}{24}z\right),$$

for PC[AB3,AM4], and

$$\mathcal{C}_4(\zeta) = \zeta^4 - \left(\frac{55}{64}z^2 + \frac{7}{6}z + 1 \right) \zeta^3 + \left(\frac{59}{64}z^2 + \frac{5}{24}z \right) \zeta^2 - \left(\frac{37}{64}z^2 + \frac{1}{24}z \right) \zeta + \frac{9}{64}z^2,$$

for PC[AB4,AM4], where $z = \lambda\Delta t$, and ζ is a complex-valued amplification factor. The regions of absolute stability for the two methods are the set of complex values z for which the roots of their characteristic polynomials satisfy $|\zeta| \leq 1$. These regions are plotted in the top graph of Figure 2. For a typical problem, a stable value for Δt would be determined by the diffusivity so that $\Delta t \propto h^2$, but we observe that there are also stable values of Δt if $\nu = 0$ (i.e. pure imaginary) for the PC[AB2,AM4] method.

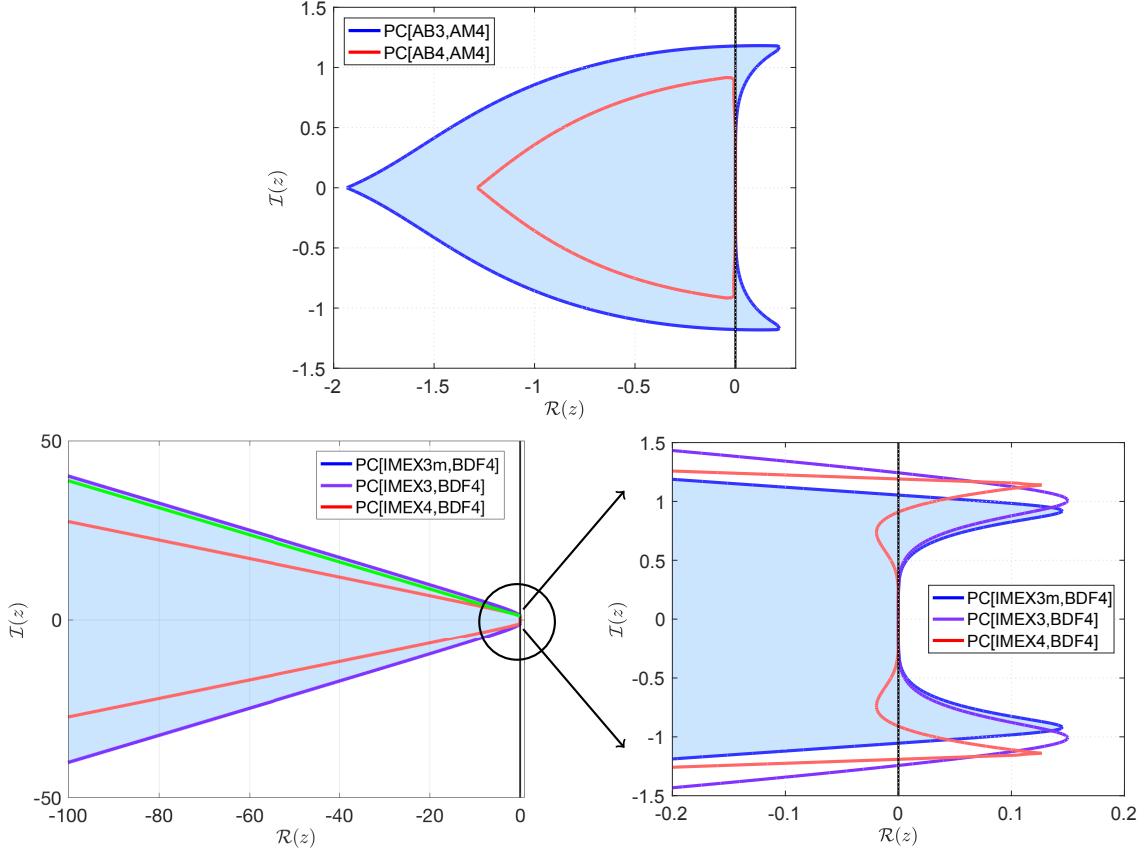


Figure 2: Regions of absolute stability for different predictor-corrector time-stepping schemes. Here, $z = \lambda\Delta t$, where λ is the time-stepping eigenvalue. Top: stability regions for the explicit PC[AB3,AM4] and PC[AB4,AM4] schemes. Bottom: Stability regions for the PC[IMEX3,BDF4], PC[IMEX3m,BDF4] and PC[IMEX4,BDF4] methods with a zoomed version near the imaginary axis on the right.

3.2. IMEX predictor-corrector method

For problems in which viscous effects are important so that the time-step required for an explicit method is constrained by $\Delta t \propto h^2$, we consider an IMEX predictor-corrector method. For this method, the viscous terms of the momentum equations denoted by \mathcal{L}_I in (7) are treated implicitly, while the pressure gradient and nonlinear advection terms denoted by \mathcal{L}_E are still integrated explicitly. The IMEX-PC methods we consider are based on BDF-type time-integration schemes. The corrector step uses the standard BDF4, which is fourth-order accurate, while there is some flexibility in the choice of the BDF-type method for the predictor step (similar to the previous discussion for the explicit-PC method). We consider three choices for the integration of the momentum equations in the predictor step all having the form

$$\mathbf{U}^{(p)} + \sum_{j=1}^4 \hat{\alpha}_j \mathbf{U}^{n+1-j} = \Delta t \hat{\beta}_0 \mathcal{L}_I(\mathbf{U}^{(p)}) + \Delta t \sum_{j=1}^4 \hat{\beta}_j \mathcal{L}_E(\mathbf{U}^{n+1-j}, P^{n+1-j}), \quad (13)$$

where $(\hat{\alpha}_j, \hat{\beta}_j)$ are coefficients. The coefficient for the schemes, referred to as IMEX3, IMEX3m and IMEX4, are listed in Table 1. The update of the predicted pressure, $P^{(p)}$, is determined by (9b) as in the explicit-PC method. The corrector step employs a BDF4 scheme given by

$$\mathbf{U}^{n+1} + \sum_{j=1}^4 \alpha_j \mathbf{U}^{n+1-j} = \Delta t \beta_0 \left[\mathcal{L}_I(\mathbf{U}^{n+1}) + \mathcal{L}_E(\mathbf{U}^{(p)}, P^{(p)}) \right], \quad (14)$$

with coefficients (α_j, β_0) listed in Table 2. The pressure update for P^{n+1} is given by (10b) as before.

For reference, Tables 1 and 2 also include the coefficients for an IMEX-PC scheme based on an IMEX2 predictor and a BDF2 corrector. The resulting second-order accurate PC[IMEX2,BDF2] scheme will be used later to compare results between the new fourth-order PC schemes and a second-order scheme.

IMEX predictor-step coefficients									
Method	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$
IMEX2	-4/3	1/3	0	0	2/3	4/3	-2/3	0	0
IMEX3	-18/11	9/11	-2/11	0	6/11	18/11	-18/11	6/11	0
IMEX3m	-48/25	36/25	-16/25	3/25	12/25	36/25	-36/25	12/25	0
IMEX4	-48/25	36/25	-16/25	3/25	12/25	48/25	-72/25	48/25	-12/25

Table 1: Coefficients $(\hat{\alpha}_j, \hat{\beta}_j)$ for the IMEX2, IMEX3, IMEX3m and IMEX4 predictor schemes.

IMEX corrector-step coefficients					
Method	α_1	α_2	α_3	α_4	β_0
BDF2	-4/3	1/3	0	0	2/3
BDF4	-48/25	36/25	-16/25	3/25	12/25

Table 2: Coefficients (α_j, β_0) for the BDF2 and BDF4 corrector schemes.

The schemes denoted by IMEX3 and IMEX3m for the predictor step are both third-order accurate, while the IMEX4 scheme is fourth-order accurate. However, the three PC-IMEX schemes based on IMEX3, IMEX3m and IMEX4 are all fourth-order accurate since the corrector step given by BDF4 is fourth-order accurate. Note that if $\hat{\beta}_0 = \beta_0$, then the coefficient matrix formed for the implicit solves of the velocity are the same for the predictor and corrector steps, which generally implies a significant savings in computational cost. This reduced cost would occur for the PC[IMEX3m,BDF4] and PC[IMEX4,BDF4] schemes, in contrast to the PC[IMEX3,BDF4] scheme where $\hat{\beta}_0 \neq \beta_0$. In fact, IMEX3m is a modification of the IMEX3 scheme implied by BDF3 that specifically chooses $\hat{\beta}_0 = \beta_0$. Other differences between the three fourth-order schemes can be seen in terms of their regions of absolute stability as shown in the bottom two graphs in Figure 2. These regions are obtained by considering Fourier-mode solutions of the model equation in (11), as before, which leads to the test equation (12) for the amplitude $\eta(t)$ of the Fourier modes. Associating the real part of the time-stepping eigenvalue λ in (12) with \mathcal{L}_I (viscous term) and the imaginary part with \mathcal{L}_E (advective term), and applying the IMEX-PC scheme in (13) and (14) leads to the characteristic polynomial given by

$$\mathcal{C}_{\text{IMEX}}(\zeta) = \zeta^4 + \frac{1}{1 - \tilde{x}\beta_0} \sum_{j=1}^4 \left\{ \alpha_j + \frac{i\tilde{y}\beta_0}{1 - \tilde{x}\hat{\beta}_0} (\hat{\alpha}_j - i\tilde{y}\hat{\beta}_j) \right\} \zeta^{4-j},$$

where $z = \lambda\Delta t = \tilde{x} + i\tilde{y}$. The regions of stability in Figure 2 are given by the set of values of z such that the roots of $\mathcal{C}_{\text{IMEX}}(\zeta)$ satisfy $|\zeta| \leq 1$. We note that the stability regions for all three fourth-order PC-IMEX schemes (and the second-order scheme) contain the negative real axis which is desirable for viscous dominated problems. On the other hand, observe that the stability region for PC[IMEX4,BDF4] does not contain an interval of the imaginary axis, while the regions for PC[IMEX3,BDF4] and PC[IMEX3m,BDF4] do contain an interval of the imaginary axis. Thus, these latter two schemes would be favored for problems

that involve both viscous dominated regions of the computational domain as well as regions where advection dominates. In view of the computational savings associated with the choice $\hat{\beta}_0 = \beta_0$ noted above and the favorable stability region shown in Figure 2, we use PC[IMEX3m,BDF4] as the IMEX-PC scheme for the subsequent calculations in the paper. The lower left graph of Figure 2 shows that for large negative \tilde{x} , the stability regions for all fourth-order accurate schemes expand in the \tilde{y} direction. As $\tilde{x} \rightarrow -\infty$ the boundaries of the stability region approach straight lines whose slope can be determined. This form for the stability regions implies that when viscous effects dominate advection, a rather large-time step can be taken if desired (e.g. to compute low Reynolds number steady-state solutions).

Finally, we note that in contrast to the more commonly used second-order accurate predictor-corrector method PC[AB2,AM2] where repeated applications of the corrector step generally yields a larger stability region, using more than one correction step here could deteriorate the stability region due to the fact that the stability region of the BDF4 method does not include the imaginary axis. The numerical experiments discussed later in Section 5 show that one corrector step is sufficient for fourth-order accuracy, and that the PC[IMEX3m,BDF4] method is stable for all of the tests considered.

3.3. Variable time-step

In general the time-step Δt may change, and it is convenient to have versions of the PC schemes that can handle a variable time-step. For efficiency, the time-step is usually held fixed over many successive steps (e.g. 100 steps or more) by specifying a range of the tolerances on the parameters used to determine the time-step (see Section 4.4.2). This reduces the cost of forming a new time-stepping matrix and associated preconditioner.⁴ Special versions of the IMEX predictor and BDF corrector are constructed to treat a variable time-step. For example, a variable time-step IMEX2 scheme is given by choosing

$$\hat{\alpha}_1 = -\frac{(1 + \eta_2)^2}{\eta_2(2 + \eta_2)}, \quad \hat{\alpha}_2 = -\frac{\hat{\alpha}_1}{(1 + \eta_2)^2},$$

and

$$\beta_0 = \frac{1 + \eta_2}{2 + \eta_2}, \quad \beta_1 = \frac{\beta_0(1 + \eta_2)}{\eta_2}, \quad \beta_2 = -\frac{\beta_0}{\eta_2}.$$

Here, $\eta_j = \Delta t_j / \Delta t_1$, where Δt_j , $j = 1, 2, \dots$, denote past values of the time-step with $t^{n-1} = t^n - \Delta t_1$, $t^{n-2} = t^n - \Delta t_1 - \Delta t_2$, etc., while Δt denotes the current time-step, $t^{n+1} = t^n + \Delta t$ as before. Unfortunately, with this version, β_0 depends on Δt_1 and Δt_2 which means the implicit time-stepping matrix needs to be reformed multiple times until β_0 becomes constant again. This can be avoided by enforcing β_0 to be constant, but in doing so one additional solution time-level is needed to maintain second-order accuracy. This results in the modified scheme with coefficients

$$\hat{\alpha}_1 = \frac{(1 + 3\eta_2)(1 - \eta_2 - \eta_3) - 4\eta_2}{3\eta_2(\eta_2 + \eta_3)}, \quad \hat{\alpha}_2 = \frac{-(1 - \eta_2 - \eta_3)}{3\eta_2\eta_3}, \quad \hat{\alpha}_3 = \frac{(1 - \eta_2)}{3\eta_3(\eta_2 + \eta_3)},$$

and

$$\beta_0 = \frac{2}{3}, \quad \beta_1 = \beta_0 \frac{(1 + \eta_2)}{\eta_2}, \quad \beta_2 = -\frac{\beta_0}{\eta_2},$$

where the coefficient β_0 is fixed. Now the the time-stepping matrix, which depends on $\beta_0\Delta t$, need only be reformed once during the transition from one fixed time-step to another one.

Listing 1: Maple code to compute the coefficients in a variable time-step IMEX4 scheme.

```
t1 := t-dt1: t2 := t-dt1-dt2: t3 := t-dt1-dt2-dt3: t4 := t-dt1-dt2-dt3-dt4:
t5 := t-dt1-dt2-dt3-dt4-dt5:

b0 := 12/25; # fix leading coefficient
# Corrector:
```

⁴For moving grids, however, the implicit time-stepping matrix is reformed at every time-step since the grid changes, and so keeping a fixed time-step is not important in this case.

```

fp := t^p + ( a1*t1^p + a2*t2^p + a3*t3^p + a4*t4^p + a5*t5^p ) - dt1*b0*p*t^(p-1):
f1 := subs(p=0,fp); f2 := subs(p=1,fp): f3 := subs(p=2,fp): f4 := subs(p=3,fp): f5 := subs(p=4,fp):
s1 := solve({f1=0,f2=0,f3=0,f4=0,f5=0},{a1,a2,a3,a4,a5}): assign(s1):

# Predictor:
fpe := t^p + ( a1*t1^p + a2*t2^p + a3*t3^p + a4*t4^p + a5*t5^p )
- dt1*( b1*p*t1^(p-1) + b2*p*t2^(p-1) + b3*p*t3^(p-1)+ b4*p*t4^(p-1) ):
f4 := subs(p=1,fpe): f5 := subs(p=2,fpe): f6 := subs(p=3,fpe): f7 := subs(p=4,fpe):
s2 := solve({f4=0,f5=0,f6=0,f7=0},{b1,b2,b3,b4}): assign(s2):

```

It is straightforward to derive the higher-order accurate variable time-step schemes. Consider, for example, deriving the coefficients in the variable-time-step IMEX4 scheme, with one additional time-level so that β_0 is constant. The coefficients $\hat{\alpha}_j$, $j = 1, 2, 3, 4, 5$ can be computed by enforcing that the modified BDF corrector is exact for the monomials t^p , $p = 0, 1, 2, 3, 4$. The values of β_j , $j = 1, 2, 3, 4$, can then be computed by enforcing that the IMEX predictor be exact for the monomials t^p , $p = 1, 2, 3, 4$. Although conceptually straightforward, the resulting formulae are long and unwieldy, and so rather than explicitly writing them down, we instead provide a Maple code, in Listing 1, to evaluate the formulae. In this listing, $\hat{\alpha}_j$, $j = 1, 2, 3, 4$ are denoted by $a1$, $a2$, etc, while β_j , $j = 1, 2, 3, 4$ are denoted by $b1$, $b2$, etc. Also note that the variable time-step schemes reduce to the corresponding fixed time-step schemes when $\Delta t_j = \Delta t$ for all j , i.e. when $\eta_j = 1$.

