AN EMBEDDED BOUNDARY METHOD FOR THE NAVIER-STOKES EQUATIONS ON A TIME-DEPENDENT DOMAIN*

GREGORY H. MILLER† AND DAVID TREBOTICH‡

Abstract. We present a new conservative Cartesian grid embedded boundary method for the solution of the incompressible Navier-Stokes equations in a time-dependent domain. It is a Godunov-projection fractional step scheme in which hyperbolic advection and a variety of implicit and explicit Helmholtz operations are performed on time-stationary domains. The transfer of data from one fixed domain to another uses third-order interpolation. The method is second order accurate in L_1 and first order in L_∞. The algorithm is verified on flow geometries with prescribed boundary motion.

Key words. Navier-Stokes, embedded boundary, finite volume, moving domain.

AMS subject classifications. 35Q30, 35R37, 65M08

1. Introduction. The incompressible Navier-Stokes equations on a time-dependent domain

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \nu \Delta \mathbf{u} \tag{1.1a}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{1.1b}$$

approximate fluid behavior in a range of important applications. Here $\mathbf{u}(\mathbf{x}, t)$ is the velocity of the fluid whose density is assumed to be unity. $\mathbf{x}$ is the spatial coordinate, t is time, P is pressure, and ν is the kinematic viscosity. We are particularly concerned with reaction-diffusion equations in porous media where reactive transport can alter the subsurface pore structure due to precipitation or dissolution. Other motivating applications include the dynamics of biological membranes and lipid bilayer analogs, and modeling rod-climbing and die-swell behavior of certain viscoelastic fluids. In these examples, the evolution of the fluid domain is coupled to the motion of the fluid. Prescribed domain motion occurs in pumps, stirred vessels, and other mechanical systems.

There are two categories of approaches to discretizing moving domains: (i) gridding schemes that conform to the domain boundary, e.g., unstructured grids obeying Lagrangian dynamics; and (ii) structured, Cartesian grids where the domain influences the solution through a forcing as in the immersed boundary method [30] or the immersed interface method [18], or through cut cell methods where the finite volume quadrature is modified on those Cartesian cells overlain by the domain boundary, otherwise known as embedded boundary methods. Cut cell methods are confronted by a small-cell stability problem: finite volume discretizations are unstable on cells whose volume fraction vanishes. Approaches to this problem include cell merging techniques (Noh's "blending" [27]), the h-box technique that references a cell of nonvanishing

*This material is based upon work supported by the National Science Foundation under grant number DMS-0810939, and is supported as part of the Center for Nanoscale Control of Geologic CO_2, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, and the Office of Advanced Scientific Computing, under Award Number DE-AC02-05CH11231, and by DOE grant number DE-SC0001981.

†Department of Chemical Engineering and Materials Science, University of California, 1 Shields Avenue, Davis CA 95616.

‡Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, MS 50A-1148, 1 Cyclotron Road, Berkeley, CA 94720.

size [2], and hybridization – use of a stable but non-conservative quadrature with subsequent reestablishment of conservation in a neighborhood [3]. Our approach is an embedded boundary method, with cut-cell stability through hybridization. This strategy has proven accurate, robust, and scalable in large scale simulations of reactive transport in fixed irregular domains [33].

Projection methods [7, 6, 8] use the unique Hodge decomposition of a vector field to determine the divergence-free component, and the gradient of a potential that can be associated with the pressure gradient. Godunov-projection methods are fractional step methods that first compute an intermediate velocity with a high-order Godunov approach, which is made discrete divergence-free by a Hodge projection. Other approaches can achieve high order without reference to the intermediate state, for example computation of $\mathbf{u}\cdot\nabla\mathbf{u}$ via an Adams-Bashforth approach. Our approach is based on the second-order projection method of Bell et al. [1], with a second-order unsplit Godunov method for the intermediate velocity [10], and using approximate projections after Lai [16]. For hyperbolic flow problems, high-order Godunov methods do a superior job of resolving steep gradients. Minion and Brown [26] compare a number of approaches to solving incompressible Navier-Stokes. Their examples show that the Godunov-projection approach does a good job of resolving incompressible Navier-Stokes flows with steep gradients without introducing spurious high-frequency oscillations created by some other approaches. This is a significant benefit for reacting flows where steep gradients exist and reaction rates can be sensitive to high-frequency oscillation.

There have been many recent developments in projection methods for the moving domain Navier-Stokes problem. Pan et al. [28] use a Godunov-projection method with multiblock structured ALE (arbitrary Lagrangian-Eulerian) grids. Udaykumar et al. [39] use an Adams-Bashforth projection approach with finite volume discretization. They locate the interface with Lagrangian marker particles, and address the small cell problem with cell merging. Marella et al. [21] employ a similar method, with interface information derived from a discrete level set. Tan et al. [36] also use level sets to represent the interface, and combine the immersed interface method with an Adams-Bashforth projection method. Liau et al. [19] combine an Adams-Bashforth projection method with the immersed boundary method. Chiu et al. [4] use the immersed boundary method with a different second-order projection discretization. All of these methods claim or demonstrate second-order accuracy. Strict conservation is necessary to accurately capture wave behavior [17], a property essential for combustion and reactive flows. Such conservation is readily obtained with ALE and finite volume methods but is a very delicate issue for immersed boundary methods [15].

In this work we present a new conservative Godunov-projection method on Cartesian grids for the solution of the incompressible Navier-Stokes equations (1.1a) on a time-dependent domain $\Omega(t)$, with boundary conditions

$$\mathbf{u} = \mathbf{s}(\mathbf{x}, t) \tag{1.2a}$$

on moving walls, where $\mathbf{s}$ is the velocity of the boundary;

$$\mathbf{u} = \mathbf{u}_{\text{in}}(\mathbf{x}, t) \tag{1.2b}$$

on inflow boundaries; and

$$\mathbf{n}\cdot\nabla\mathbf{u} = 0 \tag{1.2c}$$

on outflow boundaries where $\mathbf{n}$ is normal to the domain boundary. We represent the domain boundary as the zero of a distance function level set, and derive all geometric descriptions at the moving front from the discrete level set. In this work, the boundary motion is prescribed.

We discretize space in uniform Cartesian cells which we label with index $\mathbf{i}$, an integer vector in D space dimensions. The center of cell $\mathbf{i}$ has spatial coordinate $\mathbf{x} = h(\mathbf{i} + \frac{1}{2}\mathbf{1})$ where h is the length of the cell, and $\mathbf{1}$ is the vector of ones in $\mathbb{Z}^D$. Time is discretized in uniform increments Δt, and $t^n = n\Delta t$ is the time at step n. $\mathbf{u}_\mathbf{i}^n$ denotes the value of fluid velocity $\mathbf{u}$ at the center of cell $\mathbf{i}$ at time t^n, and with $\mathbf{e}_d$ the d^{th} unit basis vector, $\mathbf{u}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}^{n+\frac{1}{2}}$ denotes the fluid velocity at the half time step $t^{n+\frac{1}{2}}$ and the center of $\mathbf{i}$'s cell face in direction $+d$. With this discretization, an outline of the approach is:

1. Extrapolate $\mathbf{u}_\mathbf{i}^n$ to $\Omega^{n+\frac{1}{2}}$, the fluid domain at time $t^{n+\frac{1}{2}}$. For those cells $\mathbf{i}$ in $\Omega^{n+\frac{1}{2}} \setminus \Omega^n$ (Figures 1.1,1.2), this extrapolation is based on the algorithm proposed by McCorquodale et al. [24]: three cells, whose centers together with $\mathbf{i}$ are collinear and approximately aligned with the interface normal, define a quadratic interpolation function determining $\mathbf{u}_\mathbf{i}$.

2. On $\Omega^{n+\frac{1}{2}}$, use high-order Godunov methods to compute time– and edge–centered values $\mathbf{u}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_j}^{n+\frac{1}{2}}$, $j = 1, ..., D$ [10], and make this field discrete divergence-free with a MAC projection [14].

3. Compute a nonconservative but stable flux difference, a conservative but unstable flux difference, and a stable hybrid flux difference for the hyperbolic treatment of $\mathbf{u}_t = -\mathbf{u} \cdot \nabla \mathbf{u}$ [3].

4. Modify the hybrid field $\mathbf{u} \cdot \nabla \mathbf{u}$ so that it obeys global conservation, i.e., so that $\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} = 0$ is equivalent to the mathematically identical conservation form $\mathbf{u}_t + \nabla \cdot (\mathbf{u}\mathbf{u}) = 0$ in the weak sense.

5. Extrapolate $\mathbf{u} \cdot \nabla \mathbf{u}$ and lagged estimate $\nabla P^{n-\frac{1}{2}}$ to Ω^{n+1}. On Ω^{n+1} solve the heat equation $\mathbf{u}_t = \nu \Delta \mathbf{u} + \mathbf{f}$ with source term $\mathbf{f} = -\nabla P - \mathbf{u} \cdot \nabla \mathbf{u}$;

$$\tilde{\mathbf{u}} = \mathcal{L}_{\text{TGA}} \left(\mathbf{u}^n, \ -(\nabla P)^{n-\frac{1}{2}} - (\mathbf{u} \cdot \nabla \mathbf{u})^{n+\frac{1}{2}} \right) \tag{1.3}$$

where $\mathcal{L}_{\text{TGA}}$ is a particular discretization of the heat operator defined later by (2.49).

6. Make $\mathbf{u}^{n+1}$ discrete divergence free with a cell-centered projection $\mathbb{P}$ (to be defined by (2.6)). The projection computes $\nabla P^{n+\frac{1}{2}}$ on Ω^{n+1}:

$$\mathbf{u}^* = \tilde{\mathbf{u}} + \Delta t (\nabla P)^{n-\frac{1}{2}} \tag{1.4a}$$

$$\mathbf{u}^{n+1} = \mathbb{P}(\mathbf{u}^*) \tag{1.4b}$$

$$(\nabla P)^{n+\frac{1}{2}} = \frac{1}{\Delta t}(\mathbf{I} - \mathbb{P})(\mathbf{u}^*). \tag{1.4c}$$

In section 2 additional details of the algorithm will be presented, with emphasis on those aspects that are new to this work. We will emphasize the algorithm as implemented on a single grid. The components of this algorithm have been shown elsewhere to operate on a hierarchy of nested grids, enabling an adaptive mesh capability. Our implementation includes this capability, and runs in 2D and 3D with SIMD parallelism. A numerical demonstration of convergence rates is presented in section 3.

2. Algorithm details. In subsection 2.1 the existence of a Hodge decomposition for the moving domain problem is described. This discussion justifies the projections

FIG. 1.1. *Newly uncovered cells. Domain boundary $\delta\Omega^n$ is shown with a dashed curve; Ω^n is the enclosed volume, and domain boundary $\delta\Omega^{n+1}$ is shown with the solid curve. The region $\Omega^{n+1} \setminus \Omega^n$ (shaded) contains fluid at t^{n+1} but not at t^n; it is a newly uncovered region. If a cell contains a newly uncovered region, and also contains fluid at time t^n, then the value of the field in Ω^{n+1} is copied from the same cell in Ω^n. But, if a newly uncovered region does not contain t^n values, the values in the extended domain must be estimated by extrapolation. Such cells are indicated with check marks.*

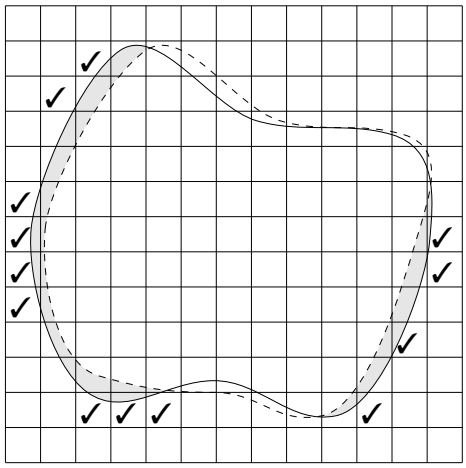
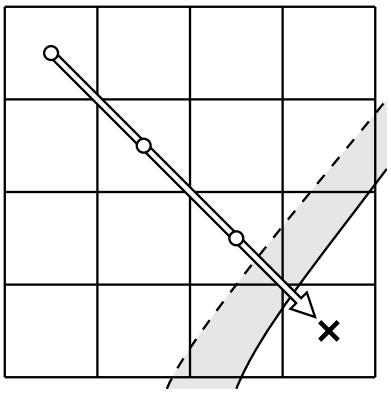


FIG. 1.2. *Extrapolation to newly uncovered cells. $\delta\Omega^n$ is shown as a dashed curve, and $\delta\Omega^{n+1}$ is a solid curve. Symbol X indicates the cell center of a newly uncovered cell. The arrow is aligned with the vector comprised of values 0 and ± 1 that is most nearly parallel the normal to $\delta\Omega^{n+1}$. Points along that arrow (open circles) are used to construct a quadratic, i.e., third-order, extrapolation polynomial.*



used in outline steps 2 and 6. The implementation of the projection for cell-centered $\mathbf{u}$ has been described in [37] and implementation details related to adaptive meshes are given by [22, 23].

