An Operator-Integration-Factor Splitting (OIFS) method for Incompressible Flows in Moving Domains

Argonne Leadership Computing Facility

An Operator-Integration-Factor Splitting (OIFS) method for Incompressible Flows in Moving Domains

prepared by
Saumil S. Patel[1], Paul F. Fischer[2,3], Misun Min[2], and Ananias G. Tomboulides[2,4]
[1]Argonne Leadership Computing Facility, Argonne National Laboratory
[2]Mathematics and Computer Science Division, Argonne National Laboratory
[3]Department of Mechanical Science and Engineering and Department of Computer Science, University of Illinois at Urbana-Champaign
[4]Aristotle University of Thessaloniki, Greece

October 21, 2017

Contents

Abstract

In this paper, we present a characteristic-based numerical procedure for simulating incompressible flows in domains with moving boundaries. Our approach utilizes an operator-integration-factor splitting technique to help produce an efficient and stable numerical scheme. Using the spectral element method and an arbitrary Lagrangian-Eulerian formulation, we investigate flows where the convective acceleration effects are non-negligible. Several examples, ranging from laminar to turbulent flows, are considered. Comparisons with a standard, semi-implicit time-stepping procedure illustrate the improved performance of the scheme.

0.1 Introduction

Predicting the motion of a fluid in a domain with moving boundaries has been one of the major challenges in computational physics. Specific examples of moving boundary problems include free-surface flows, flows in blood vessels, and in-cylinder flows within internal combustion (IC) engines. A variety of numerical algorithms have been proposed to tackle such problems. Most techniques incorporate a Lagrangian description of the boundary [1, 2, 3]. Such schemes may be considered to be in the class of arbitrary Lagrangian-Eulerian (ALE) methods [4], where the entire mesh is updated, perhaps at every timestep, in order to conform to the new boundary while the flow field is solved using an Eulerian approach.

The main feature of moving domain problems is the unsteadiness of the flow field. This feature is apparent in IC engines where multiple mechanisms generate turbulent flow fields [5]. Turbulence in IC engines presents a challenge for computational fluid dynamic (CFD) algorithms, due primarily to the broad range of length and time scales that need to be resolved. Specifically, simulations need to predict the evolution of a variety of flow structures (boundary layers, eddys, vortices, etc.) in the vicinity of complex domains that are moving. Executing these simulations accurately and in a reasonable amount of time can ultimately lead to engine design concepts with improved efficiency.

Advances in high-performance parallel computers, scalable iterative solvers, and high-order discretizations have led to much progress toward direct numerical simulation (DNS) and large eddy simulation (LES) of turbulent flows in IC engines. LES has become a popular method of choice to study turbulence in IC engines [6, 7]. Due to the use of subgrid scale models, LES allows for a lower grid resolution while being able to represent more flow structure (eddies and vortices) than a traditional Reynolds Averaged Navier Stokes (RANS) approach. Progress and reviews of LES models for IC engines can be found in [8]. DNS calculations have also been performed with spectral element methods (SEM), making significant advances in the investigations of cycle-to-cycle flow-field variations [9] and wall heat transfer [10] under engine-relevant conditions.

Since its introduction by Patera [11], several developments have made the spectral element method a powerful tool for the simulation of turbulence in complex geometries. Key advances include ALE formulations based on the $\mathbb{P}_N - \mathbb{P}_{N-2}$ spectral element method for incompressible and low-Mach number flows [10, 12, 13, 14, 15]; stable fluid-structure interaction (FSI) solution strategies[15]; high-order operator splitting strategies that lead to decoupled linear symmetric positive-definite subproblems at each timestep [16, 17, 18, 19, 20]; fast multilevel preconditioners [21, 22, 23] coupled with scalable parallel coarse grid solvers [24, 25, 26]; stable formulations for the convective operator [27, 28, 29]; and high-performance implementations [30].

The focus of this work is to build upon the current state of ALE formulations by incorporating a time-integration scheme that can solve the incompressible Navier-Stokes equations faster than standard techniques. Using the $\mathbb{P}_N - \mathbb{P}_N$ spectral element method [15, 31], we develop a scheme whose aim is to resolve turbulent flows with prescribed boundary motion. The specific application is for the flow generated due to piston and intake valve motion in IC engines.

The performance of the engine relies heavily on how well turbulence can promote large and small-scale mixing between the fuel and air prior to combustion [5]. Indeed, Schiffman et.al [7] have indicated that the early phase of the intake "jet" flow in an IC engine is critical to the evolution of the flow within a combustion chamber (or cylinder). Their results indicate that flow through the small intake valve region required refined grids for accuracy. This requirement came at a substantial computational cost, resulting in simulations that ran for three times longer than less-resolved grids [7].

The standard approach to efficient simulation of turbulent flow is to treat the nonlinear terms (i.e. convection operator) explicitly in time. Explicit treatment of the convection operator requires a timestep restriction of $\Delta t = O(\Delta x / |U|)$ to ensure stability, where $|U|$ and Δx are respectively the characteristic sizes of the velocity and grid spacing. For moving boundary problems, such as IC engines, the mesh requires an update procedure that causes Δx to change over time. During early moments of the intake stroke, a small gap exists in the valve region (in reality the valve is closed where the spacing has zero volume). This translates to

extremely small values of Δx at the early and final stages of the intake stroke. Thus the stability requirement $\Delta t = O(\Delta x)$ becomes overly constraining for IC engines.

In response to this computational constraint, we consider a characteristic-based scheme for moving domain problems. In particular, we apply the operator-integration-factor-splitting (OIFS) scheme of Maday *et al.* [17] to the ALE method. We seek to overcome the constraints of the standard time-stepping approach described above by decoupling the convection (nonlinear) and diffusion (linear) operators. This is accomplished by writing the inertial term as a material derivative and discretizing it using kth-order backward finite-differences BDFk. Specific terms in the BDFk, known as "feet" of the characteristic, are determined via a tailored form of the OIFS scheme that is solved with the Runge-Kutta 4th-order time-integration scheme. As a result, solving for the linear, implicit Stokes term occurs less often and augments stability for a given mesh spacing Δx.

This paper is organized as follows. Section 2 describes the governing equations and our OIFS scheme on a static and moving domain, provided with details on the formulation of the Runge-Kutta time integration scheme. Section 3 presents our computational results that include convergence studies for the OIFS scheme and several fluid dynamic problems, ranging from laminar to turbulent flows. We demonstrate the speedup of the OIFS scheme compared to a standard, semi-implicit time-stepping procedure. Section 4 provides our conclusion.

0.2 Characteristic-Based Formulation for the Incompressible Navier-Stokes Equations

We consider unsteady incompressible flow in a given computational domain $\Omega(t)$ at time t governed by the Navier-Stokes equations,

$$\frac{\partial \mathbf{u}}{\partial t} \;+\; \mathbf{u} \cdot \nabla \mathbf{u} \;=\; -\nabla p \;+\; \tfrac{1}{Re}\nabla^2 \mathbf{u}, \qquad \nabla \cdot \mathbf{u} = 0, \tag{1}$$

subject to prescribed velocity conditions on the domain boundary, $\partial\Omega(t)$. Here, $\mathbf{u}(\mathbf{x}, t) = (u_1, u_2, u_3)$ represents the fluid velocity components as a function of space $\mathbf{x} = (x_1, x_2, x_3)$ and time t; p is the pressure field; and $Re = L_0 U_0 / \nu_0$ is the Reynolds number based on a characteristic length scale L_0, velocity scale U_0, and kinematic viscosity of the fluid ν_0. We are interested in moving-geometry simulations where the motion of the domain boundary $\partial\Omega(t)$, is prescribed.

To motivate our formulation, we briefly summarize the standard numerical approach to solve the time-dependent Navier-Stokes equations. In particular, a semi-implicit procedure is used to discretize (1) where the symmetric Stokes operator is handled implicitly and the convection term is treated explicitly. This can be accomplished in a variety of ways. For example, one common approach is to approximate the time derivative at t^n with a kth-order backward difference formula (BDFk) while the nonsymmetric convection term is approximated at t^n by an extrapolation of order k (EXTk). Specific details on the EXTk/BDFk approach, as they apply to spectral element methods, can be found in [15, 31]. A consequence of this approach is that the stability of the scheme is determined by the explicit treatment of the convection term with the timestep restriction of $\Delta t = O(\Delta x_{min}/|U_0|)$, where Δx_{min} is the smallest grid spacing in an unstructured mesh. For spectral element methods, Δx_{min} scales as $O(N^{-2})$ [31] where N is the polynomial order of the spectral element basis functions. This has dramatic implications for moving-domain problems, such as IC engines [7], where regions exist with small gaps and large flow velocities. As a result, a severe timestep resctriction increases the time-to-solution.

In order to overcome these challenges, we present a characteristic-based ALE formulation, effectively decoupling the convection and Stokes operators, that allows a larger Courant-Friedrichs-Lewy (CFL) number than the conventional semi-implicit schemes. Our approach is extended from the previous work by Ho and collaborators [12, 13, 14] on static domain to moving domain problems.

0.2.1 Characteristic Formulation on a Static Domain

The key idea of our characteristic approach for solving flow problems in a moving mesh can be fully described using a simple convection-diffusion equation as a simplified model of the Navier-Stokes equations (1). We begin with revisiting the formulation on a static domain and extend the same approach to a moving domain. We consider a scalar convection-diffusion equation for $u = u(\mathbf{x}, t)$ defined as

$$\frac{\partial u}{\partial t} + \mathbf{c} \cdot \nabla u = \frac{1}{Pe} \nabla^2 u, \tag{2}$$

where $Pe = L_0 U_0 / \alpha_0$ is the Peclet number, α_0 is the diffusivity, and $\mathbf{c}(\mathbf{x}, t)$ is a prescribed advecting field.

In the characteristic approach [32], we formally replace the left-hand side of (2) with the material derivative

$$\frac{Du}{Dt} := \frac{\partial u}{\partial t} + \mathbf{c} \cdot \nabla u, \tag{3}$$

to rewrite (2) as

$$\frac{Du}{Dt} = \frac{1}{Pe} \nabla^2 u. \tag{4}$$

From here, we apply a BDFk scheme to (4). For example, if we choose $k = 2$, we may write (4) in the following implicit semi-discrete form,

$$\frac{3u^n - 4\tilde{u}^{n-1} + \tilde{u}^{n-2}}{2\Delta t} = \frac{1}{Pe} \nabla^2 u^n, \tag{5}$$

where $u^n = u(\mathbf{x}, t^n)$ is the solution at a gridpoint $\mathbf{x}$ and at a time $t^n = n\Delta t$. In (5), $\tilde{u}^{n-q} := u(X, t^{n-q})$ represents the value of u at an earlier time, t^{n-q} and evaluated at the foot of the characteristic, $\mathbf{X}(t^{n-q}; \mathbf{x})$ that ends at $\mathbf{x}$. Specifically,

$$\mathbf{X}(\tau; \mathbf{x}) := \mathbf{x} + \int_{t^n}^{\tau} \mathbf{c}(\mathbf{X}, s) \, ds, \tag{6}$$

which determines the foot of the characteristic when $\tau < t^n$ and returns $\mathbf{X} = \mathbf{x}$ when $\tau = t^n$. We note in particular that (5) holds at any gridpoint $\mathbf{x}_j$ where the field values and their derivatives (e.g., the right-hand side of (5)) are known or readily computed. For each gridpoint, there is a corresponding foot,

$$\mathbf{X}_j^{n-q} := \mathbf{X}(t^{n-q}; \mathbf{x}_j), \tag{7}$$

with $\mathbf{X}_j^n = \mathbf{x}_j$. Evaluation of the left-hand side of (5) at $\mathbf{x}_j$ requires u_j^n as well as the *values* of u along the characteristic

$$\tilde{u}_j^{n-q} := u^{n-q}(\mathbf{X}_j^{n-q}). \tag{8}$$

Figure 1 illustrates values of u along the characteristic with corresponding points $\mathbf{X}_j^{n-q}$ along the characteristic that emanates from gridpoint $\mathbf{x}_j$.

Determining the value of u at the foot of the characteristics becomes the goal of our scheme. Nominally, this determination is difficult because the $\mathbf{X}_j^{n-q}$ are off-grid positions. As Pirroneau [32] suggests, evaluation of $\tilde{u}^{n-q}$ requires interpolation for u^{n-q} and for the vector field, $\mathbf{c}$, in order to accurately march back along the trajectory. This procedure is thus fairly expensive, particularly for high-order spatial discretizations, as the cost of off-grid interpolation for N-th order elements in d space dimensions has a complexity of $O(N^d)$ per interpolated value (i.e., $O(EN^{2d})$ for E elements of order N).
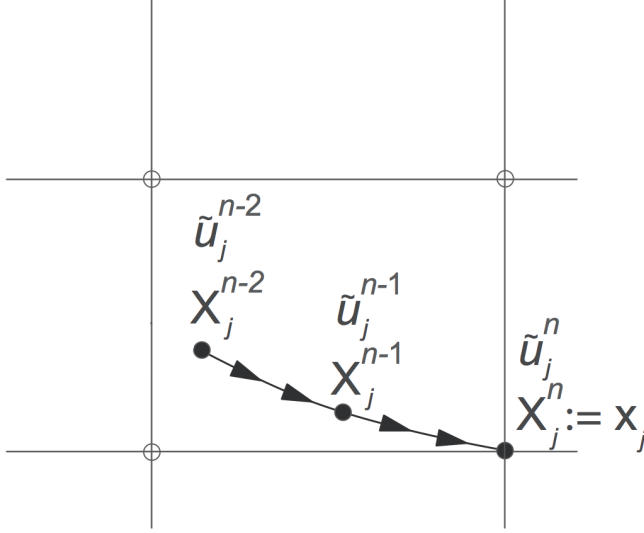
3

Figure 1: BDF2 approximation of the material derivative along the characteristic emanating from $\mathbf{x}_j$. $\mathbf{X}_j^{n-q}$ is the foot of the characteristic corresponding to $t^n - q\Delta t$

.

Thanks to the work of Maday, Patera, and Rønquist [17], expensive off-grid interpolations can be avoided through the application of the operator-integration-factor-splitting (OIFS) scheme. The OIFS scheme can be considered to be a variant of the characteristic scheme (5) where knowledge of $\mathbf{X}^{n-q}$ is not necessary in order to evaluate $\tilde{u}^{n-q}$. Rather, one solves a sequence of initial-value problems:

$$
\begin{aligned}
\frac{\partial \tilde{u}^{n-q}(\mathbf{x}, s)}{\partial s} + \mathbf{c} \cdot \nabla \tilde{u}^{n-q}(\mathbf{x}, s) &= 0, \quad s \in [t^{n-q}, t^n], \\
\tilde{u}^{n-q}(\mathbf{x}, s^{n-q}) &= u(\mathbf{x}, t^{n-q}),
\end{aligned}
\tag{9}
$$

$q = 1, \ldots, k$, that correspond to the pure convection problems. The net result of solving (9) is that the initial condition simply moves forward along the characteristic (or propagating path) of the convecting field, $\mathbf{c}$. After time $q\Delta t$, the values of $\tilde{u}^{n-q}$ at $(\mathbf{x}, t^n)$ are the desired values from the foot of the characteristics.

$$
\tilde{u}^{n-q}(\mathbf{x}, t^n) := u(\mathbf{X}^{n-q}, t^{n-q}),
\tag{10}
$$

which is exactly what is needed to solve (5). We note that no interpolation is required to find $\tilde{u}^{n-q}$. The OIFS scheme (9) effectively decouples the solution of the convection and diffusion terms in (2). As a result, the calculation of (9) amounts to a sub-integration of the convection term with a sub-timestep, $\Delta s \leq \Delta t$, which will be stable with explicit integration as long as Δs satisfies the Courant-Friedrichs-Lewy (CFL) stability constraint. (See, e.g., [31] for CFL considerations in the context of spectral elements.) Conversely, thanks to the semi-Lagrangian formulation, (5) is implicit, and Δt is not subject to CFL stability constraints.

Spatial discretization of (5) is based on the weighted residual formulation: *Find $u \in X_0^N$ such that*

$$
\frac{3}{2\Delta t}(v, u^n) - \frac{4}{2\Delta t}(v, \tilde{u}^{n-1}) + \frac{1}{2\Delta t}(v, \tilde{u}^{n-2}) = -\frac{1}{Pe}(\nabla v, \nabla u^n)
\tag{11}
$$

for all test functions $v \in X_0^N$. Here, $X_0^N \subset H^1(\Omega)$ is a finite-dimensional subset of the usual Sobolev space of square-integrable functions that vanish on $\partial\Omega$ and whose first derivatives are also square integrable on Ω [31]. Furthermore, in (11), we have introduced the $\mathcal{L}^2$ inner product, $(f, g) := \int_\Omega fg \, d\mathbf{x}$. Upon the insertion of an

appropriate nodal basis for u into (11), the fully discrete convection-diffusion equation can be given by the following linear system:

$$\left(\frac{1}{Pe}A + \frac{3}{2\Delta t}B\right)\underline{u}^n = B\underline{f}^n,\tag{12}$$

where A is the the stiffness matrix, B is the (diagonal) mass matrix, and the right-hand side is

$$\underline{f}^n = \left(\frac{4}{2\Delta t}\underline{\tilde{u}}^{n-1} - \frac{1}{2\Delta t}\underline{\tilde{u}}^{n-2}\right).\tag{13}$$

Here $\underline{\tilde{u}}^{n-q}$ represents the SE solution for the scalar field, $\tilde{u}$. Additional details on these matrix operators for the spectral element method can be found in [31].

The right-hand side values, $\underline{\tilde{u}}^{n-q}$ are determined by solving (9) with the weighted residual formulation, *Find* $\tilde{u} \in X_0^N$ *such that*

$$\frac{d}{ds}(v, \tilde{u}) = -(v, \mathbf{c} \cdot \nabla\tilde{u}) \ \forall \ v \in X_0^N,\tag{14}$$

and initial conditions given in (9). The matrix form of (14) is

$$B\frac{d\underline{\tilde{u}}}{ds} = -C\underline{\tilde{u}},\tag{15}$$

where C is the convection matrix representing the discrete form of $\mathbf{c} \cdot \nabla$ and can be written in the convective, conservative, or skew-symmetric form [31] and can be evaluated either in aliased or dealiased form [29]. (Our preference is the dealiased convective form.)

