INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 0000; **00**:1–25
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

High-Order Methods for Low Reynolds Number Flows around Moving Obstacles Based on Universal Meshes

Evan S. Gawlik[1], Hardik Kabaria[2] and Adrian J. Lew[2,1*]

[1]*Computational and Mathematical Engineering, Stanford University*
[2]*Mechanical Engineering, Stanford University*

SUMMARY

We propose a family of methods for simulating two-dimensional incompressible, low Reynolds number flow around a moving obstacle whose motion is prescribed. The methods make use of a universal mesh: a fixed background mesh that adapts to the geometry of the immersed obstacle at all times by adjusting a few elements in the neighborhood of the obstacle's boundary. The resulting mesh provides a conforming triangulation of the fluid domain over which discretizations of any desired order of accuracy in space and time can be constructed using standard finite element spaces together with off-the-shelf time integrators. We demonstrate the approach by using Taylor-Hood elements to approximate the fluid velocity and pressure. To integrate in time, we consider implicit Runge-Kutta schemes as well as a fractional step scheme. We illustrate the methods and study their convergence numerically via examples that involve flow around obstacles that undergo prescribed deformations. Copyright © 0000 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Fluid-structure interaction; viscous flow; moving interface

1. INTRODUCTION

One of the key challenges in numerical simulations of fluid flow around moving obstacles is the discretization of an evolving domain, namely, the domain occupied by the fluid. Commonly, this challenge is addressed using one of two tools: a deforming mesh, which deforms in concert with the moving fluid domain, or a fixed mesh, which triangulates or quadrangulates a larger domain in which the moving boundary is immersed for all times. In contrast, this paper presents a family of methods for simulating two-dimensional incompressible, low Reynolds number flow around moving obstacles with prescribed evolution using a *universal mesh*: a background triangulation that contains the fluid domain for all times *and conforms to its geometry* at all times by perturbing a small number of nodes in the neighborhood of the immersed fluid boundary.

The merits of this strategy are made most apparent when one considers the challenges associated with the construction of numerical methods for fluid flow around obstacles that are simultaneously robust and accurate to high order in space and time. The latter goal is particularly elusive for problems on moving domains, since errors in the discretization of the domain's geometry (and in the discretization, if any, of its temporal evolution) can dictate the order of a method. This consideration renders deforming-mesh methods attractive, especially if implemented using curved elements along the boundary. In this light, it is perhaps surprising that many examples of deforming-mesh methods in the literature, with a few noteworthy exceptions [1–10], are often restricted to at

*Correspondence to: Adrian J. Lew, Mechanical Engineering, Stanford University, Stanford, CA, USA
E-mail: lewa@stanford.edu

most second-order accuracy [11–19]. Moreover, robustness of the mesh motion poses a challenge to deforming-mesh methods [20, 21], which often prescribe a motion for the mesh by solving systems of equations (such as those of linear elasticity) for the positions of mesh nodes [22–26]. Regardless of a deforming-mesh method's mesh motion strategy, sufficiently large domain deformations may lead to element distortions (or, in more severe cases, element inversions) that are detrimental both to the accuracy of the spatial discretization and to the conditioning of the discrete governing equations [27]. These considerations can ultimately mandate that the domain be remeshed from scratch at various instants during a simulation [22, 28–30].

Fixed-mesh methods circumvent the difficulty of designing a robust mesh motion at the expense of geometric conformity. As a consequence, fixed-mesh methods require special care in order to account for the disagreement between the immersed obstacle boundary and element interfaces. A variety of techniques aim to deal with this discrepancy, including adaptive refinement in the neighborhood of the immersed boundary [31–33], cutting elements [34–38], enriching finite element spaces [39, 40], cutting elements and enriching finite element spaces [41–43], Nitsche-inspired methods [44, 45], smearing the interface [46, 47], modifying finite-difference stencils near the boundary [33, 48–50], and introducing surrogate forcing terms in lieu of the boundary conditions [51, 52]. Integration in time poses an additional challenge for fixed-mesh methods, since nodes of the background mesh may occupy differing states (fluid vs. solid) over the course of a single time step. This peculiarity is known to introduce numerical artifacts such as spurious oscillations in the pressure field for some fixed-mesh methods [37, 53, 54]. Furthermore, even if a given spatial discretization is known to deliver high-order spatial accuracy for steady flows around an embedded obstacle, its incorporation into a numerical method for unsteady flow around *moving* obstacles with high spatial and temporal accuracy is arguably a nontrivial task. These observations help to explain why many fixed mesh methods, again with a few notable exceptions [55, 56], are often restricted to first- or second-order accuracy [50, 52, 57–60].

The framework presented in this paper distinguishes itself from the preceding approaches by exhibiting the following features simultaneously. First, a universal mesh delivers a conforming representation of the evolving fluid domain at all times. This conforming mesh is obtained by perturbing the nodes of a background mesh using a mapping which supplies not only an adaptation of the background mesh, but also a mesh motion over short time intervals suitable for constructing high-order discretizations of the governing equations. Second, the mesh motion strategy is robust, in the sense that large domain deformations pose no threat to the quality of the conforming mesh, being at all times derived from a small perturbation of the background mesh. Third, our approach provides a systematic framework for constructing methods of a desired order of accuracy in space and in time for low Reynolds number flows, simply by discretizing in space with a finite element space of the appropriate order and choosing a time integrator of the appropriate order. We demonstrate this by combining high-order Taylor-Hood elements with high-order implicit Runge Kutta schemes. Finally, the framework is algorithmically simple. In its basic form, the alteration of the background mesh requires adjustments to nodal coordinates only, not the mesh's connectivity, and the nodal motions are independent and explicitly defined.

To simplify the presentation and to emphasize the main contributions of the present work, we restrict our attention to problems for which the flow has a low Reynolds number, the obstacle boundary is smooth (C^2-regular), and the obstacle motion is prescribed. Needless to say, higher Reynolds number flows pose additional challenges (the need for high resolution in boundary layers and for stabilization of convective terms in the spatial discretization) that warrant enhancements to the present strategy to ensure its viability. Likewise, the design of universal meshes for domains with lower regularity, such as domains with corners, remains an area of active research. Unprescribed obstacle motions would, of course, introduce additional complexity into the framework, but only in the sense that additional unknowns would need to be solved for concurrently with the fluid variables.

Organization. This paper is organized as follows. In Section 2, we recall the governing equations for incompressible, viscous flow around a moving obstacle with prescribed evolution, and we recast the equations in weak form. In Section 3, we propose a discretization of the aforementioned

Prepared using nmeauth.cls
DOI: 10.1002/nme

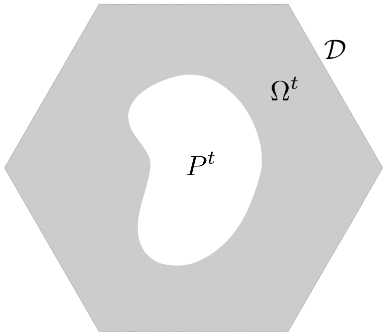


Figure 1. Fluid domain $\Omega^t = \mathcal{D} \setminus P^t$.

equations using a universal mesh in conjunction with Taylor-Hood finite elements [61]. To integrate in time, we propose the use of implicit Runge-Kutta schemes as well as a fractional step scheme. In Section 4, we apply the proposed methods to simulate flow around various obstacles with prescribed evolution: a rotating ellipse, an oscillating disk, and a rotating stirrer. We study numerically the convergence orders of the methods in the context of the rotating ellipse, where an analytical solution is readily manufactured. We close with some concluding remarks in Section 5.

2. PROBLEM

We study incompressible, viscous fluid flow around a moving obstacle immersed in a domain $\mathcal{D} \subset \mathbb{R}^2$. We denote by $P^t \subset \mathcal{D}$ the domain occupied by the obstacle at time t and by $\Omega^t = \mathcal{D} \setminus P^t$ the domain occupied by the fluid. Taking the fluid density to be everywhere unity, the governing equations for the velocity u and pressure p read

$$\frac{\partial u}{\partial t} + u \cdot \nabla_x u - \nu \Delta_x u = -\nabla_x p \qquad \text{in } \mathcal{D} \setminus P^t \tag{1}$$

$$\nabla_x \cdot u = 0 \qquad \text{in } \mathcal{D} \setminus P^t, \tag{2}$$

where $\nu > 0$ is the kinematic viscosity of the fluid. On the interface between the obstacle and the fluid, the no-slip condition holds:

$$u(x,t) = v_P(x,t), \quad x \in \partial P^t \tag{3}$$

where $v_P(x,t)$ is the prescribed velocity of the obstacle at $x \in \partial P^t$. On the remainder of the fluid boundary $\partial\mathcal{D}$, and depending on the example under consideration, we impose either the natural boundary conditions

$$pn - \nu\left(\nabla_x u + (\nabla_x u)^T\right) n = 0 \qquad \text{on } \partial\mathcal{D} \tag{4}$$

or the no-slip condition

$$u = 0 \qquad \text{on } \partial\mathcal{D}, \tag{5}$$

and in this last case the pressure field is defined up to a constant.

Weak Formulation. For later use, it is convenient to record a weak formulation of (1-2). Let us introduce two collections of function spaces, one for each choice of boundary conditions discussed above. When the boundary conditions are given by (3-4), we denote

$$\begin{aligned}
\mathring{\mathcal{V}}^t &= \{u \in H^1(\Omega^t)^2 \mid u = 0 \text{ on } \partial P^t\} \\
\mathcal{V}^t &= \{u \in H^1(\Omega^t)^2 \mid u = v_P(\cdot,t) \text{ on } \partial P^t\} \\
\mathcal{Q}^t &= L^2(\Omega^t).
\end{aligned}$$

DOI: 10.1002/nme

When the boundary conditions are given by (3) and (5), we denote

$$\begin{aligned}\mathring{\mathcal{V}}^t &= H_0^1(\Omega^t)^2\\ \mathcal{V}^t &= \{u \in H^1(\Omega^t)^2 \mid u = v_P(\cdot,t) \text{ on } \partial P^t,\, u = 0 \text{ on } \partial\mathcal{D}\}\\ \mathcal{Q}^t &= L^2(\Omega^t)/\mathbb{R}.\end{aligned}$$

In either of these two settings, a weak formulation of (1-2) reads: Find $u(\cdot,t) \in \mathcal{V}^t$ and $p(\cdot,t) \in \mathcal{Q}^t$ such that

$$m^t(\dot{u},w) + a^t(u,w) + c^t(w,p) = 0 \quad \forall w \in \mathring{\mathcal{V}}^t \tag{6}$$

$$c^t(u,q) = 0 \quad \forall q \in \mathcal{Q}^t \tag{7}$$

for every $t \in (0,T]$, where the meanings of $\mathring{\mathcal{V}}^t$, $\mathcal{V}^t$, and $\mathcal{Q}^t$ depend upon the boundary conditions under consideration, and

$$\begin{aligned}m^t(u,w) &= \int_{\Omega^t} u \cdot w \, dx\\ a^t(u,w) &= \nu \int_{\Omega^t} \left(\nabla_x u + (\nabla_x u)^T\right) : \nabla_x w \, dx + \int_{\Omega^t} (u \cdot \nabla_x u) \cdot w \, dx\\ c^t(u,p) &= -\int_{\Omega^t} (\nabla \cdot u) p \, dx.\end{aligned}$$

The well-posedness of this system in the case of a fixed domain without advection is proven in [62].

3. METHOD

In this section, we propose a discretization of (1-2) that is based upon the use of a universal mesh. Our approach follows that of [63], where a framework for constructing numerical methods for moving-boundary problems using universal meshes is introduced in the context of a parabolic model problem.

The discretization proceeds in several steps: (1) partitioning the temporal axis into short time intervals $\bigcup_{n=1}^N (t^{n-1},t^n] = (0,T]$; (2) constructing a conforming mesh $\mathcal{S}_h(t)$ for Ω^t, $t \in (t^{n-1},t^n]$, over each short time interval by adapting the universal mesh; (3) performing a Galerkin projection of the governing equations onto a finite element space associated with $\mathcal{S}_h(t)$ over each short time interval; and (4) choosing a time integrator to numerically integrate the resulting system of ODE's over $(t^{n-1},t^n]$ for each n. In the last step, the initial condition for numerical integration over $(t^{n-1},t^n]$ will come from projecting the discrete solution at time $t = t^{n-1}$ onto the finite element space associated with the triangulation $\mathcal{S}_h(t_+^{n-1})$, which generally differs from $\mathcal{S}_h(t^{n-1})$.

A distinctive feature of our discretization is the manner in which the conforming triangulation $\mathcal{S}_h(t)$ of Ω^t is constructed. As illustrated in Fig. 2, our method constructs $\mathcal{S}_h(t)$ by immersing Ω^t in a background mesh $\mathcal{T}_h$ (the *universal mesh*), identifying a subtriangulation of $\mathcal{T}_h$ that approximates the immersed domain, and adjusting a few elements so that it conforms exactly to Ω^t. This approach differs markedly from classical deforming-mesh methods, where the task of triangulating Ω^t is often handled by solving global systems of equations (such as those of linear elasticity) for nodal positions. We describe the construction of $\mathcal{S}_h(t)$ using a universal mesh in the following subsection and provide greater detail in Appendix A, which summarizes the presentation of [63].

The conditions under which a given background triangulation $\mathcal{T}_h$ can be so adjusted to conform to a family of domains Ω^t, $t \in [0,T]$, are laid forth in [64, 65] and expanded in [66]. Briefly, the procedure is guaranteed to succeed if:

(3.i) Ω^t is C^2-regular for every t.

(3.ii) $\mathcal{T}_h$ is sufficiently refined in a neighborhood of $\partial\Omega^t$ for every t.

