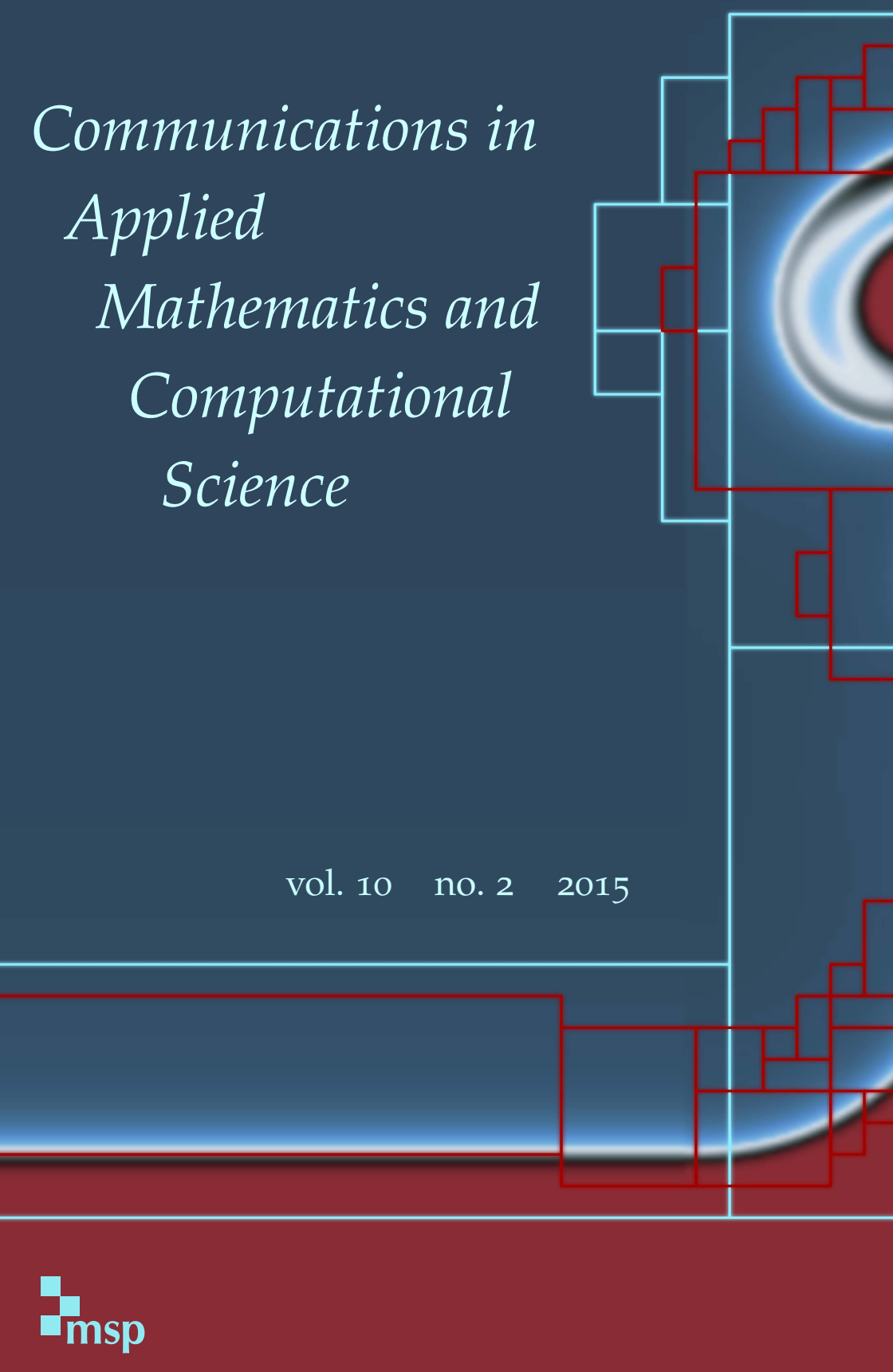

Communications in
Applied
Mathematics and
Computational
Science
vol. 10 no. 2 2015
msp

Communications in Applied Mathematics and Computational Science

msp.org/camcos

See inside back cover or msp.org/camcos for submission instructions.

The subscription price for 2015 is US $85/year for the electronic version, and $120/year (+$15, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to MSP.

Communications in Applied Mathematics and Computational Science (ISSN 2157-5452 electronic, 1559-3940 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer review and production are managed by EditFLOW® from MSP.

PUBLISHED BY

mathematical sciences publishers

nonprofit scientific publishing

http://msp.org/

COMM. APP. MATH. AND COMP. SCI.
Vol. 10, No. 2, 2015
dx.doi.org/10.2140/camcos.2015.10.97

A NITSCHE-BASED CUT FINITE ELEMENT METHOD FOR A FLUID-STRUCTURE INTERACTION PROBLEM

ANDRÉ MASSING, MATS G. LARSON,
ANDERS LOGG AND MARIE E. ROGNES

We present a new composite mesh finite element method for fluid-structure interaction problems. The method is based on surrounding the structure by a boundary-fitted fluid mesh that is embedded into a fixed background fluid mesh. The embedding allows for an arbitrary overlap of the fluid meshes. The coupling between the embedded and background fluid meshes is enforced using a stabilized Nitsche formulation that allows us to establish stability and optimal-order a priori error estimates. We consider here a steady state fluid-structure interaction problem where a hyperelastic structure interacts with a viscous fluid modeled by the Stokes equations. We evaluate an iterative solution procedure based on splitting and present three-dimensional numerical examples.

1. Introduction

In fluid-structure interaction applications, the underlying geometry of the computational domain may change significantly due to displacement of the structure. In order to deal with this situation in a standard setting with conforming elements, a mesh motion algorithm must be used. If the displacements are significant, the deformation of the mesh may lead to deteriorating mesh quality, which may ultimately require remeshing of the computational domain. Alternative, more flexible, techniques are therefore of significant practical interest.

In this paper, we consider a combination of standard moving meshes and so-called CutFEM technology [8]. Essentially, the structure or elastic solid is first embedded into a boundary-fitted fluid mesh that moves along with the deformation of the solid to keep the fluid-structure interface intact. The motion of the fluid mesh surrounding the structure is obtained by solving an elasticity problem with given displacement at the fluid-structure interface. The boundary-fitted fluid mesh is then embedded into a fixed background mesh where we allow for an arbitrary overlap of

MSC2010: 65N12, 65N30, 74B20, 76D07, 65N85.
Keywords: fluid-structure interaction, overlapping meshes, cut finite element method, embedded meshes, stabilized finite element methods, Nitsche's method.

the fluid meshes in order to facilitate the repositioning of the moving fluid mesh within the fixed background mesh. The fluid is then discretized on both the moving overlapping domain, using an arbitrary-Lagrangian–Eulerian-type (ALE) approach [14; 15], and on the fixed background mesh, using a standard discretization posed in an Eulerian frame.

The coupling at the fluid-fluid interface between the overlapping and underlying fluid meshes is handled using a stabilized Nitsche method developed for the Stokes problem in [41]. The stabilization is constructed in such a way that the resulting scheme is inf-sup stable and the resulting stiffness matrix is well-conditioned independent of the position of the overlapping fluid mesh relative to the fixed background fluid mesh. As a result, optimal-order error estimates are also established. In order to deal with the cut elements arising at the interface, we compute the polyhedra resulting from the intersections between the overlapping and background meshes. These polyhedra may then be described using a partition into tetrahedra; this partition may in turn be used to perform numerical quadrature. We refer to [39] for a detailed discussion of the implementation aspects of cut element techniques in three spatial dimensions. We remark that Nitsche-based formulations for Stokes boundary and interface problems where the surface in question is described independently of a single, fixed background mesh were proposed in [10; 40; 25; 9]. A Nitsche-based composite mesh method was first introduced for elliptic problems in [23].

One may also consider formulations where the structure is described via its moving boundary, which is immersed into a fixed background fluid mesh. Prominent examples are Cartesian grid methods, e.g., [42], the classical immersed boundary method introduced by Peskin [44; 45], its finite element pendant proposed in [7; 57; 56] and formulations based on Lagrange multipliers [57; 55; 20; 19; 46] and on Nitsche's method [24]. However, the use of an additional boundary-fitted fluid mesh as in the current work is attractive since it allows for the resolution of boundary layers and computation of accurate boundary stresses. Often, the construction of the surrounding fluid mesh can easily be generated by extending the boundary mesh in the normal direction. We plan to further investigate the properties of the fluid-structure coupling in future work.

As our proposed scheme combines an ALE-based discretization on the fluid mesh surrounding the structure with an Eulerian-based discretization on the fixed background fluid mesh, it can be classified as a hybrid Eulerian-ALE or Chimera approach. Such hybrid schemes are built upon the concept of overlapping meshes introduced for finite difference and finite volume schemes in the early works of Volkov [52], Starius [48; 49] and Steger et al. [50] and later by Chesshire and Henshaw [12] and Aftosmis et al. [1], where the primary concern was to ease the burden of mesh generation by composing individually meshed, static geometries. The idea of gluing meshes together was then explored for finite element methods

by Cebral and Löhner [11] and Löhner et al. [37; 36] to study the flow around independently meshed complex objects such as cars, collections of buildings or stents in aortic vessels. In these works, relatively simple interpolation schemes were used to communicate the solution between overlapping meshes. To achieve a physically more consistent coupling between the solution parts presented on different domains, Schwarz-type domain iteration schemes using Dirichlet–Neumann and Robin coupling on overlapping domains have been proposed for the Navier–Stokes equations in [27]. A completely different route was taken by Day and Bochev [13], who reformulated elliptic interface problems as suitable first-order systems augmented with least-square stabilizations to enforce the interface conditions between the mesh domains to be tied together.

Introducing special interpolation stencils close to the fluid-fluid interface, a finite-volume-based Chimera method for flow problems involving multiple moving rigid bodies was formulated in [54; 18] and [26], where higher-order Godunov fluxes were used. This method was then extended by Banks et al. [6] to deal with (linearly) elastic solids in two space dimensions and thus represents an instance of a hybrid ALE-fixed-grid method. This approach has barely been explored in the context of finite element methods for fluid-structure interaction problems: Wall et al. [53] and later Shahmiri et al. [47] used interpolation between fluid meshes and extended finite element techniques to couple fluid-fluid meshes and Baiges and Codina [5] introduced an auxiliary ALE step to convect information on the fixed background mesh between two consecutive time-steps.

In contrast to these contributions, our method is based on a variational finite element approach that leads to a monolithic and physically consistent coupling between the overlapping and underlying fluid meshes, which eliminates the need to introduce inconsistent interpolation operators. In addition, opposed to similar finite-element-based approaches presented, e.g., in [53; 47], our scheme used for the fluid problem is proven stable and optimally convergent, even for higher-order elements, independent of the location of the interface as shown in [41]. Thus, the new scheme for the fluid-structure interaction problem proposed in this work exhibits the necessary robustness that is essential for developing reliable hybrid ALE-fixed-mesh methods.

In the current work, we consider the steady state deformation of a hyperelastic solid immersed into a viscous fluid governed by the Stokes equations. We solve for the steady state solution using a fixed-point iteration where in each iteration the fluid, solid and mesh motion problems are solved sequentially. We present two numerical examples in three dimensions, including one example with a manufactured reference solution.

The outline of the remainder of this paper is as follows. In Section 2, we summarize the governing equations of the fluid-structure interaction (FSI) problem.

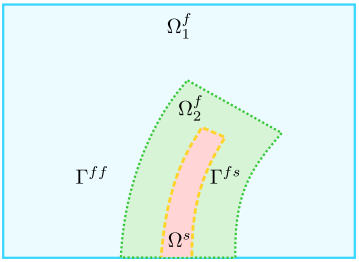


Figure 1. Fluid and structure domains for the stationary fluid-structure interaction problem.

In Section 3, we describe the overlapping mesh method. In Section 4, we present an algorithm for the solution of the stationary fluid-structure interaction model problem. In Section 5, we present three-dimensional numerical examples before drawing some conclusions in Section 6.

2. A stationary fluid-structure interaction problem

We consider a fluid-structure interaction problem posed on a domain $\Omega = \Omega^f \cup \Omega^s$ where Ω^f is the domain occupied by the fluid and Ω^s is the domain occupied by the solid. We assume that both Ω^f and Ω^s are open and bounded and that they are such that $\Omega^f \cap \Omega^s = \varnothing$. Furthermore, we decompose the fluid domain into two disjoint subdomains Ω_1^f and Ω_2^f such that $\Omega^f = \Omega_1^f \cup \Omega_2^f$. Here, Ω_2^f represents a part of the fluid domain surrounding the solid domain Ω^s; more precisely, we assume that $\partial\Omega_1^f \cap \partial\Omega^s = \varnothing$. The fluid-structure interface and the interface between the two fluid domains are denoted respectively by

$$\Gamma^{fs} = \partial\Omega_2^f \cap \partial\Omega^s \quad \text{and} \quad \Gamma^{ff} = \partial\Omega_1^f \cap \partial\Omega_2^f.$$

Here, the topological boundary ∂X for any given set X is defined by $\partial X = \overline{X} \setminus \mathring{X}$ where $\overline{X}$ and $\mathring{X}$ denote the closure and interior of X, respectively. For simplicity, we assume that the fluid domain boundary consists of two disjoint parts $\partial\Omega^f = \Gamma^{fs} \cup \partial\Omega_D^f$ and that the solid domain boundary decomposes in a similar manner: $\partial\Omega^s = \Gamma^{fs} \cup \partial\Omega_D^s$. This notation is summarized in Figure 1.

We assume that the fluid dynamics are governed by the Stokes equations of the following form: find the fluid velocity $\boldsymbol{u}^f : \Omega^f \to \mathbb{R}^3$ and the fluid pressure $p^f : \Omega^f \to \mathbb{R}$ such that

$$-\nabla \cdot (\nu^f \nabla \boldsymbol{u}^f - p^f \boldsymbol{I}) = \boldsymbol{f}^f \quad \text{in } \Omega^f, \tag{2-1}$$

$$\nabla \cdot \boldsymbol{u}^f = 0 \quad \text{in } \Omega^f, \tag{2-2}$$

where $\boldsymbol{f}^f$ is a given body force and ν^f is the fluid viscosity.

Next, we assume that the velocity is prescribed on both the fluid-structure interface and on the remainder of the fluid boundary:

$$\boldsymbol{u}^f = 0 \quad \text{on } \Gamma^{fs}, \tag{2-3}$$

$$\boldsymbol{u}^f = \boldsymbol{g}^f \quad \text{on } \partial\Omega^f_D. \tag{2-4}$$

Moreover, we enforce the continuity of the fluid velocity and of the fluid "stress" on the fluid-fluid interface by the following conditions:

$$[\boldsymbol{u}^f] = 0 \quad \text{on } \Gamma^{ff}, \tag{2-5}$$

$$[(\nu^f \nabla \boldsymbol{u}^f - p^f \boldsymbol{I}) \cdot \boldsymbol{n}] = 0 \quad \text{on } \Gamma^{ff}. \tag{2-6}$$

Here $[v] = v_1 - v_2$ denotes the jump in a function (or each component of a vector field) v over the interface Γ^{ff} where $v_i = v|_{\Omega^f_i}$ denotes the restriction of v to Ω^f_i for $i = 1, 2$. Furthermore, $\boldsymbol{n}$ is the unit normal of Γ^{ff} directed from Ω^f_2 into Ω^f_1.

Correspondingly, we assume that the structure deforms as an elastic solid satisfying the following equations: find $\boldsymbol{u}^s : \Omega^s \to \mathbb{R}^3$ such that

$$-\nabla \cdot \boldsymbol{\sigma}^s(\boldsymbol{u}^s) = \boldsymbol{f}^s \quad \text{in } \Omega^s, \tag{2-7}$$

where $\boldsymbol{\sigma}^s$ is the (Cauchy) stress tensor and $\boldsymbol{f}^s$ is a given body force. The precise form of the Cauchy stress tensor will depend on the choice of the elastic constitutive relation. In later sections, we will consider both linearly elastic and hyperelastic constitutive equations relating the displacement to the stress. As boundary conditions, we assume that the displacement of the structure is given on part of the boundary and that the structure experiences a boundary traction $\boldsymbol{t}^s_N$ on the fluid-structure interface:

$$\boldsymbol{u}^s = \boldsymbol{g}^s_D \quad \text{on } \partial\Omega^s_D, \tag{2-8}$$

$$\boldsymbol{\sigma}^s(\boldsymbol{u}^s) \cdot \boldsymbol{n} = \boldsymbol{t}^s_N \quad \text{on } \Gamma^{fs}. \tag{2-9}$$

The coupling between the fluid and the structure problems requires the fluid and solid stresses and velocities to be in equilibrium at the interface Γ^{fs}. In the stationary case considered here, these kinematic and kinetic continuity conditions are taken care of by ensuring that (2-3) and

$$\boldsymbol{t}^s_N = \boldsymbol{\sigma}^f(\boldsymbol{u}^f) \cdot \boldsymbol{n} \tag{2-10}$$

hold, where $\boldsymbol{\sigma}^f$ is the fluid stress tensor $\boldsymbol{\sigma}^f(\boldsymbol{u}^f, p^f) = 2\nu^f \epsilon(\boldsymbol{u}^f) - p^f \boldsymbol{I}$ and $\epsilon(\boldsymbol{u}^f)$ is the symmetric gradient $\epsilon(\boldsymbol{u}^f) = \frac{1}{2}(\nabla \boldsymbol{u}^f + \nabla(\boldsymbol{u}^f)^{\mathrm{T}})$.

In summary, the stationary fluid-structure interaction problem considered in this work is completely described by the set of equations (2-1)–(2-10).

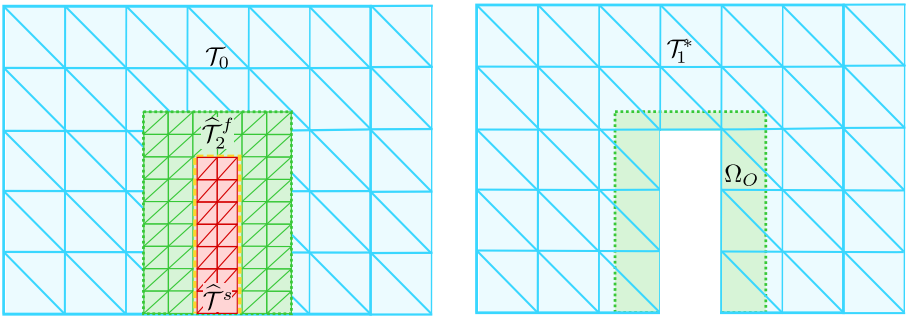


Figure 2. Chimera mesh configuration of the computational domain in the starting step of the fixed-point iteration. Left: fixed fluid background mesh $\mathcal{T}_0$ overlapped by the structure mesh $\widehat{\mathcal{T}}^s$ and a surrounding fitting fluid mesh $\widehat{\mathcal{T}}_2^f$. Right: reduced fluid background mesh $\mathcal{T}_1^*$ and fluid overlap region Ω_O.

3. An overlapping finite element discretization of the FSI problem

The nonlinear nature of the fluid-structure interaction problem (2-1)–(2-10) mandates a nonlinear solution scheme such as a Newton-type or fixed-point method. A classical and well-studied approach is to decompose the coupled problem into separate systems of equations via a Dirichlet–Neumann fixed-point iteration [43; 32; 31]. This is also the route taken here. Alternatively, more sophisticated iteration schemes based on a Robin-type reformulation of the interface conditions (2-3), (2-9) and (2-10) might be employed; see for instance [3; 4]. The basic idea of the Dirichlet–Neumann fixed-point iteration is to start with solving the fluid problem (2-1)–(2-6) on a given starting domain. The resulting fluid boundary traction acting on the fluid-structure interface then serves as Neumann data for the structure problem (2-7)–(2-10). The structure deformation dictates a displacement of the fluid domain boundary and, in turn, a new configuration of the fluid domain. This sequence of steps is repeated until convergence.

Each of the three subproblems (the fluid problem, the structure problem and the domain deformation) will be solved numerically using separate finite element discretizations. Overall, we will employ an overlapping mesh method in which a fixed background mesh is used for part of the fluid domain and a moving mesh is used for the combination of the structure domain and its surrounding fluid domain. We note that methods based on overlapping meshes (as the one considered here) are sometimes also called Chimera methods. Before describing each of the discretizations, we present an overview of the setup of the computational domains.

For simplicity, we assume that the computational domain Ω is fixed throughout the fixed-point iteration while the fluid and structure subdomains will be updated in each iteration step. In each step, we consider the following setup, illustrated in Figure 2, of the computational domains. First, we assume that Ω is tessellated by

a background mesh $\mathcal{T}_0$. Second, we assume that the current representation of the subdomains Ω_2^f and Ω^s are tessellated by meshes $\mathcal{T}_2^f$ and $\mathcal{T}^s$, respectively, and that these meshes match at their common interface. As a result, $\mathcal{T}^{fs} = \mathcal{T}_2^f \cup \mathcal{T}^s$ defines an admissible and conforming mesh of the combined domain $\Omega^{fs} = (\overline{\Omega_2^f} \cup \overline{\Omega^s})^\circ$. All meshes are assumed to be admissible and to consist of shape-regular simplices.

We further note that the background tessellation $\mathcal{T}_0$ may be decomposed into three disjoint subsets:

$$\mathcal{T}_0 = \mathcal{T}_{0,1} \cup \mathcal{T}_{0,2} \cup \mathcal{T}_{0,\Gamma}. \tag{3-1}$$

Here $\mathcal{T}_{0,1}$, $\mathcal{T}_{0,2}$ and $\mathcal{T}_{0,\Gamma}$ are defined with reference to Ω^{fs} and denote the sets of elements in $\mathcal{T}_0$ that are *not*, *completely* or *partially* overlapped by Ω^{fs}. More precisely, $\mathcal{T}_{0,1} = \{T \in \mathcal{T}_0 : T \subset \overline{\Omega_1^f}\}$, $\mathcal{T}_{0,2} = \{T \in \mathcal{T}_0 : T \subset \overline{\Omega^{fs}}\}$ and $\mathcal{T}_{0,\Gamma} = \{T \in \mathcal{T}_0 : |T \cap \Omega_1^f| > 0 \text{ and } |T \cap \Omega^{fs}| > 0\}$. In addition, we assume that $\mathcal{T}_0$ is sufficiently fine near the fluid-fluid interface in the sense that $T \cap \Omega^s = \varnothing$ for all $T \in \mathcal{T}_{0,\Gamma}$. In other words, the elements in the fluid background mesh have to be small enough close to Γ_{ff} such that a single element does not stretch from the fluid-fluid interface to the fluid-structure interface. Next, we introduce the *reduced* background mesh $\mathcal{T}_1^*$, consisting of the elements in $\mathcal{T}_0$ that are either not or only partially overlapped by Ω^{fs}, and associated domain Ω_1^*:

$$\mathcal{T}_1^* = \mathcal{T}_{0,1} \cup \mathcal{T}_{0,\Gamma}, \qquad \Omega_1^* = \bigcup_{T \in \mathcal{T}_1^*} T. \tag{3-2}$$

Note that Ω_1^* contains (but is generally larger than) Ω_1^f. We further define the so-called *fluid overlap region* $\Omega_O = \Omega_2^f \cap \Omega_1^*$. In general, for each overlapping mesh configuration described by some (background) mesh and some overlapping domain, the procedure described above defines what we shall refer to as the reduced (background) mesh.

3.1. ***An overlapping mesh method for the fluid problem.*** Here we present a finite element discretization of (2-1)–(2-6) posed on a pair of overlapping meshes, first proposed by Massing et al. [41]. The pair of meshes consist of an overlapped mesh and an overlapping mesh: in our case, the reduced background mesh $\mathcal{T}_1^*$ plays the role of the overlapped mesh while $\mathcal{T}_2^f$ is the overlapping mesh.

For any given mesh $\mathcal{T}$, let $V_h(\mathcal{T})$ be the space of continuous piecewise linear vector fields and let $Q_h(\mathcal{T})$ be the space of continuous piecewise linear scalar fields, both defined relative to $\mathcal{T}$. We define the composite finite element spaces V_h and Q_h for the overlapping fluid meshes by

$$V_h^f = V_h(\mathcal{T}_1^*) \oplus V_h(\mathcal{T}_2^f), \qquad Q_h^f = Q_h(\mathcal{T}_1^*) \oplus Q_h(\mathcal{T}_2^f). \tag{3-3}$$

Moreover, we denote by $V_{h,\boldsymbol{g}^f}^f$ the subspace of V_h^f that satisfies the boundary conditions (2-3)–(2-4) and by $V_{h,\mathbf{0}}^f$ the corresponding homogeneous version. The

overlapping mesh discretization of (2-1)–(2-6) is then: find $(\boldsymbol{u}_h^f, p_h^f) \in V_{h,\boldsymbol{g}^f}^f \times Q_h^f$ such that

$$A_h^f(\boldsymbol{u}_h^f, p_h^f; \boldsymbol{v}, q) = L_h^f(\boldsymbol{v}, q) \quad \text{for all } (\boldsymbol{v}, q) \in V_{h,\boldsymbol{0}}^f \times Q_h^f, \tag{3-4}$$

where A_h^f is defined for all $\boldsymbol{u}, \boldsymbol{v} \in V_h^f$ and all $p, q \in Q_h^f$ by

$$A_h^f(\boldsymbol{u}, p; \boldsymbol{v}, q) = a_h^f(\boldsymbol{u}, \boldsymbol{v}) + b_h^f(\boldsymbol{v}, p) + b_h^f(\boldsymbol{u}, q) + i_h^f(\boldsymbol{u}, \boldsymbol{v}) - j_h^f(p, q) \tag{3-5}$$

and the forms a_h^f, b_h^f, i_h^f and j_h^f are given by

$$a_h^f(\boldsymbol{u}, \boldsymbol{v}) = (\nabla \boldsymbol{u}, \nabla \boldsymbol{v})_{\Omega_1^f \cup \Omega_2^f} - (\langle \partial_{\boldsymbol{n}} \boldsymbol{u} \rangle, [\boldsymbol{v}])_{\Gamma^{ff}} - (\langle \partial_{\boldsymbol{n}} \boldsymbol{v} \rangle, [\boldsymbol{u}])_{\Gamma^{ff}} + \gamma (h^{-1}[\boldsymbol{u}], [\boldsymbol{v}])_{\Gamma^{ff}}, \tag{3-6}$$

$$b_h^f(\boldsymbol{v}, q) = -(\nabla \cdot \boldsymbol{v}, q)_{\Omega_1^f \cup \Omega_2^f} + ([\boldsymbol{v}] \cdot \boldsymbol{n}, \langle q \rangle)_{\Gamma^{ff}}, \tag{3-7}$$

$$i_h^f(\boldsymbol{u}, \boldsymbol{v}) = (\nabla(\boldsymbol{u}_1 - \boldsymbol{u}_2), \nabla(\boldsymbol{v}_1 - \boldsymbol{v}_2))_{\Omega_O}, \tag{3-8}$$

$$j_h^f(p, q) = \delta \sum_{T \in \mathcal{T}_1^* \cup \mathcal{T}_2^f} h_T^2 (\nabla p, \nabla q)_T \tag{3-9}$$

for $\delta > 0$. Here and throughout, $(\cdot, \cdot)_K$ denotes the $L^2(K)$ inner product over some domain K while $\langle v \rangle$ denotes a convex combination $\langle v \rangle = \alpha_1 v_1 + \alpha v_2$ with $\alpha_1 + \alpha_2 = 1$ of v across the interface Γ^{ff}. In particular, we choose $\langle v \rangle = v_2$ in accordance with Hansbo et al. [23]. Finally, the linear form L_h^f is defined by

$$L_h^f(\boldsymbol{v}, q) = (\boldsymbol{f}^f, \boldsymbol{v}) - \delta \sum_{T \in \mathcal{T}_1^* \cup \mathcal{T}_2^f} h_T^2 (\boldsymbol{f}^f, \nabla q)_T \tag{3-10}$$

for all $\boldsymbol{v} \in V_h^f$ and all $q \in Q_h^f$.

A major strength of the employed scheme for the fluid problem is that the extension of the stabilization term (3-9) from the physical domain Ω_1^f to the overlap region Ω_O in combination with the least-square stabilization (3-8) results in a well-conditioned and optimally convergent scheme, independent of the location of the overlapping mesh with respect to the fixed background mesh. Thereby, typical difficulties arising from potentially small cut cells where $|T \cap \Omega_2^f| \ll |T|$ for $T \in \mathcal{T}_{0,\Gamma}$ are completely eliminated. Consequently, for a continuous solution $(\boldsymbol{u}^f, p^f)$ satisfying (2-1)–(2-6) and a discrete solution $(\boldsymbol{u}_h^f, p_h^f)$ satisfying (3-4), the following optimal error estimate holds independently of the fluid-fluid interface position [41]:

$$|||(\boldsymbol{u}^f - \boldsymbol{u}_h^f, p^f - p_h^f)||| \leqslant Ch|\boldsymbol{u}^f|_{2,\Omega^f} + |p^f|_{1,\Omega^f}. \tag{3-11}$$

Here, $|||\cdot|||$ is an appropriate version of the standard norm on $H^1(\Omega^f) \times L^2(\Omega^f)$ accounting for the fluid overlap region Ω_O; see [41] for more details.

3.2. ***A finite element discretization of the structure problem.*** The structure problem is described by (2-7)–(2-9) in the current solid domain. As the current solid domain is actually unknown, a standard approach to discretizing such problems is to map the governing equations back to a fixed reference (Lagrangian) frame. We choose a reference domain $\widehat{\Omega}^s$ with coordinates $\hat{\boldsymbol{x}}$ and denote the deformation map from the reference to the current solid domain by $\boldsymbol{\phi}^s$:

$$\boldsymbol{x} = \boldsymbol{\phi}^s(\hat{\boldsymbol{x}}) \quad \text{for } \hat{\boldsymbol{x}} \in \widehat{\Omega}^s. \tag{3-12}$$

In general, the notation for all domains and quantities pulled back to the Lagrangian framework will be endowed with a $\hat{}$; for instance, $\widehat{\Omega}^s$ and $\hat{\boldsymbol{u}}^s$ denote the solid reference domain and solid displacement in the reference frame, respectively. In particular, $\boldsymbol{\phi}^s = \boldsymbol{I} + \hat{\boldsymbol{u}}^s$.

In the Lagrangian frame, the problem reads: find the solid displacement $\hat{\boldsymbol{u}}^s : \widehat{\Omega}^s \to \mathbb{R}^3$ such that

$$-\nabla \cdot \widehat{\boldsymbol{\Pi}}(\hat{\boldsymbol{u}}^s) = \hat{\boldsymbol{f}}^s \quad \text{in } \widehat{\Omega}^s, \tag{3-13}$$

$$\hat{\boldsymbol{u}}^s = \hat{\boldsymbol{g}}^s_D \quad \text{on } \partial\widehat{\Omega}^s_D, \tag{3-14}$$

$$\widehat{\boldsymbol{\Pi}}(\hat{\boldsymbol{u}}^s) \cdot \hat{\boldsymbol{n}} = \hat{\boldsymbol{t}}^s_N \quad \text{on } \widehat{\Gamma}^{fs}. \tag{3-15}$$

Here, the displacement $\hat{\boldsymbol{u}}^s$ and the boundary displacement $\hat{\boldsymbol{g}}^s_D$ result from the standard affine pull-back of the corresponding quantities in the current domain, for instance $\hat{\boldsymbol{u}}^s(\hat{\boldsymbol{x}}) = \boldsymbol{u}^s(\boldsymbol{x})$, and $\hat{\boldsymbol{n}}$ is the outward normal of the fluid-structure interface in the reference frame. Further, let $\boldsymbol{F}^s = \nabla\boldsymbol{\phi}^s$ and $J^s = \det \boldsymbol{F}^s$. We let $\hat{\boldsymbol{f}}^s(\hat{\boldsymbol{x}}) = J^s \boldsymbol{f}^s(\boldsymbol{x})$. Moreover, $\widehat{\boldsymbol{\Pi}}(\hat{\boldsymbol{u}}^s)$ denotes the first Piola–Kirchhoff stress tensor, resulting from a Piola transformation of the Cauchy stress tensor $\boldsymbol{\sigma}^s$:

$$\widehat{\boldsymbol{\Pi}}(\hat{\boldsymbol{u}}^s)(\hat{\boldsymbol{x}}) = J^s(\hat{\boldsymbol{x}})\boldsymbol{\sigma}^s(\boldsymbol{\phi}^s(\hat{\boldsymbol{x}}))(\boldsymbol{F}^s)^{-\mathrm{T}}(\hat{\boldsymbol{x}}). \tag{3-16}$$

In view of (2-10), we will enforce that the boundary traction acting on the solid in the reference domain is the Piola transform of the fluid traction exerted on the fluid-structure interface by the fluid in the current or physical configuration. This will be detailed in Section 4.

The governing equations (3-13)–(3-15) must be completed by a constitutive equation relating the stress to the strain. In the case of a hyperelastic material, by definition, there exists a strain energy density Ψ such that

$$\widehat{\boldsymbol{\Pi}}(\boldsymbol{F}) = \frac{\partial \Psi}{\partial \boldsymbol{F}}. \tag{3-17}$$

One example is the Saint-Venant–Kirchhoff material model, in which

$$\Psi(\boldsymbol{F}) = \mu \operatorname{tr} \boldsymbol{E}^2 + \tfrac{1}{2}\lambda(\operatorname{tr} \boldsymbol{E})^2, \quad \text{where } \boldsymbol{E} = \tfrac{1}{2}(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F} - \boldsymbol{I}), \tag{3-18}$$

for Lamé constants $\mu, \lambda > 0$.

In the special case of a linearly elastic material, we assume that the reference and physical configurations coincide so that (2-7)–(2-9) hold over $\widehat{\Omega}^s$ directly with $\boldsymbol{\sigma}^s(\boldsymbol{u}^s) = 2\mu\epsilon(\boldsymbol{u}^s) + \lambda\,\mathrm{tr}(\epsilon(\boldsymbol{u}^s))\boldsymbol{I}$.

To solve (3-13)–(3-15) numerically, let $\widehat{\mathcal{T}}^s$ be a tessellation of $\widehat{\Omega}^s$ such that $\mathcal{T}^s = \boldsymbol{\phi}^s(\widehat{\mathcal{T}}^s)$ and introduce the finite element approximation space

$$\widehat{V}^s_{h,\boldsymbol{g}} = \{\boldsymbol{v} \in V_h(\widehat{\mathcal{T}}^s) : \boldsymbol{v}|_{\partial\widehat{\Omega}^s_D} = \boldsymbol{g}\}, \tag{3-19}$$

where $V_h(\widehat{\mathcal{T}}^s)$ is the space of continuous piecewise linear vector fields defined relative to $\widehat{\mathcal{T}}^s$ as before. The finite element formulation of (3-13)–(3-15) then reads: find $\hat{\boldsymbol{u}}^s_h \in \widehat{V}^s_{h,\hat{\boldsymbol{g}}^s_D}$ such that

$$(\widehat{\Pi}(\hat{\boldsymbol{u}}^s_h), \nabla\boldsymbol{v})_{\widehat{\Omega}^s} - (\hat{\boldsymbol{t}}^s_N, \boldsymbol{v})_{\widehat{\Gamma}^{fs}} - (\hat{\boldsymbol{f}}^s, \boldsymbol{v})_{\widehat{\Omega}^s} = 0 \quad \text{for all } \boldsymbol{v} \in \widehat{V}^s_{h,\boldsymbol{0}}. \tag{3-20}$$

Note that the generally nonlinear constitutive relation and the geometric nonlinearity mandate a nonlinear solution scheme, such as a Newton method or an inner fixed-point iteration for (3-20).

3.3. ***Deformation of the surrounding fluid domain.*** The overlapping mesh method relies on keeping the background part of the fluid domain Ω^f_1 fixed while moving the part of the fluid domain Ω^f_2 surrounding the structure. This movement ensures that the mesh $\mathcal{T}^f_2$ of the latter part of the fluid domain and the structure mesh $\mathcal{T}^s$ match at the fluid-structure interface. The movement is dictated by the structure deformation only at the fluid-structure interface: the motion of the interior of the fluid domain Ω^f_2 is subject to numerical modeling. Standard approaches for the domain motion include mesh smoothing via diffusion-type equations or treating the fluid domain as a pseudoelastic structure. Here, we choose the latter approach and model the deformation of the fluid domain as a linearly elastic structure. This approach allows for typically larger deformations than a simple diffusion-equation-based mesh smoothing while avoiding unnecessary complexity.

We start with a fixed reference domain $\widehat{\Omega}^f_2$ and consider the following mesh deformation problem over this domain: find the mesh displacement $\hat{\boldsymbol{u}}^m : \widehat{\Omega}^f_2 \to \mathbb{R}^3$ such that

$$-\nabla\cdot\hat{\boldsymbol{\sigma}}^m(\hat{\boldsymbol{u}}^m) = 0 \quad \text{in } \widehat{\Omega}^f_2, \tag{3-21}$$

$$\hat{\boldsymbol{\sigma}}^m(\hat{\boldsymbol{u}}^m)\cdot\hat{\boldsymbol{n}} = 0 \quad \text{on } \widehat{\Gamma}^{ff}, \tag{3-22}$$

$$\hat{\boldsymbol{u}}^m = \hat{\boldsymbol{u}}^s \quad \text{on } \widehat{\Gamma}^{fs}, \tag{3-23}$$

where the stress tensor $\hat{\boldsymbol{\sigma}}^m$ is given by

$$\hat{\boldsymbol{\sigma}}^m(\hat{\boldsymbol{u}}^m) = 2\mu_m\epsilon(\hat{\boldsymbol{u}}^m) + \lambda_m\,\mathrm{tr}(\epsilon(\hat{\boldsymbol{u}}^m))\boldsymbol{I} \tag{3-24}$$

for chosen Lamé constants $\mu_m, \lambda_m > 0$. Let now $\widehat{\mathcal{T}}^f_2$ be a tessellation of $\widehat{\Omega}^f_2$.

We define the finite element space $\widehat{V}^m_{h,\boldsymbol{g}}$ by

$$\widehat{V}^m_{h,\boldsymbol{g}} = \{\boldsymbol{v} \in V_h(\widehat{\mathcal{T}}_2^f) : \boldsymbol{v}|_{\widehat{\Gamma}^{fs}} = \boldsymbol{g}\}. \tag{3-25}$$

The corresponding finite element formulation of the mesh problem (3-21)–(3-23) is then: find $\hat{\boldsymbol{u}}^m_h \in \widehat{V}^m_{h,\hat{\boldsymbol{u}}^s}$ such that

$$(\hat{\boldsymbol{\sigma}}^m(\hat{\boldsymbol{u}}^m_h), \boldsymbol{v})_{\widehat{\Omega}_2^f} = 0 \quad \text{for all } \boldsymbol{v} \in \widehat{V}^m_{h,\boldsymbol{0}}. \tag{3-26}$$

Finally, we define $\mathcal{T}_2^f = \boldsymbol{\phi}^m_h(\widehat{\mathcal{T}}_2^f)$ with the discrete mesh deformation $\boldsymbol{\phi}^m_h = \boldsymbol{I} + \hat{\boldsymbol{u}}^m_h$. The current surrounding fluid domain is then defined accordingly: $\Omega_2^f = \boldsymbol{\phi}^m_h(\widehat{\Omega}_2^f)$. The use of boundary condition (3-22) ensures that the fluid-structure interface is preserved in the sense that

$$\Gamma^{fs} = \partial\Omega_2^f \cap \partial\Omega^s = \boldsymbol{\phi}^m_h(\widehat{\Gamma}^{fs}) = \boldsymbol{\phi}^s_h(\widehat{\Gamma}^{fs}), \tag{3-27}$$

where $\boldsymbol{\phi}^s_h$ is the solid deformation given by the discrete solution $\hat{\boldsymbol{u}}^s_h$ of problem (3-20).

4. Solution algorithm for the discretized FSI problem

We are now in a position to give a detailed description of the overall solution scheme for the fully coupled fluid-structure interaction problem. We start with reviewing the formulation of the fluid-structure coupling in the discrete setting. For the discrete formulation, a third interface condition (3-23) needs to be added to the two interface conditions (2-3) and (2-9) due to the additional mesh deformation problem described in Section 3.3. The mesh deformation allows us to express the fluid stress tensor acting on Γ^{fs} in the reference configuration $\widehat{\Gamma}^{fs}$ via a Piola transformation. Consequently, the stress equilibrium condition (2-9) at the fluid-structure interface can be reformulated in the Lagrangian frame according to (3-15). In summary, the discrete formulation of the fluid-structure interface conditions reads:

$$\boldsymbol{u}^f = 0 \qquad \text{on } \Gamma^{fs}, \tag{4-1}$$

$$\hat{\boldsymbol{u}}^s = \hat{\boldsymbol{u}}^m \qquad \text{on } \widehat{\Gamma}^{fs}, \tag{4-2}$$

$$\widehat{\Pi}(\hat{\boldsymbol{u}}^s)(\hat{\boldsymbol{x}}) \cdot \hat{\boldsymbol{n}}(\hat{\boldsymbol{x}}) = J^m(\hat{\boldsymbol{x}})\, \boldsymbol{\sigma}^f(\boldsymbol{\phi}^m(\hat{\boldsymbol{x}}))(\boldsymbol{F}^m)^{-\mathrm{T}}(\hat{\boldsymbol{x}}) \cdot \hat{\boldsymbol{n}}(\hat{\boldsymbol{x}}) \quad \text{on } \widehat{\Gamma}^{fs}. \tag{4-3}$$

As outlined in Section 3, we employ a classical Dirichlet–Neumann fixed-point iteration approach to ensure that the interface conditions (4-1)–(4-3) are approximately satisfied by the computed solution within a user-provided tolerance. The iteration scheme is presented in detail in Algorithm 1, where the relaxation parameter ω^i was chosen dynamically to accelerate the convergence of the fixed-point iteration. Moreover, the fluid boundary traction is incorporated as Neumann data in the weak formulation of the structure problem by a properly chosen functional representing the

$\hat{\boldsymbol{u}}^{s,k} := 0$
$\hat{\boldsymbol{u}}^{m,k} := 0$
do

Update overlapping fluid meshes
$\Omega^{s,k+1} := (\boldsymbol{I} + \hat{\boldsymbol{u}}^{s,k})(\widehat{\Omega}^s)$
$\Omega_2^{f,k+1} := (\boldsymbol{I} + \hat{\boldsymbol{u}}^{m,k})(\widehat{\Omega}_2^f)$
$\Omega^{fs,k+1} := \Omega^{s,k+1} \cup \Omega_2^{f,k+1}$
Compute reduced background mesh $(\mathcal{T}_1^{f,k+1})^*$ with respect to $\Omega^{fs,k+1}$.
$\mathcal{T}^{f,k+1} := (\mathcal{T}_1^{f,k+1})^* \cup \mathcal{T}_2^{f,k+1}$

Solve fluid problem
Find $(\boldsymbol{u}_h^{f,k+1}, p_h^{f,k+1})$ such that for all $(\boldsymbol{v}_h^{f,k+1}, q_h^{f,k+1}) \in V_h^{f,k+1} \times Q_h^{f,k+1}$

$$A_h^{f,k}(\boldsymbol{u}_h^{f,k+1}, p_h^{f,k+1}; \boldsymbol{v}_h^{f,k+1}, q_h^{f,k+1}) = L^{f,k+1}(\boldsymbol{v}_h^{f,k+1}, q_h^{f,k+1}).$$

Update boundary traction functional
Define $L^{fs,k+1}(\,\cdot\,)$ by

$$L^{fs,k+1}(\hat{\boldsymbol{v}}_h^{s,k+1}) := R^{f,k+1}(\boldsymbol{u}_h^{f,k+1}, p_h^{f,k+1}; \boldsymbol{v}_h^{f,k+1}).$$

Solve structure problem
Find $\hat{\boldsymbol{u}}_h^{s,k+1}$ such that for all $\hat{\boldsymbol{v}} \in \widehat{V}_h^s$

$$A_h^s(\hat{\boldsymbol{u}}_h^{s,k+1}, \hat{\boldsymbol{v}}) = L^s(\hat{\boldsymbol{v}}) + L^{fs,k+1}(\hat{\boldsymbol{v}}).$$

Dynamic relaxation
Compute ω^{k+1} according to (4-6).
$\hat{\boldsymbol{u}}_h^{s,k+1} := \omega^{k+1}\hat{\boldsymbol{u}}_h^{s,k+1} + (1-\omega^{k+1})\hat{\boldsymbol{u}}_h^{s,k}$

Solve mesh problem
Find $\hat{\boldsymbol{u}}_h^{m,k+1}$ such that for all $\hat{\boldsymbol{v}} \in \widehat{V}_h^m$

$$\begin{aligned} A_h^m(\hat{\boldsymbol{u}}_h^{m,k+1}, \hat{\boldsymbol{v}}) &= L^s(\hat{\boldsymbol{v}}), \\ \hat{\boldsymbol{u}}_h^{m,k+1} &= \hat{\boldsymbol{u}}_h^{s,k+1} \quad \text{on } \widehat{\Gamma^{fs}}. \end{aligned}$$

while $\|\hat{\boldsymbol{u}}_h^{s,k+1} - \hat{\boldsymbol{u}}_h^{s,k}\| \leqslant \text{tol}$

Algorithm 1. Fixed-point iteration.

boundary traction weighted with some given test function. A thorough explanation of both of these intermediate steps will be given in the next sections.

4.1. ***Dynamic relaxation.*** Let U_s^k denote the coefficient vector of the finite element approximation $\hat{\boldsymbol{u}}_h^{s,k}$ of (3-20) computed in the k-th iteration step. To accelerate the convergence of the iteration scheme, a relaxation step is introduced:

$$U_s^{k+1} := \omega_k U_s^{k+1} + (1-\omega_k)U_s^k, \tag{4-4}$$

where the relaxation parameter ω_k is dynamically chosen in each iteration step.

Here, we employed Aiken's method, which is a simple scheme, yet it can greatly improve the convergence rate compared to a fixed choice of ω_k, as demonstrated by Küttler and Wall [30; 31]. Introducing the residual displacement $\Delta^k U_s$ by

$$\Delta^k U_s := U_s^k - U_s^{k-1}, \tag{4-5}$$

the new relaxation parameter ω_{k+1} is then computed by

$$\omega_k = \max\left\{\omega_{\max}, \omega_{k-1}\left(1 - \frac{\Delta^{k+1} U_s}{\|\Delta^{k+1} U_s - \Delta^k U_s\|^2}\right)\right\}, \tag{4-6}$$

where $\omega_{\max}$ is a safety parameter chosen to avoid too-large over-relaxation. The convergence of the fixed-point iteration might be accelerated further by employing more sophisticated schemes based on Robin–Robin coupling [3; 4] or vector extrapolation [31].

4.2. ***Computation of the boundary traction.*** Given the solution $\boldsymbol{u}^f$ and a pressure solution p^f of the fluid subproblem (2-1)–(2-4), the incorporation of the fluid boundary traction into the weak formulation of the structure problem (3-20) requires the evaluation of the so-called weighted fluid boundary traction on Γ^{fs} defined by

$$L^{fs}(\boldsymbol{v}) = (\boldsymbol{\sigma}^f(\boldsymbol{u}^f, p^f) \cdot \boldsymbol{n}, \boldsymbol{v})_{\Gamma^{fs}} \tag{4-7}$$

for test functions $\boldsymbol{v} \in V^s$. The functional (4-7) possesses various equivalent representations in the continuous case that are no longer equivalent when fluid velocity $\boldsymbol{u}^f$ and pressure p^f and test function $\boldsymbol{v}$ are replaced by their discrete counterparts $\boldsymbol{u}_h^f$, p_h^f and $\boldsymbol{v}_h \in V_h^s(\Omega)$, respectively. It has been observed by Dorok [16], John [28] and Giles et al. [21] that using (4-7) directly might lead to an inaccurate evaluation of the weighted boundary traction. In our work, we therefore employ an alternative formulation of the weighted boundary traction in the form

$$L^{fs}(\boldsymbol{v}_h) = (\boldsymbol{\sigma}^f(\boldsymbol{u}_h^f, p_h^f), \operatorname{Ext} \boldsymbol{v}_h)_{\Omega^f} - (\boldsymbol{f}^f, \operatorname{Ext} \boldsymbol{v}_h)_{\Omega^f}, \tag{4-8}$$

which was proposed and investigated by Giles et al. [21] in the context of a posteriori error estimation. Here, $\operatorname{Ext} \boldsymbol{v}$ is any function in $H^1(\Omega^{fs})$ such that $\operatorname{Ext}(\boldsymbol{v}_h)|_{\Gamma^{fs}} = \boldsymbol{v}_h$. Compared to the naive evaluation using (4-7), the formulation (4-8) was shown to compute the weighted boundary traction more accurately and to greatly improve the convergence of stress-related quantities such as the lift and drag coefficients.

5. Numerical results

We conclude this paper with two numerical tests, both in three spatial dimensions. The numerical experiments were carried out using the DOLFIN-OLM library. We first study the convergence rates for the finite element approximations of the fluid velocity, fluid pressure and structure displacement by constructing an artificial

fluid-structure interaction problem possessing an analytical solution. Second, we consider the flow around an elastic flap immersed in a three-dimensional channel.

5.1. ***Software for overlapping mesh variational formulations.*** The assembly of finite element tensors corresponding to standard variational formulations on conforming, simplicial meshes, such as (3-20), involves integration over elements and possibly interior and exterior facets. In contrast, the assembly of variational forms defined over overlapping meshes, such as (3-6)–(3-9) and (3-10), additionally requires integration over cut elements and cut facets. These mesh entities are of polyhedral, but otherwise arbitrary, shape. As a result, the assembly process is highly nontrivial in practice and requires additional geometry-related preprocessing, which is challenging in particular for three-dimensional meshes.

As part of this work, the technology required for the automated assembly of general variational forms defined over overlapping meshes has been implemented as part of the software library DOLFIN-OLM. This library builds on the core components of the FEniCS Project [34; 33], in particular DOLFIN [35], and the computational geometry libraries CGAL [51] and GTS [22]. DOLFIN-OLM is open source and freely available from http://launchpad.net/dolfin-olm.

There are two main challenges involved in the implementation: the computational geometry and the integration of finite element variational forms on cut cells and facets. The former involves establishing a sufficient topological and geometric description of the overlapping meshes for the subsequent assembly process. To this end, DOLFIN-OLM provides functionality for finding and computing the intersections of triangulated surfaces with arbitrary simplicial background meshes in three spatial dimensions; this functionality relies on the computational geometry libraries CGAL and GTS. These features generate topological and geometric descriptions of the cut elements and facets. Based on this information, quadrature rules for the integration of fields defined over these geometrical entities are produced. The computational geometrical aspect of this work extends, but shares many of the features of, the previous work [39] and is described in more detail in the aforementioned reference.

Further, by extending some of the core components of the FEniCS Project, in particular FFC [29; 34, Chapter 11] and UFC [34, Chapter 16], this work also provides a finite element form compiler for variational forms defined over overlapping meshes. Given a high-level description of the variational formulation, low-level C++ code can be automatically generated for the evaluation of the cut element, cut facet and surface integrals, in addition to the evaluation of integrals over the standard (uncut) mesh entities. The generated code takes as input appropriate quadrature points and weights for each cut element or facet; these are precisely those provided by the DOLFIN-OLM library.

As a result, one may specify variational forms defined over finite element spaces on overlapping meshes in high-level UFL notation [2; 34, Chapter 17], define the

overlapping fluid meshes $\{\mathcal{T}_0, \mathcal{T}_2^f\}$ and then invoke the functionality provided by the DOLFIN-OLM library to automatically assemble the corresponding stiffness matrix. In particular, the numerical experiments presented below, employing the variational formulation defined by (3-4), have been carried out using this technology.

5.2. ***Convergence test.*** While numerical studies presented in [41] confirmed the theoretically predicted convergence rates for the overlapping mesh method for the pure flow problem presented in Section 3.1, we here conduct a convergence study of the coupled FSI problem to verify the overall solution algorithm as described in Algorithm 1. To examine the convergence rates for the finite element approximations of the fluid velocity, fluid pressure and structure displacement, we construct a stationary FSI problem with a known analytical solution by employing the method of manufactured solutions as outlined in the following. The detailed analytical derivation of the fluid- and structure-related quantities are not included here to keep the presentation at an appropriate length but can be obtained as an IPython-based notebook available at http://nbviewer.ipython.org/6291921.

