COMPUTATIONAL MATHEMATICS

2016 © The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. Distributed under a Creative Commons Attribution NonCommercial License 4.0 (CC BY-NC).
10.1126/sciadv.1501869

Interfacial gauge methods for incompressible fluid dynamics

Robert Saye

Designing numerical methods for incompressible fluid flow involving moving interfaces, for example, in the computational modeling of bubble dynamics, swimming organisms, or surface waves, presents challenges due to the coupling of interfacial forces with incompressibility constraints. A class of methods, denoted interfacial gauge methods, is introduced for computing solutions to the corresponding incompressible Navier-Stokes equations. These methods use a type of "gauge freedom" to reduce the numerical coupling between fluid velocity, pressure, and interface position, allowing high-order accurate numerical methods to be developed more easily. Making use of an implicit mesh discontinuous Galerkin framework, developed in tandem with this work, high-order results are demonstrated, including surface tension dynamics in which fluid velocity, pressure, and interface geometry are computed with fourth-order spatial accuracy in the maximum norm. Applications are demonstrated with two-phase fluid flow displaying fine-scaled capillary wave dynamics, rigid body fluid-structure interaction, and a fluid-jet free surface flow problem exhibiting vortex shedding induced by a type of Plateau-Rayleigh instability. The developed methods can be generalized to other types of interfacial flow and facilitate precise computation of complex fluid interface phenomena.

INTRODUCTION

A myriad of fluid dynamics problems involve interface motion playing a distinct role in the global dynamics. Examples include bubble aeration, submersed vessel locomotion, peristaltic flow, ink jet dynamics, and crashing waves. Many of these problems can be effectively modeled with the incompressible Navier-Stokes equations, in which the domain and embedded interfaces change in time, with boundary conditions and interface jump conditions determined by forces, such as surface tension. Here, we develop a class of techniques, called interfacial gauge methods, for the numerical solution of these equations, with particular attention given to two-phase fluid flow driven by surface tension, rigid body fluid-structure interaction, and free surface flow.

Designing accurate numerical methods for interface motion coupled to incompressible fluid flow is challenging. For example, small-scale features in interface geometry, such as changes in curvature, can affect dynamics far away from the interface; at the same time, interface motion can be strikingly affected by small boundary layers in the fluid nearby the interface. A multitude of numerical methods have been developed to compute fluid interface dynamics, which, although too extensive to survey here, are often built on one or a hybrid of key methodologies, including (i) Peskin's immersed boundary method (*1*, *2*), a mixed Eulerian-Lagrangian method in which interfacial forces are represented as fluid body forces via smoothed approximations of Dirac delta functions; (ii) LeVeque and Li's immersed interface method (*3*, *4*), a sharp interface finite difference approach in which finite difference stencils are adapted to include jump conditions; (iii) level set methods (*5*), in which interfacial forces are regularized with smoothed delta functions (*6*–*8*); (iv) the ghost fluid method (*9*), a finite difference method that introduces extra grid-collocated degrees of freedom determined by extrapolation and jump conditions; and (v) the volume of fluid method (*10*), a Eulerian method in which the volume fraction of each fluid phase is tracked in each grid cell. These techniques are most commonly implemented in combination with a fixed computational grid; other numerical paradigms include (i) moving mesh methods, in which computational meshes follow the interface and adapt to changing domain shapes; (ii) cut-cell and embedded boundary finite volume and finite element methods; (iii) finite element methods that enrich the solution space with basis functions accounting for discontinuities by locally adapting to the interface shape, such as the XFEM (extended finite element method); and (iv) hybrid techniques, such as ALE (arbitrary Lagrangian-Eulerian) methods, which use a combination of moving mesh and embedded boundary techniques.

Independent of the interface treatment, essentially all of these techniques use projection methods to solve the incompressible Navier-Stokes equations. Pioneered by Chorin (*11*), projection methods can be thought of as a fractional stepping in which, at each time step, the Navier-Stokes momentum equations are used to compute an intermediate velocity field, which does not necessarily satisfy the incompressibility constraint. This intermediate velocity is then projected onto the space of divergence-free vector fields, thereby determining the fluid velocity $\mathbf{u}$ at the end of the time step. Without dismissing the wide impact and success of projection methods, in applying them to interfacial fluid dynamics, this fractional stepping approach generally creates a nonphysical coupling in the evolution of fluid velocity, pressure, and interface location. This coupling can limit the order of accuracy of numerical methods, in both space and time, especially when different fluid phases have different densities or viscosities. This issue of designing high-order accurate projection methods arises even without interfaces and has received extensive attention in the literature [for example, see previous studies (*12*–*14*)].

An alternative formulation of the incompressible Navier-Stokes equations, first introduced by Oseledets (*15*) [see also Roberts (*16*)], leads to a different class of methods. Instead of solving directly for the fluid velocity $\mathbf{u}$, these methods recast the equations to solve for an auxiliary vector field, denoted $\mathbf{m}$, whose divergence-free component gives $\mathbf{u}$. In fact, $\mathbf{m}$ is related to $\mathbf{u}$ via a gauge transformation, $\mathbf{u} = \mathbf{m} - \nabla\varphi$ for some scalar field φ, as predicted by the Helmholtz-Hodge

Mathematics Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA. Email: rsaye@lbl.gov

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

decomposition theorem. Different formulations and names for these methods have arisen, including velicity (*17*, *18*), magnetization (*19*–*22*), impulse (*20*, *23*–*26*), impetus (*27*), gauge (*28*–*30*), continuous projection (*31*), and auxiliary (*32*–*34*) methods, and there is even a connection to imaging science via dual momentum variables (*35*). These methods are also closely related to particular instances of projection methods, including the second-order method of Kim and Moin (*36*), as well as higher-order projection methods including consistent splitting schemes (*13*) and slip correction schemes (*14*). In essence, the various instances of these methods differ in the particular gauge transformation being utilized (that is, the equations determining **m** or equivalently the choice of φ and the boundary conditions coupling them), a freedom that has been explored in the context of vorticity generation at no-slip walls (*37*–*39*), in improving stability (*40*, *41*), and also in designing high-order accurate solvers for low Mach number hydrodynamics (*42*). With the exception of the works by Cortez (*23*, *24*) and Recchioni and Russo (*18*), wherein Lagrangian particle impulse methods for force-bearing filaments embedded in a two-dimensional (2D) fluid are considered, the aforementioned works have only considered single-phase fluid flow. A central theme in the present work is to capitalize on the "gauge freedom" in choosing φ, as well as the evolution equations for **m**, to design accurate and stable numerical methods for a variety of incompressible fluid flow problems with evolving interfaces. Hence, the terminology "gauge method" is adopted.

Here, a variety of gauge methods are developed for multiphase fluid flow problems. Compared to the archetypical projection methods, these "interfacial gauge methods" have a weaker coupling between fluid velocity, pressure, and interface dynamics and allow stable and high-order accurate schemes to be developed more readily. Unless it is otherwise clear from context, throughout this work, "high order" is defined to mean a scheme that computes fluid velocity, pressure, and interface position with at least second-order accuracy in both space and time, measured uniformly throughout the domain in the maximum norm (and thus also immediately adjacent to the interface). Additionally, the notion of stability refers to that of time stepping: one scheme is more stable than another if the time step Δt can be taken larger, provided the numerical solution remains bounded.

The essential concept underlying interfacial gauge methods is to appropriately choose evolution equations for **m** such that the associated projection operator is well behaved in a numerical setting. This amounts to choosing appropriate jump conditions on the evolving interface for **m** and φ. To this end, we first review the formulation of a gauge method for single-phase fluid flow in a static domain, after which interfacial gauge methods for two-phase flow, embedded rigid body motion, and free surface flow are then introduced. A numerical framework for computing these flows, which has been developed in tandem with interfacial gauge methods, is also briefly discussed—the framework is based on an implicit mesh discontinuous Galerkin (dG) method and allows for high-order accurate solutions to be computed.

Single-phase gauge methods

The essential idea of a gauge method for single-phase incompressible fluid flow is to replace the incompressible Navier-Stokes equations,

$$\left.\begin{aligned}\rho(\mathbf{u}_t+\mathbf{u}\cdot\nabla\mathbf{u})&=-\nabla p+\mu\Delta\mathbf{u}\\ \nabla\cdot\mathbf{u}&=0\end{aligned}\right\}\ \text{in}\ \Omega, \tag{1}$$

which determine the velocity $\mathbf{u}:\Omega\to\mathbb{R}^d$ and pressure $p:\Omega\to\mathbb{R}$ of a fluid of constant density ρ and constant viscosity μ in a domain $\Omega\subseteq\mathbb{R}^d$, by a new set of equations determining the evolution of an auxiliary vector field $\mathbf{m}:\Omega\to\mathbb{R}^d$ and scalar field $\varphi:\Omega\to\mathbb{R}$, where

$$\left.\begin{aligned}\rho(\mathbf{m}_t+\mathbf{u}\cdot\nabla\boldsymbol{u})&=\mu\Delta\mathbf{m}\\ \mathbf{u}&=\mathbf{m}-\nabla\varphi\\ \Delta\varphi&=\nabla\cdot\mathbf{m}\end{aligned}\right\}\ \text{in}\ \Omega, \tag{2}$$

subject to appropriate boundary conditions on ∂Ω, to be discussed shortly. Thus, instead of directly solving for the evolution of the fluid velocity **u**, one instead solves for **m**. At every instant in time, **u** is recovered from **m** via a projection of **m**, defined by the last two equations in Eq. 2: $\mathbf{u}=\mathbf{m}-\nabla\varphi$, where φ solves the Poisson problem $\Delta\varphi=\nabla\cdot\mathbf{m}$. Subject to suitable boundary conditions on φ, this defines a projection operator $\mathbf{u}=\mathbb{P}(\mathbf{m})$ whose output is the divergence-free component of its input. Simple algebraic manipulation reveals that if **m** solves Eq. 2, then $\mathbf{u}=\mathbb{P}(\mathbf{m})$ solves the equations

$$\left.\begin{aligned}\rho(\mathbf{u}_t+\mathbf{u}\cdot\nabla\mathbf{u})&=-\nabla(\rho\varphi_t-\mu\Delta\varphi)+\mu\Delta\mathbf{u}\\ \nabla\cdot\mathbf{u}&=0\end{aligned}\right\}\ \text{in}\ \Omega.$$

Therefore, under the assumption that solutions of the Navier-Stokes equations are unique, if **m** solves Eq. 2, then $\mathbf{u}=\mathbb{P}(\mathbf{m})$ solves Eq. 1 with pressure identified (up to an arbitrary constant) as $p=\rho\varphi_t-\mu\Delta\varphi$. A key point in this construction—and gauge methods in general—is that **m** is freed from having to satisfy a global incompressibility constraint in its evolution equation (Eq. 2); this holds several advantages for designing accurate and flexible time-stepping methods, especially for interfacial fluid flow, as we see later.

Although gauge methods alleviate some of the difficulties in solving the Navier-Stokes equations arising due to the incompressibility constraint, they are somewhat complicated by having to specify suitable boundary conditions on both **m** and φ in such a way that $\mathbf{u}=\mathbb{P}(\mathbf{m})$ satisfies the correct, physically prescribed boundary conditions on ∂Ω. This is intimately connected with the precise specification of the projection operator $\mathbb{P}$, determined by the choice of boundary conditions in the Poisson problem for φ. A guiding principle in the design of gauge methods, established from experience in this work, is that the projection operator should ideally be idempotent and, on the discrete level, yield an output whose normal component satisfies the correct boundary condition, independent of its input satisfying any sort of compatibility condition. This can be achieved by controlling the Neumann boundary condition on φ; in the continuum, we can simply choose these to be homogeneous. The system in Eq. 2 is then supplemented by the boundary conditions

$$\left.\begin{aligned}\mathbf{m}\cdot\mathbf{n}&=\mathbf{u}_\partial\cdot\mathbf{n}\\ \mathbf{m}\cdot\boldsymbol{\tau}&=\mathbf{u}_\partial\cdot\boldsymbol{\tau}+\nabla\varphi\cdot\boldsymbol{\tau}\\ \partial_n\varphi&=0\end{aligned}\right\}\ \text{on}\ \partial\Omega. \tag{3}$$

Here, **n** is the unit outward-pointing normal to the domain boundary, **τ** is any vector tangent to ∂Ω, $\partial_n:=\mathbf{n}\cdot\nabla$, and $\mathbf{u}_\partial$ is the given boundary conditions on the fluid velocity; no-slip walls correspond to $\mathbf{u}_\partial=0$. Thus, the normal component of **m** at the boundary is copied directly from $\mathbf{u}_\partial\cdot\mathbf{n}$, but the tangential component is coupled to ∇φ. At first glance, and in regard to the numerical solution of these equations, a predicament

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

arises: how can one solve for $\mathbf{m}$ in Eq. 2 when its boundary conditions in Eq. 3 depend on components of $\nabla\varphi$, which in turn depend on the solution to a Poisson problem whose input is $\mathbf{m}$? One straightforward approach for treating this coupling is to use a time-stepping scheme that predicts boundary data for $\mathbf{m}$ by extrapolating $\nabla\varphi$ from previous time steps, similar to a multistep method. Details of such a scheme will be given shortly.