Next, in §2.2 the high-order Godunov approach to computing edge– and time–centered values $\mathbf{u}^{n+\frac{1}{2}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$, outline step 2, is described.

In subsection 2.3 the treatment of $\mathbf{u} \cdot \nabla \mathbf{u}$ as a hyperbolic update is described. This includes the stable and conservative forms mentioned in outline step 3, and the conservation property enforced in outline step 4.

The moving domain heat problem employed in outline step 5 was first published by McCorquodale et al. [24]. They demonstrate numerically that on a single domain Ω^{n+1} one can discretize the heat problem on time interval $[t^n, t^{n+1}]$, using specially constructed boundary conditions and extrapolated source terms and initial conditions. In subsection 2.4 we present a theoretical justification for that method.

Finally, in 2.5 we present some details on the construction of geometric terms used to define the quadratures underlying the solution to Poisson's equation (projection), the Helmholtz equation (heat), and the treatment of $\mathbf{u} \cdot \nabla \mathbf{u}$ as a hyperbolic source term on a moving domain. We use an idea due to Ligocki et al. [20] that derives geometric information using a hierarchical application of the divergence theorem. Our implementation is entirely new and differs from theirs by including some relevant inequality constraints. The specialization of that approach is described in the case that the primary source of geometric information is a discretized distance function.

2.1. Hodge projection on a moving domain. To implement a projection method on a moving domain, Trebotich and Colella [11, 37] decompose a vector field $\mathbf{w}$ into three components:

$$\mathbf{w} = \underbrace{\mathbf{v} + \nabla\theta}_{\mathbf{u}} + \nabla\phi \tag{2.1a}$$

$$\Delta\theta = 0 \tag{2.1b}$$

$$\nabla \cdot \mathbf{v} = 0. \tag{2.1c}$$

In the context of incompressible Navier-Stokes, $\mathbf{u}$ is a divergence-free velocity field, consisting of a vorticity-carrying component $\mathbf{v}$ and an incompressible potential flow $\nabla\theta$. $\nabla\phi$ is the gradient of a potential, which can be used to determine ∇P. The boundary conditions for this decomposition are

1. moving walls:

$$\mathbf{n} \cdot \mathbf{v} = 0 \tag{2.2a}$$

$$\mathbf{n} \cdot \nabla\theta = \mathbf{n} \cdot \mathbf{s} \tag{2.2b}$$

$$\mathbf{n} \cdot \nabla\phi = \mathbf{n} \cdot (\mathbf{w} - \mathbf{s}); \tag{2.2c}$$

2. inflow boundaries:

$$\mathbf{v} = 0 \tag{2.3a}$$

$$\mathbf{n} \cdot \nabla\theta = \mathbf{u}_0(\mathbf{x}, t) \quad \text{prescribed} \tag{2.3b}$$

$$\mathbf{n} \cdot \nabla\phi = \mathbf{n} \cdot (\mathbf{w} - \mathbf{u}_0); \tag{2.3c}$$

3. outflow boundaries:

$$\mathbf{n} \cdot \nabla\mathbf{v} = 0 \tag{2.4a}$$

$$\mathbf{n} \cdot \nabla\theta = \bar{\mathbf{u}}_{\text{out}} \tag{2.4b}$$

$$\phi = 0. \tag{2.4c}$$

Here $\bar{\mathbf{u}}_{\text{out}}$ is the average outflow velocity given by conservation over the entire domain. These boundary conditions with $\mathbf{u} = \mathbf{v} + \nabla\theta$ are equivalent to the boundary conditions (1.2) of our problem. The Trebotich-Colella decomposition is solvable: the θ equation is well posed without null space, and $\mathbf{w} - \nabla\theta$ has boundary conditions compatible

with the Hodge decomposition (e.g., [5]). Therefore, $\mathbf{v}$, $\nabla\theta$ and $\nabla\phi$ can be determined uniquely. A projection in this framework is accomplished by

$$\phi: \quad \Delta\phi = \nabla\cdot(\mathbf{w}-\nabla\theta) \tag{2.5a}$$

$$\mathbf{v} = (\mathbf{w}-\nabla\theta) - \nabla\phi. \tag{2.5b}$$

The existence of decomposition (2.1a) does not require explicit determination of potential θ. Instead,

$$\phi: \quad \Delta\phi = \Delta(\phi+\theta) = \nabla\cdot\mathbf{w} \tag{2.6a}$$

$$\mathbf{u} = \mathbf{w} - \nabla\phi, \tag{2.6b}$$

or

$$\mathbf{u} = \mathbb{P}(\mathbf{w}), \tag{2.6c}$$

follows directly by application of (2.1b) to (2.5). The boundary conditions for projection (2.6) are

1. moving walls:

$$\mathbf{n}\cdot\mathbf{u} = \mathbf{n}\cdot\mathbf{s} \tag{2.7a}$$

$$\mathbf{n}\cdot\nabla\phi = \mathbf{n}\cdot(\mathbf{w}-\mathbf{s}); \tag{2.7b}$$

2. inflow boundaries:

$$\mathbf{n}\cdot\mathbf{u} = \mathbf{u}_0(\mathbf{x},t) \quad \text{prescribed} \tag{2.8a}$$

$$\mathbf{n}\cdot\nabla\phi = \mathbf{n}\cdot(\mathbf{w}-\mathbf{u}_0); \tag{2.8b}$$

3. outflow boundaries:

$$\mathbf{n}\cdot\nabla\mathbf{u} = 0 \tag{2.9a}$$

$$\phi = 0. \tag{2.9b}$$

These match (1.2) on $\mathbf{u}$, and for ϕ are identical to (2.1a).

Trebotich and Colella raise two concerns regarding the application of the Hodge decomposition to moving domains [11, 37]. The first is over boundary conditions, but as shown above, the existence of their velocity decomposition makes a Hodge decomposition with boundary conditions (2.7,2.8,2.9) viable. Second, they object to the use of a discrete projection that does not commute with the discrete PDE operators. While it is true that these discrete operators do not commute because of the boundary conditions on the discrete divergence, that property is not essential to the success of the method. If we assume that the mixed derivative $\mathbf{u}_{xt}$ exists and is C^0, then the *differential* operators $\nabla\cdot$ and $\partial/\partial t$ commute [35] and, without recourse to discretization, the governing PDE gives

$$\Delta P = \nabla\cdot(\nu\Delta\mathbf{u} - \mathbf{u}\cdot\nabla\mathbf{u}) \tag{2.10}$$

with boundary condition

$$\mathbf{n}\cdot\nabla P = \mathbf{n}\cdot\left(\nu\Delta\mathbf{u} - \mathbf{u}\cdot\nabla\mathbf{u} - \frac{\partial\mathbf{u}}{\partial t}\right). \tag{2.11}$$

This assumption on $\mathbf{u}_{xt}$ is required as well in the fixed-domain case (e.g., [8, Eq. (2′)]).

If

$$\mathbf{w} = \mathbf{u}^* \approx \mathbf{u}^n + \Delta t(\nu\Delta\mathbf{u} - \mathbf{u}\cdot\nabla\mathbf{u})^{n+\frac{1}{2}} \tag{2.12a}$$

$$= \mathbf{u}^{n+1} + \Delta t\nabla P^{n+\frac{1}{2}} + \mathcal{O}(\Delta t^3) + \mathcal{O}(h^2) \tag{2.12b}$$

(see (1.4a)), then with $\phi \approx \Delta t P$ and $\mathbf{u} = \mathbf{s}$ on $\delta\Omega$, the linear problems

$$\phi: \quad \Delta\phi = \nabla\cdot\mathbf{w} \tag{2.13a}$$

$$\mathbf{n}\cdot\nabla\phi = \Delta t\mathbf{n}\cdot\nabla P^{n+\frac{1}{2}} \quad \text{on } \delta\Omega^{n+1} \tag{2.13b}$$

$$\mathbf{n}\cdot\mathbf{w} = \mathbf{n}\cdot[\mathbf{s}^{n+1} + \Delta t\nabla P^{n+\frac{1}{2}}] \quad \text{on } \delta\Omega^{n+1} \tag{2.13c}$$

and

$$\phi: \quad \Delta\phi = \nabla\cdot\mathbf{w} \tag{2.14a}$$

$$\mathbf{n}\cdot\nabla\phi = 0 \quad \text{on } \delta\Omega^{n+1} \tag{2.14b}$$

$$\mathbf{n}\cdot\mathbf{w} = \mathbf{n}\cdot\mathbf{s}^{n+1} \quad \text{on } \delta\Omega^{n+1} \tag{2.14c}$$

are equivalent to $\mathcal{O}(\Delta t^3) + \mathcal{O}(h^2)$. The former (2.13) is the physical problem to be solved; the latter (2.14) is the Hodge decomposition we implement, and whose existence and uniqueness is addressed above. This approach amounts to placing the inhomogeneous boundary condition due to the moving domain in the divergence of velocity on the right-hand side of the Poisson's equation and solving the homogeneous (Neumann) problem for the pressure. The same approach maps true inflow conditions to $\mathbf{n}\cdot\mathbf{w} = \mathbf{u}_0$ and $\mathbf{n}\cdot\nabla\phi = 0$. For the outflow, conditions $\mathbf{n}\cdot\nabla\mathbf{w} = 0$ and $\phi = 0$ are literal. This discussion has used the time centering corresponding to the cell-centered projection of outline step 6. The MAC projection (outline step 2) is entirely analogous.

2.2. High-order Godunov advection. The computation of $\mathbf{u}^{n+\frac{1}{2}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$ is based on an adaptation of the embedded boundary method for hyperbolic PDEs [10]. It is a three-step process:

I. In the first step, cell-centered velocities $\mathbf{u}^n_{\mathbf{i}}$ are averaged to edges $\mathbf{u}^n_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$, and this velocity field is used to resolve Riemann problems in an advective calculation. First, the velocity is extrapolated to faces with upwind characteristics,

$$\mathbf{u}^{\pm d}_{\mathbf{i}} = \mathbf{u}^n_{\mathbf{i}} \pm \frac{1}{2}\min\left(1 \mp (\mathbf{e}_d\cdot\mathbf{u}^n_{\mathbf{i}})\frac{\Delta t}{h},\ 1\right)(\delta_d\mathbf{u}^n)_{\mathbf{i}} + \frac{\Delta t}{2}(\nu\Delta\mathbf{u}^n)_{\mathbf{i}}. \tag{2.15}$$

This initial extrapolation does not include transverse derivatives or the pressure gradient. δ is a difference operator using van Leer [40] limiting:

$$\delta_d(u) = \begin{cases} \delta_d^{\mathrm{vL}}(u) & \text{if } (u_{\mathbf{i}+\mathbf{e}_d} - u_{\mathbf{i}})(u_{\mathbf{i}} - u_{\mathbf{i}-\mathbf{e}_d}) > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{2.16}$$

