



HAL
open science

Numerical modeling of electrowetting by a shape inverse approach

Jerome Monnier, Patrick Witomski, Patrick Chow-Wing-Bom, Claire Scheid

► **To cite this version:**

Jerome Monnier, Patrick Witomski, Patrick Chow-Wing-Bom, Claire Scheid. Numerical modeling of electrowetting by a shape inverse approach. *SIAM Journal on Applied Mathematics*, 2009, 69 (5), pp.1477-1500. 10.1137/050624340 . hal-00380673

HAL Id: hal-00380673

<https://hal.science/hal-00380673>

Submitted on 4 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NUMERICAL MODELLING 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 a solid surface. The model aims to seek a drop shape minimizing its total energy (capillary, electrostatic and gravitational). We derive the equations and the shape gradient, then 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 but in a 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

1. Introduction. Electrowetting can be defined as a tool for spreading liquid droplet (e.g. water) on hydrophobic solid surfaces (e.g. polymer film). This is quite a recent technique, [1], which holds very attractive properties for manipulation of tiny liquid volumes, as it is done in biotechnologies for example. 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 to describe the droplet is the wetting angle.

Few articles treat of the experimental aspects of electrowetting and present some analytical analysis, see e.g. [1], [21], [2] and references therein. A property of electrowetting still badly understood by physicists is the contact angle saturation. Several mechanisms were proposed in [21], [22], [17], [20] to explain it. 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 higher values, instabilities of the contact line liquid-solid-gas can appear. Few hypothesis have been made to explain the saturation phenomena. Let us cite for example the air ionization, [21], or electrostatic effects near the wetting line, [4]. This limiting phenomena is still under investigation and the full modelling 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's angle (obtained when the potential is null). Also, they observe that the curvature near the contact line increases while the potential increases.

In this study, we present a mathematical approach to model and compute numerically 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 modelling is general in the sense that we do not make any assumption on the drop shape. The equations

*University of Grenoble and INRIA. Laboratory LJK (Moise project-team), 38041 Grenoble cedex 9, France (monnier@imag.fr)

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 2D axisymmetric but the method remains valid for 3D shapes. 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 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 a deviation of the shape of the drop from Lippman's predictions. In other respects, we focus on the curvature values of the droplet interface. The computed curvature is constant everywhere but in a vicinity of the contact point. If refining the surface representation near the contact line, we observe the increasing 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 solution of the external partial differential equation. The liquid volume is given and constant, it is considered as an equality constraint. Finally, the problem consists to find a min-max solution (saddle point) of an augmented lagrangian, [8]. Numerically, the solution is computed using the 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, Theorem 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 (P1) 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.

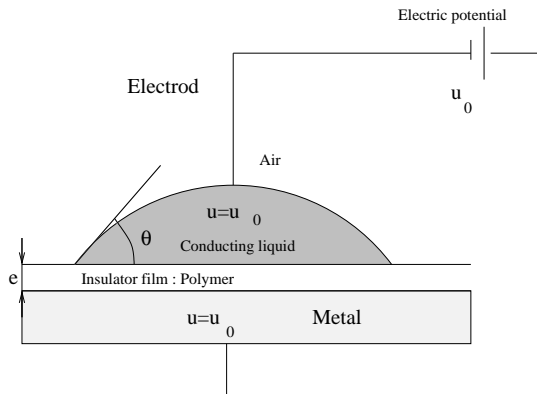
Let us consider the electrowetting process presented in Fig. 2.1.

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, the Young's equation gives:

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

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


 FIG. 2.1. *Electrowetting process*

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

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

where e is the insulator thickness, ε_0 and ε_1 are the dielectric constant. Also, we have, [1]:

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

This last equation is called Lippmann's equation also.

Let us note that this law forecasts a total spreading when the potential increases. However, if u_0 is greater a critical value u_{cr} , physicists observe a locking phenomena 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 compute numerically the liquid drop shape for u_0 lower than such 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 electro-wetting process described in the previous section as a shape inverse problem.

Assumption

- (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 wets partially the polymer (the spreading coefficient is negative).

Notations (see Fig. 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.

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.

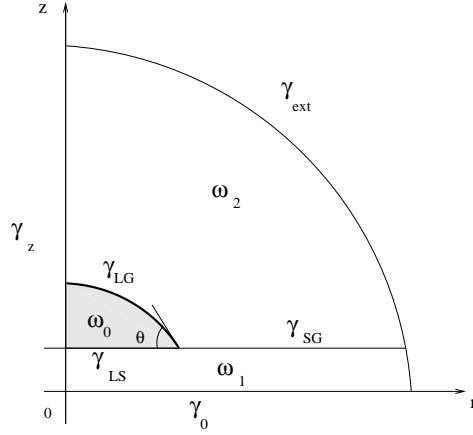


FIG. 3.1. 2D axisymmetric droplet (filled in black). Domains and boundaries notations

The questions we answer numerically are the following. Given the electric 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?