DOI: 10.1002/nme

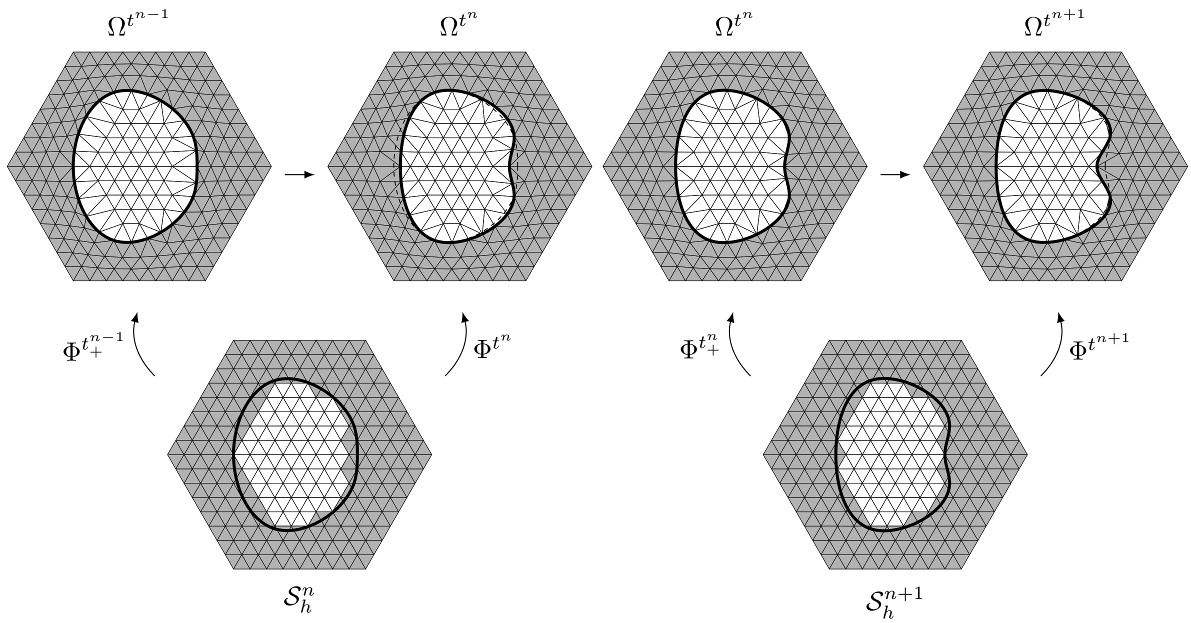


Figure 2. Illustration of the manner in which a universal mesh provides a conforming triangulation of an immersed domain Ω^t for all times t. Over a short time interval $(t^{n-1}, t^n]$, an approximating subtriangulation $\mathcal{S}_h^n$ is identified and adapted to the immersed domain using a map $\Phi^t : \mathcal{S}_h^n \to \Omega^t$, $t \in (t^{n-1}, t^n]$. Over the next short time interval $(t^n, t^{n+1}]$, a new subtriangulation $\mathcal{S}_h^{n+1}$ is identified and adapted to the immersed domain using a map $\Phi^t : \mathcal{S}_h^{n+1} \to \Omega^t$, $t \in (t^n, t^{n+1}]$. For visual clarity, the boundary of $\Omega^{t^{n-1}}$ has been juxtaposed in dashed lines onto the conforming mesh $\Phi^{t^n}(\mathcal{S}_h^n)$ for Ω^{t^n}. Likewise, the boundary of Ω^{t^n} has been juxtaposed in dashed lines onto the conforming mesh $\Phi^{t^{n+1}}(\mathcal{S}_h^{n+1})$ for $\Omega^{t^{n+1}}$

(3.iii) All triangles in $\mathcal{T}_h$ have angles bounded above by a constant $\vartheta < \pi/2$.

(3.iv) The intervals $(t^{n-1}, t^n]$ satisfy $\max_{1\le n\le N}(t^n - t^{n-1}) \le Ch$ with a sufficiently small constant C.

3.1. *Universal Mesh*

Let $\mathcal{T}_h$ be a triangulation of $\mathcal{D}$ satisfying conditions (3.i-3.iv), with h denoting the maximum diameter of an element $K \in \mathcal{T}_h$. For $i = 0, 1, 2, 3$, let $\mathcal{T}_{h,i}^t$ denote the collection of triangles $K \in \mathcal{T}_h$ for which exactly i vertices of K do not lie in the interior of Ω^t.

Fix a partition $0 = t^0 < t^1 < \cdots < t^N = T$ of the temporal axis. Our approach for constructing a conforming mesh $\mathcal{S}_h(t)$ for Ω^t, $t \in (t^{n-1}, t^n]$, will consist of identifying a subtriangulation $\mathcal{S}_h^n$ of the background triangulation $\mathcal{T}_h$ and defining a time-dependent bijection

$$\Phi^t : \mathcal{S}_h^n \to \Omega^t, \quad t \in (t^{n-1}, t^n].$$

Here and in the sequel, we abuse notation by writing $\mathcal{S}_h^n$ to denote both the triangulation (the list of vertices and their connectivities) as well as the region in $\mathbb{R}^2$ that it occupies. Our choice of $\mathcal{S}_h^n$ is

$$\mathcal{S}_h^n = \mathcal{T}_{h,0}^{t^{n-1}} \cup \mathcal{T}_{h,1}^{t^{n-1}} \cup \mathcal{T}_{h,2}^{t^{n-1}},$$

which is simply the set of triangles in the background triangulation with at least one vertex lying inside $\Omega^{t^{n-1}}$. Our choice of the map $\Phi^t : \mathcal{S}_h^n \to \Omega^t$ is that detailed in [63]. We recapitulate the explicit formulas for Φ^t in Appendix A.

For each $t \in (t^{n-1}, t^n]$, the map Φ^t delivers a conforming mesh of Ω^t having the same connectivity as $\mathcal{S}_h^n$ but consisting of triangles $\Phi^t(K)$, $K \in \mathcal{S}_h^n$. We label this curvilinear mesh

$\Phi^t(\mathcal{S}_h^n)$ and set

$$\mathcal{S}_h(t) = \Phi^t(\mathcal{S}_h^n), \quad t \in (t^{n-1}, t^n].$$

The remainder of this section is devoted to a discretization of (1-2) using finite element spaces over the evolving subtriangulation $\mathcal{S}_h(t)$. As is customary for readers familiar with ALE schemes, the resulting discretization (cf. (17)) over each short time interval $(t^{n-1}, t^n]$ will resemble a discretization of

$$\frac{Du}{Dt} + (u - v) \cdot \nabla_x u - \nu \Delta_x u = -\nabla_x p \qquad \text{in } \mathcal{D} \setminus P^t \tag{8}$$

$$\nabla_x \cdot u = 0 \qquad \text{in } \mathcal{D} \setminus P^t, \tag{9}$$

where

$$\frac{Du}{Dt} = \frac{\partial u}{\partial t} + v \cdot \nabla_x u$$

denotes the *material time derivative* of u along the path of a material particle that moves with the mesh $\mathcal{S}_h(t)$, whose velocity we denote by

$$v(\Phi^t(X), t) = \dot{\Phi}^t(X) = \left.\frac{\partial}{\partial t}\right|_X \Phi^t(X). \tag{10}$$

Since the subtriangulation $\mathcal{S}_h(t)$ changes abruptly at each t^n, $n = 1, 2, \dots, N$, a projection will be used to transfer information between finite element spaces at such instants; cf. Section 3.3.

3.2. *Galerkin Formulation over Short Time Intervals*

We now describe a spatial discretization of (1-2) that is obtained by performing a Galerkin projection of the weak equations (6-7) onto finite element subspaces $\mathring{\mathcal{V}}_h^t \subset \mathring{\mathcal{V}}^t$, $\mathcal{V}_h^t \subset \mathcal{V}^t$, and $\mathcal{Q}_h^t \subset \mathcal{Q}^t$ over a short time interval $(t^{n-1}, t^n]$. We focus on the case in which the boundary conditions are given by (3-4). The case in which the boundary conditions are given by (3) and (5) is handled similarly.

Here, we consider the use of Taylor-Hood $\mathcal{P}^k$-$\mathcal{P}^{k-1}$ finite elements with an integer $k \geq 2$ [61]. Such elements approximate the velocity field u and the pressure field p with continuous functions that are elementwise polynomials of degree at most k and $k-1$, respectively, on $\mathcal{S}_h(t)$. These finite element spaces are easy to construct with the aid of the map $\Phi^t : \mathcal{S}_h^n \to \Omega^t$ introduced in Section 3.1. Namely,

$$\begin{aligned}
\mathring{\mathcal{V}}_h^t &= \left\{ u_h \in C^0(\overline{\Omega^t})^2 \;\middle|\; u_h \circ \Phi^t\big|_K \in \mathcal{P}^k(K)^2 \, \forall K \in \mathcal{S}_h^n,\, u_h = 0 \text{ on } \partial P^t \right\} \\
\mathcal{V}_h^t &= \left\{ u_h \in C^0(\overline{\Omega^t})^2 \middle| u_h \circ \Phi^t\big|_K \in \mathcal{P}^k(K)^2 \, \forall K \in \mathcal{S}_h^n,\, u_h = i_h^t v_P(\cdot, t) \text{ on } \partial P^t \right\} \\
\mathcal{Q}_h^t &= \left\{ p_h \in C^0(\overline{\Omega^t}) \;\middle|\; p_h \circ \Phi^t\big|_K \in \mathcal{P}^{k-1}(K) \, \forall K \in \mathcal{S}_h^n \right\}.
\end{aligned}$$

Here, $i_h^t v_P(\cdot, t)$ denotes the nodal interpolant of $v_P(\cdot, t)$ onto the space of continuous functions on ∂P^t that are edgewise polynomials of degree k, i.e. $i_h^t v_P(\Phi^t(\cdot), t) \in \mathcal{P}^k(e)$ for every edge $e \subset \partial \mathcal{S}_h^n$.

The Galerkin projection of (6-7) over $(t^{n-1}, t^n]$ then reads: Find $u_h(\cdot, t) \in \mathcal{V}_h^t$ and $p_h(\cdot, t) \in \mathcal{Q}_h^t$ such that

$$m^t(\dot{u}_h, w_h) + a^t(u_h, w_h) + c^t(w_h, p_h) = 0 \quad \forall w_h \in \mathring{\mathcal{V}}_h^t \tag{11}$$

$$c^t(u_h, q_h) = 0 \quad \forall q_h \in \mathcal{Q}_h^t \tag{12}$$

for every $t \in (t^{n-1}, t^n]$.

The system (11-12) is equivalent to a system of differential-algebraic equations (DAEs). To deduce this, it is convenient to construct bases for $\mathring{\mathcal{V}}_h^t$, $\mathcal{V}_h^t$, and $\mathcal{Q}_h^t$ by composing (a subset of) shape functions on the background mesh $\mathcal{T}_h$ with the map $(\Phi^t)^{-1}$. Let

$$\begin{aligned}
\widetilde{\mathcal{V}}_h &= \left\{ U_h \in C^0(\overline{\mathcal{D}})^2 \middle| \, U_h|_K \in \mathcal{P}^k(K)^2 \, \forall K \in \mathcal{T}_h \right\} \\
\widetilde{\mathcal{Q}}_h &= \left\{ P_h \in C^0(\overline{\mathcal{D}}) \middle| \, P_h|_K \in \mathcal{P}^{k-1}(K) \, \forall K \in \mathcal{T}_h \right\}.
\end{aligned}$$

Prepared using nmeauth.cls
DOI: 10.1002/nme

In what follows, we will derive from (11-12) a system of DAEs of dimension $\mathcal{N}_u + \mathcal{N}_p$, where $\mathcal{N}_u = \dim(\widetilde{\mathcal{V}}_h)$ and $\mathcal{N}_p = \dim(\widetilde{\mathcal{Q}}_h)$.

Let $\{\widetilde{N}_a\}_{a=1}^{\mathcal{N}_u}$ and $\{\widetilde{M}_k\}_{k=1}^{\mathcal{N}_p}$ be the standard Lagrange bases for $\widetilde{\mathcal{V}}_h$ and $\widetilde{\mathcal{Q}}_h$, respectively, indexed by global degree of freedom numbers. Let $\{X_a\}_{a=1}^{\mathcal{N}_u}$ and $\{Y_k\}_{k=1}^{\mathcal{N}_p}$ denote the locations of the corresponding degrees of freedom in $\mathcal{T}_h$. Additionally, let

$$\begin{aligned}
\mathring{\mathcal{I}}_u^n &= \{1 \leq a \leq \mathcal{N}_u \mid \mathrm{supp}(\widetilde{N}_a) \cap \mathrm{int}(\mathcal{S}_h^n) \neq \emptyset,\ \widetilde{N}_a = 0 \text{ on } \partial\mathcal{S}_h^n \setminus \partial\mathcal{D}\} \\
\mathcal{I}_u^n &= \{1 \leq a \leq \mathcal{N}_u \mid \mathrm{supp}(\widetilde{N}_a) \cap \mathrm{int}(\mathcal{S}_h^n) \neq \emptyset\} \\
\mathcal{I}_p^n &= \{1 \leq k \leq \mathcal{N}_p \mid \mathrm{supp}(\widetilde{M}_k) \cap \mathrm{int}(\mathcal{S}_h^n) \neq \emptyset\}.
\end{aligned}$$

where $\mathrm{supp}(f)$ denotes the support of a function f and $\mathrm{int}(S)$ denotes the interior of a set S. Bases for the spaces $\mathring{\mathcal{V}}_h^t$, $\mathcal{V}_h^t$, and $\mathcal{Q}_h^t$ are easily constructed with the aid of the functions $n_a^t : \Omega^t \to \mathbb{R}^2$ and $m_k^t : \Omega^t \to \mathbb{R}$ given by

$$n_a^t(\Phi^t(X)) = \widetilde{N}_a(X), \quad a \in \mathcal{I}_u^n \tag{13}$$

and

$$m_k^t(\Phi^t(X)) = \widetilde{M}_k(X), \quad k \in \mathcal{I}_p^n. \tag{14}$$

Namely,

$$\begin{aligned}
\mathring{\mathcal{V}}_h^t &= \mathrm{span}\left\{ n_a^t \mid a \in \mathring{\mathcal{I}}_u^n \right\} \\
\mathcal{V}_h^t &= \mathring{\mathcal{V}}_h^t + \sum_{a \in \mathcal{I}_u^n \setminus \mathring{\mathcal{I}}_u^n} v_P(\Phi^t(X_a), t) n_a^t \\
\mathcal{Q}_h^t &= \mathrm{span}\left\{ m_k^t \mid k \in \mathcal{I}_p^n \right\}.
\end{aligned}$$