4. Discretization on overlapping grids

The spatial discretization of the governing equations is carried out on a (moving) composite overlapping grid. We begin this section with a brief overview of the overlapping grid framework, and then describe the fourth-order accurate discretization of the equations for a representative curvilinear component grid. The discretization of the viscous terms use standard centered finite differences, while the advective terms use an upwind BWENO scheme. We provide a description of the BWENO scheme along with an extension to handle the flow in the vicinity of a stagnation point. Lastly, we provide pseudo-codes that describe the fully discrete time-stepping algorithms based on the explicit and IMEX predictor-corrector methods. These pseudo-codes also describe our numerical treatment of no-slip boundary conditions.

4.1. Overlapping grids

An overlapping grid, denoted as \mathcal{G} , consists of a set of component grids $\{\mathcal{G}_g\}$, $g \in [1, \mathcal{N}]$, that cover the fluid domain Ω and overlap where they meet, see Figure 3. Each component grid \mathcal{G}_g is a logically rectangular, curvilinear grid defined by a smooth transformation \mathbf{G}_g given by

$$\mathbf{x} = \mathbf{G}_g(\mathbf{r}, t), \quad \mathbf{r} \in [0, 1]^3, \quad \mathbf{x} \in \mathbb{R}^3,$$

where $\mathbf{r} = (r_1, r_2, r_3)$ are unit computational coordinates (assuming $d = 3$ for the present discussion). The mapping may also depend on time in the case of a moving domain. The overlapping grid generator **Ogen**⁵ is used to create the composite grid, and to categorize the grid points belonging to each component grid as discretization, interpolation or unused points. In a typical overlapping grid configuration, surfaces near the boundary of the fluid domain are represented by thin boundary-fitted curvilinear grids, while the bulk of the fluid domain is covered by Cartesian grids. Holes are cut in the appropriate component grids by locating the physical boundary, thus identifying unused points based on their location. The grid generator also provides the interpolation information for all interpolation points in the overlap region between component grids. For instance, the ‘‘cylinder’’ grid displayed in the upper-right image of Figure 3 cuts a hole in the Cartesian ‘‘box’’ grid so that the latter grid has many unused points (those not being plotted in the lower-right image). The interpolation points are marked by the square points. A detailed description of the discretization of the equations on a single component grid \mathcal{G}_g will be given in Section 4.2.

⁵**Ogen** is available online at overtureframework.org.

For moving grids the governing equations are solved in a frame moving with the grids, and thus no interpolation is required to transfer the numerical solution from component grids at old times to those at new times as a result of the motion. The relative position of the component grids changes as a function of time. Accordingly, **Ogen** is called at each time step to update the boundary-fitted grids corresponding to moving boundaries, and new holes are cut in static background grids. Since the composite grid is only updated locally, this process retains much of the efficiency of a static structured grid, and is much cheaper than a complete grid regeneration.

In preparation for the discretization using overlapping grids, the governing equations and the boundary conditions are transformed from physical space coordinates to computational space coordinates for a given mapping $\mathbf{x} = \mathbf{G}_g(\mathbf{r}, t)$ using the chain rule. **Ogen** provides the metrics, $\partial r_j / \partial x_i$, $i, j = 1, 2$ or 3 , for each component-grid mapping, as well as the grid velocity, $\mathbf{w} = \dot{\mathbf{G}}_g(\mathbf{r}, t)$. In this manuscript, rigid bodies with prescribed motions of their center of mass given by $\mathbf{x}_b(t)$ are considered. This motion, in turn, determines the motion of component grids in the fluid domain fitted to the surface of the bodies. A further discussion of the issues related to moving composite grids can be found in [25].

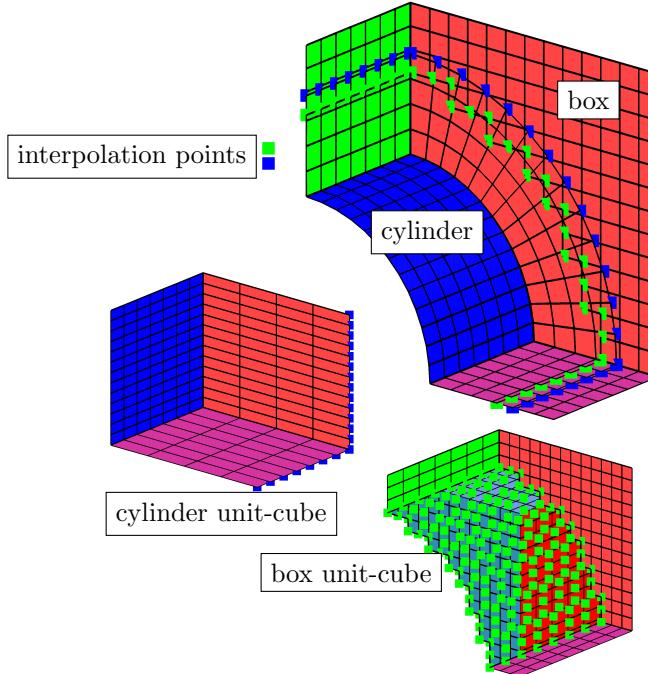


Figure 3: Three-dimensional overlapping grid for a quarter-cylinder in a box: overlapping grid in physical space (top view) and the corresponding component grids on the unit cube in parameter space (bottom views). Interpolation points at the grid overlap are marked and color-coded for each component grid.

4.2. Discretization on a component grid

We now describe the spatial discretization of the governing equations on a single curvilinear component grid within the overlapping grid system. The component grid number g will be dropped in the subsequent discussion for convenience. The discretization is carried out on a uniform grid in the mapped domain with grid points $\mathbf{r}_i = (i_1 h_1, i_2 h_2, i_3 h_3)$ on the unit cube, where $h_m = 1/N_m$ is the grid spacing in the r_m -coordinate direction for a chosen positive integer N_m . The interior and boundary points corresponding to the index space, denoted by Ω_h , are given by

$$\Omega_h = \left\{ \mathbf{i} \mid i_m = 0, 1, 2, \dots, N_m, \quad m = 1, 2, 3 \right\}.$$

Points on the boundary of the index space are denoted by $\partial\Omega_h$. In addition to the points in Ω_h , two ghost lines are included to facilitate discretization to fourth-order accuracy. Points on the first ghost line are denoted by $\partial\Omega_{G_1}$, while points on the second ghost line are given by $\partial\Omega_{G_2}$.

Let \mathbf{U}_i^n and P_i^n denote the numerical approximations to $\mathbf{u}(\mathbf{x}_i, t^n)$ and $p(\mathbf{x}_i, t^n)$. Let D_{0m} , D_{+m} and D_{-m} denote the usual centred, forward, and backward divided-difference operators defined by

$$D_{0m}w_j \stackrel{\text{def}}{=} \frac{w_{j+\mathbf{e}_m} - w_{j-\mathbf{e}_m}}{2h_m}, \quad D_{+m}w_j \stackrel{\text{def}}{=} \frac{w_{j+\mathbf{e}_m} - w_j}{h_m}, \quad D_{-m}w_j \stackrel{\text{def}}{=} \frac{w_j - w_{j-\mathbf{e}_m}}{h_m},$$

where \mathbf{e}_m is the unit vector in r_m -direction, e.g. $\mathbf{e}_2 = (0, 1, 0)$. The first and second derivatives with respect to \mathbf{r} discretized by the standard fourth-order centered difference approximations are given by

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial r_m} &\approx D_{r_m} \mathbf{U}_i \stackrel{\text{def}}{=} D_{0m} \left(1 - \frac{h_m}{6} D_{+m} D_{-m} \right) \mathbf{U}_i, \\ \frac{\partial^2 \mathbf{u}}{\partial r_m^2} &\approx D_{r_m, r_m} \mathbf{U}_i \stackrel{\text{def}}{=} D_{+m} D_{-m} \left(1 - \frac{h_m^2}{12} D_{+m} D_{-m} \right) \mathbf{U}_i, \end{aligned}$$

where the time level n has been dropped for notational convenience. Derivatives with respect to \mathbf{x} are then defined by the chain rule, and are given by

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial x_m} &\approx D_{x_m} \mathbf{U}_i \stackrel{\text{def}}{=} \sum_n \frac{\partial r_n}{\partial x_m} D_{r_n} \mathbf{U}_i, \\ \frac{\partial^2 \mathbf{u}}{\partial x_m^2} &\approx D_{x_m, x_m} \mathbf{U}_i \stackrel{\text{def}}{=} \sum_n \frac{\partial^2 r_n}{\partial x_m^2} D_{r_n, r_n} \mathbf{U}_i + \sum_{n, \ell} \frac{\partial r_n}{\partial x_m} \frac{\partial r_\ell}{\partial x_m} D_{r_n} D_{r_\ell} \mathbf{U}_i, \end{aligned}$$

where the metrics of the mapping are given by the grid generator as noted previously.

The right-hand side of the momentum equations defined by $\mathcal{L}(\mathbf{u}, p)$ in (6a) and the pressure-Poisson equation denoted by $\mathcal{P}(\mathbf{u}, p)$ in (6b) are discretized using

$$\mathcal{L}_h(\mathbf{U}_i, P_i) = -[(\mathbf{U}_i - \mathbf{w}_i) \cdot \nabla]_{\mathcal{B}} \mathbf{U}_i - \frac{1}{\rho} \nabla_h P_i + \nu \Delta_h \mathbf{U}_i + \frac{1}{\rho} \mathbf{F}_i, \quad \mathbf{i} \in \Omega_h, \quad (15a)$$

$$\mathcal{P}_h(\mathbf{U}_i, P_i) = -\Delta_h P_i - \rho \nabla_h \mathbf{U}_i : (\nabla_h \mathbf{U}_i)^T + \nabla_h \cdot \mathbf{F}_i + \alpha(\mathbf{x}_i) \nabla_h \cdot \mathbf{U}_i, \quad \mathbf{i} \in \Omega_h, \quad (15b)$$

where

$$\nabla_h P_i = (D_{x_1}, D_{x_2}, D_{x_3}) P_i, \quad \Delta_h \mathbf{U}_i = (D_{x_1, x_1} + D_{x_2, x_2} + D_{x_3, x_3}) \mathbf{U}_i,$$

$$\nabla_h \cdot \mathbf{U}_i = \sum_m D_{x_m} U_{m,i}, \quad \nabla_h \mathbf{U}_i : (\nabla_h \mathbf{U}_i)^T = \sum_m \sum_n (D_{x_n} U_{m,i}) (D_{x_m} U_{n,i}).$$

The discretization of the advection terms in (15a), denoted by $[(\mathbf{U}_i - \mathbf{w}_i) \cdot \nabla]_{\mathcal{B}} \mathbf{U}_i$, use the BWENO scheme as discussed in the next subsection. The explicit and implicit terms in the momentum equations defined by $\mathcal{L}_E(\mathbf{u}, p)$ in (8a) and $\mathcal{L}_I(\mathbf{u})$ in (8b) are discretized using

$$\mathcal{L}_{E,h}(\mathbf{U}_i, P_i) = -[(\mathbf{U}_i - \mathbf{w}_i) \cdot \nabla]_{\mathcal{B}} \mathbf{U}_i - \frac{1}{\rho} \nabla_h P_i + \frac{1}{\rho} \mathbf{F}_i, \quad \mathbf{i} \in \Omega_h, \quad (16a)$$

$$\mathcal{L}_{I,h}(\mathbf{U}_i) = \nu \Delta_h \mathbf{U}_i, \quad \mathbf{i} \in \Omega_h. \quad (16b)$$

4.3. Discretization of the advective terms

The discretization of the advective terms in the momentum equations is carried out using the BWENO scheme [6]. For regions of the flow near stagnation points, the stabilizing upwind dissipation from the original BWENO scheme vanishes, as is common for upwind schemes near stagnation points, and so we also describe an extension to ensure that dissipation is active even near stagnation points.

4.3.1. The BWENO upwind scheme

The application of the BWENO scheme described here for the discretization of the advective terms in the momentum equations uses a fourth-order centered finite-difference approximation for well-represented solutions, but can also introduce numerical dissipation when the solution varies rapidly and cannot be represented on a given computational grid (such as in a boundary layer). The BWENO scheme used here is

constructed using a combination of third-order accurate one-sided approximations at the cell faces of the grid. For smooth regions, the biased approximations are given equal weight, which yields the centered scheme. For under-resolved regions, upwind dissipation is provided by applying more weight to the upwind-biased stencil. In this way, the BWENO scheme prescribes a solution-dependent switch to transition smoothly between a fourth-order centered finite-difference approximation and a third-order upwind approximation. One important feature of the scheme is that it preserves the five-point discretization stencil of the centered fourth-order approximation, and thus no extra ghost points are needed to facilitate the discretization near the boundary.

For a curvilinear grid defined by a known mapping with the metrics, $\partial r_\ell / \partial x_n$, $\ell, n \in \{1, 2, 3\}$, the k -th component of the advection term in (15a) is given analytically by

$$[(\mathbf{u} - \mathbf{w}) \cdot \nabla] u_k = \sum_\ell v_\ell \frac{\partial u_k}{\partial r_\ell}, \quad v_\ell \stackrel{\text{def}}{=} \sum_n (u_n - w_n) \frac{\partial r_\ell}{\partial x_n}.$$

The form of the corresponding BWENO discretization is

$$[(\mathbf{U}_i - \mathbf{w}_i) \cdot \nabla]_{\mathcal{B}} U_{k,i} = \sum_\ell \mathcal{B}_h (V_{\ell,i}, U_{k,i}), \quad V_{\ell,i} = \sum_n (U_{n,i} - w_{n,i}) \left(\frac{\partial r_\ell}{\partial x_n} \right)_i. \quad (17)$$

To describe the discrete BWENO operator, denoted $\mathcal{B}_h (V_{\ell,i}, U_{k,i})$ in (17), we focus on a representative term with $\ell = k = 1$. For this choice, and to simplify notation, we drop the ℓ and k subscripts and consider only the first component of the multi-index \mathbf{i} since the other two components remain fixed. With these simplifications, the BWENO operator yields an approximation to vu_r evaluated at r_i , and is defined as

$$\mathcal{B}_h (V_i, U_i) \stackrel{\text{def}}{=} V_i \left(\frac{\hat{U}_{i+\frac{1}{2}} - \hat{U}_{i-\frac{1}{2}}}{h} \right), \quad (18)$$

where h is the grid spacing in the r_1 -direction, and $\hat{U}_{i \pm \frac{1}{2}}$ are chosen so that (18) is a high-order accurate approximation to vu_r . In the present case, this implies $\hat{U}_{i \pm \frac{1}{2}} \approx (u - \frac{h^2}{24} u_{rr})|_{r_{j \pm 1/2}}$. Note that additional details of the approach to defining $\hat{U}_{i \pm \frac{1}{2}}$ for arbitrary order-of-accuracy are discussed in [28, 29]. The face values of $\hat{U}_{i \pm \frac{1}{2}}$ are now expressed as a combination of lower-order biased approximations using the data $\{U_i\}$, as indicated in Figure 4. For example, $\hat{U}_{i+\frac{1}{2}}$ is given by

$$\hat{U}_{i+\frac{1}{2}} = \omega_{i+\frac{1}{2}}^L \hat{U}_{i+\frac{1}{2}}^L + \omega_{i+\frac{1}{2}}^R \hat{U}_{i+\frac{1}{2}}^R,$$

where

$$\hat{U}_{i+\frac{1}{2}}^L = \frac{1}{6} (-U_{i-1} + 5U_i + 2U_{i+1}), \quad \hat{U}_{i+\frac{1}{2}}^R = \frac{1}{6} (2U_i + 5U_{i+1} - U_{i+2}).$$

Here, $\hat{U}_{i+\frac{1}{2}}^L$ and $\hat{U}_{i+\frac{1}{2}}^R$ are left and right biased approximations indicated with the L and R superscripts respectively, and $\omega_{i+\frac{1}{2}}^L$ and $\omega_{i+\frac{1}{2}}^R$ are the associated weights. Similar formulas are used to obtain $\hat{U}_{i-\frac{1}{2}}$ in (18).

