INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Navier-Stokes ALE free surface flow with generalized Navier slip conditions. Droplet impact and attempt using Comsol Multiphysics 3.2.

Adel M. Benselama — Jérôme Monnier

N° ????

Avril 2007

Thème NUM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--????--FR+ENG

Navier-Stokes ALE free surface flow with generalized Navier slip conditions. Droplet impact and attempt using Comsol Multiphysics 3.2.

Adel M. Benselama* †, Jérôme Monnier‡*

Thème NUM — Systèmes numériques
Projet MOISE

Rapport de recherche n° ???? — Avril 2007 — 32 pages

Abstract: Droplet impact onto a solid surface is modeled and numerically simulated using an ALE formulation of the Navier-Stokes free surface equations. The triple line dynamics aspect is modeled implicitely through the Shikhmurzaev theory. Those equations involve a generalized Navier slip boundary conditions with the variation of the surface tension near the triple line (thus a local Marangoni effect). Comsol Multiphysics 3.2 potentialities are investigated to implement the different components of this ALE free surface problem. Finally, some numerical results of the spreading phase are obtained using the algorithm elaborated and implemented into Micralef, a home-developed software.

Key-words: Navier-stokes free surface, ALE, generalized Navier slip boundary conditions, triple line dynamics, droplet impact, Comsol multiphysics 3.2, Micralef.

* Lab. LJK, BP 53, F-38041 Grenoble Cedex 9, France
† <mailto:adel.benselama@imag.fr>
‡ INPG & INRIA, <mailto:jerome.monnier@imag.fr>

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Navier-Stokes ALE surface libre avec conditions de glissement de Navier generalisée. Impact de gouttelettes et tentative de simulations avec Comsol Multiphysics 3.2.

Résumé : Nous nous intéressons à la modélisation numérique d'une gouttelette impactant un solide. Les équations sont celles de Navier-Stokes surface libre en formulation ALE. La dynamique de la ligne triple est implicitement modélisée par les équations de Shikhmurzaev. Ces équations introduisent une condition de glissement de Navier généralisée, avec un gradient de tension de surface (effet Marangoni localisé). Les potentialités du logiciel Comsol Multiphysics 3.2 pour résoudre ce problème sont étudiées. Finalement, nous obtenons des résultats basés sur les algorithmes établis mais implémentés au sein de Micralef, un code de calcul spécialement développé pour cette étude.

Mots-clés : Navier-stokes surface libre, condition de glissement de Navier généralisée, dynamique de ligne triple, impact de gouttelette, Comsol multiphysics 3.2, Micralef.

Contents

1 Introduction

The aim of this work is to elaborate a mathematical and numerical model dealing with the dynamic contact line problem. Involved in the presence of three phases, where at least two of them are fluids (liquid-gas or non miscible liquid-liquid flow), the dynamics of this line significantly influences the interface as well as the bulk motion. The physical phenomenon of dynamical contact line appears in several industrial contexts such as coating and cleaning of solids by liquids. Two features of such slow, but not creeping, flows are revealed by experiments: *i)* the liquid front advances following a rolling motion: the particles of the liquid-gas interface are progressively projected onto the solide-liquid interface, in the same manner as does a rolling non-sliding sphere on a solide surface; *ii)* the contact angle depends on both its static value, determined thanks to the classical Young's equation, and the fluid velocity in the bulk. Even if succesfull attempts have been done to prescribe an explicit law of that contact angle, none of them is general because of the need to adjust their coefficients empirically. The contact angle and the triple line velocity should be treated by a more deep model, which describes and formalizes correctly the flow that occurs in the immediate vicinity of the triple line.
The mathematical modeling of the moving contact line is hard to deal with. A classical no-slip boundary condition at the solid-liquid interface implies a non-physical singularity: the fluid exerts an infinite force on the solid surface (glass filling paradox). Most of models have then been based on a fully slip description of the solid-liquid interface. Even slip condition removes the singularity, it prevents the fluid from rolling, which is actually not the case as mentioned above. Furthermore, for a normal liquid flowing over an even smooth solid, slippage is usually negligible.
So, the solution is expected to be between slipping and no-slipping conditions. In fact, according to Shikhmurzaev's theory [11], [2], , for instance, generalized Navier condition has to be applied in the vicinity the triple line. Shikhmurzaev's theory is however a much deeper translation of the phenomena that happen near the triple line than a rude technique to overcome mathematical inconsistencies or to adjust an arbitrary approximation law.

This paper is organized as follows. In section 2, we present the full Navier-Stokes equations (the 2D axisymmetric), the Shikhmurzaev surface equations (written in a planar surface co-ordinate). A link with the surfactant equations is done. In section 3, we recall the basic principles of the ALE method. In section 4, we present how the ALE method is implemented into Comsol$^{(\mathrm{TM})}$ Multiphysics 3.2. Since few important quantities (such as the curvature) are not easily computable in this software, we present some tricks to overcome these difficulties. Since the ALE implementation in Comsol Multiphysics 3.2 is not robust enough to handle large mesh distorsion, we present some preliminary numerical results obtained using our home-developed software Micralef. We compute the spreading phase of a 2D axisymmetric droplet impacting a solid surface. The algorithms implemented (in particular, the spreading algorithm) are those presented previously.

2 Mathematical model

2.1 Navier-Stokes equations

The droplet dynamics is modeled by the unsteady incompressible 2D axisymmetric Navier-Stokes equation. We denote by $\vec{u} = (u_r, u_z)^T$ the fluid velocity, p its pressure, Σ the stress tensor, D the deformation tensor and Re the Reynolds number. We denote by $(\vec{\tau}, \vec{n})$ the unit tangential and external normal vectors, respectively, such that it is direct. We set $\vec{\Sigma_n} = \vec{n} \cdot \Sigma$; $\Sigma_n = \vec{\Sigma_n} \cdot \vec{n}$; $\Sigma_\tau = \vec{\Sigma_n} \cdot \vec{\tau}$ and $\vec{f}$ the body force density. This equation writes down like

$$\rho \frac{D\vec{u}}{Dt} = \nabla \cdot \Sigma + \rho \vec{f} \tag{1}$$

For a Newtonian fluid, the stress tensor is given by

$$\Sigma = -pI + \mu \left(\nabla \vec{u} + (\nabla \vec{u})^T \right) \tag{2}$$

which gives

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) = -\nabla p + \mu \nabla^2 \vec{u} + \rho \vec{f} \quad \text{in } [0,T] \times \Omega_f \tag{3}$$

This equation is completed by the mass conservation equation

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \vec{u} = 0 \quad \text{in } [0,T] \times \Omega_f$$

which reduces for an incompressible fluid (as $D\rho/Dt = 0$) to

$$\nabla \cdot \vec{u} = 0 \quad \text{in } [0,T] \times \Omega_f \tag{4}$$

We present the modelling of the free surface and its dynamics in the ALE section below. Equations (3) and (4) are completed by suitable initial and boundary conditions.

2.2 Surface equations of the Shikhmurzaev theory

According to the Shikhmurzaev's theory, [11], [2] (see also [6] [8] for a reformulation of the equations) the surface equation to be solved writes down like

$$\frac{\partial \rho_i^s}{\partial t} - \xi_i \nabla_s \cdot (\rho_i^s \nabla_s \rho_i^s) + \nabla_s \cdot (\rho_i^s u_i) + \frac{1}{\tau^*} \rho_i^s = \frac{1}{\tau^*} \rho_i^{eq} \quad \text{in } [0,T] \times \Gamma_f^m \tag{5}$$

where $\xi_i = \xi$ for $i = 1$ and $\xi_i = \lambda$ for $i = 2$. ρ_i^s is a surface density at the interface i that is related to the local surface tension by a given law. Each of the surface differential operators has to be written in the surface co-ordinate system (s, ϕ), where s is an arbitrary azimutal parametrization of the (free) surface and ϕ is the meridional angle as depicted in figure 1.