In a manner, gauge methods transfer the incompressibility constraint in Eq. 1, to a constraint on the boundary conditions, that is, the coupling of $\mathbf{m}$ and $\nabla\varphi$ in Eq. 3. With regard to the development of high-order accurate time-stepping schemes, imposition of the divergence constraint is generally the greatest obstacle (*32*–*34*); as such, time-stepping schemes that evolve $\mathbf{m}$ are easier to design than schemes that directly evolve $\mathbf{u}$. This is especially true for multiphase fluid flow problems in which singularities arise on a moving interface, caused by, for example, abrupt changes in density or viscosity across the interface or interfacial forces, such as surface tension. In particular, the fractional stepping schemes underlying traditional Navier-Stokes projection methods, when applied to multiphase fluid flow, can cause appreciable degradation in accuracy and stability, due, in part, to the property that fractional stepping assumes that the evolution in time at a fixed location in space is smooth, which is untrue for points in space where the interface transitions during a time step.

The previous discussion motivated the design of a typical gauge method for a single fluid in a fixed, static domain. We now design several new gauge methods, denoted interfacial gauge methods, for a variety of multiphase fluid flow problems.

Interfacial gauge methods for surface tension dynamics

In two-phase fluid flow problems, for example, in oscillating soap bubbles, ink jet dynamics, or bubble rising flows, two immiscible fluids are separated by a moving interface. The resulting motion is strongly dependent on the coupling that occurs at the interface: for viscous fluids, the velocity field is continuous, whereas the stress tensor of the two fluids may exhibit a jump dependent on surface tension and interface curvature. Let Ω_1 and Ω_2 denote the (time-dependent) domains of the two fluids, each having density ρ_i and viscosity μ_i, and let $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ denote the evolving interface. Denote by Ω the whole domain, $\Omega = \Omega_1 \cup \Omega_2 \cup \Gamma$. Let $\mathbf{n}$ denote the unit normal of the interface Γ, pointing from Ω_1 into Ω_2, and let $[\cdot]$ denote the jump of a quantity across the interface, $[f] := f_1|_\Gamma - f_2|_\Gamma$. Then, the dynamics of incompressible two-phase fluid flow with surface tension are governed by the incompressible Navier-Stokes equations

$$\left.\begin{aligned}\rho_i(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}) &= -\nabla p + \mu_i\Delta\mathbf{u}\\ \nabla\cdot\mathbf{u} &= 0\end{aligned}\right\} \text{ in } \Omega_i,$$

subject to the interface jump conditions

$$\left.\begin{aligned}[\mathbf{u}] &= 0\\ [\boldsymbol{\sigma}]\mathbf{n} &= -\gamma\kappa\mathbf{n}\end{aligned}\right\} \text{ on } \Gamma, \tag{4}$$

where $\boldsymbol{\sigma} = -pI + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathsf{T}})$ is the stress tensor, $\kappa = \nabla_s\cdot\mathbf{n}$ is the mean curvature of Γ, and γ is the (constant) coefficient of surface tension.

In designing a gauge method for two-phase fluid flow, there is considerable freedom in choosing how the phase-dependent gauge variables $\mathbf{m}$ and φ are coupled across the interface. For example, one may impose jumps in $\nabla\mathbf{m}$ that directly capture interfacial forces such as surface tension. Alternatively, one may carefully construct jumps in φ that account for surface tension without ever explicitly calculating κ—one such method is described in the Supplementary Materials. Here, a wide variety of different interfacial gauge methods have been trialed; experience has shown that the most stable and accurate gauge methods for interfacial flow arise when $\mathbf{u}$ is defined as a projection of $\mathbf{m}$ that requires no special treatment of jump conditions, for example, by using homogeneous jump conditions in $\mathbf{m}$ and φ, if possible. With this in mind, we now outline two gauge methods for surface tension dynamics, treating two cases separately: first, when the two fluids have the same density and viscosity, later generalizing to the case when $[\rho] \neq 0$ or $[\mu] \neq 0$.

Two-phase flow with identical density and viscosity. For fluid flow involving interface motion, an important flexibility in gauge methods can be used to great advantage. This key feature is that the evolution equation for $\mathbf{m}$ can be modified, by adding a term of the form ∇q to the first equation in Eq. 2, without affecting the property that the projection of $\mathbf{m}$ correctly satisfies the Navier-Stokes momentum equations, independent of the choice of scalar function q. This modification only alters the formula for computing the fluid pressure p. We use this here to accurately and stably account for surface tension. Suppose that $\rho_1 = \rho_2$ and $\mu_1 = \mu_2$. Let $\mathbf{m}$ solve

$$\left.\begin{aligned}\rho(\mathbf{m}_t + \mathbf{u}\cdot\nabla\mathbf{u}) &= -\nabla q + \mu\Delta\mathbf{m}\\ \mathbf{u} &= \mathbf{m} - \nabla\varphi\\ \Delta\varphi &= \nabla\cdot\mathbf{m}\end{aligned}\right\} \text{ in } \Omega_i, \tag{5}$$

subject to the interface jump conditions

$$\left.\begin{aligned}[\varphi] = [\partial_n\varphi] &= 0\\ [\mathbf{m}] = [\partial_n\mathbf{m}] &= 0\\ [q] &= \gamma\kappa\end{aligned}\right\} \text{ on } \Gamma. \tag{6}$$

It is straightforward to show that $\mathbf{u}$ solves the Navier-Stokes momentum equations, with the pressure identified as $p = q + \rho\varphi_t - \mu\nabla\cdot\mathbf{m}$. Meanwhile, the jump conditions on φ in Eq. 6 imply that $[\nabla\varphi] = 0$ on Γ, and so $[\mathbf{u}] = [\mathbf{m}] - [\nabla\varphi] = 0$. To check the stress jump condition, we have that $[\nabla\mathbf{m}] = 0$, and so

$$\begin{aligned}[\sigma]\mathbf{n} &= [-pI + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathsf{T}})]\mathbf{n}\\ &= [-(q + \rho\varphi_t - \mu\nabla\cdot\mathbf{m})I + \mu(\nabla\mathbf{m} + \nabla\mathbf{m}^{\mathsf{T}}) - 2\mu D^2\varphi]\mathbf{n}\\ &= -\gamma\kappa\mathbf{n} - \rho[\varphi_t]\mathbf{n} - 2\mu[D^2\varphi]\mathbf{n},\end{aligned}$$

where $D^2\varphi$ is the Hessian of φ. To show that $[\varphi_t] = 0$, consider a passively advected particle $x = x(t)$ located on the interface; then, $\dot{x} = \mathbf{u}(x(t), t)$, and since $[\varphi] = 0$, we have that $\frac{d}{dt}[\varphi(x(t), t)] = 0$, which, upon application of the chain rule, gives $[\varphi_t + \mathbf{u}\cdot\nabla\varphi] = 0$. Since $[\mathbf{u}] = 0$ and $[\nabla\varphi] = 0$, it follows that $[\varphi_t] = 0$. Now consider the jump in $D^2\varphi$: deriving results regarding the Hessian of φ requires curvilinear coordinate expressions and is left to the Supplementary Materials. The conclusion is that, in this instance, $[D^2\varphi] = 0$; hence, the correct

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

stress jump conditions are recovered for this interfacial gauge method.

We have yet to uniquely define the function q satisfying the jump condition in Eq. 6. The role of q is to transfer the force of surface tension into the bulk domain, and in this work, it was found that a harmonic q led to an accurate and very stable method (as remarked later in the discussion on time-stepping methods). Thus, we choose

$$\Delta q = 0 \text{ in } \Omega_1 \cup \Omega_2, \quad [\partial_n q] = 0 \text{ on } \Gamma, \quad \partial_n q = 0 \text{ on } \partial\Omega. \quad (7)$$

Equations 5 to 7, plus the domain boundary conditions in Eq. 3, complete the specification of an interfacial gauge method for two-phase fluid flow driven by surface tension. The essential tools needed to implement the method are a numerical scheme capable of solving Poisson problems with jump conditions on internal interfaces, together with a time-stepping scheme that accounts for discrete differential operators that change in time (due to the moving interface). An example of such a framework that achieves high-order accuracy is described shortly; however, we next consider the case when the two fluids have different densities or viscosities.

Two-phase flow with jumps in density and viscosity. The interfacial gauge method derived above has homogeneous jump conditions for $\mathbf{m}$ and φ and is particularly natural in the case that $[\rho] = 0$ and $[\mu] = 0$. However, when the two fluids have different densities or viscosities, the jump conditions required by Eq. 4 involve different types of cancelation: the first condition requires that jumps in $\mathbf{m}$ cancel jumps in $\nabla\varphi$, whereas the second condition requires jumps in the effective stress tensor for $\mathbf{m}$ cancel those due to both the Hessian of $\mu\varphi$ and the contribution of $\rho\varphi_t$ to the pressure. It is impossible to specify jumps on $\mathbf{m}$ and $\nabla\varphi$ in such a way that all of the jumps in all of these quantities simultaneously and naturally cancel for an arbitrary curved interface. Essentially, this is because we can control both $[\varphi]$ and $[\nabla\varphi]$, or $[\alpha\varphi]$ and $[\alpha\nabla\varphi]$ (where α is a phase-dependent constant), but not a mixture of $[\varphi]$ and $[\alpha\nabla\varphi]$, because the latter is overconstrained when solving elliptic interface partial differential equations (PDEs). Fortunately, a compromise can be struck in which some components of the jumps in velocity and stress tensor can be enforced strongly, while others weakly, in such a way that the resulting numerical method is both consistent and numerically stable.

A design choice that significantly enhances the stability of an interfacial gauge method is to require the associated projection operator satisfy $[\mathbb{P}(\cdot)] \cdot \boldsymbol{n} = 0$. Stated differently, the normal component of the projected velocity field should be continuous across the interface, regardless of the input satisfying any sort of compatibility condition. Stated differently, the a natural requirement for inviscid flow but also aids in accuracy and stability for viscous flow. Combined with an observation regarding the contribution of φ_t to the pressure term (discussed further in the Supplementary Materials), these design choices lead to the following interfacial gauge method: let $\mathbf{m}$ solve

$$\left.\begin{array}{c} \rho_i(\mathbf{m}_t + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla q + \nabla\cdot(\mu_i(\nabla\mathbf{m} + \nabla\mathbf{m}^{\mathsf{T}})) \\ \mathbf{u} = \mathbf{m} - \rho_i^{-1}\nabla\varphi \\ \nabla\cdot\rho_i^{-1}\nabla\varphi = \nabla\cdot\mathbf{m} \end{array}\right\} \text{in } \Omega_i, \quad (8)$$

with interface jump conditions

$$\left.\begin{array}{c} [\varphi] = [\rho^{-1}\partial_n\varphi] = 0 \\ [\mathbf{m}]\cdot\mathbf{n} = 0, \quad [\mathbf{m}]\cdot\boldsymbol{\tau} = [\rho^{-1}\nabla\varphi]\cdot\boldsymbol{\tau} \\ [\mu(\nabla\mathbf{m} + \nabla\mathbf{m}^{\mathsf{T}})\cdot\mathbf{n}] = 2\left[\frac{\mu}{\rho}D^2\varphi\right]\mathbf{n} - 2\left[\frac{\mu}{\rho}\Delta\varphi\right]\mathbf{n} \\ [q] = \gamma\kappa - [\varphi_t] \end{array}\right\} \text{on } \Gamma. \quad (9)$$

Several remarks are in order. First, note that unlike Eq. 5, $\mathbf{m}$ now solves a PDE involving the viscous stress tensor divergence operator $\nabla\cdot(\nabla\mathbf{m}+\nabla\mathbf{m}^{\mathsf{T}})$. This is necessary because the stress tensor condition in Eq. 4 involves the jump of the viscous stress of $\mathbf{u}$, which cannot be simplified as before because μ_i is not assumed to be continuous across the interface. Second, the projection operator $\mathbf{u} = \mathbb{P}(\mathbf{m})$ now involves a (well-posed) variable coefficient Poisson problem such that $[\varphi] = 0$ with $[\rho^{-1}\partial_n\varphi] = 0$. [Note that inverse density–weighted Poisson problems also appear in traditional projection methods and gauge methods for variable density fluid flows; for example, see the works of Kadioglu *et al.* (*33*) and Almgren *et al.* (*43*).] Third, although a term of the form $\nabla\cdot(\nabla\mathbf{m}^{\mathsf{T}})$ has been added to the equation for $\mathbf{m}$, a straightforward manipulation of Eq. 8 shows that $\mathbf{u}$ nevertheless satisfies the Navier-Stokes momentum equations, with pressure identified as $p = q + \varphi_t - 2\mu\nabla\cdot\mathbf{m}$. Fourth, one can straightforwardly check that the jump conditions in Eq. 4 are satisfied using Eq. 9 and the formula for pressure p.

