



A spectrally accurate method for overlapping grid solution of incompressible Navier–Stokes equations



Brandon E. Merrill ^{a,*}, Yulia T. Peet ^{a,b}, Paul F. Fischer ^{c,b}, James W. Lottes ^{b,d}

^a School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85287, USA

^b Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA

^c Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

^d Google Inc., Mountain View, CA 94043, USA

ARTICLE INFO

Article history:

Received 20 June 2015

Received in revised form 27 November 2015

Accepted 27 November 2015

Available online 30 November 2015

Keywords:

Overlapping grid methods

Spectral accuracy

Spectral element method

Navier–Stokes equations

ABSTRACT

An overlapping mesh methodology that is spectrally accurate in space and up to third-order accurate in time is developed for solution of unsteady incompressible flow equations in three-dimensional domains. The ability to decompose a global domain into separate, but overlapping, subdomains eases mesh generation procedures and increases flexibility of modeling flows with complex geometries. The methodology employs implicit spectral element discretization of equations in each subdomain and explicit treatment of subdomain interfaces with spectrally-accurate spatial interpolation and high-order accurate temporal extrapolation, and requires few, if any, iterations, yet maintains the global accuracy and stability of the underlying flow solver. The overlapping mesh methodology is thoroughly validated using two-dimensional and three-dimensional benchmark problems in laminar and turbulent flows. The spatial and temporal convergence is documented and is in agreement with the nominal order of accuracy of the solver. The influence of long integration times, as well as inflow–outflow global boundary conditions on the performance of the overlapping grid solver is assessed. In a turbulent benchmark of fully-developed turbulent pipe flow, the turbulent statistics with the overlapping grids is validated against published available experimental and other computation data. Scaling tests are presented that show near linear strong scaling, even for moderately large processor counts.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Finding numerical solutions to partial differential equations (PDEs) by decomposing the computational domain into smaller subdomains is an idea that has been around for well over a century. Domain decomposition methods have been utilized for several different purposes, including straightforward parallelization [1–4], simplified mesh generation for complex geometries [5–8], and the ability to use different parameters or methods in different subdomains [9–12]. These techniques exist in many forms, and each has its strengths. Some decompose the global domain into overlapping subdomains [13–18], while others employ non-overlapping subdomains [19,9,7,20–22]. Some use explicit interpolation techniques for values at interface boundaries [9,23,20,24], and others carry out implicit interpolation [25,26,22,27,13,14,28–30]. Domain decomposi-

* Corresponding author.

E-mail addresses: brandon.merrill@asu.edu (B.E. Merrill), ypeet@asu.edu (Y.T. Peet), fischer@mcs.anl.gov (P.F. Fischer), jlottes@gmail.com (J.W. Lottes).

tion techniques have been developed for use with several numerical methods including finite difference [14,5,9,23], finite element [25], finite volume [17,31], and spectral methods [21,22].

The earliest known research in domain decomposition methods was performed by H.A. Schwarz whose work was published in 1870 [13]. The original Schwarz Alternating Method, initially proposed for analytical calculations [32], was developed for the global solution of boundary value problems for harmonic functions [15] decomposed into overlapping subdomains, $\Omega = \Omega_1 \cup \Omega_2$. The solution in the first subdomain (Ω_1 with boundaries $\partial\Omega_1 \cap \partial\Omega$ and $\Gamma_1 = \partial\Omega_1 \setminus \partial\Omega$) is found using the global boundary conditions on $\partial\Omega_1 \cap \partial\Omega$ and corresponding values from Ω_2 at the previous iteration on Γ_1 . The solution of Ω_2 is then found by using values from the solution in Ω_1 on Γ_2 . These two steps are iterated until sufficient convergence is reached (see [33,32,34]).

In the 1960's, Volkov generalized the original Schwarz Alternating Method into a numerical domain decomposition technique, in a form of the *Composite Mesh Difference Method* (CMDM) [14]. CMDM used finite difference methods to solve the 2-dimensional Poisson equation numerically on overlapping grids. His research laid the foundation for subsequent techniques that extended the use of CMDM to other elliptical and hyperbolic PDEs, and boundary value and initial value problems, with the ability to use curvilinear meshes (see [35–38,6]). Overlapping domain decomposition methods have also been developed to model complex equations and handle various difficulties in solving practical problems. The Chimera Grid Scheme, introduced in [39], employs overset (overlapping) grids for modeling flows in complex geometries. Shortly after initial development it was enhanced for use with three dimensional flows modeled by the Euler equations [40] and later with the addition of the thin-layer Navier–Stokes equations. More recently, Chimera Grid techniques have been used to model a variety of problems with complex geometries [41,42]. Subsequently, Henshaw and Schwendeman [17,4] developed a method for using overlapping mesh techniques in modeling high-speed reactive flows, in two and three dimensions.

In addition, techniques that employ non-overlapping grids (sometimes called patched grids) were developed. Examples include a zonal approach that uses a flux-vector splitting technique for the determination of interface values in Euler equations [43–46], Lions method [19] that uses an iterative technique to arrive at the correct values to be passed between non-overlapping subdomains in solving Laplace's equation and more general second-order elliptic problems, Dawson's approach [9] that solves the heat equation using an explicit finite difference formula to determine the interface values and allows for different time stepping to be used in different subdomains. Non-overlapping grid techniques have also been extended and employed in solving the advection–diffusion equation [20] and the Navier–Stokes equations [45,22]. Some of the more recently developed non-overlapping domain decomposition methods achieve high finite global accuracy [47] and some spectral accuracy [21,7,48,22].

While non-overlapping mesh methods allow some flexibility in mesh generation, the constraints in these techniques inhibit additional flexibility that is seen in overlapping mesh methods. By allowing variable overlap size, a broad range of potential mesh configurations are supported with overlapping methods, thus allowing for more simplified mesh generation. Additionally, overlapping methods provide a convenient framework for further extension towards moving domain methods, allowing for general and unconstrained motion of rigid body parts through the background stationary meshes [49–52].

So far, existing overlapping grid methods for the time-dependent PDE coupling have been traditionally relying on low-order, finite-difference or finite-volume schemes. Although some of the methods have been extended to achieve higher-order spatial convergence, using extended stencil finite-difference or compact schemes, the upper bound of the global accuracy has been usually limited to four [23,24,53], and at most six [54–56,27,57]. Recently, Brazell, Sitaraman and Mavriplis developed a high-order overlapping Discontinuous Galerkin solver for compressible equations, that uses Lagrangian interpolation at interface boundaries, and documented a polynomial convergence up to fourth order [58]. In the current paper, we introduce a spectrally-accurate overlapping mesh method for incompressible equations, that is based on a spectral-element method. The Spectral Element Method, which can be perceived as a high-order extension of the Finite Element Method, divides a domain into several conforming and adjacent elements [59,3,60]. The volume within each element is discretized using N th-order tensor-product Lagrange interpolating polynomials on Gauss–Lobatto–Legendre nodal points. Approximations in all elements are coupled at the boundaries to form a global solution [60], which achieves spectral convergence with p - (polynomial order) refinement. In this work, we combine a Spectral Element Method solver with the overlapping grid approach, to arrive at a globally spectrally-accurate method for solution of the incompressible Navier–Stokes equations on overlapping domains.

One of the inherent challenges with overlapping grid methods is to minimize the errors that are introduced due to the coupling of the individual subdomain solutions into the global solution. The coupling errors consist of spatial errors and temporal errors, and have to be treated separately. Spatial errors are introduced by the spatial interpolation stencil employed to obtain a function value at the interface points of one domain from the gridpoint values in the adjacent domains at the same time level. Some overlapping mesh methods circumvent the spatial error by requiring that the gridpoints in overlapping domains exactly coincide [61,62], thus fully conserving communicated information, with the drawback of decreased flexibility in mesh generation. Other methods that do not require the exact match of the gridpoints and thus are more flexible, use finite order interpolation schemes to determine values from adjacent subdomains. Although simple linear interpolation techniques have been popular [6,9,63,29,64], it was shown by Chesshire and Henshaw [6] that an interpolation scheme should be consistent with the accuracy of the underlying solver and higher-order interpolation is required to maintain the accuracy of the coupled solution with high-order methods, generally, of the order of one higher than the underlying solver accuracy if the overlap width scales with the grid resolution, and the same if it stays constant during grid refinement. Thus, in fourth- and sixth-order methods [58,23,27,24], a generalized Lagrangian interpolation method

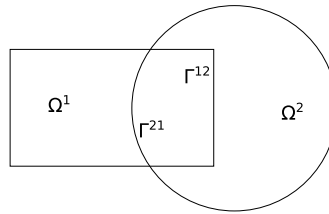


Fig. 1. Domain decomposition using two domains.

of the corresponding order was employed. In the present work, we use N th-order Lagrangian interpolation approach, on non-uniform Gauss–Lobatto–Legendre grids, to arrive at a spectrally-accurate interpolation scheme (with p -refinement) that allows us to maintain the global spectral accuracy of the coupled solution.

To avoid temporal errors due to coupling, the subdomain coupling methods have often been formulated implicitly, when either the interpolation dependencies are written into the global matrix [6,26,27], or independent subdomain solutions are coupled through the use of Schwarz-like iterations [13] to ensure interface value convergence [28–31]. Global convergence of the variables in adjacent subdomains often requires many, sometimes hundreds [29], or even thousands [31] iterations, the number generally being dependent upon the PDE, the overlap size, and timestep [65,66]. Although Henshaw [67] showed that the convergence rates when solving elliptic PDEs on overlapping subdomains can be reduced by using a multigrid solver with a smoothing algorithm near interface boundaries, this approach is difficult to implement and requires some global interventions to the solver. However, if the coupling scheme, even without iterations, does not introduce temporal errors larger than those of the global timestepping scheme, the global temporal accuracy will be preserved [68], particularly the interface values need not strictly match, but must merely be “consistent”. The notion of consistency has been previously used when constructing and analyzing interpolation and iteration schemes for the overlapping grid methods [30], where consistency was interpreted in a spatial sense. In [68], we have proposed an explicit temporal interface extrapolation algorithm for the overlapping grid methods that preserves the overall temporal accuracy, and analyzed its stability. The proposed coupling scheme is essentially “temporally consistent” (the differences between the interface values due to temporal coupling converge with the same rate as the temporal errors of the underlying flow solver).

In the present work, we implement this temporally consistent coupling scheme for coupling the solutions of incompressible Navier–Stokes equations on overlapping grids discretized with the SEM method. This novel scheme achieves a specified order of accuracy with a small number of iterations. It ensures that the temporal error between the values in the subdomains at the interfaces is equal to the temporal accuracy of the underlying SEM solver. Assuming that this holds true, the interface values need not be an exact match, while still maintaining the global accuracy of the solution. While the accuracy of the extrapolation scheme is maintained in the absence of iterations, the stability of the formulation can generally be improved by implementing a low number of Schwarz-like iterations [68].

The current paper introduces an original method for coupling Navier–Stokes equations on the overlapping domains that is spectrally accurate in space and high-order accurate in time up to a specified order. Although an overlapping Schwarz method has been previously employed as a preconditioner for the solution of the Poisson equation in a spectral element formulation of Navier–Stokes equations on a single domain [3,69], a spectrally-accurate approach that achieves a full unsteady coupling of independent solutions to Navier–Stokes equations on overlapping grids appears to be novel. In addition, our method achieves a specified temporal accuracy with the minimum number of iterations (usually two to three) while maintaining stability of the formulation. We demonstrate the spectral spatial and high-order temporal convergence of the developed solver, as well as the overall consistency of the coupled solution, on a number of two and three-dimensional examples with incompressible Navier–Stokes equations, including laminar and turbulent test cases.

The remainder of the paper is organized in the following manner: section 2 describes the computational methodology; section 3 discusses the techniques used to parallelize the overlapping grid solver; section 4 presents the results for verification and validation; and section 5 shows the scalability tests. In section 6 some final conclusions are drawn, including a brief discussion of future extensions and applications of this work.

2. Methodology

2.1. Mathematical formulation

In the present formulation, a two or three dimensional global solution domain is decomposed into two overlapping subdomains, $\Omega = \Omega^1 \cup \Omega^2$, an example of which is seen in Fig. 1. Values at interface boundaries, Γ^{12} and Γ^{21} are equal to contiguous values in the adjacent subdomain, and the solution in each subdomain is governed by the incompressible Navier–Stokes equations which are represented in non-dimensional form in space R^d , as shown below

$$\Omega^1 \begin{cases} \frac{\partial \mathbf{u}^{[1]}}{\partial t} + \mathbf{u}^{[1]} \cdot \nabla \mathbf{u}^{[1]} = -\nabla p^{[1]} + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^{[1]} \\ \nabla \cdot \mathbf{u}^{[1]} = 0 \end{cases} \quad (1)$$

and

$$\Omega^2 \begin{cases} \frac{\partial \mathbf{u}^{[2]}}{\partial t} + \mathbf{u}^{[2]} \cdot \nabla \mathbf{u}^{[2]} = -\nabla p^{[2]} + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^{[2]} \\ \nabla \cdot \mathbf{u}^{[2]} = 0 \end{cases} \quad (2)$$

where \mathbf{u} is the velocity vector, p is the pressure, and $\text{Re} = UL/\nu$ is the Reynolds number based on a characteristic velocity, length scale and kinematic viscosity. The bracketed superscript denotes the corresponding subdomain. Generalized initial and boundary conditions for each subdomain are presented below

$$\begin{aligned} \mathbf{u}^{[1],[2]}(\mathbf{x}, 0) &= \mathbf{u}_0(\mathbf{x}), & \mathbf{x} \in \Omega^{1,2} \\ \mathbf{u}^{[1],[2]}(\mathbf{x}, t) &= \mathbf{u}_d(\mathbf{x}, t), & \mathbf{x} \in \partial\Omega_D \\ \nabla \mathbf{u}^{[1],[2]}(\mathbf{x}, t) \cdot \mathbf{n} &= 0, & \mathbf{x} \in \partial\Omega_N \\ \mathbf{u}^{[1]}(\mathbf{x}, t) &= \mathbf{u}^{[2]}(\mathbf{x}, t) & \mathbf{x} \in \Gamma^{12} \cup \Gamma^{21} \end{aligned} \quad (3)$$

where $\partial\Omega_D$ corresponds to Dirichlet portions of the global boundary, $\partial\Omega_N$ represents the Neumann portions of the global boundary (with \mathbf{n} the unit outward pointing surface normal), and Γ^{ij} denotes an interface boundary, or the local boundary of a subdomain, $\partial\Omega^i$, that is not also part of the global boundary: $\Gamma^{ij} \equiv \partial\Omega^i \setminus \partial\Omega$.

A global solution is formed by weakly coupling subdomain solutions. The weak coupling occurs at subdomain interfaces, Γ^{ij} , where values from Ω^j are spatially and temporally interpolated to approximate solutions at interface boundaries of Ω^i at time t

$$\begin{aligned} \mathbf{u}^{[1]}(\mathbf{x}, t) &= \mathbf{u}_{int}^{[2]}(\mathbf{x}, t) & \mathbf{x} \in \Gamma^{12} \\ \mathbf{u}^{[2]}(\mathbf{x}, t) &= \mathbf{u}_{int}^{[1]}(\mathbf{x}, t) & \mathbf{x} \in \Gamma^{21} \end{aligned} \quad (4)$$

where the *int* subscript denotes interpolated values from the corresponding subdomain.

2.2. Numerical formulation

Aside from exchanging velocity values at interfaces, the solutions within each subdomain are computed independently of each other, and the numerical formulation within each subdomain is identical. The subdomain superscripts are thus withheld from sections in following discussion ($\Omega^i \rightarrow \Omega$ and $\mathbf{u}^{[i]} \rightarrow \mathbf{u}$), except where needed for clarity.

2.2.1. Temporal discretization

The temporal discretization in the developed methodology uses a characteristics time-stepping scheme that exploits a property of the flow that allows for calculating the solution along characteristic paths. This method allows for much larger timesteps by rewriting the time derivative and convective terms in (1) and (2) using the material derivative

$$\begin{aligned} \frac{D\mathbf{u}}{Dt} &= -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \quad (5)$$

with

$$\begin{aligned} \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0(\mathbf{x}), & \mathbf{x} \in \Omega \\ \mathbf{u}(\mathbf{x}, t) &= \mathbf{u}_d(\mathbf{x}, t), & \mathbf{x} \in \partial\Omega_D \\ \nabla \mathbf{u}(\mathbf{x}, t) \cdot \mathbf{n} &= 0, & \mathbf{x} \in \partial\Omega_N \\ \mathbf{u}^{[i]}(\mathbf{x}, t) &= \mathbf{u}_{int}^{[j]}(\mathbf{x}, t), & \mathbf{x} \in \Gamma^{ij} \end{aligned} \quad (6)$$

where the material derivative

$$\frac{D\mathbf{u}}{Dt} \equiv \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \quad (7)$$

describes the time evolution of \mathbf{u} attached to material points, or points that are convecting along characteristic lines given by the velocity field \mathbf{u} . Using this representation, the material derivatives in (5) are discretized using backward differentiation of order k (BDF k)

$$\frac{D\mathbf{u}}{Dt} \rightarrow \frac{1}{\Delta t} \left(\beta_{0k} \mathbf{u}^n + \sum_{q=1}^k \beta_{qk} \tilde{\mathbf{u}}^{n-q} \right) \quad (8)$$

Table 1
Coefficients for the BDFk scheme, $k = 1, 2, 3$ [68].

	β_{q1}	β_{q2}	β_{q3}
$q = 0$	1	3/2	11/6
$q = 1$	-1	-2	-3
$q = 2$		1/2	3/2
$q = 3$			-1/3

where the β_{qk} terms are the backwards differentiation coefficients given in Table 1 for up to third-order [60,68], and $\tilde{\mathbf{u}}^{n-q}$ represents the value of \mathbf{u} at time t^{n-q} and at the location, \mathbf{X}^{n-q} , from which it originated prior to being convected by the velocity field:

$$\tilde{\mathbf{u}}^{n-q} = \mathbf{u}(\mathbf{X}^{n-q}(\mathbf{x}), t^{n-q}). \tag{9}$$

Backward differentiation of (5), after some rearranging, then gives

$$\begin{aligned} \frac{1}{\Delta t} \beta_{0k} \mathbf{u}^n - \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^n + \nabla p^n &= \frac{1}{\Delta t} \sum_{q=1}^k \beta_{qk} \tilde{\mathbf{u}}^{n-q} \quad \text{in } \Omega \\ \nabla \cdot \mathbf{u}^n &= 0 \quad \text{in } \Omega, \end{aligned} \tag{10}$$

with initial and boundary conditions given by the temporal discretization of (6).

The main difficulty with this technique is determining the values of \mathbf{u} at the foot of the characteristic for each gridpoint. Marching backward in time along the characteristic line and evaluating \mathbf{u} at that time and location assumes interpolation since the new location will not be incident with a grid point. The velocity field \mathbf{u} must also be interpolated at each point along the characteristic to determine the correct path along which to march backward, all of which greatly increases computational cost.

The need to know \mathbf{X}^{n-q} is bypassed by using the operator-integration factor splitting (OIFS) method developed by Maday, Patera, and Rønquist [70]. The OIFS shows that $\tilde{\mathbf{u}}^{n-q}$ can be found by solving the initial-boundary value (IBV) convection problem

$$\frac{\partial \tilde{\mathbf{u}}}{\partial s} + \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} = 0, \quad s \in [t^{n-q}, t^n] \tag{11}$$

with initial and boundary conditions

$$\begin{aligned} \tilde{\mathbf{u}}(\mathbf{x}, t^{n-q}) &= \mathbf{u}(\mathbf{x}, t^{n-q}) \\ \tilde{\mathbf{u}}(\mathbf{x}, t) &= \mathbf{u}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \partial\Omega_c, \end{aligned} \tag{12}$$

where $\partial\Omega_c$ is the subset of $\partial\Omega$ where $\mathbf{u} \cdot \mathbf{n} < 0$, or the portion of $\partial\Omega$ that contains inflow characteristics. Note that, as opposed to semi-Lagrangian backwards-characteristic-tracing algorithms, the OIFS approach, by casting the problem as an IBV problem, avoids interpolation along characteristics, thus eliminating the problem of tracing along characteristics into an adjacent subdomain.

Since the terms on the right-hand side of (10) are computed separately from the global problem, they will be expressed as a forcing function to the global Stokes problem

$$\mathbf{f}^n = \frac{1}{\Delta t} \sum_{q=1}^k \beta_{qk} \tilde{\mathbf{u}}^{n-q} \tag{13}$$

which allows (10) to be expressed as

$$\begin{aligned} \frac{\beta_{0k}}{\Delta t} \mathbf{u}^n - \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^n + \nabla p^n &= \mathbf{f}^n \quad \text{in } \Omega \\ \nabla \cdot \mathbf{u}^n &= 0 \quad \text{in } \Omega \end{aligned} \tag{14}$$

with

$$\begin{aligned} \mathbf{u}^{n=0}(\mathbf{x}) &= \mathbf{u}_0(\mathbf{x}), & \mathbf{x} &\in \Omega \\ \mathbf{u}^n(\mathbf{x}) &= \mathbf{u}_d^n(\mathbf{x}), & \mathbf{x} &\in \partial\Omega_D \\ \nabla \mathbf{u}^n(\mathbf{x}) \cdot \mathbf{n} &= 0, & \mathbf{x} &\in \partial\Omega_N \\ \mathbf{u}^{[i],n}(\mathbf{x}) &= g\left(\mathbf{u}_{int}^{[j],n}(\mathbf{x}), \mathbf{u}_{int}^{[j],n-1}(\mathbf{x}), \dots, \mathbf{u}_{int}^{[j],n-p}(\mathbf{x})\right), & \mathbf{x} &\in \Gamma^{ij} \end{aligned} \tag{15}$$

where interface conditions at timestep t^n in the last equation are a function $g()$ of interpolated values from the adjacent subdomain, from the previous timesteps, as discussed in the next section. Further details regarding the characteristics time-stepping method including the OIFS method for handling inhomogeneous terms are found in [60,70].

