Computing stationary free-surface shapes in microfluidics

Cite as: Phys. Fluids **18**, 103303 (2006); https://doi.org/10.1063/1.2361291
Submitted: 04 February 2006 • Accepted: 08 September 2006 • Published Online: 16 October 2006

Michael Schindler, Peter Talkner and Peter Hänggi

Computing stationary free-surface shapes in microfluidics

Michael Schindler, Peter Talkner, and Peter Hänggi
Institut für Physik, Universität Augsburg, Universitätsstraße 1, 86159 Augsburg, Germany

(Received 4 February 2006; accepted 8 September 2006; published online 16 October 2006)

A finite-element algorithm for computing free-surface flows driven by arbitrary body forces is presented. The algorithm is primarily designed for the microfluidic parameter range where (i) the Reynolds number is small and (ii) force-driven pressure and flow fields compete with the surface tension for the shape of a stationary free surface. The free surface shape is represented by the boundaries of finite elements that move according to the stress applied by the adjacent fluid. Additionally, the surface tends to minimize its free energy and by that adapts its curvature to balance the normal stress at the surface. The numerical approach consists of the iteration of two alternating steps: The solution of a fluidic problem in a prescribed domain with slip boundary conditions at the free surface and a consecutive update of the domain driven by the previously determined pressure and velocity fields. For a Stokes problem the first step is linear, whereas the second step involves the nonlinear free-surface boundary condition. This algorithm is justified both by physical and mathematical arguments. It is tested in two dimensions for two cases that can be solved analytically. The magnitude of the errors is discussed in dependence on the approximation order of the finite elements and on a step-width parameter of the algorithm. Moreover, the algorithm is shown to be robust in the sense that convergence is reached also from initial forms that strongly deviate from the final shape. The presented algorithm does not require a remeshing of the used grid at the boundary. This advantage is achieved by a built-in mechanism that causes a smooth change from the behavior of a free surface to that of a rubber blanket if the boundary mesh becomes irregular. As a side effect, the element sides building up the free surface in two dimensions all approach equal lengths. The presented variational derivation of the boundary condition corroborates the numerical finding that a second-order approximation of the velocity also necessitates a second-order approximation for the free surface discretization. © *2006 American Institute of Physics*. [DOI: 10.1063/1.2361291]

I. INTRODUCTION

In the past decade, the development of so-called "labs-on-a-chip"[1,2] has led to an increased interest in microfluidics,[3–5] i.e., in the field of hydrodynamics with characteristic length scales of less than a millimeter. These flows are characterized by small *Reynolds numbers* and consequently governed by the Stokes equations. In the case of prescribed fluid domains with no-slip boundary conditions, standard numerical methods exist for computing their solutions.[6,7]

Recently, various experimental techniques[8,9] have been developed to induce and control flows in fluids that sit on a substrate without being confined by lateral and covering walls.[10,11] In the experiments, the fluid is kept together by its surface tensions both at the substrate and at the fluid-air interface. The stationary form that is assumed by the fluid-air interface is not given *a priori*. It results from an interplay of the internal streaming pattern, the internal pressure distribution, and the surface tension. On the other hand, the form of the interface acts back on the flow. This mutual interaction of form and flow renders free boundary value problems fascinating but difficult. The relative importance of viscous flow and pressure, each compared to the influence of the surface tension, can be quantified by two dimensionless numbers, the *capillary number* and a generalized *Bond number*, respectively.

In the present work, we consider a small water droplet (around 50 nl or less). The droplet sits on a flat substrate and is mechanically agitated by a body force.[12] Inside the droplet, this body force then causes stationary pressure and flow fields that can lead to a significant deformation of the free surface. Sufficiently strong body forces may lead to the motion of the entire droplet, but this situation will not be considered here. The values of the Reynolds number, the capillary number, and the Bond number are assumed to range from zero up to unity. This corresponds to experimentally relevant situations.[10,11,13]

Several numerical approaches for determining free surface shapes have been proposed in the past. The suitability of the approaches depends on the size of the system, typical velocities, and the material properties, as well as on the resulting deformation of the fluid domain. They can roughly be classified into two groups:[14] Either a fixed grid and a function describing the position of the free surface is used, or the computational mesh is moved together with the fluid domain, yielding a sharp surface representation by element boundaries.

An established method of the first kind is the *continuum method* proposed by Brackbill *et al.*[15] They circumvented the discretization of the normal-stress boundary condition by introducing a body-force density that is concentrated near the free surface. This force density accounts for the effect of surface tension. We have tested this method, which is imple-

mented in the commercially available fluid-dynamics program *FLUENT* using a *volume-of-fluid* discretization. For a macroscopic system, this method worked fine. The method, however, fails if the system is scaled down to the microfluidic parameter regime. In a simple test example, we found that approximation errors of the free-surface boundary condition contributed to the force balance in the Navier-Stokes equations and were amplified in an uncontrolled manner. This typically gave rise to a spurious velocity field. It even occurred when we started the iteration with the known solution. Problems with this method have also been reported by Renardy and Renardy[16] and by Popinet and Zaleski.[17] Lafaurie *et al.*[18] find the spurious velocities to be of the order surface-tension/viscosity, which is the dominant velocity scale for microfluidic systems. Thus, the existing continuum method appears to be inappropriate for the microfluidic parameter regime.

Another approach of the first kind has recently been proposed by Smolianski.[19] He uses finite elements and a level-set description for the free surface and calculates curvatures by derivatives of the distance function. He too encounters spurious velocity fields proportional to the ratio surface-tension/viscosity.

Methods of the second kind, representing the free surface by a sharp interface, are expected to work better in the microfluidic parameter regime. Algorithms in this class are often referred to as "moving mesh" or "ALE" methods and generally require more involved techniques, keeping the computational mesh feasible and not too distorted.

A technique of the second kind that has successfully been employed for tension-dominated free-surface problems is the boundary-element method.[20,21] Pinch-off effects and droplet formations can be described by this method.[14] The dimensionality of the equations is reduced to the dimensionality of the surface, which provides the basis for an efficient implementation. Unfortunately, this reduction can only be performed for Stokes equations with conservative body forces, which can be absorbed into the pressure term. In the present investigation, we allow for nonconservative body forces that are of particular experimental relevance.[9,10]

Pioneering works for the finite-element implementation of the full free-surface problem were published by Scriven and co-workers.[22,23] They used spines to parametrize the movements of the computational mesh in coating flow and implemented Newton's method for a Galerkin approximation scheme. This work was later continued under the designation "total linearization method" by Cuvelier and co-workers.[24,25] Their description requires a height function for the free-surface position, which makes it necessary to use well-adapted coordinate systems. Whether a free surface will overhang must be known in advance.

In the present paper, we extend the works of Scriven and Cuvelier to arbitrary surface geometries. In our description, the parametrization of the free surface is given directly by the finite-element boundary parametrization. Thus, neither spines nor a height function are needed. To properly account for intrinsic curvatures of the free surface, all equations are formulated in a fully covariant form that allows for all differential-geometric properties of the surface. An excellent reference for this formulation can be found in the works of Aris[26] and Scriven,[27] where the fluidic flow inside a curved free surface is described.

Recently, algorithms have been published that describe time-dependent free-surface flows, even in three dimensions.[28–30] In these works, the free surface is moved mainly due to the *kinematic* boundary condition, i.e., it is advected passively. Concerning convergence, there has been a controversy if the kinematic or rather the normal stress boundary condition should be used to move the free surface. This issue was resolved by Silliman and Scriven, who state that for capillary numbers below unity, the normal stress iteration converges well while a kinematic iteration eventually fails.[22] In addition, when the kinematic boundary condition is used for updating the free surface, the balance of normal stress that carries the effects of surface tension is not strictly imposed. It is used when implementing the weak form of the Navier-Stokes equations: In this context, an integration by parts yields an integral of the normal stress over the free surface, which is then replaced by the corresponding surface integral of the tension forces. Similar techniques are commonly used for problems with outflow boundary conditions or for Poisson's equation with Neumann boundary conditions. The correctness of the technique has been justified for the outflow problem by Renardy.[31] However, it is not evident whether it also works in the case in which the surface-tension terms dominate the whole problem. The question remains in which sense the boundary condition is satisfied. Therefore, we found it necessary in our examples to visualize the terms involved in the free-surface boundary condition, thus proving that they are correctly balanced.

An important result of our variational description of the tension terms is an improvement of the Newton algorithm controlling possible mesh distortions at the free surface. Many algorithms implementing the weak form of the capillary boundary condition encounter intrinsic instabilities of the boundary mesh when significant changes of the free surface take place. For the program *surface evolver*,[32] this manifests itself in shrinking and growing surface facets. Similar effects have been observed by Brinkmann[33] and Bänsch.[28] Our formulation of the capillary free surface is such that the free surface smoothly changes to the behavior of a rubber blanket when the boundary mesh becomes distorted. This leads to an automatic regularization of the mesh without the need of explicit remeshing or smoothing.

In Sec. II, the mathematical formulation of the problem is presented in terms of differential equations, together with the boundary conditions and the relevant parameter regime. In Sec. III, we then reestablish the bulk equations and their boundary conditions by variational techniques. For the free surface, we introduce a differential-geometric notation that allows us to write the boundary condition in a weak form. Up to this point, a continuous description is used. Section IV introduces the discretization of the problem by the computational mesh. The formulation of the tension forces as the concurrent minimization of the free-surface area of single finite elements is a necessary requirement for the mentioned automatic regularization mechanism. Section V provides a short summary of the whole algorithm. In Sec. VI, we

present examples that show the accuracy of the algorithm and two further examples for different values of the capillary and Bond numbers. Mathematical and algorithmic details are deferred to Appendixes A–D.

II. STATEMENT OF THE PROBLEM

Throughout the paper, we use tensor notation in arbitrary curvilinear coordinates. This will considerably simplify the differential geometric notation in the following sections. For the formulation of the full Navier-Stokes equations in curvilinear coordinates, we refer to Aris.[26] Repeated indices in co- and contravariant positions are summed over. Indices preceded by a comma denote covariant derivatives, and g_{ij} is the metric tensor of the underlying coordinate system.

A. The basic equations

We study incompressible and stationary flows, which are characterized by a small Reynolds number $\mathrm{Re}=\rho\bar{x}\bar{v}/\eta$. Here, ρ is the density of the fluid, η its viscosity, and $\bar{x}$ and $\bar{v}$ denote typical magnitudes of length and velocity. Under these conditions, the pressure field p and the velocity field with components v^i satisfy the Stokes equations,[34]

$$v^i_{,i}=0, \tag{1}$$

$$0=\sigma^{ij}_{,j}+f^i \quad \text{where} \tag{2}$$

$$\sigma_{ij}=-pg_{ij}+2\eta e_{ij}, \quad \text{and } e_{ij}=(v_{i,j}+v_{j,i})/2, \tag{3}$$

and where σ_{ij} is the fluidic stress tensor, e_{ij} the rate-of-strain tensor, and f^i an external body force causing nontrivial streaming and pressure patterns within a domain V. It can be split into a conservative part $f^i_{(c)}$, which can be displayed as the gradient of a potential and a nonconservative part $f^i_{(nc)}$ with vanishing divergence. The domain V may be bounded by rigid walls and by free surfaces, as, e.g., a droplet sitting on a substrate. Equations (1) and (2) then are subject to boundary conditions at the different parts of the boundary ∂V: First, the flow has to meet the *kinematic* boundary condition, requiring that the normal projection of a stationary velocity field vanishes at the boundary, i.e.,

$$v_iN^i=0. \tag{4}$$

At immobile sticky walls, we use the *no-slip* boundary condition, according to which the velocity vanishes also in the tangential directions of the boundary, implying

$$v_iT^i_\alpha=0 \quad \text{at the walls}. \tag{5}$$

Here, T^i_α denotes the ith component of the tangential vector $\mathbf{T}_\alpha$ ($\alpha=1,2$ for a two-dimensional surface). The remaining boundary is a free surface that dynamically adjusts its position such that the stress balance holds,

$$\sigma^{ij}N_j=\gamma\kappa N^i \quad \text{on free surfaces}, \tag{6}$$

with the surface tension γ and the curvature κ. Note that we have omitted a term proportional to the gradient of the surface tension and thus exclude Marangoni effects. This simplifies the following calculations but does not present a principal restriction of our description.