To complete the specification of this interfacial gauge method, q is as before chosen to be a harmonic function such that

$$\Delta q = 0 \text{ in } \Omega_1 \cup \Omega_2, \quad [\partial_n q] = 0 \text{ on } \Gamma, \quad \partial_n q = 0 \text{ on } \partial\Omega, \quad (10)$$

and last, the domain boundary conditions are suitably modified to reflect the change in the variable φ,

$$\left.\begin{array}{c} \mathbf{m}\cdot\mathbf{n} = \mathbf{u}_\partial\cdot\mathbf{n} \\ \mathbf{m}\cdot\boldsymbol{\tau} = \mathbf{u}_\partial\cdot\boldsymbol{\tau} + \rho_i^{-1}\nabla\varphi\cdot\boldsymbol{\tau} \\ \partial_n\varphi = 0 \end{array}\right\} \text{ on } \partial\Omega \cap \partial\Omega_i. \quad (11)$$

Together, Eqs. 8 to 11 define an interfacial gauge method for general two-phase incompressible fluid flow. Evidently, the method is more complicated than the previous case where we assumed $[\rho] = 0$ and $[\mu] = 0$. This is partly due to the fact that the interface stress condition, $[\sigma]\mathbf{n} = -\gamma\kappa\mathbf{n}$, involves a nontrivial interplay of the pressure and viscous stress. Moreover, and in analogy with the discussion regarding physical boundary conditions, the jump conditions on $\mathbf{m}$ in Eq. 9 depend on jumps in $\nabla\varphi$ and $D^2\varphi$, which in turn are determined by the solution of a Poisson problem with input $\mathbf{m}$, thereby forming a coupling. As in the case of the domain boundary condition and in regard to time-stepping schemes, this coupling can be simply treated by using a scheme that predicts $\nabla\varphi$ and $D^2\varphi$ from previous time steps. Consequently, the essential tools needed to implement this interfacial gauge method are high-order accurate numerical methods capable of solving elliptic interface problems with phase-dependent coefficients. Such a framework is discussed next.

An implicit mesh discontinuous Galerkin framework

To implement an interfacial gauge method, one can, in principle, apply any numerical methodology equipped to handle jump conditions on

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

evolving interfaces, for example, finite difference methods taking into account internal interfaces, cut-cell finite volume methods, or finite element methods using unstructured grids, which move and adapt to the evolving interface. As part of this work, an alternative approach has been developed, based on an implicit mesh dG method. The framework, briefly described here, has been developed to provide a high-order accurate methodology for multiphase flow, capitalizing on the flexibility of dG methods and the wide array of mathematical and computational advantages offered by implicit interface methods.

Inspired by the work of Johansson and Larson (*44*) and extending previous work (*45*), the "implicit mesh dG" method uses a background reference grid to implicitly define a mesh at every time step, given an implicit description of the interface and domain geometry. For the flows considered here, the level set method (*5*) is used to capture the evolving interface. [Briefly, the level set method defines a function $\phi : \Omega \rightarrow \mathbb{R}$ at $t = 0$ whose zero level set is Γ, and then evolves ϕ by advecting it with the fluid flow such that the interface continues to be the zero level set of ϕ for all $t > 0$. Here, this is done through the simple advection equation $\phi_t + \boldsymbol{u} \cdot \nabla\phi = 0$. See also Sethian (*46*) and Osher and Fedkiw (*47*).] The dG mesh is constructed similarly to "cut-cell" methods but avoids the troublesome ill-conditioning issues arising from "tiny" cut cells by using a simple merging procedure, as follows. Let $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ be the level set function that implicitly defines the interface $\Gamma = \{x \in \Omega : \phi(x) = 0\}$, and denote by $\Omega_1 = \{x \in \Omega : \phi(x) < 0\}$ the region occupied by phase one and $\Omega_2 = \{x \in \Omega : \phi(x) > 0\}$ the region occupied by phase two. Let U be a rectangular domain, which encloses all of Ω. Define a quadtree grid (in 2D) or octree grid (in 3D) such that $U = \cup_i U_i$, where U_i are rectangular elements. This defines a collection of "cells" $U_i \cap \Omega_j$, most of which are rectangular and fall entirely within one phase (denoted entire cells), whereas others intersect $\partial\Omega$ or Γ (partial cells), with the remainder falling outside of Ω (empty) (see Fig. 1). Partial cells that are deemed "small" are merged with nearby cells to ultimately form curved elements: many possibilities exist for deciding whether a cell is small; here, a cell $U_i \cap \Omega_j$ is deemed small if its volume fraction is less than 40%, that is, $|U_i \cap \Omega_j| < 0.4|U_i|$. Small cells are merged with neighboring cells in the same phase according to the following scheme: of the cells sharing a face, edge, or vertex with the small cell, the cell that has the largest volume fraction is used. The result of this process is a mesh composed of standard rectangular elements, elements with curved boundaries, and elements that have been merged with small cells. Figure 1 shows a 2D example.

Note that throughout this procedure, the mesh is never actually explicitly constructed; instead, at every time step, the mesh is implicitly defined by the background grid and the implicitly defined interface. The resulting mesh is automatically body-conforming, and the interface is sharply represented by the faces between elements belonging to different phases. Subsequently, jump conditions on the interface can be imposed with high-order accuracy.

dG methods are amenable to unstructured meshes such as these and easily allow for adaptive mesh refinement. Essentially, the only requirement is to compute the elemental mass matrices and face quadrature rules to integrate numerical fluxes on element boundaries. Most of the elements and their faces are rectangular, hence, a standard reference element can be used. For curved elements, curved faces, or flat faces that are cut by the interface or domain boundary, we use the high-order accurate quadrature algorithms developed by Saye (*45*). These algorithms have been designed specifically for implicitly defined surface and volume geometry, and yield quadrature schemes with the same order of accuracy as tensor-product Gaussian quadrature. The order of accuracy is a user-chosen parameter, which is typically chosen large enough to minimize any associated "variational crimes" in the dG method. Computation of the curved element mass matrices is often the most expensive part of constructing the implicit mesh; nevertheless, the cost of constructing the mesh is generally a small fraction of the overall cost per time step. (Evaluation of the projection operator is typically the most expensive component of each time step; here, the associated Poisson problems are solved with a multigrid-preconditioned conjugate gradient algorithm.)

We defer the precise details of the dG method to forthcoming work and instead briefly describe the essential characteristics. The dG solution space is piecewise polynomial such that each element is a tensor-product polynomial of degree p, therefore having $(p + 1)^d$ degrees of freedom per element. (In the theory of dG methods, what basis is used or how the degrees of freedom are interpreted is irrelevant. In practice, it is advisable to use a well-conditioned basis—here, each element has a nodal basis, corresponding to the tensor-product Gauss-Lobatto nodes attached to the element's "parent" cell in the quadtree/octree.) Discrete differential operators are defined using upwind

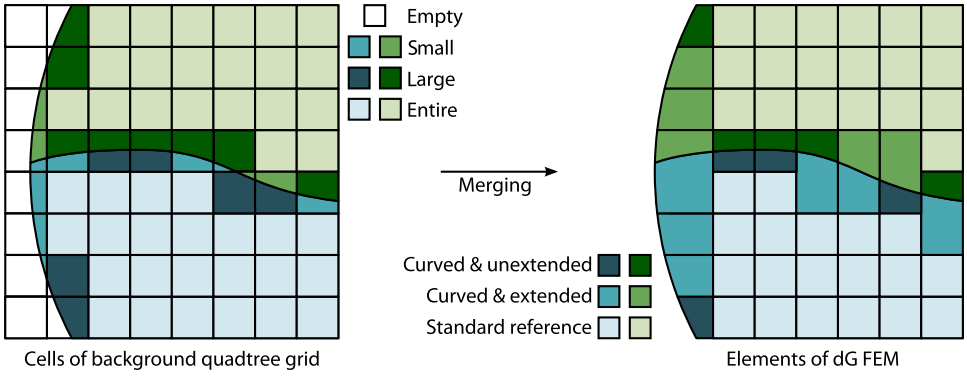

Fig. 1. Defining the mesh in the implicit mesh dG framework. (Left) The cells of a Cartesian grid are classified according to whether they fall entirely within one phase ("entire," light blue/green and rectangular) or entirely outside the domain ("empty," white), or else are denoted "partial." Partial cells are classified according to whether they have a small volume fraction (medium shade blue/green) or large intersection (dark blue/green). (Right) Small cells are merged with neighboring cells in the same phase to form a finite element mesh composed of standard rectangular elements and elements with curved, implicitly defined boundaries.

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

numerical fluxes for the advection operators $\nabla \cdot (\mathbf{uu})$ and $\nabla \cdot (\mathbf{u}\phi)$ in conservative form, whereas for the elliptic operators, a local dG (LDG) (*48*, *49*) formulation is used. The LDG formulation has the advantage that the discrete Laplacian is of the form $-\mathrm{adj}(G)G$, where G is a discrete gradient operator and $-\mathrm{adj}(G)$ is a discrete divergence operator. Consequently, the projection operator $\mathbb{P} = I - G(\mathrm{adj}(G)G)^{-1}\mathrm{adj}(G)$ is discretely idempotent (up to small penalty parameters within the LDG method). A similar discrete operator exists for the stress tensor divergence operator $\nabla\cdot(\nabla\cdot + \nabla(\cdot)^{\mathsf{T}})$ and is symmetric positive semi-definite. All of these discrete differential operators take into account prescribed jump conditions through a suitable definition of the numerical flux on interfacial mesh faces.

In summary, the spatial discretization for the interfacial gauge methods presented in this work is as follows. At each time step, the geometry (domain, interface, or rigid body) is defined implicitly through a piecewise polynomial level set function. The above construction defines a mesh implicitly; as the interface evolves, so does the mesh: for most of the time steps, the cell classifications (empty/small/large/entire) remain the same; hence, the state vectors ($\mathbf{m}$, $\mathbf{u}$, φ, ϕ, etc.) can be "injected" from the old mesh into the new. When the interface moves far enough, the mesh must necessarily change topology—when this occurs, we define the state on the new mesh as an L^2 projection of the state on the old mesh. (The mesh topology update and projection procedure occurs at most $\mathcal{O}(Th^{-1})$ many times in the course of a simulation, where T is the final time and h is the element size. Numerous convergence tests showed that it does not deteriorate the overall order of accuracy.) Other details regarding the implicit mesh dG method, including geometric multigrid methods for efficient solution of the elliptic interface problems with phase-dependent coefficients, dG methods for the level set method, accurate evaluation of curvature κ, and parallelization, will be provided elsewhere.

Time stepping

One appealing aspect of gauge methods is that, because $\mathbf{m}$ is freed from having to satisfy a divergence constraint in its evolution equation, high-order accurate time-stepping schemes can be designed more readily. For example, fourth- and sixth-order time-stepping schemes have been developed (*32*–*34*) for single-phase flow with periodic and no-slip boundary conditions. Here, the coupling that arises between the gauge variables $\mathbf{m}$ and φ [for example, as in the domain boundary condition (Eq. 3) or the interface jump conditions (Eq. 9)] is straightforwardly treated with a predictor of φ computed by extrapolating φ from previous time steps. This predictor is used only as a source term for boundary conditions—at the end of every time step, a new gauge variable is obtained, replacing the predictor. Nonetheless, for fluid flow with moving interfaces, because the location of jump conditions changes in time, it is important to use the correct discrete differential operators evaluated at the appropriate point in time. In the context of the implicit mesh dG method discussed above, this means that the spatial derivative operators change from one mesh to the next.

