



A Hierarchical Space-Time Spectral Element and Moment-of-Fluid Method for Improved Capturing of Vortical Structures in Incompressible Multi-phase/Multi-material Flows

Chaoxu Pei¹ · Mehdi Vahab² · Mark Sussman¹ · M. Yousuff Hussaini¹

Received: 15 April 2019 / Revised: 15 October 2019 / Accepted: 28 October 2019 /

Published online: 9 November 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

A novel block structured adaptive space-time spectral element and moment-of-fluid method is described for computing solutions to incompressible multi-phase/multi-material flows. The new method implements a space-time spectrally accurate method in the bulk regions of a multi-phase/multi-material flow and implements the cell integrated semi-Lagrangian moment-of-fluid method in the vicinity of mixed material computational cells. In the new method, the space-time order can be prescribed to be $2 \leq p_\ell^{(x)} \leq 16$ (space) and $2 \leq p_\ell^{(t)} \leq 16$ (time) respectively. ℓ represents the adaptive mesh refinement level. Regardless of the space-time order, only one ghost layer of cells is communicated between neighboring grid patches that are on different compute nodes or different adaptive levels ℓ . The new method is first tested on incompressible vortical flow benchmark tests, then the new method is tested on the following incompressible multi-phase/multi-material problems: (i) vortex shedding past a tilted cone and (ii) atomization and spray of a liquid jet in a gas cross-flow.

Keywords Space-time · Multi-phase flow · Multi-material flow · Spectral accuracy · Adaptive mesh refinement · Scalable algorithm

Mathematics Subject Classification 65B05 · 65M70

✉ Mark Sussman
sussman@math.fsu.edu

Chaoxu Pei
chaoxup@gmail.com

Mehdi Vahab
mvahab@fsu.edu

M. Yousuff Hussaini
myh@math.fsu.edu

¹ Department of Mathematics, Florida State University, Tallahassee, FL 32306, USA

² Department of Mechanical Engineering, FAMU-FSU College of Engineering, Tallahassee, FL 32310, USA

1 Introduction

Vortical structures appear in multi-material/multi-phase flows that arise from atomization and spray phenomena [36], bubbly flows [30,37,38], or the shedding of vortices from flow past objects [10,59].

Some specialized computational techniques for improving the simulation of the dynamics of vortical structures in fluid flow include turbulence modeling techniques [30], vortex confinement approaches [60], the vorticity stream-function formulation [39], approaches that directly solve the vorticity transport equation [11], vortex particle method [63], vortex particle mesh method [12], or adaptive mesh refinement [28].

In this article a unique algorithm is presented in which the space-time spectral element method [49] is hybridized with the Cell Integrated Semi-Lagrangian Moment-Of-Fluid method (CISL-MOF) [25,35]. It is demonstrated that our algorithm, an Eulerian grid method, for the simulation of complex vortical flow problems, preserves total kinetic energy and total vorticity to within a fraction of a percent. For smooth vortical problems, the maximum vorticity error and the maximum velocity error are as low as governed by computer round-off error. For complex problems which involve the break up of vortex patches into smaller patches, or the stretching of vortex patches, the Enstrophy error improves with increasing space-time order. We also show that the Enstrophy error improves with increasing grid refinement. The behavior of the Enstrophy error versus time for the dynamics of vortical flow is reported in Figs. 18 and 20. The evolution of the maximum vorticity magnitude as a vortical structure propagates downstream is reported in Table 9.

In order to motivate our research, we give a timeline, leading up to the present, of related research in the development of space-time Eulerian grid numerical methods for predicting fluid flows. A common objective in all of these approaches, including ours, is “time accuracy,” in which the terminal behavior of important fluid dynamics quantities such as, mass, momentum, total energy, kinetic energy, vorticity, enstrophy, and helicity are captured as accurately as possible by a numerical method.

Morinishi et al. [44] developed second and fourth order finite difference algorithms for computing single material incompressible flow with constant density. Morinishi et al.’s algorithm has the property that the error in total kinetic energy is dependent only on the local truncation error of the time discretization. Liu and Wang [40] developed a second order in space, fourth order in time (standard RK4) method in order to compute solutions to the 2D and 3D Magneto Hydrodynamics equations. Liu and Wang’s method has the property that admitted solutions are both energy and helicity preserving. Kadioglu et al. [26] developed a fourth order finite volume algorithm in space and a fourth order spectral deferred correction method in time for computing single material, variable density, incompressible flow. Excellent results were obtained by Kadioglu et al for their smooth “traveling vortex problem.” Almgren et al. [2] also developed a fourth order finite volume algorithm in space and a fourth order spectral deferred correction method in time for computing single material, incompressible flow. They demonstrate that their fourth order method has an advantage over a second order method for accurately predicting the “compensated spectrum” $\kappa^{5/3} E(\kappa)$ where $E(\kappa)$ is the energy of the κ th mode determined by taking the discrete Fourier Transform of the velocity field and “computing the energy in κ space.” Dumbser et al. [19] developed a third order “one-step ADER-WENO” method for computing solutions to the compressible gas dynamics and MHD equations. Dumbser et al. implemented their method on a tree based adaptive mesh grid and tested their algorithm on shock wave interaction problems, and vortex pair interaction problems. Don et al. [14] hybridized the fifth order WENO-Z method (in

regions detected to contain a shock) with a sixth order compact scheme for simulating the interaction of shock waves with themselves or a rigid structure. The time discretization for the hybrid method was the 3rd order TVD Runge–Kutta method. Zhang [67] developed a fourth order projection method for computing solutions to the constant density and constant viscosity Navier–Stokes equations with no-slip boundary conditions at the computational domain walls. The space derivatives are discretized with fourth order finite difference methods and the time derivative is either discretized with the fourth order explicit Runge–Kutta method (if the time step constraint associated with viscosity is not “stiff”) or the fourth order Implicit-Explicit Runge Kutta method (if the explicit time step constraint associated with viscosity is too stringent). Saye [53,54] developed a numerical method for simulating multi-material problems in implicitly defined (deforming) domains in which the temporal order of accuracy was two and the space order of accuracy can be four. Saye’s work is unique in that high order of space-time accuracy was gained at implicitly defined deforming boundaries.

We are not aware of any algorithm that has the following combination of properties that our novel hybrid method has:

- the ability to simulate problems with realistic density and viscosity ratios corresponding to liquid and gas (e.g. 800:1 and 56:1 respectively),
- robust to complex deformations of material boundaries such as simulating interfaces that merge or split,
- the time order of accuracy in the “bulk” fluid regions can be selected from $2 \leq p^{(t)} \leq 16$ and the space order of accuracy can be selected from $2 \leq p_{\ell}^{(x)} \leq 16$ where ℓ is an adaptive mesh refinement level,
- one can select a lower spatial order, $p_{\ell}^{(x)}$, on higher resolution levels in order to have a balance between accuracy gained via grid refinement versus accuracy gained by increasing the order $p_{\ell}^{(x)}$,
- for parallel implementations of our space-time spectral element method, only one layer of ghost cells is required to be communicated between grids stored on different computer nodes, regardless of the space-time order (see Fig. 14).

Remarks:

- In the “spectral regions,” the pressure gradient, non-linear advective terms, and the viscosity force are discretized with the $p_{\ell}^{(x)}$ ($2 \leq p_{\ell}^{(x)} \leq 16$) order spectral element method in space [49] and the $p^{(t)}$ order ($2 \leq p^{(t)} \leq 16$) Multi-Implicit Spectral Deferred Correction method (MISDC) [9,26,34,45,47,49] in time.
- In the “Cell Integrated Semi-Lagrangian Moment-of-Fluid” (CISL-MOF) elements, the non-linear advective terms are discretized using the first order directionally split Cell Integrated Semi-Lagrangian method (CISL), the pressure gradient force term is discretized using the (formally) second order accurate finite volume method, and the viscosity force term is discretized using the (formally) second order accurate finite volume method in space and the first order accurate backward Euler method in time. We refer the reader to our previous research on the CISL-MOF and finite volume methods: [25,35,61]. Also we refer the reader to Fig. 6 which illustrates our finite volume discretization of the viscosity force terms.
- Our rationale for selecting the MISDC method [49] for coupling the advection, pressure gradient, and viscosity force terms, is that the MISDC method is a robust operator split “divide and conquer” method in contrast to a “monolithic” method. A “Monolithic” method will simultaneously integrate in time all force terms. “Monolithic” methods [21,29,48,50,51,58,64] exhibit no splitting errors, but the performance of a “monolithic”

method relies on the resulting large sparse system of equations not being ill-conditioned. For many time dependent partial differential equations, there is no guarantee that a “monolithic method” will not break down (the sparse system solver will not diverge). We claim that the MISDC time discretization scheme results in a more robust method than “monolithic” schemes, while at the same time not incurring splitting errors.

2 Governing Equations

The objective is to simulate the fluid flow of a multi-phase/multi-material system consisting of \mathcal{M} materials, each material governed by the incompressible Navier–Stokes equations for immiscible flow:

- *Material domains and interfaces:* The *material* level set function ϕ^M represents the physical domain for material M ,

$$\phi^M(\mathbf{x}, t) = \begin{cases} > 0 & \mathbf{x} \in \text{material } M, \\ \leq 0 & \text{otherwise,} \end{cases} \quad (1)$$

where \mathbf{x} is the position vector in space and t is time. The *interface* level set represents the interface between materials M_1 and M_2

$$\phi^{M_1, M_2}(\mathbf{x}, t) = \begin{cases} > 0 & \mathbf{x} \in \text{material } M_1, \\ < 0 & \mathbf{x} \in \text{material } M_2, \\ = 0 & \mathbf{x} \text{ along } (M_1, M_2) \text{ interface.} \end{cases} \quad (2)$$

The normal and curvature defined based on these level set functions are:

$$\mathbf{n}^{M_1, M_2} = \frac{\nabla \phi^{M_1, M_2}}{|\nabla \phi^{M_1, M_2}|}, \quad \kappa^{M_1, M_2} = \nabla \cdot \frac{\nabla \phi^{M_1, M_2}}{|\nabla \phi^{M_1, M_2}|}. \quad (3)$$

- *Conservation of mass:* We assume that each material is incompressible, so that the velocity field $\mathbf{u} = (u, v, w)$ is divergence free within the bulk of each material:

$$\nabla \cdot \mathbf{u} = 0. \quad (4)$$

- *Conservation of momentum:* The conservation of momentum for each material in its domain is given by

$$(\mathbf{u} \rho^M)_t + \nabla \cdot (\mathbf{u} \otimes \mathbf{u} \rho^M + p^M \mathbb{I}) = \nabla \cdot (2\mu^M \mathbb{D}) + \rho^M \mathbf{g} \quad \text{if } \phi^M(\mathbf{x}, t) > 0, \quad (5)$$

where p^M and μ^M are pressure and viscosity of material M respectively, \mathbf{g} is the gravitational acceleration vector, and $\mathbb{D} = (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)/2$ is the rate of deformation tensor.

- *Interfacial jump conditions:* The jump condition between two materials M_1 and M_2 is described by

$$[2\mu^M \mathbb{D} - p^M \mathbb{I}] \cdot \mathbf{n}^{M_1, M_2} = \sigma^{M_1, M_2} \kappa^{M_1, M_2} \mathbf{n}^{M_1, M_2}, \quad (6)$$

where σ^{M_1, M_2} is the surface tension coefficient between materials M_1 and M_2 .

- *Interface evolution:* Assume \mathbf{v} is the velocity of a material interface defined only on the interface. The front movement is governed by the level set equation,

$$(\phi^M)_t + \mathbf{v}_{\text{ext}} \cdot \nabla \phi^M = 0, \quad M = 1 \dots \mathcal{M}, \quad (7)$$

where \mathbf{v}_{ext} is the extended version of \mathbf{v} defined also away from the interface.

3 Staggered Grid Projection Method

The computational domain is tessellated by rectangular spectral elements. We denote a spectral element on level ℓ as,

$$\Omega_\ell^e = \{(x, y, z) | x_e \leq x \leq x_e + \Delta x_{\ell,e}, y_e \leq y \leq y_e + \Delta y_{\ell,e}, z_e \leq z \leq z_e + \Delta z_{\ell,e}\}$$

If the spatial order on level ℓ is $p_\ell^{(x)}$, then the Legendre Gauss–Gauss–Gauss nodes within Ω_ℓ^e are located at:

$$\begin{aligned} & (x_{e,i,j,k}^g, y_{e,i,j,k}^g, z_{e,i,j,k}^g) \quad 0 \leq i, j, k \leq p_\ell^{(x)} - 1 \\ & x_{e,i,j,k}^g = x_e + \frac{\Delta x_{\ell,e}}{2}(s_i^g + 1) \\ & y_{e,i,j,k}^g = y_e + \frac{\Delta y_{\ell,e}}{2}(s_j^g + 1) \\ & z_{e,i,j,k}^g = z_e + \frac{\Delta z_{\ell,e}}{2}(s_k^g + 1) \end{aligned}$$

s_i^g are the roots of the $p_\ell^{(x)}$ order Legendre polynomial. Throughout this article, we assume that,

$$p_\ell^{(x)} = p_\ell^{(y)} = p_\ell^{(z)}.$$

The Legendre Gauss–Lobatto–Gauss–Gauss nodes within Ω_ℓ^e are located at:

$$\begin{aligned} & (x_{e,i,j,k}^{gl}, y_{e,i,j,k}^{gl}, z_{e,i,j,k}^{gl}) \quad 0 \leq i \leq p_\ell^{(x)}, 0 \leq j, k \leq p_\ell^{(x)} - 1 \\ & x_{e,i,j,k}^{gl} = x_e + \frac{\Delta x_{\ell,e}}{2}(s_i^{gl} + 1) \\ & y_{e,i,j,k}^{gl} = y_e + \frac{\Delta y_{\ell,e}}{2}(s_j^{gl} + 1) \\ & z_{e,i,j,k}^{gl} = z_e + \frac{\Delta z_{\ell,e}}{2}(s_k^{gl} + 1) \end{aligned}$$

s_i^{gl} are the roots of the derivative of the $p_\ell^{(x)}$ order Legendre polynomial combined with the endpoints: $s_0^{gl} = -1$, $s_{p_\ell^{(x)}}^{gl} = 1$.

The Legendre Gauss–Gauss–Lobatto–Gauss nodes and Legendre Gauss–Gauss–Gauss–Lobatto nodes are defined analogously as the Legendre Gauss–Lobatto–Gauss–Gauss nodes.

The approximate pressure is discretized at the Legendre Gauss–Gauss–Gauss nodes and the velocity vector is defined on the marker-and-cell (MAC) grid. The MAC grid consists of Legendre Gauss–Lobatto–Gauss–Gauss points for the velocity component u in the x -direction, Legendre Gauss–Gauss–Lobatto–Gauss points for the velocity component v in the y -direction, and Legendre Gauss–Gauss–Gauss–Lobatto points for the velocity component w in the z -direction.

The locations of the Legendre Gauss–Gauss–Gauss points and the MAC grid points are illustrated in Fig. 1.

Referring to Fig. 6, prior to the beginning of integrating the solution of (5) and (7) in a space-time slab (see Fig. 7), we tag elements near material interfaces as “Cell Integrated Semi-Lagrangian Moment-of-Fluid” (CISL-MOF) elements and we always discretize the Navier Stokes equations for multiphase flow using the CISL-MOF approach in the CISL-MOF elements. An element is tagged as a CISL-MOF element if multiple materials (≥ 2 materials

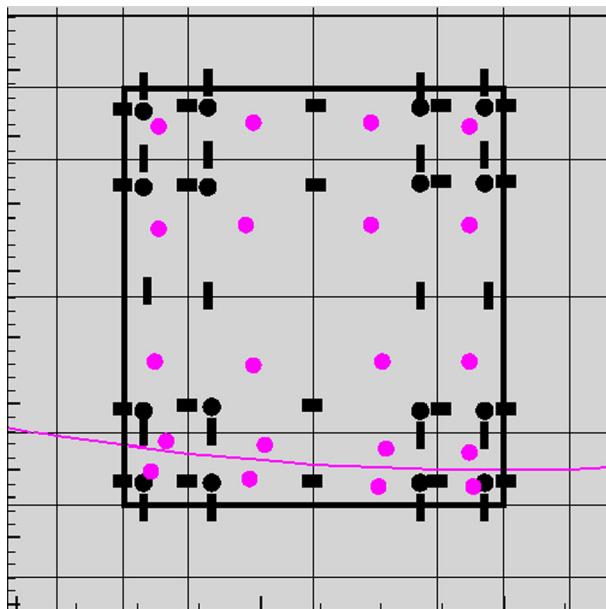


Fig. 1 An illustration of the Gauss–Gauss nodes and the MAC grid nodes in the spatial element Ω_e . Regardless of whether an element is classified as a “spectral element” or low order CISL-MOF sub-elements, the fluid variables are discretized either on the Gauss–Gauss grid (pressure—black circles), Gauss–Lobatto–Gauss grid (u black horizontal rectangles), Gauss–Gauss–Lobatto grid (v black vertical rectangles), or material centroids within a sub-element (volume-of-fluid function F or centroid X magenta circles) (Color figure online)

with a volume fraction that exceeds 10^{-8}) in the element, or its’ immediate neighbors which share an element face, are found. A CISL-MOF element is broken up into $p^{(x)} p^{(y)} p^{(z)}$ sub-elements and each sub-element is discretized using the CISL advection algorithm and standard finite volume discretization for the pressure and viscosity equation. In the description of the CISL-MOF algorithm, the MISDC indexes n and k are dropped for brevity:

- Directionally split CISL-MOF advection (Volume Fraction):

$$F_{i,j,k}^{M,m+1} = \frac{\int_{\Omega_{i,j,k}^{Depart}} \chi^{M,m}(\mathbf{x}) d\mathbf{x}}{|\Omega_{i,j,k}^{Depart}|} \quad M = 1 \dots \mathcal{M}$$

Referring to Fig. 2, $\chi^{M,m}(\mathbf{x})$ is a characteristic function which is equal to one if \mathbf{x} is a member of the piecewise linear Moment-of-fluid reconstructed material region Ω^M . Figure 3 illustrates the CISL-MOF algorithm for integrating the solution for the volume-of-fluid function F^M .

