SIAM J. APPL. MATH.
Vol. 69, No. 5, pp. 1477–1500

NUMERICAL MODELING OF ELECTROWETTING BY A SHAPE INVERSE APPROACH*

JÉRÔME MONNIER†, PATRICK WITOMSKI†, PATRICK CHOW-WING-BOM†, AND CLAIRE SCHEID†

Abstract. We model an electrified droplet spreading on a solid surface. The model aims to seek a drop shape that minimizes its total energy (capillary, electrostatic, and gravitational). We derive the equations and the shape gradient; then we detail the shape optimization algorithm and present some numerical results. Up to a critical applied voltage value, the computed angles fit the predictions of Lippman's equation (plane capacitor approximation). Then, when increasing the voltage, we observe an overestimate of the Lippman prediction. Numerical computations of the curvature show that it remains constant everywhere except in the vicinity of the contact point, where it increases sharply.

Key words. shape optimal, design, electrowetting, contact point, energy minimization, curvature

AMS subject classifications. 49Q10, 74M15

DOI. 10.1137/050624340

1. Introduction.

Electrowetting can be defined as a tool for spreading liquid droplets (e.g., water) on hydrophobic solid surfaces (e.g., polymer film). This is quite a recent technique (see [1]) which holds very attractive properties for manipulation of tiny liquid volumes, as is done, for example, in biotechnologies. The principle of electrowetting is to apply an electric field between the conductor liquid droplet and the solid surface in order to change the droplet spreading on the surface. Given the liquid volume, the main feature for describing the droplet is the wetting angle.

Several articles discuss the experimental aspects of electrowetting and present some analytical analysis; see, e.g., [1], [21], [2], and the references therein. One property of electrowetting still poorly understood by physicists is the contact angle saturation. Several mechanisms for explaining it were proposed in [21], [22], [17], [20]. When increasing the applied electric voltage, the liquid droplet spreads onto the solid and the wetting angle decreases. Nevertheless, this is true only if the value of the applied voltage is less than a certain critical value. Up to this critical value, the contact angle can be derived from the Lippman equation using the plane capacitor approximation. For higher values, one observes a saturation of the wetting angle and for even higher values, instabilities of the contact line liquid-solid-gas can appear. A few hypotheses have been made to explain the saturation phenomenon. Let us cite, for example, the air ionization (see [21]) or electrostatic effects near the wetting line (see [4]). This limiting phenomenon is still under investigation and the full modeling of electrowetting remains an open problem. In other respects, the authors of [5] show that the contact angle does not depend on the potential. It remains equal to the static Young angle (obtained when the potential is null). Also, they observe that the curvature near the contact line increases while the potential increases.

*Received by the editors February 14, 2008; accepted for publication (in revised form) November 6, 2008; published electronically February 25, 2009.
http://www.siam.org/journals/siap/69-5/62434.html
†University of Grenoble and INRIA, Laboratory LJK (Moise project-team), 38041 Grenoble cedex 9, France (monnier@imag.fr, patrick.witomski@imag.fr, chow-wing-bom@imag.fr, claire.scheid@imag.fr).

In this study, we present a mathematical approach for modeling and numerically computing the drop shape, given an applied voltage. The model is based on the shape optimal design methods; see, e.g., [6], [11]. We seek the drop shape (a free surface) such that it minimizes its total energy. The total energy is the sum of the capillary energy, the gravitational energy, and the electrostatic energy. Our numerical modeling is general in the sense that we do not make any assumption on the drop shape. The equations are fully solved and the shape is defined in a general family of surfaces. We assume that the drop shape is steady state and remains two-dimensional (2D) axisymmetric but the method remains valid for three-dimensional (3D) shapes as well. Of course, in the 3D case, the implementation is much more complex and time-consuming than in the present 2D axisymmetric case. This 2D axisymmetric assumption is valid for applied voltages up to the value leading to the instabilities mentioned above.

We obtain numerical results which are consistent with the plane capacitor approximation (Lippman's equation) only for low voltages. For higher voltages, we observe an overestimate of the Lippman predictions. Nevertheless, with the present model, we do not retrieve the wetting angle saturation but instead a deviation from Lippman's predictions of the shape of the drop. In other respects, we focus on the curvature values of the droplet interface. The computed curvature is constant everywhere except in the vicinity of the contact point. If we refine the surface representation near the contact line, we will observe an increase of the curvature—we noticed this behavior for all potentials applied.

The paper is organized as follows. In section 2, we present the electrowetting process and the plane capacitor approximation. In section 3, we derive our mathematical model. It is a shape inverse problem—we seek the drop shape such that it minimizes its total energy. The energy depends on the electric field, which is the solution of the external partial differential equation. The liquid volume is given and constant; it is considered an equality constraint. Finally, the problem consists in finding a min-max solution (saddle point) of an augmented Lagrangian (see [8]). Numerically, the solution is computed using Uzawa's algorithm and a quasi Newton optimization algorithm (BFGS). In section 4, we define the mathematical framework of shape optimization, and we derive the shape derivative of the augmented Lagrangian (the continuous gradient; see Theorem 4.1). In section 5, we detail the discretization of the equations and the shape derivative. The partial differential equation is solved using a standard linear P_1-Lagrange finite element method. The shape parameters and the shape deformation basis are defined; then the shape gradient and the optimization parameters are deduced from section 4. The full optimization process is presented in section 6. It has been implemented in C++. The code uses a public finite element library and a public mesh generator with automatic mesh refinement. In section 7, we present the algorithm we use to compute the droplet curvatures. It is based on a local least square approximation of the control points (second order Bezier approximation). In section 8, we present the numerical results.

2. Electrowetting process. Let us consider the electrowetting process presented in Figure 2.1. We denote by σ_{LS}, σ_{SG}, and σ_{LG} the surface tension coefficients of the liquid-solid interface, solid-gas interface, and liquid-gas interface, respectively. We denote by θ the wetting angle.

When the applied electrical potential u_0 is null, Young's equation gives

$$\cos(\theta_0) = \frac{\sigma_{SG} - \sigma_{LS}}{\sigma_{LG}},$$

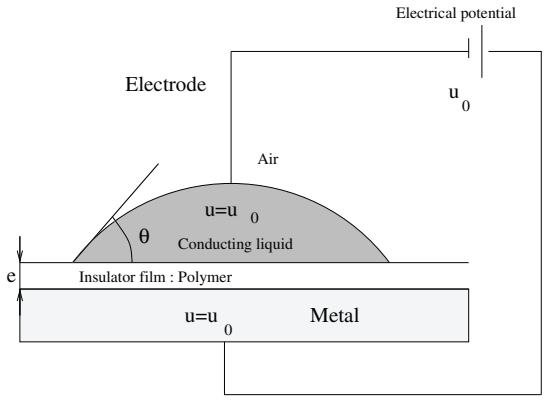

FIG. 2.1. *Electrowetting process.*

where θ_0 is the wetting angle at $u_0 = 0$.

Under the assumption that the system behaves as a plane capacitor with negligible boundary effects, the drop shape obeys the Young equation with the surface tension coefficient modified as follows (see [1]):

$$\sigma_{LS}(u_0) = \sigma_{LS} - \frac{\varepsilon_0 \varepsilon_1}{2e} u_0^2,$$

where e is the insulator thickness and ε_0 and ε_1 are the dielectric constants.

Also, we have (see [1])

$$\cos(\theta) = \cos(\theta_0) + \frac{\varepsilon_0 \varepsilon_1}{2\sigma_{LG}\, e} u_0^2.$$

This last equation is also called Lippman's equation.

Let us note that this law predicts total spreading when the potential increases. However, if u_0 is greater than a critical value u_{cr}, physicists observe a locking phenomenon limiting the spreading of the droplet on the polymer film. Such experiments are studied in [1], [21], [2].

The aim of the present study is to model and numerically compute the liquid drop shape for u_0 lower than the critical value u_{cr}. These computations include the wetting angle θ and the curvature κ of the liquid surface.

3. Mathematical modeling. We model the electrowetting process described in the previous section as a shape inverse problem.

Assumptions.

(i) The applied electrical potential u_0 is continuous.

(ii) The liquid drop is a perfect conductor.

(iii) The drop geometry is 2D axisymmetric.

(iv) Electrostatic effects are negligible far away from the drop.

(v) For $u_0 = 0$, the liquid partially wets the polymer (the spreading coefficient is negative).

Notation (see Figure 3.1). We denote by $u(x)$ the electrical potential at point x, ω_0 the liquid drop, ω_1 the polymer domain, ω_2 the artificially bounded gas domain, and γ_{ext} its external boundary. The external boundary γ_{ext} is supposed to be located far enough from the liquid drop.

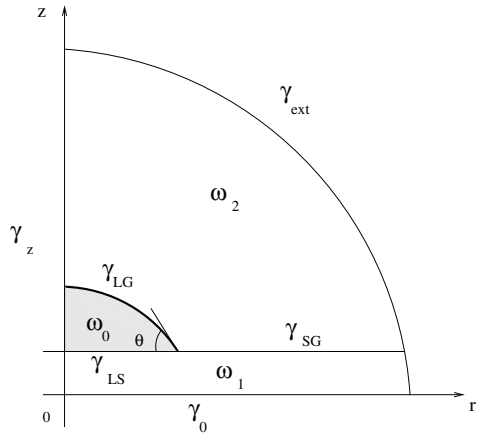

FIG. 3.1. *2D axisymmetric droplet (shaded gray). Domains and boundaries notations.*

We denote by γ_{LS}, γ_{SG}, and γ_{LG} the liquid-solid interface, solid-gas interface, and liquid-gas interface, respectively. We set $\omega = \omega_1 \cup \omega_2 \cup \gamma_{SG}$. We have $\partial\omega_0 = \gamma_{Lz} \cup \gamma_{LG} \cup \gamma_{LS}$ and $\partial\omega = \gamma_0 \cup \gamma_{Sz} \cup \gamma_{LG} \cup \gamma_{Gz} \cup \gamma_{ext}$, with $\gamma_z = \gamma_{Gz} \cup \gamma_{Lz} \cup \gamma_{Sz}$. We set $B = \omega_0 \cup \omega \cup \gamma_{LG} \cup \gamma_{LS}$. The liquid domain ω_0 will be variable; on the other hand, the domain B is given and fixed.