For all of the interfacial gauge methods presented in this work, a second-order accurate predictor-corrector scheme is used. For simplicity of presentation, we describe the scheme for the case of two-phase flow with no jumps in density or viscosity, that is, the system of Eqs. 5 to 7 and Eq. 3. The time-stepping method is a mixture of explicit and implicit schemes: advection terms are computed explicitly, whereas viscous terms are computed semi-implicitly. We denote by $\mathbf{u}^n$, φ^n, etc., state quantities at time step n; ∇^n, Δ^n, etc., denote discrete differential operators at time step n, derived from and corresponding to the mesh at time step n. Predictor quantities, which are first-order approximations at time step $n+1$, are denoted with a $\star$. For simplicity, the time step Δt is assumed constant.

Given the state at time step n, the predictor step is as follows:

(i) Update the level set function: $\phi^\star = \phi^n - \Delta t\nabla^n \cdot (\mathbf{u}^n \phi^n)$.

(ii) Define a new mesh using the predictor interface $\Gamma^\star = \{\phi^\star = 0\}$.

(iii) Calculate a predictor q by solving $\Delta^\star q^\star = 0$ in Ω, $[q^\star] = \gamma\kappa^\star$ and $[\partial_n q^\star] = 0$ on $\Gamma^\star$, and $\partial_n q^\star = 0$ on $\partial\Omega$, where $\kappa^\star = \nabla \cdot (\nabla\phi^\star/|\nabla\phi^\star|)$ is the mean curvature of the predictor interface.

(iv) Form a second-order predictor of the gauge variable at time step $n+1$: $\varphi^\star := 2\varphi^n - \varphi^{n-1}$.

(v) Calculate a predictor of $\mathbf{m}$ using backward Euler: solve $\rho\mathbf{m}^\star - \Delta t\mu\Delta^\star\mathbf{m}^\star = \rho\mathbf{m}^n - \Delta t(\nabla^n \cdot (\mathbf{u}^n\mathbf{u}^n) + \nabla^\star q^\star)$ subject to the interface jump conditions $[\mathbf{m}^\star] = 0$ and $[\partial_n\mathbf{m}^\star] = 0$ on $\Gamma^\star$ and boundary conditions $\mathbf{m}^\star\cdot\mathbf{n} = \mathbf{u}_\partial^{n+1}\cdot\mathbf{n}$ and $\mathbf{m}^\star\cdot\boldsymbol{\tau} = \mathbf{u}_\partial^{n+1}\cdot\boldsymbol{\tau} + \nabla^\star\varphi^\star\cdot\boldsymbol{\tau}$ on $\partial\Omega$.

(vi) Project $\mathbf{m}^\star$ to find $\mathbf{u}^\star$: solve $\Delta^\star\varphi = (\nabla\cdot)^\star\mathbf{m}^\star$ subject to the jump conditions $[\varphi] = [\partial_n\varphi] = 0$ on $\Gamma^\star$ and boundary condition $\partial_n\varphi = 0$ on $\partial\Omega$; compute $\mathbf{u}^\star = \mathbf{m}^\star - \nabla^\star\varphi$.

With the predictor in hand, the corrector stage uses a mixture of the explicit trapezoid rule and Crank-Nicolson to achieve second-order accuracy:

(i) Update the level set function: $\phi^{n+1} = \phi^n - \frac{1}{2}\Delta t(\nabla^n\cdot(\mathbf{u}^n\phi^n) + \nabla^\star\cdot(\mathbf{u}^\star\phi^\star))$.

(ii) Define a new mesh with the new interface $\Gamma^{n+1} = \{\phi^{n+1} = 0\}$.

(iii) Calculate q^{n+1} by solving $\Delta^{n+1}q^{n+1} = 0$ in Ω, $[q^{n+1}] = \gamma\kappa^{n+1}$ and $[\partial_n q^{n+1}] = 0$ on Γ^{n+1}, and $\partial_n q^{n+1} = 0$ on $\partial\Omega$, where $\kappa^{n+1} = \nabla \cdot (\nabla\phi^{n+1}/|\nabla\phi^{n+1}|)$.

(iv) Compute $\mathbf{m}^{n+1}$ using Crank-Nicolson: solve $\rho\mathbf{m}^{n+1} - \frac{1}{2}\Delta t\mu\Delta^{n+1}\mathbf{m}^{n+1} = \rho\mathbf{m}^n - \frac{1}{2}\Delta t(\nabla^n\cdot(\mathbf{u}^n\mathbf{u}^n) + \nabla^\star\cdot(\mathbf{u}^\star\mathbf{u}^\star) + \nabla^n q^n + \nabla^{n+1}q^{n+1} - \mu\Delta^n\mathbf{m}^n)$ subject to the interface jump conditions $[\mathbf{m}^{n+1}] = [\partial_n\mathbf{m}^{n+1}] = 0$ on Γ^{n+1} and boundary conditions $\mathbf{m}^{n+1}\cdot\mathbf{n} = \mathbf{u}_\partial^{n+1}\cdot\mathbf{n}$ and $\mathbf{m}^{n+1}\cdot\boldsymbol{\tau} = \mathbf{u}_\partial^{n+1}\cdot\boldsymbol{\tau} + \nabla^{n+1}\varphi^\star\cdot\boldsymbol{\tau}$ on $\partial\Omega$.

(v) Project $\mathbf{m}^{n+1}$ to find $\mathbf{u}^{n+1}$: solve $\Delta^{n+1}\varphi^{n+1} = (\nabla\cdot)^{n+1}\mathbf{m}^{n+1}$ subject to the jump conditions $[\varphi^{n+1}] = [\partial_n\varphi^{n+1}] = 0$ on Γ^{n+1} and boundary condition $\partial_n\varphi^{n+1} = 0$ on $\partial\Omega$; compute $\mathbf{u}^{n+1} = \mathbf{m}^{n+1} - \nabla^{n+1}\varphi^{n+1}$.

Upon conclusion of the corrector step, $\mathbf{u}^{n+1}$, $\mathbf{m}^{n+1}$, φ^{n+1}, ϕ^{n+1}, etc., are second-order accurate in time. Several remarks are in order. First, note that an explicit trapezoid rule is used for the advection term. This is necessary because the regularity of $\mathbf{u}^n$ and $\mathbf{u}^\star$ conforms precisely to Γ^n and $\Gamma^\star$, respectively. In other words, although $\mathbf{u}$ is continuous across the interface, derivatives of $\mathbf{u}$ are generally discontinuous. It follows that one cannot use the explicit midpoint method because the regularity of the linear combination $\frac{1}{2}(\mathbf{u}^n + \mathbf{u}^\star)$ does not correspond with any well-defined interface. A similar consideration applies to the curvature source term, in which an explicit trapezoid rule is used to calculate $\frac{1}{2}(\nabla^n q^n + \nabla^{n+1}q^{n+1})$. Second, an extensive array of numerical experiments indicated that the above treatment of the curvature term leads to a stable method: numerical methods for surface tension–driven flow often entail a time-step constraint of $\Delta t < Ch^{3/2}$, which relates to a grid-dependent capillary wave speed, where C is a constant depending on density and γ, and h is the smallest element size (*6*); however, no such stability constraint has been observed here to be necessary. Instead, experiments indicated that the only time-step constraint required for stability is the usual advective constraint of $\Delta t \le C\min h/|\mathbf{u}|$, where h is the local element size and $|\mathbf{u}|$ is the local

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

fluid speed; empirically, $C \approx 0.1$ sufficed, consistent with typical stability constraints for dG methods of the chosen order [wherein, more generally, C decreases with larger p (*50*)]. Third, except for previous values of the gauge variable, for example, φ^{n-1}, the predictor-corrector scheme is a one-step method—this aids in simplicity of implementation, as then the discrete differential operators do not need to be maintained for previous time steps; instead, it is possible to implement the above scheme in such a way that only one mesh exists at any point in time. Fourth, if a first-order method is sufficient, then one can neglect the corrector step: the computed results of the predictor step lead to a stable first-order accurate time-stepping method that is high-order accurate in space. (In contrast, first-order accurate archetypical projection methods, which solve directly for **u**, often exhibit numerical boundary layers adjacent to the interface that affect spatial accuracy, regardless of how high-order accurate the spatial discretization may be.) For the results computed in this work, second-order accuracy in time has been sufficient. In principle, time-stepping schemes that are higher than second-order accurate could be developed along the same lines, provided each stage of the temporal scheme appropriately considers the regularity of the intermediate quantities.

The above predictor-corrector scheme is suitable for the interfacial gauge method for surface tension having no jumps in density and viscosity. In the case that the density or viscosity is different in the two phases, that is, when Eqs. 8 to 11 constitute the system of equations to be solved, a similar predictor-corrector scheme can be used. The main difference is that the interface jump conditions on **m** and q now depend on the Hessian of φ and on φ_t. These can be computed from the Hessian of the gauge predictor $\varphi^\star$, whereas a predictor of φ_t can be formed by using φ^{n-2} in addition to φ^{n-1} and φ^n. Last, we remark that it is typically straightforward to initialize an interfacial gauge method by choosing $\mathbf{m}^0 = \mathbf{u}^0$ and $\varphi \equiv 0$ at $t = 0$. If φ_t is needed as a source term and is not zero at $t = 0$, then typical bootstrapping methods can be used to compute an effective φ^{-1} and φ^{-2} by using, for example, a first-order time-stepping method as part of an iterative scheme.

Convergence tests. Numerous convergence tests have been performed for all of the interfacial gauge methods presented in this work. The analysis consists mainly of grid convergence tests, using a maximum-norm metric to confirm that high-order convergence is obtained uniformly throughout the domain, including immediately adjacent to the interface. In all cases, in all computed quantities, second-order temporal accuracy has been confirmed. To illustrate the spatial order of accuracy, which depends on the order p of the implicit mesh dG method, Fig. 2 shows the results of a 2D grid convergence analysis (described in more detail in the Supplementary Materials) of the interfacial gauge methods for surface tension dynamics. These tests used a fixed time step Δt, small enough to ensure that temporal errors are negligible, and a long enough time horizon to ensure that key dynamics are being examined. Figure 2 shows the maximum-norm error (in space and time) of the fluid velocity **u**, pressure p, and interface location ϕ: for $p = 2$ and $p = 3$, second-order accuracy is obtained in all quantities; and for $p = 4$, fourth-order accuracy is obtained in all quantities. This dependence on the parity of p is directly due to the numerical treatment of evaluating κ (see also a discussion in the Supplementary Materials). For tensor-product degree p^d polynomials, the optimal order of accuracy one can typically expect in a dG method is $p + 1$. Because curvature is a second-order differential, up to two orders of accuracy compared to the optimal may be lost, which our results show is the case for $p = 3$. Therefore, it is favorable that $p = 2$ yields second-order accuracy and $p = 4$ yields fourth-order accuracy. Regardless of the treatment of curvature, we emphasize two aspects. First, note that high-order accuracy in pressure is obtained—certain types of projection methods yield numerical boundary layers in pressure, often leading to first-order accuracy (*12*). Second, high-order accuracy is obtained in the velocity, showing that there are no "parasitic currents" next to the interface; such parasitic currents often arise in methods that regularize interfacial forces by using smoothed delta functions. For more details, including an analogous convergence analysis in three dimensions attaining similar convergence rates, refer to the Supplementary Materials.