In the reference configuration, the fluid domain $\widehat{\Omega}^f$ consists of a straight tube of length $L = 1.0$ and diameter $R^f = 0.4$. We decompose $\widehat{\Omega}^f$ into a tube of radius $R_1^f = 0.3$ and a cylinder annulus satisfying $0.3 \leqslant r \leqslant 0.4 = R_2^f$. The solid domain $\widehat{\Omega}^s$ is given by a cylinder annulus of thickness $H^s = 0.1$ surrounding the fluid domain $\widehat{\Omega}^f$. Using cylinder coordinates, the displacement $\hat{\boldsymbol{u}}^s$ of the solid domain is prescribed by a purely radial, z-dependent translation

$$\hat{\boldsymbol{u}}^s(r, \varphi, z) = H(z)\boldsymbol{e}_r, \tag{5-1}$$

where $H(z) = H^s 2z(1 - z)$. Correspondingly, the deformation of the fluid domain is determined by a radial stretching of the form

$$\hat{\boldsymbol{u}}^m(r, \varphi, z) = \rho(1 + H(z)/R^f)\boldsymbol{e}_r. \tag{5-2}$$

The reference and physical configuration of the various domains are depicted in Figure 3.

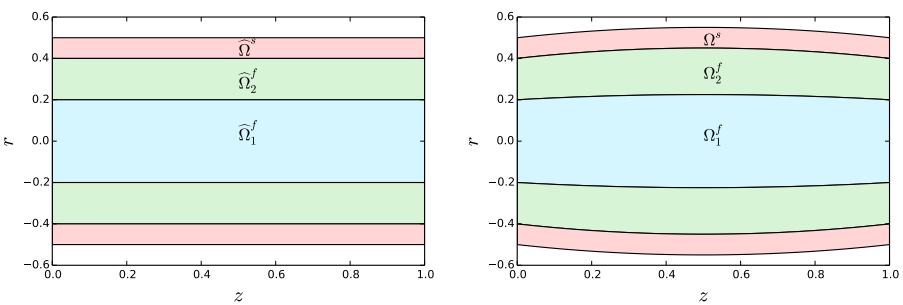


Figure 3. Cross-section through the cylinder-symmetric reference (left) and physical (right) domains for the analytical FSI reference problem.

To obtain a divergence-free velocity field in the final physical configuration, the fluid velocity is defined as a simple parabolic channel flow on the reference domain and then mapped to the physical domain via the Piola transformation induced by the fluid domain deformation (5-2). For the pressure, we simply choose $p(x, y, z) = 1 - z$. Since the interface condition (4-3) is not satisfied exactly, we introduce an auxiliary traction $\boldsymbol{t}_a$ given by the nonvanishing jump in the normal stresses:

$$\boldsymbol{t}_a = \big(\widehat{\Pi}(\hat{\boldsymbol{u}}^s)(\hat{\boldsymbol{x}}) - J^m(\hat{\boldsymbol{x}})\boldsymbol{\sigma}^f(\boldsymbol{\phi}^m(\hat{\boldsymbol{x}}))(\boldsymbol{F}^m)^{-\mathrm{T}}(\hat{\boldsymbol{x}})\big) \cdot \hat{\boldsymbol{n}}^s \quad \text{on } \widehat{\Gamma}^{fs}. \tag{5-3}$$

Regarding the remaining boundary parts, the solid displacement is uniquely determined by imposing the given displacement $\hat{\boldsymbol{u}}^s$ as a Dirichlet boundary condition on $\partial\widehat{\Omega}^s \setminus \widehat{\Gamma}^{fs}$. For the fluid problem, we prescribed the velocity profile on the inlet and impose the zero pressure on the outlet.

In the reference configuration, a discretization of the solid domain $\widehat{\Omega}^s$ and the fluid domain $\widehat{\Omega}_2^f$ is provided by two fitted and conforming meshes $\widehat{\mathcal{T}}^s$ and $\widehat{\mathcal{T}}_2^f$, respectively, while the fluid domain $\widehat{\Omega}_1^f$ is represented by a structured Cartesian mesh $\widehat{\mathcal{T}}_1^f$ overlapped by the mesh $\widehat{\mathcal{T}}_2^f$; see Figures 4 and 5. The numerical approximation

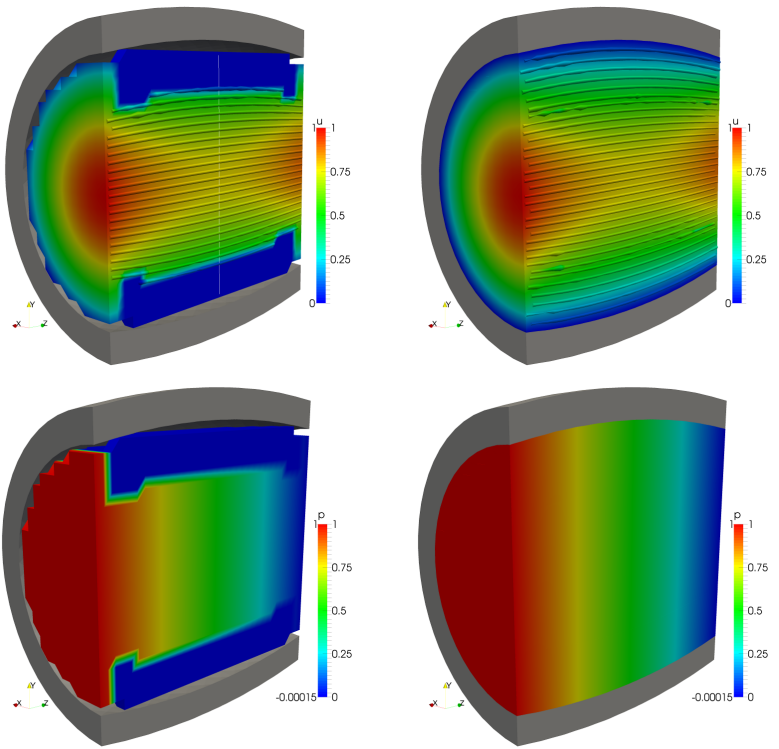


Figure 4. Computed velocity (top) and pressure (bottom) solutions on the fixed fluid background mesh $\mathcal{T}_1^f$ (left) and entire overlapping fluid mesh $\{\mathcal{T}_1^f, \mathcal{T}_2^f\}$ (right) for the analytical FSI problem.

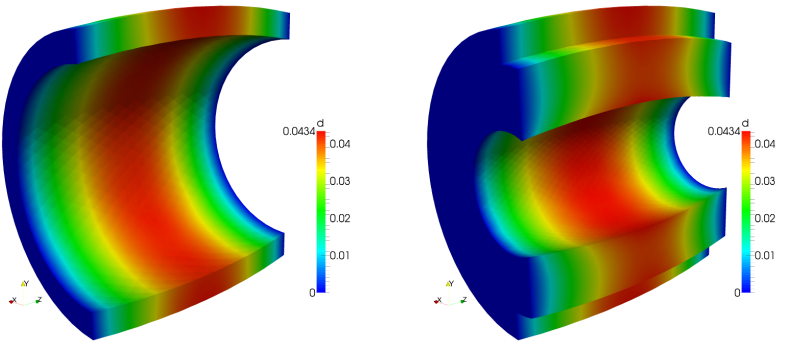


Figure 5. Displacements for the analytical FSI reference problem. Left: structure displacement of the solid tube. Right: displacement of the fluid mesh added.

of the fluid velocity, fluid pressure and structure displacement are then computed on a sequence of four overlapping meshes. The mesh sizes of the initial meshes $\widehat{\mathcal{T}}_1^f$, $\widehat{\mathcal{T}}_2^f$ and $\widehat{\mathcal{T}}^s$ are 0.246, 0.14 and 0.212, respectively, and each of the subsequent meshes is generated from the previous one by uniformly refining each mesh. Based on the manufactured exact solution, the experimental order of convergence (EOC) is then computed by

$$\mathrm{EOC}(k) = \frac{\log(E_{k-1}/E_k)}{\log 2},$$

where E_k denotes the error of the numerical solution computed at refinement level k. The numerical experiment was conducted using $\nu^f = 0.001$ for the fluid viscosity and Lamé parameters given by

$$\mu = E/(2+2\nu), \qquad \lambda = E \cdot \nu/((1+\nu)(1-2\nu)) \tag{5-4}$$

in Ω^s with $E = 10$ and $\nu = 0.3$.

For the penalty parameters in the stabilized overlapping mesh method for the fluid problem, we pick $\gamma = 10$ and $\delta = 0.5$. Since the overall computational time is dominated by the assembly and solution of the fluid system, the displacement field is conveniently solved using a direct solver while the linear system arising from the fluid problem is solved by applying a transpose-free quasiminimal residual solver with an algebraic multigrid preconditioner.

Using continuous piecewise linear functions for the approximation of the fluid velocity, fluid pressure and the structure displacement, the theoretically predicted convergence rate for a corresponding uncoupled problem is at least 1.0 when measuring the velocity and displacement error in the H^1-norm and the pressure error in the L^2-norm. Note that it is common to observe a higher experimental order of convergence of ~ 1.5 for the pressure approximation when stabilized, equal-order interpolation elements are used to discretize the flow problem. Assuming

Refinement	$\|u_h^f - u^f\|_1$	EOC	$\|p_h^f - p^f\|_0$	EOC	$\|u_h^s - u^s\|_1$	EOC
0	1.01188		$3.61948 \cdot 10^{-3}$		$3.87181 \cdot 10^{-4}$	
1	0.51000	0.99	$1.55216 \cdot 10^{-3}$	1.22	$1.40771 \cdot 10^{-4}$	1.46
2	0.21912	1.22	$3.70746 \cdot 10^{-4}$	2.06	$4.39062 \cdot 10^{-5}$	1.68
3	0.12485	0.81	$1.29430 \cdot 10^{-4}$	1.52	$1.17800 \cdot 10^{-5}$	1.9

Table 2. Convergence rates of the overlapping mesh finite element method for the analytical FSI problem.

at most quadratic convergence of the displacement solution in the L^2-norm, the L^2-error will be reduced by approximately $0.5^{2 \cdot 3} \approx 0.016$ after three uniform mesh refinements. To not pollute the overall convergence rate with the iteration error, we therefore chose tol $= 0.001$ for the relative L^2-error between two consecutive displacement solutions computed in the iteration loop. With the given tolerance, the Dirichlet–Neumann iteration converged after 5–7 iterations for each refinement level. The resulting errors for the sequence of refined meshes are summarized in Table 2. For the fluid velocity and fluid pressure, the observed convergence rates are in agreement with the theoretical error decrease expected from an uncoupled problem. For the solid displacement, the observed convergence rates 1.46–1.9 for the H^1-error are better than the theoretically expected rate of ~ 1.

5.3. ***Flow around an elastic flap.*** In the second numerical example, we consider a channel flow around an elastic flap for different orientations of the flap with respect to the channel geometry. Here, we can take full advantage of the developed method and techniques as the overlapping mesh approach handles large deformation within a single simulation easily. As an additional benefit, our proposed scheme allows us to seamlessly reposition the flap for a series of numerical experiments and thus has great potential for future applications in design and optimization processes that involve fluid-structure interaction problems in their forward simulation; see for instance [38; 17].

Within the channel domain $\Omega = [0, L] \times [0, W] \times [0, H]$ with $L = 2.5$ and $W = H = 0.41$, the bottom side of the flap of dimensions $L^s = 0.06$, $W^s = 0.2$ and $H^s = 0.24$ is centered around the point $(L/2, W/2, 0)$. In the first numerical experiment, the flap is clamped on the boundary $[(L - L^s)/2, (L + L^s)/2] \times [(W - W^s)/2, (W + W^s)/2] \times \{0\}$ while the flap is rotated 65° around the z-axis in a second experiment. For the numerical experiment, we assume that the flow can be described by the Stokes equations with fluid viscosity $\nu^f = 0.001$ while the flap is modeled as an hyperelastic material satisfying the Saint-Venant–Kirchhoff constitutive equation (3-18) with the Lamé constants μ and λ defined by (5-4) for $E^s = 15$ and $\nu^s = 0.3$. We set the inflow profile $\boldsymbol{u}^f = (16 \cdot 0.45 y(W - y)z(H - z), 0, 0)$ at the inlet $\{0\} \times [0, W] \times [0, H]$, a "do-nothing" boundary condition given by

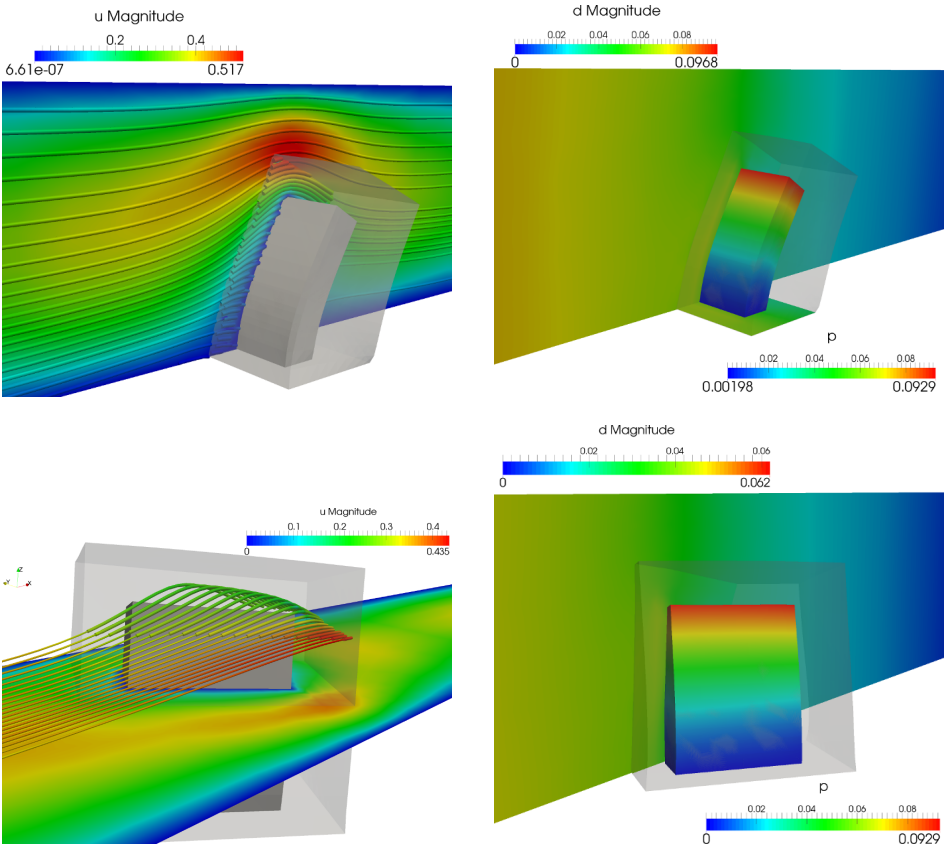


Figure 6. Flow around an elastic flap for two different flap orientations. Left: magnitude and streamlines of the velocity approximation in x-z (top) and x-y (bottom) cross-sections. The transparent block around the gray-colored flap visualizes the fluid mesh $\mathcal{T}_2^f$ surrounding the structure. The streamlines within $\mathcal{T}_2^f$ are drawn slightly thicker to illustrate the smooth transition of the velocity approximation from the outer to the inner fluid domain. Right: pressure distribution and magnitude of the structure displacement.

$\nu \partial_{\boldsymbol{n}} u - p\boldsymbol{n} = 0$ at the outlet $\{L\} \times [0, W] \times [0, H]$ and a no-slip condition $\boldsymbol{u} = \boldsymbol{0}$ elsewhere on the boundary.

The numerical results for aligned and rotated flaps are shown in Figure 6. We especially note the smooth transition of the velocity and pressure solutions from fluid background $\mathcal{T}_1^f$ to the solid-surrounding fluid mesh $\mathcal{T}_2^f$; the interface is not visible. The meshes used for simulation of the rotated flap are shown in Figure 7.

6. Conclusions

We presented a Nitsche-based cut and composite mesh method for fluid-structure interaction problems. The method utilizes a Nitsche-type coupling between two

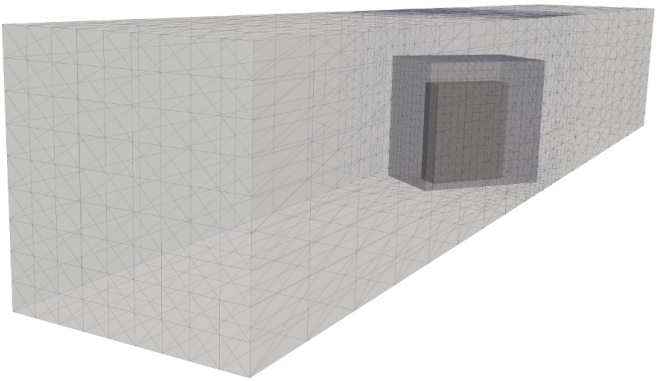


Figure 7. Background fluid mesh, structure mesh and its surrounding fluid mesh in the reference configuration.

fluid meshes: one fixed background mesh and one moving overlapping fluid mesh that is fitted to the boundary of a hyperelastic object and deforms with the object. The fluid-fluid coupling is monolithic in the sense that it determines a coupled system involving both the underlying and overlapping degrees of freedom. In previous work [41], we have shown that the coupling is stable and that the solution has optimal-order convergence for a stationary model problem.

To solve for the steady state solution of a fluid-structure interaction problem with large elastic deformations, we consider a fixed-point iteration where we solve for the fluid, compute a boundary traction for the solid, solve for the solid, solve for the mesh motion of the overlapping fluid mesh and finally update the geometry. This involves computing new intersections between underlying and overlapping meshes. Employing a provably stable overlapping mesh method for fluid-fluid coupling, the proposed scheme for the fluid-structure problem is guaranteed to be robust and insensitive to the overlap configuration.

We verified the expected convergence rates for a model problem with a manufactured solution and demonstrated the flexibility of our approach by computing the steady state solution for an elastic flap in a channel at two different orientations. It should be noted that the overlapping mesh method allows the flap to be repositioned in the channel without requiring the generation of a single *conforming* fluid mesh for each configuration. Only an elementwise, local representation of the cut cells near the interface together with some appropriate quadrature schemes are required; see for instance [39].

Future work involves extending our method to fully time-dependent flow governed by the incompressible Navier–Stokes equations. We note that the nonlinear convection term can be handled in our setting using a discontinuous Galerkin coupling with up-winding and that, from a computational point of view, taking a

time step is closely related to taking one step in our fixed-point iteration algorithm. Another area of interest is the direct coupling between fluids and solids.

Acknowledgments

This work is supported by an Outstanding Young Investigator grant from the Research Council of Norway, NFR 180450. This work is also supported by a Center of Excellence grant from the Research Council of Norway to the Center for Biomedical Computing at Simula Research Laboratory. The authors would like to thank the anonymous referee for the valuable comments and suggestions that helped to improve the presentation of this work.

This research was supported in part by the Swedish Foundation for Strategic Research Grant No. AM13-0029, the Swedish Research Council Grant 2013-4708, and the Swedish strategic research programme eSSENCE.

References

[1] M. J. Aftosmis, M. J. Berger, and J. E. Melton, *Robust and efficient Cartesian mesh generation for component-based geometry*, AIAA J. **36** (1998), no. 6, 952–960.

[2] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells, *Unified form language: a domain-specific language for weak formulations and partial differential equations*, ACM Trans. Math. Software **40** (2014), no. 2, 9. MR 3181899 Zbl 1308.65175

[3] S. Badia, F. Nobile, and C. Vergara, *Fluid-structure partitioned procedures based on Robin transmission conditions*, J. Comput. Phys. **227** (2008), no. 14, 7027–7051. MR 2009e:74026 Zbl 1140.74010

[4] ———, *Robin–Robin preconditioned Krylov methods for fluid-structure interaction problems*, Comput. Methods Appl. Mech. Engrg. **198** (2009), no. 33–36, 2768–2784. MR 2010i:65226 Zbl 1228.76079

[5] J. Baiges and R. Codina, *The fixed-mesh ALE approach applied to solid mechanics and fluid-structure interaction problems*, Internat. J. Numer. Methods Engrg. **81** (2010), no. 12, 1529–1557. MR 2010m:76105 Zbl 1183.74258

[6] J. W. Banks, W. D. Henshaw, and D. W. Schwendeman, *Deforming composite grids for solving fluid structure problems*, J. Comput. Phys. **231** (2012), no. 9, 3518–3547. MR 2902406 Zbl 06034758

[7] D. Boffi and L. Gastaldi, *A finite element approach for the immersed boundary method*, Comput. & Structures **81** (2003), no. 8–11, 491–501. MR 2004f:76081

[8] E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing, *CutFEM: discretizing geometry and partial differential equations*, Internat. J. Numer. Methods Engrg. (2014), online publication December.

[9] E. Burman, S. Claus, and A. Massing, *A stabilized cut finite element method for the three field Stokes problem*, Preprint, 2015, To appear in SIAM J. Sci. Comput. arXiv 1408.5165v2

[10] E. Burman and P. Hansbo, *Fictitious domain methods using cut elements, III: A stabilized Nitsche method for Stokes' problem*, ESAIM Math. Model. Numer. Anal. **48** (2014), no. 3, 859–874. MR 3264337 Zbl 06302445

[11] J. R. Cebral and R. Löhner, *Efficient simulation of blood flow past complex endovascular devices using an adaptive embedding technique*, IEEE T. Med. Imaging. **24** (2005), no. 4, 468–476.

[12] G. Chesshire and W. D. Henshaw, *Composite overlapping meshes for the solution of partial differential equations*, J. Comput. Phys. **90** (1990), no. 1, 1–64. MR 91f:76043 Zbl 0709.65090

[13] D. Day and P. Bochev, *Analysis and computation of a least-squares method for consistent mesh tying*, J. Comput. Appl. Math. **218** (2008), no. 1, 21–33. MR 2009e:65194 Zbl 1154.65088

[14] J. Donea, S. Giuliani, and J. P. Halleux, *An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid-structure interactions*, Comput. Methods Appl. Mech. Engrg. **33** (1982), no. 1–3, 689–723. Zbl 0508.73063

[15] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran, *Arbitrary Lagrangian–Eulerian methods*, Encyclopedia of computational mechanics (E. Stein, R. de Borst, and T. J. R. Hughes, eds.), vol. 1, Wiley, Chichester, 2014, pp. 413–437. MR 2007j:00004a Zbl 1190.76001

[16] O. Dorok, *Eine stabilisierte Finite-Elemente-Methode zur Lösung der Boussinesq-Approximation der Navier–Stokes-Gleichungen*, Ph.D. thesis, Otto-von-Guericke-Universität Magdeburg, 1995. Zbl 0868.76046

[17] B. Eguzkitza, G. Houzeaux, H. Calmet, M. Vázquez, B. Soni, S. Aliabadi, A. Bates, and D. Doorly, *A gluing method for non-matching meshes*, Comput. & Fluids **110** (2015), 159–168.

[18] R. E. English, L. Qiu, Y. Yu, and R. Fedkiw, *An adaptive discretization of incompressible flow using a multitude of moving Cartesian grids*, J. Comput. Phys. **254** (2013), 107–154. MR 3143360

[19] A. Gerstenberger and W. A. Wall, *Enhancement of fixed-grid methods towards complex fluid-structure interaction applications*, Internat. J. Numer. Methods Fluids **57** (2008), no. 9, 1227–1248. MR 2009f:74025 Zbl 05303306

[20] ———, *An eXtended Finite Element Method/Lagrange multiplier based approach for fluid-structure interaction*, Comput. Methods Appl. Mech. Engrg. **197** (2008), no. 19–20, 1699–1714. MR 2009b:74040 Zbl 1194.76117

[21] M. Giles, M. Larson, M. Levenstam, and E. Süli, *Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow*, Tech. report, 1997.

[22] *GNU Triangulated Surface Library*, software package, 2014.

[23] A. Hansbo, P. Hansbo, and M. G. Larson, *A finite element method on composite grids based on Nitsche's method*, Math. Model. Numer. Anal. **37** (2003), no. 3, 495–514. MR 2004f:65184 Zbl 1031.65128

[24] P. Hansbo and J. Hermansson, *Nitsche's method for coupling non-matching meshes in fluid-structure vibration problems*, Comput. Mech. **32** (2003), no. 1–2, 134–139. Zbl 1035.74055

[25] P. Hansbo, M. G. Larson, and S. Zahedi, *A cut finite element method for a Stokes interface problem*, Appl. Numer. Math. **85** (2014), 90–114. MR 3239219 Zbl 1299.76136

[26] W. D. Henshaw and D. W. Schwendeman, *Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow*, J. Comput. Phys. **216** (2006), no. 2, 744–779. MR 2007d:76184 Zbl 1220.76052

[27] G. Houzeaux and R. Codina, *A Chimera method based on a Dirichlet/Neumann (Robin) coupling for the Navier–Stokes equations*, Comput. Methods Appl. Mech. Engrg. **192** (2003), no. 31–32, 3343–3377. MR 1992793 Zbl 1054.76049

[28] V. John, *Parallele Lösung der inkompressiblen Navier–Stokes Gleichungen auf adaptiv verfeinerten Gittern*, Ph.D. thesis, Otto-von-Guericke-Universität Magdeburg, 1997. Zbl 0908.76054

[29] R. C. Kirby and A. Logg, *A compiler for variational forms*, ACM Trans. Math. Software **32** (2006), no. 3, 417–444. MR 2007j:65105

[30] U. Küttler and W. A. Wall, *Fixed-point fluid-structure interaction solvers with dynamic relaxation*, Comput. Mech. **43** (2008), no. 1, 61–72. Zbl 1236.74284

[31] _____, *Vector extrapolation for strong coupling fluid-structure interaction solvers*, J. Appl. Mech. **76** (2009), no. 2, 021205.

[32] P. Le Tallec and J. Mouro, *Fluid structure interaction with large structural displacements*, Comput. Methods Appl. Mech. Engrg. **190** (2001), no. 24–25, 3039–3067. Zbl 1001.74040

[33] A. Logg, *Automating the finite element method*, Arch. Comput. Methods Eng. **14** (2007), no. 2, 93–138. MR 2008f:65259 Zbl 1158.74048

[34] A. Logg, K.-A. Mardal, and G. N. Wells (eds.), *Automated solution of differential equations by the finite element method: the FEniCS book*, Lect. Notes Comput. Sci. Eng., no. 84, Springer, Berlin, 2012. MR 3075806 Zbl 1247.65105

[35] A. Logg and G. N. Wells, *DOLFIN: automated finite element computing*, ACM Trans. Math. Software **37** (2010), no. 2, 20. MR 2011i:65219

[36] R. Löhner, J. R. Cebral, F. E. Camelli, S. Appanaboyina, J. D. Baum, E. L. Mestreau, and O. A. Soto, *Adaptive embedded and immersed unstructured grid techniques*, Comput. Methods Appl. Mech. Engrg. **197** (2008), no. 25–28, 2173–2197. MR 2412819 Zbl 1158.76408

[37] R. Löhner, J. R. Cebral, F. F. Camelli, J. D. Baum, E. L. Mestreau, and O. A. Soto, *Adaptive embedded/immersed unstructured grid techniques*, Arch. Comput. Methods Eng. **14** (2007), no. 3, 279–301. MR 2008g:65182 Zbl 1127.76047

[38] E. Lund, H. Møller, and L. A. Jakobsen, *Shape design optimization of stationary fluid-structure interaction problems with large displacements and turbulence*, Struct. Multidiscip. O. **25** (2003), no. 5–6, 383–392.

[39] A. Massing, M. G. Larson, and A. Logg, *Efficient implementation of finite element methods on nonmatching and overlapping meshes in three dimensions*, SIAM J. Sci. Comput. **35** (2013), no. 1, C23–C47. MR 3033077 Zbl 1264.65194

[40] A. Massing, M. G. Larson, A. Logg, and M. E. Rognes, *A stabilized Nitsche fictitious domain method for the Stokes problem*, J. Sci. Comput. **61** (2014), no. 3, 604–628. MR 3268662 Zbl 06389878

[41] _____, *A stabilized Nitsche overlapping mesh method for the Stokes problem*, Numer. Math. **128** (2014), no. 1, 73–101. MR 3248049 Zbl 06341753

[42] S. M. Murman, M. J. Aftosmis, and M. J. Berger, *Implicit approaches for moving boundaries in a 3-D Cartesian method*, 41st Aerospace Sciences Meeting and Exhibit (Reno, NV, 2003), AIAA, Reston, VA, 2003, p. 1119.

[43] F. Nobile, *Numerical approximation of fluid-structure interaction problems with application to haemodynamics*, Ph.D. thesis, École polytechnique fédérale de Lausanne, 2001.

[44] C. S. Peskin, *Numerical analysis of blood flow in the heart*, J. Comput. Phys. **25** (1977), no. 3, 220–252. MR 58 #9389 Zbl 0403.76100

[45] _____, *The immersed boundary method*, Acta Numer. **11** (2002), 479–517. MR 2004h:74029 Zbl 1123.74309

[46] M. A. Puso, E. Kokko, R. Settgast, J. Sanders, B. Simpkins, and B. Liu, *An embedded mesh method using piecewise constant multipliers with stabilization: mathematical and numerical aspects*, Int. J. Numer. Meth. Eng. (2014), online publication October.

[47] S. Shahmiri, A. Gerstenberger, and W. A. Wall, *An XFEM-based embedding mesh technique for incompressible viscous flows*, Internat. J. Numer. Methods Fluids **65** (2011), no. 1–3, 166–190. MR 2012a:76097 Zbl 05837893

[48] G. Starius, *Composite mesh difference methods for elliptic boundary value problems*, Numer. Math. **28** (1977), no. 2, 243–258. MR 57 #1923 Zbl 0363.65078

[49] ———, *On composite mesh difference methods for hyperbolic differential equations*, Numer. Math. **35** (1980), no. 3, 241–255. MR 82b:65089 Zbl 0475.65059

[50] J. L. Steger, F. C. Dougherty, and J. A. Benek, *A Chimera grid scheme*, Advances in grid generation (Houston, TX, 1983) (K. N. Ghia and U. Ghia, eds.), Fluids Engineering Division, no. 5, ASME, New York, 1983, pp. 59–70.

[51] The CGAL Project, *CGAL user and reference manual*, 4.6.1 ed., CGAL Editorial Board, 2015.

[52] E. A. Volkov, *The method of composite meshes for finite and infinite regions with piecewise smooth boundary*, Trudy Mat. Inst. Steklov. **96** (1968), 117–148, In Russian; translated in Proc. Steklov Inst. Math. **96** (1968), 145–185. MR 42 #8719 Zbl 0207.09502

[53] W. A. Wall, A. Gerstenberger, P. Gamnitzer, C. Förster, and E. Ramm, *Large deformation fluid-structure interaction: advances in ALE methods and new fixed grid approaches*, Fluid-structure interaction: modelling, simulation, optimisation (Hohenwart, Germany, 2005) (H.-J. Bungartz and M. Schäfer, eds.), Lect. Notes Comput. Sci. Eng., no. 53, Springer, Berlin, 2006, pp. 195–232. MR 2007f:74030 Zbl 1097.76002

[54] Z. J. Wang and V. Parthasarathy, *A fully automated Chimera methodology for multiple moving body problems*, Int. J. Numer. Meth. Fl. **33** (2000), no. 7, 919–938. Zbl 0984.76073

[55] Z. Yu, *A DLM/FD method for fluid/flexible-body interactions*, J. Comput. Phys. **207** (2005), no. 1, 1–27. Zbl 1177.76304

[56] L. T. Zhang and M. Gay, *Immersed finite element method for fluid-structure interactions*, J. Fluid. Struct. **23** (2007), no. 6, 839–857.

[57] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu, *Immersed finite element method*, Comput. Methods Appl. Mech. Engrg. **193** (2004), no. 21–22, 2051–2067. MR 2071550 Zbl 1067.76576

Received November 11, 2013. Revised February 21, 2015.

ANDRÉ MASSING: andre.massing@math.umu.se
Department of Mathematics and Mathematical Statistics, Umeå University, 901 87 Umeå, Sweden

MATS G. LARSON: mats.larson@math.umu.se
Department of Mathematics and Mathematical Statistics, Umeå University, 901 87 Umeå, Sweden

ANDERS LOGG: logg@chalmers.se
Department of Mathematical Sciences, Chalmers University of Technology, 412 96 Göteborg, Sweden

and

Department of Mathematical Sciences, University of Gothenburg, 412 61 Göteborg, Sweden

MARIE E. ROGNES: meg@simula.no
Simula Research Laboratory, 1364 Fornebu, Norway

COMM. APP. MATH. AND COMP. SCI.
Vol. 10, No. 2, 2015
dx.doi.org/10.2140/camcos.2015.10.121

AN ADAPTIVE MULTIBLOCK HIGH-ORDER FINITE-VOLUME METHOD FOR SOLVING THE SHALLOW-WATER EQUATIONS ON THE SPHERE

PETER MCCORQUODALE, PAUL A. ULLRICH,
HANS JOHANSEN AND PHILLIP COLELLA

We present a high-order finite-volume approach for solving the shallow-water equations on the sphere, using multiblock grids on the cubed sphere. This approach combines a Runge–Kutta time discretization with a fourth-order-accurate spatial discretization and includes adaptive mesh refinement and refinement in time. Results of tests show fourth-order convergence for the shallow-water equations as well as for advection in a highly deformational flow. Hierarchical adaptive mesh refinement allows solution error to be achieved that is comparable to that obtained with uniform resolution of the most refined level of the hierarchy but with many fewer operations.

1. Introduction

In this paper, we present a method of local refinement applied to the 2D shallow-water equations, using test cases that capture some of the essential features that arise in 3D atmospheric models. We extend a uniform-grid finite-volume discretization on the surface of a sphere to a locally refined, nested grid hierarchy that can evolve in time, and can therefore resolve or track small-scale and synoptic features, without refining the entire computational domain. Similar high-accuracy block-structured adaptive mesh refinement (AMR) approaches have been applied to problems in compressible gas dynamics [32; 19]. For climate applications, AMR techniques hold the promise of spanning global and regional scales as well as tracking synoptic features that contribute significantly to climate means in the Earth system. Computational cost limits the finest resolution of uniform-resolution climate models to around 10 km, far larger than the grid spacing necessary for resolving clouds and features of regional climate. The highest-resolution simulations have become

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the United States Department of Energy under Contract Number DE-AC02-05CH11231.
MSC2010: primary 35L40, 65M50, 65M08; secondary 35L65, 86-08.
Keywords: high order, finite-volume method, cubed sphere, shallow-water equations, adaptive mesh refinement.

important for regional planning issues, which rely on accurate representation of changes in the behavior of mesoscale storm systems, pressure-blocking events driven by topography (responsible for heat waves and cold spells), mountain snowpack, wildfires, topographically driven precipitation, watershed-level hydrology, and urban development and agriculture. As emphasized in [58], addressing these challenges requires high-resolution regional climate modeling via either dynamical downscaling or highly refined grids. Moving synoptic features, such as extratropical and tropical cyclones, would benefit from space-time adaptivity to better resolve their dynamics. Thus, AMR can both improve the resolution of atmospheric flows and help test physical parametrizations across spatial and temporal scales in a global context, without refining the entire computational domain.

As a first step in the development of a global atmospheric modeling system, in this paper, we solve the 2D shallow-water equations, which capture many of the important properties of the equations of motion for the atmosphere. In particular, the dynamical character of the global shallow-water equations is governed by features common with atmospheric motions, including barotropic Rossby waves and inertia-gravity waves, without the added complexity of a vertical dimension. There already exists a comprehensive literature on the development of numerical methods for the global shallow-water equations spanning the past several decades. Examples include the spectral-transform method [25], semi-Lagrangian methods [41; 4; 53; 63; 54; 38], finite-difference methods [21; 42], Godunov-type finite-volume methods [43; 57], staggered finite-volume methods [29; 39; 40], multimoment finite-volume methods [8; 27; 7], and finite-element methods [51; 12; 52; 17; 33; 26; 11; 2].

As of the time of writing, work targeting AMR for the global shallow-water equations is much more sparse. Two adaptive numerical methods (finite-volume on a latitude-longitude grid and nonconservative finite-element on a cubed-sphere grid) are described in [49]. A discontinuous Galerkin formulation for global tsunami simulation is described in [5]. The multimoment finite-volume approach has also been extended to an adaptive formulation by [9]. The present article introduces an AMR approach for the shallow-water equations that also supports refinement in time.

Atmospheric models include a wide variety of computational grids on the sphere such as the latitude-longitude mesh [62; 28], icosahedral and hexagonal grids [16; 48; 18; 45; 59], and cubed-sphere meshes [56; 13; 36]. In particular, icosahedral, hexagonal, and cubed-sphere meshes have become popular over the last decade as they provide an almost-regular grid-point coverage on the sphere. The uniform distribution of elements avoids the coordinate singularities at the poles that complicate the design of stable and accurate methods for such coordinate systems.

The approach in this paper is based on the finite-volume mapped-grid technology in [10], which is extended to work with AMR in [19]. We apply these methods on cubed-sphere meshes, which consist of six panels with a separate mapping on each

panel. To coordinate the different mappings along panel boundaries, we use the mapped-multiblock approach of [31] with the following modifications:

(1) Because the computational domain is on the surface of a sphere, which is a 2D manifold in a 3D space, the evolution equations must include metric terms.

(2) Because we have vector quantities (velocities and momenta) that are expressed in different bases on different panels, the procedure for coordinating them across a panel boundary must include a basis transformation.

For smooth solutions, this approach can provide fourth-order-accurate results as also achieved in [57]. Comparing these results to those of [43] shows the advantage of fourth-order over second-order methods in avoiding artifacts at the boundaries of the cubed-sphere panels. The dispersive properties of this method have been analyzed by [55], where it was demonstrated that the use of a fourth-order finite-volume discretization led to a doubling of the effective resolution compared to a second-order approach. High-order accuracy is also necessary in the context of grid refinement since there is a formal drop of one order of accuracy (in the maximum norm) at grid-refinement boundaries. Hence, a second-order adaptive method would drop to first-order accuracy in the presence of grid refinement, with disastrous consequences to the quality of the solution, whereas a fourth-order method only drops to third-order. Further, compared to other numerical methods, including standard finite-element discretizations, central finite-volume methods provide the largest maximum stable time-step size and do not suffer from issues such as the "spectral gap" that arise from nonuniform treatments. In the absence of limiters and explicit dissipation, these schemes are also energy-conservative up to temporal truncation order.

2. Partial differential equations in cubed-sphere coordinates

The equiangular cubed-sphere grid [44; 42] consists of a cube with six Cartesian patches arranged along each face, which is then "deflated" onto a tangent spherical shell, as shown in Figure 1. It is a *quasiuniform spherical grid*; that is, it is in the class of grids that provide an approximately uniform tiling of the sphere (see [50], for example, for a review of different options for global grids). The equiangular cubed-sphere grid has the advantage of being among the most uniform of cubed-sphere grids: at high resolutions, the ratio of largest to smallest grid cell approaches $\sqrt{2}$, compared to the equidistant gnomonic cubed-sphere grid, which approaches a ratio of $3\sqrt{3}$, and the conformal cubed-sphere grid, where this ratio is unbounded. Although even more uniformity can be attained via the application of grid-relaxation techniques such as spring dynamics (see, for example, [37]), these techniques also lead to nonanalytical forms of the curvature metrics, which in turn increases the complexity of the discretization.

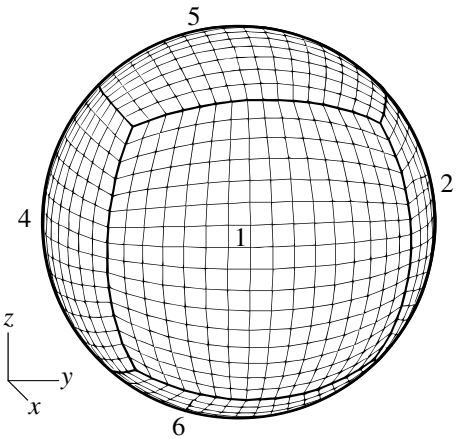


Figure 1. A cubed-sphere grid, shown with labels on panels. Panels 1–4 all straddle the equator ($z = 0$) of the unit sphere. Panel 5 is centered on the north pole ($z = +1$) and Panel 6 on the south pole ($z = -1$). On the cubed-sphere grid shown here, $N_c = 16$ (each panel contains 16×16 grid cells).

On the equiangular cubed-sphere grid, coordinates are given as (α, β, n_p), with central angles $\alpha, \beta \in [-\pi/4, \pi/4]$ and panel index $n_p \in \{1, 2, 3, 4, 5, 6\}$. By convention, we choose Panels 1–4 to be along the equator and Panels 5 and 6 to be centered on the northern and southern poles, respectively.

We will also use spherical coordinates (λ, ϕ) with longitude $\lambda \in [0, 2\pi]$ and latitude $\phi \in [-\pi/2, \pi/2]$ for plotting and specification of tests. Coordinate transforms between spherical and equiangular coordinates can be found in [56, Appendix A].

2.1. ***Metrics.*** Coordinates (X, Y) are related to equiangular coordinates (α, β) via the transform

$$X = \tan\alpha, \qquad Y = \tan\beta. \tag{1}$$

Any straight line in (X, Y) coordinates is also a great circle arc, which is not the case for general line segments in equiangular coordinates. Throughout this paper, we will be making use of the metric term

$$\delta = (1 + \tan^2\alpha + \tan^2\beta)^{1/2}, \tag{2}$$

which appears frequently in geometric calculations on the cubed-sphere grid.

Cartesian coordinates are related to the equiangular coordinates of a particular cubed-sphere panel by $\boldsymbol{x}(\alpha, \beta) = (x(\alpha, \beta), y(\alpha, \beta), z(\alpha, \beta))$. The natural basis vectors of the equiangular coordinate system are $\boldsymbol{g}_\alpha = (\partial \boldsymbol{x}/\partial\alpha)_\beta$ and $\boldsymbol{g}_\beta = (\partial \boldsymbol{x}/\partial\beta)_\alpha$, which have units of length.

The covariant 2D metric on the cubed sphere of radius r is given by

$$g_{pq} = \boldsymbol{g}_p \cdot \boldsymbol{g}_q = \frac{r^2(1+X^2)(1+Y^2)}{\delta^4} \begin{pmatrix} 1+X^2 & -XY \\ -XY & 1+Y^2 \end{pmatrix}, \tag{3}$$

with contravariant inverse

$$g^{pq} = \frac{\delta^2}{r^2(1+X^2)(1+Y^2)} \begin{pmatrix} 1+Y^2 & XY \\ XY & 1+X^2 \end{pmatrix}. \tag{4}$$

The Jacobian on the manifold is then

$$J = \sqrt{\det g_{pq}} = \frac{r^2(1+X^2)(1+Y^2)}{\delta^3} \tag{5}$$

and induces the infinitesimal area element $dA = J\, d\alpha\, d\beta$.

For a comprehensive mathematical description of the equiangular cubed-sphere grid, see [33, Appendices A, B, and C] or [56, Appendices A and B].

2.2. ***The shallow-water equations in cubed-sphere coordinates.*** In conservative coordinate-invariant form, the 2D shallow-water equations on the sphere can be written as

$$\frac{\partial H}{\partial t} + \nabla \cdot (h\boldsymbol{u}) = 0, \tag{6}$$

$$\frac{\partial h\boldsymbol{u}}{\partial t} + \nabla \cdot \left(h\boldsymbol{u}\boldsymbol{u} + \mathcal{I}\frac{Gh^2}{2} \right) = -Gh\nabla z_s - f\boldsymbol{g}_r \times (h\boldsymbol{u}), \tag{7}$$

where H denotes the fluid surface height above the reference depth $z = 0$, h is the fluid depth above the bottom topography $z = z_s(\lambda, \phi)$, $\boldsymbol{u}$ is the velocity vector, $\boldsymbol{u}\boldsymbol{u}$ denotes the outer product of the velocity, $\mathcal{I}$ is the identity matrix, $G = 9.80616\,\mathrm{m\cdot s^{-2}}$ is the acceleration due to gravity, $f = 2\Omega \sin\phi$ is the Coriolis parameter in terms of the rotation rate $\Omega = 7.292 \times 10^{-5}\,\mathrm{s^{-1}}$, and $\boldsymbol{g}_r$ is the unit vector perpendicular to the surface of the sphere. The quantities H, h, and z_s are related via $H = h + z_s$.

Under equiangular coordinates, the velocity field is written as

$$\boldsymbol{u} = u^\alpha \boldsymbol{g}_\alpha + u^\beta \boldsymbol{g}_\beta. \tag{8}$$

The coefficients u^α and u^β are known as the contravariant components of the velocity vector and have units of rad/s in the natural basis.

The height evolution equation (6) then takes the form

$$\frac{\partial H}{\partial t} + \frac{1}{J}\frac{\partial}{\partial \alpha}(Jhu^\alpha) + \frac{1}{J}\frac{\partial}{\partial \beta}(Jhu^\beta) = 0. \tag{9}$$

The momentum evolution equation (7) can be decomposed into an evolution equation for hu^α and hu^β,

$$\frac{\partial}{\partial t}\begin{pmatrix} hu^\alpha \\ hu^\beta \end{pmatrix} + \frac{1}{J}\frac{\partial}{\partial \alpha}\begin{pmatrix} J\mathcal{T}^{\alpha\alpha} \\ J\mathcal{T}^{\beta\alpha} \end{pmatrix} + \frac{1}{J}\frac{\partial}{\partial \beta}\begin{pmatrix} J\mathcal{T}^{\alpha\beta} \\ J\mathcal{T}^{\beta\beta} \end{pmatrix} = \boldsymbol{\Psi}_M + \boldsymbol{\Psi}_B + \boldsymbol{\Psi}_C, \tag{10}$$

where $\mathcal{T}^{kn} = hu^k u^n + g^{kn}\frac{1}{2}Gh^2$ and $\boldsymbol{\Psi}_M$, $\boldsymbol{\Psi}_B$, and $\boldsymbol{\Psi}_C$ denote source terms due to the curvature of the manifold, bottom topography, and Coriolis force, respectively.

The manifold source term takes the form

$$\boldsymbol{\Psi}_{\mathrm{M}} = \begin{pmatrix} -\Gamma^{\alpha}_{\ nk}\mathcal{T}^{kn} \\ -\Gamma^{\beta}_{\ nk}\mathcal{T}^{kn} \end{pmatrix} = \frac{2}{\delta^2}\begin{pmatrix} -XY^2 hu^{\alpha}u^{\alpha} + Y(1+Y^2)hu^{\alpha}u^{\beta} \\ X(1+X^2)hu^{\alpha}u^{\beta} - X^2 Y hu^{\beta}u^{\beta} \end{pmatrix}, \tag{11}$$

where $\Gamma^{m}_{\ nk}$ are the Christoffel symbols of the second kind associated with the metric. The source term due to bottom topography can be written in terms of derivatives of z_{s} as

$$\boldsymbol{\Psi}_{\mathrm{B}} = -Gh\begin{pmatrix} g^{\alpha k}\nabla_k z_{\mathrm{s}} \\ g^{\beta k}\nabla_k z_{\mathrm{s}} \end{pmatrix} = -Gh\begin{pmatrix} g^{\alpha\alpha} & g^{\alpha\beta} \\ g^{\beta\alpha} & g^{\beta\beta} \end{pmatrix}\begin{pmatrix} \partial z_{\mathrm{s}}/\partial\alpha \\ \partial z_{\mathrm{s}}/\partial\beta \end{pmatrix}. \tag{12}$$

The Coriolis source term differs depending on whether the underlying panel is equatorial or polar since

$$\sin\phi = \begin{cases} Y/\delta & \text{if } n_{\mathrm{p}} \in \{1,2,3,4\}, \\ p/\delta & \text{if } n_{\mathrm{p}} \in \{5,6\}, \end{cases} \tag{13}$$

where p is a panel indicator given by, for instance,

$$p = \operatorname{sign}\phi = \begin{cases} +1 & \text{on the northern panel } (n_{\mathrm{p}} = 5), \\ -1 & \text{on the southern panel } (n_{\mathrm{p}} = 6). \end{cases} \tag{14}$$

For equatorial panels, the Coriolis source term is given by

$$\boldsymbol{\Psi}_{\mathrm{C,eq}} = \frac{2\Omega}{\delta^2}\begin{pmatrix} -XY^2 & Y(1+Y^2) \\ -Y(1+X^2) & XY^2 \end{pmatrix}\begin{pmatrix} hu^{\alpha} \\ hu^{\beta} \end{pmatrix} \tag{15}$$

and on polar panels by

$$\boldsymbol{\Psi}_{\mathrm{C,pol}} = \frac{2p\Omega}{\delta^2}\begin{pmatrix} -XY & (1+Y^2) \\ -(1+X^2) & XY \end{pmatrix}\begin{pmatrix} hu^{\alpha} \\ hu^{\beta} \end{pmatrix}. \tag{16}$$

Multiplying both sides of the shallow-water equations (9)–(10) by J and using the fact that J and the topography $z_{\mathrm{s}} = H - h$ are independent of t, these evolution equations can be written

$$\frac{\partial}{\partial t}(J\boldsymbol{U}) + \nabla\cdot(J\vec{\boldsymbol{F}}) = J\boldsymbol{\Psi}, \tag{17}$$

where

$$\boldsymbol{U} = \begin{pmatrix} h \\ hu^{\alpha} \\ hu^{\beta} \end{pmatrix}, \qquad \boldsymbol{F}^k = \begin{pmatrix} hu^k \\ \mathcal{T}^{\alpha k} \\ \mathcal{T}^{\beta k} \end{pmatrix}, \qquad \boldsymbol{\Psi} = \begin{pmatrix} 0 \\ \boldsymbol{\Psi}_{\mathrm{M}} + \boldsymbol{\Psi}_{\mathrm{B}} + \boldsymbol{\Psi}_{\mathrm{C}} \end{pmatrix}. \tag{18}$$

Here $\boldsymbol{U}$ contains the *conserved* variables, which are functions of the *primitive* variables,

$$\boldsymbol{W} = \begin{pmatrix} h \\ u^\alpha \\ u^\beta \end{pmatrix}. \tag{19}$$

The components of $\vec{\boldsymbol{F}}$ are functions of the primitive variables and the metric.

2.3. ***Advection in cubed-sphere coordinates.*** In conservative coordinate-invariant form, the 2D advection equation on the sphere is just the first equation of (17):

$$\frac{\partial}{\partial t}(J\boldsymbol{U}) + \nabla \cdot (J\vec{\boldsymbol{F}}) = 0 \tag{20}$$

with only one component, $\boldsymbol{U} = h$ and $\boldsymbol{F}^k = hu^k$. Here, h is interpreted as the density of the advected quantity, and $\boldsymbol{u}(\alpha, \beta, t)$ is a prescribed velocity vector field.

3. Finite-volume discretization on cubed-sphere grids

3.1. ***Discretization of the cubed sphere.*** The discrete resolution of the cubed sphere is typically written in the form c$\{N_c\}$, where each coordinate direction consists of N_c grid cells. For instance, the cubed-sphere grid shown in Figure 1 is c16. The total number of grid cells on a cubed sphere is $N_c \times N_c \times 6$. A grid cell on a particular panel is denoted by $V_{i,j}$ with indices $(i, j) \in [0, \ldots, N_c - 1]^2$, which refers to the region bounded by

$$\alpha \in [i\Delta\alpha - \tfrac{1}{4}\pi, (i+1)\Delta\alpha - \tfrac{1}{4}\pi], \qquad \beta \in [j\Delta\beta - \tfrac{1}{4}\pi, (j+1)\Delta\beta - \tfrac{1}{4}\pi], \tag{21}$$

where on an equiangular grid the grid spacing is

$$\Delta\alpha = \Delta\beta = \frac{\pi}{2N_c}. \tag{22}$$

The center of $V_{i,j}$ is the point (α_i, β_j) with

$$\alpha_i = (i + \tfrac{1}{2})\Delta\alpha - \tfrac{1}{4}\pi, \qquad \beta_j = (j + \tfrac{1}{2})\Delta\beta - \tfrac{1}{4}\pi. \tag{23}$$

Some properties of the cubed-sphere grid for a variety of resolutions are given in Table 1.