For each subproblem q, the respective initial conditions and final results are

$$\begin{aligned}\underline{\tilde{u}}^{n-q,0} &= \underline{u}^{n-q} \\ \underline{\tilde{u}}^{n-q} &= \underline{\tilde{u}}^{n-q,qM},\end{aligned}$$

$q = 1, \ldots, k$. The classical RK4 is used to integrate (15) with $\Delta s = \frac{\Delta t}{M}$, where $M \geq 1$ is the number of substeps. For each BDF term, $q = 1, ..., k$, and substep $m = 0, \ldots, qM - 1$, we have

$$\underline{\tilde{u}}^{n-q,m+1} = \underline{\tilde{u}}^{n-q,m} + \frac{\Delta s}{6}(\underline{g}_0 + 2\underline{g}_1 + 2\underline{g}_2 + \underline{g}_3),\tag{16}$$

with RK4 stages

$$\begin{aligned}\underline{g}_0 &= \mathbb{F}(t^{n-q} + m\Delta s, \underline{\tilde{u}}^{n-q,m}) \\ \underline{g}_1 &= \mathbb{F}(t^{n-q} + [m + \tfrac{1}{2}]\Delta s, \underline{\tilde{u}}^{n-q,m} + \frac{\Delta s}{2}\underline{g}_0) \\ \underline{g}_2 &= \mathbb{F}(t^{n-q} + [m + \tfrac{1}{2}]\Delta s, \underline{\tilde{u}}^{n-q,m} + \frac{\Delta s}{2}\underline{g}_1) \\ \underline{g}_3 &= \mathbb{F}(t^{n-q} + [m + 1]\Delta s, \underline{\tilde{u}}^{n-q,m} + \Delta s\underline{g}_2).\end{aligned}$$

Here, the function is

$$\mathbb{F}(t^n, \underline{\tilde{u}}^{n,m}) := B^{-1}C\underline{\tilde{u}}^{n,m},\tag{17}$$

where the superscript m refers to the mth sub-timestep within the qth subproblem (9). The advantage of an effective high-order diagonal mass matrix, such as provided by the SEM (or, in block form, by discontinuous Galerkin methods), is evident given that (17) requires solution of systems in B for each stage. A temporal
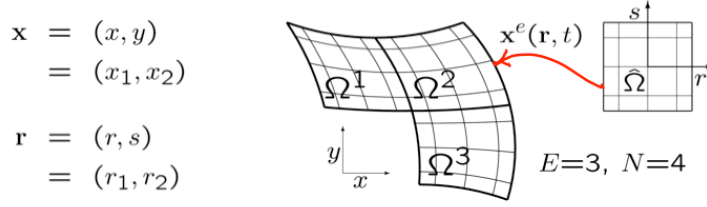
Figure 2: Two-dimensional illustration of a spectral-element domain decomposition.

superscript is not required for the mass and convection matrices since the domain is coincident with the static mesh, Ω. For applications with a moving domain, the solution of (9) requires a careful treatment in order to account for the motion of the gridpoints, $\mathbf{x}_j$.

As it stands, the OIFS scheme has a complexity that is quadratic in the temporal order, k, for there are k integrations to be performed over time intervals ranging from Δt to $k\Delta t$ in length. This complexity can be reduced to $O(k)$ by recognizing that (9) is *linear* and that the $\underline{\tilde{u}}^{n-q}$ appear as a linear combination in (5). Therefore superposition can be used to compute the k solutions in a single pass. We replace (9) with the following system: For $q = k, k-1, \ldots, 1$:

$$\frac{\partial \psi^q}{\partial s} + \mathbf{c} \cdot \nabla \psi^q = 0, \qquad s \in [t^{n-q}, t^{n-q+1}] \tag{18}$$
$$\psi^q(\mathbf{x}, s^{n-q}) = \psi^q(\mathbf{x}, s^{n-q}) + \beta_q \tilde{u}(\mathbf{x}, s^{n-q}).$$

With $\psi^{k+1}(\mathbf{x}, s^{n-k}) := 0$, the final result is

$$\psi^1(\mathbf{x}, t^n) = \sum_{q=1}^{k} \beta_q \tilde{u}^{n-q}, \tag{19}$$

which is the desired contribution to the right-hand side of (12). (For simplicity, we continue with (9) in the sequel, but use the form (18) in practice.)

0.2.2 Characteristic/OIFS Formulation on a Moving Domain

In this section, we develop the characteristic/OIFS formulation for a moving domain, $\Omega := \Omega(t)$, with boundary, $\partial\Omega := \partial\Omega(t)$. Specifically, for the spectral element method, $\Omega(t) = \bigcup_e \Omega^e(t)$, where each element is represented by an invertible map $\mathbf{x}^e(\mathbf{r}, t)$, where $\mathbf{r} \in \hat{\Omega} := [-1, 1]^d$ and d is the number of space dimensions. Such a decomposition is illustrated in Fig. 2 for $d=2$. Similar to the static case, our goal is to determine $\underline{\tilde{u}}^{n-q}$; however, we now develop a weak formulation to the subproblem (9) for moving meshes.

We begin by examining the temporal derivative of (9) on a moving domain:

$$\frac{d}{dt}\left(\int_{\Omega(t)} \tilde{u}(\mathbf{x}, t)v(\mathbf{x}, t)d\mathbf{x}\right)$$
$$= \int_{\hat{\Omega}} v(\mathbf{r})\frac{d}{dt}\left(\tilde{u}(\mathbf{r}, t)J(\mathbf{r}, t)\right) d\mathbf{r}$$
$$= \int_{\hat{\Omega}} v(\mathbf{r})\left(\frac{\partial \tilde{u}}{\partial t} + \mathbf{w} \cdot \nabla \tilde{u}\right) J(\mathbf{r}, t)d\mathbf{r} + \int_{\hat{\Omega}} v\tilde{u}\frac{d}{dt}J(\mathbf{r}, t)d\mathbf{r}, \tag{20}$$

6

where $d\mathbf{r} = dr_1 dr_2 dr_3$ and v is a properly chosen test function such that $v(\mathbf{x}, t) \in H^1(\Omega(t))$ with $\mathbf{x} = \mathbf{x}(\mathbf{r}, t)$. We note that the test functions v are functions of time as a result of the motion of $\Omega(t)$. In practice, however, all integrals are evaluated in a fixed reference frame and they are stationary basis functions in this frame, integrated against the time-evolving functions with the appropriate Jacobian, J. As a result, $v(\mathbf{x}, t)$ on Ω becomes $v(\mathbf{r})$ in Eq. (20) with no time dependency on the reference domain $\hat{\Omega}$. The *mesh velocity* $\mathbf{w} = (w_1, w_2, w_3)$ is defined by

$$\mathbf{w} = \frac{d\mathbf{x}}{dt}. \tag{21}$$

The material derivative of the last term in (20) is further simplified as

$$\frac{dJ(\mathbf{r}, t)}{dt} = \epsilon_{ijk} \left(\frac{d}{dt} \left(\frac{\partial x_i}{\partial r_1} \right) \frac{\partial x_j}{\partial r_2} \frac{\partial x_k}{\partial r_3} + \frac{\partial x_i}{\partial r_1} \frac{d}{dt} \left(\frac{\partial x_j}{\partial r_2} \right) \frac{\partial x_k}{\partial r_3} + \frac{\partial x_i}{\partial r_1} \frac{\partial x_j}{\partial r_2} \frac{d}{dt} \left(\frac{\partial x_k}{\partial r_3} \right) \right),$$

where $J(\mathbf{r}, t) = \epsilon_{ijk} \frac{\partial x_i}{\partial r_1} \frac{\partial x_j}{\partial r_2} \frac{\partial x_k}{\partial r_3}$. From the fact that differentiation can be interchanged, we have $\frac{d}{dt} \left(\frac{\partial x_i}{\partial r_j} \right) = \frac{\partial}{\partial r_j} \left(\frac{dx_i}{dt} \right) = \frac{\partial}{\partial r_j} (w_i) = \frac{\partial w_i}{\partial x_m} \left(\frac{\partial x_m}{\partial r_j} \right)$ which gives

$$\frac{dJ(\mathbf{r}, t)}{dt} = (\nabla \cdot \mathbf{w}) J(\mathbf{r}, t). \tag{22}$$

Additional details on the derivation of Eq. (22) can be found in the appendix. The final form of (20) can be further simplified as

$$\frac{d}{dt} \left(\int_{\Omega(t)} \tilde{u}(\mathbf{x}, t) v(\mathbf{x}, t) d\mathbf{x} \right)$$
$$= \int_{\hat{\Omega}} v(\mathbf{r}) \left(\frac{\partial \tilde{u}}{\partial t} + \mathbf{w} \cdot \nabla \tilde{u} \right) J(\mathbf{r}, t) d\mathbf{r} + \int_{\hat{\Omega}} v\tilde{u}(\nabla \cdot \mathbf{w}) J(\mathbf{r}, t) d\mathbf{r}. \tag{23}$$

Using the inner product notation, we can rewrite (23) as

$$\frac{d}{dt}(v, \tilde{u})_{\Omega(t)} = (v, \frac{\partial \tilde{u}}{\partial t})_{\Omega(t)} + (v, \mathbf{w} \cdot \nabla \tilde{u})_{\Omega(t)} + (v, (\nabla \cdot \mathbf{w})\tilde{u})_{\Omega(t)}. \tag{24}$$

Now, consider the weak formulation of (9) with time variable, t:

$$(v, \frac{\partial \tilde{u}}{\partial t})_{\Omega(t)} + (v, \mathbf{c} \cdot \nabla \tilde{u})_{\Omega(t)} = 0. \tag{25}$$

Substituting (25) into (24), the weak formulation becomes

$$\frac{d}{dt}(v, \tilde{u})_{\Omega(t)} + (v, \mathbf{c} \cdot \nabla \tilde{u})_{\Omega(t)} - (v, \mathbf{w} \cdot \nabla \tilde{u})_{\Omega(t)} - (v, \tilde{u}(\nabla \cdot \mathbf{w}))_{\Omega(t)} = 0, \tag{26}$$

or simply written as

$$\frac{d}{dt}(v, \tilde{u})_{\Omega(t)} + (v, (\mathbf{c} - \mathbf{w}) \cdot \nabla \tilde{u})_{\Omega(t)} - (v, (\nabla \cdot \mathbf{w})\tilde{u})_{\Omega(t)} = 0. \tag{27}$$

In contrast to Eq. (14), Eq. (27) includes a relative velocity $\mathbf{c} - \mathbf{w}$ and a term that accounts for the divergence of the mesh velocity. For a static mesh or volume conserving mesh velocities (i.e. $\nabla \cdot \mathbf{w} = 0$), the last term in Eq. (27) vanishes.

7

0.2.3   Runge-Kutta Time Integration on a Moving Domain

Similar to the static case, we employ the explicit RK4 method to integrate (27). Upon applying the spectral element spatial discretization to (27), we arrive at the following semi-discrete representation:

$$\frac{d\underline{\Phi}}{ds} = \mathbb{F}(s, \underline{\Phi}).\tag{28}$$

where $\underline{\Phi} = B\underline{\tilde{u}}$. Because we are concerned with the subproblem of determining the value at the foot of the characteristic, we revert to the time variable s. Setting $\Delta s = \frac{\Delta t}{M}$ (with $M \geq 1$ and integer), the RK4 scheme becomes

$$\underline{\tilde{\Phi}}^{n-q,m+1} = \underline{\tilde{\Phi}}^{n-q,m} + \frac{\Delta s}{6}(\underline{K}_0 + 2\underline{K}_1 + 2\underline{K}_2 + \underline{K}_3), \quad q = 1,...,k,\tag{29}$$

where

$$\begin{aligned}
\underline{K}_0 &= \mathbb{F}(t_{m_0}, \underline{\tilde{\Phi}}^{t_{m_0}})\\
&= C_w{}^{t_{m_0}}\underline{\tilde{u}}^{t_{m_0}} - D_w^{t_{m_0}}\underline{\tilde{u}}^{t_{m_0}}\\
\underline{\tilde{\Phi}}^{t_{m_0}} &:= (B\underline{\tilde{u}})^{t_{m_0}}; \; t_{m_0} := t^{n-q,m} := t^{n-q} + m\Delta s
\end{aligned}$$

$$\begin{aligned}
\underline{K}_1 &= \mathbb{F}(t_{m_1}, \underline{\tilde{\Phi}}^{t_{m_1}})\\
&= C_w{}^{t_{m_1}}\underline{\tilde{u}}^{t_{m_1}} - D_w^{t_{m_1}}\underline{\tilde{u}}^{t_{m_1}}\\
\underline{\tilde{\Phi}}^{t_{m_1}} &:= (B\underline{\tilde{u}})^{t_{m_1}} := \underline{\tilde{\Phi}}^{t_{m_0}} + \frac{\Delta s}{2}\underline{K}_0; \; t_{m_1} := t^{n-q} + [m + \frac{1}{2}]\Delta s
\end{aligned}$$

$$\begin{aligned}
\underline{K}_2 &= \mathbb{F}(t_{m_2}, \underline{\tilde{\Phi}}^{t_{m_2}})\\
&= C_w{}^{t_{m_2}}\underline{\tilde{u}}^{t_{m_2}} - D_w^{t_{m_2}}\underline{\tilde{u}}^{t_{m_2}}\\
\underline{\tilde{\Phi}}^{t_{m_2}} &:= (B\underline{\tilde{u}})^{t_{m_2}} := \underline{\tilde{\Phi}}^{t_{m_0}} + \frac{\Delta s}{2}\underline{K}_1; \; t_{m_2} := t^{n-q} + [m + \frac{1}{2}]\Delta s
\end{aligned}$$

$$\begin{aligned}
\underline{K}_3 &= \mathbb{F}(t_{m_3}, \underline{\tilde{\Phi}}^{t_{m_3}})\\
&= C_w{}^{t_{m_3}}\underline{\tilde{u}}^{t_{m_3}} - D_w^{t_{m_3}}\underline{\tilde{u}}^{t_{m_3}}\\
\underline{\tilde{\Phi}}^{t_{m_3}} &:= (B\underline{\tilde{u}})^{t_{m_3}} := \underline{\tilde{\Phi}}^{t_{m_0}} + \frac{\Delta s}{2}\underline{K}_2; \; t_{m_3} := t^{n-q} + [m + 1]\Delta s,
\end{aligned}$$

with $\underline{\tilde{u}}^{n-q,m=0} = \underline{u}^{n-q}$ and $\underline{u}^{n-q} = \underline{\tilde{u}}^{n-q,qM}$ for $m = 0,...,qM-1$. This implementation of the RK4 procedure now reflects temporal change in the mass and convection matrices. C_w represents the convection matrix which reflects the relative velocity, $\mathbf{c} - \mathbf{w}$. D_w is the divergence operator on the mesh velocity. We note that the mass matrices, B, must be determined at each stage of the RK substep in order to reflect the updated geometry.

Finally, to solve the convection-diffusion equation in the moving domain, a weak formulation is applied to (5). The fully discrete form is given by

$$\left(\frac{1}{Pe}A^n + \frac{3}{2\Delta t}B^n\right)\underline{u}^n = \underline{\Phi}^n,\tag{30}$$

where (using the preceding notation in Eq. (30))

$$\underline{\Phi}^n = \frac{4\underline{\tilde{\Phi}}^{n-1} - \underline{\tilde{\Phi}}^{n-2}}{2\Delta t},\tag{31}$$

8

and

$$\tilde{\underline{\Phi}}^{n-q} = \tilde{\underline{\Phi}}^{n-q,qM} = B^{n-q}\tilde{\underline{u}}^{n-q}. \tag{32}$$

Note that the stiffness matrix in (30) is determined at the desired time level, n.

0.2.4 Extension to the Navier-Stokes equation

In this section, we apply the ALE/OIFS method for moving domains to the unsteady Navier-Stokes equations (1). Following the procedure for the convection-diffusion equation, we rewrite the convection term as the material derivative as

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u}, \qquad \nabla \cdot \mathbf{u} = 0, \qquad \text{in } \Omega(t), \tag{33}$$

and we apply BDFk to (33) that results in

$$\frac{\beta_0}{\Delta t}\mathbf{u}^n + \frac{1}{\Delta t}\sum_{j=1}^{k}\beta_j\tilde{\mathbf{u}}^{n-j} = -\nabla p^n + \frac{1}{Re}\nabla^2\mathbf{u}^n, \qquad \nabla \cdot \mathbf{u}^n = 0, \tag{34}$$

where β_js are the standard coefficients for BDFk derived from polynomial interpolation through timepoints $[t^n, t^{n-1}, \ldots, t^{n-k}]$.

Here we apply the OIFS approach to determine the values at the foot of the characteristic, $\tilde{\mathbf{u}}^{n-j}$, by solving the following convective initial-value subproblems,

$$\begin{aligned}
\frac{\partial\tilde{\mathbf{u}}}{\partial s} + \mathbf{u} \cdot \nabla\tilde{\mathbf{u}} &= 0, \quad s \in [t^{n-j}, t^n], \\
\tilde{\mathbf{u}}(\mathbf{x}, t^{n-j}) &= \mathbf{u}(\mathbf{x}, t^{n-j}).
\end{aligned} \tag{35}$$

In solving (35), one treats $\tilde{\mathbf{u}}$ as a passive vector field. This field is convected by $\mathbf{u}$, which is the solution to (34). Values of $\mathbf{u}$ on the interval $[t^{n-j}, t^n]$ are computed by interpolating from the known fields $\mathbf{u}^{n-j}, \mathbf{u}^{n-j+1}, ..., \mathbf{u}^{n-1}$. In contrast to the convection-diffusion equation, the convecting field $\mathbf{u}^n$ is not known, so $\mathbf{u}$ is effectively extrapolated on $(t^{n-j}, t^n]$.