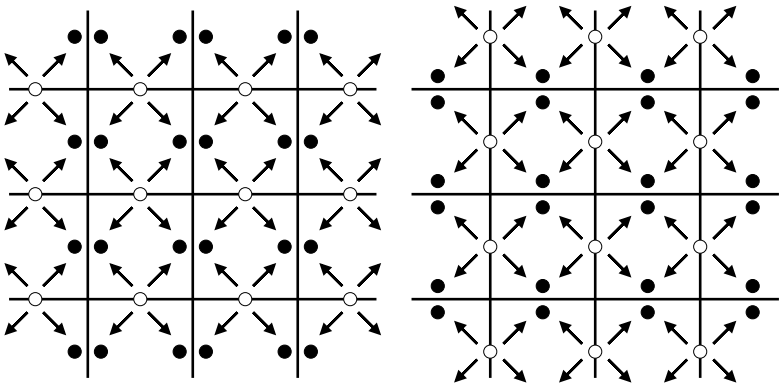
$$\delta_d^{\mathrm{vL}}(u) = \operatorname{sign}(u_{\mathbf{i}+\mathbf{e}_d} - u_{\mathbf{i}-\mathbf{e}_d}) \times \min\left(2\left|u_{\mathbf{i}} - u_{\mathbf{i}-\mathbf{e}_d}\right|,\ 2\left|u_{\mathbf{i}+\mathbf{e}_d} - u_{\mathbf{i}}\right|,\ \frac{1}{2}\left|u_{\mathbf{i}+\mathbf{e}_d} - u_{\mathbf{i}-\mathbf{e}_d}\right|\right). \tag{2.17}$$

$\mathbf{u}^{+d}_{\mathbf{i}}$ is the value of velocity extrapolated to the right side of cell $\mathbf{i}$ in direction d, and $\mathbf{u}^{-d}_{\mathbf{i}+\mathbf{e}_d}$ is the value extrapolated to the same edge from cell $\mathbf{i}+\mathbf{e}_d$. A single-valued

result is obtained by resolving the Riemann problem, which amounts to upwinding

$$\bar{\mathbf{u}}^{n+\frac{1}{2}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \begin{cases} \mathbf{u}^{+d}_{\mathbf{i}} & \text{if } (\mathbf{e}_d \cdot \mathbf{u})^n_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} > 0 \\ \mathbf{u}^{-d}_{\mathbf{i}+\mathbf{e}_d} & \text{if } (\mathbf{e}_d \cdot \mathbf{u})^n_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} < 0 \\ \frac{1}{2}\left(\mathbf{u}^{+d}_{\mathbf{i}} + \mathbf{u}^{-d}_{\mathbf{i}+\mathbf{e}_d}\right) & \text{if } (\mathbf{e}_d \cdot \mathbf{u})^n_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = 0. \end{cases} \tag{2.18}$$

FIG. 2.1. *Transverse flux correction in 2D. Double-valued edge states* $\mathbf{u}^{\pm d}$ *are indicated by filled circles, and single-valued states* $\bar{\mathbf{u}}$ *are indicated by open circles. Differences in* $\bar{\mathbf{u}}$ *across a given cell provide flux correction to the states* $\mathbf{u}^{\pm d}$ *associated with that cell, but in transverse directions.*



The output of this Riemann problem is used to provide transverse flux corrections. In 2D (Figure 2.1),

$$\mathbf{u}^{\pm d}_{\mathbf{i}} := \mathbf{u}^{\pm d}_{\mathbf{i}} - \frac{\Delta t}{2h}\left(\bar{\mathbf{u}}^{n+\frac{1}{2}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_{d'}} - \bar{\mathbf{u}}^{n+\frac{1}{2}}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_{d'}}\right), \quad d' \neq d \tag{2.19}$$

followed by another Riemann solution. In 3D the transverse flux correction is more complicated [32].

II. A discrete MAC projection is used to make the advected velocities divergence-free.

$$\Phi: \quad \Delta\Phi = \nabla \cdot \bar{\mathbf{u}}; \qquad (\nabla \cdot \bar{\mathbf{u}})_{\mathbf{i}} = \frac{1}{h}\sum_d \left(\bar{\mathbf{u}}^{n+\frac{1}{2}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - \bar{\mathbf{u}}^{n+\frac{1}{2}}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}\right) \tag{2.20a}$$

$$\mathbf{e}_d \cdot \mathbf{u} = \mathbf{e}_d \cdot \bar{\mathbf{u}} - \nabla_d \Phi. \tag{2.20b}$$

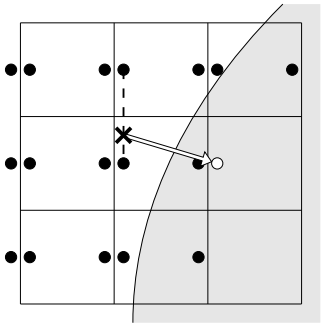
This projection only affects the normal component of the edge velocities.

III. The third step repeats step I, but the velocity used to judge upwind direction in the Riemann problem is the divergence-free edge velocity computed in step II. In this step the normal velocity components are not changed, but the transverse ones are. Finally, these transverse components are corrected to account for the pressure gradient computed in II. In 2D,

$$\begin{aligned} \mathbf{e}_{d'} \cdot \mathbf{u}^{n+\frac{1}{2}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} := \mathbf{e}_{d'} \cdot \mathbf{u}^{n+\frac{1}{2}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - \frac{1}{4}\Big[&(\nabla_{d'}\Phi)_{\mathbf{i}+\frac{1}{2}\mathbf{e}_{d'}} + (\nabla_{d'}\Phi)_{\mathbf{i}+\mathbf{e}_d+\frac{1}{2}\mathbf{e}_{d'}} + \\ &(\nabla_{d'}\Phi)_{\mathbf{i}-\frac{1}{2}\mathbf{e}_{d'}} + (\nabla_{d'}\Phi)_{\mathbf{i}+\mathbf{e}_d-\frac{1}{2}\mathbf{e}_{d'}}\Big], \end{aligned} \tag{2.21}$$

where $d' \neq d$ is the transverse direction. The generalization to 3D is straightforward.

FIG. 2.2. *Covered edge calculation, illustrated in 2D for the case of $\hat{y}$-side edges. Closed circles indicate edge values calculated by the 1D advection algorithm described here, though modified to use one-sided differences near boundaries. The open circle indicates an exterior covered edge, in this case on the right side of an edge. This right-edge covered value is extrapolated from right-edge uncovered values by interpolation (dashed line), and extrapolation in the direction of the interface normal (arrow), using the cell-centered gradient. When the uncovered values are modified to account for transverse flux correction, this calculation is repeated so the covered edge value also includes transverse corrections.*



The extension of this algorithm to embedded boundary geometries is described in [10]. One change is to employ one-sided differences where the data does not support centered stencils. Another concerns the determination of so-called covered-edge values. Covered edges are those edges of irregular cells which are not in contact with the fluid. For these edges, the upwind characteristic tracing step provides a single edge value on the fluid side of the edge. The value on the side opposite the fluid is obtained by extrapolation from edge values interior to the domain (Figure 2.2); see [10, §5.2] for details.

2.3. Hyperbolic step. We are interested here in a formulation of $\mathbf{u} \cdot \nabla \mathbf{u}$ that is consistent with the hyperbolic split of the Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = 0 \tag{2.22}$$

with $\mathbf{F} \equiv \mathbf{uu}$, $\nabla \cdot \mathbf{F} = \mathbf{u} \cdot \nabla \mathbf{u}$ when $\nabla \cdot \mathbf{u} = 0$. For this hyperbolic equation, one has a discretization

$$\mathbf{u}^{n+1}_{\text{nonconservative}} = \mathbf{u}^n - \frac{\Delta t}{h} (\mathbf{DF})^{\text{nc}} \tag{2.23}$$

with $\mathbf{DF}$ a flux difference which we approximate by

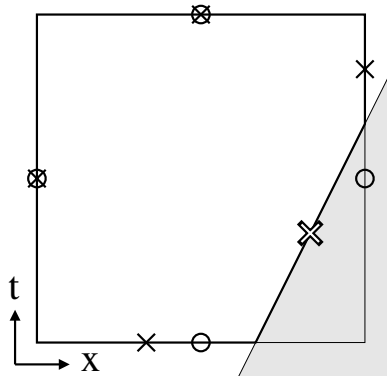
$$(\mathbf{DF})^{\text{nc}}_{\mathbf{i}} = \sum_d^D \frac{1}{h} \left(u^{n+\frac{1}{2}}_{d,\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \mathbf{u}^{n+\frac{1}{2}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - u^{n+\frac{1}{2}}_{d,\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \mathbf{u}^{n+\frac{1}{2}}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \right) \tag{2.24}$$

This discretization is second-order accurate in regular cells, but not consistent in cut cells. It is stable in both cases.

A conservative discretization of the conservation law on the irregular control volume comes from the space-time integration over the fluid in an irregular cell

(2.25)

FIG. 2.3. *Centerings: centers (open circles) and centroids (crosses). In regular domains, the discretization relies on centered quantities. A convergent stencil in irregular domains uses centroid-centered quantities.*



$$0 = \int_{t^n}^{t^{n+1}} dt \int_{\Omega_{\mathbf{i}}(t)} dV \left(\frac{\partial}{\partial t}, \nabla\right) \cdot (\mathbf{u}, \mathbf{F}) \approx \kappa_{\mathbf{i}}^{n+1} h^D \mathbf{u}_{\mathbf{i}}^{n+1} - \kappa_{\mathbf{i}}^n h^D \mathbf{u}_{\mathbf{i}}^n +$$
$$\Delta t h^{D-1} \sum_d^D (\alpha_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \mathbf{F}_{d,\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - \alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \mathbf{F}_{d,\mathbf{i}-\frac{1}{2}\mathbf{e}_d}) + A_{\mathbf{i},\mathrm{EB}} \mathbf{n}_{\mathbf{i},EB} \cdot (\mathbf{u}, \mathbf{F})_{\mathbf{i},\mathrm{EB}}$$

where

$$\Omega_{\mathbf{i}}(t) = \Omega(t) \cap [h\mathbf{i},\ h(\mathbf{i}+\mathbf{1})] \tag{2.26}$$

is the fluid-occupied volume of cell $\mathbf{i}$ at time t. Subscript EB denotes that the object is located on the embedded boundary, and EB will be used also as an abbreviation. Here κ denotes a volume fraction,

$$\kappa_{\mathbf{i}}^n = \frac{1}{h^D} \int_{\Omega_{\mathbf{i}}(t^n)} dV; \tag{2.27}$$

α a space-time area fraction (aka. "aperture"),

$$\alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} = \frac{1}{h^{D-1}\Delta t} \int_{t^n}^{t^{n+1}} dt \int_{\delta\Omega_{\mathbf{i}}(t) \cap \{\mathbf{x} | x_d = h i_d\}} dA; \tag{2.28}$$

and A_{EB} is the space-time area of the EB. $\mathbf{n}_{\mathrm{EB}}$ is the unit normal in $\mathbb{R}^{D+1}$. The $D+1$ components of $A_{\mathrm{EB}}\mathbf{n}_{\mathrm{EB}}$ can be determined from the condition $\mathrm{div}(\mathbf{e}_i) = 0$ for each of the $D+1$ directions i, giving

$$\kappa_{\mathbf{i}}^{n+1} \underset{\mathrm{cent}\mathbf{i}}{\mathbf{u}}^{n+1} = \kappa_{\mathbf{i}}^n \underset{\mathrm{cent}\mathbf{i}}{\mathbf{u}}^{n} - \frac{\Delta t}{h} \sum_d^D (\alpha_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \mathbf{F}_{d,\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - \alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \mathbf{F}_{d,\mathbf{i}-\frac{1}{2}\mathbf{e}_d}) \tag{2.29}$$
$$-(\kappa_{\mathbf{i}}^n - \kappa_{\mathbf{i}}^{n+1}) \mathbf{u}_{\mathbf{i},\mathrm{EB}} - \frac{\Delta t}{h} \sum_d^D (\alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} - \alpha_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}) \mathbf{F}_{d,\mathbf{i},\mathrm{EB}}.$$

Here we have written $\underset{\mathrm{cent}}{\mathbf{u}}$ to emphasize that the centering is at the centroid $\underset{\mathrm{cent}}{\mathbf{x}}$ for (2.29) to be consistent (Figure 2.3);

$$\underset{\mathrm{cent}\mathbf{i}}{\mathbf{x}}^{n} = \frac{1}{h^D \kappa_{\mathbf{i}}^n} \int_{\Omega_{\mathbf{i}}(t^n)} \mathbf{x} dV. \tag{2.30}$$

