Journal of Computational Physics 390 (2019) 121–151

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Moving overlapping grid methodology of spectral accuracy for incompressible flow solutions around rigid bodies in motion

Brandon E. Merrill[a], Yulia T. Peet[b,*]

[a] Raytheon Missile Systems, Modeling, Simulation, & Analysis, Tucson, AZ 85756, USA
[b] School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85287, USA

ARTICLE INFO

Article history:
Received 25 May 2018
Received in revised form 18 December 2018
Accepted 20 January 2019
Available online 7 March 2019

Keywords:
Moving grids
Domain decomposition
Spectral elements
Navier-Stokes equations

ABSTRACT

The simulation of fluid flow around moving rigid bodies has proven to be a difficult task for traditional computational fluid dynamics solvers. Decomposing the global domain into overlapping subdomains and allowing each subdomain to move independently, permits solutions to many flow problems with complex moving geometries to be determined in a straightforward manner. The present development of the moving overlapping grid method is built within a spectral element method incompressible flow solver, and uses an Arbitrary Lagrangian-Eulerian formulation of the governing equations to prescribe subdomain motions. The method maintains global spectral spatial accuracy across the subdomains with the polynomial refinement. The global high-order temporal accuracy of the method is also maintained through subdomain coupling enforced by an explicit interface temporal extrapolation scheme. The method produces aerodynamic forces and vortex shedding around two- and three-dimensional moving rigid bodies that is in line with published experimental and computational data. Additionally, the method achieves near linear computational scaling to thousands of cores.

1. Introduction

Within the engineering and physical communities, there are many important problems that involve fluid flow around moving bodies, including propellers and blades on rotor- and watercraft, stirred reactors, maneuvering aircraft, and biological flows such as blood flow through a pumping heart. Many of these problems include complex physics, often involving turbulence interactions which require high-accuracy computational methods for reliable flow field predictions. However, accurate modeling of the fluid flow around complex moving geometries has traditionally been a challenging and often impossible task.

Found among published literature are several methods used for computations of the flow around moving bodies. These methods can be categorized into two general classes: global-mesh methods and zonal-mesh methods (also referred to as embedded grid, Chimera, or domain decomposition methods). In global-mesh methods, a single grid is used to model the fluid flow around moving bodies, using either fixed-mesh or body-conforming mesh schemes.

Fixed-mesh methods, such as immersed boundary and fictitious domain, perform calculations on a global (non-decomposed) Eulerian mesh that does not conform to the fluid-solid interface. (Note that while many of these methods are commonly applied to fluid-fluid interface problems, we will focus on their applications to fluid-solid interface problems.)

* Corresponding author.
E-mail addresses: brandon.e.merrill@raytheon.com (B.E. Merrill), ypeet@asu.edu (Y.T. Peet).

https://doi.org/10.1016/j.jcp.2019.01.048

The immersed boundary method, which was originally introduced by Peskin in 1972 to model the blood flow through a pumping heart [1,2], allows for flow solutions to be computed around irregular shaped objects by tracking Lagrangian points at the interface boundary against the Eulerian (fixed) mesh where the fluid flow is calculated. The mesh within an immersed boundary solver does not need to conform to the fluid-solid interface, and the no-slip boundary conditions at fluid-solid interfaces are enforced by adding a body forcing term to the governing equations using either continuous or discrete forcing [3,4]. In the similar fictitious domain method, first introduced by Glowinski et al. [5], the boundary conditions on the surface of rigid bodies are enforced by including distributed Lagrange multipliers in the governing equations. While several advances have been made to improve efficiency of both immersed boundary and fictitious domain methods [4], the accuracy of such methods is lacking, with few fixed-mesh methods even reaching second-order spatial accuracy [6–9]. In addition, resolution near interface boundaries is inconsistent as the object moves, requiring dynamic remeshing techniques such as adaptive mesh refinement (AMR) to maintain sufficient resolution, especially when attempting to capture characteristics of boundary layer flows over moving bodies [10].

In contrast, body-conforming methods ensure more consistent resolution near moving boundaries, though mesh generation is usually a more complex task. As solid interfaces move, the mesh must also move and deform thus inhibiting large displacements which would cause detrimental mesh distortions. Body-conforming methods include Deforming-Spatial-Domain/Stabilized Space-Time (DSD/SST) and Arbitrary Lagrangian-Eulerian (ALE) methods. In the DSD/SST method, as originally presented by Tezduyar et al. [11], the governing equations for incompressible fluid flow, the Navier-Stokes equations, are defined using the space-time formulation, where the global time interval of the simulation is divided into subintervals called space-time slabs, and discretization is performed using interpolation functions in the four-dimensional space-time domain. The shape and orientation of the spatial domain is given by the shape of the individual space-time slabs, where solution on each slab is solved using a finite element formulation, thus treating moving boundaries throughout the global time interval [12]. The ALE method combines the Lagrangian and Eulerian formulations of the incompressible Navier-Stokes equations by changing the material derivative to account for additional convection introduced by the velocity of the moving grid points. Original developments into the combination of Eulerian and Lagrangian formulations began in the sixties [13,14], though the standardized ALE method was developed a decade later by a group at Los Alamos National Lab [15,16] where it was applied to finite difference methods. This ALE formulation was extended to finite element methods in the seventies and eighties [17–20], with later extension to the spectral element method by Ho and Patera [21,22]. In both of these techniques large rigid body movements cause large mesh distortions and possibly entanglement of the computational grid. To remedy this problem, many methods remesh the global domain when distortions become large, though remeshing is a computationally expensive process [4]. Another possible remedy is to allow the global mesh to move with the body in a rigid-body type of motion, but this creates challenges for the enforcement of boundary conditions at moving global boundaries.

Decomposing the global domain into a collection of subdomains, as in the class of zonal mesh methods, would allow for large rigid body movements of one domain within another, avoiding the need for remeshing and replacing global boundary conditions on moving domains with local interface conditions. Among this class, for example, are sliding mesh methods, which decompose the domain into *non*-overlapping subdomains, thus greatly restricting the types of grid motion that can be performed. While sliding mesh solvers can effectively perform simulations involving rotating geometries, they cannot be applied to important problems involving translational or deforming motion. Several variations of these solvers are commonly used for simulations of stirred reactors [23–27] and propellers/blades on rotorcraft [28–30]. Sliding mesh methods are typically coupled with finite volume methods, although other solvers are also used, such as finite element methods [31,27]. Global spatial accuracy for sliding mesh methods, which is most often limited by the accuracy of the interpolation of values among subdomains, reaches up to 2nd-order [29,30].

On the other hand, overlapping mesh methods allow for arbitrary rigid body motions, and do not have the same movement restrictions as sliding mesh solvers since constraints on mesh alignment are alleviated. While overlapping mesh methods are commonly used for rotating machinery simulations as well [32–35], they have additionally been applied to simulate flows involving other types of body motions. These include flows around moving cylinders [36,37], moving spheres [38], piston-driven flows [36,39], biological flows such as flows through moving valves and hearts [40,41], aerodynamic flows such as three element airfoils [42], plunging wings [37], and store separation from a wing [43,44,42,45], as well as flows around moving ships [46].

Moving overlapping mesh methods can be viewed as an extension of the stationary overlapping mesh methodologies which have been used for some time now to simplify mesh generation [47–49], model multiphysics problems [50,51], and improve parallelization of algorithms [52–54]. The very early idea of using overlapping grids to assist with a numerical solution of partial differential equations (PDEs) can be traced to a development of a domain decomposition technique, a methodology used to solve a boundary value problem (BVP) arising from a numerical discretization of a given PDE. A mathematical concept of overlapping domain decomposition technique was initially introduced for analytical solutions to differential equations, and was proposed by Schwarz in the late 1800's [55]. The Schwarz Alternating Method, upon which many modern numerical domain decomposition techniques are based, decomposes a global domain into smaller overlapping subdomains, which communicate values at subdomain interface boundaries (local boundaries that are not also part of the global boundary, $\Gamma^{ij} = \partial\Omega^i \setminus \partial\Omega$). Variants exist including additive Schwarz method, multiplicative Schwarz method and others [56,53]. Although originally appeared as a technique for solving elliptic problems, the methodology has been extended to other classes of partial differential equations, including parabolic [57,58] and hyperbolic systems [59,60].

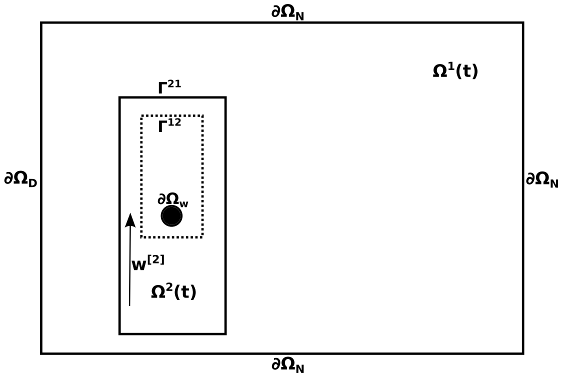


Fig. 1. Schematic of the global domain and the subdomains. The dashed line denotes a hole cut in the subdomain $\Omega^1(t)$, which is covered by $\Omega^2(t)$.

When applied to time-dependent PDEs, a global discretization of the problem in space and time is first achieved, resulting in a discrete BVP, which is then solved with domain decomposition schemes. In this setting, the corresponding solution is essentially bound to be implicit, with many iterations of the value passing and solution calculations performed each timestep until convergence to a specified tolerance is achieved [60–62]. To relax the conditions of tight coupling associated with the domain decomposition techniques, a new class of overlapping grid methods have started to emerge, where the PDEs are locally discretized on the subdomains and are coupled together through the appropriate interface conditions, such as Composite Mesh methods [48] and Chimera Grid techniques [63–65]. This, in principle, provides a greater flexibility for formulating the numerical schemes, including a possibility of different subdomain discretizations, different time step values, and explicit interface updates.

While there are multiple numerical schemes that are designed to achieve high-order accuracy in single-grid computations, it is challenging to preserve high orders of accuracy on overlapping domains since inter-grid interpolation must also be of high-order. Among stationary overlapping grid methods, a few methods have been developed that feature fourth- to sixth- order accuracy in space, and third- to fourth- order accuracy in time [66–70]. High-order developments for moving overlapping grid methods are however scarce due to the added difficulties related to the mesh movement, typically restricting the currently available methodologies to be globally of second order, both spatially and temporally [36]. While some methods use higher-order (fourth or fifth) spatial integration schemes within individual subdomains [40,71], linear interface interpolation is usually employed, which restricts their *global* spatial accuracy to second.

Our recently developed stationary overlapping mesh methodology [72] built within a spectral element method incompressible flow solver [73] maintains a global spectral accuracy in space and up to a third order accuracy in time, requiring only a few iterations. The present moving overlapping mesh methodology builds upon this development, allowing for a subdomain movement with the ALE formulation of the governing equations. Herein we discuss the development of this moving overlapping mesh methodology that achieves global (across the subdomains) spectral accuracy in space and up to a third order accuracy in time. Although the underlying ALE and multidomain coupling formulations are developed in a general framework, so that any arbitrary prescribed body motion can be handled (including deforming movements, such as, for example, in biolocomotion), this paper concentrates on prescribed solid motion as a rigid body, thus undermining the need for mesh deformation. We will, however, describe the methodology in a general sense, which would allow one to reproduce it, if needed, for any arbitrary motions. Mathematical and computational formulation of the methodology, verification of spatial and temporal global accuracy, and validation on canonical two- and three-dimensional problems involving flow over rigid moving bodies, will be given.

The remainder of the paper will be structured as follows. In Section 2 the methodology for solving fluid flow problems on moving overlapping subdomains with a spectral element method is put forth, including the mathematical derivation and numerical discretization of governing equations for fluid flow on subdomains using the ALE framework. In Section 3 we discuss the global communication framework for the parallelization of the method, and in Section 4 results are presented for two- and three-dimensional validation simulations, while in Section 5 conclusions are drawn.

2. Methodology

2.1. Mathematical formulation

2.1.1. Governing equations

In the current methodology, a two or three dimensional global solution domain is decomposed into two overlapping subdomains, $\Omega^g(t) = \Omega^1(t) \cup \Omega^2(t)$. The subdomains may be time dependent and are allowed to move with velocity $\mathbf{w}^{[i]}(\mathbf{x}, t)$, an example of which is seen in Fig. 1. For the sake of generality, we formulate the methodology as if both domains are moving, assuming that a stationary domain has $\mathbf{w}^{[i]}(\mathbf{x}, t) = 0$. In this paper, we consider a rigid body motion for the subdomains, although the Cartesian velocity $\mathbf{w}(\mathbf{x}, t)$ can still be a function of $\mathbf{x}$, for example in the case of rotation. The fluid motion in each subdomain is governed by the incompressible Navier-Stokes equations which are represented in non-dimensional form in a space R^d, as shown below

$$\Omega^1(t) \quad \begin{cases} \dfrac{D\,\mathbf{u}^{[1]}}{D\,t} = -\nabla\, p^{[1]} + \dfrac{1}{\mathrm{Re}}\nabla^2\mathbf{u}^{[1]} \\ \nabla\cdot\mathbf{u}^{[1]} = 0 \end{cases} \tag{1}$$

and

$$\Omega^2(t) \quad \begin{cases} \dfrac{D\,\mathbf{u}^{[2]}}{D\,t} = -\nabla\, p^{[2]} + \dfrac{1}{\mathrm{Re}}\nabla^2\mathbf{u}^{[2]} \\ \nabla\cdot\mathbf{u}^{[2]} = 0 \end{cases} \tag{2}$$

where $\mathbf{u}$ is the velocity vector, p is the pressure, D/Dt is the material derivative, and the Reynolds number, $\mathrm{Re} = U\,L/\nu$, is based on a characteristic velocity, length scale and kinematic viscosity. The bracketed superscript signifies the corresponding subdomain.

The global domain boundary is defined $\partial\Omega^g(t) = \partial(\Omega^1(t) \cup \Omega^2(t))$. The global boundary $\partial\Omega^g(t)$ typically contains Dirichlet, $\partial\Omega^g_D(t)$, and the Neumann, $\partial\Omega^g_N(t)$, parts, as in example of Fig. 1. Local subdomain boundaries that are not also part of the global boundary are termed interface boundaries, $\Gamma^{ij}(t) \equiv \{\partial\Omega^i(t) \setminus \partial\Omega^g(t)\} \subset \Omega^j(t)$. Conditions at moving solid wall boundaries are defined within a local subdomain, $\partial\Omega^i_W(t)$. The generalized initial and boundary conditions for each subdomain are then given as

$$\mathbf{u}^{[1],[2]}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}), \qquad \mathbf{x}\in\Omega^{1,2}(0) \tag{3a}$$

$$\mathbf{u}^{[1],[2]}(\mathbf{x},t) = \mathbf{u}_d(\mathbf{x},t), \qquad \mathbf{x}\in\partial\Omega^g_D(t) \tag{3b}$$

$$\nabla\mathbf{u}^{[1],[2]}(\mathbf{x},t)\cdot\widehat{\mathbf{n}} = 0, \qquad \mathbf{x}\in\partial\Omega^g_N(t) \tag{3c}$$

$$\mathbf{u}^{[1],[2]}(\mathbf{x},t) = \mathbf{w}^{[1],[2]}(\mathbf{x},t), \qquad \mathbf{x}\in\partial\Omega^{1,2}_W(t) \tag{3d}$$

$$\mathbf{u}^{[1]}(\mathbf{x},t) = \mathbf{u}^{[2]}(\mathbf{x},t) \qquad \mathbf{x}\in\Gamma^{12}(t)\|\Gamma^{21}(t) \tag{3e}$$

with $\widehat{\mathbf{n}}$ the unit outward pointing surface normal.

2.1.2. *Arbitrary Lagrangian-Eulerian formulation*

To represent the flow solution on moving subdomains, we adopt an Arbitrary Lagrangian-Eulerian (ALE) formulation [20,74,75], which decouples the mesh motion from the fluid motion and formulates the equations of fluid motion in a coordinate system moving with the computational mesh. Even though our subdomains are moving as rigid bodies, and the mesh deformation is not an issue in the current work, a strictly Eulerian approach would involve re-interpolation of an entire flow solution onto a new location of grid points in physical space at each time step, which can be avoided with the ALE formulation. In what follows, we present a brief description of the methodology [20,74,75].

The position of a material point, or particle, within a continuous media is expressed in terms of *material coordinates*, $\mathbf{X} = (X_1, X_2, X_3)$, that are related to the initial position of a fluid particle within the media. On the other hand, within the laboratory frame *spatial coordinates* are fixed in time $\mathbf{x} = (x_1, x_2, x_3)$, and the relationship between material and spatial coordinates can be expressed in terms of a mapping that returns the spatial coordinates of a particle at a specified time when given its material coordinates. The inverse of this mapping returns the material coordinates of a particle, given its spatial coordinates. The ALE formulation introduces a third coordinate system, also dependent on space and time, which we will call the *reference coordinate* system, $\boldsymbol{\chi}$. This coordinate system is defined for a reference domain which is representative of the computational mesh and, generally, is allowed to move in an arbitrary manner. A mapping can be defined that returns the reference coordinates of a particle when given its spatial coordinates

$$\chi_i = \mathcal{G}\,(x_i, t)\,, \tag{4}$$

with inverse mappings also defined.

Derivatives with respect to time of a physical quantity $f(x_i, t)$ related to the flow can be expressed in terms of any of the coordinate systems defined above. Thus, the following definitions,

$$\left.\frac{\partial f}{\partial t}\right|_{\mathbf{X}}, \quad \left.\frac{\partial f}{\partial t}\right|_{\mathbf{x}}, \quad \left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{\chi}}, \tag{5}$$

represent, respectively, the derivative with respect to time in the material frame (material, or Lagrangian, derivative, often denoted with a symbol D/Dt), the derivative with respect to time in the laboratory frame (Eulerian derivative), and the derivative with respect to time in a reference frame (ALE derivative, often denoted with a symbol $\delta/\delta t$).

To allow for large fluid distortions, fluid flow is typically expressed using the Eulerian formulation, using coordinates $\mathbf{x}$. Thus, convective terms arise in equations (1) and (2) to account for the motion of the particles relative to the laboratory frame [75]:

$$\frac{Df}{Dt} = \left.\frac{\partial f}{\partial t}\right|_{\mathbf{x}} + \mathbf{u} \cdot \nabla f, \tag{6}$$

where $\mathbf{u} = (\partial \mathbf{x}/\partial t)|_{\mathbf{X}}$ is the particles material velocity, or fluid velocity.

In a similar manner the mesh velocity, $\mathbf{w} = (\partial \mathbf{x}/\partial t)|_{\boldsymbol{\chi}}$, is defined using the reference domain $\boldsymbol{\chi}$, and an ALE derivative is formulated to account for convective terms that arise due to the relative motion between the reference and laboratory coordinate systems

$$\frac{\delta f}{\delta t} = \left.\frac{\partial f}{\partial t}\right|_{\mathbf{x}} + \mathbf{w} \cdot \nabla f. \tag{7}$$

The material derivative can now be formulated for use with the reference domain by expressing (6) in terms of (7) [74]

$$\frac{Df}{Dt} = \frac{\delta f}{\delta t} + (\mathbf{u} - \mathbf{w}) \cdot \nabla f. \tag{8}$$

We see that if $\mathbf{w} = 0$, the traditional formulation of the material derivative is recovered as used in the Eulerian description, and if $\mathbf{w} = \mathbf{u}$ the convective terms vanish leaving only a partial derivative holding $\boldsymbol{\chi}$ constant as used in the Lagrangian description.