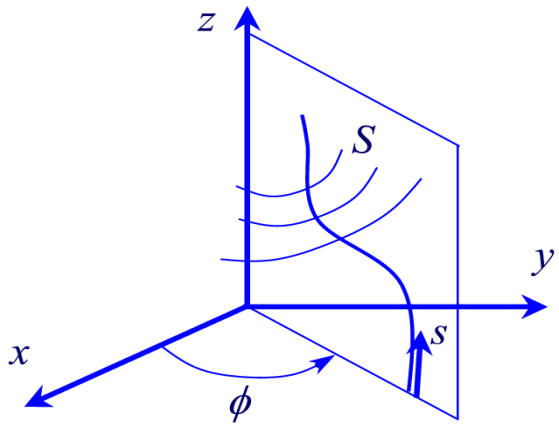

Figure 1. Surface co-ordinate system, (s, ϕ).

Injecting the expressions of the different surface differential operators given by appendix 1, the Shikhmurzaev equation (5) becomes

$$\begin{aligned} &\frac{\partial \rho_i^s}{\partial t} - \frac{1}{\sigma}\xi_i \frac{\partial}{\partial s}\left(\sigma \rho_i^s \frac{\partial \rho_i^s}{\partial s}\right) \\ &+\frac{1}{\sigma}\frac{\partial}{\partial s}\left(\sigma \rho_i^s u_{is}\right) + \frac{1}{\tau^*}\rho_i^s \quad = \quad \frac{1}{\tau_{eq}^*}\rho_i^{eq}, \quad (t,s) \in [0,T]\times[0,1] \end{aligned} \tag{6}$$

where

$$\sigma = r\left[\left(\frac{\partial r}{\partial s}\right)^2 + \left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{1}{2}} \tag{7}$$

Multiplying equation (6) by the test function $\hat{\rho}_i^s$ associated to ρ_i^s and integrating over the relevant boundary $\partial\Omega$, see figure 2, we get

$$\int_{\partial\Omega} r\hat{\rho}_i^s \frac{\partial \rho_i^s}{\partial t} ds =$$

$$
\begin{aligned}
&\left[\xi_i\hat{\rho}_i^s\left(s_1\right)\rho_i^s\left(s_1\right)\frac{\partial\hat{\rho}_i^s}{\partial s}\left(s_1\right)r\left(s_1\right)-\xi_i\hat{\rho}_i^s\left(s_0\right)\rho_i^s\left(s_0\right)\frac{\partial\hat{\rho}_i^s}{\partial s}\left(s_0\right)r\left(s_0\right)\right]\\
&-\int_{\partial\Omega}\xi_i\left[\frac{\partial\hat{\rho}_i^s}{\partial s}\frac{1}{\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{1}{2}}}+\hat{\rho}_i^s\frac{\left(\frac{\partial r}{\partial s}\right)\left(\frac{\partial^2 r}{\partial s^2}\right)+\left(\frac{\partial z}{\partial s}\right)\left(\frac{\partial^2 z}{\partial s^2}\right)}{\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{3}{2}}}\right]\\
&\times r\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{1}{2}}\rho_i^s\frac{\partial\rho_i^s}{\partial s}ds\\
&-\left[\hat{\rho}_i^s\left(s_1\right)\rho_i^s\left(s_1\right)u_{is}\left(s_1\right)r\left(s_1\right)-\hat{\rho}_i^s\left(s_0\right)\rho_i^s\left(s_0\right)u_{is}\left(s_0\right)r\left(s_0\right)\right]\\
&+\int_{\partial\Omega}\left[\frac{\partial\hat{\rho}_i^s}{\partial s}\frac{1}{\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{1}{2}}}+\hat{\rho}_i^s\frac{\left(\frac{\partial r}{\partial s}\right)\left(\frac{\partial^2 r}{\partial s^2}\right)+\left(\frac{\partial z}{\partial s}\right)\left(\frac{\partial^2 z}{\partial s^2}\right)}{\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{3}{2}}}\right]\\
&\times r\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{1}{2}}\rho_i^s u_{is}ds\\
&-\int_{\partial\Omega}r\hat{\rho}_i^s\frac{1}{\tau^*}\rho_i^s ds+\int_{\partial\Omega}r\hat{\rho}_i^s\frac{1}{\tau_{eq}^*}\rho_i^{eq}ds,\quad t\in[0,T]
\end{aligned}
\tag{8}
$$

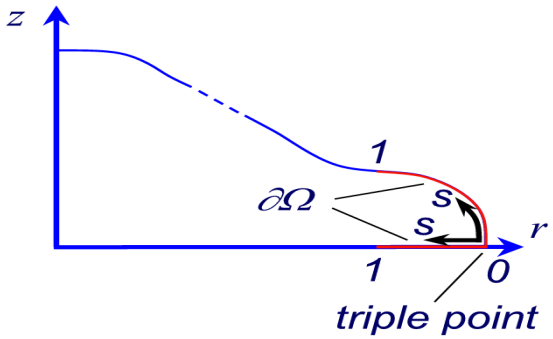

Figure 2. Mesoscopic domain $\partial\Omega$, near the triple point, over which Shikhmurzaev equation is applied.

Boundary conditions associated to this equation are

$$\rho_i^s(s_0)=\rho_0^{eq} \tag{9}$$

$$\rho_i^s(s_1)=\rho_1^{eq} \tag{10}$$

These surface equations are fully coupled with the Navier-Stokes equations (macroscopic model) through the generalized Navier slip boundary conditioins presented in next section, see [11], [2] , [6], [8] .

2.3 An analogy with the surfactant equations

The surface equatiions arising in the Shikhmurzaev's theory are similar to those modelling the dynamics of surfactants. We present in this section the surfacatants equations and their weak form.
Let us denote Γ the surface concentration of the surfactant, mesured in moles per surface area. According to [12], the surfactant mass balance over an infinitesimal section of the interface is governed by a convection-diffusion equation, in the presence of a distributed source,

$$\begin{aligned}\frac{d_s\Gamma}{dt} + \nabla_s \cdot (\Gamma \vec{u}_s) - \vec{w}_s \cdot \nabla_s \Gamma + \Gamma\kappa\,(\vec{u}\cdot\vec{n}) \\ = D_s \nabla_s^2 \Gamma, \qquad (t,s) \in [0,T]\times[0,1]\end{aligned} \tag{11}$$

$\vec{u}_s$ is the tangent to the interface component of the fluid velocity, *i.e.*

$$\vec{u}_s = \vec{n} \times (\vec{u} \times \vec{n}) \tag{12}$$

The derivative d_s/dt expresses the rate of change of a variable following an interfacial marker point moving with the velocity $\vec{w}_s$. In other words

$$\frac{d_s}{dt} = \frac{\partial}{\partial t} + \vec{w}_s \cdot \nabla_s \tag{13}$$

$\vec{w}_s$ is definetely arbitrary and it is related to the markers velocity to the interface by the relation

$$\vec{v} = \vec{u}\cdot\vec{n} + \vec{w}_s \tag{14}$$

When $\vec{w}_s$ is taken to be equal to $\vec{u}_s$, the markers will represent material fluid particles.
As done for Shikhmurzaev equation, equation (11) has to be written in the surface coordinate system (s,ϕ). According to (77), (78), (79) and (13), this is straightforward and gives

$$\begin{aligned}\frac{\partial\Gamma}{\partial t} + w_s\cdot\nabla_s\Gamma + \frac{1}{\sigma}\frac{\partial}{\partial s}(\sigma\Gamma u_s) - w_s\frac{\partial\Gamma}{\partial s} + \Gamma\kappa(\vec{u}\cdot\vec{n}) \\ = D_s\frac{1}{\sigma}\frac{\partial}{\partial s}\left(\frac{\sigma^3}{r^2}\frac{\partial\Gamma}{\partial s}\right), \qquad (t,s)\in[0,T]\times[0,1]\end{aligned} \tag{15}$$