Table 2
IEXTm schemes, $m = 1, 2, 3$ [68].

	γ_{p1}	γ_{p2}	γ_{p3}
$p = 1$	1	2	3
$p = 2$		-1	-3
$p = 3$			1

2.2.2. Temporal interface extrapolation

Here, we utilize a novel explicit temporal interface extrapolation scheme to obtain interface conditions at a current timestep, using the known solutions from the previous timesteps, with an arbitrary order of accuracy (IEXTm, where m stands for the order of accuracy) [68]. For instance, if velocity values are to be collected for use at an interface point \mathbf{x} on Γ^{ij} at time t^n , then $\mathbf{u}^{[i],n}(\mathbf{x})$ is found using interpolated values in Ω^j taken at p previous timesteps, $\mathbf{u}_{int}^{[j],n-p}$. The IEXTm scheme finds $\mathbf{u}^{[i],n}(\mathbf{x})$ with m th-order accuracy

$$\mathbf{u}^{[i],n}(\mathbf{x}) = \sum_{p=1}^m \gamma_{pm} \mathbf{u}_{int}^{[j],n-p}(\mathbf{x}), \quad \mathbf{x} \in \Gamma^{ij} \tag{16}$$

where γ_{pm} are the extrapolation weights for the IEXTm scheme, which, for up to 3rd-order schemes, can be found in Table 2 [68]. Thus, boundary conditions in (15) become

$$\begin{aligned} \mathbf{u}^n(\mathbf{x}) &= \mathbf{u}_d^n(\mathbf{x}), & \mathbf{x} \in \partial\Omega_D \\ \nabla \mathbf{u}^n(\mathbf{x}) \cdot \mathbf{n} &= 0, & \mathbf{x} \in \partial\Omega_N \\ \mathbf{u}^{[i],n}(\mathbf{x}) &= \sum_{p=1}^m \gamma_{pm} \mathbf{u}_{int}^{[j],n-p}(\mathbf{x}), & \mathbf{x} \in \Gamma^{ij} \end{aligned} \tag{17}$$

Performing temporal interface extrapolation with order of accuracy equal to the accuracy of the global temporal scheme, maintains the temporal accuracy of the global scheme. The current IEXTm scheme therefore allows to maintain the global accuracy of the solution without the penalties associated with implicit methods, such as expensive matrix inversion procedures, or excessive amount of iterations.

Iterative scheme The proposed explicit temporal interface extrapolation scheme changes the stability properties of the coupled solver. The IEXTm subdomain coupling scheme was theoretically analyzed in [68] on the example of one-dimensional heat equation discretized in time with BDF scheme, which is implicit and, in the absence of interface coupling, unconditionally stable. While the first-order accurate interface extrapolation (IEXT1) was proven to retain the unconditional stability of the global scheme (at least with first- or second-order BDF time integration), the stability of the higher-order IEXTm schemes depended on the size of overlap and the number of iterations, with a small number of iterations generally leading to a significant increase in the stability threshold [68].

To allow for stabilizing effect of Schwarz-like iterations for higher-order extrapolation schemes, the method is modified as follows

$$\begin{aligned} \frac{\beta_{0k}}{\Delta t} \mathbf{u}^{l,n} - \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^{l,n} + \nabla p^{l,n} &= \mathbf{f}^n & \text{in } \Omega \\ \nabla \cdot \mathbf{u}^{l,n} &= 0 & \text{in } \Omega \end{aligned} \tag{18}$$

with

$$\begin{aligned} \mathbf{u}^{l=1,n=0}(\mathbf{x}) &= \mathbf{u}_0(\mathbf{x}), & \mathbf{x} \in \Omega \\ \mathbf{u}^{l,n}(\mathbf{x}) &= \mathbf{u}_d^n(\mathbf{x}), & \mathbf{x} \in \partial\Omega_D \\ \nabla \mathbf{u}^{l,n}(\mathbf{x}) \cdot \mathbf{n} &= 0, & \mathbf{x} \in \partial\Omega_N \end{aligned} \tag{19}$$

where l is the iteration index, and interface conditions are defined

$$\left. \begin{aligned} \mathbf{u}^{[i],l=1,n}(\mathbf{x}) &= \sum_{p=1}^m \gamma_{pm} \mathbf{u}_{int}^{[j],l_{max},n-p}(\mathbf{x}) \\ \mathbf{u}^{[i],l>1,n}(\mathbf{x}) &= \mathbf{u}_{int}^{[j],l-1,n}(\mathbf{x}) \end{aligned} \right\} \quad \mathbf{x} \in \Gamma^{ij} \tag{20}$$

where the initial iteration uses IEXTm at subdomain interface points, while subsequent iterations use values interpolated from the most recent solution in the neighboring subdomain.

Reference [68] showed analytically that a low number of iterations (usually less than five) can sufficiently stabilize the solution of the unsteady one-dimensional parabolic problem even for large timesteps and small overlap sizes (down to 5% of total gridpoints in the overlap region). It was also shown that larger overlap sizes require smaller number of iterations for stability. Although extension of this formal stability analysis to multidimensional problems and Navier–Stokes equations will be performed in the future, the current methodology seems to exhibit similar trends: larger overlap sizes do exhibit better stability and require smaller number of iterations. In the current methodology, and based on preliminary results from the stability analysis [68], we would target to have a few elements in the region of overlap, which generally required two to four iterations for stability.

Although a low number of iterations is sometimes needed for stability, we stress that this is very different from fully implicit methods without IEXT scheme, where hundreds of iterations are needed to maintain the global accuracy.

Note that upon problem initialization at $t = 0$, the solver must use reduced order temporal schemes (for example, BDF1/IEXT1 for the first time step etc.) since previous solutions do not yet exist, a typical issue for higher-order explicit or backward-differencing temporal schemes. At restart, this can be circumvented by saving and using the corresponding number of previous solutions; at initialization, one can either use analytical solution (if available), or very small Δt for the first several timesteps, or rely on a natural “flush out” of initial start-up errors during long simulation runs.

2.2.3. Global variational form

The Stokes problem (18), with time n , iteration l , and accuracy k indices excluded for readability, is cast into variational form by multiplying the equations by a test function, $\mathbf{v} \in \mathcal{H}_0^1(\Omega)$, followed by integration, producing the variational form of the problem: Find $\mathbf{u} \in \mathcal{H}_b^1(\Omega)$, $p \in \mathcal{L}^2(\Omega)$ such that

$$\begin{aligned} \frac{\beta_0}{\Delta t} (\mathbf{u}, \mathbf{v}) - \frac{1}{\text{Re}} (\nabla^2 \mathbf{u}, \mathbf{v}) + (\nabla p, \mathbf{v}) &= (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{H}_0^1(\Omega) \\ -(q, \nabla \cdot \mathbf{u}) &= 0 \quad \forall q \in \mathcal{L}^2(\Omega), \end{aligned} \quad (21)$$

where the inner product of two scalar functions a and b is defined

$$(a, b) = \int_{\Omega} a(\mathbf{x}) b(\mathbf{x}) d\mathbf{x}, \quad \forall a, b \in \mathcal{L}^2(\Omega), \quad (22)$$

and where $\mathcal{L}^2(\Omega)$ is the space of square-integrable functions, and $\mathcal{H}^1(\Omega)$ is the space of square-integrable functions whose first derivatives are also square-integrable. $\mathcal{H}_0^1(\Omega)$ denotes the subspace of $\mathcal{H}^1(\Omega)$ with $\mathbf{v} = 0$ on Dirichlet, $\partial\Omega_D$, and interface, Γ^{ij} , boundaries, while $\mathcal{H}_b^1(\Omega)$ denotes the subspace of $\mathcal{H}^1(\Omega)$ with $\mathbf{v} = \mathbf{v}_d$ on $\partial\Omega_D$, or $\mathbf{v}^{[ij]} = \mathbf{g}(\mathbf{v}_{int}^{[ij]})$ on Γ^{ij} .

Using integration by parts,

$$(\nabla^2 \mathbf{u}, \mathbf{v}) \rightarrow \int_{\partial\Omega} \mathbf{v} \nabla \mathbf{u} \cdot \hat{\mathbf{n}} dA - \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} dV, \quad (23)$$

where “:” denotes a Frobenius inner product, on the terms $-\frac{1}{\text{Re}}(\nabla^2 \mathbf{u}, \mathbf{v})$ and $(\nabla p, \mathbf{v})$ yields the weak form of (21). Since the first integral in the right-hand side vanishes due to boundary conditions and a choice of test functions \mathbf{v} , the weak variational form of the global Navier–Stokes equations becomes

$$\begin{aligned} \frac{\beta_0}{\Delta t} (\mathbf{u}, \mathbf{v}) + \frac{1}{\text{Re}} (\nabla \mathbf{u}, \nabla \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) &= (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{H}_0^1(\Omega) \\ -(q, \nabla \cdot \mathbf{u}) &= 0 \quad \forall q \in \mathcal{L}^2(\Omega), \end{aligned} \quad (24)$$

2.2.4. Spatial discretization

In the present spectral element formulation, each computational subdomain, Ω , is decomposed into a set of E non-overlapping adjacent elements,

$$\Omega = \sum_{k=1}^E \Omega_k, \quad (25)$$

with global inner products

$$(a, b)|_{\Omega} = \sum_{k=1}^E (a_k, b_k)|_{\Omega_k}. \quad (26)$$

Spatial discretization of (24) within each element is achieved by projecting search spaces $\mathcal{H}^1(\Omega)$ and $\mathcal{L}^2(\Omega)$ onto the corresponding finite-dimensional subspaces $X^N \subset \mathcal{H}^1(\Omega)$ and $Y^N \subset \mathcal{L}^2(\Omega)$ defined as

$$\begin{aligned} X^N &= \mathcal{H}^1(\Omega) \cap \mathbb{P}_N^d \\ Y^N &= \mathcal{L}^2(\Omega) \cap \mathbb{P}_{N-2}^d, \end{aligned} \tag{27}$$

where \mathbb{P}_N is the space of all polynomials of degree less than or equal to N , and \mathbb{P}_N^d is the corresponding tensor-product in d spatial dimensions. Each arbitrary element Ω_k has a distinct mapping function to a d -dimensional reference element $\hat{\Omega} = [-1, +1]^d$:

$$\mathbf{x}_k(\mathbf{r}) \in \Omega_k \mapsto \mathbf{r} \in \hat{\Omega}, \tag{28}$$

with a well-defined inverse

$$\mathbf{r}_k(\mathbf{x}) \in \hat{\Omega} \mapsto \mathbf{x} \in \Omega_k. \tag{29}$$

The present spectral element method is a nodal scheme, where collocation points in the velocity space of the reference element, $\xi_j \in [-1, +1]$, are specified using Gauss–Lobatto Legendre (GL) quadrature, while for the pressure space, collocation points, $\eta_j \in]-1, +1[$, are specified with Gauss Legendre (G) quadrature. The basis functions used to span the polynomial space, \mathbb{P}_N for velocity and \mathbb{P}_{N-2} for pressure, are the Lagrange interpolating polynomials, $\phi_i(r)$, satisfying

$$\phi_i(\xi_j) = \delta_{ij}, \tag{30}$$

with δ_{ij} a Kronecker delta. Note that the quadrature in the velocity space contains points that lie on the element boundaries, thus maintaining basis continuity across element boundaries, while basis continuity is not enforced in the pressure space because of the Gauss Legendre quadrature.

Gaussian quadratures allow for accurate numerical integration of high order polynomials. Integration of a polynomial, $f(r)$, in a 1D reference domain, $-1 \leq r \leq 1$, takes the form (for GL quadrature)

$$\int_{-1}^1 f(r)dr = \sum_{l=0}^N \omega_l f(\xi_l), \tag{31}$$

where ξ_l are the collocation points in GL quadrature, and ω_l are their associated weights. A polynomial of up to order $2N - 1$ will be integrated exactly when using $N + 1$ collocation points represented by Gauss–Lobatto Legendre quadrature [60].

Each scalar field in Ω_k , represented by $f_k(\mathbf{x})$, in the space $\mathbb{P}_N^d(\Omega_k)$ is discretized, for example, in two-dimensions

$$f(\mathbf{x}(\mathbf{r}))|_{\Omega_k} = \sum_{i=0}^N \sum_{j=0}^N f_{k,ij} \phi_i(r_1) \phi_j(r_2), \quad r_1, r_2 \in [-1, 1]^2, \tag{32}$$

where the Lagrange polynomials satisfy (30) and \mathbf{x} maps the local coordinates to the reference geometry (28).

Restating the problem (24) in terms of quadrature and in finite-dimensional subspaces: Find $\mathbf{u} \in X_b^N$, $p \in Y^N$ such that

$$\frac{\beta_0}{\Delta t} (\mathbf{u}, \mathbf{v})_{GL} + \frac{1}{\text{Re}} (\nabla \mathbf{u}, \nabla \mathbf{v})_{GL} - (p, \nabla \cdot \mathbf{v})_G = (\mathbf{f}, \mathbf{v})_{GL} \quad \forall \mathbf{v} \in X_0^N \tag{33}$$

$$-(q, \nabla \cdot \mathbf{u})_G = 0 \quad \forall q \in Y^N \tag{34}$$

where the global inner products are determined as in equation (26).

2.2.5. One-dimensional operators

For simplicity, matrix discretizations will first be derived for one dimensional variables (unbolded) on the reference length, $[-1, 1]$. The inner products are performed on a local basis, with local variables represented using the element index, k . The inner product in the first term on the left side of equation (33) is discretized as

$$\begin{aligned} (u, v)_{k, GL} &= \int_{\hat{\Omega}} u_k v_k dr \\ &= \int_{\hat{\Omega}} \left(\sum_{i=0}^N u_{k,i} \phi_i(r) \right) \left(\sum_{j=0}^N v_{k,j} \phi_j(r) \right) dr \\ &= \sum_{j=0}^N v_{k,j} \sum_{i=0}^N u_{k,i} \int_{\hat{\Omega}} \phi_i(r) \phi_j(r) dr \\ &= \underline{v}_k^T \hat{B} \underline{u}_k \end{aligned} \tag{35}$$

where a hat over an unbolded variable denotes a one dimensional matrix operator for the reference length, and an underline denotes a vector of coefficients, $\underline{v} = (v_0, v_1, \dots, v_N)$. The mass matrix, \hat{B} , can be computed using GL quadrature weights

$$\begin{aligned}\hat{B}_{ij} &= \int_{\hat{\Omega}} \phi_i(r) \phi_j(r) dr \\ &\approx \sum_{l=0}^N \omega_l \phi_i(\xi_l) \phi_j(\xi_l).\end{aligned}\quad (36)$$

Numerical integration of the mass matrix as represented above is not exact since $N + 1$ points are being used to evaluate a polynomial of order $2N$, although the error is consistent with the overall approximation error, and accuracy increases spectrally as higher order polynomials are used. Due to the Lagrangian basis function property in (30), the mass matrix simplifies to an $(N + 1) \times (N + 1)$ diagonal matrix with the quadrature weights along the main diagonal, $\hat{B} = \text{diag}(\omega_l)$.

The inner product for the forcing term is also expressed using the mass matrix

$$(f, v)_{k, GL} = \underline{v}_k^T \hat{B} \underline{f}_k. \quad (37)$$

In a similar manner, the inner product in the second term on the left side of equation (33) is discretized

$$\left(\frac{du}{dr}, \frac{dv}{dr} \right)_{k, GL} = \underline{v}_k^T \hat{A} \underline{u}_k \quad (38)$$

where the stiffness matrix, \hat{A} , in terms of GL quadrature becomes

$$\begin{aligned}\hat{A}_{ij} &= \int_{\hat{\Omega}} \phi'_i(r) \phi'_j(r) dr \\ &= \sum_{l=0}^N \omega_l \phi'_i(\xi_l) \phi'_j(\xi_l).\end{aligned}\quad (39)$$

In terms of the spectral differentiation matrix

$$\hat{D}_{ij} \equiv \left. \frac{d\phi_j}{dr} \right|_{r=\xi_i} \quad (40)$$

and using GL quadrature, the stiffness matrix (39) can be written

$$\hat{A}_{ij} = \sum_{l=0}^N \omega_l \hat{D}_{li} \hat{D}_{lj}. \quad (41)$$

Numerical integration with GL quadrature of the stiffness operator is exact since a polynomial of order $2N - 2$ is being evaluated with $N + 1$ collocation points [60]. Note that since the basis polynomials are known, the mass and stiffness operators could be precomputed exactly, although the resulting matrices would be full, leading to much greater computational cost, especially when extended to higher dimensional operators.

The pressure term in equation (33) is represented in weak form using G quadrature. However, $\nabla \cdot \mathbf{v}$ values are defined at GL points, and therefore, the pressure values are interpolated to GL quadrature for compatibility. Thus the pressure term is discretized

$$\left(p, \frac{dv}{dr} \right)_{k, G} = \underline{v}_k^T \hat{\hat{D}}^T \underline{p}_k \quad (42)$$

where $\hat{\hat{D}}$ is a local Derivative matrix defined using a weighted interpolation operator and the weighted one-dimensional derivative matrix. The inner product in the continuity equation (34) in one dimension is discretized in a similar manner

$$\left(q, \frac{du}{dr} \right)_G = \underline{q}_k^T \hat{\hat{D}} \underline{u}_k. \quad (43)$$

See [60] for additional details regarding the discretization of the pressure and continuity terms, including the local derivative operator.

2.2.6. Higher dimensions and deformed geometries

Extensions of the above operators, \hat{B} , \hat{A} , \hat{C} , and \hat{D} , must be made for use with elements of higher dimensions and deformed geometries. Extension to higher dimensions is generally a straightforward task by applying tensor products of operators. The most trivial extension is seen in the derivation of the mass matrix for a multi-dimensional reference element. For example, in three dimensions, we write

$$\underline{\hat{B}} = \hat{B} \otimes \hat{B} \otimes \hat{B}, \tag{44}$$

where the higher dimension operator is represented with the underline, and its dimensions are $(N + 1)^d \times (N + 1)^d$. The inner product, (35), for scalar variables in higher dimension reference elements is expressed as

$$(u, v)_{k, GL} = \underline{v}_k^T \underline{\hat{B}} \underline{u}_k, \tag{45}$$

where \underline{u}_k and \underline{v}_k now represent coefficients of multidimensional scalar functions, $u(\mathbf{x})$.

The operators are additionally modified for use with deformed geometries. As discussed above, a mapping function exists to map coordinates in a physical element to a reference element, (28), with an inverse mapping function to perform the opposite mapping, (29). The transformation Jacobian $J_k(\mathbf{r})$ for element k is derived for an element by calculating the determinant of a matrix constructed using the chain rule for mapping the derivatives of one space to the other, and is used in integration to ensure operator matrices are derived for the proper space. The term derived in (35), for example, can be derived for a two-dimensional physical element, Ω_k :

$$(u(\mathbf{x}), v(\mathbf{x}))_{k, GL} = \underline{v}_k^T \underline{B}_k \underline{u}_k, \tag{46}$$

where the mass matrix hat has been dropped, and an element index, k , has been added since the operator is now extended for use beyond reference element geometry to a configuration that is specific to element k . The mass matrix is now defined using the Jacobian with GL quadrature

$$\begin{aligned} \underline{B}_{k, ij} &= \int_{\hat{\Omega}} \phi_i \phi_j \phi_j J_k(r_1, r_2) \, d\mathbf{r} \\ &\approx \sum_{l=0}^N \sum_{m=0}^N \omega_l \omega_m \phi_i(\xi_l) \phi_j(\xi_l) \phi_j(\xi_m) \phi_j(\xi_m) J_k(\xi_l, \xi_m). \end{aligned} \tag{47}$$

Using the Lagrangian basis function property in (30), the local mass matrix again simplifies to a diagonal matrix expressed as tensor product of one dimensional mass matrices multiplied by a Jacobian matrix

$$\underline{B}_k = \underline{J}_k (\hat{B} \otimes \hat{B}). \tag{48}$$

Mass matrices for deformed geometries remain of size $(N + 1)^d \times (N + 1)^d$. Other matrix operators are extended for use with higher dimension elements with deformed geometries in a similar manner [60].

2.2.7. Element coupling

All derivations thus far have been towards the development of operators on a local element-wise basis. Local inner products for all elements are added together to express unassembled operators for the global subdomain, as shown below for the mass matrix

$$(u, v)_{GL} = \sum_{k=1}^E \underline{v}_k^T \underline{B}_k \underline{u}_k = \underline{v}_L^T \underline{B}_L \underline{u}_L \tag{49}$$

where the subscript “L” denotes a global unassembled array, with the block diagonal mass matrix $\underline{B}_L \equiv \text{diag}(\underline{B}_k)$. Summation of the other matrix operators is likewise performed to create unassembled operators in the global subdomain which are also block diagonal, $\underline{A}_L \equiv \text{diag}(\underline{A}_k)$ and $\underline{D}_{L,i} \equiv \text{diag}(\underline{D}_{k,i})$. The subscript i in the interpolating derivative operator arises since a slightly different matrix is used for each dimension of the problem due to direction dependence in multidimensional vector lexicographical ordering. Notice that in unassembled arrays, which contain $E \cdot (N + 1)^d$ values, continuity at element boundaries is not enforced, since adjacent elements need not contain identical values at their shared boundaries. Continuity is thus imposed by mapping an unassembled array (\underline{u}_L) into an assembled array (\underline{u}), in which adjacent elements reference the same values at shared boundaries. This process is performed with a connectivity operator, \underline{Q} , that maps unassembled arrays to assembled arrays, and vice versa, through *gather*

$$\underline{u} = \underline{Q}^T \underline{u}_L \tag{50}$$

and *scatter*

$$\underline{u}_L = \underline{Q} \underline{u} \tag{51}$$

operations. The absence of a subscript denotes a global assembled array, in which inter-element continuity is enforced by shared values at adjacent element boundaries. Assembled arrays thus contain fewer than $E \cdot (N + 1)^d$ values, since no duplicate entries exist for values that lie on boundaries of adjacent elements. The operator, \underline{Q} , is given by a matrix composed of “ones” and “zeros” which are arranged so that when \underline{Q} is multiplied with an assembled array, values located at an element boundary are assigned to every element that includes the boundary in the global unassembled array. Likewise, the transpose of \underline{Q} maps an unassembled array to a more condensed assembled array. The connectivity operator can act on both coefficient arrays and other operators [60]. The combined *gather-scatter* operations are used to perform global operations containing both assembled and unassembled arrays. By way of example, unassembled velocity coefficient arrays, as in (49), can be expressed, and determined, in global arrays

$$(\mathbf{u}, \mathbf{v})_{GL} = \mathbf{v}_L^T \underline{B}_L \underline{u}_L \rightarrow \mathbf{v}^T \underline{Q}^T \underline{B}_L \underline{Q} \underline{u}. \quad (52)$$

Similar *gather-scatter* operations are performed for the other terms in (33) and (34).