With the above formulation, Navier-Stokes equations (1) and (2) become

$$\Omega^1(t) \begin{cases} \dfrac{\delta \mathbf{u}^{[1]}}{\delta t} + (\mathbf{u}^{[1]} - \mathbf{w}^{[1]}) \cdot \nabla \mathbf{u}^{[1]} = -\nabla p^{[1]} + \dfrac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}^{[1]} \\ \nabla \cdot \mathbf{u}^{[1]} = 0 \end{cases} \tag{9}$$

and

$$\Omega^2(t) \begin{cases} \dfrac{\delta \mathbf{u}^{[2]}}{\delta t} + (\mathbf{u}^{[2]} - \mathbf{w}^{[2]}) \cdot \nabla \mathbf{u}^{[2]} = -\nabla p^{[2]} + \dfrac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}^{[2]} \\ \nabla \cdot \mathbf{u}^{[2]} = 0 \end{cases} \tag{10}$$

where initial and boundary conditions as given in (3) remain unchanged. We see that the governing equations are now formulated with respect to the moving mesh coordinate system, which allows for convenient numerical integration of equations of motion in the moving domains.

2.1.3. Mesh velocity

In the current methodology, the body motion is prescribed with an a-priori specified function $\mathbf{w}(\mathbf{x}, t)$. This allows us to specify boundary conditions for the fluid velocities at the moving wall boundaries in a manner given by equation (3d), and these values are also equal to the mesh velocity at the moving wall boundaries. In a general formulation, an elastic solver can be employed to find a smooth mesh velocity function inside a moving subdomain [74]. The present paper in particular deals with translational and rotational rigid body motions, which allows us to avoid an elastic solution and simply prescribe mesh velocities throughout the whole domain by extending the function $\mathbf{w}(\mathbf{x}, t)$ onto the corresponding grid points.

2.2. Numerical formulation

Equations (9) and (10) in moving overlapping subdomains are coupled through the interface conditions given in (3e). In the numerical solution of these equations, we determine the values at interfaces $\mathbf{u}^{[i]}_{interf}(\mathbf{x}, t)$ through an explicit approach utilizing an interpolated solution from the adjacent subdomain at several previous time instances, $\mathbf{u}^{[i]}_{interf}(\mathbf{x}, t) = \sum_k w_k \mathbf{u}^{[j]}_{interp}(\mathbf{x}, t - \tau_k)$, where τ_k correspond to the time shifts with respect to the previous time instances and w_k are the weights in the linear combination determined by an interpolation scheme used, to be described in Section 2.2.8. This value is used as a Dirichlet boundary condition, $\mathbf{u}^{[i]}(\mathbf{x}, t) = \mathbf{u}^{[i]}_{interf}(\mathbf{x}, t)$, for the subdomain Ω^i on the boundary Γ^{ij} at time t. Thus, solutions of (9) and (10) can be computed independently after interface values are exchanged. Hence, the numerical formulation within each subdomain is identical. In the following sections, we omit the subdomain superscripts ($\Omega^i \to \Omega$, $\mathbf{u}^{[i]} \to \mathbf{u}$, $\mathbf{w}^{[i]} \to \mathbf{w}$, $p^{[i]} \to p$), except where additional clarity is needed.

2.2.1. Global variational form

The ALE formulation of the governing equations, (9) and (10), is cast into variational form by multiplying each equation with a test function $\mathbf{v}(\mathbf{x},t) \in \mathcal{H}_0^1(\Omega(t))$, and then integrating over the subdomain. Note that both the test function, $\mathbf{v}(\mathbf{x},t)$ and the subdomain $\Omega(t)$ are dependent on time due to the moving reference frame. The variational form of the problem becomes: Find $\mathbf{u}(\mathbf{x},t) \in \mathcal{H}_b^1(\Omega(t))$, $p(\mathbf{x},t) \in \mathcal{L}^2(\Omega(t))$ such that

$$\begin{aligned}
&\left(\frac{\delta \mathbf{u}}{\delta t}, \mathbf{v}\right) + ((\mathbf{u}-\mathbf{w}) \cdot \nabla \mathbf{u}, \mathbf{v}) \\
&\quad + (\nabla p, \mathbf{v}) - \frac{1}{\mathrm{Re}}\left(\nabla^2 \mathbf{u}, \mathbf{v}\right) = 0 \quad \forall \mathbf{v}(\mathbf{x},t) \in \mathcal{H}_0^1(\Omega(t)), \\
&\qquad\qquad -(q, \nabla \cdot \mathbf{u}) = 0 \quad \forall q(\mathbf{x},t) \in \mathcal{L}^2(\Omega(t)),
\end{aligned} \tag{11}$$

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

$$(a,b) = \int\limits_{\Omega(t)} a(\mathbf{x},t)\, b(\mathbf{x},t)\, \mathrm{dV}, \quad \forall a, b \in \mathcal{L}^2(\Omega(t)), \tag{12}$$

and where $\mathcal{L}^2(\Omega(t))$ represents the space of square-integrable functions, and $\mathcal{H}^1(\Omega(t))$ represents the space of square-integrable functions whose first derivatives are square-integrable as well. $\mathcal{H}_0^1(\Omega(t))$ is the subspace of $\mathcal{H}^1(\Omega(t))$ with $\mathbf{v}(\mathbf{x},t) = 0$ on Dirichlet, $\partial\Omega_D(t)$, moving wall, $\partial\Omega_W(t)$, and interface, $\Gamma^{ij}(t)$, boundaries, while $\mathcal{H}_b^1(\Omega(t))$ is the subspace of $\mathcal{H}^1(\Omega(t))$ with $\mathbf{u}(\mathbf{x},t) = \mathbf{u}_d(\mathbf{x},t)$ on $\partial\Omega_D(t)$, $\mathbf{u}(\mathbf{x},t) = \mathbf{w}(\mathbf{x},t)$ on $\partial\Omega_W(t)$, or $\mathbf{u}^{[i]}(\mathbf{x},t) = \mathbf{u}^{[i]}_{interf}(\mathbf{x},t)$ on $\Gamma^{ij}(t)$.

The variational equation (11) can be transformed as follows. Let us first remark that, although test functions, $\mathbf{v}(\mathbf{x},t)$ are time dependent, they are defined in the ALE formulation to satisfy the property $\delta\mathbf{v}/\delta t = 0$ [74]. Using this property, the integration by parts, and the Reynolds transport theorem [76,74], the inner product of the unsteady term becomes

$$\left(\frac{\delta \mathbf{u}}{\delta t}, \mathbf{v}\right) = \frac{\delta}{\delta t} \int\limits_{\Omega(t)} \mathbf{u} \cdot \mathbf{v}\, \mathrm{dV} - \int\limits_{\Omega(t)} (\nabla \cdot \mathbf{w})\, \mathbf{u} \cdot \mathbf{v}\, \mathrm{dV}. \tag{13}$$

This formulation helps to avoid inconsistencies during the temporal discretization on non-stationary domains. Combining (13) with the convective terms, utilizing the incompressibility constraint, $\nabla \cdot \mathbf{u} = 0$, and employing an additional transformation of a viscous term via an integration by parts [74,72], the variational form of the Navier-Stokes equations in the ALE formulation becomes

$$\begin{aligned}
&\frac{\delta}{\delta t}(\mathbf{u}, \mathbf{v}) + (\nabla \cdot [\mathbf{uu} - \mathbf{uw}], \mathbf{v}) \\
&\quad -(p, \nabla \cdot \mathbf{v}) + \frac{1}{\mathrm{Re}}(\nabla \mathbf{u}, \nabla \mathbf{v}) = 0 \quad \forall \mathbf{v}(\mathbf{x},t) \in \mathcal{H}_0^1(\Omega(t)) \\
&\qquad\qquad -(q, \nabla \cdot \mathbf{u}) = 0 \quad \forall q(\mathbf{x},t) \in \mathcal{L}^2(\Omega(t)).
\end{aligned} \tag{14}$$

2.2.2. Spatial discretization

In the present implementation of the spectral element method, a moving subdomain, $\Omega(t)$, is decomposed into a union of conforming elements

$$\Omega(t) = \sum_{k=1}^{E} \Omega_k(t), \tag{15}$$

where aggregate inner products in a subdomain are defined through a summation across elements

$$(a,b)|_{\Omega(t)} = \sum_{k=1}^{E} (a_k, b_k)|_{\Omega_k(t)}\,. \tag{16}$$

The governing equations (14) are spatially discretized within each element by defining finite dimensional subspaces, $X^N \subset \mathcal{H}^1(\Omega(t))$ and $Y^N \subset \mathcal{L}^2(\Omega(t))$, onto which the search spaces, $\mathcal{H}^1(\Omega(t))$ and $\mathcal{L}^2(\Omega(t))$, are projected,

$$\begin{aligned}
X^N &= \mathcal{H}^1(\Omega(t)) \cap \mathbb{P}_N^d \\
Y^N &= \mathcal{L}^2(\Omega(t)) \cap \mathbb{P}_{N-2}^d,
\end{aligned} \tag{17}$$

with $\mathbb{P}_N^d$ the space comprised of a d-dimensional tensor-product of all Nth or lower-order polynomials.

A function, independent of time, exists for each element Ω_k that maps it from the ALE frame with reference coordinates $\boldsymbol{\chi} \in \Omega_k$, to a d-dimensional primary element, $\hat{\Omega} = [-1, +1]^d$, with primary coordinates $\mathbf{r} \in \hat{\Omega}$: $\mathbf{r} = \widetilde{\mathcal{M}}_k(\boldsymbol{\chi})$ with the inverse $\boldsymbol{\chi} = \widetilde{\mathcal{M}}_k^{-1}(\mathbf{r})$. To relate the primary coordinates to the physical coordinates in laboratory frame, however, the mapping function must be time-dependent, such as

$$\mathbf{r} = \mathcal{M}_k(\mathbf{x}, t) \equiv \widetilde{\mathcal{M}}_k\big(\mathcal{G}(\mathbf{x}, t)\big), \tag{18}$$

with the inverse

$$\mathbf{x} = \mathcal{M}_k^{-1}(\mathbf{r}, t) \equiv \mathcal{G}^{-1}\big(\widetilde{\mathcal{M}}_k^{-1}(\mathbf{r}), t\big), \tag{19}$$

with the function $\mathcal{G}(\mathbf{x}, t)$ defined as in (4).

Collocation points within the primary element are defined with Gauss-Lobatto Legendre (GL) quadrature in the velocity space, $\xi_j \in [-1, +1]$, and Gauss Legendre (G) quadrature in the pressure space, $\eta_j \in]-1, +1[$. Thus problem (14) is posed with finite-dimensional subspaces: Find $\mathbf{u}(\mathbf{r}, t) \in X_b^N$, $p(\mathbf{r}, t) \in Y^N$ such that

$$\frac{\delta}{\delta t}(\mathbf{u}, \mathbf{v})_{GL} + (\nabla \cdot [\mathbf{uu} - \mathbf{uw}], \mathbf{v})_{GL}$$

$$+\frac{1}{\mathrm{Re}}(\nabla \mathbf{u}, \nabla \mathbf{v})_{GL} - (p, \nabla \cdot \mathbf{v})_G = 0 \qquad \forall \mathbf{v}(\mathbf{r}, t) \in X_0^N \tag{20}$$

$$-(q, \nabla \cdot \mathbf{u})_G = 0 \qquad \forall q(\mathbf{r}, t) \in Y^N \tag{21}$$

where the aggregate inner products (16) on G and GL nodes are denoted.

Lagrange interpolating polynomials, $\phi_i(r)$, are employed as basis functions to span the discretized velocity space $\mathbb{P}_N$, and pressure space $\mathbb{P}_{N-2}$, with the Lagrange polynomials satisfying the following property

$$\phi_i(\xi_j) = \delta_{ij}, \tag{22}$$

where δ_{ij} is the Kronecker delta, giving the following finite-dimensional approximation for a scalar field $f_k(x, t)$ in $\Omega_k(t)$ in one dimension

$$f_k\left(\mathcal{M}_k^{-1}(r, t)\right)|_{\Omega_k(t)} \approx \sum_{i=0}^{N} f_{k,i}(t)\phi_i(r), \quad r \in [-1, +1], \tag{23}$$

with extension to multiple dimensions performed through the tensor product operations [74].

2.2.3. Discrete matrix operators

In this section, we present the matrix operators for the inner products in (20) and (21). We will start with the non-linear convective term, which has a different form in the present ALE methodology as compared to a stationary grid formulation [72]. For this term, noting that $(\nabla \cdot [\mathbf{uu} - \mathbf{uw}], \mathbf{v}) = ([(\mathbf{u} \cdot \nabla)\,\mathbf{u} - (\mathbf{w} \cdot \nabla)\,\mathbf{u} - (\nabla \cdot \mathbf{w})\,\mathbf{u}], \mathbf{v})$, we write, first, for a one-dimensional variable (unbolded) on a reference interval $r = [-1, +1]$

$$\Big(u(r,t)u'(r,t), v(r,t)\Big)_{k,GL} = \int_{\hat{\Omega}} u_k(r,t)u_k'(r,t)v_k(r,t)\mathrm{d}r$$

$$\approx \int_{\hat{\Omega}} \left(\sum_{i=0}^{N} u_{k,i}(t)\phi_i(r)\right)\left(\sum_{j=0}^{N} u_{k,j}(t)\phi_j'(r)\right)\left(\sum_{m=0}^{N} v_{k,m}(t)\phi_m(r)\right)\mathrm{d}r$$

$$\approx \sum_{m=0}^{N} v_{k,m}(t)\sum_{j=0}^{N} u_{k,j}(t)\sum_{i=0}^{N} u_{k,i}(t)\int_{\hat{\Omega}} \phi_i(r)\phi_j'(r)\phi_m(r)\mathrm{d}r \tag{24}$$

and similarly for the other two terms. The prime superscript here denotes the derivative in r. Combining all the three terms gives

$$\big(\big[uu' - wu' - w'u\big], v\big)_{k,GL} \approx \underline{v}_k^T(t)\hat{C}^k(t)\underline{u}_k(t), \tag{25}$$

with the time dependent convective operator, $\hat{C}^k(t)$, computed using GL quadrature weights as

$$\hat{C}_{ij}^k(t) = \big(u_{k,i}(t) - w_{k,i}(t)\big)\omega_i\hat{D}_{ij} - \sum_{m=0}^{N} w_{k,m}(t)\omega_i\hat{D}_{im}\delta_{ij}, \tag{26}$$

where $\hat{D}_{ij}$ is the spectral differentiation matrix

$$\hat{D}_{ij} \equiv \phi_j'(r)\Big|_{r=\xi_i}. \tag{27}$$

The property (22) of the Lagrange interpolating polynomials was utilized in deriving Eq. (26). Here, a hat over an unbolded variable denotes a one dimensional matrix operator for the reference interval, and an underline denotes a vector of coefficients which are defined at the discrete collocation points, $\underline{v} = (v_0, v_1, \dots, v_N)$. Note that the convective operator $\hat{C}^k(t)$ is nonlinear, time dependent, and local to each element k due to its dependence on velocity.

The discretization of the other terms in (20) and (21) is briefly summarized as follows (please see Refs. [72,74,77] for more details). The first and the third terms on the left side of equation (20) are discretized as

$$\big(u(r,t), v(r,t)\big)_{k,GL} \approx \underline{v}_k^T(t)\hat{B}\underline{u}_k(t), \tag{28}$$

$$\big(u'(r,t), v'(r,t)\big)_{k,GL} \approx \underline{v}_k^T(t)\hat{A}\underline{u}_k(t), \tag{29}$$

giving rise to the mass and stiffness matrices, respectively

$$\hat{\mathrm{B}}_{ij} = \int\limits_{\hat{\Omega}} \phi_i(\mathrm{r})\phi_j(\mathrm{r})\mathrm{dr} \approx \sum_{l=0}^{N} \omega_l \phi_i(\xi_l)\phi_j(\xi_l), \tag{30}$$

$$\hat{A}_{ij} = \int\limits_{\hat{\Omega}} \phi_i'(r)\,\phi_j'(r)\,\mathrm{d}r = \sum_{l=0}^{N} \omega_l \phi_i'(\xi_l)\,\phi_j'(\xi_l)\,. \tag{31}$$

Note that due to the property (22) for the Lagrangian basis functions, the mass matrix simplifies to a diagonal matrix with GL quadrature weights in the main diagonal, $\hat{B} = \mathrm{diag}(\omega_l)$.

The terms involving pressure in equation (20) and the left-hand side of the continuity equation (21) use G quadrature in the weak form

$$\big(p(r,t), v'(r,t)\big)_{k,G} \approx \underline{v}_k^T(t)\hat{\tilde{D}}^T\underline{p}_k(t), \tag{32}$$

$$\big(q(r,t), u'(r,t)\big)_{k,G} \approx \underline{q}_k^T(t)\hat{\tilde{D}}\underline{u}_k(t), \tag{33}$$

with a derivative matrix, $\hat{\tilde{D}}$, composed of a weighted spectral differentiation matrix interpolated onto the G nodes, and a weighted interpolation operator from GL to G nodes to form a consistent quadrature [74,72]. Note that, unlike the convective operator, all other (linear) operators in SEM are not element-dependent and not time-dependent.

The discrete inner products presented above can be extended to multiple dimensions with the use of the tensor product operations as, for example,

$$\big(u(\mathbf{r},t), v(\mathbf{r},t)\big)_{k,GL} = \underline{v}_k^T(t)\underline{\hat{B}}\,\underline{u}_k(t), \tag{34}$$

on a reference element $\mathbf{r} = [-1,+1]^d$, where the operator with the underline refers to its multi-dimensional counterpart defined through the tensor products, e.g. $\underline{\hat{B}} = \hat{B} \otimes \hat{B} \otimes \hat{B}$. Here, $\underline{u}_k(t)$ and $\underline{v}_k(t)$ denote coefficients of scalar variables in higher dimensions, $u(\mathbf{x},t)$, $v(\mathbf{x},t)$, stored in a vector form in a lexicographical order [74,72]. Additional modifications are made for geometries not conforming with the primary element definition, $[-1,+1]^d$, where the Jacobians, J_k, of the mapping functions (18), (19) are used to formulate the discrete inner products on deformed elements, such as

$$\left(u\left(\mathbf{x},t\right), v\left(\mathbf{x},t\right)\right)_{k,GL} = \underline{v}_k^T(t)\underline{B}_k(t)\underline{u}_k(t), \tag{35}$$

where $\underline{B}_k(t)$ is now defined, for example, in two dimensions,

$$\begin{aligned} \underline{B}_{k,i\hat{\imath}j\hat{\jmath}}(t) &= \int\limits_{\hat{\Omega}} \phi_i\phi_{\hat{\imath}}\phi_j\phi_{\hat{\jmath}}\,J_k(r_1,r_2,t)\,\mathrm{d}\mathrm{r} \\ &\approx \sum_{l=0}^{N}\sum_{m=0}^{N} \omega_l\omega_m\phi_i(\xi_l)\phi_{\hat{\imath}}(\xi_l)\phi_j(\xi_m)\phi_{\hat{\jmath}}(\xi_m)\,J_k(\xi_l,\xi_m,t). \end{aligned} \tag{36}$$

The hat is dropped for the full geometry operators that are no longer conforming to the primary element definition. Since Jacobian is time-dependent for moving grids, all discrete matrix operators are both element- and time-dependent.

The operators expressed so far were defined for local (albeit deformed) elements. Coupling across elements is accomplished through the use of connectivity operators. An aggregation of elemental inner products is performed to generate *unassembled* subdomain-wide operators (with the subscript “L” denoting unassembled arrays)

$$(u(\mathbf{x},t),v(\mathbf{x},t))_{GL}=\sum_{k=1}^{E}\underline{v}_k^T(t)\underline{B}_k(t)\underline{u}_k(t)=\underline{v}_L^T(t)\underline{B}_L(t)\underline{u}_L(t). \tag{37}$$

To enforce the velocity continuity across element boundaries, which is otherwise not enforced in unassembled arrays, the *gather* (Q^T) and *scatter* (Q) connectivity operators are introduced to map the unassembled arrays onto assembled ones and vice versa

$$\underline{u}(t)=\underline{Q}^T\underline{u}_L(t),\ \ \underline{u}_L(t)=\underline{Q}\,\underline{u}(t), \tag{38}$$

with sudomain-wide assembled arrays not carrying the subscript L. Values at element boundaries in assembled arrays are shared between adjacent elements, enforcing continuity [74]. Global inner products can be expressed via assembled arrays utilizing unassembled matrix operators as

$$(u(\mathbf{x},t),v(\mathbf{x},t))_{GL}=\underline{v}_L^T(t)\underline{B}_L(t)\underline{u}_L(t)=\underline{v}^T(t)\underline{Q}^T\underline{B}_L(t)\underline{Q}\,\underline{u}(t). \tag{39}$$

Although in a current moving domain formulation, an element moves and/or deforms, its relative position with respect to the other elements in the subdomain remains the same, thus the connectivity operators are independent of time. That would not be the case if the elements were to be added or removed from the subdomains during the run time, which is not implemented in the current procedure.