σ being given by expression (7). Equation (15) has to be multiplied by the test function $\hat{\Gamma}$ associated to Γ and integrated over the whole free interface so to obtain

$$\int_{\partial\Omega}\hat{\Gamma}\frac{\partial\Gamma}{\partial t}ds+\int_{\partial\Omega}\hat{\Gamma}\frac{1}{\sigma}\frac{\partial}{\partial s}\left(\sigma\Gamma\vec{u}_s\right)ds-\int_{\partial\Omega}\hat{\Gamma}w_s\frac{\partial\Gamma}{\partial s}ds \\ +\int_{\partial\Omega}\hat{\Gamma}\Gamma\kappa(\vec{u}\cdot\vec{n})ds=\int_{\partial\Omega}\hat{\Gamma}D_s\frac{1}{\sigma}\frac{\partial}{\partial s}\left(\frac{\sigma^3}{r^2}\frac{\partial\Gamma}{\partial s}\right)ds, \\ t\in[0,T] \tag{16}$$

Integrating this equation by parts yields to

$$\int_{\partial\Omega}\hat{\Gamma}\frac{\partial\Gamma}{\partial t}ds+\left[\hat{\Gamma}\Gamma u_s\right]_{s_0}^{s_1}+\int_{\partial\Omega}\hat{\Gamma}\Gamma u_s\frac{\varpi}{\sigma}ds-\int_{\partial\Omega}\frac{\partial\hat{\Gamma}}{\partial s}\Gamma u_s ds \\ -\int_{\partial\Omega}\hat{\Gamma}w_s\frac{\partial\Gamma}{\partial s}ds+\int_{\partial\Omega}\hat{\Gamma}\Gamma\kappa(\vec{u}\cdot\vec{n})ds \\ =\int_{\partial\Omega}\hat{\Gamma}D_s\frac{\varpi\sigma}{r^2}\frac{\partial\Gamma}{\partial s}ds-\int_{\partial\Omega}\frac{\partial\hat{\Gamma}}{\partial s}D_s\frac{\sigma^2}{r^2}\frac{\partial\Gamma}{\partial s}ds+\left[\hat{\Gamma}D_s\frac{\sigma^2}{r^2}\frac{\partial\Gamma}{\partial s}\right]_{s_0}^{s_1}, \\ t\in[0,T] \tag{17}$$

where ϖ is the quantity defined by

$$\varpi=\frac{1}{\left(\left[\frac{\partial r}{\partial s}\right]^2+\left[\frac{\partial z}{\partial s}\right]^2\right)^{1/2}}\left(r\frac{\partial r}{\partial s}\frac{\partial^2 r}{\partial s^2}+r\frac{\partial z}{\partial s}\frac{\partial^2 z}{\partial s^2}+\left[\frac{\partial r}{\partial s}\right]^3+\frac{\partial r}{\partial s}\left[\frac{\partial z}{\partial s}\right]^2\right) \tag{18}$$

Thus, if we manage to implement and solve correctly the Shikhmurzaev's equations, next we can consider the coupling with the present surfactant model.

2.4 Droplet geometry and mesoscopic boundary conditions

In the Shikhmurzaev's theory, [2] , the previous "local surface model" of the Shikhmurzaev theory are fully coupled with the present boundary conditions for the Navier-Stokes equations . We refer to [2] , [6], [8] for more details.

Let us consider the following droplet geometry.

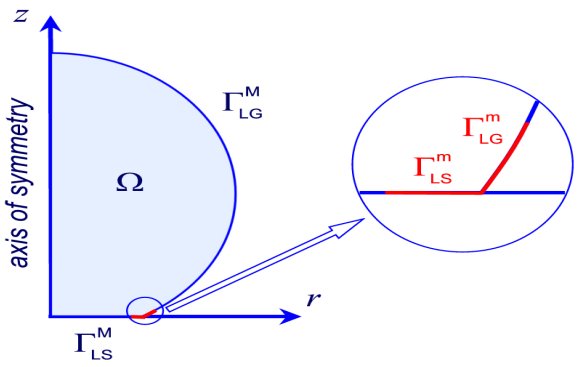

Figure 3. A droplet put down on a solid surface.

The generalized boundary conditions on the free surface (liquid-gas) are:

$$\vec{\Sigma}_n = (-p_{ext} + \frac{\kappa}{Ca})\vec{n} + h\vec{\tau} \qquad \text{in } [0,T] \times \Gamma_f \tag{19}$$

where Ca is the Capillary number, κ is the mean curvature and p_{ext} is the external pressure. The extra term h is given (it is a surface tension gradient). We have classical boundary conditions on the symmetry axis. We decompose the liquid-solid interface into two parts Γ_{ad} and Γ_{sl}. Γ_{sl} denotes a "small" part of the liquid-solid interface near the triple point. We consider a generalized Navier slip type boundary condition (local slipping)

$$\left\{ \begin{array}{lcll} \vec{u}.\vec{n} & = & 0 & \text{in } [0,T] \times \Gamma_{sl} \\ \Sigma_\tau & = & -[\beta\vec{u} + \vec{g}].\vec{\tau} & \text{in } [0,T] \times \Gamma_{sl} \end{array} \right. \tag{20}$$

where β is a given sliding coefficient. The extra term $\vec{g}$ is given, it models either a surface tension gradient in the Shikhmurzaev's theory or a uncompensated Young stress in the Qian-Wang-Sheng theory, see below. On Γ_{ad}, we impose adherence boundary conditions $\vec{u} = \vec{0}$.

2.5 Contact line dynamics modeling

We consider two different types of model for the contact line dynamics an explicit model (Tanner type law) and an "implicit" one deriving from the Shikhmurzaev theory.

Mobility relation (Tanner type Law). The contact line velocity and the wetting angle are related by

$$U_{CL}(t) \;=\; k\,\frac{(\theta(t) - \theta^{eq})^\iota}{(\theta^{eq})^\iota} \quad \text{for } t \in]0,T[\tag{21}$$

where U_{CL} is the contact line velocity, θ^{eq} is the wetting angle at equilibrium (Young's law), θ is the (dynamic) wetting angle. k and ι are parameters usually determined using

experimental results.

Local flow modelling and Shikhmurzaev's theory. The "implicit" model does not impose the contact line velocity and the wetting angle but consider them as a response of the model. This model is based on the Shikhmurzaev's theory, [11], [2], which makes introduce the generalized Navier slip condition (20) and the condition (19). These conditions are local since the extra terms $\vec{g}$ and h vanish except in a vicinity of the triple point. The basic idea of this theory is to consider that the rolling motion observed in experiments, [3], implies that particles of the liquid-gas interface become an element of the solid-liquid in a finite time. Then, the surface tension value associated to this particle must change to its new equilibrium value relative to the solid-liquid interface. This process would gives rise to a surface tension gradient in a small vicinity of the advancing contact line (hence a local Marangoni effect). In other respect, the Young equation would remain valid at any time. In this theory,

$$\vec{g} \;=\; -\frac{1}{2Ca}\,\vec{\nabla}\gamma \text{ and } h \;=\; \frac{1}{Ca}\,\vec{\nabla}\gamma_{LG}.\vec{\tau} \tag{22}$$

where γ and γ_{LG} are the liquid-solid and the liquid-gas surface tension coefficient respectively. In [8] and [7], a mathematical and numerical study presents some qualitative behaviors of g and h arising from Shikhmurzaev's theory.