3.2. ***PDE discretization.*** We can integrate a PDE of the form

$$\frac{\partial}{\partial t}(J\boldsymbol{U}) + \nabla \cdot (J\vec{\boldsymbol{F}}) = J\boldsymbol{\Psi} \tag{24}$$

over a grid cell $V_{i,j}$, giving

$$\frac{d}{dt}\iint_{V_{i,j}} J\boldsymbol{U}\, d\alpha\, d\beta + \iint_{V_{i,j}} \nabla \cdot (J\vec{\boldsymbol{F}})\, d\alpha\, d\beta = \iint_{V_{i,j}} J\boldsymbol{\Psi}\, d\alpha\, d\beta. \tag{25}$$

Resolution	Δx	A_{avg}	$A_{\text{min}}/A_{\text{max}}$	$\text{RLL}_{\text{equiv}}$	T_{equiv}
c16	625 km	3.321×10^5 km^2	0.7434	6.5°	$T17$
c32	313 km	8.302×10^4 km^2	0.7249	3.2°	$T34$
c64	156 km	2.076×10^4 km^2	0.7159	1.6°	$T68$
c128	78.2 km	5.189×10^3 km^2	0.7115	0.82°	$T136$
c256	39.1 km	1.297×10^3 km^2	0.7093	0.41°	$T272$

Table 1. Properties of the cubed-sphere grid for different resolutions. Here Δx is the grid spacing at the equator, A_{avg} is the average area of all cubed-sphere grid cells, A_{min} is the minimum cell area, and A_{max} is the maximum cell area. $\text{RLL}_{\text{equiv}}$ denotes the equivalent grid spacing (in degrees) on the regular latitude-longitude grid with the same number of cells, and T_{equiv} denotes the approximate triangular truncation of a spectral transform method.

Then applying the divergence theorem to the second term on the left-hand side of (25):

$$\frac{d}{dt}\iint_{V_{i,j}} J\boldsymbol{U}\,d\alpha\,d\beta + \oint_{\partial V_{i,j}} J\vec{\boldsymbol{F}}\cdot\hat{\boldsymbol{n}}\,d\ell = \iint_{V_{i,j}} J\boldsymbol{\Psi}\,d\alpha\,d\beta. \tag{26}$$

We can represent the integrals in (26) in terms of averages over $V_{i,j}$ and its faces. The notation for an average of a quantity $A(\alpha,\beta)$ over $V_{i,j}$ is

$$\langle A\rangle_{i,j} = \frac{\displaystyle\iint_{V_{i,j}} A(\alpha,\beta)\,d\alpha\,d\beta}{\displaystyle\iint_{V_{i,j}} d\alpha\,d\beta} = \frac{\displaystyle\int_{\beta_j-\frac{1}{2}\Delta\beta}^{\beta_j+\frac{1}{2}\Delta\beta}\int_{\alpha_i-\frac{1}{2}\Delta\alpha}^{\alpha_i+\frac{1}{2}\Delta\alpha} A(\alpha,\beta)\,d\alpha\,d\beta}{\Delta\alpha\,\Delta\beta}. \tag{27}$$

Averages over faces of $V_{i,j}$ with constant $\alpha = \alpha_i \pm \frac{1}{2}\Delta\alpha$ and $\beta = \beta_j \pm \frac{1}{2}\Delta\beta$ are denoted, respectively,

$$\langle A\rangle_{i\pm\frac{1}{2},j} = \frac{\displaystyle\int_{\beta_j-\frac{1}{2}\Delta\beta}^{\beta_j+\frac{1}{2}\Delta\beta} A(\alpha_i \pm \tfrac{1}{2}\Delta\alpha,\beta)\,d\beta}{\Delta\beta}, \tag{28}$$

$$\langle A\rangle_{i,j\pm\frac{1}{2}} = \frac{\displaystyle\int_{\alpha_i-\frac{1}{2}\Delta\alpha}^{\alpha_i+\frac{1}{2}\Delta\alpha} A(\alpha,\beta_j \pm \tfrac{1}{2}\Delta\beta)\,d\alpha}{\Delta\alpha}. \tag{29}$$

Then dividing both sides of (26) by $\Delta\alpha\,\Delta\beta$ and substituting the averages as defined in (27)–(29):

$$\frac{d}{dt}\langle J\boldsymbol{U}\rangle_{i,j} = -\frac{1}{\Delta\alpha}\left(\langle J\boldsymbol{F}^{\alpha}\rangle_{i+\frac{1}{2},j} - \langle J\boldsymbol{F}^{\alpha}\rangle_{i-\frac{1}{2},j}\right) - \frac{1}{\Delta\beta}\left(\langle J\boldsymbol{F}^{\beta}\rangle_{i,j+\frac{1}{2}} - \langle J\boldsymbol{F}^{\beta}\rangle_{i,j-\frac{1}{2}}\right) + \langle J\boldsymbol{\Psi}\rangle_{i,j}. \tag{30}$$

3.3. ***Temporal discretization.*** We apply the classical fourth-order Runge–Kutta method to integrate (30), which can be written in the form

$$\frac{d}{dt}\langle J\boldsymbol{U}\rangle_{i,j} = K(\langle J\boldsymbol{U}\rangle)_{i,j} \tag{31}$$

over grid cell $V_{i,j}$, where

$$K(\langle J\boldsymbol{U}\rangle)_{i,j} = -\frac{1}{\Delta\alpha}\left(\langle J\boldsymbol{F}^{\alpha}\rangle_{i+\frac{1}{2},j} - \langle J\boldsymbol{F}^{\alpha}\rangle_{i-\frac{1}{2},j}\right) - \frac{1}{\Delta\beta}\left(\langle J\boldsymbol{F}^{\beta}\rangle_{i,j+\frac{1}{2}} - \langle J\boldsymbol{F}^{\beta}\rangle_{i,j-\frac{1}{2}}\right) + \langle J\boldsymbol{\Psi}\rangle_{i,j}. \tag{32}$$

In Section 3.4, we show how to derive fourth-order accurate approximations to $K(\langle J\boldsymbol{U}\rangle)$ on grid cells given $\langle J\boldsymbol{U}\rangle$ on grid cells.

The classical Runge–Kutta method applied to the ordinary differential equation (31) integrated over time step Δt starting with $\langle J\boldsymbol{U}\rangle^{(0)}$ at the initial time is

$$k_1 = K(\langle J\boldsymbol{U}\rangle^{(0)})\Delta t, \tag{33}$$

$$\langle J\boldsymbol{U}\rangle^{(1)} = \langle J\boldsymbol{U}\rangle^{(0)} + \tfrac{1}{2}k_1, \qquad k_2 = K(\langle J\boldsymbol{U}\rangle^{(1)})\Delta t, \tag{34}$$

$$\langle J\boldsymbol{U}\rangle^{(2)} = \langle J\boldsymbol{U}\rangle^{(0)} + \tfrac{1}{2}k_2, \qquad k_3 = K(\langle J\boldsymbol{U}\rangle^{(2)})\Delta t, \tag{35}$$

$$\langle J\boldsymbol{U}\rangle^{(3)} = \langle J\boldsymbol{U}\rangle^{(0)} + k_3, \qquad k_4 = K(\langle J\boldsymbol{U}\rangle^{(3)})\Delta t. \tag{36}$$

Then to integrate one time step:

$$\langle J\boldsymbol{U}\rangle(t^n + \Delta t) = \langle J\boldsymbol{U}\rangle(t^n) + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + O((\Delta t)^5). \tag{37}$$

With local truncation error of $O((\Delta t)^5)$, as shown in (37), the accumulated error for the classical Runge–Kutta method is then $O((\Delta t)^4)$.

3.4. ***Spatial discretization.*** If Ω is the set of ordered pairs of indices (i, j) over which we find $\langle J\boldsymbol{U}\rangle_{i,j}$, then let $\mathcal{G}_{m,n}(\Omega)$, with m and n integers, be the set of grid cells Ω expanded by m layers of additional cells at both ends in the α direction and n layers of additional cells at both ends in the β direction. These additional cells are called *ghost cells*. For a set of indices Λ of grid cells and ghost cells, let $\mathcal{F}^{\alpha}(\Lambda)$ be the set of their faces of constant α and $\mathcal{F}^{\beta}(\Lambda)$ the set of their faces of constant β.

In the remainder of this section, we show how to compute the right-hand side of (30), the evolution equation for $\langle J\boldsymbol{U}\rangle$. The method is motivated by that in [32] for Cartesian grids, extended to mapped grids in [10] and to mapped multiblock grids in [31]. What is new here is that we are calculating on a 2D manifold in 3D and also that we have vector components that require a basis transformation (Step (2) below).

The discrete undivided-difference formulae denoted by D_{α} and D_{β} with various superscripts are defined in Appendix A.

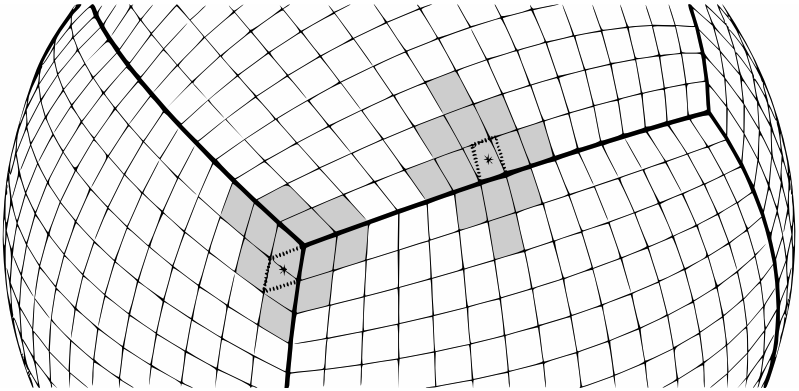


Figure 2. Sample interpolation stencils of two different ghost cells, used in Step (2). The procedure for finding the stencil is explained in [31]. The set of grid cells in the stencil is found as follows. First, find the center of the ghost cell on the cubed sphere, as marked with $*$ in this figure, and let $\boldsymbol{c}$ be the valid grid cell on a neighboring panel that contains that point. The stencil set then consists of all the valid cells sharing a vertex with $\boldsymbol{c}$ and also the cells two away from $\boldsymbol{c}$ in both directions along both coordinate dimensions, making the appropriate transformation when crossing a panel boundary.

(1) From $\langle \boldsymbol{JU} \rangle$ on Ω and $\langle J \rangle$ on $\mathcal{G}_{1,1}(\Omega)$, obtain $\langle \boldsymbol{U} \rangle$ on Ω by using (B-32), with adjustments at panel boundaries as described in Appendix B4. We then have $\langle \boldsymbol{U} \rangle$ accurate to fourth order in $\Delta\alpha = \Delta\beta$.

(2) Interpolate $\langle \boldsymbol{U} \rangle$ from Ω to the ghost cells $\mathcal{G}_{3,3}(\Omega) - \Omega$ by the method of least squares from stencils in [31]. See Figure 2 for an illustration of interpolation stencils for two sample ghost cells.

Once we find the set of stencil cells for a particular ghost cell, we rotate the entire sphere so that the center of the ghost cell lies on the equator. Let λ and ϕ denote the latitudinal and longitudinal displacements, respectively, of any point from the ghost cell's new center on the equator. For each stencil cell indexed by $\boldsymbol{s}$, let $\lambda_{\boldsymbol{s}}$ and $\phi_{\boldsymbol{s}}$ be the latitudinal and longitudinal displacements of its center from the rotated ghost-cell center. Define the stencil's average angular distance

$$\bar{\theta} = \frac{1}{N} \sum_{\boldsymbol{s}} \sqrt{\lambda_{\boldsymbol{s}}^2 + \phi_{\boldsymbol{s}}^2}, \tag{38}$$

where N is the number of stencil cells.

For the scalar component $\langle h \rangle$ of $\langle \boldsymbol{U} \rangle$, we follow the procedure in [31], approximating h by a Taylor polynomial over latitude and longitude and finding its coefficients a_{pq} for $p, q \geq 0$ and $p + q \leq 3$ satisfying as closely as possible the overdetermined system of N equations

$$\sum_{p,q \geq 0;\, p+q \leq 3} a_{pq} \left\langle \left(\frac{\lambda}{\bar{\theta}} \right)^p \left(\frac{\phi}{\bar{\theta}} \right)^q \right\rangle_{\boldsymbol{s}} = \langle h \rangle_{\boldsymbol{s}} \tag{39}$$

for all stencil cells $\boldsymbol{s}$, where the notation $\langle\cdot\rangle_{\boldsymbol{s}}$ represents averaging over cell $\boldsymbol{s}$ and (λ, ϕ) ranges over its values in cell $\boldsymbol{s}$. The system is overdetermined because there are 10 coefficients a_{pq} for which to solve and the number of equations, N, is either 12 or 13. (It is 12 only if the ghost cell is near the intersection of three panels.) We then evaluate the Taylor polynomial averaged over the ghost cell $\boldsymbol{g}$:

$$\langle h\rangle_{\boldsymbol{g}} = \sum_{p,q\ge 0;\, p+q\le 3} a_{pq}\left\langle \left(\frac{\lambda}{\bar{\lambda}}\right)^p \left(\frac{\phi}{\bar{\phi}}\right)^q \right\rangle_{\boldsymbol{g}}. \tag{40}$$

The procedure above applies to the scalar component $\langle h\rangle$ of $\langle \boldsymbol{U}\rangle$, but $\langle \boldsymbol{U}\rangle$ also contains $\langle hu^\alpha\rangle$ and $\langle hu^\beta\rangle$, which are components in different bases in adjacent panels, so in order to find $\langle hu^\alpha\rangle$ and $\langle hu^\beta\rangle$ at the ghost cell, a basis transformation must be made.

At a point (λ, ϕ), let the basis-transformation matrix from a source panel $\mathcal{S}$, containing a stencil cell, to a destination panel $\mathcal{D}$, containing the ghost cell, be denoted

$$T_{\mathcal{S}\to\mathcal{D}}(\lambda,\phi) = \begin{pmatrix} T^{\alpha\alpha}_{\mathcal{S}\to\mathcal{D}}(\lambda,\phi) & T^{\alpha\beta}_{\mathcal{S}\to\mathcal{D}}(\lambda,\phi) \\ T^{\beta\alpha}_{\mathcal{S}\to\mathcal{D}}(\lambda,\phi) & T^{\beta\beta}_{\mathcal{S}\to\mathcal{D}}(\lambda,\phi) \end{pmatrix}.$$

Then our modification to (39)–(40) is to find coefficients b_{pq} and c_{pq} of two Taylor polynomials in the basis of the panel $\mathcal{P}(\boldsymbol{g})$ containing the ghost cell $\boldsymbol{g}$, satisfying as closely as possible the overdetermined system of $2N$ equations

$$\sum_{p,q\ge 0;\, p+q\le 3} b_{pq}\left\langle T^{\alpha\alpha}_{\mathcal{P}(\boldsymbol{s})\to\mathcal{P}(\boldsymbol{g})}(\lambda,\phi)\left(\frac{\lambda}{\bar{\lambda}}\right)^p\left(\frac{\phi}{\bar{\phi}}\right)^q\right\rangle_{\boldsymbol{s}} + \sum_{p,q\ge 0;\, p+q\le 3} c_{pq}\left\langle T^{\alpha\beta}_{\mathcal{P}(\boldsymbol{s})\to\mathcal{P}(\boldsymbol{g})}(\lambda,\phi)\left(\frac{\lambda}{\bar{\lambda}}\right)^p\left(\frac{\phi}{\bar{\phi}}\right)^q\right\rangle_{\boldsymbol{s}} = \langle hu^\alpha\rangle_{\boldsymbol{s}}, \tag{41}$$

$$\sum_{p,q\ge 0;\, p+q\le 3} b_{pq}\left\langle T^{\beta\alpha}_{\mathcal{P}(\boldsymbol{s})\to\mathcal{P}(\boldsymbol{g})}(\lambda,\phi)\left(\frac{\lambda}{\bar{\lambda}}\right)^p\left(\frac{\phi}{\bar{\phi}}\right)^q\right\rangle_{\boldsymbol{s}} + \sum_{p,q\ge 0;\, p+q\le 3} c_{pq}\left\langle T^{\beta\beta}_{\mathcal{P}(\boldsymbol{s})\to\mathcal{P}(\boldsymbol{g})}(\lambda,\phi)\left(\frac{\lambda}{\bar{\lambda}}\right)^p\left(\frac{\phi}{\bar{\phi}}\right)^q\right\rangle_{\boldsymbol{s}} = \langle hu^\beta\rangle_{\boldsymbol{s}} \tag{42}$$

for all stencil cells $\boldsymbol{s}$, where $\mathcal{P}(\boldsymbol{s})$ is the panel containing cell $\boldsymbol{s}$. Then we evaluate

$$\langle hu^\alpha\rangle_{\boldsymbol{g}} = \sum_{p,q\ge 0;\, p+q\le 3} b_{pq}\left\langle \left(\frac{\lambda}{\bar{\lambda}}\right)^p \left(\frac{\phi}{\bar{\phi}}\right)^q \right\rangle_{\boldsymbol{g}}, \tag{43}$$

$$\langle hu^\beta\rangle_{\boldsymbol{g}} = \sum_{p,q\ge 0;\, p+q\le 3} c_{pq}\left\langle \left(\frac{\lambda}{\bar{\lambda}}\right)^p \left(\frac{\phi}{\bar{\phi}}\right)^q \right\rangle_{\boldsymbol{g}}. \tag{44}$$

(3) On cells in $\mathcal{G}_{3,3}(\Omega)$, deconvolve from averages $\langle \boldsymbol{U} \rangle$ to $\boldsymbol{U}$ at centers by

$$\boldsymbol{U}_{i,j} = \langle \boldsymbol{U} \rangle_{i,j} - \tfrac{1}{24}(D_\alpha^{2c} \langle \boldsymbol{U} \rangle)_{i,j} - \tfrac{1}{24}(D_\beta^{2c} \langle \boldsymbol{U} \rangle)_{i,j} \quad \text{for } (i,j) \in \mathcal{G}_{2,2}(\Omega). \tag{45}$$

This formula is from (B-23) in Appendix B3 and is accurate to fourth order in $\Delta\alpha = \Delta\beta$.

(4) Obtain averages $\langle \boldsymbol{W} \rangle$ of primitive variables on $\mathcal{G}_{2,2}(\Omega)$ as follows. Set

$$\boldsymbol{W}_{i,j} = \boldsymbol{W}(\boldsymbol{U}_{i,j}) \qquad \text{for } (i,j) \in \mathcal{G}_{2,2}(\Omega), \tag{46}$$

$$\overline{\boldsymbol{W}}_{i,j} = \boldsymbol{W}(\langle \boldsymbol{U} \rangle_{i,j}) \quad \text{for } (i,j) \in \mathcal{G}_{3,3}(\Omega) \tag{47}$$

with $\boldsymbol{W}(\boldsymbol{U})$ being the pointwise function converting conserved variables to primitive variables. Then convolve:

$$\langle \boldsymbol{W} \rangle_{i,j} = \boldsymbol{W}_{i,j} + \tfrac{1}{24}(D_\alpha^{2c} \overline{\boldsymbol{W}})_{i,j} + \tfrac{1}{24}(D_\beta^{2c} \overline{\boldsymbol{W}})_{i,j} \quad \text{for } (i,j) \in \mathcal{G}_{2,2}(\Omega). \tag{48}$$

The result is accurate to fourth order in $\Delta\alpha = \Delta\beta$ because it uses (B-22) from Appendix B3, and $\overline{\boldsymbol{W}}_{i,j} - \boldsymbol{W}_{i,j}$ is second-order in $\Delta\alpha = \Delta\beta$. In (48), we apply the difference operators to $\overline{\boldsymbol{W}}$ instead of $\boldsymbol{W}$ to reduce the depth of ghost cells required, without dropping order.

(5) Interpolate $\langle \boldsymbol{W} \rangle$ from averages over grid cells and ghost cells to averages over faces, using the fourth-order-accurate formulae from [32]:

$$\langle \boldsymbol{W} \rangle_{i+\frac{1}{2},j} = \tfrac{7}{12}(\langle \boldsymbol{W} \rangle_{i,j} + \langle \boldsymbol{W} \rangle_{i+1,j}) - \tfrac{1}{12}(\langle \boldsymbol{W} \rangle_{i-1,j} + \langle \boldsymbol{W} \rangle_{i+2,j}) \quad \text{for } (i+\tfrac{1}{2}, j) \in \mathcal{F}^\alpha(\mathcal{G}_{0,1}(\Omega)), \tag{49}$$

$$\langle \boldsymbol{W} \rangle_{i,j+\frac{1}{2}} = \tfrac{7}{12}(\langle \boldsymbol{W} \rangle_{i,j} + \langle \boldsymbol{W} \rangle_{i,j+1}) - \tfrac{1}{12}(\langle \boldsymbol{W} \rangle_{i,j-1} + \langle \boldsymbol{W} \rangle_{i,j+2}) \quad \text{for } (i, j+\tfrac{1}{2}) \in \mathcal{F}^\beta(\mathcal{G}_{1,0}(\Omega)). \tag{50}$$

(6) Deconvolve from face-averaged $\langle \boldsymbol{W} \rangle$ to face-centered $\boldsymbol{W}$, using (B-27) to obtain $\boldsymbol{W}_{i+\frac{1}{2},j}$ for $(i+\frac{1}{2}, j) \in \mathcal{F}^\alpha(\Omega)$ and (B-29) to obtain $\boldsymbol{W}_{i,j+\frac{1}{2}}$ for $(i, j+\frac{1}{2}) \in \mathcal{F}^\beta(\Omega)$. These are fourth-order-accurate in $\Delta\alpha = \Delta\beta$.

(7) Set face-centered fluxes:

$$\boldsymbol{F}^\alpha_{i+\frac{1}{2},j} = \boldsymbol{F}(\boldsymbol{W}_{i+\frac{1}{2},j}) \qquad \text{for } (i+\tfrac{1}{2}, j) \in \mathcal{F}^\alpha(\Omega), \tag{51}$$

$$\overline{\boldsymbol{F}}^\alpha_{i+\frac{1}{2},j} = \boldsymbol{F}(\langle \boldsymbol{W} \rangle_{i+\frac{1}{2},j}) \quad \text{for } (i+\tfrac{1}{2}, j) \in \mathcal{F}^\alpha(\mathcal{G}_{0,1}(\Omega)), \tag{52}$$

$$\boldsymbol{F}^\beta_{i,j+\frac{1}{2}} = \boldsymbol{F}(\boldsymbol{W}_{i,j+\frac{1}{2}}) \qquad \text{for } (i, j+\tfrac{1}{2}) \in \mathcal{F}^\beta(\Omega), \tag{53}$$

$$\overline{\boldsymbol{F}}^\beta_{i,j+\frac{1}{2}} = \boldsymbol{F}(\langle \boldsymbol{W} \rangle_{i,j+\frac{1}{2}}) \quad \text{for } (i, j+\tfrac{1}{2}) \in \mathcal{F}^\beta(\mathcal{G}_{1,0}(\Omega)). \tag{54}$$

The difference $\boldsymbol{F}^\alpha_{i+\frac{1}{2},j} - \overline{\boldsymbol{F}}^\alpha_{i+\frac{1}{2},j}$ is second-order in $\Delta\alpha = \Delta\beta$ as is the difference $\boldsymbol{F}^\beta_{i,j+\frac{1}{2}} - \overline{\boldsymbol{F}}^\beta_{i,j+\frac{1}{2}}$.

(8) Convolve face-centered $\boldsymbol{F}^\alpha$ to obtain face averages $\langle \boldsymbol{F}^\alpha \rangle$ and convolve face-centered $\boldsymbol{F}^\beta$ to obtain face averages $\langle \boldsymbol{F}^\beta \rangle$ with the fourth-order accurate formulae

$$\langle \boldsymbol{F}^\alpha \rangle_{i+\frac{1}{2},j} = \boldsymbol{F}^\alpha_{i+\frac{1}{2},j} + \tfrac{1}{24}(D^{2\mathrm{f}}_\beta \bar{\boldsymbol{F}}^\alpha)_{i+\frac{1}{2},j} \quad \text{for } (i+\tfrac{1}{2}, j) \in \mathscr{F}^\alpha(\Omega), \tag{55}$$

$$\langle \boldsymbol{F}^\beta \rangle_{i,j+\frac{1}{2}} = \boldsymbol{F}^\beta_{i,j+\frac{1}{2}} + \tfrac{1}{24}(D^{2\mathrm{f}}_\alpha \bar{\boldsymbol{F}}^\beta)_{i,j+\frac{1}{2}} \quad \text{for } (i, j+\tfrac{1}{2}) \in \mathscr{F}^\beta(\Omega). \tag{56}$$

We take derivatives of $\bar{\boldsymbol{F}}$ instead of $\boldsymbol{F}$ in order to reduce the depth of ghost cells required. Since $\bar{\boldsymbol{F}}$ and $\boldsymbol{F}$ differ only by second order in $\Delta\alpha = \Delta\beta$, we see from (A-14) that including $\bar{\boldsymbol{F}}$ rather than $\boldsymbol{F}$ in (55)–(56) results in a difference in $\langle \boldsymbol{F}^\alpha \rangle_{i+\frac{1}{2},j}$ or $\langle \boldsymbol{F}^\beta \rangle_{i,j+\frac{1}{2}}$ that is fourth-order in $\Delta\alpha = \Delta\beta$.

(9) Add artificial dissipation: to smooth out oscillations due to the central difference operator, we add an artificial dissipation to the fluxes. The effect of this modification is a sixth-order diffusive operator, which retains the order of accuracy of the underlying scheme.

First set $v_{\max}$ to be the maximum wave speed over the whole domain, which for advection is the maximum of $r(|u^\alpha| + |u^\beta|)$ and for shallow-water equations is the maximum of $\sqrt{Gh} + r\max\{|u^\alpha|, |u^\beta|\}$, where h, u^α, and u^β are the components of $\boldsymbol{W}$. Then modify the fluxes with fifth undivided differences:

$$\langle \boldsymbol{F}^\alpha \rangle_{i+\frac{1}{2},j} = \langle \boldsymbol{F}^\alpha \rangle_{i+\frac{1}{2},j} - \gamma v_{\max}(D^{5\mathrm{f}}_\alpha \langle \boldsymbol{U} \rangle)_{i+\frac{1}{2},j} \quad \text{for } \mathscr{F}^\alpha(\Omega), \tag{57}$$

$$\langle \boldsymbol{F}^\beta \rangle_{i,j+\frac{1}{2}} = \langle \boldsymbol{F}^\beta \rangle_{i,j+\frac{1}{2}} - \gamma v_{\max}(D^{5\mathrm{f}}_\beta \langle \boldsymbol{U} \rangle)_{i,j+\frac{1}{2}} \quad \text{for } \mathscr{F}^\beta(\Omega), \tag{58}$$

where $\gamma = \frac{1}{128}$ for advection and $\gamma = \sqrt{2}/64$ for shallow-water equations. The coefficient γ has been chosen empirically so that the artificial dissipation is enough to smooth out oscillations but not so large as to detract from accuracy.

(10) Find the fourth-order convolution products

$$\langle J\boldsymbol{F}^\alpha \rangle_{i+\frac{1}{2},j} = \langle J \rangle_{i+\frac{1}{2},j} \langle \boldsymbol{F}^\alpha \rangle_{i+\frac{1}{2},j} + \tfrac{1}{12}(D^{1\mathrm{f}}_\beta \langle J \rangle)_{i+\frac{1}{2},j} (D^{1\mathrm{f}}_\beta \langle \bar{\boldsymbol{F}}^\alpha \rangle)_{i+\frac{1}{2},j} \quad \text{for } \mathscr{F}^\alpha(\Omega), \tag{59}$$

$$\langle J\boldsymbol{F}^\beta \rangle_{i,j+\frac{1}{2}} = \langle J \rangle_{i,j+\frac{1}{2}} \langle \boldsymbol{F}^\beta \rangle_{i,j+\frac{1}{2}} + \tfrac{1}{12}(D^{1\mathrm{f}}_\alpha \langle J \rangle)_{i,j+\frac{1}{2}} (D^{1\mathrm{f}}_\alpha \langle \bar{\boldsymbol{F}}^\beta \rangle)_{i,j+\frac{1}{2}} \quad \text{for } \mathscr{F}^\beta(\Omega). \tag{60}$$

We take differences of $\bar{\boldsymbol{F}}^\alpha$ and $\bar{\boldsymbol{F}}^\beta$ instead of $\langle \boldsymbol{F}^\alpha \rangle$ and $\langle \boldsymbol{F}^\beta \rangle$ in order to reduce the depth of ghost cells required, without dropping order. These approximations are fourth-order-accurate in $\Delta\alpha = \Delta\beta$.

(11) For each grid-cell face that is shared by two panels, after $\langle J\boldsymbol{F}^\alpha \rangle$ or $\langle J\boldsymbol{F}^\beta \rangle$ is computed on that face separately for each panel in Step (10), replace it by its average with the corresponding $\langle J\boldsymbol{F}^\alpha \rangle$ or $\langle J\boldsymbol{F}^\beta \rangle$ calculated on the same face in

the other panel that shares it. Note that $\langle J\boldsymbol{F}^{\alpha}\rangle$ or $\langle J\boldsymbol{F}^{\beta}\rangle$ from the other panel may need to be reoriented as follows:

- Faces that are shared with equatorial panels 2 or 4 and either of the polar panels, 5 or 6, have constant β on the equatorial panel and constant α on the polar panel; hence, on these faces, $\langle J\boldsymbol{F}^{\beta}\rangle$ on the equatorial panel is averaged with $\langle J\boldsymbol{F}^{\alpha}\rangle$ on the polar panel.
- Before averaging, sign changes are required for faces on the other panel along the following interfaces: $\langle J\boldsymbol{F}^{\beta}\rangle$ on Panel 2 with $\langle J\boldsymbol{F}^{\alpha}\rangle$ on Panel 5, $\langle J\boldsymbol{F}^{\beta}\rangle$ on Panel 4 with $\langle J\boldsymbol{F}^{\alpha}\rangle$ on Panel 6, and $\langle J\boldsymbol{F}^{\beta}\rangle$ on Panel 3 with $\langle J\boldsymbol{F}^{\beta}\rangle$ on either of the polar panels, 5 or 6.

For the vector fluxes, $\mathcal{T}^{\alpha k}$ and $\mathcal{T}^{\beta k}$, this is more complicated because the components are in different bases in different panels. Write

$$\Phi^{\alpha} = J\begin{pmatrix}\mathcal{T}^{\alpha\alpha}\\ \mathcal{T}^{\beta\alpha}\end{pmatrix} \quad \text{on faces of constant } \alpha, \tag{61}$$

$$\Phi^{\beta} = J\begin{pmatrix}\mathcal{T}^{\alpha\beta}\\ \mathcal{T}^{\beta\beta}\end{pmatrix} \quad \text{on faces of constant } \beta. \tag{62}$$

Then we set the following from (52), (54), and (59)–(60):

- $\langle\Phi^{\alpha}\rangle_{i+\frac{1}{2},j}$, vector components of $\langle J\boldsymbol{F}^{\alpha}\rangle_{i+\frac{1}{2},j}$, for $(i+\frac{1}{2}, j) \in \mathcal{F}^{\alpha}(\Omega)$,
- $\overline{\Phi}^{\alpha}_{i+\frac{1}{2},j}$, vector components of $J_{i+\frac{1}{2},j}\overline{\boldsymbol{F}}^{\alpha}_{i+\frac{1}{2},j}$, for $(i+\frac{1}{2}, j) \in \mathcal{F}^{\alpha}(\mathcal{G}_{0,1}(\Omega))$,
- $\langle\Phi^{\beta}\rangle_{i,j+\frac{1}{2}}$, vector components of $\langle J\boldsymbol{F}^{\beta}\rangle_{i,j+\frac{1}{2}}$, for $(i, j+\frac{1}{2}) \in \mathcal{F}^{\beta}(\Omega)$,
- $\overline{\Phi}^{\beta}_{i,j+\frac{1}{2}}$, vector components of $J_{i,j+\frac{1}{2}}\overline{\boldsymbol{F}}^{\beta}_{i,j+\frac{1}{2}}$, for $(i, j+\frac{1}{2}) \in \mathcal{F}^{\beta}(\mathcal{G}_{1,0}(\Omega))$.

We deconvolve to face centers

$$\Phi^{\alpha}_{i+\frac{1}{2},j} = \langle\Phi^{\alpha}\rangle_{i+\frac{1}{2},j} - \tfrac{1}{24}(D^{2\mathrm{f}}_{\beta}\overline{\Phi}^{\alpha})_{i+\frac{1}{2},j} \quad \text{for } (i+\tfrac{1}{2}, j) \in \mathcal{F}^{\alpha}(\Omega), \tag{63}$$

$$\Phi^{\beta}_{i,j+\frac{1}{2}} = \langle\Phi^{\beta}\rangle_{i,j+\frac{1}{2}} - \tfrac{1}{24}(D^{2\mathrm{f}}_{\alpha}\overline{\Phi}^{\beta})_{i,j+\frac{1}{2}} \quad \text{for } (i, j+\tfrac{1}{2}) \in \mathcal{F}^{\beta}(\Omega) \tag{64}$$

and convert to the orthonormal frame with orthonormalization matrices $\mathbb{O}^{\alpha}_{i+\frac{1}{2},j}$ and $\mathbb{O}^{\beta}_{i,j+\frac{1}{2}}$ (see [56]) at face centers:

$$\Theta^{\alpha}_{i+\frac{1}{2},j} = \mathbb{O}^{\alpha}_{i+\frac{1}{2},j}\Phi^{\alpha}_{i+\frac{1}{2},j} \quad \text{for } (i+\tfrac{1}{2}, j) \in \mathcal{F}^{\alpha}(\Omega), \tag{65}$$

$$\Theta^{\beta}_{i,j+\frac{1}{2}} = \mathbb{O}^{\beta}_{i,j+\frac{1}{2}}\Phi^{\beta}_{i,j+\frac{1}{2}} \quad \text{for } (i, j+\tfrac{1}{2}) \in \mathcal{F}^{\beta}(\Omega), \tag{66}$$

$$\overline{\Theta}^{\alpha}_{i+\frac{1}{2},j} = \mathbb{O}^{\alpha}_{i+\frac{1}{2},j}\overline{\Phi}^{\alpha}_{i+\frac{1}{2},j} \quad \text{for } (i+\tfrac{1}{2}, j) \in \mathcal{F}^{\alpha}(\mathcal{G}_{0,1}(\Omega)), \tag{67}$$

$$\overline{\Theta}^{\beta}_{i,j+\frac{1}{2}} = \mathbb{O}^{\beta}_{i,j+\frac{1}{2}}\overline{\Phi}^{\beta}_{i,j+\frac{1}{2}} \quad \text{for } (i, j+\tfrac{1}{2}) \in \mathcal{F}^{\beta}(\mathcal{G}_{1,0}(\Omega)). \tag{68}$$

On each face of a panel boundary, we replace each of Θ^α and $\overline{\Theta}^\alpha$, or each of Θ^β and $\overline{\Theta}^\beta$, with the averages from the two panels sharing that face. In the case of faces shared by either Panel 2 or 4 and either Panel 5 or 6, we flip the sign of the quantity from the opposite panel before averaging.

Finally, we set the vector components of $\langle J\boldsymbol{F}^\alpha\rangle_{i+\frac{1}{2},j}$ and $\langle J\boldsymbol{F}^\beta\rangle_{i,j+\frac{1}{2}}$ to

$$\langle\widetilde{\Phi}^\alpha\rangle_{i+\frac{1}{2},j} = (\mathbb{O}^\alpha_{i+\frac{1}{2},j})^{-1}\Theta^\alpha_{i+\frac{1}{2},j} + \tfrac{1}{24}(D^{2\mathrm{f}}_\beta((\mathbb{O}^\alpha)^{-1}\overline{\Theta}^\alpha))_{i+\frac{1}{2},j} \quad \text{for } (i+\tfrac{1}{2}, j) \in \mathscr{F}^\alpha(\Omega), \quad (69)$$

$$\langle\widetilde{\Phi}^\beta\rangle_{i,j+\frac{1}{2}} = (\mathbb{O}^\beta_{i,j+\frac{1}{2}})^{-1}\Theta^\beta_{i,j+\frac{1}{2}} + \tfrac{1}{24}(D^{2\mathrm{f}}_\beta((\mathbb{O}^\beta)^{-1}\overline{\Theta}^\beta))_{i,j+\frac{1}{2}} \quad \text{for } (i, j+\tfrac{1}{2}) \in \mathscr{F}^\beta(\Omega). \quad (70)$$

Now for $(i, j) \in \Omega$, we have fourth-order-accurate $\langle J\boldsymbol{F}^\alpha\rangle_{i\pm\frac{1}{2},j}$ and $\langle J\boldsymbol{F}^\beta\rangle_{i,j\pm\frac{1}{2}}$ on the right-hand side of (30), the evolution equation for $\langle J\boldsymbol{U}\rangle_{i,j}$.

The source term $\langle J\boldsymbol{\Psi}\rangle_{i,j}$ in (30) is computed as follows. From (46), we have $\boldsymbol{W}_{i,j}$ on centers of grid cells $(i, j) \in \mathscr{G}_{2,2}(\Omega)$. Since $\boldsymbol{\Psi}$ is a function of $\boldsymbol{W}$, we can find $\boldsymbol{\Psi}_{i,j}$ for $(i, j) \in \mathscr{G}_{1,1}(\Omega)$, multiply it by $J_{i,j}$, and apply the convolution formula (B-22) to find the averaged $\langle J\boldsymbol{\Psi}\rangle_{i,j}$ for $(i, j) \in \Omega$ to fourth-order accuracy.

4. Adaptive mesh refinement

With adaptive mesh refinement (AMR), we extend the approach of [19] on single-block mapped grids to the mapped-multiblock grids of the cubed sphere. What makes the cubed sphere different from single-block mapped grids is that the solution is on a manifold, we are able to use analytic formulae for integrals of $\langle J\rangle$, and adjacent panels have different mappings.

To implement adaptive mesh refinement, we make use of the Chombo library for parallel AMR [1] and follow the strategies used therein. Adaptive-mesh-refinement calculations are performed on a hierarchy of nested meshes $\Omega^\ell \subset \Gamma^\ell$, with $\Omega^\ell \supset \mathscr{C}_{n^\ell_{\mathrm{ref}}}(\Omega^{\ell+1})$ where n^ℓ_{ref} denotes the refinement ratio between levels ℓ and $\ell+1$ and $\mathscr{C}_{n^\ell_{\mathrm{ref}}}$ denotes coarsening by this ratio. At level ℓ, we label all cells inside Ω^ℓ as being valid and all cells outside Ω^ℓ (such as ghost cells) as being invalid. Typically, Ω^ℓ is decomposed into a disjoint union of rectangles in order to perform calculations efficiently. We assume that there are a sufficient number of cells on level ℓ separating the level-$(\ell+1)$ cells from the level-$(\ell-1)$ cells such that interpolations to fill invalid ghost cells on finer levels can be independently performed. We will refer to grid hierarchies that meet this condition as being *properly nested*.

The top-level procedure for advancing level ℓ from time t^ℓ by a time step of length Δt^ℓ is shown in Figure 3.

Advance$(\ell, t^\ell, \Delta t^\ell)$:

(1) Regrid levels finer than ℓ if required (see Section 4.1).

(2) Advance level ℓ using the methods described in Section 3 with a Runge–Kutta time-stepping method.

(3) Interpolate to the invalid ghost cells surrounding level $\ell+1$ (see Section 4.2). A least-squares algorithm is used to compute the interpolating polynomial in each coarse cell. The interpolation need not be conservative because the resulting values in the ghost cells are only used to reconstruct the flux on the faces of the valid cells.

(4) Start level $\ell+1$ at Step (1). Level $\ell+1$ is refined in time (subcycled) with a time step $\Delta t^{\ell+1} = \Delta t^\ell / n^\ell_{\text{ref}}$.

(5) Average the solution from level $\ell+1$ and correct fluxes at coarse-fine interfaces to ensure conservation.

Figure 3. Pseudocode for advancing level ℓ from time t^ℓ to time $t^\ell + \Delta t^\ell$.

4.1. ***Regridding.*** Periodically, it is necessary to change the grid hierarchy in response to changes in the solution. During a regrid, we generate a new grid hierarchy, $\{\Omega^{\ell,\text{new}}\}_{\ell=\ell_{\text{base}}+1,\ldots,\ell_{\max}}$ leaving the mesh at ℓ_{base} and all coarser levels unchanged.

For $\ell = \ell_{\text{base}}, \ldots, \ell^{\text{new}}_{\max} - 1$, we use a least-squares algorithm to interpolate ghost values. For each ghost cell $V_{i,j}$, let $\mathcal{I}(i,j)$ denote the set of grid cells of its interpolation stencil. We solve a least-squares system for the coefficients $a^{i,j}_{p,q}$ of a polynomial interpolant of $\boldsymbol{U}$,

$$\sum_{p\geq 0; q\geq 0; p+q\leq 3} a^{i,j}_{p,q} \langle \alpha^p \beta^q \rangle_{i',j'} = \langle \boldsymbol{U} \rangle_{i',j'} \quad \text{for all } (i',j') \in \mathcal{I}(i,j) \tag{71}$$

(where α^p and β^q indicate powers of α and β), subject to a conservation constraint on $J\boldsymbol{U}$,

$$\sum_{(i',j')\in \mathcal{C}^{-1}(\{(i,j)\})} \; \sum_{p\geq 0; q\geq 0; p+q\leq 3} a^{i,j}_{p,q} \langle J\alpha^p \beta^q \rangle_{i',j'} = \langle J\boldsymbol{U} \rangle_{i,j}. \tag{72}$$

The moments $\langle \alpha^p \beta^q \rangle$ can be determined analytically, and the $\langle J\alpha^p \beta^q \rangle$ are computed using the product formula. Given this interpolant, we can construct $\langle J\boldsymbol{U} \rangle$ on the grid cells at level $\ell+1$ within $V_{i,j}$:

$$\langle J\boldsymbol{U} \rangle_{i',j'} = \sum_{p\geq 0; q\geq 0; p+q\leq 3} a^{i,j}_{p,q} \langle J\alpha^p \beta^q \rangle_{i',j'} \quad \text{for all } (i',j') \in \mathcal{C}^{-1}(\{(i,j)\}). \tag{73}$$

This interpolation is conservative.

4.2. ***Interpolating to ghost cells at next finer level.*** As shown in Section 3.4, advancing one time step by the method of Section 3 requires three layers of ghost cells. In Step (3) of the algorithm of Figure 3, we must interpolate $\langle J\boldsymbol{U}\rangle$ from level ℓ to the ghost cells of level $\ell+1$. In particular, after Step (2) of **Advance**$(\ell, t^\ell, \Delta t^\ell)$ advances the solution at level ℓ from time t^ℓ to time $t^\ell+\Delta t^\ell$, Step (3) interpolates the level-ℓ solution to ghost cells of level $\ell+1$ at times $t^\ell+s\Delta t^{\ell+1}$ for $s=0,\ldots,n^\ell_{\text{ref}}-1$, where $\Delta t^{\ell+1}=\Delta t^\ell/n^\ell_{\text{ref}}$ is the length of the time step at level $\ell+1$. Step (3) has the following substeps:

(a) Interpolate $\langle J\boldsymbol{U}\rangle$ on grid cells of level ℓ to the same grid cells at the intermediate times $t^\ell+s\Delta t^{\ell+1}$ for $s=1,\ldots,n^\ell_{\text{ref}}-1$. This temporal interpolation uses initial $\langle J\boldsymbol{U}\rangle^{(0)}=\langle J\boldsymbol{U}\rangle(t^\ell)$ and k_1, k_2, k_3, and k_4 in the Runge–Kutta method defined in (33)–(36) in Section 3.3. As derived in [20], for $0\le\chi\le1$, we have

$$\langle J\boldsymbol{U}\rangle(t^\ell+\chi\Delta t^\ell)=\langle J\boldsymbol{U}\rangle(t^\ell)+\chi k_1+\tfrac{1}{2}\chi^2(-3k_1+2k_2+2k_3-k_4)+\tfrac{2}{3}\chi^3(k_1-k_2-k_3+k_4)+O((\Delta t^\ell)^4). \quad (74)$$

(b) At each of the times $t^\ell+s\Delta t^{\ell+1}$ for $s=0,\ldots,n^\ell_{\text{ref}}-1$, fill in $\lceil(L+2)/n^\ell_{\text{ref}}\rceil$ layers of extrapanel ghost cells of $\langle J\boldsymbol{U}\rangle$ at level ℓ, by the method of least squares using interpolation stencils, described in Step (2) of Section 3.4.

(c) Fill in ghost cells of level $\ell+1$, by least-squares interpolation from the valid cells and ghost cells at level ℓ.

The temporal interpolation in Step (a) is the same as in [32]. With error of $O((\Delta t^\ell)^4)$, this interpolation preserves the order of the Runge–Kutta temporal discretization of Section 3.3. The spatial interpolation of Steps (b)–(c) is also fourth-order in the grid spacing.

5. Numerical tests

The Courant–Friedrichs–Lewy (CFL) number is

$$\frac{\Delta t}{\Delta\alpha}c_{\max}, \quad (75)$$

where Δt is the time step and $c_{\max}$ is the maximum wave speed.

As shown in [10], the stability constraint for the classical Runge–Kutta method we use is that the CFL number satisfy

$$\frac{\Delta t}{\Delta\alpha}c_{\max}\lesssim 2.06. \quad (76)$$

For advection, $c_{\max}$ is the maximum over the domain of $r(|u^\alpha|+|u^\beta|)$. For shallow-water equations, $c_{\max}$ is the maximum over the domain of the characteristic velocity $2\sqrt{Gh}+r(|u^\alpha|+|u^\beta|)$.

We note that the results presented here are for a method that does not employ any limiters or nonlinear filters that would suppress oscillations at discontinuities. We have constructed limiters for the Cartesian versions of the method in [32; 6]. While the extension of the approach used in that work to the present setting is straightforward, we have chosen not to apply it here, in order to obtain a clean assessment of the properties of the basic high-order method. There is a separate issue regarding positivity preservation, which historically has been an additional goal in the design of limiters. Our thinking on this issue is that the use of limiters for positivity preservation is an excessive constraint on the design choices in the method. Typically, a limiter can be thought of as a nonlinear hybridization of low- and high-order fluxes. To obtain a positivity-preserving limiter, it is a necessary condition for the low-order method to be positivity-preserving. For the case of advection, it is easy to construct a combination of a discretely divergence-free velocity field and a density distribution such that the only positivity-preserving field is donor-cell plus an explicit diffusion, which has a CFL time-step constraint that scales with the inverse of the dimensionality of the problem. Such a time-step constraint is stricter than that of the high-order methods of the type described here, even in 3D. For that reason, we are pursuing a different approach to positivity preservation based on redistribution of mass as a postprocessing step at the end of each time step [22]. Such an approach greatly expands the design space of limiter-based methods; for a discussion, see [6].

5.1. ***Deformational flow.*** To test the performance of the model under horizontal tracer transport, the deformational flow test [34, Test 4] is employed. This test is significantly more challenging than the solid-body rotation test of [61] since it not only tests divergent-free advection but also includes deformational stretching and the formation of thin filaments in the tracer field followed by subsequent recovery of the original profile. To obtain an analytical reference solution, the deformational-flow test reverses the time-varying flow field after half the total simulation period. The availability of an analytical reference solution at the final time means that error norms can be easily computed. Further, the addition of a solid-body rotation component to the flow field prevents the possible cancellation of errors when the flow is reversed.

In the transport equation (20) for h, the longitudinal component u_λ and latitudinal component u_ϕ of the flow field $\boldsymbol{u}$ take the form

$$u_\lambda = k \sin^2(\lambda') \sin(2\phi) \cos\left(\frac{\pi t}{T}\right) + \frac{2\pi}{T} \cos\phi, \tag{77}$$

$$u_\phi = k \sin(2\lambda') \cos\phi \cos\left(\frac{\pi t}{T}\right), \tag{78}$$

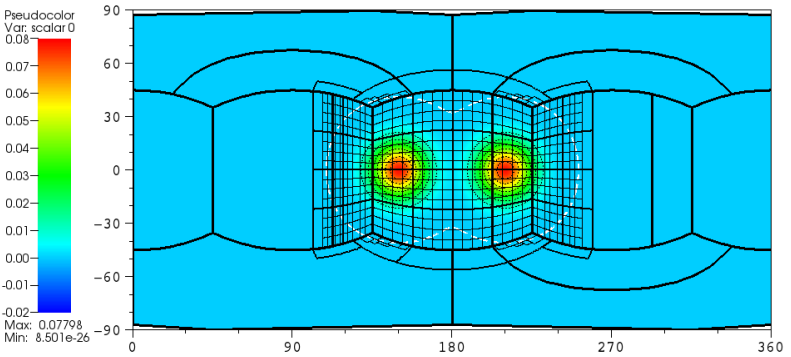


Figure 4. Plot of h at initial time in the deformational-flow test example of Section 5.1, with grids of resolutions c32/c128/c512. At this time, there is 34.4% c128 coverage and 27.9% c512 coverage. A dashed white contour line is drawn for h at the common refinement threshold of 5×10^{-5}, and dotted black contour lines are drawn at values of the positive tick marks in the legend.

where $\lambda' = \lambda - 2\pi t/T$, $k = 2$, $T = 5$ days, and $k = 2$. The height field consists of two superimposed smooth 2D Gaussian surfaces,

$$h(\lambda, \phi) = \sum_{i \in \{1,2\}} h_i(\lambda, \phi), \tag{79}$$

$$h_i(\lambda, \phi) = h_{\max} \exp\{-b_0 \delta_{\mathrm{xyz}}(\lambda, \phi; \lambda_i, \phi_i)\}, \tag{80}$$

where $i \in \{1, 2\}$, $h_{\max} = 1$, $b_0 = 10$, and $\delta_{\mathrm{xyz}}(\lambda, \phi; \lambda_i, \phi_i)$ is the 3D absolute Cartesian distance between (λ, ϕ) and (λ_i, ϕ_i) on the unit sphere,

$$\delta_{\mathrm{xyz}}(\lambda, \phi; \lambda_i, \phi_i) = \big[(\cos\phi \cos\lambda - \cos\phi_i \cos\lambda_i)^2 + (\cos\phi \sin\lambda - \cos\phi_i \sin\lambda_i)^2 + (\sin\phi - \sin\phi_i)^2\big]^{1/2}. \tag{81}$$

The centers of the Gaussian surfaces are located at $(\lambda_1, \phi_1) = (5\pi/6, 0)$ and $(\lambda_2, \phi_2) = (7\pi/6, 0)$. Although [34] has the setting $b_0 = 5$, here we instead set $b_0 = 10$ to narrow the width of the Gaussian surfaces, in order to highlight the benefits of AMR.

We run this example with the following resolutions:

- uniform resolution, with N_c a power of 2 from 32 through 1024,
- on two levels, the coarser level N_c a power of 2 from 32 through 256 and the finer level consisting of grids that are a factor of 4 finer and are located in regions where $|h| \geq 8 \times 10^{-4}/(N_c/64)^4$, and
- on three levels, the coarsest level N_c either 32 or 64, the middle level consisting of grids that are a factor of 4 finer and are located in regions where $|h| \geq 8 \times 10^{-4}/(N_c/16)^4$, and the finest level consisting of grids that are a factor of 4 finer than the middle-level grids and are located in the same regions.