The multidimensional vector representation of the problem is given by the summation of individual scalar arrays

$$\big(\mathbf{u}(\mathbf{x},t),\mathbf{v}(\mathbf{x},t)\big)=\sum_{i=1}^{d}\big(u_i(\mathbf{x},t),v_i(\mathbf{x},t)\big) \tag{40}$$

yielding

$$\big(\mathbf{u}(\mathbf{x},t),\mathbf{v}(\mathbf{x},t)\big)_{GL}=\underline{\mathbf{v}}^T(t)\underline{\mathbf{Q}}^T\underline{\mathbf{B}}_L(t)\underline{\mathbf{Q}}\underline{\mathbf{u}}(t) \tag{41}$$

where bolded variables represent vector-valued coefficient arrays and operators that act on such arrays [74,72].

2.2.4. Spatially discretized equations

Upon spatial discretization, the weak form of the Navier-Stokes equations (1) and (2) given by Eq. (14) becomes

$$\frac{\delta}{\delta t}\big(\underline{\mathbf{B}}(t)\,\underline{\mathbf{u}}(t)\big)=-\underline{\mathbf{C}}(t)\,\underline{\mathbf{u}}(t)-\frac{1}{\mathrm{Re}}\underline{\mathbf{A}}(t)\,\underline{\mathbf{u}}(t)+\underline{\widetilde{\mathbf{D}}}^T(t)\,\underline{p}(t) \tag{42}$$

$$-\underline{\widetilde{\mathbf{D}}}(t)\,\underline{\mathbf{u}}(t)=0 \tag{43}$$

where the time-dependent subdomain-wide matrix operators are given by

$$\begin{aligned}
\underline{\mathbf{A}}(t)&=\underline{\mathbf{Q}}^T\underline{\mathbf{A}}_L(t)\underline{\mathbf{Q}},\\
\underline{\mathbf{B}}(t)&=\underline{\mathbf{Q}}^T\underline{\mathbf{B}}_L(t)\underline{\mathbf{Q}},\\
\underline{\mathbf{C}}(t)&=\underline{\mathbf{Q}}^T\underline{\mathbf{C}}_L(t)\underline{\mathbf{Q}},\\
\underline{\widetilde{\mathbf{D}}}(t)&=\underline{\widetilde{\mathbf{D}}}_L(t)\underline{\mathbf{Q}}.
\end{aligned} \tag{44}$$

2.2.5. Temporal discretization

In the current implementation of the methodology, the spatially discretized governing equations are temporally discretized using kth-order backward differentiation (BDFk) for the time derivative, which is based on truncated Taylor series [74]. The method gives, assuming constant time stepping,

$$\frac{\delta}{\delta t}\big[\underline{\mathbf{B}}(t)\,\underline{\mathbf{u}}(t)\big]_{BDF}^{n}=\frac{1}{\Delta t}\sum_{p=0}^{k}\beta_{pk}\underline{\mathbf{B}}^{n-p}\underline{\mathbf{u}}^{n-p}, \tag{45}$$

where $\frac{\delta}{\delta t}\big[\underline{\mathbf{B}}(t)\,\underline{\mathbf{u}}(t)\big]_{BDF}$ denotes the BDF derivative operator, superscript n denotes the current timestep and the coefficients β_{pk} are given in Table 1. To avoid an implicit treatment of a non-linear convective term, it is approximated by kth-order explicit extrapolation operator (EXTk) as

$$\mathcal{E}\big[\underline{\mathbf{C}}(t)\,\underline{\mathbf{u}}(t)\big]_{EXT}^{n}=\sum_{q=1}^{k}\gamma_{qk}\underline{\mathbf{C}}^{n-q}\,\underline{\mathbf{u}}^{n-q}, \tag{46}$$

with the coefficients γ_{qk} also given in Table 1.

Table 1
Coefficients for the BDFk and EXTk schemes, k=1,2,3 [78].

	β_{p1}	β_{p2}	β_{p3}	γ_{p1}	γ_{p2}	γ_{p3}
p=0	1	3/2	11/6			
p=1	−1	−2	−3	1	2	3
p=2		1/2	3/2		−1	−3
p=3			−1/3			1

The BDFk scheme with an EXTk treatment of the convective term and an implicit treatment of the viscous and pressure terms applied to the spatially discretized momentum equation (42) thus takes the form

$$\frac{1}{\Delta t}\sum_{p=0}^{k}\beta_{pk}\underline{\mathbf{B}}^{n-p}\underline{\mathbf{u}}^{n-p} = -\sum_{q=1}^{k}\gamma_{qk}\underline{\mathbf{C}}^{n-q}\,\underline{\mathbf{u}}^{n-q} - \frac{1}{\mathrm{Re}}\underline{\mathbf{A}}^{n}\underline{\mathbf{u}}^{n} + \underline{\widetilde{\mathbf{D}}}^{T,n}\underline{p}^{n}. \tag{47}$$

Note that the BDFk discretization is applied to the ALE derivative in the current formulation, i.e. the derivative defined with respect to the moving reference frame. To show that this discretization maintains kth order temporal accuracy, we first note that the values $\underline{\mathbf{B}}^{n-p}\underline{\mathbf{u}}^{n-p}$ are defined at spatial locations with respect to the mesh configuration at a timestep t^{n-p}, i.e. at locations $\mathbf{x}-\Delta\mathbf{x}_p$, where $\mathbf{x}$ are the physical coordinates associated with the current mesh position, and $\Delta\mathbf{x}_p$ is the mesh displacement during p previous time steps,

$$\underline{\mathbf{B}}^{n-p}\underline{\mathbf{u}}^{n-p} = \mathcal{B}(\mathbf{x}-\Delta\mathbf{x}_p, t-p\Delta t), \tag{48}$$

where $\mathcal{B}(\mathbf{x},t)$ would correspond to $\underline{\mathbf{B}}^{n}\underline{\mathbf{u}}^{n}$. The mesh position increment $\Delta\mathbf{x}_p$ can be approximated using the current mesh velocity $\mathbf{w}(\mathbf{x},t)$ as

$$\Delta\mathbf{x}_p = \mathbf{w}(p\,\Delta t) - \frac{1}{2!}\mathbf{w}_t(p\,\Delta t)^2 + \frac{1}{3!}\mathbf{w}_{tt}(p\,\Delta t)^3 + \ldots \tag{49}$$

Performing a multivariable Taylor series expansion of the approximation (48) around $\mathcal{B}(\mathbf{x},t)$ and substituting into the ALE derivative (45), one can easily get

$$\frac{\delta}{\delta t}\left[\underline{\mathbf{B}}(t)\,\underline{\mathbf{u}}(t)\right]_{BDF} = \frac{\partial\mathcal{B}}{\partial t} + \mathbf{w}\frac{\partial\mathcal{B}}{\partial\mathbf{x}} + \mathcal{O}[\Delta t^k], \tag{50}$$

which is, indeed, a kth order temporal approximation to the ALE derivative defined in (7).

Using the same arguments as before, i.e. defining

$$\underline{\mathbf{C}}^{n-p}\,\underline{\mathbf{u}}^{n-p} = \mathcal{C}(\mathbf{x}-\Delta\mathbf{x}_p, t-p\,\Delta t),$$

substituting expression (49) for $\Delta\mathbf{x}_p$, and expanding each term $\mathcal{C}(\mathbf{x}-\Delta\mathbf{x}_p, t-p\,\Delta t)$ in Taylor series around $\mathcal{C}(\mathbf{x},t)$ with Δt as a small parameter, one can illustrate that the EXTk scheme maintains the desired temporal order of accuracy on the moving meshes,

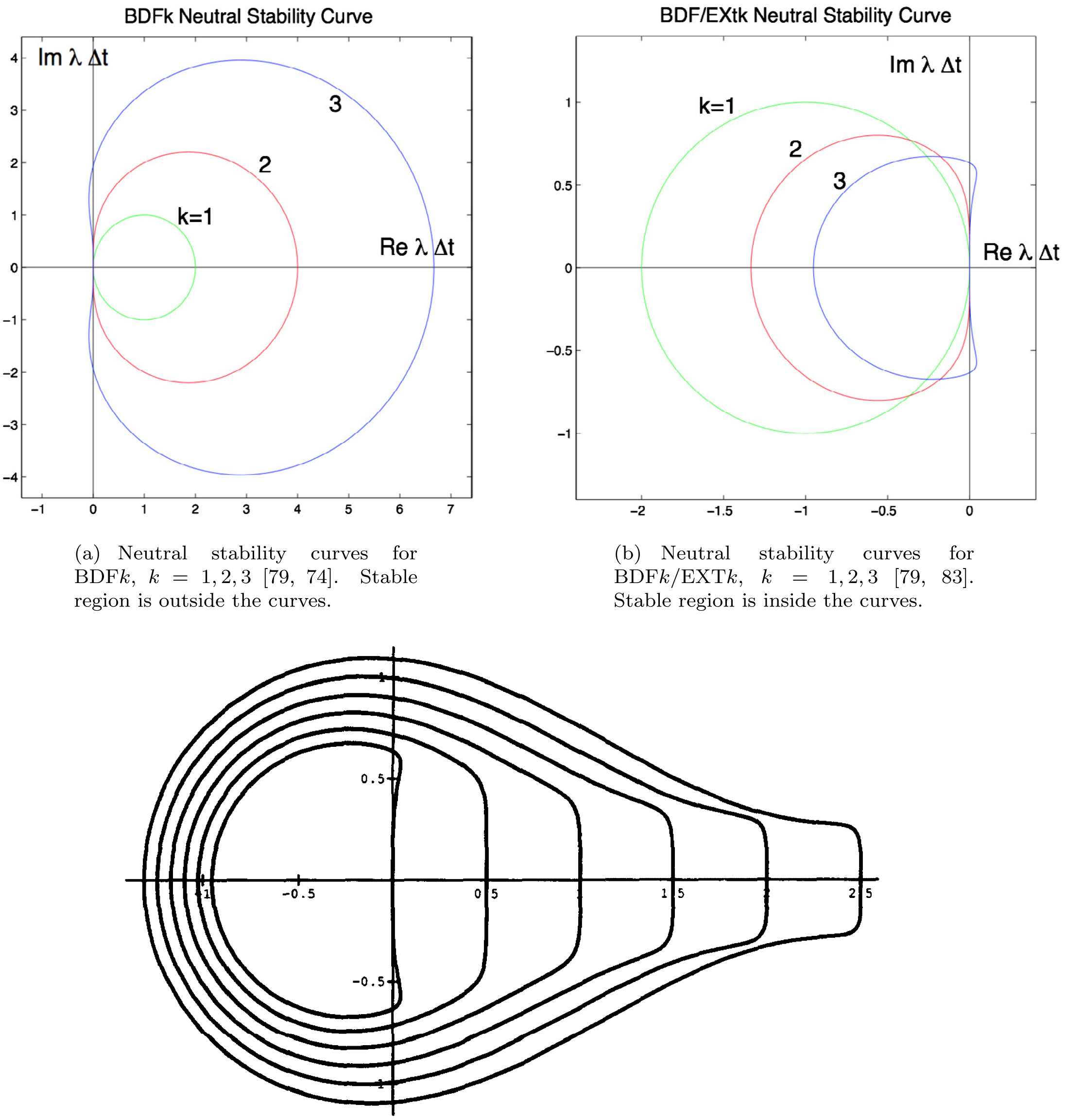
$$\mathcal{E}\left[\underline{\mathbf{C}}(t)\,\underline{\mathbf{u}}(t)\right]_{EXT} = \mathcal{C}(\mathbf{x},t) + \mathcal{O}[\Delta t^k]. \tag{51}$$

2.2.6. Stability

Stability regions of a pure implicit BDFk scheme and an explicit BDFk/EXTk scheme inferred from a model equation $du/dt=\lambda u$ for stationary domains are shown in Fig. 2(a), (b), respectively. It can be seen that BDFk scheme is always stable for systems with real negative eigenvalues, such as viscous problems, and BDFk/EXTk is stable for convective problems (which have imaginary eigenvalues), provided the corresponding CFL condition is met [74,79]. For the mixed explicit-implicit methods, when the two schemes are combined (typically extending an implicit, BDFk, treatment to a viscous term, and an explicit, BDFk/EXTk, treatment to a convective term, such as in Eq. (47)), the stability properties are determined by a relative strength of the convective and viscous contributions, but, generally, an addition of an implicit viscous term enlarges the stability region of a BDFk/EXTk scheme [80], see Fig. 2(c).

Hereafter, we will be referring to a scheme used to discretize Eq. (47) as BDFk, with the understanding that the convective term is treated explicitly with EXTk, and viscous and pressure terms are treated implicitly.

In general, analyzing stability of the time integration on the moving domains, i.e. in the ALE formulation, is not a trivial task [81,82]. However, in the current case of a rigid body motion of a subdomain, the situation simplifies quite a bit, since the mesh velocity $\mathbf{w}(\mathbf{x},t)$ in this case obeys the relation $\nabla\mathbf{w}=0$. It can be seen that the problematic terms of an undetermined sign that hinder the stability proofs for the high-order time accurate schemes in the ALE formulation are proportional to $\nabla\mathbf{w}$ (see [82]). Stability characteristics of time stepping schemes thus should not change when applied to the moving domains that move as rigid bodies, however the corresponding CFL condition must refer to the relative fluid velocity with respect to the moving mesh.



(a) Neutral stability curves for BDFk, $k = 1, 2, 3$ [79, 74]. Stable region is outside the curves.

(b) Neutral stability curves for BDFk/EXTk, $k = 1, 2, 3$ [79, 83]. Stable region is inside the curves.

(c) Neutral stability curves for the mixed explicit-implicit methods of order 3 [80]. Stable region is inside the curves. The inner-most curve corresponds to an explicit BDF3/EXT3 scheme, while the outer curves correspond to progressively larger values of the implicit viscous damping.

Fig. 2. Stability characteristics of the BDFk, BDFk/EXTk and mixed methods.

2.2.7. Stokes problem

After some rearrangement, the temporally and spatially discretized Navier-Stokes equations (42) and (43) become

$$\underline{\mathbf{H}}^n \underline{\mathbf{u}}^n - \widetilde{\underline{\mathbf{D}}}^{T,n} \underline{p}^n = \underline{\mathbf{f}}^n \tag{52}$$

$$\widetilde{\underline{\mathbf{D}}}^n \underline{\mathbf{u}}^n = 0 \tag{53}$$

where the Helmholtz operator

$$\underline{\mathbf{H}}^n = \left(\frac{\beta_0}{\Delta t} \underline{\mathbf{B}}^n + \frac{1}{\mathrm{Re}} \underline{\mathbf{A}}^n \right), \tag{54}$$

and the right hand side

$$\underline{\mathbf{f}}^n = -\sum_{p=1}^{k} \beta_{pk}\underline{\mathbf{B}}^{n-p}\underline{\mathbf{u}}^{n-p} - \sum_{q=1}^{k} \gamma_{qk}\underline{\mathbf{C}}^{n-q}\underline{\mathbf{u}}^{n-q}. \tag{55}$$

We see that explicit handling of the nonlinear convective terms permits the discretized governing equations to be expressed as a Stokes problem. Note that the Stokes problem given by Eqs. (52)–(53) that resulted from a temporal discretization of an index 2 differential-algebraic equation system given by Eqs. (42)–(43) due to incompressibility constraint, is fully coupled and is difficult to solve directly. In the current methodology, it is solved using a pressure-velocity splitting method in the pressure correction formulation [74] by first rewriting the original Stokes system (52)–(53) in a modified form

$$\begin{pmatrix} \underline{\mathbf{H}} & -\underline{\mathbf{H}}\,\underline{\mathbf{W}}\,\underline{\widetilde{\mathbf{D}}}^T \\ -\underline{\widetilde{\mathbf{D}}} & 0 \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}^n \\ \underline{p}^n - \underline{p}^{n-1} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{f}}^n + \underline{\widetilde{\mathbf{D}}}^T\,\underline{p}^{n-1} \\ 0 \end{pmatrix} + \begin{pmatrix} \underline{\mathbf{r}}^n \\ \underline{0} \end{pmatrix}, \tag{56}$$

where $\underline{\mathbf{W}}$ is the matrix operator to be determined, and p^{n-1} is the pressure from the previous time step. The term $\mathbf{r}^n$ in the right-hand side is the residual that accounts for the difference between Eq. (56) and its exact representation (52)–(53) whose value depends on the choice of $\underline{\mathbf{W}}$. The residual is discarded in the pressure-velocity split solution and represents the splitting error. For example, the choice $\underline{\mathbf{W}} = \underline{\mathbf{H}}^{-1}$ would result in a zero splitting error and would lead to a fully implicit Uzawa algorithm, which implementation is however rather difficult and costly [74]. The choice $\underline{\mathbf{W}} = \frac{\Delta t}{\beta_0}\underline{\mathbf{B}}^{-1}$ which is employed in classical fractional-step methods, would lead to a high splitting error (order one with Δt for projection methods and order two for pressure correction methods [74,83]). Higher-order approximation for $\underline{\mathbf{W}}$ can be constructed by retaining more terms in a Taylor series expansion of $\underline{\mathbf{H}}^{-1}$. The following form, for example,

$$\underline{\mathbf{W}} = \frac{\Delta t}{\beta_0}\underline{\mathbf{B}}^{-1} - \left(\frac{\Delta t}{\beta_0}\right)^2 \underline{\mathbf{B}}^{-1}\underline{\mathbf{A}}\,\underline{\mathbf{B}}^{-1} + \left(\frac{\Delta t}{\beta_0}\right)^3 (\underline{\mathbf{AB}}^{-1})^2\,\underline{\mathbf{B}}^{-1}, \tag{57}$$

leads to a decoupling error that is of order three for projection methods and of order four for pressure correction methods [74],

$$\underline{\mathbf{r}}^n = \left(\frac{\Delta t}{\beta_0}\right)^3 (\underline{\mathbf{AB}}^{-1})^2\underline{\widetilde{\mathbf{D}}}^T \left(\underline{p}^n - \underline{p}^{n-1}\right) = O(\Delta t^4), \tag{58}$$

and is employed in the current methodology. With this choice of $\underline{\mathbf{W}}$ and omitting the residual term, the factorized form of Eq. (56) leads to a decoupled system in pressure and velocity

$$\begin{pmatrix} \underline{\mathbf{H}} & -\underline{\mathbf{H}}\,\underline{\mathbf{W}}\,\underline{\widetilde{\mathbf{D}}}^T \\ \mathbf{0} & \underline{\widetilde{\mathbf{D}}}\,\underline{\mathbf{W}}\,\underline{\widetilde{\mathbf{D}}}^T \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}^n \\ \underline{p}^n - \underline{p}^{n-1} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{f}}^n + \underline{\widetilde{\mathbf{D}}}^T\,\underline{p}^{n-1} \\ -\underline{\widetilde{\mathbf{D}}}\,\underline{\mathbf{H}}^{-1}(\underline{\mathbf{f}}^n + \underline{\widetilde{\mathbf{D}}}^T\,\underline{p}^{n-1}) \end{pmatrix}, \tag{59}$$

which, due to its low splitting error, allows for up to the fourth order temporal integration of the N.-S. equations, restricted by the order of the time integrator in (42).