A connection with Qian-Wang-Sheng theory. From molecular dynamics simulations on immiscible fluids, [13] shows that the relative slipping between the fluid and the solid wall follows a generalized Navier slip b.c. similar to (20) (if one phase fluid only is considered). In this theory, the extra term $\vec{g}$ in (20) would model an interfacial uncompensated Young stress. The extra tangential stress $\vec{g}$ is defined as follows $\int_{\Gamma_{int}} g\,dy \;=\; \gamma(cos\theta - cos\theta^{eq})$, where $\int_{\Gamma_{int}}\; dy$ denotes the integral across the interface Γ_f.

3 Basic principles of the ALE method

In this section is presented the basic principles of the the so-called *Arbitrary Lagrangian Eulerian* (ALE) method for tracking moving boundaries. The computational grid therefore follows the physical interface. Although other methods have proven their high efficiency to simulate fluid flows with moving interfaces (level set method, VOF method, front-tracking method), the choice of the ALE approach is motivated here by two arguments. First, no topological changes are expected to happen in the problem we are dealing with. Second, a high precision to achieve on the position of the interface is of major interest in triple line simulations, and one must impose some extra physical equations on this surface (surface equations arising from the Shikhmurzaev's theoery or from surfactant modelling). Thus, we believe that this can be achieved more easily and more accurately with ALE methods rather than with a level set approach for instance.

The ALE formulation for *one* fluid with a free interface has been widely used by many authors, see e.g. [4], [5].

We now introduce some notations. For $t > 0$, we define a reference configuration $\tilde{\Omega}$. The liquid-gas interface is denoted Γ_{LG} and the solide-liquid interface is denoted Γ_{SL}. The normal $\vec{n}$ to each interface is directed outward the liquid domain. We define a reference configuration $\tilde{\Omega} = \Omega_{t=0}$ and we consider a mapping $\tilde{A}_t$ which associates to a point $\tilde{x} \in \tilde{\Omega}$ a point $x \in \Omega$. Throughout, the mapping $\tilde{A}_t$ will be supposed to be smooth egough in $\tilde{x}$, invertibble with a smooth inverse, and differentiable with respect to time, t. For a function $\psi(\cdot, t)$ defined on Ω, we denote by $\tilde{\psi}(\cdot, t)$ the function defined on $\tilde{\psi}$ satisfying $\tilde{\psi}(\tilde{x}, t) = \psi(x, t)$, with $x = \tilde{A}_t(\tilde{x})$. By a classical abuse of notation, we will denote $\frac{\partial \psi}{\partial t}|_{\tilde{x}}$ the time derivative on the ALE frame

$$\left.\frac{\partial \psi}{\partial t}\right|_{\tilde{x}} (x, t) = \frac{\partial \tilde{\psi}}{\partial t}(\tilde{x}, t), \quad \text{with} \quad \tilde{x} = \tilde{A}_t^{-1}(x) \tag{23}$$

The domain velocity $\vec{v}$ is defined by

$$\vec{v}(x, t) = \tilde{\vec{v}}(\tilde{x}, t) = \frac{\partial A_t}{\partial t}(\tilde{x}), \quad \text{with} \quad \tilde{x} = \tilde{A}_t^{-1}(x) \tag{24}$$

Since, according to (14), we have

$$\vec{u} \cdot \vec{n} = \vec{v} \cdot \vec{n} \quad \text{on} \quad \partial\Omega, \tag{25}$$

which means that the domains Ω_t follows the free interface. Thus,

$$\text{for } \tilde{x} \in \tilde{\Omega}, \quad x = \tilde{A}_t(\tilde{x}) \in \Omega_t \tag{26}$$

Although the domains Ω and $\tilde{\Omega}$ are in fact the same domain of $\mathbb{R}^d$ (d is the space dimension), we will keep both notations for the sake of clarity.

We now recall some standard formulae whith will be useful in the sequel. In other words, all the differential operators $(\nabla, \nabla\cdot, \nabla^2)$ will be take with respect to the Eulerian variable x. We have

$$\left.\frac{\partial \psi}{\partial t}\right|_{\tilde{x}} (x, t) = \frac{\partial \tilde{\psi}}{\partial t}(\tilde{x}, t) = \frac{\partial \psi}{\partial t}(x, t) + \vec{v} \cdot \nabla \psi(x, t) \tag{27}$$

We denote by $\tilde{\mathbf{J}}_t$ the Jacobian matrix of $\tilde{A}_t$,

$$\tilde{\mathbf{J}}_t = \left[\frac{\partial \tilde{A}_t}{\partial \tilde{x}_j}\right]$$

and by $\tilde{J}_t$ its determinant. Then we have the Euler formula,

$$\frac{\partial \tilde{J}_t}{\partial t}\left(\tilde{A}_t^{-1}(x), t\right) = \tilde{J}_t\left(\tilde{A}_t^{-1}(x), t\right) \nabla \cdot \vec{v}(x, t) \tag{28}$$

Using this relation, we get in particular

$$\begin{aligned}\frac{d}{dt}\int_{\Omega_t}\psi(x,t)dx &= \frac{d}{dt}\int_{\Omega_t}\tilde{\psi}(\tilde{x},t)\tilde{J}_t d\tilde{x}\\ &= \int_{\Omega_t}\frac{\partial}{\partial t}\left(\tilde{\psi}(\tilde{x},t)\tilde{J}_t\right)d\tilde{x}\\ &= \int_{\omega_t}\left(\frac{\partial\tilde{\psi}}{\partial t}(\tilde{x},t)\tilde{J}_t+\tilde{\psi}(\tilde{x},t)\tilde{J}_t\nabla\cdot\vec{v}\right)d\tilde{x}\end{aligned}\tag{29}$$

and thus

$$\frac{d}{dt}\int_{\Omega_t}\psi(x,t)\;dx=\int_{\Omega_t}\left.\frac{\partial\psi}{\partial t}\right|_{\tilde{x}}dx+\int_{\Omega_t}\psi(x,t)\nabla\cdot\vec{v}\;dx\tag{30}$$

This equation will be used when replacing the mapping ψ by any defined given physical mapping (velocity, pressure, etc.).

We now propose a weak formulation of equations (3) and (4). The following functional spaces will be needed $\mathbb{H}^1_0(\Omega)=\left\{\vec{w}\in\left(H^1(\Omega)\right)^d\right\}$, $\mathbb{P}^1_0(\Omega)=\left\{q\in H^1(\Omega)\right\}$. We define: $W=L^2(0,T;\mathbb{H}^1_0(\Omega))$, $\quad P=L^2(0,T;\mathbb{P}^1_0(\Omega))$. In order to express the weak form associted to the mass and momentum equations in the current frame, we introduce the test function spaces defined on the reference frame: $\tilde{W}=\mathbb{H}^1_0(\tilde{\Omega})$, $\quad\tilde{P}=\mathbb{H}^1_0(\tilde{\Omega})$. On the current domain, the test function spaces are defined by

$$W_0=\{\vec{w}:\Omega_t\times[0,T]\to\mathbb{R}^d,\vec{w}(x,t)=\tilde{\vec{w}}\left(\tilde{A}_t^{-1}(x)\right),\tilde{\vec{w}}\in\tilde{W}\},$$

$$P_0=\{q:\Omega_t\times[0,T]\to\mathbb{R},q(x,t)=\tilde{q}\left(\tilde{A}_t^{-1}(x)\right),\tilde{q}\in\tilde{P}\}.$$

It is worthwhile to notice that the test functions do not depend on time on the reference frame whereas they do on the current one. More precisely, denoting by $(w_i)_{i=1,\ldots,d}$ the components of $\vec{w}\in W_0$, we have

$$\left.\frac{\partial w_i}{\partial t}\right|_{\tilde{x}}=\frac{\partial w_i}{\partial t}+\vec{v}\cdot\nabla w_i=0\tag{31}$$

Similar relation holds for the functions $q\in P_0$.