The weights in the formulas for $\hat{U}_{i \pm \frac{1}{2}}$ are designed such that the approximation in (18) is fourth-order accurate at grid points where the solution is smooth. Note that if $\omega_{i \pm \frac{1}{2}}^L = \omega_{i \pm \frac{1}{2}}^R = \frac{1}{2} \stackrel{\text{def}}{=} \omega^*$, then (18) reduces to a standard fourth-order centered approximation of vu_r at r_i . In order to maintain fourth-order accuracy, the weights must converge to ω^* as h approaches zero. Following [30], it is sufficient to require

$$\omega_{i \pm \frac{1}{2}}^L = \omega^* + \mathcal{O}(h^2), \quad \omega_{i \pm \frac{1}{2}}^R = \omega^* + \mathcal{O}(h^2),$$

with

$$\omega_{i \pm \frac{1}{2}}^L + \omega_{i \pm \frac{1}{2}}^R = 1.$$

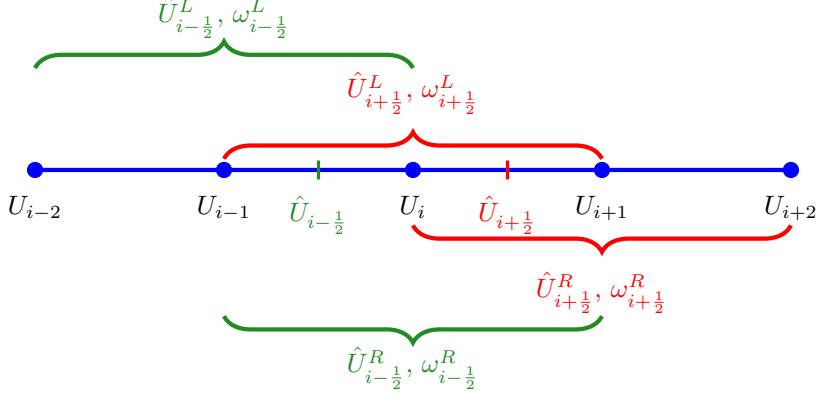


Figure 4: The approximated velocity $\hat{U}_{i+\frac{1}{2}}$ (marked in red) is defined as a convex combination of a left approximated velocity $\hat{U}_{i+\frac{1}{2}}^L$ and a right approximated velocity $\hat{U}_{i+\frac{1}{2}}^R$ with weights $\omega_{i+\frac{1}{2}}^L$ and $\omega_{i+\frac{1}{2}}^R$, respectively. $\hat{U}_{i-\frac{1}{2}}$ (marked in green) is constructed in the same fashion using $\hat{U}_{i-\frac{1}{2}}^L$ and $\hat{U}_{i-\frac{1}{2}}^R$.

Focusing on the pair, $\omega_{i+\frac{1}{2}}^s$, $s = L, R$, we set

$$\omega_{i+\frac{1}{2}}^s = \frac{\alpha_{i+\frac{1}{2}}^s}{\alpha_{i+\frac{1}{2}}^L + \alpha_{i+\frac{1}{2}}^R}, \quad \alpha_{i+\frac{1}{2}}^s = \frac{1}{(\epsilon + \beta_{i+\frac{1}{2}}^s)^2}, \quad s = L, R, \quad (19)$$

where ϵ is a small parameter (typically $\epsilon = 1 \times 10^{-40}$) and $\beta_{i+\frac{1}{2}}^s$ is a grid function designed to measure the smoothness of the solution. Following [6], we set

$$\beta_{i+\frac{1}{2}}^s = h \int_{r_i}^{r_{i+1}} \left(\frac{d}{dr} P_{i+\frac{1}{2}}^s(r) \right)^2 dr + h^3 \int_{r_i}^{r_{i+1}} \left(\frac{d^2}{dr^2} P_{i+\frac{1}{2}}^s(r) \right)^2 dr, \quad s = L, R,$$

where $P_{i+\frac{1}{2}}^s(r)$, $s = L, R$, are quadratic polynomial fits to the data $\{U_i\}$ given by

$$\begin{aligned} P_{i+\frac{1}{2}}^L(r) &= U_i + (r - r_i) D_0 U_i + \frac{1}{2} (r - r_i)^2 D_+ D_- U_i, \\ P_{i+\frac{1}{2}}^R(r) &= U_{i+1} + (r - r_{i+1}) D_0 U_{i+1} + \frac{1}{2} (r - r_{i+1})^2 D_+ D_- U_{i+1}. \end{aligned}$$

Note that if the solution is smooth on the grid, then $\beta_{i+\frac{1}{2}}^L \approx \beta_{i+\frac{1}{2}}^R$ and thus $\omega_{i+\frac{1}{2}}^s \approx \omega^*$. As with traditional WENO schemes, convergence rates near critical points (e.g. local minimum in U_i) may be less than optimal for the weights given in (19). To restore optimal convergence for most types of critical points, we follow the analysis in [30] and use a mapping of the weights defined by

$$\omega_{i+\frac{1}{2}}^s \leftarrow \frac{\gamma_{i+\frac{1}{2}}^s}{\gamma_{i+\frac{1}{2}}^L + \gamma_{i+\frac{1}{2}}^R}, \quad s = L, R, \quad (20)$$

where

$$\gamma_{i+\frac{1}{2}}^s = g(\omega_{i+\frac{1}{2}}^s) = \frac{1}{2} + \frac{1}{2} \left(\frac{\omega_{i+\frac{1}{2}}^s}{\omega^*} - 1 \right)^3. \quad (21)$$

Note that the function $g(\omega)$ in (21) has the properties $g(0) = 0$, $g(1) = 1$, $g(\omega^*) = \omega^*$ and $g'(\omega^*) = g''(\omega^*) = 0$, and it is identical to the one suggested in [30] for the special case $\omega^* = \frac{1}{2}$. Thus, the mapping in (20) with (21) has the effect of increasing the order of the zero about the ideal weight near critical points. As a final step, we ensure that upwind dissipation is added in the BWENO approximation by assigning the larger

of the two weights $(\omega_{i+\frac{1}{2}}^L, \omega_{i+\frac{1}{2}}^R)$ to $\omega_{i+\frac{1}{2}}^s$ according the sign of V_i defined originally in (17). Specifically, we set

$$\left. \begin{array}{l} \omega_{i+\frac{1}{2}}^L \leftarrow \max(\omega_{i+\frac{1}{2}}^L, \omega_{i+\frac{1}{2}}^R) \\ \omega_{i+\frac{1}{2}}^R \leftarrow \min(\omega_{i+\frac{1}{2}}^L, \omega_{i+\frac{1}{2}}^R) \end{array} \right\}, \quad \text{if } V_i \geq 0,$$

and

$$\left. \begin{array}{l} \omega_{i+\frac{1}{2}}^L \leftarrow \min(\omega_{i+\frac{1}{2}}^L, \omega_{i+\frac{1}{2}}^R) \\ \omega_{i+\frac{1}{2}}^R \leftarrow \max(\omega_{i+\frac{1}{2}}^L, \omega_{i+\frac{1}{2}}^R) \end{array} \right\}, \quad \text{if } V_i < 0.$$

We note that BWENO discretization of the advective terms defined in (17) and using the weights described above converges at fourth-order accuracy for smooth flows. The discretization also uses the same stencil width as the standard centered fourth-order accurate finite-difference approximation. Moreover, the discretization smoothly transitions from a fourth-order accurate centered approximation to a third-order accurate one-sided biased approximation, introducing a suitable artificial dissipation, in regions where the solution is under-resolved.

4.3.2. Modified BWENO scheme

Two special cases have been found for which the BWENO scheme described above does not provide sufficient numerical dissipation. Both cases occur near stagnation points in flows that are under-resolved on the grid. For such cases, the fractional-step time-stepping scheme becomes unstable and thus a modification of the BWENO scheme is needed. The essential reason for the instability is that the artificial dissipation introduced in the BWENO scheme is the only source of spatial dissipation in the time-stepping scheme and it is proportional to the flow velocity. Thus, near a stagnation point the BWENO dissipation vanishes and the scheme may become unstable. The two cases that have been found for this instability involve stagnation points in the interior of the flow domain (such as in a shear layer), and at points along a no-slip wall.

For the case of a stagnation point in the interior of the flow domain, a modification in the BWENO scheme given in (17) is used if one or more components of the velocity $V_{\ell,i}$ changes sign. As before, we focus the discussion on the case with $\ell = k = 1$ for simplicity, and only consider the first component i in the multi-index \mathbf{i} . For a point r_i with $V_{i-1}V_{i+1} < 0$, we replace $\mathcal{B}_h(V_i, U_i)$ in (18) with a modified advective discretization given by

$$\mathcal{B}_{h,\text{mod}}(V_i, U_i) = V_i D_0 \left(1 - \frac{h}{6} D_+ D_- \right) U_i + \frac{V_{\max}}{12h} \left(h^2 D_+ D_- \right)^2 U_i, \quad (22)$$

where

$$V_{\max} = \max\{|V_{i-1}|, |V_i|, |V_{i+1}|\},$$

is the local maximum of the magnitude of the velocity about r_i . Note that the first term on the right-hand side of (22) is a standard fourth-order accurate centered approximation of the advective derivative, while the second term provides the dissipation in proportion to V_{\max} . If the solution is well-resolved on the grid, then $V_{\max} = \mathcal{O}(h)$ and the dissipation term is $\mathcal{O}(h^4)$ so that the truncation error remains fourth order. If the solution is under-resolved, then the dissipation term can be larger, and it provides a sufficient dissipation to stabilize the time-stepping scheme.

Another important case where a modification in the BWENO scheme is needed occurs at points along a no-slip boundary. The approach to adding numerical dissipation follows a similar one given in (22). The details are slightly different, however, as the modification is coupled to the application of the numerical boundary conditions. As a result, the details of this modification are described later in Section 4.4.1 along with a discussion of the boundary conditions.

4.4. Fully discrete time-stepping algorithms

We now present a detailed description of the explicit and IMEX fractional-step methods for the incompressible Navier-Stokes equations. In the following two subsections, we present pseudo-codes describing one time-step of the explicit-PC and IMEX-PC time-stepping schemes with a particular focus on the discretization of the boundary conditions. For clarity, we only describe the application of a no-slip boundary condition.

Other types of boundary conditions (inflow, outflow, slip-wall, etc.) can be dealt with in a similar fashion, although the details of their implementation may vary.

Algorithm 1 Explicit PC[AB3,AM4] fractional-step scheme

1. Advance the overlapping grid \mathcal{G} using the user-defined grid velocity \mathbf{w}_i^{n+1} .
- // Prediction steps
2. Compute the predicted velocity on the interior and boundary points explicitly:
$$\begin{cases} \mathbf{U}_i^{(p)} = \mathbf{U}_i^n + \frac{\Delta t}{12} [23\mathcal{L}_h(\mathbf{U}_i^n, P_i^n) - 16\mathcal{L}_h(\mathbf{U}_i^{n-1}, P_i^{n-1}) + 5\mathcal{L}_h(\mathbf{U}_i^{n-2}, P_i^{n-2})], & \mathbf{i} \in \Omega_h \\ \mathbf{U}_i^{(p)} = \mathbf{w}_i^{n+1}, & \mathbf{i} \in \Omega_{\partial h} \end{cases} \quad :: \text{no-slip condition}$$
3. Specify the predicted velocity on the ghost points:
$$\begin{cases} \nabla_h \cdot \mathbf{U}_i^{(p)} = 0, & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{divergence-free cond.} \\ D_{n,h}(\nabla_h \cdot \mathbf{U}_i^{(p)}) = 0, & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{derivative of divergence-free cond.} \\ \mathbf{t}_m \cdot [\mu \Delta_h \mathbf{U}_i^{(p)} + \mathcal{D}_d(\mathbf{U}_i^{(p)})] = \mathbf{t}_m \cdot (\rho \dot{\mathbf{w}}_i^{n+1} + \nabla_h P_i^{(e)} - \mathbf{F}_i^{n+1}), & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{compatability cond.} \\ \mathbf{t}_m \cdot D_{+n}^5 \mathbf{U}_i^{(p)} = 0, & \mathbf{i} \in \Omega_{G_2} \quad :: \text{extrapolation} \end{cases}$$
4. Update the predicted pressure:
$$\begin{cases} \Delta_h P_i^{(p)} = -\rho \nabla_h \mathbf{U}_i^{(p)} : (\nabla_h \mathbf{U}_i^{(p)})^T + \nabla_h \cdot \mathbf{F}_i^{n+1} + \alpha(\mathbf{x}_i) \nabla_h \cdot \mathbf{U}_i^{(p)}, & \mathbf{i} \in \Omega_h \cup \Omega_{\partial h} \\ D_{n,h} P_i^{(p)} = \mathbf{n} \cdot (-\rho \dot{\mathbf{w}}_i^{n+1} - \mu \nabla_h \times \nabla_h \times \mathbf{U}_i^{(p)} + \mathbf{F}_i^{n+1}), & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{curl-curl condition} \\ D_{+n}^5 P_i^{(p)} = 0, & \mathbf{i} \in \Omega_{G_2} \quad :: \text{extrapolation} \end{cases}$$
- // Correction steps
5. Compute the corrected velocity on the interior and boundary points explicitly:
$$\begin{cases} \mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{24} [9\mathcal{L}_h(\mathbf{U}_i^{(p)}, P_i^{(p)}) + 19\mathcal{L}_h(\mathbf{U}_i^n, P_i^n) - 5\mathcal{L}_h(\mathbf{U}_i^{n-1}, P_i^{n-1}) + \mathcal{L}_h(\mathbf{U}_i^{n-2}, P_i^{n-2})], & \mathbf{i} \in \Omega_h \\ \mathbf{U}_i^{n+1} = \mathbf{w}_i^{n+1}, & \mathbf{i} \in \Omega_{\partial h} \end{cases}$$
6. Specify the corrected velocity on the ghost points:
$$\begin{cases} \nabla_h \cdot \mathbf{U}_i^{n+1} = 0, & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{divergence-free cond.} \\ D_{n,h}(\nabla_h \cdot \mathbf{U}_i^{n+1}) = 0, & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{derivative of divergence-free cond.} \\ \mathbf{t}_m \cdot [\mu \Delta_h \mathbf{U}_i^{n+1} + \mathcal{D}_d(\mathbf{U}_i^{n+1})] = \mathbf{t}_m \cdot (\rho \dot{\mathbf{w}}_i^{n+1} + \nabla_h P_i^{(p)} - \mathbf{F}_i^{n+1}), & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{compatability cond.} \\ \mathbf{t}_m \cdot D_{+n}^5 \mathbf{U}_i^{n+1} = 0, & \mathbf{i} \in \Omega_{G_2} \quad :: \text{extrapolation} \end{cases}$$
7. Update the corrected pressure:
$$\begin{cases} \Delta_h P_i^{n+1} = -\rho \nabla_h \mathbf{U}_i^{n+1} : (\nabla_h \mathbf{U}_i^{n+1})^T + \nabla_h \cdot \mathbf{F}_i^{n+1} + \alpha(\mathbf{x}_i) \nabla_h \cdot \mathbf{U}_i^{n+1}, & \mathbf{i} \in \Omega_h \cup \Omega_{\partial h} \\ D_{n,h} P_i^{n+1} = \mathbf{n} \cdot (-\rho \dot{\mathbf{w}}_i^{n+1} - \mu \nabla_h \times \nabla_h \times \mathbf{U}_i^{n+1} + \mathbf{F}_i^{n+1}), & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{curl-curl condition} \\ D_{+n}^5 P_i^{n+1} = 0, & \mathbf{i} \in \Omega_{G_2} \quad :: \text{extrapolation} \end{cases}$$

4.4.1. Fractional-step algorithms

Algorithm 1 describes the steps for the fractional-step method based on the explicit PC[AB3,AB4] time-stepping scheme discussed in Section 3.1. The first step of the algorithm moves the composite grid according to the user-defined grid motion \mathbf{w}_i^{n+1} . The overlapping grid generator **Ogen** is called to update the overlapping grid connectivity information, which includes an updated classification of points as discretization, unused, or interpolation points (as described previously). The numerical solution is now advanced in time according to the prediction steps 2–4. In Step 2, the momentum equations are advanced to t^{n+1} with the explicit multistep predictor scheme (AB3). At this step, the predicted velocity, denoted by $\mathbf{U}_i^{(p)}$, is obtained on the interior points using known grid functions for the velocity and pressure at previous time-levels on the interior, boundary and ghost points of the grid denoted by Ω_h , $\Omega_{\partial h}$ and Ω_G , respectively. A no-slip boundary condition is also applied at this step, while values for the predicted velocity at ghost points are specified in Step 3. Conditions for the two ghost lines are obtained from fourth-order accurate discretizations of the divergence-free condition in (2b) and its normal derivative, and a compatibility condition involving the momentum equation projected into the tangent directions denoted by \mathbf{t}_m , $m = 1, 2$. In this compatibility condition, $\dot{\mathbf{w}}_i^{n+1}$ is the (known) grid acceleration at t^{n+1} and $P_i^{(e)}$ is an extrapolated value of pressure given

by

$$P_{\mathbf{i}}^{(e)} = 3P_{\mathbf{i}}^n - 3P_{\mathbf{i}}^{n-1} + P_{\mathbf{i}}^{n-2}. \quad (23)$$

Also appearing in the compatibility condition is a dissipation operator, $\mathcal{D}_d(\mathbf{U}_{\mathbf{i}}^{(p)})$, given by

$$\mathcal{D}_d(\mathbf{U}_{\mathbf{i}}^{(p)}) = \delta_d \max_{1 \leq m \leq d} |D_{+m} \mathbf{V}_{\mathbf{i}}| \sum_{m=1}^d \left(h_m^2 D_{+m} D_{-m} \right)^2 \mathbf{U}_{\mathbf{i}}, \quad (24)$$

where each component of $\mathbf{V}_{\mathbf{i}}$ is defined in (17) and δ_d is a tunable coefficient, usually taken to be 1, although larger values may be needed for very poorly resolved problems. The dissipation in (24) is similar to the one introduced previously for the modified BNEWO scheme in (22). Note that the conditions in Step 3 centered at $\Omega_{\partial h}$ involve points on the first and second ghost lines, $\partial\Omega_{G_1}$ and $\partial\Omega_{G_2}$. Note also that fifth-order extrapolation is required to ensure fourth-order accuracy as discussed in [4], while the fourth-order extrapolation in time given in (23) is sufficient to maintain third-order accuracy of the AB3 predictor step.