Equations (1)–(6) have been simplified by assuming that at the free surface the fluid always stays in contact with a medium of low viscosity such as in the case of a water-air interface. Therefore, the viscous stress contribution of the air does not show up in the balance equation (6). We further assume that the ambient pressure p_0 is constant. Since the pressure is determined by the Stokes equations only up to a constant, we can split it into a part p_1 with vanishing average and use the ambient pressure p_0 as an offset parameter that enters only in the normal stress balance (6),

$$p(\mathbf{x})=p_0+p_1(\mathbf{x}) \quad \text{with} \int_V p_1 dV=0. \tag{7}$$

B. The parameter regime

By transforming both the bulk equation (2) and the free boundary condition (6) into dimensionless form employing viscosity scaling, one observes that the system may be characterized by two relevant ratios of forces, given by the dimensionless numbers

$$\mathrm{Bo}=\frac{\bar{f}_{(c)}\bar{x}^2}{\gamma} \quad \text{and } \mathrm{Ca}=\frac{\eta\bar{v}}{\gamma}. \tag{8}$$

Here, $\bar{x}$, $\bar{v}$, and $\bar{f}_{(c)}$ denote typical magnitudes of length, velocity, and the conservative part of the force density, respectively. Bo is a generalization of the Bond number, which is usually defined in terms of gravitational forces only. The capillary number Ca measures the viscous contribution to the surface deformation. In a system with static boundaries and vanishing Reynolds number, we can express the velocity scale by the typical magnitude $\bar{f}_{(nc)}$ of the nonconservative part of the driving force, namely $\bar{v}=\bar{x}^2\bar{f}_{(nc)}/\eta$. This yields an alternative definition of the capillary number similar to that of the Bond number,

$$\mathrm{Ca}=\frac{\bar{f}_{(nc)}\bar{x}^2}{\gamma}. \tag{9}$$

These two numbers reflect the different effects of the conservative and nonconservative parts of the driving. In this sense, Ca also provides a measure for the spatial changes of the velocity field. For small Ca, the flow is slow and changes smoothly, whereas for large Ca it may exhibit drastic gradients.

We propose a numerical scheme for the parameter regime where both Ca and Bo are of order unity or less. Thus, pressure gradients and viscous forces can deform the free surface significantly. The surface tension is large enough, however, to keep the whole fluid domain together; a pinch-off is excluded. The viscosity renders the velocity field smooth over the whole fluid domain and prevents the existence of boundary layers. For the considered case of stationary flows on resting substrates in stationary domains, the contact lines always stay pinned. A rolling or slipping droplet would raise additional challenges regarding the stress near the contact line that are beyond the scope of this paper.

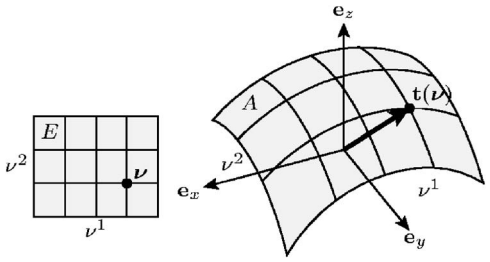

FIG. 1. A sketch of the coordinate system on a two-dimensional surface A, embedded into the three-dimensional space. The surface coordinates $\boldsymbol{\nu}$ are mapped from the reference domain E (left) onto the surface A (right) via the parametrization vector $\mathbf{t}(\boldsymbol{\nu})$.

III. CONTINUOUS DESCRIPTION OF THE PROBLEM

In order to formulate the numerical treatment of the free-surface boundary condition, we first explore the physical origins of the active forces. We will then express each of them by the first variation of a functional. A modified Newton's method requires us to calculate the second variation.

A. Physical aspects of the free-surface boundary condition: First variations

The surface tension term $\gamma\kappa N_i$ in the boundary condition (6) arises from the fact that an interface between two different phases "costs" free energy.[34] To find the optimal configuration, the surface is continually probing positions in its vicinity in order to minimize its free energy. For the case of an applied conservative force $f_i=-\Phi_{,i}$ the system is static ($v^i=0$), and the free-surface boundary condition is equivalent to a minimization of a free-energy expression. This calculation is performed in Appendix A.

Due to its thermodynamic origin, the surface tension term results from a first variation of a functional. This carries over to the dynamic case ($v^i\neq 0$) in which the boundary condition (6) must hold at any instant of time. The contribution of the free surface A to the free energy is given by the integral of the surface tension γ over A,

$$F=\int_A \gamma dA, \tag{10}$$

where dA denotes the infinitesimal surface area. Any smooth surface in a D-dimensional space may be parametrized by $D-1$ surface coordinates ν^α ($\alpha=1,\ldots,D-1$), which determine the coordinates $t^i(\nu^\alpha)$ of points in D-dimensional space on the surface. Both surface and space coordinates are illustrated in Fig. 1. In our numerical studies, we restrict ourselves to $D=2$. The general framework, however, remains valid also for $D=3$. The surface coordinates ν^α are taken from the parameter set $E\subset\Re^{D-1}$. With $\boldsymbol{\nu}$ running through E, the whole free surface A is covered,

$$t^iE:\rightarrow\Re:\boldsymbol{\nu}\mapsto t^i(\boldsymbol{\nu}), \tag{11}$$

$$A=\{\mathbf{e}_{(i)}t^i(\boldsymbol{\nu})|\boldsymbol{\nu}\in E\}. \tag{12}$$

Here, $\mathbf{e}_{(i)}$ is the ith basis vector in space. Throughout the paper, we will use Greek letters for surface indices and Latin ones for space indices. The connection between surface and space coordinates is conveniently described by the surface derivatives of the parametrization functions (cf. Ref. 26, p. 215), i.e.,

$$t^i_{,\alpha}=\frac{\partial t^i}{\partial\nu^\alpha}. \tag{13}$$

The D-dimensional contravariant space-vector, $t^i_{,\alpha}$, represents the components of the αth tangential vector $\mathbf{T}_\alpha$ of the surface. At the same time, $t^i_{,\alpha}$ is a covariant surface-vector. The scalar products of these tangential vectors determine the components of the surface metric tensor $a_{\alpha\beta}$,

$$a_{\alpha\beta}=g_{ij}t^i_{,\alpha}t^j_{,\beta}\quad\text{and}\quad a=\det(a_{\alpha\beta}). \tag{14}$$

The normal vector follows as the normalized cross product of two tangential vectors,

$$N_i=\tfrac{1}{2}\varepsilon_{ijk}\varepsilon^{\alpha\beta}t^j_{,\alpha}t^k_{,\beta}, \tag{15}$$

and the curvature κ is given as the trace of the tensor $b_{\alpha\beta}$ of the second fundamental form of the surface,

$$\kappa=a^{\alpha\beta}b_{\alpha\beta}\quad\text{with} \tag{16}$$

$$b_{\alpha\beta}=t^i_{,\alpha\beta}N_i. \tag{17}$$

Using the parametrization (11) of the free surface, we demonstrate in Appendix B that the change of the free-energy contribution F with respect to a variation of the surface positions t^i is given by

$$\delta F[\delta\mathbf{t}]=\int_A \gamma t^j_{,\beta}g_{ij}a^{\alpha\beta}\delta t^i_{,\alpha}dA. \tag{18}$$

By an integration by parts, this expression can be cast into a form containing the curvature term of the free-surface boundary condition (6), i.e.,

$$\delta F[\delta\mathbf{t}]=-\int_A \gamma\kappa N_i\delta t^i dA \tag{19}$$

(see Appendix C also for the case of varying surface tension). In this way, the curvature term κ in Eq. (19) that contains second spatial derivatives is replaced by a product of two terms, each containing a first derivative in Eq. (18). Especially for numerical applications, it is much more favorable to work only with first derivatives. This trick has been used in the literature in different contexts.[21,32,35,36,28] Seen from a physical perspective, version (18) of the equation is the more natural one. Here, one directly deduces that forces pulling along the tangential direction attempt to minimize the facet area of the boundary. On the basis of single finite elements, this perspective will be used below for stabilizing the computational mesh.

The left-hand side of Eq. (6), namely $\sigma_{ij}N^j$, is the normal fluidic stress at the boundary. We now recapitulate how this term can be understood as the result of a variational principle. In a stationary system with rigid immobile boundaries, the variational principle, which is attributed to Helmholtz and Korteweg,[37–40] states that the Stokes equations yield those velocity and pressure fields that represent a stationary point of the functional

$$P = \int_V \mathcal{P} dV = \int_V [(-pg^{ij} + \eta e^{ij}) v_{i,j} - f^i v_i] dV. \tag{20}$$

Vice versa, from the vanishing first variation of P,

$$\delta P[\delta p] = \int_V \frac{\partial \mathcal{P}}{\partial p} \delta p dV = -\int_V v^i_{,i} \delta p dV, \tag{21}$$

$$\delta P[\delta \mathbf{v}] = \int_V \left(\frac{\partial \mathcal{P}}{\partial v_i} \delta v_i + \frac{\partial \mathcal{P}}{\partial v_{i,j}} \delta v_{i,j} \right) dV \tag{22}$$

$$= \int_V (-f^i \delta v_i + \sigma^{ij} \delta v_{i,j}) dV \tag{23}$$

$$= -\int_V (f^i + \sigma^{ij}_{,j}) \delta v_i dV + \oint_{\partial V} \sigma^{ij} N_j \delta v_i dA, \tag{24}$$

the Stokes equations follow by setting the bulk contributions to zero. The boundary integral in Eq. (24) provides the fluidic stress contribution to the free-surface boundary condition.

At this point, we see that the two terms in the stress balance Eq. (6) have different physical origins. The surface tension is of thermodynamic (or rather of "thermostatic") nature while the fluidic stress stems from dynamic considerations. The first minimizes a free energy, while the second minimizes a power. Formally, this difference is expressed by the distinct variations δv^i and δt^i in the expressions $\sigma_{ij} N^j \delta v^i$ in Eq. (24) and $\kappa \gamma N_i \delta t^i$ in Eq. (19). Already for dimensionality reasons they cannot be equal, nor can the functionals F and P be directly combined into a single variational principle.

From an algorithmic point of view, a choice has to be made whether the free-surface boundary condition is approximated by means of either δv^i or δt^i as test functions. In the Galerkin implementation of the problem, we will use the ansatz functions of the finite elements as test functions. In order to acquire a consistent numerical algorithm, in this case both the velocity and the geometry parametrization must be approximated by ansatz functions of the very same order. This is the first central result of the present work.