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, see e.g. [2]:

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

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 , $i = 1, 2$, 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:

$$\left\{ \begin{array}{l} \text{Find } \omega_0^* \text{ such that:} \\ \mathcal{E}_{\omega_0^*} = \min_{(\omega_0; \int_{\omega_0} dx = vol)} \mathcal{E}_{\omega_0} \end{array} \right.$$

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:

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

with the following Dirichlet boundary conditions:

$$(3.2) \quad \left\{ \begin{array}{ll} u_1 = u_0 & \text{on } \gamma_{LG} \\ u_2 = u_0 & \text{on } \gamma_{LS} \\ u_2 = 0 & \text{on } \gamma_0 \end{array} \right.$$

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

$$(3.3) \quad \left\{ \begin{array}{ll} u_1 = u_2 & \text{on } \gamma_{SG} \\ \varepsilon_1 \nabla u_1 n_1 = -\varepsilon_2 \nabla u_2 n_2 & \text{on } \gamma_{SG} \end{array} \right.$$

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

$$(3.4) \quad \varepsilon_i \nabla u_i n_i = 0 \quad \text{on } \gamma_{ext}^i, \quad i = 1, 2$$

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:

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

where

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

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

In vertu of Lax-Milgram theorem, state equation (3.5) has one and only solution $u^\omega \in X_t(\omega)$.

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

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

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:

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

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

$$(3.8) \quad \begin{cases} \text{Find } \omega^* \in \mathcal{D} \text{ such that} \\ j(\omega^*) = \min_{(\omega; \int_{\omega_0} r dx = vol/2\pi)} j(\omega) \end{cases}$$

Let us point out that the variable is not the whole domain ω but more precisely the liquid-gas interface γ_{LG} , see Fig. 3.1.

We assume 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 $L_\tau : \mathcal{D} \times \mathbb{R} \rightarrow \mathbb{R}$, defined by, see e.g. [8]:

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

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

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

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:

$$(3.11) \quad \begin{cases} \text{Find } (\omega^*, \lambda^*) \in \mathcal{D} \times \mathbb{R} \text{ such that} \\ L_\tau(\omega^*, \lambda^*) = \min_\omega \max_\lambda L_\tau(\omega, \lambda) \end{cases}$$

We will solve (3.11) using the classical Uzawa's algorithm, see e.g. [8]. This algorithm uses a gradient type algorithm (BFGS) which requires 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 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 then we discretize them, thus obtaining the discretized continuous gradient; ii) we discretize the equations then we differentiate them, thus obtaining the discrete gradient; iii) we differentiate directly the direct code (typically, using automatic differentiation). In the present study, we follow the 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 to prove all differentiabilitys required. These derivatives are discretized in next section leading to shape gradients.

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

The 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 (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, regularity required to well define the (volume and surface) transported integrals. 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: $\hat{\Omega} = \Omega_1 \cup \Omega_2 \cup \hat{\Gamma}_{SG}$. Ω_1 represents the solid part and Ω_2 represents the gas part.

We distinguish the variable part of $\hat{\Omega}$ from its fixed part, see Fig. 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 Fig. 4.1.

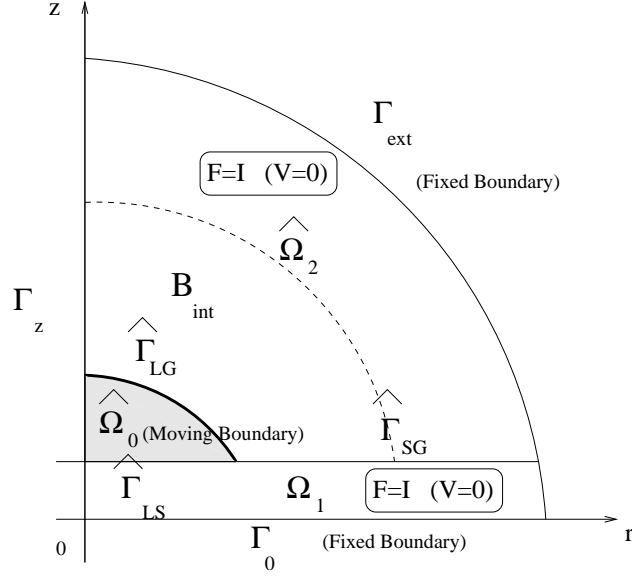
We set the function space:

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

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 domains space \mathcal{D} as follows

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

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

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 homeomorphisms space defined in Ω , Fig. 4.2: $\mathcal{F} = \{F, F = \hat{F} \circ \hat{F}_0^{-1}, \hat{F} \in \hat{\mathcal{F}}\}$, and its affine subspace: $\mathcal{F}_0 = \{F, F = \hat{F} \circ \hat{F}_0^{-1}, \hat{F} \in \hat{\mathcal{F}}_0\}$.

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 $\hat{\Omega} \setminus B_{int}$.