In terms of efficiency of the solver, we remark that the velocity and pressure are decoupled in the compatibility conditions by the use of the extrapolation in (23). Also, we note that values of the predicted velocity in the two ghost lines are coupled through the numerical boundary conditions in Step 4, and they are computed as a system of equations. Since the system only contains unknowns in the ghost points, the coupled problem is one dimension less than the full domain problem. Therefore, solving this system is relatively inexpensive computationally.

The predicted pressure is computed in Step 4 by solving a discrete Poisson problem. This Poisson problem involves a fourth-order accurate discretization of the curl-curl compatibility condition in (5) and a fifth-order extrapolation of the pressure. As before, the conditions for the two ghost lines are coupled, but these are now combined with the full linear system constructed to compute $P_{\mathbf{i}}^{(p)}$. The resulting linear systems are computed with direct solution methods for small problems, and Krylov subspace methods and geometric multigrid solvers for larger problems.

Algorithm 2 IMEX PC[IMEX3m,BDF4] fractional-step scheme

1. Advance the overlapping grid \mathcal{G} using the user-defined grid velocity $\mathbf{w}_{\mathbf{i}}^{n+1}$.
- // Prediction steps
2. Compute the predicted velocity on the interior and boundary points implicitly:
$$\begin{cases} [\mathbf{I} - \Delta t \hat{\beta}_0 \mathcal{L}_{I,h}] \mathbf{U}_{\mathbf{i}}^{(p)} = -\sum_j \hat{\alpha}_j \mathbf{U}_{\mathbf{i}}^{n+1-j} + \Delta t \sum_j \hat{\beta}_j \mathcal{L}_{E,h}(\mathbf{U}_{\mathbf{i}}^{n+1-j}, P_{\mathbf{i}}^{n+1-j}), & \mathbf{i} \in \Omega_h \\ \mathbf{U}_{\mathbf{i}}^{(p)} = \mathbf{w}_{\mathbf{i}}^{n+1}, & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{no-slip condition} \\ D_{+n}^5 \mathbf{U}_{\mathbf{i}}^{(p)} = 0, & \mathbf{i} \in \Omega_G \quad :: \text{extrapolation} \end{cases}$$
3. Specify the predicted velocity on the ghost points:
(Same procedure as Step 3 of Algorithm 1)
4. Update the predicted pressure:
(Same procedure as Step 4 in Algorithm 1)
- // Correction steps
5. Compute the corrected velocity on the interior and boundary points:
$$\begin{cases} [\mathbf{I} - \Delta t \beta_0 \mathcal{L}_{I,h}] \mathbf{U}_{\mathbf{i}}^{n+1} = -\sum_j \alpha_j \mathbf{U}_{\mathbf{i}}^{n+1-j} + \Delta t \beta_0 \mathcal{L}_{E,h}(\mathbf{U}_{\mathbf{i}}^{(p)}, P_{\mathbf{i}}^{(p)}), & \mathbf{i} \in \Omega_h \\ \mathbf{U}_{\mathbf{i}}^{n+1} = \mathbf{w}_{\mathbf{i}}^{n+1}, & \mathbf{i} \in \Omega_{\partial h} \quad :: \text{no-slip condition} \\ D_{+n}^5 \mathbf{U}_{\mathbf{i}}^{n+1} = 0, & \mathbf{i} \in \Omega_G \quad :: \text{extrapolation} \end{cases}$$
6. Specify the corrected velocity on the ghost points:
(Same procedure as Step 6 in Algorithm 1)
7. Update the corrected pressure:
(Same procedure as Step 7 in Algorithm 1)

Steps 5–7 in Algorithm 1 describe the corrector steps, and these are similar to ones for the predictor steps. The main difference is that an AM4 scheme is used to compute the corrected velocity to fourth-order

accuracy in Step 5. Step 6 completes the specification of the corrected velocity, and the corrected pressure is obtained in Step 7 by solving a discrete Poisson problem as before.

The fully discrete IMEX-PC time-stepping scheme, based on the PC[IMEX3m,BDF4] method developed in Section 3.2, is given in Algorithm 2. The overall structure of the IMEX-PC time-stepping scheme is similar to that of the explicit-PC scheme. The overlapping grid is advanced in the first step of the algorithm according to a user-defined motion, and then Steps 2–4 and 5–7 describe the predictor and corrector steps, respectively. The principal difference in the two time-stepping algorithms lies in the approaches used to compute the predicted and corrected velocities in Steps 2 and 5. For the present IMEX-PC scheme (with coefficients given previously in Tables 1 and 2), the discrete viscous terms in the momentum equations are treated implicitly. This implicit system includes the primary no-slip boundary condition specified at grid points on the boundary of the domain, as well as a fifth-order extrapolation to the first line of ghost points. This is done so that the components of the fluid velocities decouple, and thus each velocity component can be determined by separate linear system solves. In Steps 3 and 6, the velocities in the first and second ghost lines are updated using the previous numerical boundary conditions, i.e. the divergence-free condition, the normal derivative of the divergence-free condition, the compatibility condition as well as the extrapolation condition are updated with known values of velocity on the interior and the boundary. The predicted and corrected pressures are obtained in Steps 4 and 7, respectively, using the predicted and corrected velocities as before.

4.4.2. Time-step determination

The time-step, Δt , for the fractional-step schemes are estimated from a local von-Neumann analysis. This analysis determines a time-stepping eigenvalue

$$\lambda = \lambda_r + i\lambda_i, \quad \lambda_r, \lambda_i \in \mathbb{R},$$

that estimates the eigenvalue of largest magnitude to the discrete spatial operator from a method-of-lines approximation. The real part of the time-stepping eigenvalue comes primarily from the viscous terms in the equations while the imaginary part comes primarily from the advection terms (similar to the model problem analysis considered previously in Section 3). For a Cartesian grid, the real and imaginary parts of the time-stepping eigenvalue are determined from

$$\lambda_r = C_v \sum_{m=1}^d \frac{1}{\Delta x_m^2}, \quad \lambda_i = C_a \max_{\mathbf{i}} \sum_{m=1}^d \frac{|\mathbf{V}_i|}{\Delta x_m},$$

where $\mathbf{V}_i = \mathbf{U}_i - \mathbf{w}_i$ is the relative velocity vector, C_a and C_v are constants depending on the discrete approximations, and Δx_m is the grid spacing in m -th physical-space direction. Similar formulae are used for curvilinear grids. Given the time-stepping eigenvalue, the time-step Δt is determined by fitting an ellipse to the stability region of the scheme (see Figure 2),

$$\left(\frac{\lambda_r \Delta t}{a_e} \right)^2 + \left(\frac{\lambda_i \Delta t}{b_e} \right)^2 \leq (C_{\text{CFL}})^2, \quad (25)$$

where C_{CFL} is the CFL-safety factor normally taken to be 0.9. For the fourth-order accurate explicit-PC scheme we take⁶ $C_a = 20/12$ and $C_v = 64/12$ with $a_e = -1.7$ and $b_e = 1.15$. For the IMEX-PC scheme we again choose $C_a = 20/12$ but set $C_v = 0$, since the viscous terms impose no time-step restriction, and take $b_e = 1.05$. Note that for some problems, e.g. a slow start when the velocities are small, one may want to choose the time-step based on accuracy considerations rather than solely on stability considerations. In this case a time-step based on some error estimator, such as a standard ODE type error estimator, may be appropriate. Currently, however, we choose the time-step heuristically in this situation to be proportional to the mesh size divided by the anticipated velocity scale.

⁶The values of the coefficients C_a and C_v arise from the amplification of a plus-minus mode when applied to the the fourth-order accurate central difference approximations to the first and second-derivatives, respectively.

4.4.3. Remarks on implementation of the algorithms

In this section we make a few additional remarks concerning the practical implementation of the fractional-step schemes.

Extrapolation of the pressure. A key ingredient of the fractional-step algorithm is the choice of boundary conditions that allows the solution of the velocity and pressure to be treated in separate steps. The boundary conditions for the velocity in Step 3 of Algorithms 1 and 2 make use of an extrapolated value for the pressure gradient, $\nabla_h P^{(e)}$. Note that the predictor-step only needs to be third-order accurate in time for the overall predictor-corrector scheme to be fourth-order accurate in time. This explains the uses of a third-order accurate extrapolation formula for $P^{(e)}$ in (23). A second-order accurate extrapolation would not be accurate enough, while using a high-order extrapolation is prone to be less stable.

Divergence damping. The divergence damping coefficient, $\alpha(\mathbf{x})$, that appears in the pressure equation (3b), is taken to be constant, and is determined by

$$\alpha = \min \left\{ C_d \sum_{m=1}^d \frac{\nu}{\Delta x_m^2}, C_{\Delta t} \frac{1}{\Delta t} \right\},$$

for the case of a Cartesian grid (with a similar form used for curvilinear grids). Here, the first damping coefficient C_d is usually taken to be 1, while the second damping coefficient is set to $C_{\Delta t} = 0$ for the explicit time-stepping scheme and $C_{\Delta t} = 0.25$ for the IMEX scheme. The value of α is limited for the IMEX scheme since otherwise it could reduce the maximum stable time-step Δt .

Linear system solver. The linear systems for the velocity in Algorithm 2 and the pressure update in both algorithms can be solved using various approaches for sparse systems. We typically use a multigrid solver [31] or one of the Krylov solvers from PETSc [32] such as bi-CG-Stab with an ILU preconditioner. In some cases, such as a domain with no-slip boundary conditions for all $\mathbf{x} \in \partial\Omega$, the Poisson problem for the pressure is singular with the pressure only determined up to a constant. For such cases, the linear system for the pressure can be augmented with an additional unknown and a constraint that sets the constant, as discussed in [3, 17].

Other boundary conditions. To this point the focus has been on the treatment of no-slip wall boundary conditions. Other useful boundary conditions include

$$\text{Inflow (velocity): } \mathbf{u}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t), \quad (26a)$$

$$\text{Inflow (pressure): } \begin{cases} p(\mathbf{x}, t) = g(\mathbf{x}, t), \\ \mathbf{t}_m^T \mathbf{u}(\mathbf{x}, t) = g_m(\mathbf{x}, t), \end{cases} \quad (26b)$$

$$\text{Slip-wall: } \begin{cases} \mathbf{n} \cdot \mathbf{v}(\mathbf{x}, t) = 0, \\ \partial_n(\mathbf{t}_m^T \cdot \mathbf{u}) = 0, \end{cases} \quad (26c)$$

$$\text{Outflow: } \begin{cases} \alpha p + \beta \partial_n p = g(\mathbf{x}, t), \\ \partial_n \mathbf{u} = 0 \text{ or extrapolation of } \mathbf{u}. \end{cases} \quad (26d)$$

For many of these boundary conditions, the divergence constraint can be added, $\nabla \cdot \mathbf{u} = 0$, as well as extra compatibility conditions and extrapolation of ghost points where appropriate. Most of the boundary conditions (26) can be considered to be artificial boundary conditions that are applied where the domain has been truncated to a finite region. The outflow condition (26d), for example, is somewhat ad-hoc, but has been found to be useful in practice at allowing disturbances in the flow to exit the domain. The Neumann condition on the velocity at outflow in (26d) is used instead of extrapolation when there may be local regions of inflow at the outflow boundary (as may happen when a strong vortex leaves the domain).

5. Numerical results

We now present computational results which verify the implementation, and demonstrate the stability and accuracy properties, of the newly developed fourth-order accurate time-stepping methods and the fourth-order accurate spatial approximations and boundary conditions. The tests are also used to illustrate the effectiveness of the new BWENO scheme for the advection terms. We demonstrate that the BWENO scheme becomes virtually indistinguishable from the centered scheme for resolved solutions, while for under-resolved simulations the BWENO scheme remains robust and stable for cases when the centered scheme is unstable. Note that all numerical studies in this section are obtained using the Overture-based Cgns solver [33, 34].

For this section we introduce shorter labels for the following key schemes evaluated:

- IMEX22 : the semi-implicit second-order accurate scheme PC[IMEX2,BDF2].
- IMEX24 : the second-order in time coupled to fourth-order in space scheme PC[IMEX2,BDF2].
- IMEX44 : the semi-implicit fourth-order accurate scheme PC[IMEX3m,BDF4].
- IMEX44-BWENO : the IMEX44 scheme with BWENO upwind dissipation.
- PC44 : the explicit fourth-order accurate scheme PC[AB3,AM4].
- PC44-BWENO : the PC44 scheme with BWENO upwind dissipation.

Here, the notations PC_{mn} and $IMEX_{mn}$ indicate schemes with m^{th} -order accuracy in time and n^{th} -order accuracy in space. Note that the IMEX24 scheme introduced here is used for comparative purposes in Sections 5.1.3 and 5.3 to assess the contributions to the error from the temporal and spatial approximations.

5.1. Convergence analysis for two-dimensional flows

We begin by considering several test problems in two dimensions. These include verification tests for a problem using a manufactured solution as well as two physical problems for which exact solutions are known. Careful grid refinement studies are performed in order to confirm the accuracy of the new schemes.

5.1.1. Manufactured solution for a rotating disk in a square

We consider a two-dimensional problem defined for a domain involving a rotating disk embedded in a square. The outer boundary of the fluid domain Ω is chosen to be the perimeter of the square $[0, 2] \times [0, 2]$, while the inner boundary is the perimeter of a solid disk of radius equal to 0.2 centered at $\mathbf{x} = (1, 1)$. A counterclockwise rotation of the disk about its center with an angular velocity equal to 0.2π is imposed. The computational domain is covered by a circle-in-channel composite grid, $\mathcal{G}_{\text{cic}}^{(j)}$. The composite grid, shown in Figure 5, consists of a Cartesian grid and a boundary-fitted annular grid attached to the rotating inner boundary. The number of grid lines in each coordinate direction for the two component grids are chosen so that the grid spacings in both directions are equal to $h_j = 1/(80j)$ approximately for the composite grid $\mathcal{G}_{\text{cic}}^{(j)}$ with resolution index j .

To demonstrate the spatial and temporal accuracy of the solver, a convergence study is carried out using a manufactured solution given by

$$\begin{aligned} u_1(\mathbf{x}, t) &= \frac{1}{2} \cos(kx_1) \cos(kx_2) \cos(\omega t) + \frac{1}{2}, \\ u_2(\mathbf{x}, t) &= \frac{1}{2} \sin(kx_1) \sin(kx_2) \cos(\omega t) + \frac{1}{2}, \\ p(\mathbf{x}, t) &= \cos(kx_1) \cos(kx_2) \cos(\omega t) + \frac{1}{2}, \end{aligned}$$

with $k = \omega = 1.1\pi$. Note that the velocity, $\mathbf{u} = (u_1, u_2)$, of the manufactured solution is divergence free. No-slip conditions are applied at the outer boundary of the fluid domain, as well as along the rotating inner wall. Numerical solutions are computed using the PC44 and IMEX44 schemes with and without the BWENO discretization of the advective terms. The fluid density and viscosity are chosen as $\rho = 1$ and $\nu = 0.01$, respectively, and the equations are integrated to $t_{\text{final}} = 0.5$.

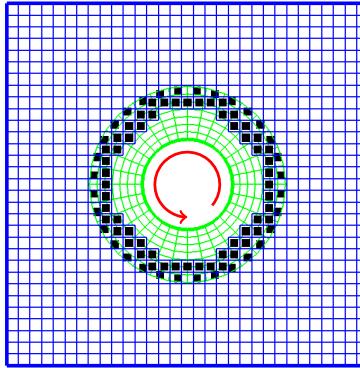


Figure 5: Coarse version of the composite grid \mathcal{G}_{cic} for the two-dimensional test using manufactured solutions. The grid consists of a Cartesian component grid (blue) and a boundary-fitted annular grid (green) connected by the interpolation points (marked points). The boundary-fitted grid rotates with an angular velocity equal to 0.2π .