Integration of (34) proceeds via a fractional step-method and thereby decouples the nonlinear, divergence-free, and viscous terms into separate subproblems. It starts with

$$\mathbf{u}^* = \sum_{j=0}^{k}\beta_j\tilde{\mathbf{u}}^{n-j}, \tag{36}$$

where the values of $\tilde{\mathbf{u}}^{n-j}$ are determined by the OIFS scheme of (35). Next, we apply the divergence operation to the momentum equation of (34) to arrive at the pressure Poisson equation

$$-\nabla \cdot (\nabla p^n) = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} - \frac{1}{Re}\nabla \cdot (\nabla^2\mathbf{u}^n). \tag{37}$$

The pressure and velocity at t^n in (37) are decoupled using subsitutions found in [31, 33]. As a result, (37) can be rewritten as

$$-\nabla \cdot (\nabla p^n) = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} - \frac{1}{Re}\nabla \cdot (\sum_{j=1}^{k}\alpha_j\nabla \times \omega^{n-j}), \tag{38}$$

where $\omega = \nabla \times \mathbf{u}$ is the vorticity and α_j are the coefficients associated with a k-th order extrapolation procedure. Additional details on the boundary conditions for (38) can also be found in [18]. In this work, (38) is solved using the projection techniques of Fischer [34]. The final step requires the solution of

$$-\frac{1}{Re}\nabla^2\mathbf{u}^n + \frac{\beta_k}{\Delta t}\mathbf{u}^n = \frac{\mathbf{u}^{**}}{\Delta t}, \tag{39}$$

where $\mathbf{u}^{**}$ is an incompressible velocity field given by

$$\mathbf{u}^{**} = \mathbf{u}^* - \Delta t \nabla p^n. \tag{40}$$

In practice, Eq. (34) is solved via (36)-(40) where the spatial discretization is performed using a weighted residual formulation for each subproblem. To illustrate the method, we apply the weighted residual formulation for $k = 2$ to (34), which becomes: *Find* $(\mathbf{u}, p) \in X_b^N(\Omega(t)) \times Y^N(\Omega(t))$ *such that*

$$\frac{3}{2\Delta t}(\mathbf{v}, \mathbf{u}^n)_{\Omega(t)} - \frac{4}{2\Delta t}(\mathbf{v}, \tilde{\mathbf{u}}^{n-1})_{\Omega(t)} + \frac{1}{2\Delta t}(\mathbf{v}, \tilde{\mathbf{u}}^{n-2})_{\Omega(t)}$$
$$= (\nabla \cdot \mathbf{v}, p^n)_{\Omega(t)} - \frac{1}{Re}(\nabla\mathbf{v}, \nabla\mathbf{u}^n)_{\Omega(t)},$$

$$(\nabla \cdot \mathbf{u}^n, q)_{\Omega(t)} = 0, \tag{41}$$

for all test functions $(\mathbf{v}, q) \in X_0^N(\Omega(t)) \times Y^N(\Omega(t))$. Here, we employ a discretization where the velocity and pressure have the same polynomial order. This is the so-called $\mathbb{P}_N - \mathbb{P}_N$ method [15, 31], where $X^N(\Omega(t)) \subset H^1(\Omega(t))$ is the set of continuous Nth-order spectral element basis functions described in [31].

X_b^N is the subset of X^N satisfying the Dirichlet conditions on $\partial\Omega(t)$; X_0^N is the subset of X^N satisfying homogeneous Dirichlet conditions on $\partial\Omega(t)$; Y^N is the space of continuous SE basis functions of degree N; and H^1 is the usual Sobolev space of functions that are square integrable on $\Omega(t)$, whose derivatives are also square integrable.

The $\tilde{\mathbf{u}}^{n-j}$ are obtained by the weighted residual form for (35) given by an analysis similar to that in Sec. 0.2.2. Using index notation for the velocity components, we give the weak form as follows: Find $\tilde{u}_i^{n-j} \in X_b^N(\Omega(t))$ such that

$$\frac{d}{dt}(\tilde{u}_i^{n-j}, v)_{\Omega(t)} + ((\mathbf{u} - \mathbf{w}) \cdot \nabla\tilde{u}_i^{n-j}, v)_{\Omega(t)} - (\tilde{u}_i^{n-j}(\nabla \cdot \mathbf{w}), v)_{\Omega(t)} = 0, \tag{42}$$

for all test functions $v \in X_0^N(\Omega(t))$. The temporal integration of (42) is given by the RK4 scheme on moving domains (see Sec. 0.2.3).

0.3 Spectral Element Method

Here, we describe the spectral element bases, operator evaluation, and implementation of inhomogeneous boundary conditions that are central to our moving-domain simulations. A critical aspect of the SEM is that neither the global nor the *local* stiffness matrices are ever formed. Elliptic problems are solved iteratively and thus require only the action of matrix-vector multiplication. Preconditioning is based on either diagonal scaling or hybrid multigrid-Schwarz methods with local smoothing effected through the use of separable operators [21, 35, 22, 23]. Exclusive reliance on matrix-free forms is particularly attractive in an ALE context because the overhead to update the operators as the mesh evolves is effectively nil.

We illustrate the basic components by considering the scalar elliptic problem,

$$-\nabla \cdot \mu \nabla u + \gamma u = f, \quad u = g \text{ on } \partial\Omega_D, \quad \nabla u \cdot \hat{\mathbf{n}} = 0 \text{ on } \partial\Omega \backslash \partial\Omega_D, \tag{43}$$

10

with Dirichlet conditions imposed on $\partial\Omega_D$ and Neumann conditions on the remainder of the boundary, $\partial\Omega\backslash\partial\Omega_D$. The coefficients and data satisfy $\mu > 0$, $\gamma \geq 0$, $f \in \mathcal{L}^2(\Omega)$, and $g \in C^0(\partial\Omega_D)$. This boundary value problem arises in many contexts in our Navier-Stokes solution process. With $\gamma = \beta_0/\Delta t$ and ν a constant, it is representative of the implicit subproblem for the velocity components in (39). For constant μ, the Poisson problem reflects the pressure substep of (38).

The discrete variational formulation of (43) is as follows: *Find $u(\mathbf{x})$ in X_b^N such that*

$$(\nabla v, \mu\nabla u) + (v, \gamma u) = (v, f) \qquad \forall\, v \in X_0^N, \tag{44}$$

where, as in the Navier-Stokes case, X_b^N (X_0^N) denotes the space of functions in X^N that satisfy $u = g$ ($u = 0$) on $\partial\Omega_D$. We symmetrize (44) by moving the boundary data to the right-hand side. If u_b is any known function in X_b^N, the reformulated system is as follows: *Find $u_0(\mathbf{x})$ in X_0^N such that*

$$(\nabla v, \mu\nabla u_0) + (v, \gamma u_0) = (v, f) - (\nabla v, \mu\nabla u_b) - (v, \gamma u_b) \qquad \forall\, v \in X_0^N, \tag{45}$$

with $u := u_0 + u_b$.

We formally introduce a *global representation* of $u(\mathbf{x})$, which is never used in practice but which affords compact representation of the global system matrices. Let any $u \in X^N$ be represented in terms of a Lagrange (nodal) interpolating basis,

$$u(\mathbf{x}) = \sum_{\hat{j}=1}^{\bar{n}} u_{\hat{j}}\phi_{\hat{j}}(\mathbf{x}), \tag{46}$$

with basis functions $\phi_{\hat{j}}(\mathbf{x})$ that are continuous on Ω. The number of coefficients, $\bar{n}$, corresponds to all basis functions in X^N. Ordering the coefficients with boundary nodes numbered last yields n interior nodes such that $X_0^N = \mathrm{span}\{\phi_{\hat{j}}\}_1^n$. Let I_n be the $n \times n$ identity matrix and $R = [I_n\ O]$ be an $n \times \bar{n}$ restriction matrix whose last $(\bar{n} - n)$ columns are empty. For any function $u(\mathbf{x}) \in X^N$ we will denote the set of $\bar{n}$ basis coefficients by $\underline{\bar{u}}$ and the set of n interior coefficients by $\underline{u}$. Note that $\underline{u} = R\underline{\bar{u}}$ always holds, whereas $\underline{\bar{u}}_0 = R^T\underline{u}_0$ holds only for functions $u_0 \in X_0^N$.

We define the stiffness $\bar{A}$ and and mass $\bar{B}$ matrices having entries

$$\bar{A}_{ij} := (\nabla\phi_i, \mu\nabla\phi_j), \qquad \bar{B}_{ij} := (\phi_i, \phi_j), \qquad i,j \in \{1, \ldots, \bar{n}\}^2. \tag{47}$$

The systems governing the interior coefficients of u_0 are the $n \times n$ restricted stiffness and mass matrices, $A = R\bar{A}R^T$ and $B = R\bar{B}R^T$, respectively. A is invertible if $n < \bar{n}$. We refer to $\bar{A}$ as the Neumann operator because it is the stiffness matrix that would result if there were no Dirichlet boundary conditions. It has a null space of dimension one, corresponding to the constant function.[1]

With the preceding definitions, the discrete equivalent of (45) is

$$\underline{v}^T A\,\underline{u}_0 + \gamma\underline{v}^T B\,\underline{u}_0 = \underline{v}^T R\left[\bar{B}\,\underline{f} - \bar{A}\,\underline{\bar{u}}_b - \gamma\bar{B}\,\underline{\bar{u}}_b\right]. \tag{48}$$

Here, we have exploited the fact that u_0 and v are in X_0^N, and for illustration we have made the simplifying assumptions that γ is constant and that $f \in X^N$. Neither of these assumptions is binding. Full variability, including jumps in μ, γ, and f across element boundaries, can be handled in the SEM.

Because (48) holds for all $\underline{v} \in \mathbb{R}^n$, the linear system for the unknown interior basis coefficients is

$$H\,\underline{u}_0 = R\left[\bar{B}\,\underline{f} - \bar{H}\,\underline{\bar{u}}_b\right], \tag{49}$$

---

[1] We remark that $\bar{A}$ governs the pressure in certain Navier-Stokes formulations when the system is closed. A pressure with zero mean is readily computed iteratively by projecting the constant mode out of the right-hand side and out of the pressure with each iteration.

with $\bar{H} := \bar{A} + \gamma\bar{B}$ and $H := R\bar{H}R^T$. The full solution to (43) then is given by (46) plus

$$\underline{\bar{u}} = R^T \underline{u}_0 + \underline{\bar{u}}_b. \tag{50}$$

For the case $f = 0$, we recognize in (49)–(50) the energy-minimizing projection,

$$\underline{\bar{u}} = \underline{\bar{u}}_b - R^T \left(R\bar{H}R^T\right)^{-1} R\bar{H}\,\underline{\bar{u}}_b, \tag{51}$$

which extends the trace of u_b into the interior of Ω in a smooth way provided that γ is also smooth.

Spectral Element Bases. In the SEM, the global bases ϕ_j are never formed. Rather, all operations are evaluated locally within each of E nonoverlapping hexahedral (curvilinear brick) elements whose union forms the domain $\Omega = \bigcup_{e=1}^{E} \Omega^e$. Functions in X^N are represented as tensor-product polynomials in the reference element, $\hat{\Omega} := [-1, 1]^d$, whose image is mapped isoparametrically to each of the elements, as illustrated for the case $d = 2$ in Fig. 2. As an example, a scalar field $u(\mathbf{r})$ on Ω^e in three dimensions would be represented in terms of local basis coefficients u_{ijk}^e as

$$u^e(\mathbf{r}) = \sum_{k=0}^{N}\sum_{j=0}^{N}\sum_{i=0}^{N} h_i(r)\,h_j(s)\,h_k(t)\,u_{ijk}^e. \tag{52}$$

Here, $\mathbf{r} = [r, s, t] = [r_1, r_2, r_3] \in \hat{\Omega}$ are the computational coordinates[2], and $h_i(\xi)$ are Nth-order Lagrange polynomials having nodes at the Gauss-Lobatto-Legendre (GLL) quadrature points, $\xi_j \in [-1, 1]$. This choice of nodes provides a stable basis and allows the use of pointwise quadrature, resulting in significant savings in operator evaluation. Typical discretizations involve $E = 10^2$–10^7 elements of order $N = 8$–16 (corresponding to 512–4,096 points per element). Vectorization and cache efficiency derive from the local lexicographical ordering within each element and from the fact that the action of discrete operators, which nominally have $O(EN^6)$ nonzeros, can be evaluated in only $O(EN^4)$ work and $O(EN^3)$ storage through the use of tensor-product-sum factorization [31, 36].

The geometry, $\mathbf{x}^e(\mathbf{r})$, takes exactly the same form as (52), and derivatives are evaluated by using the chain rule. For example, the pth component of the gradient of u at the GLL node $\xi_{ijk} := (\xi_i, \xi_j, \xi_k)$ is computed as

$$\left.\frac{\partial u}{\partial x_p}\right|_{\xi_{ijk}} = \left.\frac{\partial r_1}{\partial x_p}\right|_{\xi_{ijk}} \sum_{i'=0}^{N} \hat{D}_{ii'} u_{i'jk} + \left.\frac{\partial r_2}{\partial x_p}\right|_{\xi_{ijk}} \sum_{j'=0}^{N} \hat{D}_{jj'} u_{ij'k} + \left.\frac{\partial r_3}{\partial x_p}\right|_{\xi_{ijk}} \sum_{k'=0}^{N} \hat{D}_{kk'} u_{ijk'},$$

where $\hat{D}$ is the one-dimensional derivative matrix on $[-1, 1]$. $\hat{D}_{ij} = \left.\frac{dh_j}{dr}\right|_{\xi_i}$. We note that if the metric terms $\frac{\partial r_q}{\partial x_p}$ are precomputed, then the work to evaluate all components of the gradient, $\frac{\partial u}{\partial x_p}$, is $(6N+15)EN^3 \approx (6N+15)\bar{n}$, and the number of memory accesses is $O(\bar{n})$. The work to compute the metrics $\frac{\partial r_q}{\partial x_p^e}$ is similarly $O(N\bar{n})$. Using $\hat{D}$, one evaluates the 3×3 matrix $\frac{\partial x_q^e}{\partial r_p}$, then inverts this matrix pointwise in $O(N^3)$ operations to obtain $(F_{pq}^e)_{\xi_{ijk}} := \left.\frac{\partial r_q}{\partial x_p^e}\right|_{\xi_{ijk}}$. If $\underline{u}^e$ is the lexicographically ordered set of basis coefficients on element Ω^e, its gradient can be compactly expressed as

$$\underline{w}_p^e = \sum_{q=1}^{3} F_{pq}^e D_q \underline{u}^e, \quad p = 1, 2, \text{ or } 3, \tag{53}$$

---

[2] In this section, we occasionally use "t" to represent the third coordinate in the reference domain $\hat{\Omega}$. It should not be confused with time because no temporal variation exists in the current context.

where $D_1 = I \otimes I \otimes \hat{D}$, $D_2 = I \otimes \hat{D} \otimes I$, $D_3 = \hat{D} \otimes I \otimes I$, and, for each p, q and e, F_{pq}^e is a diagonal matrix.

The high order of the SEM coupled with the use of GLL-based Lagrangian interpolants allows the integrals in (47) to be accurately approximated by using pointwise quadrature. In particular, the mass matrix becomes diagonal. For a single element one has

$$
\begin{aligned}
B_{\hat{i}\hat{i}'}^e \quad &:= \quad \int_{\Omega^e} \phi_{\hat{i}}\, \phi_{\hat{i}'}\, d\mathbf{x} = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \left[h_i(r) h_j(s) h_k(t) \right] \left[h_{i'}(r) h_{j'}(s) h_{k'}(t) \right] \mathcal{J}^e\, dr\, ds\, dt \\
&\approx \quad \sum_{i'' j'' k''} \rho_{i''} \rho_{j''} \rho_{k''} \left[h_i(\xi_{i''}) h_j(\xi_{j''}) h_k(\xi_{k''}) \right] \left[h_{i'}(\xi_{i''}) h_{j'}(\xi_{j''}) h_{k'}(\xi_{k''}) \right] \mathcal{J}_{i'' j'' k''}^e \\
&= \quad \rho_i \rho_j \rho_k \mathcal{J}_{ijk}^e\, \delta_{ii'} \delta_{jj'} \delta_{kk'},
\end{aligned}
\tag{54}
$$

where $\mathcal{J}^e = \left| \frac{\partial x_p^e}{\partial r_q} \right|$ is the pointwise Jacobian associated with the mapping $\mathbf{x}^e(\mathbf{r})$, ρ_j is the quadrature weight corresponding to the GLL point ξ_j, and $\delta_{ii'}$ is the Kronecker delta. For compactness, we have also introduced the lexicographical ordering $\hat{i} := i + (N+1)(j-1) + (N+1)^2(k-1)$. The same map takes the trial function (i', j', k') to $\hat{i}'$. The tensor-product form of the local mass matrix is $B^e = J^e(\hat{B} \otimes \hat{B} \otimes \hat{B})$, where $\hat{B} = \text{diag}(\rho_k)$ is the 1D mass matrix containing the GLL quadrature weights and J^e is the diagonal matrix of Jacobian values at the quadrature points.

Combining the mass matrix with the gradient operator yields the local stiffness matrix as typically applied in the SEM, namely,

$$
A^e \quad = \quad \sum_{p=1}^3 \sum_{q=1}^3 D_p^T \left(\mu\, G_{pq}^e \right) D_q, \qquad G_{pq}^e \quad := \quad B^e \sum_{q'=1}^3 F_{q'p}^e F_{q'q}^e.
\tag{55}
$$

We note that $G_{pq}^e = G_{qp}^e$ is a symmetric tensor field that amounts to six diagonal matrice of size $(N+1)^3$ for each element Ω^e. Likewise, the variable diffusivity μ is understood to be a diagonal matrix evaluated at each gridpoint, ξ_{ijk}^e. We emphasize that for the general curvilinear element case A^e is completely *full*, with $(N+1)^6$ nonzeros, which makes it prohibitive to form for $N > 3$. However, the factored form (55) is *sparse*, with only $6(N+1)^3$ nonzeros for all the geometric factors G_{pq}^e plus $O(N^2)$ for derivative matrices (and an additional $(N+1)^3$ if μ is variable). The total storage for the general factored stiffness matrix is $\sim 7n_l$, where $n_l = E(N+1)^3$ is the total number of gridpoints in the domain. Moreover, the total work per matrix-vector product is only $\sim 12Nn_l$, and this work is effectively cast as highly vectorizable matrix-matrix products [31, 37, 30].