2.2.8. Vector-valued functions

Extension from scalar to vector space is accomplished by summing scalar arrays to represent the multidimensional problem

$$\begin{aligned} (\mathbf{u}, \mathbf{v}) &= \sum_{i=1}^d (u_i, v_i) \\ \Rightarrow (\mathbf{u}, \mathbf{v})_{GL} &= \sum_{i=1}^d \mathbf{v}_{i,L} \underline{B}_L \underline{u}_{i,L} \\ \Rightarrow (\mathbf{u}, \mathbf{v})_{GL} &= \mathbf{v}_L \underline{B}_L \underline{u}_L. \end{aligned} \quad (53)$$

The vector-valued arrays $\underline{\mathbf{u}}$ are defined by appending the scalar-valued arrays $(\underline{u}_1, \underline{u}_2, \underline{u}_3)^T$, and the matrix operator \underline{B}_L for the vector-valued arrays is the block-diagonal matrix containing the corresponding operators \underline{B}_L for the scalar arrays.

2.2.9. Boundary conditions

Neumann boundaries Homogeneous Neumann boundary conditions, on $\partial\Omega_N$, are satisfied in the weak form of the governing equations (24) due to integration-by-parts performed on the inner product, $(\nabla^2 \mathbf{u}, \mathbf{v})$, which is presented in (23). The resulting integral over the domain boundary, $\partial\Omega$, contains the term $\nabla \mathbf{u} \cdot \hat{\mathbf{n}}$, and thus vanishes due to the homogeneous Neumann boundary conditions, $\nabla \mathbf{u}(\mathbf{x}, t) \cdot \mathbf{n} = 0$, $\mathbf{x} \in \partial\Omega_N$.

Dirichlet and interface boundaries Satisfying boundary conditions at Dirichlet and interface boundaries need an additional constraint. Homogeneous Dirichlet boundary conditions are enforced using an unassembled Mask matrix, \underline{M}_L , that, when multiplied with a coefficient array, sets coefficient values corresponding with points on $\partial\Omega_D$ to zero, while keeping all other values the same. To carry out this operation, the Mask matrix is essentially constructed as an Identity matrix, but with “zeros” replacing the “ones” in positions that correspond to points on $\partial\Omega_D$. This will enforce the boundary condition, $\mathbf{u} = 0$ on $\partial\Omega_D$. For example, the mask matrix enforces homogeneous Dirichlet boundary conditions on (52) by

$$\mathbf{v}^T \underline{Q}^T \underline{B}_L \underline{Q} \underline{u} \rightarrow \mathbf{v}^T \underline{Q}^T \underline{M}_L \underline{B}_L \underline{M}_L \underline{Q} \underline{u}, \quad (54)$$

and the corresponding expression for the vector fields is

$$(\mathbf{u}, \mathbf{v})_{GL} = \mathbf{v}^T \underline{Q}^T \underline{M}_L \underline{B}_L \underline{M}_L \underline{Q} \underline{u}. \quad (55)$$

The same procedure is used for masking out the interface points, that is the corresponding mask matrix $\{\underline{M}_L\}_{k,lm}$ will be zero for all the Dirichlet points on $\partial\Omega_D$, and the interface points on Γ^{ij} , for a given subdomain, and one for all the other diagonal points. Note that an assembled mask matrix, \underline{M} , that acts on assembled arrays, may also be used, although the order of the matrix multiplications will change [60].

Inhomogeneous boundary conditions To set up inhomogeneous boundary conditions for the Dirichlet and interface boundaries, the semi-discrete boundary conditions given by Eqs. (19), (20), need to be spatially discretized. Although the operation is quite straightforward for global Dirichlet boundaries, a spatial discretization of interface conditions needs to be derived.

Spatial interface interpolation To maintain spectral accuracy of the global solution during the subdomain coupling, we propose to use a Lagrangian interpolation of N th order, consistent with the SEM spatial discretization (32). Thus, the following interpolation formula can be used in the domain Ω^j to find the interpolated value of velocity $\mathbf{u}_{int}^{[j]}(\mathbf{x}^{[i]})$ that corresponds to the interface point $\mathbf{x}^{[i]} \in \Gamma^{ij}$ of the domain Ω^i

$$\mathbf{u}_{int}^{[j]}(\mathbf{x}^{[i]}) = \sum_{l=0}^N \sum_{m=0}^N \mathbf{u}_{k,lm}^{[j]} \phi_l(r_1^{[j]}) \phi_m(r_2^{[j]}), \quad \mathbf{x}^{[i]} \in \Gamma^{ij}. \quad (56)$$

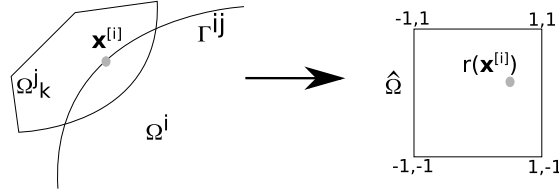


Fig. 2. Find coordinates of ξ^i , which is coincident with Ω_k^j , in the space of the reference element, $\hat{\Omega}$.

According to the SEM discretization, the velocity values $\underline{\mathbf{u}}_{k,lm}^{[j]}$ at the GL points are known, and the interpolation weights equal to the values of Lagrange interpolating polynomials $\phi_l(r^{[j]})$ can be found if the reference coordinates $\mathbf{r}^{[j]}$ in the domain Ω^j corresponding to the interface point $\mathbf{x}^{[i]} \in \Gamma^{ij}$ of the domain Ω^i are known. The difficulty is that these coordinates are not easily available, since the mapping function (28) is defined only locally in a subdomain. The main goal of interpolation becomes one of determining the coordinates of a GL point, $\mathbf{x}^{[i]} \in \Gamma^{ij}$ in terms of its position and reference coordinates within the adjacent subdomain, Ω^j , i.e. the global mapping $\mathbf{x}^{[i]} \in \Gamma^{ij} \rightarrow \mathbf{r}^{[j]} \in \hat{\Omega}$.

Before the reference coordinates are found, the element in the subdomain Ω^j that encapsulates the physical coordinates of $\mathbf{x}^{[i]} \in \Gamma^{ij}$ must be determined. This is done by testing the coordinates, $\mathbf{x}^{[i]}$, against the bounding lines (or faces) of each element. Once the coincident element, Ω_k^j , is found, the physical coordinates of the point $\mathbf{x}^{[i]}$ are mapped to corresponding reference coordinates, $\mathbf{r}^{[j]} \in [-1, 1]^d$, in Ω_k^j , as shown in Fig. 2. The reference coordinates, $\mathbf{r}^{[j]}$, within Ω_k^j are found by treating the search as an optimization problem

$$\min_{\mathbf{r} \in [-1, 1]^d} \mathbf{h}(\mathbf{r}), \tag{57}$$

where $\mathbf{h}(\mathbf{r}) = |\mathbf{x}^{[i]} - \mathbf{x}^{[j]}(\mathbf{r})|^2$, and $\mathbf{x}^{[j]}(\mathbf{r})$ is the result of function (29) that maps the coordinates in the reference element, \mathbf{r} , to physical coordinates, $\mathbf{x}^{[j]}$, in the deformed element Ω_k^j . The optimization problem is completed using the iterative Newton–Raphson method

$$\mathbf{r}^{p+1} = \mathbf{r}^p - \mathbf{J}(\mathbf{r}^p)^{-1} \mathbf{h}(\mathbf{r}^p), \tag{58}$$

where p is the iteration index, and $\mathbf{J}(\mathbf{r}) = \partial \mathbf{h} / \partial \mathbf{r}$. Note that while precise analytical coordinate mappings can be calculated for simple mesh structures (as in most of the test problems presented here), the method is designed to handle complex quadrilateral/hexahedral mesh configurations, such as elements composed of both curved and straight faces. Once reference coordinates $\mathbf{r}^{[j]}$ are found, the interpolated value $\mathbf{u}_{int}^{[j]}$ is evaluated using the equation (56) and communicated to the adjacent domain, where it is utilized as a boundary condition (20). The developed interpolation procedure is thus as accurate as the original solution approximation and maintains the spectral accuracy with the polynomial refinement.

Enforcing inhomogeneous boundary conditions Once the inhomogeneous boundary conditions have been spatially discretized, their enforcement is carried out by decomposing the solution, \mathbf{u} , into a homogeneous (\mathbf{u}_0) and inhomogeneous (\mathbf{u}_b) terms: $\mathbf{u} = \mathbf{u}_0 + \mathbf{u}_b$. The homogeneous term \mathbf{u}_0 that is a solution to the discretized SEM equations satisfies homogeneous boundary conditions on Dirichlet and interface boundaries. The inhomogeneous term \mathbf{u}_b is given by a continuous function that satisfies the Dirichlet conditions \mathbf{u}_d on $\partial \Omega_D$ and interface conditions \mathbf{u}_{int} on Γ^{ij} , that modifies the right-hand side of the discretized equations as shown below. Any inexpensive projection step can be employed to construct the function \mathbf{u}_b that is continuous on a subdomain [60].

2.2.10. Discretized equations

Thus, the fully discretized equations on overlapping domains take the form

$$\text{In } \Omega^1 : \begin{pmatrix} \underline{\mathbf{H}} & -\underline{\tilde{\mathbf{D}}}^T \\ -\underline{\tilde{\mathbf{D}}} & 0 \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_0^{[1]} \\ \underline{p}_L^{[1]} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{F}}^{[1]}(\underline{\mathbf{u}}^{[2]}) \\ \underline{F}_p^{[1]}(\underline{\mathbf{u}}^{[2]}) \end{pmatrix}, \tag{59}$$

$$\text{In } \Omega^2 : \begin{pmatrix} \underline{\mathbf{H}} & -\underline{\tilde{\mathbf{D}}}^T \\ -\underline{\tilde{\mathbf{D}}} & 0 \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_0^{[2]} \\ \underline{p}_L^{[2]} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{F}}^{[2]}(\underline{\mathbf{u}}^{[1]}) \\ \underline{F}_p^{[2]}(\underline{\mathbf{u}}^{[1]}) \end{pmatrix}. \tag{60}$$

As it can be seen, the coupling occurs explicitly through the interpolated terms in the right-hand sides in the equations. Here, the following operators have been defined:

$$\underline{\mathbf{A}} = \underline{\mathbf{Q}}^T \underline{\mathbf{M}}_L \underline{\mathbf{A}}_I \underline{\mathbf{M}}_L \underline{\mathbf{Q}}, \tag{61}$$

$$\underline{\mathbf{B}} = \underline{\mathbf{Q}}^T \underline{\mathbf{M}}_L \underline{\mathbf{B}}_I \underline{\mathbf{M}}_L \underline{\mathbf{Q}}, \tag{62}$$

$$\underline{\tilde{\mathbf{D}}} = \underline{\tilde{\mathbf{D}}}_L \underline{\mathbf{M}}_L \underline{\mathbf{Q}}, \tag{63}$$

$$\underline{\mathbf{H}}_L = \left(\frac{\beta_0}{\Delta t} \underline{\mathbf{B}}_L + \frac{1}{\text{Re}} \underline{\mathbf{A}}_L \right), \quad (64)$$

$$\underline{\mathbf{H}} = \left(\frac{\beta_0}{\Delta t} \underline{\mathbf{B}} + \frac{1}{\text{Re}} \underline{\mathbf{A}} \right), \quad (65)$$

with the vectors

$$\underline{\mathbf{F}}^{[i]}(\underline{\mathbf{u}}^{[j]}) = \underline{\mathbf{Q}}^T \underline{\mathbf{M}}_L \left(\underline{\mathbf{B}}_L \underline{\mathbf{f}}_L^{[i]} - \underline{\mathbf{H}}_L \underline{\mathbf{u}}_{b,L}^{[i]}(\underline{\mathbf{u}}^{[j]}) \right) \quad (66)$$

$$\underline{\mathbf{F}}_p^{[i]}(\underline{\mathbf{u}}^{[j]}) = \underline{\tilde{\mathbf{D}}}_L \underline{\mathbf{u}}_{b,L}^{[i]}(\underline{\mathbf{u}}^{[j]}), \quad (67)$$

where dependence on the adjacent domain solution comes from the dependence of the interpolated interface values in the inhomogeneous term $\underline{\mathbf{u}}_{b,L}$. After the coupling term in the right-hand side is updated globally at the beginning of each iteration, the two Stokes problems are solved independently on the overlapping subdomains, as described next. Note that all the operators shown in bold for the vector-valued arrays are block-diagonal extension of the scalar-valued operators, while the derivative operator $\underline{\tilde{\mathbf{D}}}$ is defined as $\underline{\tilde{\mathbf{D}}} = [\underline{\tilde{\mathbf{D}}}_1 \ \underline{\tilde{\mathbf{D}}}_2 \ \underline{\tilde{\mathbf{D}}}_3]$, so that $\underline{\tilde{\mathbf{D}}}^T$ acts on a scalar-valued pressure array p_L .

Recall that $\underline{\mathbf{f}}_L^{[i]}$ is defined in equation (13) where OIFS is used for explicit treatment of the nonlinear convective term. Spatial discretization of the OIFS problem for approximation of $\underline{\mathbf{u}}$ is performed using similar integration procedures to those discussed in Section 2.2, and gives

$$\underline{\mathbf{B}}_L \frac{d}{ds} \underline{\tilde{\mathbf{u}}}(s) = -\underline{\mathbf{C}}_L \underline{\tilde{\mathbf{u}}}(s), \quad t^{n-q} \leq s \leq t^n, \quad (68)$$

where $\underline{\mathbf{C}}$ is the convective operator, the derivation of which is found in [60]. Equation (68) is temporally discretized using an explicit fourth-order Runge–Kutta (RK4) scheme with $\Delta s \leq \Delta t$ (typically $\Delta s = \Delta t/5$ for stability). Note that, although spatial interpolation is not required in the OIFS technique, as discussed in Section 2.2.1, since $\underline{\mathbf{u}}$ is only known at discrete times t^{n-q} , it will need to be temporally interpolated at the intermediate RK stages $t^{n-q} + \Delta s$, $t^{n-q} + 2\Delta s$ etc., with the help of, for example, Fornberg's interpolation.

2.2.11. Stokes solver

The discretized Stokes problem

$$\begin{pmatrix} \underline{\mathbf{H}} & -\underline{\tilde{\mathbf{D}}}^T \\ -\underline{\tilde{\mathbf{D}}} & 0 \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_0^{l,n} \\ \underline{p}^{l,n} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{F}}^{l,n} \\ \underline{\mathbf{F}}_p^{l,n} \end{pmatrix} \quad (69)$$

is solved by decoupling the pressure from the velocity field calculations. Here we omit all the indices related to the subdomain description, as well as drop the subscript “ L ” for unassembled arrays; but we add the superscripts n and l denoting the current timestep and the iteration.

Solving for the solution at timestep n and iteration l , a matrix, $\underline{\mathbf{W}}$, is introduced to the problem along with the pressure solution from the previous timestep

$$\begin{pmatrix} \underline{\mathbf{H}} & -\underline{\mathbf{H}}\underline{\mathbf{W}}\underline{\tilde{\mathbf{D}}}^T \\ -\underline{\tilde{\mathbf{D}}} & 0 \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_0^{l,n} \\ \underline{p}^{l,n} - \underline{p}^{n-1} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{F}}^{l,n} + \underline{\tilde{\mathbf{D}}}^T \underline{p}^{n-1} \\ \underline{\mathbf{F}}_p^{l,n} \end{pmatrix} + \begin{pmatrix} \underline{\mathbf{r}}^{l,n} \\ \underline{\mathbf{0}} \end{pmatrix}. \quad (70)$$

The pressure p^{n-1} from the previous timestep is defined from the final iteration, i.e. $p^{n-1} = p^{l_{\max},n-1}$, and we omit the iteration index for clarity. The value $\underline{\mathbf{r}}^{l,n}$ is the residual, or error due to splitting, taking care of the difference between the original Stokes problem (69) and (70). For this approach

$$\underline{\mathbf{r}}^{l,n} = (\underline{\mathbf{H}}\underline{\mathbf{W}} - \underline{\mathbf{I}}) \underline{\tilde{\mathbf{D}}}^T (\underline{p}^{l,n} - \underline{p}^{n-1}) \quad (71)$$

If $\underline{\mathbf{W}} = \underline{\mathbf{H}}^{-1}$ we arrive at the Uzawa algorithm which gives $\underline{\mathbf{r}}^{l,n} = 0$, but is costly to compute. The choice

$$\underline{\mathbf{W}} = \frac{\Delta t}{\beta_0} \underline{\mathbf{B}}^{-1} - \left(\frac{\Delta t}{\beta_0} \right)^2 \underline{\mathbf{B}}^{-1} \underline{\mathbf{A}} \underline{\mathbf{B}}^{-1} \quad (72)$$

is inexpensive to compute, does not introduce any additional steady-state temporal errors, and preserves the correct boundary conditions [60]. This choice of $\underline{\mathbf{W}}$ gives splitting error

$$\underline{\mathbf{r}}^{l,n} = \left(\frac{\Delta t}{\beta_0} \right)^2 \frac{1}{\text{Re}} (\underline{\mathbf{A}} \underline{\mathbf{B}}^{-1})^2 \underline{\tilde{\mathbf{D}}}^T (\underline{p}^{l,n} - \underline{p}^{n-1}), \quad (73)$$

where it is shown in [71] that the time splitting, $(\underline{p}^{l,n} - \underline{p}^{n-1})$, is $O(\Delta t)$ resulting in residual, $\underline{\mathbf{r}}^{l,n}$, of $O(\Delta t^3)$. Since the velocity in (70) is multiplied by a factor of $\frac{1}{\Delta t}$, neglecting the residual gives fourth-order local truncation error. Dropping the

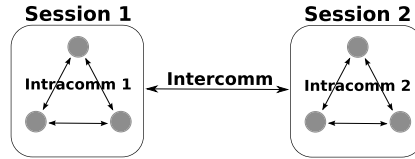


Fig. 3. Schematic illustrating the dual-session communication framework for parallelization, including intracommunicators and the intercommunicator.

residual term and performing block Gaussian elimination the Stokes system of equations (70) yields a decoupled discretized Stokes problem

$$\begin{pmatrix} \underline{\mathbf{H}} & -\underline{\mathbf{H}}\underline{\mathbf{W}}\underline{\tilde{\mathbf{D}}}^T \\ \mathbf{0} & \underline{\tilde{\mathbf{D}}}\underline{\mathbf{W}}\underline{\tilde{\mathbf{D}}}^T \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_0^{l,n} \\ \underline{\mathbf{p}}^{l,n} - \underline{\mathbf{p}}^{n-1} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{f}}^{l,n} + \underline{\tilde{\mathbf{D}}}^T \underline{\mathbf{p}}^{n-1} \\ \underline{\phi}^{l,n} \end{pmatrix} \quad (74)$$

where $\phi^{l,n}$ represents the inhomogeneous terms resulting from the algebraic manipulation. The Helmholtz system is iteratively solved with a Jacobi preconditioned conjugate gradient method, while the Poisson equation for pressure projection is solved iteratively with a multilevel GMRES solver and a Spectral-Element multigrid solver at the coarse-grid level, which, upon determination, is used to compute the correct velocity field (see [60] for greater details regarding the Stokes problem). We see that (74) allows for decoupled pressure solutions which can, in turn, be utilized for velocity solutions [60]. The homogeneous velocity solution \mathbf{u}_0 is corrected at the end of the iteration by adding inhomogeneous term \mathbf{u}_b , as discussed in Section 2.2.9.

3. Parallelization

Communication among separate subdomains is currently accomplished through a dual-session communication framework, which allows for two sessions to perform computations separately while enabling real-time communication with each other in a single computational platform. Each session runs its own independent instance of the SEM solver for its subdomain where MPI intracommunicators are established for communication among the local processors. An intercommunicator is established for global communications between sessions, see Fig. 3. A session uses the intercommunicator to receive interpolated values for use at its interface points and/or send interpolated values for use at the adjacent subdomain's interface points.

Upon initialization, each processor in each session searches for interface points among the elements for which it is responsible. When an interface point is found, it is marked with a flag and boundary conditions at that point are changed to standard Dirichlet. Each processor divides the interface points it found equally between the processors of the neighboring session and communicates the partitioned interface points' coordinates and identities to the corresponding processors in the neighboring session, via the intercommunicator, for efficient load balancing of the intercommunications. Also during initialization, a geometric hash table is created for each subdomain which allows for efficient determination of a point's location within a given subdomain. Once the interface points are collected, processors within the local session compare coordinates of the received flagged boundary points with the boundaries of its subdomain. If a flagged point lies outside of the subdomain, the point is unflagged and sent back to its original session, where it is treated with generic Dirichlet boundary conditions. A search procedure is then performed, comparing the coordinates of each flagged point with element boundaries to determine which processor should be responsible for interpolation procedures for each interface point. In the chance that a point lies exactly on an element boundary the element chosen may be decided by floating point error, though the specific element chosen will not matter so long as the interpolated field is continuous at element boundaries. The interface point information is then redistributed locally within the session to the processors which handle the needed elements. Rapid communication is established locally through a procedure based on the Crystal Router algorithm [72], which uses all-to-all communications for fast data transfer among the session processors, and is built into the SEM solver [73].