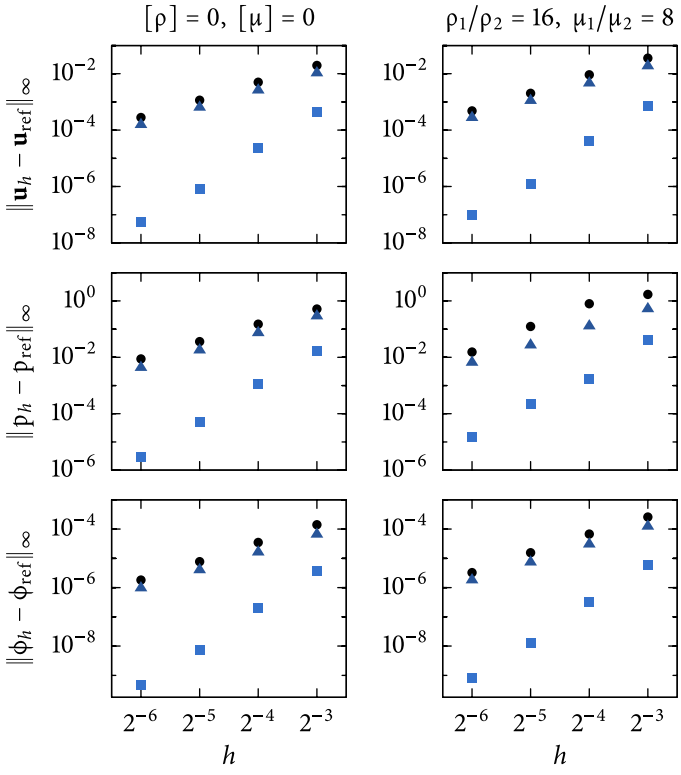

Fig. 2. Results of a grid convergence analysis (described in more detail in the Supplementary Materials) for the interfacial gauge method without jumps in ρ or μ (left column) and a jump ratio of 16 in ρ and 8 in μ for the more general interfacial gauge method (right column). Here, $\mathbf{u}_h$, p_h, and ϕ_h are the computed solutions obtained with the implicit mesh dG method using a uniform reference Cartesian grid of cell size h, whereas $\mathbf{u}_{ref}$, etc., is the reference solution computed with the highest p on a grid with $h = 2^{-7}$; • denotes $p = 2$, ▲ denotes $p = 3$, and ■ denotes $p = 4$.

Capillary waves in two-phase surface tension dynamics

To demonstrate the interfacial gauge methods for surface tension dynamics, two examples are given, involving capillary wave dynamics revealing intricate fine-scale flow features. The initial condition, shown in the top left panels of Figs. 3 and 4, consists of two fluids separated by an interface resembling a soap bubble shortly after coalescing with a flat film. The two examples consist of (i) a case where both phases have the same density $\rho = 1$ and viscosity $\mu = 0.001$, utilizing the interfacial gauge method in Eqs. 5 to 7, and (ii) a case where the fluid on

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

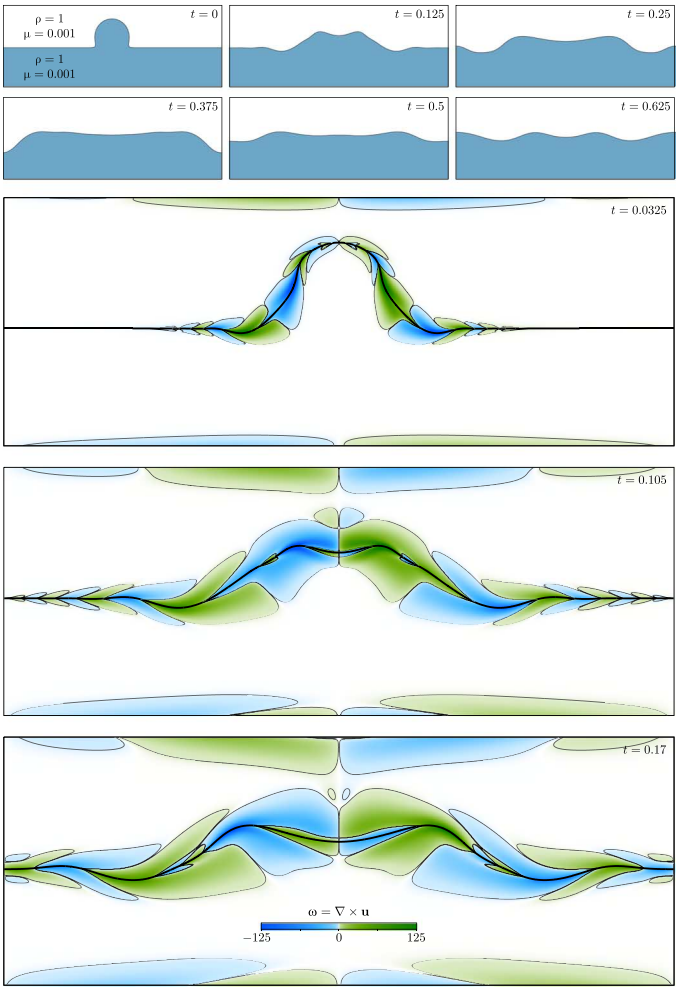

Fig. 3. Capillary waves in surface tension dynamics: An example in which [ρ] = 0 and [μ] = 0. (Top) Initial condition together with a sample of frames to provide an indication of the resulting wave dynamics. (Bottom) A plot of vorticity $\omega = \nabla \times \mathbf{u}$ at three moments in time—the heart-shaped features precisely correspond with the wavelengths of the capillary waves. Note also the vorticity generation at the no-slip walls near the top and bottom. Computed with the interfacial gauge method (Eqs. 5 to 7 plus 3) and the implicit mesh dG method with $p = 3$.

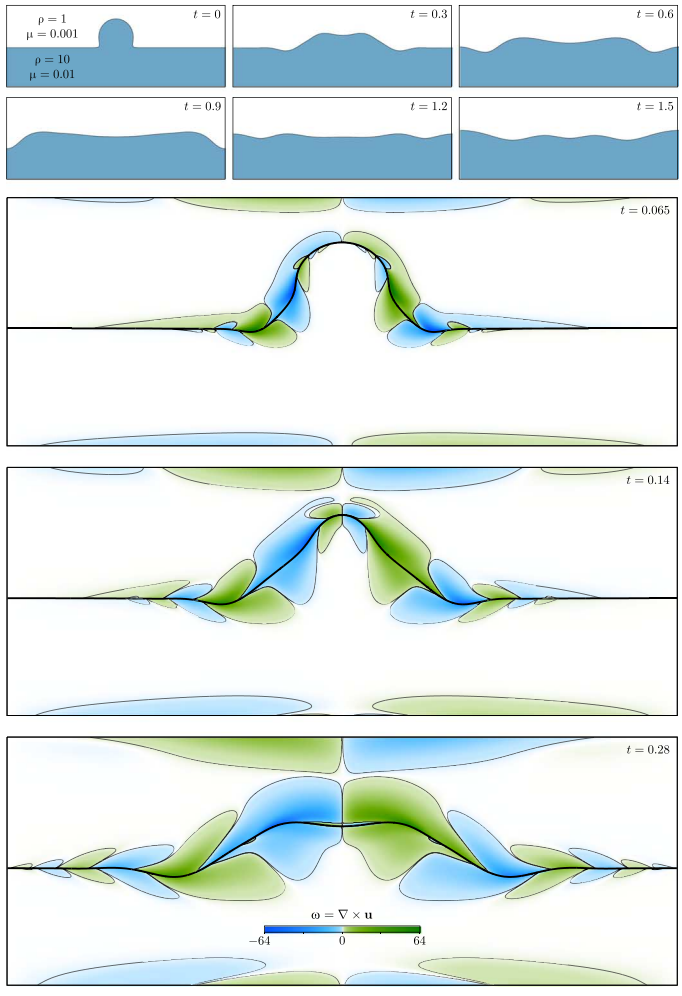

Fig. 4. Capillary waves in surface tension dynamics: An example involving a jump ratio of 10 in both density and viscosity. (Top) Initial condition together with a sample of frames to provide an indication of the resulting wave dynamics. (Bottom) A plot of vorticity $\omega = \nabla \times \mathbf{u}$ at three moments in time—similar to Fig. 3, there are heart-shaped features corresponding to the wavelengths of the capillary waves, but in this case, the vorticity is discontinuous across the interface. Computed with the interfacial gauge method (Eqs. 8 to 11) and the implicit mesh dG method with $p = 3$.

top has a density of $\rho = 1$ and a viscosity of $\mu = 0.001$ and the fluid on bottom has a density of $\rho = 10.0$ and a viscosity of $\mu = 0.01$, utilizing the interfacial gauge method in Eqs. 8 to 11. In both cases, the surface tension coefficient is $\gamma = 0.1$, no-slip boundary conditions are prescribed on the top and bottom horizontal boundaries, and the flow is periodic in the horizontal direction and is initially stationary, $\mathbf{u} \equiv 0$ at $t = 0$.

Figures 3 and 4 illustrate the resulting dynamics, wherein the surface tension of the initial "bubble" causes it to collapse, sending out capillary waves in the process. The figures plot the vorticity $\omega = \nabla \times \mathbf{u} = \partial_x u_2 - \partial_y u_1$ of the flow, which has intricate "heart"-shaped features that align with the wavelengths of the capillary waves. In particular, for the case with no jump in viscosity, the vorticity is continuous across the interface (because $[\nabla \mathbf{u}] = 0$, as per the derivation in the Supplementary Materials), whereas for $[\mu] \neq 0$, the vorticity is generally discontinuous. These discontinuities (including those present in pressure and higher derivatives of $\mathbf{u}$) are sharply captured by the implicit mesh dG method and are high-order accurate owing to the design of interfacial gauge methods. Using a maximum fluid speed of $|\mathbf{u}| \approx 1.8$ (respectively, 0.6) and the length scale of the domain, the Reynolds number for the flow in Fig. 3 is approximately 1800 (respectively, 600 for Fig. 4).

Interfacial gauge methods for rigid body fluid-structure interaction

A challenging aspect of modeling fluid-structure interaction problems is the coupling between the dynamics of the fluid and the physics of the structure. This coupling is often numerically treated in one of two ways: either with a fractional stepping approach (in which the structure is temporarily frozen for part of the time step and the fluid is updated,

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

and then the fluid is frozen and the position of the structure is updated to complete the time step) or with a "monolithic"-type approach (where both the fluid and structure are solved for simultaneously, as part of a large, fully coupled nonlinear system). In particular, for incompressible fluids, challenges arising from the coupling are due, in part, to the incompressibility constraint. Gauge methods provide an alternative means for treating this aspect; this is demonstrated here by developing an interfacial gauge method for a rigid body embedded in an incompressible fluid. This approach does not require the solution of a large fully coupled system, yet high-order accuracy in space and time can still be obtained.

Consider a rigid body Ω_b surrounded by a fluid of density ρ and viscosity μ. The motion of the body is determined by the net force and torque exerted on the body by the fluid, which, in turn, is determined by integrating the normal component of the fluid stress $\boldsymbol{\sigma}$ around the body surface $\Gamma := \partial\Omega_b$. Because the body is rigid, the velocity field inside the body is given by $\mathbf{u}_b(x) = v_b + \omega \times (x - x_c)$, where v_b is its linear velocity, ω is its angular velocity, and x_c is its center of mass. These quantities evolve according to conservation of linear and angular momentum, leading to

$$\begin{aligned} \dot{x}_c &= v_b, \\ m\dot{v}_b &= \textstyle\int_\Gamma \boldsymbol{\sigma}\mathbf{n} + mg, \\ \frac{d}{dt}(I_b\omega) &= \textstyle\int_\Gamma (x - x_c) \times \boldsymbol{\sigma}\mathbf{n}, \end{aligned} \tag{12}$$

where m is the body's mass, g is gravity, I_b is the body's moment of inertia, and $\mathbf{n}$ is the unit normal on Γ pointing into the fluid. The fluid in the domain $\Omega_f = \Omega \setminus \Omega_b$ solves the Navier-Stokes equations with gravitational forcing, subject to the boundary condition $\mathbf{u} = \mathbf{u}_b$ on Γ.