- Directionally split CISL-MOF advection (Centroid):

$$X_{i,j,k}^{M,m+1} = \frac{\int_{\mathcal{T}(\Omega_{i,j,k}^{M,Depart})} \mathbf{x} \chi^{M,m}(\mathcal{T}^{-1}(\mathbf{x})) d\mathbf{x}}{|\mathcal{T}(\Omega_{i,j,k}^{M,Depart})|} \quad M = 1 \dots \mathcal{M}$$

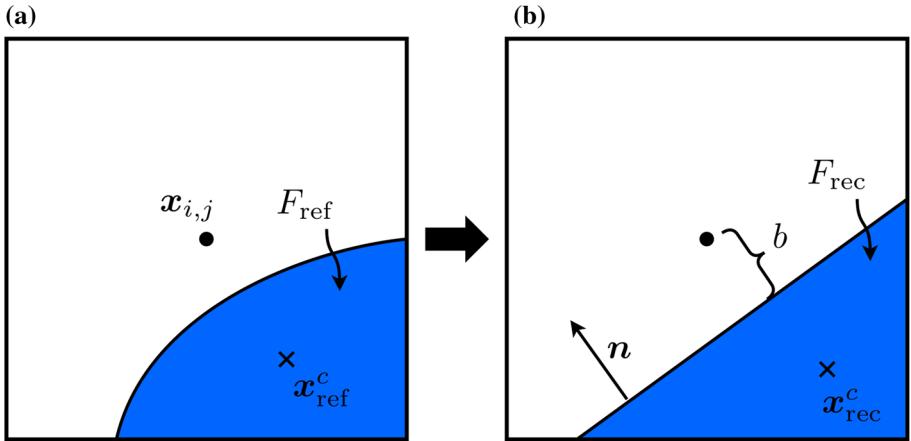


Fig. 2 **a** Material domain, $\Omega_{i,j}^{M,\text{exact}}$, at cell $\{i, j\}$ for one of the phases, and corresponding reference volume fraction and centroid. **b** Piecewise linear reconstruction of the interface using the MOF method. The approximate material domain is represented by a linearly cut cell: $\Omega_{i,j}^M = \Omega_{i,j} \cap \{x | \mathbf{n} \cdot (x - x_{i,j}) + b = 0\}$. We define a characteristic function $\chi^M(x)$ in which $\chi^M(x) = 1$ if $x \in \Omega_{i,j}^M$

- Directionally split CISL-MOF advection (Level set functions):

$$\phi_{i,j,k}^{M,m+1} = \frac{\int_{\Omega_{i,j,k}^{M,\text{Depart}}} \phi^{M,m}(x) dx}{|\Omega_{i,j,k}^{\text{Depart}}|} \quad M = 1 \dots \mathcal{M}$$

- Directionally split CISL-MOF advection (MAC velocity):

$$u_{i-1/2,j,k}^{adv,m+1} = \frac{\int_{\Omega_{i-1/2,j,k}^{\text{Depart}}} \sum_{M=1}^{\mathcal{M}} \rho^M \chi^{M,m}(x) u^m(x) dx}{\int_{\Omega_{i-1/2,j,k}^{\text{Depart}}} \sum_{M=1}^{\mathcal{M}} \rho^M \chi^{M,m}(x) dx} \quad (8)$$

$$v_{i,j-1/2,k}^{adv,m+1} = \frac{\int_{\Omega_{i,j-1/2,k}^{\text{Depart}}} \sum_{M=1}^{\mathcal{M}} \rho^M \chi^{M,m}(x) v^m(x) dx}{\int_{\Omega_{i,j-1/2,k}^{\text{Depart}}} \sum_{M=1}^{\mathcal{M}} \rho^M \chi^{M,m}(x) dx} \quad (9)$$

$$w_{i,j,k-1/2}^{adv,m+1} = \frac{\int_{\Omega_{i,j,k-1/2}^{\text{Depart}}} \sum_{M=1}^{\mathcal{M}} \rho^M \chi^{M,m}(x) w^m(x) dx}{\int_{\Omega_{i,j,k-1/2}^{\text{Depart}}} \sum_{M=1}^{\mathcal{M}} \rho^M \chi^{M,m}(x) dx} \quad (10)$$

Figures 4 and 5 illustrate the CISL-MOF algorithm for integrating the solution for the velocity, $u_{i-1/2,j,k}$, $v_{i,j-1/2,k}$, and $w_{i,j,k-1/2}$.

- Viscosity:

1. create a provisional cell centered velocity field:

$$\mathbf{u}^{adv,CELL} = \mathcal{I}_{MAC}^{CELL} \mathbf{u}^{adv,m+1}$$

- 2.

$$\mathbf{u}^{visc,CELL} = \mathbf{u}^{adv,CELL} + \Delta t_m \frac{\nabla \cdot (2\mu \mathbb{D})^{visc,CELL}}{\rho} \quad (11)$$

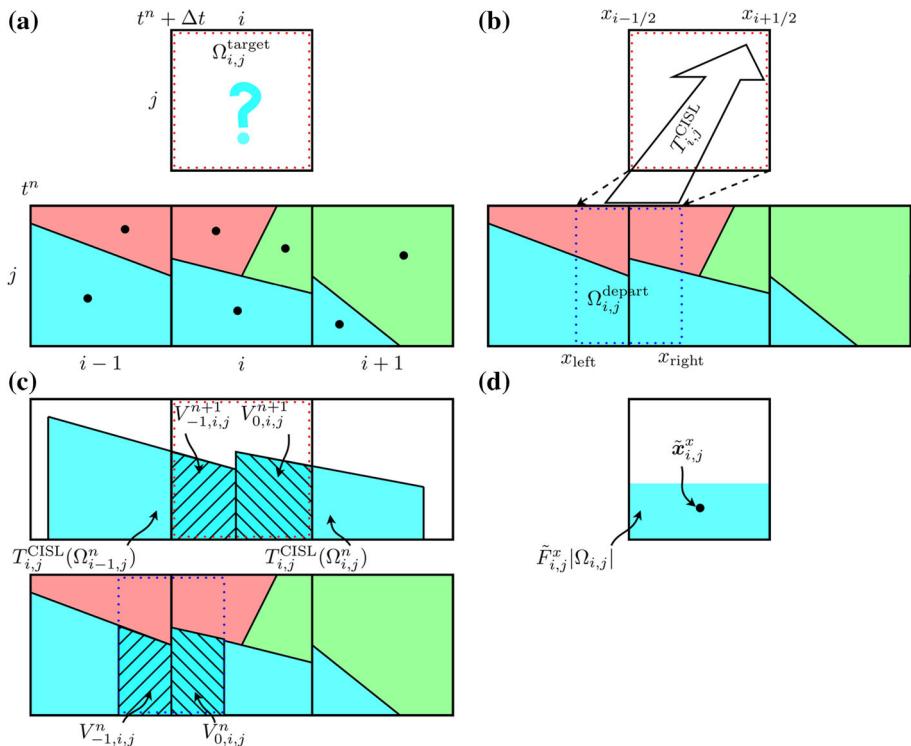


Fig. 3 CISL-MOF method for mass distribution and centroid position. The backward characteristic tracing is shown for the blue material in x direction. **a** Material distribution, centroid position and interface reconstruction for cells $\{i-1, j\}$, $\{i, j\}$ and $\{i+1, j\}$ at time t^n , and target region at time $t^n + \Delta t$. Centroids are shown in black dots. **b** Characteristic tracing and calculation of the mapping function. **c** Transformation of material domain using the mapping function and finding the intersection with the target region. **d** Combining the volume fraction and centroid in the target region (Color figure online)

3.

$$\mathbf{u}^{visc,m+1} = \mathbf{u}^{adv,m+1} + \mathcal{I}_{CELL}^{MAC}(\mathbf{u}^{visc,CELL} - \mathbf{u}^{adv,CELL})$$

– Pressure projection:

$$\begin{aligned} \mathcal{F}_{surfacetension} &= \frac{\sum_{M_1, M_2} \nabla \cdot ((\mathbb{I} - \mathbf{n}^{M_1, M_2} (\mathbf{n}^{M_1, M_2})^T) \sigma^{M_1, M_2} \chi^{M_1, M_2} \delta(\phi^{M_1, M_2}))}{\rho} \\ \mathbf{u}^{*,m+1} &= \mathbf{u}^{visc,m+1} + \Delta t_m \mathbf{g} + \Delta t_m \mathcal{F}_{surfacetension} \\ -\nabla \cdot \frac{\nabla p^{m+1}}{\rho} &= -\frac{\nabla \cdot \mathbf{u}^{*,m+1}}{\Delta t_m} \\ \mathbf{u}^{m+1} &= \mathbf{u}^{*,m+1} - \Delta t_m \frac{\nabla p^{m+1}}{\rho} \end{aligned} \quad (12)$$

Remarks:

- We choose the staggered grid velocity and pressure coupling in this article over the collocated formulation in our previous work, [25,35], since we experienced stability problems

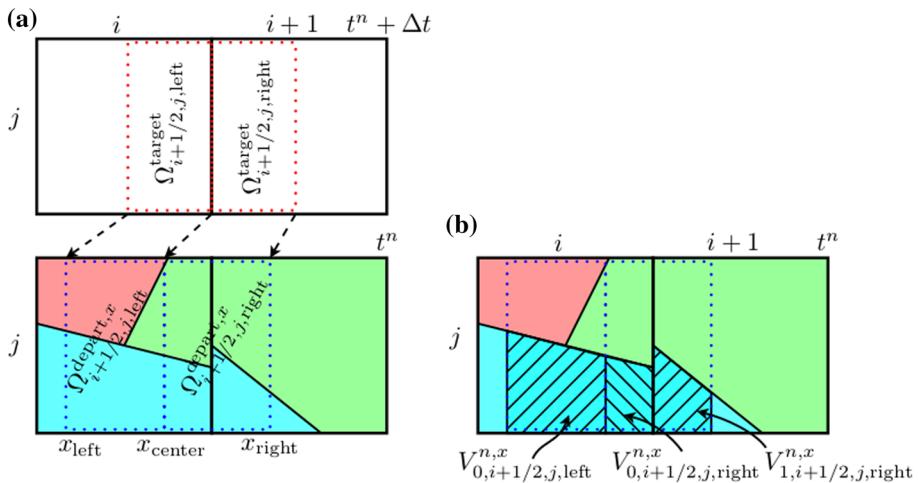


Fig. 4 CISL-MOF method for x-momentum advection. The backward characteristic tracing is shown for the blue material in x direction. **a** Left and right departure regions and their boundaries. **b** Finding intersections of departure regions and the domain of material M (Color figure online)

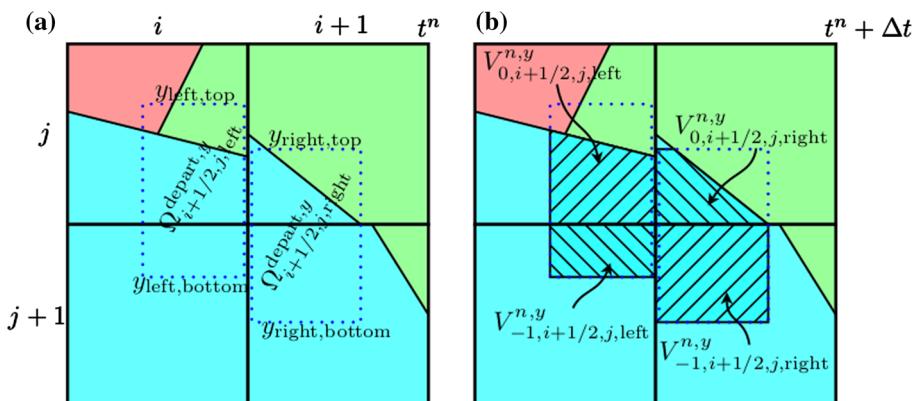
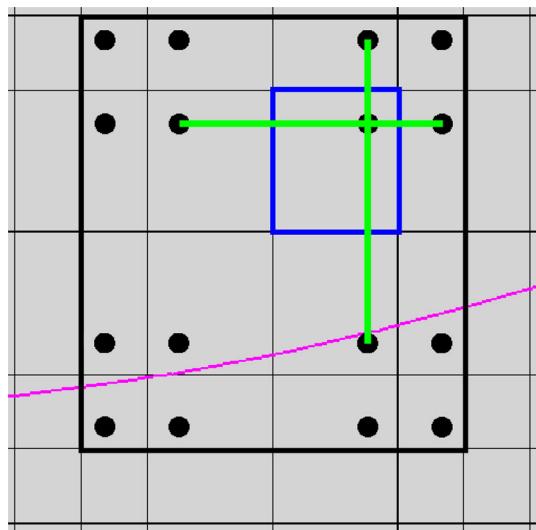


Fig. 5 CISL-MOF method for momentum advection. The backward characteristic tracing is shown for the blue material in y direction. Target regions are the same as shown in Fig. 4a. **a** Left and right departure regions. **b** Finding the intersections of departure regions and the domain of material M (Color figure online)

with the collocated discretization; it is possible for the velocity in the collocated formulation to have a gradient component that is not completely eliminated by the approximate cell centered projection.

- In the “CISL-MOF” elements, we discretize the viscosity force in the present work using an implicit backwards Euler time integration scheme, rather than an explicit sub-cycling scheme used in our previous work [25,35], or the semi-implicit scheme used in [31,61], since either the explicit sub-cycling approach or the semi-implicit approach would be inefficient for computing problems with highly viscous materials.
- The linear, large sparse matrix systems associated with (i) the pressure projection equation [(12) in the CISL-MOF region and (24) in the spectral region] and (ii) the viscosity equation [(11) in the CISL-MOF region and (23) in the spectral region] are solved using

Fig. 6 Illustration of a 4th order spectral element that contains five segments of the piecewise linear MOF reconstructed interface (purple curve). This element is tagged as a low order CISL-MOF element and the spectral element is broken up into 16 low order sub-elements. See for example, the blue sub-element. For advection, variables have a piecewise constant or slope limited piecewise linear representation within each sub-element. The discretization of $\nabla \cdot \frac{1}{\rho} \nabla p$ or $\nabla \cdot (2\mu \mathbb{D})$ uses the finite volume method in the blue square in which ∇p is approximated using finite difference methods along the 5 point green stencil and \mathbb{D} would use a 9 point stencil (Color figure online)



the iterative nested Krylov Subspace method: the “Multigrid-GMRES Preconditioned BiCGStab” method. The discretization for the Multigrid smoother always corresponds to the low order finite volume method illustrated in Fig. 6. We describe our “Multigrid-GMRES Preconditioned BiCGStab” algorithm in Sect. 7.

- At the beginning of each time slab (Fig. 7), the CISL-MOF elements are identified. Since an element at a new slab might be converted from a low order CISL-MOF element to a space-time spectral element, or vice versa, the velocity might not be discretely divergence free in the whole computational domain at the start of a new time slab. As a consequence, a method that discretizes the nonlinear advection terms in the following form,

$$\nabla \cdot (\mathbf{u} \otimes \mathbf{u}), \quad (13)$$

is not guaranteed to be “free stream preserving.” In otherwords, the advection by \mathbf{u} of a constant scalar function S is not guaranteed to preserve the constant property of S . We overcome this issue by discretizing (13) in the spectral element regions using the following (analytically identical) form instead:

$$\nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \mathbf{u} \nabla \cdot \mathbf{u}. \quad (14)$$

- The CISL-MOF advection algorithm is a directionally split algorithm [25,55] which alternates between the Eulerian-Implicit and Lagrangian-Explicit CISL-MOF algorithm. After CISL-MOF advection, the level set function ϕ^M is overwritten with the exact signed distance to the piecewise linear moment-of-fluid reconstructed interface. The moment-of-fluid reconstruction uses the gradient of the initial level set function has an initial guess for the reconstructed slope.