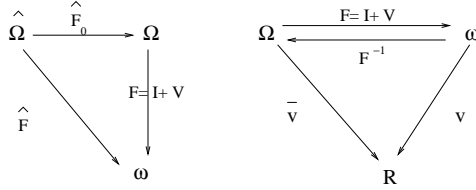


FIG. 4.2. Change of variables

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

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

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

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

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

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

with: $\operatorname{div}_{\Gamma} V = (\operatorname{div}(V) - \langle n, {}^T DV n \rangle)$, n is 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 $V \in \mathcal{C}^1(\bar{\Omega}, \mathbb{R}^2)$,

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

Proof. It is done in 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 defined 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:

$$\begin{aligned} \bar{a}(F; \bar{u}, \bar{v}) &= a_{F(\Omega)}(\bar{u} \circ F^{-1}, \bar{v} \circ F^{-1}) = a_{\omega}(u, v) \\ &= \int_{\Omega} \bar{\varepsilon} \bar{r} \langle ({}^T DF^{-1} \circ F) \nabla \bar{u}, ({}^T DF^{-1} \circ F) \nabla \bar{v} \rangle |det DF| d\bar{x} \end{aligned}$$

with: $\bar{u} = u \circ F$, $\bar{v} = v \circ F$, $\bar{x} = x \circ F$ and $\bar{\varepsilon} = \varepsilon \circ F$, see Fig. 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 respect, the Dirichlet's data u_0 is constant, hence $u_0 = u_0 \circ F$. Then, since state equation (3.5) has an unique solution u^ω , the transported state equation:

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

has an 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: $\bar{j}(F) = \bar{\mathcal{E}}(F; \bar{u}^F)$,

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

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

Also, we define:

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

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 $F \mapsto \bar{u}^F : \mathcal{F}_0 \rightarrow X_t(\Omega)$ in a neighborhood \mathcal{V}_I of I .

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

Step 3. Expression of the exact differential.

By definition, we have: $\frac{d\bar{j}}{d\omega}(\Omega) \cdot V = \frac{d\bar{j}}{dF}(I) \cdot V$, $\forall V \in C^1(\bar{\Omega}, \mathbb{R}^2)$.

Then, using the classical adjoint technique, we have:

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

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:

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

Hence $p^\Omega \in X_0(\Omega)$ and $a_\Omega(p^\Omega, v) = 0 \quad \forall v \in X_0(\Omega)$. Therefore: $p^\Omega = 0$.

Hence,

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

Using (4.6) and the classical expression of the derivatives of $|det(DF)|$, $(DF^{-1} \circ F)$ and $(\|{}^T DF^{-1}.n\|_{\mathbb{R}^2})$ -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

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 $V \in \mathcal{C}^1(\bar{\Omega}, \mathbb{R}^2)$,*

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

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 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, $\omega = \cup_{T \in \mathcal{T}_h} T$. 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; Shape parameters.. Let $\hat{\Omega}$ be an open set of reference; typically $\hat{\Omega}$ is a quarter of a disk, see Fig. 4.1. The domain of reference $\hat{\Omega}$ is defined using a parametric function:

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

where $\{s_i(t)\}_{i=0..N-1}$ 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:

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

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

Initially, these points define $\hat{\Gamma}_{LG}$ as follows, see Fig. 5:

$$\begin{aligned} \hat{P}_i &= (0, R)^T \\ \hat{P}_i &= (R \cos(\frac{(N-1-i)\pi}{2(N-1)}), R \sin(\frac{(N-1-i)\pi}{2(N-1)})^T \quad i = 2..N-1 \\ \hat{P}_1 &= (R \cos(\frac{(N-2)\pi}{2(N-1)})/2, R)^T \end{aligned}$$

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

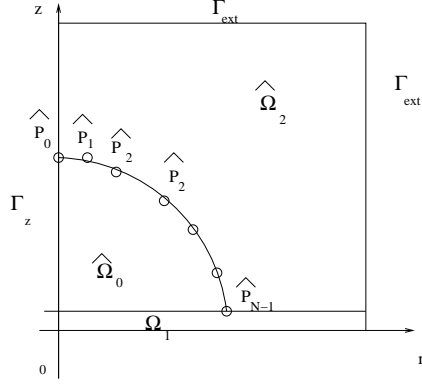


FIG. 5.1. Reference domain. Parametrization.

The shape deformation space.. Let us discretize the shape deformation V , $V \in C^1(\hat{\Omega}, \mathbb{R}^2)$. 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 $C^1(\hat{\Omega}, \mathbb{R}^2)$ by \hat{S}_H the vectorial space spanned by $\{\hat{V}_i\}_{i=0..N-1}$:

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

where the set of vectors $\{\hat{V}_i\}_{i=0..N-1}$ 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:

$$(5.1) \quad V_H = \sum_{i=0}^{N-1} \eta_i V_i$$

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

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

$$(5.2) \quad \hat{V}_H = \sum_{i=0}^{N-1} \eta_i \hat{V}_i$$

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

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

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

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

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

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:

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

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

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

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

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

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

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 (quasi-Newton method). This gives:

- Initially, we set: $\eta_i^0 = 0$, $i = 0 \dots N - 1$; $\lambda_0 = 0$.
- We compute the volume constraint $c(\eta^0)$.
- While the volume constraint ($|c(\eta^{k+1})| > eps1$):
 - set $\lambda_{k+1} = \lambda_k + \rho c(\eta^k)$
 - compute η_i^{k+1} $i = 0 \dots N - 1$, such that $L_\tau(\eta^{k+1}, \lambda_{k+1}) < L_\tau(\eta^k, \lambda_{k+1})$ using BFGS algorithm
 - 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 $\frac{|j(\eta^{k+2}) - j(\eta^{k+1})|}{j(\eta^{k+1})} < \text{eps2}$ or if $\|(G^H)^{k+2}\| < \text{eps3}$.