If we adopt the convention that $n_a^t = 0$ in Ω^t for $a \notin \mathcal{I}_u^n$ and $m_k^t = 0$ in Ω^t for $k \notin \mathcal{I}_p^n$, we may expand

$$u_h(x,t) = \sum_{a=1}^{\mathcal{N}_u} \mathbf{u}_a(t) n_a^t(x) \tag{15}$$

and

$$p_h(x,t) = \sum_{k=1}^{\mathcal{N}_p} \mathbf{p}_k(t) m_k^t(x) \tag{16}$$

as linear combinations of the shape functions n_a^t and m_a^t, bearing in mind that

$$u_h(\cdot, t) \in \mathcal{V}_h^t \implies \mathbf{u}_a(t) = v_P(\Phi^t(X_a), t) \ \forall a \in \mathcal{I}_u^n \setminus \mathring{\mathcal{I}}_u^n.$$

In the expansions above, we adopt the convention that $\mathbf{u}_a(t) = 0$ for $a \notin \mathcal{I}_u^n$ and $\mathbf{p}_k(t) = 0$ for $k \notin \mathcal{I}_p^n$. Observe that by (13),

$$\begin{aligned}
\dot{u}_h(x,t) &= \sum_{a=1}^{\mathcal{N}_u} \dot{\mathbf{u}}_a(t) n_a^t(x) + \sum_{a=1}^{\mathcal{N}_u} \mathbf{u}_a(t) \frac{\partial n_a^t}{\partial t}(x) \\
&= \sum_{a=1}^{\mathcal{N}_u} \dot{\mathbf{u}}_a(t) n_a^t(x) + \sum_{a=1}^{\mathcal{N}_u} \mathbf{u}_a(t)(-v(x,t) \cdot \nabla_x n_a^t(x)) \\
&= \sum_{a=1}^{\mathcal{N}_u} \dot{\mathbf{u}}_a(t) n_a^t(x) - v(x,t) \cdot \nabla_x u_h(x,t)
\end{aligned}$$

where v is given by (10).

Prepared using nmeauth.cls
DOI: 10.1002/nme

It follows that (11-12) is equivalent to the system of DAEs

$$\mathbf{M}(t)\begin{pmatrix}\dot{\mathbf{u}}(t)\\0\end{pmatrix}+\mathbf{K}(t)\begin{pmatrix}\mathbf{u}(t)\\\mathbf{p}(t)\end{pmatrix}+\begin{pmatrix}\mathbf{b}(\mathbf{u}(t),t)\\0\end{pmatrix}=\begin{pmatrix}\mathbf{f}(t)\\0\end{pmatrix} \tag{17}$$

where

$$\mathbf{M}(t)=\begin{pmatrix}\mathbf{M}_u(t) & 0\\ 0 & 0\end{pmatrix}$$
$$\mathbf{K}(t)=\begin{pmatrix}\mathbf{K}_u(t) & \mathring{\mathbf{C}}(t)^T\\ \mathbf{C}(t) & \mathbf{Z}\end{pmatrix}$$

and the entries of $\mathbf{M}_u(t),\mathbf{K}_u(t)\in\mathbb{R}^{\mathcal{N}_u\times\mathcal{N}_u}$, $\mathbf{C}(t),\mathring{\mathbf{C}}(t)\in\mathbb{R}^{\mathcal{N}_p\times\mathcal{N}_u}$, $\mathbf{Z}\in\mathbb{R}^{\mathcal{N}_p\times\mathcal{N}_p}$, and $\mathbf{b}(\mathbf{u}(t),t),\mathbf{f}(t)\in\mathbb{R}^{\mathcal{N}_u}$ are given by

$$\mathbf{M}_{u,ab}(t)=\begin{cases}m^t(n_b^t,n_a^t) & \text{if } a\in\mathring{\mathcal{I}}_u^n, b\in\mathcal{I}_u^n\\ 0 & \text{otherwise}\end{cases}$$
$$\mathbf{K}_{u,ab}(t)=\begin{cases}a_\nu^t(n_b^t,n_a^t) & \text{if } a\in\mathring{\mathcal{I}}_u^n, b\in\mathcal{I}_u^n\\ \delta_{ab} & \text{otherwise}\end{cases}$$
$$\mathbf{C}_{kb}(t)=\begin{cases}c^t(n_b^t,m_k^t) & \text{if } k\in\mathcal{I}_p^n, b\in\mathcal{I}_u^n\\ 0 & \text{otherwise}\end{cases}$$
$$\mathring{\mathbf{C}}_{kb}(t)=\begin{cases}c^t(n_b^t,m_k^t) & \text{if } k\in\mathcal{I}_p^n, b\in\mathring{\mathcal{I}}_u^n\\ 0 & \text{otherwise}\end{cases}$$
$$\mathbf{Z}_{kl}=\begin{cases}0 & \text{if } k\in\mathcal{I}_p^n\\ \delta_{kl} & \text{otherwise}\end{cases}$$
$$\mathbf{b}_a(\mathbf{u}(t),t)=\begin{cases}b^t(u_h-v,u_h,n_a^t) & \text{if } a\in\mathring{\mathcal{I}}_u^n\\ 0 & \text{otherwise}\end{cases}$$
$$\mathbf{f}_a(t)=\begin{cases}v_P(\Phi^t(X_a),t) & \text{if } a\in\mathcal{I}_u^n\setminus\mathring{\mathcal{I}}_u^n\\ 0 & \text{otherwise.}\end{cases}$$

Here, δ_{ab} denotes the Kronecker delta, and

$$a_\nu^t(u,w)=\nu\int_{\Omega^t}\left(\nabla_x u+(\nabla_x u)^T\right):\nabla_x w\,dx$$
$$b^t(u_1,u_2,w)=\int_{\Omega^t}(u_1\cdot\nabla_x u_2)\,w\,dx.$$

In hindsight, it is now evident that (17) is a discretization of the equations (8-9) that were alluded to earlier.

Remark. In the preceding paragraphs, we opted to construct a system of DAEs of dimension $\mathcal{N}_u+\mathcal{N}_p$ over each short time interval $(t^{n-1},t^n]$, even though a portion of those DAEs correspond to degrees of freedom in the background mesh that do not belong to $\mathcal{S}_h^n$. This was accomplished by incorporating the set of trivial DAEs $\mathbf{u}_a(t)=0$ for $a\notin\mathcal{I}_u^n$ and $\mathbf{p}_k(t)=0$ for $k\notin\mathcal{I}_p^n$ into the system via the prescriptions $\mathbf{K}_{u,ab}(t)=\delta_{ab}$ for $a\notin\mathcal{I}_u^n$ and $\mathbf{Z}_{kl}(t)=\delta_{kl}$ for $k\notin\mathcal{I}_p^n$, respectively. The boundary conditions $\mathbf{u}_a(t)=v_P(\Phi^t(X_a),t)$ for $a\in\mathcal{I}_u^n\setminus\mathring{\mathcal{I}}_u^n$ were incorporated similarly via the prescriptions of $\mathbf{K}_u(t)$ and $\mathbf{f}(t)$.

We did this to highlight an important feature of the universal mesh: it permits the use of the same data structures (e.g., the matrices $\mathbf{M}(t)$ and $\mathbf{K}(t)$) over the complete duration of the simulation, not merely over the intervals during which the mesh evolves continuously. The sparsity patterns of these data structures are invariant since the connectivity of the background mesh never changes.

Having said that, it is worth noting that one could, in principle, choose to replace the background mesh with a new one satisfying (3.i-3.iv) at any temporal node t^n. Such a strategy may be useful if, for example, a local refinement or coarsening is desired at a particular stage of the simulation. Needless to say, the sizes and sparsity patterns of the data structures would generally change in this scenario. However, the theory presented in [66] suggests that the order of accuracy of the method is maintained as long as such replacements of the background mesh occur a number of times that remains bounded under refinement.

3.3. *Initial Condition on each Short Time Interval*

In order to complete the prescription of $(\mathbf{u}(t), \mathbf{p}(t))$ over a short time interval $(t^{n-1}, t^n]$, the system of DAEs (17) must be supplemented with an initial condition $\mathbf{u}(t_+^{n-1})$. Note that an initial condition for the pressure $\mathbf{p}$ is unnecessary.

Since the spaces $\mathcal{V}_h^{t^{n-1}}$ and $\mathcal{V}_h^{t_+^{n-1}}$ generally need not coincide, a projection is needed in order transfer information between finite element spaces. To this end, we set

$$u_h(\cdot, t_+^{n-1}) = i_h^{t_+^{n-1}} u_h(\cdot, t^{n-1}), \tag{18}$$

where i_h^t is the nodal interpolant onto $\mathcal{V}_h^t$ [62, Chapter 1]. The corresponding vector $\mathbf{u}(t_+^{n-1})$ then consists of the coefficients $\mathbf{u}_a(t_+^{n-1})$ in the expansion (15).

We remark that more generally, one may consider the use of other surjective, linear projectors onto $\mathcal{V}_h^t$, such as the orthogonal projector onto $\mathcal{V}_h^t$ with respect to the L^2- or H^1-inner products. The theory presented in [66] supports the use of the L^2-projection, though the use of interpolation has always proven satisfactory in our numerical examples.

The influence of repeated projections such as (18) on the accuracy of the method is analyzed in [66, 67] and discussed in [63]. In brief, the projections introduce a half-order reduction in the method's order of accuracy in the L^2-norm for linear parabolic problems when the L^2-projector is adopted. Conservation of total energy and momentum are of course also influenced, and it may be desirable in some situations to consider projectors designed with these considerations in mind; see, for example [68].

3.4. *Temporal Discretization*

The setup we have described thus far offers the freedom to employ a time integrator of one's choosing to numerically integrate (17), a system of DAEs of index 2, from $t = t^{n-1}$ to $t = t^n$. Below we present two examples of integration schemes: a Singly Diagonally Implicit Runge-Kutta (SDIRK) scheme [69, 70], and a fractional step scheme [71–73]. In accordance with common guidelines for numerically solving DAEs, the SDIRK schemes we consider are stiffly accurate (and hence L-stable) methods [70, 74]. The same schemes are considered by, for instance, [75, 76], in their studies of high-order methods for the Navier-Stokes equations on fixed domains.

For the forthcoming discussion, we remind the reader that the temporal nodes t^n demarcate changes in the reference triangulation $\mathcal{S}_h^n$; hence, the time step Δt adopted during integration from t^{n-1} to t^n must be less than or equal to $t^n - t^{n-1}$ for every n. In practice, we often take $\Delta t = t^n - t^{n-1}$, though this is by no means a necessity. Recall also that, in accordance with (3.iv), the time intervals $(t^n - t^{n-1})$ scale with the mesh spacing h.

Singly Diagonally Implicit Runge-Kutta. Consider the use of a stiffly accurate s-stage Singly Diagonally Implicit Runge-Kutta (SDIRK) scheme of order $\leq s$ with a time step $\Delta t \leq (t^n - t^{n-1})$. At a given time $\tau_0 \in [t^{n-1}, t^n]$, such an integrator advances the current numerical solution $(\mathbf{u}_0, \mathbf{p}_0) \approx (\mathbf{u}(\tau_{0+}), \mathbf{p}(\tau_{0+}))$ to time $t = \tau_0 + \Delta t$ by solving a sequence of s systems of equations, as detailed below. The coefficients $\gamma > 0$ and $\beta_{ij} \in \mathbb{R}$, $i = 1, 2, \ldots, s$, $j = 0, 1, \ldots, i-1$, for various SDIRK methods are tabulated in B, Tables I-IV.

DOI: 10.1002/nme

Algorithm 3.1 SDIRK scheme for integration from $t = \tau_0 \in [t^{n-1}, t^n]$ to $t = \tau_0 + \Delta t \in [t^{n-1}, t^n]$.

Require: Initial condition $(\mathbf{u}_0, \mathbf{p}_0) \approx (\mathbf{u}(\tau_{0+}), \mathbf{p}(\tau_{0+}))$.

1: **for** $i = 1, 2, \dots, s$ **do**

2: Set

$$\tau_i = \sum_{j=0}^{i-1} \beta_{ij} \tau_j + \gamma \Delta t$$

and

$$\mathbf{u}_* = \sum_{j=0}^{i-1} \beta_{ij} \mathbf{u}_j.$$

3: Solve

$$(\mathbf{M}(\tau_i) + \gamma \Delta t \mathbf{K}(\tau_i)) \begin{pmatrix} \mathbf{u}_i \\ \mathbf{p}_i \end{pmatrix} + \gamma \Delta t \begin{pmatrix} \mathbf{b}(\mathbf{u}_i, \tau_i) \\ 0 \end{pmatrix} = \mathbf{M}(\tau_i) \begin{pmatrix} \mathbf{u}_* \\ 0 \end{pmatrix} + \gamma \Delta t \begin{pmatrix} \mathbf{f}(\tau_i) \\ 0 \end{pmatrix}$$

for $\begin{pmatrix} \mathbf{u}_i \\ \mathbf{p}_i \end{pmatrix}$.

4: **end for**

5: **return** $(\mathbf{u}_s, \mathbf{p}_s) \approx (\mathbf{u}(\tau_0 + \Delta t), \mathbf{p}(\tau_0 + \Delta t))$.

Pragmatically, implementing an SDIRK method amounts to computing s "backward-Euler" steps, with the initial condition at the i^{th} stage given by a linear combination of the solutions at the previous stages. Also, notice that the deformed mesh $\mathcal{S}_h(t)$ is constructed at each one of the s stages of the integration step, namely, for $t \in \{\tau_1, \dots, \tau_s\}$.

Fractional Step Scheme. Our second example of a time integrator is a fractional step scheme with a time step $\Delta t \leq (t^n - t^{n-1})$. The scheme we propose is an adaptation of classical fractional step schemes [71–73] to the setting in which the fluid domain evolves with time.