However, the elliptic operators we use are based on a cell-centered discretization $\underset{\text{cc}}{\mathbf{u}}$, which suggests the modification

$$\underset{\text{cc}}{\mathbf{u}}_{\mathbf{i}}^{n+1} - \underset{\text{cc}}{\mathbf{u}}_{\mathbf{i}}^{n} = -\frac{\Delta t}{h}(\mathbf{DF})^c \tag{2.31}$$

$$\begin{aligned}(\mathbf{DF})^c = \frac{1}{\kappa_{\mathbf{i}}^{n+1}}\Bigg[\frac{h}{\Delta t}\Bigg(&\kappa_{\mathbf{i}}^{n}(\underset{\text{cc}}{\mathbf{x}} - \underset{\text{cent}}{\mathbf{x}})_{\mathbf{i}}^{n}\cdot(\nabla\mathbf{u})_{\mathbf{i}}^{n} - \kappa_{\mathbf{i}}^{n+1}(\underset{\text{cc}}{\mathbf{x}} - \underset{\text{cent}}{\mathbf{x}})_{\mathbf{i}}^{n+1}\cdot(\nabla\mathbf{u})_{\mathbf{i}}^{n}\\ &+(\kappa_{\mathbf{i}}^{n+1}-\kappa_{\mathbf{i}}^{n})\underset{\text{cc}}{\mathbf{u}}_{\mathbf{i}}^{n}\Bigg) + \sum_{d}^{D}(\alpha_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}\mathbf{F}_{d,\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - \alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}\mathbf{F}_{d,\mathbf{i}-\frac{1}{2}\mathbf{e}_d})\\ &+\frac{h}{\Delta t}(\kappa_{\mathbf{i}}^{n}-\kappa_{\mathbf{i}}^{n+1})\mathbf{u}_{\mathbf{i},\text{EB}} + \sum_{d}^{D}(\alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} - \alpha_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d})\mathbf{F}_{d,\mathbf{i},\text{EB}}\Bigg].\end{aligned} \tag{2.32}$$

(2.31) has $\mathcal{O}(h)$ discretization error in irregular cells (when $\kappa < 1$), and is second-order in regular cells. The velocity at the centroid of the EB is $\mathbf{s}(\mathbf{x}, t)$, the prescribed boundary condition (1.2a). Fluxes at the centroids of cell faces are calculated by interpolating the velocity field to the centroid

$$\underset{\text{cent}}{\mathbf{x}}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} = \frac{1}{h^{D-1}\Delta t\alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}}\int_{t^n}^{t^{n+1}} dt\int_{\delta\Omega_{\mathbf{i}}(t)\cap\{\mathbf{x}|x_d=hi_d\}}\mathbf{x}dA \tag{2.33a}$$

$$\underset{\text{cent}}{t}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} = \frac{1}{h^{D-1}\Delta t\alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}}\int_{t^n}^{t^{n+1}} tdt\int_{\delta\Omega_{\mathbf{i}}(t)\cap\{\mathbf{x}|x_d=hi_d\}}dA. \tag{2.33b}$$

The data interpolated is taken from all available data in a 5^D-cell region centered at the point where $\mathbf{F}$ is required. This makes $\mathbf{F}$ on an irregular edge, say $\mathbf{i}+\frac{1}{2}\mathbf{e}_d$ independent of the cell, $\mathbf{i}$ or $\mathbf{i}+\mathbf{e}_d$, that shares it. Interpolation is second order in space and time, and implemented by solving an overdetermined set of linear equations with Householder decomposition.

To make the method stable we employ the hybridized flux difference

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \frac{\Delta t}{h}\left[\kappa^{n+1}(\mathbf{DF})^{\text{c}} + (1-\kappa^{n+1})(\mathbf{DF})^{\text{nc}}\right]. \tag{2.34}$$

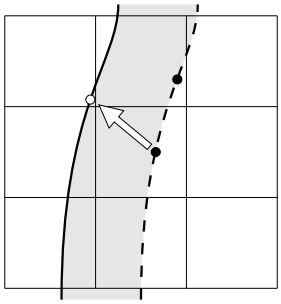
In the limit that cells become regular on $[t^n, t^{n+1}]$ the conservative, nonconservative, and hybrid flux differences are all equivalent to the stable second-order result, and (2.34) reduces to (2.31).

The generalized mass difference is redistributed. The mass excess is

$$\delta\mathbf{m} = h^D\kappa^{n+1}(\mathbf{u}^{n+1} - \mathbf{u}^{n+1}_{\text{unstable}}) = \Delta t h^{D-1}\kappa^{n+1}(1-\kappa^{n+1})\left[(\mathbf{DF})^{\text{c}} - (\mathbf{DF})^{\text{nc}}\right]. \tag{2.35}$$

The negative of this quantity is to be distributed in a volume-weighted sense to neighboring cells [3, 29, 25]. Let $\tilde{\mathbf{u}}$ be $\mathbf{u}^{n+1}$ evaluated by (2.34), then modified by redistribution. Then $h(\mathbf{u}^n - \tilde{\mathbf{u}})/\Delta t$ is what we refer to in outline step 4 as a conservation-preserving calculation of $\mathbf{u}\cdot\nabla\mathbf{u}$.

FIG. 2.4. *Extrapolation of boundary conditions. The dashed curve is $\delta\Omega^n$, and boundary conditions are known at the centroid of EB segments in each cell. The solid curve is $\delta\Omega^{n+1}$, and boundary conditions are needed at the centroids of this EB in each cell. For each t^{n+1} centroid (e.g., the open circle), the neighboring cell with the greatest boundary area is chosen. In this picture there are two candidates (closed circles). The boundary condition is then extrapolated using the inner product of the cell-centered gradient in the t^n cell and the relative coordinates (arrow).*



2.4. The time-dependent heat problem. For the problem

$$u_t = K\Delta u + f \quad \mathbf{x} \in \Omega(t) \tag{2.36a}$$

$$u(\mathbf{x}, t^n) = u_0(\mathbf{x}) \quad \mathbf{x} \in \Omega(t^n) \tag{2.36b}$$

$$u(\mathbf{x}, t) = u_{\rm bc}(\mathbf{x}, t) \quad \mathbf{x} \text{ on } \delta\Omega(t) \tag{2.36c}$$

McCorquodale et al. [24] propose the following algorithm:

1. Interpolate the boundary conditions $u_{\rm bc}(\mathbf{x}, t^n)$ to the boundary $\delta\Omega^{n+1}$ with

$$u_{\rm bc}^{\rm interp}(\mathbf{x}') = u_{\rm bc}(\mathbf{x}, t^n) + (\mathbf{x}' - \mathbf{x}) \cdot \nabla u_0(\mathbf{x}', t^n), \tag{2.37}$$

where $\mathbf{x}'$ on $\delta\Omega^{n+1}$, $\mathbf{x}$ on $\Delta\Omega^n$, and $|\mathbf{x} - \mathbf{x}'|$ is $\mathcal{O}(h)$. Specifically, let $\mathbf{i}$ be the cell containing $\mathbf{x}$, and let $\mathbf{i}'$ be the cell containing $\mathbf{x}'$. For a given $\mathbf{i}$, cell $\mathbf{i}'$ is chosen to be the neighbor of $\mathbf{i}$ with greatest boundary area (Figure 2.4).

2. On $\delta\Omega^{n+1}$, boundary conditions for any time in $[t^n, t^{n+1}]$ are obtained by linear interpolation of $u_{\rm bc}^{\rm interp}(\mathbf{x})$ and $u_{\rm bc}(\mathbf{x}, t^{n+1})$.
3. Extrapolate u_0 to Ω^{n+1} using the approach described in outline step 1.
4. Extrapolate $f(\mathbf{x}, t^{n+\frac{1}{2}})$ from $\Omega^{n+\frac{1}{2}}$ to Ω^{n+1} with this same procedure.
5. On Ω^{n+1}, solve the heat equation by the method of Twizell et al. [38]:

$$\begin{aligned}\mathbf{u}^{n+1} = (\mathbf{I} - \mu_1\Delta t\Delta_1^h)^{-1}(\mathbf{I} - \mu_2\Delta t\Delta_2^h)^{-1} \times \\ \left[(\mathbf{I} + \mu_3\Delta t\Delta_3^h)\mathbf{u}^{n,\rm extrap} + (\mathbf{I} + \mu_4\Delta t\Delta_4^h)\Delta t\mathbf{f}^{n+\frac{1}{2},\rm extrap}\right],\end{aligned} \tag{2.38}$$

with $\mu_1 = \mu_2 = 1 - 1/\sqrt{2}$, $\mu_3 = \sqrt{2} - 1$, and $\mu_4 = \sqrt{2} - 3/2$. $\mathbf{u}^{n,\rm extrap}$ is the field $\mathbf{u}$ centered at time t^n, but extrapolated from Ω^n to Ω^{n+1}, and likewise $\mathbf{f}^{n+\frac{1}{2},\rm extrap}$ is the source term f centered at $t^{n+\frac{1}{2}}$ and extrapolated from $\Omega^{n+\frac{1}{2}}$ to Ω^{n+1}. Δ_1^h is the discrete Laplacian on Ω^{n+1} with boundary conditions at t^{n+1}; Δ_2^h has boundary conditions at $t^{n+1} - \mu_1\Delta t$ (by interpolation); and Δ_3^h has boundary conditions at t^n. The boundary conditions on Δ_4^h are homogeneous Dirichlet.

A justification of this algorithm follows.

The ODE

$$u' = A(t)u + f(t) \tag{2.39}$$

has solution

$$u^{n+1} = R(\Delta t)u^n + R(\Delta t)\int_0^{\Delta t} R^{-1}(s)f(s)ds \tag{2.40}$$

where R is an integrating factor

$$R(\Delta t) = \exp\left(\int_{t^n}^{t^n+\Delta t} A(\tau)d\tau\right). \tag{2.41}$$

Expanding A in a Taylor series,

$$A(t^n + s) = \sum_{i=0}^{\infty} A_i s^i, \tag{2.42}$$

facilitates constructing an approximation to R:

$$R(\Delta t) \approx \frac{1 + \mu_3\alpha_3\Delta t}{(1 - \mu_1\alpha_1\Delta t)(1 - \mu_2\alpha_2\Delta t)}, \tag{2.43}$$

where

$$\mu_1 = \mu_2 = 1 - 1/\sqrt{2} \tag{2.44a}$$

$$\mu_3 = \sqrt{2} - 1. \tag{2.44b}$$

These coefficients μ_i minimize the discretization error of this approximation in the case that A is independent of time: the case described by Twizell et al. [38] (Twizell et al. introduce a factor ϵ of order machine precision to lift the degeneracy of (2.44a) and enable a partial fraction representation of (2.43). McCorquodale et al. [24] include this ϵ factor but do not use partial fractions). The factors α_i are different time centerings of $A(t)$:

$$\alpha_1 = A_0 + A_1 c_1 \Delta t \tag{2.45a}$$

$$\alpha_2 = A_0 + A_1 c_2 \Delta t \tag{2.45b}$$

$$\alpha_3 = A_0 + A_1 c_3 \Delta t \tag{2.45c}$$

with coefficients c_i to be determined. When A is time-varying, the approximation to $R(\Delta t)$ differs from (2.41) by $\mathcal{O}(\Delta t^3)$ provided

$$(c_1 + c_2)(2 - \sqrt{2}) + 2c_3(\sqrt{2} - 1) = 1. \tag{2.46}$$

The solution [24]

$$c_1 = 1 \tag{2.47a}$$

$$c_2 = 1/\sqrt{2} \tag{2.47b}$$

$$c_3 = 0 \tag{2.47c}$$

satisfies this consistency requirement, but not uniquely. Expanding $f(t)$ in a Taylor series about $t^{n+1/2}$ leads to a discretization of that source term. Combined,

$$u^{n+1} = (1 - \mu_1\alpha_1\Delta t)^{-1}(1 - \mu_2\alpha_2\Delta t)^{-1}\left[(1 + \mu_3\alpha_3\Delta t)u^n + (1 + \mu_4\alpha_4\Delta t)\Delta t f^{n+1/2}\right] \tag{2.48}$$

where $\mu_4 = \sqrt{2} - 3/2$, and α_4 is an arbitrary centering of A.