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

The simulator does the following:

- it computes the new shape and the new mesh defined by:

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

- 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), its gradient G^H defined by (5.5) and the volume constraint c defined by (3.10).

The full optimization process is represented in Fig. 6.1.

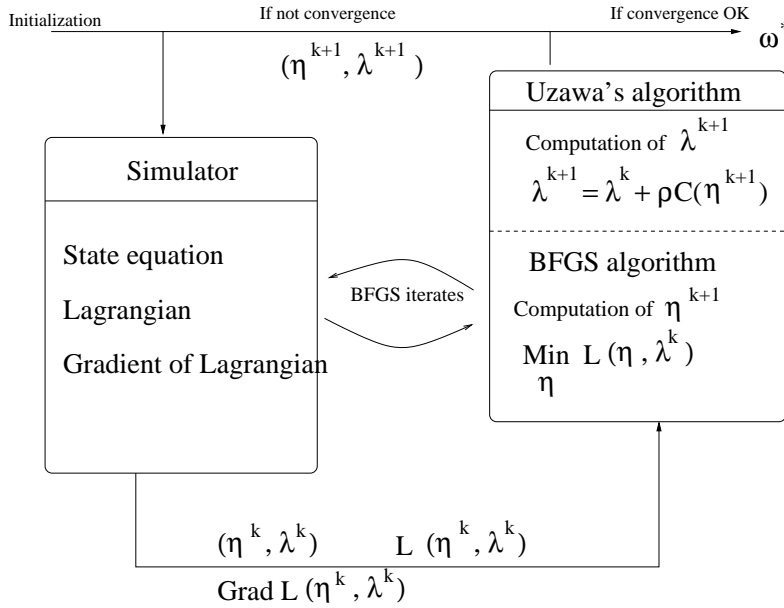


FIG. 6.1. *The optimization process*

7. Curvature Computation. In the next section, we consider the evaluation of the droplet curvature, particularly near the contact line. It has been 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 modelling approach allows to observe such change of curvature values near the triple point.

Computing accurately the droplet curvature is a difficult task since its interface is a piecewise linear curve, hence not C^2 differentiable. In addition, points defining this piecewise linear curve result from the full shape optimal design process, hence may comprise some non-negligible numerical errors.

Then, 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 CAGD context, surfaces are 3D and triangularized, and the objectives are to smooth the mesh, to 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 accurate 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 then evaluate its curvature, since it leads to inaccurate results and unexpected behavior. Following [9, 14], we consider a local least-square approximation 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 $X_i = (r_i, z_i)^T$, $i = 1..N$, defining the liquid-gas interface, the basic idea is to approximate these data using a local least-square approximation by a Bezier's curve.

The Bezier's curve $\mathcal{C}(t)$ is given by:

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

where $P_j = (\alpha_j, \beta_j)^T \in \mathbb{R}^2$ are the control points and $\{B_j^m(t)\}_{0 \leq j \leq M-1}$ is the classical Bernstein's basis, $B_j^m \in P_m$, $B_j^m(t) = C_j^m (1-t)^{m-j} t^j$, C_j^m being the binomial coefficients.

We set $M = 3$, hence we consider 3 points of control P_j and second degree curves. For an inner point X_i , see Fig. 7.1, we compute the least square approximation of the 4 points $\{X_{i-2}, \dots, X_{i+2}\}$ by a 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 an uniform subdivision of $[0, 1]$. The unique minimum is computed by solving the corresponding normal equations.

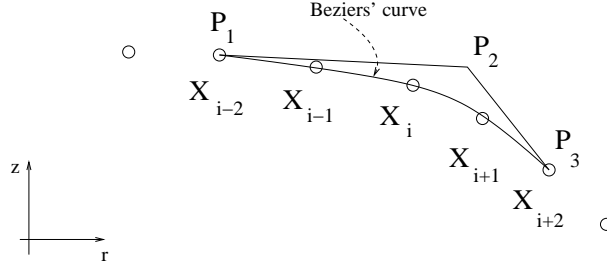


FIG. 7.1. Inner point X_i . Local least square approximation using Bezier's curve.

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

Curvature expression. Once a Bezier's 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, t_i being the parameter value related to X_i .

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

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 z-coordinate of data X_i , $i = 1..N$, then we have:

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

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

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

To this end, we consider an "oscillating-curve", Fig. 7.2, defined by N points as follows:

$r(s) = (R + \epsilon \cos(a \cdot s)) \cos(\frac{\pi}{2}s)$, $z(s) = (R + \epsilon \cos(a \cdot s)) \sin(\frac{\pi}{2}s)$ 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 change 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 $X_i = (r_i, z_i)^T$, $i = 1..N$, then without noise both lead to a similar accuracy: the two methods give a very precise approximation.