To design an interfacial gauge method for rigid body fluid interaction, we can again make use of a q term in the equations for $\mathbf{m}$. In the surface tension–driven flows considered previously, the role of q was to represent the force of surface tension on the bulk fluid. In analogy, q can be used here to represent the normal force of the body on the fluid. Consider a particle $x = x(t)$ fixed to the surface of the body. Then, its velocity is $\dot{x} = v_b + \omega \times (x - x_c)$ and acceleration is $\ddot{x} = \dot{v}_b + \dot{\omega} \times (x - x_c) + \omega \times (\dot{x} - \dot{x}_c) = \dot{v}_b + \dot{\omega} \times (x - x_c) + \omega \times (\omega \times (x - x_c))$. Since ∇q is a body force, we therefore choose Neumann boundary conditions for q such that $-\partial_n q = \rho(\dot{v}_b + \dot{\omega} \times (x - x_c) + \omega \times (\omega \times (x - x_c))) \cdot \mathbf{n}$. This leads to the following interfacial gauge method: let $\mathbf{m}$ solve

$$\left.\begin{aligned} \rho(\mathbf{m}_t + \mathbf{u} \cdot \nabla\mathbf{u}) &= -\nabla q + \mu\Delta\mathbf{m} \\ \mathbf{u} &= \mathbf{m} - \nabla\varphi \\ \Delta\varphi &= \nabla \cdot \mathbf{m} \\ \Delta q &= 0 \end{aligned}\right\} \text{ in } \Omega_f, \tag{13}$$

subject to the interface conditions

$$\left.\begin{aligned} \mathbf{m}\cdot\mathbf{n} &= \mathbf{u}_b\cdot\mathbf{n} \\ \mathbf{m}\cdot\boldsymbol{\tau} &= \boldsymbol{u}_b\cdot\boldsymbol{\tau} + \nabla\varphi\cdot\boldsymbol{\tau} \\ \partial_n\varphi &= 0 \\ \partial_n q &= -\rho\left(\dot{v}_b + \dot{\omega}\times(x - x_c) + \omega\times\left(\omega\times(x - x_c)\right)\right)\cdot\mathbf{n} \end{aligned}\right\} \text{ on } \Gamma. \tag{14}$$

Then, $\mathbf{u}$ solves the Navier-Stokes equations in Ω_f with gravitational forcing, with pressure identified as $p = q + \rho\varphi_t - \mu\nabla \cdot \mathbf{m} + \rho G$, where $G = G(x) := g \cdot x$ is the gravitational potential. Equations 12 to 14, plus Eq. 3 for the physical boundary, constitute the interfacial gauge method.

A similar predictor-corrector time-stepping method can be used to solve the system of equations. In this case, both the predictor and corrector step involve the computation of the surface integrals in Eq. 12, which can be carried out with high-order accuracy in the implicit mesh dG framework. In particular, because the body remains rigid, we can use a fixed level set function ϕ that implicitly defines its shape, defined in "body coordinates"; the body coordinates can then be mapped to world coordinates with the aid of a rotation matrix R that evolves in correspondence with ω; in fact, $\dot{R} = S(\omega)R$, where $S(\cdot)$ is the cross-product matrix. Orthogonality of the rotation matrix can be maintained by incorporating Rodrigues' rotation formula into the predictor-corrector scheme. The time-stepping scheme is conceptually straightforward, and for brevity of presentation, details are omitted.

A tumbling rigid body. Figure 5 demonstrates this interfacial gauge method with an example consisting of an ellipsoid falling under the action of gravity in a tall column of liquid periodic in the vertical direction. The ellipsoid has a nonuniform density and is initially top-heavy, that is, its center of mass starts above its center of buoyancy. As the body falls, viscous effects initially resist the body's inclination to rotate, but these effects are shortly overcome and the body rotates, subsequently "kissing" the channel boundaries. In this example, the fluid has a density of $\rho = 1$ and a viscosity of $\mu = 0.001$, gravity is $|g| = 0.1$ pointing down, and the average body density is approximately seven times more than the fluid. Using the width of the channel as the length scale and maximum speed of the body ($\approx$0.6), the Reynolds number for this flow is approximately 600. The fluid velocity and pressure are sharply resolved adjacent to the body surface, and convergence tests indicate that they, in addition to the body's position, are computed with high-order accuracy in space (at least second order, depending on p) and second-order accuracy in time.

Interfacial gauge methods for free surface flow

Our last application of gauge methods in evolving domains concerns that of free surface flow, for example, traveling waves on the surface of a lake. In a free surface flow problem, the physics taking place exterior to the fluid is taken as negligible. Hence, there are no Dirichlet boundary conditions for the fluid velocity on the free surface, but instead, a boundary condition exists for the normal component of the fluid stress: for a surface exhibiting surface tension, as considered here, the normal stress of the fluid balances the force of surface tension. In the presence of gravity, where $G = G(x) = g \cdot x$ is the gravitational potential, the velocity field $\mathbf{u}$ of the fluid satisfies

$$\begin{aligned} \left.\begin{aligned} \rho(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}) &= -\nabla p + \mu\Delta\mathbf{u} + \rho\nabla G \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned}\right\} &\text{ in } \Omega, \\ \mathbf{u} = \mathbf{u}_\partial \quad &\text{on } \partial\Omega_{\text{in/out}}, \\ \boldsymbol{\sigma}\mathbf{n} = -\gamma\kappa\mathbf{n} \quad &\text{on } \Gamma, \end{aligned} \tag{15}$$

where $\mathbf{n}$ points exterior to the fluid. The domain Ω, the inflow and outflow parts of the domain boundary $\partial\Omega_{\text{in/out}}$, and the free surface Γ all change in time.

For a free surface flow problem, in which $\mathbf{u} \cdot \mathbf{n}$ on Γ is a priori unknown, it is most natural to choose homogeneous Dirichlet boundary conditions for φ on the evolving interface (as discussed further in the

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

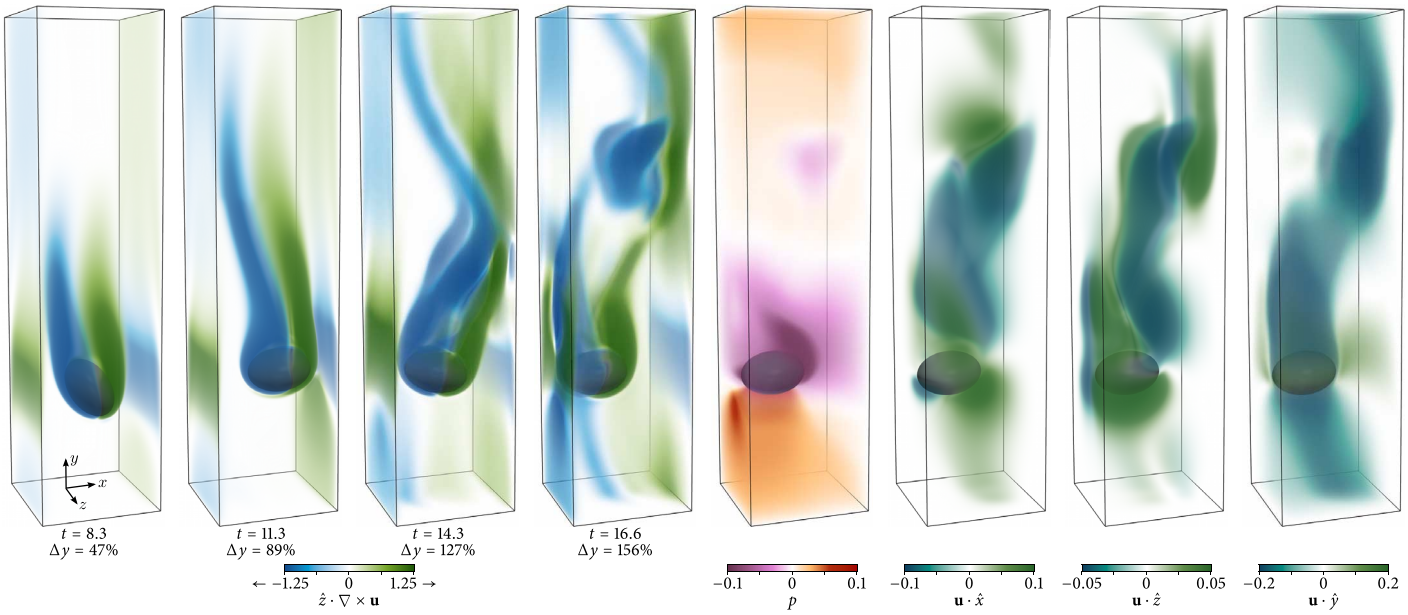

Fig. 5. A tumbling rigid body submersed in a tall channel of liquid falling under the action of gravity. (Left four) Volume renderings of the outward-pointing component of vorticity; from left to right: (i) body rotating clockwise, shortly before flipping over, (ii) shortly after flipping over, (iii) after it has rebounded and unsteadily moved to the other side, and (iv) nearly touching the left wall. The images are shown in a frame of reference attached to the body's center of mass; the Δy values indicate how far the body has fallen as a percentage of the height of the channel. (Right four) Volume renderings of other state quantities at $t = 16.6$ showing pressure p (excluding the hydrostatic component) and the components of the velocity field. Computed with the interfacial gauge method (Eqs. 12 to 14), and the implicit mesh dG method with $p = 3$.

Supplementary Materials). This leads to the following interfacial gauge method: let $\mathbf{m}$ solve

$$\left.\begin{aligned} \rho(\mathbf{m}_t + \mathbf{u}\cdot\nabla\mathbf{u}) &= -\nabla q + \mu\nabla\cdot(\nabla\mathbf{m} + \nabla\mathbf{m}^{\mathsf{T}}) \\ \mathbf{u} &= \mathbf{m} - \nabla\varphi \\ \Delta\varphi &= \nabla\cdot\mathbf{m} \\ \Delta q &= 0 \end{aligned}\right\} \text{ in } \Omega, \qquad (16)$$

subject to the free surface interface conditions

$$\left.\begin{aligned} \varphi &= 0 \\ \mu(\nabla\mathbf{m} + \nabla\mathbf{m}^{\mathsf{T}})\mathbf{n} &= 2\mu\Big((D^2\varphi)\mathbf{n} - (\Delta\varphi)\mathbf{n}\Big) \\ q &= \gamma\kappa - \rho G - \rho\varphi_t \end{aligned}\right\} \text{ on } \Gamma, \qquad (17)$$

and domain boundary conditions

$$\left.\begin{aligned} \mathbf{m}\cdot\mathbf{n} &= \mathbf{u}_\partial\cdot\mathbf{n} \\ \mathbf{m}\cdot\boldsymbol{\tau} &= \mathbf{u}_\partial\cdot\boldsymbol{\tau} + \nabla\varphi\cdot\boldsymbol{\tau} \\ \partial_n\varphi &= \partial_n q = 0 \end{aligned}\right\} \text{ on } \partial\Omega\backslash\Gamma. \qquad (18)$$

Then, $\mathbf{u} = \mathbb{P}(\mathbf{m})$ solves the Navier-Stokes free surface equations (Eq. 15) with gravitational forcing. Note that, similar to the interfacial gauge method for two-phase flow with different densities/viscosities, $\mathbf{m}$ solves a PDE involving the stress tensor divergence operator $\nabla\cdot(\nabla\mathbf{m} + \nabla\mathbf{m}^{\mathsf{T}})$, which is key to enforcing the normal stress boundary condition in Eq. 15. In a similar fashion, one can straightforwardly design a predictor-corrector scheme that can be used as a second-order accurate time-stepping method.

Plateau-Rayleigh instability–induced water ripples. To illustrate the interfacial gauge method for free surface flow, we reproduce an intricate fluid phenomena that can be readily observed at home: when a steady stream of water with a diameter of 3 to 5 mm exits a tap and downstream is obstructed, for example, by a finger, mildly steady ripples are seen in the stream immediately above the obstruction. These ripples are caused by surface tension and the Plateau-Rayleigh instability that acts to collapse a cylinder of liquid; capillary waves are created, which are then sustained by the steady inflow of liquid from the tap.