The questions we will answer numerically are the following. Given the electrical potential u_0,

- What is the drop shape?
- What is the wetting angle value θ?
- What is the curvature κ value of the drop surface?

The shape inverse formulation. We model this steady-state free surface problem as a shape inverse problem. We follow the approach presented in [4].

The total energy $\mathcal{E}$ is the sum of the gravitational energy, the capillary energy, and the electrostatic energy. In the 3D case, its expression is the following (see, e.g., [2]):

$$\mathcal{E}_{\omega_0} = \mathcal{E}^{grav}_{\omega_0} + \mathcal{E}^{cap}_{\omega_0} + \mathcal{E}^{elec}_{\omega_0};$$

with the gravitational energy:

$$\mathcal{E}^{grav} = \rho\ g \int_{\omega} z dx;$$

with the capillary balance energy:

$$\mathcal{E}^{cap} = \int_{\gamma_{LS}} (\sigma_{LS} - \sigma_{GS}) ds + \int_{\gamma_{LG}} \sigma_{LG} ds;$$

and the electrostatic energy:

$$\mathcal{E}^{elec} = -\frac{1}{2} \int_{\omega} \varepsilon |\nabla u|^2 dx,$$

where ρ is the liquid density, g is the gravity constant, $\varepsilon = \varepsilon_i$ in ω_i, $i = 1, 2$, and ε_i is the relative dielectric permittivity of ω_i; i.e., $\varepsilon_0 \varepsilon_i$, $i = 1, 2$ is the polymer and the gas permittivity, respectively.

The shape inverse formulation is as follows:

$$\left\{ \begin{array}{l} \text{Find } \omega_0^\star \text{ such that} \\ \mathcal{E}_{\omega_0^\star} \;=\; \min\limits_{(\omega_0;\ \int_{\omega_0} dx = vol)} \mathcal{E}_{\omega_0}, \end{array} \right.$$

where *vol* is the given drop volume.

We set $u_i = u|_{\omega_i}$, $i = 1, 2$. Then, the potential u_i is the solution of the equation

$$-\mathrm{div}(\varepsilon_i \nabla u_i) \;=\; 0 \quad \text{in } \omega_i,\ i = 1, 2, \tag{3.1}$$

with the following Dirichlet boundary conditions:

$$\left\{ \begin{array}{rcll} u_1 & = & u_0 & \text{on } \gamma_{LG}, \\ u_2 & = & u_0 & \text{on } \gamma_{LS}, \\ u_2 & = & 0 & \text{on } \gamma_0. \end{array} \right. \tag{3.2}$$

On the solid-gas interface, we have the transmission boundary conditions

$$\left\{ \begin{array}{rcl} u_1 & = & u_2 \qquad\quad \text{on } \gamma_{SG}, \\ \varepsilon_1 \nabla u_1 n_1 & = & -\varepsilon_2 \nabla u_2 n_2 \quad \text{on } \gamma_{SG}. \end{array} \right. \tag{3.3}$$

On the artificial boundary $\gamma_{ext} = \gamma_{ext}^1 \cup \gamma_{ext}^2$, we impose

$$\varepsilon_i \nabla u_i n_i \;=\; 0 \qquad \text{on } \gamma_{ext}^i,\ i = 1, 2. \tag{3.4}$$

Therefore, the present mathematical problem is a shape optimal control problem for a system governed by a linear steady-state partial differential equation.

2D axisymmetric equations. As mentioned previously, we assume that the drop shape is 2D axisymmetric. We present below the weak formulation of the model. We set

$$X_0(\omega) = \{v \in H^1(\omega);\ v = 0 \text{ on } \gamma_0 \cup \gamma_{LS} \cup \gamma_{LG}\},$$

$$X_t(\omega) = \{v \in H^1(\omega);\ v = 0 \text{ on } \gamma_0;\ v = u_0 \text{ on } \gamma_{LS} \cup \gamma_{LG}\}.$$

The weak formulation of (3.1)–(3.4) in the 2D axisymmetric case is

$$\left\{ \begin{array}{l} \text{Find } u^\omega \in X_t(\omega) \text{ such that} \\ \forall v \in X_0(\omega),\ a_\omega(u^\omega, v) = 0, \end{array} \right. \tag{3.5}$$

where

$$a_\omega(u, v) = \int_\omega \varepsilon r \langle \nabla u, \nabla v \rangle dx,$$

$x = (r, z)$, and $\langle ., . \rangle$ is the inner product of $\mathbb{R}^2$.

It follows from the Lax–Milgram theorem that state equation (3.5) has only one solution for $u^\omega \in X_t(\omega)$.

The shape inverse problem. In its dimensionless form, the drop energy is

$$\mathcal{E}_{\omega_0}(u^\omega) \;=\; \alpha \int_\omega z dx + \int_{\gamma_{LG}} r ds + \mu \int_{\gamma_{LS}} r dr - \delta \int_\omega \varepsilon r |\nabla u^\omega|^2 dx, \tag{3.6}$$

where u^ω is the unique solution of (3.5), $\alpha = \frac{\rho g (L^*)^2}{\sigma_{LG}}$, $\mu = -cos(\theta_0) = \frac{\sigma_{LS}-\sigma_{GS}}{\sigma_{LG}}$, $\delta = \frac{1}{2\sigma_{LG}L^*}$, and L^* is a characteristic length (typically $L^* \approx 10^{-4} - 10^{-3}$ m).

We set the cost function by

$$j(\omega) = \mathcal{E}_{\omega_0}(u^\omega). \tag{3.7}$$

We denote by $\mathcal{D}$ the admissible domain space. (The definition of $\mathcal{D}$ is detailed in the next section.) The shape optimal inverse problem is

$$\begin{cases} \text{Find } \omega^\star \in \mathcal{D} \text{ such that} \\ j(\omega^\star) = \min\limits_{(\omega; \int_{\omega_0} r dx = vol/2\pi)} j(\omega). \end{cases} \tag{3.8}$$

Let us point out that the variable is not the whole domain ω, but more precisely, the liquid-gas interface γ_{LG}; see Figure 3.1. We assume that the inverse shape problem (3.8) admits at least one solution. The existence of an optimal shape is not addressed in the present paper.

The augmented Lagrangian. Problem (3.8) is an optimization problem under an equality constraint. Thus, classically, we introduce the augmented Lagrangian $L_\tau : \mathcal{D} \times \mathbb{R} \longrightarrow \mathbb{R}$, defined by the following (see, e.g., [8]):

$$L_\tau(\omega, \lambda) = j(\omega) + \lambda c(\omega) + \tau c(\omega)^2, \tag{3.9}$$

where $c(\omega)$ is the volume constraint,

$$c(\omega) = \int_{\omega_0} r dx - \frac{vol}{2\pi} = \int_B r dx - \int_\omega r dx - \frac{vol}{2\pi}, \tag{3.10}$$

λ is the Lagrange multiplier, and τ is a penalty parameter.

Then, the shape optimal inverse problem (3.8) is formulated as the saddle-point problem:

$$\begin{cases} \text{Find } (\omega^\star, \lambda^\star) \in \mathcal{D} \times \mathbb{R} \text{ such that} \\ L_\tau(\omega^\star, \lambda^\star) = \min\limits_{\omega} \max\limits_{\lambda} L_\tau(\omega, \lambda). \end{cases} \tag{3.11}$$

We will solve (3.11) using the classical Uzawa algorithm; see, e.g., [8]. This algorithm uses a gradient-type algorithm (BFGS), which requires us to compute the shape derivative of the cost function $\frac{dj}{d\omega}(\Omega)$ and the shape derivative of the constraint $\frac{dc}{d\omega}(\Omega)$. The expressions of these shape derivatives are presented in the next section.

4. Shape derivatives. As mentioned above, we need to compute the shape derivative of the cost function $\frac{dj}{d\omega}(\Omega)$ and the shape derivative of the constraint $\frac{dc}{d\omega}(\Omega)$. This is done using the optimal shape design method (see [15], [6], [11]; definitions of [7], [12] are used). Three approaches are possible: (i) we differentiate the equations and then we discretize them, thus obtaining the discretized continuous gradient; (ii) we discretize the equations and then we differentiate them, thus obtaining the discrete gradient; (iii) we directly differentiate the direct code (typically, using automatic differentiation). In the present study, we follow approach (i). This requires some extra mathematical definitions and tools, but this approach is rigorous; it leads to synthetic expressions of derivatives and it allows us to prove all the differentiabilities required. These derivatives are discretized in the next section, leading to shape gradients. The

family of shapes considered is large enough in the sense that it includes those observed in the experiments.

This section is organized as follows. We define the admissible domain space $\mathcal{D}$ (Lipschitz domains); then we use the classical definition of shape derivatives based on domain deformations (a method of transport with $\mathcal{C}^1$ transformations). We prove the differentiability of the cost function j and the constraint function c with respect to the domain ω. Then, by introducing the adjoint state equation (in our case the adjoint state vanishes), we obtain the differential of j and c (Theorem 4.1). The shape derivative of the augmented Lagrangian L_τ follows (Corollary 4.2).

4.1. Mathematical framework: Domain variations and shape derivatives. We consider a family of Lipschitz domains. We define the space of admissible domains and the derivative with respect to the domain in a classical manner. The domain space is the set of domains homeomorphic to a reference domain. The transformations are $\mathcal{C}^1$ homeomorphisms. This regularity is necessary for all transported integrals to be well defined. The shape derivative of a real valued function is the derivative of the transported function with respect to the transformation. We refer to [15], [6], and we follow the definitions and properties presented in [7], [12].

Admissible domain space. Let $\hat{\Omega}$, a bounded open subset of $\mathbb{R}^2$ with a Lipschitz boundary, be the reference domain $\hat{\Omega} = \Omega_1 \cup \hat{\Omega}_2 \cup \hat{\Gamma}_{SG}$. Ω_1 represents the solid part and $\hat{\Omega}_2$ the gas part. We distinguish the variable part of $\hat{\Omega}$ from its fixed part; see Figure 4.1. We set $\partial\hat{\Omega} = \hat{\Gamma}_{Var} \cup \Gamma_{Fix}$, where $\hat{\Gamma}_{Var} = \hat{\Gamma}_{LG} \cup \hat{\Gamma}_{LS}$ is the variable boundary and Γ_{Fix} is the fixed boundary. We denote by B_{int} a neighborhood of $\hat{\Gamma}_{Var}$, B_{int} large enough; see Figure 4.1.