But in the presence of noise, the direct approximation does not give any good approximation. At 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 non-perturbed curve.

We present in Fig. 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 , $i = 1..N$. Its magnitude is about 0.5%.

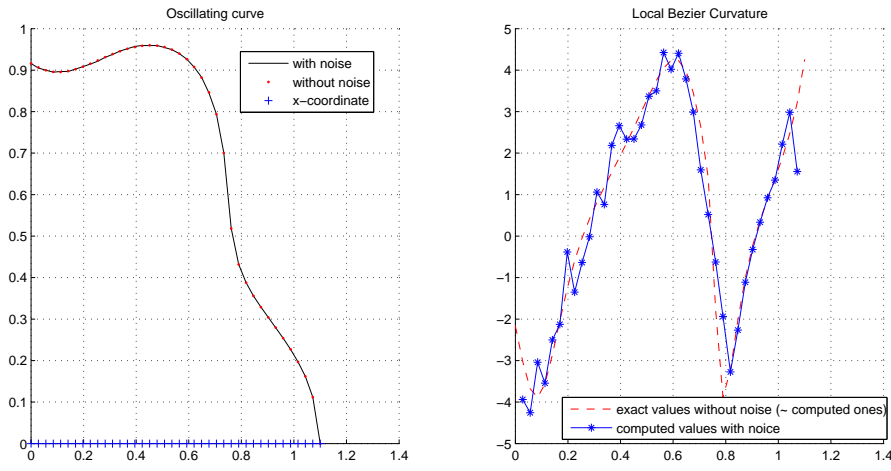


FIG. 7.2. Left: Oscillating curve. Right: Computed curvature value when noise is introduced.

8. Numerical Results. The full optimization process described in the previous section has been 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 Fig. 8.2 a typical mesh with the adaptive mesh in the edge.

Numerical data.. Numerical data considered are the following:

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

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

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

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. 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 BFGS algorithm: $eps2 = eps3 = 10^{-3}$.
- The number of control points: $NCP = 50$.

The *NCP* control points are 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 Fig. 8.4.

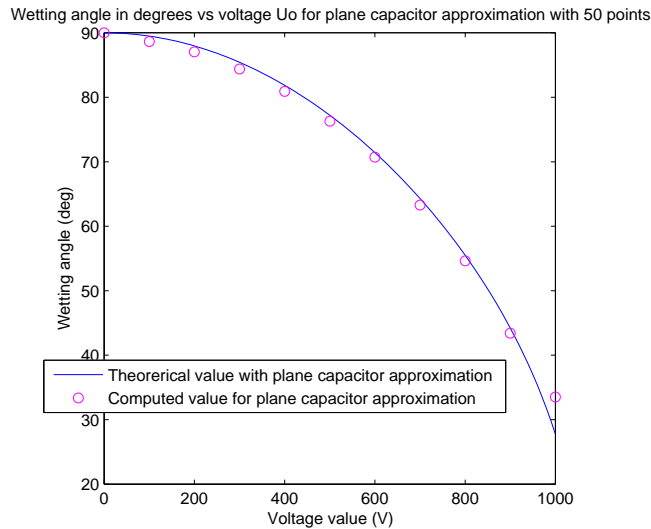
Code validation. All components of the code has been validated: direct problem, transport of mesh and shape gradient.

The computed shape gradient have 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 $10^{-4} - 10^{-6}$, 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 $u_0 = 0V$, but changing σ_{LS} for each values 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 be also given by Lippman equation. Numerically we observe a good agreement with the theory. Fig.8.1 shows the value of the contact angle found numerically (angle of the last mesh triangle) and the theoretical value given by Lippman equation.


 FIG. 8.1. *Plane capacitor approximation*

Moreover, we compute the curvature for each values of σ_{LS} (which correspond to a value of u_0). Given a value of u_0 and so 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 validate 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 Fig. 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 approximatively 4000 elements and 2000 vertices.

For each computation, the volume constraint is satisfied at less than 0.1%.

We present the cost function, the augmented lagrangian and its gradient as a function of iteration number for $u_0 = 400$ V in Fig. 8. The behavior of the algorithm for other values of u_0 is similar.

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

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

Let us recall that experimental results correspond to the Lippman equation until 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

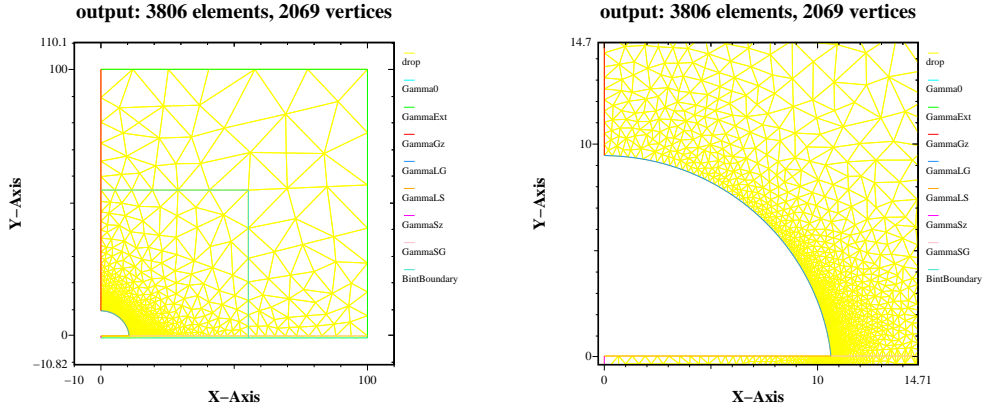


FIG. 8.2. Left, Shape and mesh for $u_0 = 400$ V; Right, Zoom near the drop.