Finest resolution	Uniform resolution max error	rate	Two levels max error	rate	Three levels max error	rate
c32	4.003×10^{-2}					
		0.88				
c64	2.162×10^{-2}					
		1.73				
c128	6.527×10^{-3}		6.544×10^{-3}			
		3.33		3.33		
c256	6.507×10^{-4}		6.506×10^{-4}			
		3.97		3.97		
c512	4.150×10^{-5}		4.150×10^{-5}		4.150×10^{-5}	
		4.00		4.00		4.00
c1024	2.586×10^{-6}		2.586×10^{-6}		2.586×10^{-6}	

Table 2. Maximum solution error at the final time, and convergence rates, for the deformational-flow test example of Section 5.1. When there is more than one level, the refinement ratio between consecutive levels is set to 4. Hence, in the two-level runs with results given here, where the finer levels are c128 through c1024, the coarser level is c32 through c256. Of the three-level runs, the first one has the refinements of the levels as c32/c128/c512 and the second has c64/256/c1024.

Figure 4 shows a plot of h at the initial time. The refinement thresholds have been selected to be comparable to the predicted asymptotically fourth-order solution error. We pick time step $\Delta t = 0.4\,\text{day}/N_c$, and we find $c_{\max} = 5.99\,\text{rad/day}$, so the CFL number from (75) is 1.53.

Table 2 shows the maximum solution error for each of the different runs. This table also shows the convergence rate of the maximum solution error, computed from two successively finer resolutions: since each successive resolution is refined by a factor of 2, this rate is the base-2 logarithm of the ratio of the errors. We see that the solution error converges to fourth order, and the error in each multilevel run is as good as that in the single-level run with the resolution of the finest level, with the level refinement criteria we use. Since the refinement criteria are such that finer grids are added where h is above a certain threshold, this example is not necessarily good for showing convergence at refinement boundaries, and so in Section 5.4, we show results of an example with fixed grids.

For the three simulations of deformational flow with coarsest level c32, Figure 5 shows plots of the error in h at the final time, where the maximum errors are the numbers shown in the first rows of the columns of Table 2. For these same three simulations, Figure 6 shows plots of h at the midpoint in time.

Figure 7 shows the fraction of the domains covered by finer-level grids during the multilevel simulations. Owing to the pattern of deformational flow, domain coverage of refined levels is highest near the midpoint in time and, in our runs, reaches its maximum of 68.2% for coverage of c128 in the c32/c128/c512 run. Because the refinement thresholds are equal, the coverage of c512 is almost the same in the c128/c512 and c32/c128/c512 runs. For the same reason, the coverage of c1024 is almost the same in the c256/c1024 and c64/c256/c1024 runs.

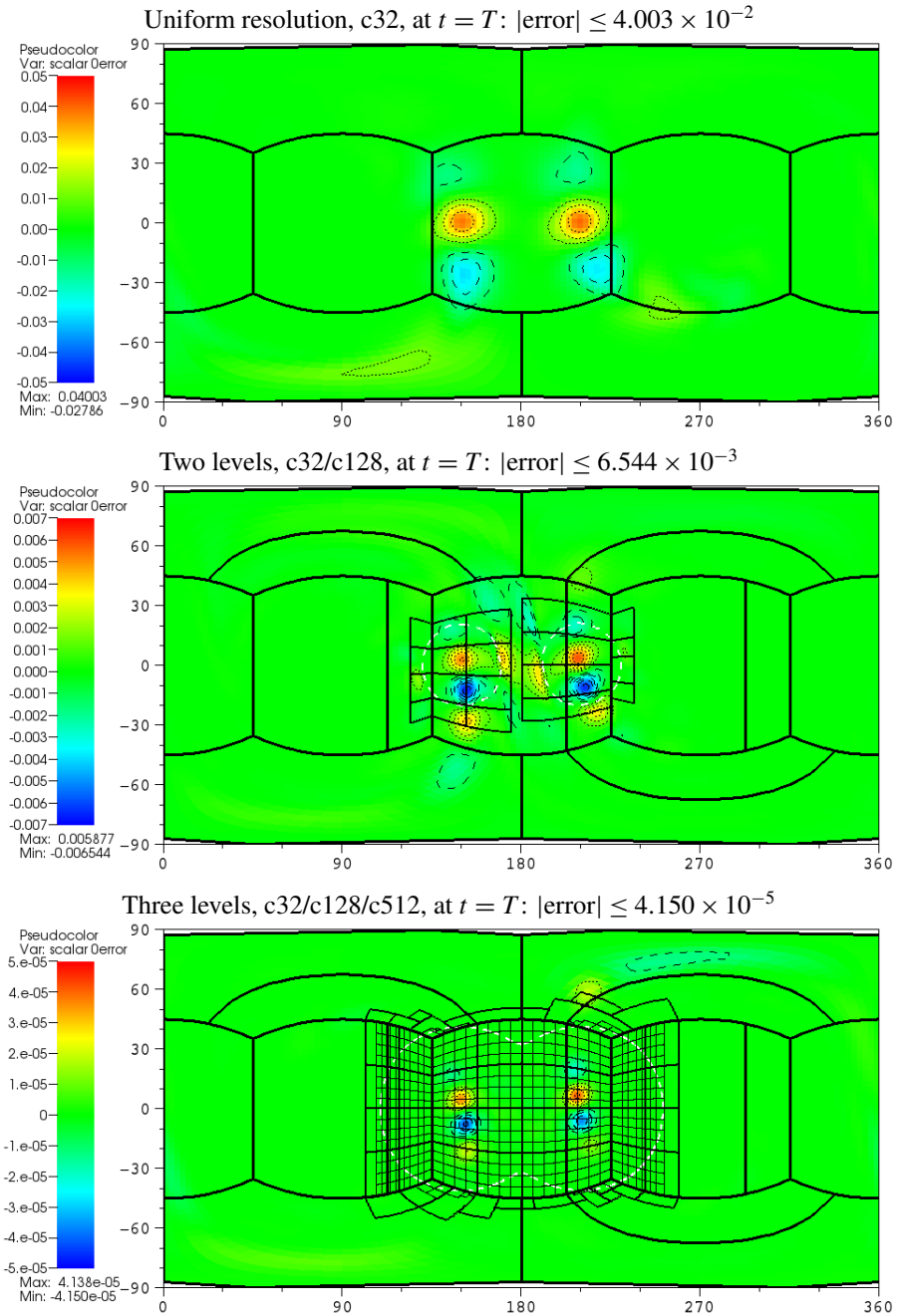


Figure 5. Plots of the error in h at the final time in the deformational-flow test example of Section 5.1, with c32 at the coarsest level. Grids at all levels at this time are shown. Black contour lines are drawn at values of the tick marks in the legend: dotted for positive and dashed for negative. For the two-level and three-level runs, dashed white contour lines are drawn at the refinement threshold for the calculated h at this time.

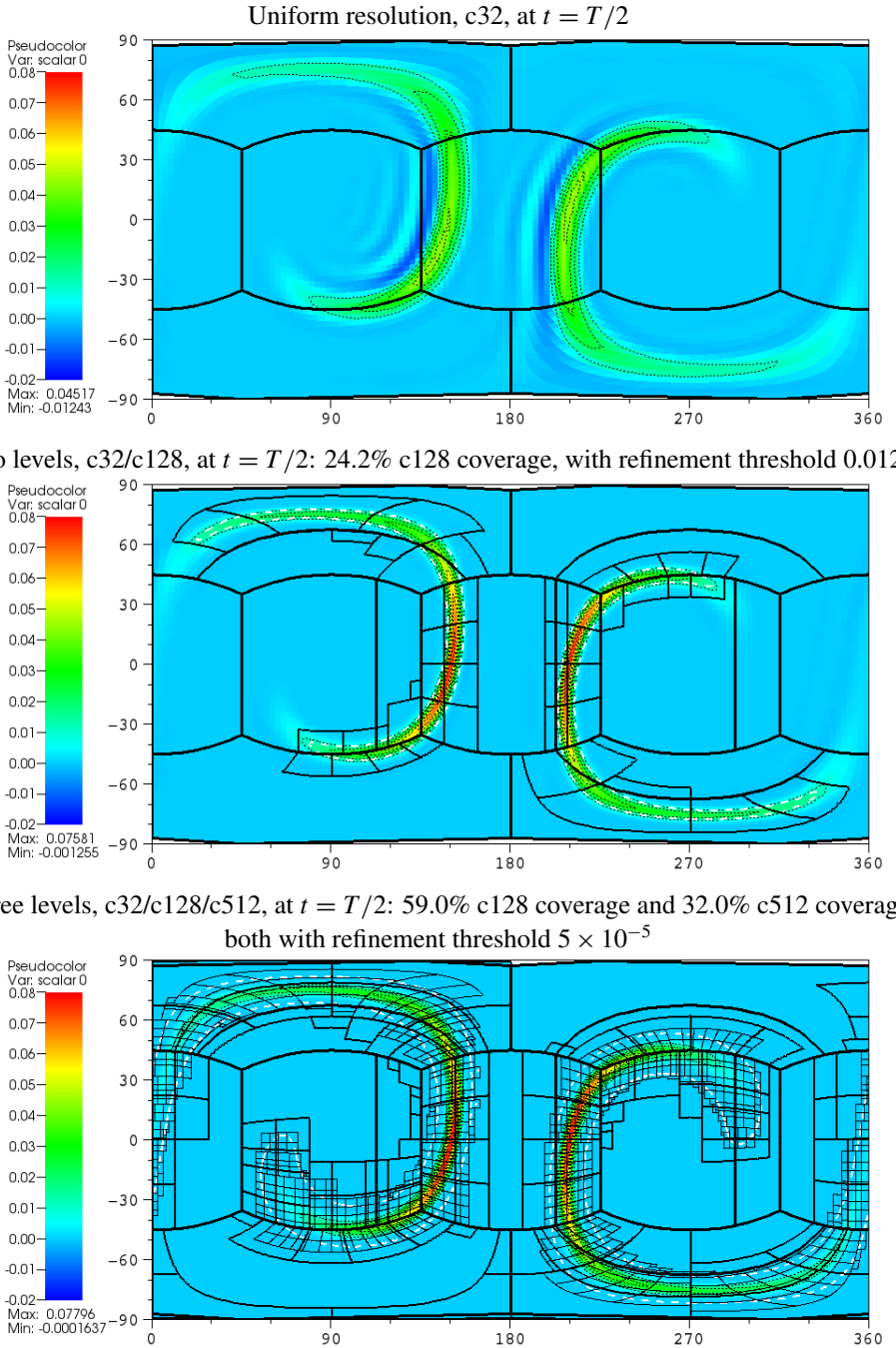


Figure 6. Plot of h at the midpoint in time, $t = T/2$, in the deformational-flow test example of Section 5.1, with c32 at the coarsest level. Grids at all levels at this time are shown. Dotted black contour lines are drawn at values of the positive tick marks in the legend, and in the two multilevel runs, dashed white contour lines are drawn at the refinement threshold.

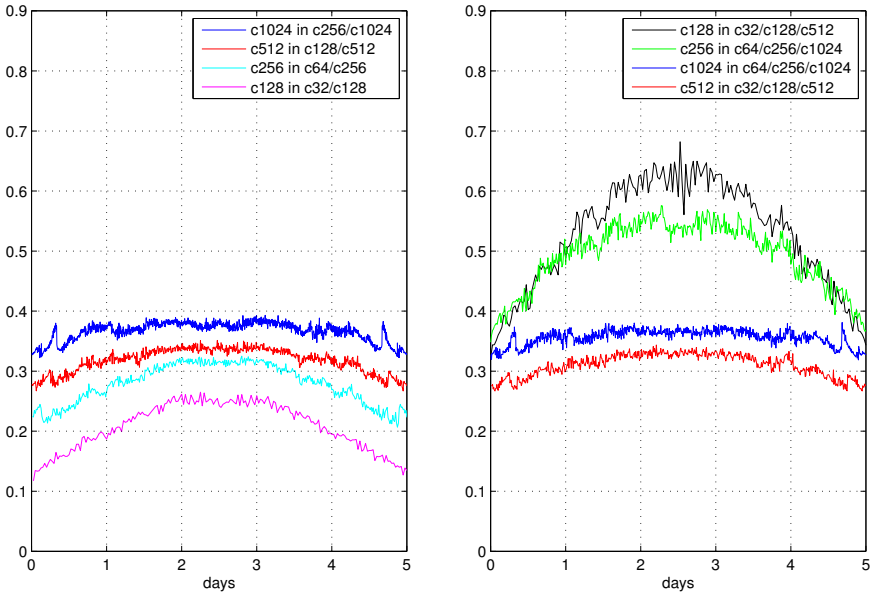


Figure 7. Plot of domain coverage of finer levels over time in the deformational-flow test of Section 5.1. Left: coverage of the finer level in two-level runs. Coverage increases with greater resolution because the refinement threshold is proportional to the fourth power of the grid spacing at the coarser level. Right: coverage of the middle and finest levels in three-level runs. As indicated by the red and dark-blue curves, coverage of the finest level in each three-level run matches coverage of the finer level in the two-level run with the same finest-level resolution because the refinement threshold is the same. In each three-level run, coverage of the middle level (black and green curves) is necessarily higher than coverage of the finest level (dark-blue and red curves) because proper-nesting conditions must be maintained. The gap between each three-level run's middle-level and finest-level coverage shrinks as resolution increases because proper-nesting conditions are expressed in terms of number of grid cells and grid cells become smaller with finer resolution.

5.2. ***Barotropically unstable jet without initial perturbation.*** The barotropic-instability test case of [15] consists of a zonal jet with compact support at a latitude of 45°. As in [24; 60], we first show the results of this test *without* the initial height perturbation that initiates the instability because we can check the order of accuracy of our method by comparing with the exact steady-state solution.

We pick time step $\Delta t = 0.25\,\text{day}/N_c$, and we find $c_{\max} = 10.1\text{rad/day}$, so the CFL number from (75) is 1.61. We run this example up to day 5 with the following resolutions:

- uniform resolution, with N_c a power of 2, from 16 through 1024,
- on two levels, the coarser level N_c a power of 2 from 16 through 256 and the finer level consisting of grids that are a factor of 4 finer and are located in regions where relative vorticity exceeds $0.32/\pi\ \text{day}^{-1} = 0.102\,\text{day}^{-1}$, and

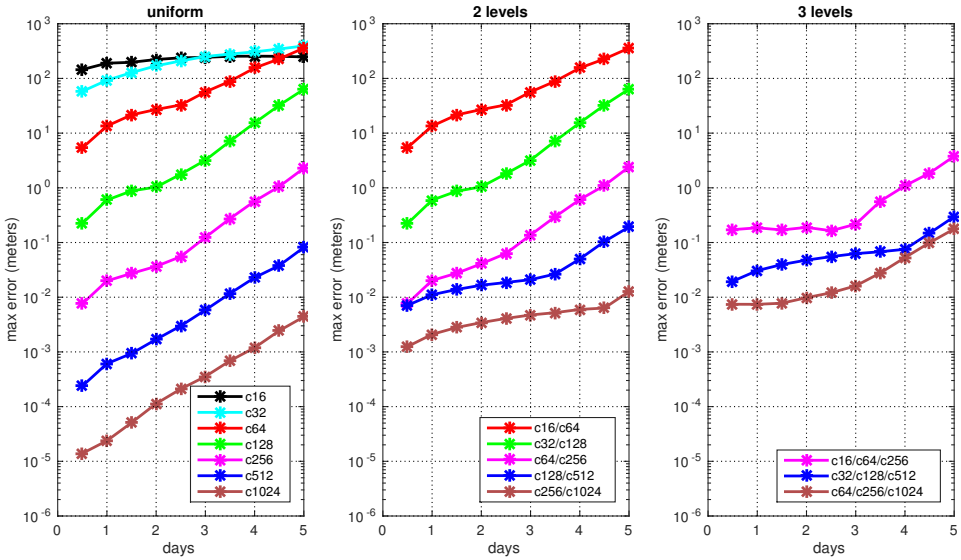


Figure 8. Plots of maximum error in height for the example in Section 5.2 of the steady-state (but unstable) jet of [15] without the initial perturbation, shown at intervals of every half day. Plots for runs with the same *finest*-level resolution have the same color.

- on three levels, the coarsest level N_c a power of 2 from 16 through 64, the middle level consisting of grids that are a factor of 4 finer and are located in regions where relative vorticity exceeds $0.32/\pi$ day$^{-1} = 0.102$ day^{-1}, and the finest level consisting of grids that are a factor of 4 finer than the middle-level grids and are located in regions where relative vorticity exceeds $1.28/\pi$ day$^{-1} = 0.407$ day^{-1}.

Figure 8 shows the maximum error in height for this example. We find that on uniform grids (left plot), the error is approximately fourth-order in the spatial resolution for c128 and finer; at coarser resolutions, the barotropic jet is not resolved, leading to a loss of convergence. For the two-level runs (center plot), the curves of maximum error over time match those of the finer level with uniform resolution for c16/c64, c32/c128, and c64/c256; but with more grid resolution, the two-level error is higher because the refinement threshold is too high to resolve it. For the three-level runs (right plot), the maximum error for c16/c64/c256 is a little higher than that for c64/c256 after day 3, and the maximum error for c32/c128/c512 is a little higher than that for c128/c512 after day 4, but the maximum errors at earlier times are higher because of the refinement threshold.

5.3. ***Barotropic instability.*** In the barotropic-instability test case of [15], a small height perturbation is added atop the jet, which leads to the controlled formation of an instability in the flow. The relative vorticity of the flow field at day 6 can then

be visually compared against a high-resolution numerically computed solution [15; 49]. For comparison, we use the simulation without additional explicit diffusion since the additional diffusion suggested in [15] leads to a significantly different flow field.

As in Section 5.2, we pick time step $\Delta t = 0.25\,\text{day}/N_c$. We now find $c_{\max} = 10.4\,\text{rad/day}$, so the CFL number from (75) is 1.66. We run this example up to day 6 with the same resolutions and refinement criteria as in Section 5.2. In the absence of an exact solution, we compare with the uniform c1024 solution as a reference.

Figure 9 shows the relative vorticity field at the final time for uniform c32, two-level c32/c128, and three-level c32/c128/c512. As shown in this figure, features are not sufficiently resolved on uniform c32, but the addition of a finer level refined by a factor of 4 improves the resolution in the region of instability (c32/c128), and resolution is further improved with the addition of a third level (c32/c128/c512).

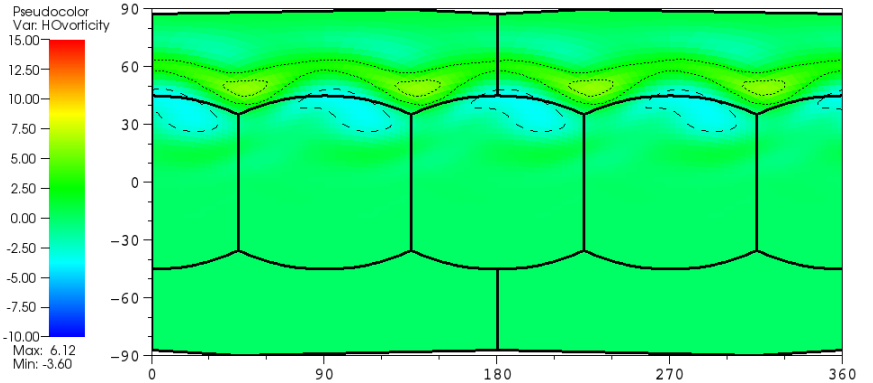
Figure 10, on the top half, shows the maximum difference in relative vorticity between uniform c1024 and each other run at half-day intervals. Above the refinement threshold of $0.102\,\text{day}^{-1}$, curves of maximum difference with c1024 look approximately the same when the finest level has the same resolution. Specifically, the result for two-level c16/c64 matches that for uniform c64, c32/c128 matches uniform c128, c64/c256 and c16/c64/c256 match uniform c256, and c128/c512 and c32/c128/c512 match uniform c512 above the refinement threshold of $0.102\,\text{day}^{-1}$. The bottom half of Figure 10 shows the maximum difference in relative vorticity between each two-level and three-level run and the corresponding run having uniform resolution of the finest level; this difference stays below the refinement threshold until approximately day 5, when the instability is fully formed.

Total energy E is invariant under the shallow-water equations and is defined by

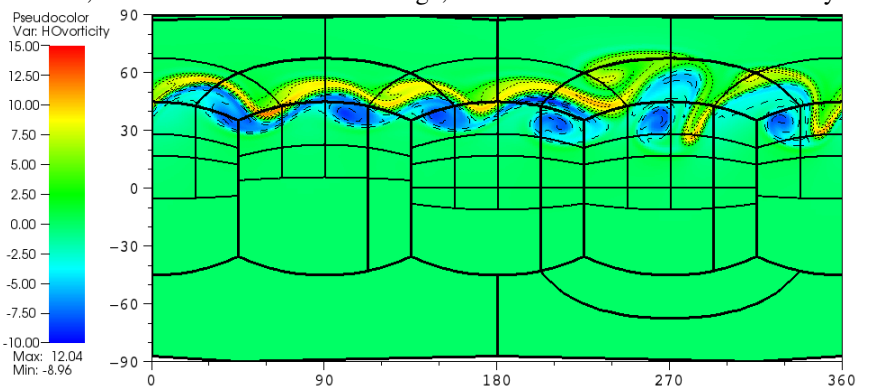
$$E = \tfrac{1}{2} h\boldsymbol{u}\cdot\boldsymbol{u} + \tfrac{1}{2} G(H^2 - z_s^2). \tag{82}$$

We calculate total energy by an area-weighted sum over the whole domain, accurate up to $O((\Delta\alpha)^2) = O((\Delta\beta)^2)$. In regions covered by grids with multiple levels of refinement, we take the sum over the finest level. Figure 11 shows the difference in total energy over time from its initial value, normalized by the initial total energy, for several runs: uniform c32, c128, and c512, two-level c32/c128, and three-level c32/c128/c512. We observe that higher spatial resolution corresponds to a substantial decrease in energy loss to numerical diffusion, with spatial convergence occurring at roughly fourth-order accuracy up to about day 4. At the highest resolutions, the calculation of total integrated shallow-water energy is dominated by truncation errors, leading to highly oscillatory behavior during the early part of the simulation. Results for the two-level c32/c128 and especially the three-level c32/c128/c512 are even more oscillatory because refinement does not necessarily preserve total energy. Nonetheless, all the simulations show a positive mean energy

Uniform resolution, c32



Two levels, c32/c128: 54.2% c128 coverage, with refinement threshold 0.102 day^{-1}



Three levels, c32/c128/c512: 54.2% c128 coverage, with refinement threshold 0.102 day^{-1}, and 32.9% c512 coverage, with refinement threshold 0.407 day^{-1}

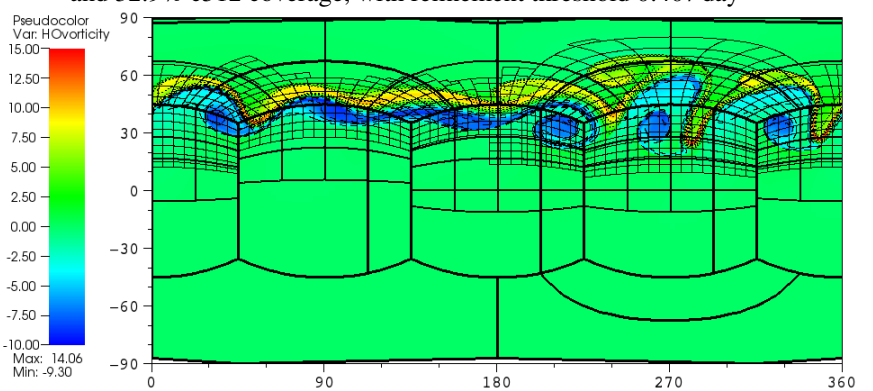


Figure 9. Relative vorticity field (in units of day^{-1}) at the final time (6 days) in the barotropic-instability test of Section 5.3, for c32 at the coarsest level. Black contour lines are drawn at values of the tick marks in the legend: dotted for positive and dashed for negative. In the two-level and three-level cases shown here, the second-level grids are the same and cover an area that coincides approximately with the northern hemisphere.

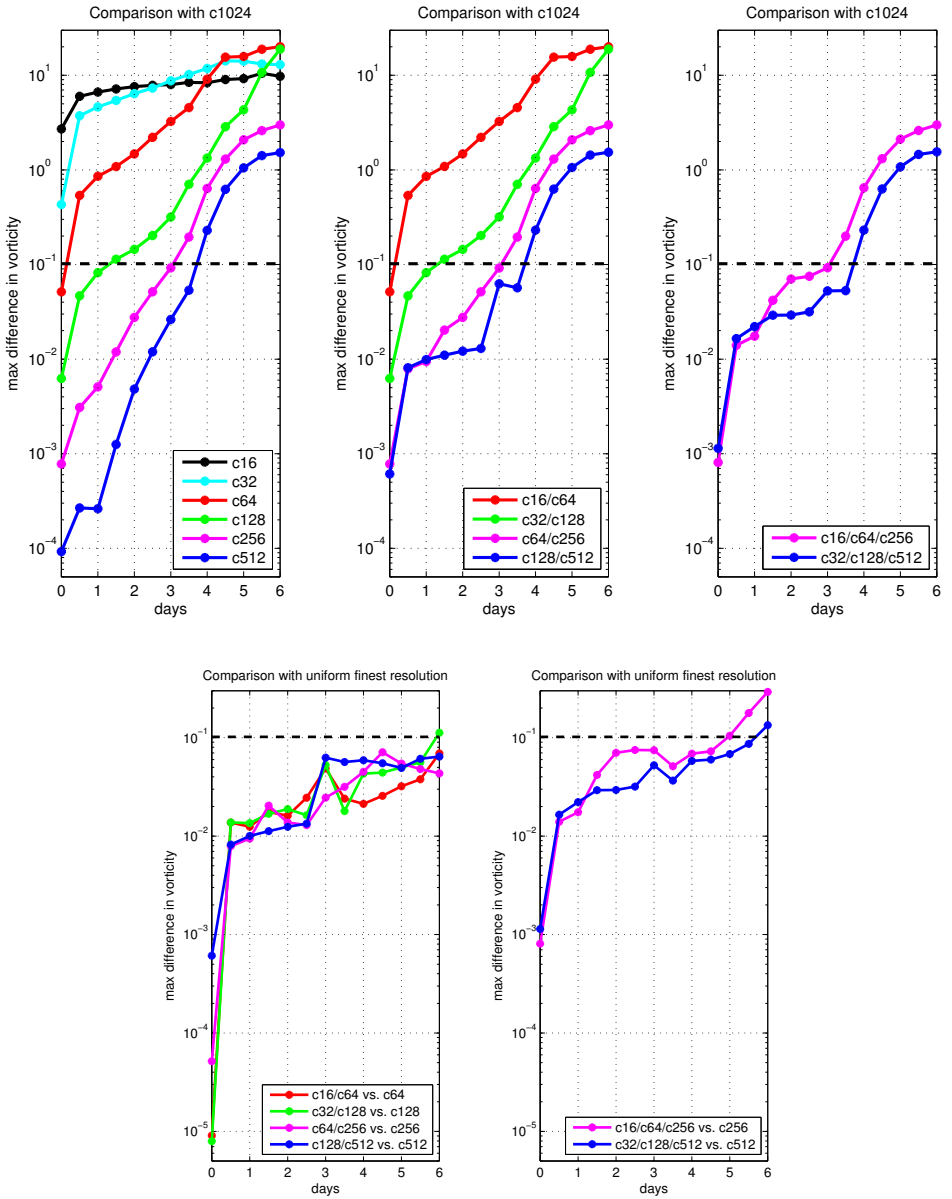


Figure 10. Plots of maximum differences in relative vorticity (in units of day^{-1}) between different runs of the barotropic-instability test of Section 5.3, shown at intervals of every half day. Plots for runs with the same *finest*-level resolution have the same color. Top: difference between uniform c1024 and (left to right) uniform, two-level, and three-level runs having resolution given in each legend. Bottom: difference between (left to right) two-level and three-level runs and the run with uniform resolution of the finest level in each case. On every plot, the refinement threshold of 0.102 day^{-1} from the coarsest level is marked with a dashed black line.

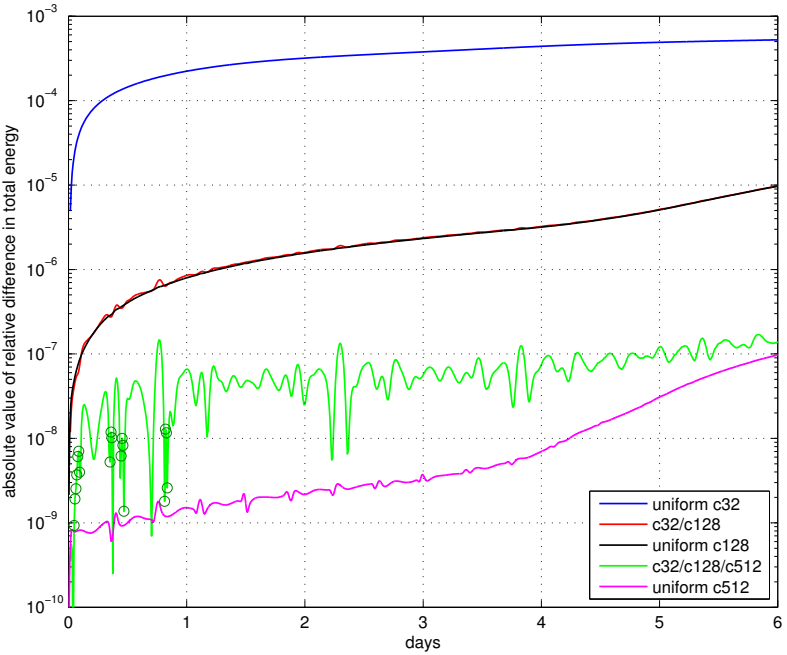


Figure 11. Plot of absolute value over time of the relative difference in total energy from initial value for five different runs of the barotropic-instability test of Section 5.3. Note that the curves for uniform c128 and for c32/c128 mostly overlap. The relative difference is *negative* at all steps after the initial time in all of these simulations with the exception of the c32/c128/c512 simulation, in which the relative difference is positive at the time steps marked with circles on the graph; as can be seen on the graph, all of these time steps occur before the end of day 1 and the relative difference never exceeds 2×10^{-8}.

loss, which suggests stability of the underlying numerical scheme. The three-level c32/c128/c512 simulation is the only one that shows total energy higher than its initial value at any stage of the simulation, but the stages where this occurs are all during the first day.

5.4. ***Gaussian pulse.*** The following example is included to test high-order convergence across refinement boundaries that are not characteristic. The initial velocity is zero, and the initial height field is a function of the latitude and is specified by a smoothed Gaussian with parameters $h_0 = 5000\,\mathrm{m}$ as background, $h_\delta = 500\,\mathrm{m}$ as maximum perturbation, and $w = \pi/10$ as angular width. With latitude ϕ, setting

$$\eta = \frac{\frac{1}{2}\pi - \phi}{w},$$

then

$$h(\eta) = \begin{cases} h_0 + h_\delta \exp(-4\eta^2)\cos^6(\frac{1}{2}\pi\eta) & \text{if } \eta < 1, \\ h_0 & \text{otherwise.} \end{cases} \tag{83}$$

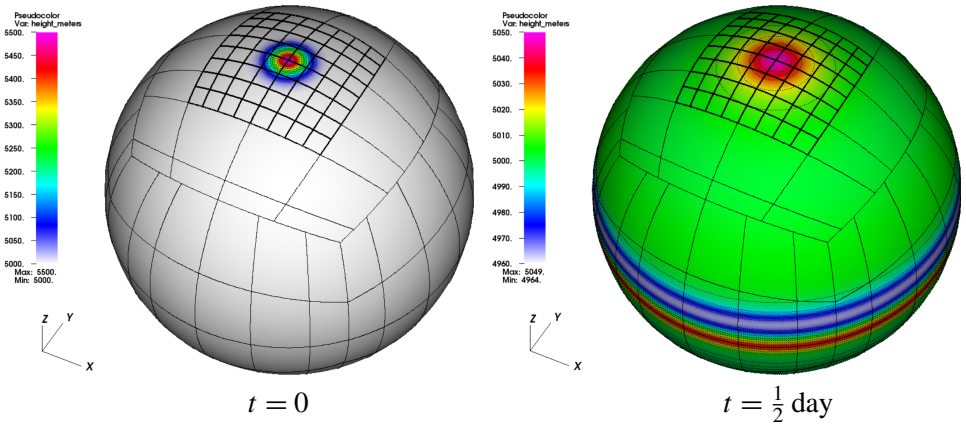


Figure 12. Total-height field for the Gaussian-pulse test case of Section 5.4 at (left) initial time $t = 0$ and (right) final time $t = \frac{1}{2}$ day. The base level is c128. There are fixed grids refined by a factor of 4 (hence a subset of c512) around the north pole, and these are shown with darker outlines than the coarse grids. Black contour lines (dotted) are drawn on each plot at values of the tick marks in the corresponding legend. Note the different color maps as initial h ranges from 5000 to 5500 meters and final h ranges from 4964 to 5049 meters.

The smoothing factor $\cos^6(\frac{1}{2}\pi\eta)$ is present in order to ensure that h is C^6. We calculate from times 0 to $\frac{1}{2}$ day, at which time the Gaussian has spread to the equator.

We pick time step $\Delta t = 0.4\,\text{day}/N_c$, and we find $c_{\max} = 6.30\,\text{rad/day}$, so the CFL number from (75) is 1.60. We run tests with uniform refinement, N_c a power of 2 and c32 up to c4096, and then with two levels, the coarser level having N_c a power of 2 and c32 up to c1024 and the finer level, with a refinement ratio of 4, consisting of grid cells encompassed by a square centered on the north pole, with side length half that of the north polar panel. Figure 12 shows h at initial time 0 and final time $\frac{1}{2}$ in a two-level c128/c512 run. The two-level runs are chosen so as to see the effect of a Gaussian initially contained within the finer level but then spreading past the coarse-fine boundary. Figure 13 shows a contour plot of calculated values of h in the two-level c128/512 run, at longitude 45°, as a function of latitude and time.

We take the solution with uniform c4096 to be a reference to compare results with the other resolutions. As seen in Table 3, the results approach fourth-order accuracy.

5.5. ***Zonal flow over an isolated mountain.*** Zonal flow over an isolated mountain is a key test of the performance of the model in the presence of topography. However, the traditionally employed shallow-water test of [61] has the disadvantage of being only C^0, hence preventing meaningful convergence studies beyond first order. Consequently, this paper uses a modified version of this test where the bottom

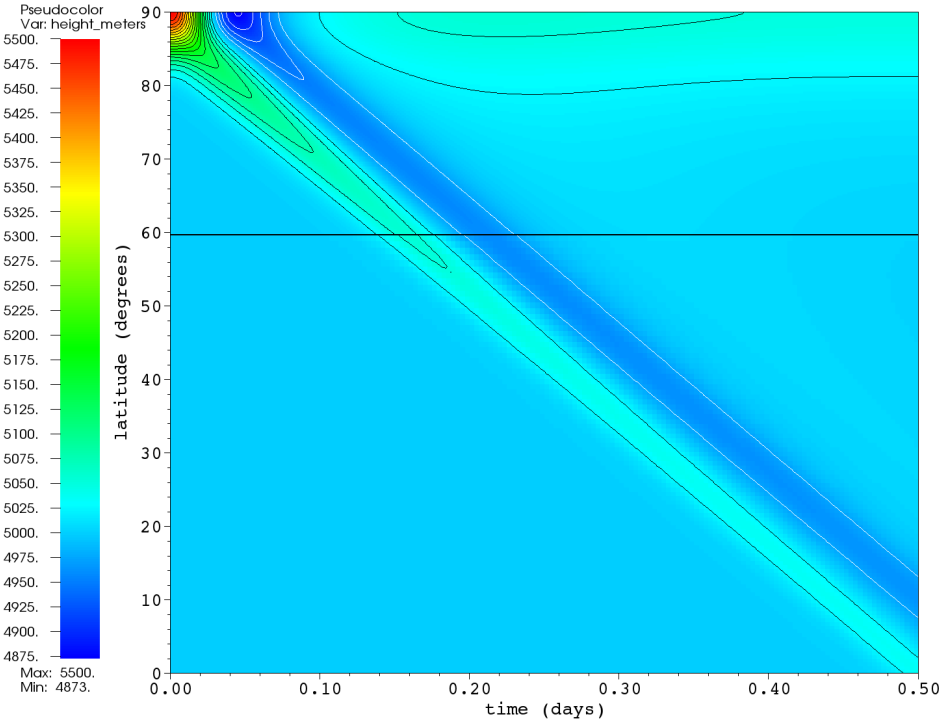


Figure 13. Total-height field for the Gaussian-pulse test case of Section 5.4 at longitude 45°, over all latitudes from initial time $t = 0$ to final time $t = \frac{1}{2}$ day. The base level is c128, and there is a finer level, a subset of c512, north of the refinement boundary indicated by the solid black line. At longitude 45°, this refinement boundary occurs at a corner of the grids shown in Figure 12. Contour lines are shown in black for every 25 meters above 5000 meters and in white for every 25 meters below 5000 meters.

topography is given by a C^3 cosine hill,

$$z_s = \frac{z_0}{4}\left[1 + \cos\left(\frac{\pi r}{R}\right)\right]^2, \tag{84}$$

where $R = \pi/9$ and $r^2 = \min\{R^2, (\lambda - \lambda_c)^2 + (\phi - \phi_c)^2\}$. The height of the mountain is $z_0 = 2000\,\mathrm{m}$, and its center is at $(\lambda_c, \phi_c) = (3\pi/2, \pi/6)$. The initial wind field is given by

$$u_\lambda = u_0 \cos\phi, \qquad u_\phi = 0 \tag{85}$$

and surface-height field by

$$H = h_0 - \frac{u_0}{2g}(u_0 + a\Omega)\sin^2\phi \tag{86}$$

with background height h_0 and velocity amplitude u_0 chosen to be

$$h_0 = 5960\,\mathrm{m}, \qquad u_0 = 20\,\mathrm{m}\cdot\mathrm{s}^{-1}. \tag{87}$$

Coarser resolution	Uniform resolution max error	rate	Two levels max error	rate
c32	1.489×10^{1}		1.286×10^{1}	
		1.20		1.12
c64	6.499×10^{0}		5.914×10^{0}	
		2.11		2.09
c128	1.509×10^{0}		1.390×10^{0}	
		2.90		2.86
c256	2.019×10^{-1}		1.912×10^{-1}	
		3.62		3.61
c512	1.641×10^{-2}		1.561×10^{-2}	
		3.95		3.95
c1024	1.059×10^{-3}		1.012×10^{-3}	

Table 3. Maximum difference between height in meters at final time with given resolutions and with uniform c4096 reference solution, and rates of convergence, for the Gaussian-pulse test case of Section 5.4. In the two-level runs, the refinement ratio between the coarser and finer levels is 4, so the resolution at the finer level is c128 through c4096.

We pick time step $\Delta t = 0.4\,\text{day}/N_c$, and we find $c_{\max} = 7.20\,\text{rad/day}$, so the CFL number from (75) is 1.83. We calculate up to 15 days with uniform refinement, N_c a power of 2 and c32 up to c1024.

Figure 14 shows the total height after 5, 10, and 15 days of the c128 solution. Although the mountain shape does not exactly match [61], we still observe an analogous appearance of a mix of large-scale Rossby waves and smaller-scale inertia-gravity waves.

We measure the error of the solution at a given time as the difference in total height between that solution and a c1024 reference solution. For runs with uniform resolutions from c32 to c512, Figure 15 shows the maximum magnitude of the error over the sphere after each day of the simulation. Note that up to day 6, the solution approaches fourth-order convergence. Figure 15 shows a jump in the maximum error in the c512 solution between day 6 and day 7 and a decrease in convergence rate to third order. In this case, the error in the c512 solution at day 7 is concentrated near one of the panel boundaries, in a region where the flow is tangent to the panel boundary. Where panel boundaries are characteristic, we expect a drop of one order of accuracy as is happening here in this case.

The longer-term solution approaches second-order convergence. This rate is expected because, as shown in [35; 47], once wave-breaking occurs the kinetic energy spectra of large-scale atmospheric flows will approach a decay rate of k^{-3}, corresponding to, at most, continuity of first derivatives of prognostic quantities.

Figure 16 shows the L^1 norm of the error after each day of the simulation, where the L^1 norm of a function is the integral of its absolute value over the sphere:

$$\|f\|_1 = \int |f|\, dA. \tag{88}$$

We see from Figure 16 that the L^1 norm of the error converges to fourth order with increasing refinement.

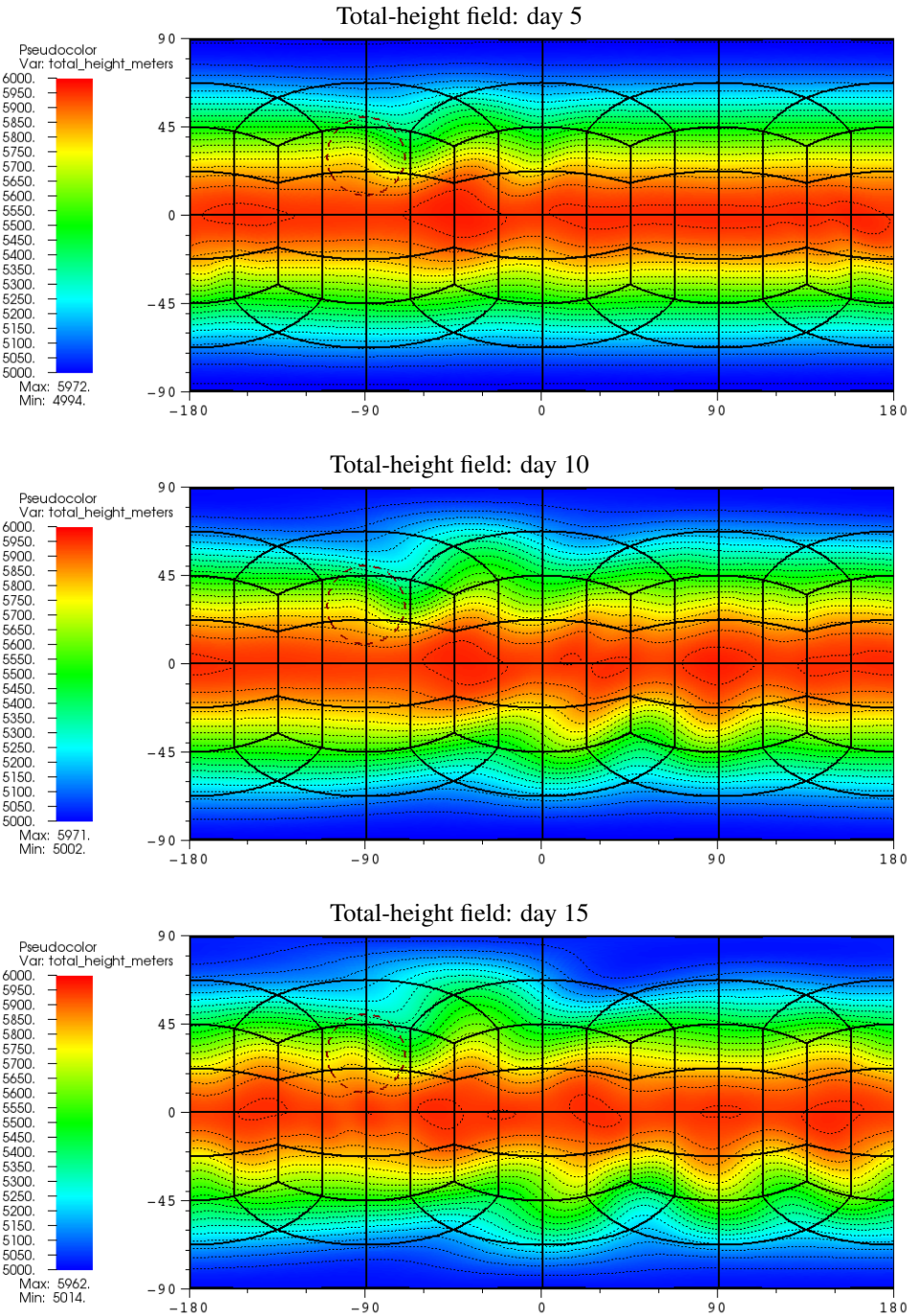


Figure 14. Total-height field for the C^3 mountain test case of Section 5.5, with c128 refinement. The base of the mountain is indicated with a dashed circle. Black contour lines (dotted) are drawn at intervals of 50 meters, at values of the tick marks in the legend.

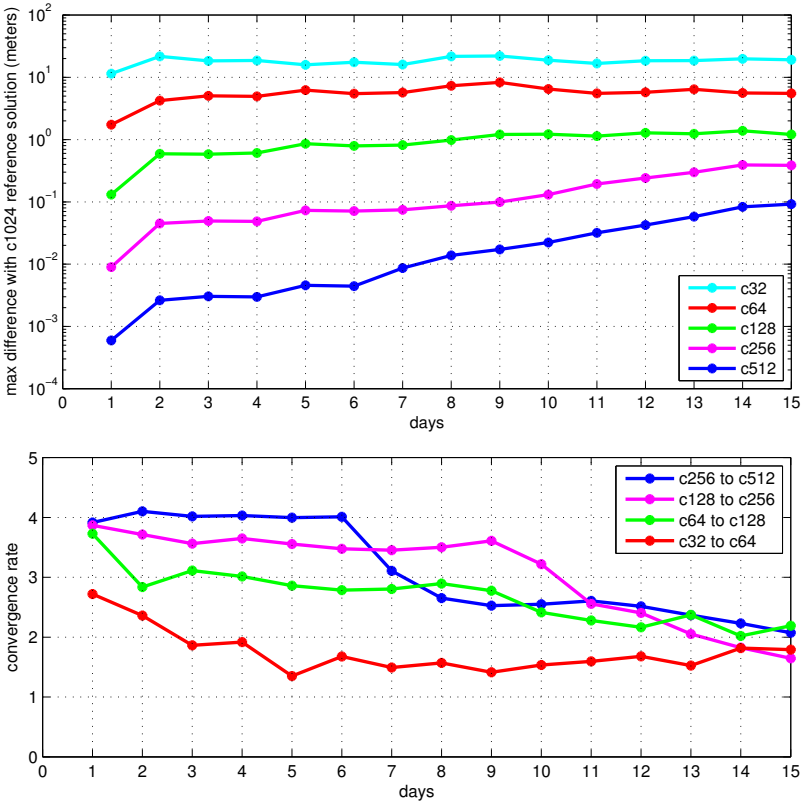


Figure 15. Top: plot of maximum differences over time between total height in meters in runs with given resolutions and the c1024 reference solution, for the C^3 mountain test case of Section 5.5. Bottom: plot of convergence rate over time, expressed as the base-2 logarithm of the ratio of the differences shown in the top plot for successive resolutions refined by a factor of 2.

6. Conclusions and future work

In this paper, we have presented a fourth-order-accurate finite-volume method on the cubed sphere. Despite formally third-order truncation-error accuracy at panel boundaries, the approach achieved fourth-order accuracy overall in smooth advection and the shallow-water equation test cases, with no evidence of panel-boundary artifacts. In addition, our results with adaptive mesh refinement show that, by using refined grids, it is possible to obtain overall solution error comparable to that on a uniform grid having the resolution of the finest level in the AMR hierarchy.

The next step is to extend this approach to the Euler equations on 3D thin spherical shells and complete a battery of dry atmospheric dynamical core tests. To that end, future work will include orography, which in 3D can be treated with several approaches such as cut-cell methods [59; 3], immersed-boundary methods [30], or

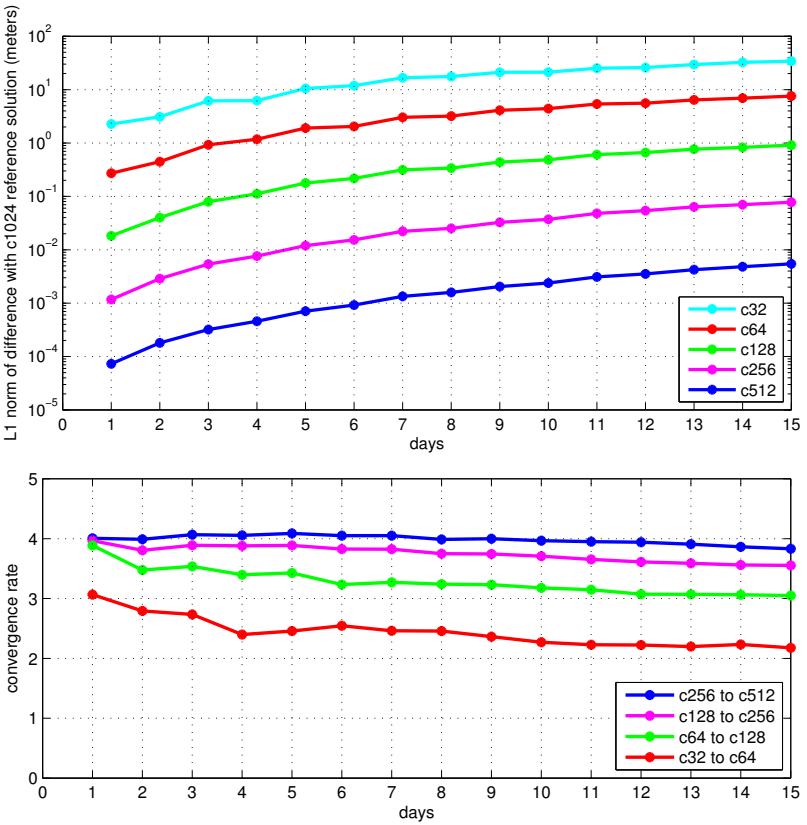


Figure 16. Top: plot of L^1 norm of differences over time between total height in meters in runs with given resolutions and the c1024 reference solution, for the C^3 mountain test case of Section 5.5. Bottom: plot of convergence rate over time, expressed as the base-2 logarithm of the ratio of the differences shown in the top plot for successive resolutions refined by a factor of 2.

terrain-following coordinates [14; 46]. In the near future, we anticipate incorporate climate cloud and radiation physics (such as that used in CESM [23]) with the goal of applying AMR to very high-resolution climate simulations.

Appendix A: Discrete undivided differences

This appendix gives the discrete undivided difference formulae that are used in Section 3 and their relationships to derivatives. The undivided differences are all denoted D with a subscript of α or β to indicate the direction in which the difference is taken and superscripts to indicate the order of the difference and whether the results are centered on the grid cells themselves (superscript c) or on their faces (superscript f).