We now give the formulation that will be used in the numerical simulations.

Weak ALE formulation. Suppose that the domain is moving such relation (14), beyong others, is satisfied. We look for $(\vec{u},p)$ in $W\times P$ such that, $\forall(\vec{w},q)\in W_0\times P_0$:

$$\begin{aligned}\frac{d}{dt}\int_\Omega\rho\vec{u}\cdot\vec{w}\;dx+\int_\Omega\rho(u\vec{-}v)\cdot\nabla\vec{u}\cdot\vec{w}\;dx-\int_\Omega\rho\vec{u}\cdot\vec{w}\nabla\cdot\vec{v}\;dx\\ =\int_\Omega\left(-\nabla p+\mu\nabla^2\vec{u}+\rho\vec{f}\right)\cdot\vec{w}\;dx\end{aligned}\tag{32}$$

$$\int_\Omega q\nabla\cdot\vec{u}\ dx = 0 \tag{33}$$

Proposition
System (3)-(4) is formally equivalent to the weak ALE formulation (32)-(33).

Proof. Let $(\vec{u}, p)$ be a solution to (3)-(4). For the momentum balance equation (32), using formula (27), we have from (3)

$$\rho\left(\left.\frac{\partial\vec{u}}{\partial t}\right|_{\tilde{x}} + (u \vec{-} v)\cdot\nabla\vec{u}\right) = \vec{h}$$

where $\vec{h} = -\nabla p + \mu\nabla^2\vec{u} + \rho\vec{f}$. By multiplying this equation by $\vec{w}\in W_0$ and integrating over Ω, we get

$$\int_\Omega \rho\left.\frac{\partial\vec{u}}{\partial t}\right|_{\tilde{x}}\cdot\vec{w}\ dx + \int_\Omega \rho(u \vec{-} v)\cdot\nabla\vec{u}\cdot\vec{w}\ dx = \int_\Omega \vec{h}\cdot\vec{w}\ dx \tag{34}$$

using (30) and (31), we can write the first term as follows

$$\begin{aligned}\int_\Omega \rho\left.\frac{\partial\vec{u}}{\partial t}\right|_{\tilde{x}}\cdot\vec{w}\ dx &= \int_\Omega \rho\left.\frac{\partial}{\partial t}\right|_{\tilde{x}}(\vec{u}\cdot\vec{w})\ dx \\ &= \frac{d}{dt}\int_\Omega \rho\vec{u}\cdot\vec{w}\ dx - \int_\Omega \rho\vec{u}\cdot\vec{w}\nabla\cdot\vec{v}\ dx \end{aligned}\tag{35}$$

Thus, the eventual momentum equation in the weak form reads

$$\begin{aligned}\frac{d}{dt}\int_\Omega \rho\vec{u}\cdot\vec{w}\ dx + \int_\Omega \rho(u \vec{-} v)\cdot\nabla\vec{u}\cdot\vec{w}\ dx - \int_\Omega \rho\vec{u}\cdot\vec{w}\nabla\cdot\vec{v}\ dx \\ = \int_\Omega \vec{h}\cdot\vec{w}\ dx\end{aligned}\tag{36}$$

In addition, multiplying by $q\in P_0$ and integrating over Ω, the weak equation associated to the mass conservation equation (4) reads as given by (33).

Conversely, and independently of boundary conditions, if $(\vec{u}, p)$ is a solution to the weak ALE formulation (36)-(33) for all $(\vec{w}, q)\in W_0\times P_0$ then $(\vec{u}, p)$ is a solution to the system (3)-(4). The demonstration is however out of the scope of this paper.

Mesh velocity law. All we did so far is to specify the mesh velocity $\vec{v}$ by the condition (14), where $\vec{w}_s$ is an arbitrary tangential to the interface velocity. Perhaps the more intuitive reaction is to take this later equal to zero, that is, no mesh sliding is allowed at the boundaries. Even if it is shown it has a good mass conservation property, using normal velocity component for free interfaces is however not always of benefit on surfaces that are not supposed to deform. Let us suppose, for any reason, we have made our mind up about this surface tangential velocity component. The next question is *how will the mesh deform*

inside the domain? The answer is given by the name of the method: arbitrarily (once more). One can take virtually any equation to be satisfied on Ω. Virtually any one and not any one, since any point from the domain Ω has to remain inside its (moving) boundary $\partial\Omega$ as time goes on. A simple way to satisfy this condition is to choose, among others, an elliptical equation to be satisfied by the mesh velocity on Ω. More precisely, a better choice is to have the mesh velocity in such a way to preserve a quality equivalent to this of the initial mesh during the whole process of deformation. Elasticity equations are a common and efficient way to achieve such a goal.

We refer to [7] for a description of the ALE algorithm implemented into Micralef, computational code used in the sequel.

4 Comsol Multiphysics 3.2 and its incompatibilities with droplet modelling

A general presentation of COMSOL Multiphysics is made in Appendix B. In what follows we focus our attention on the possibilities given by the ALE new mode that we chiefly are concerned. We present below briefly the ALE implementation in Comsol Multiphysics 3.2. Then some tricks useful to overcome some difficulties when using Comsol Multiphysics 3.2 (difficulties inherent in this software as being not a fully open package).

In summary, unfortunately the present implementation of ALE method into Comsol Multiphysics (version 3.2) does not allow to consider large mesh distortions which necessarily appear when modeling free surface flows (such as a droplet impact). Furthermore, in order to simulate a free surface flow driven by surface tension forces one must be able to compute the curvature forces. The present implementation in Comsol Multiphysics prevent to do it straightforwardly. Also, we encoutered serious difficulties to implement an algorithm to treat the triple line dynamics. Nevertheless, we present below some tricks useful to overcome some of these difficulties.

For the description of an efficient implementation of the ALE method and the description of a triple line dynamics algorithm as required for a free surface droplet flow, we refer to [7]. The latter approach has been implemented into our home-developed software Micralef, which led to the preliminary numerical results presented in next section.

4.1 The ALE implementation in Comsol Multiphysics 3.2

The ALE algorithm is implemented as a new physics mode strongly coupled with the original mechanical problem since Navier-Stokes equations are expressed in a domain which is determined by the ALE mode and the ALE physics is completed by kinematic boundary conditions supplied by Navier-Stokes equations mode. By default the mesh velocity is

governed by Laplace equation:

$$\nabla^2 \vec{v} = 0 \tag{37}$$

which is completed with suitable boundary conditions

$$\vec{v}|_{interface} = (\vec{u} \cdot \vec{n})\, \vec{n}|_{interface} \tag{38}$$

COMSOL Multiphysics also offers the possibility to implement one's own mesh governing equations but many difficulties arose to do so in practice.
As mentioned previously, one must define the mesh velocity in such a way to preserve a good quality of the mesh during the whole process of deformation. An efficient way to acheive this goal is to define the mesh velocity by solving a simple linear elasticity system. Suprisingly, this is has not been foreseen into the ALE new mode of Comsol Mulitphysics. Comsol uses instead the Laplace equation or Winslow equations which is not suitable for large distorsions as this appears in the present application (and in free surface flows in general). Such an approach is probably robust enough for fluid-structure interactions since in that case, mesh distortions are much smaller, but not in our case. Furthermore, in order to implement the present mathematical model (eg the curvature term , and the dynamics of the triple point), one needs to define and / or obtain some essential informations. We present below how we circumvented the standard use of Comsol in order to reach (partially only) our goal.