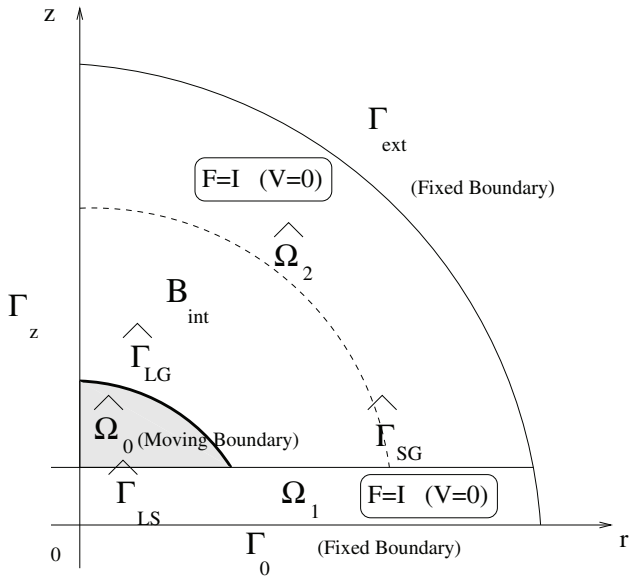

FIG. 4.1. *The reference domain* $\hat{\Omega}$.

We set the function space

$$\hat{\mathcal{F}} = \{\hat{F},\ \hat{F} \text{ bijection of } \hat{\Omega} \text{ onto } \hat{F}(\hat{\Omega});\ \hat{F} \in \mathcal{C}^1(\bar{\hat{\Omega}}, \mathbb{R}^d),\ \hat{F}^{-1} \in \mathcal{C}^1(\bar{\hat{F}}(\hat{\Omega}), \mathbb{R}^d)\} \tag{4.1}$$

and its affine subspace $\hat{\mathcal{F}}_0 = \{\hat{F} \in \hat{\mathcal{F}};\ \hat{F} = I \text{ in } \hat{\Omega} \setminus B_{int}\}$, where I denotes the identity

of $\mathbb{R}^d$. Then, we define the admissible domain space $\mathcal{D}$ as follows:

$$\mathcal{D} = \{\omega = \hat{F}_0(\hat{\Omega});\ \hat{F}_0 \in \hat{\mathcal{F}}_0\}. \tag{4.2}$$

One knows that if $\hat{F}$ is close enough to I in $\hat{\mathcal{F}}_0$ ($(\hat{F} - I)$ small enough), then $\hat{F}(\hat{\Omega})$ is an open set of $\mathbb{R}^2$ with a Lipschitz boundary and $F(\hat{\Gamma}_{Var}) \subset B_{int}$.

Shape derivative of a real valued function. For $\hat{F}_0 \in \hat{\mathcal{F}}_0$, $(\hat{F}_0 - I)$ small enough, we define the domain Ω by $\Omega = \hat{F}_0(\hat{\Omega})$ and $\Gamma_{Var} = \hat{F}_0(\hat{\Gamma}_{Var})$. We set the homeomorphism space defined in Ω (see Figure 4.2) as $\mathcal{F} = \{F,\ F = \hat{F} \circ \hat{F}_0^{-1},\ \hat{F} \in \hat{\mathcal{F}}\}$, and its affine subspace as $\mathcal{F}_0 = \{F,\ F = \hat{F} \circ \hat{F}_0^{-1},\ \hat{F} \in \hat{\mathcal{F}}_0\}$.

Let $F \in \mathcal{F}_0$; we define $\omega = F(\Omega)$ and $V \in \mathcal{C}^1(\bar{\Omega}, \mathbb{R}^d)$ by $V = F - I$. We have $V = 0$ in $\hat{\Omega} \setminus B_{int}$.

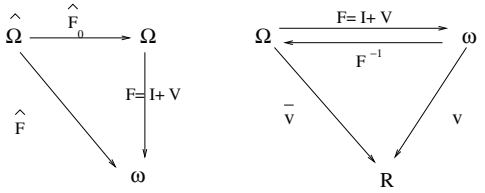

FIG. 4.2. *Change of variables.*

For a given cost function j, $j : \omega \in \mathcal{D} \mapsto j(\omega) \in \mathbb{R}$, we define the "transported" cost function $\bar{\jmath}$ by $\bar{\jmath} : \mathcal{F}_0 \to \mathbb{R}\ :\ F \mapsto \bar{\jmath}(F) = j(F(\Omega)) = j(\omega)$. Then, the derivative with respect to the domain is defined as follows (see, e.g., [15], [7] for more details):

$$\frac{dj}{d\omega}\ (\Omega) \cdot V = \frac{d\bar{\jmath}}{dF}\ (I) \cdot V \quad \forall V \in \mathcal{C}^1(\bar{\Omega}, \mathbb{R}^d). \tag{4.3}$$

4.2. Shape derivatives. We present below the expressions of the exact differentials with respect to the shape ω.

THEOREM 4.1. *There exists $\mathcal{V}_I$, a neighborhood of I in $\mathcal{F}_0$, such that*

(i) *the cost function $j : \mathcal{D} \to \mathbb{R}$; $\omega \mapsto j(\omega) = \mathcal{E}_{\omega_0}(u^\omega)$ belongs to $\mathcal{C}^1$ for all $\omega = F(\Omega)$, $F \in \mathcal{V}_I$. Additionally, for all $V \in \mathcal{C}^1(\bar{\Omega}, \mathbb{R}^2)$, we have*

$$\frac{dj}{d\omega}(\Omega).V = \frac{\partial \mathcal{E}_{\Omega_0}}{\partial \omega}(u^\Omega).V, \tag{4.4}$$

with u^Ω the solution of the state equation (3.5) *posed in Ω and*

$$\begin{aligned}
\frac{\partial \mathcal{E}_{\Omega_0}}{\partial \omega}(u^\Omega).V = {} & \alpha \int_\Omega z \circ V\ dx + \alpha \int_\Omega z \mathrm{div}(V)\ dx \\
& + \int_{\Gamma_{LG}} r \circ V\ ds + \int_{\Gamma_{LG}} r\ \mathrm{div}_\Gamma V\ ds \\
& + \mu \int_{\Gamma_{LS}} r \circ V\ dr + \mu \int_{\Gamma_{LS}} r\ \mathrm{div}_\Gamma V\ dr \\
& - \delta \int_\Omega \varepsilon\ (r \circ V)\ |\nabla u^\Omega|^2\ dx - \delta \int_\Omega \varepsilon r |\nabla u^\Omega|^2\ \mathrm{div}(V)\ dx \\
& + \delta \int_\Omega \varepsilon r < (\ {}^T DV + DV) \nabla u^\Omega, \nabla u^\Omega > dx,
\end{aligned}$$

with $\mathrm{div}_\Gamma V = (\mathrm{div}(V) - \langle n,\ {}^T DV n \rangle)$, *$n$ the external normal, and $x = (r, z)$.*

(ii) *The volume constraint* $c(\omega)$ *belongs to* $\mathcal{C}^1$ *for all* $\omega = F(\Omega)$, $F \in \mathcal{V}_I$ *and, for all* $V \in \mathcal{C}^1(\bar{\Omega}, \mathbb{R}^2)$,

$$\frac{dc}{d\omega}(\Omega).V = -\int_\Omega r \circ V dx - \int_\Omega r \mathrm{div}(V) dx. \tag{4.5}$$

Proof. The proof follows with three steps: 1. transport of equations; 2. differentiability with respect to ω; 3. use of the adjoint technique leading to the expression of the exact differential.

Step 1. *Transport of equations.* As noted previously, we need to transport the cost function j in order to compute its shape derivative. To this end, we need to transport all the equations on the reference domain $\Omega = F^{-1}(\omega)$.

For any $u, v \in X_0(\omega)$, we let

$$\bar{a}(F; \bar{u}, \bar{v}) = a_{F(\Omega)}(\bar{u} \circ F^{-1}, \bar{v} \circ F^{-1}) = a_\omega(u, v)$$
$$= \int_\Omega \bar{\varepsilon}\bar{r} <^T (DF^{-1} \circ F)\nabla\bar{u}, ^T (DF^{-1} \circ F)\nabla\bar{v} > |\mathrm{det}DF| d\bar{x},$$

with $\bar{u} = u \circ F$, $\bar{v} = v \circ F$, $\bar{x} = x \circ F$, and $\bar{\varepsilon} = \varepsilon \circ F$; see Figure 4.2.

The mapping $v \in X_0(F(\Omega)) \mapsto v \circ F \in X_0(\Omega)$ is an isomorphism for $F \in \mathcal{F}_0$. In other respects, the Dirichlet data u_0 is constant; hence $u_0 = u_0 \circ F$. Then, since state equation (3.5) has a unique solution u^ω, the transported state equation

$$\text{Find } \bar{u}^F \in X_t(\Omega) : \ \bar{a}(F; \bar{u}, \bar{v}) = 0 \quad \forall \bar{v} \in X_0(\Omega)$$

has a unique solution $\bar{u}^F = u^\omega \circ F$.

Similarly, for any $u \in X_0(\omega)$, we let $\bar{\mathcal{E}}(F; \bar{u}) = \mathcal{E}_{F(\Omega_0)}(\bar{u} \circ F^{-1}) = \mathcal{E}_{\omega_0}(u)$. We have $\bar{\jmath}(F) = \bar{\mathcal{E}}(F; \bar{u}^F)$,

$$\begin{aligned} \bar{\jmath}(F) = \alpha &\int_\Omega \bar{z} \ |\mathrm{det}DF| \ d\bar{x} \\ &+ \int_{\Gamma_{LG}} \bar{r} \ Jac(F) \ d\bar{s} + \mu \int_{\Gamma_{LS}} \bar{r} \ Jac(F) \ d\bar{r} \\ &- \delta \int_\Omega \bar{\varepsilon} \ \bar{r} \ | \ ^T(DF^{-1} \circ F)\nabla\bar{u}^F|^2 \ |\mathrm{det}DF| \ d\bar{x}, \end{aligned} \tag{4.6}$$

with $Jac(F) = |\mathrm{det}DF| \ \| \ ^T DF^{-1}.n\|_{\mathbb{R}^2}$.