The choice $c_1 = 1$ is optimal in that the final implicit solve will satisfy its given boundary conditions exactly. An interpretation of this result is that the α_3 operation carries u^n to $u^{n+\mu_3}$, then the α_2 operation carries the solution to $u^{n+1-\mu_1}$, with the final operation α_1 terminating at u^{n+1}. This suggests that $0 \le c_3 \le \mu_3/2$ in order that $\mu_3 \le c_2 \le \mu_3 + \mu_2$, i.e., that the boundary conditions lie within the interval of the associated operation.

Connecting ODE (2.39) to the heat PDE by the method of lines, this analysis suggests

(2.49a)
$$\mathbf{u}^{n+1} = (\mathbf{I} - \mu_1\mathcal{L}_1\Delta t)^{-1}(\mathbf{I} - \mu_2\mathcal{L}_2\Delta t)^{-1}\left[(\mathbf{I} + \mu_3\mathcal{L}_3\Delta t)\mathbf{u}^n + (\mathbf{I} + \mu_4\mathcal{L}_4\Delta t)\Delta t\mathbf{f}^{n+\frac{1}{2}}\right],$$

which we abbreviate as

$$\mathbf{u}^{n+1} = \mathcal{L}_{\mathrm{TGA}}\left(\mathbf{u}^n,\ \mathbf{f}^{n+\frac{1}{2}}\right): \tag{2.49b}$$

the solution at t^{n+1} to $\mathbf{u}_t = \mathcal{L}\mathbf{u} + \mathbf{f}$. When $\mathcal{L}$ is a negative definite operator, this discretization is L_0 stable and second-order accurate in time. Since $\mu_1 + \mu_2 + \mu_3 = 1$, the principle of superposition requires that boundary conditions on $\mathcal{L}_4$ be homogeneous. It remains to be shown that all operators $\mathcal{L}_i$ can be discretized on the domain Ω^{n+1} to $\mathcal{O}(h^2)$. Operators $\mathcal{L}_i$ must be centered correctly, as given by (2.45,2.47) to second-order in time, except for $\mathcal{L}_4$ which may be first-order in time.

Consider the heat equation

$$\begin{aligned} u_t &= K\Delta u + f \\ u(\mathbf{x},0) &= u_0(\mathbf{x}) \\ u(\mathbf{x},t) &= u_{\mathrm{bc}}(\mathbf{x},t) \quad \text{on } \delta\Omega. \end{aligned} \tag{2.50}$$

Let $\mathbf{x}^0$ be a point on $\delta\Omega$, and let $\mathbf{x}^1$ be an arbitrary point $\mathcal{O}(h)$ away from $\mathbf{x}^0$. Then a Taylor series expansion gives

$$u(\mathbf{x}^1,t) = u(\mathbf{x}^0,t) + (\mathbf{x}^1 - \mathbf{x}^0)\cdot\nabla u(\mathbf{x}^0,0) + \mathcal{O}(h^2) + \mathcal{O}(h\Delta t). \tag{2.51}$$

Therefore, if Ω^{n+1} and $\Omega(t)$ are close (in the sense that for any point $\mathbf{x}$ on $\delta\Omega^{n+1}$ there is a point $\mathbf{x}'$ on $\delta\Omega(t)$ with $|\mathbf{x} - \mathbf{x}'| = \mathcal{O}(h)$), and if one uses Dirichlet boundary conditions on Ω^{n+1} given by

$$u_{\mathrm{bc}}(\mathbf{x}^{n+1},t) := u_{\mathrm{bc}}(\mathbf{x}(t),t) + (\mathbf{x}^{n+1} - \mathbf{x}(t))\cdot\nabla u_0,$$

and if $\Delta t \propto h$, then the solution at $\mathbf{x}(t)$ on $\delta\Omega(t)$ will be obtained to second order in h.

The solution on the interior of a domain Ω is a linear functional of its boundary conditions, initial conditions, and forcing. For example,

$$\begin{aligned} u(\mathbf{x},t) = &\int_\Omega dV' G(\mathbf{x} \mid \mathbf{x}',t)u_0(\mathbf{x}') + \\ &\int_0^t dt' \int_\Omega dV' G(\mathbf{x} \mid \mathbf{x}', t-t')f(\mathbf{x}',t') \\ &\int_0^t dt' \int_{\delta\Omega} dS' \mathbf{n}'\cdot\nabla' G(\mathbf{x}' \mid \mathbf{x}, t-t')u_{\mathrm{bc}}(\mathbf{x}',t') \end{aligned} \tag{2.52}$$

solves (2.50) where G is the Green's function solving

$$\begin{aligned} G_t &= K\Delta G + \delta(\mathbf{x}-\mathbf{x}') \\ G(\mathbf{x} \mid \mathbf{x}', 0) &= 0 \\ G(\mathbf{x} \mid \mathbf{x}', t) &= 0 \quad \forall\ \mathbf{x} \text{ on } \delta\Omega. \end{aligned} \tag{2.53}$$

Therefore, on a domain Ω differing from Ω^{n+1} by $\mathcal{O}(h)$, where u_0 and f are continued by high-order interpolation, and where u_{bc} is second-order accurate, the solution interior to Ω will be second-order accurate. Solved by a discrete method, the error will be the lower of the order of the method or h^2, in the present case $O(h^2)+O(\Delta t)$ for the solution by forward or backward Euler, and $O(h^2)+O(\Delta t^2)$ embedded in the Twizell et al. framework (2.49a).

The discretization of this heat solver is based on the conservative but unstable discretization of the Laplacian for time-stationary geometries

$$\Delta u = \nabla\cdot\mathbf{F}, \quad \mathbf{F} = \nabla u \tag{2.54a}$$

$$u^{n+1} = u^n + \mathcal{L}_{\mathbf{i}}(\mathbf{u}) \tag{2.54b}$$

(2.54c)

$$\mathcal{L}_{\mathbf{i}}(\mathbf{u}) = \frac{\nu\Delta t}{\kappa_{\mathbf{i}} h}\sum_d^D \left(\left[\alpha_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \mathbf{F}^{n+\frac{1}{2}}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - \alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \mathbf{F}^{n+\frac{1}{2}}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \right] + \left[\alpha_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} - \alpha_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right] \mathbf{F}_{d,\mathbf{i},\text{EB}} \right).$$

Note that while $\mathcal{L}_{\mathbf{i}}$ is unstable in the limit $\kappa_{\mathbf{i}} \to 0$, $\kappa_{\mathbf{i}}\mathcal{L}_{\mathbf{i}}$ is stable. The overall sequence can be written in a stable manner as follows:

$$\boldsymbol{\psi}^1 = \boldsymbol{\kappa}(\mathbf{I}+\mu_4\mathcal{L})\mathbf{f}^{n+1/2} \tag{2.55a}$$

$$\boldsymbol{\psi}^2 = \boldsymbol{\kappa}(\mathbf{I}+\mu_3\mathcal{L})\mathbf{u}^n \tag{2.55b}$$

$$\boldsymbol{\psi}^3 = \Delta t\boldsymbol{\psi}^1 + \boldsymbol{\psi}^2 \tag{2.55c}$$

$$\boldsymbol{\psi}^4 = [\boldsymbol{\kappa}(\mathbf{I}-\mu_2\mathcal{L})]^{-1}\,\boldsymbol{\psi}^3 \tag{2.55d}$$

$$\boldsymbol{\psi}^5 = \boldsymbol{\kappa}\boldsymbol{\psi}^4 \tag{2.55e}$$

$$\mathbf{u}^{n+1} = [\boldsymbol{\kappa}(\mathbf{I}-\mu_1\mathcal{L})]^{-1}\,\boldsymbol{\psi}^5. \tag{2.55f}$$

2.5. Computation of space-time geometry. We base our geometry calculation on a hierarchical application of the divergence theorem proposed by Ligocki et al. [20], here specialized to the case where geometric information is to be determined from cell– and time-centered discrete values of a level set function ψ. This method assumes only that ψ is a sufficiently differentiable level set, not necessarily a distance function.

2.5.1. Governing equations. In D dimensions use the following multiindex convention,

$$\mathbf{x}^{\mathbf{p}} = x_1^{p_1}x_2^{p_2}\cdots x_D^{p_D} \tag{2.56a}$$

$$\mathbf{p}! = p_1!p_2!\cdots p_D! \tag{2.56b}$$

$$\nabla^{\mathbf{r}} = \frac{\partial^{r_1}}{\partial x_1^{r_1}}\frac{\partial^{r_2}}{\partial x_2^{r_2}}\cdots\frac{\partial^{r_D}}{\partial x_D^{r_D}}, \tag{2.56c}$$

and in this application all components of a multiindex are nonnegative. We will say multiindex integer $\mathbf{p}$ is even if all p_i are even, and for the magnitude, $P = |\mathbf{p}| = \sum p_i$, etc.

Consider the volume integral of $\nabla\cdot(\mathbf{x}^{\mathbf{p}}\mathbf{e}_d) = p_d\mathbf{x}^{\mathbf{p}-\mathbf{e}_d}$ with the divergence theorem:

$$p_d\int_V \mathbf{x}^{\mathbf{p}-\mathbf{e}_d}dV = \int_{A_d^+}\mathbf{x}^{\mathbf{p}}dA - \int_{A_d^-}\mathbf{x}^{\mathbf{p}}dA + \int_{A_{\rm EB}}\mathbf{x}^{\mathbf{p}}\mathbf{n}\cdot\mathbf{e}_d dA \tag{2.57}$$

where EB denotes the embedded boundary, and $\mathbf{n}$ is the unit normal vector. With the boundary having curvature, $\mathbf{n}$ is spatially varying. Account for this spatial variance with a truncated Taylor series;

$$p_d\int_V \mathbf{x}^{\mathbf{p}-\mathbf{e}_d}dV - n_d\int_{A_{\rm EB}}\mathbf{x}^{\mathbf{p}}dA = \int_{A_d^+}\mathbf{x}^{\mathbf{p}}dA - \int_{A_d^-}\mathbf{x}^{\mathbf{p}}dA + \sum_{1\le|\mathbf{r}|\le R}\frac{\nabla^{\mathbf{r}}n_d}{\mathbf{r}!}\int_{A_{\rm EB}}\mathbf{x}^{\mathbf{r}+\mathbf{p}}dA + \mathcal{O}(h^{D+R+P}). \tag{2.58}$$

Here V designates a generalized "volume", and A designates a codimension-1 subspace – a generalized "area". In (2.58), $\mathbf{n}$ is the normal to the EB in the space of volume V. This equation expresses moments on $[-h/2,h/2]^D$ and on the codimension-1 EB in terms of higher moments on lower dimensional spaces. Each of these lower dimensional spaces can be analyzed with a similar specialization of (2.58). For example if the area A_{d^+} is bounded by subspaces L (lines)

$$p_{d'}\int_{A_{d^+}}\mathbf{x}^{\mathbf{p}-\mathbf{e}_{d'}}dA - n_{d'}\int_{L_{\rm EB}}\mathbf{x}^{\mathbf{p}}dL = \int_{L_{d'}^+}\mathbf{x}^{\mathbf{p}}dL - \int_{L_{d'}^-}\mathbf{x}^{\mathbf{p}}dL + \sum_{1\le|\mathbf{r}|\le R}\frac{\nabla^{\mathbf{r}}n_{d'}}{\mathbf{r}!}\int_{L_{\rm EB}}\mathbf{x}^{\mathbf{r}+\mathbf{p}}dL + \mathcal{O}(h^{D'+R+P}) \tag{2.59}$$

where in (2.59) $\mathbf{n}$ is the interface normal in the subspace A_{d^+}, and $D' = D-1$ if we consider p_d (the component of $\mathbf{p}$ in the dimension orthogonal to space A_{d^+}) to be zero. (This assumption can be made without loss of generality. If $M(\mathbf{p})$ is a given moment on surface $A_{d^\pm}$ with $p_d = 0$, then $M(\mathbf{p}+k\mathbf{e}_d) = (\pm h/2)^k M(\mathbf{p})$: the generation of moments for which $p_d \neq 0$ is trivial.) Equation (2.58) can be applied as many times as needed until the the subspaces contain trivial normal vectors $\mathbf{n}$: when the space V of (2.58) is 1D, the normal vector is ±1 and has no derivative.