To complete the problem statement, we need to assemble the local stiffness and mass matrices, B^e and A^e, and apply the boundary conditions, both of which imply restrictions on the nodal values u_{ijk}^e and v_{ijk}^e. For any $u(\mathbf{x}) \in X^N$ we can associate a single nodal value u_g for each unique $\mathbf{x}_g \in \Omega$. where $g \in \{1, \ldots, \bar{n}\}$ is a global index. Let $g = g_{ijk}^e$ be an integer that maps any $\mathbf{x}_{ijk}^e$ to $\mathbf{x}_g$; let $l = i + (N+1)(j-1) + (N+1)^2(k-1) + (N+1)^3(e-1)$ represent a lexicographical ordering of the local nodal values; and let $m = E(N+1)^3$ be the total number of local nodes. We define Q^T as the $\bar{n} \times m$ Boolean gather-scatter matrix whose lth column is $\hat{\underline{e}}_{g(l)}$, where $g(l)$ is the local-to-global pointer and $\hat{\underline{e}}_g$ is the gth column of the $\bar{n} \times \bar{n}$ identity matrix. For any $u \in X^N$ we have the global-to-local map $\underline{u}_L = Q\underline{u}$, where $\underline{u}_L = \{\underline{u}^e\}_{e=1}^E$ is the collection of *local* basis coefficients. With these definitions, the discrete bilinear form for the Laplacian becomes

$$
(\nabla v, \mu \nabla u) \quad = \quad \sum_{e=1}^E (\underline{v}^e)^T A^e \underline{u}^e \quad = \quad \underline{v}_L^T A_L\, \underline{u}_L \quad = \quad (Q\underline{v})^T A_L\, Q\underline{u} \quad = \quad \underline{v}^T Q^T A_L\, Q\underline{u}, \quad = \quad \underline{v}^T \bar{A}\underline{u}.
$$

Here $A_L = \text{block-diag}\{A^e\}$ is termed the *unassembled* stiffness matrix, and $\bar{A} = Q^T A_L Q$ is the *assembled*

stiffness matrix. To obtain the mass matrix, we consider the inner product,

$$(v, u) = \sum_{e=1}^{E} (\underline{v}^e)^T B^e \underline{u}^e = \underline{v}_L^T B_L \underline{u}_L = (Q\underline{v})^T B_L Q\underline{u} = \underline{v}^T Q^T B_L Q\underline{u}, = \underline{v}^T \bar{B}\underline{u}.$$

Here, B_L= block-diag$\{B^e\}$ and $\bar{B} = Q^T B_L Q$ are, respectively, the diagonal unassembled and assembled mass matrices comprising local mass matrices, B^e.

We close this section on basis functions by defining elements of the pressure space. For the $\mathbb{P}_N - \mathbb{P}_N$ formulation described in [15], we take $Y^N = X^N$. That is, the pressure is continuous and represented by basis functions having the form (52). For the $\mathbb{P}_N - \mathbb{P}_{N-2}$ formulation of Maday and Patera [38], the elements of Y^N have the tensor-product form of (52) except that the index ranges from 0 to N-2 and the nodal points are chosen to be the Gauss-Legendre quadrature points rather than the GLL points. Furthermore, interelement continuity is not enforced on either the pressure, p, or the corresponding test function, q. Element-to-element interaction for the pressure derives from the fact that the velocity $\mathbf{u}$ and test functions $\mathbf{v}$ are in $X^N \subset H^1$. We refer to [38, 21, 31] for additional detail concerning the SEM bases and implementation of the $\mathbb{P}_N - \mathbb{P}_{N-2}$ formulation.

0.4 Computational Results

Here, we consider several examples that illustrate the accuracy and performance of the characteristic/OIFS scheme introduced in the preceding section. These methods have been implemented in Nek5000, which is an open source spectral element code for fluid, thermal, and combustion simulations that scales to over a million processors [39]. For most of this section, we will refer to the proposed characteristic/OIFS scheme as CHAR/BDFk while we refer to the standard semi-implicit or non-characteristic scheme as EXTk/BDFk.

Before proceeding to the examples we briefly discuss guidelines for choosing the number of RK4 sub-steps, M. This choice is driven by the practitioner's target Courant number, C defined as

$$C = \max_i \frac{|\mathbf{u}_i - \mathbf{w}_i|\Delta t}{\Delta x_i}, \tag{56}$$

where the subscript i is used to indicate the range of gridpoints in the domain. We note that the velocity reflects the relative velocity between the flow field and the mesh velocity. If one wishes for a value of $C = 3$, then the unconditional stability of the BDFk allows for $\Delta t \approx \frac{3\Delta x}{|\mathbf{u}_o|}$. For simplicity, $\mathbf{u}_o$ is a characteristic velocity which may reflect mesh motion. In the characteristic approach, there is a sub-timestep, Δs, given by $\Delta s = \frac{\Delta t}{M}$, where M is an integer. For the RK4 procedure there is a conditional stability associated with Δs. The stability diagram for RK4 [31] intersects the imaginary axis at ± 2.82. As a result the stable Δs is given by

$$\Delta s \leq \frac{2.82\Delta x}{|\mathbf{u}_o|S}, \tag{57}$$

where S is a margin of safety that is approximately 1.5 for $N = 2, 3$ and gradually settles to the range of $1.2 \to 1.16$ for $N = 16 \to 256$ [31]. We may now write

$$\frac{3\Delta x}{|\mathbf{u}_o|} \approx \Delta t = M\Delta s \leq M\left(\frac{2.82\Delta x}{|\mathbf{u}_o|S}\right), \tag{58}$$

which suggests

$$M \geq \frac{3S}{2.82}. \tag{59}$$

In practice, we use a value of $M = 2$ or greater to ensure a stable timestep.

14

0.4.1 Temporal-Spatial Accuracy

Using the $\mathbb{P}_N - \mathbb{P}_N$ discretization and an ALE formulation, we test the accuracy of the CHAR/BDFk scheme by considering the family of exact eigenfunctions for the incompressible Navier-Stokes equations (derived by Walsh [40]) in a moving domain.

For all integer pairs (m,n) satisfying $\lambda = -(m^2 + n^2)$, families of eigenfunctions can be formed by defining streamfunctions ψ that are linear combinations of the functions

$$\cos(mx)\cos(ny),\ \sin(mx)\cos(ny),\ \cos(mx)\sin(ny),\ \sin(mx)\sin(ny).$$

With the eigenfunction $\mathbf{u}^0 := (-\psi_y, \psi_x)$ as an initial condition, a solution to the Navier-Stokes equations is $\mathbf{u} = e^{\nu\lambda t}\mathbf{u}^0(\mathbf{x})$.

We add a relatively high-speed mean flow $\bar{\mathbf{u}}$ with an exact solution given as

$$\tilde{\mathbf{u}}(\mathbf{x}, \mathbf{t}) = \bar{\mathbf{u}} + e^{\nu\lambda t}\mathbf{u}^0[\mathbf{x} - \bar{\mathbf{u}}t], \tag{60}$$

where the brackets imply that the argument is modulo 2π in x and y. By varying $\bar{\mathbf{u}}$, one can advect the solution a significant number of characteristic lengths before the eigensolution decays.

We take an initial configuration $(x^0, y^0) \in \Omega^0 = [0,7]^2$ and evolve this with the prescribed mesh velocity

$$\dot{x} = \omega\cos(\omega t)\ \sin(\pi y^0/7), \tag{61}$$

$$\dot{y} = \omega\cos(\omega t/2)\ \sin(\pi x^0/7)(2y-1), \tag{62}$$

with $\omega = 5$. The mesh consists of a 16×16 array of square spectral elements. We have inhomogeneous Dirichlet conditions on all of $\partial\Omega$ corresponding to $\tilde{\mathbf{u}}$ (60). In the present case, since we have an exact solution as a function of space and time, we can run the Dirichlet case with the solution prescribed on all four sides of the domain. Starting with the initial condition we take $\nu = .01$ and $\bar{\mathbf{u}} = (1, .3)$ and evolve the solution to a final time T. Figure 3 shows a picture of the velocity magnitude in the domain at two timepoints.

Figure 4 shows the maximum pointwise error in the x-component of the velocity at a final time of $T = 7.0$. The left panel shows that the characteristic scheme yields expected kth-order accuracy for $k = 2$ and 3. The right panel shows the error (using $k = 3$) as a function of Δt for several values of N. The general trend is that the error is dominated by spatial error for sufficiently small values of Δt and becomes dominated by temporal error as Δt is increased.

Figure 5 shows the maximum pointwise error as a function of the Courant number, C, defined by (56). Results are shown for $k = 2$ (left) and 3 (right) with a fixed polynomial order of $N = 11$ in each case. As a result, C becomes a strong function of the chosen timestep, Δt. Each data point is calculated for a timestep that is fixed for the duration of the simulation. The solution is evolved to $T = 1.0$, where C is then calculated. In each panel, we compare error and the maximum C that can be attained for CHAR/BDFk and EXTk/BDFk. For CHAR/BDFk we choose the number of sub-steps to be $M = 1$ which means $\Delta s = \Delta t$. Interestingly, for $k = 2$ (left) the characteristic scheme (CHAR/BDFk) exhibits a lesser degree of accuracy than its EXTk/BDFk counterpart. On the other hand, the characteristic scheme achieves a Courant number (or timestep) that is more than twice that for EXTk/BDFk. This is almost true for $k = 3$ (right). In particular, the characteristic scheme is able to attain a higher C, although it is not twice as high as the EXTk/BDFk scheme. At their largest timesteps, however, the characteristic scheme is as accurate as the EXTk/BDFk scheme.

0.4.2 Flow in a Channel with a Moving Indentation

In this section we validate our characteristic-based approach by studying the two-dimensional, unsteady flow of a viscous, incompressible fluid in a channel with a time-dependent indentation in one wall. This problem
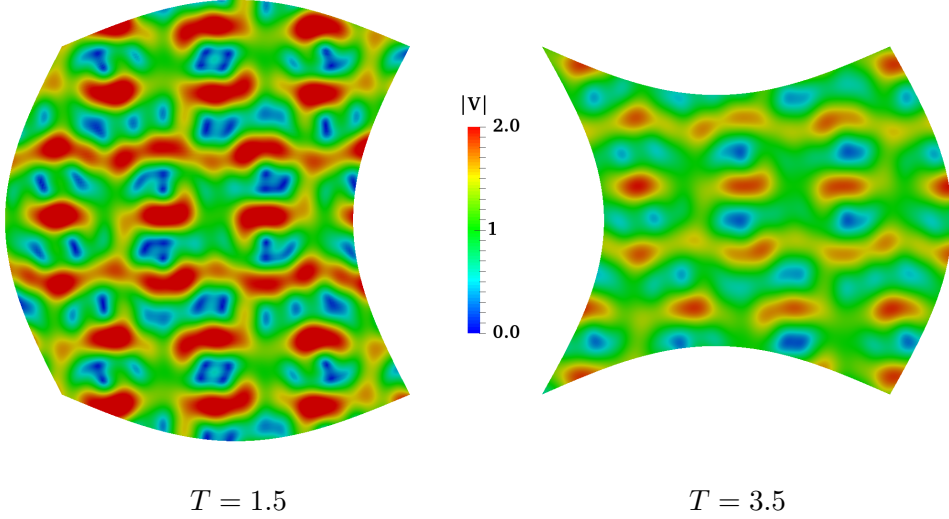
$$T = 1.5 \qquad\qquad\qquad T = 3.5$$

Figure 3: Eddy solution results (CHAR/BDFk) at Re=100 for the moving domain problem: left and right images show velocity magnitude and domain configuration at two timepoints with $k = 3$.
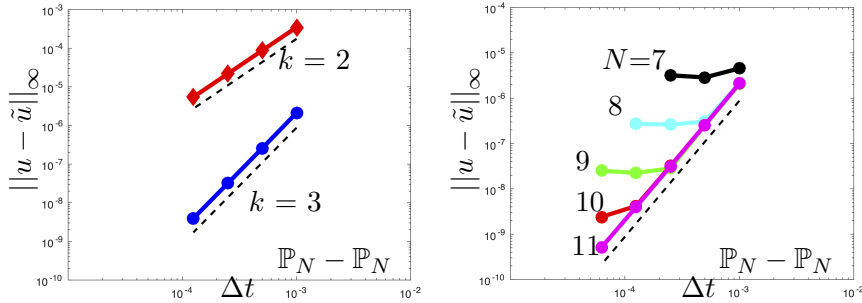

Figure 4: Eddy solution results (CHAR/BDFk) at Re=100 for the moving domain problem: (left) 2nd- and 3rd-order temporal accuracy based on maximum pointwise error is shown for polynomial order $N = 12$, (right) maximum pointwise error for $N = 7$–11 using $k = 3$. Dashed line shows the slope for each BDFk scheme. The slopes agree with temporal accuracy of $k = 2$ and $k = 3$, respectively.

has been the subject of several studies [1, 41, 42] seeking to better understand flow in collapsible tubes such as arteries and veins. The configuration for this study is shown in Fig. 6. Based upon the experiment of Pedley and Stephanoff [41], a steady Poiseuille flow or parabolic velocity profile is prescribed at the inlet. An indentation on the bottom wall moves in and out sinusoidally where its retracted position is flush with the wall. The two main nondimensional parameters are given by the Reynolds number $Re = \frac{\bar{U}_o b}{\nu} = 507$ and the Strouhal number $St = \frac{b}{\bar{U}_o T} = 0.037$. $\bar{U}_o = \frac{Q_o}{b}$ is the mean inlet velocity, where Q_o is the volumetric flow rate per unit channel depth and b is the channel height. T is the oscillation period.

The walls of the channel are defined to be at $y = b$ and $y = F(x,t)$. $F(x,t)$ is taken to be of the seperable form

$$F(x,t) \quad = \quad g(x)h(t), \tag{63}$$

16
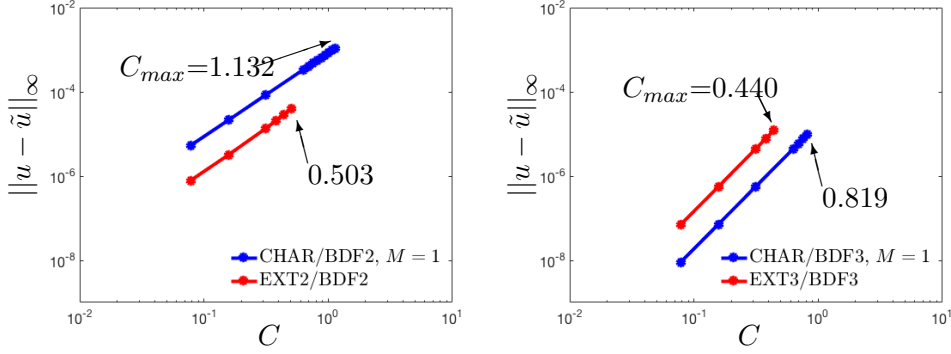
Figure 5: Eddy solution results (CHAR/BDFk (blue) vs. EXTk/BDFk (red)) where the solution is evolved to $T = 1.0$. Left image shows maximum pointwise error as a function of the Courant number, C, for $k = 2$. Right image is for $k = 3$.

where $h(t) = 0$ for $t < 0$, $h(t) \geq 0$ for $t \geq 0$ and $g(x) \geq 0$. Here we choose $h(t)$ as

$$h(t) \quad = \quad 0.5[1 - \cos 2\pi t^*], \tag{64}$$

where $t^* = \frac{t}{T}$. The shape of the indentation is desribed by $g(x)$ and is symmetric around the origin $x = 0$ (i.e. $g(-x) = g(x)$). Following the analysis in Ralph and Pedley [42], $g(x)$ is written as

$$g(x) \quad = \quad 0.5h_{max}[1 - \tanh a(|x| - x_2)], \tag{65}$$

which reproduces the wall profiles given by experiments. Here $h_{max} = 0.38b$ specifies the maximum height of the indentation, and $a = 4.14$. $x_2 = 0.5(x_1 + x_3)$, where $x_1 = 4b$ and $x_3 = 6.5b$. The entire length of the channel is given as $l_1 + l_2 = 40b$ with $l_1 = 13.75b$ and $l_2 = 26.25b$.
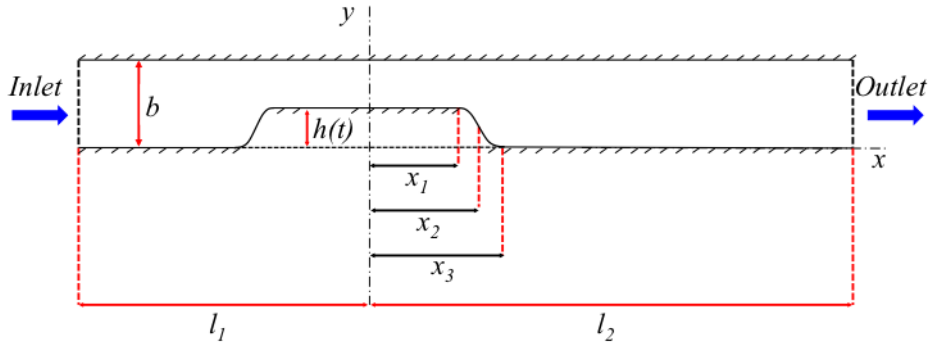
Figure 6: Schematic for flow in a channel with a moving-wall indentation.