4 Space-Time Discretization

In a space-time discretization, we introduce a *space-time domain* \mathcal{E} as $\mathcal{E} = \Omega \times [t^0, T]$. A point in the space-time domain, $\bar{\mathbf{x}} \in \mathcal{E}$, has coordinates (\mathbf{x}, t) . First, we partition the time

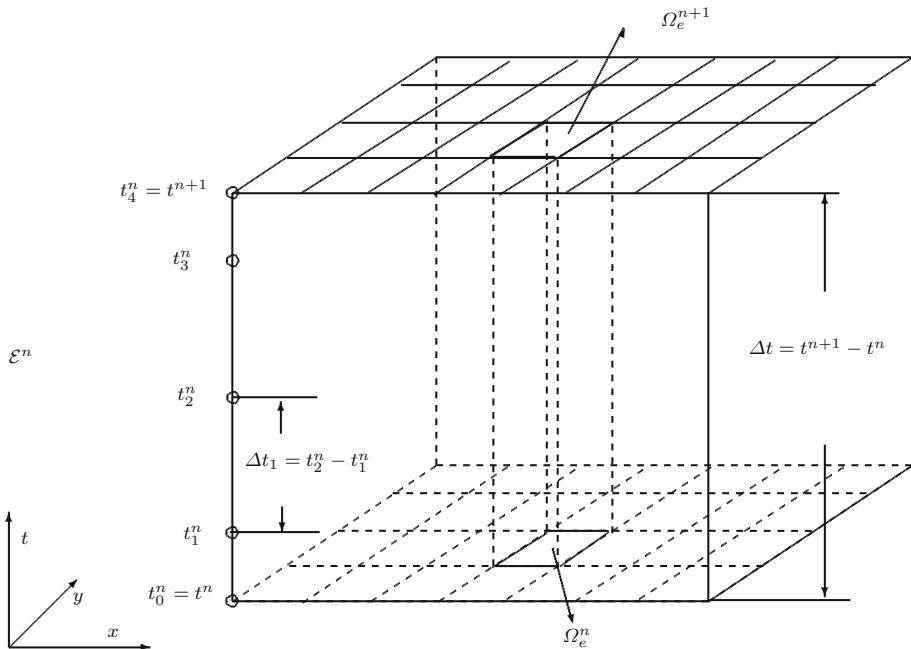


Fig. 7 An illustration of a space-time slab \mathcal{E}^n . The time interval $[t^n, t^{n+1}]$ is divided into 4 subintervals by choosing Legendre Gauss-Lobatto points t_m^n for $m = 0, \dots, 4$. $\Delta t = t^{n+1} - t^n$ is referred to as a *time step size* while $\Delta t_m = t_{m+1}^n - t_m^n$ is referred to as a *time sub-step size*. The small rectangle cube is the space-time element \mathcal{K}_e^n constructed by connecting Ω_e^n and Ω_e^{n+1}

interval $[t^0, T]$ uniformly by the time levels $0 = t^0 < t^1 < \dots < t^{E^{(t)}} = T$. The space-time domain \mathcal{E} is then divided into $E^{(t)}$ *space-time slabs*. The n -th space-time slab is denoted as $\mathcal{E}^n = \mathcal{E} \cap I^n$, where $I^n = [t^n, t^{n+1}]$ is the n -th time interval with length $\Delta t = t^{n+1} - t^n$. Next, we divide the spatial domain Ω into $E^{(x)} \times E^{(y)} \times E^{(z)}$ non-overlapping rectangular spatial elements. Let Ω_e^n and Ω_e^{n+1} be the spatial element e at time level t^n and t^{n+1} , respectively. A space-time element \mathcal{K}_e^n is then obtained by connecting Ω_e^n and Ω_e^{n+1} . The tessellation of the space-time domain is denoted as $\tilde{\mathcal{T}}_h$.

In each space-time slab \mathcal{E}^n , the time interval $I^n = [t^n, t^{n+1}]$ is divided into $p^{(t)}$ subintervals by choosing Legendre Gauss-Lobatto points t_m^n for $m = 0, \dots, p^{(t)}$, that is, $t^n = t_0^n < t_1^n < \dots < t_{p^{(t)}}^n = t^{n+1}$. In the following, the length Δt ($\Delta t = t^{n+1} - t^n$) is referred to as a *time step size* while Δt_m ($\Delta t_m = t_{m+1}^n - t_m^n$) is referred to as a *time sub-step size*. Fig. 7 is an illustration of a space-time slab.

4.1 Basis Functions

The pressure and velocity are represented by a nodal spectral element expansion in which the approximate pressure is discretized at the Gauss–Gauss–Gauss nodes and the velocity vector is defined on the marker-and-cell (MAC) grid. The MAC grid consists of Gauss–Lobatto–Gauss–Gauss points for the velocity component u in the x -direction, Gauss–Gauss–Lobatto–Gauss points for the velocity component v in the y -direction, and Gauss–Gauss–Gauss–Lobatto points for the velocity component w in the z -direction.

The locations of the Gauss–Gauss–Gauss points and the MAC grid points are illustrated in Fig. 1.

The basis functions are Lagrange interpolation polynomials defined as follows,

$$\ell_i^g(s) = \prod_{\substack{k=0 \\ k \neq i}}^r \frac{s - s_k^g}{s_i^g - s_k^g}, \quad \ell_i^{gl}(s) = \prod_{\substack{k=0 \\ k \neq i}}^{r+1} \frac{s - s_k^{gl}}{s_i^{gl} - s_k^{gl}}, \quad (15)$$

where $\{s_i^g\}_{i=0,\dots,r}$ are the roots of the $(r + 1)$ -th order Legendre Gauss polynomials, and $\{s_i^{gl}\}_{i=0,\dots,r+1}$ are the roots of the $(r + 2)$ -th order Legendre Gauss–Lobatto polynomials. Then, the discrete pressure and the discrete velocity vector in the space-time slab \mathcal{E}^n are defined as follows,

$$p_h(\mathbf{x}, t)|_{\Omega^e} = \sum_{i,j,k,l=0}^{p^{(x)}-1, p^{(y)}-1, p^{(z)}-1, p^{(t)}} \hat{p}_{i,j,k,l} \psi_i^g(x) \psi_j^g(y) \psi_k^g(z) \psi_l^{gl}(t), \quad (16a)$$

$$u_h(\mathbf{x}, t)|_{\Omega^e} = \sum_{i,j,k,l=0}^{p^{(x)}, p^{(y)}-1, p^{(z)}-1, p^{(t)}} \hat{u}_{i,j,k,l} \psi_i^{gl}(x) \psi_j^g(y) \psi_k^g(z) \psi_l^{gl}(t), \quad (16b)$$

$$v_h(\mathbf{x}, t)|_{\Omega^e} = \sum_{i,j,k,l=0}^{p^{(x)}-1, p^{(y)}, p^{(z)}-1, p^{(t)}} \hat{v}_{i,j,k,l} \psi_i^g(x) \psi_j^{gl}(y) \psi_k^g(z) \psi_l^{gl}(t), \quad (16c)$$

$$w_h(\mathbf{x}, t)|_{\Omega^e} = \sum_{i,j,k,l=0}^{p^{(x)}-1, p^{(y)}-1, p^{(z)}, p^{(t)}} \hat{w}_{i,j,k,l} \psi_i^g(x) \psi_j^g(y) \psi_k^{gl}(z) \psi_l^{gl}(t), \quad (16d)$$

where

$$\psi_i^g(x) = \ell_i^g(s), \quad \psi_i^{gl}(x) = \ell_i^{gl}(s), \quad \text{with } x = x_e + \Delta x_{\ell,e}(s + 1)/2, \quad (17a)$$

$$\psi_j^g(y) = \ell_j^g(s), \quad \psi_j^{gl}(y) = \ell_j^{gl}(s), \quad \text{with } y = y_e + \Delta y_{\ell,e}(s + 1)/2, \quad (17b)$$

$$\psi_k^g(z) = \ell_k^g(s), \quad \psi_k^{gl}(z) = \ell_k^{gl}(s), \quad \text{with } z = z_e + \Delta z_{\ell,e}(s + 1)/2, \quad (17c)$$

$$\psi_l^{gl}(t) = \ell_l^{gl}(s), \quad \text{with } t = t^n + \Delta t(s + 1)/2. \quad (17d)$$

Here (x_e, y_e, z_e) denotes the lower bounds of the spatial element Ω^e and $(\Delta x_{\ell,e}, \Delta y_{\ell,e}, \Delta z_{\ell,e})$ represents the lengths of the spatial element Ω^e on AMR level ℓ .

5 Multi-implicit Space-Time Spectral Element Method

We have implemented the Multi-implicit space-time spectral element method for approximating the solutions of the Navier–Stokes equations (5) in the bulk fluid regions.

For improved capturing of vortical structures, and other long-time flow properties, we discretize the nonlinear advection force term, $\nabla \cdot (\mathbf{u} \otimes \mathbf{u})$, pressure gradient force term, $\nabla p/\rho$, and viscous force term $\nabla \cdot (2\mu_m \mathbb{D})$, with space-time spectral accuracy by way of the Multi-Implicit Spectral Deferred Correction (MISDC) method for time discretization and the spectral element method for space discretization [49]. We note that in some cases in which the Reynolds' number is large, and the flow is unsteady, for example see Sects. 12 and 13, there is no difference between discretizing the viscosity force term, $\nabla \cdot (2\mu_m \mathbb{D})$, with

space-time spectral accuracy versus the finite volume method in space (see Fig. 6) and the backwards Euler method in time. On the otherhand, we do observe a significant improvement in accuracy when discretizing the viscosity force term with space time spectral accuracy and the Reynolds number is 0.01 or 1.0; see Sect. 8.

5.1 Spectral Element Method in Space for the Nonlinear Advection, Pressure Gradient, and Viscosity Force Terms

5.1.1 Spectral Element Discretization in Space for the Pressure Projection Equation:

$$-\nabla \cdot \frac{\nabla p}{\rho} = -\frac{\nabla \cdot u^*}{\Delta t}$$

We use a spectral element method in the collocation form [1,24] for the spatial discretization. First, we describe the spatial discretization of the second order term $\nabla \cdot \frac{1}{\rho} \nabla p$ in the pressure projection equation (12) by two steps: discretization of the gradient operator and discretization of the divergence operator.

- *Discretization of the gradient operator.* In each spatial element, the approximate solution for pressure is located at the Gauss–Gauss–Gauss points (see Fig. 1). The x derivative of the gradient, $\frac{\partial}{\partial x}$, is located at the Gauss–Lobatto–Gauss–Gauss points, the y derivative of the gradient, $\frac{\partial}{\partial y}$, is located at the Gauss–Gauss–Lobatto–Gauss points, and the z derivative of the gradient, $\frac{\partial}{\partial z}$, is located at the Gauss–Gauss–Gauss–Lobatto points (see the MAC grid points in Fig. 1). Since the procedures of computing the x derivative is the same as the y and z derivatives, we describe the approximation of the x derivative as an example. We enhance the approximate solution stencil in a spatial element, which are referred to as *Extended-Gauss* points, by adding 2 points to the existing $p^{(x)}$ Gauss points (see the *Extended-Gauss* points in Fig. 8). The solution values are then interpolated from the set of *Extended-Gauss* points onto the Gauss–Lobatto points of $(p^{(x)} + 2)$ that has the same number points of *Extended-Gauss* points. The gradient values on the Gauss–Lobatto points of $(p^{(x)} + 1)$ are obtained by differentiating the solution interpolated from the $(p^{(x)} + 2)$ Gauss–Lobatto points and then evaluating at the $(p^{(x)} + 1)$ Gauss–Lobatto points. An illustration of nodes used in computing the x derivative is displayed in Fig. 8. For an element adjacent to the domain boundary, the set of *Extended-Gauss* points are constructed by adding the point on the domain boundary and the point from the neighbor element to enhance the stencil, see Fig. 9.
- *Discretization of the divergence operator.* In each spatial element, the divergence operator is approximated at the Gauss–Gauss–Gauss points which coincides with the location of the approximate solution (see Fig. 1). Due to the discontinuity of the gradient values at inter-element boundaries, the pressure gradient is double-valued across element boundaries. In order to define a unique flux at inter-element boundaries, we replace the double valued flux with the average of the coincident gradient values coming from each side of the inter-element face. The product of the continuous gradient values ∇p and the density coefficients $\frac{1}{\rho}$ on the MAC grid points are used to compute the differentiation. The values of the divergence operator are obtained by differentiating the product on the MAC grid points and then evaluating at the Gauss–Gauss points.

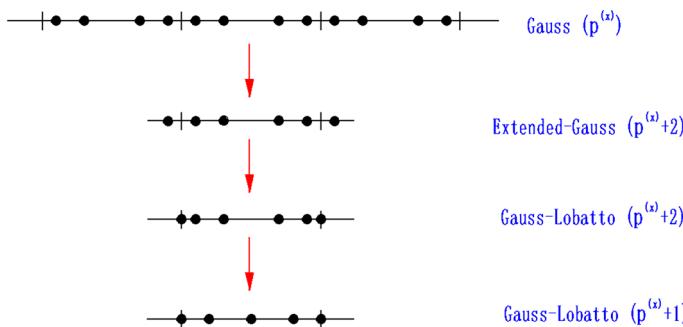
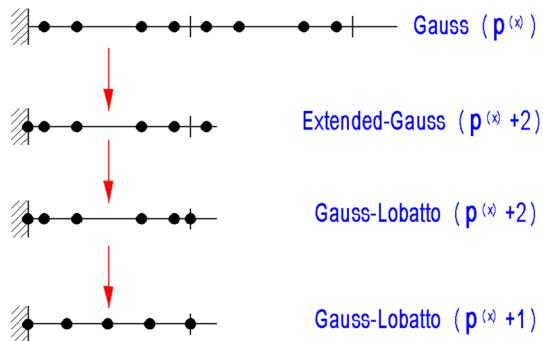


Fig. 8 An illustration of nodes used in computing the x derivative in the x -direction. Extended-Gauss points ($p^{(x)} + 2$) denotes the Gauss points together with two additional points from the neighbors, Gauss-Lobatto points ($p^{(x)} + 2$) denotes the Gauss-Lobatto points with the same number points of Extended-Gauss, and Gauss-Lobatto points ($p^{(x)} + 1$) denotes the MAC grid points in the x -direction

Fig. 9 An illustration of nodes used in computing the x derivative in the x -direction for the element near the wall. Extended-Gauss points ($p^{(x)} + 2$) denotes the Gauss points plus the point on the wall and a point from the neighbor, Gauss-Lobatto points ($p^{(x)} + 2$) denotes the Gauss-Lobatto points with the same number points of Extended-Gauss, and Gauss-Lobatto points ($p^{(x)} + 1$) denotes the MAC grid points in the x -direction



5.1.2 Spectral Element Discretization in Space for the Viscosity Equation:

$$\frac{u^* - u^{adv}}{\Delta t} - \nabla \cdot (2\mu_m \mathbb{D})^* = 0$$

To evaluate the viscous term in a spectral element, first the velocity values are interpolated from the MAC grid to the Gauss–Gauss–Gauss grid. Therefore, the evaluation of non-coupled derivative terms in \mathbb{D} are centered at the MAC grid points, while another interpolation step is needed to recenter the components of the coupled terms. To illustrate the procedure, here we illustrate the evaluation of the non-coupled u_x and coupled $u_y + v_x$ terms in 2D.

Non-coupled terms: To evaluate the derivative terms at Gauss–Lobatto–Gauss and Gauss–Gauss–Lobatto nodes the Gauss–Gauss–Gauss grids require one layer of ghost values on each side in the direction of the derivative from the outside of the spectral element to be included in the spectral derivative stencil. If a neighbor element has the same refinement as the current element, the ghost values are copied from the neighbor Gauss–Gauss–Gauss points directly. Otherwise, the ghost value is interpolated to the location that follows the current cell Gauss–Gauss–Gauss points in the derivative direction and matches the neighbor cell Gauss–Gauss–Gauss points in the derivative transverse directions (see Fig. 10).

Coupled term: The evaluation of derivatives for the coupled terms also require the extended stencil and one layer of ghost values. For the coupled terms, the ghost value positions are the mirror locations of the Gauss–Gauss–Gauss points with respect to the element boundaries. Using the extended derivative stencils, the evaluated u_y and v_x terms are centered at the MAC

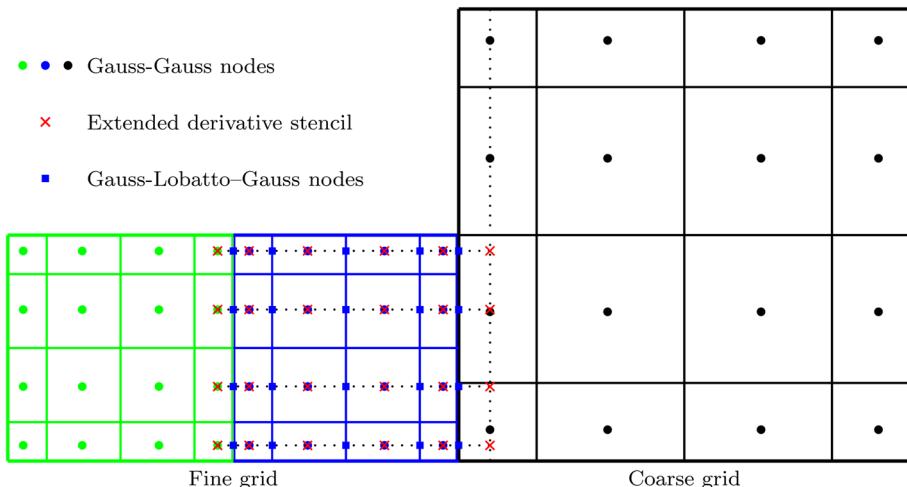


Fig. 10 Evaluation of the non-coupled term u_x in \mathbb{D} . The left and center spectral elements are on the fine grid while the right element is located on the coarse grid

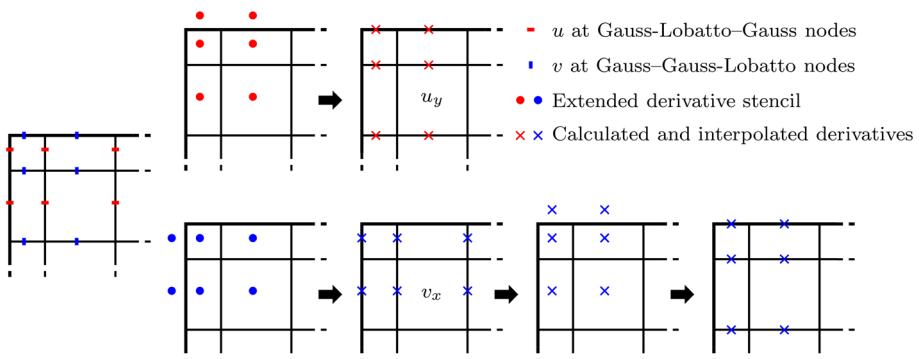


Fig. 11 Evaluation of the coupled terms $u_y + v_x$ in \mathbb{D}

grids in y and x direction respectively. v_x is then interpolated first to the Gauss–Gauss–Gauss grids and then to MAC y grids to be collocated with u_y (see Fig. 11).