4.2 Some tricks to overcome difficulties encountered

Comsol Multiphysics is reputed to be significantly an open scientific software, although some information are not straightforward or even impossible to obtain. Among these are the curvature of the boundaries, the maximum of a given space-dependent function and the current time step. In the following we explain the different stategies we adopted to get most of those quantities.

4.2.1 Curvature computation

If for some reason we do need to compute the curvature, or other boundary differential quantities, and not its weak integral over a given boundary, the simplest (not the most elegant, though) way is to solve simultanously the following trivial physics problems in the weak form (for simplicity, we suppose we are dealing with a 2D configuration using the co-ordinates r,z):

$$\widehat{coordr}\ (coordr - r) = 0 \tag{39}$$

$$\widehat{coordz}\ (coordz - z) = 0 \tag{40}$$

Thus, the variable *coordr* (resp. *coordz*) should give an approximation of the r (resp. z) co-ordinate. Once evaluated, the first and second order space derivatives are obtained according

to

$$\frac{\partial r}{\partial s} = \frac{\partial coordr}{\partial s} = coordrTr \times tr + coordrTz \times tz \tag{41}$$

$$\frac{\partial z}{\partial s} = \frac{\partial coordz}{\partial s} = coordzTr \times tr + coordzTz \times tz \tag{42}$$

$$\frac{\partial^2 r}{\partial s^2} = \frac{\partial^2 coordr}{\partial s^2} = coordrTrr + coordrTzz \tag{43}$$

$$\frac{\partial^2 z}{\partial s^2} = \frac{\partial^2 coordz}{\partial s^2} = coordzTrr + coordzTzz \tag{44}$$

where tr and tz are the components of the tangential vector to the interface, and $coordrTr$, $coordrTz$, $coordrTrr$, $coordrTzz$ are COMSOL Multiphysics syntax for the (output) quantities

$$coordrTr = \frac{\partial coordr}{\partial s} tr \tag{45}$$

$$coordrTz = \frac{\partial coordr}{\partial s} tz \tag{46}$$

$$coordrTrr = \frac{\partial^2 coordr}{\partial s^2} tr^2 \tag{47}$$

$$coordrTzz = \frac{\partial^2 coordr}{\partial s^2} tz^2 \tag{48}$$

The quantities corresponding to $coordz$ are given in a similar manner.
Hence the planer (first principal) curvature is obtained by

$$\kappa_1 = \frac{\frac{\partial^2 r}{\partial s^2}\frac{\partial z}{\partial s} - \frac{\partial^2 z}{\partial s^2}\frac{\partial r}{\partial s}}{\left[\left(\frac{\partial r}{\partial s}\right)^2 + \left(\frac{\partial z}{\partial s}\right)^2\right]^{3/2}} \tag{49}$$

Bearing in mind that due to axisymmetry, the second principal curvature is given by

$$\kappa_2 = \frac{n_z}{r} \tag{50}$$

the total curvature, given by

$$\kappa = \kappa_1 + \kappa_2, \tag{51}$$

is then definetely determined.

4.2.2 Capillary effects

To introduce forces due to capillary effects, which are surface-located, into the volumic kinetic balance, two ways are conceivable. The first is to compute directly the curvature as

given by the previous subsection **4.2.1** and take into account the term $\kappa\gamma\vec{n}$ in the surface balance applied to the interface boundary condition. Let us show how the surface balance is injected into the COMSOL interface boundary condition. For accommodation, we rewrite the normal projection of equation (19), when the external pressure is equal to zero and no tangential trailing is imposed (we delay this treatment for a next step), in the form

$$\Sigma_n - \kappa\gamma = 0 \tag{52}$$

By default, COMSOL does not propose such a boundary condition. To make it support this condition, we have to relax the boundary condition, by imposing "initially" neutral condition, and then use the Lagrange multipliers technic in order to constrain the desired condition, namely equation (52), to be satisfied. Let λ be the Lagrange multiplier we are using. The equation this new unkown has to satisfy (and we have to introduce) is

$$\hat{\lambda}\left(\Sigma_n - \kappa\gamma\right) - \lambda\left(\widehat{u\cdot n}\right) = 0 \tag{53}$$

We remind that $\hat{}$ (hat) symbol denotes the associated test function of the underlying quantity. This equation expresses only the fact that the condition $\Sigma_n - \kappa\gamma$ is applied to the normal projection of the momentum equation by means of the associated quantity $\widehat{u\cdot n}$. The resolution of fluid dynamic problem is hence done with boundary conditions (at least some of them) handled as weak constraints, since equation (53) is by definition a weak form.

4.2.3 Maximum of a space-dependent function

Perhaps one of the most tedious tasks was to get the maximum of a space- dependent function. Let us first explain why have we to compute such a value and then how will we do it.
In the case of a spreading droplet over a solid surface, we have to apply different boundary conditions whether solid-liquid surface or liquid-gas interface are concerned. For instance on the liquid-gas interface:

$$\Sigma_n - \kappa\gamma = 0 \tag{54}$$

and on the solid-liquid surface:

$$\vec{u}\cdot\vec{n} = 0 \tag{55}$$

$$\Sigma_t = 0 \tag{56}$$

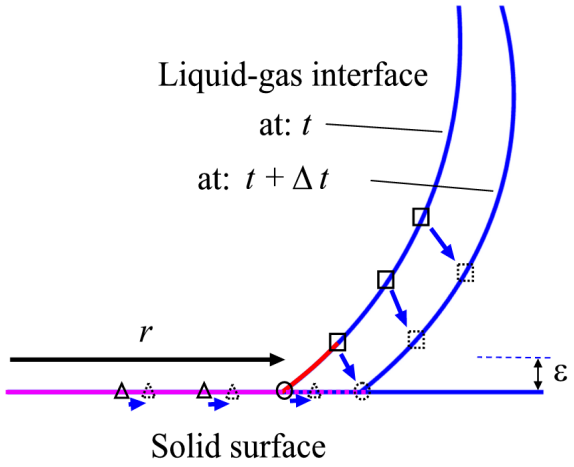

Figure 1. Switch from free boundary condition to sliding boundary condition, as the droplet spreads and its free interface rolls up.

The point is during the droplet spreading a given part of the liquid-gas interface will rolls up and sticks up on the solid-liquid interface (see figure 5). This what we call projection of the liquid-gas interface. So, at time t, the boundary conditions applied on the red element are given by (56) while at time $t+\Delta t$ they will be given by (54) and (55). Two questions have to be answered. The first is how to switch between one type of boundary conditions to another. The second is what is the criterion that permits to choose one boundary condition rather than the other. We will answer to the second question first. The criterion is merely belonging to the solid-liquid interface or not. In other words, a point belong to the solid-liquid surface if it satisfies:

$$r - r_p < 0 \tag{57}$$

$$z - z_p < \epsilon \tag{58}$$

where (r_p, z_p) are the coordinates of the triple-point and ϵ is a tolerence much smaller than the smallest mesh size. Now the answer to the first question. The easiest way in COMSOL Multiphysics to switch from a boundary condition to another is to use weak-constraint handling through Lagrange multipliers in the following way:

$$\begin{aligned} &\hat{\lambda}\left[\left(\Sigma_n - \kappa\gamma\right) H(z - z_p - \epsilon)\right. \\ &\left. + \left(\vec{u}\cdot\vec{n}\right) H(-z + z_p + \epsilon)\right] - \lambda\left(\widehat{u\vec{\cdot}n}\right) = 0 \end{aligned} \tag{59}$$

where H is the Heaviside function (or a smoothed function that behaves alike). The triple-point coordinate, given by (r_p, z_p), have to be evaluated. The triple line z-co-ordinate

is obtained in straightforward manner, while the r-co-ordinate has to be conputed in the following way:

$$r_p = max\left\{r/P(r,z) \in \Gamma \wedge z - z_p < \epsilon\right\} \tag{60}$$

Unfortunately there is no way to directly obtain a maximun of a space dependent-function using COMSOL Multiphysics. However an indirect tricky (and time consuming) way to do so exists using the following property of positive functions ($a < b$):

$$\lim_{n\to+\infty} \frac{1}{b-a}\left(\int_a^b |f(x)|^n\, dx\right)^{1/n} = \max_{x\in[a,b]}\left(|f(x)|\right) \tag{61}$$

In our case the function to be maximized along the concerned boundaries is $rH(z_p - z + \epsilon)$. From a numerical point of view, and according to (61), the larger n is the better the maximum is computed. A bad news is that this argument is no longer correct because round off error accumulated during computing increases as n gets larger. The other bad news is that error increases with respect to n before getting a good approximation of the desired maximum, as shown by figure 6.