Also, we define

$$\bar{c}(F) = \int_B r dx - \int_\Omega \bar{r} \ |\mathrm{det}DF| \ d\bar{x} - \frac{vol}{2\pi}. \tag{4.7}$$

Step 2. *Differentiability with respect to* ω. The mapping $\bar{a}(F; \bar{u}, \bar{v})$ is C^1 with respect to $(F; \bar{u})$. It follows from the implicit function theorem that the transported state equation defines a C^1-mapping $F \mapsto \bar{u}^F : \mathcal{F}_0 \to X_t(\Omega)$ in a neighborhood $\mathcal{V}_I$ of I.

Then, since the mapping $\bar{\mathcal{E}}$ is of class $\mathcal{C}^1(\mathcal{F} \times X_0(\Omega))$, the cost function j is continuously differentiable. Also, the constraint function c is continuously differentiable.

Step 3. *Expression of the exact differential.* By definition, we have $\frac{dj}{d\omega}(\Omega) \cdot V = \frac{d\bar{\jmath}}{dF}(I) \cdot V$ for all $V \in \mathcal{C}^1(\bar{\Omega}, \mathbb{R}^2)$.

Then, using the classical adjoint technique, we have

$$\frac{d\bar{j}}{dF}(I).V = \frac{\partial\bar{\mathcal{E}}}{\partial F}(I;u^\Omega).V - \frac{\partial\bar{a}}{\partial F}(I;u^\Omega,p^\Omega).V \quad \forall V \in \mathcal{C}^1(\bar{\Omega},\mathbb{R}^2),$$

where u^Ω is the solution of the state equation posed in Ω and $p^\Omega \in X_0(\Omega)$ is the adjoint state, unique solution of the following adjoint equation:

$$\frac{\partial\bar{a}}{\partial u}(I;u^\Omega,p^\Omega).v = \frac{\partial\bar{\mathcal{E}}}{\partial u}(I;u^\Omega).v \quad \forall v \in X_0(\Omega).$$

We have

$$\frac{\partial\bar{a}}{\partial u}(I;u^\Omega,p^\Omega).v = a_\Omega(p^\Omega,v) \quad \text{and} \quad \frac{\partial\bar{\mathcal{E}}}{\partial u}(I;u^\Omega).v = -2\delta a_\Omega(u^\Omega,v) = 0 \ \ \forall v \in X_0(\Omega).$$

Hence, $p^\Omega \in X_0(\Omega)$ and $a_\Omega(p^\Omega,v) = 0$ for all $v \in X_0(\Omega)$. Therefore, $p^\Omega = 0$.

Hence,

$$\frac{dj}{d\omega}(\Omega).V = \frac{\partial\bar{\mathcal{E}}}{\partial F}(I;u^\Omega).V \quad \forall V \in C^1(\bar{\Omega},\mathbb{R}^2).$$

Using (4.7) and the classical expression of the derivatives of $|\det(DF)|$, $(DF^{-1}\circ F)$, and $(\| {}^TDF^{-1}.n\|_{\mathbb{R}^2})$ (see, e.g., [15, Chap. IV]), we obtain the result (i).

The result (ii) follows from (4.7) and the expression of the derivative of $|\det(DF)|$. □

Then, we have straightforwardly the following result.

COROLLARY 4.2. *At (λ,τ) given in $\mathbb{R}\times\mathbb{R}^+$, the augmented Lagrangian L_τ is locally and continuously differentiable with respect to ω. And for all $V \in \mathcal{C}^1(\bar{\Omega},\mathbb{R}^2)$,*

$$\frac{\partial L_\tau}{\partial\omega}(\Omega,\lambda).V = \frac{dj}{d\omega}(\Omega).V + \lambda\frac{dc}{d\omega}(\Omega).V + 2\tau c(\Omega)\frac{dc}{d\omega}(\Omega).V, \tag{4.8}$$

where $\frac{dj}{d\omega}(\Omega).V$ and $\frac{dc}{d\omega}(\Omega).V$ are defined by (4.4) *and* (4.5)*, respectively.*

5. Discretization. In this section, we discretize the shape derivative of the augmented Lagrangian L_τ defined by (4.8); then we define the shape parameters and obtain the shape gradient. Then, we detail the full optimization process. We follow [7], [12]; see also [13].

Let us recall that the expression $\frac{\partial L_\tau}{\partial\omega}(\Omega,\lambda).V$ depends on u, the unique solution of (3.5).

Let $(\mathcal{T}_h)$ be a regular family of triangulation, where $\omega = \cup_{T\in\mathcal{T}_h}T$. We compute an approximation of u using the classical piecewise linear conforming finite element method (P_1-Lagrange). This finite element approximation is denoted by u_h, where the parameter h denotes a characteristic mesh size.

Discretization of the boundary and the shape parameters. Let $\hat{\Omega}$ be an open set of reference; typically $\hat{\Omega}$ is a quarter of a disk; see Figure 4.1. The domain of reference $\hat{\Omega}$ is defined using a parametric function:

$$s_{\hat{\Omega}}(t) = \sum_{i=0}^{N-1} \hat{P}_i\, s_i(t)\,,\ t \in [0,1],$$

where $\{s_i(t)\}_{i=0,\dots,N-1}$ are piecewise linear functions, $s_i(\frac{j}{N-1}) = \delta_{ij}$; δ_{ij} denotes the Kronecker symbol, and $\hat{P}_i = ((\hat{P}_r)_i,(\hat{P}_z)_i)^T$ are the control points. We set $(\hat{P}_z)_1 = (\hat{P}_z)_0$.

We have $\Omega = \hat{F}_0(\hat{\Omega})$ with $\hat{F}_0 \in \hat{\mathcal{F}}_0$. Similarly, we define the variable boundary Γ_{LG} (the unknown of the problem) by

$$s_\Omega(t) = \sum_{i=0}^{N-1} P_i \, s_i(t) \, , \; t \in [0,1].$$

Hence, the boundary Γ_{LG} is defined by N control points $P_i, i = 0, \ldots, N-1$.

Initially, these points define $\hat{\Gamma}_{LG}$ as follows (see Figure 5.1):

$$\hat{P}_i = (0,R)^T,$$

$$\hat{P}_i = \left(R \cos\left(\frac{(N-1-i)\pi}{2(N-1)} \right), \; R \sin\left(\frac{(N-1-i)\pi}{2(N-1)} \right) \right)^T, \quad i = 2, \ldots, N-1,$$

$$\hat{P}_1 = \left(R \cos\left(\frac{(N-2)\pi}{2(N-1)} \right) /2, R \right)^T.$$

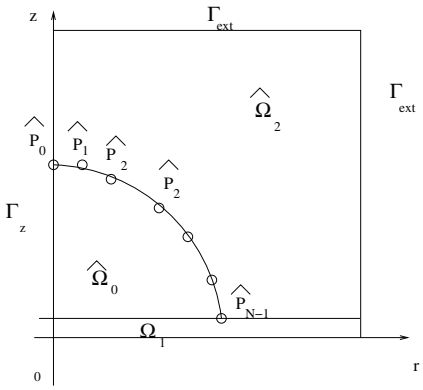

FIG. 5.1. *Reference domain. Parametrization.*

Therefore, during the optimization process, we compute a new domain that requires computing new control points P_i, $i = 0, \ldots, N-1$.

The shape deformation space. Let us discretize the shape deformation V, $V \in \mathcal{C}^1(\bar{\Omega}, \mathbb{R}^2)$. We have $\Omega = \hat{F}_0(\hat{\Omega})$ with $\hat{F}_0 \in \hat{\mathcal{F}}_0$. We set $V = \hat{V} \circ \hat{F}_0^{-1}$. V is defined in Ω, while $\hat{V}$ is defined in $\hat{\Omega}$.

We approximate $C^1(\bar{\hat{\Omega}}, \mathbb{R}^2)$ by $\hat{S}_H$, the vectorial space spanned by $\{\hat{V}_i\}_{i=0,\ldots,N-1}$:

$$\hat{S}_H = Span\{\hat{V}_i\}_{i=0,\ldots,N-1},$$

where the set of vectors $\{\hat{V}_i\}_{i=0,\ldots,N-1}$ is detailed below.

We set $H = \frac{1}{N-1}$. The parameter H denotes a characteristic size of the shape deformation space.

Then, the deformation field V is approximated by

$$V_H = \sum_{i=0}^{N-1} \eta_i V_i, \tag{5.1}$$

where $V_i = \hat{V}_i \circ \hat{F}_0^{-1}$ and η_i, $i = 0, \ldots, N-1$ are real coefficients.

We have $V_H = (\hat{V}_H \circ \hat{F}_0^{-1})$ with

$$\hat{V}_H = \sum_{i=0}^{N-1} \eta_i \hat{V}_i. \tag{5.2}$$

Finally, $C^1(\bar{\Omega}, \mathbb{R}^2)$ is approximated by $S_H = Span\{V_i = \hat{V}_i \circ \hat{F}_0^{-1}\}_{i=0,\dots,N-1}$.

The shape deformation basis. We have $\hat{F}_0 = (I + \hat{V})$, and $\hat{V}$ is approximated by $\hat{V}_H$, which was defined by (5.2).