5.1.3 Spectral Element Discretization in Space for the Nonlinear Advective Force Terms

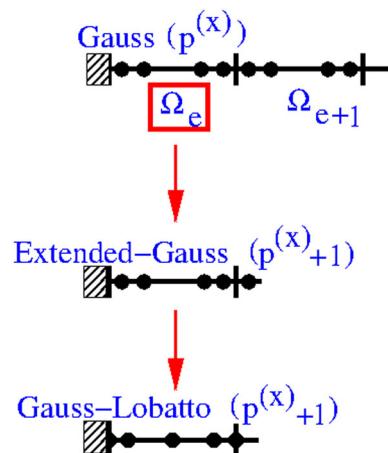
$$\frac{\mathbf{u}^{adv}-\mathbf{u}^n}{\Delta t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = 0$$

For the spatial discretization of the advective term,

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

we first interpolate the velocity (u, v, w) from the staggered MAC grid to the Gauss–Gauss–Gauss grid. U represents either u , v , or w after interpolation. \mathbf{u} is the original staggered MAC grid velocity. Once U has been interpolated to the Gauss–Gauss–Gauss grid, there are two parts for approximating $\nabla \cdot (\mathbf{u}U)$; (i) Including the two *Extended-Gauss* points in order to approximate U at the MAC grid points, and (ii) discretization of the divergence operator:

Fig. 12 An illustration of nodes used in computing the x derivative in the x -direction for an element next to a wall in which numerical boundary conditions are required; i.e. an illustration of the numerical stencil when the characteristics point out of the computational domain.
Extended-Gauss points ($p^{(x)} + 1$) denotes the Gauss points plus the point from the interior neighbor, and Gauss–Lobatto points ($p^{(x)} + 1$) denotes the MAC grid points in the x -direction



(i) *Including the two Extended-Gauss points in order to interpolate U at the MAC grid points.*

The use of the *Extended-Gauss* points for computing the x derivative is illustrated in Fig. 8 and is explained in detail as follows. To reconstruct the advective flux on the MAC grid points, the values of the approximate solution need to be interpolated from the Gauss–Gauss–Gauss points onto the MAC grid points. Take the x -component for example. We enhance the approximate solution stencil in a spatial element, which are referred to as *Extended-Gauss* points, by adding 2 points to the existing $p^{(x)}$ Gauss points (see the *Extended-Gauss* points in Fig. 8). The solution values are then interpolated from the set of *Extended-Gauss* points of $(p^{(x)} + 2)$ onto the Gauss–Lobatto points of $(p^{(x)} + 1)$. If a spectral element touches a Dirichlet boundary condition wall and the characteristics are pointing into the computational domain, then the extended node is located on the boundary of the element and the value associated with the node is the boundary condition; see Fig. 9. If the characteristics point out of the computational domain, then we apply numerical boundary conditions in which case there is no extended value on the wall; see Fig. 12.

- (ii) *Discretization of the divergence operator.* Due to the discontinuity of the solution values, the advective flux values are double-valued across the inter-element faces. The endpoints are then overwritten by the upwind values at the inter-element faces. The direction for upwinding is determined from the sign of the original MAC velocity \mathbf{u} . The values of the derivative are obtained by differentiating $\mathbf{u}U$ on the $(p^{(x)} + 1)$ Gauss–Lobatto points and then evaluating at the $p^{(x)}$ Gauss points.

After having approximated $\nabla \cdot (\mathbf{u}U) - U \nabla \cdot \mathbf{u}$ at the Gauss–Gauss–Gauss nodes, the advective force term is interpolated back to the MAC grid using extended interpolation.

5.2 Spectral Deferred Correction Method; Coupling of the Nonlinear Advection, Pressure Gradient, and Viscosity Force Terms

The multi-implicit spectral deferred correction (MISDC) method introduced by Bourlioux et al. [9] is a variant of the classical spectral deferred correction (SDC) method. SDC methods are derived from representing an evolution equation as an integral in time, and approximating this integral by high order quadrature rules. To reduce the integration error, a series of correction

equations are designed and solved by a low-order time-integration scheme. These correction equations can be applied iteratively to achieve arbitrary high-order accuracy in time. The choices of correction equations and the efficiency of variances of the SDC methods have been discussed in great detail by Layton [32,33].

A key feature of the MISDC method is that it iteratively couples all physical processes together by including the effects of each process during the integration of any particular process. In contrast, traditional operator-splitting methods ignore the effects of other processes, that is, each process is discretized in isolation. After the spatial discretization of the Navier Stokes equation, the resulting ODE system is rewritten as

$$\frac{\partial \mathbf{W}}{\partial t} = \mathbf{F}_A(\mathbf{W}) + \mathbf{F}_D(\mathbf{W}) + \mathbf{F}_{GP}(\mathbf{W}) = \mathbf{F}(\mathbf{W}(t), t), \quad (18)$$

where $\mathbf{F}_A(w)$ denotes the spatial discretization of the nonlinear advection term $\nabla \cdot (\mathbf{u} \otimes \mathbf{u})$, \mathbf{F}_D denotes the spatial discretization of the viscosity term $\frac{\nabla \cdot (2\mu \mathbb{D})^{visc}}{\rho}$, and \mathbf{F}_{GP} denotes the spatial discretization of the pressure gradient term $\frac{\nabla p}{\rho}$.

We use an explicit time discretization scheme for the nonlinear advection (the strongly stable first order I-scheme) building block, the backwards Euler method for the viscosity force term building block, and an implicit “projection” method for the pressure gradient force term building block. A complicated system of different processes can be solved in a decoupled fashion by applying operator splitting, but the overall order of accuracy is restricted by the splitting error. For example, the solution advanced in one time step by a naive operator splitting method can be obtained as follows,

$$\mathbf{W}_A(t^{n+1}) = \mathbf{W}(t^n) + \int_{t^n}^{t^{n+1}} \mathbf{F}_A(\mathbf{W}_A(\tau)) d\tau, \quad (19)$$

$$\mathbf{W}_D(t^{n+1}) = \mathbf{W}_A(t^{n+1}) + \int_{t^n}^{t^{n+1}} \mathbf{F}_D(\mathbf{W}_D(\tau)) d\tau, \quad (20)$$

$$\mathbf{W}(t^{n+1}) = \mathbf{W}_D(t^{n+1}) + \int_{t^n}^{t^{n+1}} \mathbf{F}_{GP}(\mathbf{W}(\tau)) d\tau. \quad (21)$$

One can show that the above approximation is $\mathcal{O}(\Delta t)$ globally unless the operators associated with \mathbf{F}_A , \mathbf{F}_D and \mathbf{F}_{GP} commute. In comparison, the MISDC method can achieve higher order of temporal accuracy by the iterative coupling strategy which reduces both the splitting error and the integration error. Even though the time discretization error of our advection, viscosity, and pressure gradient building blocks are all first order when applied separately, when these separate parts are combined strategically, we can obtain, in a non-intrusive manner, a resulting method that is spectrally accurate in time [49].

We developed the following simple MISDC coupling strategy [49]:

$$\begin{aligned} \mathbf{W}_{A,m+1}^{n,k+1} &= \mathbf{W}_m^{n,k+1} + \int_{t_m^n}^{t_m^{n+1}} [\mathbf{F}_A(\mathbf{W}_A^{k+1}) - \mathbf{F}_A(\mathbf{W}_A^k)] d\tau \\ &\quad + \int_{t_m^n}^{t_m^{n+1}} [\mathbf{F}_A(\mathbf{W}^k)] d\tau \end{aligned} \quad (22)$$

$$\begin{aligned} \mathbf{W}_{AD,m+1}^{n,k+1} &= \mathbf{W}_{A,m}^{n,k+1} + \int_{t_m^n}^{t_m^{n+1}} [\mathbf{F}_D(\mathbf{W}_{AD}^{k+1}) - \mathbf{F}_D(\mathbf{W}_{AD}^k)] d\tau \\ &\quad + \int_{t_m^n}^{t_m^{n+1}} [\mathbf{F}_D(\mathbf{W}^k)] d\tau. \end{aligned} \quad (23)$$

$$\begin{aligned} \mathbf{W}_{m+1}^{n,k+1} = & \mathbf{W}_{AD,m}^{n,k+1} + \int_{t_m^n}^{t_{m+1}^n} [\mathbf{F}_{GP}(\mathbf{W}^{k+1}) - \mathbf{F}_{GP}(\mathbf{W}^k)] d\tau \\ & + \int_{t_m^n}^{t_{m+1}^n} [\mathbf{F}_{GP}(\mathbf{W}^k)] d\tau. \end{aligned} \quad (24)$$

In order to construct an arbitrary high-order time integration scheme, the second integrals in (22) through (24) are evaluated by Gaussian quadrature. In order to derive the Gaussian quadrature approximation for the integral,

$$\int_{t_m^n}^{t_{m+1}^n} \mathbf{F}(\tau) d\tau, \quad (25)$$

one first interpolates $\mathbf{F}(\tau)$ from the Gauss–Lobatto time quadrature nodes,

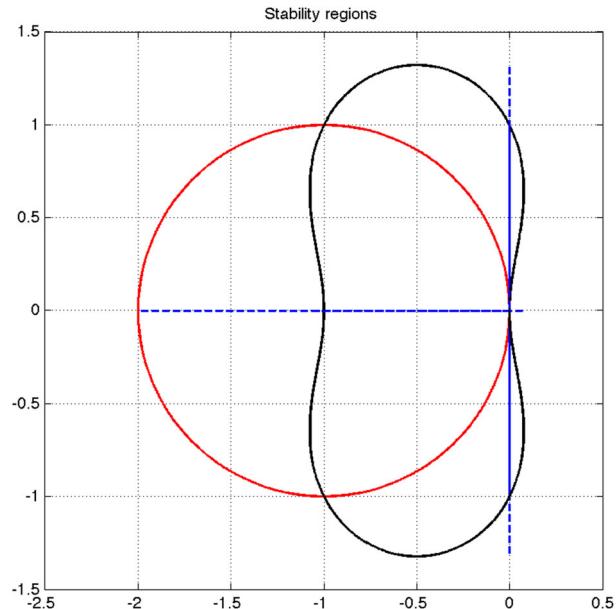
$$[t_0^n, \dots, t_{p^{(t)}}^n],$$

to the $p^{(t)} + 1$ Gauss time quadrature nodes which are distributed along the interval $[t_m^n, t_{m+1}^n]$, then one uses standard Gaussian quadrature in order to approximate (25). The first integrals in (22,...,24) are discretized by a low-order time integration scheme, e.g., a first order time integration scheme.

First order I-scheme for low order time integration of the nonlinear advection force terms

We have found that using the standard explicit Euler time integration scheme as a low order building block for integrating the nonlinear advection force terms, $\mathbf{F}_A(\mathbf{W}) = \nabla \cdot (\mathbf{u} \otimes \mathbf{u})$, results in an overall unstable numerical method. Instead of the forward Euler method, we have implemented the first-order I-stable building block time integration scheme developed

Fig. 13 A comparison of the stability region of the first order I-stable scheme (black) with the first order forward Euler scheme (red) (Color figure online)



by Bao and Jin [5]:

$$\mathbf{W}_*^{n,k+1} = \mathbf{W}_m^{n,k+1} + \Delta t_m \mathbf{F}_A(\mathbf{W}_m^{n,k+1}, t_m^n), \quad (26a)$$

$$\mathbf{W}_{m+1}^{n,k+1} = \mathbf{W}_m^{n,k+1} + \Delta t_m \mathbf{F}_A(\mathbf{W}_*^{n,k+1}, t_{m+1}^n). \quad (26b)$$

The region of absolute stability of an I-stable method contains part of the imaginary axis. In Fig. 13, the stability region of the first order I-stable scheme (black) is compared to that of the forward Euler (red) scheme.

Overall coupling strategy for our multi-implicit space-time spectral element method for the Navier–Stokes equations

The discretization of the present multi-implicit space-time spectral element method, which corresponds to coupling strategy “1” described in Pei et al. [49], is described in Algorithm 1 below:

Algorithm 1 The multi-implicit space-time spectral element method uses the first order I-stable building block scheme for the nonlinear advection force terms, the backwards Euler method building block scheme for the viscosity force terms, and the projection method for the pressure gradient force terms. The non-symmetric large sparse matrix systems (viscosity and pressure gradient force terms) are solved using the MG-GMRES preconditioned BiCGStab iterative method (Section 7)

```

1: for  $m = 0, \dots, p^{(t)} - 1$  do
2:    $\mathbf{F}_A(\mathbf{W}_m^0, t_m^0) = 0$ .
3:    $\mathbf{F}_A(\mathbf{W}_*^0, t_m^0) = 0$ .
4:    $I_m^{m+1}(\mathbf{F}_A(\mathbf{W}^0)) = 0$ .
5:    $\mathbf{F}_D((\mathbf{W}_{AD})_{m+1}^0, t_{m+1}^0) = 0$ .
6:    $I_m^{m+1}(\mathbf{F}_D(\mathbf{W}^0)) = 0$ .
7:    $\mathbf{F}_{GP}(\mathbf{W}_{m+1}^0, t_{m+1}^0) = 0$ .
8:    $I_m^{m+1}(\mathbf{F}_{GP}(\mathbf{W}^0)) = 0$ .
9: end for
10: for  $k = 1, \dots, K$  do
11:    $\mathbf{W}_0^k = \mathbf{W}(t^n)$ .
12:   for  $m = 0, \dots, p^{(t)} - 1$  do
13:     Compute explicitly  $\mathbf{F}_A(\mathbf{W}_m^k, t_m^n)$  (Section 5.1.3). ▷ Advection process
14:      $(\mathbf{W}_A)_*^k = \mathbf{W}_m^k + \Delta t_m \mathbf{F}_A(\mathbf{W}_m^k, t_m^n)$ .
15:     Compute  $\mathbf{F}_A(\mathbf{W}_*^k, t_m^n)$  (Section 5.1.3).
16:      $(\mathbf{W}_A)_{m+1}^k = \mathbf{W}_m^k + \Delta t_m [\mathbf{F}_A((\mathbf{W}_A)_*^k, t_{m+1}^n) - \mathbf{F}_A((\mathbf{W}_A)_*^{k-1}, t_{m+1}^n)]$ 
         $+ I_m^{m+1}(\mathbf{F}_A(\mathbf{W}^{k-1}))$ .
17:     Compute implicitly  $\mathbf{F}_D((\mathbf{W}_{AD})_{m+1}^k, t_{m+1}^n)$  (Section 5.1.2). ▷ Diffusion process
18:      $(\mathbf{W}_{AD})_{m+1}^k = (\mathbf{W}_A)_{m+1}^k + \Delta t_m [\mathbf{F}_D((\mathbf{W}_{AD})_{m+1}^k, t_{m+1}^n) - \mathbf{F}_D((\mathbf{W}_{AD})_{m+1}^{k-1}, t_{m+1}^n)]$ 
         $+ I_m^{m+1}(\mathbf{F}_D(\mathbf{W}^{k-1}))$ .
19:     Compute implicitly  $\mathbf{F}_{GP}(\mathbf{W}_{m+1}^k, t_{m+1}^n)$  (Section 5.1.1). ▷ Projection process
20:      $\mathbf{W}_{m+1}^k = (\mathbf{W}_{AD})_{m+1}^k + \Delta t_m [\mathbf{F}_{GP}(\mathbf{W}_{m+1}^k, t_{m+1}^n) - \mathbf{F}_{GP}(\mathbf{W}_{m+1}^{k-1}, t_{m+1}^n)]$ 
         $+ I_m^{m+1}(\mathbf{F}_{GP}(\mathbf{W}^{k-1}))$ .
21:   end for
22:   Compute  $I_m^{m+1}(\mathbf{F}_A(\mathbf{W}^k))$ ,  $I_m^{m+1}(\mathbf{F}_D(\mathbf{W}^k))$ , and  $I_m^{m+1}(\mathbf{F}_{GP}(\mathbf{W}^k))$  with the updated solutions  $\mathbf{W}^k$ .
23: end for
```

With a coupling strategy used in a series of deferred correction steps, the temporal splitting error is eliminated iteratively. We conjecture that our multi-implicit space-time spectral element method, Algorithm 1, has an overall order of accuracy

$$\min\{p^{(x)} + 1, p^{(t)} + 1, K\}, \quad (27)$$

where K is the number of iterations in the multi-implicit SDC method, $p^{(x)}$ is the polynomial order in the x , y , or z directions, and $p^{(t)}$ is the polynomial order in the temporal direction.

6 Scalable, Hierarchical, Space-Time Spectral Element Method Using the BoxLib Libraries

The BoxLib libraries (see [68] which describes the version that we are using in order to produce the results in this paper), now called AMReX, are developed in order to simplify the development of numerical algorithms that use dynamic, scalable, hierarchical, block structured, adaptive mesh refinement [3,6,8,17,18,27,62]. In Fig. 14 we show an example of the grid structure used in adaptive mesh refinement if each grid patch in the hierarchy consists of fourth order elements. We have modified BoxLib so that while internally grid points on grid patches are assumed uniformly spaced, externally, we map grid cells and faces so that their locations correspond to Legendre Gauss and Legendre Gauss Lobatto points respectively. See Fig. 1. The grid hierarchy is composed of different levels of refinement ranging from coarsest, $\ell = 0$, to finest, $\ell = \ell_{max}$. Each level is represented as the union of rectangular grid patches of a given resolution. In our computations the refinement ratio between levels is 2. Thus we have for the case of uniformly spaced cells, i.e. $p_\ell^{(x)} \equiv 1$, $\Delta x_{\ell+1} = \Delta x_\ell/2$, $\Delta y_{\ell+1} = \Delta y_\ell/2$, and $\Delta z_{\ell+1} = \Delta z_\ell/2$. The grids are properly nested, in the sense that the union of grids at level $\ell + 1$ is contained in the union of grids at level ℓ for $0 \leq \ell < \ell_{max}$. Furthermore, the containment is strict in the sense that, except at physical boundaries, the level ℓ grids are large enough to guarantee that there is a border at least one level ℓ cell wide surrounding each level $\ell + 1$ grid.