The dual-session communication framework, as used in the present method, requires only one global communication per iteration, while the codes can run locally in all the remaining operations, leading to natural parallelization and load balancing. The most expensive procedure in the multidomain communication is the original initialization and the search step which requires solving Newton–Raphson optimization problem, Eqs. (57), (58), for each flagged interface point, in order to identify its local reference coordinates and interpolation weights. This procedure is only needed to be performed once for the whole computation, before the time advancement begins. The interpolation procedure itself, Eq. (56), after the interpolation weights have been found, is based on standard matrix–vector product operations which cost is equivalent to the other basic operations in the SEM [18] thus not effecting the scalability of the solver. Parallel performance is optimized with the Crystal Router Algorithm [73], leading to good scalability and parallel performance of the current method, as documented in the section 5. Our current formulation uses this framework to couple two instances of the same solver, although potential extension to three and more domains and to different codes is possible.

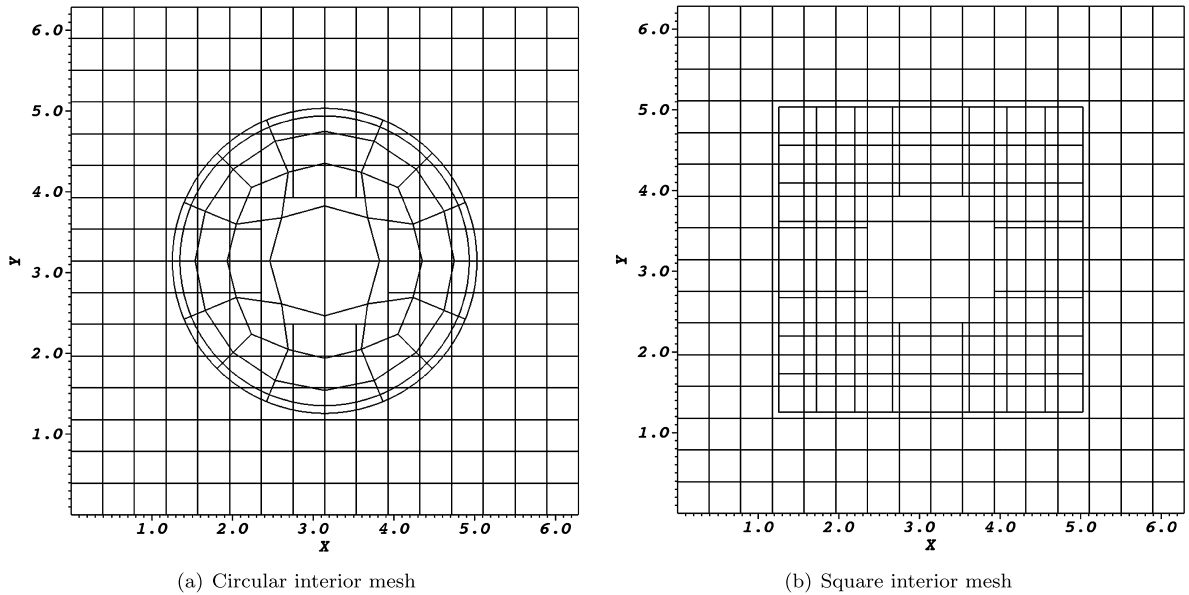


Fig. 4. Two-mesh domains for convecting eddy simulations, with element boundaries shown.

4. Results

In this section, we demonstrate performance of the developed overlapping grid method, on two- and three-dimensional laminar and turbulent problems, with the focus on spatial and temporal orders of convergence, accuracy in the presence of outflow disturbances and long integration times, and the ability of the method to reproduce correct turbulence characteristics on overlapping domains.

4.1. Convecting Walsh's eddies

A simulation of convecting eddies in a two-dimensional periodic global domain was used to assess the spatial and temporal accuracy of the overlapping mesh method since an exact solution to the problem is readily available [74]. Walsh's discussion of non-convecting eddy solutions to the Navier–Stokes Equations states that if the initial conditions are set as $\tilde{\mathbf{u}} = \mathbf{a}$ where \mathbf{a} satisfies $\Delta \mathbf{a} = \lambda \mathbf{a}$, $\nabla \cdot \mathbf{a} = 0$ in Ω , then $\tilde{\mathbf{u}} = e^{\nu \lambda t} \mathbf{a}$, ν is kinematic viscosity, is a solution with the pressure defined as $\nabla \tilde{p} = -\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}$. Here by tilde we denote the values corresponding to a frame of reference where the locations of the eddies do not change with time (or convective frame of reference if the eddies convect). The eigenvalues λ can be expressed in terms of n and m , arbitrary non-negative integers, $\lambda = -(n^2 + m^2)$, with corresponding eigenfunctions formed by the product of $\sin[n\tilde{x}]$ or $\cos[n\tilde{x}]$ with $\sin[m\tilde{y}]$ or $\cos[m\tilde{y}]$. Linear combinations of the eigenfunctions form stream functions satisfying the Navier–Stokes Equations, from which velocity and pressure solutions can be found. For the eddies, convecting with the constant velocity (u_0, v_0) , the Walsh's solutions need to be transformed into a stationary frame of reference by using the coordinate transformation: $x = \tilde{x} + u_0 t$, $y = \tilde{y} + v_0 t$. For our test case, the following exact solutions were used,

$$\begin{aligned}
 u(x, y, t) &= e^{-25tv} (-\cos[5(y - v_0 t)] + \cos[4(y - v_0 t)] \sin[3(x - x_0 t)]) + u_0, \\
 v(x, y, t) &= e^{-25tv} (-\sin[5(x - x_0 t)] - \frac{3}{4} \cos[3(x - x_0 t)] \sin[4(y - v_0 t)]) + v_0,
 \end{aligned} \tag{75}$$

where $\nu = 0.05$, $(u_0, v_0) = (1, 0.3)$. The corresponding solution for pressure (not presented in the original paper by Walsh [74]) is documented in Appendix A, together with the further discussion of the derivation. Since pressure in the current $\mathbb{P}_N - \mathbb{P}_{N-2}$ formulation is not prescribed as a boundary condition, it is defined up to an arbitrary constant, thus we compute the pressure with the zero mean, both in numerical and exact solution.

4.1.1. Mesh configuration

The overlapping mesh configuration used for the present simulation consists of an exterior mesh with a vacancy in the middle and an interior mesh, either circular or square, that covers the vacancy (Fig. 4). Note that the grid points (including on the interface boundaries) are not coincident with grid points in the adjacent subdomain. This is done in order to test for correct interpolation of values between the two meshes. Single mesh simulations, which were performed for comparison with two-mesh simulations, used the same mesh as the exterior mesh shown in Fig. 4, but with the vacancy filled.

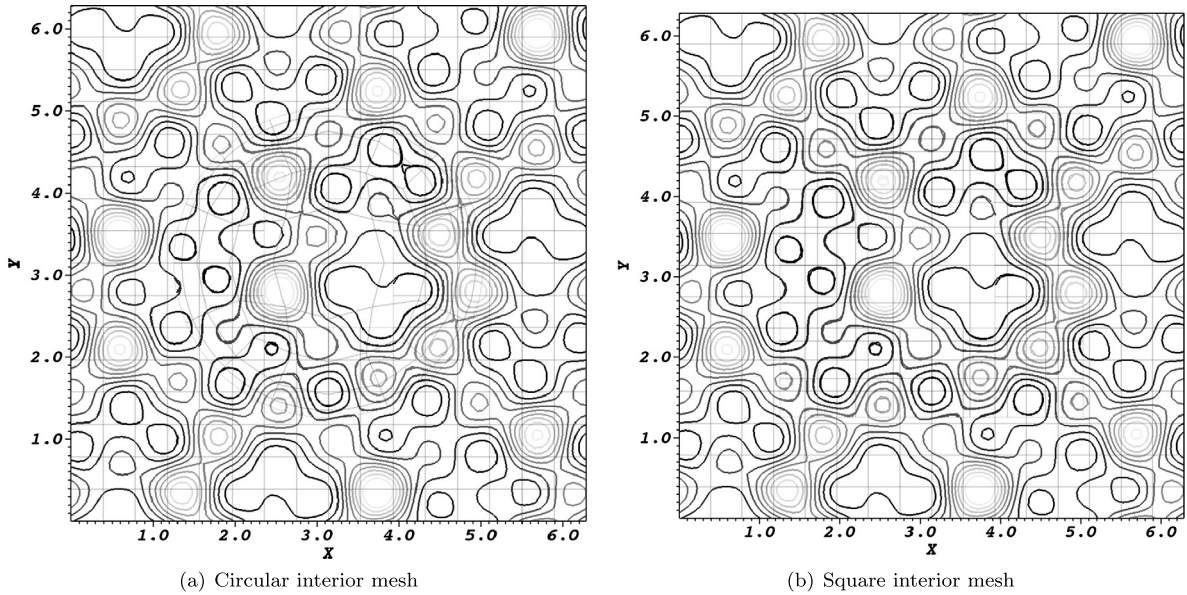


Fig. 5. Pressure contours taken after the 600th timestep using the 7th order polynomial approximation and third order temporal scheme with $\Delta t = 1 \cdot 10^{-4}$. IEXT3 was performed using two iterations. The darkest lines represent a pressure of 0.8078 while the faintest lines represent a pressure of -2.710 .

Fig. 5 presents the pressure contours for the overlapping simulations with circular and square interior meshes. The snapshots contain two sets of contour lines in sections where the subdomains overlap, although the continuity between the values in the two meshes make their differences almost indistinguishable.

4.1.2. Accuracy

In this section, we investigate spatial and temporal convergence of the grid overlapping method. We calculate the L^2 velocity error in each subdomain as

$$L^2 \text{ error } (\mathbf{u}^{[i]}) = \sqrt{\frac{\int_{\Omega^i} (\mathbf{u}_{\text{ex}}^{[i]} - \mathbf{u}_{\text{comp}}^{[i]})^2 dV}{2 V^{[i]}}}, \tag{76}$$

where $V^{[i]}$ is the volume of the subdomain Ω^i (the factor of two in denominator arises from the fact that two volume integrals are summed together, one for u velocity and one for v velocity, thus dividing by two results in the average), and the L^2 pressure error as

$$L^2 \text{ error } (p^{[i]}) = \sqrt{\frac{\int_{\Omega^i} (p_{\text{ex}}^{[i]} - p_{\text{comp}}^{[i]})^2 dV}{V^{[i]}}}, \tag{77}$$

where $\mathbf{u}_{\text{ex}}^{[i]}$, $p_{\text{ex}}^{[i]}$ are the exact values and $\mathbf{u}_{\text{comp}}^{[i]}$, $p_{\text{comp}}^{[i]}$ are the computational approximations.

The spatial accuracy plots (Fig. 6) show clear spectral convergence with p -refinement with respect to L^2 -errors in velocity and pressure. Notice also that a comparison of the plots for the circular and square interior meshes yields very similar results; the spatial errors for the simulation with a circular interior mesh being slightly larger than those for the square interior mesh case. This effect is due to a slightly slower (albeit still exponential) convergence of the SEM method with curvilinear meshes attributed to additional errors introduced when the mapping Jacobian is used in the derivation of the operators [75,60,76], and not due to the overlapping procedure. We also recognize that the spatial errors reported for single mesh simulations are slightly lower than their double mesh counter-parts. This is attributed to the lower h -resolution of the interior mesh (see Fig. 4), which also affects the accuracy of values used at interface boundaries of the exterior mesh. The ratio of element sizes in single/exterior/interior meshes is about 1/1/1.44.

The temporal accuracy plots (Figs. 7 and 8) contain the results of simulations performed with IEXT k ($k = 1-3$) and third order time stepping compared with Δt , $(\Delta t)^2$, $(\Delta t)^3$ plots. The temporal convergence analysis was performed with 17th order polynomials to make certain that the problem was fully converged spatially to ensure more accurate determination of temporal convergence rates. However, since full spatial convergence is achieved in the current test problems for polynomial orders greater than 12, subsequent temporal convergence tests as discussed below use lower order polynomials. As expected, the order of interface extrapolation is the limiting factor for the temporal accuracy, even though the third order temporal scheme was used in all cases. The results of the simulations generally meet the expected order of temporal convergence.

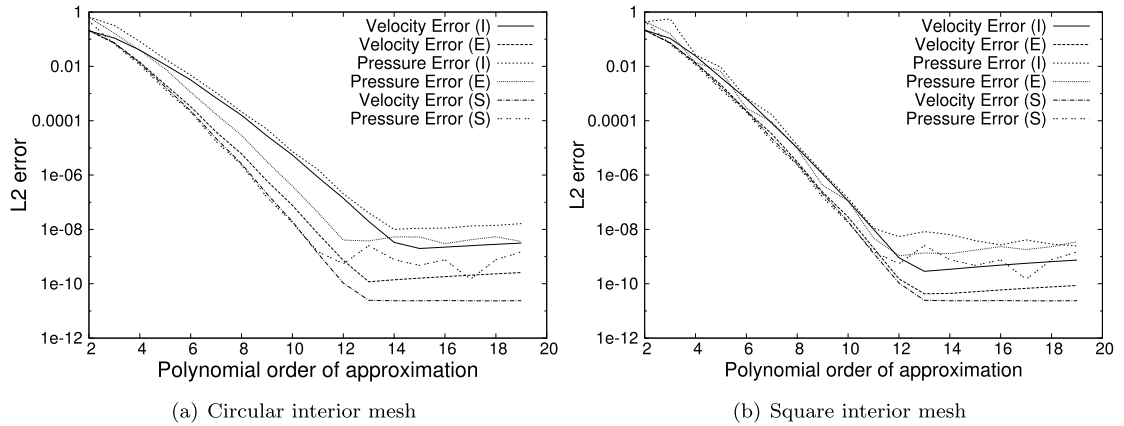


Fig. 6. Spatial accuracy for simulations using the third order time stepping scheme, with IEXT3 using three iterations. The I, E, and S in the key refer to the interior mesh in the two mesh simulation, the exterior mesh in the two mesh simulation, and the single mesh (square) simulation, respectively. The errors were collected after 1000 timesteps for simulations with $\Delta t = 1 \cdot 10^{-4}$.

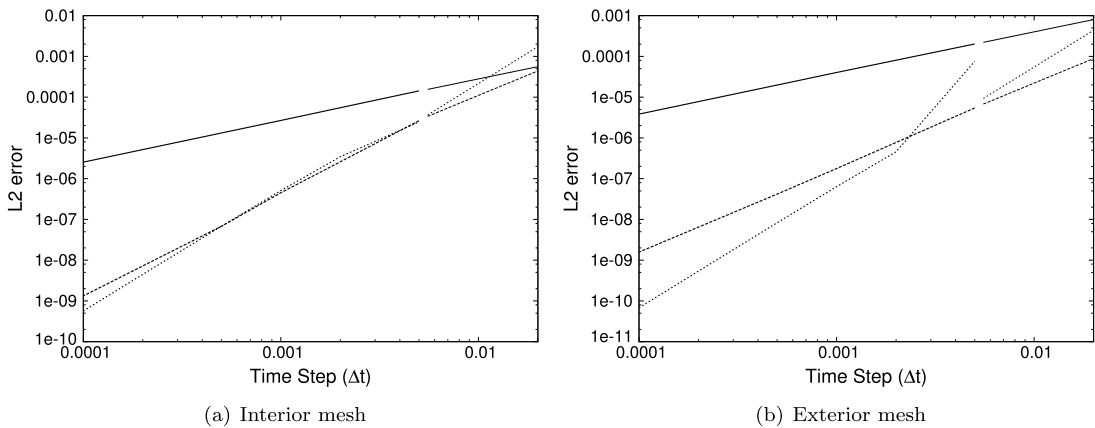


Fig. 7. Temporal accuracy of the velocity for the square interior mesh simulation. The solid line represents the results for IEXT1, the dashed line represents IEXT2, and the dotted line represents IEXT3. In each case, the IEXTk scheme used three iterations. The lines to the right of the break are plots of Δt , $(\Delta t)^2$, $(\Delta t)^3$ for comparison with the simulation results. These results were collected for the 17th polynomial order of approximation, at time $1 \cdot 10^{-1}$.

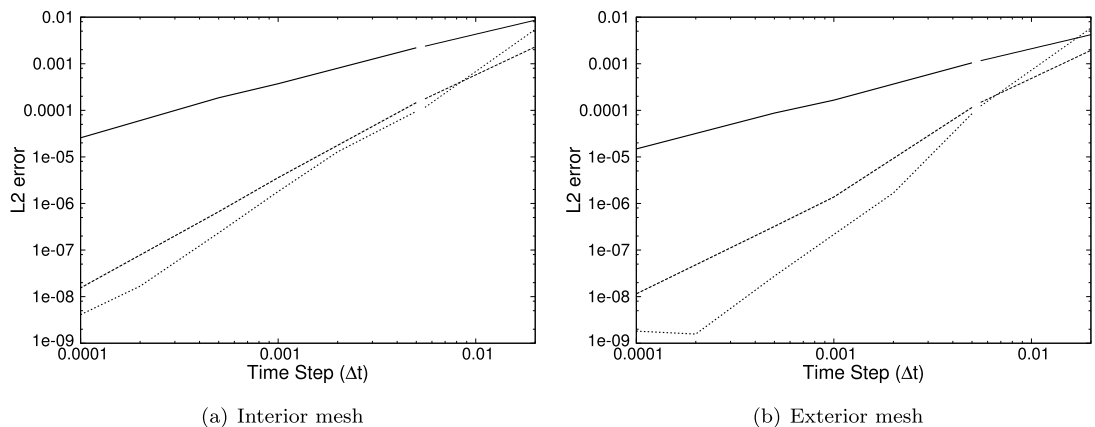


Fig. 8. Temporal accuracy of the pressure for the square interior mesh simulation. The solid line represents the results for IEXT1, the dashed line represents IEXT2, and the dotted line represents IEXT3. In each case, the interface extrapolation used three iterations. The lines to the right of the break are plots of Δt , $(\Delta t)^2$, $(\Delta t)^3$ for comparison with the simulation results. These results were collected for the 17th polynomial order of approximation, at time $1 \cdot 10^{-1}$.

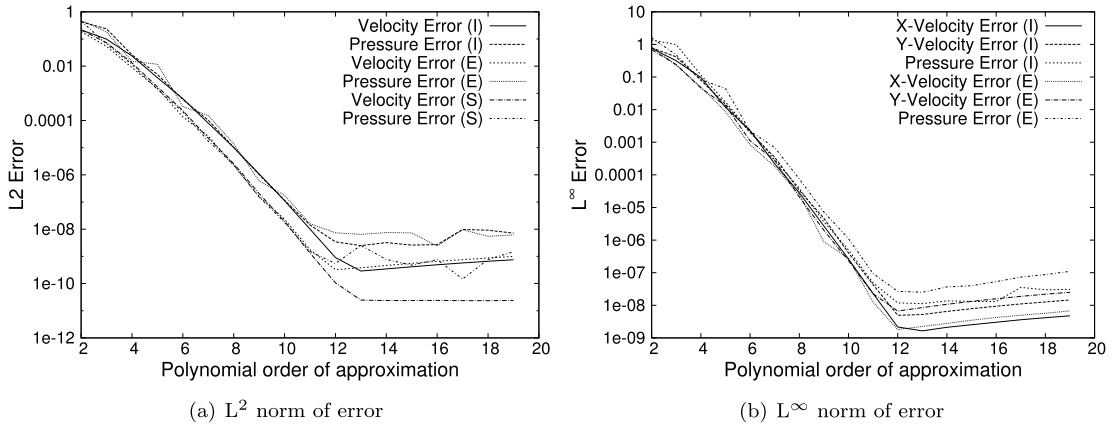


Fig. 9. Spatial accuracy for square interior mesh tests with global Dirichlet boundary conditions using the third order time stepping scheme, with IEXT3 using three iterations. The I, E, and S in the key refer to the interior mesh in the two mesh simulation, the exterior mesh in the two mesh simulation, and the single mesh (square) simulation, respectively. The errors were collected after 1000 timesteps for simulations with $\Delta t = 1 \cdot 10^{-4}$.

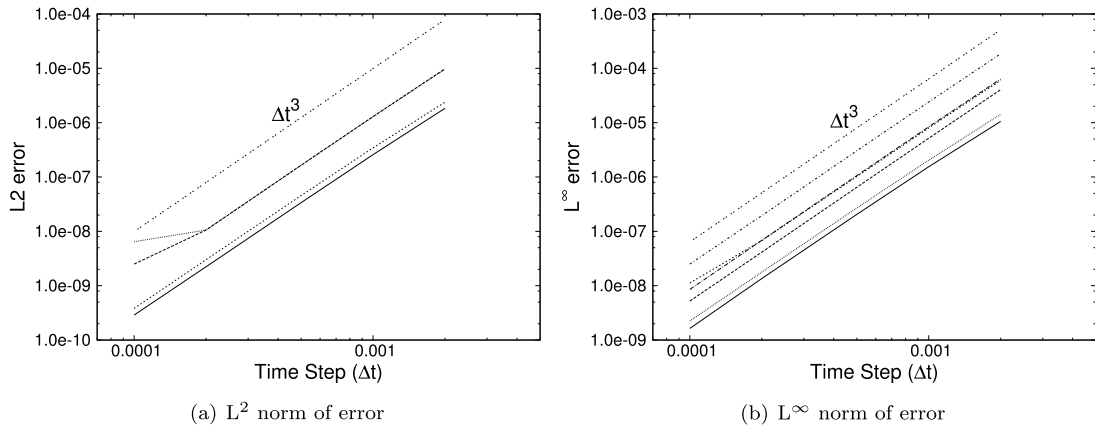


Fig. 10. Temporal accuracy for square interior mesh simulations with global Dirichlet boundary conditions using the third order time stepping scheme, with IEXT3 using three iterations. The labeled line shows the expected order of temporal convergence, while the lines for velocities and pressures are displayed according to the legends in Fig. 9 (single mesh data is not included here). These results were collected for the 13th polynomial order of approximation, at time $1 \cdot 10^{-1}$.