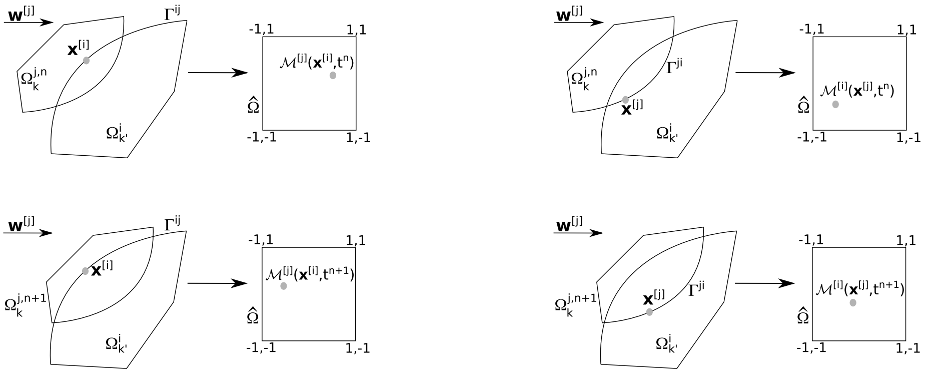
2.2.8. Boundary conditions

Homogeneous Neumann boundary conditions are naturally satisfied with weak formulation [74]. Boundary conditions at moving wall, Dirichlet and interface boundaries are satisfied by adding an additional constraint to the problem. We first describe how homogeneous boundary conditions are prescribed and proceed to setting up inhomogeneous boundary conditions.

Homogeneous conditions An unassembled Mask matrix, $\underline{M}_L$, is used in the discretized representation of the problem for the satisfying of homogeneous Dirichlet boundary conditions [74,72]. In the current moving mesh implementation, boundary condition types are assigned only upon initialization of the simulation, so the Mask matrix is independent of time. The Mask matrix enforces homogeneous boundary conditions on, for example, the term involving the Mass operator as follows

$$\big(\mathbf{u}(\mathbf{x},t), \mathbf{v}(\mathbf{x},t)\big) \rightarrow \underline{\mathbf{v}}^{T,n}\underline{\mathbf{Q}}^T\underline{\mathbf{M}}_L\underline{\mathbf{B}}_L^n\underline{\mathbf{M}}_L\underline{\mathbf{Q}}\underline{\mathbf{u}}^n. \tag{60}$$

Inhomogeneous conditions To impose inhomogeneous Dirichlet, moving wall and interface boundary conditions, the solution, $\underline{\mathbf{u}}^n$ is separated into a homogeneous part, $\underline{\mathbf{u}}_0^n$, that satisfies the homogeneous conditions on Dirichlet, moving wall and interface boundaries, and an inhomogeneous part, $\underline{\mathbf{u}}_b^n$, that satisfies the corresponding inhomogeneous conditions. The homogeneous part is a solution of the discretized equations (52) and (53), and it remains on the left-hand side of the problem. The inhomogeneous part is given by any function that is continuous through the domain and satisfies the conditions on $\partial\Omega_D^{g,n}$, $\partial\Omega_W^n$ and $\Gamma^{ij,n}$, and it thus alters the right-hand side of equations (52) and (53). Such a function, $\underline{\mathbf{u}}_b^n$, can be constructed using any inexpensive projection method [74]. The discrete values of boundary arrays $\underline{\underline{\mathbf{u}}}_d^n$ and $\underline{\underline{\mathbf{w}}}^n$ at the Dirichlet and moving wall boundaries are obtained in a straightforward manner by discretizing the corresponding equations (3b) and (3d). Double lines denote the discrete boundary condition arrays defined at the boundaries. We will devote the rest of this section to describing of how the interface boundary arrays $\underline{\underline{\mathbf{u}}}_{interf}^n$ are obtained at each time step.



(a) Boundary point on stationary interface, Γ^{ij}

(b) Boundary point on moving interface Γ^{ji}

Fig. 3. Illustrative representation of points on interface boundaries being mapped to reference coordinates at time t^n and t^{n+1}. $\Omega^i_{k'}$ represents a stationary domain, Ω^j_k–moving domain.

Setting up the interface conditions The procedure of specifying the interface conditions consists of two steps: a search step served to locate the interface points within the neighboring subdomain, and an interpolation step, to obtain the current interface velocity values $\underline{\underline{\mathbf{u}}}^n_{interf}$ at these points. In what follows, we describe these two steps.

a) Search step In the present methodology, a search procedure is performed once at the beginning of each timestep and is described in detail in Ref. [72]. The difference with a stationary overlapping method is that, since each subdomain now moves relative to the other, the locations of interface points within the other subdomain now change at each timestep. Upon the occasion that an interface node moves outside of the other subdomain, user imposed Dirichlet boundary conditions could be enforced at that point, but we did not encounter such situations in the problems treated in the current paper.

In a search procedure, first, the element in $\Omega^{j,n}$ that encompasses the physical coordinates of the interface point $\mathbf{x}^{[i],n} \in \Gamma^{ij,n}$ $(\Omega^{j,n}_k)$ is determined. When it is determined, the coordinates of the interface point in the laboratory frame, $\mathbf{x}^{[i],n}$ are mapped to primary coordinates, $\mathbf{r}^{[j],n} \in [-1,+1]^d$, of the element $\Omega^{j,n}_k$. Fig. 3 shows this mapping from physical to primary coordinates during the mesh movement. We see that the search procedure must necessarily be performed at each timestep for both moving and stationary domains. The search is then treated as an optimization procedure to determine an interface point's primary coordinates, $\mathbf{r}^{[j],n}$, within the element $\Omega^{j,n}_k$,

$$\min_{\mathbf{r}\in[-1,+1]^d} h(\mathbf{r}), \tag{61}$$

where $h(\mathbf{r}) = \left|\mathbf{x}^{[i],n} - \mathcal{M}^{-1,[j]}_k(\mathbf{r},t^n)\right|^2$, $\mathcal{M}^{-1}_k(\mathbf{r},t^n)$ is the function (19) that maps primary coordinates, $\mathbf{r}$, to physical coordinates, $\mathbf{x}^{[j],n}$, at time t^n in the physical element $\Omega^{j,n}_k$. Superscripts "$[j]$" and "n" for the primary coordinates $\mathbf{r}$ were omitted here for readability. Newton's method is used for optimization as

$$\mathbf{r}^{p+1} = \mathbf{r}^p - \left[\mathcal{H}h(\mathbf{r}^p)\right]^{-1}\nabla h(\mathbf{r}^p), \tag{62}$$

where p is the iteration index, $\nabla h(\mathbf{r})$ is the gradient vector and $\mathcal{H}h(\mathbf{r})$ is the Hessian matrix of the optimization function.

b) Interpolation step Interpolation step consists of a spatial interpolation and a temporal interpolation to set up the corresponding spatially and temporally accurate boundary conditions for the interface values at time step t^n.

Spatial interpolation Spatial interpolation is performed in a similar manner as in a corresponding stationary overlapping grid method [72] and consists of a Lagrange interpolation of a computed solution from the adjacent subdomain

$$\mathbf{u}^{[j],n}_{interp}\big|_{\mathbf{x}^{[i],n}} = \sum_{l=0}^{N}\sum_{m=0}^{N} \underline{\mathbf{u}}^{[j],n}_{k,lm}\phi_l\left(r^{[j],n}_1\right)\phi_m\left(r^{[j],n}_2\right),\ \mathbf{x}^{[i],n} \in \Gamma^{ij,n} \to \mathbf{r}^{[j],n} \in \Omega^{j,n}_k, \tag{63}$$

(triple summation in 3 dimensions). Here, the subscript "*interp*" stands for an *interpolated* value inside the $\Omega^{j,n}$ subdomain. The Lagrange interpolation (63) is consistent with the accuracy of the polynomial approximation within the SEM spatial scheme, and returns a spectrally accurate value for $\mathbf{u}^{[j],n}_{interp}\big|_{\mathbf{x}^{[i],n}}$ with polynomial refinement.

Temporal extrapolation To ensure specified temporal accuracy of the global time stepping scheme for the moving overlapping mesh methodology, an interface coupling scheme based on mth order temporal extrapolation (IEXTm) is designed, that maintains its accuracy without the need for iterations,

$$\mathbf{u}_{interf}^{[i],n}\big|_{\mathbf{x}^{[i],n}} = \sum_{p=1}^{m} \gamma_{pm}\mathbf{u}_{interp}^{[j],n-p}\big|_{\mathbf{x}^{[i],n-p}}, \quad \mathbf{x}^{[i],n-p} \in \Gamma^{ij,n-p}, \tag{64}$$

where subscript *interf* refers to interface boundary conditions used by the interface points $\mathbf{x}^{[i],n} \in \Gamma^{ij,n}$, γ_{pm} are the same extrapolation coefficients as defined in Table 1, and $\mathbf{u}_{interp}^{[j],n-p}$ is a spatially interpolated value from Eq. (63) at time step t^{n-p}. Interface conditions defined with equation (64) for all the points $\mathbf{x}^{[i],n} \in \Gamma^{ij,n}$ are then collected to form the boundary arrays $\underline{\underline{\mathbf{u}}}_{interf}^{[i],n}$ which are used to set up inhomogeneous interface conditions on $\Gamma^{ij,n}$ at time step t^n in a weak form.

To show that the IEXTm scheme maintains the desired temporal order of accuracy with respect to the laboratory frame in the ALE formulation, the same procedure that led to Eq. (51) can be used by rewriting

$$\mathbf{u}_{interp}^{[j],n-p}\big|_{\mathbf{x}^{[i],n-p}} = \mathcal{U}(\mathbf{x}^{[i]} - \Delta\mathbf{x}_p^{[i]}, t - p\Delta t), \tag{65}$$

and performing the corresponding multivariable Taylor series expansions such that

$$\mathbf{u}_{interf}^{[i]}(\mathbf{x}^{[i]}, t) = \mathcal{U}(\mathbf{x}^{[i]}, t) + \mathcal{O}[\Delta t^m]. \tag{66}$$

Note that, since the expressions (65), (66) are concerned with the interface points $\mathbf{x}^{[i]} \in \Gamma^{ij}$ that reside in the subdomain Ω^i, both their coordinates, $\mathbf{x}^{[i]}$, and their increment, $\Delta\mathbf{x}_p^{[i]}$, in the laboratory frame are determined with the mesh velocity $\mathbf{w}^{[i]}(\mathbf{x}, t)$ of the subdomain Ω^i.

Iterations Explicit temporal extrapolation of the interface terms in the overlapping grid scheme reduces the stability region of the underlying time integrator. The effects of explicit extrapolation on stability for the BDFk/IEXTm schemes were previously analyzed in [78] for stationary overlapping grids using a model problem of one-dimensional heat equation. It was found that, while first-order temporal schemes were unconditionally stable, larger mesh overlap sizes and an increased number of Schwarz-like subdomain iterations improved the stability properties of high-order schemes for the model problem. It was also found that, in general, more iterations were required to stabilize the third-order schemes as compared to the second-order schemes [78]. To enhance stability properties of the current method, Schwarz-like iterations are performed in addition to the IEXTm scheme, similar to our previous stationary overlapping grid solver [72]. Interface conditions (3e) for the lth iteration are defined

$$\left.\begin{aligned} \mathbf{u}_{interf}^{[i],l=1,n}\Big|_{\mathbf{x}^{[i],n}} &= \sum_{p=1}^{m} \gamma_{pm}\mathbf{u}_{interp}^{[j],l_{max},n-p}\Big|_{\mathbf{x}^{[i],n-p}} \\ \mathbf{u}_{interf}^{[i],l>1,n}\Big|_{\mathbf{x}^{[i],n}} &= \mathbf{u}_{interp}^{[j],l-1,n}\Big|_{\mathbf{x}^{[i],n}} \end{aligned}\right\} \quad \mathbf{x}^{[i],n} \in \Gamma^{ij,n} \tag{67}$$

where the first iteration uses extrapolated values (64) from m previous timesteps at interface nodes, and succeeding iterations use the latest iteration solution in the other subdomain for interpolated values. While velocity and pressure values are dependent on the iteration count, the linear operators are not since geometry and position do not change with iterations. Since the nonlinear convection operator is treated explicitly, only values from previous timesteps are used in its calculation, and thus iterations do not affect its values at the current timestep.

While a theoretical stability analysis of the overlapping grid interpolation on the moving domains will be addressed in our future work, the test cases show a very similar behavior of the overlapping moving and overlapping stationary domain methods in terms of their stability properties [72,78]. Thus, in the present moving overlapping mesh formulation, no more than two iterations were needed to attain stable solutions with the second-order scheme (BDF2/IEXT2), and four to seven iterations for the third-order scheme (BDF3/IEXT3) for the presented cases.

2.2.9. Fully-discretized equations

Decomposing $\underline{\mathbf{u}}$ into homogeneous and inhomogeneous solutions gives the full discretized equations for two moving overlapping subdomains in block matrix form with iteration and timestep indices as

$$\text{In } \Omega^1: \begin{pmatrix} \underline{\underline{\mathbf{H}}}^{[1],n} & -\underline{\underline{\widetilde{\mathbf{D}}}}^{T,[1],n} \\ -\underline{\underline{\widetilde{\mathbf{D}}}}^{[1],n} & 0 \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_0^{[1],l,n} \\ \underline{p}_L^{[1],l,n} \end{pmatrix} = \begin{pmatrix} \underline{\underline{\mathbf{F}}}^{[1],l,n}(\underline{\mathbf{u}}^{[2],l_\alpha,n_\alpha}) \\ \underline{F}_p^{[1],l,n}(\underline{\mathbf{u}}^{[2],l_\alpha,n_\alpha}) \end{pmatrix}, \tag{68}$$

$$\text{In } \Omega^2: \begin{pmatrix} \underline{\underline{\mathbf{H}}}^{[2],n} & -\underline{\underline{\widetilde{\mathbf{D}}}}^{T,[2],n} \\ -\underline{\underline{\widetilde{\mathbf{D}}}}^{[2],n} & 0 \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}_0^{[2],l,n} \\ \underline{p}_L^{[2],l,n} \end{pmatrix} = \begin{pmatrix} \underline{\underline{\mathbf{F}}}^{[2],l,n}(\underline{\mathbf{u}}^{[1],l_\alpha,n_\alpha}) \\ \underline{F}_p^{[2],l,n}(\underline{\mathbf{u}}^{[1],l_\alpha,n_\alpha}) \end{pmatrix}, \tag{69}$$

where the double underlined assembled operators are now modified with the unassembled Mask matrix for satisfying Dirichlet, moving wall, and interface boundary conditions

$$\underline{\underline{\mathbf{A}}}^{[i],n} = \underline{\mathbf{Q}}^{[i]\,T}\underline{\mathbf{M}}_L^{[i]}\underline{\mathbf{A}}_L^{[i],n}\underline{\mathbf{M}}_L^{[i]}\underline{\mathbf{Q}}^{[i]}, \tag{70}$$

$$\underline{\underline{\mathbf{B}}}^{[i],n} = \underline{\mathbf{Q}}^{[i]\,T}\underline{\mathbf{M}}_L^{[i]}\underline{\mathbf{B}}_L^{[i],n}\underline{\mathbf{M}}_L^{[i]}\underline{\mathbf{Q}}^{[i]}, \tag{71}$$

$$\underline{\underline{\mathbf{C}}}^{[i],n} = \underline{\mathbf{Q}}^{[i]\,T}\underline{\mathbf{M}}_L^{[i]}\underline{\mathbf{C}}_L^{[i],n}\underline{\mathbf{M}}_L^{[i]}\underline{\mathbf{Q}}^{[i]}, \tag{72}$$

$$\underline{\underline{\widetilde{\mathbf{D}}}}^{[i],n} = \underline{\widetilde{\mathbf{D}}}_L^{[i],n}\underline{\mathbf{M}}_L^{[i]}\underline{\mathbf{Q}}^{[i]} \tag{73}$$

and

$$\underline{\underline{\mathbf{H}}}^{[i],n} = \left(\frac{\beta_0}{\Delta t}\underline{\underline{\mathbf{B}}}^{[i],n} + \frac{1}{\mathrm{Re}}\underline{\underline{\mathbf{A}}}^{[i],n}\right). \tag{74}$$

The arrays on the right-hand side of (68) and (69) depend on the inhomogeneous solution $\underline{\mathbf{u}}_b^{[i],l,n}$, due to inhomogeneous Dirichlet, moving wall, and interface boundary conditions. It is important to note that the inhomogeneous solution $\underline{\mathbf{u}}_b^{[i],l,n}$ depends on the solutions from the adjacent subdomain (from previous iterations and/or time steps l_α, n_α) through the dependence of interface conditions via equations (63) and (67), which defines the subdomain coupling

$$\underline{\underline{\mathbf{F}}}^{[i],l,n}(\underline{\mathbf{u}}^{[j],l,n}) = \underline{\mathbf{f}}^{[i],n} - \underline{\mathbf{Q}}^{[i]\,T}\underline{\mathbf{M}}_L^{[i]}\underline{\mathbf{H}}_L^{[i],n}\underline{\mathbf{u}}_{b,L}^{[i],l,n}(\underline{\mathbf{u}}^{[j],l_\alpha,n_\alpha}), \tag{75}$$

$$\underline{F}_p^{[i],l,n}(\underline{\mathbf{u}}^{[j],l,n}) = \underline{\widetilde{\mathbf{D}}}_L^{[i],n}\,\underline{\mathbf{u}}_{b,L}^{[i],l,n}(\underline{\mathbf{u}}^{[j],l_\alpha,n_\alpha}), \tag{76}$$

with

$$\underline{\mathbf{H}}_L^{[i],n} = \left(\frac{\beta_0}{\Delta t}\underline{\mathbf{B}}_L^{[i],n} + \frac{1}{\mathrm{Re}}\underline{\mathbf{A}}_L^{[i],n}\right). \tag{77}$$

Note that assembled operators are subdomain dependent, since they depend on different Jacobians, and time dependent, since Jacobians change as the mesh moves, although they are not iteration dependent for the reasons stated in Section 2.2.8. Mask matrices and connectivity operators are subdomain dependent, although they do not depend upon time or iteration count in the current methodology.

The coupling terms in the right-hand side of (68) and (69) are updated at the beginning of each iteration and the Stokes problems are solved independently for each subdomain. Note that the Stokes problem arising in the coupled formulation differs from the corresponding Stokes problem of a single domain given by Eqs. (52)–(53) only by the presence of the coupling terms in the right-hand side of equations, thus the matrix factorization with high-order splitting methods for the coupled Stokes problem proceeds in the same way as discussed in Section 2.2.5, see also [72]. Additionally, the pressure correction term $p^n - p^{n-1}$ will be replaced by $p^{l,n} - p^{l_{max},n-1}$ due to the addition of iterations in the coupled problem. While bolded operators represent block-diagonal formulations of the scalar-valued operators, as discussed previously, the interpolating derivative operator is defined $\underline{\underline{\widetilde{\mathbf{D}}}}^{[i],n} = \left[\underline{\underline{\widetilde{D1}}}^{[i],n}\ \underline{\underline{\widetilde{D2}}}^{[i],n}\ \underline{\underline{\widetilde{D3}}}^{[i],n}\right]$ to ensure proper multiplication of $\underline{\underline{\widetilde{\mathbf{D}}}}^{T,[i],n}$ and $\underline{p}_L^{[i],l,n}$. In the current formulation, the pressure field obtained in each subdomain is defined up to an arbitrary constant. The pressure field is further adjusted by matching the subdomain pressure solutions at a single spatial location that allows for a pressure field that is consistently defined throughout the global domain.

2.3. Schematic of the procedure

A schematic of the coupled solution procedure is given in Fig. 4. As discussed above, the procedure in each subdomain is fairly autonomous, with, apart from initialization, only search and iteration steps requiring global communication. Since the mesh geometry is updated once per timestep, the search step is also required once per timestep to determine the correct primary coordinates of each interface point in terms of the coincident elements in the neighboring mesh. The interpolation step, however, is required once per iteration to ensure that the most recent solution set is used for the interface conditions. In the next section, we describe the parallelization strategy for the methodology, focusing on global communication procedures, as single-domain parallelization procedures were described elsewhere [73].