We have modified the BoxLib libraries in order to allow one to select a different space order corresponding to different levels of refinement. The interpolation order of our “space-time BoxLib” method from level ℓ to level $\ell + 1$ is $p_\ell^{(x)}$ and the interpolation order from level $\ell + 1$ to level ℓ is $p_{\ell+1}^{(x)}$. See Fig. 14. In order to simplify the extension of “Hierarchical BoxLib” to “space-time Spectral Element Hierarchical BoxLib” we have the following restrictions:

- In the “spectral regions,” the time order, $2 \leq p^{(t)} \leq 16$, is the same for each level and must be a power of two. The space order, $2 \leq p_\ell^{(x)} \leq 16$ is allowed to depend on ℓ . In the interface regions, our method automatically reduces to the lower order cell integrated semi-Lagrangian method [25] (see Fig. 6).
- The space order on level ℓ , $p_\ell^{(x)}$, must be a power of two and either $p_\ell^{(x)} = p_{\ell+1}^{(x)}$ or $p_\ell^{(x)} = 2p_{\ell+1}^{(x)}$.
- In “BoxLib,” [3,17,62,68] a blocking factor must be prescribed on each level: $bfact_\ell$. For any grid box on level ℓ , each side of the box must be divisible by the blocking factor. For our hierarchical space-time method, we require that $bfact_\ell \geq bfact_{\ell+1}$ and that $bfact_\ell$ is a power of two.
- Spectral elements at level ℓ must be exactly aligned with elements at level $\ell - 1$ (see Fig. 14). This means the blocking factor must satisfy: $bfact_\ell \geq 2p_{\ell-1}^{(x)}$.

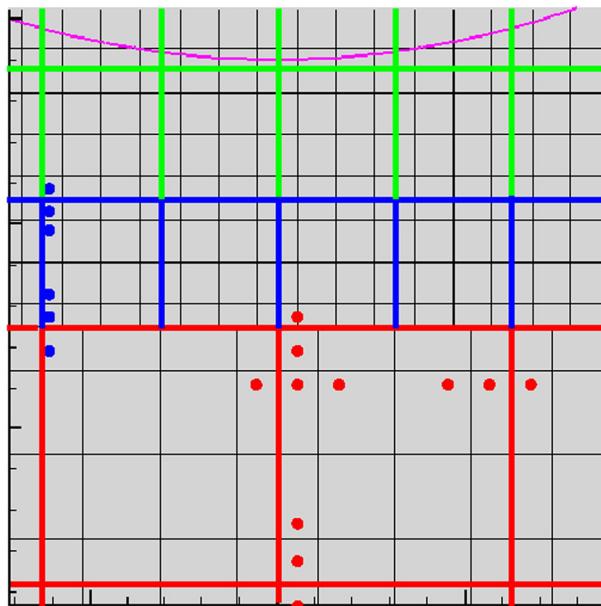


Fig. 14 Illustration of extended stencil used for the spectral element approximation of gradients and fluxes on a hierarchy of levels. The green fourth order elements either contain a moment-of-fluid reconstructed interface or neighbor an element that contains an interface. The green elements are broken up into low order CISL-MOF sub-elements. The blue elements are fourth order elements on level $\ell = 1$. The red elements are fourth order elements on level $\ell = 0$. The location of nodes within an element correspond to Legendre Gauss quadrature points. The grid lines coincide with Legendre Gauss Lobatto quadrature points. The Gauss quadrature positioning of nodes and sub-elements is essential so that only one neighbor in the normal direction needs to be communicated. The blue circle in the red element ($\ell = 0$) is an extended Gauss point for a stencil of points on level $\ell = 1$ and is interpolated from the red element in the x direction (tangent to the adjoining face); it is unnecessary to interpolate to the blue circle's location in the y direction (normal to the adjoining face). The red circle in the blue element ($\ell = 1$) is an extended Gauss point for a stencil of points on level $\ell = 0$ and is interpolated from the blue element in the x direction (tangent to the adjoining face); it is unnecessary to interpolate to the red circle's location in the y direction (normal to the adjoining face). For initializing coarse level boundary conditions for a finer level, Message passing is done first on level ℓ followed by interpolation in the tangential direction(s) to level $\ell + 1$. For the opposite case, interpolation is done first in the tangential direction(s) on level $\ell + 1$ prior to message passing on level ℓ (Color figure online)

- The depth of level ℓ cells that buffer uncovered level $\ell - 1$ grid cells from a level $\ell + 1$ grid must be at least $p_\ell^{(x)}$ (the exception being that a level $\ell + 1$ grid box coincides with the computational domain boundary). This means that we must prescribe $\text{bfact}_\ell \geq 2p_\ell^{(x)}$. We call this condition the “spectral element proper nesting condition.”
- The element size on level ℓ is $\Delta x_{\ell,e}$ and satisfies,

$$\Delta x_{\ell,e} = 2 \frac{p_\ell^{(x)}}{p_{\ell+1}^{(x)}} \Delta x_{\ell+1,e} \quad (28)$$

7 Nested Krylov Subspace Method: The “Multigrid-GMRES Preconditioned BiCGStab” Algorithm on a Hierarchy of Levels

The pressure projection equation can be written as [see (12) or (24)],

$$\nabla \cdot \beta \nabla p = f, \quad (29)$$

and the viscosity equation can be written as [see (11) or (23)],

$$\alpha u - \nabla \cdot (2\mu \mathbb{D}) = f. \quad (30)$$

Both the discretization of (29) and (30) lead to non-singular, non-symmetric matrices. The discretization of (29) leads to a non-symmetric matrix because the spectral element method (5.1.1) is used in elements that do not neighbor mixed material elements and the finite volume method (Fig. 6) is used in all sub-elements of elements that neighbor multi-material cells. The discretization of (30) leads to a non-symmetric matrix too not only because the spectral element method (5.1.2) is used in elements that do not neighbor mixed material elements, but also because one-sided (first order) finite difference techniques are used for discretizing the viscous coupling terms in multi-material (partial volume fraction) sub-elements [61]. The finite volume method (Fig. 6) using central finite difference techniques is used in all single material (volume fraction is one) sub-elements of elements that neighbor multi-material cells.

We write the matrix system for either (29) or (30) as,

$$Ax = b, \quad (31)$$

where A is an $\mathcal{N} \times \mathcal{N}$ matrix. Our algorithm for solving the non-symmetric system, (31), is similar to the hierarchical Multigrid preconditioned conjugate gradient method described in Algorithm 3 of [18] except that (1) the Multigrid-GMRES preconditioner [41, 52, 66] replaces the Multigrid preconditioner used in [18] and (2) the Preconditioned BiCGStab method with $K = K_1 K_2$ and $K_1 = I$ (see [65, p. 638]) replaces the preconditioned conjugate gradient method used in [18]. It is stated in [65] that, in the absence of numerical round off error, the BiCGStab method is guaranteed to converge for non-symmetric matrices in less than or equal to \mathcal{N} steps. It is also stated in [65] that the magnitude of the residual decreases more evenly for the BiCGStab method than the Conjugate Gradient-Squared (CG-S) method.

A crucial difference between the present algorithm and a standard Preconditioned BiCGStab Algorithm [65] is that the multigrid smoother and bottom solver are not derived from A , instead the smoother and bottom solver are derived from the symmetric matrix A_{FVM} . In other-words, the multigrid part of the multigrid-GMRES preconditioner is written as

$$M_{FVM}z = r,$$

which is meant as an approximation to,

$$A_{FVM}z = r. \quad (32)$$

r is a residual that is passed from the GMRES preconditioner (in our algorithm, the GMRES method preconditions the BiCGStab method) and M_{FVM} (e.g. the red-black ILU smoother [18]) is derived from A_{FVM} instead of A . A_{FVM} is the matrix derived from the low order finite volume discretization (see Fig. 6) of the operator,

$$Lu = \alpha u - \nabla \Lambda \nabla u \quad \Lambda \text{ is a diagonal matrix.} \quad (33)$$

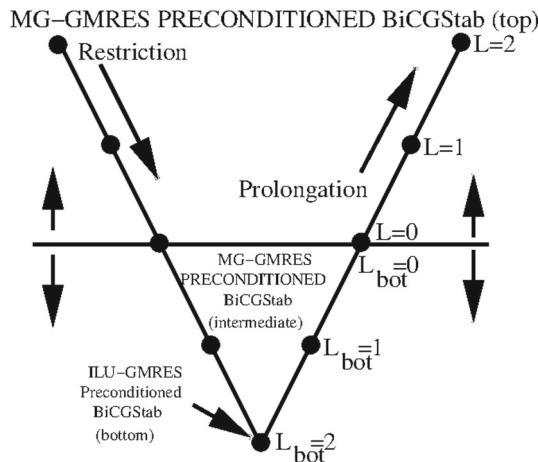


Fig. 15 Illustration of the V-cycle algorithm for a multigrid preconditioner on a hierarchy of levels. Levels $\ell = 1$ and $\ell = 2$ that are above the $\ell = 0$ line are adaptive mesh refinement levels and do not necessarily cover the whole computational domain. The “Multigrid-GMRES Preconditioned BiCGStab” method (top solver) is used on the levels above the $\ell = 0$ line. At level $\ell_{bot} = 0$ and below, the “Multigrid-GMRES Preconditioned BiCGStab” method is also used as an “intermediate solver” with a tolerance that is two orders of magnitude smaller than the top solvers’ tolerance. The multigrid smoother at all levels is derived from A_{FVM} (32). At the very bottom, $\ell_{bot} = 2$, the “red black ILU-GMRES Preconditioned BiCGStab” method has been implemented as the “bottom solver” with a tolerance that is one order of magnitude smaller than the “intermediate solver” tolerance

We report the performance of our “Multigrid-GMRES Preconditioned BiCGStab method” in Sect. 9. An illustration of the V-cycle for our hierarchical “Multigrid-GMRES Preconditioner” is shown in Fig. 15.

Remarks:

1. The idea of using a simpler variant of A , e.g. A_{FVM} , in the Multigrid part of the Multigrid-GMRES preconditioner was reported earlier in [20] in which the original matrix A corresponded to the indefinite matrix derived from the Helmholtz equation, and the (non-singular) matrix in the Multigrid preconditioner is derived from a shifted Helmholtz equation.
2. It is crucial that the location of the nodal points used to derive A_{FVM} correspond to the Gauss–Gauss–Gauss points and that the flux locations correspond to the Gauss–Lobatto–Gauss MAC grid points, etc. (see Fig. 1).
3. We believe that we overcome the BiCGStab round-off error difficulties described in [23, 42, 43, 56, 57] by implementing a very effective preconditioner: the “Multigrid-GMRES preconditioner. As shown in Tables 6, 10, and 11, the number of BiCGStab iterations to reach convergence is consistently less than 15.
4. In all of our simulations, the number of iteration steps of our GMRES preconditioner is four for solving the pressure equation (Sect. 5.1.1) and eight for solving the viscosity equation (Sect. 5.1.2).

Table 1 The maximum vorticity (36) and velocity (37) errors at $t = 1/2$ as a function of the space and time order for the inviscid ($\mu = 0$), smooth, advecting vortices problem

$E^{(x)}$	$E^{(t)}$	$p^{(t)}$	$p^{(x)}$	\mathcal{E}^ω	\mathcal{E}^u	u error ratio
16	98	2	2	$1.44E-1$	$1.29E-2$	
8	98	4	4	$2.13E-4$	$1.18E-5$	1093
16	192	4	4	$8.48E-6$	$3.94E-7$	29.9
4	95	8	8	$3.65E-7$	$5.83E-9$	2024
2	94	16	16	$8.35E-12$	$2.61E-13$	22337

Table 2 The maximum vorticity (36) and velocity (37) errors at $t = 1/2$ as a function of the space and time order for the viscous ($\mu = 0.01$), smooth, advecting vortices problem

$E^{(x)}$	$E^{(t)}$	$p^{(t)}$	$p^{(x)}$	\mathcal{E}^ω	\mathcal{E}^u	u error ratio
16	94	2	2	$9.15E-2$	$8.79E-3$	
8	94	4	4	$2.57E-4$	$2.04E-5$	431
4	91	8	8	$1.14E-7$	$3.96E-9$	5152
2	90	16	16	$1.47E-12$	$7.44E-14$	53226

8 Vorticity Error for Inviscid, Smooth, Advecting Vortices Problem (“Translating Taylor–Green Vortex”)

We approximate the solutions to the translating Taylor-Green Vortex problem, also presented in [54], and report the the maximum vorticity (36) and velocity (37) errors as a function of the space and time order of accuracy in Tables 1 ($\mu = 0$), 2 ($\mu = 0.01$), and 3 ($\mu = 1.0$) at $t = 1/2$:

$$\nabla \cdot \mathbf{u} = 0 \quad (34)$$

$$\mathbf{u}_t + \nabla \cdot (\mathbf{u} \otimes \mathbf{u} + p\mathbb{I} - 2\mu^M \mathbb{D}) = 0 \quad (35)$$

$$u(0, x, y) = -\sin(2\pi x) \cos(2\pi y) + 2$$

$$v(0, x, y) = \sin(2\pi y) \cos(2\pi x) + 2$$

The computational domain is the periodic unit box. The time step constraint is:

$$\max_m \Delta t_m \leq \frac{1}{2} \frac{\min \Delta x_{\ell,e} (s_1^{gl} + 1)}{2 \max |u| + |v| + |w|}$$

The exact solution is:

$$u(t, x, y) = e^{-8\pi^2\mu t} (-\sin(2\pi(x - 2t)) \cos(2\pi(y - 2t))) + 2$$

$$v(t, x, y) = e^{-8\pi^2\mu t} (\sin(2\pi(y - 2t)) \cos(2\pi(x - 2t))) + 2$$

Since the initial pressure is assumed to be unknown, we divide the initial time step by $2^{p^{(t)}-1}$ and we do first order in time discretization for the initial time step.

The errors \mathcal{E}^ω and \mathcal{E}^u are defined as:

$$\mathcal{E}^\omega = \max ||\nabla \times \mathbf{u}^{approx} - \nabla \times \mathbf{u}^{exact}||_2 \quad (36)$$

Table 3 The maximum vorticity (36) and velocity (37) errors $0 \leq t \leq 1/2$ as a function of the space and time order for the viscous ($\mu = 1.0$), smooth, advecting vortices problem

$E^{(x)}$	$E^{(t)}$	$p^{(t)}$	$p^{(x)}$	$\max_{0 \leq t \leq 1/2} \mathcal{E}^\omega$	$\max_{0 \leq t \leq 1/2} \mathcal{E}^u$	u error ratio
16	78	2	2	$3.21E-2$	$2.59E-3$	
8	94	4	4	$9.21E-5$	$9.28E-6$	279
4	91	8	8	$3.04E-8$	$1.79E-9$	5184
2	90	16	16	$1.55E-11$	$1.89E-13$	9471

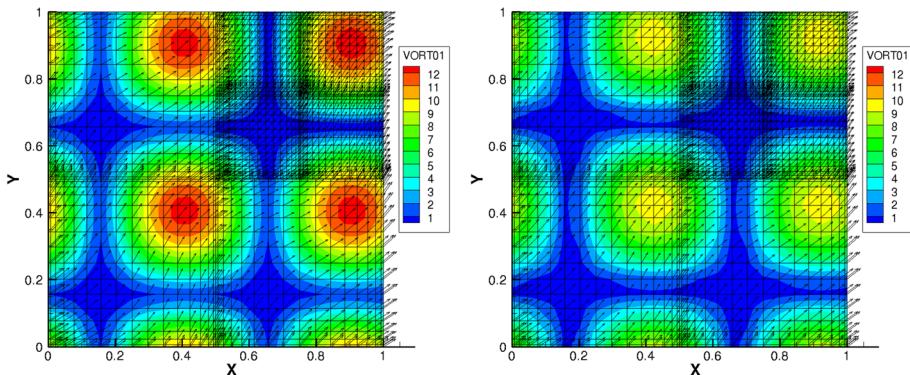


Fig. 16 Vorticity contours at $t = 0.327$ (left, $\mu = 0$) and $t = 0.336$ (right, $\mu = 0.01$) for the advection of smooth vorticity patches in a periodic domain. A stationary AMR grid patch is artificially inserted in the upper right hand quadrant of the computational domain. The space and time order is 16. There are 3 16th order coarse level elements and 4 16th order fine level elements. We note that the tecplot contour plotter assumes a piecewise linear representation (in contrast to the order 16 accuracy of our space-time algorithm) of the vorticity data, so that the contours do not represent the solution that is actually computed. The maximum vorticity error is $2.95E-8$ for the inviscid case (corresponding to the left plot), and $1.72E-9$ for the $\mu = 0.01$ case (corresponding to the right plot)

$$\mathcal{E}^u = \max ||\mathbf{u}^{approx} - \mathbf{u}^{exact}||_2 \quad (37)$$

Referring to the inviscid test results in Table 1, we note that in order to measure spectral accuracy for this test, the error ratio from order 4 to order 8 should be about $1.0E + 6$ instead of $2.0E + 3$ and the error ratio from order 8 to order 16 should be about $4.0E + 6$ instead of $2.2E + 4$. We attribute this discrepancy between theory and results due to limitations of using finite precision (double precision) arithmetic. The stopping criteria for the BiCGStab “Multigrid-GMRES preconditioner bottom solver” is that the L_2 norm of the residual falls below $1.0E-13$. The stopping criteria for the overall Multigrid-GMRES preconditioned BiCGStab solver is when the L_2 norm of the residual falls below $1.0E-11$. Already the velocity error (in the L_∞ norm!) is less than $6.0E-9$ for the eighth order space-time approximation.