To interpret the order $D'+R+P$, begin by specifying S as the desired order of accuracy. On the original space $R = S-1$, and $P = 0,1$ is required at a minimum to obtain the centroid of the EB. However, with $R = 1$ and $P = 1$, EB moments with $P = 2$ are required on the right hand sides. This causes the maximum P, $P_{\max}$, to depend on S and D' in a systematic way:

$$P_{\max}(D') = S - 1 + [D - \max(D',\ 2)] \qquad D \ge 2, \tag{2.60}$$

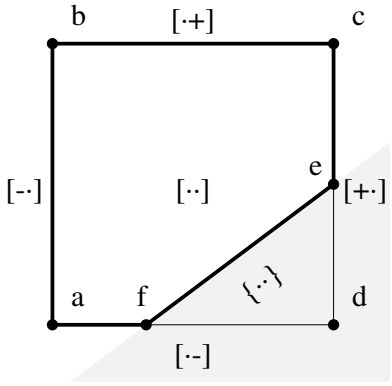
and, for each magnitude $P = 0, ..., P_{\max}$,

$$R = \max(S-1-P,\ 0). \tag{2.61}$$

Table 2.1 displays some convergence results in multiple dimensions for the case $S = 2$.

2.5.2. Order of operations. For each dimension, the system of equations implied by (2.58) is overdetermined and nonsingular. Ligocki et al. propose evaluating this hierarchical system in a particular way, grouping equations on a common subspace and with common P. This makes each overdetermined set small, minimizing

FIG. 2.5. *Notation for 2D example. The fluid region abcde is denoted* $[\cdot\cdot]$*; the EB* $\{\cdot\cdot\}$*,* $\overline{ef}$*, separates the fluid from the shaded exterior region def. The 1D subregion* $[+\cdot]$ *is the line segment* $\overline{cd}$*, etc. The calculation begins with* $P = 0$ *moments on the 1D subregions, e.g.,* $(00)_{[\cdot-]} = af$*, then the* $P = 1$ *moments; e.g.,* $(10)_{[\cdot-]} = x_f^2/2 - h^2/8$*, and* $(01)_{[\cdot-]} = -(af)h/2$ *which is simply* $(00)_{[\cdot-]}$ *multiplied by the* x_1 *coordinate of the edge,* $-h/2$*.*



the cumulative cost of the associated linear algebra. Here, we first describe the order of operation as described by Ligocki et al., then discuss constraints and modifications to the operation order that are made to accommodate them.

To illustrate these ideas, consider the 2D case. Let us write as a subscript $[\cdot\cdot]$ to indicate that the volume being integrated over is $[-h/2, h/2] \times [-h/2, h/2]$, and $[+\cdot]$ to indicate the $+x_0$ edge on which the integral runs $[-h/2, h/2]$ in the x_1 direction (Figure 2.5). We will write $(p_0 p_1)$ to represent a given moment. Thus, $(10)_{[\cdot-]}$ is the first x moment of the bottom edge of the cell.

In support of the 2D computation, we need the $P = 0$ and $P = 1$ moments over each edge (each of the four 1D bounding spaces). These quantities are determined by interpolation of the discrete level set data using stencils and methods described below (§2.5.4).

Once these 1D moments are known, one can proceed to evaluate the moments in 2D. In volume $[\cdot\cdot]$ we are interested in the $P = 0$ moment $(00)_{[\cdot\cdot]}$ which gives the volume fraction. We are also interested in the $P = 0$ and $P = 1$ moments over the EB, which together specify the centroid. The EB in volume $[\cdot\cdot]$ will be written $\{\cdot\cdot\}$. The first block of equations come from (2.58) with $P = 1$. In order, these are from $\mathbf{p} = (1, 0)$ with $d = 0$, then $d = 1$, followed by $\mathbf{p} = (0, 1)$ with $d = 0$, then $d = 1$:

(2.62)

$$\begin{aligned} 1(00)_{[\cdot\cdot]} - n_{0[\cdot\cdot]}(10)_{\{\cdot\cdot\}} &= (10)_{[+\cdot]} - (10)_{[-\cdot]} \\ -n_{1[\cdot\cdot]}(10)_{\{\cdot\cdot\}} &= (10)_{[\cdot+]} - (10)_{[\cdot-]} \\ -n_{0[\cdot\cdot]}(01)_{\{\cdot\cdot\}} &= (01)_{[+\cdot]} - (01)_{[-\cdot]} \\ 1(00)_{[\cdot\cdot]} - n_{1[\cdot\cdot]}(01)_{\{\cdot\cdot\}} &= (01)_{[\cdot+]} - (01)_{[\cdot-]}. \end{aligned}$$

With the unknowns on the left hand side, there are 4 equations to determine 3 variables. The next set of equations come from (2.58) with $P = 0$, $\mathbf{p} = (0, 0)$, with $d = 0$ followed by $d = 1$:

(2.63)

$$-n_{0[\cdot\cdot]}(00)_{\{\cdot\cdot\}} = (00)_{[+\cdot]} - (00)_{[-\cdot]} + n^{(10)}_{0[\cdot\cdot]}(10)_{\{\cdot\cdot\}} + n^{(01)}_{0[\cdot\cdot]}(01)_{\{\cdot\cdot\}}$$
$$-n_{1[\cdot\cdot]}(00)_{\{\cdot\cdot\}} = (00)_{[\cdot+]} - (00)_{[\cdot-]} + n^{(10)}_{1[\cdot\cdot]}(10)_{\{\cdot\cdot\}} + n^{(01)}_{1[\cdot\cdot]}(01)_{\{\cdot\cdot\}};$$

two equations in one unknown. This system requires the normal and its gradient, which may be constructed from a degree-2 Taylor series expansion of the level set.

From these computations, volume fraction, centroids and apertures are (e.g.)

$$\kappa = \frac{(00)_{[\cdot\cdot]}}{h^2} \tag{2.64}$$

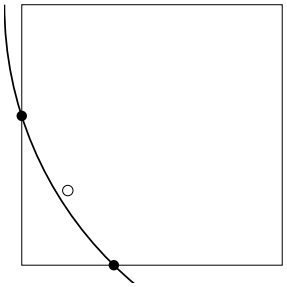
$$\mathbf{x}^{\text{cent}}_{[-\cdot]} = \frac{1}{(00)_{[-\cdot]}} \begin{pmatrix} (10)_{[-\cdot]} \\ (01)_{[-\cdot]} \end{pmatrix} \tag{2.65}$$

$$\mathbf{x}^{\text{cent}}_{\{\cdot\cdot\}} = \frac{1}{(00)_{\{\cdot\cdot\}}} \begin{pmatrix} (10)_{\{\cdot\cdot\}} \\ (01)_{\{\cdot\cdot\}} \end{pmatrix} \tag{2.66}$$

$$\alpha_{[-\cdot]} = \frac{(00)_{[-\cdot]}}{h} \tag{2.67}$$

(cf. (2.27,2.33a,2.28)). While an EB area is calculated by this method, finite volume discretizations use the projected area and the normal that come from the requirement that $\nabla \cdot (\mathbf{e}_i) = 0$ [29].

FIG. 2.6. *Recentering to improve accuracy. When the 1D edges of a given volume are evaluated, the intersections of the edges with $\psi = 0$ are discovered (filled circles). In the evaluation of higher-dimensional volumes, here a 2D face, the mean of the intersection points of those edges associated with this face gives a centering point (open circle) which approximates the centroid of the EB.*



Ligocki (personal communication) noted that the quality of the least squares solutions can be dramatically improved by recentering the calculation from the center of a given $[-h/2, h/2]^D$ volume to a point close to the centroid of the EB. Specifically, we recenter the linear equation systems and the constraint equations prior to solution of the over-determined data fitting equations by Householder reduction, then recenter the computed result to the center of the given volume. The estimated centroid is the average of the intersections of $\psi = 0$ with the 1D edges of the volume being evaluated (Figure 2.6).

2.5.3. Incorporation of constraints. The moments appearing in this expansion are subject to certain inequality constraints. If $\bar{\mathbf{p}}$ is even, then the corresponding volume integral is nonnegative and, if not on the EB, can be bounded from above:

$$0 \le \int_V \mathbf{x}^{\bar{\mathbf{p}}} dV \le \frac{h^{P+D}\mathbf{p}!}{2^P(\mathbf{p}+\mathbf{1})!}. \tag{2.68}$$

If $\mathbf{p}$ differs from an even multiindex $\bar{\mathbf{p}}$ by addition of a unit basis vector $\mathbf{e}_j$, then

$$\min_V(x_j)\int_V \mathbf{x}^{\bar{\mathbf{p}}}dV \le \int_V \mathbf{x}^{\bar{\mathbf{p}}+\mathbf{e}_j}dV \le \max_V(x_j)\int_V \mathbf{x}^{\bar{\mathbf{p}}}dV \tag{2.69}$$

by the mean value theorem.

In the second-order 2D example described above, simple positivity constraints are

$$(00)_{[\cdot\cdot]} \ge 0 \tag{2.70a}$$

$$(00)_{\{\cdot\cdot\}} \ge 0, \tag{2.70b}$$

and there is a physical constraint

$$(00)_{[\cdot\cdot]} \le h^2; \tag{2.70c}$$

volume fraction is positive but less than or equal to one, and the EB area is positive. Constraints of the second type are

$$-\frac{h}{2}(00)_{\{\cdot\cdot\}} \le (10)_{\{\cdot\cdot\}} \le +\frac{h}{2}(00)_{\{\cdot\cdot\}} \tag{2.71a}$$

$$-\frac{h}{2}(00)_{\{\cdot\cdot\}} \le (01)_{\{\cdot\cdot\}} \le +\frac{h}{2}(00)_{\{\cdot\cdot\}}. \tag{2.71b}$$

Constraints of type (2.68) can be implemented with any organization of the divergence theorem hierarchy. However, to incorporate those derived from the mean value theorem (2.69) while minimizing the overall least squares problem, it is necessary to solve for all necessary moments of a given volume simultaneously. This can be seen by noting that constraints (2.71) combine EB area values $(00)_{\{\cdot\cdot\}}$ and EB moment values $(10)_{\{\cdot\cdot\}}$ and $(01)_{\{\cdot\cdot\}}$ which are determined in different blocks (2.63) and (2.62), respectively, of the Ligocki et al. algorithm.

Incorporation of constraints in that setting means that the first linear system is solved without constraints, then constraints may be incorporated in subsequent solves. This would be analogous to weighing system (2.62) in preference to (2.63). This relative priority cannot be justified. To correct this weighting problem we solve simultaneously for all moments of a given subspace: (i) 1D moments as above, (ii) solve system (2.62,2.63) together. We explicitly weigh each equation by h^{-P} so that, unconstrained, they carry similar weights as in the Ligocki et al. method.

All linear systems are solved with Householder **QR** reduction. The constrained least squares problem is equivalent to the constrained positive definite quadratic programming problem solved by Goldfarb and Idnani [12, 13]: minimize $(\mathbf{Ax}-\mathbf{b})^T(\mathbf{Ax}-\mathbf{b})$ with respect to $\mathbf{x}$ subject to linear inequality constraints. Their method begins with the Cholesky $\mathbf{LL}^T$ decomposition of the Hessian $\mathbf{A}^T\mathbf{A}$, and with $\mathbf{Q}$ unitary the setup phase of their method is trivial: $\mathbf{L} = \mathbf{R}^T$, the transpose of $\mathbf{R}$ from the Householder decomposition. The quadratic form $(\mathbf{Ax}-\mathbf{b})^T(\mathbf{Ax}-\mathbf{b})$ never need be explicitly constructed.

2.5.4. Stencils. Here algorithms are described that determine the moments on 1D subspaces, and derivatives of the normal vector, from cell– and time–centered level set discretizations.