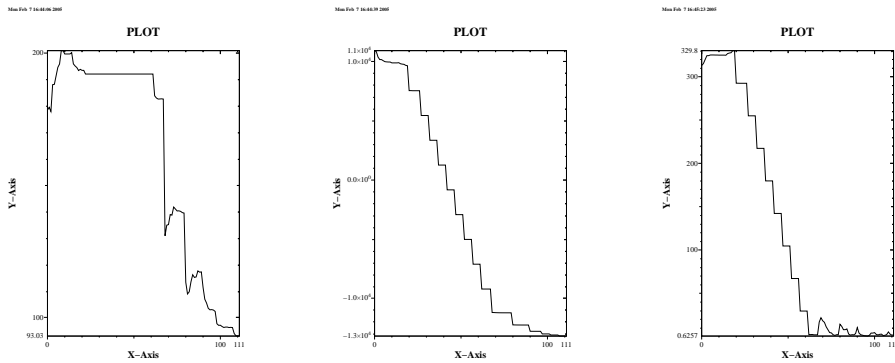


FIG. 8.3. $u_0 = 400$ V. Left. Cost function j vs iterations Middle. Augmented Lagrangian L_τ . Right. Gradient of L_τ .

angle (locking phenomena), see e.g. [21]. As mentioned previously, the explanation of this locking phenomena is still badly 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 $NPC = 15$, 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.

When increasing the number of control points to $NPC = 50$, we still obtain a good

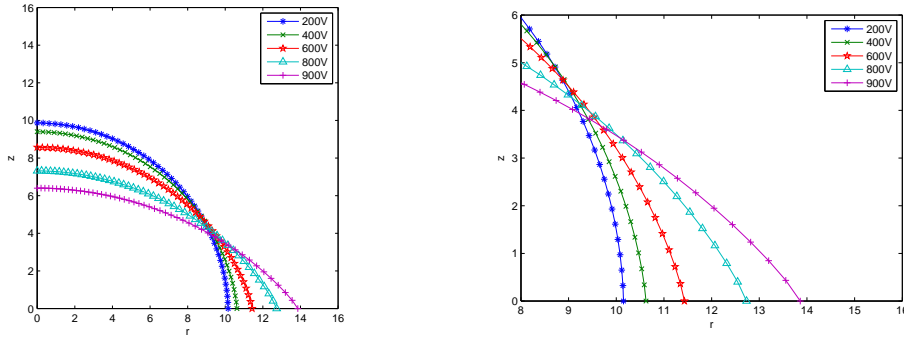


FIG. 8.4. Droplet surfaces for different u_0 values. At right: zoom near the triple point.

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 angles values computed with 50 points are higher than those obtained for 15 points for $u_0 > 500$ V.

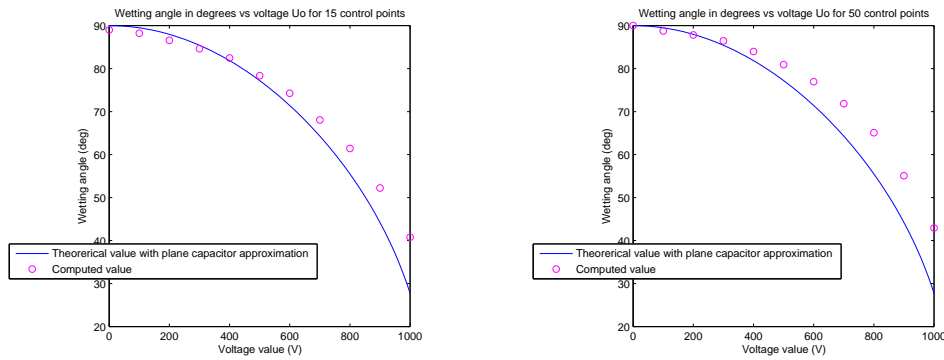


FIG. 8.5. Wetting angle. Computed values and Lippman's equation predictions. Left. With $NPC = 15$ Right. With $NPC = 50$

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

Let us precise that we did not manage to increase more NPC 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 points density becomes similar to the finite element points density.