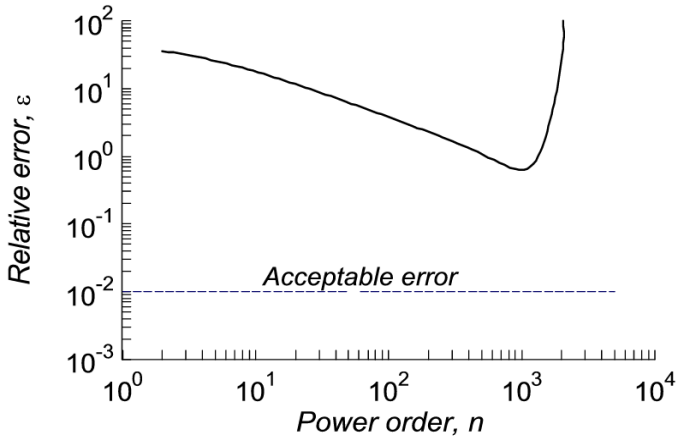

Figure 6. Variations of the relative error with respect to the power n used in expression (61).

Now the good news (even here there are some). Another way to get the maximum is to use the expression (61) with a relative small value of n not once only, but many times in a iterative way. Figure 7 shows the variations of the relative error with the iteration number when $n = 64$.

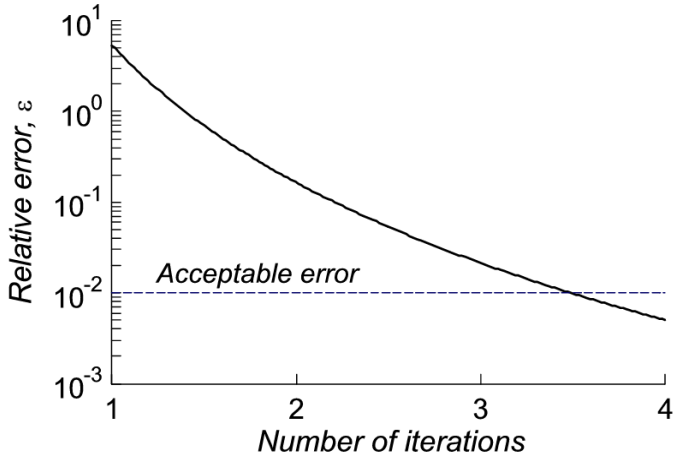

Figure 7. Variations of the relative error with the number of iteration of expression (61) with $n = 64$.

Unfortunately, we did not manage to obtain neither large mesh distortions nor valid result of the spreading of the droplet when using Comsol Multiphysics 3.2.. Therefore, we do not present any figures obtained.

5 Numerical results obtained using Micralef software: spreading phase

In the present section, we present some preliminary numerical results of the spreading of a droplet impacting a solid surface. These results are obtained when solving the model presented previously (see also [7] for more details). The numerical scheme and algorithms have been implemented into Micralef, a home-developed software.

We consider a water droplet. The reference length and velocity are: $L_{ref} = 2.3\,mm$, $U_{ref} = 0.98\,m/s$. Then, $Re = \dfrac{\rho U_{ref} L_{ref}}{\mu} \approx 46$, Weber number $We = \dfrac{\rho U_{ref}^2 L_{ref}}{\sigma} \approx 68$ and $C_a = \dfrac{\sigma}{\mu U_{ref}} \approx 1.5$.

5.1 Spreading phase using a Tanner type law

We start by testing the efficiency of the algorithm implemented into Micralef. To this end, we consider the Tanner type law as triple line dynamic modelling, and we make fit this law with available experimental data, see e.g. [1], related to a spreading phase. The triple

point position is imposed at each time. We focus on the volume conservation, the height and diameter of the spreading (splat radius) and the deformation of the mesh (number of re-meshing necessary), see Fig. 5.1. With a time step $dt = 5.10^{-5}s$ and a coarse mesh (≈ 600 elements), volume lost after 650 iterations is roughly 3%. The loss occurs mainly at the very beginning of spreading. Mesh transport is efficient hence re-meshing occurs only when a projection occurs (see previous section). From a qualitative point of view, numerical results are similar to experimental data presented in [1].

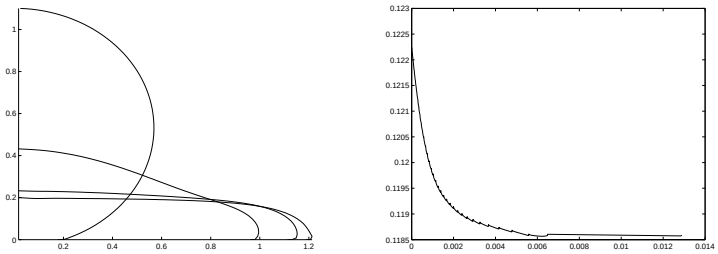

Figure 5.1: Spreading: droplet profiles and volume in function of time steps

5.2 Spreading phase using an implicit triple line dynamics

We consider the spreading phase using the Shikhmurzaev's model and the algorithm presented previously. The source term g in the generalized Navier slip condition acts like a control on the position of the triple point. A decrease for $g < 0$ translates in a faster spreading. If g is small, this influence is likely not to be seen in the first part of the spreading, when the inertial forces dominate all other forces in presence. By increasing $|g|$ over a threshold, which depends on β, it accelerates the spreading and modifies the flow nature. For $|g|$ large, large interface distortions occurs and mesh generator fails to re-mesh.

In order to observe the influence of the g-term, we fix $\beta = 1000$, and we perform 1000 iterations with a step size of $dt = 5.0 \times 10^{-6}$ for $g \in \{-10, -100\}$, Fig. 5.2. One can observe that after the inertial phase, the triple point position is farther to the right with an increasing $|g|$.

Fig. 5.3 shows a droplet subjected to large distortion (similar to breakup) when $|g|$ is getting larger. Simulation can go no further because of the change of topology it implies is not supported by the ALE method we are using.