Nominally, we assume that the EB $\psi = 0$ will intersect each 1D edge at most once. If this is true, then interpolated values of ψ at the corners of a cell determine

FIG. 2.7. *Stencils for the construction of 1D moments for the case where all dimensions are spatial, illustrated in 2D for the case of order $S = 3$ or 4. To achieve a final order S, stencils of half width $K = \lceil S/2 \rceil$ are constructed. a) To compute the moments on the left edge of the center cell (bold line), a stencil consists of $2K$ points transverse to the edge, and $2K+1$ points in the direction of the edge (open circles). The first step is to interpolate to the line in the transverse direction. The points being interpolated lie on the dashed lines, and the resulting interpolants are given by crosses. The calculation of the derivatives of* $\mathbf{n}$ *for the center cell is based on a least squares fit of all $(2K+1)^D$ cells (the squares) to determine Taylor coefficients in a centered expansion of ψ. b) The top $2K$ interpolants are interpolated to deduce the value at the top end of the bold line segment (filled circle). The polynomial given by this filled circle and the bottom $2K$ crosses is identical to the polynomial given by all crosses alone, so the top cross may be dropped when the filled circle is added to the list of support points. Similarly, the bottom $2K$ points are used to interpolate the value at the bottom of the line segment. The result is that the corner values of the cell are computed from a symmetric $(2K)^D$ set of points, and for all cells that share a given corner the stencil is identical (e.g., the value at the corner indicated by the filled circle is determined by the set of points in the bold square, regardless of the edge under consideration). c) The resulting $2K+1$ interpolation points (equivalently the $2K+1$ crosses of part (a)) define an interpolation polynomial whose roots are the intersection of $\psi = 0$ with the given edge.*

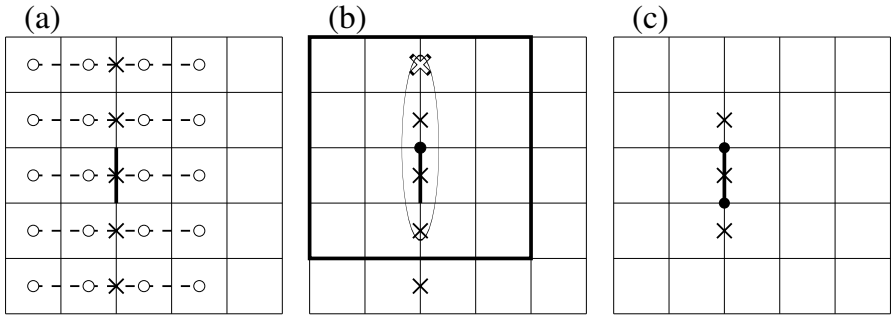
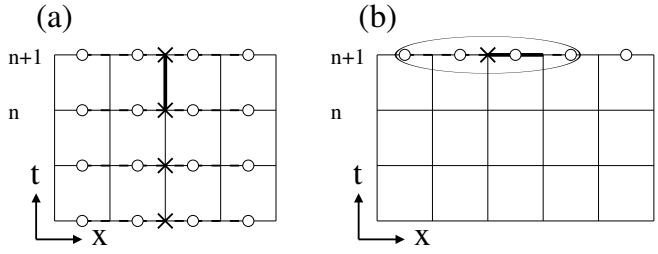


FIG. 2.8. *Stencils for the case where one dimension is temporal, illustrated in 1D+1D for the case of order $S = 4$. Let $K = \lceil S/2 \rceil$. Because data is centered at time levels, stencils for time and space edges are different. a) To find moments on a temporal edge (bold line segment) S time levels are interpolated (crosses) each from $2K$ spatial interpolations (circles on dashed lines). b) For a spatial edge, data on a single time level is treated by the stencil described in Figure 2.7. To evaluate derivatives of ψ at the center of a space-time volume, the stencil uses $2K+1$ points in each spatial direction and $S+1$ time levels.*



which edges are intersected by the EB, which are covered (by the wall), and which are regular. It is important to the robustness of the method that these corner values be accurate, and that each edge's notion of the corner be identical: the corner values must be cell– and edge–invariant. In order that quantities like the aperture α be invariant, it is also important that the intersection point of $\psi = 0$ with a given edge be cell-invariant. These symmetry considerations impact the interpolation algorithms by rounding up the stencil width in some cases.

To compute the moments on 1D edges, one finds the intersection of the edge with

$\psi = 0$ (say a point $\boldsymbol{\zeta}$), then constructs the moments explicitly. In a frame where the cell center is at the origin, one has

$$(p_0 p_1)_{[-,\cdot]} = \left(-\frac{h}{2}\right)^{p_0} \times \begin{cases} \int_{-h/2}^{\zeta} y^{p_1} dy & n_y = -1 \\ \int_{\zeta}^{h/2} y^{p_1} dy & n_y = +1, \end{cases} \tag{2.72}$$

for example.

The intersection point $\boldsymbol{\zeta}$ is determined by constructing an interpolating polynomial using data interpolated to the line coincident with the edge. We seek its roots with bisection until Smale's criterion [34] indicates that Newton-Raphson will converge quadratically. Roots are then refined with Newton-Raphson.

For the general case of arbitrary dimension D and arbitrary order S, $\mathcal{O}(h^S)$ accuracy on the 1-dimensional subspaces requires an interpolation polynomial with S support points. The symmetry invariance requirement of the method modifies this stencil. If $K = \lceil S/2 \rceil$, then $2K$ support points are required in the transverse direction and $2K + 1$ in the normal direction (Figure 2.7).

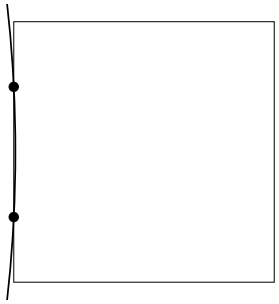
The support requirements in the case of space-time interpolation are simpler since data exists on the time edges so interpolation to integer time levels is not required (Figure 2.8).

To achieve order S accuracy, $S - 1$ order derivatives of the normal vector are required, which are based on S order derivatives of the discrete level set using

$$\mathbf{n}^{(\mathbf{p})} = \frac{d^{\mathbf{p}}}{d\mathbf{x}^{\mathbf{p}}} \frac{\nabla\psi}{\sqrt{(\nabla\psi)\cdot(\nabla\psi)}}. \tag{2.73}$$

These derivatives are based on a Taylor series centered at the center of the relevant subspace, fit to data with stencil width $S + 1$. Where possible the stencil is made symmetric by rounding up to width $2K + 1$.

FIG. 2.9. *Irregular cells whose interpolated corner values have uniform sign. In this example, the cell is a square of length h and $\psi = 0$ is a circle centered $5\frac{1}{2}h$ units to the left of the cell center. The radius is chosen so the circle intersects the left cell boundary at $\pm h/4$. The area to be measured is $\approx 2.08 \times 10^{-3} h^2$.*



2.5.5. Underresolved and nonconforming geometry. Underresolved geometries may fail under the standard algorithm. The geometry in Figure 2.9 will fail because the interpolated value of ψ at the corners of the square cell are all positive. The algorithm therefore misses the fact that the EB crosses the left edge twice. One way to detect these problems is to estimate the minimum and maximum values of ψ on the cell. If these have different signs, then the cell is irregular even when the corner values have uniform sign, and even if ψ is not a distance function.

TABLE 2.1

Convergence of EB area for sections of a hypersphere for order $S = 2$. In 2D the area of a unit circle is computed on one quadrant. In 3D, the area of a unit sphere in one octant. In 2+1D, a section of the unit sphere from the midplane to $x_2 = 1/16$. In 3+1D, the area of a unit hypersphere from the midplane to $x_3 = 1/16$. Calculations used cell-centered values of the signed distance function to derive all quantities.

2D		
$1/h$	error	rate
16	-8.143e-4	
32	-2.226e-4	1.87
64	-5.409e-5	2.04
128	-1.378e-5	1.97

3D		
$1/h$	error	rate
16	-3.108e-3	
32	-7.873e-4	1.98
64	-1.958e-4	2.01
128	-4.898e-5	2.00

(2+1)D		
$1/h$	error	rate
16	-6.705e-5	
32	-1.796e-5	1.90
64	-4.416e-6	2.02
128	-1.115e-6	1.98

(3+1)D		
$1/h$	error	rate
16	-2.107e-4	
32	-5.330e-5	1.98
64	-1.327e-5	2.01
128	-3.315e-6	2.00

TABLE 2.2

Relative area error (i.e., $(A^h - A)/A$ where A is exact and A^h is computed) using one level of bisection to resolve Fig. 2.9. Without subdivision, the relative error is 1.

order S	relative error
2	-0.595
3	-0.028,1
4	-0.030,4
5	0.008,32
6	-0.000,256

An algorithm to estimate the range of values the differentiable function ψ takes on the cell is given by Rivlin [31]. The basic idea is to sample the domain $\Omega_{\mathbf{i}}$ by overlaying it with a grid of length δ. If $\psi(\boldsymbol{\xi})$ is an extremum in cell $\mathbf{i}$, and $\mathbf{x_k}$ is a point on the δ-grid, then

$$\psi(\mathbf{x_k}) = \psi(\boldsymbol{\xi}) + \sum_{|\mathbf{r}|=2} \frac{(\mathbf{x_k} - \boldsymbol{\xi})^{\mathbf{r}}}{\mathbf{r}!} \psi^{\mathbf{r}}(\boldsymbol{\chi}) \tag{2.74a}$$

$$\max_{\mathbf{x}\in\Omega_{\mathbf{i}}} |\psi(\mathbf{x}) - \psi(\boldsymbol{\xi})| \le \Delta \equiv \frac{\delta^2}{4} \max_{\boldsymbol{\chi}} \sum_{|\mathbf{r}|=2} \frac{1}{\mathbf{r}!} |\psi^{(\mathbf{r})}(\boldsymbol{\chi})| \tag{2.74b}$$

for some $\boldsymbol{\chi} \in [\mathbf{x_k}, \boldsymbol{\xi}]$, and so

$$\min_{\mathbf{x}\in\Omega_i} \psi(\mathbf{x}) > \min_{\mathbf{x_k}} \psi(\mathbf{x_k}) - \Delta \tag{2.75a}$$

$$\max_{\mathbf{x}\in\Omega_i} \psi(\mathbf{x}) < \min_{\mathbf{x_k}} \psi(\mathbf{x_k}) + \Delta. \tag{2.75b}$$

We estimate Δ using the Taylor series, which we center at the center of cell $\Omega_{\mathbf{i}}$:

$$\psi(\mathbf{x}) = \sum_P \sum_{|\mathbf{p}|=P} \frac{\mathbf{x}^{\mathbf{p}}}{\mathbf{p}!}\psi^{(\mathbf{p})}(\mathbf{0}) \tag{2.76a}$$

$$\Delta = \frac{\delta^2}{4} \max_{\boldsymbol{\chi}} \sum_{|\mathbf{r}|=2} \frac{1}{\mathbf{r}!} \left| \sum_P \sum_{|\mathbf{p}|=P} \frac{\boldsymbol{\chi}^{\mathbf{p}-\mathbf{r}}}{(\mathbf{p}-\mathbf{r})!}\psi^{(\mathbf{p})}(\mathbf{0}) \right| \tag{2.76b}$$

$$\leq \frac{\delta^2}{4} \sum_{|\mathbf{r}|=2} \frac{1}{\mathbf{r}!} \sum_P \sum_{|\mathbf{p}|=P} \frac{(\frac{1}{2}\mathbf{h})^{\mathbf{p}-\mathbf{r}}}{(\mathbf{p}-\mathbf{r})!} \left|\psi^{(\mathbf{p})}(\mathbf{0})\right| \tag{2.76c}$$

where $\mathbf{h}$ is the vector cell edge lengths. In support of (2.73), derivatives of ψ through order S are known. So, for any order $S \geq 2$ sufficient information will be available to employ Rivlin's method. Given a desired tolerance Δ; (i) approximate the Taylor series by least squares, (ii) estimate δ from (2.76c), then compute the bounds by sampling the polynomial. If the product of bounds $\psi_{\min}\psi_{\max}$ is negative, then subdivision is applied. Otherwise, the cell is regular $\kappa = 1$ or covered $\kappa = 0$.

For the situation in Figure 2.9, a single bisection (in all directions) permits identification of the cell as an irregular one. The resulting volume calculations are summarized in Table 2.2.