A1. ***First differences on grid cells:*** $\boldsymbol{D^{1c\{C,L,R\}}_{\{\alpha,\beta\}}}$**.** First differences D^{1cC}_{α} and D^{1cC}_{β} on a grid cell take the 3-point centered finite-difference stencils:

$$(D^{1cC}_{\alpha}a)_{i,j} = \frac{a_{i+1,j} - a_{i-1,j}}{2}, \qquad (D^{1cC}_{\beta}a)_{i,j} = \frac{a_{i,j+1} - a_{i,j-1}}{2}. \tag{A-1}$$

One-sided differences $D^{1c\{L,R\}}_{\alpha}$ are given by

$$(D^{1cL}_{\alpha}a)_{i,j} = \frac{-3a_{i,j} + 4a_{i+1,j} - a_{i+2,j}}{2}, \tag{A-2}$$

$$(D^{1cR}_{\alpha}a)_{i,j} = \frac{a_{i-2,j} - 4a_{i-1,j} + 3a_{i,j}}{2} \tag{A-3}$$

and one-sided differences $D^{1c\{L,R\}}_{\beta}$ by

$$(D^{1cL}_{\beta}a)_{i,j} = \frac{-3a_{i,j} + 4a_{i,j+1} - a_{i,j+2}}{2}, \tag{A-4}$$

$$(D^{1cR}_{\beta}a)_{i,j} = \frac{a_{i,j-2} - 4a_{i,j-1} + 3a_{i,j}}{2}. \tag{A-5}$$

These differences are related to partial derivatives as

$$D^{1c\{C,L,R\}}_{\alpha}a = \Delta\alpha\frac{\partial a}{\partial\alpha} + O((\Delta\alpha)^3), \qquad D^{1c\{C,L,R\}}_{\beta}a = \Delta\beta\frac{\partial a}{\partial\beta} + O((\Delta\beta)^3). \tag{A-6}$$

A2. ***Second differences on grid cells:*** $\boldsymbol{D^{2c}_{\{\alpha,\beta\}}}$**.** Second differences D^{2c}_{α} and D^{2c}_{β} take the 3-point centered finite-difference stencils:

$$(D^{2c}_{\alpha}a)_{i,j} = a_{i+1,j} - 2a_{i,j} + a_{i-1,j}, \qquad (D^{2c}_{\beta}a)_{i,j} = a_{i,j+1} - 2a_{i,j} + a_{i,j-1}. \tag{A-7}$$

These differences are related to partial derivatives as

$$D^{2c}_{\alpha}a = (\Delta\alpha)^2\frac{\partial^2 a}{\partial\alpha^2} + O((\Delta\alpha)^4), \qquad D^{2c}_{\beta}a = (\Delta\beta)^2\frac{\partial^2 a}{\partial\beta^2} + O((\Delta\beta)^4). \tag{A-8}$$

A3. ***First transverse differences on faces of grid cells:*** $\boldsymbol{D^{1f}_{\{\alpha,\beta\}}}$**.** The first transverse differences, D^{1f}_{β} on faces of constant α and D^{1f}_{α} on faces of constant β, take the 3-point centered finite-difference stencils:

$$(D^{1f}_{\beta}a)_{i+\frac{1}{2},j} = \frac{a_{i+\frac{1}{2},j+1} - a_{i+\frac{1}{2},j-1}}{2}, \tag{A-9}$$

$$(D^{1f}_{\alpha}a)_{i,j+\frac{1}{2}} = \frac{a_{i+1,j+\frac{1}{2}} - a_{i-1,j+\frac{1}{2}}}{2}. \tag{A-10}$$

These differences are related to partial derivatives as

$$D^{1f}_{\beta}a = \Delta\beta\frac{\partial a}{\partial\beta} + O((\Delta\beta)^3), \qquad D^{1f}_{\alpha}a = \Delta\alpha\frac{\partial a}{\partial\alpha} + O((\Delta\alpha)^3). \tag{A-11}$$

A4. ***Second transverse differences on faces of grid cells:*** $\boldsymbol{D^{2\mathrm{f}}_{\{\alpha,\beta\}}}$**.** The second transverse differences, $D^{2\mathrm{f}}_{\beta}$ on faces of constant α and $D^{2\mathrm{f}}_{\alpha}$ on faces of constant β, take the 3-point centered finite-difference stencils:

$$(D^{2\mathrm{f}}_{\beta}a)_{i+\frac{1}{2},j} = a_{i+\frac{1}{2},j-1} - 2a_{i+\frac{1}{2},j} + a_{i+\frac{1}{2},j+1}, \tag{A-12}$$

$$(D^{2\mathrm{f}}_{\alpha}a)_{i,j+\frac{1}{2}} = a_{i-1,j+\frac{1}{2}} - 2a_{i,j+\frac{1}{2}} + a_{i+1,j+\frac{1}{2}}. \tag{A-13}$$

These differences are related to partial derivatives as

$$D^{2\mathrm{f}}_{\beta}a = (\Delta\beta)^2\frac{\partial^2 a}{\partial\beta^2} + O((\Delta\beta)^4), \qquad D^{2\mathrm{f}}_{\alpha}a = (\Delta\alpha)^2\frac{\partial^2 a}{\partial\alpha^2} + O((\Delta\alpha)^4). \tag{A-14}$$

A5. ***Fifth differences on faces of grid cells:*** $\boldsymbol{D^{5\mathrm{f}}_{\{\alpha,\beta\}}}$**.** For the artificial dissipation in Step (9) in Section 3.4, we need fifth undivided differences on faces, from data on grid cells:

$$(D^{5\mathrm{f}}_{\alpha}a)_{i+\frac{1}{2},j} = 10(a_{i+1,j} - a_{i,j}) - 5(a_{i+2,j} - a_{i-1,j}) + a_{i+3,j} - a_{i-2,j}, \tag{A-15}$$

$$(D^{5\mathrm{f}}_{\beta}a)_{i,j+\frac{1}{2}} = 10(a_{i,j+1} - a_{i,j}) - 5(a_{i,j+2} - a_{i,j-1}) + a_{i,j+3} - a_{i,j-2}. \tag{A-16}$$

These differences are related to partial derivatives as

$$D^{5\mathrm{f}}_{\alpha}a = (\Delta\alpha)^5\frac{\partial^5 a}{\partial\alpha^5} + O((\Delta\alpha)^7), \qquad D^{5\mathrm{f}}_{\beta}a = (\Delta\beta)^5\frac{\partial^5 a}{\partial\beta^5} + O((\Delta\beta)^7). \tag{A-17}$$

Appendix B: High-order averages over grid cells and faces

We use angle brackets $\langle\cdot\rangle_{i,j}$ to denote the average of a quantity over a computational grid cell $V_{i,j}$. An average over the face of $V_{i,j}$ where $\alpha = \alpha_i \pm \frac{1}{2}\Delta\alpha$ and $\beta \in [\beta_j - \frac{1}{2}\Delta\beta, \beta_j + \frac{1}{2}\Delta\beta]$ is denoted by $\langle\cdot\rangle_{i\pm\frac{1}{2},j}$, and an average over the face where $\beta = \beta_j \pm \frac{1}{2}\Delta\beta$ and $\alpha \in [\alpha_i - \frac{1}{2}\Delta\alpha, \alpha_i + \frac{1}{2}\Delta\alpha]$ is denoted by $\langle\cdot\rangle_{i,j\pm\frac{1}{2}}$.

B1. ***Exact*** $\boldsymbol{\langle J\rangle}$ ***on grid cells.*** For J defined in (5), the average $\langle J\rangle$ on a grid cell $V_{i,j}$ can be computed exactly:

$$\begin{aligned}\langle J\rangle_{i,j} &= \frac{1}{\Delta\alpha\,\Delta\beta}\int_{\beta_j-\frac{1}{2}\Delta\beta}^{\beta_j+\frac{1}{2}\Delta\beta}\int_{\alpha_i-\frac{1}{2}\Delta\alpha}^{\alpha_i+\frac{1}{2}\Delta\alpha} J\,d\alpha\,d\beta \\ &= \frac{r^2}{\Delta\alpha\,\Delta\beta}\sum_{p=0}^{1}\sum_{q=0}^{1}(-1)^{p+q}\tan^{-1}\frac{X_pY_q}{\sqrt{1+X_p^2+Y_q^2}},\end{aligned} \tag{B-18}$$

where $X_0 = \tan(\alpha_i - \frac{1}{2}\Delta\alpha)$, $X_1 = \tan(\alpha_i + \frac{1}{2}\Delta\alpha)$, $Y_0 = \tan(\beta_j - \frac{1}{2}\Delta\beta)$, and $Y_1 = \tan(\beta_j + \frac{1}{2}\Delta\beta)$.

B2. ***Exact $\langle J \rangle$ on faces of grid cells.*** We can also compute exactly the average of J over faces of grid cells.

- On faces of constant $\alpha = \alpha_i + \frac{1}{2}\Delta\alpha$, with β extending from $\beta_j - \frac{1}{2}\Delta\beta$ to $\beta_j + \frac{1}{2}\Delta\beta$:

$$\langle J \rangle_{i+\frac{1}{2},j} = \int_{\beta_j - \frac{1}{2}\Delta\beta}^{\beta_j + \frac{1}{2}\Delta\beta} J \, d\beta = \frac{r^2 Y_1}{\sqrt{1 + X^2 + Y_1^2}} - \frac{r^2 Y_0}{\sqrt{1 + X^2 + Y_0^2}}, \tag{B-19}$$

  where $X = \tan(\alpha)$, $Y_0 = \tan(\beta_j - \frac{1}{2}\Delta\beta)$, and $Y_1 = \tan(\beta_j + \frac{1}{2}\Delta\beta)$.

- On faces of constant $\beta = \beta_j + \frac{1}{2}\Delta\beta$, with α extending from $\alpha_i - \frac{1}{2}\Delta\alpha$ to $\alpha_i + \frac{1}{2}\Delta\alpha$:

$$\langle J \rangle_{i,j+\frac{1}{2}} = \int_{\alpha_i - \frac{1}{2}\Delta\alpha}^{\alpha_i + \frac{1}{2}\Delta\alpha} J \, d\alpha = \frac{r^2 X_1}{\sqrt{1 + X_1^2 + Y^2}} - \frac{r^2 X_0}{\sqrt{1 + X_0^2 + Y^2}}, \tag{B-20}$$

  where $X_0 = \tan(\alpha_i - \frac{1}{2}\Delta\alpha)$, $X_1 = \tan(\alpha_i + \frac{1}{2}\Delta\alpha)$, and $Y = \tan(\beta)$.

B3. ***High-order conversion between averaged and centered values.***

- If we have a at centers of grid cells, then by expanding Taylor series, we can obtain averages of a over grid cells:

$$\langle a \rangle_{i,j} = a_{i,j} + \frac{(\Delta\alpha)^2}{24}\left(\frac{\partial^2 a}{\partial\alpha^2}\right)_{i,j} + \frac{(\Delta\beta)^2}{24}\left(\frac{\partial^2 a}{\partial\beta^2}\right)_{i,j} + O((\Delta\alpha)^4, (\Delta\alpha)^2(\Delta\beta)^2, (\Delta\beta)^4). \tag{B-21}$$

  Using the discrete-differences notation of Appendix A2, this can be written as

$$\langle a \rangle_{i,j} = a_{i,j} + \tfrac{1}{24}(D_\alpha^{2c} a)_{i,j} + \tfrac{1}{24}(D_\beta^{2c} a)_{i,j} + O((\Delta\alpha)^4, (\Delta\alpha)^2(\Delta\beta)^2, (\Delta\beta)^4), \tag{B-22}$$

$$a_{i,j} = \langle a \rangle_{i,j} - \tfrac{1}{24}(D_\alpha^{2c} \langle a \rangle)_{i,j} - \tfrac{1}{24}(D_\beta^{2c} \langle a \rangle)_{i,j} + O((\Delta\alpha)^4, (\Delta\alpha)^2(\Delta\beta)^2, (\Delta\beta)^4). \tag{B-23}$$

- With a at centers of faces of grid cells, we can also expand the Taylor series to obtain an approximation to averages over faces:

$$\langle a \rangle_{i+\frac{1}{2},j} = a_{i+\frac{1}{2},j} + \frac{(\Delta\beta)^2}{24}\left(\frac{\partial^2 a}{\partial\beta^2}\right)_{i+\frac{1}{2},j} + O((\Delta\beta)^4), \tag{B-24}$$

$$\langle a \rangle_{i,j+\frac{1}{2}} = a_{i,j+\frac{1}{2}} + \frac{(\Delta\alpha)^2}{24}\left(\frac{\partial^2 a}{\partial\alpha^2}\right)_{i,j+\frac{1}{2}} + O((\Delta\alpha)^4). \tag{B-25}$$

Hence, taking the discrete differences of Appendix A4,

$$\langle a\rangle_{i+\frac{1}{2},j} = a_{i+\frac{1}{2},j} + \tfrac{1}{24}(D_\beta^{2\mathrm{f}} a)_{i+\frac{1}{2},j} + O((\Delta\beta)^4), \tag{B-26}$$

$$a_{i+\frac{1}{2},j} = \langle a\rangle_{i+\frac{1}{2},j} - \tfrac{1}{24}(D_\beta^{2\mathrm{f}} \langle a\rangle)_{i+\frac{1}{2},j} + O((\Delta\beta)^4), \tag{B-27}$$

$$\langle a\rangle_{i,j+\frac{1}{2}} = a_{i,j+\frac{1}{2}} + \tfrac{1}{24}(D_\alpha^{2\mathrm{f}} a)_{i,j+\frac{1}{2}} + O((\Delta\alpha)^4), \tag{B-28}$$

$$a_{i,j+\frac{1}{2}} = \langle a\rangle_{i,j+\frac{1}{2}} - \tfrac{1}{24}(D_\alpha^{2\mathrm{f}} \langle a\rangle)_{i,j+\frac{1}{2}} + O((\Delta\alpha)^4). \tag{B-29}$$

B4. ***High-order product formulae.***

- As shown in [10], the average of a product of a and b over a grid cell is

$$\langle ab\rangle = \langle a\rangle\langle b\rangle + \frac{(\Delta\alpha)^2}{12}\frac{\partial a}{\partial\alpha}\frac{\partial b}{\partial\alpha} + \frac{(\Delta\beta)^2}{12}\frac{\partial a}{\partial\beta}\frac{\partial b}{\partial\beta} + O((\Delta\alpha)^4, (\Delta\alpha)^2(\Delta\beta)^2, (\Delta\beta)^4). \tag{B-30}$$

Hence on $V_{i,j}$, using (A-6) with the undivided differences $D_\alpha^{1\mathrm{cC}}$ and $D_\beta^{1\mathrm{cC}}$ from Appendix A1,

$$\langle ab\rangle_{i,j} = \langle a\rangle_{i,j}\langle b\rangle_{i,j} + \tfrac{1}{12}(D_\alpha^{1\mathrm{cC}} a)_{i,j}(D_\alpha^{1\mathrm{cC}} b)_{i,j} + \tfrac{1}{12}(D_\beta^{1\mathrm{cC}} a)_{i,j}(D_\beta^{1\mathrm{cC}} b)_{i,j} + O((\Delta\alpha)^4, (\Delta\alpha)^2(\Delta\beta)^2, (\Delta\beta)^4), \tag{B-31}$$

and the average of one of the factors can be obtained from the average of the product by

$$\langle b\rangle_{i,j} = \frac{\langle ab\rangle_{i,j} - \tfrac{1}{12}\left(D_\alpha^{1\mathrm{cC}}\frac{\langle ab\rangle}{\langle a\rangle}\right)_{i,j}(D_\alpha^{1\mathrm{cC}}\langle a\rangle)_{i,j} - \tfrac{1}{12}\left(D_\beta^{1\mathrm{cC}}\frac{\langle ab\rangle}{\langle a\rangle}\right)_{i,j}(D_\beta^{1\mathrm{cC}}\langle a\rangle)_{i,j}}{\langle a\rangle_{i,j}} + O((\Delta\alpha)^4, (\Delta\alpha)^2(\Delta\beta)^2, (\Delta\beta)^4). \tag{B-32}$$

In (B-32), we can substitute the one-sided $D_\alpha^{1\mathrm{cL}}$ or $D_\alpha^{1\mathrm{cR}}$ for the centered $D_\alpha^{1\mathrm{cC}}$ if $V_{i-1,j}$ or $V_{i+1,j}$, respectively, is not a grid cell of the panel containing $V_{i,j}$. Similarly, we can substitute $D_\beta^{1\mathrm{cL}}$ or $D_\beta^{1\mathrm{cR}}$ for $D_\beta^{1\mathrm{cC}}$ if $V_{i,j-1}$ or $V_{i,j+1}$, respectively, is not a grid cell of the panel containing $V_{i,j}$.

- Also from [10] and using (A-11), the average of a product of a and b over the face of a grid cell with constant α is

$$\langle ab\rangle_{i+\frac{1}{2},j} = \langle a\rangle_{i+\frac{1}{2},j}\langle b\rangle_{i+\frac{1}{2},j} + \tfrac{1}{12}(D_\beta^{1\mathrm{f}} a)_{i+\frac{1}{2},j}(D_\beta^{1\mathrm{f}} b)_{i+\frac{1}{2},j} + O((\Delta\beta)^4) \tag{B-33}$$

and over the face of a grid cell with constant β is

$$\langle ab\rangle_{i,j+\frac{1}{2}} = \langle a\rangle_{i,j+\frac{1}{2}}\langle b\rangle_{i,j+\frac{1}{2}} + \tfrac{1}{12}(D_\alpha^{1\mathrm{f}} a)_{i,j+\frac{1}{2}}(D_\alpha^{1\mathrm{f}} b)_{i,j+\frac{1}{2}} + O((\Delta\alpha)^4). \tag{B-34}$$

References

[1] M. Adams, P. Colella, D. T. Graves, J. N. Johnson, H. S. Johansen, N. D. Keen, T. J. Ligocki, D. F. Martin, P. W. McCorquodale, D. Modiano, P. O. Schwartz, T. D. Sternberg, and B. Van Straalen, *Chombo software package for AMR applications: design document*, Tech. Report LBNL-6616E, Lawrence Berkeley National Laboratory, 2014.

[2] L. Bao, R. D. Nair, and H. M. Tufo, *A mass and momentum flux-form high-order discontinuous Galerkin shallow water model on the cubed-sphere*, J. Comput. Phys. **271** (2014), 224–243. MR 3209599

[3] M. F. Barad, P. Colella, and S. G. Schladow, *An adaptive cut-cell method for environmental fluid mechanics*, Internat. J. Numer. Methods Fluids **60** (2009), no. 5, 473–514. MR 2010d:76022

[4] J. R. Bates, F. H. M. Semazzi, R. W. Higgins, and S. R. M. Barros, *Integration of the shallow water equations on the sphere using a vector semi-Lagrangian scheme with a multigrid solver*, Mon. Weather Rev. **118** (1990), no. 8, 1615–1627.

[5] S. Blaise and A. St-Cyr, *A dynamic hp-adaptive discontinuous Galerkin method for shallow-water flows on the sphere with application to a global tsunami simulation*, Mon. Weather Rev. **140** (2012), no. 3, 978–996.

[6] C. Chaplin and P. Colella, *A single stage flux-corrected transport algorithm for high-order finite-volume methods*, Tech. report, 2015, Submitted to Commun. Appl. Math. Comput. Sci. arXiv 1506.02999v1

[7] C. Chen, X. Li, X. Shen, and F. Xiao, *Global shallow water models based on multi-moment constrained finite volume method and three quasi-uniform spherical grids*, J. Comput. Phys. **271** (2014), 191–223. MR 3209598

[8] C. Chen and F. Xiao, *Shallow water model on cubed-sphere by multi-moment finite volume method*, J. Comput. Phys. **227** (2008), no. 10, 5019–5044. MR 2009e:86001

[9] C. Chen, F. Xiao, and X. Li, *An adaptive multimoment global model on a cubed sphere*, Mon. Weather Rev. **139** (2011), no. 2, 523–548.

[10] P. Colella, M. R. Dorr, J. A. F. Hittinger, and D. F. Martin, *High-order, finite-volume methods in mapped coordinates*, J. Comput. Phys. **230** (2011), no. 8, 2952–2976. MR 2012d:65245 Zbl 1218.65119

[11] R. Comblen, S. Legrand, E. Deleersnijder, and V. Legat, *A finite element method for solving the shallow water equations on the sphere*, Ocean Model. **38** (2009), no. 1–3, 12–23.

[12] J. Côté and A. Staniforth, *An accurate and efficient finite-element global model of the shallow-water equations*, Mon. Weather Rev. **118** (1990), no. 12, 2707–2717.

[13] J. M. Dennis, J. Edwards, K. J. Evans, O. Guba, P. H. Lauritzen, A. A. Mirin, A. St-Cyr, M. A. Taylor, and P. H. Worley, *CAM-SE: a scalable spectral element dynamical core for the Community Atmosphere Model*, Int. J. High Perform. C. **26** (2012), no. 1, 74–89.

[14] T. Gal-Chen and R. C. J. Somerville, *On the use of a coordinate transformation for the solution of the Navier–Stokes equations*, J. Computational Phys. **17** (1975), 209–228. MR 51 #2470 Zbl 0297.76020

[15] J. Galewsky, R. K. Scott, and L. M. Polvani, *An initial-value problem for testing numerical models of the global shallow-water equations*, Tellus A **56** (2004), no. 5, 429–440.

[16] A. Gassmann, *A global hexagonal C-grid non-hydrostatic dynamical core (ICON-IAP) designed for energetic consistency*, Q. J. Roy. Meteor. Soc. **139** (2013), no. 670, 152–175.

[17] F. X. Giraldo, J. S. Hesthaven, and T. Warburton, *Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations*, J. Comput. Phys. **181** (2002), no. 2, 499–525. MR 2003g:86004 Zbl 1178.76268

[18] M. Govett, J. Middlecoff, and T. Henderson, *Running the NIM next-generation weather model on GPUs*, 10th IEEE/ACM international conference on cluster, cloud and grid computing (Melbourne, 2010) (M. Parashar and R. Buyya, eds.), IEEE Computer Society, Los Lamitos, CA, 2010, pp. 792–796.

[19] S. M. Guzik, P. McCorquodale, and P. Colella, *A freestream-preserving high-order finite-volume method for mapped grids with adaptive-mesh refinement*, 50th AIAA aerospace sciences meeting (Nashville, TN, 2012), AIAA, Reston, VA, 2012, p. 0574.

[20] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations, I: Nonstiff problems*, 2nd ed., Springer Series in Computational Mathematics, no. 8, Springer, Berlin, 1993. MR 94c:65005

[21] R. Heikes and D. A. Randall, *Numerical integration of the shallow-water equations on a twisted icosahedral grid, I: Basic design and results of tests*, Mon. Weather Rev. **123** (1995), no. 6, 1862–1880.

[22] J. Hilditch and P. Colella, *A projection method for low Mach number fast chemistry reacting flow*, 35th AIAA aerospace sciences meeting (Reno, NV, 1997), AIAA, Reston, VA, 1997, p. 0263.

[23] J. W. Hurrell, M. M. Holland, P. R. Gent, S. Ghan, J. E. Kay, P. J. Kushner, J.-F. Lamarque, W. G. Large, D. Lawrence, K. Lindsay, W. H. Lipscomb, M. C. Long, N. Mahowald, D. R. Marsh, R. B. Neale, P. Rasch, S. Vavrus, M. Vertenstein, D. Bader, W. D. Collins, J. J. Hack, J. Kiehl, and S. Marshall, *The community earth system model: a framework for collaborative research*, B. Am. Meteorol. Soc. **94** (2013), no. 9, 1339–1360.

[24] S. Ii and F. Xiao, *A global shallow water model using high order multi-moment constrained finite volume method and icosahedral grid*, J. Comput. Phys. **229** (2010), no. 5, 1774–1796. MR 2010i:65179

[25] R. Jakob-Chien, J. J. Hack, and D. L. Williamson, *Spectral transform solutions to the shallow water test set*, J. Comput. Phys. **119** (1995), no. 1, 164–187. Zbl 0878.76059

[26] M. Läuter, F. X. Giraldo, D. Handorf, and K. Dethloff, *A discontinuous Galerkin method for the shallow water equations in spherical triangular coordinates*, J. Comput. Phys. **227** (2008), no. 24, 10226–10242. MR 2467951 Zbl 1218.76028

[27] X. Li, D. Chen, X. Peng, K. Takahashi, and F. Xiao, *A multimoment finite-volume shallow-water model on the Yin–Yang overset spherical grid*, Mon. Weather Rev. **136** (2008), no. 8, 3066–3086.

[28] S.-J. Lin, *A "vertically Lagrangian" finite-volume dynamical core for global models*, Mon. Weather Rev. **132** (2004), no. 10, 2293–2307.

[29] S.-J. Lin and R. B. Rood, *An explicit flux-form semi-Lagrangian shallow-water model on the sphere*, Q. J. Roy. Meteor. Soc. **123** (1997), no. 544, 2477–2498.

[30] K. A. Lundquist, F. K. Chow, and J. K. Lundquist, *An immersed boundary method for the weather research and forecasting model*, Mon. Weather Rev. **138** (2010), no. 3, 796–817.

[31] P. McCorquodale, M. R. Dorr, J. A. F. Hittinger, and P. Colella, *High-order finite-volume methods for hyperbolic conservation laws on mapped multiblock grids*, J. Comput. Phys. **288** (2015), 181–195. MR 3320206

[32] P. McCorquodale and P. Colella, *A high-order finite-volume method for conservation laws on locally refined grids*, Commun. Appl. Math. Comput. Sci. **6** (2011), no. 1, 1–25. MR 2012h:65181 Zbl 1252.65163

[33] R. D. Nair, S. J. Thomas, and R. D. Loft, *A discontinuous Galerkin transport scheme on the cubed sphere*, Mon. Weather Rev. **133** (2005), no. 4, 814–828.

[34] R. D. Nair and P. H. Lauritzen, *A class of deformational flow test cases for linear transport problems on the sphere*, J. Comput. Phys. **229** (2010), no. 23, 8868–8887. MR 2011f:86010 Zbl 1282.86012

[35] G. D. Nastrom and K. S. Gage, *A climatology of atmospheric wavenumber spectra of wind and temperature observed by commercial aircraft*, J. Atmos. Sci. **42** (1985), no. 9, 950–960.

[36] W. M. Putman and S.-J. Lin, *A finite-volume dynamical core on the cubed-sphere grid*, Numerical modeling of space plasma flows: ASTRONUM 2008 (Saint John, United States Virgin Islands, 2008) (N. V. Pogorelov, E. Audit, P. Colella, and G. P. Zank, eds.), ASP Conference Series, no. 406, Astronomical Society of the Pacific, San Francisco, 2009, pp. 268–276.

[37] W. M. Putman and S.-J. Lin, *Finite-volume transport on various cubed-sphere grids*, J. Comput. Phys. **227** (2007), no. 1, 55–78. MR 2008j:86001 Zbl 1126.76038

[38] A. Qaddouri, J. Pudykiewicz, M. Tanguay, C. Girard, and J. Côté, *Experiments with different discretizations for the shallow-water equations on a sphere*, Q. J. Roy. Meteor. Soc. **138** (2012), no. 665, 989–1003.

[39] T. Ringler, L. Ju, and M. Gunzburger, *A multiresolution method for climate system modeling: application of spherical centroidal Voronoi tessellations*, Ocean Dynam. **58** (2008), no. 5–6, 475–498.

[40] T. D. Ringler, D. Jacobsen, M. Gunzburger, L. Ju, M. Duda, and W. Skamarock, *Exploring a multiresolution modeling approach within the shallow-water equations*, Mon. Weather Rev. **139** (2011), no. 11, 3348–3368.

[41] H. Ritchie, *Application of the semi-Lagrangian method to a spectral model of the shallow water equations*, Mon. Weather Rev. **116** (1988), no. 8, 1587–1598.

[42] C. Ronchi, R. Iacono, and P. S. Paolucci, *The "cubed sphere": a new method for the solution of partial differential equations in spherical geometry*, J. Comput. Phys. **124** (1996), no. 1, 93–114. MR 96k:86001

[43] J. A. Rossmanith, *A wave propagation method for hyperbolic systems on the sphere*, J. Comput. Phys. **213** (2006), no. 2, 629–658. MR 2007i:65065 Zbl 1089.65088

[44] R. Sadourny, *Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids*, Mon. Weather Rev. **100** (1972), no. 2, 136–144.

[45] M. Satoh, T. Matsuno, H. Tomita, H. Miura, T. Nasuno, and S. Iga, *Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations*, J. Comput. Phys. **227** (2008), no. 7, 3486–3514. MR 2009b:86006 Zbl 1132.86311

[46] C. Schär, D. Leuenberger, O. Fuhrer, D. Lüthi, and C. Girard, *A new terrain-following vertical coordinate formulation for atmospheric prediction models*, Mon. Weather Rev. **130** (2002), no. 10, 2459–2480.

[47] W. C. Skamarock, *Evaluating mesoscale NWP models using kinetic energy spectra*, Mon. Weather Rev. **132** (2004), no. 12, 3019–3032.

[48] W. C. Skamarock, J. B. Klemp, M. G. Duda, L. D. Fowler, S.-H. Park, and T. D. Ringler, *A multiscale nonhydrostatic atmospheric model using centroidal Voronoi tesselations and C-grid staggering*, Mon. Weather Rev. **140** (2012), no. 9, 3090–3105.

[49] A. St-Cyr, C. Jablonowski, J. M. Dennis, H. M. Tufo, and S. J. Thomas, *A comparison of two shallow-water models with nonconforming adaptive grids*, Mon. Weather Rev. **136** (2008), no. 6, 1898–1922.

[50] A. Staniforth and J. Thuburn, *Horizontal grids for global weather and climate prediction models: a review*, Q. J. Roy. Meteor. Soc. **138** (2012), no. 662, 1–26.

[51] M. Taylor, J. Tribbia, and M. Iskandarani, *The spectral element method for the shallow water equations on the sphere*, J. Comput. Phys. **130** (1997), no. 1, 92–108. Zbl 0868.76072

[52] S. J. Thomas and R. D. Loft, *The NCAR spectral element climate dynamical core: semi-implicit Eulerian formulation*, J. Sci. Comput. **25** (2005), no. 1-2, 307–322. MR 2007b:86019 Zbl 1203.86013

[53] M. A. Tolstykh, *Vorticity-divergence semi-Lagrangian shallow-water model of the sphere based on compact finite differences*, J. Comput. Phys. **179** (2002), no. 1, 180–200. MR 2003e:76085a Zbl 1060.76086

[54] M. A. Tolstykh and V. V. Shashkin, *Vorticity-divergence mass-conserving semi-Lagrangian shallow-water model using the reduced grid on the sphere*, J. Comput. Phys. **231** (2012), no. 11, 4205–4233. MR 2911791

[55] P. A. Ullrich, *Understanding the treatment of waves in atmospheric models, I: The shortest resolved waves of the 1D linearized shallow-water equations*, Q. J. Roy. Meteor. Soc. **140** (2014), no. 682, 1426–1440.

[56] P. A. Ullrich and C. Jablonowski, *MCore: a non-hydrostatic atmospheric dynamical core utilizing high-order finite-volume methods*, J. Comput. Phys. **231** (2012), no. 15, 5078–5108. MR 2929934 Zbl 1247.86007

[57] P. A. Ullrich, C. Jablonowski, and B. van Leer, *High-order finite-volume methods for the shallow-water equations on the sphere*, J. Comput. Phys. **229** (2010), no. 17, 6104–6134. MR 2011d:76069

[58] US CLIVAR Scientific Steering Committee, *US climate variability & predictability program science plan*, Tech. Report 2013-7, US CLIVAR Project Office, 2013.

[59] R. L. Walko and R. Avissar, *The Ocean-Land-Atmosphere Model (OLAM), II: Formulation and tests of the nonhydrostatic dynamic core*, Mon. Weather Rev. **136** (2008), no. 11, 4045–4062.

[60] H. Weller, *Controlling the computational modes of the arbitrarily structured C grid*, Mon. Weather Rev. **140** (2012), no. 10, 3220–3234.

[61] D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, *A standard test set for numerical approximations to the shallow water equations in spherical geometry*, J. Comput. Phys. **102** (1992), no. 1, 211–224. MR 93d:86006 Zbl 0756.76060

[62] N. Wood, A. Staniforth, A. White, T. Allen, M. Diamantakis, M. Gross, T. Melvin, C. Smith, S. Vosper, M. Zerroukat, and J. Thuburn, *An inherently mass-conserving semi-implicit semi-Lagrangian discretization of the deep-atmosphere global non-hydrostatic equations*, Q. J. Roy. Meteor. Soc. **140** (2014), no. 682, 1505–1520.

[63] M. Zerroukat, N. Wood, A. Staniforth, A. A. White, and J. Thuburn, *An inherently mass-conserving semi-implicit semi-Lagrangian discretisation of the shallow-water equations on the sphere*, Q. J. Roy. Meteor. Soc. **135** (2009), no. 642, 1104–1116.

Received June 24, 2014. Revised May 26, 2015.

PETER MCCORQUODALE: pwmccorquodale@lbl.gov
Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A1148, Berkeley, CA 94720, United States

PAUL A. ULLRICH: paullrich@ucdavis.edu
Department of Land, Air and Water Resources, University of California, Davis, 1 Shields Avenue, Davis, CA 95616, United States

HANS JOHANSEN: hjohansen@lbl.gov
Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A1148, Berkeley, CA 94720, United States

PHILLIP COLELLA: pcolella@lbl.gov
Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A1148, Berkeley, CA 94720, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 10, No. 2, 2015
dx.doi.org/10.2140/camcos.2015.10.163

LOW MACH NUMBER FLUCTUATING HYDRODYNAMICS OF BINARY LIQUID MIXTURES

ANDY NONAKA, YIFEI SUN, JOHN B. BELL AND ALEKSANDAR DONEV

Continuing on our previous work (A. Donev, A. Nonaka, Y. Sun, T. G. Fai, A. L. Garcia and J. B. Bell, *Comm. App. Math. and Comp. Sci.* **9** (2014), no. 1, 47–105), we develop semi-implicit numerical methods for solving low Mach number fluctuating hydrodynamic equations appropriate for modeling diffusive mixing in isothermal mixtures of fluids with different densities and transport coefficients. We treat viscous dissipation implicitly using a recently developed variable-coefficient Stokes solver (M. Cai, A. J. Nonaka, J. B. Bell, B. E. Griffith and A. Donev, Commun. Comput. Phys. **16** (2014), no. 5, 1263–1297). This allows us to increase the time step size significantly for low Reynolds number flows with large Schmidt numbers compared to our earlier explicit temporal integrator. Also, unlike most existing deterministic methods for low Mach number equations, our methods do not use a fractional time-step approach in the spirit of projection methods, thus avoiding splitting errors and giving full second-order deterministic accuracy even in the presence of boundaries for a broad range of Reynolds numbers including steady Stokes flow. We incorporate the Stokes solver into two time-advancement schemes, where the first is suitable for inertial flows and the second is suitable for the overdamped limit (viscous-dominated flows), in which inertia vanishes and the fluid motion can be described by a steady Stokes equation. We also describe how to incorporate advanced higher-order Godunov advection schemes in the numerical method, allowing for the treatment of (very) large Péclet number flows with a vanishing mass diffusion coefficient. We incorporate thermal fluctuations in the description in both the inertial and overdamped regimes. We validate our algorithm with a series of stochastic and deterministic tests. Finally, we apply our algorithms to model the development of giant concentration fluctuations during the diffusive mixing of water and glycerol, and compare numerical results with experimental measurements. We find good agreement between the two, and observe propagative (nondiffusive) modes at small wavenumbers (large spatial scales), not reported in published experimental measurements of concentration fluctuations in fluid mixtures. Our work forms the foundation for developing low Mach number fluctuating hydrodynamics methods for miscible multispecies mixtures of chemically reacting fluids.

MSC2010: primary 76T99; secondary 65M08.

Keywords: fluctuating hydrodynamics, binary mixtures, giant fluctuations, Stokes solver, low Mach flow.

I. Introduction

Flows of realistic mixtures of miscible fluids exhibit several features that make them more difficult to simulate numerically than flows of simple fluids. Firstly, the physical properties of the mixture depend on the concentration of the different species composing the mixture. This includes both the density of the mixture at constant pressure, and transport coefficients such as viscosity and mass diffusion coefficients. Common simplifying assumptions such as the Boussinesq approximation, which assumes a constant density and thus incompressible flow, or assuming constant transport coefficients, are uncontrolled and not appropriate for certain mixtures of very dissimilar fluids. Secondly, for liquid mixtures there is a large separation of time scales between the various dissipative processes, notably, mass diffusion is much slower than momentum diffusion. The large Schmidt numbers $\mathrm{Sc} \sim 10^3$–10^4 typical of liquid mixtures lead to extreme stiffness and make direct temporal integration of the hydrodynamic equations infeasible. Lastly, flows of mixtures exhibit all of the numerical difficulties found in single component flows, for example, well-known difficulties caused by advection in the absence of sufficiently strong dissipation (diffusion of momentum or mass), and challenges in incorporating thermal fluctuations in the description. Here we develop a low Mach number approach to isothermal binary fluid mixtures that resolves many of the above difficulties, and paves the way for incorporating additional physics such as the presence of more than two species [5], chemical reactions [11; 1], multiple phases and surface tension [50; 15], and others.

Stochastic fluctuations are intrinsic to fluid dynamics because fluids are composed of molecules whose positions and velocities are random. Thermal fluctuations affect flows from microscopic to macroscopic scales [26; 56] and need to be consistently included in all levels of description. Fluctuating hydrodynamics (FHD) incorporates thermal fluctuations into the usual Navier–Stokes–Fourier laws in the form of stochastic contributions to the dissipative momentum, heat, and mass fluxes [22]. FHD has proven to be a very useful tool in understanding complex fluid flows far from equilibrium [31; 50; 54; 4]; however, theoretical calculations are often only feasible after making many uncontrolled approximations [22], and numerical schemes used for fluctuating hydrodynamics are usually far behind state-of-the-art deterministic computational fluid dynamics (CFD) solvers.

In this work, we consider binary mixtures and restrict our attention to isothermal flows. We consider a specific equation of state (EOS) suitable for mixtures of incompressible liquids or ideal gases, but otherwise account for advective and diffusive mass and momentum transport in full generality. Recently, some of us developed finite-volume methods for the incompressible equations [6]. We have also developed low Mach number isothermal fluctuating equations [28], which

eliminate the stiffness arising from the separation of scales between acoustic and vortical modes [38; 47; 49]. The low Mach number equations account for the fact that for mixtures of fluids with different densities, diffusive and stochastic mass fluxes create local expansion and contraction of the fluid. In these equations the incompressibility constraint should be replaced by a "quasi-incompressibility" constraint [49; 40], which introduces some difficulties in constructing conservative finite-volume techniques [46; 48; 21; 43; 42; 28]. In Section II we review the low Mach number equations of fluctuating hydrodynamics for a binary mixture of miscible fluids, as first proposed in Ref. [28].

The numerical method developed in Ref. [28] uses an explicit temporal integrator. This requires using a small time step and is infeasible for liquid mixtures due to the stiffness caused by the separation of time scales between fast momentum diffusion and slow mass diffusion. In recent work [24], some of us developed temporal integrators for the equations of fluctuating hydrodynamics that have several important advantages. Notably, these integrators are semi-implicit, allowing one to treat fast momentum diffusion (viscous dissipation) implicitly, and other transport processes explicitly. These temporal integrators are constructed to be second-order accurate for the equations of linearized fluctuating hydrodynamics (LFHD), which are suitable for describing thermal fluctuations around stable macroscopic flows over a broad range of length and time scales [22]. Importantly, the linearization of the fluctuating equations is carried out *automatically* by the code, making the numerical methods very similar to standard deterministic CFD schemes. Finally, specific integrators are proposed in Ref. [24] to handle the extreme separation of scales between the fast velocity and the slow concentration by taking an *overdamped* limit of the inertial equations.

In this work, we extend the semi-implicit temporal integrators proposed in Ref. [24] for incompressible flows to account for the quasi-incompressible nature of low Mach number flows. We apply these temporal integrators to the staggered-grid conservative finite-volume spatial discretization developed in Ref. [28], and additionally generalize the treatment of advection to allow for the use of monotonicity-preserving higher-order Godunov schemes [8; 9; 41; 44].

Our work relies heavily on several prior works, which we will only briefly summarize in the present paper. The spatial discretization we describe in more detail in Section III B is identical to that proposed by Donev et al. [28], which itself relies heavily on the treatment of thermal fluctuations developed in Refs. [6; 28]. A key development that makes the algorithm presented here feasible for large-scale problems is recent work by some of us [13] on efficient multigrid-based iterative methods for solving unsteady and steady variable-coefficient Stokes problems on staggered grids. Our high-order Godunov method for mass advection is based on the work of Bell et al. [9; 41; 44].

The temporal integrators developed in Section III D are a novel approach to low Mach number hydrodynamics even in the deterministic context. In high-resolution finite-volume methods, the dominant paradigm has been to use a splitting (fractional-step) or projection method [16] to separate the pressure and velocity updates [3; 21; 7; 2; 37]. We followed such a projection approach to construct an explicit temporal integrator for the low Mach number equations [28]. When viscosity is treated implicitly, however, the splitting introduces a commutator error that leads to the appearance of spurious or "parasitic" modes in the presence of physical boundaries [32; 12; 23]. There are several techniques to reduce (but not eliminate) these artificial boundary layers [12], and for sufficiently large Reynolds number flows the time step size dictated by advective stability constraints makes the splitting error relatively small in practice. At small Reynolds numbers, however, the splitting error becomes larger as viscous effects become more dominant, and projection methods do not apply in the steady Stokes regime for problems with physical boundary conditions. Methods that do not split the velocity and pressure updates but rather solve a combined Stokes system for velocity and pressure have been used in the finite-element literature for some time, and have more recently been used in the finite-volume context for incompressible flow [35]. Here we demonstrate how the same approach can be effectively applied to the low Mach number equations for a binary fluid mixture [28], to construct a method that is second-order accurate up to boundaries, for a broad range of Reynolds numbers including steady Stokes flow.

We test our ability to accurately capture the static structure factor for equilibrium fluctuation calculations. Then, we test our methods deterministically on two variable density and variable viscosity low Mach number flows. First, we confirm second-order deterministic accuracy in both space and time for a lid-driven cavity problem in the presence of a bubble of a denser miscible fluid. Next, we simulate the development of a Kevin–Helmholtz instability as a lighter less viscous fluid streams over a denser more viscous fluid. These tests confirm the robustness and accuracy of the methods in the presence of large contrasts, sharp gradients, and boundaries. Next we focus on the use of fluctuating low Mach number equations to study giant concentration fluctuations. In Section V we apply our methods to study the development of giant fluctuations [58; 19; 56; 55] during free diffusive mixing of water and glycerol. We compare simulation results to experimental measurements of the time-correlation function of concentration fluctuations during the diffusive mixing of water and glycerol [19]. The relaxation times show signatures of the rich deterministic dynamics, and a transition from purely diffusive relaxation of concentration fluctuations at large wavenumbers, to more complex buoyancy-driven dynamics at smaller wavenumbers. We find reasonably good agreement given the large experimental uncertainties, and observe the appearance of propagative modes at small wavenumbers, which we suggest could be observed in experiments as well.

II. Low Mach number equations

At mesoscopic scales, in typical liquids, sound waves are very low amplitude and much faster than momentum diffusion; hence, they can usually be eliminated from the fluid dynamics description. Formally, this corresponds to taking the zero Mach number singular limit $\mathrm{Ma} \to 0$ of the well-known compressible fluctuating hydrodynamics equations system [39; 22]. In the compressible equations, the coupling between momentum and mass transport is captured by the equation of state (EOS) for the pressure $P(\rho, c; T_0)$ as a local function of the density $\rho(\boldsymbol{r}, t)$ and mass concentration $c(\boldsymbol{r}, t)$ at a specified temperature $T_0(\boldsymbol{r})$, assumed to be time-independent in our isothermal model.

The low Mach number equations can be obtained by making the ansatz that the thermodynamic behavior of the system is captured by a reference pressure $P_0(\boldsymbol{r}, t)$, with the additional pressure contribution $\pi(\boldsymbol{r}, t) = O(\mathrm{Ma}^2)$ capturing the mechanical behavior while not affecting the thermodynamics. We will restrict consideration to cases where stratification due to gravity causes negligible changes in the thermodynamic state across the domain. In this case, the reference pressure is spatially constant and constrains the system so that the evolution of ρ and c remains consistent with the thermodynamic EOS

$$P\big(\rho(\boldsymbol{r}, t), c(\boldsymbol{r}, t); T_0(\boldsymbol{r})\big) = P_0(t). \tag{1}$$

Physically this means that any change in concentration must be accompanied by a corresponding change in density, as would be observed in a system at thermodynamic equilibrium held at the fixed reference pressure and temperature. The EOS defines density $\rho\big(c(\boldsymbol{r}, t); T_0(\boldsymbol{r}), P_0(t)\big)$ as an implicit function of concentration in a binary liquid mixture. The EOS constraint (1) can be rewritten as a constraint on the divergence of the fluid velocity $\boldsymbol{v}(\boldsymbol{r}, t)$,

$$\rho \boldsymbol{\nabla} \cdot \boldsymbol{v} = -\beta \, \boldsymbol{\nabla} \cdot \boldsymbol{F}, \tag{2}$$

where $\boldsymbol{F}$ is the total diffusive mass flux defined in (10), and the solutal expansion coefficient

$$\beta(c) = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial c}\right)_{P_0, T_0}$$

is determined by the specific form of the EOS.

In this work we consider a specific *linear* EOS,

$$\frac{\rho_1}{\bar{\rho}_1} + \frac{\rho_2}{\bar{\rho}_2} = \frac{c\rho}{\bar{\rho}_1} + \frac{(1-c)\rho}{\bar{\rho}_2} = 1, \tag{3}$$

where $\bar{\rho}_1$ and $\bar{\rho}_2$ are the densities of the pure component fluids ($c = 1$ and $c = 0$,

respectively), giving

$$\beta = \rho\left(\frac{1}{\bar{\rho}_2} - \frac{1}{\bar{\rho}_1}\right) = \frac{\bar{\rho}_1 - \bar{\rho}_2}{c\bar{\rho}_2 + (1-c)\bar{\rho}_1}. \tag{4}$$

It is important that for this specific form of the EOS β/ρ is a material constant independent of the concentration; this allows us to write the EOS constraint (9) in conservative form $\nabla \cdot \boldsymbol{v} = -\nabla \cdot (\beta\rho^{-1}\boldsymbol{F})$ and take the reference pressure P_0 to be independent of time. The specific form of the density dependence (4) on concentration arises if one assumes that two incompressible fluids do not change volume upon mixing, which is a reasonable assumption for liquids that are not too dissimilar at the molecular level. Surprisingly the EOS (3) is also valid for a mixture of ideal gases. If the specific EOS (3) is not a very good approximation over the entire range of concentration $0 \le c \le 1$, it may be a very good approximation over the range of concentrations of interest if $\bar{\rho}_1$ and $\bar{\rho}_2$ are adjusted accordingly. Our choice of the specific form of the EOS will aid significantly in the construction of simple conservative spatial discretizations that strictly maintain the EOS without requiring complicated nonlinear iterative corrections.

In fluctuating hydrodynamics, stochastic contributions to the momentum and mass fluxes are formally modeled as follows [6]:

$$\begin{aligned} \boldsymbol{\Sigma} &= \sqrt{\eta k_B T}(\boldsymbol{\mathcal{W}} + \boldsymbol{\mathcal{W}}^T), \\ \boldsymbol{\Psi} &= \sqrt{2\chi\rho\mu_c^{-1}k_B T}\,\widetilde{\boldsymbol{\mathcal{W}}}, \end{aligned} \tag{5}$$

where k_B is Boltzmann's constant, η is the shear viscosity, χ is the diffusion coefficient, $\mu(c; T_0, P_0)$ is the chemical potential of the mixture with $\mu_c = (\partial\mu/\partial c)_{P_0,T_0}$, and $\boldsymbol{\mathcal{W}}(\boldsymbol{r}, t)$ and $\widetilde{\boldsymbol{\mathcal{W}}}(\boldsymbol{r}, t)$ are standard zero mean, unit variance random Gaussian tensor and vector fields, respectively, with uncorrelated components,

$$\left\langle \boldsymbol{\mathcal{W}}_{ij}(\boldsymbol{r}, t)\boldsymbol{\mathcal{W}}_{kl}(\boldsymbol{r}', t')\right\rangle = \delta_{ik}\delta_{jl}\,\delta(t-t')\delta(\boldsymbol{r}-\boldsymbol{r}'),$$

and similarly for $\widetilde{\boldsymbol{\mathcal{W}}}$.

A standard asymptotic low Mach analysis [38], formally treating the stochastic forcing as smooth, leads to the *isothermal low Mach number* equations for a binary mixture of fluids in conservation form [28],

$$\partial_t(\rho\boldsymbol{v}) + \nabla\pi = -\nabla\cdot(\rho\boldsymbol{v}\boldsymbol{v}^T) + \nabla\cdot(\eta\overline{\nabla}\boldsymbol{v} + \boldsymbol{\Sigma}) + \rho\boldsymbol{g} \tag{6}$$

$$\partial_t(\rho_1) = -\nabla\cdot(\rho_1\boldsymbol{v}) + \nabla\cdot\boldsymbol{F} \tag{7}$$

$$\partial_t(\rho_2) = -\nabla\cdot(\rho_2\boldsymbol{v}) - \nabla\cdot\boldsymbol{F} \tag{8}$$

$$\nabla\cdot\boldsymbol{v} = -\nabla\cdot(\beta\rho^{-1}\boldsymbol{F}), \tag{9}$$

where the deterministic and stochastic diffusive mass fluxes are denoted by

$$\boldsymbol{F} = \rho\chi\nabla c + \boldsymbol{\Psi}. \tag{10}$$

Here $\overline{\nabla} = \nabla + \nabla^T$ is a symmetric gradient, $\rho_1 = \rho c$ is the density of the first component, $\rho_2 = (1 - c)\rho$ is the density of the second component, and $\boldsymbol{g}$ is the gravitational acceleration. The gradient of the nonthermodynamic component of the pressure π (Lagrange multiplier) appears in the momentum equation as a driving force that ensures the EOS constraint (9) is obeyed. We note that the bulk viscosity term gives a gradient term that can be absorbed in π and therefore does not explicitly need to appear in the equations. Temperature dynamics and fluctuations are neglected in these equations; however, this type of approach can be extended to include thermal effects. The shear viscosity $\eta(c; T_0, P_0)$ and the mass diffusion coefficient $\chi(c; T_0, P_0)$ in general depend on the concentration. Note that the two density equations (7) and (8) can be combined to obtain the usual continuity equation for the total density,

$$\partial_t \rho = -\nabla \cdot (\rho \boldsymbol{v}), \tag{11}$$

and the primitive (nonconservation law) form of the concentration equation,

$$\rho(\partial_t c + \boldsymbol{v} \cdot \nabla c) = \nabla \cdot \boldsymbol{F}. \tag{12}$$

Our conservative numerical scheme is based on Equations (6), (7), (9), and (11).

In Ref. [28] we discussed the effect of the low Mach constraint on the thermal fluctuations, suitable boundary conditions for the low Mach equations, and presented a gauge formulation of the equations that formally eliminates pressure in a manner similar to the projection operator formulation for incompressible flows. Importantly, the gauge formulation demonstrates that although the low Mach equations have the appearance of a constrained system, one can write them in an unconstrained form by introducing a gauge degree of freedom for the pressure. For the purposes of time integration, one can therefore treat these equations as standard initial-value problems [28] and use the temporal integrators developed in Ref. [24].

A. ***Linearized low Mach fluctuating hydrodynamics.*** It is important to note that the equations of fluctuating hydrodynamics should be interpreted as a mesoscopic coarse-grained representation of the mass, momentum and energy transport in fluids [45]. As such, these equations implicitly contain a mesoscopic coarse-graining length and time scale that is larger than molecular scales [33] and can only formally be written as stochastic partial differential equations (SPDEs). A coarse-graining scale can explicitly be included in the SPDEs [27; 29]; such a coarse-graining scale explicitly enters in our finite-volume spatiotemporal discretization through the grid spacing (equivalently, the volume of a grid cell, or more precisely, the number of molecules per grid cell). Additional difficulties are posed by the fact that in general the noise in the nonlinear equations is multiplicative, requiring a careful stochastic

interpretation; the Mori–Zwanzig projection formalism [34] suggests the correct stochastic interpretation is the kinetic one [36].

For compressible and incompressible flows, the SPDEs of *linearized* fluctuating hydrodynamics (LFHD) [22] can be given a precise continuum meaning [20; 30; 23; 24]. In these linearized equations one splits each variable into a deterministic component and small fluctuations around the deterministic solution, e.g., $c(\boldsymbol{r}, t) = \bar{c}(\boldsymbol{r}, t) + \delta c(\boldsymbol{r}, t)$, where $\bar{c}$ is a solution of the deterministic equations (6), (7), (9) and (11), with $\boldsymbol{\Psi} = 0$ and $\boldsymbol{\Sigma} = 0$. Here δc is the solution of a *linear* additive-noise equation obtained by linearizing (12) to first order in the fluctuations and evaluating the noise amplitude at the deterministic solution; more precisely, LFHD is an expression of the central limit theorem in the limit of weak noise. In this work, in the stochastic setting we restrict our attention to LFHD equations. As discussed in Ref. [24], we do not need to write down the (rather tedious) complete form of the linearized low Mach number equations (for an illustration, see the next subsection) since the numerical method will perform this linearization automatically. Namely, the complete nonlinear equations are essentially equivalent to the LFHD equations when the noise is sufficiently weak, i.e., when the hydrodynamic cells contain many molecules.