The development of the flow can be seen in Figs. 7–8 and is similar to the flow field found in the experiments of Pedley and Stephanoff [41]. At some time between $t^* = 0.20$ and $t^* = 0.30$ flow separates on the lee side of the moving indentation, and the resulting eddy grows rapidly. Around $t^* = 0.40$, a second
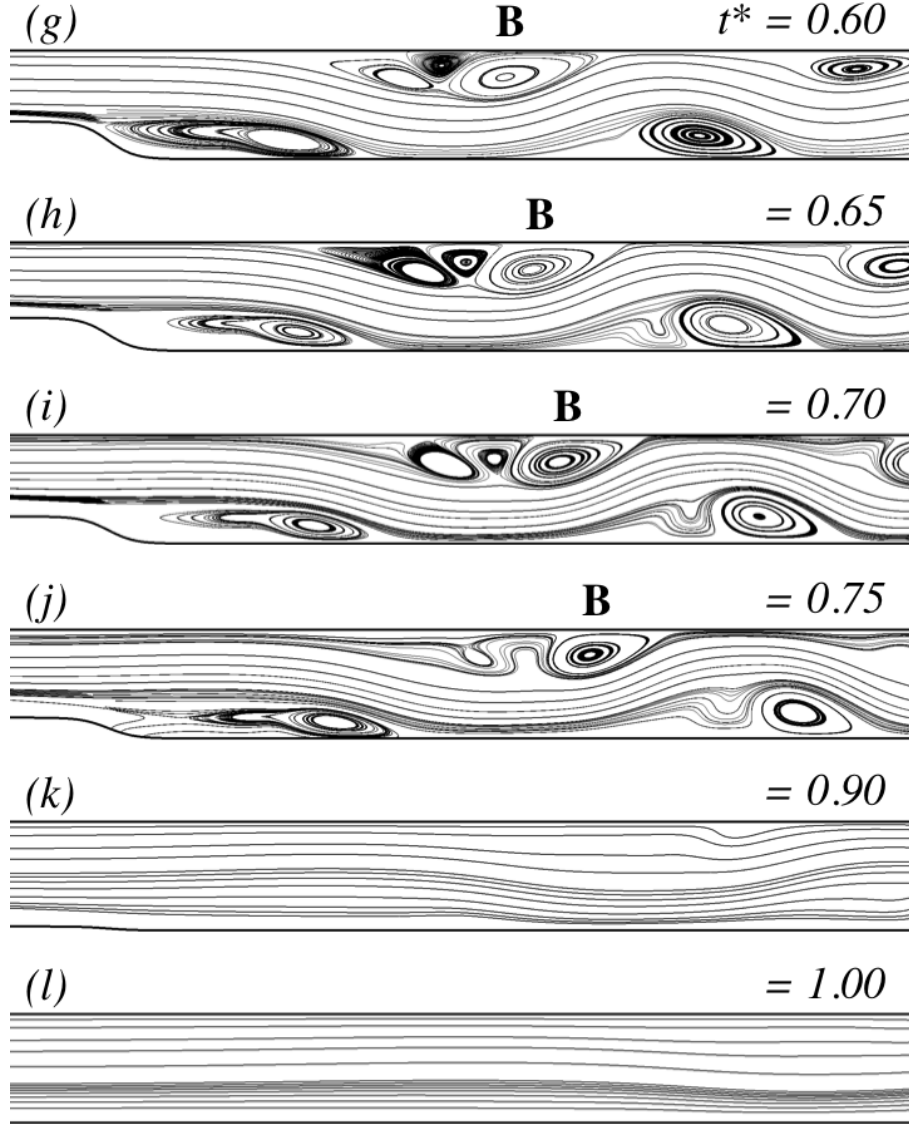
17
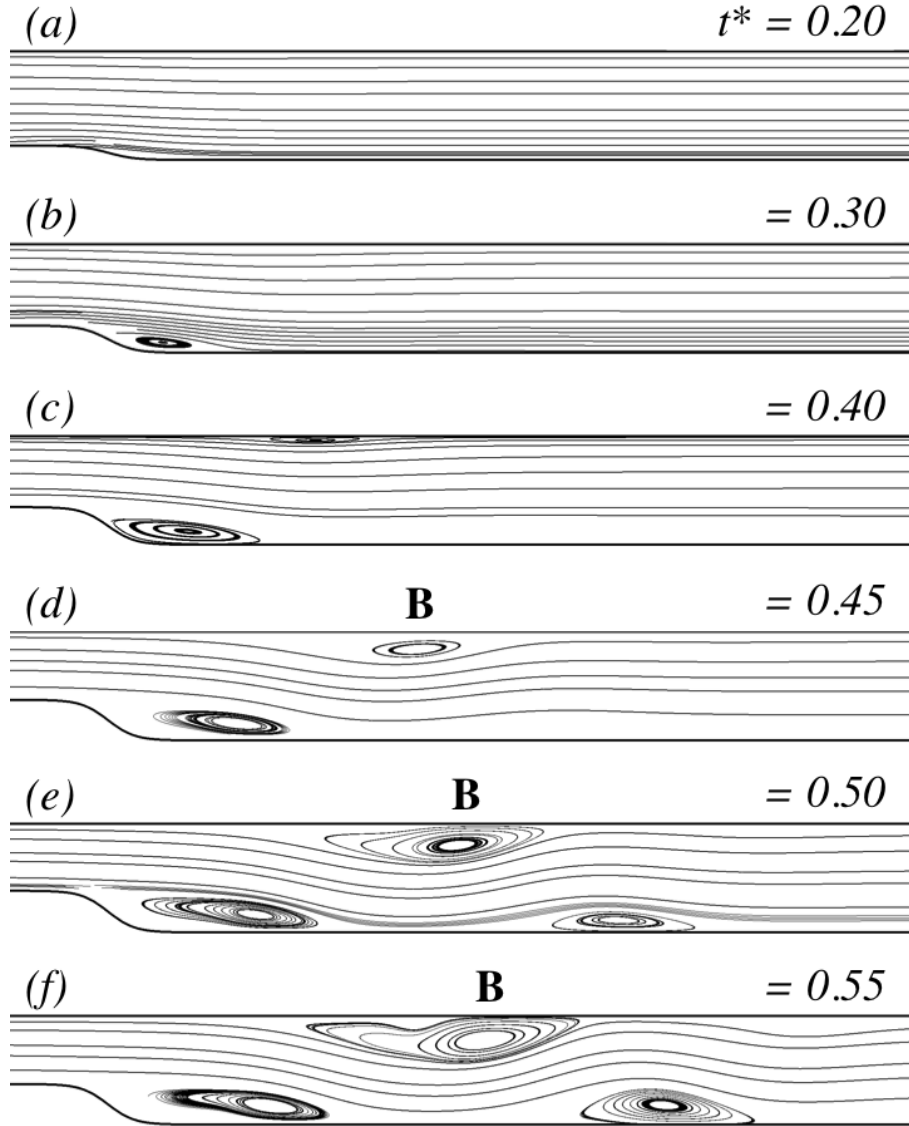
Figure 7: Wave formation and propagation downstream of a moving indentation for $Re = 507$. The total number of elements $E = 60 \times 8 = 480$ with polynomial order $N = 11$. Results shown for CHAR/BDFk with k=2 and $C = 2$.

eddy forms on the upper wall. Adopting the convention in the experiment [41], we label the second eddy as B. Trifurcation of eddy B into smaller corotating recirculations can be seen in Figs. 7(g)-(j) and is consistent with the results found in [42]. Figure 9 tracks the motion of eddy B by measuring the trough position, or lowest point of the eddy along a scaled abscissa coordinate, x^*, over time. The coordinate is defined according

Figure 8: Wave formation and propagation downstream of a moving indentation for $Re = 507$. The total number of elements $E = 60 \times 8 = 480$ with polynomial order $N = 11$. Results shown for CHAR/BDFk with k=2 and $C = 2$.

to the analysis of Pedley and Stephanoff [41] and is given by

$$x^* = \left(\frac{x - x_1}{b}\right)(10St)^{(\frac{1}{3})}. \tag{66}$$

The unfilled triangular symbols represent results from the proposed characteristic-based scheme. We note that results for the larger Courant number, $C = 2$, which is realized by a larger timestep, shows good agreeement
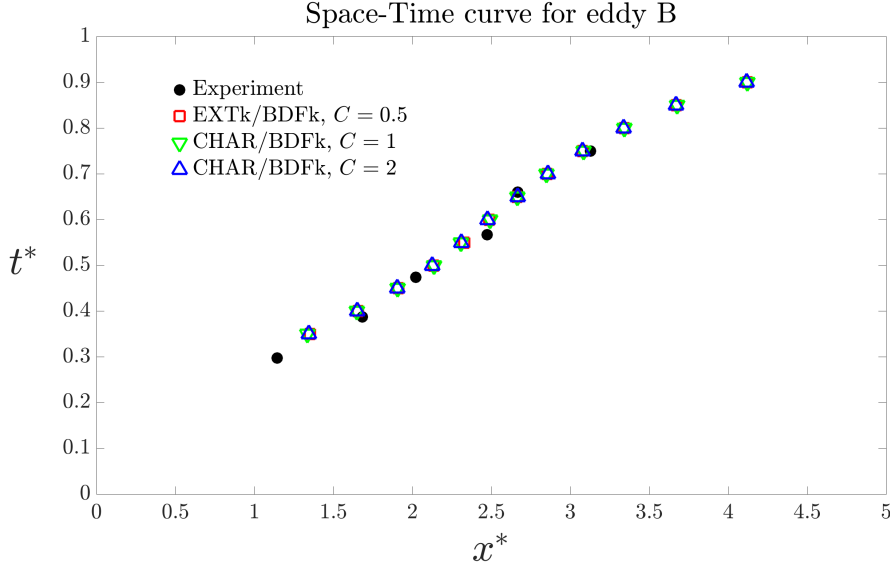
Figure 9: Space-time (x-t) curve for eddy B. Filled circular symbol represents experimental results from Pedley and Stephanoff [41]. Unfilled symbols represent $\mathbb{P}_N - \mathbb{P}_N$ approach. Square symbol represents the standard semi-implicit approach EXTk/BDFk. Triangular symbols are based on the proposed characteristic scheme CHAR/BDFk. Results are presented for k=2.
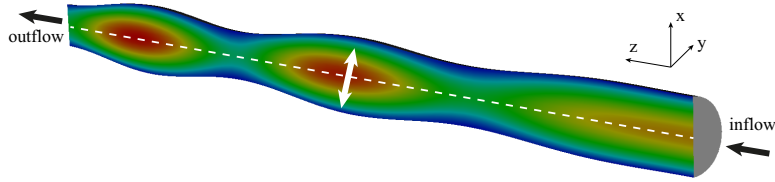
Figure 10: Distribution of velocity magnitude for the CHAR/BDFk and $\mathbb{P}_N - \mathbb{P}_N$ approach on a vertical slice at t = 50.

with the experimental results.

0.4.3 Peristaltic Pumping

For the 3D case, verification of the CHAR/BDFk implementation begins with the tubular setup of Fig. 10. The moving mesh in this case generates a peristaltic pumping that strongly influences the temporal and spatial evolution of the velocity field. The pipe has a base radius $R = 1/2$ and length $L = 16$. The prescribed mesh velocity is

$$w_1 = -W \frac{x}{R} \cos(kz - \omega t), \quad w_2 = -W \frac{y}{R} \cos(kz - \omega t), \quad w_3 = 0, \tag{67}$$

where $W = A/\omega$ is the velocity amplitude and $A = 0.1 \tanh(0.2z) \tanh(0.2t)$ is the amplitude of the displacement. The prescribed wavenumber is $k = \pi/3$, and the frequency is $\omega = 1$. The subscripts 1, 2, and 3 in (67) correspond to the x, y, and z directions respectively. The Reynolds number is always below 200 so that the flow remains laminar. At the inflow a steady parabolic velocity profile with a maximum axial velocity $u_3 = 1$ at the cylinder center is imposed while at the outflow zero-Neumann boundary conditions are
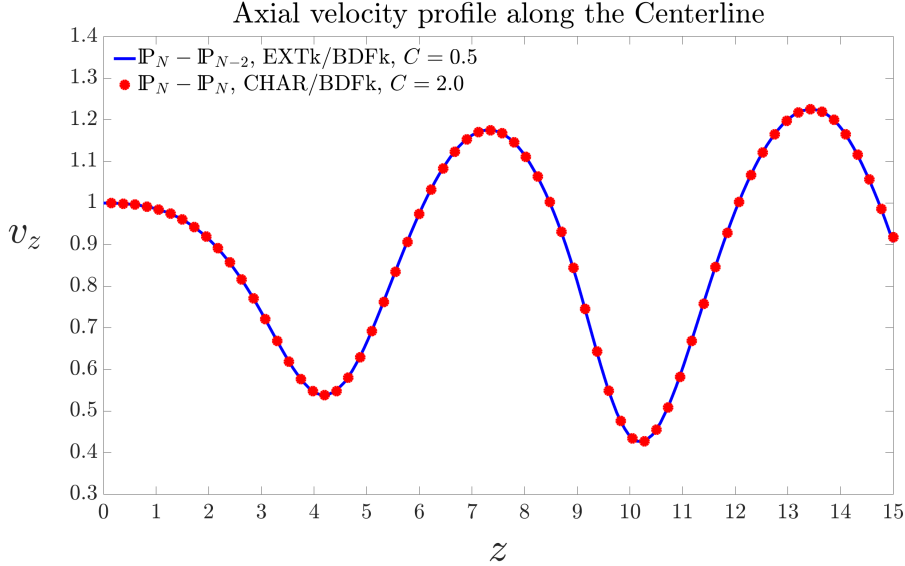
Figure 11: Instantaneous axial velocity profile at $T = 50$ during peristaltic pumping. Results are shown for two cases. The blue line is the benchmark study using the EXTk/BDFk and $\mathbb{P}_N - \mathbb{P}_{N-2}$ approach. Red symbols are for the proposed CHAR/BDFk and $\mathbb{P}_N - \mathbb{P}_N$ approach. Both simulations have the same number of elements $E = 192$ and polynomial order $N = 9$. Results are presented for k=2.

Method	Time to Solution (Sec.) - BDF2	Time to Solution (Sec.) - BDF3
CHAR/BDFk	2.10×10^3	2.53×10^3
EXTk/BDFk	5.88×10^3	5.97×10^3

Table 1: Peristaltic pumping performance. Time-to-solution results are measured at final time $t_{final} = T = 50$. In both cases, the polynomial order is $N = 9$ and the $\mathbb{P}_N - \mathbb{P}_N$ formulation is used. For CHAR/BDFk, $M = 1$ and a maximum stable timestep is chosen such that $C > 1$.

used. At the pipe walls the velocity is set equal to the mesh velocity in order to prevent a flow across the walls. The numerical setup including the mesh is given in the example `peris` of the Nek5000 package.

Figure 11 shows the axial velocity distribution along the centerline of the pipe at time, $T = 50$. For this calculation, the total number of elements is $E = 192$ with a polynomial order of $N = 9$. The present characteristic-based (CHAR/BDFk), $\mathbb{P}_N - \mathbb{P}_N$ formulation is compared with the non-characteristic (EXTk/BDFk), $\mathbb{P}_N - \mathbb{P}_{N-2}$ approach of Ho and Patera [43, 44]. The circle markers indicate results from the CHAR/BDFk approach where the number of RK4 substeps is $M = 2$ which achieves the $C = 2$. The axial velocity shows excellent agreement between the two formulations.

Table 1 shows time-to-solution data comparing the CHAR/BDFk approach with the EXTk/BDFk approach using the same $\mathbb{P}_N - \mathbb{P}_N$ formulation. These calculations were performed on an Apple MacBook Pro which runs a 2.3 GHz Intel Core i7 processor. Using $M = 1$, we chose a maximum timestep such that the calculation remains numerically stable. This results in a $C > 1$. As the table shows, the characteristic approach achieves a reduction in the time to solution by more than a factor of 2 over the noncharacteristic approach.

21

0.4.4   TCC-III Engine

In this section, we illustrate a practical application of the CHAR/BDFk method. In particular, we assess the performance of this approach in a simulation of incompressible flow during the intake stroke for the Transparent Combustion Chamber-III (TCC-III) engine [7]. For this investigation, we modify the geometry of the TCC-III engine by simulating flow that enters an intake-valve and empties into a pancake-shaped combustion chamber. The image on the left of Fig. 12 is a schematic describing the geometry. During the intake stroke, the valve opens (i.e., moves down) thus allowing fluid to enter into the combustion chamber. As the valve moves down, the piston is simultaneously moving downward with velocity, v_{piston}. The diameter of the cylinderical combustion chamber, known as the bore, represents the characteristic length, L_o. The diagram on the right shows the geometric layout for the piston pin (P), crank pin (N) and crank center (O). The piston pin is connected to the crank pin through a connecting rod of length, l. Here x is the piston pin position, r is the crank radius (also called the half stroke), and a is the crank angle which is measured from the bore centerline or the top dead center (TDC) position of the piston. The crankshaft angular velocity, ω, is related to the engine revolutions per minutes (RPM) by $\omega = \frac{2\pi RPM}{60}$. Assuming a constant angular velocity, the crank angle is given by $a = \omega t$. The position of the piston, x, and the velocity, v_{piston}, are given as

$$
\begin{aligned}
x &= r\cos a + \sqrt{l^2 - (r\sin a)^2}, \\
v_{piston} &= \frac{dx}{da} \times \frac{da}{dt} = \left(-r\sin a - \frac{r^2 \sin a \cos a}{\sqrt{l^2 - r^2 \sin^2 a}} \right) \times \omega.
\end{aligned}
\tag{68}
$$

Figure 13 illustrates the piston velocity and the motion of the intake valve. The valve motion from the online database [45] is described in terms of the valve lift (i.e. distance from the horizontal chamber walls) versus the crank-angle (CA) degree.

The reference velocity for this calculation is the maximum piston speed, $U_o = v_{piston,max}$. Based upon Fig. 13, the maximum piston speed occurs at approximately 440° CA, as the piston is moving downward toward the bottom dead center (i.e., lowest position). DNS calculations are shown in Fig. 14, which show non-dimensional velocity distribution snapshots for $Re = 5000$. For this DNS we use the following parameters: RPM = 50, Bore (L_o) = 9.2 cm, Stroke ($2r$) = 8.6 cm, Rod length (l) = 23.10 cm, and $v_{piston,max} = 22.90$ cm/s. According to the snapshots, we observe a fluid entering the pancake-shaped chamber and evolves into a highly complex and turbulent flow field. Developing a turbulent flow field inside the chamber is crucial to preparing the fuel mixture for combustion [5].

The simulation begins at 362° CA, where the initial condition is a quiescent flow. The moving boundaries for this problem are the valve and piston surface, while the remaining boundaries (cylindrical chamber) are considered to be static walls. Due to the incompressible assumption, the inflow boundary condition (at the top of the intake-valve) is imposed such that the volumetric flow rate is the same as the volumetric change in the chamber due to the piston motion. This ensures that flow enters the valve and cylindrical chamber as the piston surface moves downward. These results were captured by using the CHAR/BDF3 scheme with a target Courant number of $C = 3$ and setting $M = 3$. The total number of spectral elements used is $E = 6784$ with polynomial order of $N = 11$.

The mesh displacement is computed by integrating Eq. (21) in time where the mesh velocity $\mathbf{w}$ is subject to the kinematic condition,

$$
\mathbf{w} \cdot \hat{\mathbf{n}}|_{\partial\Omega} = \mathbf{u} \cdot \hat{\mathbf{n}}|_{\partial\Omega}.
\tag{69}
$$

where $\hat{\mathbf{n}}$ is the unit normal at the domain surface, $\partial\Omega(t)$. In order to smoothly blend boundary data into the domain interior we solve Laplace's equation (i.e., $-\nabla \cdot \mu \nabla u = 0$) in order to blend the surface velocities to the interior, relying on the maximum principle to give a bounded interpolant. Here, u represents a displacement factor which is applied at all gridpoints in the domain.