It was stated by Bänsch[28] (p. 42, cf. also citations 49 and 50 therein) that a second-order approximation of the surface parametrization yields a "good discrete curvature," whereas a first-order one does not. The same can be seen below in Fig. 2. We are now able to substantiate Bänsch's numerical observation with the underlying physical mechanism. The argument is similar to that for the celebrated Ladyzhenskaya-Babuska-Brezzi requirement that velocity gradients have to be approximated by the same order as the pressure. From a physical perspective, this is not astonishing, because both are components of the stress tensor.

B. Splitting the problem into two numerical systems

For free boundaries, a twofold problem must be solved: (i) The unknown fluid domain V has to be determined and (ii) the Stokes equations (1) and (2) have to be solved in V, with the boundary conditions (4)–(6). The latter depend on the shape of V via the normal vector at the boundary. Both parts of this problem cannot be processed independently.

In principle, two options exist to approach this combined problem. The first one is to implement a single numerical system for both the flow variables p and v^i together with the geometric variables t^i. We will not follow this direction but rather consecutively solve two smaller systems, one for the flow variables, depending on the current domain V, and a second one for the parametrization of the boundary. We have chosen this approach because the problem is linear in the flow variables but highly nonlinear in the geometric variables t^i. The nonlinearity is caused by the appearance of the inverse surface metric $a^{\alpha\beta}$ and of the Jacobi determinant contained in dA in Eq. (18). Thus, solving the Stokes equations in the *fluidic system*, as we call it, will be a standard problem, while the nonlinear search for the correct boundary shape will be done in the *geometric system*. Both systems are solved consecutively:

(1) Choose an initial domain V.
(2) Until convergence, repeat the following steps:
   (a) Solve the fluidic system within the domain V.
   (b) Solve the geometric system using fixed values for the pressure and velocity variables. This results in an updated domain V.

In three-dimensional space, Eqs. (4)–(6) pose four boundary conditions. There are one too many for the linear fluidic system to be fully determined. One boundary condition is thus used for updating the parametrization of the free surface.[24] The main challenge is the proper assignment of specific boundary conditions for the two systems in order to make them solvable, uniquely determined, and robust. It is clear that the no-slip boundary condition (5) at sticky walls applies only to the fluidic system. The free-surface boundary condition needs further consideration.

Here, again, a physical argument helps to choose the proper boundary condition. Either the stress by the fluid or its velocity is employed for moving the free surface. Accordingly, either the normal stress balance (6) or the kinematic boundary condition (4) leads to an update of the surface in the geometric system (see the discussion by Saito and Scriven[22] and our remarks in the Introduction). We choose our approach according to the following principle: The fluidic system should be well defined as a stationary system even if the boundary is fixed and is not part of the problem. The kinematic boundary condition must then apply to the velocity field to prevent the fluid from passing through the free surface. Thus, the kinematic boundary condition cannot be used for updating the surface.

Before the correct boundary shape is reached, the viscous stress of the flow may force the free surface into an arbitrary direction. By its very nature, however, the tension force always stays normal on the free surface. Only normal forces can be compensated by a free surface. As a consequence, the tangential projection of the normal stress has to vanish.[41] In the proposed scheme with two separated sys-

tems, it is the fluidic system that must ensure the tangential components of the free boundary condition (6), i.e.,

$$(v_{i,j}+v_{j,i})N^i t^j_{,\alpha}=0 \quad \text{for all } \alpha. \tag{25}$$

As in Eq. (6), the surface tension γ has been set constant. For the velocity variables, this constitutes a perfect-slip boundary condition, which resembles a Neumann boundary condition. We thus find the fluidic system to be fully determined and physically well defined even for fixed boundaries by the conditions (4), (5), and (25).

The geometric system is then responsible for the remaining normal component of the stress balance (6),

$$-p+\eta(v_{i,j}+v_{j,i})N^iN^j=\gamma\kappa, \tag{26}$$

which is used as the update equation for the boundary. A given trial position of the free surface is updated if Eq. (26) is violated.

C. Second variation with respect to the surface parametrization

In a first implementation we used a direct and explicit update algorithm moving the boundary into a normal direction with a step width that is determined by a parameter τ and the residual of Eq. (26). The discretization of this update can be found below in Eq. (54). Depending on the value of τ, this method exhibited strong instabilities. Although advanced techniques for determining an apt value for τ seem to exist (cf. the program *Surface Evolver* by Brakke[32]), we prefer a modified Newton-Raphson iterative method. This has the advantages of faster convergence and less strong dependence on τ. It requires an additional variation of the surface free energy for the assembly of the geometric system. Using the same calculus as in Appendix B, we find the second variation of the free-energy contribution F of a one-dimensional free surface,

$$\delta^2F[\delta\mathbf{t},\delta\mathbf{t}]=\delta\left(\int_A \gamma g_{ij}a^{\alpha\beta}t^j_{,\beta}\delta t^i_{,\alpha}dA\right) \tag{27}$$

$$=\int_A \gamma\delta t^i_{,\alpha}\delta t^j_{,\beta}(N_iN_ja^{\alpha\beta}+t^\alpha_i t^\beta_j-t^\beta_i t^\alpha_j)dA, \tag{28}$$

where the contravariant surface indices are introduced in Appendix B. Note that the precise form of the second variation influences only the convergence of the iteration, but not the solution, which solely depends on the first variation.

Also the flow velocity, and by this the viscous stress, depends on the shape of the surface. The formulation of the modified Newton method requires also the change of the fluidic stress integral due to changes of the free boundary,

$$\delta\left(\int_A \sigma_{ij}N^j\delta t^i dA\right)[\delta\mathbf{t}]=\int_A \delta t^i\sigma_{ij}\delta N^j[\delta\mathbf{t}]dA + \int_E \delta t^i\sigma_{ij}N^j\delta\sqrt{a}[\delta\mathbf{t}]d\boldsymbol{\nu} + \int_A \delta t^i\delta\sigma_{ij}[\delta\mathbf{t}]N^j dA. \tag{29}$$

The first two integrals on the right-hand side contain the changes of the normal vector (15) and the infinitesimal surface area $dA=\sqrt{a}d\boldsymbol{\nu}$ due to changes of the boundary shape. Both can be calculated along the lines of Appendix B. The third integral expresses the change of the fluidic stress σ_{ij} on the boundary caused by changes of its position. The shape changes act on the velocity field via the boundary conditions (4) and (25) of the Stokes equations. Unfortunately, this indirect response of the viscous stress tensor to the changes of shape cannot be expressed explicitly. We therefore assume that this dependence is weak. The boundary conditions affect only the velocity field, not the pressure, which is determined only by the applied external force. A Taylor expansion of the pressure field around the free surface yields for the change of the stress tensor,

$$\delta\sigma_{ij}[\delta\mathbf{t}]\approx -g_{ij}p_{,k}\delta t^k. \tag{30}$$

This approximation assumes that the fluidic and the geometric systems are decoupled to the extent that the viscous part of the stress tensor in the vicinity of the boundary is not affected by small boundary changes. Note that this assumption influences only the rate of convergence but not the final result. The difficulty to describe the mutual dependence of velocity field and surface geometry is not a consequence of splitting the problem into two separate systems, but a general problem that applies equally to the combined approach. Altogether, the right-hand side of Eq. (29) becomes approximately

$$\int_A \{(\delta t^i\sigma_{ij}N^j)(\delta t^k_{,\alpha}g_{kl}a^{\alpha\beta}t^l_{,\beta})-(\delta t^i\sigma_{ij}t^j_{,\alpha})a^{\alpha\beta}(\delta t^k_{,\beta}N_k) - \delta t^iN_ip_{,k}\delta t^k\}dA. \tag{31}$$

IV. DISCRETIZATION OF THE PROBLEM

We propose a discretization of the above equations by means of a Galerkin approximation scheme that is known to work well for minimization problems. As variables we introduce the velocity components u and v in the x and y direction, respectively, the pressure p, and additional variables r and s for the coordinates of the boundary parametrization vector $\mathbf{t}$. The continuous fields are discretized using ansatz functions, weighted with the corresponding degrees of freedom (DoF),

$$u(\mathbf{x})=\sum_d u_d\phi_d(\mathbf{x}), \quad v(\mathbf{x})=\sum_d v_d\phi_d(\mathbf{x}), \tag{32}$$

$$p(\mathbf{x})=\sum_d p_d\psi_d(\mathbf{x}), \tag{33}$$

$$r(\mathbf{x})=\sum_d r_d\chi_d(\mathbf{x}), \quad s(\mathbf{x})=\sum_d s_d\chi_d(\mathbf{x}), \tag{34}$$

where the sum runs over all DoFs. The fluid velocity components u,v are approximated by the second-order finite elements (FEs) ϕ and the pressure variable p by first-order FEs ψ. For the position variables r,s we predominantly used second-order FEs, but for accuracy and other testing reasons

we also tried first-order FEs. We denote the position FEs with χ. All FEs are of the Lagrange family,[6] having ansatz functions that are 1 at exactly one node of the mesh and 0 at all others. The DoFs are then equal to the function values at the nodes. This property is most convenient for the position variables (r_d, s_d) that coincide with the coordinates of the node d.

A. The fluidic system

The fluidic system is formulated in a standard way. Equation (2) is tested with the second-order FEs ϕ, while the continuity equation (1) is tested with the first-order FEs ψ. The system consists of the following linear equations for the DoFs, which are collected in the vectors $\vec{u}, \vec{v}, \vec{p}$ with components u_d, v_d, p_d, respectively,

$$\begin{pmatrix} K_{uu} & 0 & K_{up} \\ 0 & K_{vv} & K_{vp} \\ K_{pu} & K_{pv} & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{v} \\ \vec{p} \end{pmatrix} = \begin{pmatrix} L_u \\ L_v \\ 0 \end{pmatrix} \tag{35}$$

with the entry matrices K and entry vectors L given by

$$[K_{uu}]_{de} = \eta \int_V \nabla \phi_d \cdot \nabla \phi_e dV - \eta \oint_{\partial V} \phi_d \mathbf{N} \cdot \nabla \phi_e dA, \tag{36}$$

$$[K_{vv}]_{de} = \eta \int_V \nabla \phi_d \cdot \nabla \phi_e dV - \eta \oint_{\partial V} \phi_d \mathbf{N} \cdot \nabla \phi_e dA, \tag{37}$$

$$[K_{up}]_{de} = -\int_V (\partial_x \phi_d) \psi_e dV + \oint_{\partial V} \phi_d \psi_e N_x dA, \tag{38}$$

$$[K_{vp}]_{de} = -\int_V (\partial_y \phi_d) \psi_e dV + \oint_{\partial V} \phi_d \psi_e N_y dA, \tag{39}$$

$$[K_{pu}]_{de} = -\int_V \psi_d \partial_x \phi_e dV, \tag{40}$$

$$[K_{pv}]_{de} = -\int_V \psi_d \partial_y \phi_e dV, \tag{41}$$

$$[L_u]_d = \int_V \phi_d f_x dV, \tag{42}$$

$$[L_v]_d = \int_V \phi_d f_y dV. \tag{43}$$

All integrals are assembled in a loop over the elements and the sides of the mesh, using a fifth-order Gaussian quadrature rule. The fluidic system could likewise implement the stationary Navier-Stokes equations with a small Reynolds number; we choose the Stokes equation for simplicity reasons here.