Figs. 9 and 10 illustrate the errors given by the same eddy simulation with Dirichlet boundary conditions prescribing the exact velocity at the global domain boundary as a more stringent test for convergence. Both L^2 and L^∞ errors are plotted for spatial and temporal convergence. Looking at the L^∞ norm of the error, defined by

$$L^\infty \text{ error } (u^{[i]}) = \frac{\max \left(\left| u_{\text{ex}}^{[i]} - u_{\text{comp}}^{[i]} \right| \right)}{\max \left(\left| u_{\text{ex}}^{[i]} \right| \right)}, \tag{78}$$

shows that with this error norm the solution within the global domain also achieves spectral spatial convergence and 3rd order temporal convergence for both velocity and pressure.

Fig. 11 shows the absolute errors of velocity magnitude in the interior and exterior subdomains. Since Dirichlet conditions are used on the global boundaries, it is expected that the largest errors in the exterior subdomain should be near the interface boundaries, as is the case. Interestingly, the largest errors within the interior subdomain do not lie near interface boundaries, but correspond to areas of largest fluid velocity magnitude. While there are distinct concentrated areas of larger error, all errors throughout the global domain are of the same order of magnitude.

Investigation of the convergence rates (of the L^∞ norm of error) with respect to refinement of the average element size is illustrated in Figs. 12 and 13 for square and circular interior mesh simulations with solutions approximated by 5th and 7th order polynomials. We see that the h -refinement convergence tests generally show rates for velocity error of order h^N with square meshes, and convergence rates for pressure error of order h^{N-1} since pressure solutions are approximated on G rather than GL quadrature. Circular interior mesh simulations give velocity error convergence rates that are of order h^{N-1} , while pressure error generally maintains convergence rates of order h^{N-1} , as seen previously. Errors in circular mesh cases cease to decline at coarser resolutions than when square meshes are used. The lower order h -refinement convergence rate

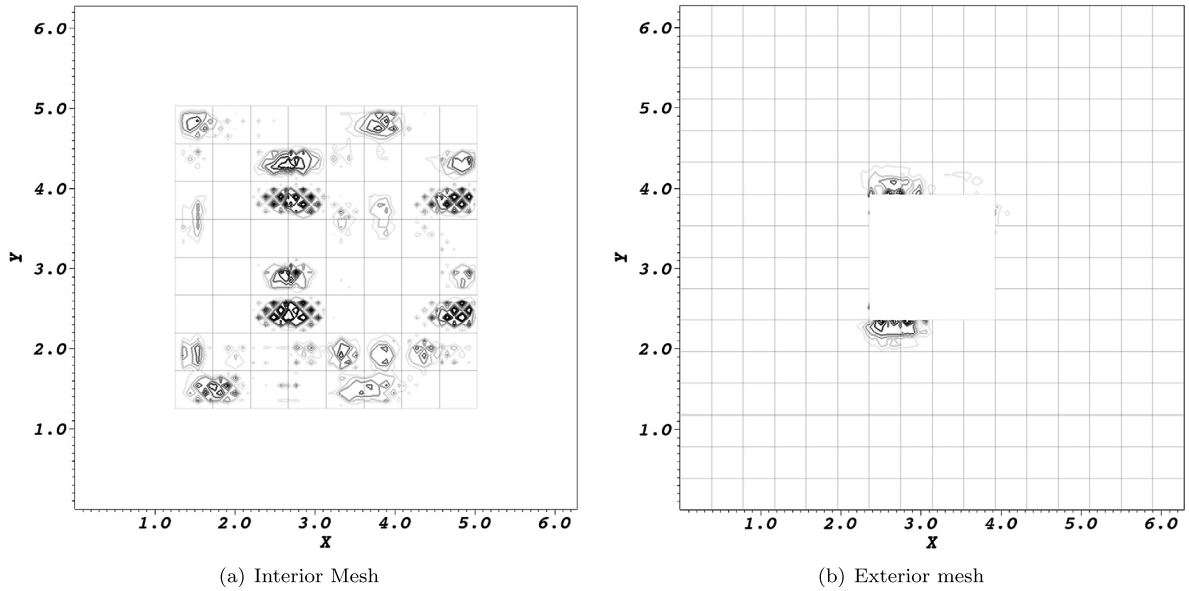


Fig. 11. Contour plot of the absolute error in the interior and exterior subdomains for a simulation with global Dirichlet boundary conditions, using 7th order polynomials, third order timestepping, and IEXT3 using three iterations. The snapshot is taken at time $1 \cdot 10^{-1}$. The darkest contour lines correspond to a value of $2.5 \cdot 10^{-4}$ and lightest contour lines $1.0 \cdot 10^{-4}$. The exterior subdomain has a maximum error of $2.799 \cdot 10^{-4}$ and the interior subdomain has a maximum error of $3.48 \cdot 10^{-4}$.

of the circular mesh is likely the leading cause of the slower, though still exponential, p -refinement convergence rate of the circular mesh illustrated in Fig. 6 and can be explained by the same mechanisms of loss of accuracy in numerical quadrature in the presence of Jacobian matrices [75,60,76].

The velocity and pressure profile plots for the overlapping mesh simulation are presented in Fig. 14. As illustrated by the profile plots, good continuity exists in the transitions between the two meshes.

4.2. Convecting Taylor vortex

We have performed a computation of a two-dimensional convecting vortex traveling through a long, narrow domain composed of two overlapping subdomains with inflow–outflow global boundary conditions to test: 1) error behavior over long integration times; 2) influence of inexact outflow conditions on the overlapping grid solution. For that, we have constructed two overlapping meshes to be long and narrow, with an overlapping region in the center of the global domain (see Fig. 15). Single mesh tests were performed on a grid with the same global dimensions which has the same spatial resolution as the left mesh. The vortex enters the left boundary of the global domain, traveling with a constant prescribed convecting velocity U_∞ , ultimately exiting the right boundary. The top, bottom, and left boundaries of the global domain are assigned Dirichlet boundary conditions for velocity, and the right boundary is assigned outflow conditions, which, in the current $\mathbb{P}_N - \mathbb{P}_{N-2}$ formulation where the pressure at the boundary is not prescribed, reduces to the homogeneous Neumann conditions on velocity, $\nabla \mathbf{u} \cdot \mathbf{n} = 0$ on $\partial\Omega_O$, where $\partial\Omega_O$ is the outflow boundary, and \mathbf{n} is the vector normal to the outflow boundary.

The tangential velocity of a single incompressible Taylor vortex [77,12,78] in a convective frame of reference is prescribed by the equation

$$\tilde{u}_\theta = \frac{\mathcal{M}r}{16\pi\nu^2t^2} \exp\left(\frac{-\tilde{r}^2}{4\nu t}\right), \quad (79)$$

where \mathcal{M} is an invariant of the flow. As in the previous section, the variables with a tilde correspond to a convective frame of reference. The parameters t , ν , and \mathcal{M} are altered to obtain the desired vortex radius \tilde{r}_d , which is defined as the distance from the center of the vortex to the location of the maximum tangential velocity, and the desired maximum tangential velocity, $\tilde{u}_{\theta \max}$. A relationship for the vortex radius is found by solving $\frac{d\tilde{u}_\theta(r)}{d\tilde{r}} = 0$, giving

$$\tilde{r}_d = \sqrt{2\nu t_0} \quad (80)$$

from which ν and an arbitrarily chosen time, t_0 , can be manipulated to give the desired radius. Using this relationship, an equation is derived for \mathcal{M} that gives a vortex with the desired radius and maximum tangential velocity

$$\mathcal{M} = \frac{16\pi \cdot \tilde{u}_{\theta \max} \cdot \nu^2 \cdot t_0^2}{\sqrt{2\nu t_0}} e^{1/2}. \quad (81)$$

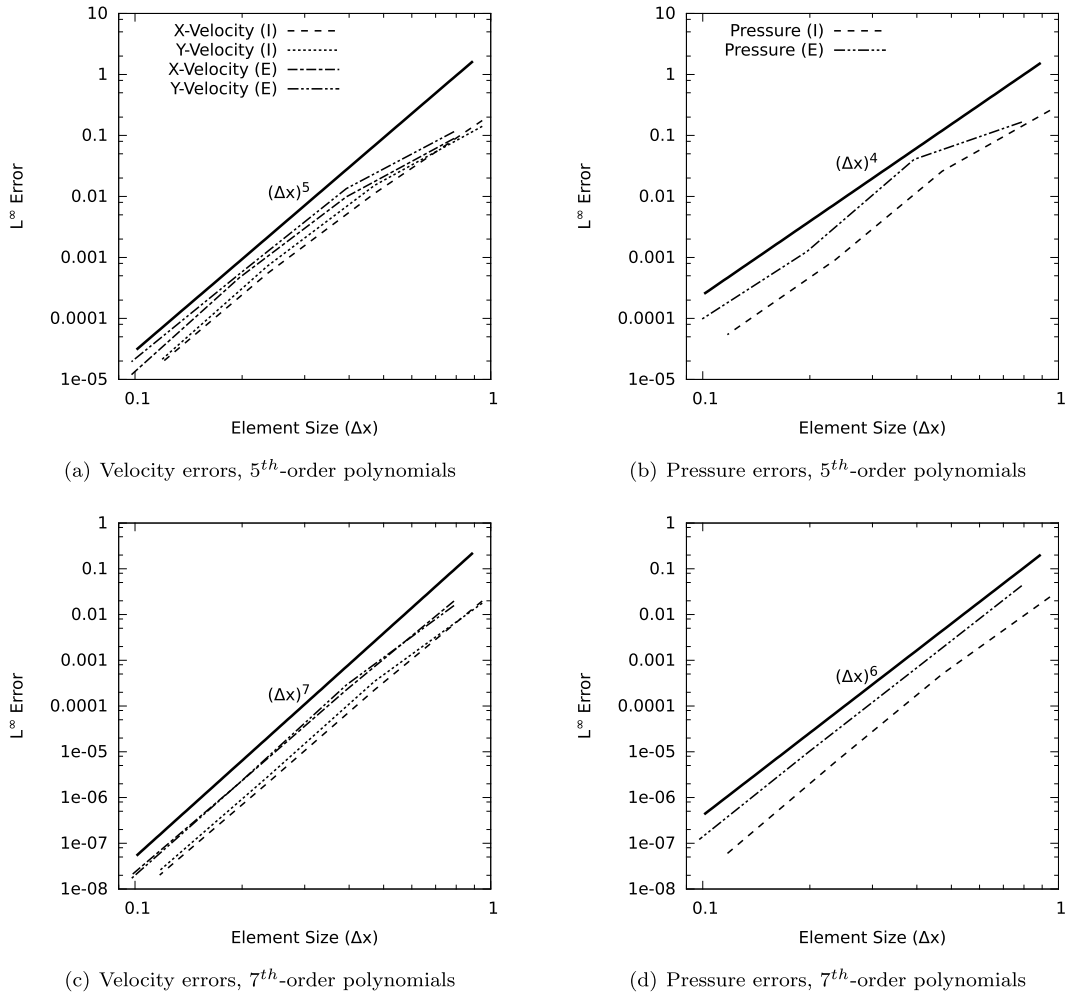


Fig. 12. L^∞ error with respect to element size (Δx) for square interior mesh simulations. Simulations were performed with $\Delta t = 1 \times 10^{-4}$ with the third order temporal scheme and IEXT3 using three iterations. Bold solid lines are the convergence rate reference lines, and are labeled with the corresponding convergence rate.

Since only a tangential velocity component exists, the momentum equation in the radial direction reduces to

$$\frac{\tilde{u}_\theta^2}{\tilde{r}} = -\frac{1}{\tilde{\rho}} \frac{\partial \tilde{p}}{\partial \tilde{r}}. \tag{82}$$

Thus, solving for the pressure of a non-convecting Taylor vortex gives

$$\tilde{p} = -\frac{\mathcal{M}^2 \tilde{\rho}}{256\pi^2 \nu^3 t^3} \exp\left(\frac{-\tilde{r}^2}{2\nu t}\right), \tag{83}$$

where \mathcal{M} is defined in equation (81).

Solution of the convecting vortex in a stationary frame of reference is then obtained by transforming polar coordinates of a convective frame into Cartesian coordinates of a stationary frame and adding $(U_\infty, 0)$ to the corresponding (\tilde{u}, \tilde{v}) velocity, while pressure is obtained from (83) with the account for coordinate transformation, as in Appendix A. As in the previous section, pressure with the zero mean is taken in both exact and numerical solutions.

In the present simulations, the maximum tangential velocity is given as a percentage of the convecting velocity, namely $\tilde{u}_{\theta \max}/U_\infty = 100\%$, which represents a very strong disturbance and a demanding test for the numerical method. Simulations were performed at a Reynolds number ($\text{Re} = U_\infty \tilde{r}_d/\nu$) of 140.

4.2.1. Accuracy

As the vortex travels through the domain the errors remain fairly constant (Fig. 16). Note that in Fig. 16 we do not include the errors as the vortex leaves the global domain through the outflow boundary to concentrate our attention on the errors as

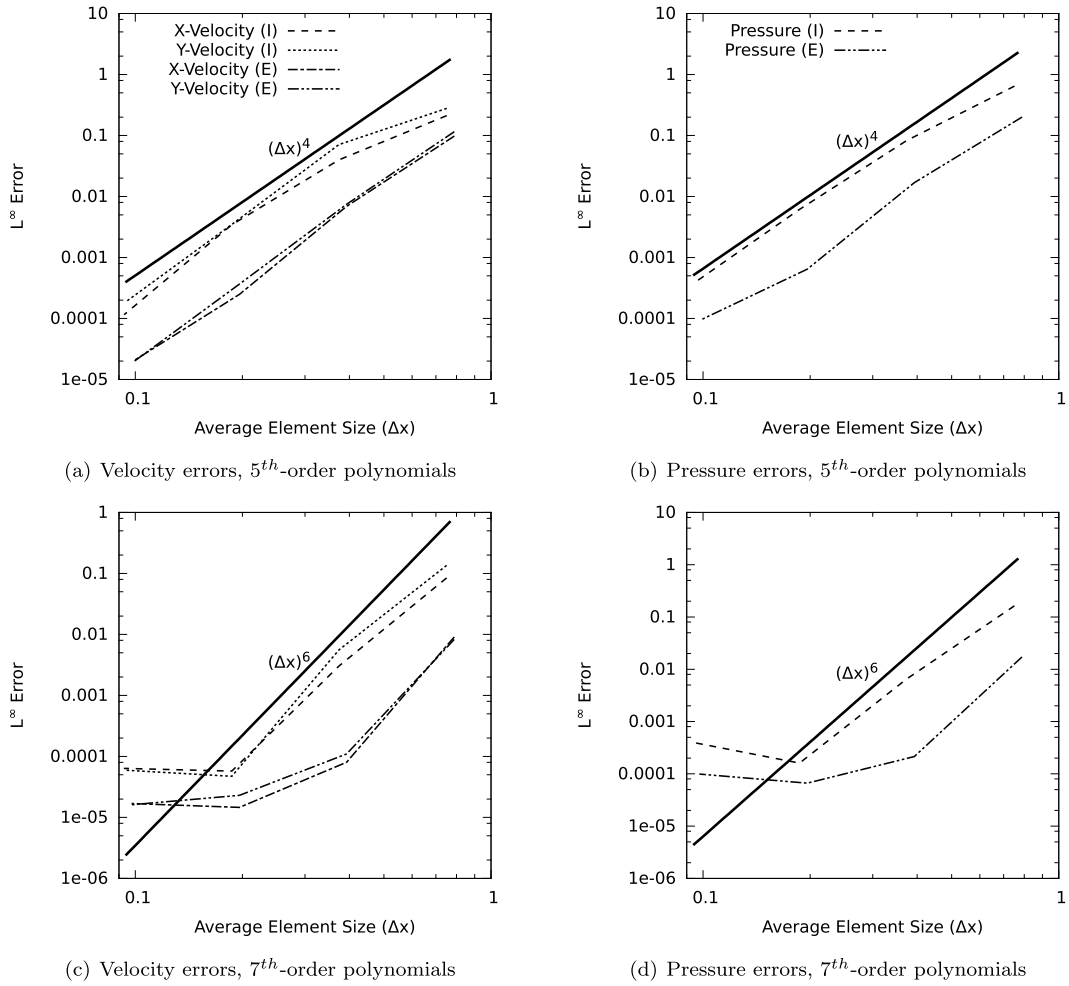


Fig. 13. L^∞ error with respect to average element size (Δx) for circular interior mesh simulations. See caption of Fig. 12 for runtime parameters.

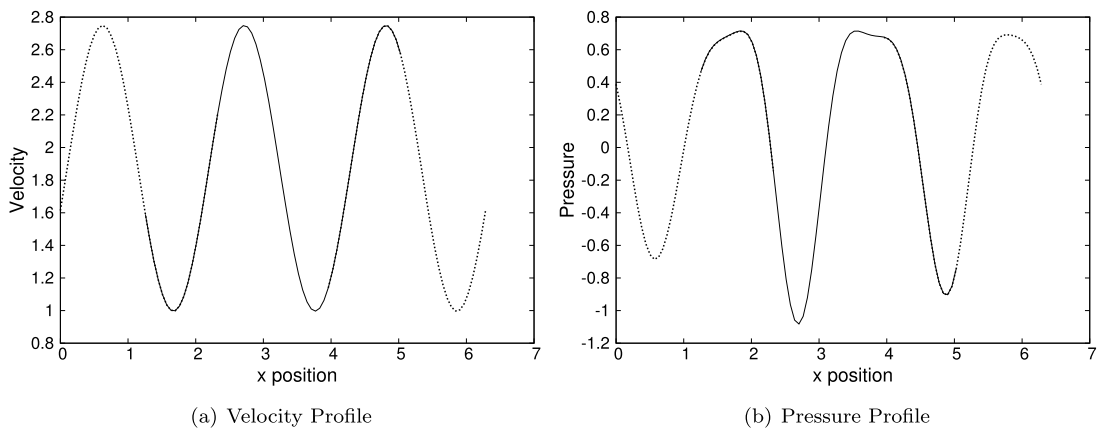


Fig. 14. Cross-section profiles of the square interior mesh case through the center of the domain: The solid line illustrates the interior mesh data and the dotted line illustrates the exterior mesh data. These profiles were taken after 1000 timesteps from a third order simulation with IEXT3 using three iterations. The 7th order polynomial approximation was used with $\Delta t = 1 \cdot 10^{-4}$.

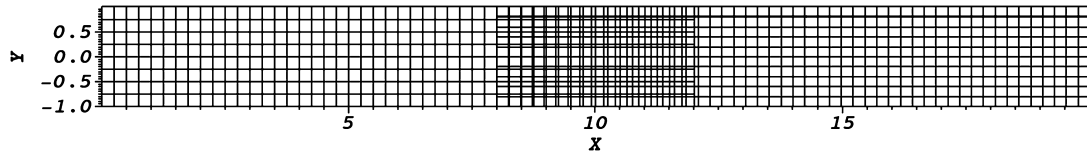


Fig. 15. Single vortex mesh configuration showing element boundaries.

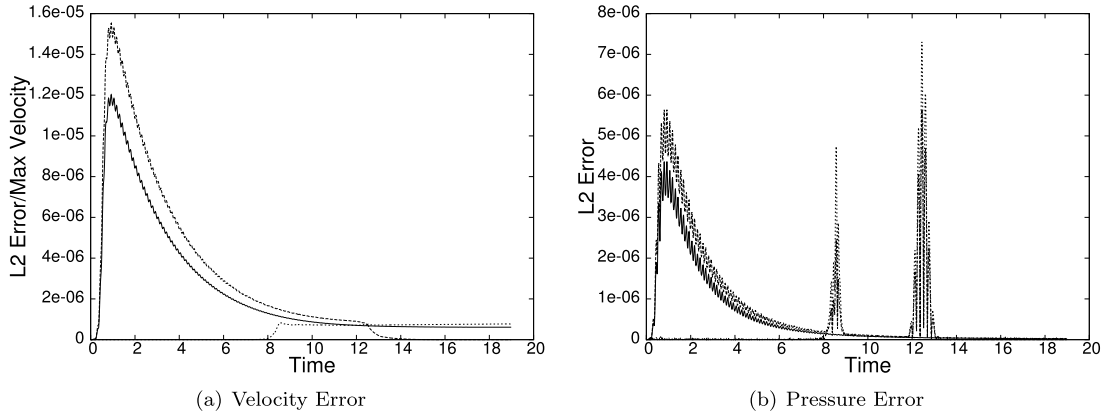


Fig. 16. L^2 -errors as the vortex travels through the domain, where 9th order polynomials are used for spatial approximations and third order time-stepping was used with $\Delta t = 1 \times 10^{-4}$. IEXT3 with four iterations was used for interface values. The solid line represents the errors of the single mesh configuration, the dashed line represents the errors in the left mesh of the two-mesh case, and the dotted line represents the errors in the right mesh. The velocity errors have been normalized by the corresponding value of the maximum velocity at each time.