3. Parallel communication

Parallel implementation of the moving overlapping domain method is based on a dual-session communication framework previously developed by the authors [84,72]. The framework allows for fully independent computations in each session, while enabling the sessions to communicate with each other at certain global checkpoints. Within each session, communication among local processors is handled with MPI Intracommunicators, while global communication is accomplished through the establishment of an MPI Intercommunicator [72].

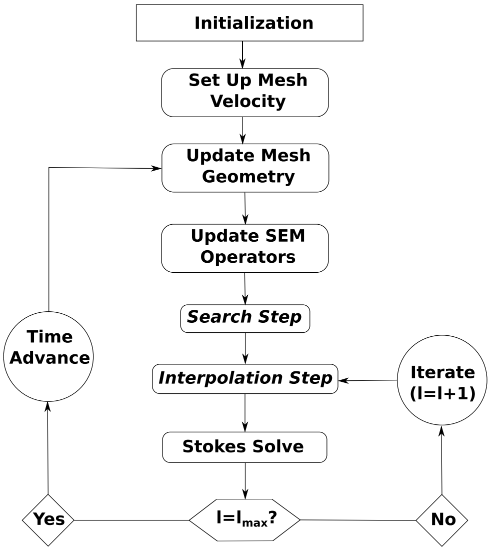


Fig. 4. Basic procedures for computation of moving overlapping mesh methodology. Steps requiring global communication are in italic.

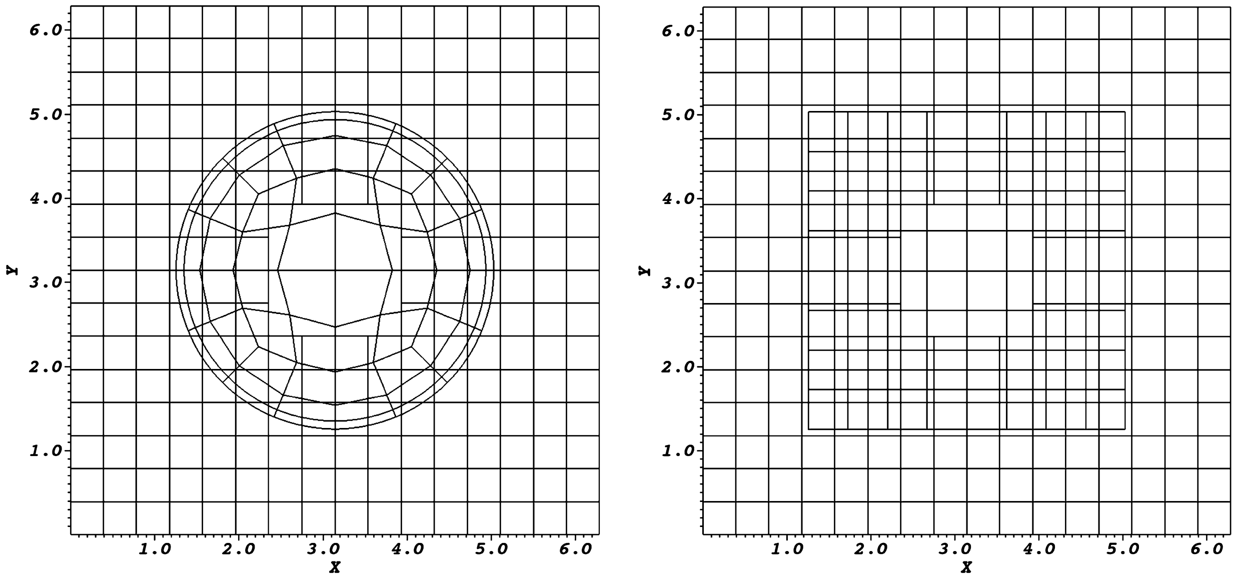
In a computation involving the moving overlapping domain method, global communications can be categorized into two groups associated with the interface conditions exchange procedure as described in Section 2.2.8: *Search step* and *Interpolation step*. During *Search step*, each processor at a given subdomain scans through the GL points in its custody that are flagged as "interface" points. Since interface boundaries in the current implementation are not altered throughout the simulation, this flagging is performed only once per simulation, usually during a grid construction process. The total number of interface points found by each processor at a given session during the *Search step* is divided equally among the processors in the other session, and the point identities (including physical coordinates in the laboratory frame) are distributed via the Intercommunicator among the processors in the other session. Upon receipt of point identities, the sessions independently determine whether each "interface" point lies within their subdomain boundary, and if it does not, the point is sent back to its originating session where it is treated with user prescribed Dirichlet condition instead of interpolation. For all accepted interface points within the subdomain's boundaries, the *Search Step* proceeds to determine which local element encompasses each of these points. The points are then locally distributed to the corresponding "owner" processors via a Crystal Router Algorithm [85] which uses fast all-to-all communications for efficient data transfer, and is built into the SEM solver. The *Search step* continues by determining and tabulating the primary coordinates of each point with respect to the element in which it lies. During the *Interpolation Step*, the parallel interface and the tabulated point coordinates built during the *Search step* are efficiently utilized to perform, via a Crystal Router, a spectral interpolation of the corresponding solution variables to the interface points. The interpolated values are passed back to the neighboring session via the Intercommunicator, where the information is unwrapped and sent locally (through Intracommunicators) back to the processors requiring each interface point data. The interpolated values are utilized as inhomogeneous Dirichlet conditions at the interface boundaries at each subdomain.

As opposed to our previous *stationary* overlapping grid methodology [72], where the *Search step* needed to be performed only once per simulation, in the case of the moving grids, it needs to be performed at every time step, since the relative position between the grids changes dynamically. However, it does not have to be performed at every iteration step, as opposed to *Interpolation step*, which does. Although *Search step* is the most expensive part of the parallel communication procedure, an efficient implementation of the current dual-session framework together with utilization of optimized Crystal Router routines whenever possible, does not hinder the scalability and efficiency of the algorithm, as demonstrated in Section 4.4 via scalability tests.

Note that the current numerical formulation of the moving overlapping methodology is general enough so that the flexible/deforming domain motions can be handled as well, although the current paper is concerned strictly with the rigid body motions, where grid deformation is not necessary. The methodology however does not allow at the present moment for the addition or removal of the elements from the subdomains, so that the "hole-cutting" is currently not considered [47, 86]. All the present test cases employ grids that are designed to preserve the topology of the overlap region during the subdomain motion.

4. Results

The capability of the developed moving overlapping grid method will be demonstrated through two- and three-dimensional example problems. The spatial and temporal accuracy of the scheme will be documented, and the method's ability to realistically model fluid flow when influenced by moving rigid bodies will be illustrated. Computational param-



(a) Circular interior mesh

(b) Square interior mesh

Fig. 5. Two-mesh domains for convecting eddy simulations, with element boundaries shown.

eters for the test cases presented in the current paper were chosen primarily based on the availability of the previously published data for a direct comparison, as is detailed below.

4.1. Convecting two-dimensional eddies

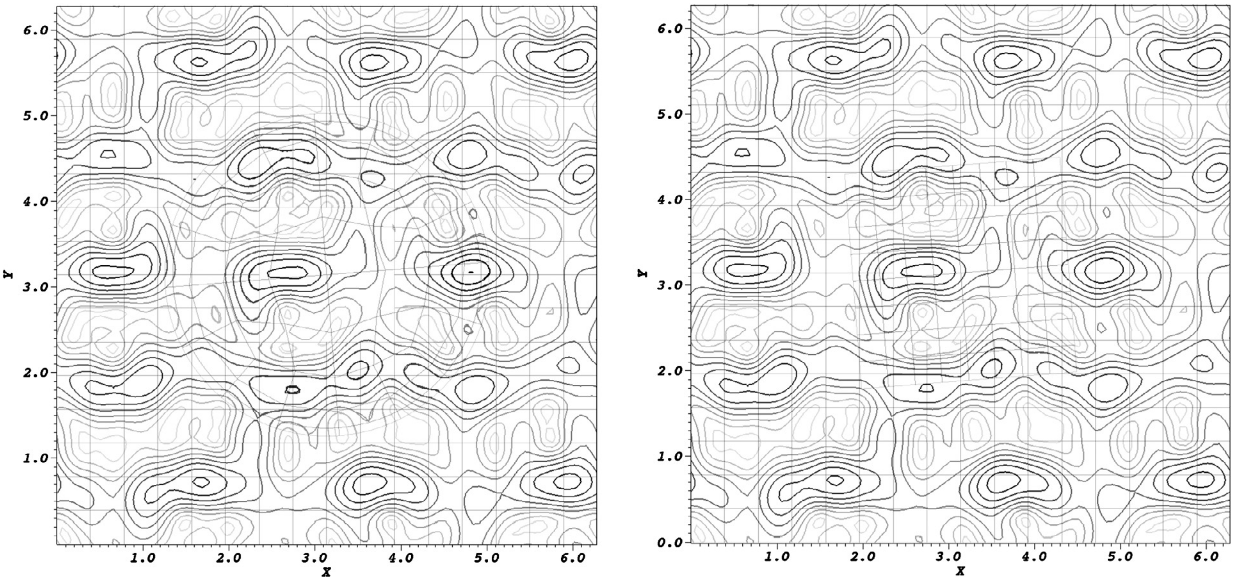
The spatial and temporal convergence rates are determined from a simulation of convecting eddies in a periodic global domain, comprised of two subdomains, for which an exact solution exists [87]. The original Walsh solution [87] reflects a case of stationary eddies decaying in time and is represented by a linear combination of the eigenfunctions that are products of Sin[nx] or Cos[nx] with Sin[my] or Cos[my], with the corresponding eigenvalues $\lambda = -(n^2 + m^2)$, where n and m are non-negative integers. For the eddies convecting with the constant velocity (u_0, v_0), the solution can be found by transforming the Walsh's solution from convecting into a stationary frame of reference. Please, see [72] for more details of the convecting eddy case including the derivation of the pressure solution. The same exact solution as in [72] was used for the present test case. For the pressure comparison, the pressure field that has a zero mean value on the global domain is used for both exact and computational solutions. The mean pressure value of the solution on the global domain can be calculated by treating grid points within the overlap region with a weight of one-half during the calculation of the global averages, see [72] for more details. The calculated global mean value then can be subtracted from both subdomain solutions to arrive at a zero-mean pressure field.

The global $2\pi \times 2\pi$ two-dimensional domain is decomposed into an interior and exterior mesh. The exterior mesh contains a vacancy at its center which is covered by the interior mesh. Two configurations of interior grids were used, a circular mesh and a square mesh, as seen in Fig. 5. In all simulations the exterior mesh was held stationary and the interior mesh was constrained to move in a prescribed fashion. Note that this motion is a type of "pseudo", or "virtual", motion as the movement of the mesh has no effect on the flow since there are no solid structures contained within. In the first set of verification cases, both circular and square interior meshes were constrained to rotate, while in succeeding simulations sliding motion was prescribed.

Simulations were performed with Re = 20, where the Reynolds number is defined, $\mathrm{Re} = L^* U^* / \nu^*$, with L^* the reference length, U^* the reference velocity, and ν^* the kinematic viscosity. Quantities with stars denote the dimensional quantities, and the ones without stars are non-dimensional. All non-dimensional length and velocity variables are given here, with the presumption that corresponding dimensional counterparts were normalized by L^* (taken here as $L^* = L_x^*/(2\pi)$, where L_x^* is the domain length in x direction), and by U^* equal to the eddy convection velocity in x-direction.

For the rotating mesh test cases, interior meshes are rotated counter-clockwise as rigid bodies about their center of mass, which coincides with the center of the global domain $[\pi, \pi]$, with a non-dimensional angular velocity $\Omega = \Omega^* L^* / U^* = \pi/4$. For sliding mesh cases, the interior mesh slides to the right with the non-dimensional velocity $w = w^*/U^* = 1.5$.

In Fig. 6, we show velocity magnitude contours of rotating interior mesh simulations with fluid convection velocity $u_0 = u_0^*/U^* = 1$ and $v_0 = v_0^*/U^* = 0.3$, one with the circular and one with the square interior mesh. This particular convection velocity was chosen so that direct comparison with the results of a stationary overlapping mesh case published in [72] was possible. Note that in the overlap region, two sets of contour lines are presented, although the continuity of values between the two subdomains is such that discrepancies between values in the overlap region are difficult to visually detect.



(a) Circular interior mesh

(b) Square interior mesh

Fig. 6. Velocity magnitude contours of rotating interior mesh simulations at a non-dimensional time $t = U^*t^*/L^* = 0.1$. The darkest contour lines represent a non-dimensional velocity magnitude of three, while the lightest lines, zero. The simulations were performed with 8th-order polynomial approximations, and $\Delta t = 1 \times 10^{-4}$ using second order temporal accuracy and IEXT2, with two iterations per timestep.

We proceed by looking at L^2-norms of the velocity and pressure errors, which are calculated for each subdomain:

$$L^2\text{-error } (\boldsymbol{u}^{[i]}) = \sqrt{\frac{1}{dV^{[i]}} \int\limits_{\Omega^i} (\boldsymbol{u}_{\text{ex}}^{[i]} - \boldsymbol{u}_{\text{comp}}^{[i]})^2 \text{dV}}, \tag{78}$$

$$L^2 \text{ error } (p^{[i]}) = \sqrt{\frac{1}{V^{[i]}} \int\limits_{\Omega^i} \left(p_{\text{ex}}^{[i]} - p_{\text{comp}}^{[i]}\right)^2 \text{dV}}, \tag{79}$$

where $V^{[i]}$ is the volume of the subdomain Ω^i, d is the spatial dimension ($d = 2$ in the current problem), and $_{\text{ex}}$ and $_{\text{comp}}$ subscripts are used to denote exact and computational values, respectively. Errors for all simulations were collected at a non-dimensional time $t = 0.1$.

Results are compared with the data from our previously validated overlapping mesh methodology for stationary meshes [72], where Fig. 7 shows velocity errors with respect to polynomial refinement. Note that for sliding meshes, we present data only for the square interior mesh for compactness, however the results for the circular interior mesh show similar trends.

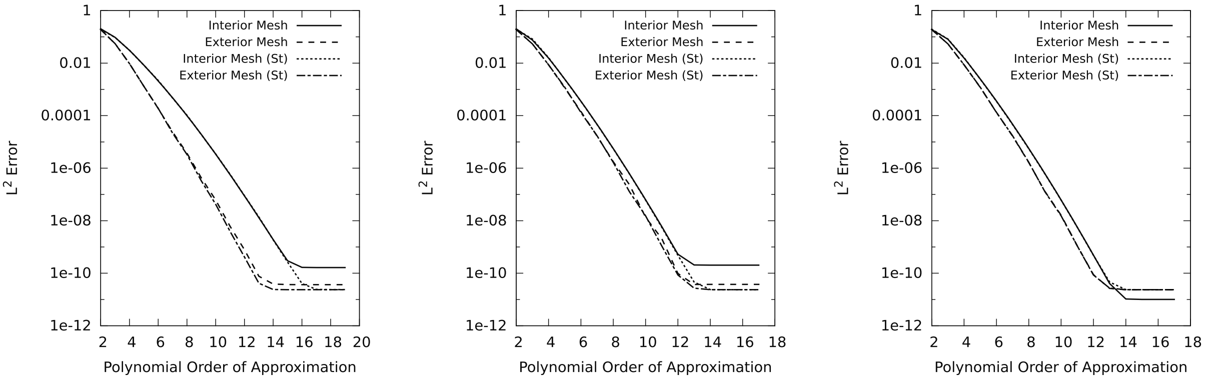
Results from the moving overlapping mesh methodology closely correlate with the stationary overlapping mesh data. Both interior and exterior meshes achieve full convergence at the polynomial orders between 13th and 15th, using the non-dimensional timestep of $\Delta t = 1 \times 10^{-4}$, with exterior meshes usually converging first. In addition, convergence of both meshes is slightly lower for the circular interior mesh cases. A slightly lower convergence rate for circular interior mesh cases is associated with a slower convergence of the SEM on curvilinear meshes [88,74,89,72], and not a result of the moving overlapping mesh methodology. In spite of different slope values in the convergence curves, the convergence with rotating and sliding meshes is still exponential in each case.

Temporal accuracy for both the rotating and sliding interior meshes is presented in Fig. 8. The accuracy in each case achieves the expected third-order convergence rate, where BDF3 and IEXT3 were used.

Pressure accuracy also achieves spectral spatial convergence and the desired third-order temporal convergence rate for each subdomain in all the cases considered. Figs. 9 and 10 show the data for the cases with a square interior mesh.

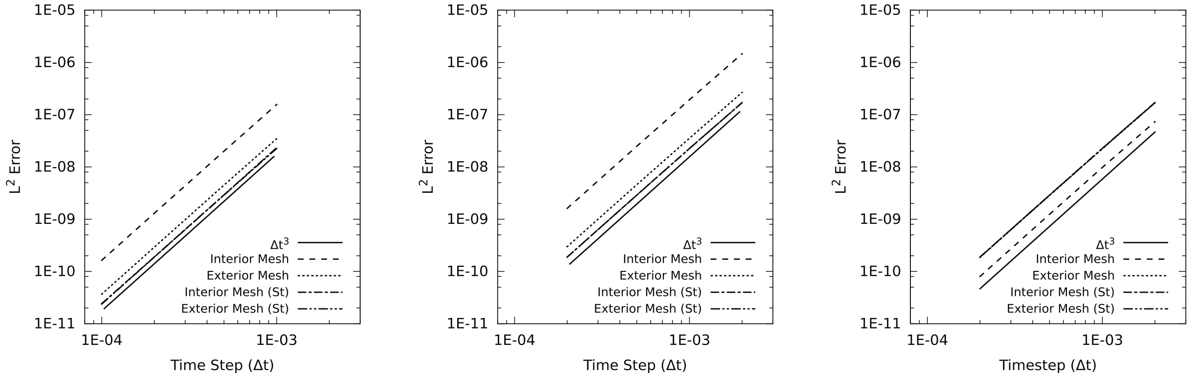
4.2. Two-dimensional oscillating cylinder

The next example considers the flow over an oscillating cylinder and tests the performance of the method in the presence of a solid moving object. The domain is assigned a uniform free stream velocity U_∞^* at the left boundary with outflow conditions specified on the opposing side. Outflow conditions are defined, in non-dimensional form, as $\nabla \boldsymbol{u} \cdot \widehat{\mathbf{n}} = 0$ on $\partial\Omega_O$, where $\widehat{\mathbf{n}}$ is the normal unit vector and $\partial\Omega_O$ is the outflow boundary. Here, all spatial variables are non-dimensionalized



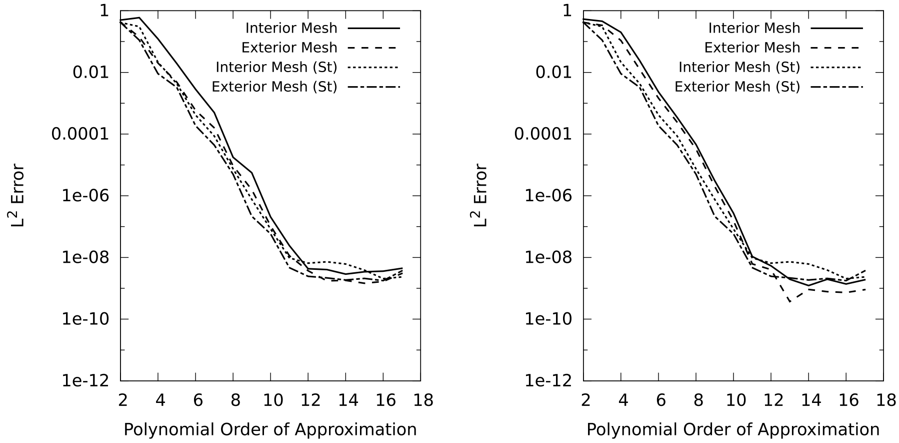
(a) Rotating circular interior mesh (b) Rotating square interior mesh (c) Sliding square interior mesh

Fig. 7. Velocity errors with respect to changing solution polynomial order. The tests were performed with timestep $\Delta t = 1 \times 10^{-4}$ using BDF3 and IEXT3, with four iterations per timestep, and errors were collected at $t = 0.1$. The (St) denotes data from stationary overlapping mesh simulations.