Rotating disk in a square: PC44-BWENO scheme									
h_j	Δt_j	E_j^p	σ_j^p	$E_j^{u_1}$	$\sigma_j^{u_1}$	$E_j^{u_2}$	$\sigma_j^{u_2}$	$E_j^{\nabla \cdot \mathbf{u}}$	$\sigma_j^{\nabla \cdot \mathbf{u}}$
1/80	1/50	8.40e-5	—	1.42e-4	—	1.17e-4	—	1.78e-3	—
1/160	1/200	4.30e-6	4.29	7.34e-6	4.27	5.60e-6	4.38	9.89e-5	4.17
1/240	1/450	7.87e-7	4.19	1.28e-6	4.31	9.59e-7	4.35	1.60e-5	4.49
1/320	1/800	2.59e-7	3.86	3.72e-7	4.30	2.75e-7	4.34	4.79e-6	4.19
rate		4.2		4.3		4.4		4.3	

Table 3: Maximum-norm errors of the solution components computed using the PC44 scheme with BWENO upwinding.

The results of grid convergence studies using the PC44-BWENO and IMEX44-BWENO schemes are reported in Tables 3 and 4, respectively. Numerical solutions are computed using the $\mathcal{G}_{\text{cic}}^{(j)}$ grid with $j = 1, 2, 3$ and 4 , and maximum-norm errors in the components of the solutions are found for each case. The time-steps are taken to be $\Delta t_j = 1/(50j^2)$ for the explicit scheme and $\Delta t_j = 1/(40j)$ for the IMEX scheme. The columns labeled σ_j^κ in the tables are pairwise estimates of the convergence rates based on the errors at grid resolutions j and $j - 1$ given by

$$\sigma_j^\kappa = \frac{\log(E_{j-1}^\kappa/E_j^\kappa)}{\log(h_{j-1}/h_j)}, \quad (27)$$

where E_j^κ is the maximum-norm error in component κ at grid resolution j . The estimated convergence rates listed along the bottom row of the tables are determined by least-squares fits to the errors for each component of the solution (as well as the computed divergence of the velocity). We observe that the estimated convergence rate for each component is close to four. We also note that the errors in the solution obtained

Rotating disk in a square: IMEX44-BWENO									
h_j	Δt_j	E_j^p	σ_j^p	$E_j^{u_1}$	$\sigma_j^{u_1}$	$E_j^{u_2}$	$\sigma_j^{u_2}$	$E_j^{\nabla \cdot \mathbf{u}}$	$\sigma_j^{\nabla \cdot \mathbf{u}}$
1/80	1/40	8.77e-5	—	1.42e-4	—	1.17e-4	—	1.78e-3	—
1/160	1/80	4.39e-6	4.32	7.35e-6	4.27	5.31e-6	4.46	9.90e-5	4.17
1/240	1/120	7.72e-7	4.29	1.27e-6	4.33	9.06e-7	4.36	1.65e-5	4.42
1/320	1/160	2.63e-7	3.74	3.74e-7	4.25	2.60e-7	4.34	4.69e-6	4.37
rate		4.2		4.3		4.4		4.3	

Table 4: Maximum-norm errors of the solution components computed using the IMEX44 scheme with BWENO upwinding.

using the two time-stepping schemes are nearly the same for each component, even though the time-step for the IMEX scheme is much larger at the finest resolution than that of the explicit method. This suggests that the error in the spatial discretization (at fourth-order accuracy) dominates for this problem.

Rotating disk in a square: PC44 scheme									
h_j	Δt_j	E_j^p	σ_j^p	$E_j^{u_1}$	$\sigma_j^{u_1}$	$E_j^{u_2}$	$\sigma_j^{u_2}$	$E_j^{\nabla \cdot \mathbf{u}}$	$\sigma_j^{\nabla \cdot \mathbf{u}}$
1/80	1/40	8.56e-5	—	1.35e-4	—	1.00e-4	—	1.46e-3	—
1/160	1/80	4.30e-6	4.32	6.88e-6	4.29	5.24e-6	4.25	8.94e-5	4.03
1/240	1/120	7.63e-7	4.26	1.28e-6	4.15	9.13e-7	4.31	1.65e-5	4.17
1/320	1/160	2.62e-7	3.72	3.74e-7	4.28	2.60e-7	4.37	4.69e-6	4.37
rate		4.2		4.2		4.3		4.1	

Table 5: Maximum-norm errors of the solution components computed using the PC44 method in time and with a centered difference discretization for all spatial derivatives, including the advective terms.

A final mesh refinement study is carried out for this problem using the PC44 method in time and centered differences for all spatial derivatives, including the advective terms. Results are reported in Table 5. We note that the numerical solution is smooth for all grid resolutions, and thus the maximum-norm errors of the solution components computed with the centered discretization of the advective terms are very close to those in Table 4 obtained using the BWENO discretization. We note that the errors in the solution on the coarsest grid ($j = 1$) for the fully centered scheme are slightly smaller than those in the solution computed using the BWENO scheme, but on the finest grid resolution ($j = 4$) the errors are nearly the same. This trend is in agreement with the fact that the BWENO discretization generally approaches a centered scheme when the solution is well resolved.

5.1.2. Taylor-Green vortex flow

We next consider the Taylor-Green vortex problem in a square geometry $\Omega = [0, 1] \times [0, 1]$. The exact solution is given by

$$\begin{aligned} u_1(\mathbf{x}, t) &= \sin(kx_1) \cos(kx_2) \exp(-2k^2\nu t), \\ u_2(\mathbf{x}, t) &= -\cos(kx_1) \sin(kx_2) \exp(-2k^2\nu t), \\ p(\mathbf{x}, t) &= \frac{\rho}{4} [\cos(2kx_1) + \cos(2kx_2)] \exp(-4k^2\nu t), \end{aligned}$$

where k is a wave number [35]. The parameters are taken as $k = 4\pi$, $\rho = 1$ and $\nu = 0.05$. The boundary conditions are taken to be periodic in both directions. Numerical solutions are computed using a Cartesian grid, $\mathcal{G}_{\text{TG}}^{(j)}$, where the grid spacing is chosen to be $h_j = 1/(16j)$. Maximum-norm errors for the velocity (maximum of the errors for the two components) and the pressure are computed at $t_{\text{final}} = 0.2$ on a sequence of grids of increasing resolution with $j = 1, 2, 4, 8, 16$. Three different schemes are compared: IMEX22, IMEX44 and PC44. Figure 6 shows the behavior of the errors as a function of h_j along with shaded contour plots of the u_1 and p components on the solution. The results obtained using IMEX22 are seen to converge at close to second-order accuracy, while the results found using IMEX44 and PC44 both converge at close to fourth-order accuracy. The errors for the two fourth-order accurate schemes are similar in size and become much smaller than the errors for the second-order accurate scheme on the finer grids.

5.1.3. Circular Couette flow

As a final convergence test for a flow in two dimensions, we consider a circular Couette flow in an annular region given by $a \leq r \leq b$, where $r^2 = x_1^2 + x_2^2$. It is assumed that the inner and outer boundaries rotate with angular velocities ω_a and ω_b , respectively, and that the radial component of the velocity is zero, while the circumferential component u_θ is a function of (r, t) only. Under these assumptions, the velocity is divergence free, and a solution of the circumferential component of the momentum equations can be taken as

$$u_\theta(r, t) = \frac{A}{r} + Br + \alpha [J_1(\lambda r)Y_1(\lambda a) - Y_1(\lambda r)J_1(\lambda a)] e^{-\lambda^2\nu t},$$

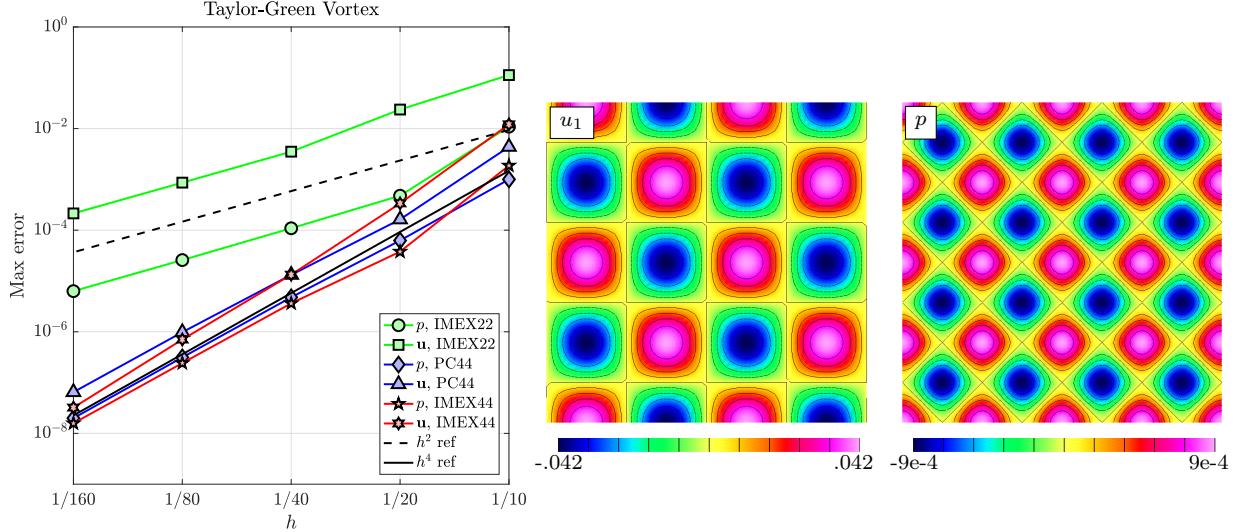


Figure 6: Taylor-Green vortex. Left: maximum-norm errors in velocity and pressure versus grid spacing. Middle and right: contours of u_1 and p at $t_{\text{final}} = 0.2$.

where

$$A = \frac{a^2 b^2 (\omega_a - \omega_b)}{b^2 - a^2}, \quad B = \frac{b^2 \omega_b - a^2 \omega_a}{b^2 - a^2},$$

and λ is a root (eigenvalue) of the transcendental equation

$$\phi(\lambda) = J_1(\lambda b) Y_1(\lambda a) - Y_1(\lambda b) J_1(\lambda a).$$

Here, J_1 and Y_1 are Bessel functions of order one. For the purposes of the convergence test, we choose $(a, b) = (0.5, 1)$ and $\alpha = 1$, and take λ to be the smallest root of $\phi(\lambda) = 0$, which is $\lambda \approx 6.3912$. The angular velocities of the boundaries are set as $(\omega_a, \omega_b) = (1, 2)$, while the viscosity is taken to be $\nu = 0.05$. The fluid pressure is determined by the momentum equation in the radial direction, and it is given by

$$p(r, t) = \int_a^r \rho [u_\theta(\bar{r}, t)]^2 \frac{d\bar{r}}{\bar{r}}. \quad (28)$$

Note that an arbitrary constant could be added to the pressure in (28) since the pressure is only determined up to a constant for this problem.

Numerical solutions for this problem are computed using the composite grid $\mathcal{G}_{\text{da}}^{(j)}$, with resolution index j , consisting of two overlapping boundary-fitted annular component grids as shown in Figure 7. For both component grids, the grid spacing is chosen to be approximately $h_j = 1/(10j)$. The initial conditions are taken from the exact solution and no-slip boundary conditions are specified on the inner and outer boundaries. On the inner boundary the velocity is set to match the angular velocity ω_a . The outer grid rotates with angular velocity ω_b . Figure 7 shows the maximum-norm errors at $t_{\text{final}} = 0.1$ for four different schemes as the mesh is refined. In addition to errors from the second-order accurate IMEX22 scheme, and the fourth-order accurate PC44 and IMEX44 schemes, we also plot the errors obtained using the IMEX24 scheme which is second-order accurate in time and fourth-order accurate in space. The solutions obtained using the IMEX22 and IMEX24 schemes are seen to converge at close to second order, while the solutions given by the two fourth-order accurate schemes are seen to converge at close to fourth order. The errors for the two fourth-order accurate schemes are similar in size. The IMEX44 scheme, however, is more efficient than the explicit PC44 scheme on the finer grids since it can use a larger time-step. We note that similar maximum-norm errors are obtained from the results when the outer annular grid is static and has the velocity set on its boundary, and from the results obtained using a composite grid with just one annular component grid covering the whole domain and thus requiring no grid overlap.

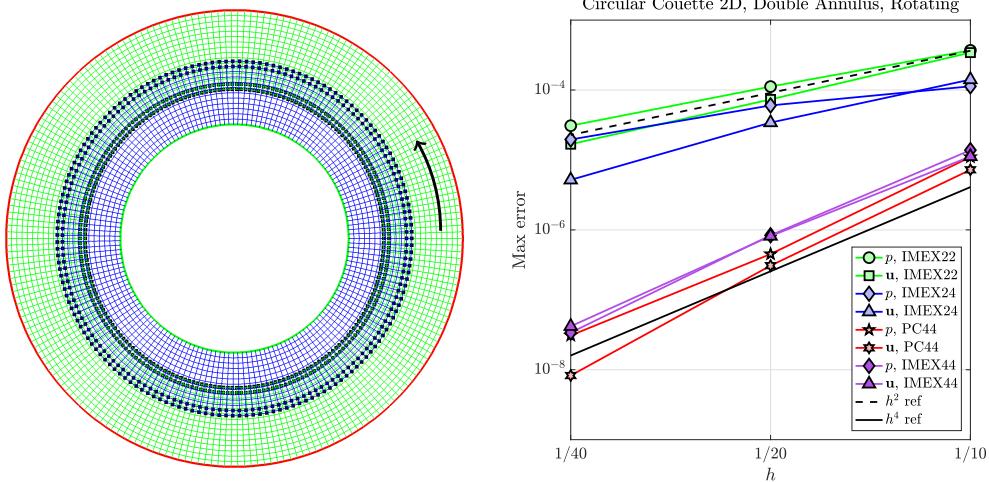


Figure 7: Circular Couette flow. Left: composite grid for the annular region; the outer green grid rotates to match the velocity of the outer boundary while the inner blue grid is static with the velocity on its boundary specified. Right: maximum-norm errors versus grid spacing for the second-order accurate IMEX22 scheme, the second-order accurate in time and fourth-order accurate in space IMEX24 scheme, the fourth-accurate explicit PC44 scheme and the fourth-order accurate IMEX44 scheme.

5.2. Convergence analysis for three-dimensional flows

In this section, we continue our verification of the new schemes by considering two three-dimensional problems for which exact solutions are either manufactured or available.

5.2.1. Manufactured solution for a rotating sphere in a box

As a first test of the time-stepping schemes for a flow in three dimensions, we consider a manufactured solution for a domain consisting of a rotating sphere inside a box. The box is given by $[-1.5, 1.5]^3$ and the sphere is centered at the origin with radius equal to 0.5. Similar to the previous problem in two dimensions, the sphere rotates in the positive direction about the x_3 -axis with an angular velocity equal to 0.2π . The computational domain is covered by a sphere-in-box composite grid, $\mathcal{G}_{\text{sib}}^{(j)}$, consisting of a Cartesian grid and two boundary-fitted orthographic grid patches attached to the surface of the sphere. A coarse version of the composite grid is shown in Figure 8. The number of grid lines in each coordinate direction for the composite grid with resolution index j is chosen so that the grid spacing is approximately $h_j = 1/(40j)$ in every direction.

Appropriate forcing functions are specified so that the exact solution is given by

$$\begin{aligned} u_1(\mathbf{x}, t) &= \cos(kx_1) \cos(kx_2) \cos(kx_3) \cos(\omega t), \\ u_2(\mathbf{x}, t) &= \frac{1}{2} \sin(kx_1) \sin(kx_2) \cos(kx_3) \cos(\omega t), \\ u_3(\mathbf{x}, t) &= \frac{1}{2} \sin(kx_1) \sin(kx_2) \sin(kx_3) \cos(\omega t), \\ p(\mathbf{x}, t) &= \frac{1}{2} \sin(kx_1) \cos(kx_2) \cos(kx_3) \sin(\omega t), \end{aligned}$$

with $k = \omega = 1.1\pi$ as in the corresponding two-dimensional problem. No-slip boundary conditions are applied at the outer surface of the box and on the surface of the rotating sphere. The exact solution is divergence free, and it is used to compute the velocity on the boundary and the initial conditions. The fluid density and viscosity are set to be $\rho = 1$ and $\nu = 0.1$. Numerical solutions are computed to $t_{\text{final}} = 0.4$ for the IMEX scheme and to $t_{\text{final}} = 0.05$ for the explicit scheme. Since the time-step for the explicit scheme is much smaller than that for the IMEX scheme, a smaller value for t_{final} is chosen for the explicit scheme to speed up the computations.