To preserve the resolution of critical boundary-layer elements, we increase the diffusivity (μ) near specific walls so that the mesh velocity tends to match that of the nearby object. The bulk of the mesh deformation
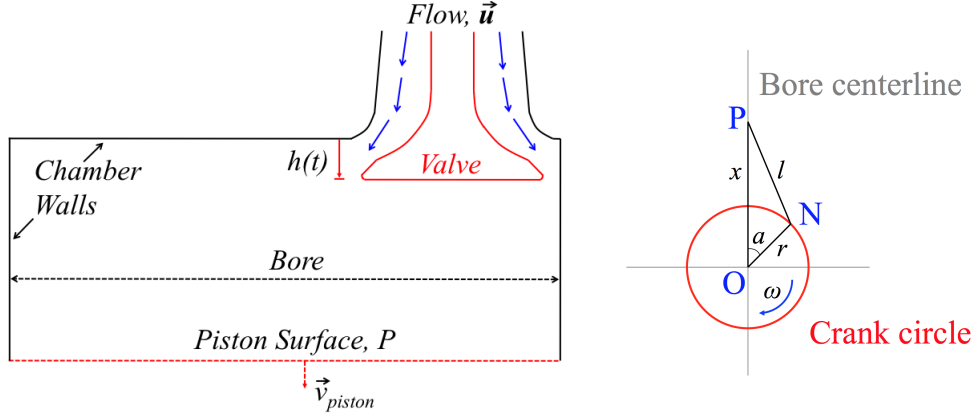
Figure 12: Schematic for the modified TCC-III internal combustion engine (Left). The image on the right is a diagram showing the geometric layout of the piston pin (P), crank pin (N) and crank center(O).

is effectively pushed into the far field, where elements are larger and thus better able to absorb significant deformation. The idea of using variable diffusivity has been explored in finite-element contexts where the coefficient is based on local element volumes (e.g., [46, 47]) and can also be applied to the elasticity equations.

We set $\mu(\mathbf{x}) = 1 + \alpha e^{-\delta^2}$ with $\alpha = 3$ and $\delta := d/\Delta$ the distance to the wall normalized by a chosen length scale, Δ. In the absence of any other scale information, we set Δ equal to the average thickness of the first layer of spectral elements in contact with the given object. To compute d, we use a Euclidian graph-based approximation to the true distance function. A variable diffusivity is especially beneficial in the vicinity of the valve and piston surfaces since boundary motion is in the same direction as the normal to the surface. Note, boundary motion along the cylinder chamber is orthogonal to the surface where the boundary-layer is preserved throughout the entire simulation.

0.4.5   TCC-III Engine: Computational Performance

We use this section to illustrate the performance for the CHAR/BDFk scheme by monitoring the computational cost or wall time required to execute an early phase of the intake stroke from 362° CA to 375° CA. The early phase of the intake stroke is critical in the evolution of the flow due to the fluid's dramatic acceleration in the small valve-gap region (also known as the valve seat). The high-speed flow in this region has a lasting effect on the evolution of the flow structure in the chamber. Figure 15 illustrates this phenomena by showing the nondimensional velocity distributions for 364° CA (left) and 375° CA (right). From the figure, we observe that the largest velocities exist in the small-gap region. In terms of computational performance, it becomes critical for the practitioner to choose a timestepping scheme that attains the largest stable Courant number, C.

Figure 16 shows the wall-time results for the simulation to execute from 362° CA to 375° CA. These calculations were performed on 4096 processors of an IBM BG/Q machine maintained by the Argonne Leadership Computing Facility. The gain in using the CHAR/BDFk scheme is notable, where we see that it delivers results sooner than the EXTk/BDFk scheme. Early in the simulation (CA $< 363^o$), the CHAR/BDFk scheme shows little gain which is likely due to the piston velocity being relavitvely small. However, it's advantage becomes apparent at larger CA where we see a notable difference in the time-to-solution. The
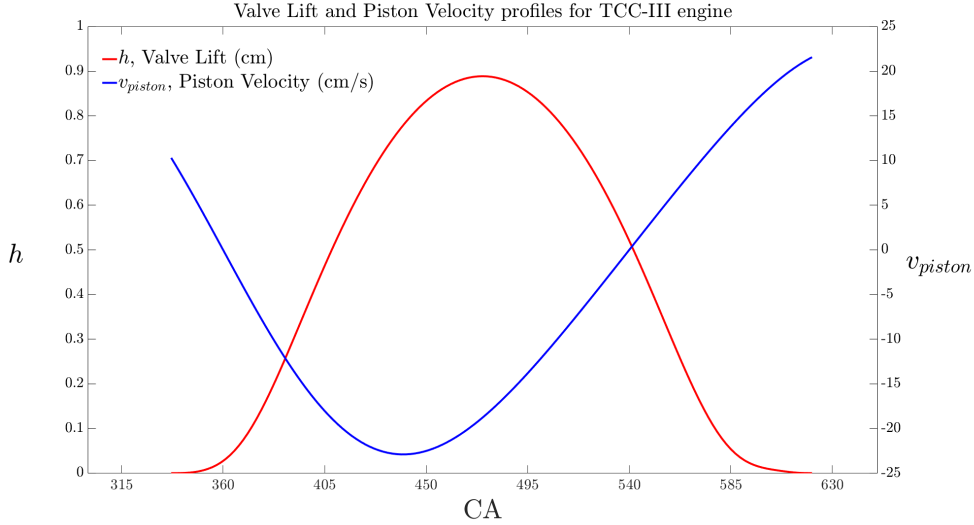
23

Figure 13: Intake-Valve and Piston surface motion. The red line indicates the valve-lift distance while the blue line shows the piston velocity. Both are functions of the crank-angle.

influence of the CHAR/BDFk scheme is particularly dramatic for $C = 1$ or 2. Indeed, figure 17, shows the speedup as a function of the crank-angle (CA) degree. In this work, we define speedup as the ratio

$$\text{speedup} \quad = \quad \frac{\text{Time-To-Solution (EXTk/BDFk)}}{\text{Time-To-Solution (CHAR/BDFk)}}, \tag{70}$$

where the numerator and denominator signify that the method being used is the noncharacteristic (EXTk/BDFk) and characteristic (CHAR/BDFk) method, respectively. At approximately 365° CA, the maximum speedup is achieved. Using CHAR/BDFk we gather the timing information at 375° CA and note the overall speedup of 2.7 (for target C=4) over the EXTk/BDFk scheme which is limited to a stable Courant number of $C = 0.5$. For a target Courant number of $C \geq 2$ we observe that a speedup factor of at least 2 is attained.

Figure 18 shows the change in the timestep Δt over the course of the simulation. As expected we can achieve larger (and more stable) timesteps with higher C (or M - number of Runge-Kutta timesteps). The large drop in the timestep size for CA > 364 is due to the large flow velocities which are moving through the valve-seat region. While the overall average timestep increases by a factor of approximately 8, the wall-time results show the effect of diminishing returns due, in part, to the additional number (M) of Runge-Kutta timesteps between each foot along the characteristic. In addition, the influence on wall time is affected by the number of pressure iterations, which can be seen in figure 19. Due to larger timesteps, the number of pressure iterations in the preconditioned conjugate gradient algorithm increases which adds to the overall computational work that is performed for each timestep. Nonetheless, these results illustrate the advantage of the characteristics/OIFS scheme, which shows a a reduction in computational time over standard methods by more than a factor of two.

0.4.6   A Pinched Pipe

In order to assess the limits and benefits of our characteristic-based approach, we conduct a simple exercise of a pinched or constricted pipe. A schematic can be seen in figure 20, where the axial boundary undergoes a radial constriction. The center of the pipe (along the axial direction) undergoes the maximum radial

24

constriction and the effect is observed linearly towards the ends (i.e. inlet and outlet). As a real-life analogue, one may imagine the local constriction of a garden hose as water is rushing through it. If we consider a incompressible fluid entering the pipe at a fixed flow rate then the constriction will increase the velocity of the fluid as it passes through the pipe. Computationally, a singularity in the mesh would occur if constriction proceed without bound. This would mean that $\Delta x_{min} \to 0$ as time proceeds. As such, the Courant number (56), would increase for a fixed timestep.

Our aim in this section is to assess if our implementation of the CHAR/BDFk approach can realize Courant numbers larger than the stable Courant numbers for EXTk/BDFk. In order to do this, we track the Courant number in time for the constricted pipe. We fix our timestep and change the number of RK4 substeps (i.e. M) that are being performed. Figure **??** compares the realized Courant number for a given temporal discretization. In each case, the Courant number is calculated at each timestep until the algorithm becomes numerically unstable and produces diverging/erroneous results. This occurs because of the mesh singularity in the mid-axial plane of the pipe. The top left image of figure **??** shows the realized Courant numbers for the EXTk/BDFk scheme. This plot shows how the solution becomes unstable at Courant numbers that are well below 10. In fact, the stable Courant number for EXTk/BDFk is known to be approximately 0.5. Courant numbers that are larger than 0.5 are past the stability limit and producing erroneous results. The other plots show achievable Courant numbers using the CHAR/BDFk scheme for $M = 1, 2, 3, 4$ and 8. As the plots show, a larger value for M will achieve a higher Courant number. It is worth noting that while the x-axis indicates the temporal variable, it is also an indication of the change in radius for the smallest cirucular cross-section (i.e. mid-axial plane) of the pipe. That said, these plots suggest that a larger M also provides the user with the means to capture flow-fields in the small grid limit without changing the timestep (Δt). To confirm that behavior, figure 21 shows the smallest gap that is being achieved for a given timestep and temporal discretization. Timestep values are the same as the ones that are used in figure 23. The gap (ordinate) compares the minimum radius to the original radius at the start of the simulation. The circular symbol shows the results for EXTk/BDFk which attains the largest gaps before numerical instability occurs. The other symbols show results for CHAR/BDFk. Larger values for M show that the smaller gaps are achieved for a given timestep. Figure 22 shows the maximum value for the axial component of the velocity versus time. Similar to figure 23, we plot this component until the solution becomes numerically unstable. We confirm that larger velocities are captured for the CHAR/BDFk schemes versus the EXTk/BDFk scheme.

0.5   Appendix

0.5.1   Outflow Boundary Condition

This section illustrates the unique outflow boundary condition that was used in the peristaltic pumping example of section 0.4.3. The so-called turbulent outflow boundary condition [48] is currently employed in order to avoid the locally negative flux at the outflow boundary. Such negative fluxes may occur in turbulent flows where vortices might allow for flow characteristics to enter into the domain rather than exit. As is noted in [48], the incoming characteristics can rapidly lead to numerical instability. In order to avoid this phenomena, a mean axial component to the velocity field is added to the layer of element adjacent to the outflow boundary. This is effectively accomplished by added a divergence in the velocity field to this layer of elements. In particular, this divergence is set so that it ramps from zero at the upstream end of the layer to a fixed positive value at the exit.

The effect on the flow field is only noticed at this last layer of elements and does not effect the other parts of the domain where the interest of the study remains. In the peris example, there are no incoming characteristics which lead to instabilities. We investigate if the turbulent outflow boundary condition does drastically affect the flow-field. Figure 24 shows the results for the persitaltic pumping example with the turbulent outflow boundary condition. In particular, the top row shows the axial velocity along the centerline of the domain. Each plot compares the EXTk/BDFk (IFCHAR = F) scheme with the CHAR/BDFk

(IFCHAR = T) scheme. The left image shows the situations when the turbulent outflow boundary condition is employed. In this image, can see the sharp rise in the axial velocity. The right image shows the smooth axial profile without the turbulent outflow boundary condtion. One can see that the upstream profile is unaffected by the additional divergence. The second row shows the maximum axial velocity in the entire domain over a given time period. When the special boundary condition is employed one can see the larger values for velocity which is due to the added divergence. Similar phenomena can be seen in Figure 25 for the pressure.

0.5.2 Material Derivative of the Jacobian

$$
\begin{aligned}
\frac{dJ(\mathbf{r},t)}{dt} &= \epsilon_{ijk}\frac{\partial}{\partial r_1}\left(\frac{dx_i}{dt}\right)\frac{\partial x_j}{\partial r_2}\frac{\partial x_k}{\partial r_3} + \epsilon_{ijk}\frac{\partial x_i}{\partial r_1}\frac{\partial}{\partial r_2}\left(\frac{dx_j}{dt}\right)\frac{\partial x_k}{\partial r_3} + \epsilon_{ijk}\frac{\partial x_i}{\partial r_1}\frac{\partial x_j}{\partial r_2}\frac{\partial}{\partial r_3}\left(\frac{dx_k}{dt}\right) \\
&= \epsilon_{ijk}\frac{\partial}{\partial r_1}\left(w_i\right)\frac{\partial x_j}{\partial r_2}\frac{\partial x_k}{\partial r_3} + \epsilon_{ijk}\frac{\partial x_i}{\partial r_1}\frac{\partial}{\partial r_2}\left(w_j\right)\frac{\partial x_k}{\partial r_3} + \epsilon_{ijk}\frac{\partial x_i}{\partial r_1}\frac{\partial x_j}{\partial r_2}\frac{\partial}{\partial r_3}\left(w_k\right) \\
&= \epsilon_{ijk}\frac{\partial w_i}{\partial x_m}\left(\frac{\partial x_m}{\partial r_1}\right)\frac{\partial x_j}{\partial r_2}\frac{\partial x_k}{\partial r_3} + \epsilon_{ijk}\frac{\partial x_i}{\partial r_1}\frac{\partial w_j}{\partial x_m}\left(\frac{\partial x_m}{\partial r_2}\right)\frac{\partial x_k}{\partial r_3} + \epsilon_{ijk}\frac{\partial x_i}{\partial r_1}\frac{\partial x_j}{\partial r_2}\frac{\partial w_k}{\partial x_m}\left(\frac{\partial x_m}{\partial r_3}\right) \\
&= \epsilon_{ijk}\frac{\partial w_1}{\partial x_1}\left(\frac{\partial x_i}{\partial r_1}\right)\frac{\partial x_j}{\partial r_2}\frac{\partial x_k}{\partial r_3} + \epsilon_{ijk}\frac{\partial x_i}{\partial r_1}\frac{\partial w_2}{\partial x_2}\left(\frac{\partial x_j}{\partial r_2}\right)\frac{\partial x_k}{\partial r_3} + \epsilon_{ijk}\frac{\partial x_i}{\partial r_1}\frac{\partial x_j}{\partial r_2}\frac{\partial w_3}{\partial x_3}\left(\frac{\partial x_k}{\partial r_3}\right) \\
&= \frac{\partial w_1}{\partial x_1}J(\mathbf{r},t) + \frac{\partial w_2}{\partial x_2}J(\mathbf{r},t) + \frac{\partial w_3}{\partial x_3}J(\mathbf{r},t) \\
&= (\nabla \cdot \mathbf{w})J(\mathbf{r},t) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (71)
\end{aligned}
$$

0.6 Conclusions

A new characteristic-based spectral element method has been introduced for incompressible flows in moving-domain problems. By adopting a taylored form of the operator-integration-factor-splitting (OIFS) scheme, this method decouples the nonlinear and linear terms in the unsteady Navier-Stokes equations and reduce timestepping constraints compared to the standard semi-implicit schemes. In conjunction with the $\mathbb{P}_N - \mathbb{P}_N$ formulation, we show 2nd- and 3rd-order temporal convergence rates for our characteristic-based scheme and demonstrate good agreement with physical experiments with small variation in the results as target Courant numbers are increased. We apply our method for direct numerical simulations (DNS) of turbulent flow fields to study the intake stroke of the modified TCC-III engine. Performance results for resolving fast-moving flow through the small-gap region of the intake valve from 362° to 375° CA indicate that the characteristic-based method can reduce the time to solution by more than a factor of 2.
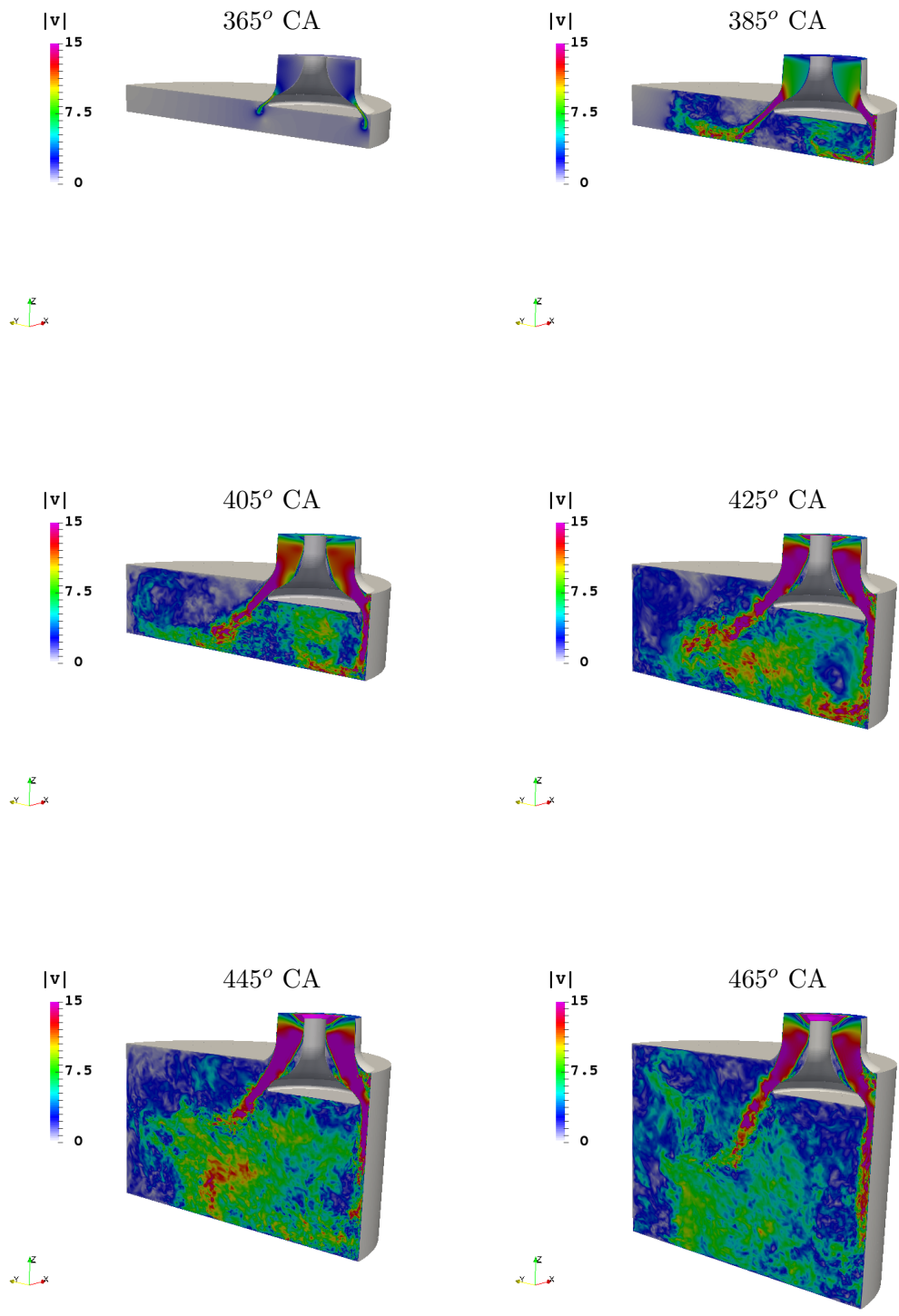
Figure 14: TCC-III Engine. Centerplane, non-dimensional velocity magnitude distribution for the $\mathbb{P}_N - \mathbb{P}_N$ approach. Results were determined using the CHAR/BDFk scheme with a CFL = 3. Top Row: 365°, and 385° CA (Left to Right); 405°, and 425° CA (Left to Right - Middle Row); 445° and 465° CA (Left to Right - Bottom Row)