The low Mach number equations pose additional difficulties because they represent a coarse-graining of the dynamics not just in space but also in time. As such, even the linearized equations cannot directly be interpreted as describing a standard diffusion process. This is because the stochastic mass flux $\boldsymbol{\Psi}$ in the EOS constraint (9) makes the velocity formally white-in-time [28]. We note, however, that the analysis in Ref. [27] shows that there is a close connection between mass diffusion and advection by the thermally fluctuating velocity field, and thus between $\boldsymbol{\Psi}$ and velocity fluctuations. This suggests that a precise interpretation of the low Mach constraint in the presence of stochastic mass fluxes requires a very delicate mathematical analysis. In this work we rely on the implicit coarse-graining in time provided by the finite time step size in the temporal integration schemes to regularize the low Mach equations [28]. Furthermore, for the applications we study here, we can neglect stochastic mass fluxes and assume $\boldsymbol{\Psi} \approx 0$, in which case the difficulties related to a white-in-time velocity disappear.

B. ***Overdamped limit.*** At small scales, flows in liquids are viscous-dominated and the inertial momentum flux $\rho \boldsymbol{v}\boldsymbol{v}^T$ can often be neglected in a zero Reynolds number approximation. In addition, in liquids, there is a large separation of time scales between the fast momentum diffusion and slow mass diffusion, i.e., the Schmidt number $\mathrm{Sc} = \eta/(\rho\chi)$ is large. This makes the relaxation times of velocity modes at sufficiently large wavenumbers much smaller than those of the concentration modes. Formally treating the stochastic force terms as smooth for the moment,

the separation of time scales implies that we can replace the *inertial* momentum equation (6) with the *overdamped* steady-Stokes equation

$$\begin{aligned} -\nabla\cdot(\eta\overline{\nabla}\boldsymbol{v})+\nabla\pi &= \nabla\cdot\boldsymbol{\Sigma}+\rho\boldsymbol{g},\\ \nabla\cdot\boldsymbol{v} &= -(\bar{\rho}_2^{-1}-\bar{\rho}_1^{-1})\nabla\cdot(\chi\rho\nabla c+\boldsymbol{\Psi}). \end{aligned} \tag{13}$$

The above equations can be used to eliminate velocity as a variable, leaving only the concentration equation (12). Note that the density equation (11) simply defines density as a function of concentration and thus is not considered an independent equation.

The solution of the Stokes system

$$\begin{aligned} -\nabla\cdot(\eta\overline{\nabla}\boldsymbol{v})+\nabla\pi &= \boldsymbol{f},\\ \nabla\cdot\boldsymbol{v} &= -h, \end{aligned} \tag{14}$$

where $\boldsymbol{f}(\boldsymbol{r},t)$ and $h(\boldsymbol{r},t)$ are applied forcing terms, can be expressed in terms of a generalized inverse Stokes linear operator[1] $\mathcal{L}^{-1}[\eta(\cdot,t)]$ that is a *functional* of the viscosity (and thus the concentration),

$$\boldsymbol{v}=\mathcal{L}^{-1}[\eta](\boldsymbol{f},\,h).$$

In the linearized fluctuating equations, one must linearize around the (time dependent) solution of the deterministic nonlinear equation

$$\bar{\rho}(\partial_t\bar{c}+\bar{\boldsymbol{v}}\cdot\nabla\bar{c})=\nabla\cdot(\bar{\rho}\bar{\chi}\nabla\bar{c}), \tag{15}$$

where we have used the shorthand notation $\bar{\rho}=\rho(\bar{c})$, $\bar{\eta}=\eta(\bar{c})$, $\bar{\chi}=\chi(\bar{c})$. Here the velocity is an implicit function of concentration defined via

$$\begin{aligned} -\nabla\cdot(\bar{\eta}\overline{\nabla}\boldsymbol{v})+\nabla\bar{\pi} &= \bar{\rho}\boldsymbol{g}\\ \nabla\cdot\bar{\boldsymbol{v}} &= -(\bar{\rho}_2^{-1}-\bar{\rho}_1^{-1})\nabla\cdot(\bar{\chi}\bar{\rho}\nabla\bar{c}), \end{aligned}$$

which we can write in shorthand notation as

$$\bar{\boldsymbol{v}}=\mathcal{L}^{-1}[\bar{\eta}]\big(\bar{\rho}\boldsymbol{g},\ (\bar{\rho}_2^{-1}-\bar{\rho}_1^{-1})\nabla\cdot(\bar{\chi}\bar{\rho}\nabla\bar{c})\big). \tag{16}$$

Here we develop second-order integrators for the deterministic overdamped low Mach equation (15)+(16).

In the stochastic setting, the solution of (13) is white in time because the stochastic mass and momentum fluxes are white in time. This means that the advective term $\boldsymbol{v}\cdot\nabla c$ requires a specific stochastic interpretation, in addition to the usual regularization (smoothing) in space required to interpret all nonlinear terms appearing in formal fluctuating hydrodynamics SPDEs. By performing a precise (albeit formal) adiabatic

[1] More generally, in the presence of inhomogeneous boundary conditions, the solution operator for (14) is an affine rather than a linear operator.

mode elimination of the fast velocity variable under the assumption of infinite separation of time scales, Donev et al. arrive at a Stratonovich interpretation of the random advection term $\boldsymbol{v}\cdot\nabla c$ (see Appendix A of Ref. [27]). This analysis does not, however, directly extend to the low Mach number equations since it relies in key ways on the incompressibility of the fluid. Generalizing this sort of analysis to the case of variable fluid density is nontrivial, likely requiring the use of the gauge formulation of the low Mach equations, and appears to be beyond the scope of existing techniques. Variable (i.e., concentration-dependent) viscosity and mass diffusion coefficients can be handled using existing techniques although there are subtle nonlinear stochastic effects arising from the fact that the noise in the velocity equation is multiplicative and the invariant measure (equilibrium distribution) of the fast velocity depends on the slow concentration.

In the linearized setting, however, the difficulties associated with the interpretation of stochastic integrals and multiplicative noise disappear. The complete form of the linearized equations contains many terms and is rather tedious. Since we will never need to explicitly write this form let us illustrate the procedure by assuming χ and η to be constant. For the concentration, we obtain the linearized equation

$$\bar{\rho}(\partial_t(\delta c)+(\delta\boldsymbol{v})\cdot\nabla\bar{c}) = \nabla\cdot\left(\bar{\rho}\chi\nabla(\delta c)+\bar{\rho}'\chi(\nabla\bar{c})\delta c\right)-\bar{\rho}^{-1}\bar{\rho}'\nabla\cdot(\bar{\rho}\chi\nabla\bar{c})\delta c, \quad (17)$$

where $\bar{\rho}' = d\rho(\bar{c})/dc = \bar{\rho}\beta(\bar{\rho})$ relates concentration fluctuations to density fluctuations via the EOS. Here we split $\delta\boldsymbol{v} = \delta\boldsymbol{v}_c + \delta\boldsymbol{v}_f$ into a component $\delta\boldsymbol{v}_c$ that is continuous in time and a component $\delta\boldsymbol{v}_f$ that is white in time,

$$\delta\boldsymbol{v}_c = \mathcal{L}^{-1}[\bar{\eta}]\left(\bar{\rho}'\boldsymbol{g}\delta c,\ (\bar{\rho}_2^{-1}-\bar{\rho}_1^{-1})\nabla\cdot(\bar{\rho}\chi\nabla(\delta c))\right),$$
$$\delta\boldsymbol{v}_f = \mathcal{L}^{-1}[\bar{\eta}](\nabla\cdot\boldsymbol{\Sigma},\ \nabla\cdot\boldsymbol{\Psi}).$$

The term $\bar{\rho}(\delta\boldsymbol{v}_f)\cdot\nabla\bar{c}$ in (17) is interpreted as an additive noise term with a rather complicated and potentially time-dependent (via $\bar{\eta}(\bar{c}(\boldsymbol{r},t))$) spatial correlation structure. In this work we develop numerical methods that solve the overdamped linearized Equation (17) to second-order weakly [24].

III. Spatiotemporal discretization

Our baseline spatiotemporal discretization of the low Mach equations is based on the method of lines approach where we first discretize the (S)PDEs in space to obtain a system of (S)ODEs, which we then solve using a single-step multistage temporal integrator. The conservative finite-volume spatial discretization that we employ here is essentially identical to that developed in our previous works, Refs. [28; 13]. In summary, scalar fields such as concentration and densities are cell-centered, while velocity is face-centered. In order to ensure conservation, the conserved momentum $\rho\boldsymbol{v}$ and mass densities ρ_1 and ρ are evolved rather than the primitive variables $\boldsymbol{v}$

and c. Diffusion of mass and momentum is discretized using standard centered differences, leading to compact stencils similar to the standard Laplacian. Stochastic mass fluxes are associated with the faces of the regular grid, while for stochastic momentum fluxes we associate the diagonal elements with the cell centers and the off-diagonal elements with the nodes (in 2D) or edges (in 3D) of a regular grid with grid spacing Δx.

Here we focus our discussion on three new aspects of our spatiotemporal discretization. After summarizing the dimensionless numbers that control the appropriate choice of advection method and temporal integrator, in Section B we describe our implementation of two advection schemes and a discussion of the advantages of each. In Section C we describe our implicit treatment of viscous dissipation using a GMRES solver for the coupled velocity-pressure Stokes system. In Section D we describe our overall temporal discretization strategies for the inertial and overdamped regimes.

A. ***Dimensionless numbers.*** The suitability of a particular temporal integrator or advection scheme depends on the following dimensionless numbers:

$$\text{cell Reynolds number } \mathrm{Re}_c = \frac{U\Delta x}{\nu},$$

$$\text{cell Péclet number } \mathrm{Pe}_c = \frac{U\Delta x}{\chi},$$

$$\text{Schmidt number } \mathrm{Sc} = \frac{\nu}{\chi} = \frac{\mathrm{Pe}}{\mathrm{Re}},$$

where $\nu = \eta/\rho$ is the kinematic viscosity. Observe that the first two depend on the spatial resolution and the typical flow speed U, while the Schmidt number is an intrinsic material property of the mixture. Also note that the physically relevant Reynolds Re and Péclet Pe numbers would be defined with a length scale much larger than Δx, such as the system size, and thus would be much larger than the discretization-scale numbers above. In this work, we are primarily interested in small-scale flows with $\mathrm{Re}_c \lesssim 1$ and large Sc (liquid mixtures).

The choice of advection scheme for concentration (partial densities) is dictated by Pe_c. If $\mathrm{Pe}_c \gtrsim 2$, centered advection schemes will generate nonphysical oscillations, and one must use the Godunov advection scheme described below. However, it is important to note that in this case the spectrum of fluctuations will not be correctly preserved by the advection scheme; if fluctuations need to be resolved it is advisable to instead reduce the grid spacing and thus reduce the cell Péclet number to $\mathrm{Pe}_c \lesssim 2$ and use centered advection.

The choice of the temporal integrator, on the other hand, is determined by the importance of inertia and the time scale of interest. If Re_c is not sufficiently small, then there is no alternative to resolving the inertial dynamics of the velocity. Now

let us assume that $Re \ll 1$, i.e., viscosity is dominant. If the time scale of interest is the advective timescale L/U, where L is the system size, then one should use the inertial equations. However, the inertial temporal integrator described in Section III D will be rather inefficient if the time scale of interest is the diffusive time scale L^2/χ, as is the case in the study of diffusive mixing presented in Section V. This is because the Crank–Nicolson (implicit midpoint) scheme used to treat viscosity in our methods is only A-stable, and, therefore, if the viscous Courant number $\nu\Delta t/\Delta x^2$ is too large, unphysical oscillations in the solution will appear (note that this problem is much more serious for fluctuating hydrodynamics due to the presence of fluctuations at *all* scales). In order to be able to use a time step size on the diffusive time scale, one must construct a *stiffly accurate* temporal integrator. This requires using an L-stable scheme to treat viscosity, such as the backward Euler scheme, which is however only first-order accurate.

Constructing a second-order stiffly accurate implicit-explicit integrator in the context of variable density low Mach flows is rather nontrivial. Furthermore, using an L-stable scheme leads to a damping of the velocity fluctuations at large wavenumbers and is inferior to the implicit midpoint scheme in the context of fluctuating hydrodynamics [23]. Therefore, in this work we choose to consider separately the overdamped limit $\mathrm{Re} \to 0$ and $\mathrm{Sc} \to \infty$ (note that the value of Pe is arbitrary). In this limit we analytically eliminate the velocity as an independent variable, leaving only the concentration equation, which evolves on the diffusive time scale. We must emphasize, however, that the overdamped equations should be used with caution, especially in the presence of fluctuations. Notably, the validity of the overdamped approximation requires that the separation of time scales between the fast velocity and slow concentration be uniformly large over *all* wavenumbers, since fluctuations are present at *all* length scales. In the study of giant fluctuations we present in Section V, buoyancy effects speed up the dynamics of large-scale concentration fluctuations and using the overdamped limit would produce physically incorrect results at small wavenumbers. In microgravity, however, the overdamped limit is valid and we have used it to study giant fluctuations over very long time scales in a number of separate works [14; 27].

B. ***Advection.*** We have implemented two advection schemes for cell-centered scalar fields, and describe under what conditions each is more suitable. The first is a simpler nondissipative centered advection discretization described in our previous work [28]. This scheme preserves the skew-adjoint nature of advection and thus maintains fluctuation-dissipation balance in the stochastic context. However, when sharp gradients are present, centered advection schemes require a sufficient amount of dissipation (diffusion) in order to avoid the appearance of Gibbs-phenomenon instabilities. Higher-order Godunov schemes have been used successfully with

cell-centered finite volume schemes for some time [8; 9; 41; 44]. In these semi-Lagrangian advection schemes, a construction based on characteristics is used to estimate the average value of the advected quantity passing through each cell face during a time step. These averages are then used to evaluate the advective fluxes. Our second scheme for advection is the higher-order Godunov approach of Bell, Dawson, and Shubin (BDS) [9]. Additional details of this approach are provided in the next subsection.

The BDS scheme can only be used to advect cell-centered scalar fields such as densities. This is because the scheme operates on control volumes, and therefore applying it to staggered fields requires the use of disjoint control volumes, thereby greatly complicating the advection procedure for non-cell-centered data. We therefore limit ourselves to using the skew-adjoint centered advection scheme described in Refs. [6; 23] to advect momentum. Although some Godunov schemes for advecting a staggered momentum field have been developed [53; 35], they are not at the same level of sophistication as those for cell-centered scalar fields. For example, in Ref. [53] a piecewise constant reconstruction is used, and in Ref. [35] extrapolation is performed in space only, and not in time. In our target applications, there is sufficient viscous dissipation to stabilize centered advection of momentum (note that the mass diffusion coefficient is several orders of magnitude smaller than the kinematic viscosity in typical liquids).

The BDS advection scheme is not skew-adjoint and thus adds some dissipation in regions of sharp gradients that are not resolved by the underlying grid. Thus, unlike the case of using centered advection, the spatially discrete (but still continuous in time) fluctuating equations do not obey a strict discrete fluctuation-dissipation principle [30; 23]. Nevertheless, in high-resolution schemes such as BDS artificial dissipation is added locally in regions where centered advection would have failed completely due to insufficient spatial resolution. Furthermore, the BDS scheme offers many advantages in the deterministic context and allows us to simulate high Péclet number flows with little to no mass diffusion. For well-resolved flows with sufficient dissipation there is little difference between BDS and centered advection. Note that both advection schemes are spatially second-order accurate for smooth flows.

1. *BDS advection.* Simple advection schemes, such as the centered scheme described in our previous work [28], directly computes the divergence of the advective flux $\boldsymbol{f} = \phi \boldsymbol{v}$ evaluated at a specific point in time, where ϕ is a cell-centered quantity such as density, and $\boldsymbol{v}$ is a specified face-centered velocity. By contrast, the BDS scheme uses the multidimensional characteristic geometry of the advection equation

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \boldsymbol{v}) = q, \tag{18}$$

to estimate time-averaged fluxes through cell faces over a time interval Δt, given ϕ^n, as well as a face-centered velocity field $\boldsymbol{v}$ and a cell-centered source q that are assumed *constant* over the time interval. In actual temporal discretizations $\boldsymbol{v} \approx \boldsymbol{v}(t^{n+1/2}) \approx \boldsymbol{v}(t^n + \Delta t/2)$ is a midpoint (second-order) approximation of the velocity over the time step. Similarly, $q \approx q(t^{n+1/2})$ will be a centered approximation of the divergence of the diffusive and stochastic fluxes over the time step. In the description of our temporal integrators, we will use the shorthand notation BDS to denote the approximation to the advective fluxes used in the BDS scheme for solving (18),

$$\phi^{n+1} = \phi^n - \Delta t\, \nabla \cdot (\mathrm{BDS}(\phi^n,\ \boldsymbol{v},\ q,\ \Delta t)) + \Delta t\, q.$$

BDS is a conservative scheme based on computing time-averaged advective fluxes through every face of the computational grid, for example, in two dimensions,

$$\mathrm{BDS}_{i+1/2,j} = f_{i+1/2,j} = \phi_{i+1/2,j} v_{i+1/2,j},$$

where $v_{i+1/2,j}$ is the given normal velocity at the face, and $\phi_{i+1/2,j}$ represents the space-time average of ϕ passing through face-$(i+1/2,\ j)$ in the time interval Δt. The extrapolated face-centered states $\phi_{i+1/2,j}$ are computed by first reconstructing a piecewise continuous profile of $\phi(\boldsymbol{r}, t)$ in every cell that can, optionally, be limited based on monotonicity considerations. The multidimensional characteristic geometry of the flow in space-time is then used to estimate the time-averaged flux; see the original papers [9; 41; 44] for a detailed description. In the original advection BDS schemes in two dimensions [9] and three dimensions [44], a piecewise-bilinear (in two dimensions) or trilinear (in three dimensions) reconstruction of ϕ was used. Subsequently, the schemes were extended to a quadratic reconstruction in two dimensions [41]. Note that handling boundary conditions in BDS properly requires additional investigations, and the construction of specialized one-sided reconstruction stencils near boundaries. In our implementation we rely on cubic extrapolation based on interior cells and the specified boundary condition (Dirichlet or Neumann) to fill ghost cell values behind physical boundaries, and then apply the BDS procedure to the interior cells using the extrapolated ghost cell values.

BDS advection, as described in [9; 41; 44], does not strictly preserve the EOS constraint, unlike centered advection. The characteristic extrapolation of densities to space-time midpoint values on the faces of the grid, $(\rho_1)_{i+1/2,j}$ and $\rho_{i+1/2,j}$, are not necessarily consistent with the EOS, unlike centered advection where they are simple averages of values from neighboring cells, and thus guaranteed to obey the EOS by linearity. A simple fix that makes BDS preserve the EOS, without affecting its formal order of accuracy, is to enforce the EOS on each face by projecting the extrapolated values $(\rho_1)_{i+1/2,j}$ and $\rho_{i+1/2,j}$ onto the EOS. In the L_2 sense, such a

projection consists of the update

$$(\rho_1)_{i+1/2,j} \leftarrow \frac{\bar{\rho}_1^2}{\bar{\rho}_1^2+\bar{\rho}_2^2}(\rho_1)_{i+1/2,j} - \frac{\bar{\rho}_1\bar{\rho}_2}{\bar{\rho}_1^2+\bar{\rho}_2^2}(\rho_2)_{i+1/2,j},$$

and similarly for ρ_2, or equivalently, $\rho = \rho_1 + \rho_2$. Note that this projection is done on each face only for the purposes of computing advective fluxes and is distinct from any projection onto the EOS performed globally.

C. ***GMRES solver.*** The temporal discretization described in our previous work [28] was fully explicit, whereas the discretization we employ here is implicit in the viscous dissipation. The implicit treatment of viscosity is traditionally handled by time-splitting approaches, in which a velocity system is solved first, without strictly enforcing the constraint. The solution is then projected onto the space of vector fields satisfying the constraint [16]. This type of time-splitting introduces several artifacts, especially for viscous-dominated flows; here we avoid time-splitting by solving a combined velocity-pressure Stokes linear system, as discussed in detail in Ref. [13].

The implicit treatment of viscosity in the temporal integrators described in Section III D requires solving discretized unsteady Stokes equations for a velocity $\boldsymbol{v}$ and a pressure π,

$$\theta\rho\boldsymbol{v} - \boldsymbol{\nabla}\cdot(\eta\overline{\boldsymbol{\nabla}}\boldsymbol{v}) + \boldsymbol{\nabla}\pi = \boldsymbol{f},$$
$$\boldsymbol{\nabla}\cdot\boldsymbol{v} = h,$$

for given spatially varying density ρ and viscosity η, right-hand sides $\boldsymbol{f}$ and h, and a coefficient $\theta \geq 0$. We solve these linear systems using a GMRES Krylov solver preconditioned by the multigrid-based preconditioners described in detail in Ref. [13]. This approach requires only standard velocity (Helmholtz) and pressure (Poisson) multigrid solvers, and requires about two to three times more multigrid iterations than solving an uncoupled pair of velocity and pressure subproblems (as required in projection-based splitting methods).

There are two issues that arise with the Stokes solver in the context of temporal integration that need special care. In fluctuating hydrodynamics, typically the average flow $\bar{\boldsymbol{v}}$ changes slowly and is much larger in magnitude than the fluctuations around the flow $\delta\boldsymbol{v}$. In the predictor stages of our temporal integrators the convergence criterion in the GMRES solver is based on relative tolerance. Because the right-hand side of the linear system and the residual are dominated by the deterministic flow, it is hard to determine when the fluctuating component of the flow has converged to the desired relative accuracy. In the corrector stage of our predictor-corrector schemes, we use the predicted state as a reference, and switch to using absolute error as the convergence criterion in GMRES, using the same residual error tolerance as was used in the predictor stage. This ensures that the corrector stage GMRES

converges quickly if the predicted state is already a sufficiently accurate solution of the Stokes system. Another issue that has to be handled carefully is the imposition of inhomogeneous boundary conditions, which leads to a linear system of the form

$$\boldsymbol{A}\boldsymbol{x}^{\text{new}} + \boldsymbol{b}_{BC} = \boldsymbol{b},$$

where $\boldsymbol{b}_{BC}$ comes from nonhomogeneous boundary conditions. Both of these problems are solved by using a residual correction technique to convert the Stokes linear system into one for the *change* in the velocity and pressure $\Delta\boldsymbol{x} = \boldsymbol{x}^{\text{new}} - \bar{\boldsymbol{x}}^{\text{old}}$ relative to an initial guess or *reference* state $\bar{\boldsymbol{x}}^{\text{old}}$, which is typically the last known velocity and pressure, *except* that the desired inhomogeneous boundary conditions are imposed; this ensures that boundary terms vanish and the Stokes problem for $\Delta\boldsymbol{x}$ is in homogeneous form. Note that any Dirichlet boundary conditions for the normal component of velocity should be consistent with h, and any Dirichlet boundary conditions for the tangential component of the velocity should be evaluated at the same point in time (e.g., beginning, midpoint, or endpoint of the time step) as h.

D. ***Temporal discretization.*** In this section we construct temporal integrators for the spatially discretized low Mach number equations, in which we treat viscosity semi-implicitly. For our target applications, the Reynolds number is sufficiently small and the Schmidt number is sufficiently large that an explicit viscosity treatment would lead to an overall viscous time step restriction,

$$\frac{\eta\Delta t}{\Delta x^2} < \frac{1}{2d},$$

We present temporal integrators in which we avoid fractional time stepping and ensure *strict* (to within solver and roundoff tolerances) conservation and preservation of the EOS constraint. The key feature of the algorithms developed here is the implicit treatment of viscous dissipation, without, however, using splitting between the velocity and pressure updates, as discussed at more length in the introduction. The feasibility of this approach relies an efficient solver for Stokes systems on a staggered grid [13]; see also Section III C for additional details.

In the temporal integrators developed here, we treat advection explicitly, which limits the advective Courant number to

$$\frac{v_{\max}\Delta t}{\Delta x} < C \sim 1.$$

Mass diffusion is also treated explicitly since it is typically much slower than momentum diffusion and in many examples also slower than advection. Explicit treatment of mass diffusion leads to an additional stability limit on the time step

since the diffusive Courant number must be sufficiently small,

$$\frac{\chi \Delta t}{\Delta x^2} < \frac{1}{2d},$$

where d is the number of spatial dimensions and Δx is the grid spacing. Implicit treatment of mass diffusion is straightforward for incompressible flows, see Algorithm 2 in Ref. [24], but is much harder for the low Mach number equations due to the need to maintain the EOS constraint (3) via the constraint (9). Even with explicit mass diffusion, provided that the Reynolds is sufficiently small and the Schmidt number is sufficiently large, a semi-implicit viscosity treatment results in a much larger allowable time step.

1. *Predictor-corrector time stepping schemes.* In Algorithm 1 we give the steps involved in advancing the solution from time level n by a time interval Δt to time level $n+1$, using a semi-implicit trapezoidal temporal integrator [24] for the inertial fluctuating low Mach number equations (6), (7), (9) and (11). In Algorithm 2 we give an explicit midpoint temporal integrator [24] for the overdamped low Mach number equations (7), (11) and (13).

In order to ensure strict conservation of mass and momentum, we evolve the momentum density $\boldsymbol{m} = \rho \boldsymbol{v}$ and the mass densities ρ_1 and ρ (an equally valid choice is to evolve ρ_1 and ρ_2). Whenever required, the primitive variables $\boldsymbol{v} = \boldsymbol{m}/\rho$ and $c = \rho_1/\rho$ are computed from the conserved quantities. Unlike the incompressible equations, the low Mach number equations require the enforcement of the EOS constraint (3) at every update of the mass densities ρ_1 and ρ, notably, both in the predictor and the corrector stages. This requires that the right-hand side of the velocity constraint (9) be consistent with the corresponding diffusive fluxes used to update ρ_1. In order to preserve the EOS and also maintain strict conservation, Algorithms 1 and 2 use a splitting approach, in which we first update the mass densities and then we update the velocity using the updated values for the density ρ and the diffusive fluxes that will be used to update ρ_1. Note, however, that after many time steps the small errors in enforcing the EOS due to roundoff and solver tolerances can accumulate and lead to a systematic drift from the EOS. This can be corrected by periodically projecting the solution back onto the EOS using an L_2 projection, see Section III.C in Ref. [28].

In our presentation of the temporal integrators, we use superscripts to denote where a given quantity is evaluated, for example, $\eta^{p,n+1} \equiv \eta(c^{p,n+1})$. Even though we use continuum notation for the divergence, gradient and Laplacian operators, it is implicitly understood that the equations have been discretized in space. The white-noise random tensor fields $\mathcal{W}(\boldsymbol{r}, t)$ and $\widetilde{\mathcal{W}}(\boldsymbol{r}, t)$ are represented via one or two collections of i.i.d. uncorrelated normal random variables $\boldsymbol{W}$ and $\widetilde{\boldsymbol{W}}$, generated independently at each time step, as indicated by superscripts and subscripts [23;

24]. Spatial discretization adds an additional factor of $\Delta V^{-1/2}$ due to the delta function correlation of white-noise, where ΔV is the volume of a grid cell [23]. For simplicity of notation we set $\overline{\boldsymbol{W}} = \boldsymbol{W} + \boldsymbol{W}^T$.

Several variants of the inertial Algorithm 1 preserve deterministic second-order accuracy. For example, in the corrector stage for ρ_1, for centered advection we use a trapezoidal approximation to the advective flux,

$$\tfrac{1}{2}(\rho_1 \boldsymbol{v})^n + \tfrac{1}{2}(\rho_1 \boldsymbol{v})^{\star,n+1}, \tag{19}$$

but we could have also used a midpoint approximation

$$\left(\frac{\rho_1^n + \rho_1^{\star,n+1}}{2}\right)\left(\frac{\boldsymbol{v}^n + \boldsymbol{v}^{\star,n+1}}{2}\right). \tag{20}$$

without affecting the second-order weak accuracy [24]. Note that BDS advection by construction requires a midpoint approximation to the advective velocity; no analysis of the order of stochastic accuracy is available for BDS advection at present. In the corrector step for velocity, in Algorithm 1 we use corrected values for the viscosity, but one can also use the values from the predictor $\eta^{*,n+1}$.

2. *Order of accuracy.* For explicit temporal integrators, we relied on a gauge formulation to write the low Mach equations in the form of a standard unconstrained initial-value problem, thus allowing us to use standard integrators for ODEs [28]. In the semi-implicit case, however, we do not use a gauge formulation because the Stokes solver we use works directly with the pressure and velocity. This makes proving second-order temporal accuracy nontrivial even in the deterministic context; we therefore rely on empirical convergence testing to confirm the second-order deterministic accuracy.

In the stochastic context, there is presently no available theoretical analysis when BDS advection is employed; existing analysis [30; 23; 24] assumes a method of lines (MOL) discretization in which space is discretized first to obtain a system of SODEs. For centered advection, which does lead to an MOL discretization, the algorithms used here are based on the second-order weak temporal integrators developed in Ref. [24]. In particular, for the case of the inertial equations (6), (7), (9) and (11), we base our temporal integrator on an implicit trapezoidal method. It should be emphasized however that the analysis in Ref. [24] applies to unconstrained Langevin systems, while the low Mach equations are constrained by the EOS. Nevertheless, the deterministic accuracy of the method is crucial even when fluctuations are of primary interest, because in linearized fluctuating hydrodynamics the fluctuations are linearized around the solution of the deterministic equations, which must itself be computed numerically accurately [24] in order to have any chance of computing the fluctuations accurately. For the case of the overdamped equations (7), (11) and (13), we base our temporal integrator on an implicit midpoint method. In

1. Compute the diffusive / stochastic fluxes for the predictor. Note that these can be obtained from step 5. of the previous time step,

$$\boldsymbol{F}^n = (\rho\chi\nabla c)^n + \sqrt{\frac{2(\chi\rho\mu_c^{-1})^n k_B T}{\Delta t\,\Delta V}}\,\widetilde{\boldsymbol{W}}^n.$$

2. Take a predictor forward Euler step for ρ_1, and similarly for ρ_2, or, equivalently, for ρ,

$$\rho_1^{\star,n+1} = \rho_1^n + \Delta t\,\nabla\cdot\boldsymbol{F}^n - \Delta t\,\nabla\cdot\begin{cases}\mathrm{BDS}(\rho_1^n,\ \boldsymbol{v}^n,\ \nabla\cdot\boldsymbol{F}^n,\ \Delta t) & \text{for BDS,}\\ \rho_1^n\boldsymbol{v}^n & \text{for centered.}\end{cases}$$

3. Compute $c^{\star,n+1} = \rho_1^{\star,n+1}/\rho^{\star,n+1}$ and calculate corrector diffusive fluxes and stochastic fluxes,

$$\boldsymbol{F}^{\star,n+1} = (\rho\chi\nabla c)^{\star,n+1} + \sqrt{\frac{2(\chi\rho\mu_c^{-1})^{\star,n+1}k_B T}{\Delta t\,\Delta V}}\,\widetilde{\boldsymbol{W}}^n.$$

4. Take a predictor Crank–Nicolson step for the velocity, using $\boldsymbol{v}^n$ as a reference state for the residual correction form of the Stokes system,

$$\begin{aligned}&\frac{\rho^{\star,n+1}\boldsymbol{v}^{\star,n+1} - \rho^n\boldsymbol{v}^n}{\Delta t} + \nabla\pi^{\star,n+1}\\ &\qquad = \nabla\cdot(-\rho\boldsymbol{v}\boldsymbol{v})^n + \rho^n\boldsymbol{g} + \tfrac{1}{2}\nabla\cdot(\eta^n\overline{\nabla}\boldsymbol{v}^n + \eta^{\star,n+1}\overline{\nabla}\boldsymbol{v}^{\star,n+1}) + \nabla\cdot\left(\sqrt{\frac{\eta^n k_B T}{\Delta t\,\Delta V}}\,\overline{\boldsymbol{W}}^n\right),\\ &\nabla\cdot\boldsymbol{v}^{\star,n+1} = -\nabla\cdot(\beta\rho^{-1}\boldsymbol{F}^{\star,n+1}).\end{aligned}$$

Take a corrector step for ρ_1, and similarly for ρ_2, or, equivalently, for ρ,

$$\rho_1^{n+1} = \rho_1^n + \frac{\Delta t}{2}\nabla\cdot\boldsymbol{F}^{\star,n+1/2} - \Delta t\,\nabla\cdot\begin{cases}\mathrm{BDS}(\rho_1^n,\ \boldsymbol{v}^{\star,n+1/2},\ \nabla\cdot\boldsymbol{F}^{\star,n+1/2},\ \Delta t) & \text{for BDS,}\\ \frac{1}{2}(\rho_1\boldsymbol{v})^n + \frac{1}{2}(\rho_1\boldsymbol{v})^{\star,n+1} & \text{for centered,}\end{cases}$$

where $\boldsymbol{F}^{\star,n+1/2} = (\boldsymbol{F}^n + \boldsymbol{F}^{\star,n+1})/2$ and $\boldsymbol{v}^{\star,n+1/2} = (\boldsymbol{v}^n + \boldsymbol{v}^{\star,n+1})/2$.

5. Compute $c^{n+1} = \rho_1^{n+1}/\rho^{n+1}$ and compute

$$\boldsymbol{F}^{n+1} = (\rho\chi\nabla c)^{n+1} + \sqrt{\frac{2(\chi\rho\mu_c^{-1})^{n+1}k_B T}{\Delta t\,\Delta V}}\,\widetilde{\boldsymbol{W}}^{n+1}.$$

6. Take a corrector step for velocity by solving the Stokes system, using $\boldsymbol{v}^{\star,n+1}$ as a reference state,

$$\begin{aligned}&\frac{\rho^{n+1}\boldsymbol{v}^{n+1} - \rho^n\boldsymbol{v}^n}{\Delta t} + \nabla\pi^{n+1/2}\\ &\qquad = \tfrac{1}{2}\nabla\cdot((-\rho\boldsymbol{v}\boldsymbol{v})^n + (-\rho\boldsymbol{v}\boldsymbol{v})^{\star,n+1}) + \tfrac{1}{2}(\rho^n + \rho^{n+1})\boldsymbol{g} + \tfrac{1}{2}\nabla\cdot(\eta^n\overline{\nabla}\boldsymbol{v}^n + \eta^{n+1}\overline{\nabla}\boldsymbol{v}^{n+1})\\ &\qquad\qquad + \frac{1}{2}\nabla\cdot\left[\left(\sqrt{\frac{\eta^n k_B T}{\Delta t\,\Delta V}} + \sqrt{\frac{\eta^{n+1}k_B T}{\Delta t\,\Delta V}}\right)\overline{\boldsymbol{W}}^n\right],\\ &\nabla\cdot\boldsymbol{v}^{n+1} = -\nabla\cdot(\beta\rho^{-1}\boldsymbol{F}^{n+1}).\end{aligned}$$

Algorithm 1. Semi-implicit trapezoidal temporal integrator for the inertial fluctuating low Mach number equations (6), (7), (9) and (11).

1. Calculate predictor diffusive fluxes and generate stochastic fluxes for a half step to the midpoint,
$$\boldsymbol{F}^n = (\rho\chi\nabla c)^n + \sqrt{\frac{2(\chi\rho\mu_c^{-1})^n k_B T}{(\Delta t/2)\,\Delta V}}\,\widetilde{\boldsymbol{W}}_A^n.$$
2. Generate a random advection velocity by solving the steady Stokes equation with random forcing,
$$\nabla\pi^n = \nabla\cdot(\eta^n\overline{\nabla}\boldsymbol{v}^n) + \nabla\cdot\left(\sqrt{\frac{\eta^n k_B T}{(\Delta t/2)\,\Delta V}}\,\overline{\boldsymbol{W}}_A^n\right) + \rho^n\boldsymbol{g}$$
$$\nabla\cdot\boldsymbol{v}^n = -\nabla\cdot(\beta\rho^{-1}\boldsymbol{F}^n).$$
3. Take a predictor midpoint Euler step for ρ_1, and similarly for ρ_2, or, equivalently, for ρ,
$$\rho_1^{\star,n+1/2} = \rho_1^n + \frac{\Delta t}{2}\nabla\cdot\boldsymbol{F}^n - \frac{\Delta t}{2}\nabla\cdot\begin{cases}\mathrm{BDS}(\rho_1^n,\ \boldsymbol{v}^n,\ \nabla\cdot\boldsymbol{F}^n,\ \frac{\Delta t}{2}) & \text{for BDS,}\\ \rho_1^n\boldsymbol{v}^n & \text{for centered,}\end{cases}$$
and compute $c^{\star,n+1/2} = \rho_1^{\star,n+1/2}/\rho^{\star,n+1/2}$.
4. Calculate corrector diffusive fluxes and generate stochastic fluxes,
$$\boldsymbol{F}^{\star,n+1/2} = (\rho\chi\nabla c)^{\star,n+1/2} + \sqrt{\frac{2(\chi\rho\mu_c^{-1})^{\star,n+1/2}k_B T}{\Delta t\,\Delta V}}\left(\frac{\widetilde{\boldsymbol{W}}_A^n + \widetilde{\boldsymbol{W}}_B^n}{\sqrt{2}}\right),$$
where $\widetilde{\boldsymbol{W}}_B^n$ is a collection of random numbers generated independently of $\widetilde{\boldsymbol{W}}_A^n$.
5. Solve the corrected steady Stokes equation
$$\nabla\pi^{\star,n+1/2} = \nabla\cdot(\eta^{\star,n+1/2}\overline{\nabla}\boldsymbol{v}^{\star,n+1/2}) + \nabla\cdot\left[\sqrt{\frac{\eta^{\star,n+1/2}k_B T}{\Delta t\,\Delta V}}\left(\frac{\overline{\boldsymbol{W}}_A^n + \overline{\boldsymbol{W}}_B^n}{\sqrt{2}}\right)\right] + \rho^{\star,n+1/2}\boldsymbol{g}$$
$$\nabla\cdot\boldsymbol{v}^{\star,n+1/2} = -\nabla\cdot(\beta\rho^{-1}\boldsymbol{F}^{\star,n+1/2}).$$
6. Correct ρ_1, and similarly for ρ_2, or, equivalently, for ρ,
$$\rho_1^{n+1} = \rho_1^n + \Delta t\,\nabla\cdot\boldsymbol{F}^{\star,n+1/2} - \Delta t\,\nabla\cdot\begin{cases}\mathrm{BDS}(\rho_1^n,\ \boldsymbol{v}^{\star,n+1/2},\ \nabla\cdot\boldsymbol{F}^{\star,n+1/2},\ \Delta t) & \text{for BDS,}\\ (\rho\boldsymbol{v})^{\star,n+1/2} & \text{for centered,}\end{cases}$$
and set $c^{n+1} = \rho_1^{n+1}/\rho^{n+1}$.

Algorithm 2. A time step of our implicit midpoint temporal integrator for the overdamped equations (7), (11) and (13).

this case the analysis presented in Ref. [24] does apply since the velocity is not a variable in the overdamped equations and the limiting equation for concentration is unconstrained. This analysis indicates that the overdamped temporal integrator in Algorithm 2 is second-order weakly accurate for the *linearized* overdamped low Mach number equations.

IV. Validation and testing

In this section we apply the inertial and overdamped low Mach algorithms described in Section III in a stochastic and several deterministic contexts. First, we demonstrate our ability to accurately model equilibrium fluctuations by analyzing the static spectrum of the fluctuations. Next, we confirm the second-order deterministic order of accuracy of our methods on a low Mach number lid-driven cavity test. Next, we confirm that the BDS advection scheme enables robust simulation in cases when there is little or no mass diffusion (i.e., nearly infinite Péclet number). Lastly, we use the inertial algorithm to simulate the development of a Kelvin–Helmholtz instability when a lighter less viscous fluid is impulsively set in motion on top of a heavier more viscous fluid.

A. ***Equilibrium fluctuations.*** One of the key quantities used to characterize the intensity of *equilibrium* thermal fluctuations is the static structure factor or static spectrum of the fluctuations at thermodynamic equilibrium. We examine the static structure factors in both the inertial and overdamped regimes. We use arbitrary units with $T = 1$, $k_B = 1$, molecular masses $m_1 = 1$, $m_2 = 2$, and pure component densities $\bar{\rho}_1 = 2/3$, $\bar{\rho}_2 = 2$. We initialize the domain with $c = 0.5$, which gives $\rho = 1$. The diffusion coefficient was constant $\chi = 1$, whereas the viscosity varies linearly from $\eta = 1$ to $\eta = 10$ (for the inertial tests), and from $\eta = 1$ to $\eta = 100$ (for the overdamped tests) as c varies from 0 to 1, but note that at equilibrium the concentration fluctuations are small so the viscosity varies little over the domain. We assume an ideal mixture, giving chemical potential $\mu_c^{-1} k_B T = c(1-c) \times [cm_2 + (1-c)m_1]$ (see Refs. [28; 10]). At these conditions, the equilibrium density variance is $\Delta V \langle (\delta\rho)^2 \rangle = S_\rho = 0.375$, where ΔV is the volume of a grid cell (see Appendix A1 in Ref. [28]). We use a periodic system with 32×32 grid cells with $\Delta x = \Delta y = 1$, with the thickness in the third direction set to give a large $\Delta V = 10^6$ and thus small fluctuations, ensuring consistency with linearized fluctuating hydrodynamics. A total of 10^5 time steps are skipped in the beginning to allow equilibration of the system, and statistics are then collected for an additional 10^6 steps. We run both the inertial and overdamped algorithms using three different time steps, $\Delta t = 0.1$, 0.05, and 0.025, the largest of which corresponds to 40% of the maximum allowable time step by the explicit mass diffusion CFL condition.

In Table 1 we observe that as we reduce the time step by a factor of two, we see a reduction in error in the average value of S_ρ over all wavenumbers by a factor of $\sim$4 (second-order convergence) for the inertial algorithm, and a factor of $\sim$8 (third-order convergence) for the overdamped algorithm (the latter being consistent with the fact that the explicit midpoint method is third-order accurate for static covariances [23]). In Figure 1 we show the spectrum of density fluctuations at equilibrium for three different time step sizes. At thermodynamic equilibrium, the

	Δt	S_ρ	|Error|	Order
Inertial	0.1	0.3201	0.0549	
	0.05	0.3624	0.0126	2.12
	0.025	0.3722	0.0029	2.14
Overdamped	0.1	0.4192	0.0442	
	0.05	0.3786	0.0036	3.63
	0.025	0.3755	0.0005	2.92

Table 1. Equilibrium static structure factor S_ρ averaged over all wavevectors for the inertial and overdamped algorithms using three different time steps. The exact solution from theory is $S_\rho = 0.375$, allowing us to estimate an order of accuracy from the average error over all wavenumbers. Note that there are significant statistical errors present, especially at small wavenumbers, and these make it difficult to reliably estimate the asymptotic order of accuracy empirically when the error is very small (as for the overdamped integrator).

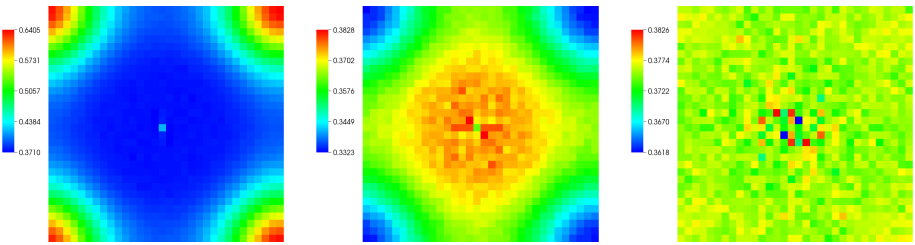


Figure 1. Equilibrium static structure factor S_ρ as a function of wavevector (zero being at the center of the figures) for the overdamped simulations with $\Delta t = 0.1$ (left), $\Delta t = 0.05$ (middle), and $\Delta t = 0.025$ (right). The correct result, which is recovered in the limit $\Delta t \to 0$, is $S_\rho = 0.375$. The artifacts decrease by roughly a factor of 8 as the time step is reduced in half.

static structure factors are independent of the wavenumber due to the local nature of the correlations. Since we include mass diffusion using an explicit temporal integrator, for larger time steps we expect to see additional deviation from a flat spectrum at the largest wavenumbers (i.e., for $k \sim \Delta x^{-1}$) [30; 23]. In the limit of sufficiently small time steps, we recover the correct flat spectrum, demonstrating that our model and numerical scheme obey a fluctuation-dissipation principle.

B. ***Deterministic lid-driven cavity convergence test.*** In this section, we simulate a smooth test problem and empirically confirm deterministic second-order accuracy of Algorithms 1 and 2 even in the presence of boundary conditions, inertial effects, and gravity, as well as nonconstant density, mass diffusion coefficient, and viscosity. The problem is a deterministic lid-driven cavity flow, following previous work by Boyce Griffith for incompressible constant-density and constant-viscosity flow [35].

We use CGS units (centimeters for length, seconds for time, grams for mass). We consider a square (two dimensions) or cubic (three dimensions) domain with

side of length $L = 1$ bounded on all sides by no-slip walls moving with a specified velocity. The bottom and top walls (y-direction) are no-slip walls moving in equal and opposite directions, setting up a circular flow pattern, while the remaining walls are stationary. The top wall has a specified velocity given in two dimensions by

$$u(x,t) = \begin{cases} \frac{1}{4}\left[1+\sin\left(2\pi x - \frac{\pi}{2}\right)\right]\left[1+\sin\left(2\pi t - \frac{\pi}{2}\right)\right], & t < \frac{1}{2}, \\ \frac{1}{2}\left[1+\sin\left(2\pi x - \frac{\pi}{2}\right)\right], & t \geq \frac{1}{2}, \end{cases} \tag{21}$$

and in three dimensions by

$$\begin{aligned} &u(x,z,t) = w(x,z,t) \\ &= \begin{cases} \frac{1}{8}\left[1+\sin\left(2\pi x - \frac{\pi}{2}\right)\right]\left[1+\sin\left(2\pi z - \frac{\pi}{2}\right)\right]\left[1+\sin\left(2\pi t - \frac{\pi}{2}\right)\right], & t < \frac{1}{2}, \\ \frac{1}{4}\left[1+\sin\left(2\pi x - \frac{\pi}{2}\right)\right]\left[1+\sin\left(2\pi z - \frac{\pi}{2}\right)\right], & t \geq \frac{1}{2}. \end{cases} \end{aligned} \tag{22}$$

Note that the wall velocity tapers to zero at the corners in order to regularize the corner singularities [35]; similarly, the velocity smoothly increases with time to its final value in order to avoid potential loss of accuracy due to an impulsive start of the flow. The two liquids have pure-component densities $\bar{\rho}_1 = 2$ and $\bar{\rho}_2 = 1$. The initial conditions are $\boldsymbol{v} = 0$ for velocity, and a Gaussian bump of higher density for the concentration, $c(\boldsymbol{r}, t) = \exp(-75r^2)$, where r is the distance to the center of the domain. The viscosity varies linearly as a function of concentration, such that $\eta = 0.1$ when $c = 0$ and $\eta = 1$ when $c = 1$. Similarly, the mass diffusion coefficient varies linearly as a function of concentration, such that $\chi = 10^{-4}$ when $c = 0$ and $\chi = 10^{-3}$ when $c = 1$. In order to confirm that second-order accuracy is preserved even in the limit of infinite Péclet number if BDS advection is employed, we also perform simulations with $\chi = 0$. Figure 2 illustrates the initial and final (at time $t = 2$) configurations of concentration and velocity in two dimensions.

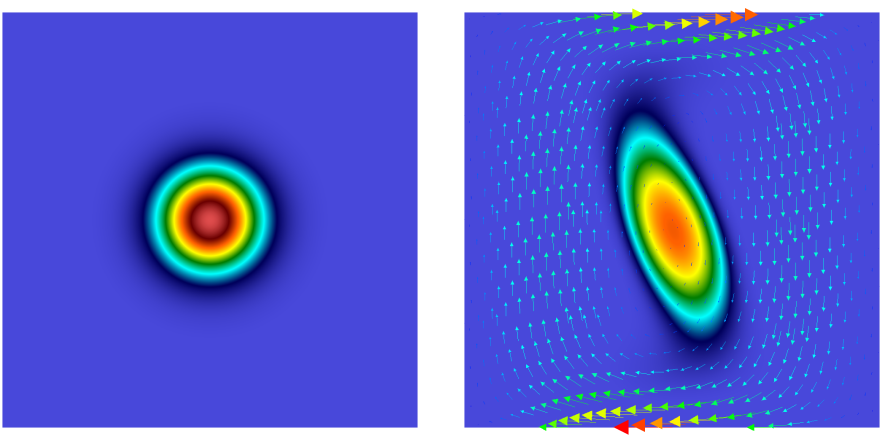


Figure 2. Initial ($t = 0$) and final ($t = 2$) concentration (scalar color field) and velocities (vector field) for the low Mach number lid-driven cavity test problem.

Recall that advection of the concentration can be treated using centered advection or the BDS advection scheme (see Section III B 1). BDS advection can use either a bilinear (trilinear in 3D) or a quadratic reconstruction (2D only), and can further be limited to avoid the appearance of spurious local extrema. Here we present convergence results for the following test problems:

- Test 1: Centered advection, nonzero χ
- Test 2: Unlimited bilinear BDS advection, nonzero χ
- Test 3: Unlimited quadratic BDS advection, nonzero χ
- Test 4: Unlimited bilinear BDS advection, $\chi = 0$.

We perform Tests 1–4 using both the inertial Algorithm 1 and the overdamped Algorithm 2. The Reynolds number in this test is of order unity and there is only a small difference in the results for the inertial and overdamped equations. Recall that concentration is the *only* independent variable in the overdamped equations.

In two dimensions, we discretize the problem on a grid of 64^2, 128^2, 256^2 or 512^2 cells. The time step size for the coarsest simulation is $\Delta t = 5 \times 10^{-3}$ and it is halved as the resolution doubles. This corresponds to an advective Courant number of $v_{\max} \Delta t / \Delta x \sim 0.3$ for each simulation. The diffusive Courant number is $\chi \Delta t / \Delta x^2 \sim 0.16$ (recall that the stability limit is $1/4 = 0.25$ in two dimensions) for the finest simulation, reducing by a factor of 2 with each successive grid coarsening. We simulate the flow and compute error norms at time $t = 2$. In Table 2 we

refinement		64–128	order	128–256	order	256–512
Test 1:	u	1.93×10^{-3}	1.91	5.12×10^{-4}	1.96	1.32×10^{-4}
	v	8.69×10^{-4}	1.99	2.19×10^{-4}	2.00	5.49×10^{-5}
	c	3.02×10^{-4}	1.99	7.60×10^{-4}	2.00	1.90×10^{-4}
Test 2:	u	1.92×10^{-3}	1.91	5.11×10^{-4}	1.95	1.32×10^{-4}
	v	9.08×10^{-4}	1.96	2.34×10^{-4}	1.99	5.91×10^{-5}
	c	2.63×10^{-3}	1.72	7.99×10^{-4}	1.92	2.11×10^{-4}
Test 3:	u	1.92×10^{-3}	1.91	5.11×10^{-4}	1.95	1.32×10^{-4}
	v	8.62×10^{-4}	1.95	2.23×10^{-4}	1.98	5.64×10^{-5}
	c	1.95×10^{-3}	1.99	4.91×10^{-4}	2.00	1.23×10^{-4}
Test 4:	u	1.91×10^{-3}	1.92	5.06×10^{-4}	1.96	1.30×10^{-4}
	v	9.78×10^{-4}	2.01	2.43×10^{-4}	2.02	6.00×10^{-5}
	c	4.29×10^{-3}	1.90	1.15×10^{-3}	1.97	2.93×10^{-4}

Table 2. Convergence of errors in the L_∞ norm for a *two-dimensional inertial* low Mach lid-driven cavity problem as the grid is refined in space and time, for the components of the velocity $\boldsymbol{v} = (u, v)$ and concentration c. The order of convergence is estimated from the error ratio between two successive refinements.

refinement	64–128	order	128–256	order	256–512
Test 1:	3.57×10^{-3}	2.01	8.89×10^{-4}	2.00	2.22×10^{-4}
Test 2:	2.70×10^{-3}	1.78	7.87×10^{-4}	1.92	2.08×10^{-4}
Test 3:	1.95×10^{-3}	1.98	4.95×10^{-4}	1.89	1.34×10^{-4}
Test 4:	4.23×10^{-3}	1.93	1.11×10^{-3}	1.96	2.86×10^{-4}

Table 3. Convergence of errors in the L_∞ norm for a *two-dimensional overdamped* low Mach lid-driven cavity problem as the grid is refined in space and time, for concentration c. The order of convergence is estimated from the error ratio between successive refinements.

present estimates of the order of convergence in the L_∞ (max) norm for the velocity components and concentration for the inertial equations. We see clear second-order pointwise convergence, without any artifacts near the boundaries. Similar results are obtained for the concentration in the overdamped limit, as shown in Table 3.