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

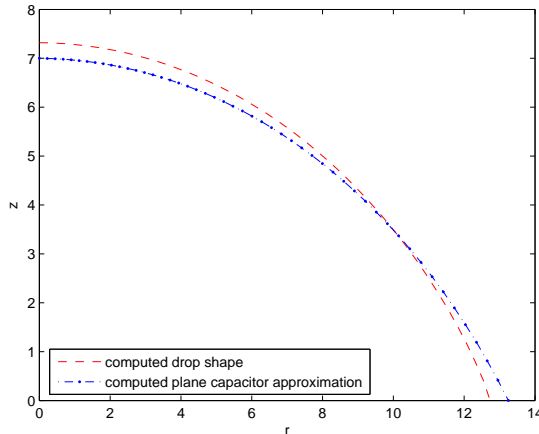


FIG. 8.6. *Computed shape compared to plane capacitor approximation shape for $u_0 = 800 V$*

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

In Fig. 8.8, we present the curvature values for different electrical potential u_0 values with 50 control points (curvature values corresponding to Fig. 8.7, but for different u_0 values).

For all computations we performed, curvatures behave like those showed in Fig. 8.8.

Then, we can make three main remarks :

- For the curvature, the results are more accurate with 50 points than with 15 points or 30 points. With 15 points or 30 points, the behavior of the curvature near the triple points appears to be less clear than with 50 points. This is due to the too small number of points near the triple line.
- For a given potential u_0 , the curvature remains constant except when we approach the triple line, where the curvature increases. We can see near the triple point that the curvature is higher than away from it. (see Fig. 8.7 for 800V, for other voltage the curvature has the same behavior, see Fig. 8.8.)
- If we look at the evolution of the curvature for an increasing potential u_0 , we notice that:
 - 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 away from the triple point decreases is in accordance with the fact that globally the drop should be a portion of sphere with an increasing radius as u_0 increases. We noticed that the curvature increases near the triple line, it is in accordance with the fact that the contact angle is higher than the Lippman

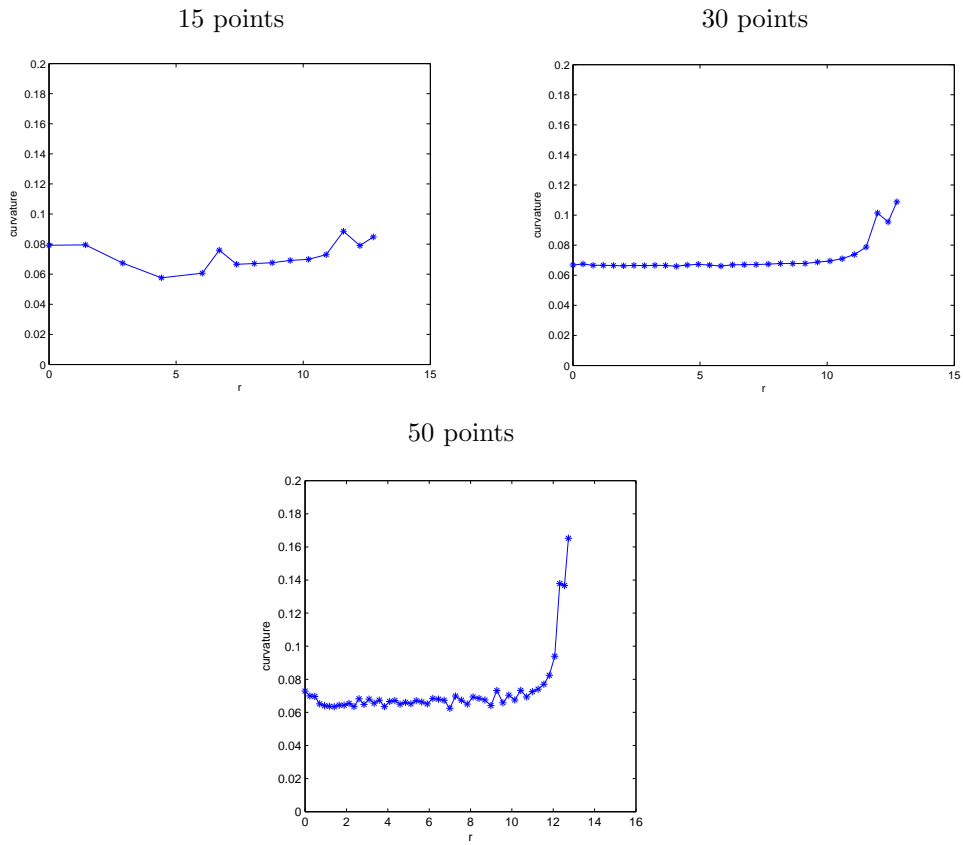


FIG. 8.7. Curvature values at $u_0 = 800V$ for 15 points, 30 points and 50 points respectively.

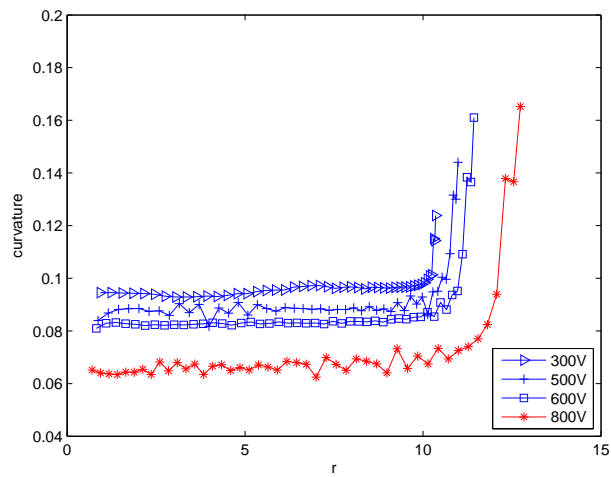


FIG. 8.8. Curvature of the drop for few u_0 values with 50 control points

predicted value.