3. Results. We demonstrate the method and show its convergence by computing the flow past a sphere in a bounded domain, Figure 3.1. In arbitrary units, the domain has length 4 and height 2. The top and bottom boundaries are stationary no-slip walls, the right boundary is outflow, and the left domain boundary is inflow with velocity having a Poiseuille profile with maximum velocity 1.5. Viscosity is 0.1. A sphere centered at $(1, 1)$ obstructs the flow. Its radius depends on time as $0.2 + 0.1\cos\omega t$, with $\omega = \pi/1.2$. The finest discretization of the domain is 1024×512, with $\Delta t = 1.5 \times 10^{-3}$ fixed. To determine rates of convergence we also use coarser grids: a 512×256 grid with $\delta t = 3.0 \times 10^{-3}$, etc., through the coarsest discretization of 64×32 with $\Delta t = 2.4 \times 10^{-2}$. The maximum CFL over the course of this simulation is 0.8.

Errors and rates of convergence are shown in Table 3.1 after 352 time steps on the finest grid through 22 time steps on the coarsest. In L_1 the velocity is second-order accurate, while in L_∞ it is first-order. The errors reported are Richardson estimates obtained by comparing computations with different resolution:

$$\|u\|_1^{h,2h} = \frac{1}{V}\int_\Omega |u^h - u^{2h}| dV = \frac{\sum_{\mathbf{i}} \kappa_{\mathbf{i}} |u^h - u^{2h}|_{\mathbf{i}}}{\sum_{\mathbf{i}} \kappa_{\mathbf{i}}} \tag{3.1a}$$

$$\|u\|_\infty^{h,2h} = \max_{\mathbf{x}\in\Omega} |u^h - u^{2h}| = \max_{\mathbf{i}} |u^h - u^{2h}|_{\mathbf{i}}. \tag{3.1b}$$

In these expressions, $\mathbf{i}$ is a cell index in the $2h$-grid, and

$$|u^h - u^{2h}|_{\mathbf{i}} = \left| \mathbf{u}_{\mathbf{i}}^{2h} - \frac{1}{2^D} \sum_{\mathbf{j}} \frac{\kappa_{\mathbf{j}}^h}{\kappa_{\mathbf{i}}^{2h}} \mathbf{u}_{\mathbf{j}}^h \right| \tag{3.2}$$

with the sum being over h-grid cells $\mathbf{j}$ that lie in the $2h$-grid cell. The convergence rate is given by

$$r = \frac{1}{\ln 2} \ln \frac{\|u\|^{2h,4h}}{\|u\|^{h,2h}}. \tag{3.3}$$

FIG. 3.1. *Flow past a shrinking sphere on 2:1 domain. Circles represent initial and final sphere surface. Curves are streamlines. Color corresponds to* $|\mathbf{u}|$ *from 0 (blue) to 1.7 (red). Note that the streamlines attach to the sphere because it is moving. Times are* 0.6 *and* 1.2*, respectively*

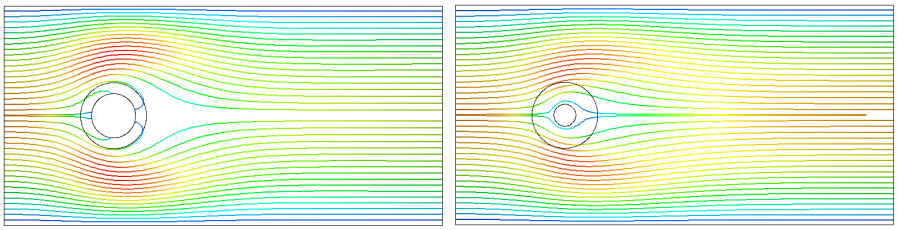


TABLE 3.1
Richardson error convergence study for flow past a shrinking sphere

N_x	$\|u_x \text{ error}\|_\infty$	rate	$\|u_x \text{ error}\|_1$	rate
64/128	5.28×10^{-1}		4.37×10^{-3}	
128/256	2.24×10^{-1}	1.24	9.24×10^{-4}	2.24
256/512	8.76×10^{-2}	1.35	2.22×10^{-4}	2.06
512/1024	4.64×10^{-2}	0.92	5.36×10^{-5}	2.05

N_x	$\|u_y \text{ error}\|_\infty$	rate	$\|u_y \text{ error}\|_1$	rate
64/128	4.01×10^{-1}		3.12×10^{-3}	
128/256	1.86×10^{-1}	1.11	7.07×10^{-4}	2.14
256/512	8.28×10^{-2}	1.16	1.68×10^{-4}	2.07
512/1024	3.97×10^{-2}	1.06	3.95×10^{-5}	2.09

The first-order convergence in L_∞ is expected because of the discretization error of the quadrature formula (2.31) for the hyperbolic part of the governing equations. As anticipated by Colella [9], the truncation error in irregularly shaped finite volumes is lower order than regularly shaped volumes. Thus, any fully conservative and consistent finite volume hyperbolic method based on a quadrature rule consisting of one point per bounding surface will be first-order in L_∞. This expectation applies also to approaches like cell merging.

Acknowledgments. This work builds on a flow solver for the incompressible Navier Stokes equations on a fixed domain, first written by D. T. Graves at the Lawrence Berkeley National Laboratory. We thank him for many helpful suggestions in the course of this project. T. J. Ligocki, also at LBNL, was a valuable resource for computational geometry.

REFERENCES

[1] 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), pp. 257–283.
[2] M. J. BERGER, C. HELZEL, AND R. J. LEVEQUE, *H-box methods for the approximation of one-dimensional conservation laws on irregular grids*, SIAM J. Numer. Anal., 41 (2003), pp. 893–918.
[3] I.-L. CHERN AND P. COLELLA, *A conservative front tracking method for hyperbolic conservation laws*, tech. report, Lawrence Livermore National Laboratory, 1987. UCRL-97200.

[4] P. H. Chiu, R. K. Lin, and T. W. H. Sheu, *A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier-Stokes equations in time-varying complex geometries*, J. Comput. Phys., 229 (2010), pp. 4476–4500.
[5] A. Chorin and J. Marsden, *A mathematical introduction to fluid mechanics*, Springer-Verlag, New York, third ed., 1993.
[6] A. J. Chorin, *Numerical solution of incompressible flow problems*, Stud. Numer. Anal., 2 (1968), pp. 64–71.
[7] ———, *Numerical solution of Navier-Stokes equations*, Math. Comput., 22 (1968), pp. 745–762.
[8] ———, *On convergence of discrete approximations to Navier-Stokes equations*, Math. Comput., 23 (1969), pp. 341–353.
[9] P. Colella, *Volume-of-fluid methods for partial differential equations*, in Godunov Methods: Theory and Applications, E. F. Toro, ed., New York, 2001, Kluwer, pp. 161–176.
[10] P. Colella, D. T. Graves, B. J. Keen, and D. Modiano, *A Cartesian grid embedded boundary method for hyperbolic conservation laws*, J. Comput. Phys., 211 (2006), pp. 347–366.
[11] P. Colella and D. Trebotich, *Numerical simulations of incompressible viscous flow in deforming domains*, Proc. Natl. Acad. Sci., 96 (1999), pp. 5378–5381.
[12] D. Goldfarb and A. Idnani, *Dual and primal-dual methods for solving strictly convex quadratic programs*, in Numerical Analysis: Proceedings of the third IIMAS workshop held at Cocoyoc, Mexico, Jan. 19–23, 1981, J.P. Hennart, ed., New York, 1982, Springer-Verlag, pp. 226–239.
[13] ———, *A numerically stable dual method for solving strictly convex quadratic programs*, Math. Programming, 27 (1983), pp. 1–33.
[14] F. H. Harlow and J. E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, Phys. Fluids, 8 (1965), pp. 2182–2189.
[15] T. Ikeno and T. Kajishima, *Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations*, J. Comput. Phys., 226 (2007), pp. 1485–1508.
[16] M. F. Lai, *A Projection Method for Reacting Flow in the Zero Mach Number Limit*, PhD thesis, University of California, Berkeley, 1994.
[17] P. D. Lax, *Weak solutions of nonlinear hyperbolic equations and their numerical computation*, Comm. Pure Appl. Math., 7 (1954), pp. 159–193.
[18] R. J. LeVeque and Z. Li, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources*, SIAM J. Numer. Anal., 31 (1994), pp. 1019–1044.
[19] C.-C. Liau, Y.-W. Chang, C.-A. Lin, and J. M. McDonough, *Simulating flows with moving rigid boundary using immersed-boundary method*, Comput. Fluids, 39 (2010), pp. 152–167.
[20] T. J. Ligocki, P. O. Schwartz, J. Percelay, and P. Colella, *Embedded boundary grid generation using the divergence theorem, implicit functions, and constructive solid geometry*, J. Phys: Conference Series, 125 (2008), pp. 1–5.
[21] S. Marella, S. Krishnan, H. Liu, and H. S. Udaykumar, *Sharp interface Cartesian grid method I: An easily implemented technique for 3D moving boundary computations*, J. Comput. Phys., 210 (2005), pp. 1–31.
[22] D. F. Martin and K. L. Cartwright, *Solving Poisson's equation using adaptive mesh refinement*, tech. report, Electronics Research Laboratory, University of California, Berkeley, 1996. UCB/ERL M96/66.
[23] D. F. Martin, P. Colella, and D. Graves, *A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions*, J. Comput. Phys., 227 (2008), pp. 1863–1886.
[24] P. McCorquodale, P. Colella, and H. Johansen, *A Cartesian grid embedded boundary method for the heat equation on irregular domains*, J. Comput. Phys., 173 (2001), pp. 620–635.
[25] G. H. Miller and P. Colella, *A conservative three-dimensional Eulerian method for coupled solid–fluid shock capturing*, J. Comput. Phys., 183 (2002), pp. 26–82.
[26] M. L. Minion and D. L. Brown, *Performance of under-resolved two-dimensional incompressible flow simulations, II*, J. Comput. Phys., 138 (1997), pp. 734–765.
[27] W. F. Noh, *CEL: a time-dependent, two-space-dimensional, coupled Eulerian-Lagrangian code*, in Methods in Computational Physics, B. Alder, S. Fernbach, and M. Rotenberg, eds., vol. 3, New York, 1964, Academic Press, pp. 117–179.
[28] H. Pan, L. S. Pan, D. Xu, T. Y. Ng, and G. R. Liu, *A projection method for solving incompressible viscous flows on domains with moving boundaries*, Int. J. Numer. Meth. Fluids, 45 (2004), pp. 53–78.
[29] R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome, *An adaptive Cartesian grid method for unsteady compressible flow in irregular regions*, J.

Comput. Phys., 120 (1995), pp. 278–304.
[30] C. S. Peskin, *Numerical analysis of blood flow in the heart*, J. Comput. Phys., 25 (1977), pp. 220–252.
[31] T. J. Rivlin, *Bounds on a polynomial*, J. Res. Nat. Bur. Stand., 74B (1970), pp. 47–54.
[32] J. Saltzman, *An unsplit 3D upwind method for hyperbolic conservation laws*, J. Comput. Phys., 115 (1994), pp. 153–168.
[33] C. Shen, D. Trebotich, S. Molins, D. T. Graves, B. Van Straalen, D. T. Graves, T. Ligocki, and C. I. Steefel, *High performance computations of subsurface reactive transport processes at the pore scale*, J. Phys: Conference Series, (2011). to appear.
[34] S. Smale, *Newton's method estimates from data at one point*, in The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics, R.E. Ewing, K.I. Gross, and C.F. Martin, eds., New York, 1986, Springer-Verlag, pp. 185–196.
[35] S. K. Stein and A. Barcellos, *Calculus and Analytic Geometry*, McGraw-Hill, New York, fifth ed., 1992.
[36] Z. Tan, K. M. Lim, and B. C. Khoo, *A level set-based immersed interface method for solving incompressible viscous flows with the prescribed velocity at the boundary*, Int. J. Numer. Meth. Fluids, 62 (2010), pp. 267–290.
[37] D. Trebotich and P. Colella, *A projection method for incompressible viscous flow on moving quadrilateral grids*, J. Comput. Phys., 166 (2001), pp. 191–217.
[38] E. H. Twizell, A. B. Gumel, and M. A. Arigu, *Second-order L_0-stable methods for the heat equation with time-dependent boundary conditions*, Adv. Comput. Math., 6 (1996), pp. 333–352.
[39] H. S. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna, *A sharp interface Cartesian grid method for simulating flows with complex boundaries*, J. Comput. Phys., 174 (2001), pp. 345–380.
[40] B. van Leer, *Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method*, J. Comput. Phys., 32 (1979), pp. 101–136.