In order to test our hypothesis, we simulated the fourth order case on a refined grid. This test corresponds to $E^{(x)} = 16$ in Table 1. In order to compensate for the fact that the very first time step is performed without the MISDC algorithm, the first time step for the $E^{(x)} = 16$ 4th order case is a fifth of the $E^{(x)} = 8$ 4th order case. We find that the error ratio between the $E^{(x)} = 8$ and $E^{(x)} = 16$ 4th order cases is 29.9 which indicates at least 4th order accuracy measured for the 4th order space-time algorithm.

Table 4 The maximum velocity and vorticity errors at $t = 1/2$ as a function of the space and time order for the inviscid, smooth, advecting vortices test problem and an artificially inserted, and stationary, AMR grid patch

$E_{\ell=0}^{(x)}$	$E^{(t)}$	$p^{(t)}$	$p^{(x)}$	\mathcal{E}^ω	\mathcal{E}^u	u error ratio
16	193	2	2	$4.68E-1$	$6.48E-3$	
8	192	4	4	$4.68E-2$	$4.54E-4$	14.3
16	381	4	4	$4.17E-3$	$2.63E-5$	17.3
4	187	8	8	$3.13E-4$	$2.54E-6$	178.7
2	183	16	16	$2.79E-8$	$1.79E-10$	14189.9

Table 5 The maximum velocity and vorticity errors at $t = 1/2$ as a function of the space and time order for the viscous ($\mu = 0.01$), smooth, advecting vortices test problem and an artificially inserted, and stationary, AMR grid patch

$E_{\ell=0}^{(x)}$	$E^{(t)}$	$p^{(t)}$	$p^{(x)}$	\mathcal{E}^ω	\mathcal{E}^u	u error ratio
16	186	2	2	$3.33E-1$	$3.90E-3$	
8	185	4	4	$8.04E-3$	$4.26E-5$	91.5
16	368	4	4	$1.03E-3$	$2.35E-6$	18.13
4	180	8	8	$5.88E-6$	$3.42E-8$	1245.6
2	176	16	16	$1.55E-9$	$5.27E-12$	6489.6

9 Vorticity Error and MG-GMRES BiCGStab Solver Performance with Artificially Inserted AMR Grid Patch

In this section, we report the maximum vorticity and velocity errors as in Sect. 8 except that we artificially insert an AMR grid patch in the upper right hand quadrant of the computational domain. In other-words, we have two adaptive levels for this test, $\ell = 0$ and $\ell = 1$, and the fine level occupies the region,

$$\left\{(x, y) \mid \frac{1}{2} \leq x \leq 1 \quad \frac{1}{2} \leq y \leq 1\right\}.$$

For this test problem, $p_\ell^{(x)} = p_0^{(x)} = p_1^{(x)}$. See Fig. 16. The maximum vorticity and velocity errors as a function of the order of accuracy are reported in Tables 4 ($\mu = 0$) and 5 ($\mu = 0.01$) at $t = 1/2$. Table 6 reports the performance of our Multigrid-GMRES Preconditioned BiCGStab solver which is described in Sect. 7.

We note that in order to measure spectral accuracy for the test results shown in Table 4, the error ratio from order 4 to order 8 should be about 204 instead of 178 and the error ratio from order 8 to order 16 should be about $3.2E+4$ instead of $1.4E+4$. We attribute this discrepancy between theory and results due to limitations of using finite precision (double precision) arithmetic. The extent of the discrepancy is not as large as that reported in Sect. 8 since the errors in this section do not approach zero as rapidly. Nonetheless, the maximum velocity error for the space-time order 16 case is $1.8E-10$ for which we hypothesize again that this error is effected by numerical round-off error.

In order to test our hypothesis, we simulated the fourth order case on a refined grid. This test corresponds to $E^{(x)} = 16$ in Table 4. In order to compensate for the fact that the very first

Table 6 Performance statistics for our hierarchical Multigrid-GMRES preconditioned BiCGStab method (described in Sect. 7) for solving the large sparse matrix system that arises from discretizing the pressure projection equation [see (12) or (24)]

$E_{\ell=0}^{(x)}$	$p^{(t)}$	$p^{(x)}$	#iter (top) max,median #iter (bot)	1 core, mpic++/mpif90 (g++/gfortran) (s) 4 cores, mpic++/mpif90 (s)
32	2	2	4	6.1 (5.1) (s)
			4, 3	2.1 (s)
16	4	4	5	5.6 (4.9) (s)
			5, 4	2.0 (s)
8	8	8	5	8.3 (8.0) (s)
			6, 5	2.9 (s)
4	16	16	12	33.0 (31.6) (s)
			8, 7	11.5 (s)

The statistics are measured at time $t = 0.05$ for the smooth advecting vortices test problem illustrated in Fig. 16. The total number of sub-elements on level $\ell = 0$ is always 64×64 and the total number of sub-elements on level $\ell = 1$, always located in the upper right hand quadrant of the computational domain, is also 64×64 . The stopping criterion for our iterative algorithm is when the $L2$ norm of the residual is below $1.0e-11$.

time step is performed without the MISDC algorithm, the first time step for the $E^{(x)} = 16$ 4th order case is a fifth of the $E^{(x)} = 8$ 4th order case. We find that the error ratio between the $E^{(x)} = 8$ and $E^{(x)} = 16$ 4th order cases is 17.2 which indicates 4th order accuracy measured for the velocity corresponding to the 4th order space-time algorithm.

10 Break-Up of Inviscid Vortex Patch into Smaller Vortex Patches

In this section we analyze our space-time spectral element method for the problem of the break-up of a vortex patch in a periodic unit box. We approximate solutions to the inviscid incompressible Euler equations in two dimensions [see (34) and (35)]. The initial conditions are,

$$\epsilon = \frac{1}{30}$$

$$r = \sqrt{(x - 1/2)^2 + (y - 1/2)^2} - 1/4$$

$$\alpha = \frac{1 - \tanh(\frac{r}{\epsilon})}{2}$$

$$u(0, x, y) = \alpha(y - 1/2) \quad (38)$$

$$v(0, x, y) = -\alpha(x - 1/2). \quad (39)$$

For the case of $\epsilon = 0$, which is an idealized circular vortex patch, it is predicted by [13, 15, 16] that infinitesimally small perturbations in the patch boundary will lead to uncontrolled growth in the patch boundary perimeter. It is also indicated by [13] that the patch area will be preserved. In Fig. 17, $\epsilon = 1/30$, we show the vorticity magnitude contour plots computed using our eighth order space-time method. The initial vortex patch breaks up into 5 smaller patches. The plots for the second and fourth order space-time methods are qualitatively very similar to the eighth order method that we show.

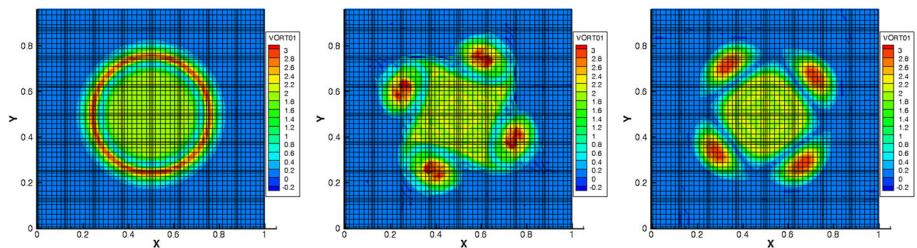


Fig. 17 Computed evolution of the vorticity magnitude in which the initial velocity field is given by (38) and (39). The times for the plots are $t = 0$, $t = 15.7$, and $t = 40.0$. The simulation was made using the 8th order space and time algorithm in which there were 8×8 space elements. The total kinetic energy and vorticity integral fluctuates at most 0.12% and 0.25%, respectively, throughout the whole simulation

In Table 7 we report the maximum fluctuation in the overall kinetic energy, $E_k(\mathbf{u})$ (40),

$$E_k(\mathbf{u}) = \int_{\Omega} \frac{1}{2} \rho |\mathbf{u}|^2 dx, \quad (40)$$

and vorticity integral (41),

$$I_{\omega} = \int_{\Omega} \omega dx, \quad (41)$$

as a function of the space and time order for the time period $0 \leq t \leq 40$.

In Fig. 18 we plot the Enstrophy, $\text{En}(\omega)$ (42),

$$\text{En}(\omega) = \int_{\Omega} |\omega|^2 dx, \quad (42)$$

versus time and we observe that the Enstrophy error improves with increasing space-time accuracy, but not nearly at the same rate as that reported for the smooth test problem reported in Sect. 8 (in Sect. 8, the maximum vorticity error falls below computer round-off error by the 16th order).

Remarks:

1. The highest rate of Enstrophy loss occurs during the period when the original vortex breaks up into smaller vortices. The rate of Enstrophy loss after the break-up phase is relatively very low, the size of the smaller vortices being smaller than an element notwithstanding (see the $t = 40.0$ results in Fig. 17). The percent loss of Enstrophy for $20 < t < 40$ for the fourth and eighth order cases was two percent, during which time the small satellite vortices traveled one full revolution which is 13 times their length.
2. [46] have developed an Enstrophy preserving method by solving numerically an equation for velocity \mathbf{u} , vorticity ω , and pressure p . The velocity \mathbf{u} and vorticity ω in the formulation by [46] are decoupled. They did not report the difference between $\nabla \times \mathbf{u}$ and ω .

11 Single Phase Inviscid Jet in Doubly Periodic Geometry

In this example we report numerical results, as a function of the space-time order, for the problem of a single phase inviscid jet in doubly periodic geometry [7]. We approximate

Table 7 Errors for the simulation of the break-up of a single vortex patch into multiple smaller vortex patches

$E^{(x)}$	$E^{(t)}$	$p^{(t)}$	$p^{(x)}$	Max fluctuation of $E_k(\mathbf{u})$ (%)	Max fluctuation of I_ω (%)
32	822	2	2	0.8	0.03
16	823	4	4	0.16	0.2
8	798	8	8	0.12	0.25

The maximum fluctuation of the total kinetic energy $E_k(\mathbf{u})$ (40) and vorticity integral I_ω (41), $0 \leq t \leq 40$, as a function of the space-time order are reported

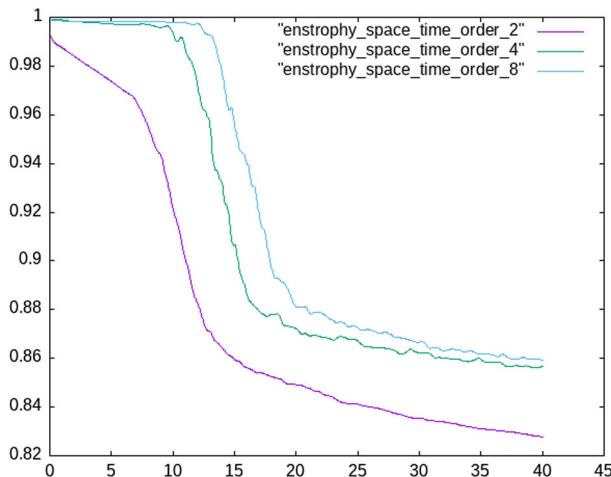


Fig. 18 Plot of the Enstrophy versus time for the break up of a vortex patch into smaller patches. The initial velocity field is given by (38) and (39). The vertical axis is the total Enstrophy [see (42)]

solutions to the inviscid incompressible Euler equations in two dimensions [see (34) and (35)]. The initial conditions are,

$$\rho = \frac{1}{30} \quad \delta = \frac{1}{20}$$

$$u(0, x, y) = \begin{cases} \tanh\left(\frac{y-\frac{1}{4}}{\rho}\right) & y \leq \frac{1}{2} \\ \tanh\left(\frac{\frac{3}{4}-y}{\rho}\right) & y > \frac{1}{2} \end{cases} \quad (43)$$

$$v(0, x, y) = \delta \sin(2\pi x) \quad (44)$$

In Fig. 19, we show the vorticity magnitude contour plots computed using our eighth order space-time method for the cases when there are 16×16 and 32×32 elements respectively.

In Table 8 we report the maximum fluctuation in the overall kinetic energy, $E_k(\mathbf{u})$ (40), and vorticity integral, I_ω (41), as a function of the space and time order for the time period $0 \leq t \leq 2$.

In Fig. 20 we plot the Enstrophy, $\text{En}(\omega)$ (42), versus time and we observe that the Enstrophy error improves with increasing space-time accuracy, but not nearly at the same rate as that reported for the smooth test problem reported in Sect. 8.

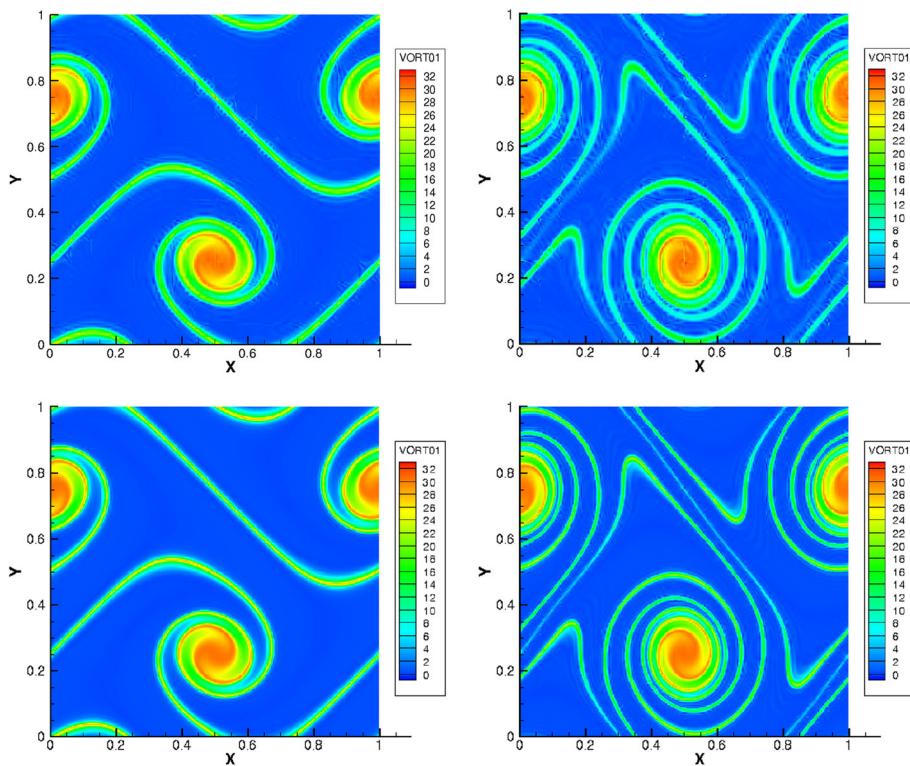


Fig. 19 Computed evolution of the vorticity magnitude in which the initial velocity field is given by (43) and (44). The times for the plots are $t = 1.2$ and $t = 1.8$. The simulation was made using the 8th order space and time algorithm in which there were 16×16 (top) and 32×32 (bottom) space elements respectively. The total kinetic energy fluctuates at most 0.018% for both of the resolutions

Table 8 Errors for the simulation of the roll-up of a vortex sheet

$E^{(x)}$	$E^{(t)}$	$p^{(t)}$	$p^{(x)}$	Max fluctuation of $E_k(\mathbf{u})$ (%)	Max fluctuation of I_ω (%)
16	523	8	8	0.018	2.5E-12
128	1073	2	2	0.023	2.0E-12
64	1072	4	4	0.0046	4.0E-12
32	1065	8	8	0.0023	4.0E-12

The maximum fluctuation of the total kinetic energy $E_k(\mathbf{u})$ (40) and vorticity integral I_ω (41), $0 \leq t \leq 2$, as a function of the space-time order are reported

12 Vortex Shedding for Flow Past a Reverse Slanted Cone

In this example we report numerical results, as a function of the space-time order, for the problem of vortex shedding past a tilted cone (see Figs. 21, 22 and 23). The computational domain dimensions for the two dimensional case is,

$$-1 \leq x \leq 1 \quad 0 \leq y \leq 1,$$

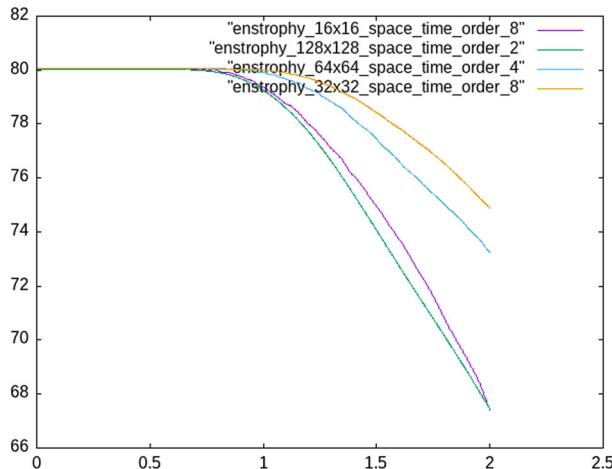


Fig. 20 Plot of the Enstrophy versus time for the roll up of a vortex sheet. The initial velocity field is given by (43) and (44). The vertical axis is the total Enstrophy [see (42)]

and the computational domain dimensions for the three dimensional case is,

$$-1 \leq x \leq 1 \quad 0 \leq y \leq 1 \quad 0 \leq z \leq 1/2.$$

The boundary condition between the cone and the fluid is the no-slip condition $\mathbf{u} = 0$. For the two dimensional problem, the boundary conditions are $u = u_\infty$ and $v = 0$ at all domain walls. For the the three dimensional problem, the boundary conditions are $u = u_\infty$, $v = 0$, and $w = 0$ at the $x = -1$, $x = 1$, $y = 0$, $y = 1$, and $z = 1/2$ walls, and symmetric boundary conditions at the $z = 0.0$ wall.