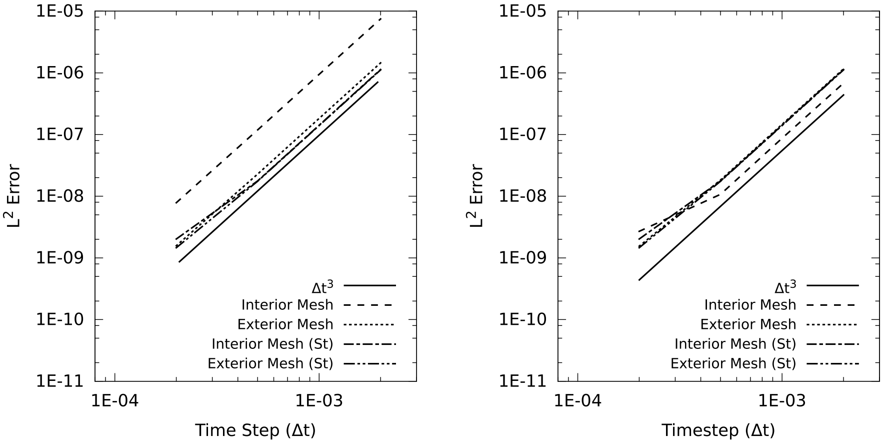
(a) Rotating circular interior mesh (b) Rotating square interior mesh (c) Sliding square interior mesh

Fig. 8. Velocity errors with respect to changing timestep. The tests were performed with 17th-order polynomials using BDF3 and IEXT3, with four iterations per timestep, and errors were collected at $t = 0.1$. The (St) denotes data from stationary overlapping mesh simulations, and a reference line is displayed for comparison to the expected convergence rate.



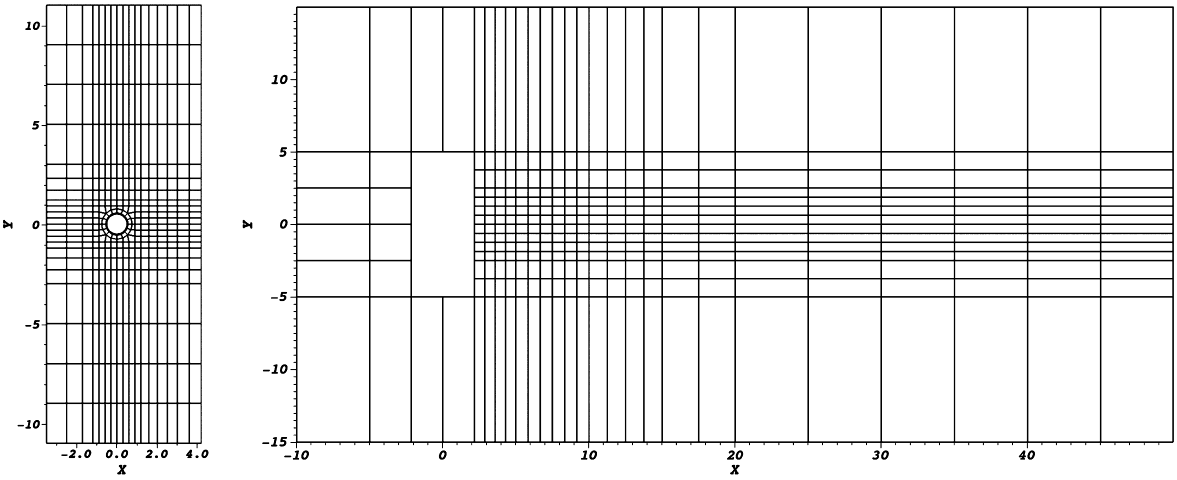
(a) Rotating square interior mesh (b) Sliding square interior mesh

Fig. 9. Spatial convergence of pressure errors. The tests were performed with timestep $\Delta t = 1 \times 10^{-4}$ using BDF3 and IEXT3, with four iterations per timestep, and errors were collected at $t = 0.1$. The (St) denotes data from stationary overlapping mesh simulations.



(a) Rotating square interior mesh (b) Sliding square interior mesh

Fig. 10. Temporal convergence of pressure errors. The tests were performed with 17th-order polynomials using BDF3 and IEXT3, with four iterations per timestep, and errors were collected at $t = 0.1$. The (St) denotes data from stationary overlapping mesh simulations.



(a) Interior mesh (b) Exterior mesh

Fig. 11. Geometry of the oscillating cylinder case, with element boundaries shown. All distances are in terms of the cylinder diameter.

with the cylinder diameter D^*, and velocity variables with the free-stream velocity U^*_∞, with $\mathrm{Re} = U^*_\infty D^*/\nu^*$. Symmetry boundary conditions are imposed on the top and bottom of the domain. The horizontal velocity of the cylinder is set to zero, while the vertical component of motion is governed by the equation

$$y(t) = y_0 + A\sin(2\pi f_0 t), \tag{80}$$

where $A = y^*_{max}/D^*$ (y^*_{max} is the largest vertical displacement of the cylinder) represents the non-dimensional amplitude and $f_0 = f^*_0 D^*/U^*_\infty$ represents the non-dimensional frequency of oscillation. The cylinder is initially placed at (0,0) such that $y_0 = 0$ in Eq. (80), with the global domain spanning from $x = -10$ to 50 diameters, and $y = -15$ to 15 diameters. We also define the frequency ratio $F = f^*_0/f^*_v$ where f^*_v is the vortex shedding frequency of the *fixed* cylinder.

The global domain is decomposed into two overlapping subdomains, with the exterior mesh (see Fig. 11) containing a vacancy for the cylinder movement, and the interior mesh containing the 2D cylinder itself. During the simulations, the exterior mesh remains stationary while the interior mesh is constrained to move with the equation of the oscillation given by Eq. (80). The interior mesh is constructed not to slide out of the global domain or out of the vacancy during the simulations.

4.2.1. Stationary cylinder

For an accurate comparison with the previously published data [90–94], the Strouhal number, or vortex shedding frequency, from the flow around the *stationary* cylinder was first determined using our previously validated stationary

Table 2
Results for a stationary cylinder with Re = 500 in uniform flow using overlapping meshes compared with data presented in [95]. The present data used BDF2/IEXT2 time stepping with 7th order polynomial approximations, $\Delta t = 2.5 \times 10^{-3}$ and two iterations per timestep. Present averages were taken over the interval $300 \leq t \leq 600$, with the solution reaching an asymptotically periodic state at $t \approx 100$.

	Present data	Blackburn et al. [95]
St	0.2281	0.2280
$\widehat{C_l}$	1.202	1.200
$\overline{C_d}$	1.461	1.460
$-\overline{C_{pb}}$	1.504	1.506

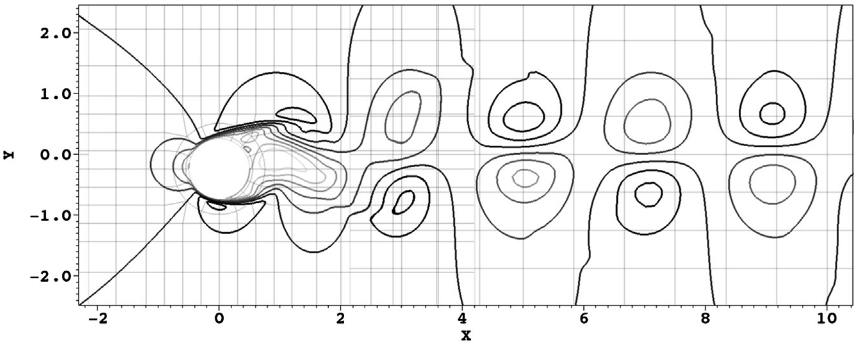


Fig. 12. Enlarged plot of x-direction velocity contours of oscillating cylinder simulation at Re=500 with A=0.25 and F=1.0, at $t = 100$. The lightest lines represent velocity of $u^*/U^*_\infty = -0.5$ while the darkest lines represent $u^*/U^*_\infty = 1.5$. Solution approximations are calculated using BDF2/IEXT2 time stepping, with 7th-order polynomials, $\Delta t = 2.5 \times 10^{-3}$, and two iterations per timestep.

Table 3
Results for an oscillating cylinder with Re = 500, $F = 1.0$ and $A = 0.25$ in a uniform flow compared with results in [95]. BDF2/IEXT2 was run with 7th-order polynomial approximations, $\Delta t = 2.5 \times 10^{-3}$ and two iterations per timestep while BDF3/IEXT3 was run with 9th-order polynomial approximations, $\Delta t = 2 \times 10^{-3}$ and seven iterations per timestep. The averages for both schemes were taken over the interval $250 \leq t \leq 550$, with an asymptotically periodic state reached at $t \approx 75$.

	BDF2/IEXT2	BDF3/IEXT3	Blackburn et al. [95]
$\widehat{C_l}$	1.781	1.781	1.776
$\overline{C_d}$	1.417	1.415	1.414
$-\overline{C_{pb}}$	1.377	1.374	1.377

overlapping mesh methodology [72]. Strouhal number is given as $St \equiv f^*_v D^*/U^*_\infty$, or just f_v with the current normalization. At Reynolds number of 200, our simulations produced a Strouhal number of 0.1998 compared with the Strouhal number of 0.198 found in Udaykumar et al. [94] and of 0.197 found in Williamson et al. [93] at the same Reynolds number. The mean drag of our simulation was 1.372, while Udaykumar et al. [94] found a value of 1.38.

Table 2 compares the values from our overlapping mesh methodology for fixed meshes, at Re = 500, with values published by Blackburn et al. [95] for the stationary cylinder in uniform flow at the same Re. The Strouhal number is denoted by St, while $\widehat{C_l}$ represents the peak coefficient of lift, $\overline{C_d}$ is the mean (time-averaged) coefficient of drag, and $\overline{C_{pb}}$ is the mean base pressure coefficient calculated from the pressures at the furthest upstream (p^*_0) and downstream (p^*_{180}) points on the cylinder surface, $C_{pb} = 1 + 2(p^*_{180} - p^*_0)/\rho^* U^{*2}_\infty$.

4.2.2. Oscillating cylinder

Fig. 12 depicts the x-direction velocity contours of vortices shed from an oscillating cylinder as they travel from the interior mesh across the interface boundary to the exterior mesh. The overlap region contains two sets of contour lines, and visual inspection shows good continuity between the values reported by the two meshes in the overlap region.

The peak lift forces, the mean drag forces, and the mean pressure differences on the oscillating cylinder are compared with the results from [95] in Table 3 for a frequency ratio of 1.0, non-dimensional amplitude 0.25, and Re = 500 for both sets of data. To assess the stability and performance of the third-order accurate temporal scheme for moving rigid bodies, this case was also run with the BDF3/IEXT3 scheme. As can be seen from the Table 3, agreement between the second- and third-order temporal schemes is excellent and both results are very close to the Blackburn et al. [95] data.

In his experimental investigations, Koopmann [91] observed certain frequencies and amplitudes of oscillation that enabled 'lock-in,' where the vortex shedding frequency is equal to the oscillation frequency of the cylinder. Outside of this

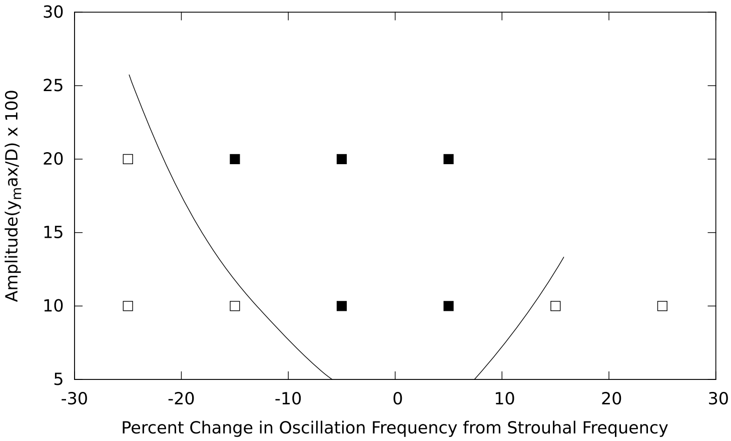


Fig. 13. 'Lock-In' data compared with experimental data from Koopmann [91]. The region inside the solid lines represents Koopmann's lock-in region, the squares represent simulations performed using the moving overlapping mesh method (filled squared displayed lock-in, hollow squares did not). These simulations were performed using BDF2/IEXT2 time stepping with 7th order polynomial approximations, $\Delta t = 2.5 \times 10^{-3}$ and two iterations per timestep. Re = 200 was used for both experimental and numerical data.

lock-in region, the vortex shedding frequency converges to the fixed-cylinder vortex shedding frequency denoted as Strouhal frequency [91,94]. Lock-in behavior of the vortex shedding is a result of the relationship between the vortex formation near the surface of the cylinder and the changing velocity and acceleration of the cylinder with respect to the fluid that causes portions of the forming vortices to detach at different positions with respect to the cylinder's cycle [92]. Fig. 13 shows lines that represent the lock-in region established by Koopmann [91], with data from our moving overlapping mesh method showing which simulations exhibited the lock-in behavior and which didn't. The present data displays good correlation with the experimental results reported in [91].

4.3. Rotating 2D cylinder and 3D sphere

In this section, we consider an interaction of the incoming stream with the rotating solid body. This case serves two purposes: in the first part, we verify the accuracy of the approach in the presence of moving solid boundaries, and in the second part, we extend the methodology to a three-dimensional example.

4.3.1. Rotating 2D cylinder

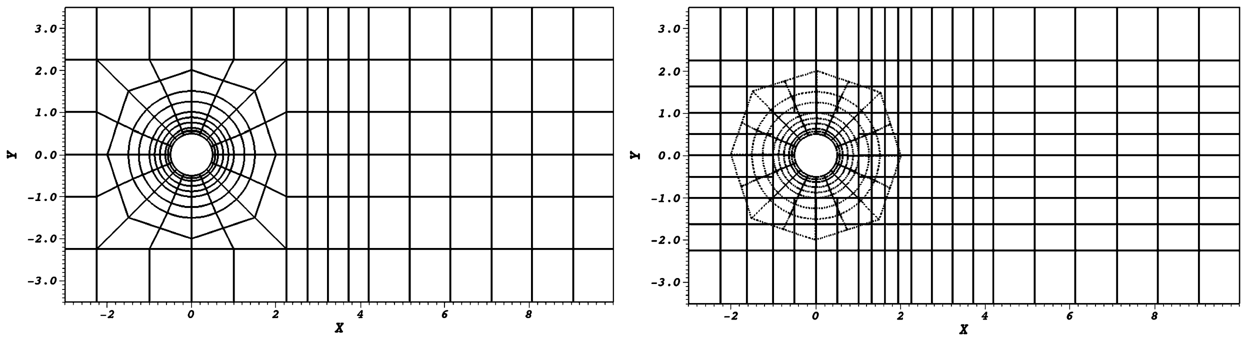
The spatial convergence analysis was performed on a rotating 2D cylinder. The left boundary is assigned uniform freestream velocity, U_∞^*, with outflow conditions prescribed at the opposite side, and symmetry conditions on the top and bottom boundaries. Spatial variables are non-dimensionalized using the cylinder diameter D^*, and velocity variables with the free-stream velocity U_∞^*, with $\mathrm{Re} = U_\infty^* D^*/\nu^*$.

A baseline case was developed as a single mesh with an embedded cylinder (see Fig. 14 a) for a spatial self convergence analysis. Dirichlet boundary conditions were prescribed at the cylinder wall to ensure a cylinder angular velocity of $\Omega = 2.0$, where the non-dimensional angular velocity is given by $\Omega = \Omega^* D^*/(2U_\infty^*)$, and the prescribed Reynolds number was 200. The simulation was performed using 17th order polynomial approximations and 3rd order timestepping (BDF3) with timestep $\Delta t = 1 \times 10^{-5}$.

A moving overlapping mesh case (see Fig. 14 b) was developed for comparison to the baseline case. The interior, moving, mesh was prescribed a rotational velocity $\Omega = 2.0$, while the exterior mesh was held stationary. All moving overlapping mesh simulations were performed using BDF3/IEXT3 and 7 iterations per timestep with timestep $\Delta t = 1 \times 10^{-5}$, at Re = 200. Fig. 15 shows exponential spatial convergence for velocity in both the interior and exterior meshes as the polynomial order nears that used for the single mesh case.

4.3.2. Rotating 3D sphere

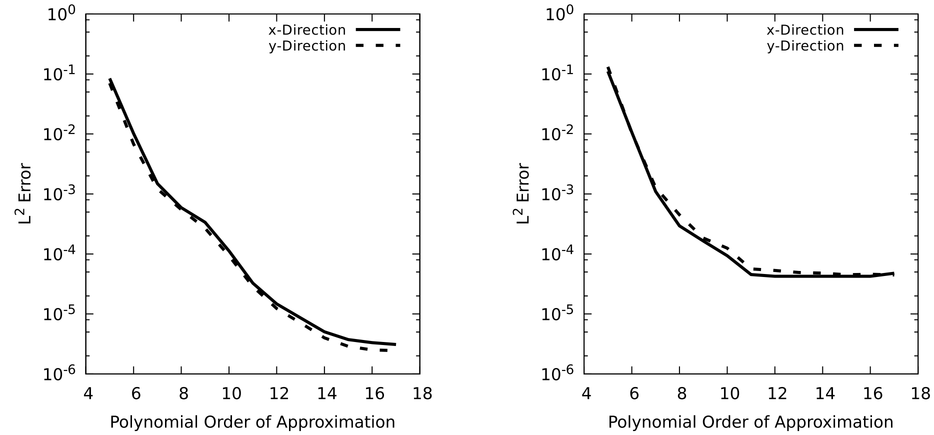
In this test case, we extend our benchmarking to a three-dimensional situation and compare to the previously published data [96–98]. The setup is similar to the one in the rotating cylinder case, albeit in three dimensions, with the three-dimensional subdomain mesh topologies illustrated in Fig. 16. A spherical mesh is formed around the solid walled sphere, which is positioned to cover a vacancy within a large outer mesh. The inner mesh is created to ensure that at least nine grid points lie within the laminar boundary layer for all the simulations performed. Steady inflow velocity U_∞^* is prescribed at one end of the outer mesh and outflow boundary conditions are set at the other, while a steady rotation of the sphere is imposed by establishing the inner mesh velocity with the ALE formulation. Note that the current setup could have been accomplished through a single stationary mesh by assigning non-zero, rotating Dirichlet velocity conditions at



(a) Single Mesh

(b) Moving Overlapping Meshes

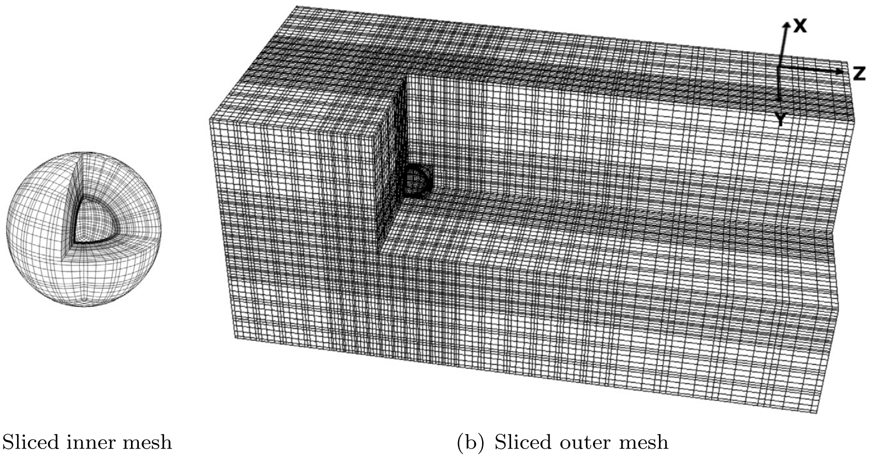
Fig. 14. Geometry of the rotating cylinder case, with element boundaries shown. Distances are presented in terms of the cylinder diameter.