In three dimensions, we discretize the problem on a grid of 32^3, 64^3, 128^3 or 256^3 cells. The time step size for the coarsest simulation is $\Delta t = 1.25 \times 10^{-2}$, which corresponds to an advective Courant number of ~ 0.4, and diffusive Courant number of ~ 0.10 (stability limit is $1/6 \approx 0.17$) for the finest resolution simulation. We simulate the flow and compute error norms at time $t = 1$. We limit our study here to inertial flow and only perform Tests 1 and 2 (note that there is presently no available unlimited quadratic BDS advection scheme in three dimensions, so test 3 cannot be performed). We also try a higher-order one-sided difference for the tangential velocity at the no-slip boundaries, which does not affect the asymptotic rate of convergence, but it can significantly reduce the magnitude of the errors, and enables us to reach the asymptotic regime for smaller grid sizes [35]. The numerical convergence results shown in Table 4 demonstrate the second-order deterministic accuracy of our method in three dimensions.

C. ***Deterministic sharp interface limit.*** In this section we verify the ability of the BDS advection scheme to advect concentration and density without creating spurious oscillations or instabilities, even in the absence of mass diffusion, $\chi = 0$, and in the presence of sharp interfaces. The problem setup is similar to the inertial lid-driven cavity test presented above, with the following differences. First, the gravity is larger, $\boldsymbol{g} = (0, -5)$, so that the higher density region falls downward a significant distance. Secondly, the initial conditions are a constant background of $c = 0$ with a square region covering the central 25% of the domain initialized to $c = 1$ (see the left panel of Figure 3). The correct solution of the equations must remain a binary field, $c = 1$ inside the advected square curve, and $c = 0$ elsewhere. In this test we employ limited quadratic BDS, and use a grid of 256^2 cells and a fixed time step size $\Delta t = 2.5 \times 10^{-3}$, corresponding to an advective CFL number of ~ 0.6. In Figure 3, we show the concentration at several points in time, observing

		32–64	Rate	64–128	Rate	128–256
Test 1:	u	7.66×10^{-3}	1.75	2.27×10^{-3}	1.88	6.16×10^{-4}
	v	3.12×10^{-3}	1.96	8.02×10^{-4}	1.99	2.02×10^{-4}
	w	7.66×10^{-3}	1.75	2.27×10^{-3}	1.88	6.16×10^{-4}
	c	1.22×10^{-2}	2.00	3.06×10^{-3}	2.00	7.64×10^{-4}
Test 1:	u	2.30×10^{-3}	1.97	5.88×10^{-4}	2.02	1.45×10^{-4}
with higher-order	v	9.01×10^{-4}	2.23	1.92×10^{-4}	1.99	4.82×10^{-5}
boundary stencil	w	2.30×10^{-3}	1.97	5.88×10^{-4}	2.02	1.45×10^{-4}
	c	1.21×10^{-2}	1.99	3.05×10^{-3}	2.00	7.62×10^{-4}
Test 2:	u	7.67×10^{-3}	1.75	2.28×10^{-3}	1.89	6.16×10^{-4}
	v	3.11×10^{-3}	1.96	8.01×10^{-4}	1.99	2.02×10^{-4}
	w	7.67×10^{-3}	1.75	2.28×10^{-3}	1.89	6.16×10^{-4}
	c	9.79×10^{-3}	1.91	2.61×10^{-3}	1.97	6.68×10^{-4}
Test 2:	u	2.30×10^{-3}	1.96	5.89×10^{-4}	2.01	1.46×10^{-4}
with higher-order	v	8.90×10^{-4}	2.21	1.92×10^{-4}	1.99	4.82×10^{-5}
boundary stencil	w	2.30×10^{-3}	1.96	5.89×10^{-4}	2.01	1.46×10^{-4}
	c	9.70×10^{-3}	1.91	2.59×10^{-3}	1.96	6.67×10^{-4}

Table 4. Convergence of errors in the L_∞ norm for a *three-dimensional inertial* low Mach lid-driven cavity problem as the grid is refined in space and time, for the components of the velocity $\boldsymbol{v} = (u, v, w)$ and concentration c. The order of convergence is estimated from the error ratio between successive refinements.

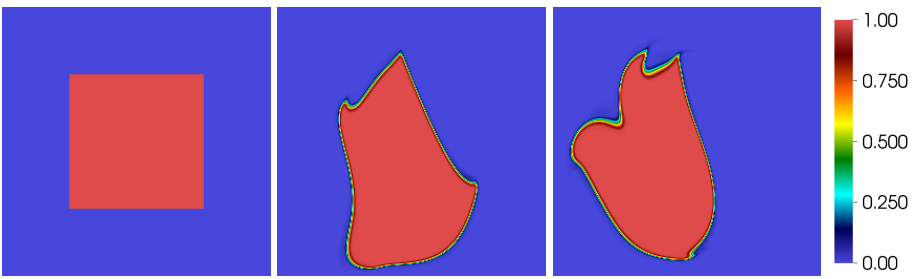


Figure 3. Advection of a square bubble in a lid-driven cavity flow, using the limited quadratic BDS scheme. Concentration is shown as a color plot at times $t = 0, 2, 4$.

very little smearing of the interface, even as the deformed bubble passes near the bottom boundary.

D. ***Deterministic Kelvin–Helmholtz instability.*** We simulate the development of a Kelvin–Helmholtz instability in three dimensions in order to demonstrate the robustness of our inertial time-advancement scheme in a deterministic setting. Our simulation uses $256 \times 128 \times 256$ computational cells with grid spacing $\Delta x = 1/256$. We use periodic boundary conditions in the x and z directions, a no-slip condition on

the y boundaries, with prescribed velocity $\boldsymbol{v}(x, y=0, z)=0$ on the bottom boundary and $\boldsymbol{v}(x, y=0.5, z)=(1, 0, 0)$ on the top boundary. We use an adaptive time step size Δt adjusted to maintain a maximum advective CFL number $v_{max}\Delta t/\Delta x \leq 0.9$. The binary fluid mixture has a 10:1 density contrast with $\bar{\rho}_1 = 10$ and $\bar{\rho}_2 = 1$. Viscosity varies linearly with concentration, such that $\eta = 10^{-4}$ for $c = 0$ and $\eta = 10^{-3}$ for $c = 1$. The mass diffusion coefficient is fixed at $\chi = 10^{-6}$, which makes the diffusive CFL number $\chi\Delta t/\Delta x^2 \sim 10^{-4}$, making it necessary to use BDS advection in order to avoid instabilities due to sharp gradients at the interface. We employ the bilinear BDS advection scheme [9] with limiting in order to preserve strict monotonicity and maintain concentration within the bounds $0 \leq c \leq 1$.

The initial condition is $c = 1$ in the lower-half of the domain, and $c = 0$ in the upper-half of the domain, so that light fluid sits on top of heavy fluid with a discontinuity in the concentration and velocity at the interface. The initial momentum is set to $\rho\boldsymbol{v} = (1, 0, 0)$ in the upper-half of the domain and $\rho\boldsymbol{v} = 0$ in the lower-half of the domain. Gravity has a magnitude of $g = 0.1$ acting in the downward y-direction. In order to set off the instability, in a row of cells at the centerline, c is initialized to a random value between 0 and 1. The subsequent temporal evolution of the density (which is related to concentration via the EOS) is displayed in Figure 4, showing the development of the instability with no visible numerical artifacts. We also observe uniformly robust convergence of the GMRES Stokes solver throughout the simulation.

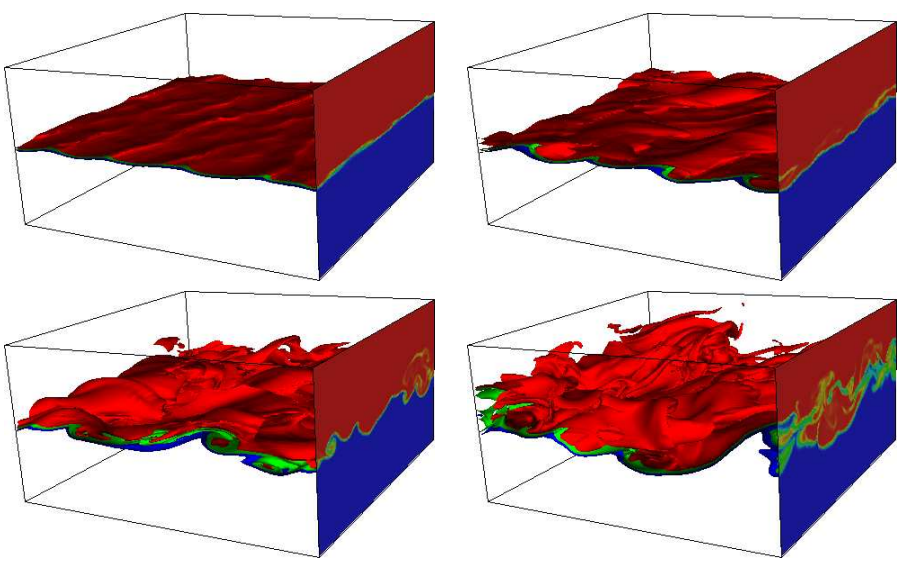


Figure 4. The development of a Kelvin–Helmholtz instability as a lighter less-viscous fluid streams over a ten times denser and more viscous fluid. Contour surfaces of the density, ranging from $\rho = 1$ (red) to $\rho = 10$ (blue), are shown at times $t = 1.72,\ 3.16,\ 4.53,$ and 5.85 s.

V. Giant concentration fluctuations

Advection of concentration by thermal velocity fluctuations in the presence of large concentration gradients leads to the appearance of *giant fluctuations* of concentration, as has been studied theoretically and experimentally for more than a decade [57; 58; 19; 56; 55]. In this section, we use our algorithms to simulate experiments measuring the temporal evolution of giant concentration fluctuations during free diffusive mixing in a binary liquid mixture. Croccolo et al. [19] report experimental measurements of the temporal evolution of the time-correlation functions of concentration fluctuations during the diffusive mixing of water and glycerol. In the experiments, a solution of glycerol in water with mass fraction of $c = 0.39$ is carefully injected in the bottom half of the experimental domain, under the $c = 0$ pure water in the top half. The two fluids slowly mix over the course of many hours while a series of measurements of the concentration fluctuations are performed.

In the experiments, quantitative shadowgraphy is used to observe and measure the strength of the fluctuations in the concentration via the change in the index of refraction. The observed light intensity, once corrected for the optical transfer function of the equipment, is proportional to the intensity of the fluctuations in the concentration averaged along the vertical (gradient) direction,

$$c_\perp(x, z;\ t) = H^{-1}\int_{y=0}^{H} c(x, y, z;\ t)\, dy,$$

where H is the thickness of the sample in the vertical direction. The quantity of interest is the correlation function of the Fourier coefficients $\widehat{\delta c}_\perp(k_x, k_z;\ t)$ of $c_\perp(x, z;\ t)$,

$$C(\tau;\ t, k) = \left\langle (\widehat{\delta c}_\perp(k_x, k_z;\ t+\tau))(\widehat{\delta c}_\perp(k_x, k_z;\ t))^\star \right\rangle,$$

where $k = \sqrt{k_x^2 + k_z^2}$ is the wavenumber (in our two dimensional simulations $k_z = 0$), τ is a delay time, and t is the elapsed time since the beginning of the experiment. In principle, the averaging above is an ensemble average but in the experimental analysis, and also in our processing of the simulation results, a time averaging over a specified time window T is performed in lieu of ensemble averaging. This approximation is justified because the system is ergodic and the evolution of the deterministic (background) state occurs via slow diffusive mixing of the water and glycerol solutions, and thus happens on a much longer time scale (hours) than the time delays of interest (a few seconds).

The Fourier transform (in time) of $C(\tau)$ is called the dynamic structure factor. The equal-time correlation function

$$S(k;\ t) = C(\tau = 0;\ t, k)$$

is the static structure factor, and is more difficult to measure in experiments [19]. For this reason, the experimental results are presented in the form of normalized time-correlation functions,

$$\widetilde{C}(\tau;\ t,k) = \frac{C(\tau;\ t,k)}{S(k;\ t)}.$$

The wavenumbers observed in the experiment and simulation are $k = \kappa \cdot 2\pi/L$, where κ is an integer and L is the horizontal extent of the observation window or the simulation box size. When evaluating the theory, we account for errors in the discrete approximation to the continuum Laplacian by using the effective wavenumber

$$k_\perp = k_x \frac{\sin(k_x \Delta x/2)}{(k_x \Delta x/2)} \tag{23}$$

instead of the actual discrete wavenumber k_x [6].

The confinement in the vertical direction is expected to play a small role because of the large thickness (2cm) of the sample, and a simple quasiperiodic (bulk) approximation can be used. Approximate theoretical analysis [22] suggests that at steady state the dominant nonequilibrium contribution to the static structure factor,

$$S(k;t) = \frac{k_B T}{(\eta\chi k^4 - \rho\beta g h)}\, h^2, \tag{24}$$

exhibits a k^{-4} power-law decay at large wavenumbers, and a plateau to $k_B T\, h/(\rho\beta g)$ for wavenumbers smaller than a rollover $k_c^4 = \rho\beta g h/(\eta\chi)$ due to the influence of gravity (buoyancy). Here $h(t) = d\bar{c}(y;t)/dy$ is the deterministic (background) concentration gradient, which decays slowly with time due to the continued mixing of the water and glycerol solutions.

An overdamped approximation suggests that the time correlations decay exponentially, $\widetilde{C}(\tau;\ t,k) = \exp(-\tau/\tau_k)$, with a relaxation time or decay time

$$\tau_k^{-1} = \chi k^2 \left[1 + \frac{\rho\beta g h}{\eta\chi k^4}\right], \tag{25}$$

that has a minimum at $k = k_c$ with value $\tau_{\min}^{-1} = 2\chi k_c^2 \sim \sqrt{hg}$. For wavenumbers $k < k_c$ the relaxation time becomes *smaller* and can in fact become very small at the smallest wavenumbers, requiring small time step sizes in the simulations to resolve the dynamics and ensure stability of the temporal integrators. In the presence of gravity, at small wavenumbers the separation of time scales used to justify the overdamped limit fails and the fluid inertia has to be taken into account [24]. This changes the prediction for the time correlation function to be a sum of two exponentials with relaxation times,

$$\tau_{1/2}^{-1} = \tfrac{1}{2}(\nu+\chi)k^2 \pm \tfrac{1}{2}\sqrt{k^4(\nu-\chi)^2 - 4\beta g h}, \tag{26}$$

where $\nu = \eta/\rho$. This expression becomes complex-valued for

$$k \lesssim k_p = \left(\frac{4\beta g h}{\nu^2}\right)^{1/4} = \left(\frac{4\chi}{\nu}\right)^{1/4} k_c,$$

indicating the appearance of *propagative* rather than diffusive modes for small wavenumbers, closely related to the more familiar gravity waves. While experimental measurements over wavenumbers $k \lesssim k_p$ are not reported by Croccolo et al. [19], their experimental data does contain several wavenumbers in that range. We report here simulation results for propagative concentration modes at small wavenumbers. To our knowledge, experimental observation of propagative modes has only been reported for temperature fluctuations [52].

Because it is essentially impossible to analytically solve the linearized fluctuating equations in the presence of spatially inhomogeneous density and transport coefficients and nontrivial boundary conditions, the existing theoretical analysis of the diffusive mixing process [58] makes a quasiperiodic constant-coefficient and constant-gradient incompressible approximation [22]. This approximation, while sufficient for qualitative studies, is not appropriate for quantitative studies because the viscosity η and mass diffusion coefficient χ vary by about a factor of three from the bottom to the top of the sample. In our simulations we account for the full dependence of density, viscosity and diffusion coefficient on concentration.

A. ***Simulation parameters.*** For LFHD there is no difference between the two and three dimensional problems due to the symmetries of the problem [22]. Because very long simulations with a small time step size are required for this study, we perform two-dimensional simulations. Furthermore, in these simulations we do not include a stochastic flux in the concentration equation, i.e., we set $\mathbf{\Psi} = 0$, so that all fluctuations in concentration arise from the coupling to the fluctuating velocity. With this approximation we do not need to model the chemical potential of the mixture and obtain μ_c. This approximation is justified by the fact that it is known experimentally that the nonequilibrium fluctuations are much larger than the equilibrium ones for the conditions we consider [19]; in fact, the fluctuations due to nonzero $\mathbf{\Psi}$ are smaller than solver or even roundoff tolerances in the simulations reported here.

We base our parameters on the experimental studies of diffusive mixing in a water-glycerol mixture, as reported by Croccolo et al. [19]. In the actual experiments the fluid sample is confined in a cylinder 2 cm in diameter and 2 cm thick in the vertical direction. In our simulations, the two-dimensional physical domain is 1.132 cm × 1.132 cm discretized on a uniform 256 × 256 two dimensional grid, with a thickness of 1 cm along the z direction. This large thickness makes the magnitude of the fluctuations very small since the cell volume ΔV contains a

very large number of molecules, and puts us in the linearized regime [24]. The width of the domain $L = 1.132\,\text{cm}$ is chosen to match the observation window in the experiments, and thus also match the discrete set of wavenumbers between the simulations and experiments. Earth gravity $g = -9.81\,\text{m}^2/\text{s}$ is applied in the negative y (vertical) direction; for comparison we also perform a set of simulations without gravity. Periodic boundary conditions are applied in the x-direction and impenetrable no-slip walls are placed at the y boundaries. The initial condition is $c = 0.39$ in the bottom half of the domain and $c = 0$ in the top half, with velocity initialized to zero. The temperature is kept constant at 300 K throughout the domain. Centered advection is used to ensure fluctuation-dissipation balance over the whole range of wavenumbers represented on the grid.

A very good fit to the experimental equation of state (dependence of density on concentration at standard temperature and pressure) over the whole range of concentrations of interest is provided by the EOS (3) with the density of water set to $\bar{\rho}_2 = 1\,\text{g/cm}^3$ and the density of glycerol set to $\bar{\rho}_1 = 1.29\,\text{g/cm}^3$. Experimentally, the dependence of viscosity on glycerol mass fraction has been fit to an exponential function [19], which we approximate with a rational function over the range of concentrations of interest [51],

$$\eta(c) \approx \frac{1.009 + 1.1262\,c}{1 - 1.5326\,c} \cdot 10^{-2}\,\frac{\text{g}}{\text{cm s}}. \tag{27}$$

The diffusion coefficient dependence on the concentration has been studied experimentally, and we employ the fit proposed in Ref. [25],

$$\chi(c) = \frac{1.024 - 1.002\,c}{1 + 0.663\,c} \cdot 10^{-5}\,\frac{\text{cm}^2}{\text{s}}, \tag{28}$$

which is in reasonable but not perfect agreement with a Stokes–Einstein relation $\eta(c)\chi(c) = \text{const}$. Note that the Schmidt number $S_c = \nu/\chi \sim 10^3$. In Ref. [19], based on the experimental measurements and the approximate theoretical model, it is suggested that $\chi \approx 10^{-5}\,\text{cm}^2/\text{s}$ is constant over the range of concentrations present. For comparison, we also perform simulations in which we keep the diffusion coefficient independent of concentration, while still taking into account the concentration dependence of viscosity. It is worth noting that there is a notable disagreement between experimental measurements of $\chi(c)$ using different experimental techniques [25] and the true dependence is not known with the same accuracy as that of $\eta(c)$.

When gravity is present, we use the inertial Algorithm 1, with a rather small time step size $\Delta t = 0.01375\,\text{s}$ due to the fact that the smallest relaxation time measured is on the order of 0.1 s. For this time step size, the viscous CFL number is $\nu\Delta t/\Delta x^2 \sim 10$–$30$, indicating that the viscous dynamics is resolved except at the wavenumbers comparable to the grid spacing. In the absence of gravity we use

Starting Time (s)	Total Time Steps	End Time (s)
600.6	3328	$600 + 3328\Delta t = 646.36$
3003	10784	3151.28
8108.1	4992	8176.74
15015	4768	15080.6

Table 5. Time intervals over which we average the dynamic correlation functions used to compute the relaxation times shown in Figure 5.

the overdamped Algorithm 2, which allows us to use a much larger time step size (on the diffusive time scale), $\Delta t = 0.22$ s, giving a diffusive CFL number on the order of $\chi \Delta t/\Delta x^2 \lesssim 0.1$. Using larger time step sizes than this would require an implicit treatment of mass diffusion.

B. ***Results.*** Our simulations closely mimic the experiments of Croccolo et al. [19]. We perform a long (stochastic) run of the diffusive mixing up to physical time $t =$ 21,021 s, saving a snapshot and statistics every 21,840 time steps, which corresponds to 300 seconds of physical time. We then perform 8 short runs with different random seeds starting from the saved snapshots, and compute time correlation functions averaged over a short time interval. Note that in the experiments a similar procedure is used in which data is collected over short time intervals during a single long mixing process. Croccolo et al. report measurements at $t = 600$ s, 3060 s, 8160 s, and 14,880 s. Table 5 lists the time intervals over which we collect statistics in the simulations, which match those in the experiments as well as possible. The time interval between successive snapshots used in the computation of the time correlation function is four time steps or 0.055 s, which is four times smaller than the interval used in the experimental analysis. In the experiments averaging is performed over a range of wavenumbers in the (k_x, k_z) plane with similar magnitude. Since we perform two dimensional simulations we average over the 8 independent simulations; in the end the statistical errors are lower in the simulation results since experiments are subject to large experimental noise not present in the simulations.

1. *Dynamic structure factors.* In order to extract a relaxation time, we fit the numerical results for the normalized time-correlation function to an analytical formula. For the first four wavenumbers $k = (1, 2, 3, 4) \cdot 2\pi/L$, clear oscillations (propagative modes) were observed, as illustrated in the top panel of Figure 5. For these wavenumbers we used the fit

$$\widetilde{C}(t) = \exp(-t/\tau)\big(A \sin(2\pi t/T) + \cos(2\pi t/T)\big), \tag{29}$$

where the relaxation time τ, the coefficient A and the period of oscillation T are the fitting parameters. For the remaining wavenumbers, we used a double-exponential decay for the fitting,

$$\widetilde{C}(t) = \alpha \exp(-t/\tau_1) + (1-\alpha)\exp(-t/\tau_2), \tag{30}$$

where α, τ_1 and τ_2 are the fitting parameters. This leads to good fits for $k > k_p \sim 32\,\mathrm{cm}^{-1}$; for a few transitional wavenumbers such as $k \sim 28\,\mathrm{cm}^{-1}$ the fit is not as good, as illustrated in the top panel of Figure 5. From the fit (30) we obtain the relaxation time τ as the point at which the amplitude decays by $\widetilde{C}(\tau) = 1/e$.

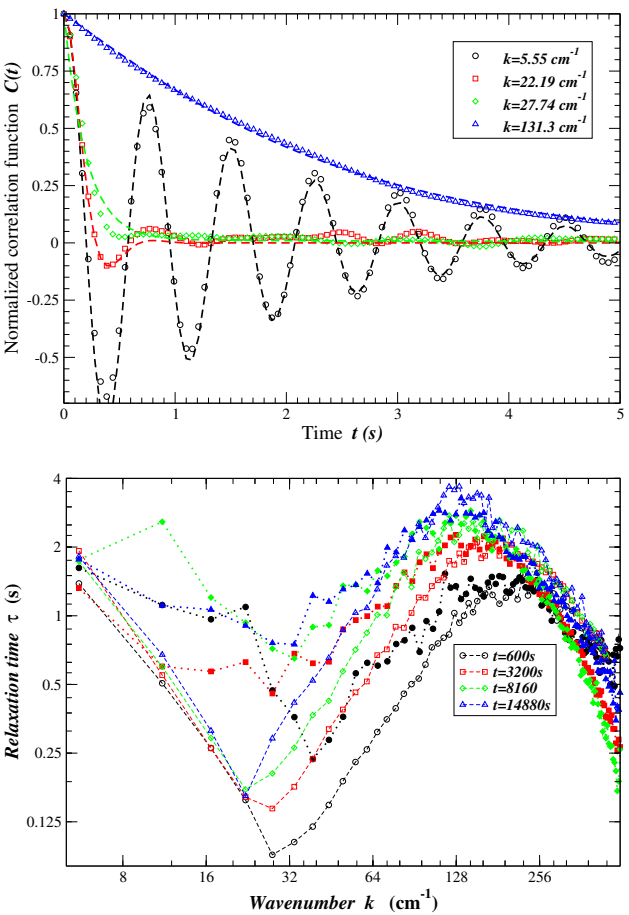


Figure 5. Dynamics of concentration fluctuations during free-diffusive mixing of water and glycerol. (Top) Numerical results for the time correlation functions for several selected wavenumbers about 8160 s from the beginning of the experiment. Symbols indicate results from the simulations and lines of the same color indicate the fit to (29) for the first ($k \approx 5.6\,\mathrm{cm}^{-1}$) and fourth ($k \approx 22.2\,\mathrm{cm}^{-1}$) wavenumbers, or to (30) for the remaining wavenumbers. Note that the statistical errors due to the finite averaging increase with time and the tails of the correlation functions are not reliably estimated. (Bottom) Relaxation or decay times as a function of wavenumber at several points in time. Empty symbols show results from computer simulations, and filled from experimental measurements [19].

A similar procedure was also used to obtain the relaxation time from the experimental data of Croccolo et al. [19] for all wavenumbers[2]. The experimental data shows monotonically decaying correlation functions $\widetilde{C}(\tau;\ t, k)$ for *all* measured wavenumbers, not consistent with the oscillatory correlation function observed for the four smallest wavenumbers in our simulations [24]; see the top panel of Figure 5. We believe that this mismatch is due to the way measurements for different wavenumbers of similar modulus are averaged in the experimental calculations. In our two-dimensional simulations, we do not perform any averaging over wavenumbers. We believe that the experimentally measured time correlation functions capture the *real* part of the decay times *only* and thus have the form of a sum of exponentials. Due to the lower time resolution and the fact that the static structure factor is not known, for the experimental data we used a single exponential fit and added an offset to account for the background noise,

$$C(t) = A\exp(-t/\tau) + B.$$

In the bottom panel of Figure 5 we compare simulation and experimental results for the real part of the decay or relaxation time τ_k, at several points in time measured from the beginning of the experiment. Good agreement is observed between the two with the same qualitative trends: a diffusive relaxation time $\tau_k^{-1} \approx \chi k^2$ for large wavenumbers, with a maximum at $k \approx k_c$, and then another minimum at $k \approx k_p$. Note that decay times are not reported by Croccolo et al. [19] for wavenumbers $k \lesssim k_p$ since that work focuses on the effect of gravity for $k \lesssim k_c$. In our analysis of the experimental data we included all measured wavenumbers, including those for which propagative modes are observed. Here, the diffusion coefficient varies with concentration according to (28); very similar results for the relaxation times were obtained by keeping $\chi \approx 10^{-5}\,\mathrm{cm}^2/\mathrm{s}$ constant, as suggested by Croccolo et al. [19]. This indicates that the dynamic correlations are not very sensitive to the concentration dependence $\chi(c)$. In future work we will perform a more careful comparison to experiments.

2. *Static structure factors.* Extracting the static structure factor from experimental measurements is complicated by several factors, including the presence of optical prefactors such as the transfer function of the instrument, and the appearance of additional contributions to the scattered light intensity such as shot noise, contributions due to giant temperature fluctuations [52], and capillary waves [17; 18]. We therefore study the evolution of the static structure factor using simulations only. In the top panel of Figure 6 we show numerical results for the static structure factor $S(k;\ t)$ of the discrete concentration field averaged along the y-axes, at a

[2]The experimental data for the time correlation functions were graciously given to us by Fabrizio Croccolo.

series of times t chosen to match those of the experimental measurements. Instead of ensemble averaging, here we performed a temporal average of the spectrum of the vertically averaged concentration over a period of 300 s, *ending* at the time indicated in the legend of the figure. The characteristic k^{-4} power law decay at large wavenumbers and the plateau at small wavenumbers predicted by (24) are clearly observed in Figure 6, consistent with a value of h decreasing with time. A

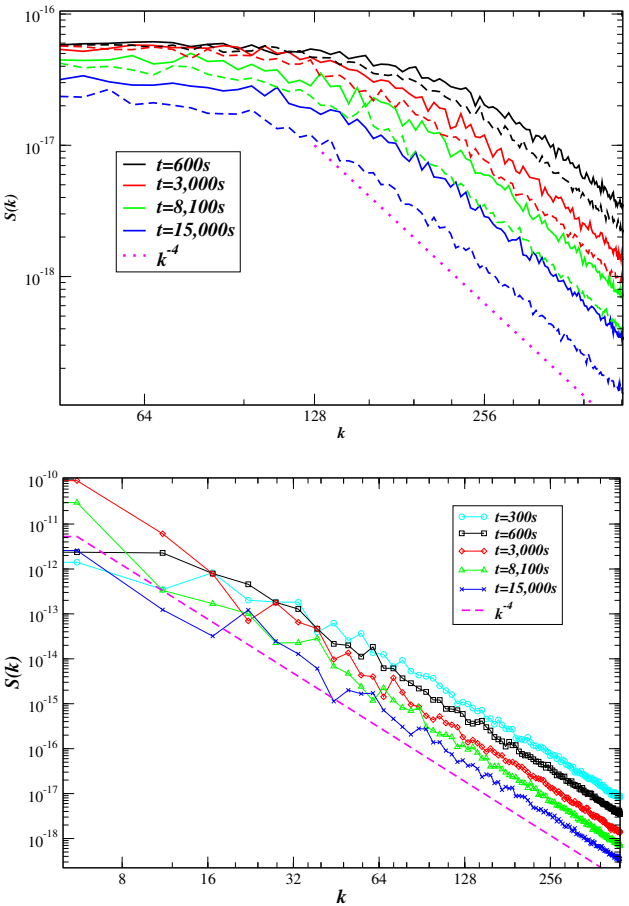


Figure 6. Evolution of the static structure factor during free-diffusive mixing of water and glycerol. (Top) With Earth gravity. Solid lines show results for simulations in which $\chi(c)$ depends on concentration according to (28), while dashed lines of the same color correspond to simulations in which $\chi \approx 10^{-5}\,\mathrm{cm}^2/\mathrm{s}$ is constant. Fluctuations at large wavenumbers follow a k^{-4} power law but are damped by gravity at small wavenumbers. (Bottom) Without gravity. Observe the large difference in the vertical axes showing "giant" fluctuations in the microgravity case. Note that these are results from a single simulation, mimicking a single experiment, and therefore there are large statistical uncertainties at small wavenumbers (large decorrelation times).

quantitative difference is seen between the results for variable and constant diffusion coefficients, consistent with a different value of the imposed concentration gradient h due to the somewhat different evolution of $\bar{c}(y,t)$.

In the bottom panel of Figure 6 we show numerical results for the static structure factor that would be obtained had the experiment been performed in microgravity ($g = 0$). In this case, we use the overdamped Algorithm 2 since there is a persistent large separation of time scales between the slow concentration and fast velocity. We see clear development of a k^{-4} power law as predicted by (24) for $g = 0$. Note that here the concentration gradient is established instantaneously, in fact, it is the largest in the initial configuration and then decays on the diffusive time scale; this is different from simulations of the development of giant fluctuations in microgravity during the GRADFLEX experiment reported in [24], in which the concentration gradient is slowly established on a diffusive scale. The results in Figure 6 show that it takes some time for the giant fluctuations at smallest wavenumbers to develop; the diffusive relaxation time corresponding to the smallest wavenumber studied, $k_{\min} \approx 5\,\mathrm{cm}^{-1}$, is $\tau_{\max} = (\chi k_{\min}^2)^{-1} \sim 4,000\,\mathrm{s}$. After a time $\sim (\chi k^2)^{-1}$, the amplitude of the fluctuations $S(k) \sim k^{-4}h^2(t)$ decays slowly due to the diffusive mixing, and eventually the system will fully mix and reach thermodynamic equilibrium.

VI. Conclusions

We have developed a low Mach number algorithm for diffusively mixing mixtures of two liquids with potentially different density and transport coefficients. In the low Mach number setting, the incompressible constraint is replaced by a quasicompressible constraint that dictates that stochastic and diffusive mass fluxes must create local expansion and contraction of the fluid to maintain a constant thermodynamic (base) pressure.

We employed a uniform-grid staggered-grid spatial discretization [6]. Following prior work in the incompressible simple-liquid case [35], we treated viscosity implicitly without splitting the pressure update, relying on a recently developed variable-coefficient Stokes solver [13] for efficiency. This approach works well for any Reynolds number, including the viscous-dominated overdamped (zero Reynolds number) limit, even in the presence of nontrivial boundary conditions. Furthermore, by using a high-resolution BDS scheme [9] to advect the concentration we robustly handled the case of no mass diffusion (no dissipation in the concentration equation). In our spatial discretization we strictly preserved mass and momentum conservation, as well as the equation of state (EOS) constraint, by using a finite-volume (flux-based) discretization of advective fluxes in which fluxes are computed using extrapolated values of concentration and density that obey the EOS. Our

temporal discretization used a predictor-corrector integrator that treats all terms except momentum diffusion (viscosity) explicitly [24].

We empirically verified second-order spatiotemporal accuracy in the deterministic method. In the stochastic context, establishing the weak order of accuracy is nontrivial in the general low Mach number setting. For centered advection our temporal integration schemes can be shown to be second-order accurate for the special case of a Boussinesq constant-density (incompressible) approximation, or in the overdamped (inertia-free) limit. Existing stochastic analysis does not apply to the case of BDS advection because Godunov schemes do not fit a method-of-lines approach, but rather, employ a space-time construction of the fluxes. The presence of nontrivial density differences between the pure fluid components and nonzero mass diffusion coefficient, complicates the analysis even for centered advection, due to the presence of a nontrivial EOS constraint on the fluid dynamics. It is a challenge for future work to develop improved numerical analysis of our schemes in both the deterministic and the stochastic setting.

In future work, we will demonstrate how to extend the algorithms proposed here to multispecies mixtures of liquids using a generalization of the low Mach number constraint. The nontrivial multispecies formulation of the diffusive and stochastic mass fluxes has already been developed by some of us in the compressible setting [5].

It is also possible to include thermal effects in our formulation, by treating the temperature in a manner similar to the way we treated concentration here. Two key difficulties are constructing a spatial discretization that ensures preservation of an appropriately generalized EOS, as well as developing temporal integrators that can handle the moderate separation of time scales between the (typically) slower heat diffusion and (typically) faster momentum diffusion. In particular, it seems desirable to also treat temperature implicitly. Such implicit treatment of mass or heat diffusion is nontrivial because it would require solving coupled (via the EOS constraint) velocity-temperature or velocity-concentration linear systems, and requires further investigation.

In the staggered-grid based discretization developed here, we can only employ existing higher-order Godunov advection schemes for the cell-centered scalar fields such as concentration and density. It is a challenge for future work to develop comparable methods to handle advection of the staggered momentum field. This would enable simulations of large Reynolds number flows. It should be noted, however, that our unsplit approach is most advantageous at small Reynolds numbers.

A challenge for future work on low Mach number fluctuating hydrodynamics is to account for the effects of surface tension in mixtures of immiscible or partially miscible liquids. This can be most straightforwardly accomplished by using a diffuse-interface model, as some of us recently did in the compressible setting for

a single-fluid multiphase system [15]. One of the key challenges is handling the fourth-order derivative term in the concentration equation in a way that ensures stability of the temporal integrator, as well as developing a consistent discretization of the Korteweg stresses on a staggered grid [50].

The semi-implicit temporal integrators we described here can deal well with a broad range of Reynolds or Schmidt numbers in the deterministic (smooth) setting. In the context of fluctuating hydrodynamics, however, all modes are thermally excited and treatment of viscosity based on a Crank–Nicolson method (implicit midpoint rule) are bound to fail for sufficiently large Schmidt numbers (or sufficiently low Reynolds numbers). In this work we solved this problem for the case of infinite Schmidt, zero Reynolds number flows by taking an overdamped limit of the original inertial equations before temporal discretization. It is a notable challenge for the future to develop uniformly accurate temporal integrators that work over a broad range of Reynolds or Schmidt numbers, including the asymptotic overdamped limit, in the presence of thermal fluctuations.

Acknowledgments

We would like to thank Fabrizio Croccolo and Alberto Vailati for sharing their experimental data on water-glycerol mixing, as well as numerous informative discussions. This material is based upon work supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-SC0008271 and under contract No. DE-AC02-05CH11231. Additional support for A. Donev was provided by the National Science Foundation under grant DMS-1115341.

References

[1] C. Almarcha, P. M. J. Trevelyan, P. Grosfils, and A. De Wit, *Chemically driven hydrodynamic instabilities*, Phys. Rev. Lett. **104** (2010), 044501.

[2] A. S. Almgren, J. B. Bell, C. A. Rendleman, and M. Zingale, *Low Mach number modeling of type Ia supernovae. I. Hydrodynamics*, The Astrophysical Journal **637** (2006), no. 2, 922–936.

[3] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations*, J. Comput. Phys. **142** (1998), no. 1, 1–46. MR 99k:76096 Zbl 0933.76055

[4] P. J. Atzberger, *Stochastic Eulerian Lagrangian methods for fluid-structure interactions with thermal fluctuations*, J. Comput. Phys. **230** (2011), no. 8, 2821–2837. MR 2012c:74040 Zbl 05909504

[5] K. Balakrishnan, A. L. Garcia, A. Donev, and J. B. Bell, *Fluctuating hydrodynamics of multi-species nonreactive mixtures*, Phys. Rev. E **89** (2014), 013017.

[6] F. Balboa Usabiaga, J. B. Bell, R. Delgado-Buscalioni, A. Donev, T. G. Fai, B. E. Griffith, and C. S. Peskin, *Staggered schemes for fluctuating hydrodynamics*, Multiscale Model. Simul. **10** (2012), no. 4, 1369–1408. MR 3022043 Zbl 1310.76108

[7] J. Bell, M. Day, C. Rendleman, S. Woosley, and M. Zingale, *Adaptive low mach number simulations of nuclear flame microphysics*, Journal of Computational Physics **195** (2004), no. 2, 677 – 694. Zbl 1115.85302

[8] J. B. Bell, P. Colella, and H. M. Glaz, *A second-order projection method for the incompressible Navier–Stokes equations*, J. Comput. Phys. **85** (1989), no. 2, 257–283. MR 90i:76002 Zbl 0681.76030

[9] J. B. Bell, C. N. Dawson, and G. R. Shubin, *An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions*, Journal of Computational Physics **74** (1988), no. 1, 1–24. Zbl 0684.65088

[10] J. B. Bell, A. L. Garcia, and S. A. Williams, *Computational fluctuating fluid dynamics*, M2AN Math. Model. Numer. Anal. **44** (2010), no. 5, 1085–1105. MR 2011h:76030

[11] A. K. Bhattacharjee, K. Balakrishnan, A. L. Garcia, J. B. Bell, and A. Donev, *Fluctuating hydrodynamics of multi-species reactive mixtures*, The Journal of Chemical Physics **142** (2015), no. 22, 224107.

[12] D. L. Brown, R. Cortez, and M. L. Minion, *Accurate projection methods for the incompressible Navier–Stokes equations*, J. Comput. Phys. **168** (2001), no. 2, 464–499. MR 2002a:76112 Zbl 1153.76339

[13] M. Cai, A. Nonaka, J. B. Bell, B. E. Griffith, and A. Donev, *Efficient variable-coefficient finite-volume Stokes solvers*, Commun. Comput. Phys. **16** (2014), no. 5, 1263–1297. MR 3256967

[14] R. Cerbino, Y. Sun, A. Donev, and A. Vailati, *Dynamic scaling for the growth of non-equilibrium fluctuations during thermophoretic diffusion in microgravity*, Tech. report, 2015, To appear in Sci. Reps. arXiv 1502.03693

[15] A. Chaudhri, J. B. Bell, A. L. Garcia, and A. Donev, *Modeling multiphase flow using fluctuating hydrodynamics*, Phys. Rev. E **90** (2014), 033014.

[16] A. J. Chorin, *Numerical solution of the Navier–Stokes equations*, Math. Comp. **22** (1968), 745–762. MR 39 #3723 Zbl 0198.50103

[17] P. Cicuta, A. Vailati, and M. Giglio, *Equilibrium and nonequilibrium fluctuations at the interface between two fluid phases*, Phys. Rev. E **62** (2000), 4920–4926.

[18] P. Cicuta, A. Vailati, and M. Giglio, *Capillary-to-bulk crossover of nonequilibrium fluctuations in the free diffusion of a near-critical binary liquid mixture*, Appl. Opt. **40** (2001), no. 24, 4140–4145.

[19] F. Croccolo, D. Brogioli, A. Vailati, M. Giglio, and D. S. Cannell, *Nondiffusive decay of gradient-driven fluctuations in a free-diffusion process*, Phys. Rev. E **76** (2007), 041112.

[20] G. Da Prato, *Kolmogorov equations for stochastic PDEs*, Birkhäuser, Basel, 2004. MR 2005m:60002 Zbl 1066.60061

[21] M. S. Day and J. B. Bell, *Numerical simulation of laminar reacting flows with complex chemistry*, Combustion Theory and Modelling **4** (2000), no. 4, 535–556. Zbl 0970.76065

[22] J. M. O. De Zarate and J. V. Sengers, *Hydrodynamic fluctuations in fluids and fluid mixtures*, Elsevier, Amsterdam, 2006.

[23] S. Delong, B. E. Griffith, E. Vanden-Eijnden, and A. Donev, *Temporal integrators for fluctuating hydrodynamics*, Phys. Rev. E **87** (2013), 033302.

[24] S. Delong, Y. Sun, B. E. Griffith, E. Vanden-Eijnden, and A. Donev, *Multiscale temporal integrators for fluctuating hydrodynamics*, Phys. Rev. E **90** (2014), 063312.

[25] G. D'Errico, O. Ortona, F. Capuano, and V. Vitagliano, *Diffusion coefficients for the binary system glycerol + water at 25° C. A velocity correlation study*, Journal of Chemical & Engineering Data **49** (2004), no. 6, 1665–1670.

[26] A. Donev, J. B. Bell, A. de la Fuente, and A. L. Garcia, *Diffusive transport by thermal velocity fluctuations*, Phys. Rev. Lett. **106** (2011), 204501.

[27] A. Donev, T. G. Fai, and E. Vanden-Eijnden, *A reversible mesoscopic model of diffusion in liquids: from giant fluctuations to Fick's law*, Journal of Statistical Mechanics: Theory and Experiment **2014** (2014), no. 4, P04004.

[28] A. Donev, A. Nonaka, Y. Sun, T. G. Fai, A. Garcia, and J. B. Bell, *Low Mach number fluctuating hydrodynamics of diffusively mixing fluids*, Commun. Appl. Math. Comput. Sci. **9** (2014), no. 1, 47–105. MR 3212867 Zbl 06443295

[29] A. Donev and E. Vanden-Eijnden, *Dynamic density functional theory with hydrodynamic interactions and fluctuations*, The Journal of Chemical Physics **140** (2014), no. 23, 234115.

[30] A. Donev, E. Vanden-Eijnden, A. L. Garcia, and J. B. Bell, *On the accuracy of finite-volume schemes for fluctuating hydrodynamics*, Commun. Appl. Math. Comput. Sci. **5** (2010), no. 2, 149–197. MR 2012d:65017 Zbl 1277.76089

[31] B. Dünweg and A. Ladd, *Lattice Boltzmann simulations of soft matter systems*, Advanced Computer Simulation Approaches for Soft Matter Sciences III (2009), 89–166.

[32] W. E and J.-G. Liu, *Gauge method for viscous incompressible flows*, Commun. Math. Sci. **1** (2003), no. 2, 317–332. MR 2004c:76039 Zbl 1160.76329

[33] P. Español, J. G. Anero, and I. Zúñiga, *Microscopic derivation of discrete hydrodynamics*, The Journal of Chemical Physics **131** (2009), no. 24, 244117.

[34] H. Grabert, *Projection operator techniques in nonequilibrium statistical mechanics*, Springer Tracts in Modern Physics, no. 95, Springer, Berlin, 1982. MR 84k:82001

[35] B. E. Griffith, *An accurate and efficient method for the incompressible Navier–Stokes equations using the projection method as a preconditioner*, J. Comput. Phys. **228** (2009), no. 20, 7565–7595. MR 2561832 Zbl 05615504

[36] M. Hütter and H. Christian Öttinger, *Fluctuation-dissipation theorem, kinetic stochastic integral and efficient simulations*, J. Chem. Soc., Faraday Trans. **94** (1998), 1403–1405.

[37] S. Y. Kadioglu, R. Klein, and M. L. Minion, *A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics*, J. Comput. Phys. **227** (2008), no. 3, 2012–2043. MR 2009g:76102 Zbl 1146.76035

[38] S. Klainerman and A. Majda, *Compressible and incompressible fluids*, Comm. Pure Appl. Math. **35** (1982), no. 5, 629–651. MR 84a:35264 Zbl 0478.76091

[39] L. D. Landau and E. M. Lifshitz, *Fluid mechanics*, Course of Theoretical Physics, no. 6, Pergamon Press, Oxford, 1959. MR 21 #6839

[40] J. Lowengrub and L. Truskinovsky, *Quasi-incompressible Cahn–Hilliard fluids and topological transitions*, R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci. **454** (1998), no. 1978, 2617–2654. MR 2000e:82022 Zbl 0927.76007

[41] S. May, A. Nonaka, A. Almgren, and J. Bell, *An unsplit, higher-order Godunov method using quadratic reconstruction for advection in two dimensions*, Commun. Appl. Math. Comput. Sci. **6** (2011), no. 1, 27–61. MR 2012f:65150 Zbl 1231.65142

[42] B. Müller, *Low-Mach-number asymptotics of the Navier–Stokes equations*, J. Engrg. Math. **34** (1998), no. 1-2, 97–109. MR 99f:76111 Zbl 0924.76095

[43] F. Nicoud, *Conservative high-order finite-difference schemes for low-Mach number flows*, J. Comput. Phys. **158** (2000), no. 1, 71–97. MR 2000j:76112 Zbl 0973.76068

[44] A. Nonaka, S. May, A. S. Almgren, and J. B. Bell, *A three-dimensional, unsplit Godunov method for scalar conservation laws*, SIAM J. Sci. Comput. **33** (2011), no. 4, 2039–2062. MR 2012j:65287 Zbl 05987096

[45] H. C. Öttinger, *Beyond equilibrium thermodynamics*, Wiley, New York, 2005.

[46] R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, W. A. Fiveland, and J. P. Jessee, *An adaptive projection method for unsteady, low-Mach number combustion*, Combustion Science and Technology **140** (1998), no. 1-6, 123–168.

[47] R. G. Rehm and H. R. Baumt, *The equation of motion for thermally driven, buoyant flows*, J. Research of the National Bureau of Standards **83** (1978), no. 3, 297–308.

[48] T. Schneider, N. Botta, K. J. Geratz, and R. Klein, *Extension of finite volume compressible flow solvers to multi-dimensional, variable density zero Mach number flows*, J. Comput. Phys. **155** (1999), no. 2, 248–286. MR 2000g:76081 Zbl 0968.76054

[49] J. Sethian and A. Majda, *The derivation and numerical solution of the equations for zero Mach number combustion*, Combustion science and technology **42** (1985), no. 3–4, 185–205.

[50] B. Z. Shang, N. K. Voulgarakis, and J.-W. Chu, *Fluctuating hydrodynamics for multiscale simulation of inhomogeneous fluids: Mapping all-atom molecular dynamics to capillary waves*, The Journal of Chemical Physics **135** (2011), no. 4, 044111.

[51] P. N. Shankar and M. Kumar, *Experimental determination of the kinematic viscosity of glycerol-water mixtures*, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences **444** (1994), no. 1922, 573–581.

[52] C. J. Takacs, G. Nikolaenko, and D. S. Cannell, *Dynamics of long-wavelength fluctuations in a fluid layer heated from above*, Phys. Rev. Lett. **100** (2008), 234502.

[53] E. Y. Tau, *A second-order projection method for the incompressible Navier–Stokes equations in arbitrary domains*, J. Comput. Phys. **115** (1994), no. 1, 147–152. MR 95g:76036 Zbl 0811.76064

[54] S. P. Thampi, I. Pagonabarraga, and R. Adhikari, *Lattice-Boltzmann–Langevin simulations of binary mixtures*, Phys. Rev. E **84** (2011), 046709.

[55] A. Vailati, R. Cerbino, S. Mazzoni, M. Giglio, C. J. Takacs, and D. S. Cannell, *Gradient-driven fluctuations in microgravity*, Journal of Physics: Condensed Matter **24** (2012), no. 28, 284134.

[56] A. Vailati, R. Cerbino, S. Mazzoni, C. J. Takacs, D. S. Cannell, and M. Giglio, *Fractal fronts of diffusion in microgravity*, Nature Comm. **2** (2011), 290.

[57] A. Vailati and M. Giglio, *Giant fluctuations in a free diffusion process*, Nature **390** (1997), no. 6657, 262–265.

[58] ———, *Nonequilibrium fluctuations in time-dependent diffusion processes*, Phys. Rev. E **58** (1998), 4361–4371.

Received October 8, 2014. Revised May 11, 2015.

ANDY NONAKA: AJNonaka@lbl.gov
Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A-1148, Berkeley, CA 94720, United States

YIFEI SUN: yifei@cims.nyu.edu
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, United States

JOHN B. BELL: jbbell@lbl.gov
Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, MS 50A-1148, 1 Cyclotron Road, Berkeley, CA 94720, United States

ALEKSANDAR DONEV: donev@courant.nyu.edu
Courant Institute of Mathematical Sciences, New York University, 1016 Warren Weaver Hall, 251 Mercer St., New York, NY 10012, United States

COMM. APP. MATH. AND COMP. SCI.
Vol. 10, No. 2, 2015
dx.doi.org/10.2140/camcos.2015.10.205

PARAMETER ESTIMATION BY IMPLICIT SAMPLING

MATTHIAS MORZFELD, XUEMIN TU,
JON WILKENING AND ALEXANDRE J. CHORIN

Implicit sampling is a weighted sampling method that is used in data assimilation to sequentially update state estimates of a stochastic model based on noisy and incomplete data. Here we apply implicit sampling to sample the posterior probability density of parameter estimation problems. The posterior probability combines prior information about the parameter with information from a numerical model, e.g., a partial differential equation (PDE), and noisy data. The result of our computations are parameters that lead to simulations that are compatible with the data. We demonstrate the usefulness of our implicit sampling algorithm with an example from subsurface flow. For an efficient implementation, we make use of multiple grids, BFGS optimization coupled to adjoint equations, and Karhunen–Loève expansions for dimensional reduction. Several difficulties of Markov chain Monte Carlo methods, e.g., estimation of burn-in times or correlations among the samples, are avoided because the implicit samples are independent.

1. Introduction

We wish to compute a set of parameters θ, an m-dimensional vector, so that simulations with a numerical model that require these parameters are compatible with data z (a k-dimensional vector) we have collected. We assume that some information about the parameter is available before we collect the data and this information is summarized in a prior probability density function (pdf) $p(\theta)$. For example, one may know a priori that some of the parameters are positive. The numerical model, e.g., a partial differential equation (PDE), defines the likelihood $p(z|\theta)$, which describes how the parameters are connected with the data. Bayes' rule combines the prior and likelihood to find the posterior density

$$p(\theta|z) \propto p(\theta)p(z|\theta);$$

see, e.g., [40]. This posterior pdf defines which parameters of the numerical model are compatible with the data z. The goal in parameter estimation is to compute the posterior pdf.

MSC2010: 86-08, 65C05.
Keywords: importance sampling, implicit sampling, Markov chain Monte Carlo.