The basis $\{\hat{V}_i\}_{i=0,\dots,N-1}$, is defined in $\hat{\Omega}$ as follows. For $i = 0, \dots, N-1$, we solve

$$\begin{cases} \Delta(\hat{V}_r)_i &=& 0 & \text{in } \hat{\Omega} \cap Bint, \\ (\hat{V}_r)_i &=& 0 & \text{in } \hat{\Omega}/Bint, \\ (\hat{V}_r)_i &=& 0 & \text{on } \Gamma_{Gz} \cup \Gamma_{Sz}, \\ (\hat{V}_r)_i &=& \frac{(\hat{P}_r)_i}{||\hat{P}_i||}\, s_i & \text{on } \hat{\Gamma}_{LG}, \end{cases} \tag{5.3}$$

$$\begin{cases} \Delta(\hat{V}_z)_i &=& 0 & \text{in } \hat{\Omega} \cap Bint, \\ (\hat{V}_z)_i &=& 0 & \text{in } \hat{\Omega}/Bint, \\ (\hat{V}_z)_i &=& 0 & \text{on } \Gamma_0 \cup \hat{\Gamma}_{LS} \cup \hat{\Gamma}_{SG}, \\ (\hat{V}_z)_i &=& \frac{(\hat{P}_z)_i}{||\hat{P}_i||}\, s_i & \text{on } \hat{\Gamma}_{LG}, \end{cases} \tag{5.4}$$

where $\hat{V}_i = ((\hat{V}_r)_i, (\hat{V}_z)_i)^T$, $\hat{P}_i = ((\hat{P}_r)_i, (\hat{P}_z)_i)^T$, and $\|\hat{P}_i\| = [(\hat{P}_r)_i^2 + (\hat{P}_z)_i^2]^{\frac{1}{2}}$.

Let us note that we could have extended this vector field over the whole domain by solving a linear elasticity system.

The shape gradient. We approximate V by V_H (see (5.1)), and we have

$$\frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).V \approx \frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).V_H = \sum_{i=0}^{N-1} \eta_i \frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).V_i.$$

Then, the shape gradient denoted by G^H is the vector

$$\begin{aligned} G^H = (G_i^H)_{i=0,\dots,N-1} &= \left(\left[\frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).V_i \right] \right)_{i=0,\dots,N-1} \\ &= \left(\left[\frac{\partial L_\tau}{\partial \omega}(\Omega, \lambda).(\hat{V}_i \circ \hat{F}_0^{-1}) \right] \right)_{i=0,\dots,N-1}, \end{aligned}$$

where $\Omega = \hat{F}_0(\hat{\Omega})$.

Finally, we have for all $i = 0, \dots, N-1$ (see Corollary 4.2),

$$G_i^H = \frac{dj}{d\omega}(\Omega).(\hat{V}_i \circ \hat{F}_0^{-1}) + \lambda \frac{dc}{d\omega}(\Omega).(\hat{V}_i \circ \hat{F}_0^{-1}) + 2\tau c(\Omega) \frac{dc}{d\omega}(\Omega).(\hat{V}_i \circ \hat{F}_0^{-1}). \tag{5.5}$$

Variables of optimization. Since $\Omega = \hat{F}_0(\hat{\Omega}) = (I + \hat{V})(\hat{\Omega}) \approx (I + \hat{V}_H)(\hat{\Omega})$ with $\hat{V}_H$ defined by (5.2), and $\hat{V}_i$ defined by (5.3), (5.4), the variables of optimization are the N coefficients η_i, $i = 0, \dots, N-1$.

6. Optimization process. As mentioned previously, we solve (3.11), an optimization problem with constraint, using Uzawa's algorithm; see, e.g., [8]. This algorithm requires a descent algorithm which is in the present case BFGS (the quasi Newton method). This gives the following:

- Initially, we set $\eta_i^0 = 0$, $i = 0, \ldots, N-1$; $\lambda_0 = 0$.
- We compute the volume constraint $c(\eta^0)$.
- While the volume constraint ($|c(\eta^{k+1})| > eps1$) is not satisfied,
  - set $\lambda_{k+1} = \lambda_k + \rho\, c(\eta^k)$,
  - compute η_i^{k+1} $i = 0, \ldots, N-1$ such that $L_\tau(\eta^{k+1}, \lambda_{k+1}) < L_\tau(\eta^k, \lambda_{k+1})$ using the BFGS algorithm, and
  - compute the volume constraint $c(\eta^{k+1})$.

Classically, we set $\rho = \tau$; see [8].

The BFGS algorithm is implemented with bounding constraints. The linear search is done using a dichotomic process.

We stop the BFGS algorithm either if $\frac{|j(\eta^{k+2}) - j(\eta^{k+1})|}{j(\eta^{k+1})} < eps2$ or if $\|(G^H)^{k+2}\| < eps3$.

As usual, each call of the algorithm BFGS implies a few calls to the simulator.

The simulator does the following:

- It computes the new shape and the new mesh defined by

$$\Omega = \left(I + \sum_{i=0}^{N-1} \eta_i \hat{V}_i \right) (\hat{\Omega}).$$

- It solves the state equation (3.5) posed in Ω by a P_1-Lagrange finite element method (with or without automatic mesh refinement).
- It computes the augmented Lagrangian L_τ defined by (3.9), with its gradient G^H defined by (5.5), and the volume constraint c defined by (3.10).

The full optimization process is represented in Figure 6.1.

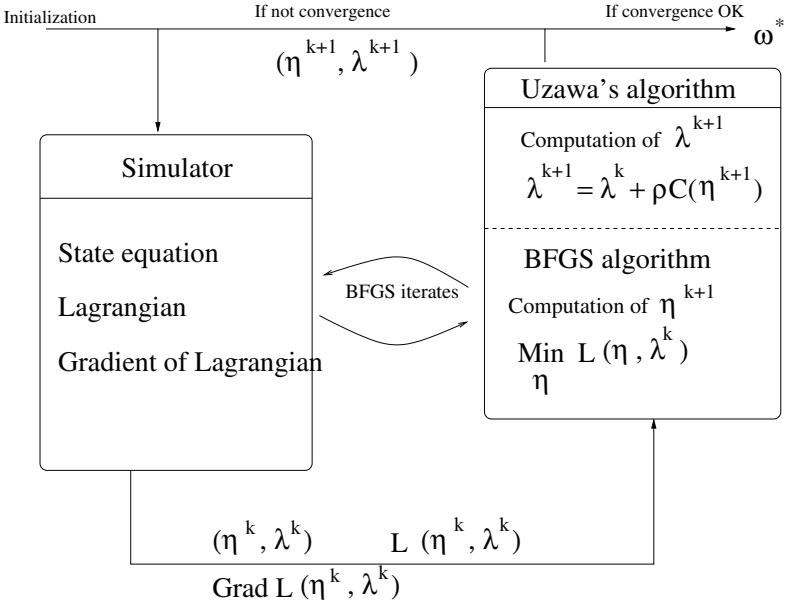

FIG. 6.1. *The optimization process.*

7. Curvature computation. In the next section, we consider the evaluation of the droplet curvature, particularly near the contact line. It was shown in [5] that

the contact angle approaches Young's angle, independently of the applied electrical potential value. Observations show that the curvature is not constant. Then, it would be interesting to see if the present modeling approach allows us to observe such changes of curvature values near the triple point.

Accurately computing the droplet curvature is a difficult task since its interface is a piecewise linear curve, and hence is not C^2 differentiable. In addition, points defining this piecewise linear curve result from the full shape optimal design process and hence may comprise some nonnegligible numerical errors. Thus, we seek to estimate the curvature of an underlying smooth surface.

Computing a discrete surface curvature is a classical (and difficult) problem. Usually in the computer aided geometric design context, surfaces are 3D and triangularized, and the objectives are to smooth the mesh and simplify it, but not to quantify a local variation of curvature; see, e.g., [10].

We are facing the following dilemma. We seek to get rid of numerical errors on the points defining the curve while we try to detect as accurately as possible a local significant variation of curvature.

We do not consider a direct computation by a finite difference method since it is very sensitive to data error. We do not consider a polynomial reconstruction of the underlying smooth surface and then evaluate its curvature, since this leads to inaccurate results and unexpected behavior. Following [9], [14], we consider a local least square approximation and then we evaluate the curvature. In the present algorithm, we consider a second order local Bezier approximation; see [14]. As the numerical tests presented below show, this method filters noise reasonably.

7.1. The algorithm. Given N points $X_i = (r_i, z_i)^T$, $i = 1, \dots, N$ defining the liquid-gas interface, the basic idea is to approximate these data using a local least square approximation by a Bezier curve.

The Bezier curve $\mathcal{C}(t)$ is given by

$$\mathcal{C}(t) = (r(t), z(t))^T = \sum_{j=1}^{M} P_j B_{j-1}^{M-1}(t) \qquad \text{for } t \in [0,1],$$

where $P_j = (\alpha_j, \beta_j)^T \in \mathbb{R}^2$ are the control points and $\{B_j^m(t)\}_{0 \le j \le M-1}$ is the classical Bernstein basis, with $B_j^m \in P_m$, $B_j^m(t) = C_j^m (1-t)^{m-j} t^j$, C_j^m being the binomial coefficients.

We set $M = 3$; hence we consider three points of control P_j and second degree curves.

For an inner point X_i (see Figure 7.1), we compute the least square approximation of the four points $\{X_{i-2}, \dots, X_{i+2}\}$ by Bezier's curve as follows. We minimize

$$J(P_1, P_2, P_3) = \sum_{l=i-2}^{i+2} \left\| \sum_{j=1}^{3} P_j B_{j-1}^2(t_l) - X_l \right\|^2,$$

where $\{t_{i-2} = 0, \dots, t_{i+2} = 1\}$ is a uniform subdivision of $[0,1]$. The unique minimum is computed by solving the corresponding normal equations.

For the extremal point X_1, we consider a Bezier curve approximating the points X_i for $i = 1, \dots, 4$. For X_2, we consider a Bezier curve approximating the points X_i for $i = 1, \dots, 5$.

For the extremal points X_{N-1} and X_N, the principle is similar.

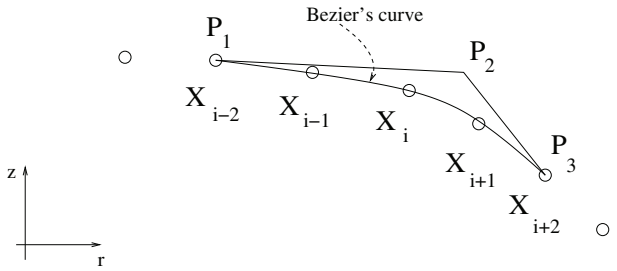

FIG. 7.1. *Inner point X_i. Local least square approximation using Bezier's curve.*

Curvature expression. Once a Bezier curve $\mathcal{C}(t) = (r(t), z(t))^T$ is computed for each point X_i, we evaluate the curvature as follows:

$$\kappa_i \equiv \kappa(t_i) = \frac{r'z" - r"z'}{(r'^2 + z'^2)^{\frac{3}{2}}}(t_i),$$

where (r', z') and (r'', z'') are computed using de Casteljau's algorithm, with t_i being the parameter value related to X_i.