364° CA 375° CA
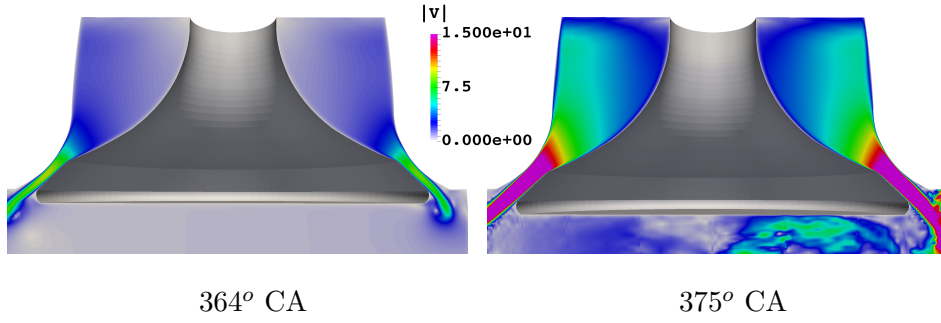
Figure 15: Closeup view of the intake valve. Centerplane, nondimensional velocity distributions during an intake stroke for an incompressible flow at CA 364° (left) and 375° (right). $Re = 5000$.
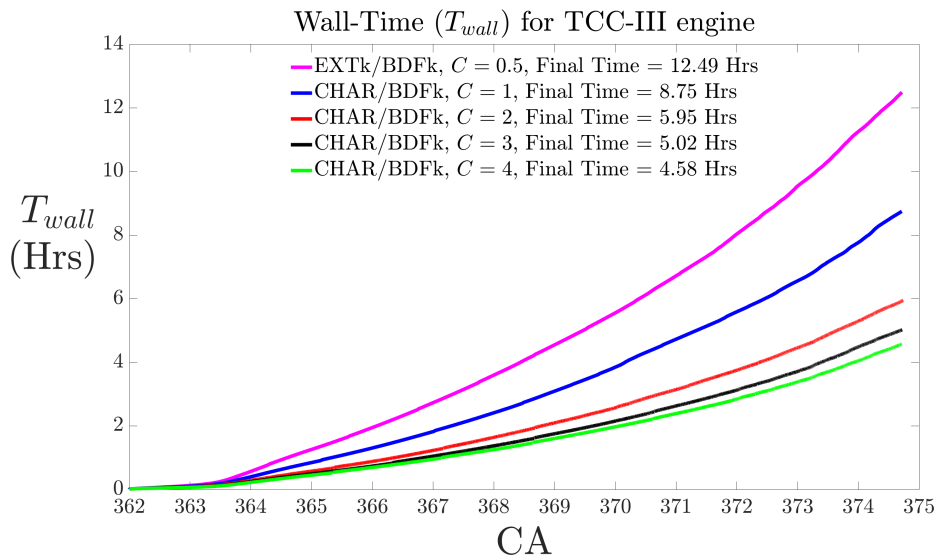
Figure 16: Wall Time (vertical-axis) since start of engine simulation (left). x-axis represents CA, crank-angle degrees. Polynomial order is $N = 11$ with a BDF3 temporal discretization. Calculations are carried out from 362° CA to 375° CA. The inset shows speedup for the CHAR/BDFk for consecutively large Courant number C. Speed-up is defined by Eq. (70). Note: piston velocity for CA < 364 is nearly zero due to TDC position of the piston (i.e. small piston velocity), thus contributing to comparable wall-time results for CA < 364. These results were found using 4096 processors on Cetus - an IBM BG/Q machine maintained by the Argonne Leadership Computing Facility.
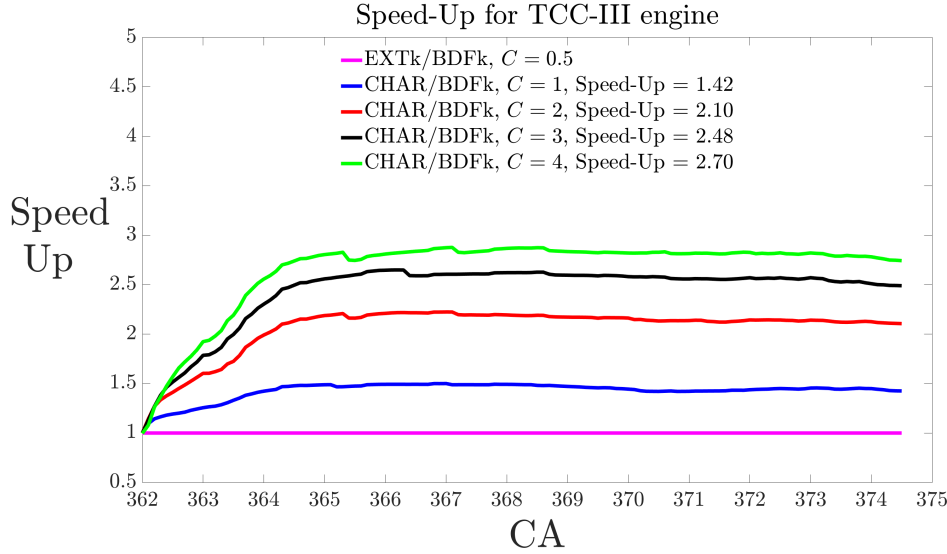
Figure 17: Speed-Up (vertical axis) since start of engine simulation (left). x-axis represents CA, crank-angle degrees. Polynomial Order is $N = 11$ with a BDF3 temporal discretization. Calculations are carried out from 362° CA to 375° CA. Speed-up is defined by Eq. (70). These results were found using 4096 processors on Cetus - an IBM BG/Q machine maintained by the Argonne Leadership Computing Facility.
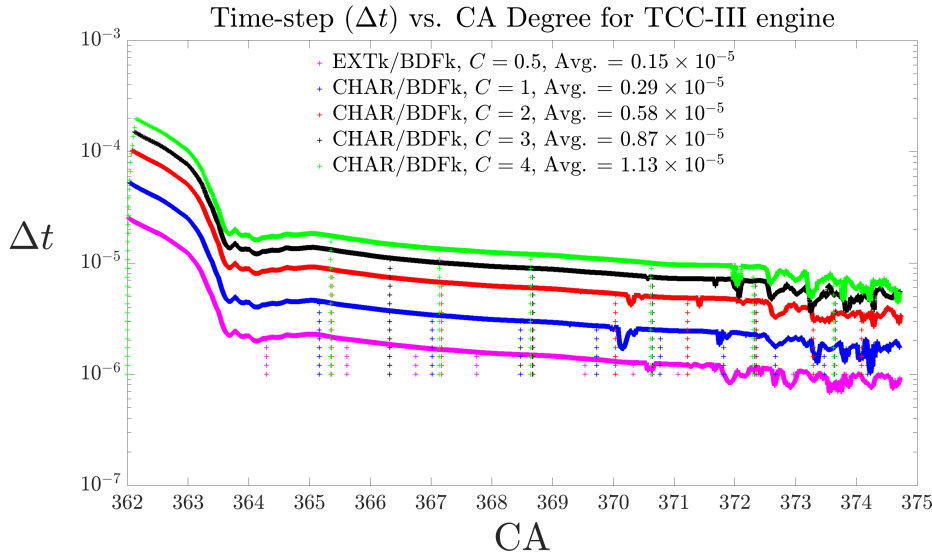
Figure 18: Δt (vertical-axis) as a function of CA. Results show how a variable timestepping implementation changes with the choice of C that is used for the CHAR/BDFk method. Polynomial order is $N = 11$ with a BDF3 temporal discretization.
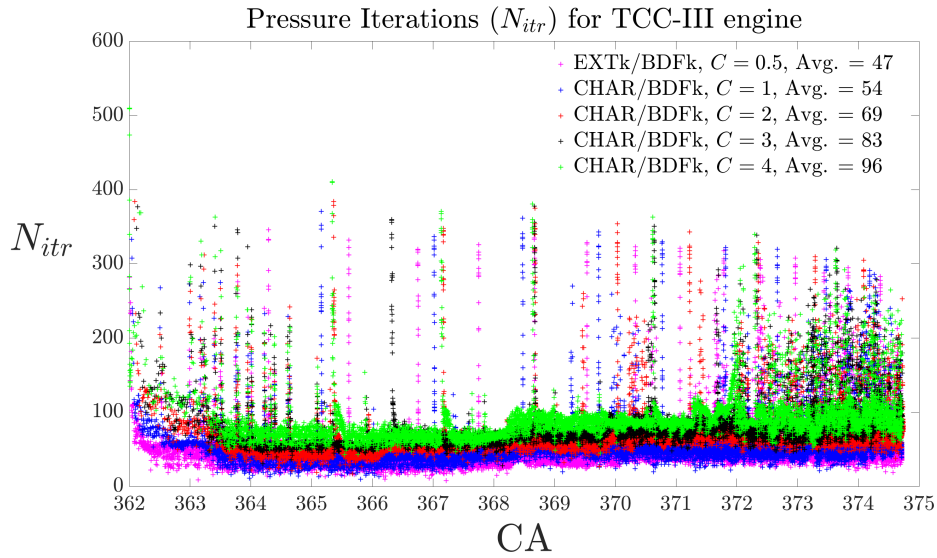
Figure 19: Pressure iterations as a function of crank-angle degree for 362°–375° CA. The peaks in the figure represent the effect of the restart procedure.
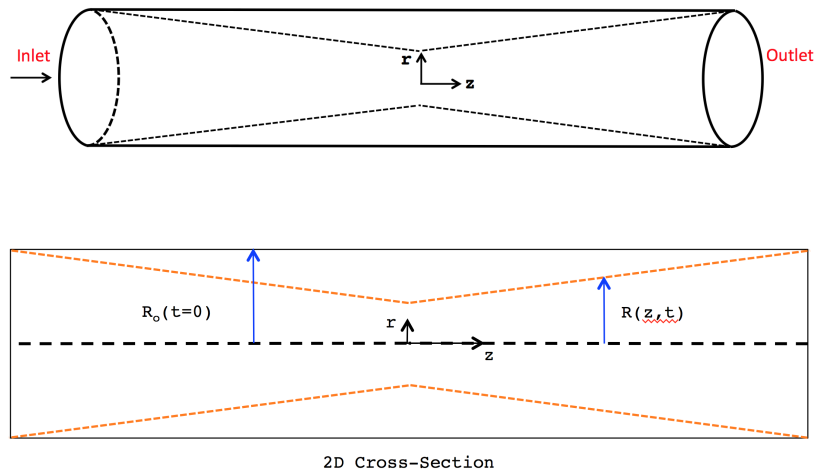
Figure 20: Schematic for a pinched pipe.
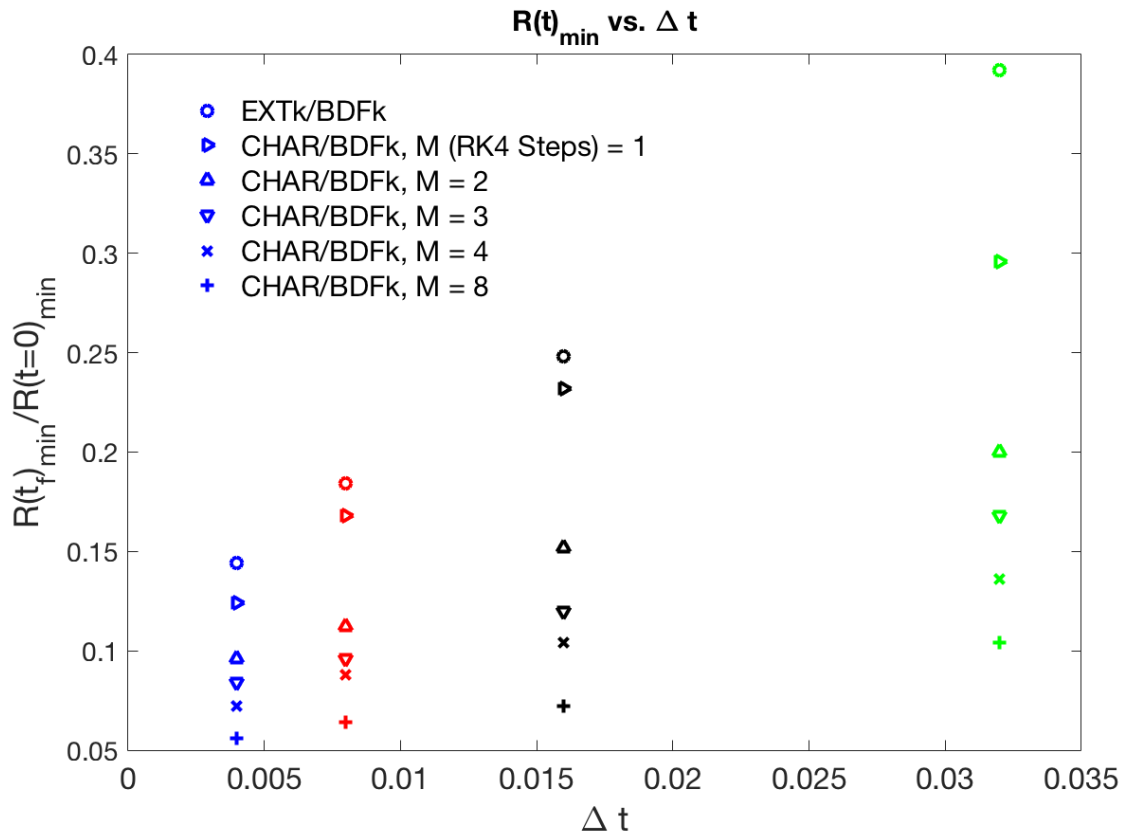
Figure 21: Constricted Pipe. Smallest achievable radius before time-stepping became unstable and produced diverging results.
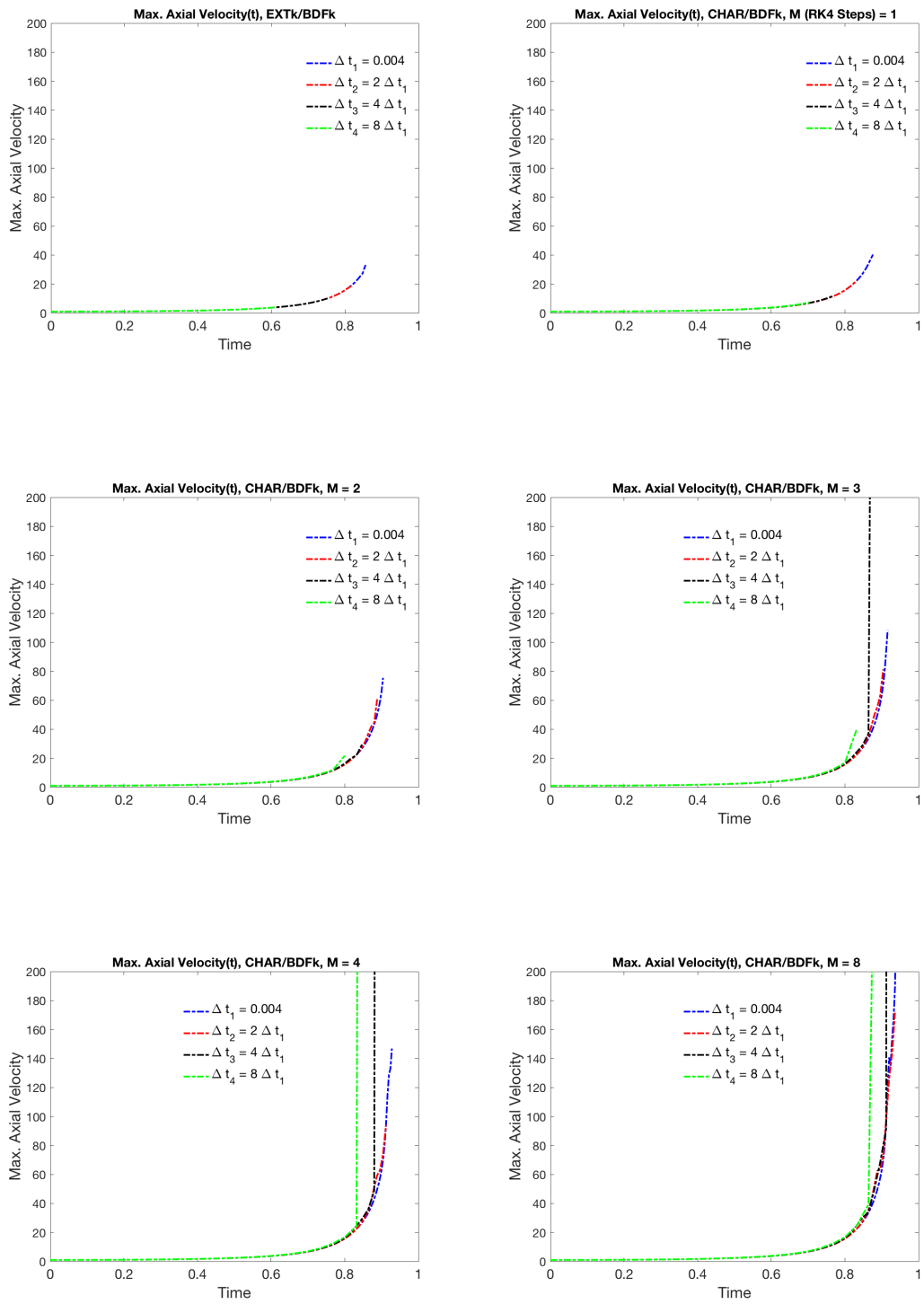
Figure 22: Constricted Pipe. Axial Velocity versus time.
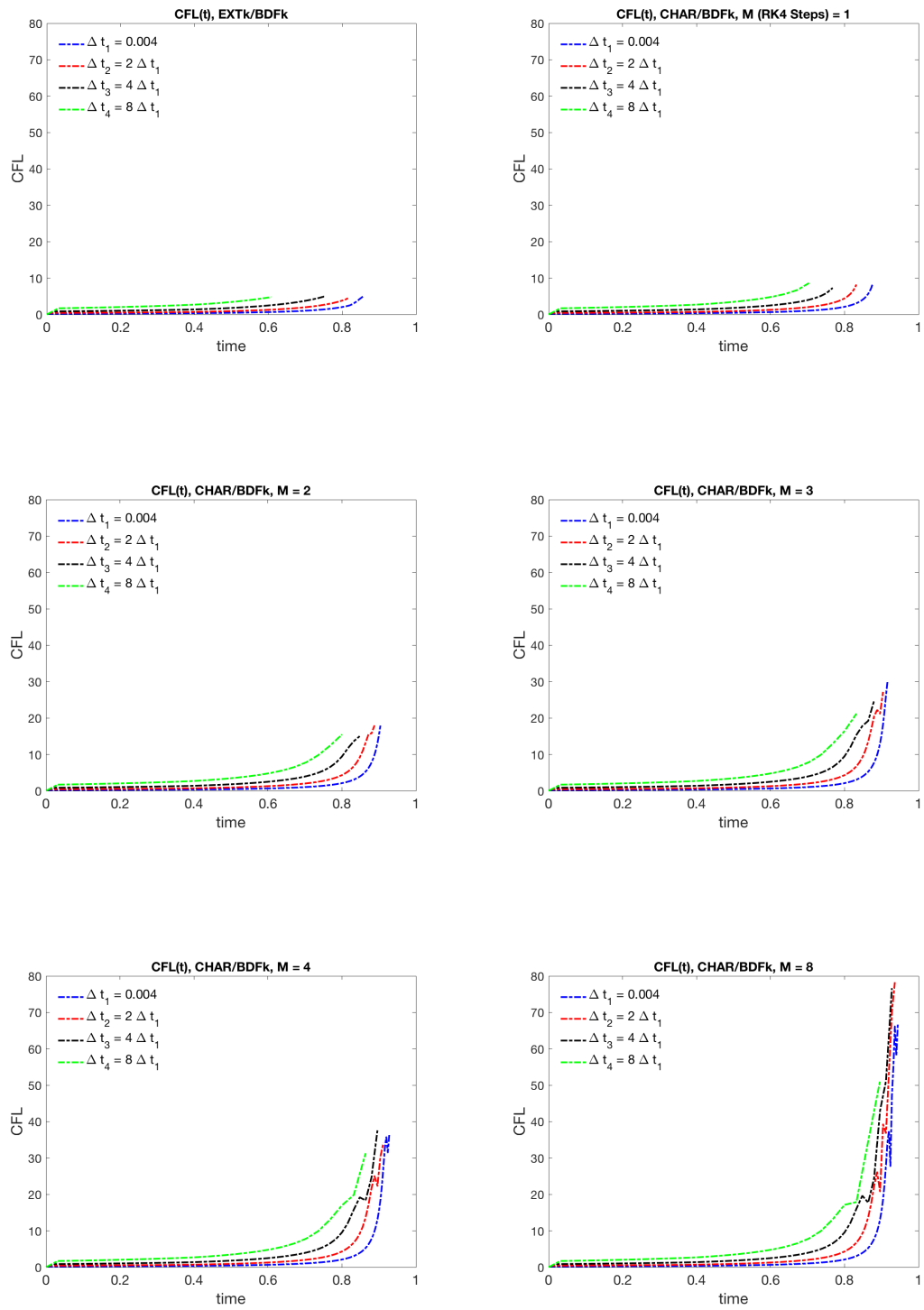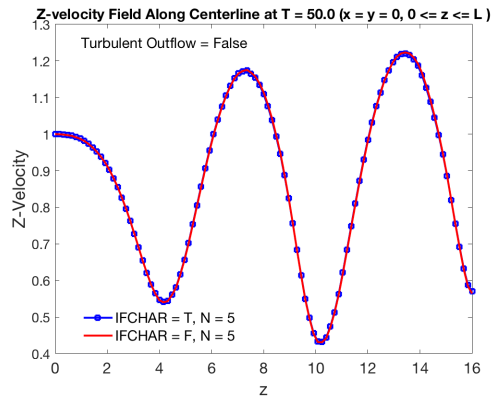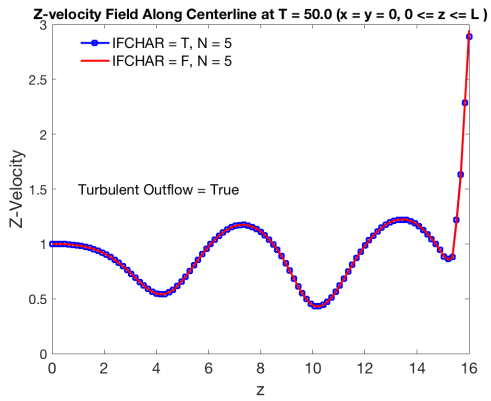
Figure 23: Constricted Pipe. CFL versus time.
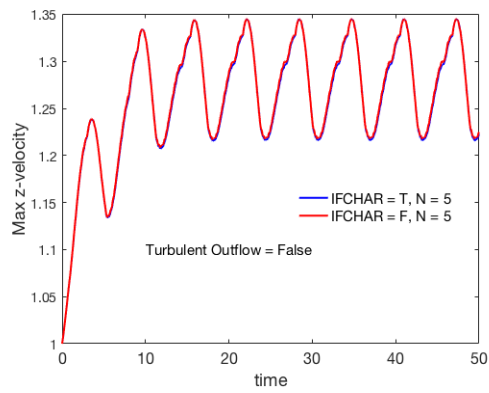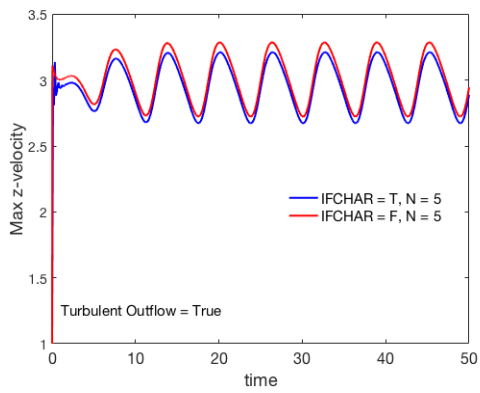
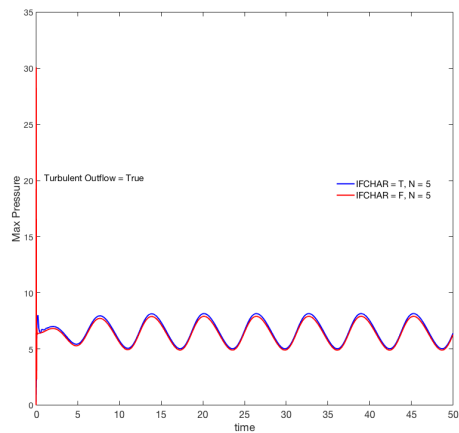Figure 24: Turbulent Outflow Boundary Condition Phenomena: The effect on axial velocity for the peristaltic pump.
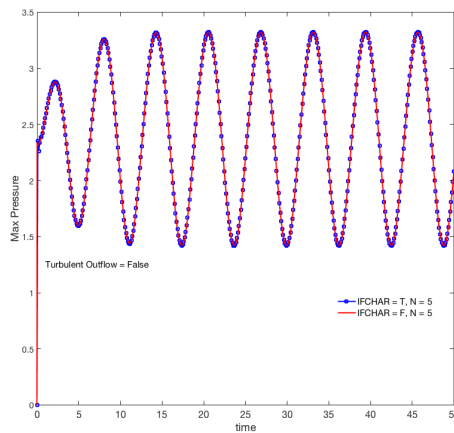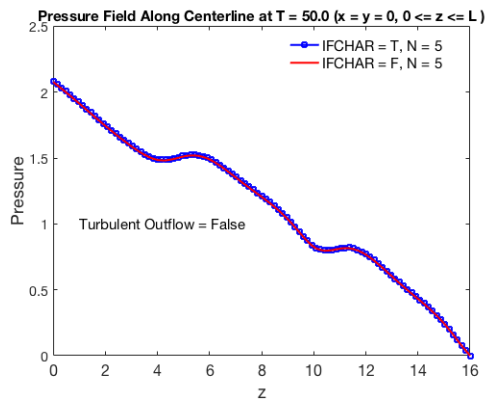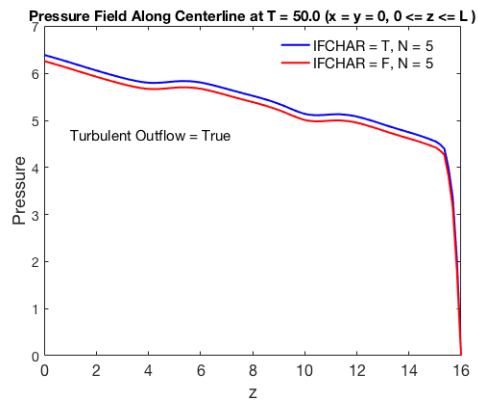
Figure 25: Turbulent Outflow Boundary Condition Phenomena: The effect on pressure for the peristaltic pump.