(a) Interior Mesh

(b) Exterior Mesh

Fig. 15. Spatial velocity convergence for the rotating cylinder case. All simulations were performed with timestep $\Delta t = 1 \times 10^{-5}$ using BDF3/IEXT3, with seven iterations per timestep, and errors were computed at t=40.



(a) Sliced inner mesh

(b) Sliced outer mesh

Fig. 16. Mesh geometry of transversely rotating sphere simulation [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 4
A tabular listing of regimes observed in the present simulations for each rotating sphere case performed in the present study.

Re = 100 Ω	Observed regime	Re = 250 Ω	Regime	Re = 300 Ω	Observed regime
0.0	Steady	0.0	Steady	0.0	Unsteady
0.05	Steady	0.05	Transitional	0.05	Unsteady
		0.078	Unsteady		
0.1	Steady	0.1	Unsteady	0.1	Unsteady
		0.2	Unsteady		Unsteady
0.25	Steady			0.25	Unsteady
0.3	Steady	0.3	Transitional	0.3	Unsteady
0.5	Steady	0.5	Steady	0.5	Steady
0.6	Steady	0.6	Steady	0.6	Steady
0.8	Steady	0.8	Steady	0.8	Transitional
1.0	Steady	1.0	Steady	1.0	Transitional

a sphere boundary [96,99,97,98], and the aforementioned data will be used here for validation of our moving overlapping mesh method.

Results are presented for several different inner mesh angular velocities, using Reynolds numbers ($\mathrm{Re} = D^*U^*_\infty/\nu^*$) of 100, 250, and 300, which correspond with those reported in a computational study of Giacobello et al. [98]. These Reynolds numbers lie within three different regimes for uniform flow around a *stationary* sphere, of which a detailed investigation and description can be found in [100]. The flow around a stationary sphere at $\mathrm{Re} = 20 - 210$ is axisymmetric and steady, while a steady transition regime appears as Reynolds numbers increase beyond 210 and the flow becomes non-symmetric, until $\mathrm{Re} \approx 270$ where the flow becomes unsteady and vortex shedding begins to occur. When the rotation is added to the sphere, the flow patterns change depending on the angular velocity. The list of the calculated cases is presented in Table 4.

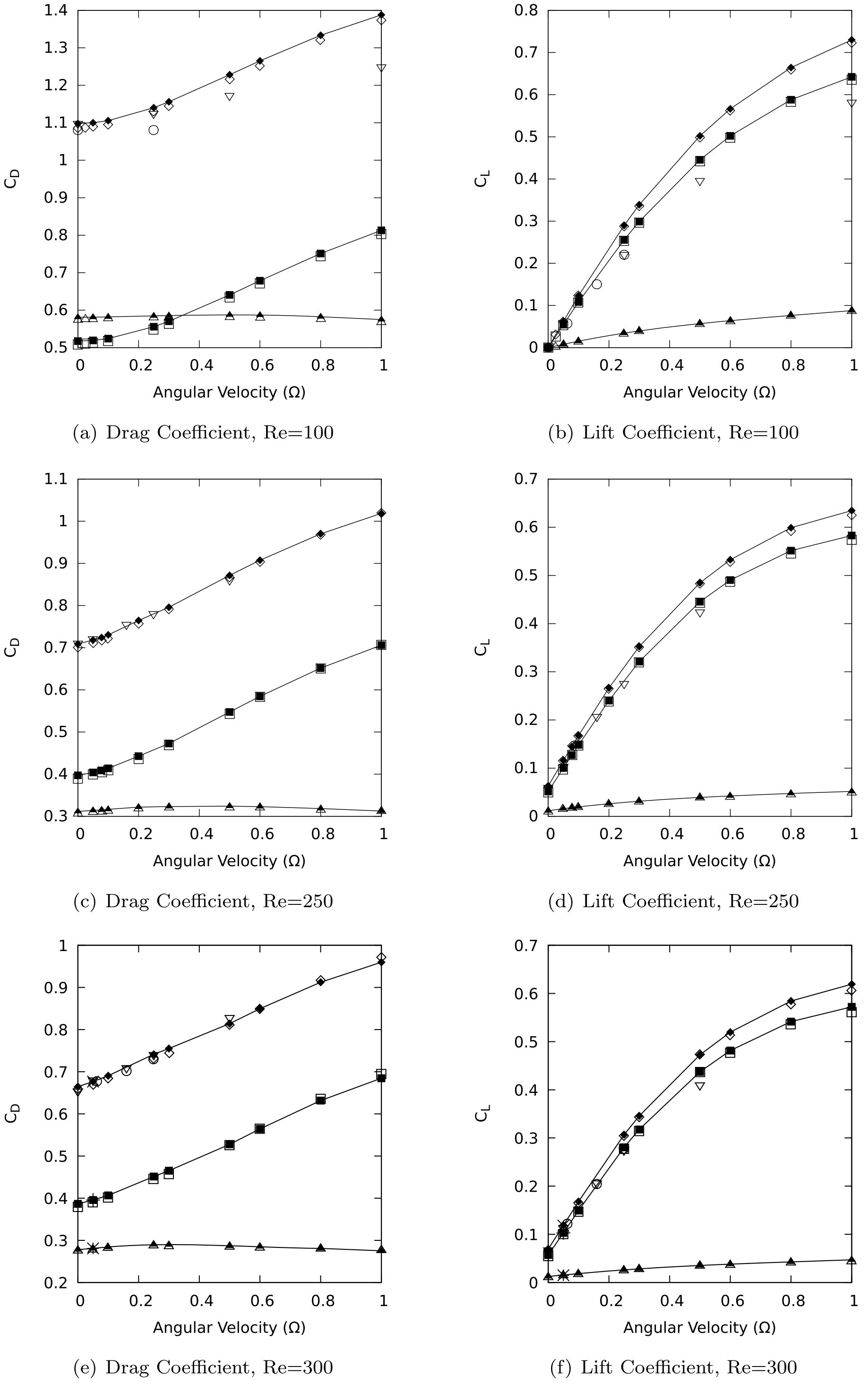
The drag and lift coefficients, determined by forces on the sphere: $C_D = F^*_D/(\frac{1}{8}\rho^*_f U^{*2}_\infty \pi D^{*2})$ and $C_L = F^*_L/(\frac{1}{8}\rho^*_f U^{*2}_\infty \pi D^{*2})$, where ρ^*_f is the fluid density, were used in comparison with results reported in [96–98] and can be seen in Fig. 17 where the non-dimensional angular velocity is given by $\Omega = \Omega^* D^*/(2U^*_\infty)$. Specific viscous and pressure contributions to the total force are reported for additional comparison. As can be seen from the figure, the forces calculated using our moving overlapping mesh method correlate well with the published results found in [96–98]. As with the oscillating cylinder case, one of the rotating sphere cases ($\mathrm{Re} = 300$, $\Omega = 0.05$) was also run with the third-order time stepping to address the stability of the scheme in this setup. The drag, lift and pressure forces calculated with the third-order scheme are indistinguishable from those of the second-order scheme, and the Strouhal number matches as well, as can be seen in Figs. 17 and 18 respectively.

For two of the higher Reynolds numbers, $\mathrm{Re} = 250$ and $\mathrm{Re} = 300$, Giacobello et al. [98] observed three different regimes depending on the angular velocity: steady regime when the vortex shedding is not observed throughout a duration of the simulations; transitional regime when the vortex shedding initially occurs and then dies out; and fully unsteady regime, with the vortex shedding persisting until the end of the simulations (the flow was always in a steady regime for $\mathrm{Re} = 100$ for the tested angular velocities). We noticed a similar behavior in our simulations, with corresponding regimes for our cases listed in Table 4.

For the Reynolds numbers $\mathrm{Re} = 250$ and $\mathrm{Re} = 300$, we calculated Strouhal numbers of the vortex shedding for the tested angular velocities and compared them with the data of [96–98] in Fig. 18. Following Giacobello et al. [98], for the transitional cases (for which the vortex shedding initially occurs and then dies out), we present initial and not final vortex shedding frequency (which would be zero). Zero frequency in this plot thus corresponds to steady cases. As can be seen, a comparison of Strouhal numbers for different angular velocities and Reynolds numbers (see Fig. 18) also shows excellent correlation with the published data. Visual comparison of trailing vortices between our results and those of Ref. [98] is performed in Fig. 19 for $\mathrm{Re} = 250$ and 300. Vortices are visualized using the λ_2-criteria [101]. As can be seen, the observed vortical structures reveal excellent resemblance with the figures reported in [98].

4.4. Computational scaling

Scaling analyses were performed on Stampede supercomputing resources in conjunction with a computational resources allocation through XSEDE. The test simulation for scaling studies consisted of a three-dimensional turbulent pipe flow with an exterior and interior overlapping meshes (a cross-section of each is shown with element boundaries in Fig. 20) so that a one-to-one comparison with the corresponding scaling analysis of the stationary overlapping method documented in [72] could be made. The interior and exterior meshes overlapped along the entire length of the pipe, and no-slip boundary conditions were enforced on the outer cylindrical boundary of the exterior mesh, representing the wall, while periodic boundary conditions were given at the pipe ends. The global problem contained a total of 51.2k elements (12.8k in interior mesh, 38.4k in exterior mesh), with the total number of grid points varied based upon the polynomial order used for solution approximations. The simulations were performed with Reynolds number of 5300 based on the pipe diameter. Note that the mesh resolution for this Reynolds number is much higher than is needed for a Direct Numerical Simulation of the



(a) Drag Coefficient, Re=100

(b) Lift Coefficient, Re=100

(c) Drag Coefficient, Re=250

(d) Lift Coefficient, Re=250

(e) Drag Coefficient, Re=300

(f) Lift Coefficient, Re=300

Fig. 17. Comparison of force coefficients at Re=100, 250, and 300. Closed symbols represent present data, and open symbols represent data reported in previous publications [96–98]. Closed upright triangles represent the viscous contribution, closed squares represent the pressure contribution, and closed diamonds denote the total force coefficient in the present data computed with BDF2/IEXT2. A cross symbol corresponds to $Re = 300, \Omega = 0.05$ case computed with BDF3/IEXT3. Open circles, open inverted triangles, and open diamonds represent data given for the total force coefficient in [96], [97], and [98] respectively. Open squares denote the pressure contribution, and open upright triangles the viscous contribution, as reported in [98]. The simulations with BDF2/IEXT2 were performed using 5th-order polynomial approximations, $\Delta t = 5 \times 10^{-3}$ and two iterations per timestep, while the simulation with BDF3/IEXT3 was performed using 5th-order polynomial approximations, $\Delta t = 5 \times 10^{-3}$ and five iterations per timestep.

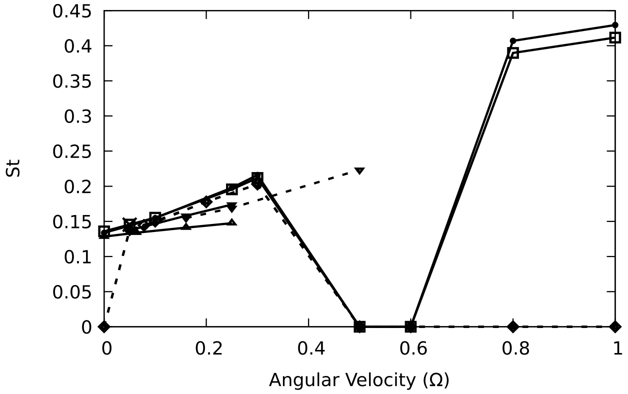
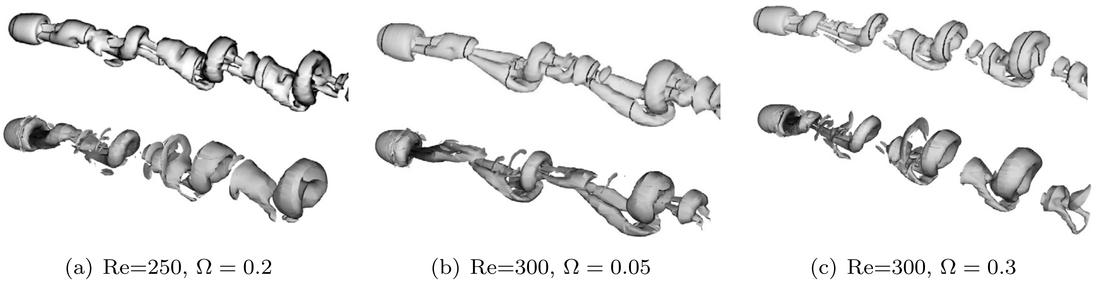


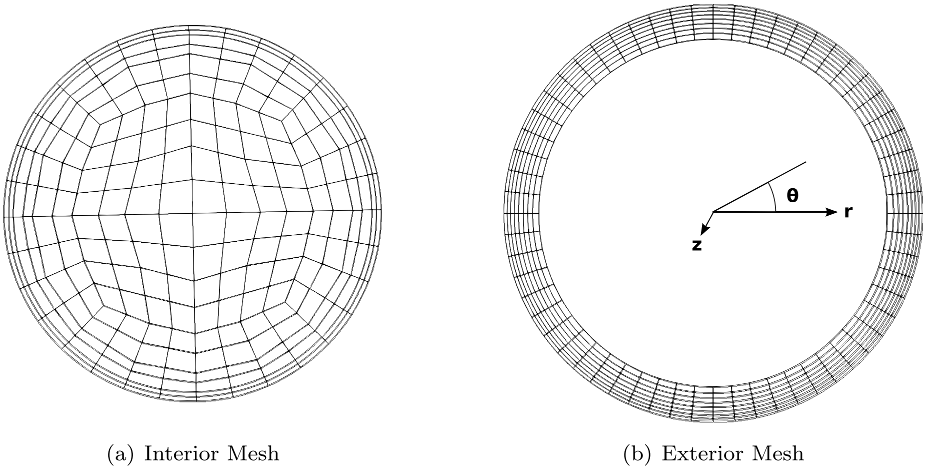
Fig. 18. Comparison of Strouhal numbers given by the present simulations and data presented in [96–98]. All data at $\mathrm{Re} = 250$ is given with dashed lines, while all data at $\mathrm{Re} = 300$ is given with solid lines computed with BDF2/IEXT2. A cross symbol corresponds to $Re = 300, \Omega = 0.05$ case computed with BDF3/IEXT3. Symbols represent data as follows: hollow diamonds - present data with Re=250, hollow squares - present data with $\mathrm{Re} = 300$, hollow upright triangles - Kurose and Komori [96] with $\mathrm{Re} = 300$, hollow inverted triangles - Niazmand and Renksizbulut [97] with $\mathrm{Re} = 250$, filled inverted triangles - Niazmand and Renksizbulut [97] with $\mathrm{Re} = 300$, hollow circles - Giacobello et al. [98] with $\mathrm{Re} = 250$, filled circles - Giacobello et al. [98] with $\mathrm{Re} = 300$. The simulations with BDF2/IEXT2 were performed using 5th-order polynomial approximations, $\Delta t = 5 \times 10^{-3}$ and two iterations per timestep, while the simulation with BDF3/IEXT3 was performed using 5th-order polynomial approximations, $\Delta t = 5 \times 10^{-3}$ and five iterations per timestep.


(a) Re=250, $\Omega = 0.2$

(b) Re=300, $\Omega = 0.05$

(c) Re=300, $\Omega = 0.3$

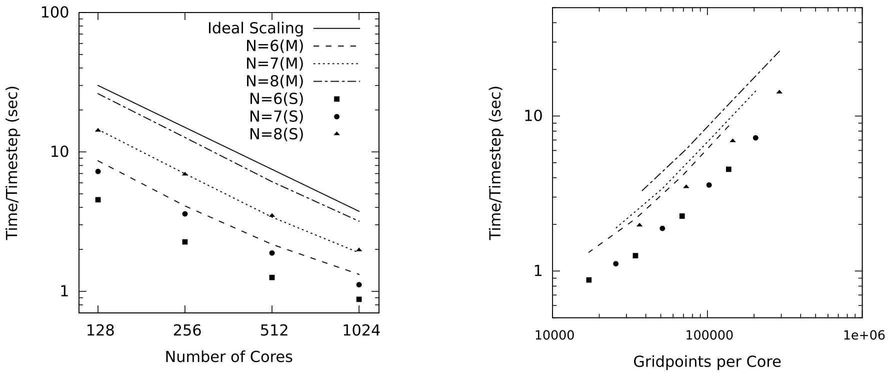
Fig. 19. Visual comparison of shed vortices as reported in [98] (top visualization in each subfigure) with present results (bottom visualization in each subfigure). Vortex iso-surfaces are presented using λ_2 criteria [101] at $\lambda_2 = -8 \times 10^{-4}$ (same for both sets of data). The presented data is for BDF2/IEXT2 scheme with 5th-order polynomial approximations, $\Delta t = 5 \times 10^{-3}$ and two iterations per timestep. Visualizations of the wake are displayed up to 16 diameters downstream.



(a) Interior Mesh

(b) Exterior Mesh

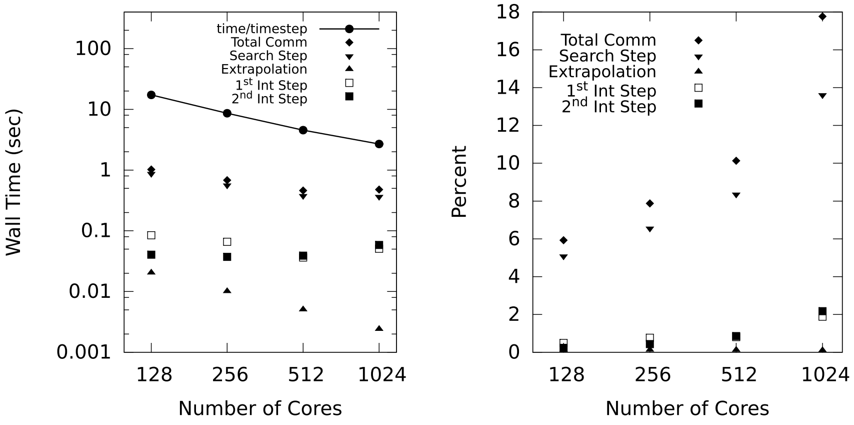
Fig. 20. Transverse cross-section of the grids for two-mesh pipe flow simulations used in scalability analyses. Element boundaries are shown. The displayed meshes extend 6 pipe diameters in the z-direction.



(a) Overlapping moving mesh (lines) simulation scaling compared with overlapping stationary mesh (points) scaling

(b) Scaling of moving and stationary overlapping mesh simulations with respect to the number of gridpoints per core. Line patterns correspond to those shown in Figure 21(a).

Fig. 21. Scaling of overlapping mesh simulations using 6th, 7th, and 8th order polynomial approximations. Meshes are identical for moving and stationary mesh cases. Parallel partitioning was utilized assigning cores to each sub-domain proportional to the corresponding mesh sizes. The timestepping scheme BFD2 with IEXT2 and two iterations per timestep was used for each simulation, with $\Delta t = 1 \times 10^{-3}$.



(a) Wall time used per timestep in different components handling inter-mesh communication.

(b) Percentage of wall time per timestep used in different components handling inter-mesh communication.

Fig. 22. Time spent in different components of the moving overlapping mesh methodology. Filled circles represent total time per timestep, filled diamonds - time spent in all components handling inter-mesh communication, filled inverted triangles - the search step, filled upright triangles - extrapolation, hollow squares - 1st interpolation step, filled squares - 2nd interpolation step. For computational parameters see Fig. 21.

pipe flow, and large element counts were used mainly for scalability analysis on large number of processors. The interior mesh was constrained to rotate in the moving mesh simulations with a rotational velocity of $\Omega \equiv \Omega^* D^*/U_b^* = 1$, where U_b^* is the bulk flow rate of fluid in the pipe, which, as in the test case of convecting eddies, represents a "virtual movement" and does not affect the flow. The current test case (including turbulent statistics) was validated in [72] for a fully developed turbulent flow with the same Reynolds number and zero interior mesh velocity. Turbulent statistics were not collected for the rotating mesh cases, that were used primarily for scalability studies.