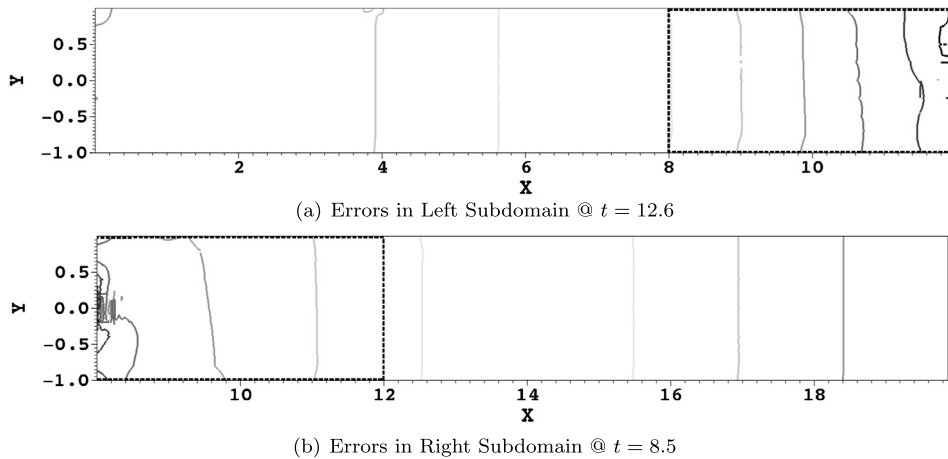


Fig. 17. Pressure errors ($|p_{\text{exact}} - p_{\text{comp}}|$) throughout the domain as the vortex enters (b) and leaves (a) the overlapping region. The overlapping region within each subdomain is outlined with dashed lines. 9th order polynomials are used for spatial approximations and third order time-stepping was used with $\Delta t = 1 \times 10^{-4}$. IEXT3 with four iterations was used for interface values. The darkest contour lines represent an error value of 1.4×10^{-6} and the lightest contours represent an error value of 2.3×10^{-7} .

the vortex travels into and out of the overlapping region. The errors as the vortex leaves through the outflow boundary are several orders of magnitude larger (on the order of 10^{-3} for velocity and pressure) due to the outflow boundary prescribing boundary conditions that do not correspond to the exact solution. It is known that approximate outflow boundary conditions are detrimental for outgoing disturbances. The errors due to the outflow conditions are however localized in time and space and correspond to the moment when the vortex leaves the global domain and are confined to the outflow boundary region, see Fig. 18. The contours are given on a log-scale so that the faintest line represents a pressure error value of 5×10^{-6} while the darkest lines, representing a pressure error value of 1×10^{-3} , are localized at the outflow boundary. The large errors experienced as the vortex crosses the global outflow boundary occur in both single mesh and two mesh problems. The initial conditions prescribed in this problem place the vortex to the left of the global domain, giving nearly uniform unidirectional flow throughout the computational domain, thus initial computational errors are very low. As the vortex convects into the computational domain we see that the error values increase, as expected, for both single mesh and

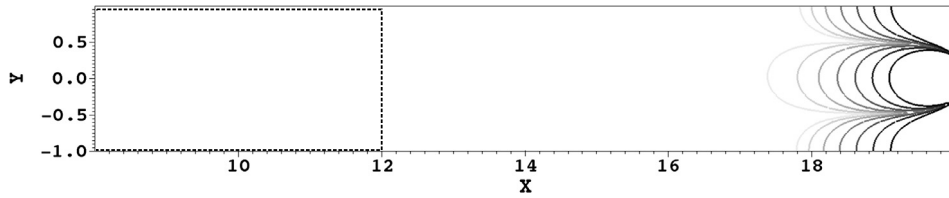


Fig. 18. A snapshot of the pressure error contours at $t = 20.5$ when the errors due to the vortex leaving the global domain reach a maximum; right subdomain is shown. The overlapping region is outlined with dashed lines. The contour lines are given in logscale with the darkest lines representing a value of 1×10^{-3} and the lightest lines representing 5×10^{-6} .

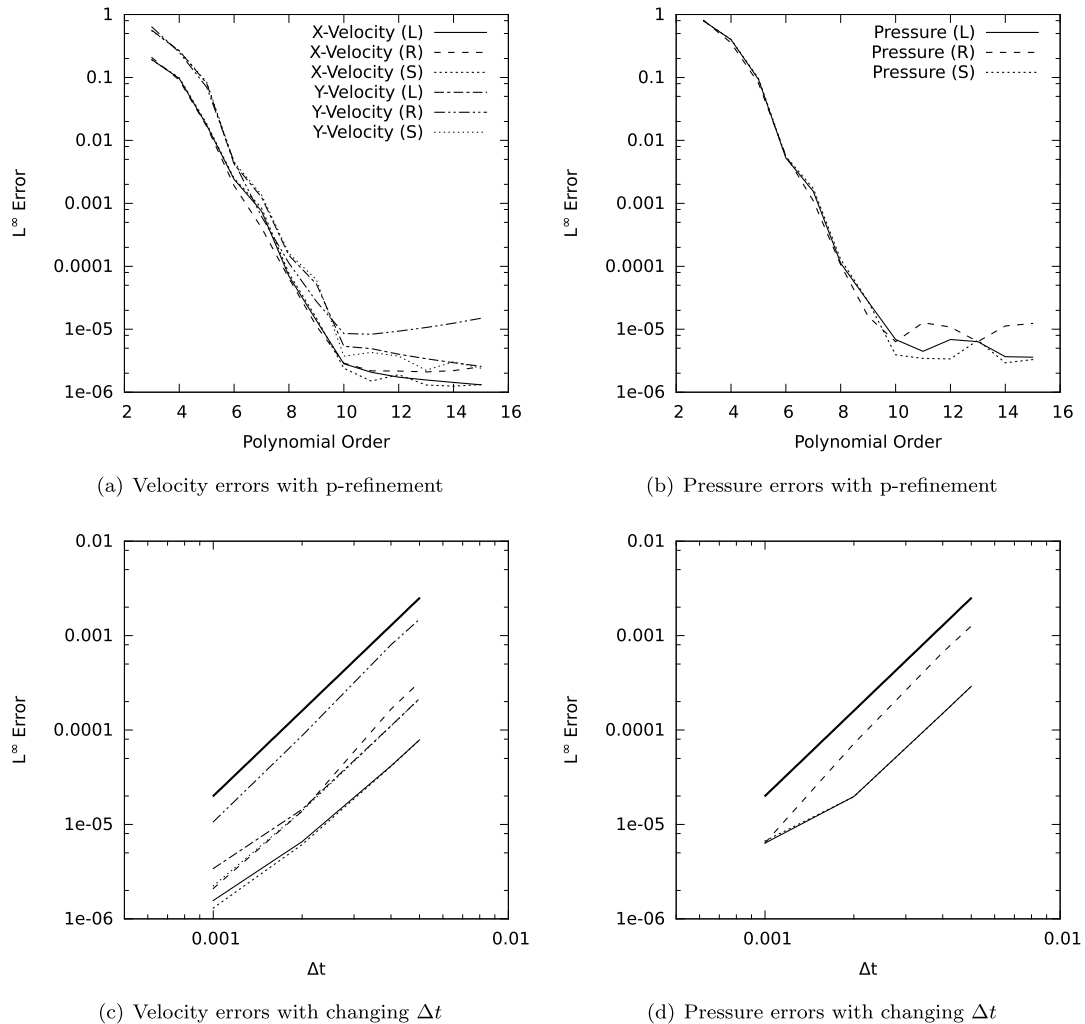


Fig. 19. L^∞ errors with respect to polynomial order and Δt refinement for the left (L) and right (R) meshes in two mesh simulations and for single mesh (S) simulations. In the temporal convergence plots, the line types correspond to those used in the spatial convergence plots, and the bold solid line is the Δt^3 reference. Spatial convergence plots used timestep $\Delta t = 1 \times 10^{-3}$, and temporal convergence plots were performed with 14th-order polynomials. In all cases, the BDF3/EXT3 schemes were used with four iterations per timestep.

two mesh problems, after which the error gradually decreases. As the vortex enters and leaves the overlapping region the pressure errors spike since pressure is not interpolated at interface boundaries, although the errors are localized and do not significantly propagate into the domain (see Fig. 17). The velocity errors remain largely unaffected by the vortex movement across the interfaces. Convergence tests, illustrated in Fig. 19, show spectral convergence with respect to polynomial-order refinement, and the expected third-order temporal convergence. Errors were recorded when the vortex was centered in the overlapping region.

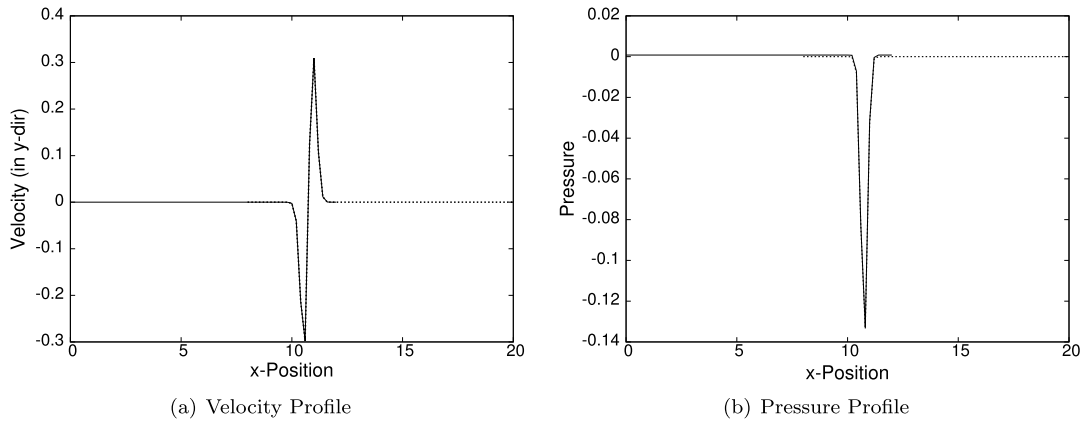


Fig. 20. Profile plots through the center of the vortex when the vortex resides in the overlapping region: The left mesh data is illustrated by a solid line while the right mesh data is illustrated by a dotted line. 9th order polynomials and a third order temporal scheme with $\Delta t = 1 \times 10^{-4}$ were used for solution approximations. Four iterations were used each timestep with the IEXT3 scheme.

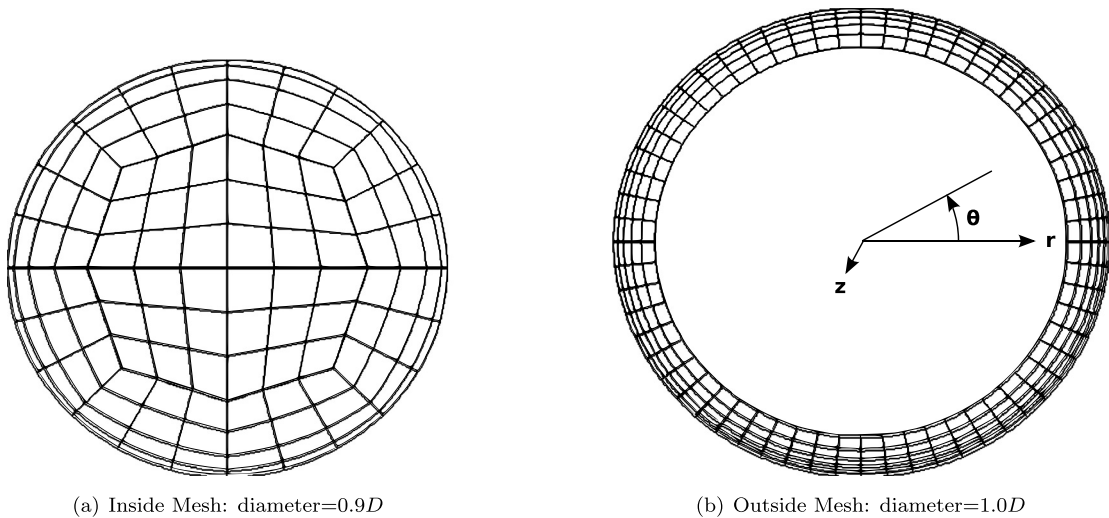


Fig. 21. Cross section views of the inside and outside meshes for turbulent pipe flow (only the element boundaries are shown and not the GL points).

The profile plots in Fig. 20 are taken through the center of the vortex when the vortex resides in the overlapping region. The values in the overlapping region show strong agreement between results in the two subdomains.

4.3. Turbulent pipe flow

In this section, we perform a simulation of fully-developed turbulent pipe flow to assess the applicability of the developed overlapping grid method to three-dimensional turbulent flow cases. A direct numerical simulation (DNS) of the flow was performed with a bulk Reynolds number $Re_b = \frac{\bar{U}_b D}{\nu} = 5300$ to correlate with experiments performed in [79]. Sinusoidal perturbations in the θ and z directions were given in the initial flow conditions to enable more rapid transition to a turbulent state. The results were compared with the experimental and numerical data widely available from [79].

4.3.1. Mesh configuration

A cross-sectional view of the inside and outside meshes is illustrated in Fig. 21, the length of the global pipe domain $L = 6D$, where D is the pipe diameter. The inside mesh contains a total of 1716 elements while the outside mesh contains 6000 elements, giving $\sim 0.6M$ and $\sim 2.1M$ gridpoints in the inside and outside meshes respectively using 6th-order polynomial approximations. Periodic boundary conditions are assigned at the ends of the global pipe domain. Single mesh simulations were performed using the rescaled version of the inside mesh of Fig. 21, though with 7th-order polynomial approximations.

The overlapping grid methodology allows for efficient placement of refined elements in the locations where it is most needed. Thus, our outside mesh has much finer resolution in all radial, azimuthal and streamwise directions, enabling accurate representation of the boundary layer structures, while the inside mesh can be significantly coarser, minimizing the total number of elements and improving aspect ratios of the cells compared to a single domain. Since Gauss–Lobatto

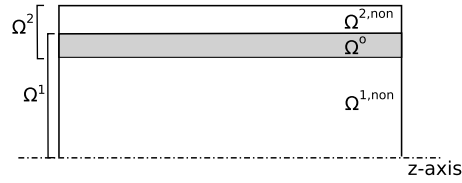


Fig. 22. An axisymmetric cross section view of the inside and outside subdomains. Shaded areas denote regions where inside and outside subdomains overlap. Different regions are represented by different superscripts, namely, 'o' denotes an overlapping region, and 'non' a non-overlapping region, while subdomain reference numbers are also included.

quadrature is used to determine the positions of the nodes between element boundaries, the spacing is not constant. Given this grid structure, the largest spacing among the nodes in elements near the center of the pipe (given in Cartesian coordinates) are $\Delta x^+ = \Delta y^+ \approx 8.121$ ($\Delta x = \Delta y \approx (2.250 \times 10^{-2})D$) and $\Delta z^+ \approx 39.044$ ($\Delta z \approx (1.082 \times 10^{-1})D$). (Wall units, or plus units, are given by $x^+ = u_\tau x/\nu$ and so forth for the other coordinate directions, where $u_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity.) The smallest spacing among nodes in the center-most elements are $\Delta x^+ = \Delta y^+ \approx 2.941$ ($\Delta x = \Delta y \approx (8.150 \times 10^{-3})D$) and $\Delta z^+ \approx 14.138$ ($\Delta z \approx (3.918 \times 10^{-2})D$). The largest spacings among nodes in the outermost element of the outside mesh are $\Delta r^+ \approx 0.7311$ ($\Delta r \approx (2.026 \times 10^{-3})D$), $(\Delta r\theta)^+ \approx 4.4$ ($\Delta r\theta \approx (1.219 \times 10^{-2})D$), $\Delta z^+ \approx 25.4$ ($\Delta z \approx (7.039 \times 10^{-2})D$), while the smallest spacings (including nearest the wall) are $\Delta r^+ \approx 0.2648$ ($\Delta r \approx (7.338 \times 10^{-4})D$), $(\Delta r\theta)^+ \approx 1.6$ ($\Delta r\theta \approx (4.434 \times 10^{-3})D$), $\Delta z^+ \approx 9.2$ ($\Delta z \approx (2.550 \times 10^{-2})D$). We note that the width of the outermost element near the wall is $\Delta r_{el}^+ = 3.1$ ($\Delta r_{el} = (8.590 \times 10^{-3})D$) which means that this entire element lies well within the viscous sublayer which has a thickness of about 5 wall units. Comparatively, the point nearest the wall given in [79] is $\Delta r^+ = 0.94$. The global domain size in wall units, for a reference, is $D^+ \approx 360.9$ and $L^+ \approx 2165$.

4.3.2. Forcing

In the absence of a prescribed pressure gradient, the flow rate in a periodic pipe will eventually tend to zero due to viscous dissipation. Thus, a mechanism for sustaining the flow rate must be implemented in the simulations. In this work, we utilize a flow control scheme based upon incremental adjustment of the pressure gradient for maintenance of the bulk flow rate, U_b . The adjusted pressure gradient is determined using the current and previous values of the bulk flow rate, U_b^n and U_b^{n-1} , and the target bulk flow rate, \bar{U}_b :

$$\left(\frac{\partial p}{\partial z}\right)^n = \alpha \left(\frac{1 + \bar{U}_b/U_b^n}{2}\right) \left(\frac{\partial p}{\partial z}\right)^{n-1}, \quad (84)$$

where

$$\alpha = 1 - \frac{D}{\bar{U}_b \Delta t} \left(\frac{U_b^n - U_b^{n-1}}{\bar{U}_b}\right). \quad (85)$$

In single domain simulations the determination of the current bulk flow rate is a straightforward task using fluid velocity and the domain volume

$$U_b^n = \frac{1}{V} \int_{\Omega} u^n dV. \quad (86)$$

However, in overlapping mesh simulations, a simple summation of the flow rates within each subdomain accounts for the value in the overlapping region twice,

$$U_b^n \neq \frac{1}{V^{[1]}} \int_{\Omega^1} u^{[1],n} dV + \frac{1}{V^{[2]}} \int_{\Omega^2} u^{[2],n} dV. \quad (87)$$

Thus, a modified determination of the global bulk flow rate must be devised to circumvent the repeated values. For that, the modified flow rates at individual subdomains are defined by

$$U_b^{[i],n} = \frac{1}{V^{[i]}} \int_{\Omega^i} \gamma u^{[i],n} dV, \quad (88)$$

where

$$\gamma = \begin{cases} 1 & \text{in } \Omega^{i,non}, i = 1, 2 \\ \frac{1}{2} & \text{in } \Omega^o \end{cases} \quad (89)$$

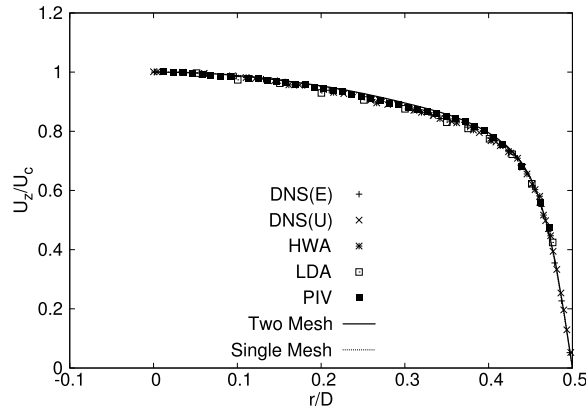


Fig. 23. Average velocity profile of turbulent pipe flow normalized by the centerline velocity: The symbols represent the data presented in [79] (DNS(E): direct numerical simulation performed by Eggels and Nieuwstadt, DNS(U): direct numerical simulation performed by Unger and Friedrich, HWA: hot-wire anemometry, LDA: laser Doppler anemometry, PIV: particle image velocimetry) while the lines represent the data found using the simulation discussed here. Lines for the inside and outside meshes are both displayed on the plot, though the same line style is used to improve readability. The present simulation data was found using 6th-order polynomials and second order temporal scheme with $\Delta t = 1 \times 10^{-3}$. IEXT2 was performed at interfaces with two iterations for multimesh simulations.

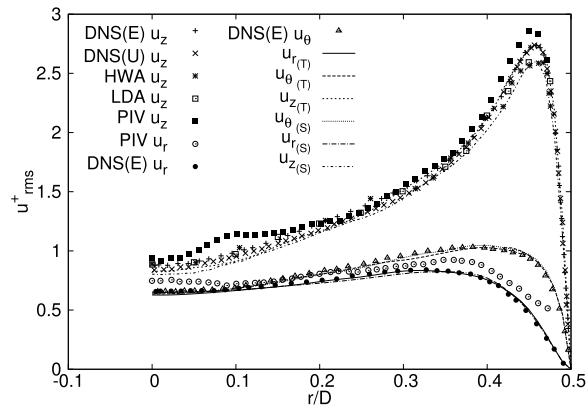


Fig. 24. u_{rms}^+ profile of turbulent pipe flow. The symbols represent the data presented in [79] while the lines represent the data obtained from the simulation discussed here. The T and S subscripts refer to the present two-mesh and the single mesh simulations, respectively. Refer to the caption of Fig. 23 for parameter specifications and details regarding abbreviations.

and domain superscripts correspond to those illustrated in Fig. 22. The sum of the modified flow rates gives the correct value for the global flow rate:

$$U_b^n = U_b^{[1],n} + U_b^{[2],n}. \tag{90}$$

In general, the overlap region does not coincide with the element boundaries, thus the weights γ for the integration in Eq. (88) are assigned on a GL pointwise basis. Note that if an element partially lies within the overlap region, a non-smooth function γ across an element would result in a reduced order of accuracy for the GL quadrature integration. While decreased accuracy in the calculation of U_b for a small portion of the domain will affect the value given for the global bulk flow rate, its effect will be small since the lower-order integration is localized only to affected elements along the interface.

The global bulk flow rate (90) of the decomposed domain is used to adjust the global pressure gradient (84), thereby maintaining a prescribed value of \bar{U}_b . Pressure gradient (84) is subsequently used to calculate the wall shear stress $\tau_w = -\frac{D}{4} \frac{\partial p}{\partial z}$ and friction velocity $u_\tau = \sqrt{\tau_w/\rho}$ for the normalization of turbulent statistics.