In the example considered here, we aim to validate the numerical methods developed in this work by comparing them with an experimental study of the Plateau-Rayleigh instability conducted by Hancock and Bush (*51*) (see Fig. 6 inset). In that work, it is argued that sufficiently far upstream of the ripples, the jet flow can be considered as plug flow, that is, with uniform velocity across its circular cross section. This profile is adopted here as an inflow boundary condition, where, based on quoted experimental parameters and photographs (*51*), the jet speed is estimated to be 84 cm s^{-1}, uniform across the jet with a diameter of 2.2 mm, with an estimated jet height of 16 mm, while the remaining parameters are taken as $\rho = 1$ g cm^{-3}, $\mu = 0.012$ g cm^{-1} s^{-1}, $\gamma = 70.5$ dynes cm^{-1}, and $g = 980$ cm s^{-2}. Furthermore, in the cited experiments, the water jet strikes a large reservoir of water. However, because modeling a large container of water is computationally expensive, we here instead consider a mildly deep "well," situated underneath a flat

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

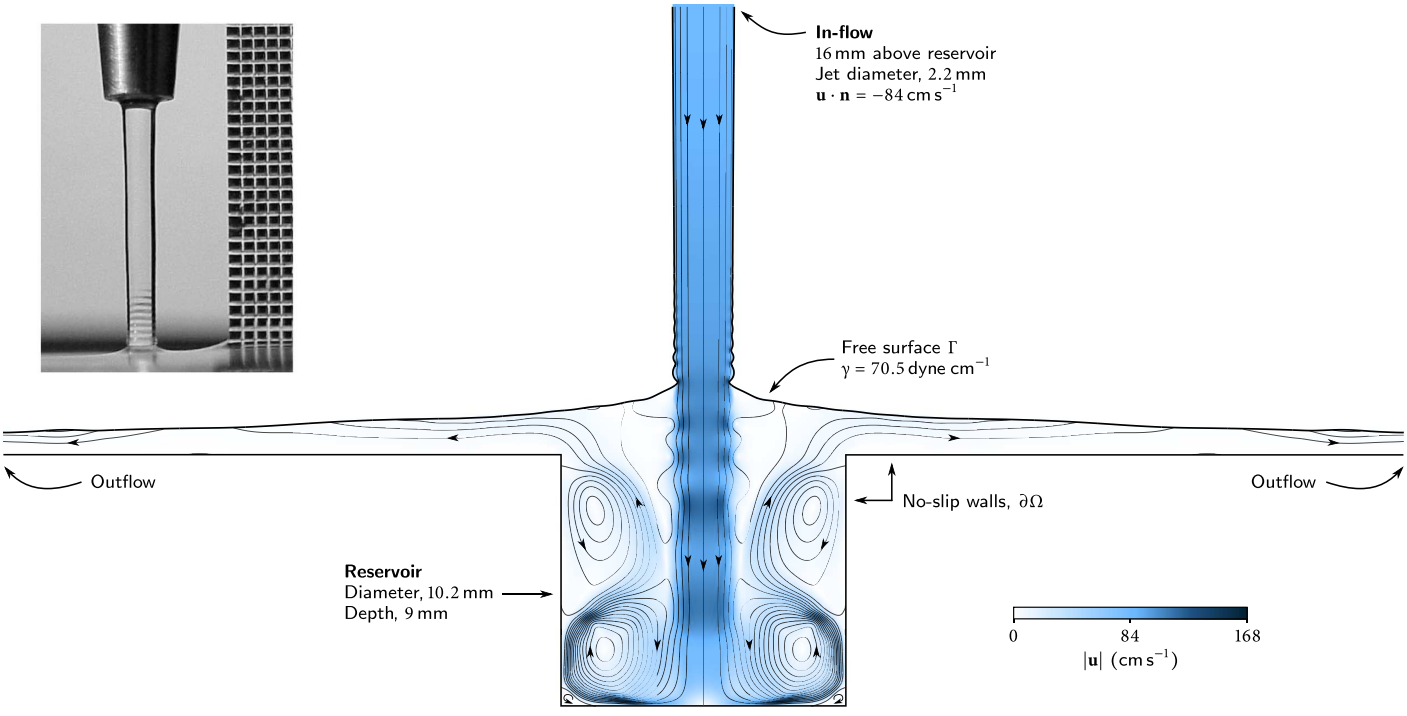

Fig. 6. Water ripples in a free surface flow. (**Inset**) Experimental image reproduced with permission from Hancock and Bush (*51*) showing a jet of water exiting a nozzle and entering a reservoir. When the stream is steady, surface tension capillary waves can be seen traveling up the stream. These ripples are caused by the Plateau-Rayleigh instability, which acts to collapse a cylindrical tube of liquid; however, in this circumstance, the waves are steadily maintained by the constant influx of water from the jet. The grid seen in the photo is millimetric. (**Main**) Results of a numerical simulation aimed at reproducing this ripple phenomenon. Shown are the streamlines of the flow and the speed of the water, as well as the chosen domain layout and boundary conditions: the jet radius and inflow speed, as well as physical parameters, such as water density and viscosity, match those of the experiment—note that the computed results successfully reproduce the ripple phenomenon at the base of the jet, the shape and wavelengths of which are in good agreement with the experimental result. See also Fig. 7 for zoomed-in plots of the computed fluid pressure and vorticity. Results computed with the interfacial gauge method (Eqs. 16 to 18) and the implicit mesh dG method with $p = 3$ in cylindrical coordinates in which axisymmetric solutions are sought.

table, to ensure that the outflow boundary conditions are far away from the jet and do not affect the ripple dynamics; Fig. 6 shows the domain layout used. Last, we capitalize on the symmetry of the problem by computing in cylindrical coordinates, seeking an axisymmetric solution.

In this parameter regime, the viscosity of water has little effect on the wavelength of the ripples; however, it does affect their amplitude decay up the stream (*51*), as well as their stability. The small viscosity of water also makes for a challenging computation, due, in part, to the lack of an ideal initial condition. For example, if the jet of liquid is simply instantly turned on, or even slowly turned on, then a high degree of turbulent flow is obtained, taking an impractical number of time steps to find a steady state. As an alternative strategy, the initial condition used here consists of a full reservoir of liquid and a steady jet stream but with a liquid 15 times more viscous than water—this allows a steady-state free surface flow to be achieved more quickly, exhibiting no ripples. After this steady state is found, the viscosity of the liquid is then slowly ramped down to water's, allowing the ripples to slowly form. Once the final prescribed viscosity is attained, this mechanism results in a flow in which the free surface is approximately stationary.

Figure 6 shows the computed flow profile a short time after the steady ripple profile is attained. In regard to the ripple geometry, a good agreement between the experiment and computation (utilizing the interfacial gauge method, Eqs. 16 to 18) is obtained. Nevertheless, numerical experiments indicate that the ensuing dynamics and ripple profiles are remarkably sensitive to the jet speed, surface tension coefficient, and viscosity of the liquid. There is also a wide breadth of scales involved: Fig. 6 shows that the top part of the jet and the outflow profile are relatively steady, whereas the reservoir exhibits many eddies of different sizes, caused, in part, by the vortices that are shed at the base of the ripples, themselves having smaller-scaled dynamics. This latter phenomenon is shown more clearly in Fig. 7 with a zoomed-in view of the ripple profile, fluid pressure, and vorticity. A time series of the vortex shedding is shown in Fig. 8; the (axisymmetric) vortex tubes, which are not visible in the experiments of Hancock and Bush (*51*), are shed with nearly constant frequency, moving down into the reservoir along with the bulk fluid from the jet, later merging with the largest eddy at the base of the reservoir. To provide an indication of the numerical resolution required for the overall flow, using the maximum fluid velocity (realized at the base of the ripples; ≈ 170 cm s^{-1}) and the radius of the reservoir, the Reynolds number is approximately 7200; the computed

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

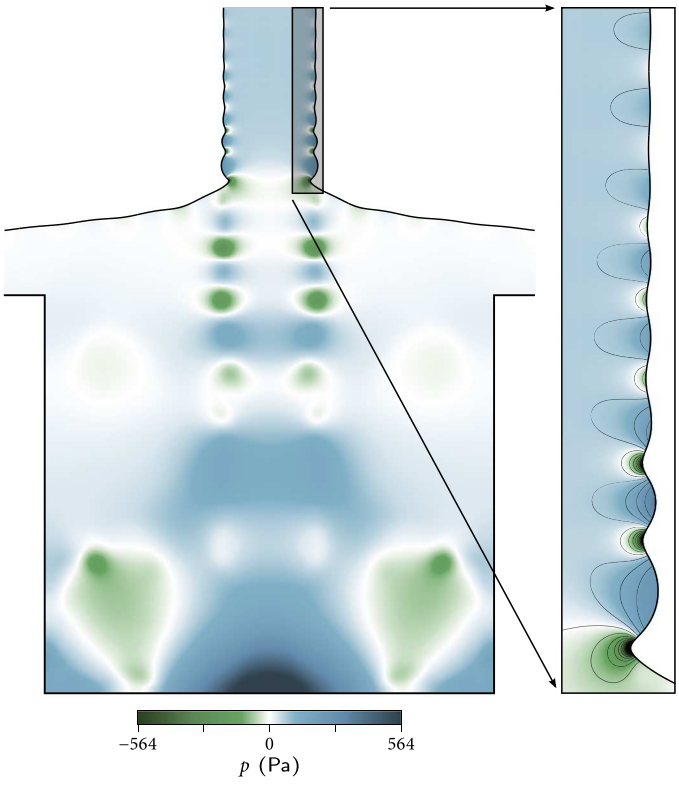

Fig. 7. Fluid pressure and vorticity in the free surface flow of Fig. 6. In both figures, a zoomed-in view of the ripple profile is given, illustrating the wide range of scales taking place within this particular fluid flow phenomenon.

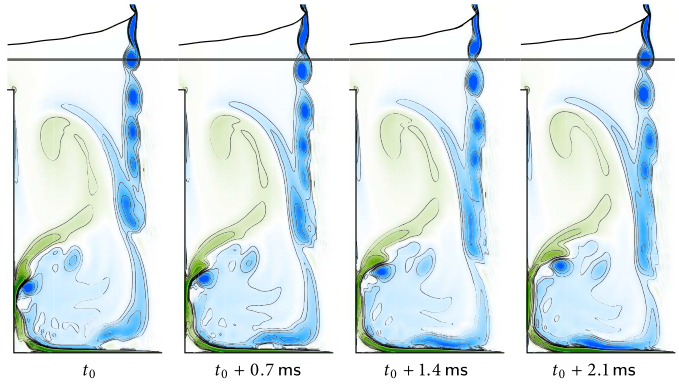

Fig. 8. Vortex shedding in the free surface flow of Fig. 6. Illustrated is a time series of the fluid vorticity $\nabla \times \mathbf{u}$ (sharing the same scale as Fig. 7) restricted to the left half; overlaid is a horizontal guide line to highlight the downward movement of vortices (vortex tubes by axisymmetry) over a period of 2.1 ms.

results required approximately 30,000 elements, each one a bicubic polynomial in the dG framework, to smoothly and correctly resolve the ripple-induced dynamics.

CONCLUSION

Here, a variety of interfacial gauge methods have been introduced, extending single-phase gauge methods to multiphase incompressible fluid flow and free surface flow. These methods take advantage of the gauge freedom in choosing φ and the evolution equations for **m**, as well as their interfacial jump conditions, such that there is a relatively weak coupling of fluid velocity, pressure, and interface geometry, allowing stable and high-order accurate numerical methods to be designed. Using the implicit mesh dG framework, second- and fourth-order accuracy in space was demonstrated, in the maximum norm, in all computed quantities, including pressure. A second-order accurate time-stepping method has been sufficient here; nevertheless, building higher-order temporal schemes for interfacial gauge methods is likely possible. Furthermore, the methods also work in curved domains with time-dependent velocity boundary conditions, although no such examples were included in this work.

It is important to note the similarities, and dissimilarities, with traditional projection methods for multiphase fluid flow. If one was to implement a gauge method, but at the end of every time step overwrite $\mathbf{m}^{n+1}$ with $\mathbf{u}^{n+1}$ and set $\varphi^{n+1} \equiv 0$, then a conventional projection method is obtained. Essentially, in such a setting, **m** becomes the intermediate velocity field that is projected at the end of every time step, and φ adopts a role directly related to fluid pressure. However, such a procedure often introduces numerical boundary layers into the computed pressure and therefore reduces the order of accuracy of the computed velocity field (*12*–*14*). This degradation is due, in part, to the projected velocity field satisfying physically inconsistent tangential boundary conditions, and in the context of interfacial flow, the velocity field is also likely to satisfy physically inconsistent tangential jump conditions on the moving interface (sometimes referred to as parasitic currents). As was demonstrated by Brown *et al.* (*12*) for

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

physical boundaries, interfacial gauge methods could be used to analyze these interfacial numerical boundary layers. Thus, although both archetypical projection methods (that directly solve for **u**) and interfacial gauge methods (which solve for **m**) make essential use of projection operators, they differ in how the boundary conditions and interface jump conditions are numerically treated.