The scaling analyses (Fig. 21) show near linear strong scaling for both stationary and moving overlapping simulations. The timing data for different polynomial orders collapses when scaled with respect to the number of gridpoints per core (Fig. 21 (b)) indicating robust parallel performance.

Fig. 22 shows a breakdown of the percentage of wall time per timestep spent in components of the moving overlapping mesh code that deal with inter-mesh communications. Notice that the percentage of wall time spent in the interpolation and extrapolation steps is very small while a much more sizable percentage of wall time per timestep is expended in the search step. The total percentage of time spent in inter-mesh communications is still significantly smaller than the time spent in computations.

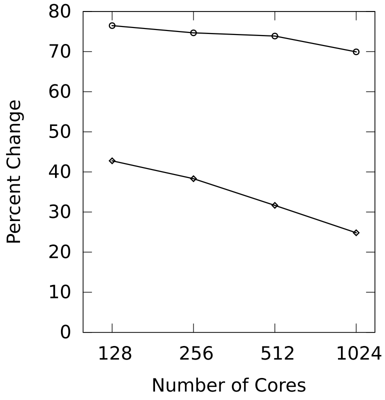


Fig. 23. Percent change in global computation time, comparing the present moving overlapping mesh methodology (hollow circles) with the traditional single mesh implementation, and also our stationary overlapping mesh methodology [72] (hollow diamonds) with the traditional single mesh implementation. For computational parameters see Fig. 21.

Fig. 23 illustrates the percent change in wall time per timestep going from the traditional single mesh solver to the moving overlapping mesh solver (of the same total element count), and from our stationary overlapping mesh solver [72] to the moving overlapping mesh solver. While more wall time per timestep is understandably spent in moving overlapping mesh simulations, the results, again, confirm that the overall scalability of the global code is maintained.

5. Conclusion

Knowing the characteristics of fluid flow around moving solid bodies is important in many real world problems. Traditionally, solvers that can handle the flow around moving bodies, including body conforming and fixed mesh methods, achieve lower orders of convergence. Our developed moving overlapping mesh methodology maintains high orders of global accuracy in space and time, by utilizing a spectral-element computational code combined with a spectral spatial interpolation and a high-order temporal extrapolation at the interface nodes. Simulations of convecting eddies on moving meshes, where exact solutions are known, showed expected rates of spatial and temporal convergence consistent with the underlying accuracy of the global numerical scheme. A self convergence test with the two-dimensional rotating solid cylinder also exhibited an expected exponential spatial convergence in the presence of solid moving boundaries. Solving for the incompressible flow around a two-dimensional oscillating cylinder produced aerodynamic forces in line with published computational data, and showed lock-in behavior of vortex shedding corresponding with Koopman's experimental findings [91]. Extension of the method to three-dimensional domains has shown a robust modeling of the vortices shed from a rotating sphere and produced a quantitative data that are consistent with the results found in literature. The presented incompressible flow solver with moving overlapping meshes achieves near linear scaling for large processor counts. While a theoretical stability analysis of the overlapping grid interpolation on the moving domains will be addressed in our future work, the test cases show a stabilizing behavior of the subdomain iterations, similar to the stationary overlapping domain method [72,78]. In the present work, only two iterations were needed to attain stable solutions with the second-order temporal scheme for the cases tested, and between four and seven iterations for the third-order temporal scheme. While stability when using the explicit interface extrapolation scheme is enhanced with a small number of iterations, this is vastly different from implicit coupling methods that require iteration counts in the order of hundreds solely to achieve a desired order of temporal accuracy. The current method lends itself as a reliable tool for performing high-fidelity simulations of moving bodies in the presence of unsteady, transitional and turbulent environments (see, for example, Ref. [102], where Direct Numerical Simulations of a cyclically pitching airfoil in a wake of an upstream cylinder were performed with the current methodology for the Reynolds number Re_c=44,000 based on the airfoil chord length). A limitation of the present implementation of the methodology is the inability to add or remove elements from the subdomains on the fly, so that the "hole-cutting" is not currently a feature. This inflicts some restrictions onto a generality of the relative subdomain motions that can be considered, which can be further relaxed when hole-cutting is implemented, and is the subject of our future work.

Acknowledgements

This work was supported by the NSF grant CMMI #1250124. 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.

References

[1] C.S. Peskin, Flow Patterns Around Heart Valves: A Digital Computer Method for Solving the Equations of Motion, PhD thesis, Albert Einstein College of Medicine, Bronx, NY, 1972.

[2] C.S. Peskin, The fluid dynamics of heart valves: experimental, theoretical, and computational methods, Annu. Rev. Fluid Mech. 14 (1) (1982) 235–259.

[3] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.

[4] S. Haeri, J.S. Shrimpton, On the application of immersed boundary, fictitious domain and body-conformal mesh methods to many particle multiphase flows, Int. J. Multiph. Flow 40 (2012) 38–55.

[5] R. Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 112 (1) (1994) 133–148.

[6] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2) (2000) 705–719.

[7] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2) (2007) 2118–2137.

[8] A. Mark, B.G.M. van Wachem, Derivation and validation of a novel implicit second-order accurate immersed boundary method, J. Comput. Phys. 227 (13) (2008) 6660–6680.

[9] S.V. Apte, M. Martin, N.A. Patankar, A numerical method for fully resolved simulation (FRS) of rigid particle–flow interactions in complex flows, J. Comput. Phys. 228 (8) (2009) 2712–2738.

[10] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. 153 (2) (1999) 509–534.

[11] T.E. Tezduyar, J. Liou, M. Behr, A new strategy for finite element computations involving moving boundaries and interfaces–the DSD/ST procedure: I. The concept and the preliminary numerical tests, Comput. Methods Appl. Mech. Eng. 94 (3) (1992) 339–351.

[12] R. Torii, M. Oshima, T. Kobayashi, K. Takagi, T.E. Tezduyar, Computer modeling of cardiovascular fluid–structure interactions with the deforming-spatial-domain/stabilized space–time formulation, Comput. Methods Appl. Mech. Eng. 195 (13) (2006) 1885–1895.

[13] W.F. Noh, CEL: A Time-Dependent, Two-Space-Dimensional, Coupled Eulerian–Lagrange Code, Technical report, Lawrence Radiation Lab., Univ. of California, Livermore, 1963.

[14] R.M. Franck, R.B. Lazarus, Mixed Eulerian–Lagrangian method, Methods Comput. Phys. 3 (1964) 47–67.

[15] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.

[16] W.E. Pracht, Calculating three-dimensional fluid flows at all speeds with an Eulerian–Lagrangian computing mesh, J. Comput. Phys. 17 (2) (1975) 132–159.

[17] J. Donéa, P. Fasoli-Stella, S. Giuliani, Lagrangian and Eulerian finite element techniques for transient fluid-structure interaction problems, in: Structural Mechanics in Reactor Technology, 1977.

[18] T. Belytschko, J.M. Kennedy, D.F. Schoeberle, Quasi-Eulerian finite element formulation for fluid-structure interaction, J. Press. Vessel Technol. 102 (1) (1980) 62–69.

[19] T.J.R. Hughes, W.K. Liu, T.K. Zimmermann, Lagrangian–Eulerian finite element formulation for incompressible viscous flows, Comput. Methods Appl. Mech. Eng. 29 (3) (1981) 329–349.

[20] B. Ramaswamy, M. Kawahara, Arbitrary Lagrangian–Eulerian finite element method for unsteady, convective, incompressible viscous free surface fluid flow, Int. J. Numer. Methods Fluids 7 (10) (1987) 1053–1075.

[21] L.-W. Ho, A.T. Patera, A Legendre spectral element method for simulation of unsteady incompressible viscous free-surface flows, Comput. Methods Appl. Mech. Eng. 80 (1) (1990) 355–366.

[22] L.-W. Ho, A.T. Patera, Variational formulation of three-dimensional viscous free-surface flows: natural imposition of surface tension boundary conditions, Int. J. Numer. Methods Fluids 13 (6) (1991) 691–698.

[23] J.Y. Murthy, S.R. Mathur, D. Choudhury, CFD simulation of flows in stirred tank reactors using a sliding mesh technique, in: Institution of Chemical Engineers Symposium Series, vol. 136, Hemsphere Publishing Corporation, 1994, p. 341.

[24] A. Bakker, R.D. LaRoche, M. Wang, R.V. Calabrese, Sliding mesh simulation of laminar flow in stirred reactors, Chem. Eng. Res. Des. 75 (1) (1997) 42–44.

[25] K. Ng, N.J. Fentiman, K.C. Lee, M. Yianneskis, Assessment of sliding mesh CFD predictions and LDA measurements of the flow in a tank stirred by a Rushton impeller, Chem. Eng. Res. Des. 76 (6) (1998) 737–747.

[26] J.M. Bujalski, Z. Jaworski, W. Bujalski, A.W. Nienow, The influence of the addition position of a tracer on CFD simulated mixing times in a vessel agitated by a Rushton turbine, Chem. Eng. Res. Des. 80 (8) (2002) 824–831.

[27] C.A. Rivera, M. Heniche, F. Bertrand, R. Glowinski, P.A. Tanguy, A parallel finite element sliding mesh technique for the simulation of viscous flows in agitated tanks, Int. J. Numer. Methods Fluids 69 (3) (2012) 653–670.

[28] Y.M. Park, O.J. Kwon, Simulation of unsteady rotor flow field using unstructured adaptive sliding meshes, J. Am. Helicopter Soc. 49 (4) (2004) 391–400.

[29] G. Barakos, R. Steijl, K. Badcock, A. Brocklehurst, Development of CFD capability for full helicopter engineering analysis, in: 31st European Rotorcraft Forum, Florence, Italy, Sept, 2005.

[30] R. Steijl, G. Barakos, Sliding mesh algorithm for CFD analysis of helicopter rotor–fuselage aerodynamics, Int. J. Numer. Methods Fluids 58 (5) (2008) 527–549.

[31] A. Buffa, Y. Maday, F. Rapetti, A sliding mesh-mortar method for a two dimensional eddy currents model of electric engines, Modél. Math. Anal. Numér. 35 (2) (2001) 191–228.

[32] R.L. Meakin, Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulations, AIAA Paper, vol. 3350, 1993, p. 1993.

[33] 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.

[34] H. Pomin, S. Wagner, Aeroelastic analysis of helicopter rotor blades on deformable chimera grids, J. Aircr. 41 (3) (2004) 577–584.

[35] M. Potsdam, H. Yeo, W. Johnson, Rotor airloads prediction using loose aerodynamic/structural coupling, J. Aircr. 43 (3) (2006) 732–742.

[36] 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.

[37] D. Chandar, J. Sitaraman, D. Mavriplis, Dynamic overset grid computations for CFD applications on graphics processing units, in: Proceedings of the Seventh International Conference on Computational Fluid Dynamics, 2012.

[38] A.W. Vreman, A staggered overset grid method for resolved simulation of incompressible flow around moving spheres, J. Comput. Phys. 333 (2017) 269–296.

[39] J.W. Banks, W.D. Henshaw, D.W. Schwendeman, Deforming composite grids for solving fluid structure problems, J. Comput. Phys. 231 (9) (2012) 3518–3547.

[40] C. Kiris, D. Kwak, S. Rogers, I.D. Chang, Computational approach for probing the flow through artificial heart devices, J. Biomech. Eng. 119 (4) (1997) 452–460.

[41] L. Ge, H.-L. Leo, F. Sotiropoulos, A.P. Yoganathan, Flow in a mechanical bileaflet heart valve at laminar and near-peak systole flow rates: CFD simulations and experiments, J. Biomech. Eng. 127 (5) (2005) 782–797.

[42] 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.
[43] R.L. Meakin, Computations of the unsteady flow about a generic wing/pylon/finned-store configuration, AIAA Paper 4568 (1992) 1992.
[44] N.C. Prewitt, D.M. Belk, W. Shyy, Parallel computing of overset grids for aerodynamic problems with moving objects, Prog. Aerosp. Sci. 36 (2) (2000) 117–172.
[45] R.W. Noack, D.A. Boger, R.F. Kunz, P.M. Carrica, Suggar++: an improved general overset grid assembly capability, in: Proceedings of the 47th AIAA Aerospace Science and Exhibit, 2009, pp. 22–25.
[46] P.M. Carrica, R.V. Wilson, R.W. Noack, F. Stern, Ship motions using single-phase level set with dynamic overset grids, Comput. Fluids 36 (9) (2007) 1415–1433.
[47] 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.
[48] G. Chesshire, W.D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comput. Phys. 90 (1) (1990) 1–64.
[49] R.L. Meakin, Composite overset structured grids, in: Handbook of Grid Generation, 1999, pp. 1–20.
[50] 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.
[51] Y. Peet, S. Lele, Computational framework for coupling compressible and low Mach number codes, AIAA J. 46 (8) (2008) 1990–2001.
[52] T.F. Chan, T.P. Mathew, Domain decomposition algorithms, Acta Numer. 3 (1994) 61–143.
[53] B.F. Smith, P. Bjorstad, W. Gropp, Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, 1996.
[54] 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.
[55] H.A. Schwarz, Ueber einige abbildungsaufgaben, J. Reine Angew. Math. 70 (1869) 105–120.
[56] P.-L. Lions, On the Schwarz alternating method. I, in: First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris, France, SIAM, Philadelphia, 1988, pp. 1–42.
[57] X.-C. Cai, Additive Schwarz algorithms for parabolic convection diffusion equations, Numer. Math. 60 (1991) 41–61.
[58] X.-C. Cai, Multiplicative Schwarz methods for parabolic problems, SIAM J. Sci. Comput. 15 (3) (1994) 587–603.
[59] G. Starius, On composite mesh difference methods for hyperbolic differential equations, Numer. Math. 35 (3) (1980) 241–255.
[60] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Number CMCS-BOOK-2009-019, Oxford University Press, 1999.
[61] P. Bjorstad, W. Gropp, Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, 2004.
[62] A. Toselli, O. Widlund, Domain Decomposition Methods: Algorithms and Theory, vol. 3, Springer, 2005.
[63] J.L. Steger, F.C. Dougherty, J. Benek, A Chimera grid scheme, in: Advances in Grid Generation, ASME Fluids Engineering Conference, Houston, TX, 1983.
[64] J.A. Benek, P.G. Buning, J.L. Steger, A 3-D Chimera grid embedding technique, in: Proceedings 7th AIAA Computational Physics Conference, Cincinnati, OH, 1985 (AIAA 1985-1523).
[65] 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.
[66] W. Henshaw, A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids, J. Comput. Phys. 113 (1) (1994) 13–25.
[67] 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.
[68] P. Morgan, M. Visbal, D. Rizzetta, A parallel overset grid high-order flow solver for large eddy simulation, J. Sci. Comput. 19 (2) (2016) 165–200.
[69] S.E. Sherer, J.N. Scott, High-order compact finite-difference methods on general overset grids, J. Comput. Phys. 210 (2005) 459–496.
[70] M.J. Brazell, J. Sitaraman, D.J. Mavriplis, An overset mesh approach for 3D mixed element high order discretizations, J. Comput. Phys. 322 (2016) 33–51.
[71] P.M. Carrica, J. Huang, R. Noack, D. Kaushik, B. Smith, F. Stern, Large-scale DES computations of the forward speed diffraction and pitch and heave problems for a surface combatant, Comput. Fluids 39 (7) (2010) 1095–1111.
[72] B.E. Merrill, Y.T. Peet, P.F. Fischer, J.W. Lottes, A spectrally accurate method for overlapping grid solution of incompressible Navier–Stokes equations, J. Comput. Phys. 307 (2016) 60–93.
[73] P.F. Fischer, J.W. Lottes, S.G. Kerkemeier, Nek5000 web page, http://nek5000.mcs.anl.gov, 2008.
[74] M.O. Deville, P.F. Fischer, E.H. Mund, High-Order Methods for Incompressible Fluid Flow, vol. 9, Cambridge University Press, 2002.
[75] J. Donea, A. Huerta, J.-Ph. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian–Eulerian methods, in: Encyclopedia of Computational Mechanics, 2004.
[76] C. Truesdell, R. Toupin, The Classical Field Theories, Springer, 1960.
[77] L.W. Ho, A Legendre Spectral Element Method for Simulation of Incompressible Unsteady Viscous Free-Surface Flows, PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, 1989.
[78] Y. Peet, P.F. Fischer, Stability analysis of interface temporal discretization in grid overlapping methods, SIAM J. Numer. Anal. 50 (6) (2012) 3375–3401.
[79] P.F. Fischer, Time-dependent PDEs, http://fischerp.cs.illinois.edu/cs598/docs/notes_time1b.pdf, 2017.
[80] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 97 (2) (1991) 414–443.
[81] L. Formaggia, F. Nobile, A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements, East-West J. Numer. Math. 7 (2) (1999) 105–131.
[82] L. Formaggia, F. Nobile, A stability analysis of second-order time accurate schemes for ALE-FEM, Comput. Methods Appl. Mech. Eng. 193 (2004) 4097–4116.
[83] J. Van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Sci. Stat. Comput. 7 (1986) 870–891.
[84] 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).
[85] P. Fischer, J. Lottes, D. Pointer, A. Siegel, Petascale algorithms for reactor hydrodynamics, J. Phys. Conf. Ser. 125 (2008) 012076.
[86] N.A. Petterson, An algorithm for assembling overlapping grid systems, SIAM J. Sci. Comput. 20 (6) (1999) 1995–2022.
[87] O. Walsh, Eddy solutions of the Navier–Stokes equations, in: The Navier-Stokes Equations II - Theory and Numerical Methods, Springer, 1992, pp. 306–309.
[88] 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.
[89] Y. Peet, P.F. Fischer, Legendre spectral element method with nearly incompressible materials, Eur. J. Mech. A, Solids 44 (2013) 91–103.
[90] R.E.D. Bishop, A.Y. Hassan, The lift and drag forces on a circular cylinder oscillating in a flowing fluid, Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci. 277 (1368) (1964) 51–75.

[91] G.H. Koopmann, The vortex wakes of vibrating cylinders at low Reynolds numbers, J. Fluid Mech. 28 (3) (1967) 501–512.
[92] C.H.K. Williamson, A. Roshko, Vortex formation in the wake of an oscillating cylinder, J. Fluids Struct. 2 (4) (1988) 355–381.
[93] C.H.K. Williamson, Vortex dynamics in the cylinder wake, Annu. Rev. Fluid Mech. 28 (1) (1996) 477–539.
[94] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys. 174 (1) (2001) 345–380.
[95] H.M. Blackburn, R.D. Henderson, A study of two-dimensional flow past an oscillating cylinder, J. Fluid Mech. 385 (1999) 255–286.
[96] R. Kurose, S. Komori, Drag and lift forces on a rotating sphere in a linear shear flow, J. Fluid Mech. 384 (1999) 183–206.
[97] H. Niazmand, M. Renksizbulut, Surface effects on transient three-dimensional flows around rotating spheres at moderate Reynolds numbers, Comput. Fluids 32 (10) (2003) 1405–1433.
[98] M. Giacobello, A. Ooi, S. Balachandar, Wake structure of a transversely rotating sphere at moderate Reynolds numbers, J. Fluid Mech. 621 (2009) 103–130.
[99] D. Kim, H. Choi, Laminar flow past a sphere rotating in the streamwise direction, J. Fluid Mech. 461 (2002) 365–386.
[100] T.A. Johnson, V.C. Patel, Flow past a sphere up to a Reynolds number of 300, J. Fluid Mech. 378 (1999) 19–70.
[101] J. Jeong, F. Hussain, On the identification of a vortex, J. Fluid Mech. 285 (1995) 69–94.
[102] B.E. Merrill, Y.T. Peet, Effect of impinging wake turbulence on the dynamic stall of a pitching airfoil, AIAA J. 55 (12) (2017) 4094–4112.