The boundary conditions are imposed by a constraints technique for the matrix and for the right-hand side in Eq. (35). A constrained DoF u_d is expressed by an inhomogeneity plus a weighted sum of other DoFs,

$$u_d = w_d + \sum_{e \neq d} w_{de} u_e, \tag{44}$$

which represent the boundary condition in question. The DoF u_d is then completely eliminated from the linear system (35). By such constraint equations we write the kinematic boundary condition (4) as

$$0 = \sum_e (u_e N_x + v_e N_y) \phi_e, \tag{45}$$

the no-slip condition at the walls

$$0 = \sum_e (u_e T_x + v_e T_y) \phi_e, \tag{46}$$

and a weak version of the tangential projection of the free-surface boundary condition (25),

$$0 = \sum_e \begin{pmatrix} u_e \\ v_e \end{pmatrix} \cdot \begin{pmatrix} 2T_x N_x & T_x N_y + T_y N_x \\ T_x N_y + T_y N_x & 2T_y N_y \end{pmatrix} \times \int_{\partial V} \phi_d \begin{pmatrix} \partial_x \phi_e \\ \partial_y \phi_e \end{pmatrix} dA. \tag{47}$$

The constraint equations differ only in the values of w_{de}. The inhomogeneity w_d is zero in all three equations. Nonzero inhomogeneities would result in the presence of gradients of the surface tension or in the case of tangentially moving rigid boundaries.

For the boundary condition (47), which is equivalent to a perfect-slip condition, an improper choice of the normal direction can cause spurious contributions to the velocity field (see Behr,[42] Walkley *et al.*,[43] and our remarks stated in the Introduction). In the presence of conservative forces only, we did not find such spurious flows in our results.

The fact that the free boundary condition (47) cross-links all DoFs residing at boundary nodes presents a serious problem. Each of the DoFs is in principle linked to all its neighbors on the boundary. This leads to a nearly filled system matrix that is unfavorable regarding memory capacity and computing time. We found that an iterative method can overcome this problem. Instead of cross-linking a boundary DoF with all its neighbors, for some of them we take their old values, as is detailed in Appendix D. After some iterations, the full boundary condition (47) is established. The drawback of this scheme is that the constraint equations have to be reassembled after every solution step of the fluidic system.

B. The geometric system

In the geometric system, a modified Newton method is employed to perform the nonlinear search for the correct boundary position. This scheme corresponds to a minimization of the free energy F, while taking the fluidic stress into account. The boundary update equation can be written in a discretized form as

$$0 = [L_r]_d(\vec{r},\vec{s},\vec{u},\vec{v},\vec{p}) := \frac{\partial F}{\partial r_d} + \int_A \chi_d \sigma^{xj} N_j dA,$$
$$0 = [L_s]_d(\vec{r},\vec{s},\vec{u},\vec{v},\vec{p}) := \frac{\partial F}{\partial s_d} + \int_A \chi_d \sigma^{yj} N_j dA, \tag{48}$$

where d enumerates the geometric variables, and L and F are functions of the arrays $\vec{r},\vec{s}$, etc. containing the DoFs. For the original Newton-Raphson method, the geometric system repeatedly has to solve the linear system of equations[44]

$$\begin{pmatrix} \partial[L_r]_d/\partial r_e & \partial[L_r]_d/\partial s_e \\ \partial[L_s]_d/\partial r_e & \partial[L_s]_d/\partial s_e \end{pmatrix}^{(\mathrm{old})} \begin{pmatrix} \vec{r}_e^{(\mathrm{new})} - \vec{r}_e^{(\mathrm{old})} \\ \vec{s}_e^{(\mathrm{new})} - \vec{s}_e^{(\mathrm{old})} \end{pmatrix} = -\tau \begin{pmatrix} [L_r]_d \\ [L_s]_d \end{pmatrix}^{(\mathrm{old})}, \tag{49}$$

where $\tau \in [0,1]$ is a step-size parameter. In the original Newton method, the block matrices on the left-hand side of Eq. (49) consist of the second variations given by Eq. (28). This method turns out to be numerically unstable. This is a surprising fact, because the calculation that has led to Eq. (28) consists of two straightforward variations. A convergent algorithm is obtained by modifying the block matrices, representing instead of Eq. (28) the integral

$$\int_A \gamma \delta t^i_{,\alpha} \delta t^j_{,\beta} g_{ij} a^{\alpha\beta} dA. \tag{50}$$

The only fixed points of the Newton-Raphson method (49) are the zeros of the vector (L_r, L_s). These are independent of the particular choice of the matrix on the left-hand side of Eq. (49). The modified matrix leads to a convergent iteration toward the accurate solution. This is confirmed by the examples in Sec. VI B. The analogous argument applies to the approximation in Eq. (30).

The search for the correct boundary shape is strongly nonlinear in the position variables. In order to remove the main nonlinearities, which are caused by the surface metric expressions $\sqrt{a}$ and $a^{\alpha\beta}$, the nodes of the elements are moved to their corresponding coordinates (r_d, s_d) after each step of the geometric system. Then, all integrals can be performed directly on the element edges. Also the normal vector can be taken from the element sides. In the previous section, we used a convenient variational notation to express the change of the free-energy contribution F by changes of the boundary parametrization. Essentially the same equations are obtained by differentiating the discrete version of F with respect to the DoFs, which are the nodal degrees of freedom of the corresponding variables. The only difference is that the variation δt^i in the continuous formulation must be replaced by the vectorial test function $\chi_d \mathbf{e}_i$, and the variation $\delta t^i_{,\alpha}$ by its tangential derivative $\mathbf{T}_\alpha \cdot \nabla \chi_d \mathbf{e}_i$.

C. Controlling the tangential displacements of boundary nodes

For a given discretization, we must not only find the correct boundary shape, but its discretization should also remain well-proportionate. Very long and very short element sides cause badly conditioned matrices and make the algorithm unstable. Several algorithms implementing the weak form of the free-surface boundary condition encounter these intrinsic instabilities of the boundary mesh. For the program *Surface Evolver* this manifests itself in shrinking and growing surface facets. It is therefore recommended to monitor the mesh quality and remove too small or split too large elements.[32] Similar effects were reported by Brinkmann.[33]

In Sec. III B, the assignment of the boundary conditions to the fluidic and the geometric systems was described. There, we found that the presence of incompatible forces may easily destroy a free surface that essentially attempts to minimize the lengths $A_{(m)}$ of the free-surface sides in each element m. Because all fluidic stresses are constrained to have only normal components, we are free to use additional tangential force components for keeping the boundary mesh as regular as possible. This can be done during the assembly of the system matrices by weighting the surface tension by the element side length $A_{(m)}$, divided by the average length $\langle A_{(m)} \rangle$ of all element sides contributing to the free surface. Of course, this weighting factor becomes ineffective if all sides have equal length. Any length difference of adjacent sides causes an additional force that tries to equalize them. The tension forces for each element side are then equivalent to a first variation of the functional $\gamma A^2_{(m)}/(2\langle A_{(m)} \rangle)$, which describes a rubber band with Hookean forces. Instead of $\delta F[\delta \mathbf{t}]$ from Eq. (18), we thus assemble on each element

$$\frac{\gamma}{2\langle A_{(m)} \rangle} \delta(A^2_{(m)})[\delta \mathbf{t}] = \gamma \frac{A_{(m)}}{\langle A_{(m)} \rangle} \delta A_{(m)}[\delta \mathbf{t}]. \tag{51}$$

The second variations of $A_{(m)}$ and $A^2_{(m)}/2$ are not proportional to each other,

$$\frac{\gamma}{2\langle A_{(m)} \rangle} \delta^2(A^2_{(m)})[\delta \mathbf{t}, \delta \mathbf{t}] = \gamma \frac{A_{(m)}}{\langle A_{(m)} \rangle} \delta^2 A_{(m)}[\delta \mathbf{t}, \delta \mathbf{t}] + \gamma \frac{1}{\langle A_{(m)} \rangle} (\delta A_{(m)}[\delta \mathbf{t}])^2. \tag{52}$$

In the implementation we therefore took only the first term on the right-hand side of Eq. (52). In this sense, we did not strictly implement the behavior of a rubber band, but yet a stabilized version of the free-surface tension terms. After convergence, all boundary sides of the mesh representing the free surface have equal lengths, and the extra terms $A_{(m)}/\langle A_{(m)} \rangle$ do not change the behavior of the free surface.

V. SUMMARY OF THE ALGORITHM

Here, we provide a short overview of the complete algorithm. The required steps are as follows:

(1) Choose an initial mesh and initial ambient pressure p_0.
(2) Until convergence, repeat the following steps:
   (a) Smooth the inner mesh if it is too distorted.
   (b) Repeatedly solve the fluidic system for p, u, and v until the perfect-slip boundary condition is established.
   (c) Subtract the average from p.

(d) Solve the geometric system for the new boundary. At the same time, search for the value of p_0 that keeps the volume unchanged.
(e) Set the mesh boundary nodes to the parametrization values of the geometric system.

The fluidic system is assembled according to Eqs. (35)–(43) with constraints that account for the proper boundary conditions. To give the full algorithm at this point, we summarize also the terms of the geometric system. The update Eq. (49) is written as

$$\begin{pmatrix} K_{rr} & K_{rs} \\ K_{sr} & K_{ss} \end{pmatrix} \begin{pmatrix} \vec{r}^{(\mathrm{new})} \\ \vec{s}^{(\mathrm{new})} \end{pmatrix} = -\tau \begin{pmatrix} L_r \\ L_s \end{pmatrix} + \begin{pmatrix} K_{rr} & K_{rs} \\ K_{sr} & K_{ss} \end{pmatrix} \begin{pmatrix} \vec{r}^{(\mathrm{old})} \\ \vec{s}^{(\mathrm{old})} \end{pmatrix} \tag{53}$$

with entries that are assembled per element m,

$$[L_r^{(m)}]_d = \int_{A_{(m)}} \chi_d (\mathbf{e}_x \cdot \sigma \cdot \mathbf{N}) dA + \frac{\gamma A_{(m)}}{\langle A_{(m)} \rangle} \int_{A_{(m)}} (\nabla \chi_d \cdot \mathbf{T})(\mathbf{e}_x \cdot \mathbf{T}) dA, \tag{54}$$

$$[K_{rr}^{(m)}]_{de} = -\int_{A_{(m)}} \chi_d \chi_e (\mathbf{e}_x \cdot \nabla p)(\mathbf{e}_x \cdot \mathbf{N}) dA + \int_{A_{(m)}} \chi_d (\nabla \chi_e \cdot \mathbf{T})\{(\mathbf{e}_x \cdot \sigma \mathbf{N})(\mathbf{e}_x \cdot \mathbf{T}) - (\mathbf{e}_x \cdot \sigma \mathbf{T})(\mathbf{e}_x \cdot \mathbf{N})\} dA + \frac{\gamma A_{(m)}}{\langle A_{(m)} \rangle} \int_{A_{(m)}} (\nabla \chi_d \cdot \mathbf{T}) \times (\nabla \chi_e \cdot \mathbf{T}) dA, \tag{55}$$

$$[K_{rs}^{(m)}]_{de} = -\int_{A_{(m)}} \chi_d \chi_e (\mathbf{e}_x \cdot \nabla p)(\mathbf{e}_y \cdot \mathbf{N}) dA + \int_{A_{(m)}} \chi_d (\nabla \chi_e \cdot \mathbf{T})\{(\mathbf{e}_x \cdot \sigma \mathbf{N})(\mathbf{e}_y \cdot \mathbf{T}) - (\mathbf{e}_x \cdot \sigma \mathbf{T})(\mathbf{e}_y \cdot \mathbf{N})\} dA. \tag{56}$$

The remaining entries can be obtained by permutations of x and y together with r and s. Again, constraints were used to keep the contact lines pinned. In all applications we used values of τ between 0.1 and 1.0.