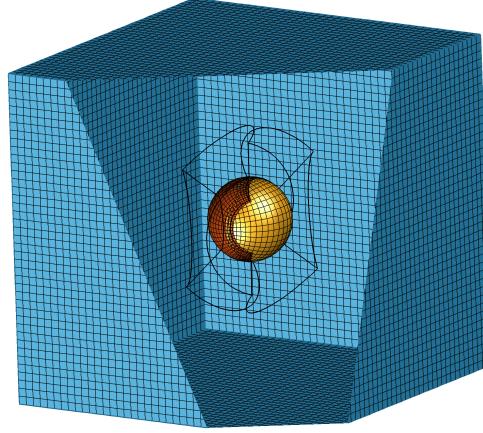


Figure 8: Composite grid for a sphere in a box. An edge of the Cartesian grid is cut away to reveal the two orthographic component grids fitted to the surface of the sphere. The sphere rotates about a line through its center with direction parallel to the vertical x_3 -axis.

Rotating sphere in a box: PC44-BWENO method											
h_j	Δt_j	E_j^p	σ_j^p	$E_j^{u_1}$	$\sigma_j^{u_1}$	$E_j^{u_2}$	$\sigma_j^{u_2}$	$E_j^{u_3}$	$\sigma_j^{u_3}$	$E_j^{\nabla \cdot \mathbf{u}}$	$\sigma_j^{\nabla \cdot \mathbf{u}}$
1/40	1/1000	1.61e-3	—	1.53e-3	—	1.43e-3	—	1.14e-3	—	1.13e-2	—
1/80	1/4000	7.44e-5	4.44	8.51e-5	4.17	6.41e-5	4.48	5.84e-5	4.29	5.61e-4	4.33
1/120	1/9000	1.46e-5	4.02	1.64e-5	4.06	1.23e-5	4.07	1.08e-5	4.16	1.13e-4	3.95
rate		4.3		4.1		4.3		4.2		4.2	

Table 6: Maximum-norm errors of solutions computed using the PC44 scheme with the BWENO upwinding.

Numerical solutions are computed using the PC44-BWENO and IMEX44-BWENO schemes for grid resolutions $j = 1, 2$ and 3 . The corresponding approximate grid spacings are $h_j = 1/40, 1/80$ and $1/120$, and stable values for the time-step, Δt_j are chosen for each case. For the explicit method, $\Delta t_j \propto h_j^2$, while we set $\Delta t_j \propto h_j$ for the IMEX method. The errors in the solution components (and the divergence of velocity), as well as the pairwise convergence rates using (27), are given in Tables 6 and 7. We observe that the pairwise convergence rates, and the rate along the bottom row obtained by least-squares fits to the errors, are all approximately equal to four indicating fourth-order convergence of the time-stepping methods for this test in three dimensions. As in the two-dimensional case, it is worth noting that the errors in the components of the solution at the finest grid resolution are similar between the two time-stepping methods, but the stable time-step required for the IMEX method is much bigger than that for the explicit method.

5.2.2. Couette-Poiseuille flow in a cylindrical pipe

As a second test problem in three space dimensions, we consider a Couette-Poiseuille flow in a cylindrical pipe. The pipe has length L and radius R . The axial component of the velocity, u_1 , is assumed to be a superposition of a steady flow driven by a prescribed (constant) body force $F_1 = \rho G$, and a time-dependent

Rotating sphere in a box: IMEX44-BWENO scheme											
h_j	Δt_j	E_j^p	σ_j^p	$E_j^{u_1}$	$\sigma_j^{u_1}$	$E_j^{u_2}$	$\sigma_j^{u_2}$	$E_j^{u_3}$	$\sigma_j^{u_3}$	$E_j^{\nabla \cdot \mathbf{u}}$	$\sigma_j^{\nabla \cdot \mathbf{u}}$
1/40	1/50	1.08e-3	—	1.21e-3	—	9.66e-4	—	8.69e-4	—	8.68e-3	—
1/80	1/100	6.22e-5	4.12	6.83e-5	4.15	5.12e-5	4.24	4.42e-5	4.30	4.53e-4	4.26
1/120	1/150	1.25e-5	3.96	1.27e-5	4.15	1.06e-5	3.88	9.12e-6	3.89	9.85e-5	3.76
rate		4.1		4.1		4.1		4.2		4.1	

Table 7: Maximum-norm errors of solutions computed using the IMEX44 scheme with BWENO upwinding.

flow given by an eigenfunction of the axial component of the momentum equations, i.e.

$$u_1(r, t) = \frac{G}{2\nu} (R^2 - r^2) + \alpha_0 J_0(\lambda_0 r) e^{-\lambda_0^2 \nu t},$$

where α_0 is an arbitrary constant and $\lambda_0 R$ is a root of the Bessel function, J_0 . The radial component of the velocity is assumed to be zero, while the circumferential component, $u_\theta(r, t)$, is taken to be

$$u_\theta(r, t) = \omega r + \alpha_1 J_1(\lambda_1 r) e^{-\lambda_1^2 \nu t},$$

where ω is the angular velocity of the pipe wall, α_1 is an arbitrary constant, and $\lambda_1 R$ is a root of J_1 . The fluid pressure (which is defined up to an arbitrary constant) is given by

$$p(r, t) = \int_0^r \rho [u_\theta(\bar{r}, t)]^2 \frac{d\bar{r}}{\bar{r}}.$$

For the grid convergence tests, we use $L = 0.5$, $R = 1$, $G = 0.1/L$, $\omega = 1$, $\nu = 0.1$ and $(\alpha_0, \alpha_1) = (1, 1)$. The smallest positive roots of J_0 and J_1 are used, and these give $(\lambda_0, \lambda_1) \approx (2.4048, 3.8317)$. Numerical solutions for this problem are computed using the composite grid $\mathcal{G}_{\text{pipe}}^{(j)}$, with resolution index j , consisting of a boundary-fitted cylindrical grid overlapping a central Cartesian grid as shown in Figure 9. For both component grids, the grid spacing is chosen to be approximately $h_j = 1/(10j)$. The initial conditions are taken from the exact solution. No-slip boundary conditions (with a specified angular velocity) are specified on the pipe boundary at $r = R$ and periodic boundary conditions are taken at $x_1 = 0$ and L . Figure 9 shows the max-norm errors at time $t_{\text{final}} = 0.1$ for the second-order accurate IMEX22 scheme along with the fourth-order accurate PC44 and IMEX44 schemes. We observe that the errors in velocity and pressure for the second-order accurate scheme are converging at rates close to two, while the errors for the two fourth-order schemes, which are similar in magnitude, are converging at rates close to four.

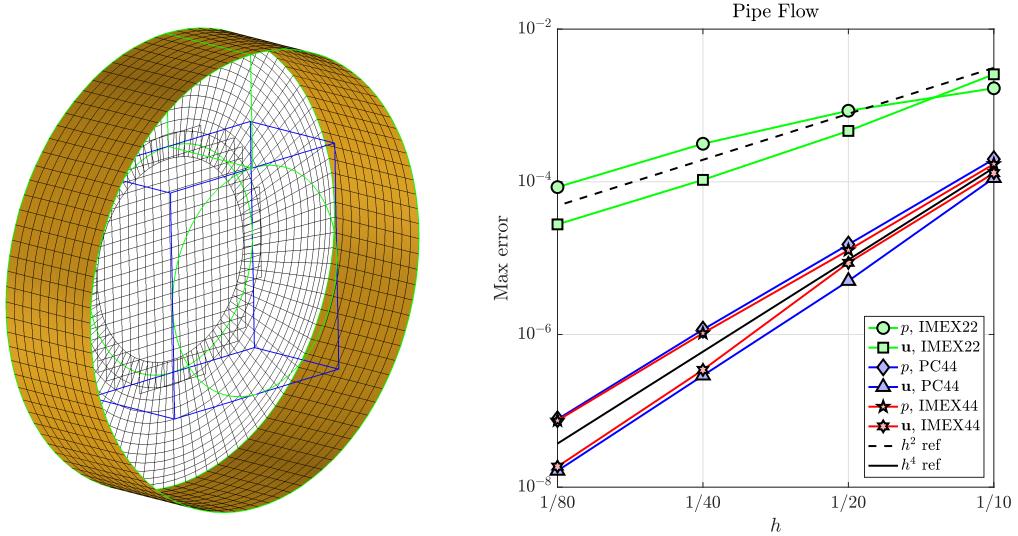


Figure 9: Pipe flow with swirl. Left: composite grid for flow in a short pipe. Right: Maximum-norm errors versus grid spacing.

5.3. Rotating disk in a box

In this section we consider the simulation of a two-dimensional disk in a square fluid domain where the disk moves along a circular path as shown in Figure 10. This example serves as a clean benchmark problem for testing the new schemes for a domain with a moving grid. A self-convergence grid-refinement study is performed to estimate the errors and convergence rates.

The composite grid for the geometry, denoted by $\mathcal{G}_d^{(j)}$, has a target grid spacing of $\Delta s^{(j)} = 1/(10j)$, and consists of a Cartesian grid for the square $[-1.5, 1.5]^2$ together with an annular grid of inner radius 0.5 and outer radius 0.85. Initially the disk is centered at $\mathbf{c} = (0.5, 0)$ as shown in the upper left plot of Figure 10. The lower left plot shows a zoomed view of the overlap region for the relatively coarse grid $\mathcal{G}_d^{(4)}$. Note that this grid, with interpolation points marked, is suitable for second-order accurate simulations. Fourth-order accurate schemes require a second layer of interpolation points (not shown in the figure) for both the (blue) Cartesian background grid and the (green) boundary-fitted annular grid.

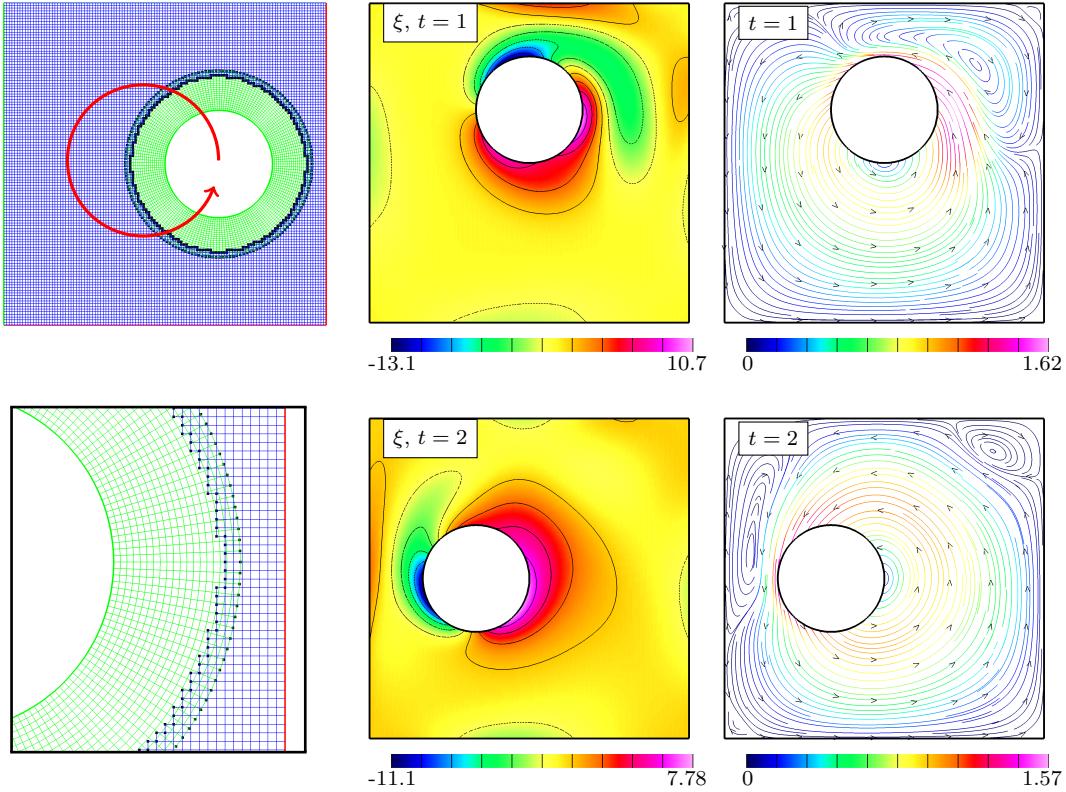


Figure 10: Rotating cylinder: Left column: Composite grid $\mathcal{G}_d^{(4)}$ and zoom. Middle and right columns: Contours of voricity and streamlines at times $t = 1$ and $t = 2$.

At each time-step the annular grid moves according to a specified motion. The motion of the disk is defined in a smooth manner so that the solution for the fluid is smooth in space and time; this allows us to verify fourth-order accuracy from the fourth-order accurate schemes. The disk translates and rotates as it moves along a circular path of radius one centered at the origin. In particular, the position over time of an arbitrary point $\mathbf{x}(t)$ on the annular grid is defined by

$$\mathbf{x}(t) = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) \\ -\sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix} \mathbf{x}(0),$$

$$\theta(t) \stackrel{\text{def}}{=} \omega_0 t \mathcal{R}(t),$$

where $\omega_0 = \pi/2$ and $\mathcal{R}(t)$ is a function that smoothly ramps from zero to one over the time interval $[0, 1]$. For $t \in [0, 1]$, this quintic ramp function is chosen to be the unique polynomial of degree eleven that has an odd symmetry about $t = 1/2$ and has five continuous derivatives at $t = 0$ and $t = 1$.

The flow is assumed to be at rest initially, and the boundary conditions on the perimeter of the box and on disk are taken to be no-slip walls. The kinematic viscosity is $\nu = 0.05$. The time-step for the semi-implicit schemes is chosen to be the minimum between Δx and the usual computation of Δt based on stability, as described in Section 4.4.2. This is done to prevent large a time-step at early times when the velocity is close

to zero. (We recall that the viscous terms provide no time-step restriction for the semi-implicit schemes.) The explicit schemes use the computed time-step based on stability and this Δt becomes quite small on the fine grids due to the contribution from the viscous terms on the time-step restriction. The semi-implicit schemes, which can take a larger time-step on the fine grids, are generally much more efficient than the explicit schemes when the viscous terms dictate a small Δt .

The plots in the middle and right columns of Figure 10 show contours of the vorticity, $\xi = \partial_x u_2 - \partial_y u_1$, and instantaneous streamlines of the solution at times $t = 1$ and $t = 2$ when the disk has made a quarter and half revolution, respectively. Vorticity is generated primarily at the disk surface although there is some vorticity generated at the walls of the square as the disk passes nearby. The streamlines indicate the main part of the flow is primarily undergoing a counterclockwise rotation, but with clockwise rotating flow in regions near some of the corners.

Figure 11 shows estimated max-norm errors of the velocity for five different schemes using results for composite grids $\mathcal{G}_d^{(j)}$, $j = 4, 8$ and 16 . The explicit PC22 scheme and the semi-implicit IMEX22 scheme are shown to converge at very close to second-order, while the explicit scheme PC44 and the semi-implicit IMEX44 are shown to converge at approximately fourth-order. The IMEX24 scheme, which uses the second-order accurate time-stepping from the IMEX22 scheme together with fourth-order accurate approximations in space, appears to be converging at close to second-order. The solutions obtained using the IMEX24 scheme have smaller errors than the ones obtained using IMEX22 and PC22 schemes, but the errors are larger than those in the solutions computed using the fourth-order accurate IMEX44 and PC44 schemes. This convergence behavior is consistent with that observed previously for the circular Couette flow problem (Section 5.1.3), indicating that a fourth-order accurate time-stepping scheme is needed for full fourth-order accuracy.

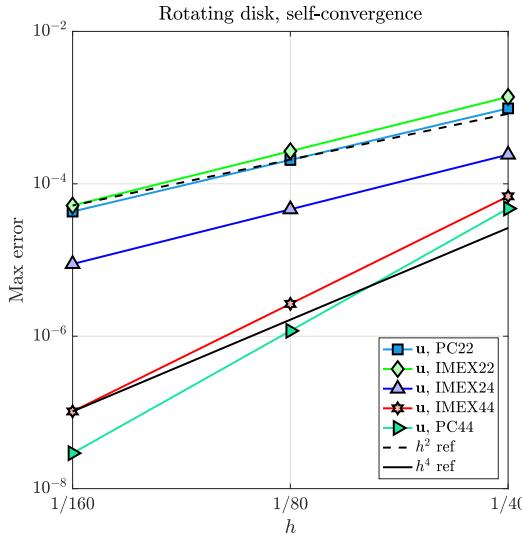


Figure 11: Rotating disk: Estimated max-norm errors of the velocity for five different schemes from a self-convergence grid refinement study at $t = 1.0$ with $\nu = 0.05$.