At a given time $\tau_0 \in [t^{n-1}, t^n]$, the fractional step scheme that we propose advances the current numerical solution $\mathbf{u}_0 \approx \mathbf{u}(\tau_{0+})$ to time $t = \tau_0 + \Delta t$ using a sequence of three steps. First, a preliminary approximation $\mathbf{u}_* \approx \mathbf{u}(\tau_0 + \Delta t)$ that need not satisfy the incompressibility constraint is computed. Next, $\mathbf{u}_*$ is projected onto the space of divergence-free vector fields by solving a Neumann problem for an auxiliary variable $\boldsymbol{\phi}$, leading to a divergence-free quantity $\mathbf{u}_1 \approx \mathbf{u}(\tau_0 + \Delta t)$ that serves as the time-Δt advancement of $\mathbf{u}_0$. Finally, an approximation $\mathbf{p}_{1/2}$ to the pressure at $t = \tau_0 + \Delta t/2$ is computed.

To present the scheme in detail, we denote by $\mathbf{M}_p(t)$ and $\mathbf{K}_p(t)$ the $\mathcal{N}_p \times \mathcal{N}_p$ matrices with entries

$$\mathbf{M}_{p,kl}(t) = \begin{cases} \displaystyle\int_{\Omega^t} m_k^t m_l^t \, dx & \text{if } k \in \mathcal{I}_p^n, l \in \mathcal{I}_p^n \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{K}_{p,kl}(t) = \begin{cases} \displaystyle\int_{\Omega^t} \nabla_x m_k^t \cdot \nabla_x m_l^t \, dx & \text{if } k \in \mathcal{I}_p^n, l \in \mathcal{I}_p^n \\ \delta_{kl} & \text{otherwise.} \end{cases}$$

We denote $\tau_{1/2} = \tau_0 + \Delta t/2$, and we use $\bar{\mathbf{p}}_{1/2}$ to denote a preliminary approximation to $\mathbf{p}(\tau_{1/2})$, which will be specified shortly. The details of the algorithm follow.

Algorithm 3.2 Fractional step scheme for integration from $t = \tau_0 \in [t^{n-1}, t^n]$ to $t = \tau_0 + \Delta t \in [t^{n-1}, t^n]$.

Require: Initial condition $\mathbf{u}_0 \approx \mathbf{u}(\tau_{0+})$, and preliminary approximation $\bar{\mathbf{p}}_{1/2} \approx \mathbf{p}(\tau_{1/2})$.

1: Solve

$$\mathbf{M}_u(\tau_{1/2}) \left(\frac{\mathbf{u}_* - \mathbf{u}_0}{\Delta t} \right) + \mathbf{K}_u(\tau_{1/2}) \left(\frac{\mathbf{u}_0 + \mathbf{u}_*}{2} \right) + \mathring{\mathbf{C}}(\tau_{1/2})^T \bar{\mathbf{p}}_{1/2} + \mathbf{b}\left(\frac{\mathbf{u}_0 + \mathbf{u}_*}{2}, \tau_{1/2} \right) = \frac{\mathbf{f}(\tau_0) + \mathbf{f}(\tau_0 + \Delta t)}{2}$$

for $\mathbf{u}_*$.

2: With $\boldsymbol{\ell} = \Delta t^{-1} \mathbf{C}(\tau_{1/2}) \mathbf{u}_*$, solve

$$\mathbf{K}_p(\tau_{1/2}) \boldsymbol{\phi} = \boldsymbol{\ell}$$

for $\boldsymbol{\phi}$.

3: Set

$$\mathbf{u}_1 = \mathbf{u}_* - \Delta t \mathbf{M}_u(\tau_{1/2})^{-1} \mathring{\mathbf{C}}(\tau_{1/2})^T \boldsymbol{\phi}$$
$$\mathbf{p}_{1/2} = \bar{\mathbf{p}}_{1/2} + \boldsymbol{\phi} + \frac{\nu \Delta t}{2} \mathbf{M}_p(\tau_{1/2})^{-1} \boldsymbol{\ell}.$$

4: Return $(\mathbf{u}_1, \mathbf{p}_{1/2}) \approx (\mathbf{u}(\tau_0 + \Delta t), \mathbf{p}(\tau_0 + \Delta t/2))$.

The precise choices that we made in the update formulas (the boundary conditions imposed on $\mathbf{u}_*$, the boundary conditions imposed on $\boldsymbol{\phi}$, and the update to the pressure) correspond to those made by the projection method "PmII" described in [73]. In particular, we prescribe the boundary values of $\mathbf{u}_*$ with the known values of the velocity field at $\tau_0 + \Delta t$, we impose homogeneous Neumann boundary conditions on $\boldsymbol{\phi}$, and we use a pressure update that is known to deliver second-order accuracy in time for both the velocity and pressure variables in the case of a fixed domain.

To understand the origin of the preceding scheme, it is instructive to consider its spatially continuous, temporally discrete counterpart on a fixed domain ($\Omega^t = \Omega^0 = \mathcal{D} \setminus P^0 \ \forall t$). In this setting, Algorithm 3.2 reduces to the following scheme, where we denote by $u_0, u_*, u_1, \phi, p_{1/2}$, and $\bar{p}_{1/2}$ the spatially continuous counterparts of $\mathbf{u}_0, \mathbf{u}_*, \mathbf{u}_1, \boldsymbol{\phi}, \mathbf{p}_{1/2}$, and $\bar{\mathbf{p}}_{1/2}$, respectively:

1. Solve

$$\frac{u_* - u_0}{\Delta t} - \frac{\nu}{2}(\Delta_x u_0 + \Delta_x u_*) + \nabla_x \bar{p}_{1/2} + \frac{u_0 + u_*}{2} \cdot \nabla_x \frac{u_0 + u_*}{2} = 0 \quad \text{in} \quad \Omega^0 \tag{19}$$
$$u_* = 0 \quad \text{on} \quad \partial\mathcal{D} \tag{20}$$
$$u_* = v_P(\cdot, \tau_0 + \Delta t) \quad \text{on } \partial P^0 \tag{21}$$

for u_*.

2. Solve

$$\Delta t \Delta_x \phi = \nabla_x \cdot u_* \quad \text{in } \Omega^0$$
$$\frac{\partial \phi}{\partial n} = 0 \qquad \text{on } \partial\Omega^0$$

for ϕ.

3. Set

$$u_1 = u_* - \Delta t \nabla_x \phi \tag{22}$$
$$p_{1/2} = \bar{p}_{1/2} + \phi - \frac{\nu \Delta t}{2} \Delta_x \phi. \tag{23}$$

Prepared using nmeauth.cls
DOI: 10.1002/nme

As mentioned earlier, the scheme above is precisely the second-order method "PmII" of [73]. We have numerical evidence (cf. Section 4.1) and heuristic reasoning to suggest that our extension of the method to moving domains is likewise second-order accurate in time, though a justification of this assertion warrants further analysis.

Note that in step 2 of the algorithm, the linear system to be solved for ϕ is singular, as it corresponds to a Neumann problem whose solution is determined up to the addition of a constant. This is because for boundary conditions (3) and (5), the pressure is defined up to a constant. Defining ϕ unambiguously requires, for example, imposing the value of one entry of the vector ϕ arbitrarily. If a boundary condition of the form (4) were to be adopted on part of $\partial\mathcal{D}$, then ϕ would need to satisfy homogeneous Dirichlet boundary conditions therein.

Finally, we describe our choice of $\bar{\mathbf{p}}_{1/2}$, which is set to be equal to the last computed value $\mathbf{p}_{-1/2}$, interpolated onto the appropriate finite element space if necessary. More precisely, we set $\bar{\mathbf{p}}_{1/2} = \mathbf{p}_{-1/2}$ if $\tau_0 \neq t^{n-1}$ and $\bar{\mathbf{p}}_{1/2} = 0$ if $\tau_0 = 0$; otherwise, we set $\bar{\mathbf{p}}_{1/2}$ equal to the vector of coefficients in the expansion

$$\sum_{k=1}^{\mathcal{N}_p} \bar{\mathbf{p}}_{1/2,k} m_k^{t_+^{n-1}} = i_h^{t_+^{n-1}} \left(\sum_{k=1}^{\mathcal{N}_p} \mathbf{p}_{-1/2,k} m_k^{t^{n-1}} \right),$$

where, abusing notation, i_h^t denotes the nodal interpolant onto $\mathcal{Q}_h^t$. Note that choosing $\bar{\mathbf{p}}_{1/2} = 0$ for $\tau_0 = 0$ reduces the accuracy of the very first time step to first order (cf. [73]), however, it is easy see that (by analogy with multi-step methods for ODEs [70]) this does not reduce the order of the scheme's global truncation error.

3.5. *Algorithm Summary*

A summary of the proposed algorithm for integration over $[0, T]$ using a universal mesh $\mathcal{T}_h$ and a temporal partition $0 = t^0 < t^1 < \cdots < t^N = T$ is as follows.

Algorithm 3.3 Integration over $[0, T]$ using a universal mesh $\mathcal{T}_h$ and a temporal partition $0 = t^0 < t^1 < \cdots < t^N = T$.

Require: Initial condition $\mathbf{u}(0)$.
1: **for** $n = 1, 2, \ldots, N$ **do**
2: Identify the subtriangulation $\mathcal{S}_h^n$ of $\mathcal{T}_h$ consisting of triangles with at least one vertex lying inside $\Omega^{t^{n-1}}$.
3: Adapt $\mathcal{S}_h^n$ to $\Omega^{t^{n-1}}$ by computing $\mathcal{S}_h(t_+^{n-1}) := \Phi^{t_+^{n-1}}(\mathcal{S}_h^n)$, where $\Phi^t : \mathcal{S}_h^n \to \Omega^t$ is the universal mesh map (29).
4: Project $\mathbf{u}(t^{n-1})$ onto the finite element space $\mathcal{V}_h^{t_+^{n-1}}$ associated with $\mathcal{S}_h(t_+^{n-1})$, through (18), giving $\mathbf{u}(t_+^{n-1})$.
5: Numerically integrate (17) from $t = t_+^{n-1}$ to $t = t^n$ using a time integrator of one's choosing with time step $\Delta t \leq (t^n - t^{n-1})$, giving $(\mathbf{u}(t^n), \mathbf{p}(t^n))$.
6: **end for**
7: **return** $(\mathbf{u}(T), \mathbf{p}(T))$

Remarks.

1. In the last step of the algorithm, the numerical integration may require the evaluation of $\mathcal{S}_h(t) = \Phi^t(\mathcal{S}_h^n)$ at intermediate times $t \in (t^{n-1}, t^n]$ in order to assemble the quantities $\mathbf{M}(t)$, $\mathbf{K}(t)$, $\mathbf{b}(\mathbf{u}(t), t)$, and $\mathbf{f}(t)$ at intermediate stages of integration.

2. When the fractional step scheme (3.2) is adopted for numerical integration, the output of step (4) in such scheme is $(\mathbf{u}(t^n), \mathbf{p}(t^n - \Delta t/2))$.

DOI: 10.1002/nme

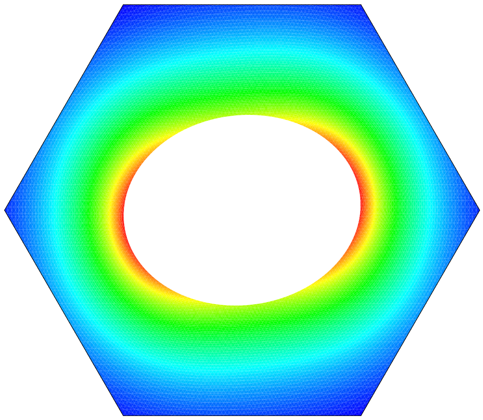


Figure 3. Velocity magnitude contours for the manufactured solution (24-26) at time $t = 0.05$.

4. NUMERICAL EXAMPLES

In this section, we apply the proposed methods to simulate flow around various obstacles with prescribed evolution. We consider three examples of obstacles: a rotating ellipse, an oscillating disk, and a rotating stirrer. We consider the rotating ellipse in order to study numerically the convergence of the methods. The remaining examples serve to illustrate the features of the methodology.