VI. NUMERICAL EXPERIMENTS

We performed all our test cases for a two-dimensional fluid. The programs were written using the open-source C++ library *libmesh*,[45] which allows changes to the element geometry in a user's routine and provides a powerful constraint method.

A. The instability of a "direct explicit update" algorithm

In a first numerical example, we did not use the modified Newton's method with the update rule (53), but instead with the direct and explicit update rule

$$\begin{pmatrix} \vec{r}^{(\mathrm{new})} \\ \vec{s}^{(\mathrm{new})} \end{pmatrix} = -\tau \begin{pmatrix} L_r \\ L_s \end{pmatrix} + \begin{pmatrix} \vec{r}^{(\mathrm{old})} \\ \vec{s}^{(\mathrm{old})} \end{pmatrix}. \tag{57}$$

The stability of this update rule sensibly depends on the step-size parameter τ. The allowed range of τ is a function of the size of the elements, the curvature, etc. In a simple situation, a homogeneous pressure field deforms the boundary into a circular arc with radius $R=-1/\kappa=p_0/\gamma$. For the combination $R=0.5$, $\gamma=1$, and $p_0=2$, we found the update rule (57) to be stable for 14 first-order FEs and a given step size $\tau=0.05$. The resulting approximation is indistinguishable from the first-order approximation in Fig. 2. In contrast, the update is unstable for the same step size and 52 first-order FEs. After a few iteration steps, starting from the exact solution, the finite-element approximation was completely destroyed. Using second-order FEs, the instability was similar.

B. Testing the accuracy of the modified Newton algorithm

In order to confirm the accuracy of the curvature approximation of the algorithm in Eqs. (53)–(56), we test two cases that can be solved analytically. Similar to the calculation in Appendix A, a prescribed pressure determines the free-surface shape. Then, the approximation in Eq. (30) becomes exact and simplifies to

$$\delta\sigma_{ij} = -g_{ij} p_{,k} \delta t^k. \tag{58}$$

Thus, possible approximation errors result only from the discretization of the curvature.

Figure 2(a) depicts the most simple situation in which a homogeneous pressure field deforms the boundary into a circular arc, as in the previous example. The free surface shape is approximated by the sides of five second-order FEs and 14 first-order FEs, respectively. In dimensionless units, the surface tension is $\gamma=1$ and the prescribed pressure $p_0=2$ produces a circle with radius $R=1/2$ as the exact solution. The initial geometry of both surface approximations was the straight connection between the fixed end points. Good convergence was reached after 100 iterations with step-size parameter $\tau=1$. The comparison of a second-order approximation with a first-order one, using more than twice as many FEs, clearly reveals the superiority of the second-order parametrization. The relative errors of the numerically resulting curvatures are about 8.6×10^{-6} and 5.0×10^{-3} for the

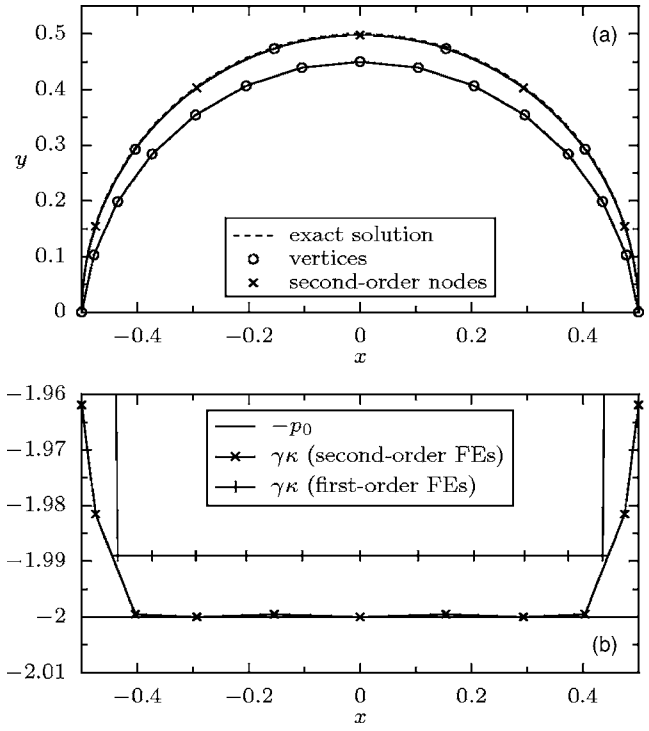

FIG. 2. Panel (a) depicts the free-surface deformation, which is caused by a prescribed homogeneous pressure $p_0=2$. The exact solution (dashed curve) in Cartesian coordinates x and y is a half-circle with radius 1/2. Two finite-element approximations are presented, one with five second-order FEs (upper solid curve) and another one with 14 first-order FEs (lower solid curve). The first-order FEs are bounded by vertices (indicated by small circles), while the second-order FEs also contain second-order nodes (small crosses) in between. Panel (b) compares an estimate of the Laplace pressure $\gamma\kappa$ and the applied pressure p_0 as a function of the spatial x coordinate. The agreement is excellent, except for the outliers near the end points, where the estimate of the curvature is less accurate.

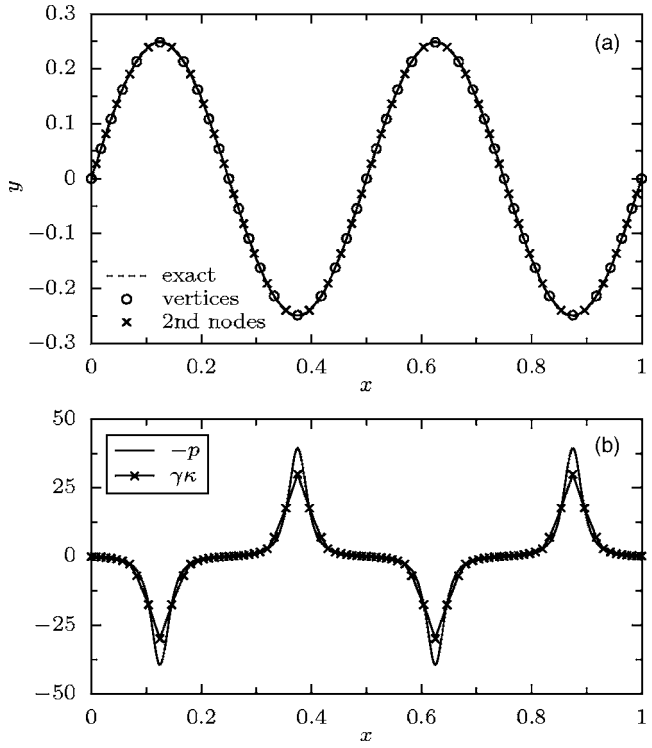

FIG. 3. Same as Fig. 2, for an expected sinusoidal boundary shape $y=h(x)=0.25\sin(4\pi x)$, generated by the prescribed pressure of Eq. (59) with surface tension $\gamma=1$. The approximated free surface consists of 40 second-order FEs.

second-order and first-order approximations, respectively. These values are obtained from the position of the topmost node.

The result of an alternative estimate of the approximation error is visualized in Fig. 2(b). The normal vectors at each node are calculated from the resulting finite-element side. Due to the elements being second-order, we obtain a unique normal vector at second-order nodes. At vertices where two elements meet and where the surface parameterization is not smooth, we average the resulting two normal vectors. The curvature estimate at a node in Fig. 2(b) is then given by the curvature of a circle sector that is determined by the two neighbors of the specific node. The sector is bounded by the two corresponding normal vectors; its chord length equals the distance between the two neighbors. This yields a reconstruction of the curvature from the change of the normal vector per arclength of the surface. It is clear by construction that the normal vector at the end points cannot be reliably estimated. This causes the outliers in Fig. 2(b). All other nodes fit well.

In the next accuracy test, depicted in Fig. 3, a variable pressure is prescribed, for which the resulting boundary shape is known. Figure 3(a) illustrates the approximation of a sinusoidal boundary height function $y=h(x)=\alpha\sin(\beta x)$, which is caused by the corresponding pressure field

$$p(x,y) = -\gamma\kappa(x) = \gamma\frac{\alpha\beta^2\sin(\beta x)}{[1+\alpha^2\beta^2\cos^2(\beta x)]^{3/2}}. \tag{59}$$

Again, the approximation in Eq. (30) becomes exact, and we expect similar discretization errors as in the previous example. The initial geometry was the straight line between the end points of the surface. Good convergence was reached after 60 iterations with step size $\tau=1$. The relative error of the curvature 6.7×10^{-3} is larger than in the previous example because the parts with the maximal curvature are discretized less densely. Nevertheless, the error is still small enough to reproduce the expected boundary shape with excellent accuracy. It decreases with the number of approximating elements. A first-order approximation leads to a much larger, possibly intolerable error of 2.0×10^{-1}. The side lengths of the elements in Fig. 3(a) vary only by ±0.007%. This small deviation demonstrates that the mesh regularization method does not influence the final behavior of the free boundary.

Concerning the discretization errors of the curvature, the accuracy test in Fig. 3 covers already the general case. According to the construction of the algorithm, the flow exerts stress on the boundary only in a normal direction. Whether this stress is of viscous nature or due to a pressure difference is irrelevant for the resulting curvature.

C. A deformed microdroplet

In order to explicitly show that the quality of the curvature discretization does not depend on the origin of the ap-

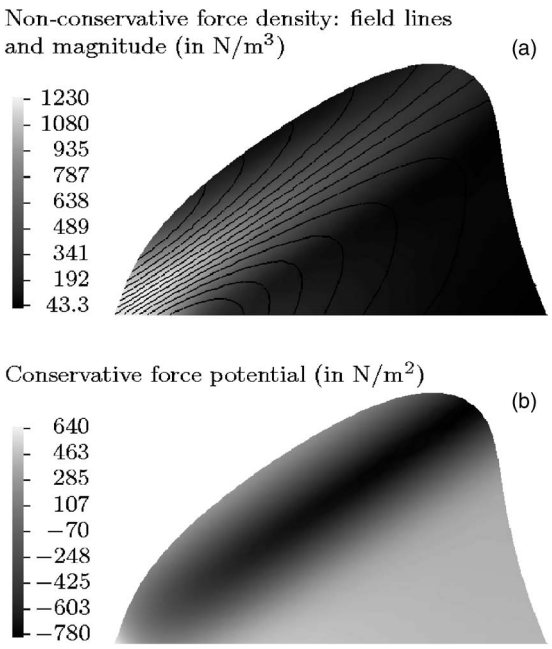

FIG. 4. The force density that models the effect of the surface-acoustic wave in the droplet given in Fig. 5. Panel (a) depicts the nonconservative part that causes the flow; (b) shows the potential of the conservative part that contributes only to the pressure.

plied normal stress, we return to the introductory motivation for the present work. The previous examples were analytically solvable. The form and internal streaming of microdroplets, however, cannot be determined analytically.

In the experiment, the internal flow is agitated by a surface-acoustic wave (SAW) due to the *acoustic streaming effect*.[46] Because the very details of the impact of the SAW are not known, we model it here by a body force that is active in the fluid only, as depicted in Fig. 4. The force is concentrated in a narrow channel that starts at the left contact point where the SAW enters and continues into the fluid. The fluid is carried along this channel, from the entry point of the SAW into the droplet, giving rise also to a back flow.[47] Additionally, the force has a strong conservative portion that is balanced by the pressure in the fluid.