Sensitivity to random noise. Since the control points defining the (optimal) droplet shape result from the full optimization process, they are perturbed by some nonnegligible numerical errors. Hence, we test the robustness of our algorithm to data perturbation below.

We set $N(r, z) = (r'z" - r"z')$ and $D(r, z) = (r'^2 + z'^2)^{\frac{3}{2}}$; hence $\kappa(r, z)(t) = \frac{N(r,z)}{D(r,z)}(t)$. Let δz be a perturbation on the z-coordinate of data X_i, $i = 1, \ldots, N$; then we have

$$\frac{\partial \kappa}{\partial z}(r, z).\delta z = \frac{N(r, \delta z)}{D(r, z)} - 3\ \frac{\kappa(r, z)}{(r'^2 + z'^2)} z'\delta z.$$

This formula expresses the curvature sensitivity to perturbation on z-coordinates. Noise introduced below is a random perturbation on the z-coordinate of data X_i, $i = 1, \ldots, N$. It is a normal distribution with mean zero and variance one.

7.2. Numerical tests. The numerical tests presented below are useful for (i) validating the present algorithm on explicit curves knowing their curvature value (the "exact" curves); and (ii) measuring the computed curvature sensitivity to random perturbation on data.

To this end, we consider an "oscillating curve" (see Figure 7.2), defined by N points as follows:

$$r(s) = (R + \epsilon\cos(a.s))\cos\left(\frac{\pi}{2}s\right), \quad z(s) = (R + \epsilon\cos(a.s))\sin\left(\frac{\pi}{2}s\right),$$

with $s \in [0, 1]$, s discretized by N points similarly to η and $\epsilon = \frac{R}{10}$, $a = 10$, $R = 1$.

The exact curvature of the "oscillating circle" is straightforwardly obtained. This curve presents smooth variations of curvature with changes of sign. If we compare the curvature values computed by the present algorithm and those computed by the second order finite difference scheme directly applied to the N data $X_i = (r_i, z_i)^T$, $i = 1, \ldots, N$, then without noise both lead to similar accuracy—the two methods give a very precise approximation.

However, in the presence of noise, the direct approximation does not give any good approximation. On the contrary, the present algorithm, based on a local least square approximation of the surface by Bezier's curve, leads to a good approximation of the curvature value of the nonperturbed curve.

We present in Figure 7.2 the curvature values obtained with the present algorithm when some noise is introduced. As mentioned above, the noise is defined as a perturbation on the z-coordinate of data X_i, $i = 1, \ldots, N$. Its magnitude is about 0.5%.

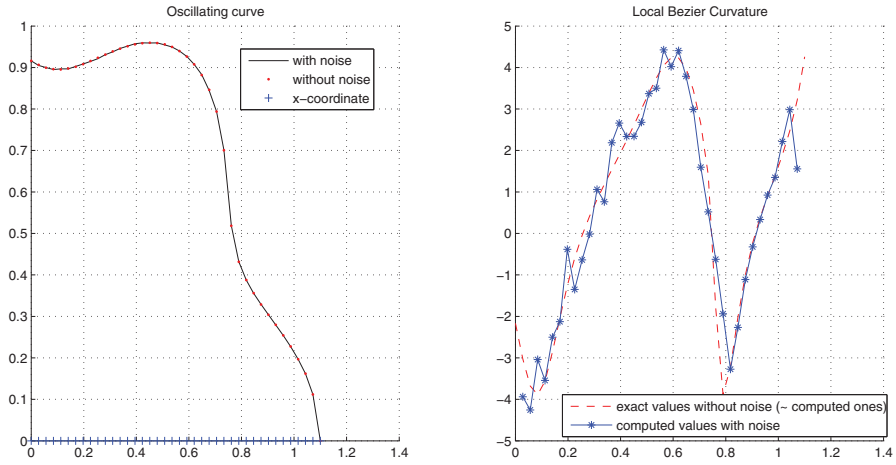

FIG. 7.2. *Left: Oscillating curve. Right: Computed curvature value when noise is introduced.*

8. Numerical results. The full optimization process described in the previous section was implemented in C++. Our software, *ElectroCap* (see [13]), is based on the public C++ finite element library Rheolef [19] and an in-house BFGS algorithm. The mesh generator used is Bamg. For each simulator call, an automatic mesh refinement is used. This mesh refinement is based on the classical a posteriori estimates. We present in Figure 8.2 a typical mesh with the adaptive mesh in the edge.

Numerical data. The numerical data considered are the following:

- the surface tension coefficients (in N/m): $\sigma_{LS} = 2.7\ 10^{-2}$, $\sigma_{LG} = 5\ 10^{-2}$;
- the wetting angle at $u_0 = 0$ (in radians): $\theta_0 = \frac{\pi}{2}$ (hence $\mu = 0$);
- the insulator thickness (in m): $e = 200\ 10^{-6}$;
- the electrical permitivities: $\varepsilon_1 = 2 \times 8.85\ 10^{-12}$ and $\varepsilon_2 = 8.85\ 10^{-12}$;
- the drop volume (in L): $vol = 40\ 10^{-9}$.

We assume that the Bond number α is small; i.e., we neglect the gravitational term. Then, the cost function is (see (3.6), (3.7))

$$j(\omega) = j_{cap}(\omega) - j_{elec}(\omega), \tag{8.1}$$

with

$$j_{cap}(\omega) = \int_{\gamma_{LG}} r ds \quad \text{and} \quad j_{elec}(\omega) = \delta \int_{\omega} \varepsilon r |\nabla u^{\omega}|^2 \, dx,$$

where $j_{cap}(\omega)$ and $j_{elec}(\omega)$ are positive cost functions. The numerical parameters are the following:

- the penalty parameter: $\tau = \rho = 10^{-3}$;
- the convergence parameter of Uzawa's algorithm: $eps1 = 10^{-3}$;
- the convergence parameter of the BFGS algorithm: $eps2 = eps3 = 10^{-3}$;
- the number of control points: $NCP = 50$.

The NCP is defined as follows. If we consider the polar coordinates in the plane, for a droplet of radius R, the N points are equidistributed in θ. Their positions are indicated in Figure 8.4.

Code validation. All components of the code have been validated—the direct problem, the transport of the mesh, and the shape gradient.

The computed shape gradient has been compared with values obtained by a finite difference method using the following approach. For each shape parameter, a finite difference shape derivative is computed using a domain perturbation of magnitude 10^{-4}. The order of magnitude of the relative error obtained between the two approaches is between 10^{-4} and 10^{-6}, depending on the imposed electrical field value u_0.

In order to validate the entire code, we simulate the Lippman approximation by using the code with $u_0 = 0$ V but changing σ_{LS} for each value of u_0 using the formula given by the approximation of the plane capacitor:

$$\sigma_{LS}(u_0) = \sigma_{LS} - \frac{\varepsilon_0 \varepsilon_1}{2e} u_0^2.$$

Thus, theoretically, the contact angle should also be given by the Lippman equation. Numerically we observe a good agreement with the theory. Figure 8.1 shows the value of the contact angle found numerically (the angle of the last mesh triangle) and the theoretical value given by the Lippman equation.

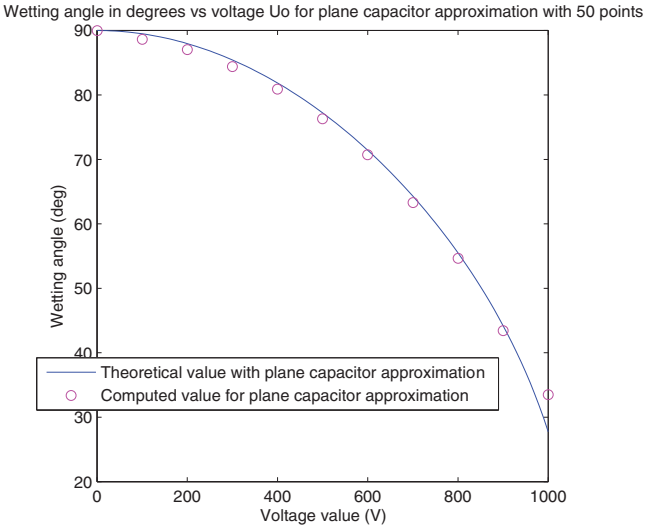

FIG. 8.1. *Plane capacitor approximation.*

Moreover, we compute the curvature for each value of σ_{LS} (which corresponds to a value of u_0). Given a value of u_0 and thus a value of σ_{LS}, we notice that the numerically computed curvature remains constant for each point of the drop. We also obtain for this case a very good agreement with the theory, which contributes to validating the code.

Now we compute the drop shape with the initial model, i.e., by considering σ_{LS} as a constant and by changing values of u_0.

Drop shape and wetting angle. We present in Figure 8.2 the drop shape (with mesh) obtained for $u_0 = 400$ V (left) and a zoom of the refined mesh near the edge (right). As a matter of fact, we use an adaptive mesh refinement near the contact point based on a posteriori estimates. All meshes contain approximately 4000 elements and 2000 vertices. For each computation, the volume constraint is satisfied at less than 0.1%.