On a related note, it can sometimes be beneficial to occasionally perform the above-mentioned reset or reinitialization, in which **m** is replaced by **u** and φ is reset to zero, provided that this reset is performed relatively infrequently over the course of a simulation. The reason for doing so is that, as **m** evolves, it may develop finer spatial scales than those exhibited in the physical velocity field **u**, which may subsequently require unnecessary spatial resolution to resolve. This phenomenon has also been observed in single-phase gauge methods (*18*, *23*, *24*, *26*, *40*), where it has been noted that the norm of **m** may grow in time without bound (depending on the dynamics taking place). For the examples presented in this work, the capillary wave dynamics (Figs. 3 and 4) did not require reinitialization, the tumbling rigid body (Fig. 5) needed occasional reinitialization as the body continued to accelerate, and the free surface flow (Fig. 6) needed occasional reinitialization with frequency proportional to that of the vortex shedding. In these instances, the decision to reinitialization was largely based on heuristics regarding the dynamics of **m** and related resolution requirements; it would be worthwhile to investigate more precise metrics for determining when to reinitialize. It may also be possible to combine the advantages offered by interfacial gauge methods together with frequent reinitialization in a way that does not incur a loss of stability or accuracy—this is the subject of ongoing work.

We conclude with some avenues of future research. Although the interfacial forces considered in this work consisted of constant coefficient surface tension, it is expected that interfacial gauge methods can also be designed for other types of interfacial flow, such as those due to Marangoni forces created by a flow of surfactant rendering γ variable along an interface, elastic forces caused by deforming elastic membranes, and other types of non–rigid body fluid-structure interaction. In designing such methods, different types of gauge freedom could be exploited to obtain stable and accurate methods. In addition, the implicit mesh dG framework allows for multiphase (three or more) fluid flow problems to be considered, wherein triple junction motion plays a crucial role (*52*, *53*); it is expected that the derived jump conditions carry over naturally to this setting as well. Along similar lines, it would be worthwhile to consider the implementation of contact angle boundary conditions, allowing slip of an interface against a wall boundary. Last, extension to inviscid interfacial flow would also be of interest—in this case, one would impose $[\mathbf{m}] \cdot \mathbf{n} = 0$, but there would be no jump conditions in the tangential direction.

SUPPLEMENTARY MATERIALS

Supplementary material for this article is available at http://advances.sciencemag.org/cgi/content/full/2/6/e1501869/DC1

section S1. Jumps of differential operators on curved surfaces

section S2. Remarks on free surface flow boundary conditions for φ

section S3. Two-phase flow with jumps in density and the contribution of φ_t to pressure

section S4. An interfacial gauge method liberated from calculating curvature

section S5. Convergence tests

fig. S1. Results of a 3D grid convergence analysis for the interfacial gauge method without jumps in ρ or μ (left column) and a jump ratio of 16 in ρ and 8 in μ for the more general interfacial gauge method (right column).

REFERENCES AND NOTES

1. C. S. Peskin, The immersed boundary method. *Acta Numerica* **11**, 479–517 (2002).
2. R. Mittal, G. Iaccarino, Immersed boundary methods. *Annu. Rev. Fluid Mech.* **37**, 239–261 (2005).
3. R. J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.* **31**, 1019–1044 (1994).
4. Z. Li, K. Ito, *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006).
5. S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* **79**, 12–49 (1988).
6. J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension. *J. Comput. Phys.* **100**, 335–354 (1992).
7. M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* **114**, 146–159 (1994).
8. J. A. Sethian, P. Smereka, Level set methods for fluid interfaces. *Annu. Rev. Fluid Mech.* **35**, 341–372 (2003).
9. R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.* **152**, 457–492 (1999).
10. C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* **39**, 201–225 (1981).
11. A. J. Chorin, Numerical solution of the Navier-Stokes equations. *Math. Comput.* **22**, 745–762 (1968).
12. D. L. Brown, R. Cortez, M. L. Minion, Accurate projection methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.* **168**, 464–499 (2001).
13. J. L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows. *Comput. Methods Appl. Mech. Eng.* **195**, 6011–6045 (2006).
14. J.-G. Liu, J. Liu, R. L. Pego, Stable and accurate pressure approximation for unsteady incompressible viscous flow. *J. Comput. Phys.* **229**, 3428–3453 (2010).
15. V. I. Oseledets, On a new way of writing the Navier-Stokes equation. The Hamiltonian formalism. *Russ. Math. Surv.* **44**, 210–211 (1989).
16. P. H. Roberts, A Hamiltonian theory for weakly interacting vortices. *Mathematika* **19**, 169–179 (1972).
17. T. F. Buttke, Velicity methods: Lagrangian numerical methods which preserve the Hamiltonian structure of incompressible fluid flow, in *Vortex Flows and Related Numerical Methods*, J.T. Beale, G.-H. Cottet, S. Huberson, Eds. (Springer, Dordrecht, 1993), vol. 395, pp. 39–57.
18. M. C. Recchioni, G. Russo, Hamilton-based numerical methods for a fluid-membrane interaction in two and three dimensions. *SIAM J. Sci. Comput.* **19**, 861–892 (1998).
19. T. F. Buttke, A. J. Chorin, Turbulence calculations in magnetization variables. *Appl. Numer. Math.* **12**, 47–54 (1993).
20. D. M. Summers, A. J. Chorin, Numerical vorticity creation based on impulse conservation. *Proc. Natl. Acad. Sci. U.S.A.* **93**, 1881–1885 (1996).
21. P. Luchini, Magnetization-vector analogy as a reformulation of the equations of fluid dynamics. *AIAA J.* **29**, 474–477 (1991).
22. A. J. Chorin, *Vorticity and Turbulence* (Springer, New York, 1994).
23. R. Cortez, Impulse-based particle methods for fluid flow. LBL Report-37206, thesis, University of California, Berkeley (1995); https://publications.lbl.gov/islandora/object/ir:101737.
24. R. Cortez, An impulse-based approximation of fluid motion due to boundary forces. *J. Comput. Phys.* **123**, 341–353 (1996).
25. R. Cortez, On the accuracy of impulse methods for fluid flow. *SIAM J. Sci. Comput.* **19**, 1290–1302 (1998).
26. R. Cortez, A vortex/impulse method for immersed boundary motion in high Reynolds number flows. *J. Comput. Phys.* **160**, 385–400 (2000).
27. J. H. Maddocks, R. L. Pego, An unconstrained Hamiltonian formulation for incompressible fluid flow. *Commun. Math. Phys.* **170**, 207–217 (1995).
28. W. E, J.-G. Liu, Gauge method for viscous incompressible flows. *Commun. Math. Sci.* **1**, 317–332 (2003).
29. R. H. Nochetto, J.-H. Pyo, The gauge–Uzawa finite element method. Part I: The Navier–Stokes equations. *SIAM J. Numer. Anal.* **43**, 1043–1068 (2005).
30. J.-H. Pyo, J. Shen, Gauge-Uzawa methods for incompressible flows with variable density. *J. Comput. Phys.* **221**, 181–197 (2007).
31. M. Liu, Y.-X. Ren, H. Zhang, A class of fully second order accurate projection methods for solving the incompressible Navier–Stokes equations. *J. Comput. Phys.* **200**, 325–346 (2004).
32. A. S. Almgren, A. J. Aspden, J. B. Bell, M. L. Minion, On the use of higher-order projection methods for incompressible turbulent flow. *SIAM J. Sci. Comput.* **35**, B25–B42 (2013).
33. S. Y. Kadioglu, R. Klein, M. L. Minion, A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics. *J. Comput. Phys.* **227**, 2012–2043 (2008).
34. M. L. Minion, Semi-implicit projection methods for incompressible flow based on spectral deferred corrections. *Appl. Numer. Math.* **48**, 369–387 (2004).

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

35. D. Mumford, P. W. Michor, On Euler's equation and 'EPDiff.' *J. Geom. Mech.* **5**, 319–344 (2013).
36. J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comput. Phys.* **59**, 308–323 (1985).
37. D. M. Summers, A. J. Chorin, Hybrid vortex/magnet methods for flow over a solid boundary. *ESAIM Proc. Surv.* **1**, 65–76 (1996).
38. D. M. Summers, A representation of bounded viscous flow based on Hodge decomposition of wall impulse. *J. Comput. Phys.* **158**, 28–50 (2000).
39. D. M. Summers, Towards an impulse-based Lagrangian model of boundary layer turbulence. *Physica D* **154**, 287–312 (2001).
40. G. Russo, P. Smereka, Impulse formulation of the Euler equations: General properties and numerical methods. *J. Fluid Mech.* **391**, 189–209 (1999).
41. W. E, J.-G. Liu, Finite difference schemes for incompressible flows in the velocity–impulse density formulation. *J. Comput. Phys.* **130**, 67–76 (1997).
42. A. Donev, A. Nonaka, Y. Sun, T. G. Fai, A. L. Garcia, J. B. Bell, Low Mach number fluctuating hydrodynamics of diffusively mixing fluids. *Commun. Applied Math. Comput. Sci.* **9**, 47–105 (2014).
43. A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, M. L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations. *J. Comput. Phys.* **142**, 1–46 (1998).
44. A. Johansson, M. G. Larson, A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary. *Numerische Mathematik* **123**, 607–628 (2013).
45. R. I. Saye, High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles. *SIAM J. Sci. Comput.* **37**, A993–A1019 (2015).
46. J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences* (Cambridge Univ. Press, Cambridge, 1999).
47. S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces* (Springer, New York, 2003).
48. B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.* **35**, 2440–2463 (1998).
49. D. N. Arnold, F. Brezzi, B. Cockburn, L. D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.* **39**, 1749–1779 (2002).
50. J. S. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications.* (Springer, New York, 2008), vol. 54.
51. M. J. Hancock, J. W. M. Bush, Fluid pipes. *J. Fluid Mech.* **466**, 285–304 (2002).
52. R. I. Saye, J. A. Sethian, The Voronoi Implicit Interface Method for computing multiphase physics. *Proc. Natl. Acad. Sci. U.S.A.* **108**, 19498–19503 (2011).
53. R. I. Saye, J. A. Sethian, Multiscale modeling of membrane rearrangement, drainage, and rupture in evolving foams. *Science* **340**, 720–724 (2013).

Acknowledgments: I thank A. J. Chorin for illuminating discussions regarding the history of gauge methods. **Funding:** This research was supported by a Luis W. Alvarez Postdoctoral Fellowship at the Lawrence Berkeley National Laboratory (LBNL), by the Laboratory Directed Research and Development Program of the LBNL, and by the Applied Mathematics Program of the U.S. Department of Energy (DOE) Office of Advanced Scientific Computing Research under contract DE-AC02-05CH11231. Some computations used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. DOE under contract DE-AC02-05CH11231. **Competing interests:** The author declares that he has no competing interests. **Data and materials availability:** All data used to obtain the conclusions in this paper are presented in the paper and/or the Supplementary Materials. Additional data related to this paper may be requested from the author.

Submitted 19 December 2015
Accepted 20 May 2016
Published 10 June 2016
10.1126/sciadv.1501869

Citation: R. Saye, Interfacial gauge methods for incompressible fluid dynamics. *Sci. Adv.* **2**, e1501869 (2016).

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

ScienceAdvances
AAAS

Interfacial gauge methods for incompressible fluid dynamics
Robert Saye (June 10, 2016)
Sci Adv 2016, 2:.
doi: 10.1126/sciadv.1501869

This article is publisher under a Creative Commons license. The specific license under which this article is published is noted on the first page.

For articles published under CC BY licenses, you may freely distribute, adapt, or reuse the article, including for commercial purposes, provided you give proper attribution.

For articles published under CC BY-NC licenses, you may distribute, adapt, or reuse the article for non-commerical purposes. Commercial use requires prior permission from the American Association for the Advancement of Science (AAAS). You may request permission by clicking here.

The following resources related to this article are available online at http://advances.sciencemag.org. (This information is current as of June 10, 2016):

Updated information and services, including high-resolution figures, can be found in the online version of this article at:
http://advances.sciencemag.org/content/2/6/e1501869.full

Supporting Online Material can be found at:
http://advances.sciencemag.org/content/suppl/2016/06/07/2.6.e1501869.DC1

This article **cites 46 articles,**3 of which you can be accessed free:
http://advances.sciencemag.org/content/2/6/e1501869#BIBL

Downloaded from http://advances.sciencemag.org/ on June 10, 2016

Science Advances (ISSN 2375-2548) publishes new articles weekly. The journal is published by the American Association for the Advancement of Science (AAAS), 1200 New York Avenue NW, Washington, DC 20005. Copyright is held by the Authors unless stated otherwise. AAAS is the exclusive licensee. The title Science Advances is a registered trademark of AAAS