4.1. *Rotating Ellipse*

To study numerically the convergence of the methods, we considered the case in which obstacle P^t is an ellipse with semi-major axis $a = 1.0$ and semi-minor axis $b = 0.8$, rotating at a fixed angular velocity $\omega = 2.5$, as depicted in Fig. 3. Using [77] for inspiration, we manufactured a solution by adding a forcing term to the right-hand side of (1) so that the exact solution is given by

$$u_1(x_1, x_2, t) = -\frac{a^2+b^2}{\sqrt{a^4-b^4}}\omega e^{-\xi}(b\cos\omega t\sin\eta + a\sin\omega t\cos\eta) \tag{24}$$

$$u_2(x_1, x_2, t) = -\frac{a^2+b^2}{\sqrt{a^4-b^4}}\omega e^{-\xi}(b\sin\omega t\sin\eta - a\cos\omega t\cos\eta) \tag{25}$$

$$p(x_1, x_2, t) = \sin(x_1)\sin(x_2), \tag{26}$$

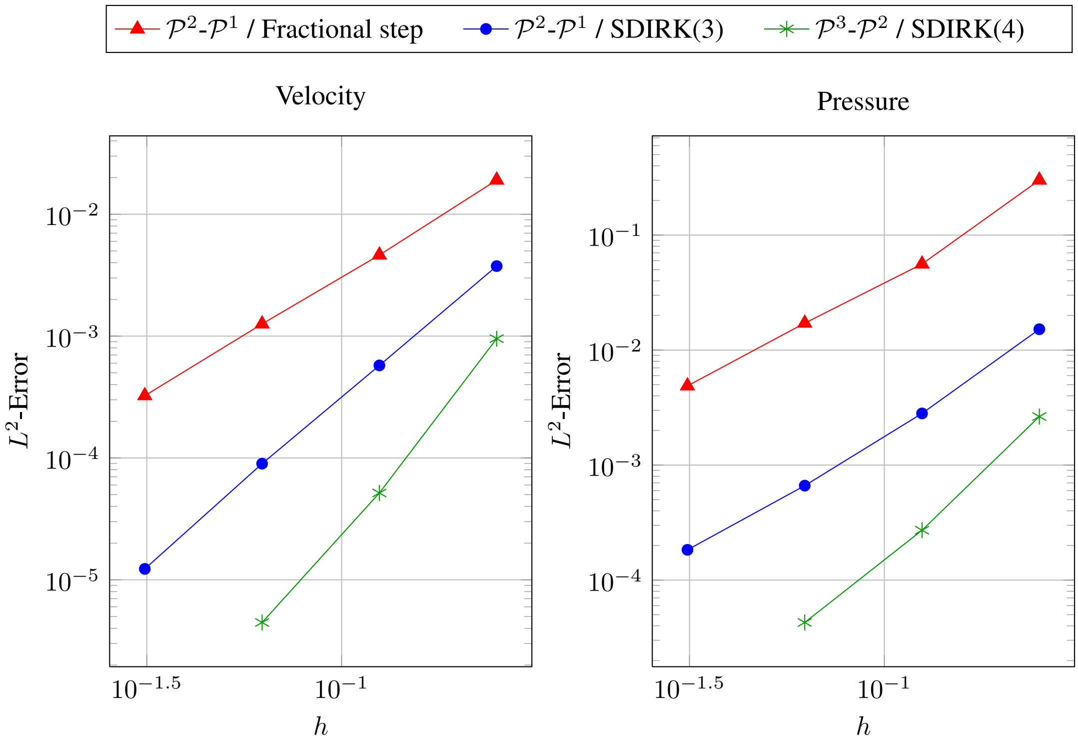
with $\xi \geq 0$ and $\eta \in [0, 2\pi)$ related to the cartesian coordinates x_1 and x_2 via

$$x_1\cos\omega t + x_2\sin\omega t = \frac{\sqrt{a^4-b^4}}{a}\cosh\xi\cos\eta$$
$$-x_1\sin\omega t + x_2\cos\omega t = \frac{\sqrt{a^4-b^4}}{b}\sinh\xi\sin\eta.$$

The velocity field so manufactured has the property that it is everywhere divergence-free and satisfies the no-slip condition (3) on ∂P^t. On the remainder of the fluid boundary, we prescribed the known values of the velocity field. We took $\nu = 1.0$ so that the Reynolds number of the flow was $Re = u_2(a, 0, 0)a/\nu = 2.5$.

We studied the L^2-error in u and p at time $T = 0.05$ on a sequence of uniform refinements of an equilateral triangle mesh with a lowest resolution mesh spacing of $h_0 = 0.25$, using a time step $\Delta t = Th/h_0$ and a temporal subdivision $t^n = n\Delta t$, $n = 0, 1, 2, \ldots, T/\Delta t$. We considered three combinations of finite elements and time integrators: Taylor-Hood $\mathcal{P}^2$-$\mathcal{P}^1$ elements together with the fractional step scheme (3.2),† Taylor-Hood $\mathcal{P}^2$-$\mathcal{P}^1$ elements together with an SDIRK scheme of

†In the case of the fractional step scheme, the error in p was measured at $t = T - \Delta t/2$ rather than at $t = T$.



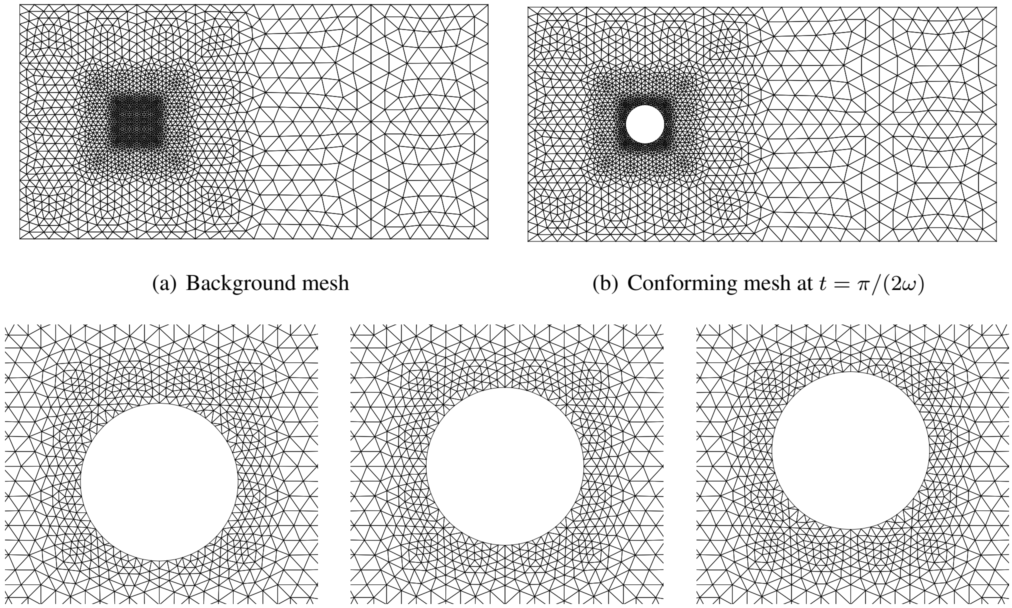
Velocity h_0/h	$\mathcal{P}^2$-$\mathcal{P}^1$ / Fractional step Error	Order	$\mathcal{P}^2$-$\mathcal{P}^1$ / SDIRK(3) Error	Order	$\mathcal{P}^3$-$\mathcal{P}^2$ / SDIRK(4) Error	Order
1	1.91e-02	-	3.75e-03	-	9.53e-04	-
2	4.63e-03	2.04	5.75e-04	2.71	5.16e-05	4.21
4	1.26e-03	1.88	8.98e-05	2.68	4.47e-06	3.53
8	3.24e-04	1.96	1.23e-05	2.87	-	-
Expected Order		1.5		2.5		3.5

Pressure h_0/h	$\mathcal{P}^2$-$\mathcal{P}^1$ / Fractional step Error	Order	$\mathcal{P}^2$-$\mathcal{P}^1$ / SDIRK(3) Error	Order	$\mathcal{P}^3$-$\mathcal{P}^2$ / SDIRK(4) Error	Order
1	3.01e-01	-	1.52e-02	-	2.64e-03	-
2	5.61e-02	2.42	2.81e-03	2.43	2.71e-04	3.28
4	1.72e-02	1.71	6.63e-04	2.09	4.28e-05	2.66
8	4.89e-03	1.81	1.83e-04	1.85	-	-
Expected Order		1.5		1.5		2.5

Figure 4. Convergence rates in the $L^2(\Omega^T)$-norm for the solution to incompressible, viscous flow around a rotating ellipse using three combinations of finite elements and time integrators with $\Delta t \propto h$: (1) Taylor-Hood $\mathcal{P}^2$-$\mathcal{P}^1$ elements together with the fractional step scheme (3.2), (2) Taylor-Hood $\mathcal{P}^2$-$\mathcal{P}^1$ elements together with a third-order implicit Runge-Kutta scheme, and (3) Taylor-Hood $\mathcal{P}^3$-$\mathcal{P}^2$ elements together with a fourth-order implicit Runge-Kutta scheme. Also shown in the tables are expected orders of convergence inferred from the theory presented in [63, 66].

order 3 (cf. Table III), and Taylor-Hood $\mathcal{P}^3$-$\mathcal{P}^2$ elements together with an SDIRK scheme of order 4 (cf. Table IV). The resulting spatial discretizations for $h_0/h = 1, 2, 4$, and 8, respectively, had 1,851, 7,155, 28,131, and 111,555 degrees of freedom (for $\mathcal{P}^2$-$\mathcal{P}^1$ elements) and 4,419, 17,283, 68,355, and 271,875 degrees of freedom (for $\mathcal{P}^3$-$\mathcal{P}^2$ elements). For each of the combinations of finite elements and time integrators considered, we observed convergence rates that are at worst suboptimal by half

DOI: 10.1002/nme



(a) Background mesh (b) Conforming mesh at $t = \pi/(2\omega)$

(c) Close-up of conforming mesh at $t = 0$ (d) Close-up of conforming mesh at $t = \pi/(2\omega)$ (e) Close-up of conforming mesh at $t = \pi/\omega$

Figure 5. Universal mesh for a disk with unit diameter oscillating with amplitude $A = 0.1$ and frequency ω.

an order. These results are consistent with the predictions of [63, 66, 67], which derive a priori error estimates that are suboptimal by half an order in the L^2-norm for schemes that adopt a universal mesh in conjunction with piecewise polynomial finite element spaces to solve a parabolic model problem.

4.2. *Oscillating Disk*

As a second example, we considered the case in which the obstacle P^t is a disk of radius $R = 1/2$ whose center oscillates vertically with amplitude A and frequency ω about a point $(x_0, y_0) = (-3, 0)$:

$$P^t = \{(x, y) \mid (x - x_0)^2 + (y - y_0 + A\cos(\omega t))^2 < R^2\}.$$

We immersed the oscillating disk in a domain $\mathcal{D} = [-6, 6] \times [-3, 3]$ and prescribed boundary conditions

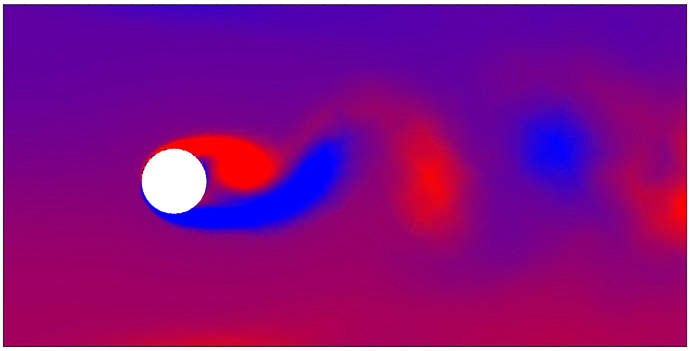
$$\begin{aligned} u &= \left(u_\infty(1 - e^{-t/2}), 0\right) && \text{on } [-6, 6] \times \{-3\} \cup [-6, 6] \times \{3\} \cup \{-6\} \times [-3, 3] \\ pn - \nu\left(\nabla_x u + (\nabla_x u)^T\right) n &= 0 && \text{on } \{6\} \times [-3, 3]. \\ u &= (0, \omega A \sin(\omega t)) && \text{on } \partial P^t, \end{aligned}$$

where $u_\infty = 1/2$.

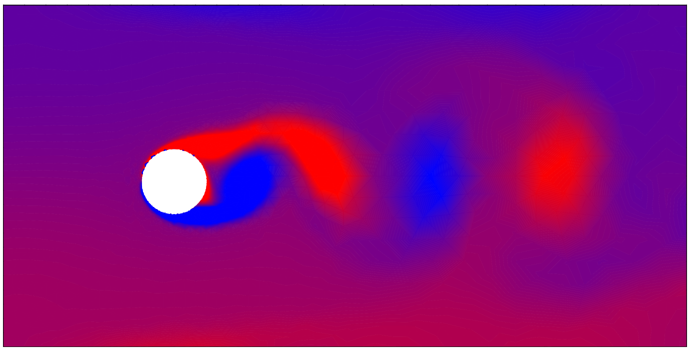
Fig. 5 shows the universal mesh that we adopted for this simulation, as well as snapshots of the resulting conforming mesh for $\Omega^t = \mathcal{D} \setminus P^t$ at a few representative instants in time when $A = 0.1$. The background mesh was constructed by inserting stencils of acute triangles into an adaptively refined quadtree; see [78] for details.

We first considered the cases in which $A = 0.1$ and $\omega = 0.8\omega_0$, ω_0, and $1.2\omega_0$, where ω_0 is the natural vortex shedding frequency for flow past a fixed disk of radius R, assuming a Strouhal number $St = (\omega_0/2\pi)(2R)/u_\infty = 0.195$. We took $\nu = 1/370$ so that $Re = (2R)u_\infty/\nu = 185$. We solved the problem using Taylor-Hood $\mathcal{P}^2$-$\mathcal{P}^1$ elements (leading to 18,701 degrees of freedom) together with the fractional step scheme (3.2), using $\Delta t = 0.2$ and $t^n = n\Delta t$. Figs. 6-7 show contours of the

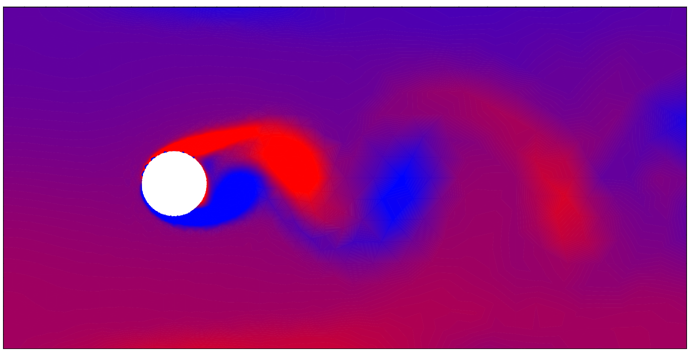
DOI: 10.1002/nme



(a) $\omega/\omega_0 = 0.8$



(b) $\omega/\omega_0 = 1$



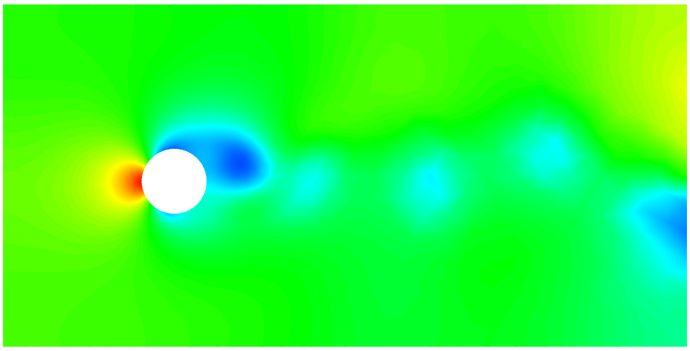
(c) $\omega/\omega_0 = 1.2$

Figure 6. Vorticity contours during flow past a disk with unit diameter oscillating with amplitude $A = 0.1$ and frequency (a) $\omega = 0.8\omega_0$, (b) $\omega = \omega_0$, and (c) $\omega = 1.2\omega_0$. The snapshot shown in each case corresponds to the largest time $t < 80$ for which the disk's vertical displacement is $-A$. A characteristic shift in the vortex shedding pattern's phase relative to the disk's oscillation occurs as ω passes through ω_0.