5.4. Flow past two cylinders in a channel

The benefit of using high-order accurate approximations for wave propagation problems has been well established as requiring fewer points-per-wavelength to achieve a given relative error in the solution [36, 37], and here we investigate the issue for under-resolved simulations of advection-dominated incompressible flow. We consider the flow past two cylinders in a two-dimensional channel as shown in Figure 12. Simulations are performed at a high Reynolds number on grids of varying resolutions to demonstrate the behavior of the new schemes for under-resolved computations. We compare results from the second-order accurate IMEX22 scheme to those from the fourth-order accurate IMEX44 scheme to assess the ability of the schemes to propagate vortices downstream with minimal non-physical dissipation. We pose the question of whether

there is any benefit in using a high-order accurate scheme for a high-Reynolds flow when the simulation is under-resolved. A similar question for turbulent flow in a three-dimensional box was discussed in [38].

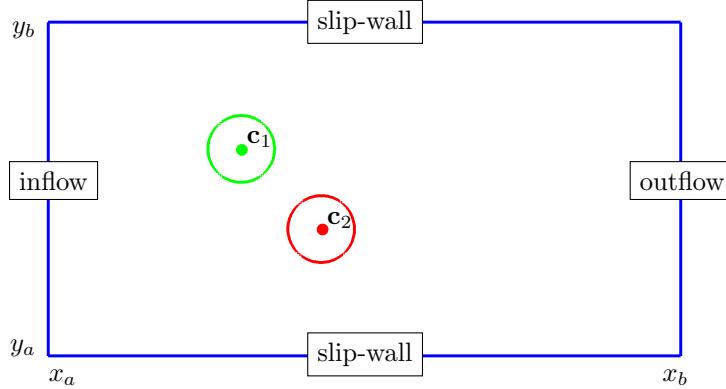


Figure 12: Geometry for flow past two cylinders.

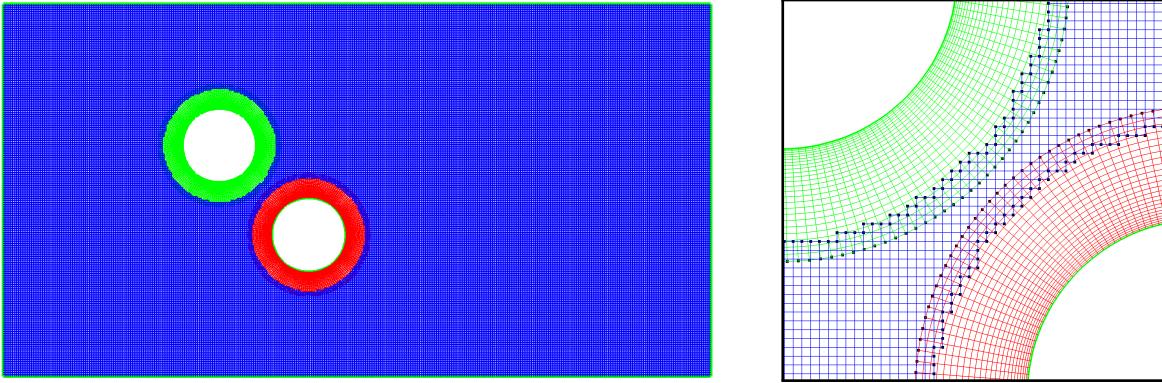


Figure 13: Composite grid $\mathcal{G}_{tc}^{(4)}$ for two cylinders in a channel. Left: full grid. Right: zoom near the cylinders showing the interpolation boundary.

The computational domain for the problem, as shown in Figure 12, consists of two circular disks embedded in a rectangular channel of dimensions $[x_a, x_b] \times [y_a, y_b] = [-3.5, 6] \times [-2.5 \times 2.5]$. The disks have radii $R_1 = R_2 = 0.5$ and centers at $\mathbf{c}_1 = (-0.6, 0.6)$ and $\mathbf{c}_2 = (0.6, -0.6)$. The composite grid for the geometry, denoted by $\mathcal{G}_{tc}^{(j)}$, consists of a background Cartesian grid and two annular grids as shown in Figure 13 with each annular grid having a radial thickness equal to 0.3. The grids have a background grid spacing of $\Delta s^{(j)} = 1/(10j)$, but the grid spacing on the annular grids is stretched in the radial direction by a factor of 5 next to the boundary of the disk to better resolve the boundary layer. The boundary conditions are taken as no-slip walls on the two disks and slip-walls (26c) on the top and bottom of the channel. The inflow boundary conditions are taken as zero tangential velocity and a specified normal component of the velocity that ramps from zero to one over $t = [0, 1]$ using the cubic ramp function given in [39]. The outflow condition (26d), with $\alpha = 1$ and $\beta = 0.1$, is given on the right side of the domain. The kinematic viscosity is taken as $\nu = 10^{-4}$, which gives a Reynolds number of $Re = 10^4$ based on unit scales for length and velocity. The time-step is chosen to be the minimum of $\Delta x/5$ and the usual computation of Δt with a CFL parameter of $C_{CFL} = 0.9$, as described in Section 4.4.2. This is done, as before, to limit the size of the time-step at early times.

Figure 14 shows the evolution of the pressure p , speed $|\mathbf{u}| = \sqrt{u_1^2 + u_2^2}$, and vorticity $\xi = \partial_x u_2 - \partial_y u_1$, at times $t = 3, 5$ and 7.5 for the fine grid $\mathcal{G}_{tc}^{(16)}$. The cylinders start to shed positive and negative vorticity at early times from the boundary layers around the two disks, and this shedding is not symmetric due to the offset of the two cylinders. Roughly speaking, strong vortex sheets are seen to originate from the top and

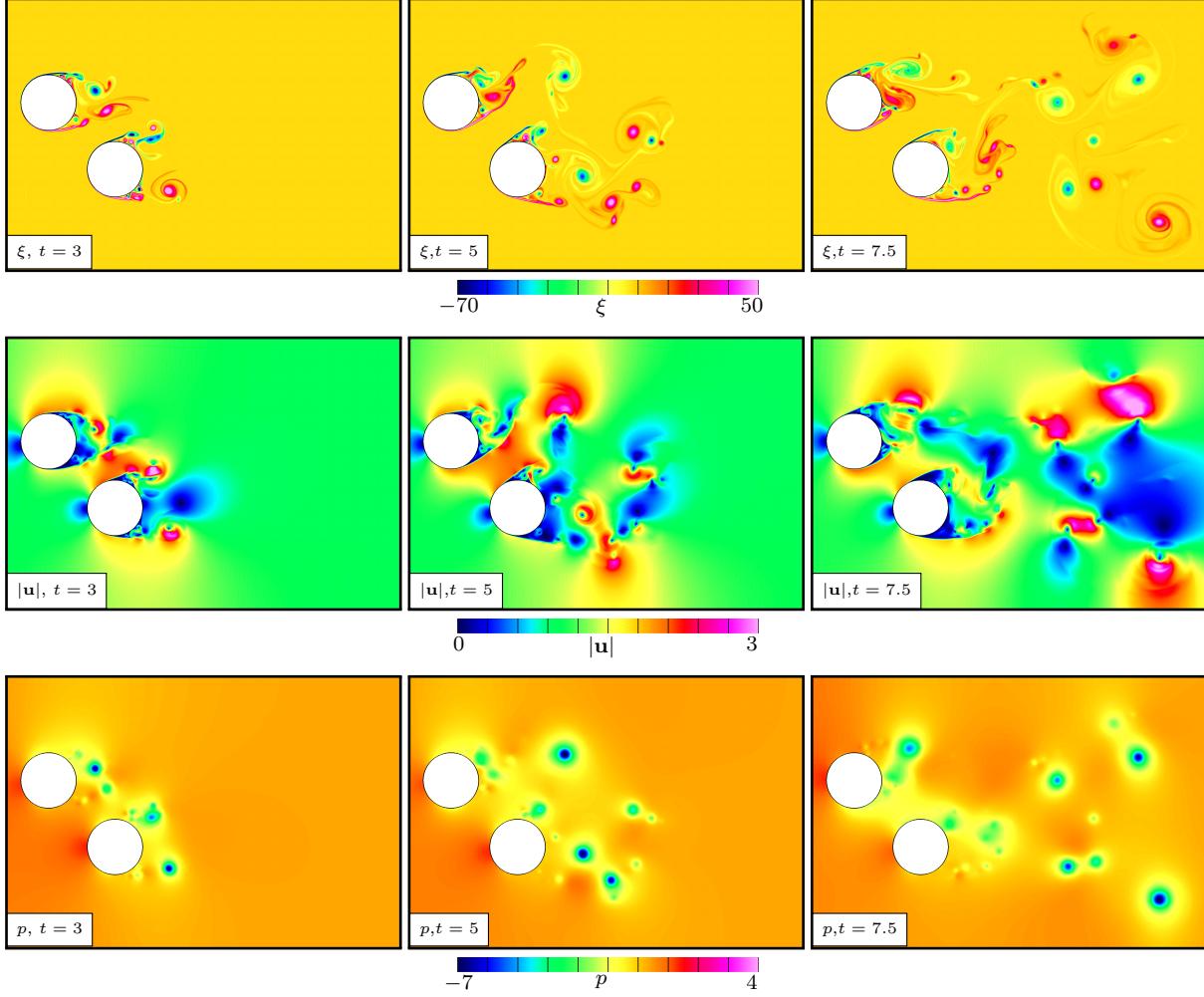


Figure 14: Contour plots of the pressure p , speed $|\mathbf{u}|$, and vorticity ξ for flow past two disks in a channel at times $t = 3, 5$ and 7.5 computed on grid $\mathcal{G}_{tc}^{(16)}$.

bottom of each cylinder and these break up into smaller vortices. Concentrated vortices are then advected down the channel. In some cases, a vortex pair of positive and negative vorticity forms (see upper right region of plots at $t = 7.5$) and this pair advects together for some time. Note that the vorticity is a derived quantity and not directly computed. The velocity, on the other hand, is computed directly, and thus the contours of flow speed, $|\mathbf{u}|$, give an indication of flow features that must be approximated by the numerical scheme. The contours of flow speed show sharp shear layers and small features near the cores of the vortices. The pressure contours show that the pressure is generally much smoother than the velocity and vorticity (roughly the pressure is one derivative smoother than the velocity and two derivatives smoother than the vorticity). The strong vortices are clearly seen in the pressure as small disks of low pressure.

The degree to which the numerical computation is under-resolved can be estimated using the smallest scale of the flow,

$$\lambda_{\min} = \sqrt{\frac{\nu}{|\nabla \mathbf{u}|}}, \quad (29)$$

which depends of the size of the local gradient of the velocity, $|\nabla \mathbf{u}|$, and which measures the length associated with the sharpest feature in the flow, such as the width of the sharpest shear layer, that is allowed before viscous smoothing becomes dominant [40–43]. To be fully resolved, the local mesh spacing, Δs , of a numerical simulation should roughly satisfy $\Delta s < \lambda_{\min}$, since the numerical solution should be able to represent the

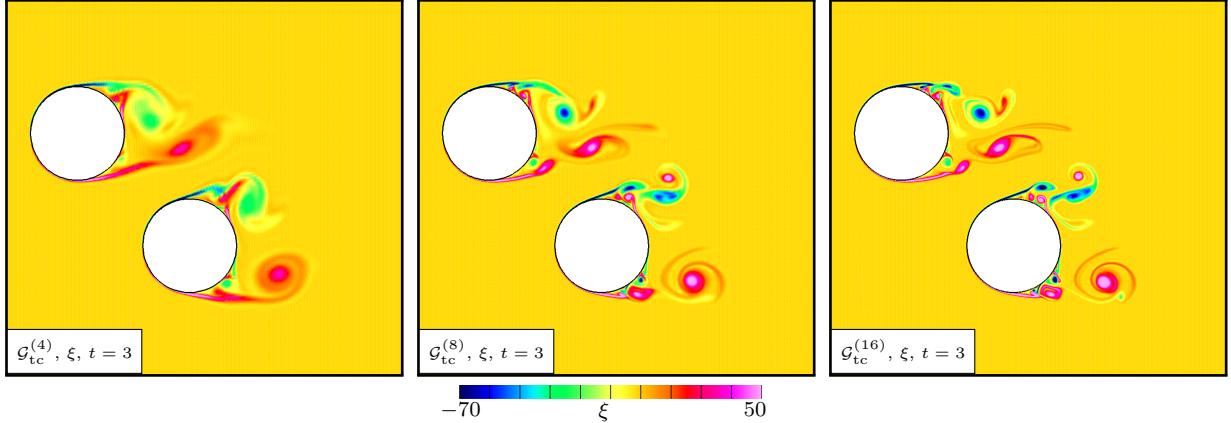


Figure 15: Contour plots of the vorticity ξ at $t = 3$ computed with the IMEX44-BWENO scheme on grids $\mathcal{G}_{tc}^{(4)}$, $\mathcal{G}_{tc}^{(8)}$ and $\mathcal{G}_{tc}^{(16)}$ showing some measure of grid convergence for the solution at time $t = 3$

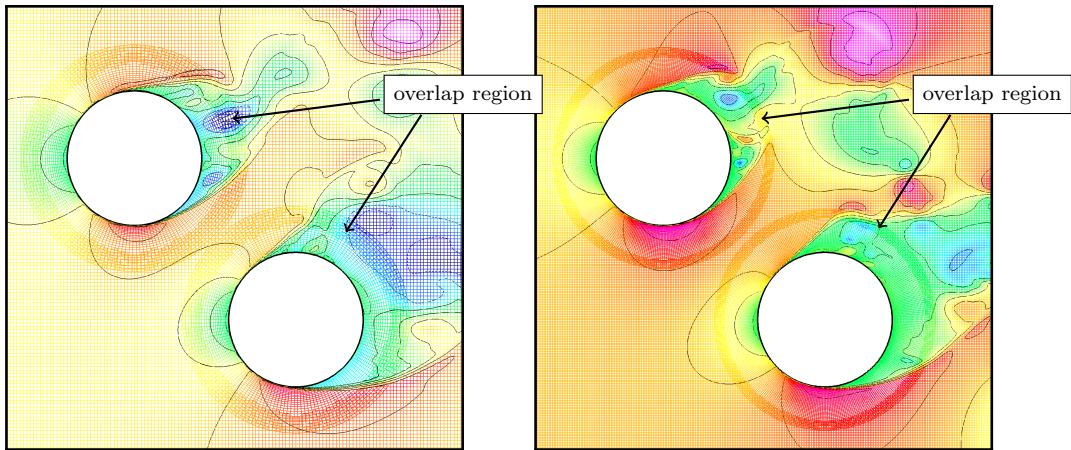


Figure 16: Wireframe contours of u_1 on the grids $\mathcal{G}_{tc}^{(4)}$ and $\mathcal{G}_{tc}^{(8)}$ showing the smooth transition of contour lines across the interpolation boundary even for under-resolved simulations.

smallest feature. The smallest scale can be estimated from the magnitude of the vorticity, $|\xi| \approx |\nabla \mathbf{u}|$. For the simulations under consideration in this section with $\nu = 10^{-4}$, the vorticity near the cylinder wall is roughly $|\xi| \approx 10^3$, which gives $\lambda_{\min} \approx \sqrt{10^{-7}} \approx 3.2 \times 10^{-4}$. The normal mesh spacing at the wall is approximately $\Delta n^{(j)} = 1/(50j)$ and solving for j from $\Delta n^{(j)} = \lambda_{\min}$ suggests that grid $\mathcal{G}_{tc}^{(j)}$ with $j = 64$ would be needed to fully resolve the flow. In Figure 15 we assess the grid convergence by examining the vorticity at time $t = 3$ on grids $\mathcal{G}_{tc}^{(4)}$, $\mathcal{G}_{tc}^{(8)}$ and $\mathcal{G}_{tc}^{(16)}$. The solution on grid $\mathcal{G}_{tc}^{(4)}$ is clearly under-resolved with the vorticity being smeared and dissipated in the near wake. Comparing the results for $\mathcal{G}_{tc}^{(8)}$ and $\mathcal{G}_{tc}^{(16)}$ shows that while the solutions are not yet fully resolved (as estimated using (29)) the solutions are in quite good agreement in terms of the shedding of the vorticity near the cylinder walls and the position, structure and strength of the vortices and shear-layers downstream of the cylinders. An independent result (not shown) from the IMEX22 scheme on a finer grid $\mathcal{G}_{tc}^{(32)}$ is also in good agreement with the IMEX44 scheme on $\mathcal{G}_{tc}^{(16)}$. These results provide an indication that the numerical solutions are converging as the mesh is refined. Note that grid convergence at longer times would require even finer mesh spacings due to the sensitivity of the solutions to small perturbations at earlier times.