4.3.3. Accuracy

The present DNS simulations were performed using second order time-stepping with $\Delta t = 1 \times 10^{-3}$, while IEXT2 was employed with two iterations for interface values. Second order temporal schemes, rather than third as in other tests, were used in this case to achieve an increase in computational speed, to allow for turbulent statistics to be collected over longer time periods. Most of the DNS reported in the literature for the turbulent pipe flows were done with second order schemes as well [79–81]. The flow was allowed to develop until the non-dimensional time $t\bar{U}_b/D = 1350$ was reached, after which turbulent statistics was collected for another 500 time units. Figs. 23 and 24 show the results of the present simulation compared to the numerical and experimental results found in [79] for the mean normalized velocity and the root

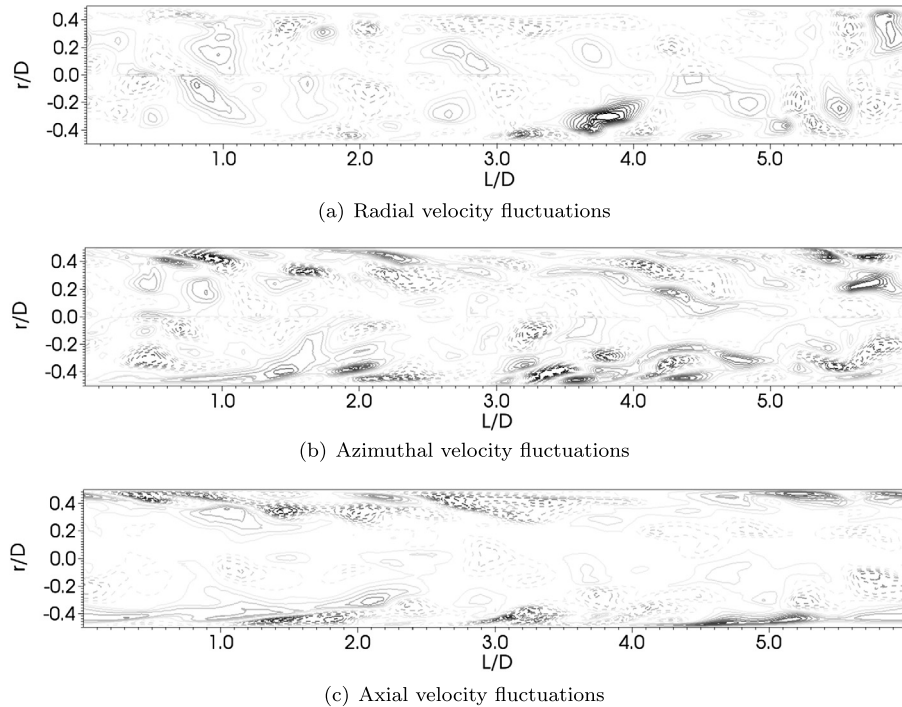


Fig. 25. Contours of fluid velocity fluctuations in a length-wise cross section of the pipe at non-dimensional time of 3000. Solid and dashed contours represent positive and negative velocity fluctuations, respectively. The darkest contour lines in the radial (a) and azimuthal (b) velocity fluctuation plots represent values of 0.2 and -0.2 for solid and dashed, with step size of 0.02 between contours. The darkest contour lines in axial velocity fluctuation plot (c) represent values of 0.45 and -0.45 for solid and dashed, with step size of 0.045 between contours. Both inner and outer domains are shown.

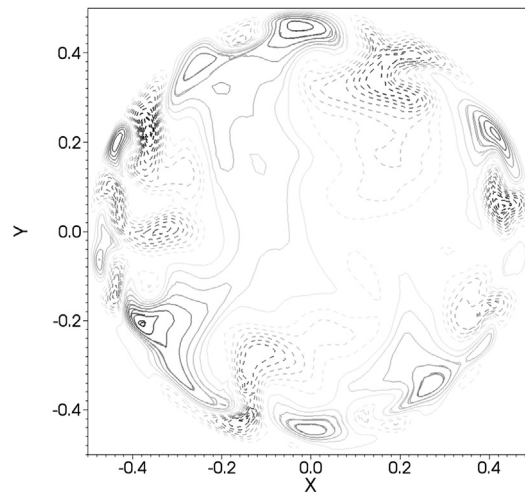


Fig. 26. Contours of axial fluid velocity fluctuations in a cylindrical cross section of the pipe. Both inner and outer domains are shown. Refer to the caption of Fig. 25 for details regarding axial velocity fluctuation contours.

mean squared velocity. The present results give turbulent statistics that correlate well with the published data from several simulations and experiments. The present simulation gives a centerline Reynold's Number $Re_c = \frac{U_c D}{\nu} = 6815$ compared to $Re_c = 6950$ in the DNS results [79], and Re_τ Reynolds number $Re_\tau = \frac{u_\tau D}{\nu} = 360.9$ compared to the value of 360 reported in [79].

The velocity fluctuation contours shown in Figs. 25 and 26 display good continuity between the two subdomains. The figures show that, in general, the velocity fluctuations near the pipe wall are larger than the fluctuations near the center. Evidence of the ejection of lower speed fluid (dashed contour lines) from the boundary layer and the sweeping motion of higher speed fluid (solid contour lines) toward the pipe wall is seen in the mushroom shaped contour lines displayed in Fig. 26, in accordance with the corresponding literature [79,80].

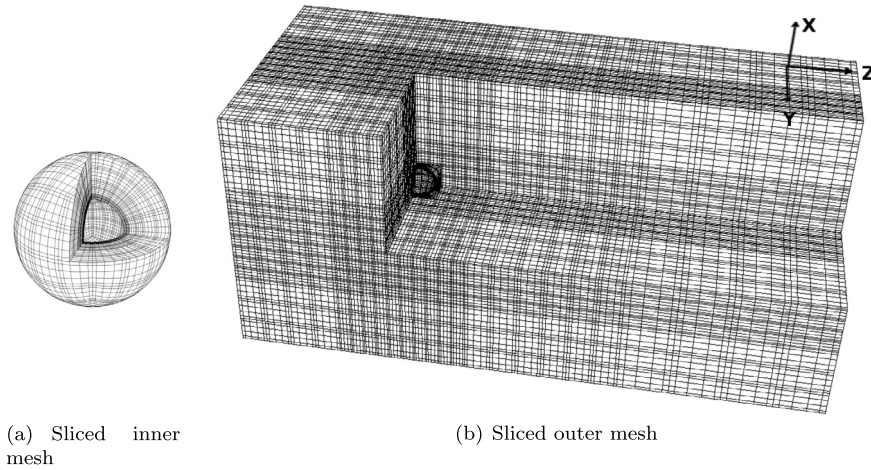


Fig. 27. Mesh geometry for simulation of steady flow around a solid sphere [not scaled relative to each other]. The subdomains displayed have sections removed for visualization of the solid sphere and vacancy for the inner and outer meshes respectively.

Table 3
Aerodynamic force coefficients and Strouhal Numbers for $Re = 250$ & 300 .

	Re	$\overline{C_D}$	$\overline{C_L}$	St
Present data	250	0.709	0.063	0.136
	300	0.665	0.070	
Johnson and Patel [82]	250		0.062	0.137
	300	0.656	0.069	
Constantinescu and Squires [83]	250	0.700	0.062	0.136
	300	0.665	0.065	
Kim and Choi [84]	250	0.702	0.060	0.134
	300	0.657	0.067	
Giacobello et al. [85]	250	0.702	0.061	0.134
	300	0.658	0.067	

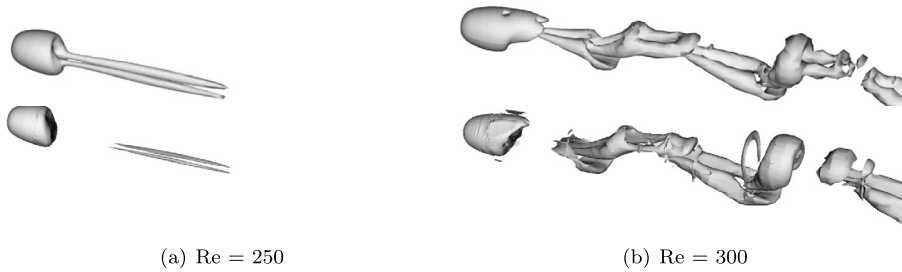


Fig. 28. Iso-surface of λ_2 -criterion at $\lambda_2 = -8 \cdot 10^{-4}$. The top illustration in each subfigure is presented as published in [85]. Snapshot (a) was taken at $t^* = \frac{tD}{U_\infty} = 200.0$, and snapshot (b) at $t^* = 152.5$.

4.4. Flow past a sphere

Here we describe the simulation of another 3D problem, namely, the flow around a stationary sphere. This problem further demonstrates the ability of our overlapping mesh methodology to handle more complex geometries with mixed rectangular and curved elements. A spherical interior mesh is constructed around a solid-walled sphere while the cuboid exterior mesh contains a vacancy wherein the sphere resides (Fig. 27). The interior mesh was created so that the boundary layer contains a minimum of nine grid points normal to the sphere wall for all simulations. Uniform inflow velocity $u = U_\infty$ is prescribed at one end of the exterior mesh and outflow boundary conditions $\nabla \mathbf{u} \cdot \mathbf{n} = 0$, are prescribed at the other. Symmetric boundary conditions ($\mathbf{u} \cdot \mathbf{n} = 0$) are enforced on the other domain boundaries.

Problems were simulated for diameter based Reynolds numbers of 250 and 300, corresponding to flow in the non-symmetric regime and unsteady vortex-shedding regime, respectively, for the flow around a stationary sphere as described by Johnson and Patel [82]. Simulations were performed with 5th-order polynomial approximations using second order temporal and extrapolation schemes with two iterations and $\Delta t = 5 \cdot 10^{-3}$. Table 3 shows the aerodynamic forces and Strouhal numbers ($St = fD/U_\infty$) determined using the present overlapping mesh methodology in comparison with other published

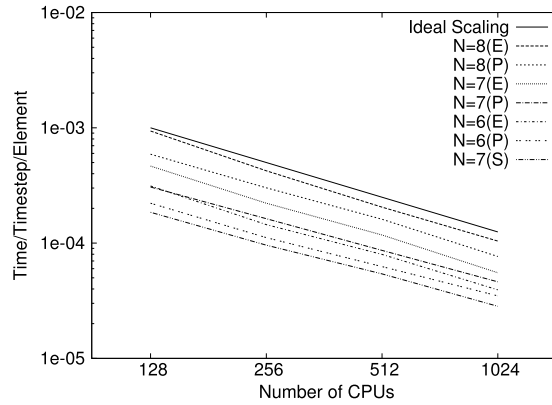


Fig. 29. Computational time vs number of CPUs for simulations performed on the SDSC Gordon cluster with different polynomial orders of approximation (N). Grids used for double mesh simulations contain a total of 51.2k elements (12.8k in inside mesh, 38.4k in outside mesh), giving $\sim 17.6\text{M}$, 26.2M , and 37.3M total gridpoints for 6th, 7th, and 8th order polynomial approximations respectively. The single mesh simulation consists of $\sim 41\text{k}$ elements giving $\sim 21.0\text{M}$ gridpoints for 7th order polynomial approximations. The 'E' denotes that the simulations were performed with an *equal* number of processors allocated to each mesh, while 'P' designates simulations where processors were assigned to each mesh *proportional* to the number of elements it contains. The single mesh simulation is signified by an 'S'.

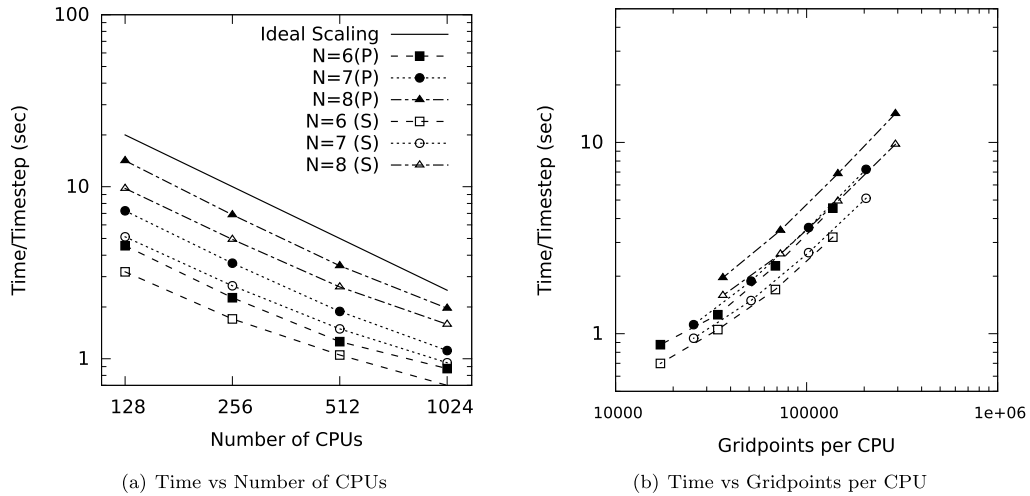


Fig. 30. Computational scaling data for simulations performed on the TACC Stampede cluster with different polynomial orders of approximation (N), using second order temporal schemes with two iterations. Grids used for double mesh simulations contain a total of 51.2k elements (12.8k in inside mesh, 38.4k in outside mesh), with single mesh simulations (S) also containing a total of 51.2k elements, giving $\sim 17.6\text{M}$, 26.2M , and 37.3M total gridpoints for 6th, 7th, and 8th order polynomial approximations respectively. Double mesh simulations used proportional (P) CPU allocation.

data. The results are in good comparison with other published data. Fig. 28 presents iso-surfaces of the λ_2 -criterion (described in [86]) at $\lambda_2 = -8 \cdot 10^{-4}$ for present data and the results published in [85]. A comparison of the two sets of data shows the same major flow structures. Considering all presented data, good correlation exists between data resulting from the present overlapping grid methodology and published data.

5. Scalability

In this section, we perform the scalability tests for the developed overlapping grid method. We utilize the previous test case of the turbulent pipe flow simulations as a platform for the scalability studies. However, the refined computational meshes were constructed, with higher element counts than in the DNS simulations described above (see label to Figs. 29 and 30) in order to enable testing on high processors counts (up to 1024). The refined outside mesh consisted of three times the number of elements and collocation points as the inside mesh. This discrepancy in the number of elements allowed us to test two methods for processor allocations: equal allocation (same number of processors for each domain, irrespective of the relative mesh sizes), and proportional allocation (processors allocated proportionally to the number of elements the mesh contains, namely, the outside domain was assigned three times as many processors as the inside domain). The results of the scalability tests, performed on the SDSC Gordon and TACC Stampede clusters, are shown in Figs. 29 and 30 with a refined single mesh pipe simulation for comparison. Fig. 29 shows the total CPU time for each simulation, divided by the

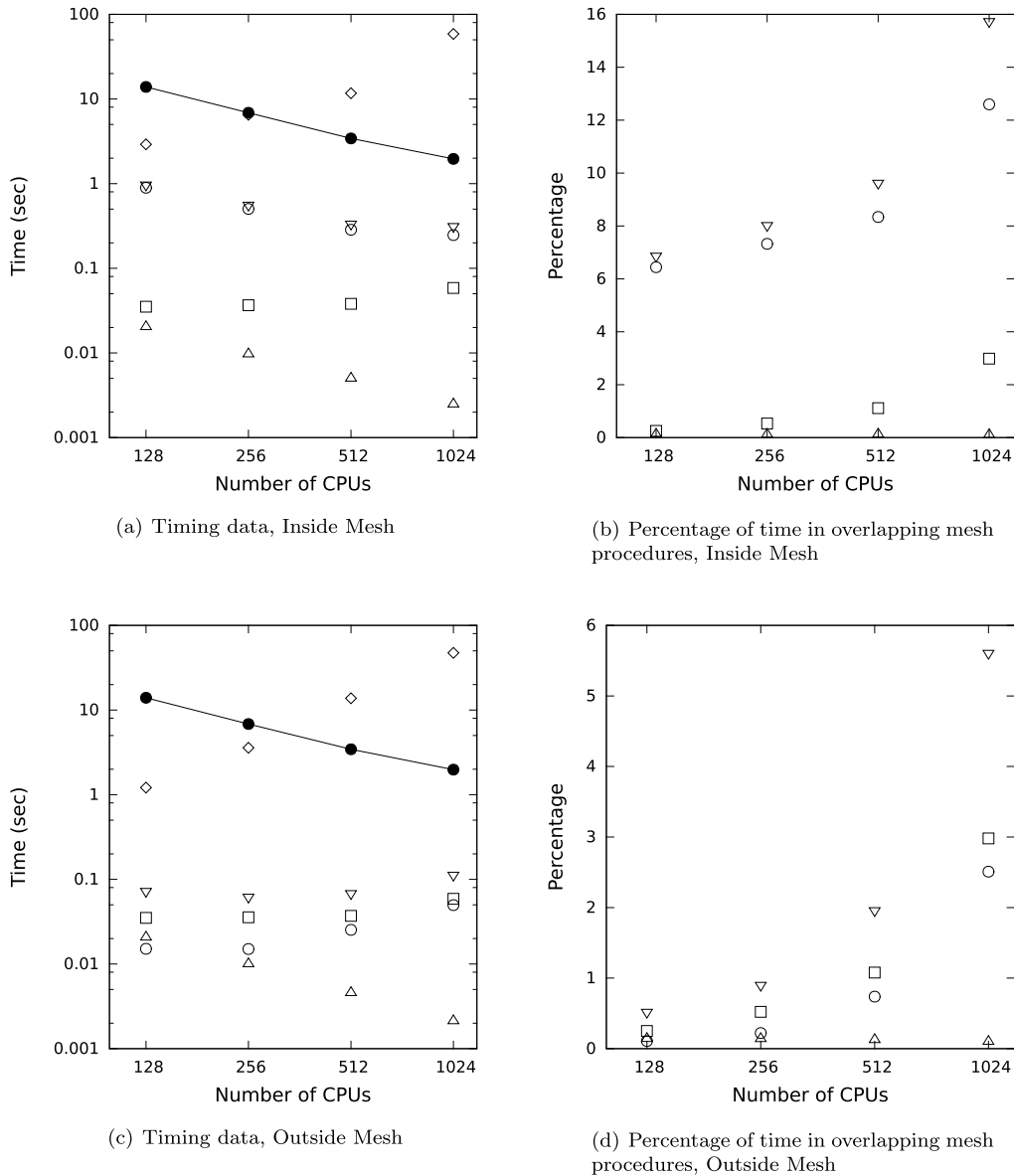


Fig. 31. Overlapping mesh timing data for simulations performed on the TACC Stampede cluster. All tests were performed with 8th order polynomials, using second order temporal schemes with two iterations using a proportional allocation of CPUs. The solid lines with filled circles denote the total computational time per timestep. Open diamonds represent the time spent in overlapping mesh initialization (performed only once, before the first timestep), open squares represent the time/timestep used for interpolation in the first iteration, open circles represent the time/timestep used for interpolation in the second iteration, open upright triangles represent the time/timestep for temporal extrapolation, and open inverted triangles represent the total overlapping mesh communication time spent per timestep. Plots (a) and (c) present the timing data, while plots (b) and (d) present the overlapping mesh communication time as a percentage of the total time per timestep.

number of timesteps and the total number of elements contained in the simulation, versus the total number of allocated CPUs, and Fig. 30 contains two plots, the second of which shows a comparison of timing data. Fig. 31 breaks down the time spent per timestep in different aspects of the overlapping mesh coupling procedures. It is seen that the scaling of our overlapping mesh methodology nearly reaches ideal scaling in each presented case. Line sections that appear to achieve better than ideal scaling are a result of inconsistent communication speeds among the nodes assigned to the jobs, for example, slower communication speeds among nodes assigned to a 128 core test compared with a 256 core test with faster communication speeds among the assigned nodes will appear to have better than ideal scaling. Several simulations were performed for each case, and averages of the timing data is presented. As expected, scaling for proportional CPU allocations yielded a speedup over equal allocations in each of the tested cases, due to more consistent load balancing. However, while the number of elements assigned per core is a good indicator of the work load, there is more to consider when seeking to attain optimal speeds. For instance, the external mesh contains 4800 element faces requiring interpolation from values in

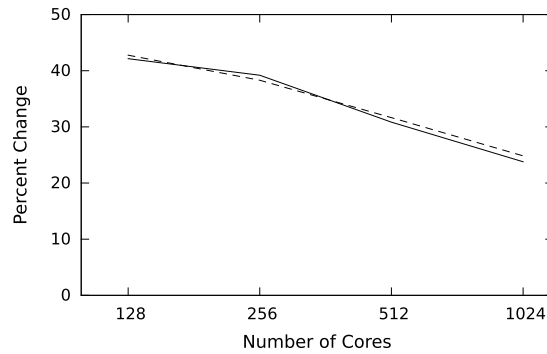


Fig. 32. Percent change in time per timestep between single mesh and overlapping mesh tests. The solid and dashed lines represent percent change in inside and outside meshes, respectively. For mesh characteristics see the caption to Fig. 30, and for runtime parameters refer to Fig. 31 caption.

the other subdomain, while the interior mesh contains only 1600 such faces. In the proportional scaling tests, the outside mesh is assigned three times the number of cores that are assigned to the inside mesh, though during the interpolation step the inside mesh must determine values for three times as many interface points. The fewer processors assigned to the inside mesh have a much larger work load during the overlapping mesh procedures, as is illustrated in the plots of Fig. 31 by the greater amount of time the inside mesh spends for overlapping mesh procedures. While a proportional allocation is a good starting point for a speed-up of the problem, further work on optimization of parallel procedures is required to ensure a more robust load balancing methodology. Fig. 32 illustrates the percent change in the time per timestep when going from single mesh to overlapping mesh simulations with the same total element count. We see that overlapping mesh tests, with proportional CPU allocation, spend 40% more time per timestep at low processor counts than single mesh tests due to additional interpolation and communication procedures, however the efficiency increases as the CPU count grows, with only 20% more time per timestep at 1024 CPUs, promising even higher benefits at larger processor counts.