Bibliography

[1] I. Demirdžić, M. Perić, Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries, International Journal for Numerical Methods in Fluids **10**(7), 771 (1990)

[2] T. Tezduyar, M. Behr, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces–the dsd/st procedure: I. the concept and the preliminary numerical tests, Computer Methods in Applied Mechanics and Engineering **94**(3), 339 (1992)

[3] T.E. Tezduyar, M. Behr, S. Mittal, J. Liou, A new strategy for finite element computations involving moving boundaries and interfacesthe deforming-spatial-domain/space-time procedure: Ii. computation of free-surface flows, two-liquid flows, and flows with drifting cylinders, Computer methods in applied mechanics and engineering **94**(3), 353 (1992)

[4] J. Donea, A. Huerta, J.P. Ponthot, A. Rodrguez-Ferran, Arbitrary LagrangianEulerian methods, Encyclopedia of computational mechanics **DOI: 10.1002/0470091355.ecm009**, 1:14 (2004)

[5] C. Arcoumanis, J. Whitelaw, Fluid mechanics of internal combustion enginesa review, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science **201**(1), 57 (1987)

[6] T.W. Kuo, X. Yang, V. Gopalakrishnan, Z. Chen, Large eddy simulation (les) for ic engine flows, Oil Gas Sci. Technol.–Rev. IFP Energies nouvelles (2012)

[7] P. Schiffmann, S. Gupta, D. Reuss, V. Sick, X. Yang, T.W. Kuo, Tcc-iii engine benchmark for large-eddy simulation of ic engine flows, Oil & Gas Science and Technology **71**(1) (2016)

[8] C. Rutland, Large-eddy simulations for internal combustion engines–a review, International Journal of Engine Research p. 1468087411407248 (2011)

[9] M. Schmitt, C.E. Frouzakis, A.G. Tomboulides, Y.M. Wright, K. Boulouchos, Direct numerical simulation of multiple cycles in a valve/piston assembly, Physics of Fluids **26**(3), 035105 (2014)

[10] M. Schmitt, C. Frouzakis, Y. Wright, A. Tomboulides, K. Boulouchos, Investigation of wall heat transfer and thermal stratification under engine-relevant conditions using DNS, Int. J. of Engine Res. **17**(1), 63 (2016)

[11] A. Patera, A spectral element method for fluid dynamics : laminar flow in a channel expansion, J. Comput. Phys. **54**, 468 (1984)

[12] L. Ho, A Legendre spectral element method for simulation of incompressible unsteady viscous free-surface flows. Ph.D. thesis, Massachusetts Institute of Technology (1989). Cambridge, MA.

[13] L. Ho, Y. Maday, A. Patera, E. Rønquist, A high-order Lagrangian-decoupling method for the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. **80**, 65 (1990)

[14] L. Ho, A. Patera, A Legendre spectral element method for simulation of unsteady incompressible viscous free-surface flows, Comput. Methods Appl. Mech. Engng. **80**, 355 (1990)

[15] P. Fischer, M. Schmitt, A. Tomboulides, Recent developments in spectral element simulations of moving-domain problems, Fields Institute Communications: Recent Progress and Modern Challenges in Applied Mathematics, Modelling and Computational Science **Springer/Fields Institute** (2016)

[16] S. Orszag, M. Israeli, M. Deville, Boundary conditions for incompressible flows., J. Sci. Comp. **1**, 75 (1986)

[17] Y. Maday, A. Patera, E. Rønquist, An operator-integration-factor splitting method for time-dependent problems: Application to incompressible fluid flow, J. Sci. Comput. **5**, 263 (1990)

[18] A. Tomboulides, M. Israeli, G. Karniadakis, Efficient removal of boundary-divergence errors in time-splitting methods, J. Sci. Comput. **4**, 291 (1989)

[19] J. Perot, An analysis of the fractional step method, J. Comp. Phys. **108**, 51 (1993)

[20] W. Couzy, Spectral element discretization of the unsteady Navier-Stokes equations and its iterative solution on parallel computers. Ph.D. thesis, Swiss Federal Institute of Technology-Lausanne (1995). Thesis nr. 1380

[21] P. Fischer, An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations, J. Comput. Phys. **133**, 84 (1997)

[22] P. Fischer, J. Lottes, in *Domain Decomposition Methods in Science and Engineering Series*, ed. by R. Kornhuber, R. Hoppe, J. Priaux, O. Pironneau, O. Widlund, J. Xu (Springer, Berlin, 2004)

[23] J.W. Lottes, P.F. Fischer, Hybrid multigrid/Schwarz algorithms for the spectral element method, J. Sci. Comput. **24**, 45 (2005)

[24] H. Tufo, P. Fischer, Fast parallel direct solvers for coarse-grid problems, J. Parallel Distrib. Comput. **61**, 151 (2001)

[25] P. Fischer, J. Lottes, W. Pointer, A. Siegel, Petascale algorithms for reactor hydrodynamics, J. Phys. Conf. Series **125**, 012076 (2008)

[26] J. Lottes, Independent quality measures for symmetric AMG components. Tech. Rep. ANL/MCS-P1820-0111, Argonne National Laboratory, Argonne, IL, USA (2011)

[27] J. Boyd, Two comments on filtering for Chebyshev and Legendre spectral and spectral element methods, J. Comput. Phys. **143**, 283 (1998)

[28] P. Fischer, J. Mullen, Filter-based stabilization of spectral element methods, Comptes rendus de l'Académie des sciences, Série I- Analyse numérique **332**, 265 (2001)

[29] J. Malm, P. Schlatter, P. Fischer, D. Henningson, Stabilization of the spectral-element method in convection dominated flows by recovery of skew symmetry, J. Sci. Comp. **57**, 254 (2013)

[30] H. Tufo, P. Fischer, in *Proc. of the ACM/IEEE SC99 Conf. on High Performance Networking and Computing, Gordon Bell Prize* (IEEE Computer Soc., CDROM, 1999)

[31] M. Deville, P. Fischer, E. Mund, *High-Order Methods for Incompressible Fluid Flow* (Cambridge University Press, Cambridge, 2002)

[32] O. Pironneau, On the transport-diffusion algorithm and its applications to the Navier-Stokes equations, Numer. Math. **38**, 309 (1982)

[33] A. Tomboulides, J. Lee, S. Orszag, Numerical simulation of low Mach number reactive flows, Journal of Scientific Computing **12**, 139 (June 1997)

[34] P. Fischer, Projection techniques for iterative solution of $A\underline{x} = \underline{b}$ with successive right-hand sides, Comput. Methods Appl. Mech. Engrg. **163**, 193 (1998)

[35] P. Fischer, N. Miller, H. Tufo, in *Parallel Solution of Partial Differential Equations*, ed. by P. Bjørstad, M. Luskin (Springer, Berlin, 2000), pp. 158–180

[36] S. Orszag, Spectral methods for problems in complex geometry, J. Comput. Phys. **37**, 70 (1980)

[37] P. Fischer, Spectral element solution of the Navier-Stokes equations on high performance distributed-memory parallel processors. Ph.D. thesis, Massachusetts Institute of Technology (1989). Cambridge, MA.

[38] Y. Maday, A. Patera, in *State-of-the-Art Surveys in Computational Mechanics*, ed. by A. Noor, J. Oden (ASME, New York, 1989), pp. 71–143

[39] P. Fischer, in *22nd AIAA Computational Fluid Dynamics Conference, AIAA Aviation* (AIAA 2015-3049, 2015)

[40] O. Walsh, in *The NSE II-Theory and Numerical Methods*, ed. by J. Heywood, K. Masuda, R. Rautmann, V. Solonikkov (Springer, 1992), pp. 306–309

[41] T. Pedley, K. Stephanoff, Flow along a channel with a time-dependent indentation in one wall: the generation of vorticity waves, Journal of Fluid Mechanics **160**, 337 (1985)

[42] M. Ralph, T. Pedley, Flow in a channel with a moving indentation, Journal of Fluid Mechanics **190**, 87 (1988)

[43] L. Ho, A. Patera, A Legendre spectral element method for simulation of unsteady incompressible viscous free-surface flows, Comput. Methods Appl. Mech. Engrg. **80**, 355 (1990)

[44] L. Ho, A Legendre spectral element method for simulation of incompressible unsteady viscous free-surface flows. Ph.D. thesis, Massachusetts Institute of Technology (1989). Cambridge, MA.

[45] V. Sick, D. Reuss, X. Yang, T.W. Kuo. TCC-III CFD Input Dataset. https://deepblue.lib.umich.edu/handle/2027.42/108382

[46] A. Masud, T.J.R. Hughes, A space-time galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems, Comput. Methods Appl. Mech. Engrg. **146**, 91 (1997)

[47] H. Kanchi, A. Masud, A 3d adaptive mesh moving scheme, Int. J. Numer. Methods Fluids **54**, 923 (2007)

[48] P. Fischer, F. Loth, S. Lee, D. Smith, H. Bassiouny, Simulation of high Reynolds number vascular flows, Comput. Methods Appl. Mech. Engrg. **196**, 3049 (2007)

Acknowledgments

Argonne Leadership Computing Facility
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 240
Argonne, IL 60439

www.anl.gov