If the prior and likelihood are Gaussian, then the posterior is also Gaussian, and it is sufficient to compute the mean and covariance of $\theta|z$ (because the mean and covariance define the Gaussian). The posterior mean and covariance are the minimizer and the inverse of the Hessian of the negative logarithm of a Gaussian posterior pdf. In nonlinear and non-Gaussian problems, one can compute the posterior mode, often called the maximum a posteriori (MAP) point, by minimizing the negative logarithm of the posterior, and use the MAP point (instead of the mean) as an approximation of the parameter θ. The inverse of the Hessian of the negative logarithm of the posterior can be used to measure the uncertainty of this approximation. This method is sometimes called linearization about the MAP point (LMAP) or the Laplace approximation [7; 24; 34; 35].

One can also use Markov chain Monte Carlo (MCMC) to solve a parameter estimation problem. In MCMC, one generates a collection of samples from the posterior pdf; see, e.g., [13; 16; 29; 36]. The samples form an empirical estimate of the posterior, and statistics, e.g., the mean or mode, can be computed from this empirical estimate by averaging over the samples. Under mild assumptions, the averages one computes from the samples converge to the expected values with respect to the posterior pdf as the number of samples goes to infinity. In practice, a finite number of samples is used and successful MCMC sampling requires that one can test if the chain has converged to the posterior pdf. The convergence can be slow due to correlations among the samples.

An alternative to MCMC is to use importance sampling. The idea is to draw samples from an importance function and to attach a weight to each sample such that the weighted samples form an empirical estimate of the posterior (see, e.g., [8]). The efficiency of importance sampling depends on the importance function which in turn defines the weights. Specifically, if the variance of the weights is large, then the weighted samples are a poor empirical estimate of the posterior and the number of samples required can increase quickly with the dimension of the problem [4; 5; 9; 39]. For this reason, importance sampling has not been used for parameter estimation problems in which the dimension is usually large. We investigate if implicit sampling which has been used before in online-filtering/data assimilation [2; 10; 11; 12; 30; 31; 42] can overcome this issue.

We will describe how to apply implicit sampling to parameter estimation problems, and it will become clear that an important step in implicit sampling is to minimize the negative logarithm of the posterior pdf, i.e., to find the MAP point. This optimization step identifies the region where the posterior probability is large, i.e., the region where the high-probability samples are. Starting from the MAP point, implicit sampling generates samples in its vicinity to explore the regions of high posterior probability. The optimization in implicit sampling represents the link between implicit sampling and LMAP. In fact, the optimization methods

used in LMAP codes can be used for implicit sampling; however, implicit sampling captures non-Gaussian characteristics of the posterior, which are usually missed by LMAP.

We illustrate the efficiency of our implicit sampling algorithm with numerical experiments using a problem from subsurface flow [3; 35]. This problem is a common test problem for MCMC algorithms, and the conditions for the existence of a posterior measure and its continuity are well understood [13]. Earlier work on this problem includes [16], where Metropolis–Hastings MC sampling is used, and [17], which uses optimal maps and is further discussed below.

The remainder of this paper is organized as follows. In Section 2, we explain how to use implicit sampling for parameter estimation and discuss an efficient implementation. Numerical examples are provided in Section 3. Conclusions are offered in Section 4.

2. Implicit sampling for parameter estimation

We wish to estimate an m-dimensional parameter vector θ from data which are obtained as follows. One measures a function of the parameters $h(\theta)$, where h is a given k-dimensional function; the measurements are noisy so that the data z satisfy the relation

$$z = h(\theta) + r, \tag{1}$$

where r is a random variable with a known distribution and the function h maps the parameters onto the data. Often, the function h involves solving a PDE. In a Bayesian approach, one obtains the pdf $p(\theta|z)$ of the conditional random variable $\theta|z$ by Bayes' rule:

$$p(\theta|z) \propto p(\theta)p(z|\theta), \tag{2}$$

where the likelihood $p(z|\theta)$ is given by (1) and the prior $p(\theta)$ is assumed to be known.

The goal is to sample the posterior and use the samples to calculate useful statistics. This can be done with importance sampling as follows [8; 25]. One can represent the posterior by M weighted samples. The samples θ_j, $j = 1, \ldots, M$, are obtained from an importance function $\pi(\theta)$ (which is chosen such that it is easy to sample from), and the j-th sample is assigned the weight

$$w_j \propto \frac{p(\theta_j)p(z|\theta_j)}{\pi(\theta_j)}.$$

A sample corresponds to a set of possible parameter values, and the weight describes how likely this set is in view of the posterior. The weighted samples $\{\theta_j, w_j\}$ form an empirical estimate of $p(\theta|z)$ so that, for a smooth function u, the sum

$$E_M(u) = \sum_{j=0}^{M} u(\theta_j)\hat{w}_j,$$

where $\hat{w}_j = w_j / \sum_{j=0}^{M} w_j$, converges almost surely to the expected value of u with respect to $p(\theta|z)$ as $M \to \infty$, provided that the support of π includes the support of $p(\theta|z)$ [8; 25].

The importance function must be chosen carefully or else sampling can become inefficient. For example, suppose you choose the prior as the importance function. In this case, the weights are proportional to the likelihood. Thus, one first draws samples from the prior and then determines their posterior probability by comparing them with the data. However, the samples one draws from the prior lie in the region where the prior probability is high and this region may not overlap with the region where the posterior probability is high. Two important scenarios in which this happens are:

(i) The prior and likelihood have (almost) disjoint support; i.e., the prior assigns probability mass in a small region of the (parameter) space in which the likelihood is small and vice versa. See Figure 1a.

(ii) The prior is broad; however, the likelihood is sharply peaked. See Figure 1b.

In either scenario, the samples we draw from the prior typically receive a low posterior probability so that the resulting empirical estimate of the posterior is inaccurate. An accurate empirical estimate requires samples with a high posterior probability, and a large number of prior samples may be required to obtain a few

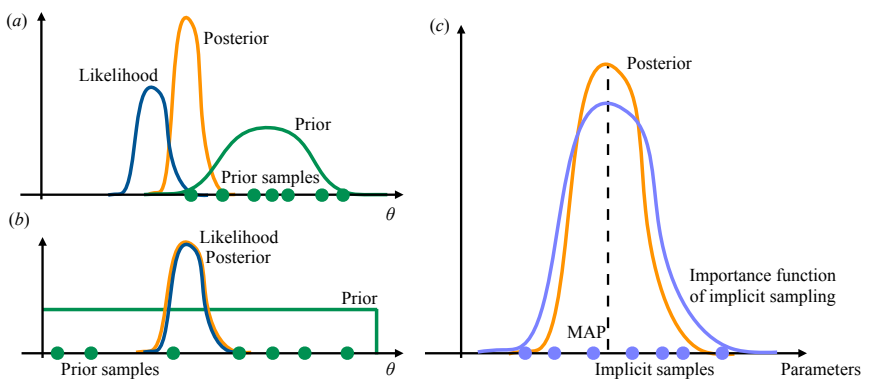


Figure 1. (*a*) The prior and likelihood are nearly mutually singular so that prior samples receive a small posterior probability. (*b*) The prior is broad and the likelihood is sharply peaked so that the majority of prior samples receives a small posterior probability. (*c*) The importance function of implicit sampling assigns probability mass to the neighborhood of the MAP point so that its overlap with the posterior pdf is significant, which leads to implicit samples that receive a large posterior probability.

samples with high posterior probability. In fact, the number of samples required can increase catastrophically with the dimension of the problem so that this importance sampling algorithm cannot be applied to high-dimensional problems [4; 5; 9; 39].

2.1. ***Basic ideas of implicit sampling.*** The idea in implicit sampling is to construct a data-informed importance function, which has a significant overlap with the posterior pdf (see Figure 1c). This requires in particular that the importance function be large where the posterior pdf is large. We can find one point where the posterior pdf is large by minimizing its negative logarithm; i.e., we find the MAP point as in LMAP methods. To set up the notation, let

$$F(\theta) = -\log\big(p(\theta)p(z|\theta)\big) \tag{3}$$

so that the MAP point is the minimizer of F,

$$\mu = \arg\min_{\theta} F(\theta).$$

Our goal is to construct an importance function that assigns high probability to the neighborhood of the MAP point. For the construction, we first use a random variable ξ with pdf $p(\xi) \propto \exp(G(\xi))$, which is easy to sample (e.g., a Gaussian). The variable ξ assigns high probability to the neighborhood of its mode, the minimizer of G. Next we define a new random variable, x, implicitly by the solutions of the algebraic equations

$$F(x) - \phi = G(\xi) - \gamma, \tag{4}$$

where $\phi = \min F$ and $\gamma = \min G$. The pdf of x can be calculated by a change of variables

$$\pi(x) = p(\xi(x)) \left|\det\left(\frac{\partial \xi}{\partial x}\right)\right|,$$

provided the map $\xi \to x$ is one-to-one and onto. Many one-to-one and onto mappings $\xi \to x$ exist because (4) is underdetermined: it is a scalar equation in m variables. The pdf $\pi(x)$ is the importance function of implicit sampling, and samples are drawn by solving (4). The weights of the samples are

$$w_j \propto \frac{p(\theta|z)}{\pi(x)} \propto \underbrace{\exp(G(\xi(x)) - F(\theta))}_{=\exp(\gamma-\phi)=\text{const.}} \left|\det\left(\frac{\partial \xi}{\partial x}\right)\right| \propto \left|\det\left(\frac{\partial \xi}{\partial x}\right)\right|, \tag{5}$$

proportional to the Jacobian of the map from x to ξ.

Note that a typical draw from the variable ξ is close to the mode of ξ so that G evaluated at a typical sample of ξ is close to its minimum γ. Thus, the left-hand side of (4) is likely to be small. A small left-hand side implies a small right-hand side so that the function F evaluated at the solution of (4) is close to its minimum ϕ. This forces the solutions of (4) to lie near the MAP point μ. Thus, by repeatedly

solving (4) for several draws of the variable ξ, we explore the neighborhood of the MAP point.

2.2. ***Solving the implicit equation.*** We describe and implement two strategies for solving (4) for a Gaussian ξ with mean 0 and covariance matrix H^{-1}, where H is the Hessian of the function F at the minimum. With this ξ, (4) becomes

$$F(\theta) - \phi = \tfrac{1}{2}\xi^T H \xi. \tag{6}$$

Both algorithms are affine invariant and, therefore, capable of sampling within flat and narrow valleys of F; see [20] for a discussion of the importance of affine invariance in Monte Carlo sampling.

2.2.1. *Random maps.* One can look for solutions of (6) in a random direction, ξ:

$$\theta = \mu + \lambda(\xi)\xi. \tag{7}$$

The stretch factor λ can be computed by substituting (7) into (6) and solving the resulting equation for the scalar $\lambda(\xi)$ with Newton's method. A formula for the Jacobian of the random map defined by (6) and (7) was derived in [22; 31]:

$$w \propto |J(\xi)| = \left| \lambda^{m-1} \frac{\xi^T H \xi}{\nabla_\theta F \cdot \xi} \right|, \tag{8}$$

where m is the number of nonzero eigenvalues of H. The Jacobian is easy to evaluate if the gradient of F is easy to compute, e.g., using the adjoint method (see below).

2.2.2. *Linear maps.* An alternative strategy is to approximate F by its Taylor expansion to second order:

$$F_0(\theta) = \phi + \tfrac{1}{2}(\theta - \mu)^T H (\theta - \mu),$$

where $\mu = \arg\min F$ is the minimizer of F (the MAP point) and H is the Hessian at the minimum. This strategy is called "implicit sampling with linear maps" and requires that one solves the quadratic equation

$$F_0(\theta) - \phi = \tfrac{1}{2}\xi^T H \xi \tag{9}$$

instead of (6). This can be done by simply shifting ξ by the mode: $\theta = \mu + \xi$. The bias created by solving the quadratic equation (9) instead of (6) can be removed by the weights [2; 10]

$$w \propto \exp(F_0(\theta) - F(\theta)). \tag{10}$$

A comparison of the linear and random map methods is given in [22], where it is found that the random map loses its advantages as the dimension of the problem increases if the posterior is a small perturbation of a Gaussian. We will confirm this theory with our numerical examples below.

2.2.3. *Connections with optimal maps.* An interesting construction, related to implicit sampling, has been proposed in [17; 38]. Suppose one wants to generate samples with the pdf $p(\theta|z)$ and have θ be a function of a variable ξ with pdf g, as above. If the samples are all to have equal weights, one must have, in the notation above,

$$p(\theta|z) = g(\xi)/J(\xi),$$

where, as above, J is the Jacobian of a map $\theta \to \xi$. Taking logs, one finds

$$F(\theta) + \log\beta = G(\xi) - \log(J(\xi)), \tag{11}$$

where $\beta = \int p(z|\theta)p(\theta)\,d\theta$ is the proportionality constant that has been elided in (2) and $G(\xi) = -\log\xi$. If one can find a one-to-one mapping from ξ to θ that satisfies this equation, one obtains an optimal sampling strategy, where the pdf of the samples matches exactly the posterior pdf. In [17], this map is found globally by choosing $g = p(\theta)$ (the prior), rather than sample-by-sample as in implicit sampling. The main differences between the implicit sampling equation (4) and (11) are the presence of the Jacobian J and of the normalizing constant β in the latter; J has shifted from being a weight to being a term in the equation that picks the samples, and the optimization that finds the probability mass has shifted to the computation of the map.

If ξ is Gaussian and the problem is linear, (11) can be solved by a linear map with a constant Jacobian and this map also solves (4) so that one recovers implicit sampling. In particular, in a linear Gaussian problem, the local (sample-by-sample) map (4) of implicit sampling also solves the global equation (11), which, for the linear problem, is a change of variables from one Gaussian to another. If the problem is not linear, the task of finding a global map that satisfies (11) is difficult (see also [15; 27; 38; 43]). The determination of optimal maps in [17], based on nonlinear transport theory, is elegant but can be computationally intensive and requires approximations that reintroduce nonuniform weights. Using (simplified) optimal maps and reweighting the samples from approximate maps is discussed in [38]. In [33], further optimal transport maps from prior to posterior are discussed. These maps are exact in linear Gaussian problems; however, in general, they are approximate, due to neglecting the Jacobian, when the problem is nonlinear.

2.3. ***Adjoint-based optimization with multiple grids.*** The first step in implicit sampling is to find the MAP point by minimizing F in (3). This can be done numerically by Newton, quasi-Newton, or Gauss–Newton methods (see, e.g., [32]). The minimization requires derivatives of the function F.

We consider parameter estimation problems in which the function h in (1) typically involves solving a PDE. In this case, adjoints are efficient for computing the gradient of F. The reason is that the complexity of solving the adjoint equation is similar to that of solving the original "forward" model. Thus, the gradient can

be computed at the cost of (roughly) two forward solutions. Adjoint methods are used widely in LMAP methods and can be used in connection with a quasi-Newton method, e.g., BFGS, or with Gauss–Newton methods. We illustrate how to use the adjoint method for BFGS optimization in the example below.

During the optimization, one can make use of multiple grids. This idea first appeared in the context of online state estimation in [2] and is similar to a multigrid finite difference method [18] and multigrid Monte Carlo [21]. However, the idea is different from the usual "multigrid" method (which is why we call it optimization with multiple grids). The idea is as follows. First, initialize the parameters and pick a coarse grid. Then perform the minimization on the coarse grid and use the minimizer to initialize a minimization on a finer grid. The minimization on the finer grid should require only a few steps, since the initial guess is informed by the computations on the coarser grid, so that the number of fine-grid forward and adjoint solves is small. This procedure can be generalized to use more than two grids (see the example below).

3. Application to subsurface flow

We illustrate the applicability of our implicit sampling method by a numerical example from subsurface flow, where we estimate subsurface structures from pressure measurements of flow through a porous medium. This is a common test problem for MCMC and has applications in reservoir simulation/management (see, e.g., [35]) and groundwater pollution modeling (see, e.g., [3]).

We consider Darcy's law

$$u = -\frac{\kappa}{\mu}\nabla p,$$

where ∇p is the pressure gradient across the porous medium, μ is the viscosity, and u is the average flow velocity; κ is the permeability and describes the subsurface structures we are interested in. Assuming, for simplicity, that the viscosity is constant, we obtain, from conservation of mass, the elliptic problem

$$-\nabla \cdot (\kappa \nabla p) = g, \tag{12}$$

on a domain Ω, with Dirichlet boundary conditions and where the source term g represents externally prescribed inward or outward flow rates. For example, if a well were drilled and a constant inflow were applied through this well, g would be a delta function with support at the well.

The uncertain quantity in this problem is the permeability; i.e., κ is a random variable, whose realizations we assume to be smooth enough so that, for each realization of κ, a unique solution of (12) exists. We would like to update our knowledge about κ on the basis of noisy measurements of the pressure at k locations

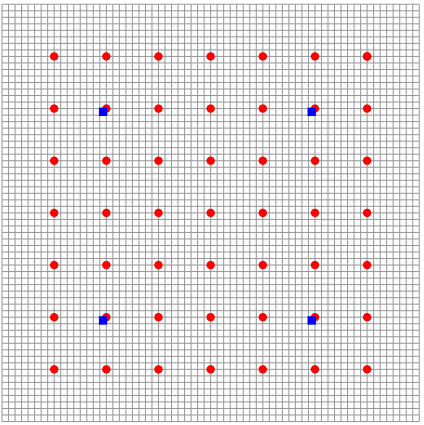


Figure 2. Mesh of the square domain (gray lines), pressure measurements (red dots), and forcing locations (delta distributions, blue squares)

within the domain Ω so that (1) becomes

$$z = h(p(\kappa), x, y) + r, \tag{13}$$

where r is a random variable.

In the numerical experiments below, we consider a 2D problem on a square domain $\Omega = [0, 1] \times [0, 1]$ and discretize (12) with a (standard) piecewise linear finite element method on a uniform $(N+1)\times(N+1)$ mesh of triangular elements [6]. We use the balancing domain decomposition by constraints method [14] to solve the resulting symmetric linear systems; i.e., we first decompose the computational domain into smaller subdomains and then solve a subdomain interface problem. The right-hand side g is a superposition of four delta distributions in the center of the domain (see Figure 2).

Our finest grid is 64×64, and the pressure measurements and forcing g are arranged such that they align with grid points of our fine and coarse grids (which we use in the multiple-grid approach). The 49 pressure measurements are collected in the center of the domain (see Figure 2).

The pressure measurements are perturbed with a Gaussian random variable $r \sim \mathcal{N}(0, R)$, with a diagonal covariance matrix R (i.e., we assume that measurement errors are uncorrelated). The variance at each measurement location is set to 30% of the reference solution. This relatively large variance brings about significant non-Gaussian features in the posterior pdf.

3.1. ***The log-normal prior, its discretization, and dimensional reduction.*** The prior for permeability fields is often assumed to be log-normal, and we follow suit. Specifically, the continuous permeability field is assumed log-normal with a

squared exponential covariance function [37],

$$K(x_1, x_2, y_1, y_2) = \exp\left(-\frac{(x_1 - x_2)^2}{l_x^2} - \frac{(y_1 - y_2)^2}{l_y^2}\right), \tag{14}$$

where (x_1, y_1) and (x_2, y_2) are two points in the domain Ω and where the correlation lengths are equal: $l_x = l_y = 0.5$. This prior models the (log-)permeability as a smooth function of x and y so that solutions of the PDE (12) uniquely exist. Moreover, the theory presented in [13; 40] applies and a well defined posterior also exists for the continuous problem.

The random permeability field is discretized on our uniform grid by a finite-dimensional random variable with a log-normal distribution. The elements of the covariance matrix Σ are obtained from the continuous correlation function (14)

$$\Sigma(i, j) = K(x_i, x_j, y_i, y_j), \quad i, j = 1, \ldots, N,$$

where N is the number of grid points in each direction. We perform a dimension reduction via Karhunen–Loève (KL) expansions [19; 26] and use the resulting low-rank approximation of the covariance matrix Σ for all subsequent computations. Specifically, the factorization of the covariance function $K(x_1, x_2, y_1, y_2)$ into the x and y directions allows us to compute the covariance matrices in each direction separately; i.e., we compute the matrices Σ_x and Σ_y with elements

$$\Sigma_x(i, j) = \sigma_x^2 \exp\left(-\frac{(x_i - x_j)^2}{l_x^2}\right), \qquad \Sigma_y(i, j) = \sigma_y^2 \exp\left(-\frac{(y_i - y_j)^2}{l_y^2}\right).$$

We then compute singular value decompositions (SVD) in each direction to form low-rank approximations $\hat{\Sigma}_x \approx \Sigma_x$ and $\hat{\Sigma}_y \approx \Sigma_y$ by neglecting small eigenvalues. These low-rank approximations define a low-rank approximation of the covariance matrix

$$\Sigma \approx \hat{\Sigma}_x \otimes \hat{\Sigma}_y,$$

where $\otimes$ is the Kronecker product. Thus, the eigenvalues and eigenvectors of $\hat{\Sigma}$ are the products of the eigenvalues and eigenvectors of $\hat{\Sigma}_x$ and $\hat{\Sigma}_y$. We obtain the low-rank approximation for the covariance matrix on the grid from the SVD of the covariance in each direction:

$$\hat{\Sigma} = V^T \Lambda V,$$

where Λ is a diagonal matrix whose diagonal elements are the m largest eigenvalues of Σ and V is an $m \times N$ matrix whose columns are the corresponding eigenvectors. Our approximate covariance $\hat{\Sigma}$ is optimal in the sense that the difference of the Frobenius norms of Σ and $\hat{\Sigma}$ is minimized. With $m = 30$ eigenvalues, we capture 99.9% of the variance (in the sense that the sum of the first 30 eigenvalues is 99% of the sum of all eigenvalues).

Thus, in reduced coordinates on the grid, the prior is

$$\hat{K} \sim \log \mathcal{N}(\hat{\mu}, \hat{\Sigma}).$$

Exponentiating followed by the linear change of variables

$$\theta = V^T \Lambda^{-0.5} \hat{K}$$

gives a prior for the "effective parameters" θ:

$$p(\theta) = \mathcal{N}(\mu, I_m), \tag{15}$$

where $\mu = V^T \Lambda^{-0.5} \hat{\mu}$. We will carry out the computations in the reduced coordinates θ. This reduces the effective dimension of the problem from N^2 (4096 for our finest grid) to $m = 30$. The model reduction follows naturally from assuming that the permeability is smooth, so that the prior is correlated, and the probability mass localizes in parameter space. A similar observation, in connection with data assimilation, was made in [9].

3.2. ***Multiple grids and adjoint-based BFGS optimization.*** Implicit sampling requires minimization of F in (3) which in reduced coordinates of this problem takes the form

$$F(\theta) = \tfrac{1}{2}\theta^T\theta + \tfrac{1}{2}(z - MP(\theta))^T R^{-1}(z - MP(\theta)),$$

where M is a $k \times N^2$ matrix that defines at which locations on the (fine) grid we collect the pressure. We solve the optimization problem using BFGS coupled to an adjoint code to compute the gradient of F with respect to θ (see also, e.g., [23; 34]).

The adjoint calculations are as follows. The gradient of F with respect to θ is

$$\nabla_\theta F(\theta) = \theta + (\nabla_\theta P(\theta))^T W,$$

where $W = -M^T R^{-1}(z - MP(\theta))$ and P is an N^2 vector that contains the pressure on the grid. We use the chain rule to derive $(\nabla_\theta P(\theta))^T W$ as follows:

$$\begin{aligned}(\nabla_\theta P(\theta))^T W &= \left(\nabla_K P(\theta) \frac{\partial K}{\partial \hat{K}} \frac{\partial \hat{K}}{\partial \theta}\right)^T W \\ &= (\nabla_K P(\theta) e^{\hat{K}} V \Lambda^{0.5})^T W = (V\Lambda^{0.5})^T (\nabla_K P(\theta) e^{\hat{K}})^T W,\end{aligned}$$

where $e^{\hat{K}}$ is an $N^2 \times N^2$ diagonal matrix whose elements are the exponentials of the components of $\hat{K}$. The gradient $\nabla_K P(\theta)$ can be obtained directly from our finite element discretization. Let $P = P(\theta)$, let K_l be the l-th component of K, and take the derivative with respect to K_l of our finite element discretization to obtain

$$\frac{\partial P}{\partial K_l} = -A^{-1} \frac{\partial A}{\partial K_l} P,$$

where A is the $N^2 \times N^2$ matrix that defines the linear system we solve and where $\partial A/\partial K_l$ are componentwise derivatives. We use this result to obtain

$$(\nabla_K P(\theta) e^{\hat{K}})^T W = -(e^{\hat{K}})^T \begin{bmatrix} P^T \dfrac{\partial A}{\partial K_1}(A^{-T}W) \\ \vdots \\ P^T \dfrac{\partial A}{\partial K_{N^2}}(A^{-T}W) \end{bmatrix}. \tag{16}$$

When P is available, the most expensive part in (16) is to evaluate $A^{-T}W$, which is equivalent to solving the adjoint problem (which is equal to itself for this self-adjoint problem). The rest can be computed elementwise by the definition of A. Note that there are only a fixed number of nonzeros in each $\partial A/\partial K_l$ so that the additional work for solving the adjoint problem in (16) is about $O(N^2)$, which is small compared to the work required for the adjoint solve.

Collecting terms we finally obtain the gradient

$$\begin{aligned} \nabla_\theta F(\theta) &= \theta + (V\Lambda^{0.5})^T (\nabla_K P(\theta) e^{\hat{K}})^T W \\ &= \theta - (V\Lambda^{0.5})^T (e^{\hat{K}})^T \begin{bmatrix} P^T \dfrac{\partial A}{\partial K_1}(A^{-T}W) \\ \vdots \\ P^T \dfrac{\partial A}{\partial K_{N^2}}(A^{-T}W) \end{bmatrix}. \end{aligned}$$

Multiplying by $(V\Lambda^{0.5})^T$ to go back to physical coordinates will require additional work of $O(mN^2)$. Note that the adjoint calculations for the gradient require only one adjoint solve because the forward solve (required for P) has already been done before the gradient calculation in the BFGS algorithm. In summary, our adjoint solves are only slightly more expensive than the forward solves. This concludes our derivation of an adjoint method for gradient computations.

We use this adjoint-based gradient computations in a BFGS method with a cubic interpolation line search [32, Chapter 3]. We use the multiple-grids approach to reduce the number of fine-grid solves. We use three grids, 16×16, 32×32, and 64×64. The required number of iterations on each grid and the number of forward solves are summarized in Table 1. After converting the cost of coarse/medium-grid solves to the cost of fine-grid solves, we estimate the cost of the multiple-grid

Grid	Iterations	Forward solves
16×16	9	32
32×32	6	14
64×64	5	12

Table 1. Required iterations and function evaluations for multiple-grid optimization.

optimization with 17 fine-grid solves. Without multiple grids, 36 fine-grid solves are needed to find the same minimum.

3.3. ***Implementation of the random and linear maps.*** We generate samples using the linear map and random map methods described above. Both require the Hessian of F at the minimum. A direct finite difference method for the Hessian would require $m(m+1) = 930$ forward solves, which is too expensive (infeasible if m becomes larger). For that reason, we approximate the Hessian by

$$H \approx \hat{H} = I - Q^T (QQ^T + R)^{-1} Q, \tag{17}$$

where $Q = M\nabla_\theta P$, as is standard in LMAP methods [24]. Here the gradient of the pressure (or the Jacobian) is computed with finite differences, which requires $m+1$ forward solves.

With this approximate Hessian, generating samples with the random map method requires solving (6) with the ansatz (7). We use a Newton method for solving these equations and observe that it usually converges quickly (within 1–4 iterations). Each iteration requires a derivative of F with respect to λ, which we implement using the adjoint method, so that each iteration requires two forward solutions. In summary, the random map method requires between 2–8 forward solutions per sample. The linear map method requires generating a Gaussian sample and weighting it by (10) so that only one forward solve is required per sample.

The quality of the weighted ensembles of the random and linear map methods can be assessed by the variance of the weights. A well distributed ensemble has a small variance of the weights. The variance of the weights is equal to $R - 1$, where

$$R = \frac{E(w^2)}{E(w)^2}.$$

In fact, R itself can be used to measure the quality of the samples [1; 41]. If the variance of the weights is small, then $R \approx 1$. Moreover, the effective sample size, i.e., the number of unweighted samples that would be equivalent in terms of statistical accuracy to the set of weighted samples, is about M/R [41], where M is the number of samples we draw. In summary, an R close to 1 indicates a well distributed ensemble.

We compute a value of R of about 1.6 for both methods. In fact, we generate 10 synthetic data sets, run implicit sampling with random and linear maps on each set, and estimate R based on 10^4 samples for each numerical experiment. We compute an $R = 1.68 \pm 0.10$ for the linear map method and $R = 1.63 \pm 0.066$ for the random map method. The random map method thus performs slightly better; however, the cost per sample is also slightly larger (because generating a sample requires solving (6), which in turn requires solving the forward problem). Since the

linear map method is less expensive and easier to program, it is a more appropriate technique for this problem.

We have also experimented with symmetrization of implicit sampling [22], which is similar in spirit to the classic trick of antithetic variates [25]. The symmetrization of the linear map method is as follows. Sample ξ, and compute a sample $x^+ = \mu + \xi$. Use the same ξ to compute $x^- = \mu - \xi$. Then pick x^+ with probability $p^+ = w(x^+)/(w(x^+) + w(x^-))$ and pick x^- with probability $p^- = w(x^-)/(w(x^+) + w(x^-))$, and assign the weight $w^s = (w(x^+) + w(x^-))/2$. This symmetrization can lead to a smaller R, i.e., a better distributed ensemble, in the small noise limit. In our example, we compute the quality measure $R = 1.4$. While this R is smaller than for the nonsymmetrized methods, the symmetrization does not pay off in this example since each sample of the symmetrized method requires two forward solves (to evaluate the weights).

3.4. ***Comparisons with other methods.*** The MAP and LMAP methods estimate parameters by computing the MAP point, i.e., the most likely parameters in view of the data, and estimate the uncertainty by a Gaussian whose covariance is the inverse of the Hessian of F at the minimum [7; 24; 34; 35]. In our example, LMAP overestimates the uncertainty since the Gaussian approximation has a standard deviation of 0.93 for the first parameter θ_1, whereas we compute 0.64 with the linear map and random map methods. The reason for the over-estimation of the uncertainty with LMAP is that the posterior is not Gaussian. This effect is illustrated in Figure 3, where we show histograms of the marginals of the posterior for the first four parameters θ_1, θ_2, θ_3, and θ_4 along with their Gaussian approximation as in LMAP. We also compute the skewness and excess kurtosis for these marginal densities. While the marginals for the parameters may become "more Gaussian" for the higher-order coefficients of the KL expansion, the joint posterior exhibits significant non-Gaussian behavior. Since implicit sampling (with random or linear maps) does not require linearizations or Gaussian assumptions, it can correctly capture these non-Gaussian features. In the present example, accounting for the non-Gaussian effects brings about a significant reduction of the uncertainty.

Note that code for LMAP can be converted into an implicit sampling code. In particular, implicit sampling with linear maps requires the MAP point and an approximation of the Hessian at the minimum. Both can be computed with LMAP codes. Non-Gaussian features of the posterior can then be captured by weighted sampling with linear maps, where each sample comes at a cost of a single forward simulation.

Another important class of methods for solving Bayesian parameter estimation problems is MCMC. We compare implicit sampling with Metropolis MCMC [28], where we use an isotropic Gaussian proposal density, for which we tuned the variance to achieve an acceptance rate of about 30%. This method requires one

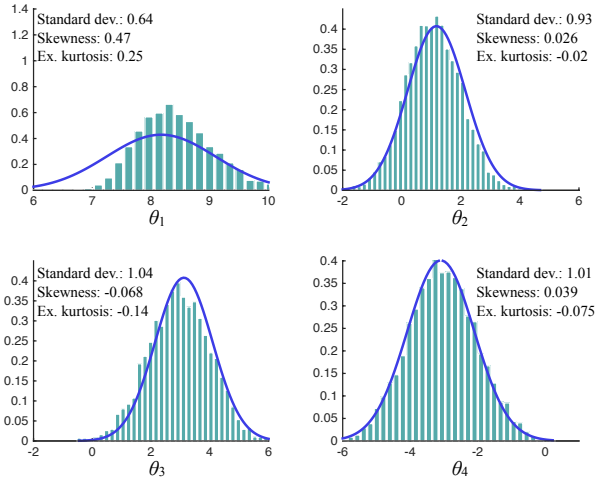


Figure 3. Marginals of the posterior computed with implicit sampling with random maps and their Gaussian approximation obtained via LMAP. Top left: $p(\theta_1|z)$. Top right: $p(\theta_2|z)$. Bottom left: $p(\theta_3|z)$. Bottom right: $p(\theta_4|z)$.

forward solution per step (to compute the acceptance probability). We start the chain at the MAP (to reduce burn-in time). In Figure 4, we show the approximation of the conditional mean of the variables θ_1, θ_2, and θ_3 as a function of the number

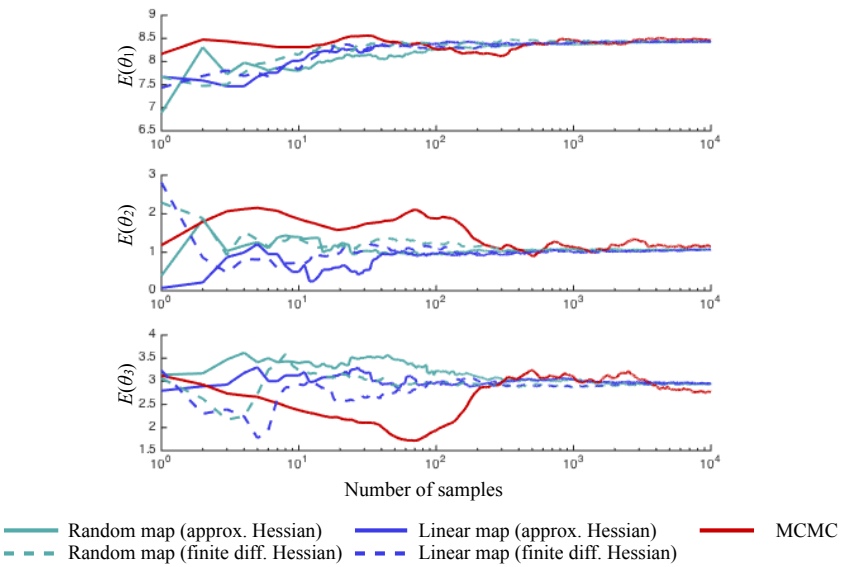


Figure 4. Expected value as a function of the number of samples. Red: MCMC. Turquoise: implicit sampling with random maps and approximate Hessian (dashed) and finite difference Hessian (solid). Blue: implicit sampling with linear maps and approximate Hessian (dashed) and finite difference Hessian (solid).

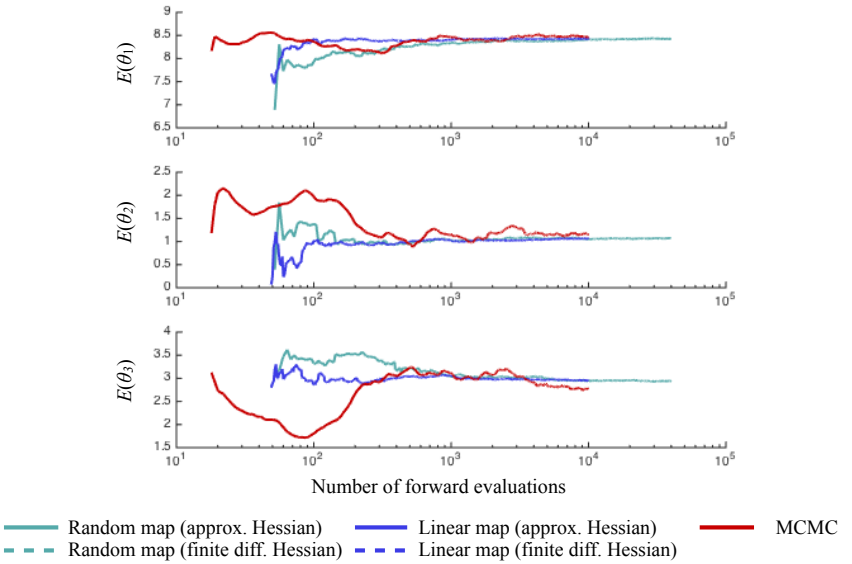


Figure 5. Expected value as a function of required forward solves. Red: MCMC. Turquoise: implicit sampling with random maps and approximate Hessian (dashed) and finite difference Hessian (solid). Blue: implicit sampling with linear maps and approximate Hessian (dashed) and finite difference Hessian (solid).

of steps in the chain. We observe that, even after 10^4 steps, the chain has not settled, in particular for the parameter θ_3 (see bottom pane).

With implicit sampling, we observe a faster convergence, in the sense that the approximated conditional mean does not change significantly with the number of samples. In fact, about 10^2 samples are sufficient for accurate estimates of the conditional mean. As a reference solution, we also show results we obtained with implicit sampling (with both random and linear maps) for which we used a Hessian computed with finite differences (rather than with the approximation in (17)).

The cost per sample of implicit sampling and the cost per step of Metropolis MCMC are different, and a fair comparison of these methods should take these costs into account. In particular, the offset cost of the minimization and computation of the Hessian, required for implicit sampling, must be accounted for. We measure the cost of the algorithms by the number of forward solves required. The results are shown for the parameters θ_1, θ_2, and θ_3 in Figure 5.

We find that the fast convergence of implicit sampling makes up for the relatively large a priori cost (for minimization and Hessian computations). In fact, the figure suggests that the random method requires only a few hundred samples, whereas Metropolis MCMC requires thousands of samples. The convergence of Metropolis MCMC can perhaps be increased by further tuning or by choosing a more advanced transition density. Implicit sampling on the other hand requires little tuning other

than deciding on standard tolerances for the optimization. Moreover, implicit sampling generates independent samples with a known distribution so that issues such as determining burn-in times, auto-correlation times, and acceptance ratios do not arise. It should also be mentioned that implicit sampling is easy to parallelize. Parallelizing Metropolis MCMC on the other hand is not trivial because it is a sequential technique.

Finally, we discuss connections of our proposed implicit sampling methods to stochastic Newton MCMC [29]. In stochastic Newton, one first finds the MAP point (as in implicit sampling or LMAP) and then starts a number of MCMC chains from the MAP point. The transition probabilities are based on local information about F and make use of the Hessian of F, evaluated at the location of the chain. Thus, at each step, a Hessian computation is required which, with our finite difference scheme, requires 31 forward solves (see above) and, therefore, is expensive (compared to generating samples with implicit sampling, which requires computing the Hessian only once). Second-order adjoints (if they were available) do not reduce that cost significantly. We have experimented with stochastic Newton in our example and have used 10–50 chains and taken about 200 steps per chain. Without significant tuning, we find acceptance rates of only a few percent, leading to a slow convergence of the method. We also observe that the Hessian may not be positive definite at all locations of the chain and, therefore, cannot be used for a local Gaussian transition probability. In summary, we find that stochastic Newton MCMC is impractical unless second-order adjoints are available to speed up the Hessian computations. Variations of stochastic Newton were explained and compared to each other in [36].

4. Conclusions

We explained how to use implicit sampling to estimate the parameters in PDE from sparse and noisy data. The idea in implicit sampling is to find the most likely state, often called the maximum a posteriori (MAP) point, and generate samples that explore the neighborhood of the MAP point. This strategy can work well if the posterior probability mass localizes around the MAP point, which is often the case when the data constrain the parameters. We discussed how to implement these ideas efficiently in the context of parameter estimation problems using multiple grids and adjoints to speed up the required optimization.

Our implicit sampling approach has the advantage that it generates independent samples so that issues connected with MCMC, e.g., estimation of burn-in times, auto-correlations of the samples, or tuning of acceptance ratios, are avoided. Our approach is also fully nonlinear and captures non-Gaussian features of the posterior

(unlike linear methods such as the linearization about the MAP point) and is easy to parallelize.

We illustrated the efficiency of our approach in numerical experiments with an elliptic inverse problem that is of importance in applications to reservoir simulation/management and pollution modeling. The elliptic forward model is discretized using finite elements, and the linear equations are solved by balancing domain decomposition by constraints. The optimization required by implicit sampling is done with a BFGS method coupled to an adjoint code. We use the fact that the solutions are expected to be smooth for model order reduction based on Karhunen–Loève expansions and found that our implicit sampling approach can exploit this low-dimensional structure. Moreover, implicit sampling is about an order of magnitude faster than Metropolis MCMC sampling (in the example we consider). We also discussed connections and differences of our approach with linear/Gaussian methods, such as linearization about the MAP, and with stochastic Newton MCMC methods. In particular, one can build an implicit sampling code starting from a MAP code by simply adding the Gaussian sampling and weighting step. At the cost of one additional forward solve per sample, the implicit sampling approach can reveal non-Gaussian features.

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract DE-AC02005CH11231 and by the National Science Foundation under grants DMS-0955078, DMS-1115759, DMS-1217065, and DMS-1419069.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*, IEEE T. Signal. Proces. **50** (2002), no. 2, 174–188.

[2] E. Atkins, M. Morzfeld, and A. J. Chorin, *Implicit particle methods and their connection with variational data assimilation*, Mon. Weather Rev. **141** (2013), no. 6, 1786–1803.

[3] J. Bear and A. Verruijt, *Modeling groundwater flow and pollution*, Theory and Applications of Transport in Porous Media, no. 2, Reidel, Dordrecht, Holland, 1987.

[4] T. Bengtsson, P. Bickel, and B. Li, *Curse-of-dimensionality revisited: collapse of the particle filter in very large scale systems*, Probability and statistics: essays in honor of David A. Freedman (D. Nolan and T. Speed, eds.), Inst. Math. Stat. Collect., no. 2, Inst. Math. Stat., Beachwood, OH, 2008, pp. 316–334. MR 2009k:93144 Zbl 1166.93376

[5] P. Bickel, B. Li, and T. Bengtsson, *Sharp failure rates for the bootstrap particle filter in high dimensions*, Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh (B. Clarke and S. Ghosal, eds.), Inst. Math. Stat. Collect., no. 3, Inst. Math. Stat., Beachwood, OH, 2008, pp. 318–329. MR 2010c:93107

[6] D. Braess, *Finite elements: theory, fast solvers, and applications in solid mechanics*, Cambridge University, 1997. MR 98f:65002 Zbl 0894.65054

[7] T. Bui-Thanh, O. Ghattas, J. Martin, and G. Stadler, *A computational framework for infinite-dimensional Bayesian inverse problems, I: The linearized case, with application to global seismic inversion*, SIAM J. Sci. Comput. **35** (2013), no. 6, A2494–A2523. MR 3126997 Zbl 1287.35087

[8] A. J. Chorin and O. H. Hald, *Stochastic tools in mathematics and science*, 3rd ed., Texts in Applied Mathematics, no. 58, Springer, New York, 2013. MR 3076304 Zbl 06150329

[9] A. J. Chorin and M. Morzfeld, *Conditions for successful data assimilation*, J. Geophys. Res. Atmos. **118** (2003), no. 20, 11522–11533.

[10] A. J. Chorin, M. Morzfeld, and X. Tu, *Implicit particle filters for data assimilation*, Commun. Appl. Math. Comput. Sci. **5** (2010), no. 2, 221–240. MR 2011m:60118 Zbl 1229.60047

[11] _____, *Implicit sampling, with application to data assimilation*, Chin. Ann. Math. Ser. B **34** (2013), no. 1, 89–98. MR 3011460 Zbl 1261.62084

[12] A. J. Chorin and X. Tu, *Implicit sampling for particle filters*, P. Natl. Acad. Sci. USA **106** (2009), no. 41, 17249–17254.

[13] M. Dashti and A. M. Stuart, *Uncertainty quantification and weak approximation of an elliptic inverse problem*, SIAM J. Numer. Anal. **49** (2011), no. 6, 2524–2542. MR 2873245 Zbl 1234.35309

[14] C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput. **25** (2003), no. 1, 246–258. MR 2004k:74099 Zbl 1038.65039

[15] A. Doucet, S. Godsill, and C. Andrieu, *On sequential Monte Carlo sampling methods for Bayesian filtering*, Stat. Comput. **10** (2000), no. 3, 197–208.

[16] Y. Efendiev, T. Hou, and W. Luo, *Preconditioning Markov chain Monte Carlo simulations using coarse-scale models*, SIAM J. Sci. Comput. **28** (2006), no. 2, 776–803. MR 2007b:65009 Zbl 1111.65003

[17] T. A. El Moselhy and Y. M. Marzouk, *Bayesian inference with optimal maps*, J. Comput. Phys. **231** (2012), no. 23, 7815–7850. MR 2972870 Zbl 06117578

[18] R. P. Fedorenko, *A relaxation method for solving elliptic difference equations*, USSR Comp. Math. Math. **1** (1962), no. 4, 1092–1096. Zbl 0163.39303

[19] R. G. Ghanem and P. D. Spanos, *Stochastic finite elements: a spectral approach*, Springer, New York, 1991. MR 91k:73102 Zbl 0722.73080

[20] J. Goodman, K. K. Lin, and M. Morzfeld, *Small-noise analysis and symmetrization of implicit Monte Carlo samplers*, Commun. Pur. Appl. Math. (2015), online publication July.

[21] J. Goodman and A. D. Sokal, *Multigrid Monte Carlo method: conceptual foundations*, Phys. Rev. D **40** (1989), no. 6, 2035–2071.

[22] J. Goodman and J. Weare, *Ensemble samplers with affine invariance*, Commun. Appl. Math. Comput. Sci. **5** (2010), no. 1, 65–80. MR 2011d:65007 Zbl 1189.65014

[23] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE constraints*, Mathematical Modelling: Theory and Applications, no. 23, Springer, New York, 2009. MR 2010h:49002 Zbl 1167.49001

[24] M. A. Iglesias, K. J. H. Law, and A. M. Stuart, *Evaluation of Gaussian approximations for data assimilation in reservoir models*, Comput. Geosci. **17** (2013), no. 5, 851–885. MR 3104638

[25] M. H. Kalos and P. A. Whitlock, *Monte Carlo methods*, vol. 1, Wiley, New York, 1986. MR 88e:65009 Zbl 0655.65004

[26] O. P. Le Maître and O. M. Knio, *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*, Springer, New York, 2010. MR 2011b:65002 Zbl 1193.76003

[27] J. S. Liu and R. Chen, *Blind deconvolution via sequential imputations*, J. Am. Stat. Assoc. **90** (1995), no. 430, 567–576. Zbl 0826.62062

[28] J. S. Liu, *Monte Carlo strategies in scientific computing*, Springer, New York, 2008. MR 2010b:65013 Zbl 1132.65003

[29] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas, *A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion*, SIAM J. Sci. Comput. **34** (2012), no. 3, A1460–A1487. MR 2970260 Zbl 1250.65011

[30] M. Morzfeld and A. J. Chorin, *Implicit particle filtering for models with partial noise, and an application to geomagnetic data assimilation*, Nonlinear Proc. Geophys. **19** (2012), 365–382.

[31] M. Morzfeld, X. Tu, E. Atkins, and A. J. Chorin, *A random map implementation of implicit filters*, J. Comput. Phys. **231** (2012), no. 4, 2049–2066. MR 2012m:62287 Zbl 1242.65012

[32] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed., Springer, New York, 2006. MR 2007a:90001 Zbl 1104.65059

[33] D. S. Oliver, *Minimization for conditional simulation: relationship to optimal transport*, J. Comput. Phys. **265** (2014), 1–15. MR 3173132

[34] D. S. Oliver and Y. Chen, *Recent progress on reservoir history matching: a review*, Computat. Geosci. **15** (2011), no. 1, 185–221. Zbl 1209.86001

[35] D. S. Oliver, A. C. Reynolds, and N. Liu, *Inverse theory for petroleum reservoir characterization and history matching*, Cambridge University, 2008.

[36] N. Petra, J. Martin, G. Stadler, and O. Ghattas, *A computational framework for infinite-dimensional Bayesian inverse problems, II: Stochastic Newton MCMC with application to ice sheet flow inverse problems*, SIAM J. Sci. Comput. **36** (2014), no. 4, A1525–A1555. MR 3233941 Zbl 1303.35110

[37] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, MIT, Cambridge, MA, 2006. MR 2010i:68131 Zbl 1177.68165

[38] N. Recca, *A new methodology for importance sampling*, master's thesis, New York University, 2011.

[39] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, *Obstacles to high-dimensional particle filtering*, Mon. Weather Rev. **136** (2008), no. 12, 4629–4640.

[40] A. M. Stuart, *Inverse problems: a Bayesian perspective*, Acta Numer. **19** (2010), 451–559. MR 2011i:65093 Zbl 1242.65142

[41] E. Vanden-Eijnden and J. Weare, *Data assimilation in the low noise regime with application to the Kuroshio*, Mon. Weather Rev. **141** (2013), no. 6, 1822–1841.

[42] B. Weir, R. N. Miller, and Y. H. Spitz, *Implicit estimation of ecological model parameters*, Bull. Math. Biol. **75** (2013), no. 2, 223–257. MR 3022352 Zbl 1310.92005

[43] V. S. Zaritskii, V. B. Svetnik, and L. I. Shimelevich, *Monte-Carlo technique in problems of optimal information processing*, Automat. Remote Control **36** (1975), no. 12, 2015–2022. MR 55 #1701 Zbl 0344.60041

Received June 23, 2015.

MATTHIAS MORZFELD: mmo@math.lbl.gov
Department of Mathematics, University of California, Berkeley, Evans Hall, Berkeley, CA 94720, United States

and

Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

XUEMIN TU: xtu@math.ku.edu
Department of Mathematics, University of Kansas, 1460 Jayhawk Boulevard, Lawrence, KS 66045, United States

JON WILKENING: wilken@math.berkeley.edu
Department of Mathematics, University of California, Berkeley, Evans Hall, Berkeley, CA 94720, United States

and

Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

ALEXANDRE J. CHORIN: chorin@math.lbl.gov
Department of Mathematics, University of California, Berkeley, Evans Hall, Berkeley, CA 94720, United States

and

Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

Guidelines for Authors

Authors may submit manuscripts in PDF format on-line at the Submission page at msp.org/camcos.

Originality. Submission of a manuscript acknowledges that the manuscript is original and and is not, in whole or in part, published or under consideration for publication elsewhere. It is understood also that the manuscript will not be submitted elsewhere while under consideration for publication in this journal.

Language. Articles in CAMCoS are usually in English, but articles written in other languages are welcome.

Required items. A brief abstract of about 150 words or less must be included. It should be self-contained and not make any reference to the bibliography. If the article is not in English, two versions of the abstract must be included, one in the language of the article and one in English. Also required are keywords and subject classifications for the article, and, for each author, postal address, affiliation (if appropriate), and email address.

Format. Authors are encouraged to use LaTeX but submissions in other varieties of TeX, and exceptionally in other formats, are acceptable. Initial uploads should be in PDF format; after the refereeing process we will ask you to submit all source material.

References. Bibliographical references should be complete, including article titles and page ranges. All references in the bibliography should be cited in the text. The use of BibTeX is preferred but not required. Tags will be converted to the house format, however, for submission you may use the format of your choice. Links will be provided to all literature with known web locations and authors are encouraged to provide their own links in addition to those supplied in the editorial process.

Figures. Figures must be of publication quality. After acceptance, you will need to submit the original source files in vector graphics format for all diagrams in your manuscript: vector EPS or vector PDF files are the most useful.

Most drawing and graphing packages (Mathematica, Adobe Illustrator, Corel Draw, MATLAB, etc.) allow the user to save files in one of these formats. Make sure that what you are saving is vector graphics and not a bitmap. If you need help, please write to graphics@msp.org with details about how your graphics were generated.

White space. Forced line breaks or page breaks should not be inserted in the document. There is no point in your trying to optimize line and page breaks in the original manuscript. The manuscript will be reformatted to use the journal's preferred fonts and layout.

Proofs. Page proofs will be made available to authors (or to the designated corresponding author) at a Web site in PDF format. Failure to acknowledge the receipt of proofs or to return corrections within the requested deadline may cause publication to be postponed.

Communications in Applied Mathematics and Computational Science

vol. 10 no. 2 2015