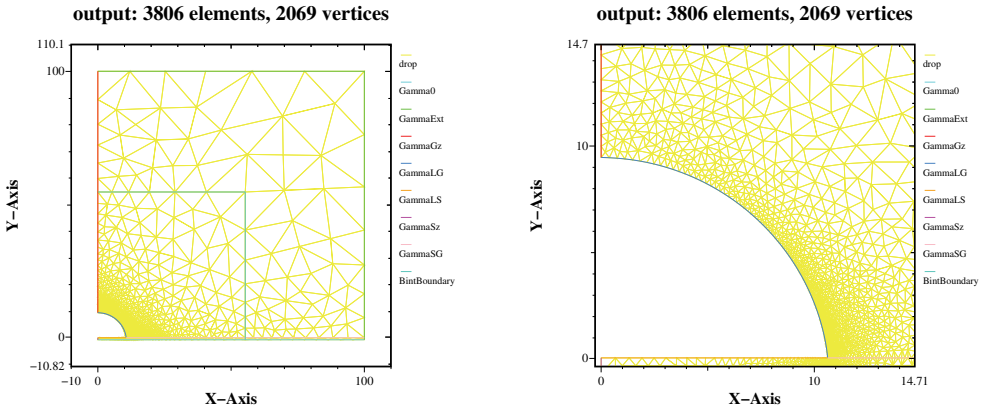

FIG. 8.2. *Left: Shape and mesh for* $u_0 = 400$ *V. Right: Zoom near the drop.*

We present the cost function, the augmented Lagrangian, and its gradient as a function of the iteration number for $u_0 = 400$ V in Figure 8.3. The behavior of the algorithm for other values of u_0 is similar.

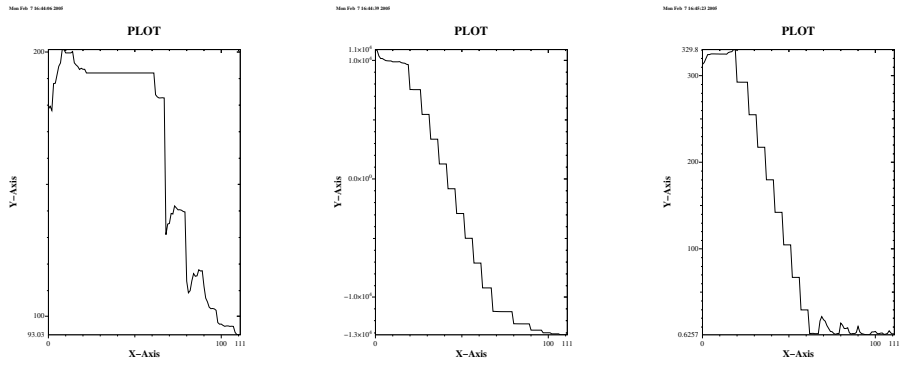

FIG. 8.3. $u_0 = 400$ *V. Left. Cost function* j *versus iterations. Middle. Augmented Lagrangian* L_τ*. Right. Gradient of* L_τ*.*

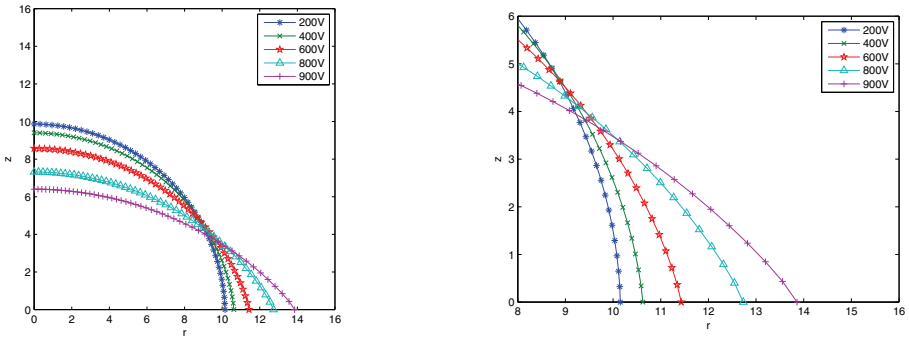

FIG. 8.4. *Droplet surfaces for different u_0 values. At right: zoom near the triple point.*

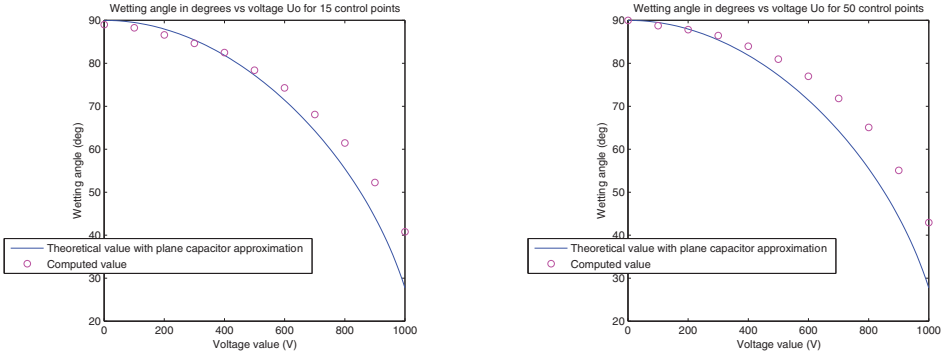

FIG. 8.5. *Wetting angle. Computed values and Lippman's equation predictions. Left. With $NCP = 15$. Right. With $NCP = 50$.*

We present in Figure 8.4 the drop shapes obtained in the function of u_0.

We present in Figure 8.5 (left and right) the wetting angle values in the function of u_0. In both figures (left and right), we plot the computed values (the angle of the last mesh triangle) and values predicted by the Lippman equation. On the left, plotted values are obtained using 15 control points ($NCP = 15$); on the right, plotted values are obtained using 50 control points ($NCP = 50$) (both with similar finite element meshes).

Let us recall that experimental results correspond to the Lippman equation up to a critical electrical potential u_{cr} (for the present case, the observed critical value $u_{cr} \approx 700$ V). For $u_0 > u_{cr}$, experimental results show a saturation of the wetting angle (locking phenomenon); see, e.g., [21]. As mentioned previously, the explanation of this locking phenomenon is still poorly understood by physicists. For $u_0 \approx 1050$ V, the Lippman equation predicts a total spreading of the drop on the substrate (the wetting angle vanishes).

With the present numerical model and with $NCP = 15$, we obtain a good agreement with the Lippman equation for $u_0 < 500$ V. For higher u_0 values, we do not model the angle saturation, but we observe that the contact angle is higher than the predicted value for the plane capacitor approximation.

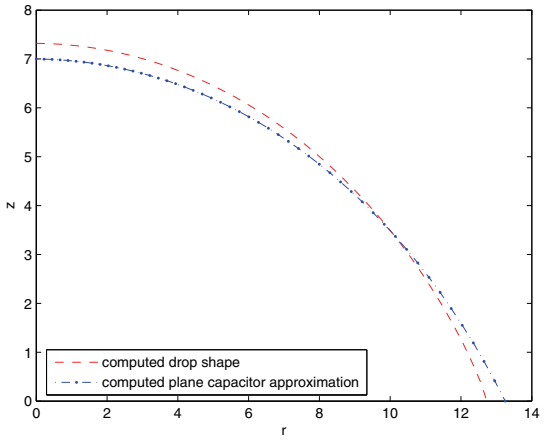

FIG. 8.6. *Computed shape compared to plane capacitor approximation shape for* $u_0 = 800$ *V.*

When increasing the number of control points to $NCP = 50$, we still obtain a good agreement with the Lippman equation for $u_0 < 400$ V. As with 15 points, we notice that the computed values are higher than the predicted values for the plane capacitor approximation. Moreover, the angle values computed with 50 points are higher than those obtained for 15 points for $u_0 > 500$ V.

Also, we compare the drop shape obtained to those obtained using the software but "forcing" the Lippman approximation (i.e., by changing σ_{LS} for each u_0 value). In Figure 8.6 the result for a drop at 400 V is presented. We again find that the wetting angle of the computed shape is higher than the Lippman predicted value.

Let us clarify that we did not manage to increase the NCP because of the well-known instability of the shape optimal design algorithms. As a matter of fact, shape optimal design algorithms become unstable when the control point density becomes similar to the finite element point density.

In summary, with the present model, we do not manage to properly simulate the locking phenomenon, but we do observe an overestimate of the Lippman predictions; this overestimate becomes more important when using a higher control point density.

Curvature. We use the algorithm described in the previous section; see also [14]. For all the computations we performed, the droplet shapes obtained had a constant curvature everywhere but in the vicinity of the triple point. In Figure 8.7, we present as an example (here $u_0 = 800$ V) the computed curvature at each control point. The results are presented with 15, 30, and 50 points, respectively.

In Figure 8.8, we present the curvature values for different electrical potential u_0 values with 50 control points (with curvature values corresponding to those in Figure 8.7, but for different u_0 values). In Figure 8.9, we present the gradient of the solution, i.e., the electric field.

For all computations we performed, the curvatures behave as those shown in Figure 8.8. Thus, we can make the following three main remarks:

- For the curvature, the results are more accurate with 50 points than with 15 or 30 points. With 15 or 30 points, the behavior of the curvature near the triple point appears to be less clear than with 50 points. This is due to the too small number of points near the triple line.