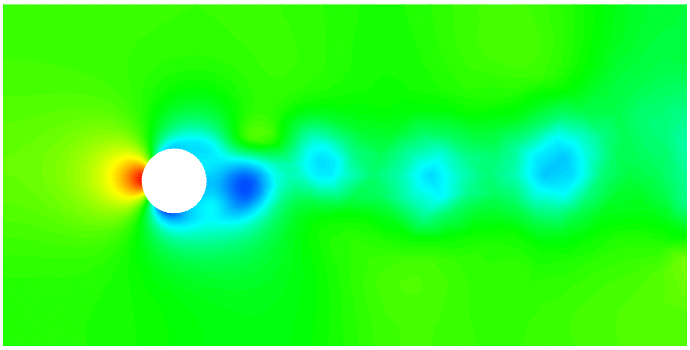
vorticity $\nabla_x \times u$ and the pressure p in each of the three cases at the largest time $t < 80$ for which the disk's vertical displacement is $-A$. We observed a characteristic shift in the vortex shedding pattern's phase relative to the disk's oscillation as ω passed through ω_0, which is consistent with past numerical and experimental studies of flow past an oscillating disk [79, 80].

Next, we fixed $\omega = 0.8\omega_0$ and $A = 0.2$ and studied the temporal evolution of the drag coefficient C_D and the lift coefficient C_L for the cases in which $\nu = 1/2$ (so that $Re = 1$) and $\nu = 1/370$

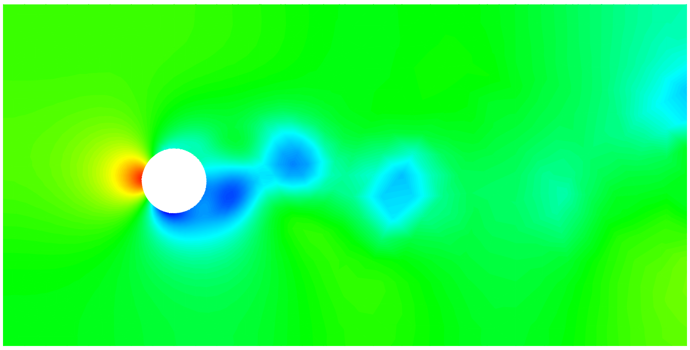
Prepared using nmeauth.cls
DOI: 10.1002/nme



(a) $\omega/\omega_0 = 0.8$



(b) $\omega/\omega_0 = 1$



(c) $\omega/\omega_0 = 1.2$

Figure 7. Pressure contours during flow past a disk with unit diameter oscillating with amplitude $A = 0.1$ and frequency (a) $\omega = 0.8\omega_0$, (b) $\omega = \omega_0$, and (c) $\omega = 1.2\omega_0$. These snapshots correspond to the same instants in time as in Fig. 6.

(so that $Re = 185$). Figs. 8-9 show the drag and lift coefficients obtained using meshes obtained from uniform refinements of the mesh depicted in Fig. 5, whose maximum element diameter is $h_0 := 0.58$. We used Taylor-Hood $\mathcal{P}^2$-$\mathcal{P}^1$ elements together with a 3^{rd}-order SDIRK scheme (cf. Table III), taking $\Delta t = 0.1h/h_0$ and $t^n = n\Delta t$. The resulting spatial discretizations had 18,701 and 74,335 degrees of freedom for $h_0/h = 1, 2$, respectively. The drag and lift coefficients were computed by direct integration over the boundary of the cylinder.

DOI: 10.1002/nme

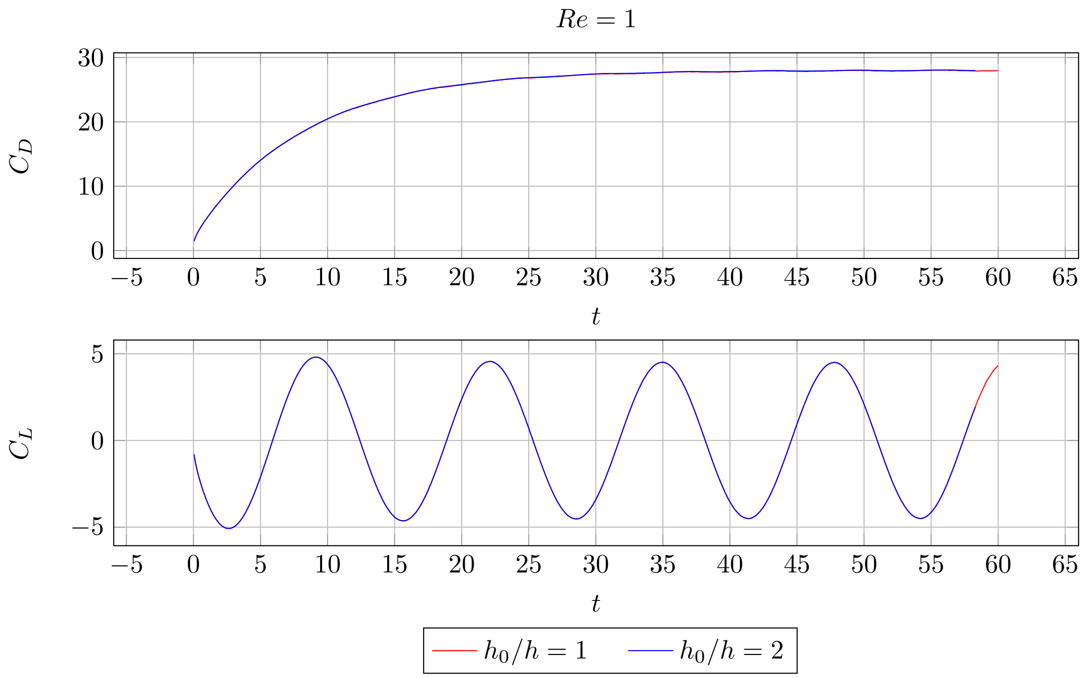


Figure 8. Drag and lift coefficients during flow past an oscillating disk at $Re = 1$. The results of two simulations are plotted, one corresponding to the mesh in Fig. 5 and one corresponding to a refinement thereof.

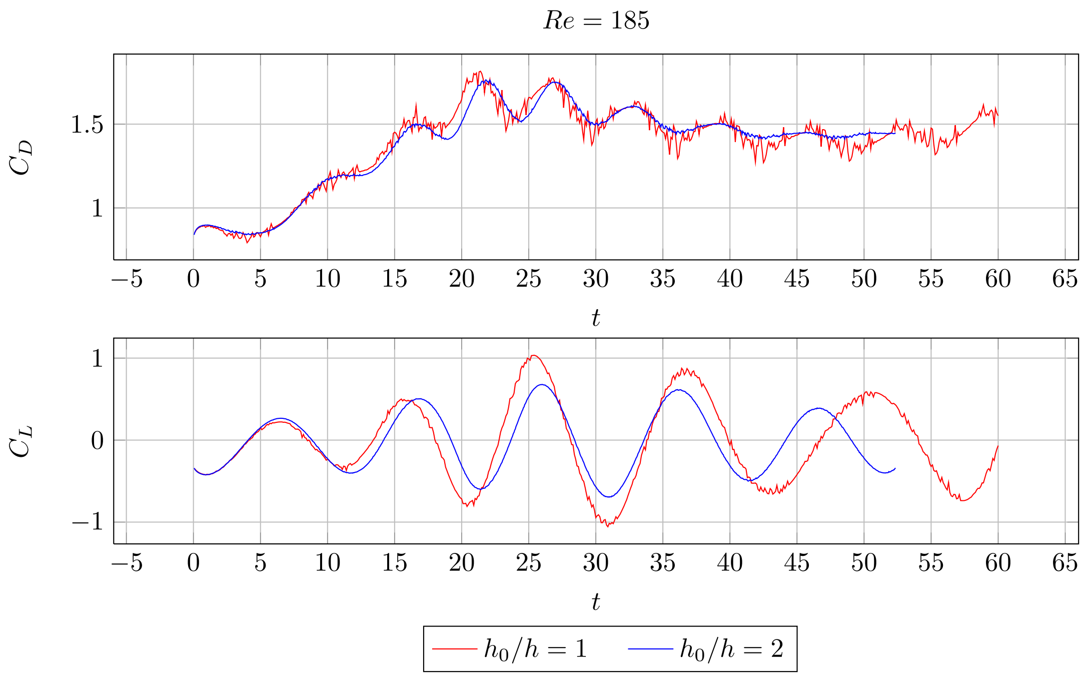


Figure 9. Drag and lift coefficients during flow past an oscillating disk at $Re = 185$. The results of two simulations are plotted, one corresponding to the mesh in Fig. 5 and one corresponding to a refinement thereof.

For $Re = 185$ (Fig. 9), the simulation on the coarsest mesh exhibits spurious oscillations of the drag and lift coefficients, but these are significantly reduced upon refinement. We suspect that the

DOI: 10.1002/nme

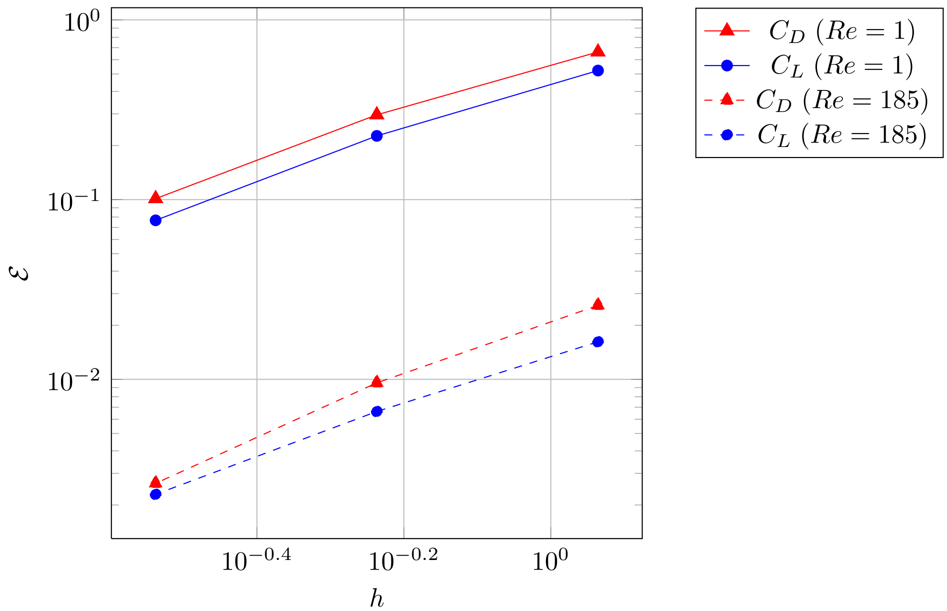


Figure 10. Convergence of the drag and lift coefficient time series under mesh refinement. The reported error $\mathcal{E}$ is the square root of (a rectangle-rule approximation of) the integrated squared error $(C_i(t) - \bar{C}_i(t))^2$, $i = L, D$, over the interval $[0, 1]$, relative to a reference solution $\bar{C}_i(t)$ obtained from a fine mesh with $h = 0.145$.

oscillations are attributable to the interpolation of the solution onto a new finite element space at each time t^n (cf. Section 3.3), since identical numerical experiments with a fixed disk rendered drag and lift coefficient time series that were free of artificial oscillations. Diffusion seems to also play a role in mitigating the artificial oscillations, as evidenced by their absence in Fig. 8, where $Re = 1$. Based upon these observations, it may be worthwhile to explore the possibility of designing more sophisticated strategies for transferring information between finite element spaces, such as projecting the velocity onto the space of divergence-free vector fields after interpolating, in order to obtain more satisfactory results on coarse meshes at high Reynolds numbers.

Fig. 10 shows the convergence of the computed drag and lift coefficients under the aforementioned mesh refinement. To measure the errors in the time series, we computed a rectangle-rule approximation $\mathcal{E}$ to the integrated error

$$\left(\int_0^1 (C_D(t) - \bar{C}_D(t))^2 \, dt \right)^{1/2}$$

between the computed solution $C_D(t)$ and a reference solution $\bar{C}_D(t)$ obtained from a fine mesh with $h = 0.145$, and likewise for the lift coefficient. The errors in all cases converged to zero at rates approximately of the order $h^{1.5}$.