As a further evaluation of the fourth-order accurate IMEX44-BWENO scheme for under-resolved flows, including the effects of overset grid boundaries, we consider the behavior of the flow velocity near the interpolation boundaries between the annular boundary-fitted grids and the background Cartesian grid. Figure 16 shows, for example, contours of the horizontal component of velocity, u_1 , in the vicinity of the two

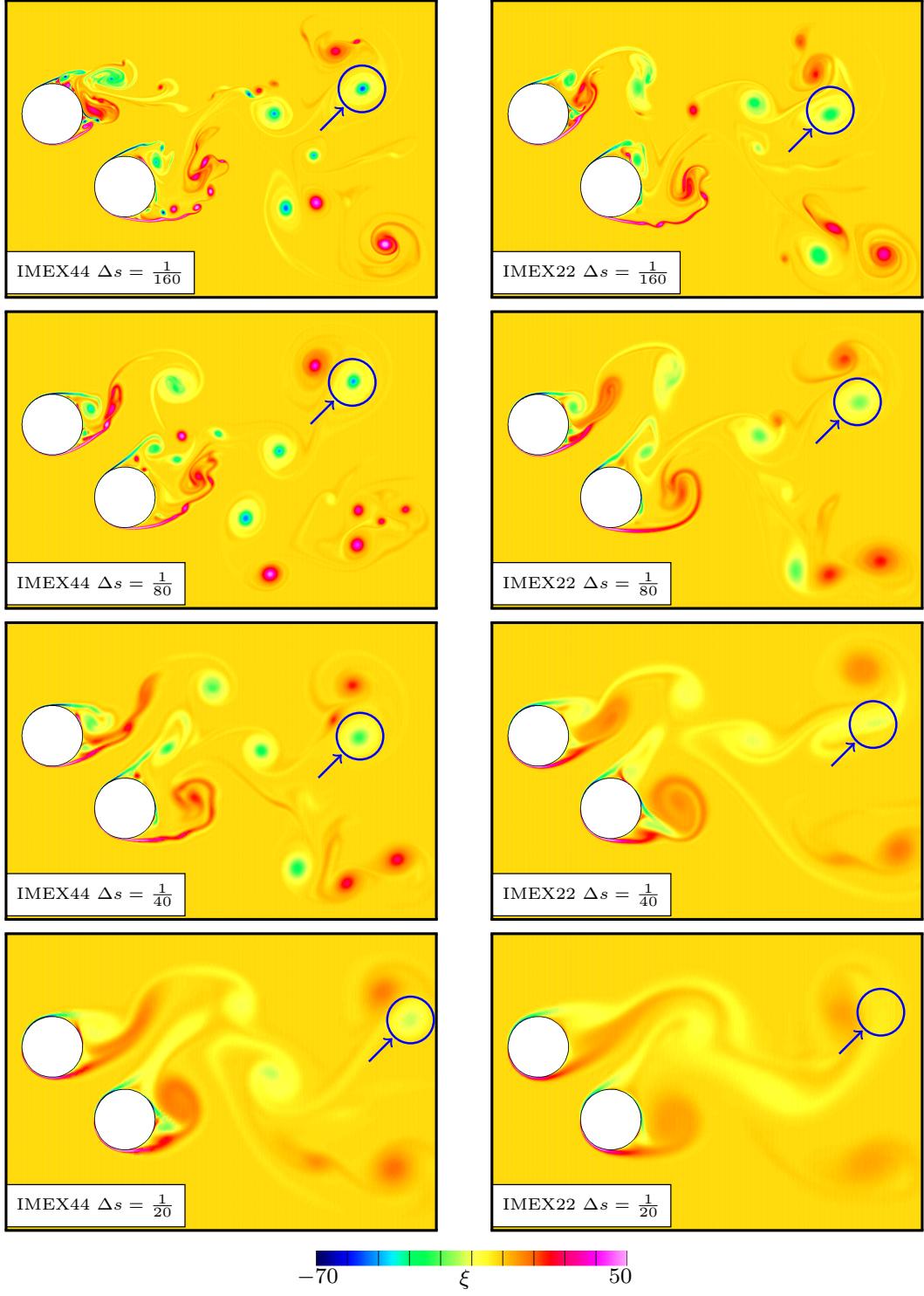


Figure 17: Flow past two cylinders with $\nu = 10^{-4}$ showing contour plots of the vorticity, ξ , at $t = 7.5$ on grids of different resolutions. Left column: fourth-order accurate IMEX44 scheme with BEWNO upwind dissipation. Right column: second-order accurate IMEX22 scheme with second-order dissipation. A representative vortex in the wake is circled for comparison. The IMEX44-BWENO scheme is seen to provide greater effective resolution for a given grid spacing compared to the IMEX22 scheme.

cylinders for computations on the medium coarse grids $\mathcal{G}_{tc}^{(4)}$ and $\mathcal{G}_{tc}^{(8)}$. Even for these highly under-resolved simulations, the contour lines are shown to smoothly cross the interpolation boundaries and show only small effects when crossing from the annular grid to the background grid, despite the presence of rather sharp shear layers. This example demonstrates the benefit of the BWENO dissipation to adaptively add some, but not too much, smoothing to the numerical solution.

We now compare the behaviours of the IMEX44-BWENO scheme, and the IMEX22 scheme with artificial dissipation, for an under-resolved flow at time $t = 7.5$ as shown in Figure 17. By this later time there are many small vortices that have been shed and advected downstream of the cylinders. The exact location of these vortices is quite sensitive to small perturbations at earlier times, and we observe large differences between the solutions even on the two finest grids shown. We now ask whether there is any benefit of using a high-order scheme for such under-resolved flows. Rather than comparing the exact location of the downstream vortices, an alternate measure of solution quality is the strength of the downstream vortices. This is an important consideration in applications such as wind turbines and rotor-craft where these vortices can interact with downstream structures. Focusing on the strength of downstream vortices in Figure 17, (a selected vortex is circled to facilitate the comparison), we compare the IMEX22 solution on the finest grid $\mathcal{G}_{tc}^{(16)}$ (with $\Delta s = 1/160$) to the IMEX44-BWENO solutions on coarser grids. We note that the representation of the circled vortex for the IMEX22 scheme on grid $\mathcal{G}_{tc}^{(16)}$ is roughly in agreement with the IMEX44-BWENO solution on grid $\mathcal{G}_{tc}^{(4)}$, which has a grid spacing, $\Delta s = 1/40$, which is four times larger. While these two simulations provide a similar solution quality in the wake, the fourth-order BWENO calculation on the coarser grid requires a CPU time that is 11 times smaller than the second-order calculation. Thus, there appears to be a benefit to using the higher-order accurate scheme under this measure of quality. Furthermore, this benefit should only increase as the vortices advect further downstream since it is well known that the advantages of high-order schemes increase as features propagate for longer distances.

6. Conclusions

We have described two efficient and fourth-order accurate fractional-step schemes for solving the incompressible Navier-Stokes (INS) equations in the *velocity-pressure* formulation where the velocity is updated in a separate stage from the pressure. The momentum equations are integrated using linear-multistep methods methods in a predictor-corrector manner. The fourth-order accurate explicit time-stepping method PC[AB3,AM4] uses a third-order accurate Adams-Basforth predictor and a fourth-order accurate Adams-Moulton corrector. The fourth-order accurate semi-implicit scheme PC[IMEX3m,BDF4] uses a third-order accurate BDF-based IMEX predictor and a partially implicit BDF corrector. The predictor and corrector both treat the viscous terms implicitly and the remaining terms explicitly. The implicit method is especially useful when the viscous terms are stiff (e.g. when using highly stretched grids in a boundary layer) since the stable time-step is not constrained by the viscous terms. Versions of the schemes that can be used with variable time-step are given, together with modified IMEX versions that avoid reforming the implicit time-stepping matrix more than once when the time-step changes from one constant value to another.

The equations are discretized in space using fourth-order accurate finite-difference approximations. Accurate boundary conditions based on compatibility conditions are used. For cases in which the boundary conditions couple the velocity and pressure, such as at a no-slip wall, we have shown how to implement these conditions in a segregated manner so that the fractional-step character of the method is maintained and the scheme remains stable. Special treatment of the boundary conditions are given that simplify the implementation of the implicit velocity solves in the IMEX schemes.

To improve robustness for under-resolved simulations, when the natural dissipation from the viscous terms is not sufficient, a WENO-type upwind dissipation is developed for discretizing the advection terms in the momentum equations. Unlike a traditional WENO approach, this new BWENO approach fits with the same five-point stencil as the centered scheme, provides upwind dissipation for under-resolved simulations, and retains fourth-order accuracy for well-resolved simulations. The new BWENO scheme also incorporates modifications to handle flows near stagnation points.

The explicit-PC and IMEX-PC time-stepping schemes are fourth-order accurate in both time and space, and they are implemented for complex moving geometry in two and three space dimensions using composite

overlapping grids⁷. The schemes are verified to be stable and fourth-order accurate using manufactured solutions and known solutions for static and moving geometry. A self-convergence grid refinement study was performed for a rotating disk in a box to provide a clean benchmark for a moving grid problem. The IMEX-PC method was also tested for under-resolved simulations of a high-Reynolds number flow past two disks to demonstrate the usefulness of the high-order scheme and the BWENO upwind dissipation.

References

- [1] G. S. Chesshire, W. D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, *J. Comput. Phys.* 90 (1) (1990) 1–64.
- [2] A. J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* 2 (1967) 12–26.
- [3] W. D. Henshaw, A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids, *J. Comput. Phys.* 113 (1) (1994) 13–25.
- [4] W. D. Henshaw, H.-O. Kreiss, L. G. M. Reyna, A fourth-order accurate difference approximation for the incompressible Navier-Stokes equations, *Comput. Fluids* 23 (4) (1994) 575–593.
- [5] W. D. Henshaw, N. A. Petersson, A split-step scheme for the incompressible Navier-Stokes equations, in: M. M. Hafez (Ed.), *Numerical Simulation of Incompressible Flows*, World Scientific, 2003, pp. 108–125.
- [6] J. W. Banks, J. A. F. Hittinger, A new class of non-linear, finite-volume methods for Vlasov simulation, *IEEE T. Plasma. Sci.* 38 (9) (2010) 2198–2207.
- [7] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1) (1996) 202–228.
- [8] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, *Tech. Rep. NASA-CR-97-206253*, NASA Langley Research Center (Nov. 1997).
- [9] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: A. Quarteroni (Ed.), *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Vol. 1697 of *Lecture Notes in Mathematics*, Springer, New York, 1998, pp. 325–432.
- [10] H. Johnston, J.-G. Liu, Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term, *J. Comput. Phys.* 199 (1) (2004) 221–259.
- [11] J. G. Liu, J. Liu, R. L. Pego, Stable and accurate pressure approximation for unsteady incompressible viscous flow, *Journal of Computational Physics* 229 (9) (2010) 3428 – 3453.
- [12] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comput. Phys.* 59 (1985) 308–323.
- [13] O. Colomés, S. Badia, Segregated Runge–Kutta methods for the incompressible Navier–Stokes equations, *International Journal for Numerical Methods in Engineering* 105 (5) (2016) 372–400.
- [14] J. W. Banks, W. D. Henshaw, D. W. Schwendeman, Q. Tang, A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part II: General formulation, *J. Comput. Phys.* 343 (2017) 469–500.
- [15] J. W. Banks, W. D. Henshaw, D. W. Schwendeman, Q. Tang, A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part I: Model problem analysis, *J. Comput. Phys.* 343 (2017) 432–468.

⁷Using the Overture framework overtureFramework.org.

- [16] A. Koblitz, S. Lovett, N. Nikiforakis, W. D. Henshaw, Direct numerical simulation of particulate flows with an overset grid method, *J. Comput. Phys.* 343 (2017) 414–431.
- [17] J. W. Banks, W. D. Henshaw, D. W. Schwendeman, Q. Tang, A stable partitioned FSI algorithm for rigid bodies and incompressible flow in three dimensions, *J. Comput. Phys.* 373 (2018) 455–492.
- [18] J. W. Banks, W. D. Henshaw, D. W. Schwendeman, An analysis of a new stable partitioned algorithm for FSI problems. Part II: Incompressible flow and structural shells, *J. Comput. Phys.* 268 (2014) 399–416.
- [19] L. Li, W. D. Henshaw, J. W. Banks, D. W. Schwendeman, G. A. Main, A stable partitioned FSI algorithm for incompressible flow and deforming beams, *J. Comput. Phys.* 312 (2016) 272–306.
- [20] J. W. Banks, W. D. Henshaw, D. W. Schwendeman, An analysis of a new stable partitioned algorithm for FSI problems. Part I: Incompressible flow and elastic solids, *J. Comput. Phys.* 269 (2014) 108–137.
- [21] D. A. Serino, J. W. Banks, W. D. Henshaw, D. W. Schwendeman, A stable added-mass partitioned (AMP) algorithm for elastic solids and incompressible flow, preprint arXiv:1812.05208, submitted for publication (2019).
- [22] D. A. Serino, J. W. Banks, W. D. Henshaw, D. W. Schwendeman, A stable added-mass partitioned (AMP) algorithm for elastic solids and incompressible flow: Model problem analysis, preprint arXiv:1812.03192, submitted for publication (2019).
- [23] F. C. Dougherty, J.-H. Kuan, Transonic store separation using a three-dimensional Chimera grid scheme, paper 89-0637, AIAA (1989).
- [24] R. Meakin, Moving body overset grid methods for complete aircraft tiltrotor simulations, paper 93-3350, AIAA (1993).
- [25] W. D. Henshaw, D. W. Schwendeman, Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow, *J. Comput. Phys.* 216 (2) (2006) 744–779.
- [26] W. M. Chan, Overset grid technology development at NASA Ames Research Center, *Comput. Fl.* 38 (3) (2009) 496–503.
- [27] N. A. Petersson, Stability of pressure boundary conditions for Stokes and Navier-Stokes equations, *J. Comput. Phys.* 172 (1) (2001) 40–70.
- [28] J. Angel, J. W. Banks, W. D. Henshaw, High-order upwind schemes for the wave equation on overlapping grids: Maxwell's equations in second-order form, *J. Comput. Phys.* 352 (2018) 534–567.
- [29] J. W. Banks, A. G. Odu, R. L. Berger, T. Chapman, W. T. Arrighi, S. Brunner, High-order accurate conservative finite difference methods for Vlasov equations in 2D+2V, *SIAM J. Sci. Comput.* (submitted).
- [30] A. K. Henrick, T. D. Aslam, J. M. Powers, Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points, *J. Comput. Phys.* 207 (2005) 542–567.
- [31] W. D. Henshaw, On multigrid for overlapping grids, *SIAM J. Sci. Comput.* 26 (5) (2005) 1547–1572.
- [32] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, The portable extensible toolkit for scientific computation, *Tech. Rep. <http://www.mcs.anl.gov/petsc/petsc.html>*, Argonne National Laboratory (1999).
- [33] W. D. Henshaw, Cgins user guide: An Overture solver for the incompressible Navier-Stokes equations on composite overlapping grids, Software Manual LLNL-SM-455851, Lawrence Livermore National Laboratory (2010).
- [34] W. D. Henshaw, Cgins reference manual: An Overture solver for the incompressible Navier-Stokes equations on composite overlapping grids, Software Manual LLNL-SM-455871, Lawrence Livermore National Laboratory (2010).

- [35] G. I. Taylor, A. E. Green, Mechanism of the production of small eddies from large ones, *Proc. R. Soc. Lond. A* 158 (1937) 499–521.
- [36] H.-O. Kreiss, J. Oliger, Comparison of accurate methods for the integration of hyperbolic equations, *Tellus* 24 (1972) 199–215.
- [37] B. Gustafsson, H.-O. Kreiss, J. Oliger, *Time Dependent Problems and Difference Methods*, John Wiley and Sons Inc., 1995.
- [38] N. Fehn, W. A. Wall, M. Kronbichler, Efficiency of high-performance discontinuous Galerkin spectral element methods for under-resolved turbulent incompressible flows, *International Journal for Numerical Methods in Fluids* 88 (1) (2018) 32–54.
- [39] D. A. Serino, J. W. Banks, W. D. Henshaw, D. W. Schwendeman, A stable added-mass partitioned (AMP) algorithm for elastic solids and incompressible flow, *J. Comput. Phys.* 399 (2019) 1–30.
- [40] W. D. Henshaw, H.-O. Kreiss, L. G. M. Reyna, On the smallest scale for the incompressible Navier-Stokes equations, *Theoretical and Computational Fluid Dynamics* 1 (1989) 65–95.
- [41] W. D. Henshaw, H.-O. Kreiss, L. G. M. Reyna, Smallest scale estimates for the incompressible Navier-Stokes equations, *Arch. Rational Mech. Anal.* 112 (1990) 21–44.
- [42] W. D. Henshaw, H.-O. Kreiss, L. G. M. Reyna, Estimates of the local minimum scale for the incompressible Navier-Stokes equations, *Numerical Functional Analysis and Optimization* 16(3&4) (1995) 315–344.
- [43] W. D. Henshaw, H.-O. Kreiss, J. Yström, Numerical experiments on the interaction between the large- and small-scale motions of the Navier-Stokes equations, *SIAM Journal of Multiscale Modeling and Simulation* 1 (1) (2003) 119–149.