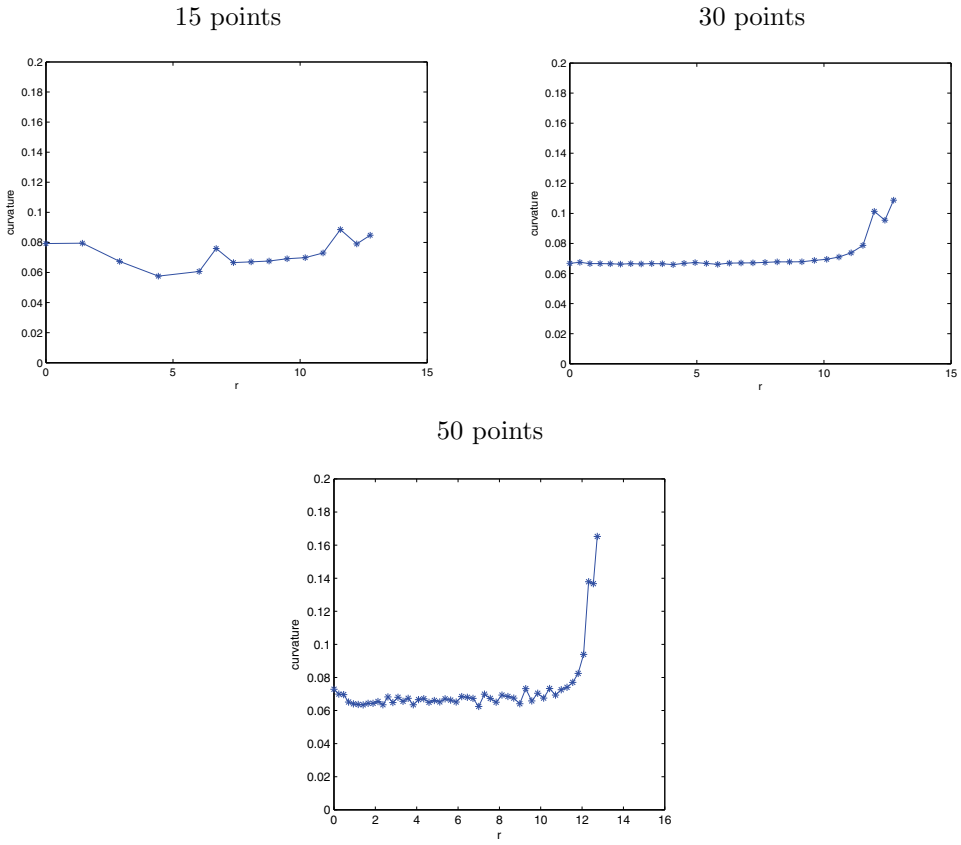

FIG. 8.7. *Curvature values at* $u_0 = 800\,V$ *for* 15, 30, *and* 50 *points, respectively.*

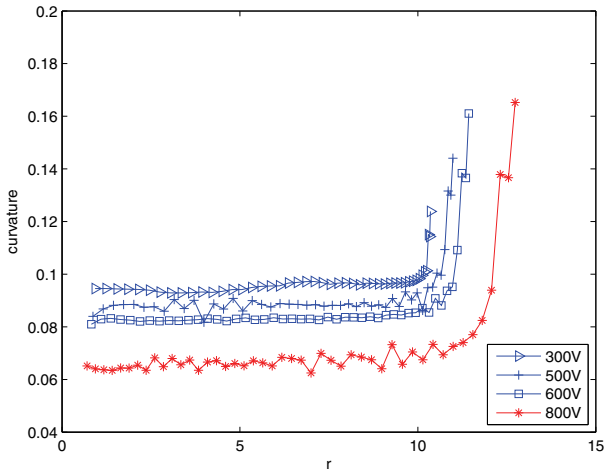

FIG. 8.8. *Curvature of the drop for several* u_0 *values with* 50 *control points.*

- For a given potential u_0, the curvature remains constant until we approach the triple line, where the curvature increases. We can see that the curvature is higher near the triple point than it is further away from it. (See Figure 8.7 for the case at 800 V. For other voltage the curvature has the same behavior; see Figure 8.8.)
- If we look at the evolution of the curvature for an increasing potential u_0, we notice the following:
  - The value of the curvature far from the triple line is constant and decreases when u_0 increases.
  - The curvature near the triple point increases, when u_0 increases. The fact that, with an increasing u_0, the curvature far from the triple point decreases is in accordance with the fact that, globally, the drop should be a portion of a sphere with an increasing radius as u_0 increases. We note that the curvature increases near the triple line; this is in accordance with the fact that the contact angle is higher than the Lippman predicted value.

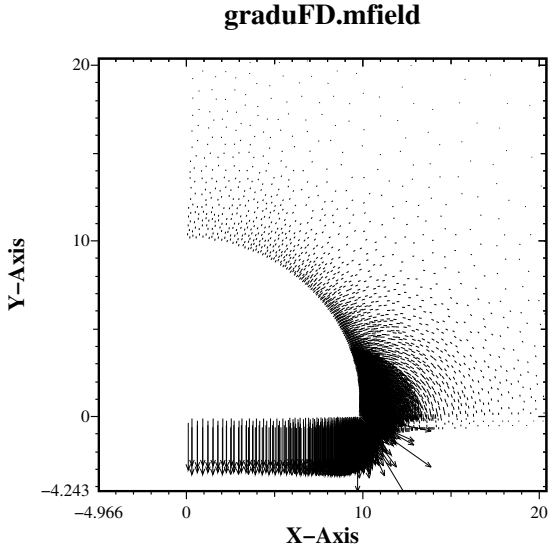

FIG. 8.9. *External electric field* $\vec{E} = \vec{\nabla} u$ *at* $u_0 = 400$ *V (zoom around the droplet).*

9. Conclusion. We have detailed and implemented a general approach for modeling the electrowetting process. The drop shape is computed as a minimum of the total energy. Our model is based on the shape optimal design methods. We test our model and software by including in the model the plane capacitor approximation (i.e., using the software with $u_0 = 0$ V and changing the value of σ_{LS} for each value of the potential). We obtain in this case an excellent agreement with the plane capacitor approximation, which contributes to validating the approach. Then, we compare numerical results obtained classically, that is to say, by changing the value of u_0, with the theoretical values for the plane capacitor approximation. In this case, the com-

puted shapes and angles are not in agreement with this theory for a voltage higher than 300 V.

Although we did not properly obtain the locking phenomenon, the drop shape obtained deviates from the predicted shape as in [16]. Also, we did not manage to observe that the contact angle remains constant; instead, the computed contact angle values are higher than those predicted by Lippman's equation. Moreover, this overestimate becomes more important when using a higher control point density.

In other respects, we compute the curvature of the droplets. These values are globally constant except in the vicinity of the contact point where the computed curvature increases sharply. These results are in accordance with experimental results obtained in [3] and [5], which noted this increase of the curvature near the triple line.

Finally, in order to properly obtain the locking phenomenon and Young's angle at the triple line as in [5], [16], [3], further investigations based on extra singular basis functions to the finite element spaces are in progress.

Acknowledgments. The authors would like to thank C. Quilliet and M. Bienia from the Laboratory of Spectrometry Physics (LSP) in Grenoble, France, for the numerous and fruitful discussions; also, the referees for their constructive comments.

REFERENCES

[1] B. Berge, *Electrocapillarité et mouillage de films isolants par l'eau*, C. R. Acad. Sci. Sér. 2, 317 (1993), pp. 157–163.

[2] M. Bienia, C. Quilliet, and M. Vallade, *Modification of drop shape controlled by electrowetting*, Langmuir, 19 (2003), pp. 9328–9333.

[3] M. Bienia, M. Vallade, C. Quilliet, and F. Mugele, *Electrical-field-induced curvature increase on a drop of conducting liquid*, Europhys. Lett., 74 (2006), pp. 103–109.

[4] B. Bouchereau, *Modélisation et simulation numérique de l'electro-mouillage*, Ph.D. thesis, University of Grenoble I, Grenoble, France, 1997.

[5] J. Buehrle, S. Herminghaus, and F. Mugele, *Interface profiles near three-phase contact lines in electric fields*, Phys. Rev. Lett., 91 (2003), 086101.

[6] J. Céa, *Conception optimale ou identification de formes: Calcul rapide de la dérivée directionnelle de la fonction coût*, M2AN Math. Model. Numer. Anal., 20 (1986), pp. 371–402.

[7] D. Chenais, J. Monnier, and J. P. Vila, *A shape optimal design problem with convective and radiative heat transfer. Analysis and implementation*, J. Optim. Theory Appl., 110 (2001), pp. 75–117.

[8] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary Value Problems*, North–Holland, Amsterdam, 1983.

[9] B. Hamann, *Curvature approximation for triangulated surfaces*, in Geometric Modelling, G. Farin, H. Hagin, and H. N. Hemeier, eds., Springer-Verlag, Vienna, 1993, pp. 139–153.

[10] M. Meyer, M. Desbrun, P. Schroder, and A. H. Barr, *Discrete differential-geometry operators for triangulated 2-manifolds*, in Visualization and Mathematics III, H.-C. Hege and K. Polthier, eds., Springer-Verlag, Berlin, 2003, pp. 35–57.

[11] B. Mohammadi and O. Pironneau, *Applied Shape Optimization for Fluids*, Oxford University Press, Oxford, 2001.

[12] J. Monnier, *Shape sensitivities in a Navier-Stokes flow with convective and grey bodies radiative thermal transfer*, Optimal Control Appl. Methods, 24 (2003), pp. 237–256.

[13] J. Monnier and P. Chow-Wing-Bom, *ElectroCap: A Shape Inverse Model for an Electro-Capillary Process*, Report INRIA RR-5617, INRIA, Grenoble, France, 2005.

[14] J. Monnier, A. Benselama, and I. Cotoi, *Flow patterns in the vicinity of triple line dynamics arising from a local surface tension model*, Int. J. Multiscale Comput. Eng., 5 (2007), pp. 417–434.

[15] F. Murat and J. Simon, *Sur le Contrôle par un Domaine Géométrique*, Laboratory of Numerical Analysis, University Paris VI, Paris, 1976.

[16] A. G. Papathanasiou and A. G. Boudouvis, *Manifestation of the connection between dielectric breakdown strength and contact angle saturation in electrowetting*, Appl. Phys. Lett., 86 (2005), 164102.

[17] V. Peykov, A. Quinn, and J. Ralston, *Electrowetting: A model for contact-angle saturation*, Colloid Polym. Sci., 278 (2000), pp. 789–793.
[18] C. Quilliet and B. Berge, *Electrowetting: A recent outbreak*, Curr. Opin. Colloid Interface Sci., 6 (2001), pp. 34–39.
[19] P. Saramito, N. Roquet, and J. Etienne, *Rheolef Home Page*, http://www-lmc.imag.fr/lmc-edp/Pierre.Saramito/rheolef/, LMC-IMAG, 2002.
[20] B. Shapiro, H. Moon, R. L. Garrell, and C.-J. Kim, *Equilibrium behavior of sessile drops under surface tension, applied external fields and material variations*, J. Appl. Phys., 93 (2003), pp. 5794–5811.
[21] M. Vallet, M. Vallade, and B. Berge, *Limiting phenomena for the spreading of water on polymer films by electrowetting*, Eur. Phys. J. B, 11 (1999), pp. 583–591.
[22] H. J. J. Verheijen and M. W. J. Prins, *Reversible electrowetting and trapping of charge: Model and experiments*, Langmuir, 15 (1999), pp. 6616–6620.