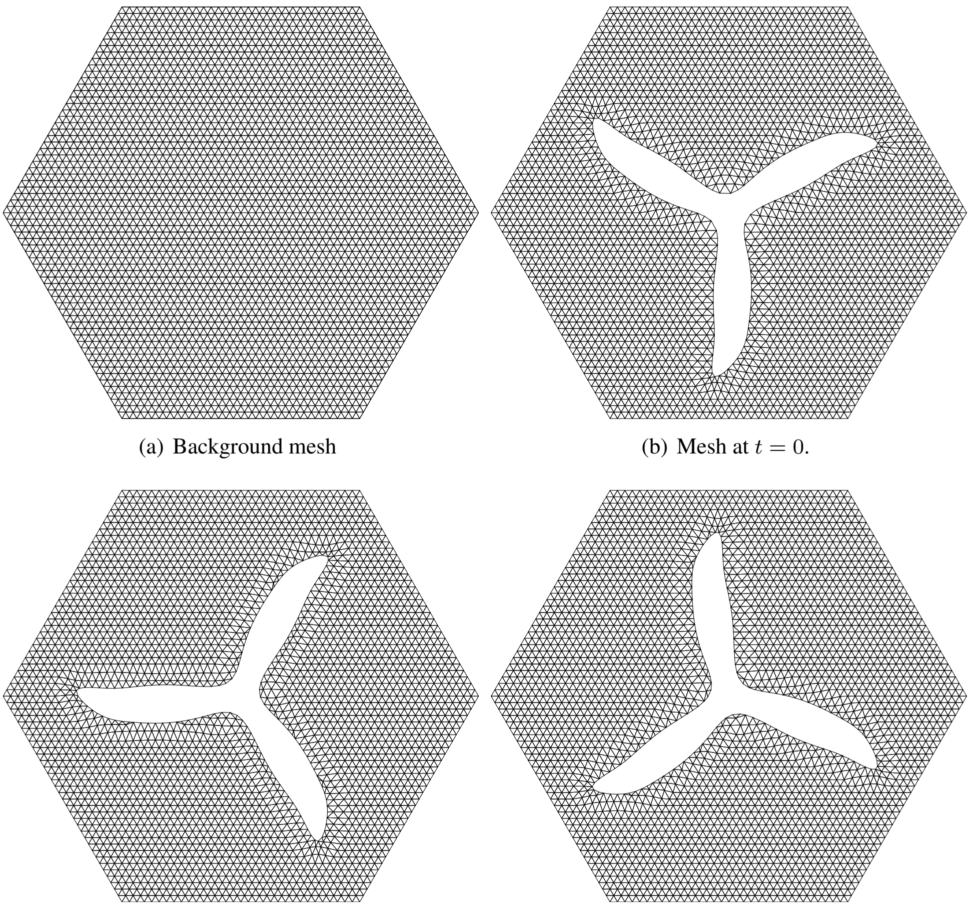
4.3. *Stirring a Viscous Fluid*

Our last example considers the case in which the obstacle boundary is a closed cubic spline in the shape of a propeller-like stirrer that rotates at a prescribed angular velocity

$$\omega(t) = \omega_0(1 - e^{-t/\tau}) \tag{27}$$

with $\omega_0 = 5.0$ and $\tau = 0.01$. The stirrer blades were of length ≈ 1.4 and average width ≈ 0.3. We took $\nu = 0.2$ in our simulations, so that the Reynold's number of the flow (treating the stirrer blade width as the characteristic length scale) was approximately 10.5. To approximate the velocity and pressure, we adopted Taylor-Hood $\mathcal{P}^2$-$\mathcal{P}^1$ elements. To integrate in time, we used an SDIRK scheme

DOI: 10.1002/nme



(a) Background mesh (b) Mesh at $t = 0$. (c) Mesh at $t = 0.125$. (d) Mesh at $t = 0.25$.

Figure 11. Universal mesh for a rotating stirrer.

of order 3. We immersed the stirrer in hexagonal domain $\mathcal{D}$ of diameter 4 and imposed Neumann boundary condtions (4) on $\partial\mathcal{D}$. We adopted a uniform background mesh of equilateral triangles ($h = 0.0625$, $18{,}701$ degrees of freedom).

Figs. 11 and 12 display snapshots of the mesh and velocity magnitude contours, respectively, at various times during the simulation. The robust nature of the methods introduced here is patent in this example, as traditional deforming-mesh methods could easily encounter difficulties with mesh entanglement upon rotation of the stirrer.

5. CONCLUDING REMARKS

We have presented a framework for computing incompressible, viscous flow around a moving obstacle with prescribed evolution using a universal mesh. By immersing the obstacle in a background mesh and adjusting a few elements in the neighborhood of obstacle's boundary, the strategy provides a conforming triangulation of the fluid domain at all times over which a spatial discretization of the fluid velocity and pressure fields of any desired order may be constructed using standard finite elements. The resulting semidiscrete equations may be integrated in time using standard time integrators for ODEs. We illustrated the framework using Taylor-Hood finite elements together with Runge-Kutta time integrators and a fractional step scheme. Numerical convergence tests confirmed the theory presented in [63, 66], which predicts orders of convergence that are

DOI: 10.1002/nme

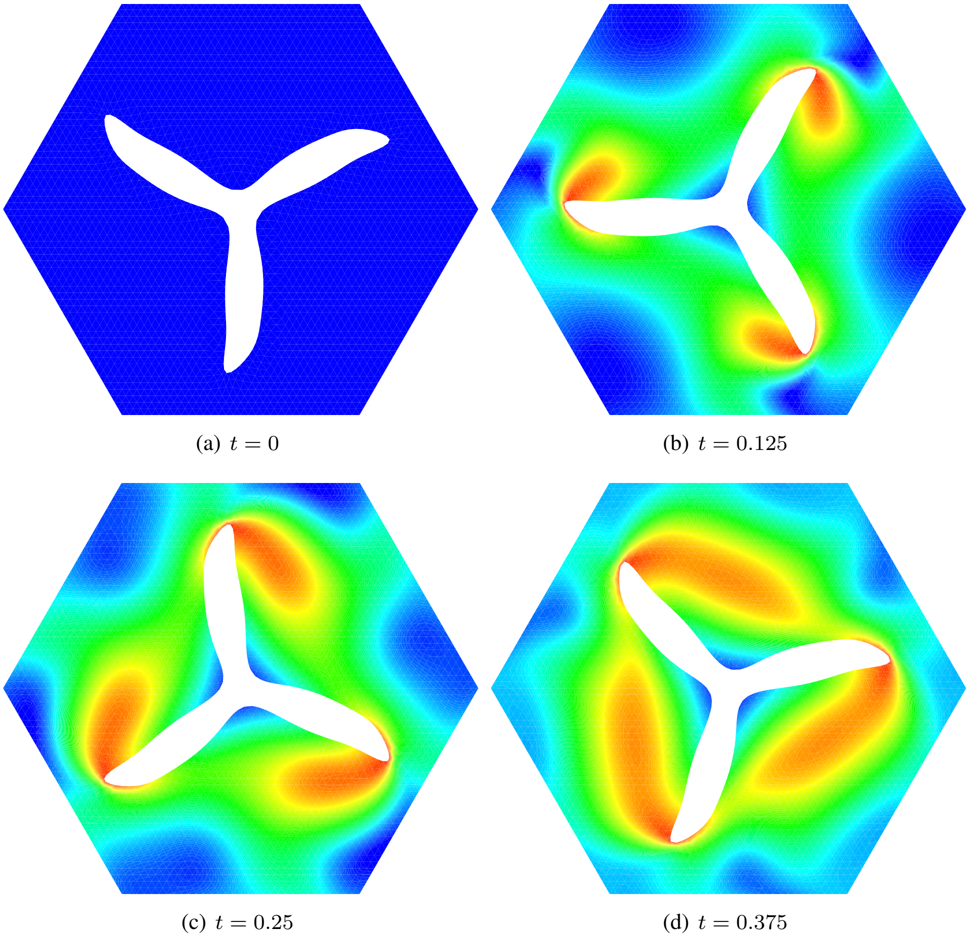


Figure 12. Velocity magnitude contours during a simulation of stirring of a viscous fluid.

suboptimal by half an order in the L^2-norm for a model parabolic problem. We demonstrated the method's versatility on numerical examples that involve flow past an oscillating disk and flow around a rotating stirrer.

All examples in the manuscript involved flows with low-to-moderate Reynolds number. This enabled us to obtain accurate solutions with relatively coarse and isotropic meshes. For larger Reynolds numbers, we expect to have to modify the spatial discretization by including a stabilization of the advection term. More importantly, meshes will have to be anisotropic and refined around the boundary of the moving obstacle, to capture the boundary layer. Doing it with a universal mesh is an open problem.

ACKNOWLEDGEMENTS

This research was supported by the U.S. Department of Energy grant DE-FG02-97ER25308; Department of the Army Research Grant, grant number: W911NF-07- 2-0027; NSF Career Award, grant number: CMMI-0747089; and NSF, grant number CMMI-1301396.

REFERENCES

1. Bonito A, Kyza I, Nochetto RH. Time-discrete higher order ALE formulations: A priori error analysis. *Numerische Mathematik* 2013; **125**(2):225–257.

2. Wang L, Persson PO. A discontinuous Galerkin method for the Navier-Stokes equations on deforming domains using unstructured moving space-time meshes 2013; .
3. Lomtev I, Kirby R, Karniadakis G. A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. *Journal of Computational Physics* 1999; **155**(1):128–159.
4. Minoli CAA, Kopriva DA. Discontinuous Galerkin spectral element approximations on moving meshes. *Journal of Computational Physics* 2011; **230**(5):1876–1902.
5. Yang Z, Mavriplis DJ. Unstructured dynamic meshes with higher-order time integration schemes for the unsteady Navier-Stokes equations. *AIAA paper* 2005; **1222**(2005):1.
6. Montefusculo F, Sousa F, Buscaglia G. High-order ALE schemes with applications in capillary flows. *(Submitted)* 2014; .
7. Hay A, Yu K, Etienne S, Garon A, Pelletier D. High-order temporal accuracy for 3D finite-element ALE flow simulations. *Computers & Fluids* 2014; **100**:204–217.
8. Mavriplis DJ, Nastase CR. On the geometric conservation law for high-order discontinuous Galerkin discretizations on dynamically deforming meshes. *Journal of Computational Physics* 2011; **230**(11):4285–4300.
9. Cori JF, Etienne S, Pelletier D, Garon A. Implicit Runge-Kutta time integrators for fluid-structure interactions. *48th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, FL*, 2010; 4–7.
10. Ferrer E, Willden RH. A high order discontinuous Galerkin–Fourier incompressible 3D Navier–Stokes solver with rotating sliding meshes. *Journal of Computational Physics* 2012; **231**(21):7037–7056.
11. Farhat C, van der Zee KG, Geuzaine P. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer methods in applied mechanics and engineering* 2006; **195**(17):1973–2001.
12. Farhat C, Rallu A, Wang K, Belytschko T. Robust and provably second-order explicit–explicit and implicit–explicit staggered time-integrators for highly non-linear compressible fluid–structure interaction problems. *International Journal for Numerical Methods in Engineering* 2010; **84**(1):73–107.
13. Geuzaine P, Grandmont C, Farhat C. Design and analysis of ALE schemes with provable second-order time-accuracy for inviscid and viscous flow simulations. *Journal of Computational Physics* 2003; **191**(1):206–227.
14. Boffi D, Gastaldi L. Stability and geometric conservation laws for ALE formulations. *Computer methods in applied mechanics and engineering* 2004; **193**(42):4717–4739.
15. Mackenzie JA, Mekwi WR. An unconditionally stable second-order accurate ALE–FEM scheme for two-dimensional convection–diffusion problems. *IMA Journal of Numerical Analysis* 2012; **32**(3):888–905.
16. Formaggia L, Nobile F. Stability analysis of second-order time accurate schemes for ALE–FEM. *Computer methods in applied mechanics and engineering* 2004; **193**(39):4097–4116.
17. Formaggia L, Nobile F. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East West Journal of Numerical Mathematics* 1999; **7**:105–132.
18. Gastaldi L. A priori error estimates for the arbitrary Lagrangian Eulerian formulation with finite elements. *Journal of Numerical Mathematics* 2001; **9**(2):123–156.
19. Kang YS, Sohn D, Kim JH, Kim HG, Im S. A sliding mesh technique for the finite element simulation of fluid–solid interaction problems by using variable-node elements. *Computers & Structures* 2014; **130**:91–104.
20. Aymone F, Luis J. Mesh motion techniques for the ALE formulation in 3D large deformation problems. *International journal for numerical methods in engineering* 2004; **59**(14):1879–1908.
21. Bar-Yoseph PZ, Mereu S, Chippada S, Kalro VJ. Automatic monitoring of element shape quality in 2-D and 3-D computational mesh dynamics. *Computational Mechanics* 2001; **27**(5):378–395.
22. J Donea JPP A Huerta, Rodriguez-Ferran A. *Encyclopedia of Computational Mechanics*, chap. 14: Arbitrary Lagrangian-Eulerian Methods. John Wiley and Sons, Ltd.: New York, 2004.
23. Farhat C, Degand C, Koobus B, Lesoinne M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer methods in applied mechanics and engineering* 1998; **163**(1):231–245.
24. Johnson AA, Tezduyar TE. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer methods in applied mechanics and engineering* 1994; **119**(1):73–94.
25. Helenbrook BT. Mesh deformation using the biharmonic operator. *International journal for numerical methods in engineering* 2003; **56**(7):1007–1021.
26. Huerta A, Liu WK. Viscous flow with large free surface motion. *Computer Methods in Applied Mechanics and Engineering* 1988; **69**(3):277–324.
27. Masud A. Effects of mesh motion on the stability and convergence of ALE based formulations for moving boundary flows. *Computational Mechanics* 2006; **38**(4-5):430–439.
28. Löhner R, Yang C. Improved ALE mesh velocities for moving bodies. *Communications in numerical methods in engineering* 1996; **12**(10):599–608.
29. Souli M, Ouahsine A, Lewin L. ALE formulation for fluid–structure interaction problems. *Computer methods in applied mechanics and engineering* 2000; **190**(5):659–675.
30. Knupp P, Margolin LG, Shashkov M. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics* 2002; **176**(1):93–128.
31. DeZeeuw D, Powell KG. An adaptively refined Cartesian mesh solver for the Euler equations. *Journal of Computational Physics* 1993; **104**(1):56–68.
32. Bayyuk SA, Powell K, Van Leer B. A simulation technique for 2-D unsteady inviscid flows around arbitrarily moving and deforming bodies of arbitrary geometry. *Ann Arbor* 1993; **1001**:48 109–2140.
33. Pember RB, Bell JB, Colella P, Curtchfield WY, Welcome ML. An adaptive Cartesian grid method for unsteady compressible flow in irregular regions. *Journal of computational Physics* 1995; **120**(2):278–304.
34. Hu X, Khoo B, Adams N, Huang F. A conservative interface method for compressible flows. *Journal of Computational Physics* 2006; **219**(2):553–578.
35. Meinke M, Schneiders L, Günther C, Schröder W. A cut-cell method for sharp moving boundaries in Cartesian grids. *Computers & Fluids* 2013; **85**:135–142.