The resulting stationary droplet shape and the internal velocity and pressure fields are presented in Fig. 5. The initial shape was a half-circle with the same two-dimensional volume. Good convergence was reached after seven iterations with a step-size parameter $\tau=0.5$. The material properties are those of water and air at room temperature, i.e., $\eta=10^{-3}$ kg/ms and $\gamma=72.8\times10^{-3}$ N/m. The deformed boundary consists of two regions, one with negative curvature (as the initial half circle) and another one at the right flank of the droplet with positive curvature. The—admittedly strange—deformation of the droplet qualitatively agrees with the experimentally observed jumping droplet in Fig. 4 of the publication by Wixforth *et al.*.[10] The deformation is due to the large conservative contribution of the driving force and the resulting pressure. The viscous forces for the given velocities are far too weak to lead to a substantial deformation of the free surface. The capillary number for the illustrated flow is $\mathrm{Ca}\approx10^{-5}$; the Bond number is around 1. Although

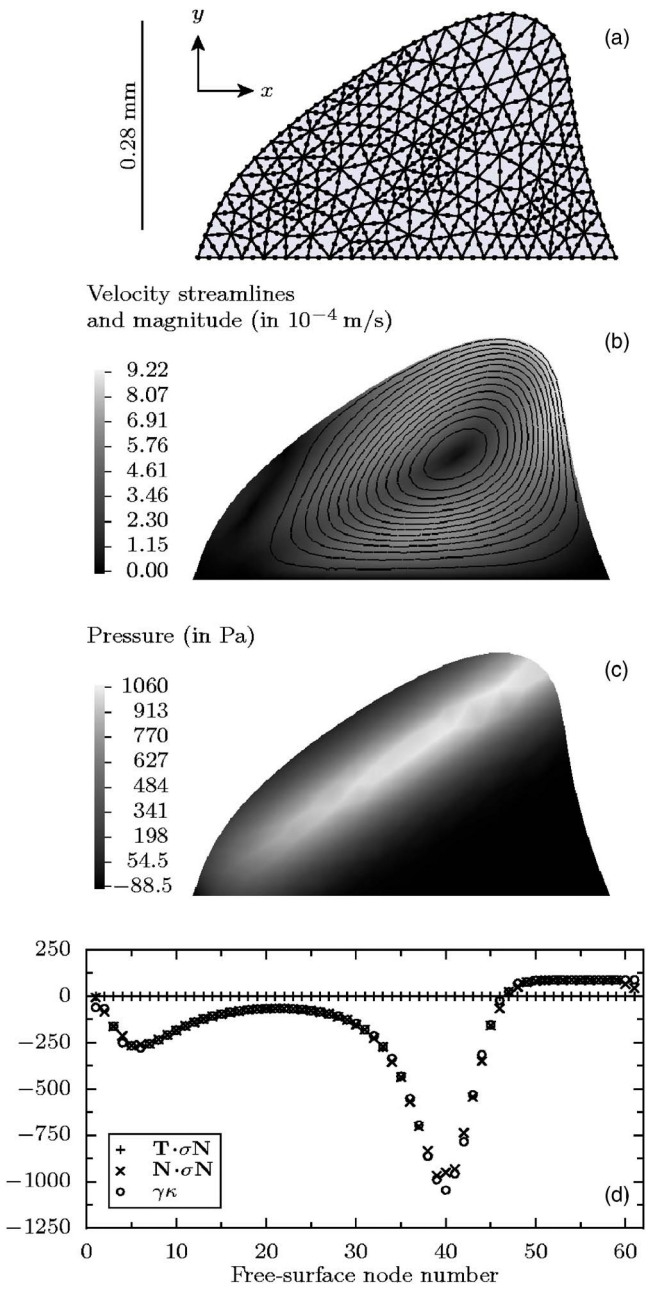

FIG. 5. A deformed microdroplet, sitting on a flat substrate with pinned contact points. The deformation is due to an internal pressure and viscous flow, both caused by the body force density illustrated in Fig. 4. The material properties are those of water surrounded by air at room temperature. The panels depict the computational grid (a), the flow (b), and the pressure field (c), respectively. Note that the deformation is predominantly caused by the pressure, which corresponds to the case in which $\mathrm{Ca}\ll\mathrm{Bo}$. In panel (d), the free-boundary condition is examined: The normal stress $\mathbf{N}\cdot\sigma\mathbf{N}$ equals the Laplace pressure $\gamma\kappa$; the tangential stress $\mathbf{T}\cdot\sigma\mathbf{N}$ vanishes.

the free surface is significantly deformed, its discretization by finite-element sides is as regular as possible. Their lengths vary only by $4.5\times10^{-5}\%$. This guarantees that the behavior of the boundary is indeed that of a free surface and is not influenced by the automatic regularization technique described in Sec. IV C. Figure 5(d) quantifies the normal stress condition. For each node, we integrated the normal and the tangential component of the normal stress, weighted with the corresponding ansatz function of the node. The tangential

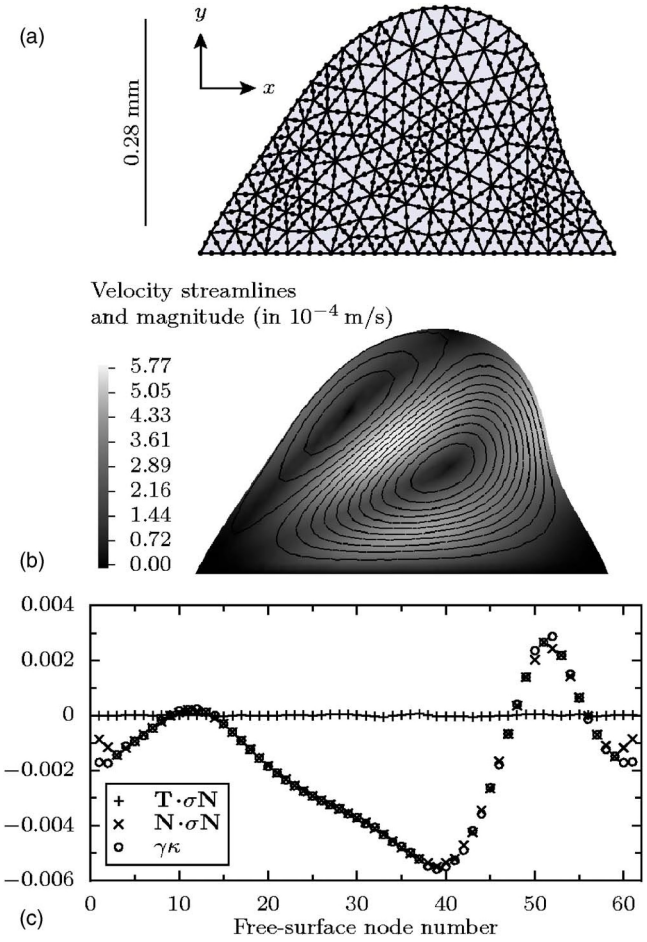

FIG. 6. A microdroplet of similar shape as in Fig. 5, deformed only by the viscous stress at the boundary. The flow is driven by the nonconservative force density of Fig. 4(a) and vanishing conservative part of the force density. The surface tension γ is 10^5 times smaller than in Fig. 5. This corresponds to the case $\mathrm{Bo} \ll \mathrm{Ca}$. Panel (c) corresponds to Fig. 5(d).

component vanishes perfectly. The normal component coincides well with the reconstruction estimate of the curvature, which was described in the previous examples. Thus, the free-surface boundary condition is indeed satisfied.

In order to prove that our algorithm can likewise produce stable results in the parameter regime $\mathrm{Bo} \ll \mathrm{Ca}$, we consider a droplet that is deformed only by viscous stress. In Fig. 6, we used the same nonconservative force that is visualized in Fig. 4, but we omitted the conservative part. Thus, the pressure was constant and $\mathrm{Bo}=0$. With the same water-air interface tension as in the previous example, the droplet would acquire an almost spherical shape. To obtain a comparable deformation as in the previous case together with $\mathrm{Ca} \approx 1$, we took an artificial 10^5 times smaller surface tension. The initial geometry was the droplet shape of Fig. 5. In this example, the stress that deforms the free surface depends much more strongly on the shape of the surface itself. Thus, the approximation in Eq. (30), expressing surface-stress variations by variations of the surface shape, becomes questionable. As a result, we had to set the numerical parameter τ to a smaller value than in the previous example, in order to reduce the step size of the surface update. This deteriorates the convergence rate of the modified Newton method, and convergence was achieved after 30 iterations.

VII. SUMMARY AND OUTLOOK

In this work, we presented a weak formulation of free-surface boundary problems in arbitrary coordinate systems. The steps of the derivation are physically and mathematically founded using variational techniques for the Stokes equations and the differential geometry of the surface. We found that the applicability of different numerical treatments of the curvature terms depends strongly on the scales of the system. Our method is designed for Bond and capillary numbers assuming values from zero up to unity. Which one is larger plays no role.

A decisive benefit of our method is the automatic control of mesh regularity at a free surface. Many algorithms implementing the weak form of the free-surface boundary condition encounter intrinsic instabilities of the boundary mesh. Often, it is therefore necessary to create a completely new mesh after several iteration steps. Our formulation includes a smooth transition to the behavior of a rubber blanket when the boundary mesh becomes distorted. This leads to an inherent regularization of the mesh without affecting the behavior of the free surface.

As another important result, we find that for physical reasons the geometric variables for the parametrization of the free surface should be approximated on the same level of accuracy as the velocity variables. This substantiates numerical observations reported by Bänsch.[28]

The quality of our numerical approach is tested by two analytically solvable examples. The numerical results for the curvature of the free surface and for the stress that causes the deformation confirm that the free-surface boundary condition is indeed satisfied in the implemented weak sense. Moreover, transcending a weak-sense result, a reliable reconstruction of the curvature by the normal vectors of the finite-element sides is feasible. Two further examples illustrate that the ratio of capillary number and Bond number has only a weak influence on the stability of the algorithm.

The presented covariant formulation opens the possibility to utilize the powerful differential geometric description of free surfaces in finite-element implementations of the Stokes equations. It thus provides a natural approach to treat surfaces and interfaces with a richer behavior, such as lipid vesicles containing bending stiffness, area constraints, and much more. Many potential applications can be found in the literature on lipid vesicle geometry, where other expressions for free-energy contributions of more complicated surfaces are in use.[48–50]

Extensions of the presented approach toward moving contact lines, time-dependent flows, and a three-dimensional implementation are possible. There are still some hurdles to be overcome that can be clearly seen in our derivation. One of them is the principally unknown mutual dependence of the stress tensor and the surface parametrization, where we introduced the approximation (30). Another one is the understanding of the numeric instabilities caused by the second variation of the free energy of the surface, see Eqs. (28) and

(50). It must be emphasized that these approximations determine the rate of convergence of the proposed method but leave the final result unchanged.

These extensions would also provide a solid basis for the theoretical understanding of particle transport in surface-acoustic-wave-driven flows.[13,51,52]

ACKNOWLEDGMENTS

We gratefully acknowledge our experimental partners in the group of Professor Achim Wixforth, University of Augsburg, and the developers of the *libmesh* project. This work was supported by the Deutsche Forschungsgemeinschaft (DFG) via Grant No. 1517/25-1, SFB 486 (Project B13), and the Graduiertenkolleg: Nichtlineare Probleme in Analysis, Geometrie und Physik.