6. Conclusions

We have developed an overlapping grid methodology for two and three dimensional applications in a Spectral Element Method incompressible flow solver. Spectral accuracy is maintained for spatial discretization due to a spectral interpolation at interface boundaries using solution approximations in an N th-order polynomial space on GL points. The global high-order (up to a third) temporal accuracy of the flow solver is also maintained with few, or no, iterations using solutions at previous timesteps to form explicit extrapolation approximation at interfaces.

Our overlapping grid methodology has been validated with several two- and three-dimensional simulations, comparing results to exact solutions and experimental data. Results from simulations of two dimensional convecting eddies were compared with exact solutions. As the polynomial order for solution approximations is increased, spectral convergence of the global solution on overlapping grids is observed. As the size of timestep is decreased, keeping the polynomial order constant, the temporal accuracy of the global scheme is maintained, as long as the interface extrapolation order is consistent with the global temporal accuracy. In addition, the order of interface extrapolation proved to be a limiting factor in the global temporal accuracy, thus, a simulation with third order global time-stepping scheme with first order interface extrapolation gives results with global first order temporal accuracy. We note that addition of subdomain iterations can improve the temporal accuracy, but at the expense of the computational cost.

Simulations of an incompressible Taylor vortex convecting through a long two-dimensional global domain helped form a more complete understanding of errors as flow features cross subdomain boundaries, and where disturbances are present downstream. It was observed that as the vortex crossed a subdomain boundary, overall errors remained small in the presence of outflow disturbances.

Our three dimensional fully-developed turbulent pipe flow simulation on two overlapping meshes showed good agreement with other physical and computational experiments. Turbulent statistics closely matched those published in [79], including Re_c , Re_τ , mean velocity profile, and u_{rms}^+ profile data. We have also performed scalability tests and demonstrated near linear computational scaling of the developed overlapping grid solver for large processor counts, up to 1024 processors.

Our future research will seek to extend the present methodology to moving overlapping grids in an effort to develop a spectrally accurate mechanism for modeling flows around rotating and moving objects. This extension will enable high-order accurate computations of problems that are very hard, or impossible, to model with conventional methods.

Acknowledgements

This work has supported by the NSF grants #1250124 and #1025359. Support by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357 and by the NSF RTG

grant at Northwestern University is also acknowledged. We thank the NSF XSEDE program for providing the allocations on the SDSC Gordon and TACC Stampede clusters where the scalability and timing tests were performed.

Appendix A

A.1. Pressure solution for Walsh's eddies

Convecting frame of reference Eddy solutions in a convecting frame of reference can be found using the technique proposed by Walsh [74] and discussed in section 4.1. For the present test case, we find these solutions from the initial conditions given by the stream function

$$\tilde{\psi}(\tilde{x}, \tilde{y}) = -\frac{1}{5} \cos[5\tilde{x}] + \frac{1}{4} \sin[3\tilde{x}] \sin[4\tilde{y}] - \frac{1}{5} \sin[5\tilde{y}], \quad (91)$$

leading to the following exact solution for \mathbf{u} in a convecting frame of reference:

$$\begin{aligned} \tilde{u}(\tilde{x}, \tilde{y}, t) &= e^{-25tv} (-\cos[5\tilde{y}] + \cos[4\tilde{y}] \sin[3\tilde{x}]), \\ \tilde{v}(\tilde{x}, \tilde{y}, t) &= e^{-25tv} (-\sin[5\tilde{x}] - \frac{3}{4} \cos[3\tilde{x}] \sin[4\tilde{y}]). \end{aligned} \quad (92)$$

As in section 4.1, by tilde we denote the variables in a convecting frame of reference.

The solution for pressure is found from the relation $\nabla \tilde{p} = -\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}$, which, in the present case, leads to two partial differential equations for pressure:

$$\frac{\partial \tilde{p}}{\partial \tilde{x}} = -\tilde{u} \frac{\partial \tilde{u}}{\partial \tilde{x}} - \tilde{v} \frac{\partial \tilde{u}}{\partial \tilde{y}} \quad (93)$$

$$\frac{\partial \tilde{p}}{\partial \tilde{y}} = -\tilde{u} \frac{\partial \tilde{v}}{\partial \tilde{x}} - \tilde{v} \frac{\partial \tilde{v}}{\partial \tilde{y}} \quad (94)$$

Solving this set of equations by integration gives the solution for the pressure of non-convecting eddies

$$\begin{aligned} \tilde{p}(\tilde{x}, \tilde{y}, t) &= \frac{1}{64} e^{-50tv} (-16 + 16 \cos[6\tilde{x}] + 8 \cos[8\tilde{x} - 4\tilde{y}] \\ &\quad - 32 \cos[2(\tilde{x} - 2\tilde{y})] + 9 \cos[8\tilde{y}] - 8 \cos[4(2\tilde{x} + \tilde{y})] \\ &\quad + 32 \cos[2(\tilde{x} + 2\tilde{y})] - 4 \sin[3(\tilde{x} - 3\tilde{y})] + 32 \sin[5(\tilde{x} - \tilde{y})] \\ &\quad + 36 \sin[3\tilde{x} - \tilde{y}] - 32 \sin[5(\tilde{x} + \tilde{y})] + 36 \sin[3\tilde{x} + \tilde{y}] \\ &\quad - 4 \sin[3(\tilde{x} + 3\tilde{y})]) \end{aligned} \quad (95)$$

Stationary frame of reference To convert the above solutions to the stationary frame of reference, we need to use the coordinate transformation

$$\begin{aligned} x &= \tilde{x} + u_0 t \\ y &= \tilde{y} + v_0 t, \end{aligned} \quad (96)$$

where (u_0, v_0) is the constant velocity of convection. That leads to the following solution for velocity

$$\begin{aligned} u(x, y, t) &= e^{-25tv} (-\cos[5(y - v_0 t)] + \cos[4(y - v_0 t)] \sin[3(x - u_0 t)]) + u_0, \\ v(x, y, t) &= e^{-25tv} (-\sin[5(x - u_0 t)] - \frac{3}{4} \cos[3(x - u_0 t)] \sin[4(y - v_0 t)]) + v_0, \end{aligned} \quad (97)$$

while solution for pressure is obtained analogously as

$$p(x, y, t) = \tilde{p}(x - u_0 t, y - v_0 t, t) \quad (98)$$

and not spelled out here due to a rather cumbersome final expression.

References

- [1] T.F. Chan, T.P. Mathew, Domain decomposition algorithms, *Acta Numer.* 3 (1994) 61–143.
- [2] B.F. Smith, P. Bjorstad, W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
- [3] P.F. Fischer, An overlapping Schwarz method for spectral element solution of the incompressible Navier–Stokes equations, *J. Comput. Phys.* 133 (1) (1997) 84–101.
- [4] W. Henshaw, D. Schwendeman, Parallel computation of three-dimensional flows using overlapping grids with adaptive mesh refinement, *J. Comput. Phys.* 227 (16) (2008) 7469–7502.

- [5] J.L. Steger, J.A. Benek, On the use of composite grid schemes in computational aerodynamics, *Comput. Methods Appl. Mech. Eng.* 64 (1) (1987) 301–320.
- [6] G. Cheshire, W.D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, *J. Comput. Phys.* 90 (1) (1990) 1–64.
- [7] C. Bernardi, Y. Maday, A. Patera, Domain decomposition by the mortar element method, in: *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters*, Springer, 1993, pp. 269–286.
- [8] R.L. Meakin, Composite overset structured grids, in: *Handbook of Grid Generation*, 1999, pp. 1–20.
- [9] C. Dawson, Q. Du, T. Dupont, A finite difference domain decomposition algorithm for numerical solution of the heat equation, *Math. Comput.* 57 (195) (1991) 63–71.
- [10] J.U. Schlüter, X. Wu, S. Kim, S. Shankaran, J.J. Alonso, H. Pitsch, A framework for coupling Reynolds-averaged with large-eddy simulations for gas turbine applications, *J. Fluids Eng.* 127 (4) (2005) 806–815.
- [11] S. Hahn, K. Duraisamy, G. Iaccarino, S. Nagarajan, J. Sitaraman, X. Wu, J.J. Alonso, J.D. Baeder, S.K. Lele, P. Moin, F. Schmitz, Coupled high-fidelity URANS simulation for helicopter applications, Center for Turbulence Research, Annual Research Briefs (2006).
- [12] Y. Peet, S. Lele, Computational framework for coupling compressible and low Mach number codes, *AIAA J.* 46 (8) (2008) 1990–2001.
- [13] H.A. Schwarz, Ueber einige Abbildungsaufgaben, *J. Reine Angew. Math.* 70 (1869) 105–120.
- [14] E.A. Volkov, The method of composite meshes for finite and infinite regions with piecewise smooth boundary, *Tr. Mat. Inst. Steklova* 96 (1968) 117–148.
- [15] P.-L. Lions, On the Schwarz alternating method. I, in: *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Paris, France, 1988, pp. 1–42.
- [16] Z.J. Wang, A fully conservative interface algorithm for overlapped grids, *J. Comput. Phys.* 122 (1) (1995) 96–106.
- [17] W. Henshaw, D. Schwendeman, An adaptive numerical scheme for high-speed reactive flow on overlapping grids, *J. Comput. Phys.* 191 (2) (2003) 420–447.
- [18] Y. Peet, P. Fischer, Heat transfer LES simulations in application to wire-wrapped fuel pins, in: *Proceedings 10th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, Chicago, IL, 2010 (AIAA 2010-4318).
- [19] P.-L. Lions, On the Schwarz alternating method. III: a variant for nonoverlapping subdomains, in: *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, vol. 6, SIAM, Philadelphia, PA, 1990, pp. 202–223.
- [20] M.H. Carpenter, J. Nordström, D. Gottlieb, A stable and conservative interface treatment of arbitrary spatial accuracy, *J. Comput. Phys.* 148 (2) (1999) 341–365.
- [21] Y. Maday, C. Mavriplis, A. Patera, Nonconforming Mortar Element Methods: Application to Spectral Discretizations, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, 1988.
- [22] M. Manna, A. Vacca, M.O. Deville, Preconditioned spectral multi-domain discretization of the incompressible Navier–Stokes equations, *J. Comput. Phys.* 201 (1) (2004) 204–223.
- [23] W. Henshaw, A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids, *J. Comput. Phys.* 113 (1) (1994) 13–25.
- [24] G. Desquesnes, M. Terracol, E. Manoha, P. Sagaut, On the use of a high order overlapping grid method for coupling in CFD/CAA, *J. Comput. Phys.* 220 (1) (2006) 355–382.
- [25] L.F. Pavarino, O.B. Widlund, A polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions, *SIAM J. Numer. Anal.* 33 (4) (1996) 1303–1335.
- [26] T.M. Burton, J.K. Eaton, Analysis of a fractional-step method on overset grids, *J. Comput. Phys.* 177 (2) (2002) 336–364.
- [27] S.E. Sherer, J.N. Scott, High-order compact finite-difference methods on general overset grids, *J. Comput. Phys.* 210 (2005) 459–496.
- [28] P.K. Moore, J.E. Flaherty, Adaptive local overlapping grid methods for parabolic systems in two space dimensions, *J. Comput. Phys.* 98 (1) (1992) 54–63.
- [29] J.C. Strikwerda, C.D. Scarbnick, A domain decomposition method for incompressible viscous flow, *SIAM J. Sci. Comput.* 14 (1) (1993) 49–67.
- [30] Y. Zang, R.L. Street, A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains, *Int. J. Numer. Methods Fluids* 20 (5) (1995) 341–361.
- [31] H.S. Tang, S.C. Jones, F. Sotiropoulos, An overset-grid method for 3D unsteady incompressible flows, *J. Comput. Phys.* 191 (2) (2003) 567–600.
- [32] P. Bjorstad, W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 2004.
- [33] A. Quarteroni, A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Number CMCS-BOOK-2009-019, Oxford University Press, 1999.
- [34] A. Toselli, O. Widlund, *Domain Decomposition Methods: Algorithms and Theory*, vol. 3, Springer, 2005.
- [35] G. Starius, Composite mesh difference methods for elliptic boundary value problems, *Numer. Math.* 28 (2) (1977) 243–258.
- [36] G. Starius, Constructing orthogonal curvilinear meshes by solving initial value problems, *Numer. Math.* 28 (1) (1977) 25–48.
- [37] G. Starius, On composite mesh difference methods for hyperbolic differential equations, *Numer. Math.* 35 (3) (1980) 241–255.
- [38] B. Kreiss, Construction of a curvilinear grid, *SIAM J. Sci. Stat. Comput.* 4 (2) (1983) 270–279.
- [39] J.L. Steger, F.C. Dougherty, J. Benek, A Chimera grid scheme, in: *Advances in Grid Generation*, ASME Fluids Engineering Conference, Houston, TX, 1983.
- [40] P.G. Buning, A 3-D Chimera grid embedding technique, in: *Proceedings 7th AIAA Computational Physics Conference*, Cincinnati, OH, 1985 (AIAA 1985-1523).
- [41] F.C. Dougherty, J.-H. Kuan, Transonic store separation using a three-dimensional Chimera grid scheme, in: *27th AIAA Aerospace Sciences Meeting*, Reno, NV, 1989 (AIAA 1989-637).
- [42] P.G. Buning, Navier–Stokes simulation of the Space Shuttle launch vehicle flight transonic flow field using a large scale Chimera grid system, in: *12th AIAA Applied Aerodynamics Conference*, Colorado Springs, CO, 1994 (AIAA 1994-1860).
- [43] K.A. Hennesius, T.H. Pulliam, A zonal approach to solution of the Euler equations, in: *3rd AIAA Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference*, St. Louis, MO, 1982 (AIAA 82-0969).
- [44] M.M. Rai, K.A. Hennesius, S.R. Chakravarthy, Metric-discontinuous zonal grid calculations using the Osher scheme, *Comput. Fluids* 12 (3) (1984) 161–175.
- [45] K. Hennesius, M. Rai, Applications of a conservative zonal scheme to transient and geometrically complex problems, *Comput. Fluids* 14 (1) (1986) 43–58.
- [46] M.M. Rai, A conservative treatment of zonal boundaries for Euler equation calculations, *J. Comput. Phys.* 62 (2) (1986) 472–503.
- [47] R. Chen, T. Hagstrom, p-Adaptive Hermite methods for initial value problems, *ESAIM: Math. Model. Numer. Anal.* 46 (03) (2012) 545–557.
- [48] D.A. Kopriva, A staggered-grid multidomain spectral method for the compressible Navier–Stokes equations, *J. Comput. Phys.* 143 (1) (1998) 125–158.
- [49] N.C. Prewitt, D.M. Belk, W. Shyy, Parallel computing of overset grids for aerodynamic problems with moving objects, *Prog. Aerosp. Sci.* 36 (2000) 117–172.
- [50] Z.J. Wang, V. Parthasarathy, N. Hariharan, A fully automated Chimera methodology for multiple moving body problems, *Int. J. Numer. Methods Fluids* 33 (7) (2000) 919–938.
- [51] G. Houzeaux, R. Codina, A Chimera method based on a Dirichlet/Neumann (Robin) coupling for the Navier–Stokes equations, *Comput. Methods Appl. Mech. Eng.* 192 (31) (2003) 3343–3377.
- [52] W.D. Henshaw, D.W. Schwendeman, Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow, *J. Comput. Phys.* 216 (2) (2006) 744–779.

- [53] W.D. Henshaw, A high-order accurate parallel solver for Maxwell's equations on overlapping grids, *SIAM J. Sci. Comput.* 28 (5) (2006) 1730–1765.
- [54] C.K.W. Tam, K.A. Kurbatskii, A wavenumber based extrapolation and interpolation method for use in conjunction with high-order finite difference schemes, *J. Comput. Phys.* 157 (2) (2000) 588–617.
- [55] A.K. Chaniotis, D. Poulidakos, High order interpolation and differentiation using B-splines, *J. Comput. Phys.* 197 (1) (2004) 253–274.
- [56] R.E. Gordnier, M.R. Visbal, Implicit LES computation of a vortical-gust/wing interaction for transitional flow, in: 53rd AIAA Aerospace Sciences Meeting, 2015 (AIAA 2015-1067).
- [57] B. Sjögreen, H.C. Yee, Variable high order multiblock overlapping grid methods for mixed steady and unsteady multiscale viscous flows, *Commun. Comput. Phys.* 5 (2009) 730–744.
- [58] M.J. Brazell, J. Sitaraman, D.J. Mavriplis, An overset mesh approach for 3D mixed element high order discretizations, in: 53rd AIAA Aerospace Sciences Meeting, 2015 (AIAA 2015-1739).
- [59] A.T. Patera, A spectral element method for fluid dynamics: laminar flow in a channel expansion, *J. Comput. Phys.* 54 (1984) 468–488.
- [60] M.O. Deville, P.F. Fischer, E.H. Mund, *High-Order Methods for Incompressible Fluid Flow*, vol. 9, Cambridge University Press, 2002.
- [61] T. Holst, S.D. Thomas, U. Kaynak, K.L. Gundy, J. Flores, N.M. Chaderjian, *Computational Aspects of Zonal Algorithms for Solving the Compressible Navier–Stokes Equations in Three Dimensions*, NASA Ames Research Center, Moffett Field, CA, United States, 1985 (19860003828).
- [62] Ü. Kaynak, T. Holst, B. Cantwell, Computation of transonic separated wing flows using an Euler/Navier–Stokes zonal approach, *NASA STI/Recon Technical Report N. 88:11089*, 1986.
- [63] M. Hinatsu, J.H. Ferziger, Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique, *Int. J. Numer. Methods Fluids* 13 (8) (1991) 971–997.
- [64] H.S. Tang, T. Zhou, On nonconservative algorithms for grid interfaces, *SIAM J. Numer. Anal.* 37 (1) (1999) 173–193.
- [65] X.C. Cai, Additive Schwarz algorithms for parabolic convection diffusion equations, *Numer. Math.* 60 (1991) 41–61.
- [66] H. Rui, Multiplicative Schwarz methods for parabolic problems, *Appl. Math. Comput.* 136 (2) (2003) 593–610.
- [67] W.D. Henshaw, On multigrid for overlapping grids, *SIAM J. Sci. Comput.* 26 (5) (2005) 1547–1572.
- [68] Y. Peet, P.F. Fischer, Stability analysis of interface temporal discretization in grid overlapping methods, *SIAM J. Numer. Anal.* 50 (6) (2012) 3375–3401.
- [69] J.W. Lottes, P.F. Fischer, Hybrid multigrid/Schwarz algorithms for the spectral element method, *J. Sci. Comput.* 24 (1) (2005) 45–78.
- [70] Y. Maday, A.T. Patera, E.M. Rønquist, An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow, *J. Sci. Comput.* 5 (4) (1990) 263–292.
- [71] J. Van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Stat. Comput.* 7 (3) (1986) 870–891.
- [72] G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, D. Walker, *Solving Problems on Concurrent Computers*, vol. 19, Prentice-Hall, Englewood Cliffs, New Jersey, 1988, p. 88.
- [73] P. Fischer, J. Lottes, D. Pointer, A. Siegel, Petascale algorithms for reactor hydrodynamics, *J. Phys. Conf. Ser.* 125 (2008) 012076.
- [74] O. Walsh, Eddy solutions of the Navier–Stokes equations, in: *The Navier–Stokes Equations II – Theory and Numerical Methods*, Springer, 1992, pp. 306–309.
- [75] C. Schneidesch, M. Deville, Chebyshev collocation method and multi-domain decomposition for Navier–Stokes equations in complex curved geometries, *J. Comput. Phys.* 106 (1993) 234–257.
- [76] Y. Peet, P.F. Fischer, Legendre spectral element method with nearly incompressible materials, *Eur. J. Mech. A, Solids* 44 (2013) 91–103.
- [77] T. Colonius, S.K. Lele, P. Moin, The free compressible viscous vortex, *J. Fluid Mech.* 230 (1991) 45–73.
- [78] G.I. Taylor, On the dissipation of eddies, *The Scientific Papers of Sir Geoffrey Ingram Taylor* 2 (1918) 96–101.
- [79] J.G.M. Eggels, F. Unger, M.H. Weiss, J. Westerweel, R.J. Adrian, R. Friedrich, F.T.M. Nieuwstadt, Fully developed turbulent pipe flow: a comparison between direct numerical simulation and experiment, *J. Fluid Mech.* 268 (1994) 175–210.
- [80] X. Wu, P. Moin, A direct numerical simulation study on the mean velocity characteristics in turbulent pipe flow, *J. Fluid Mech.* 608 (2008) 81–112.
- [81] J.M.J. Den Toonder, M.A. Hulsen, G.D.C. Kuiken, F.T.M. Nieuwstadt, Drag reduction by polymer additives in a turbulent pipe flow: numerical and laboratory experiments, *J. Fluid Mech.* 337 (1997) 193–231.
- [82] T.A. Johnson, V.C. Patel, Flow past a sphere up to a Reynolds number of 300, *J. Fluid Mech.* 378 (1999) 19–70.
- [83] G.S. Constantinescu, K.D. Squires, LES and DES investigations of turbulent flow over a sphere, in: *Proceedings 38th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2000 (AIAA 2000-0540).
- [84] D. Kim, H. Choi, Laminar flow past a sphere rotating in the streamwise direction, *J. Fluid Mech.* 461 (2002) 365–386.
- [85] M. Giacobello, A. Ooi, S. Balachandar, Wake structure of a transversely rotating sphere at moderate Reynolds numbers, *J. Fluid Mech.* 621 (2009) 103–130.
- [86] J. Jeong, F. Hussain, On the identification of a vortex, *J. Fluid Mech.* 285 (1995) 69–94.