The initial velocity field for the 2D or 3D simulations is generated by first initializing the velocity field to be $\mathbf{u}^* = (u_\infty, 0, 0)$ and then numerically solving,

$$\begin{aligned} \nabla \cdot \nabla p &= \nabla \cdot \mathbf{u}^* \\ \mathbf{u}(t = 0, \mathbf{x}) &= \mathbf{u}^* - \nabla p. \end{aligned} \tag{45}$$

The boundary conditions in (45) at the solid-fluid boundary is

$$\nabla p \cdot \mathbf{n} = 0$$

where \mathbf{n} is the outward facing normal to the cone. We use a very simple low order discretization at the fluid cone boundary [4].

The physical properties prescribed for this test problem are,

$$u_\infty = 0.2143 \frac{\text{m}}{\text{s}} \quad \mu = 1.825 \frac{\text{kg}}{\text{ms}} \quad \rho = 1.204 \frac{\text{kg}}{\text{m}^3}$$

Table 9 and Figs. 21 and 22 compare the computed vorticity magnitude for two dimensional simulations using the first, second, and fourth order space time methods respectively. Table 10 reports the performance of our Multigrid-GMRES Preconditioned BiCGStab solver which is described in Sect. 7.

In Fig. 23 we compare the computed vorticity magnitude for three dimensional simulations using the first and second order space time methods respectively.

For all simulations, the low order CISL-MOF method is used in mixed material elements or elements that neighbor a mixed material element. For the first order simulations in Figs. 21

Fig. 21 Vorticity magnitude contours at $t = 3.68$ for vortices shed from an angled cone in two dimensions. Top: space and time order is one for both levels $\ell = 0$ and $\ell = 1$. Middle: space and time order is two for both levels $\ell = 0$ and $\ell = 1$. Bottom: space and time order is four for both levels $\ell = 0$ and $\ell = 1$. The criteria for adaptivity is that either the cone level set function changes sign or the vorticity magnitude exceeds $|\omega| > 4$ for the first order case or exceeds $|\omega| > 10$ for the second and fourth order cases. Note the order in space and time is always one in the multi-material/mixed CISL-MOF elements

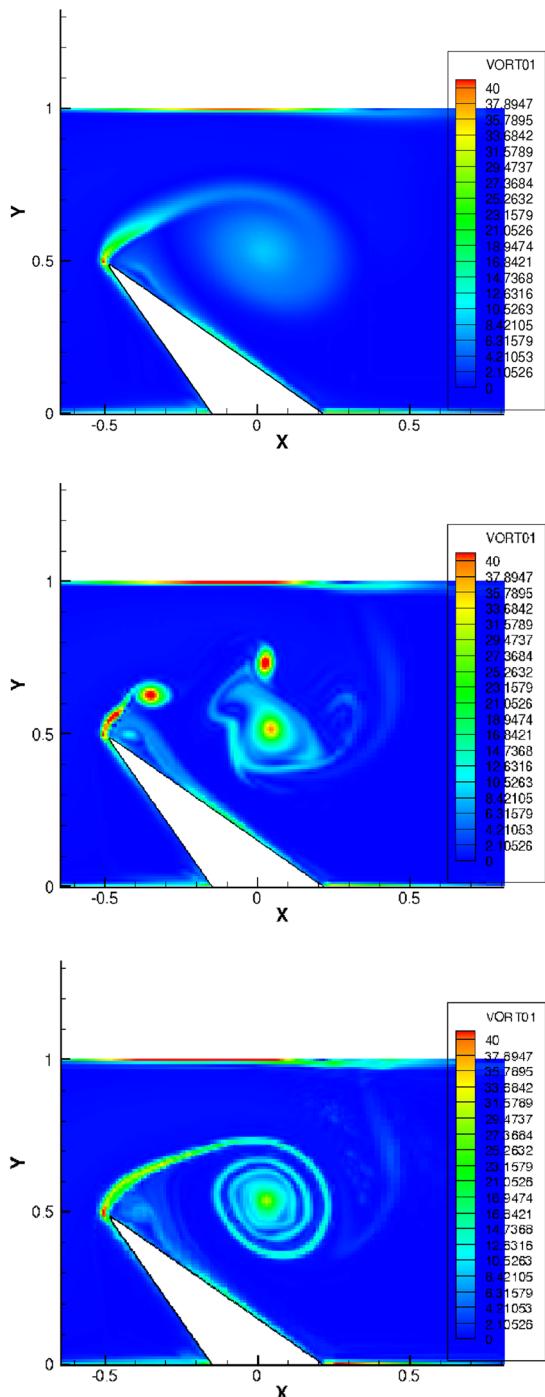
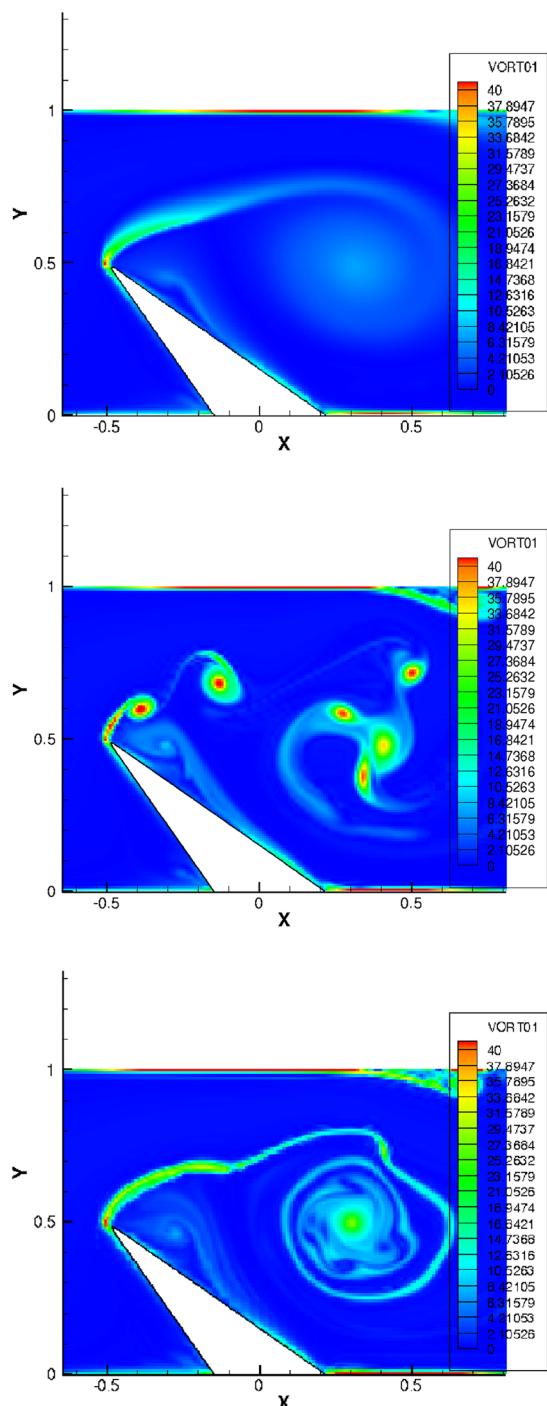


Fig. 22 Vorticity magnitude contours at $t = 5.53$ for vortices shed from an angled cone in two dimensions. Top: space and time order is one for both levels $\ell = 0$ and $\ell = 1$. Middle: space and time order is two for both levels $\ell = 0$ and $\ell = 1$. Bottom: space and time order is four for both levels $\ell = 0$ and $\ell = 1$. The criteria for adaptivity is that either the cone level set function changes sign or the vorticity magnitude exceeds $|\omega| > 4$ for the first order case or exceeds $|\omega| > 10$ for the second and fourth order cases. Note the order in space and time is always one in the multi-material/mixed CISL-MOF elements



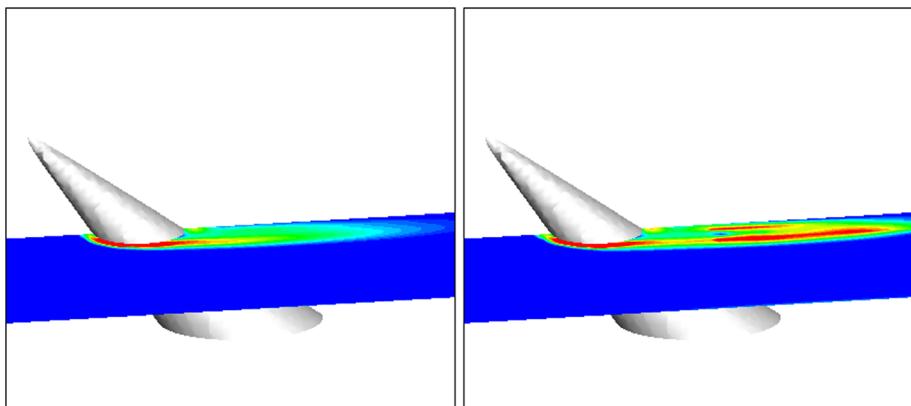


Fig. 23 Vorticity magnitude contours at $t = 20.0$ for vortices shed from an angled cone in three dimensions. Left: space and time order is one. Right: space and time order is two. There are 20 vorticity magnitude contours ranging from 1 to 5. Note the order in space and time is always one in the multi-material/mixed CISL-MOF elements

Table 9 Maximum vorticity magnitude of shedded vortex patch at $t = 3.68$ and $t = 5.53$, $\max_{x \in \Omega_{patch}} \|\omega\|_2$, for flow past an angled cone obstacle in two dimensions

$E_{\ell=0}^{(x)}$	$E_{\ell=0}^{(y)}$	$E^{(t)}, 0 \leq t \leq 5.53$	$p^{(t)}$	$p_{\ell=0}^{(x)}$	$p_{\ell=1}^{(x)}$	$t = 3.68$	$t = 5.53$
128	64	1180	1	1	1	8.85	7.09
64	32	1201	2	2	2	49.3, 40.9	45.7, 41.7, 41.7, 36.2
32	16	1050	4	4	4	30.6	28.5

and 22, there are 128×64 elements on level $\ell = 0$ and the CISL-MOF method is used everywhere on both levels. For the second order simulations in the same figures, there are 64×32 second order elements on level $\ell = 0$ and second order elements are used on level $\ell = 1$ too. For the fourth order simulation, there are 32×16 fourth order elements on level $\ell = 0$ and fourth order elements are used on level $\ell = 1$.

For the three dimensional simulations (Fig. 23), we do not use AMR, and there are either $128 \times 64 \times 32$ elements (first order) or $64 \times 32 \times 16$ elements (second order).

13 The Break Up of a Liquid Jet into Droplets Due to Cross Flowing Air

In this section we compare the simulation results using the first order CISL-MOF algorithm to the second order space-time algorithm for the break up of a liquid jet in cross flow. See Fig. 24 for an illustration of the bending liquid jet and a cross-section of the vorticity field. Table 11 reports the performance of our Multigrid-GMRES Preconditioned BiCGStab solver which is described in Sect. 7.

We computed this problem previously in [25] using the CISL-MOF method and related simulations have been reported by [22,36].

The dimensions of the computational domain are,

$$0 \leq x \leq 5.12 \text{ cm} \quad 0 \leq y \leq 1.28 \text{ cm} \quad 0 \leq z \leq 2.56 \text{ cm},$$

Table 10 Performance statistics for our hierarchical Multigrid-GMRES preconditioned BiCGStab method (described in Sect. 7) for solving the large sparse matrix system that arises from discretizing the pressure projection equation [see (12) or (24)]

	$E_{\ell=0}^{(x)}$	$p^{(t)}$	$p_{\ell=0}^{(x)}$	$p_{\ell=1}^{(x)}$	#iter (top) max,median #iter (bot)
128	1	1	1	3	7, 5
					4
64	2	2	2	4	8, 5
					4
32	4	4	4	8, 7	8, 7

The statistics are measured at time $t = 5.53$ for the 2D cone problem illustrated in Fig. 22. The total number of sub-elements on level $\ell = 0$ is always 128×64 and level $\ell = 0$ cells are tagged for adaptivity if either the solid level set function changes sign or the vorticity exceeds $\omega = 10$. The stopping criterion for our iterative algorithm is when the $L2$ norm of the residual is below $1.0e-11$

the surface tension coefficient is 24.0 dyne/cm, the liquid and gas viscosity coefficients are 0.013 and 0.000142 g/(cm s) respectively, the liquid and gas density are 0.78 and 0.00125 g/cm³ respectively, the diameter of the liquid orifice at $(x, y, z) = (1.28 \text{ cm}, 0.0 \text{ cm}, 0.0 \text{ cm})$ is $d_{liq} = 0.0762 \text{ cm}$, the gas inflow velocity at $x = 0$ is $u_g = 6900 \text{ cm/s}$, and the liquid inflow velocity at the liquid orifice is $u_l = 1165 \text{ cm/s}$. These parameters correspond to a Weber number defined as,

$$We = \frac{\rho_g u_g^2 d_{liq}}{\sigma} = 189.0.$$

The boundary conditions for the computational domain are inflow at $x = 0$, outflow at $(x, y, z) = (5.12 \text{ cm}, 1.28 \text{ cm}, 2.56 \text{ cm})$, reflecting at $y = 0.0$, and inflow at $z = 0$ where the orifice is.

In the simulations used to create the results shown in Fig. 24, the base $\ell = 0$ grid had $64 \times 16 \times 32$ first order elements for the low order simulation and $32 \times 8 \times 16$ second order elements for the second order simulation. There were 4 levels of adaptivity.

We conjecture that the second order results shown in Fig. 24 are more realistic than the first order results. The second order results preserve the vortical behavior of the gas flow beneath the bended jet whereas the first order results are much more diffused. We remark that [36] used finer meshes and captured similar dynamics of the gas in comparison to our second order results. The gas flow is important as it influences the stability of the liquid jet as is apparent from our results.

14 Summary and Conclusions

A novel hybrid space-time spectral method and CISL-MOF method has been developed for simulating complex multiphase/multi-material flows on a dynamic adaptive mesh refinement grid. For smooth problems, the new method simulates flows in which the maximum vorticity error is governed by computer round-off error rather than error in the method. See Tables 1, 2, 3, 4, and 5. For problems in which vortex patches break-up or otherwise undergo large deformations, the maximum fluctuation of kinetic energy and vorticity is below 0.25% for all cases. See Tables 7 and 8. For the period in which vortices stay intact, we observed

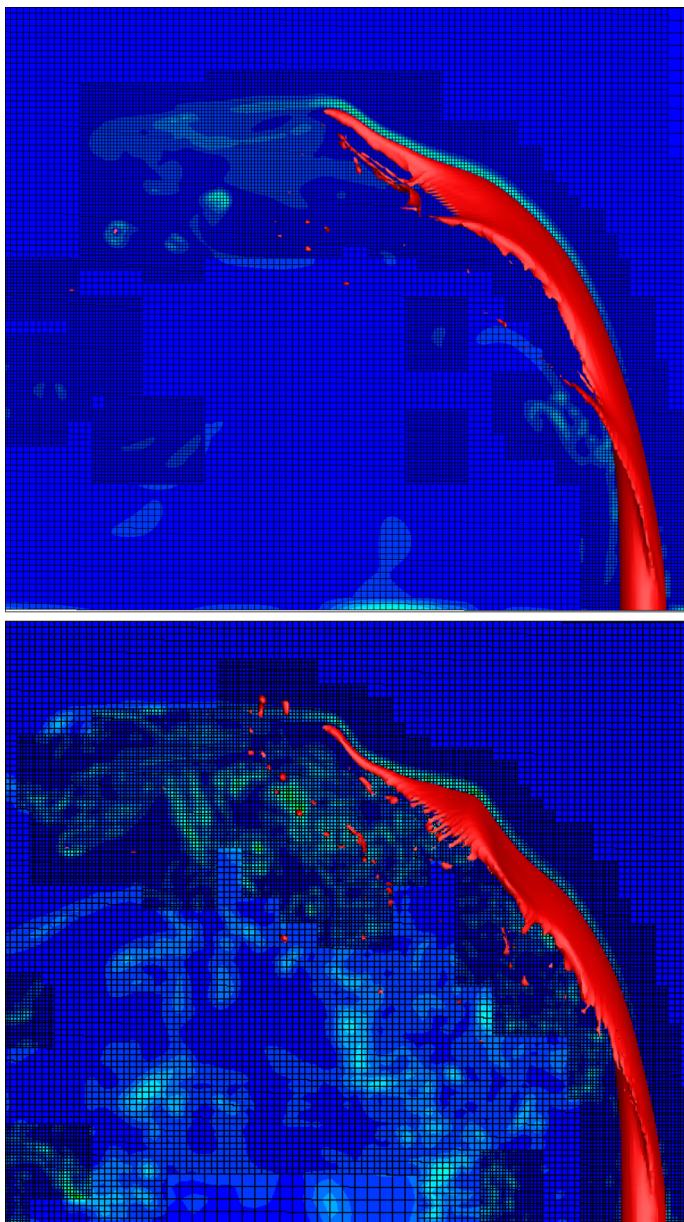


Fig. 24 A comparison of the gas vorticity magnitude at $t = 0.000684$ for the break up of a liquid jet (red isosurface in the plot) in a gas cross flow. Top: space and time order is one. Bottom: space and time order is two. Note the order in space and time is always one in the multi-material/mixed CISL-MOF elements (Color figure online)

Table 11 Performance statistics for our hierarchical Multigrid-GMRES preconditioned BiCGStab method (described in Sect. 7) for solving the large sparse matrix system that arises from discretizing the pressure projection equation [see (12) or (24)]

$p^{(t)}$	$p_{\ell=0}^{(x)}$	$p_{\ell=4}^{(x)}$	#iter (top) max,median #iter (bot)
1	1	1	4
			3, 3
2	2	2	5
			3, 3

The statistics are measured at time $t = 0.000684$ for the 3D Liquid jet in a cross-flow problem illustrated in Fig. 24. The total number of sub-elements on level $\ell = 0$ is always $64 \times 16 \times 32$ and the fraction of the computational domain covered by the finest level ($\ell = 4$) is 1.3%. The stopping criterion for our iterative algorithm is when the L_2 norm of the residual is below $1.0e-11$

less than 0.15% Enstrophy fluctuation for a vortex resolved by 4×4 eighth order elements (see Fig. 18, $0 < t < 10$) and less than 2.0% Enstrophy fluctuation for satellite vortices resolved by just one eighth order element [see Figs. 18 ($20 < t < 40$) and 17 ($t = 40$)]. For the multi-material and multi-phase flow tests (see Table 9 and Figs. 21, 22, 23, and 24), the error in vortex magnitude consistently improves with increasing space-time order of accuracy. For the problem of the break-up of a liquid jet in a gas cross-flow (Fig. 24), the ability to capture the proper vortical flow dynamics in the gas enhances the numerical methods' ability to predict the subtle wavy structure of the liquid jet prior to its breakup.