DOI: 10.1002/nme

36. Schneiders L, Hartmann D, Meinke M, Schröder W. An accurate moving boundary formulation in cut-cell methods. *Journal of Computational Physics* 2013; **235**:786–809.
37. Seo JH, Mittal R. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *Journal of computational physics* 2011; **230**(19):7347–7363.
38. Baiges J, Codina R. The fixed-mesh ALE approach applied to solid mechanics and fluid–structure interaction problems. *International journal for numerical methods in engineering* 2010; **81**(12):1529–1557.
39. Zilian A, Legay A. The enriched space–time finite element method (EST) for simultaneous solution of fluid–structure interaction. *International Journal for Numerical Methods in Engineering* 2008; **75**(3):305–334.
40. Gerstenberger A, Wall WA. An extended finite element method/Lagrange multiplier based approach for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**(19):1699–1714.
41. Lew AJ, Buscaglia GC. A discontinuous-Galerkin-based immersed boundary method. *International Journal for Numerical Methods in Engineering* 2008; **76**(4):427–454.
42. Rangarajan R, Lew A, Buscaglia GC. A discontinuous-Galerkin-based immersed boundary method with non-homogeneous boundary conditions and its application to elasticity. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(17):1513–1534.
43. Ausas RF, Sousa FS, Buscaglia GC. An improved finite element space for discontinuous pressures. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(17):1019–1031.
44. Dolbow J, Harari I. An efficient finite element method for embedded interface problems. *International journal for numerical methods in engineering* 2009; **78**(2):229–252.
45. Hansbo A, Hansbo P. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer methods in applied mechanics and engineering* 2004; **193**(33):3523–3540.
46. Anderson D, McFadden GB, Wheeler A. Diffuse-interface methods in fluid mechanics. *Annual review of fluid mechanics* 1998; **30**(1):139–165.
47. Feng JJ, Liu C, Shen J, Yue P. An energetic variational formulation with phase field methods for interfacial dynamics of complex fluids: Advantages and challenges. *Modeling of soft matter*. Springer, 2005; 1–26.
48. Almgren AS, Bell JB, Colella P, Marthaler T. A Cartesian grid projection method for the incompressible Euler equations in complex geometries. *SIAM Journal on Scientific Computing* 1997; **18**(5):1289–1309.
49. Marella S, Krishnan S, Liu H, Udaykumar H. Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations. *Journal of Computational Physics* 2005; **210**(1):1–31.
50. Udaykumar HS, Mittal R, Rampunggoon P, Khanna A. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics* 2001; **174**(1):345–380.
51. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; **11**(0):479–517.
52. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics* 2000; **161**(1):35–60.
53. Lee J, Kim J, Choi H, Yang KS. Sources of spurious force oscillations from an immersed boundary method for moving-body problems. *Journal of computational physics* 2011; **230**(7):2677–2695.
54. Yang X, Zhang X, Li Z, He GW. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics* 2009; **228**(20):7821–7836.
55. Cortez R, Minion M. The blob projection method for immersed boundary problems. *Journal of Computational Physics* 2000; **161**(2):428–453.
56. Linnick MN, Fasel HF. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics* 2005; **204**(1):157–192.
57. Kim D, Choi H. Immersed boundary method for flow around an arbitrarily moving body. *Journal of Computational Physics* 2006; **212**(2):662–680.
58. Griffith BE, Peskin CS. On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems. *Journal of Computational Physics* 2005; **208**(1):75–105.
59. Xu S, Wang Z. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics* 2006; **216**:454–493.
60. Li Z, Lai MC. The immersed interface method for the Navier–Stokes equations with singular forces. *Journal of Computational Physics* 2001; **171**(2):822–842.
61. Taylor C, Hood P. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers & Fluids* 1973; **1**(1):73–100.
62. Ern A, Guermond JL. *Theory and Practice of Finite Elements*. Springer: New York, 2004.
63. Gawlik ES, Lew AJ. High-order finite element methods for moving-boundary problems with prescribed boundary evolution. *Computer Methods in Applied Mechanics and Engineering* 2014; **278**:314–346.
64. Rangarajan R, Lew AJ. Analysis of a method to parameterize planar curves immersed in triangulations. *SIAM Journal on Numerical Analysis* 2013; **51**(3):1392–1420.
65. Rangarajan R, Lew AJ. Universal meshes: A method for triangulating planar curved domains immersed in nonconforming triangulations. *International Journal for Numerical Methods in Engineering* 2014; **98**(4):236–264.
66. Gawlik ES, Lew AJ. Unified analysis of finite element methods for problems with moving boundaries. *(Submitted)* 2014; .
67. Gawlik ES, Lew AJ. Supercloseness of orthogonal projections onto nearby finite element spaces. *Mathematical Modeling and Numerical Analysis (to appear)* 2014; .
68. Farrell P, Piggott M, Pain C, Gorman G, Wilson C. Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(33):2632–2642.
69. Burrage K, Butcher JC, Chipman FH. An implementation of singly-implicit Runge-Kutta methods. *BIT Numerical Mathematics* 1980; **20**(3):326–340.
70. Hairer E, Wanner G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer: Berlin, 2002.
71. Chorin AJ. Numerical solution of the Navier-Stokes equations. *Mathematics of computation* 1968; **22**(104):745–762.

72. Temam R. Une méthode d'approximation de la solution des équations de Navier-Stokes. *Bulletin de la Société Mathématique de France* 1968; **96**:115–152.
73. Brown DL, Cortez R, Minion ML. Accurate projection methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 2001; **168**(2):464–499.
74. Brenan KE, Campbell SL, Petzold LR. *Numerical solution of initial-value problems in differential-algebraic equations*, vol. 14. SIAM, 1996.
75. Persson PO. High-order LES simulations using implicit-explicit Runge-Kutta schemes. *Proc. of the 49th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2011-684, 2011.
76. Persson PO. High-order Navier-Stokes simulations using a sparse line-based discontinuous Galerkin method. *Proc. of the 50th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2012-456, 2012.
77. Lamb H. *Hydrodynamics*. Cambridge university press, 1993.
78. Bern M, Eppstein D, Gilbert J. Provably good mesh generation. *Journal of Computer and System Sciences* 1994; **48**(3):384–409.
79. Ongoren A, Rockwell D. Flow structure from an oscillating cylinder Part 1. Mechanisms of phase shift and recovery in the near wake. *Journal of Fluid Mechanics* 1988; **191**:197–223.
80. Guilmineau E, Queutey P. A numerical simulation of vortex shedding from an oscillating circular cylinder. *Journal of Fluids and Structures* 2002; **16**(6):773–794.
81. Ying W, Henriquez CS, Rose DJ. Composite backward differentiation formula: an extension of the TR-BDF2 scheme. *Submitted to Applied Numerical Mathematics* 2009; .

A. THE UNIVERSAL MESH MAP

In this section, we detail the construction of the map $\Phi^t : \mathcal{S}_h^n \to \Omega^t$ introduced in Section 3.1. We require the definition of three auxiliary maps: a *boundary evolution map*, a *relaxation map*, and a *blend map*. In what follows, we denote by $\phi^t : \mathcal{D} \to \mathbb{R}$ the signed distance function to $\partial\Omega^t$, taken to be positive outside Ω^t and negative inside Ω^t. We denote by $\pi^t : \mathcal{D} \to \partial\Omega^t$ the closest point projection onto $\partial\Omega^t$.

Auxiliary Maps. The *boundary evolution map* $\gamma^t : \partial\mathcal{S}_h^n \to \partial\Omega^t$ provides a correspondence between the piecewise linear boundary of $\mathcal{S}_h^n$ and the boundary of Ω^t for $t \in (t^{n-1}, t^n]$. It is defined in terms of the closest point projection via

$$\gamma^t = \pi^t \circ \pi^{t^{n-1}}\Big|_{\partial\mathcal{S}_h^n}.$$

The *relaxation map* $\mathfrak{p} : \mathcal{S}_h^n \to \mathcal{S}_h^n$ identifies those vertices that lie both inside Ω^t and near $\partial\Omega^t$, and perturbs them in a direction away from $\partial\Omega^t$. It is defined in terms of the signed distance function via

$$\mathfrak{p}(x) = \begin{cases} x - \delta h \left(1 + \frac{\phi^{t^{n-1}}(x)}{Rh}\right) \nabla_x \phi^{t^{n-1}}(x) & \text{if } -Rh < \phi^{t^{n-1}} < 0 \\ x & \text{otherwise,} \end{cases}$$

with $R > 1$ a small positive integer and $(1 + 1/R)^{-1} \le \delta \le 1$. We denote by $\mathfrak{p}(\mathcal{S}_h^n)$ the triangulation obtained by applying the relaxation $\mathfrak{p}$ to the vertices of $\mathcal{S}_h^n$ while preserving the mesh's connectivity.

The *blend map* ψ^t takes each straight triangle $K \in \mathfrak{p}(\mathcal{T}_{h,2}^{t^{n-1}})$ and deforms it to a curved triangle that conforms exactly to the moving boundary. Letting u, v, w denote the vertices of K, the blend map reads

$$\begin{aligned}\psi^t(x) =& \frac{1}{2(1-\lambda_u)}[\lambda_v \gamma^t(\lambda_u u + (1-\lambda_u)v) + \lambda_u \lambda_w \gamma^t(u)] \\ &+ \frac{1}{2(1-\lambda_v)}[\lambda_u \gamma^t((1-\lambda_v)u + \lambda_v v) + \lambda_v \lambda_w \gamma^t(v)] + \lambda_w w, \end{aligned} \tag{28}$$

where $\lambda_u, \lambda_v, \lambda_w$ are the barycentric coordinates of $x \in K$. Here, we have employed the convention that the vertex w is the unique vertex of K lying inside $\Omega^{t^{n-1}}$.

The Universal Mesh Map. We now define Φ^t over each triangle $K \in \mathcal{S}_h^n$ with vertices u, v, w according to

$$\Phi^t(x) = \begin{cases} \lambda_u \mathfrak{p}(u) + \lambda_v \mathfrak{p}(v) + \lambda_w \mathfrak{p}(w) & \text{if } K \in \mathcal{T}_{h,0}^{t^{n-1}} \\ \lambda_u \gamma^t(u) + \lambda_v \mathfrak{p}(v) + \lambda_w \mathfrak{p}(w) & \text{if } K \in \mathcal{T}_{h,1}^{t^{n-1}} \\ \psi^t(\lambda_u u + \lambda_v v + \lambda_w \mathfrak{p}(w)) & \text{if } K \in \mathcal{T}_{h,2}^{t^{n-1}}, \end{cases} \tag{29}$$

where $\lambda_u, \lambda_v, \lambda_w$ are the barycentric coordinates of $x \in K$. Once again, we have employed the convention that for triangles $K \in \mathcal{T}_{h,2}^{t^{n-1}}$, the vertex w is the unique vertex of K lying inside $\Omega^{t^{n-1}}$, and for triangles $K \in \mathcal{T}_{h,1}^{t^{n-1}}$, the vertex u is the unique vertex of K lying outside $\Omega^{t^{n-1}}$.

Table I. SDIRK(1): Coefficients β_{ij} for a $s = 1$-stage SDIRK scheme of order 1. ($\gamma = 1$)

$i \setminus j$	0
1	1

Table II. SDIRK(2): Coefficients β_{ij} for a $s = 2$-stage SDIRK scheme of order 2. ($\gamma = 1 - \sqrt{2}/2$)

$i \setminus j$	0	1
1	1	
2	$-\sqrt{2}$	$1 + \sqrt{2}$

Table III. SDIRK(3): Coefficients β_{ij} for a $s = 3$-stage SDIRK scheme of order 3. ($\gamma = 0.43586652150845899942$)

$i \setminus j$	0	1	2
1	1.00000000000000000		
2	0.352859819860479140	0.647140180139520860	
3	−1.25097989505606042	3.72932966244456977	−1.47834976738850935

Table IV. SDIRK(4): Coefficients β_{ij} for a $s = 5$-stage SDIRK scheme of order 4. ($\gamma = 1/4$)

$i \setminus j$	0	1	2	3	4
1	1				
2	-1	2			
3	$-\frac{13}{25}$	$\frac{42}{25}$	$-\frac{4}{25}$		
4	$-\frac{4}{17}$	$\frac{89}{68}$	$-\frac{25}{136}$	$\frac{15}{136}$	
5	$\frac{7}{3}$	$-\frac{37}{12}$	$-\frac{103}{24}$	$\frac{275}{8}$	$-\frac{85}{3}$

Isoparametric Approximations. In practical computations, it is convenient to approximate the map Φ^t (and hence the domain Ω^t) with a polynomial interpolant

$$\Phi^t_{\text{approx}}(x) = \sum_a \widetilde{N}_a(x)\Phi^t(\widetilde{X}_a) \tag{30}$$

constructed from shape functions $\widetilde{N}_a$ of a triangular Lagrange element with corresponding degrees of freedom $\widetilde{X}_a$ on the reference triangulation $\mathcal{S}_h^n$. Details are given in [63].

Assembly of the quantities $\mathbf{M}(t)$, $\mathbf{K}(t)$, $\mathbf{b}(\mathbf{u}(t), t)$, and $\mathbf{f}(t)$ at a given time $t \in (t^{n-1}, t^n]$ can then be accomplished by computing the positions $\Phi^t(\tilde{X}_a)$ of the degrees of freedom in the (approximately) conforming mesh $\Phi^t_{\text{approx}}(\mathcal{S}_h^n)$ and following standard practices to compute elementwise contributions over curved isoparametric elements [62]. We adopted this strategy in all of the numerical examples presented in this paper.

B. SINGLY DIAGONALLY IMPLICIT RUNGE KUTTA TIME INTEGRATORS.

Tables I-IV record the coefficients $\gamma > 0$ and $\beta_{ij} \in \mathbb{R}$, $i = 1, 2, \ldots, s$, $j = 0, 1, \ldots, i-1$ for a collection of SDIRK schemes of orders 1 through 4. Note that the structure of the Runge-Kutta stages in Algorithm (3.1) differs from the structure that is most familiar to Runge-Kutta practitioners [70]; see [63, Appendix A] and [81] for details.

DOI: 10.1002/nme