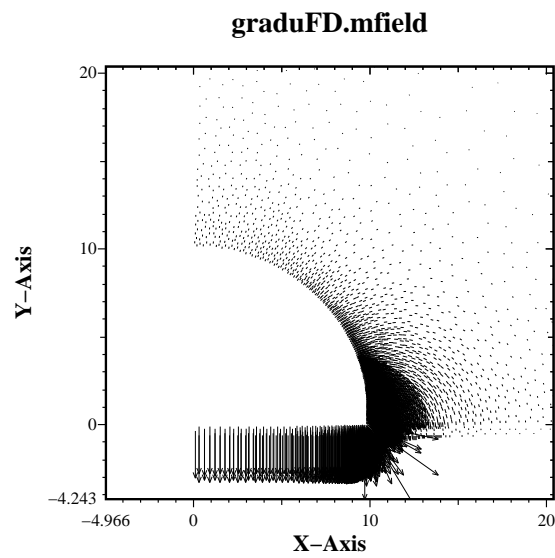


FIG. 8.9. *External electric field $\vec{E} = \vec{\nabla}u$ at $u_0 = 400V$ (zoom around the droplet)*

9. Conclusion. We have detailed and implemented a general approach to model 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 validate 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 computed shapes and angles are not in agreement with this theory for a voltage higher than 300V.

Although, we didn't obtain properly the locking phenomena, the drop shape obtained deviates from the predicted shape like in [16]. Also, we didn't manage to observe that the contact angle remains constant, but the computed contact angle values are higher than those predicted by Lippman equation. Moreover this overestimate becomes more important when using a higher control points density.

In other respects, we compute the curvature of the droplets. These values are globally constant except in a 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 noticed this increase of the curvature near the triple line.

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

Acknowledgments. The authors would like to thank C. Quiliet and M. Bienia from Laboratory of Spectrometry Physics -LSP- 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*, CRAS 317 Série II, (1993), pp. 157-163.
- [2] M. BIENIA, C. QUILLET AND M. VALLADE, *Modification of drop shape controlled by electrowetting*, Langmuir, 19 (2003), pp. 9328-9333.
- [3] M. BIENIA, M. VALLADE, C.QUILLET 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*, PhD Thesis, University of Grenoble I (1997).
- [5] J. BUEHRLE, S. HERMINGHAUS, AND F. MUGELE, *Interface profiles near three-phase contact lines in electric fields*, Phys. Rev. Lett., 91, 8 (2003), pp. 086101.
- [6] J. CÉA, *Conception optimale ou identification de formes - Calcul rapide de la dérivée directionnelle de la fonction coût*, M2AN, 20, 3 (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. Th. Appl., 110, 1, (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 et al. eds, Springer-Verlag, 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., 2003, pp. 35-57.

- [11] B. MOHAMMADI AND O. PIRONNEAU, *Applied optimal shape design for fluids*, Oxford University Press, 2001.
- [12] J. MONNIER, *Shape sensitivities in a Navier-Stokes flow with convective and gray bodies radiative thermal transfer*, *Optim. Control. Appl. Meth.*, 24 (2003), pp. 237-256.
- [13] J. MONNIER AND P. CHOW-WING-BOM, *ElectroCap: A shape inverse model for an electrocapillary process*. Report INRIA RR-5617, July 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. Comput. Multiscale Comput. Eng.*, 5 (2007), pp. 417-434.
- [15] F. MURAT AND J. SIMON, *Sur le Contrôle par un Domaine Géométrique*, Publication of the Laboratory of Numerical Analysis, University Paris VI, 1976.
- [16] A. G. PAPATHANASIOU AND A. G. BOUDOUVIS, *Manifestation of the connection between dielectric breakdown strength and contact angle saturation in electrowetting*, *Applied Physics Letters* 86 (2005), pp. 164102.
- [17] V. PEYKOV, A. QUINN AND J. RALSTON, *Electrowetting: a model for contact-angle saturation*, *Coll. Polym. Sci.*, 278 (2000), pp. 789-793.
- [18] C. QUILLIET AND B. BERGE, *Electrowetting: a recent outbreak*. *Curr. Opinion Coll. Interf. Sci.*, 6 (2001), pp. 34-39.
- [19] P. SARAMITO, N. ROQUET AND J. ETIENNE, *Rheolef home page: www-lmc.imag.fr/lmc-edp/Pierre.Saramito/rheolef/*, LMC-IMAG, (2002).
- [20] B. SHAPIRO, *Equilibrium behavior of sessile drops under surface tension, applied external fields and material variations*, *J. Appl. Physics*, 93, 9 (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. VERHELJEN AND M. W. J. PRINS, *Reversible electrowetting and trapping of charge: model and experiments*, *Langmuir*, 15 (1999), pp. 6616-6620.