APPENDIX A: A STATIC DROPLET IN CARTESIAN COORDINATES

The aim of this appendix is to recall the variational techniques in a simple three-dimensional Cartesian setup before going to arbitrary coordinates in Appendix B. The argument is similar to the one given by Cuvelier.[24]

We describe the special case in which the two-dimensional free surface A of the fluid can be described by a height function

$$A:\ z = h(x,y) \tag{A1}$$

which is nonzero over a certain region $(x,y) \in E$. The Stokes equations for the static situation with a conservative force $f_i = -\Phi_{,i}$ reduce to

$$0 = -p_{,i} - \Phi_{,i} \tag{A2}$$

with the solution $p(\mathbf{x}) = p_0 - \Phi(\mathbf{x})$. The undetermined homogeneous term p_0 will be identified as the Lagrange multiplier for the constraint of constant volume $V = \int_E dxdy h(x,y)$.

The free energy of the system consists of the surface integral of the constant surface tension and the volume integral of the potential

$$F = \gamma \int_A dA + \int_V \Phi dV = \int_E \mathcal{F}(x,y) dxdy \tag{A3}$$

with

$$\mathcal{F}(x,y) = \gamma \sqrt{1 + [\partial_x h(x,y)]^2 + [\partial_y h(x,y)]^2} + \int_0^{h(x,y)} \Phi(x,y,z) dz. \tag{A4}$$

The Euler-Lagrange equation for finding the extremal F by varying h is then

$$0 = \frac{\partial \mathcal{F}}{\partial h} - \frac{\partial}{\partial x}\frac{\partial \mathcal{F}}{\partial(\partial_x h)} - \frac{\partial}{\partial y}\frac{\partial \mathcal{F}}{\partial(\partial_y h)} \tag{A5}$$

$$= \Phi[x,y,h(x,y)] - \gamma\kappa(x,y), \tag{A6}$$

where κ is the curvature of A, given by

$$\kappa(x,y) = \frac{\partial}{\partial x}\frac{\partial_x h(x,y)}{[1 + (\partial_x h)^2 + (\partial_y h)^2]^{1/2}} + \frac{\partial}{\partial y}\frac{\partial_y h(x,y)}{[1 + (\partial_x h)^2 + (\partial_y h)^2]^{1/2}}. \tag{A7}$$

Because the pressure is given by the potential, the Euler-Lagrange equation is equivalent to the free-surface boundary condition for a static fluid,

$$-p[x,y,h(x,y)] + p_0 = \gamma\kappa(x,y). \tag{A8}$$

At this point it is easy to see that p_0 plays the role of a Lagrange multiplier for a volume constraint. Adding the term

$$\lambda V = \lambda \int_E dxdy h(x,y) \tag{A9}$$

to F gives an additional constant λ in the Euler-Lagrange equation, just as the pressure offset p_0. Because p_0 is yet undetermined, we may identify it with λ.

APPENDIX B: VARIATIONAL CALCULUS FOR THE PARAMETRIZATION OF A FREE SURFACE

In order to prove equality (18), we express the change of the free-energy functional (10) by the change of the Jacobi determinant $\sqrt{a}$ of the surface parametrization. With the infinitesimal surface area $dA = \sqrt{a} d\boldsymbol{\nu}$, the variation of the surface free energy becomes

$$\delta F[\delta \mathbf{t}] = \delta\left(\int_A \gamma dA\right) = \int_E \gamma \delta\sqrt{a} d\boldsymbol{\nu} = \int_E \gamma \frac{\partial\sqrt{a}}{\partial t^i_{,\alpha}} \delta t^i_{,\alpha} d\boldsymbol{\nu}. \tag{B1}$$

The dependence of $\sqrt{a}$ on the tangential vectors follows from its definition as the determinant of the surface metric. For a two-dimensional surface, it reads

$$a = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = \frac{1}{2}\epsilon^{\alpha\gamma}\epsilon^{\beta\delta} a_{\alpha\beta} a_{\gamma\delta} = \frac{1}{2}\epsilon^{\alpha\gamma}\epsilon^{\beta\delta} g_{ij} g_{kl} t^i_{,\alpha} t^j_{,\beta} t^k_{,\gamma} t^l_{,\delta}, \tag{B2}$$

where $\epsilon^{\alpha\beta}$ is the permutation symbol in two dimensions,

$$\epsilon^{\alpha\beta} = \begin{cases} 0 & \alpha = \beta, \\ +1 & \alpha = 1, \quad \beta = 2, \\ -1 & \alpha = 2, \quad \beta = 1, \end{cases} \tag{B3}$$

which is a relative surface tensor with weight +1. The absolute tensor results as

$$\varepsilon^{\alpha\beta} = \frac{\epsilon^{\alpha\beta}}{\sqrt{a}}. \tag{B4}$$

This is analogous to the completely antisymmetric tensor in three dimensions, described in detail by Aris.[26] With the antisymmetric tensor, we obtain the inverse surface metric as

$$a^{\alpha\beta} = \varepsilon^{\alpha\gamma}\varepsilon^{\beta\delta} a_{\gamma\delta}. \tag{B5}$$

A formal derivative of (B2) yields

$$\frac{\partial a}{\partial t^i_{,\alpha}} = 2g_{ij}t^j_{,\beta}\varepsilon^{\alpha\gamma}\varepsilon^{\beta\delta}a_{\gamma\delta} = 2ag_{ij}a^{\alpha\beta}t^j_{,\beta} \quad \text{and} \tag{B6}$$

$$\frac{\partial\sqrt{a}}{\partial t^i_{,\alpha}} = \frac{1}{2\sqrt{a}}\frac{\partial a}{\partial t^i_{,\alpha}} = \sqrt{a}g_{ij}a^{\alpha\beta}t^j_{,\beta} \tag{B7}$$

which can be inserted into (B1) to give the desired result (18).

For a one-dimensional curve in two-dimensional space, the same formula can be derived, but the notation may be somewhat confusing. Summation over the single surface index makes no sense, nevertheless we still have to distinguish between co- and contravariant relative tensors, i.e.,

$$a = a_{11} = g_{ij}t^i_{,1}t^j_{,1}, \tag{B8}$$

$$a^{11} = 1/a_{11} \quad \text{because } a^{11}a_{11} = a^{\alpha\beta}a_{\alpha\beta} = 1. \tag{B9}$$

The formal derivative then becomes

$$\frac{\partial\sqrt{a}}{\partial t^i_{,1}} = \frac{1}{2\sqrt{a}}2g_{ij}t^j_{,1} = \sqrt{a}g_{ij}t^j_{,1}\frac{1}{a_{11}} = \sqrt{a}g_{ij}t^j_{,1}a^{11}, \tag{B10}$$

which completes the result for the one-dimensional surface.

APPENDIX C: INTEGRATION BY PARTS OF THE TENSION FORCES

In order to see that Eq. (19) follows from Eq. (18), we remove the surface covariant derivative from $\delta t^i_{,\alpha}$ by an integration by parts and obtain

$$\delta F[\delta\mathbf{t}] = -\int_A \gamma a^{\alpha\beta}t^i_{,\alpha\beta}g_{ij}\delta t^j - \int_A \gamma_{,\beta}a^{\alpha\beta}t^i_{,\alpha}g_{ij}\delta t^j + \oint_{\partial A}\gamma\nu_\beta a^{\alpha\beta}t^i_{,\alpha}g_{ij}\delta t^j, \tag{C1}$$

where the covariant surface vector ν_β is tangential to A and normal to ∂A. We can express the surface derivatives $t^i_{,\alpha\beta}$ by the tensor $b_{\alpha\beta}$ of the second fundamental form of the surface from Eq. (17) (cf. Aris,[26] p. 216),

$$t^i_{,\alpha\beta} = b_{\alpha\beta}N^i, \tag{C2}$$

arriving at

$$a^{\alpha\beta}t^i_{,\alpha\beta} = a^{\alpha\beta}b_{\alpha\beta}N^i = \kappa N^i. \tag{C3}$$

We have used the definition of the curvature as the trace of the tensor of the second fundamental form as in Eq. (16). For a two-dimensional surface, this is twice the mean curvature $\kappa = 2H = a^{\alpha\beta}b_{\alpha\beta}$, for a one-dimensional surface we have only one entry $\kappa = a^{11}b_{11}$.

As is consistent with the standard literature,[26,34] the term δF from Eq. (18) comprises a curvature term in the normal direction,

$$-\gamma\kappa N^i \tag{C4}$$

and a term accounting for the surface gradient of γ. The space vector

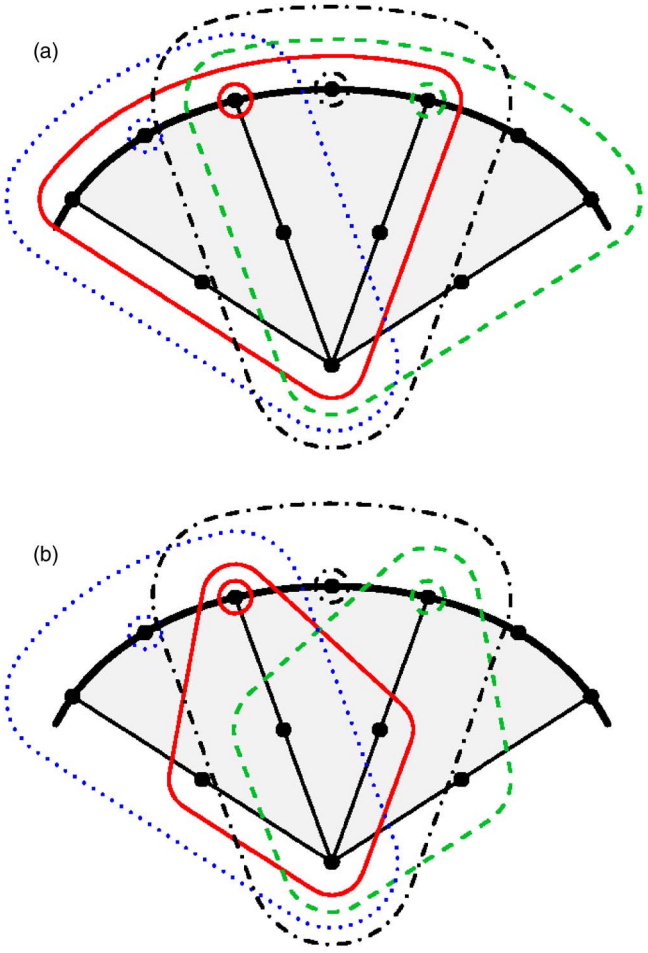

FIG. 7. A sketch of the interdependencies among the degrees of freedom (DoFs) located on three elements. The free surface is indicated by the thick curve. Constrained DoFs are surrounded by small circles. The nodes carrying the corresponding constraining DoFs are surrounded by curves drawn in the same style. Panel (a) depicts the full interdependencies while in (b) the DoFs located at vertices depend only on DoFs located at inner nodes. The missing adjacent DoFs located on the free surface are taken as inhomogeneities.

$$-t^i_{,\alpha}a^{\alpha\beta}\gamma_{,\beta} \tag{C5}$$

is tangential to the surface. The third term on the right-hand side of Eq. (C1), which is an integral over the contact line ∂A, vanishes because for a pinned droplet $\delta t^i = 0$ vanishes on the contact line.