The largest Enstrophy errors occur when the diameter or filament size of a vortex patch is below the size of a sub-element. See Figs. 18, 17, 20, and 19. As is shown in Fig. 20 (for the vortex stretching problem illustrated in Fig. 19), when the mesh is refined from 16×16 eighth order elements to 32×32 elements, the Enstrophy conservation error is greatly improved for the time interval when the vortex patch becomes thinner than a sub-element.

If preserving total enstrophy is the goal, then our results indicate that it is more efficient to increase the resolution, rather than the order (i.e. second order is best), in order to prevent artificial enstrophy fluctuations. On the otherhand, if it is important to preserve the proper distribution of kinetic energy amongst varying flow modes, then we see from Tables 1, 4, and Figs. 21 and 22 that it is optimal to increase the order, not the resolution.

In the future, we plan to extend our space-time spectral algorithm in order to simulate compressible multi-phase/multi-material flows (we refer the reader to our low order compressible multiphase flow algorithm [25]). A space-time spectrally accurate method for multiphase/multi-material compressible flows would have the following benefits:

1. Improved capturing of vortical structures in compressible materials.
2. Improved prediction of acoustic wave dynamics in multiphase flows.
3. One can extract modal information from spectral space-time data in order to detect shock-genesis [50]. Trouble elements that contain a shock can be tagged for adaptive mesh refinement and/or converted to a piecewise finite volume element (see Fig. 6).

Finally, we plan to study in the future the interaction of acoustic waves, shock waves, and vortical structures with a deforming multi-material/multi-phase interface that is changing phase.

Acknowledgements This work and the authors were supported in part by the National Science Foundation under Contract DMS 1418983.

References

1. Abbassi, H., Mashayek, F., Jacobs, G.B.: Shock capturing with entropy-based artificial viscosity for staggered grid discontinuous spectral element method. *Comput. Fluids* **98**, 152–163 (2014)
2. Almgren, A.S., Aspden, A.J., Bell, J.B., Minion, M.L.: On the use of higher-order projection methods for incompressible turbulent flow. *SIAM J. Sci. Comput.* **35**(1), B25–B42 (2013)
3. Almgren, A.S., Bell, J.B., Colella, P., Howell, L.H., Welcome, M.L.: A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations. *J. Comput. Phys.* **142**(1), 1–46 (1998)
4. Arienti, M., Sussman, M.: An embedded level set method for sharp-interface multiphase simulations of diesel injectors. *Int. J. Multiph. Flow* **59**, 1–14 (2014)
5. Bao, W., Jin, S.: Weakly compressible high-order i-stable central difference schemes for incompressible viscous flows. *Comput. Methods Appl. Mech. Eng.* **190**(37), 5009–5026 (2001)
6. Bell, J., Berger, M., Saltzman, J., Welcome, M.: Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comput.* **15**(1), 127–138 (1994)
7. Bell, J.B., Colella, P., Glaz, H.M.: A second-order projection method for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **85**(2), 257–283 (1989)
8. Berger, M., Rigoutsos, I.: An algorithm for point clustering and grid generation. *IEEE Trans. Syst. Man Cybern.* **21**(5), 1278–1286 (1991)
9. Bourlioux, A., Layton, A.T., Minion, M.L.: High-order multi-implicit spectral deferred correction methods for problems of reactive flow. *J. Comput. Phys.* **189**(2), 651–675 (2003)
10. Brown, R.E.: Rotor wake modeling for flight dynamic simulation of helicopters. *AIAA J.* **38**(1), 57–63 (2000)
11. Brown, R.E., Line, A.J.: Efficient high-resolution wake modeling using the vorticity transport equation. *AIAA J.* **43**(7), 1434–1443 (2005)
12. Chatelain, P., Curioni, A., Bergdorf, M., Rossinelli, D., Andreoni, W., Koumoutsakos, P.: Billion vortex particle direct numerical simulations of aircraft wakes. *Comput. Methods Appl. Mech. Eng.* **197**(13–16), 1296–1304 (2008)
13. Constantin, P., Titi, E.: On the evolution of nearly circular vortex patches. *Commun. Math. Phys.* **119**(2), 177–198 (1988)
14. Don, W.S., Gao, Z., Li, P., Wen, X.: Hybrid compact-weno finite difference scheme with conjugate Fourier shock detection algorithm for hyperbolic conservation laws. *SIAM J. Sci. Comput.* **38**(2), A691–A711 (2016)
15. Dou, H.S.: Stability of rotating viscous and inviscid flows. arXiv preprint physics/0503083 (2005)
16. Dou, H.S.: Mechanism of flow instability and transition to turbulence. *Int. J. Nonlinear Mech.* **41**(4), 512–517 (2006). <https://doi.org/10.1016/j.ijnonlinmec.2005.12.002>
17. Dubey, A., Almgren, A., Bell, J., Berzins, M., Brandt, S., Bryan, G., Colella, P., Graves, D., Lijewski, M., Loffler, F., O’Shea, B., Schnetter, E., Straalen, B.V., Weide, K.: A survey of high level frameworks in block-structured adaptive mesh refinement packages. *J. Parallel Distrib. Comput.* **74**(12), 3217–3227 (2014)
18. Duffy, A., Kuhnle, A., Sussman, M.: An improved variable density pressure projection solver for adaptive meshes (2012). <https://www.math.fsu.edu/~sussman/MGAMR.pdf>. Accessed 5 Apr 2011
19. Dumbser, M., Zanotti, O., Hidalgo, A., Balsara, D.S.: Ader-weno finite volume schemes with space-time adaptive mesh refinement. *J. Comput. Phys.* **248**, 257–286 (2013). <https://doi.org/10.1016/j.jcp.2013.04.017>
20. Erlangga, Y.A., Oosterlee, C.W., Vuik, C.: A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.* **27**(4), 1471–1492 (2006)
21. Fambri, F., Dumbser, M.: Spectral semi-implicit and space-time discontinuous Galerkin methods for the incompressible Navier–Stokes equations on staggered Cartesian grids. *Appl. Numer. Math.* **110**, 41–74 (2016). <https://doi.org/10.1016/j.apnum.2016.07.014>
22. Garrick, D.P., Hagen, W.A., Regele, J.D.: An interface capturing scheme for modeling atomization in compressible flows. *J. Comput. Phys.* **344**, 260–280 (2017)
23. Itoh, S., Namekawa, Y.: An improvement in DS-BICGstab (l) and its application for linear systems in lattice QCD. *J. Comput. Appl. Math.* **159**(1), 65–75 (2003)
24. Jacobs, G.B., Kopriva, D.A., Mashayek, F.: A conservative isothermal wall boundary condition for the compressible Navier–Stokes equations. *J. Sci. Comput.* **30**(2), 177–192 (2007)
25. Jemison, M., Sussman, M., Arienti, M.: Compressible, multiphase semi-implicit method with moment of fluid interface representation. *J. Comput. Phys.* **279**, 182–217 (2014)
26. Kadioglu, S.Y., Klein, R., Minion, M.L.: A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics. *J. Comput. Phys.* **227**(3), 2012–2043 (2008)

27. Kadioglu, S.Y., Sussman, M.: Adaptive solution techniques for simulating underwater explosions and implosions. *J. Comput. Phys.* **227**(3), 2083–2104 (2008)
28. Kamkar, S., Wissink, A., Sankaran, V., Jameson, A.: Feature-driven Cartesian adaptive mesh refinement for vortex-dominated flows. *J. Comput. Phys.* **230**(16), 6271–6298 (2011). <https://doi.org/10.1016/j.jcp.2011.04.024>
29. Klaij, C.M., van der Vegt, J.J.W., van der Ven, H.: Space-time discontinuous Galerkin method for the compressible Navier–Stokes equations. *J. Comput. Phys.* **217**(2), 589–611 (2006)
30. Lakehal, D.: Status and future developments of large-Eddy simulation of turbulent multi-fluid flows (leis and less). *Int. J. Multiph. Flow* **104**, 322–337 (2018). <https://doi.org/10.1016/j.ijmultiphaseflow.2018.02.018>
31. Lalanne, B., Rueda Villegas, L., Tanguy, S., Risso, F.: On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method. *J. Comput. Phys.* **301**, 289–307 (2015)
32. Layton, A.T.: On the choice of correctors for semi-implicit Picard deferred correction methods. *Appl. Numer. Math.* **58**(6), 845–858 (2008)
33. Layton, A.T.: On the efficiency of spectral deferred correction methods for time-dependent partial differential equations. *Appl. Numer. Math.* **59**(7), 1629–1643 (2009)
34. Layton, A.T., Minion, M.L.: Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics. *J. Comput. Phys.* **194**(2), 697–715 (2004). <https://doi.org/10.1016/j.jcp.2003.09.010>
35. Li, G., Lian, Y., Guo, Y., Jemison, M., Sussman, M., Helms, T., Arienti, M.: Incompressible multiphase flow and encapsulation simulations using the moment-of-fluid method. *Int. J. Numer. Methods Fluids* **79**(9), 456–490 (2015). <https://doi.org/10.1002/fld.4062>
36. Li, X., Soteriou, M.C.: High fidelity simulation and analysis of liquid jet atomization in a gaseous crossflow at intermediate weber numbers. *Phys. Fluids* **28**(8), 082,101 (2016)
37. Liovic, P., Lakehal, D.: Interface-turbulence interactions in large-scale bubbling processes. *Int. J. Heat Fluid Flow* **28**(1), 127–144 (2007). <https://doi.org/10.1016/j.ijheatfluidflow.2006.03.003>
38. Liovic, P., Rudman, M., Liow, J.L., Lakehal, D., Kothe, D.: A 3d unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction. *Comput. Fluids* **35**(10), 1011–1032 (2006). <https://doi.org/10.1016/j.compfluid.2005.09.003>
39. Liu, J.G., Shu, C.W.: A high-order discontinuous Galerkin method for 2D incompressible flows. *J. Comput. Phys.* **160**(2), 577–596 (2000)
40. Liu, J.G., Wang, W.C.: Energy and helicity preserving schemes for hydro-and magnetohydro-dynamics flows with symmetry. *J. Comput. Phys.* **200**(1), 8–33 (2004)
41. McInnes, L.C., Smith, B., Zhang, H., Mills, R.T.: Hierarchical Krylov and nested Krylov methods for extreme-scale computing. *Parallel Comput.* **40**(1), 17–31 (2014)
42. Miyauchi, T., Itoh, S., Zhang, S.L., Natori, M.: Dynamic selection of l for BI-CGstab (l). *Trans. Jpn. Soc. Ind. Appl. Math.* **11**(2), 49–62 (2001)
43. Montagnac, M., Chesneaux, J.M.: Dynamic control of a BiCGSTab algorithm. *Appl. Numer. Math.* **32**(1), 103–117 (2000)
44. Morinishi, Y., Lund, T., Vasilyev, O., Moin, P.: Fully conservative higher order finite difference schemes for incompressible flow. *J. Comput. Phys.* **143**(1), 90–124 (1998). <https://doi.org/10.1006/jcph.1998.5962>
45. Nonaka, A., Bell, J., Day, M., Gilet, C., Almgren, A., Minion, M.: A deferred correction coupling strategy for low mach number flow with complex chemistry. *Combust. Theory Model.* **16**, 1053–1088 (2012)
46. Palha, A., Gerritsma, M.: A mass, energy, enstrophy and vorticity conserving (MEEVC) mimetic spectral element discretization for the 2d incompressible Navier–Stokes equations. *J. Comput. Phys.* **328**, 200–220 (2017)
47. Pazner, W.E., Nonaka, A., Bell, J.B., Day, M.S., Minion, M.L.: A high-order spectral deferred correction strategy for low Mach number flow with complex chemistry. *Combust. Theory Model.* **20**(3), 521–547 (2016)
48. Pei, C., Sussman, M., Hussaini, M.Y.: A space-time discontinuous Galerkin spectral element method for the Stefan problem. *Discrete Contin. Dyn. Syst. B* **23**(9), 3595–3622 (2018)
49. Pei, C., Sussman, M., Hussaini, M.Y.: New multi-implicit space-time spectral element methods for advection–diffusion–reaction problems. *J. Sci. Comput.* **78**(2), 653–686 (2019). <https://doi.org/10.1007/s10915-018-0654-5>
50. Pei, C., Sussman, M., Hussaini, M.Y.: A space-time discontinuous galerkin spectral element method for nonlinear hyperbolic problems. *Int. J. Comput. Methods* **16**(01), 1850,093 (2019)
51. Rhebergen, S., Cockburn, B., van der Vegt, J.J.W.: A space-time discontinuous Galerkin method for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **233**, 339–358 (2013)
52. Saad, Y.: A flexible inner–outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.* **14**(2), 461–469 (1993)

53. Saye, R.: Implicit mesh discontinuous galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid-structure interaction, and free surface flow: part I. *J. Comput. Phys.* **344**, 647–682 (2017)
54. Saye, R.: Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: part II. *J. Comput. Phys.* **344**, 683–723 (2017)
55. Scardovelli, R., Zaleski, S.: Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection. *Int. J. Numer. Methods Fluids* **41**(3), 251–274 (2003)
56. Sleijpen, G.L., Fokkema, D.R.: BiCGStab (I) for linear equations involving unsymmetric matrices with complex spectrum. *Electron. Trans. Numer. Anal.* **1**(11), 2000 (1993)
57. Sleijpen, G.L., Van der Vorst, H.A.: Maintaining convergence properties of BiCGStab methods in finite precision arithmetic. *Numer. Algorithms* **10**(2), 203–223 (1995)
58. Sollie, W.E.H., Bokhove, O., van der Vegt, J.J.W.: Space-time discontinuous Galerkin finite element method for two-fluid flows. *J. Comput. Phys.* **230**(3), 789–817 (2011)
59. Srinivasan, G., McCroskey, W., Baeder, J., Edwards, T.: Numerical simulation of tip vortices of wings in subsonic and transonic flows. *AIAA J.* **26**(10), 1153–1162 (1988)
60. Steinhoff, J., Underhill, D.: Modification of the euler equations for “vorticity confinement”: application to the computation of interacting vortex rings. *Phys. Fluids* **6**(8), 2738–2744 (1994)
61. Stewart, P., Lay, N., Sussman, M., Ohta, M.: An improved sharp interface method for viscoelastic and viscous two-phase flows. *J. Sci. Comput.* **35**(1), 43–61 (2008)
62. Sussman, M., Almgren, A.S., Bell, J.B., Colella, P., Howell, L.H., Welcome, M.L.: An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.* **148**(1), 81–124 (1999)
63. Tan, F.J., Wen Wang, H.: Simulating unsteady aerodynamics of helicopter rotor with panel/viscous vortex particle method. *Aerosp. Sci. Technol.* **30**(1), 255–268 (2013). <https://doi.org/10.1016/j.ast.2013.08.010>
64. van der Vegt, J.J.W., Sudirham, J.J.: A space-time discontinuous Galerkin method for the time-dependent Oseen equations. *Appl. Numer. Math.* **58**(12), 1892–1917 (2008)
65. Van der Vorst, H.A.: Bi-CGStab: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **13**(2), 631–644 (1992)
66. Van der Vorst, H.A., Vuik, C.: GMRESR: a family of nested GMRES methods. *Numer. Linear Algebra Appl.* **1**(4), 369–386 (1994)
67. Zhang, Q.: Gepup: Generic projection and unconstrained ppe for fourth-order solutions of the incompressible Navier–Stokes equations with no-slip boundary conditions. *J. Sci. Comput.* **67**(3), 1134–1180 (2016)
68. Zhang, W., Almgren, A., Day, M., Nguyen, T., Shalf, J., Unat, D.: Boxlib with tiling: an adaptive mesh refinement software framework. *SIAM J. Sci. Comput.* **38**(5), S156–S172 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.