APPENDIX D: INVOKING CONSTRAINTS FOR THE PERFECT-SLIP BOUNDARY CONDITION

The tangential components of the free-surface boundary condition correspond to a perfect-slip boundary condition. When this condition is expressed as a set of constraints for the DoFs, we obtain one equation like (47) per each DoF at the free surface. Because the derivatives of the ansatz functions ϕ from (32) generally do not vanish at proximate nodes, the constraint equations contain nonvanishing weights for all DoFs that are located on the same element. Therefore, the constraints for DoFs that are connected to two adjacent elements create interdependencies of DoFs also on other elements. This is illustrated in Fig. 7(a). As a result, all DoFs

on the free surface depend implicitly on each other.

As a strategy to avoid this full dependency, we replace the constraint equation of type (44) by

$$u_d = w_d + \sum_{e \in \Lambda_d} w_{de} u_e + \sum_{e \in \bar{\Lambda}_d} w_{de} u_e^{(\mathrm{old})}, \tag{D1}$$

where the sums run over two complementary sets Λ_d and $\bar{\Lambda}_d$. The DoFs in Λ_d contribute to the constraint for u_d in the usual way, while those in $\bar{\Lambda}_d$ have been substituted by their old values $u_e^{(\mathrm{old})}$ and thus contribute to the inhomogeneity. There is some freedom in the choice, which of the participating DoFs in one element is in Λ_d and which is taken into $\bar{\Lambda}_d$. We found that the combination illustrated in Fig. 7(b) works well: For the DoFs located at element vertices, we take the DoFs that belong to adjacent nodes on the free surface as inhomogeneities; all other constraining DoFs are located at inner nodes and are not constrained. The DoFs located at the second-order nodes on the free surface acquire their full constraints. When all constrained DoFs are expressed by unconstrained DoFs, then the resulting constraint equations will only contain DoFs that are located at inner nodes of three adjacent elements. This presents a sufficient decoupling of the constraint equations to yield an efficient algorithm.

Although the boundary condition given by Eq. (D1) is not the correct one when the true velocity field has not yet been determined, it still improves as the velocity field tends to the proper solution. Thus, there is hope that the correct boundary condition is established by the successive use of Eq. (D1) using increasingly good values for the values $u_e^{(\mathrm{old})}$. In numerical experiments, the scheme for splitting the interdependencies as illustrated in Fig. 7(b) turned out to be the only one that works. In the examples of Figs. 5 and 6, it took about 20 iteration steps to establish the correct boundary condition from scratch, and 5 iteration steps to reestablish it after a change of the mesh. This could be readily observed because after the first iteration step, the velocity field exhibited oscillations at the boundary nodes that ceased during iteration.

[1] D. Figeys and D. Pinto, "Lab-on-a-chip: A revolution in biological and medical sciences," Anal. Chem. **72**, 330A (2000).

[2] H. A. Stone, A. D. Stroock, and A. Ajdari, "Engineering flows in small devices: Microfluidics toward a lab-on-a-chip," Annu. Rev. Fluid Mech. **36**, 381 (2004).

[3] N. A. Polson and M. A. Hayes, "Microfluidics controlling fluids in small places," Anal. Chem. **73**, 312A (2001).

[4] T. M. Squires and S. R. Quake, "Microfluidics: Fluid physics at the nanoliter scale," Rev. Mod. Phys. **77**, 977 (2005).

[5] U. Thiele, "Open questions and promising new fields in dewetting," Eur. Phys. J. E **12**, 409 (2003).

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

[7] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method. Volume 3: Fluid Dynamics* (Butterworth-Heinemann, Oxford 2000).

[8] A. Ramos, H. Morgan, N. G. Green, and A. Castellanos, "AC electrokinetics: A review of forces in microelectrode structures," J. Phys. D **31**, 2338 (1998).

[9] Z. Guttenberg, A. Rathgeber, S. Keller, J. O. Rädler, A. Wixforth, M. Kostur, M. Schindler, and P. Talkner, "Flow profiling of a surface-acoustic-wave nanopump," Phys. Rev. E **70**, 056311 (2004).

[10] A. Wixforth, Ch. Strobl, Ch. Gauer, A. Toegl, J. Scriba, and Z. Guttenberg, "Acoustic manipulation of small droplets," Anal. Bioanal. Chem. **379**, 982 (2004).

[11] Z. Guttenberg, H. Müller, H. Habermüller, A. Geisbauer, J. Pipper, J. Felbel, M. Kielpinski, J. Scriba, and A. Wixforth, "Planar chip device for PCR and hybridization with surface acoustic wave pump," Lab Chip **5**, 308 (2005).

[12] In an experimental realization, the body force may be caused by a *surface-acoustic wave* (SAW) that travels over the surface of a substrate and rushes into the droplet.[10] The fast motion of the SAW is damped by the fluid, giving rise to a body force via the *acoustic streaming effect*.[46]

[13] K. Sritharan, C. J. Strobl, M. F. Schneider, Z. Guttenberg, and A. Wixforth, "Acoustic mixing at low Reynolds numbers," Appl. Phys. Lett. **88**, 054102 (2006).

[14] V. Cristini and Y.-C. Tan, "Theory and numerical simulation of droplet dynamics in complex flows—A review," Lab Chip **4**, 257 (2004).

[15] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modeling surface tension," J. Comput. Phys. **100**, 335 (1992).

[16] Y. Renardy and M. Renardy, "PROST: A parabolic reconstruction of surface tension for the volume-of-fluid method," J. Comput. Phys. **183**, 400 (2002).

[17] S. Popinet and S. Zaleski, "A front-tracking algorithm for accurate representation of surface tension," Int. J. Numer. Methods Fluids **30**, 775 (1999).

[18] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, "Modelling merging and fragmentation in multiphase flows with SURFER," J. Comput. Phys. **113**, 134 (1994).

[19] A. Smolianski, "Finite-element/level-set/operator-splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces," Int. J. Numer. Methods Fluids **48**, 231 (2005).

[20] C. Pozrikidis, *Boundary Integral and Singularity Methods for Linearized Viscous Flow* (Cambridge University Press, Cambridge, 1992).

[21] A. Z. Zinchenko, M. A. Rother, and R. H. Davis "A novel boundary-integral algorithm for viscous interaction of deformable drops," Phys. Fluids **9**, 1070 (1997).

[22] H. Saito and L. E. Scriven, "Study of coating flow by the finite element method," J. Comput. Phys. **42**, 53 (1981).

[23] S. F. Kistler and L. E. Scriven, "Coating flows," in *Computational Analysis of Polymer Processing*, edited by J. R. A. Pearson (Applied Science, Barking, Essex, 1983) Chap. 8.

[24] C. Cuvelier and R. M. S. Schulkes, "Some numerical methods for the computation of capillary free boundaries governed by the Navier-Stokes equations," SIAM Rev. **32**, 355 (1990).

[25] C. Cuvelier, A. Segal, and A. A. van Steenhoven, *Finite Element Methods and Navier-Stokes Equations* (Reidel, Dordrecht, 1986).

[26] R. Aris, *Vectors, Tensors, and the Basic Equations of Fluid Mechanics* (Dover, New York, 1989).

[27] L. E. Scriven, "Dynamics of a fluid interface. Equation of motion for Newtonian surface fluids," Chem. Eng. Sci. **12**, 98 (1960).

[28] E. Bänsch, Habilitation thesis, Albert-Ludwigs-Universität Freiburg, Freiburg, 1998.

[29] R. A. Cairncross, P. R. Schunk, T. A. Baer, R. R. Rekha, and P. A. Sackinger, "A finite element method for free surface flows of incompressible fluids in three dimensions. Part I. Boundary fitted mesh motion," Int. J. Numer. Methods Fluids **33**, 375 (2000).

[30] M. A. Walkley, P. H. Gaskell, P. K. Jimack, M. A. Kelmanson, and J. L. Summers, "Finite element simulation of three-dimensional free-surface flow problems," J. Sci. Comput. **24**, 147 2005.

[31] M. Renardy, "Imposing 'no' boundary condition at outflow: Why does it work?" Int. J. Numer. Methods Fluids **24**, 413 (1997).

[32] K. A. Brakke, "The surface evolver," Exp. Math. **1**, 141 (1992).

[33] M. Brinkmann, Ph.D. thesis, Universität Potsdam, Potsdam, 2002.

[34] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics* (Pergamon, Oxford, 1963).

[35] G. Dziuk, "An algorithm for evolutionary surfaces," Numer. Math. **58**, 603 (1991).

[36] K. Deckelnick and K. G. Siebert, "$W^{1,\infty}$-convergence of the discrete free boundary for obstacle problems," IMA J. Numer. Anal. **20**, 481 (2000).

[37] H. Lamb, *Hydrodynamics* (Dover, New York, 1932).

[38] H. von Helmholtz, "Zur Theorie der stationären Ströme in reibenden Flüssigkeiten," Nat. Med. **V**, 1 (1869). Reprinted in Ref. 39, pp. 223–230.

[39] H. von Helmholtz, *Wissenschaftliche Abhandlungen* (Barth, Leipzig, 1882), Vol. 1.

[40] B. A. Finlayson, *The Method of Weighted Residuals and Variational Principles* (Academic, New York, 1972).

[41] As a simple demonstration, we consider a problem that is similar to the well-known catenary setup. A curve, representing the one-dimensional surface, in two-dimensional space is assumed to be fixed at its end points. A homogeneous force should act in the direction normal to the connecting line between the end points. If the curve is physically realized by a chain of rigid or elastic elements, its shape is given by a catenary or a parabola, respectively. But if the curve has to be realized by the free surface of a fluid, *this problem is ill-posed and does not possess a stationary solution*. The free surface then can only compensate normal forces and its shape would be a straight line. On the other hand, its curvature would then be zero and therefore could not compensate the external force. Hence, no solution exists.

[42] M. Behr, "On the application of slip boundary condition on curved boundaries," Int. J. Numer. Methods Fluids **45**, 43 (2004).

[43] M. A. Walkley, P. H. Gaskell, P. K. Jimack, M. A. Kelmanson, and J. L. Summers, "On the calculation of normals in free-surface problems," Commun. Numer. Methods Eng. **20**, 343 (2004).

[44] I. N. Bronshtein and K. A. Semendyayew, *Handbook of Mathematics* (Harri Deutsch, Thun, Frankfurt a. M., 1985).

[45] The C++ Finite Element Library *libMesh* is written by B. S. Kirk, J. W. Peterson, R. Stogner, and S. Petersen. Its source-code is available from http://libmesh.sourceforge.net.

[46] W. M. Nyborg, "Acoustic streaming," Phys. Acoust. **2B**, 265 (1965).

[47] A. Wixforth and Z. Guttenberg (private communication).

[48] J. Guven, "Membrane geometry with auxiliary variables and quadratic constraints," J. Phys. A **37**, L313 (2004).

[49] R. Capovilla, J. Guven, and J. A. Santiago, "Deformations of the geometry of lipid vesicles," J. Phys. A **36**, 6281 (2003).

[50] U. Seifert, "Configurations of fluid membranes and vesicles," Adv. Phys. **46**, 13 (1997).

[51] C. J. Strobl, C. Schäflein, U. Beierlein, J. Ebbecke, and A. Wixforth, "Carbon nanotube alignment by surface acoustic waves," Appl. Phys. Lett. **85**, 1427 (2004).

[52] M. Kostur, M. Schindler, P. Talkner, and P. Hänggi, "Chiral separation in microflows," Phys. Rev. Lett. **96**, 014502 (2006).