A first attempt to simulate the recoiling phase. We try now to see the influence of the sign of parameter g on the droplet behavior. Since we have seen that parameter g monitor the spreading, it is also supposed to do so with recoiling. In Fig. 5.4, simulation is started with initially spread droplet. Also here, the ratio $\dfrac{-g}{\beta}$ significantly influences the behavior

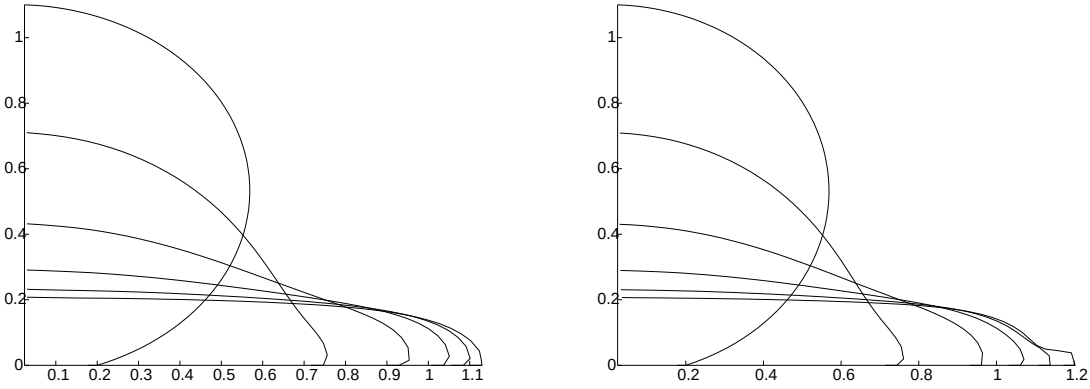

Figure 5.2: Droplet profiles and triple point position (right) for $(\beta, g) = \{(10^3, -10), (10^3, -10^2)\}$
.

of the droplet while parameter h influences more particulary on the droplet curvature than on the triple-point position.

6 Conclusion

We presented the Navier–Stokes free surface equa-tions with a generalized Navier slip condition as it appears both in the Shikhmurzaev theory and in the Qian-Wang-Sheng theory. Furthermore, we wrote the surface equations arising in the Shikhmurzaev theory in the surface co-ordinate system and in weak form. An anlmogy with the surfactant equations si done; and these equations are detailled.
Then, we consider as test case the droplet impact onto a solid surface. An attempt to model and simulate using Comsol Multiphysics 3.2 is done. Unfortunately, we show that the ALE implementation in Comsol Multiphysics 3.2 is not robust enough to handle large mesh distorsion (as it appears in many surface flows). In addition, some important quantities such as curvature, are not easily computable. We present some tricks to circumvent this latter problem.
Finally, we present some preliminary numerical results using our freeware Micralef. Miocralef has been developed especially to solve the present equations and the algorithms we elaborated (ALE, spreading and recoiling algorithms). We obtain qualitatively good results when siumulating the spreading phase. An efficient numerical model for the recoiling phase is still under progress.

Acknowledgements. The authors would like to thank P. Pham, A. Glière from LETI -CEA- Grenoble, France, for their numerous and fruitful discussions and remarks. This study has been co-funded by a research program led by Prof. A. Soucemarianadin (UJF); the authors thank him very much.

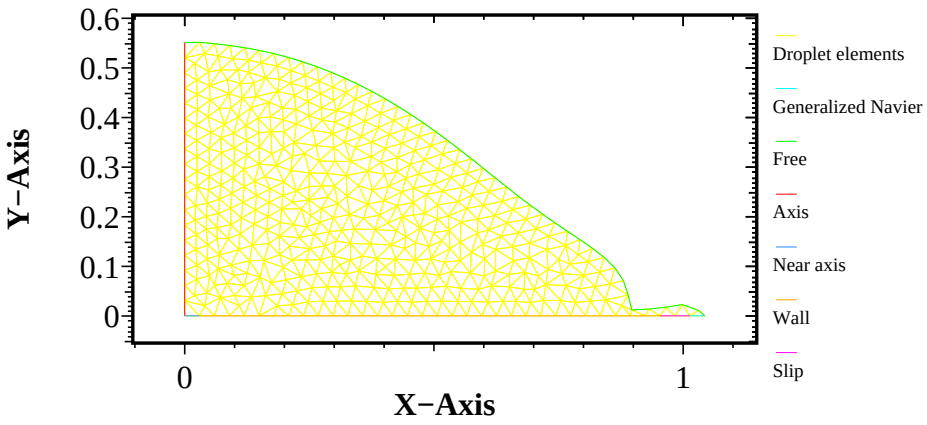

Figure 5.3: Large interface distortion, similar to a droplet breakup $g - 10^4\}$

.

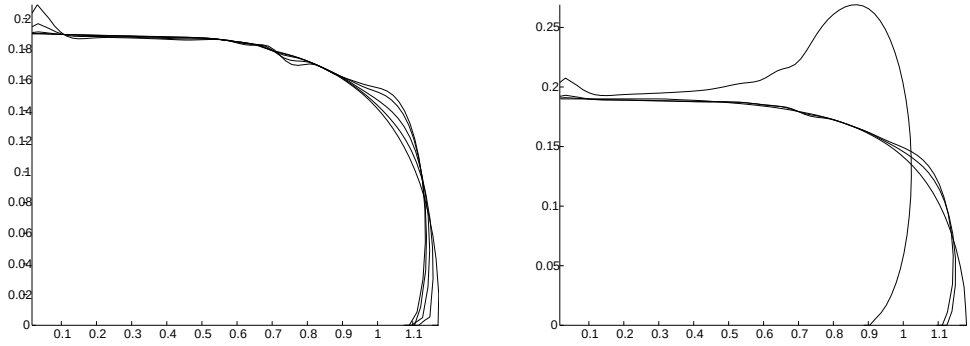

Figure 5.4: Droplet profiles and triple point position (right) for $(\beta, g, h) = \{(500, 100, 0), (100, 100, -10)\}$

.

References

[1] Allaman S., Desie G., Vadillo D., Soucemarianadin A. : Impact and spreading of microdrops onto solid substrates. MÈcaniques & Industries **4** (2003) 443-455.

[2] Blake T.D., Bracke M., Shikhmurzaev Y.D. : Experimental evidence of nonlocal hydrodynamic influence on the dynamic contact angle. Physics of fluids **11** (1999) 1995-2007.

[3] Dussan V.E.B., Davis S.H. : On the motion of a fluid-fluid interface along a solid surface. J. Fluid Mech. **65** (1974) 71-95 .

[4] Hughes, T.J.R, Liu, W., Zimmermann, T.K., Lagrangian-Eulerian Finite Element Formulation for incompressible Viscous Flows, *Comp. Methods Applied Mech. Eng.*, vol. 29, pp. 329-349 (1981).

[5] Maury, B., Pironneau, O., *Characteristics ALE method for unsteady free surface flows with surface tension* Z. Angew. Math. Mech., **76**, pp 613 (1996).

[6] Monnier J. : Modélisation Numérique de la Ligne Triple. Internal Report LMC-IMAG, Summer school CEA Grenoble. September (2003).

[7] Monnier J., Benselama A., Cotoi I. : Flow patterns in the vicinity of triple line dynamics arising from a local surface tension model. Int. J. Comput. Multiscale Comput. Eng. Accepted, To appear, 2007.

[8] Monnier J., Witomski P. : Analysis of a Local Hydrodynamic Model with Marangoni Effect. J. Sc. Comp. **21** (2004) 369-403.

[9] Pomeau Y. : Recent progress in the moving contact line problem: a review. C.R. Mécanique. **330** (2002) 207-222.

[10] Saramito, P., Roquet, N. and Etienne, J.: Rheolef: A C++ finite element library. http://ljk.imag.fr/membres/Pierre.Saramito/rheolef/

[11] Shikhmurzaev Y.D. : The moving contact line problem on a smooth solid surface. Int. J. Multiphase Flow. **19** (1993) 589-610.

[12] Slatery J.C, L. Sagis, E.S. Oh: Interfacial transport phenomena. 2nd edt. Springer (2007).

[13] Qian T., Wang X-P., Sheng P. : Generalized Navier boundary condition for the moving contact line. Comm. Math. Sci. **1** (2003) 333-341.

A Surface differential operators expressions

Let the surface S be affected by an arbitrary co-ordinate system (u^1, u^2) and let φ be a scalar function and A^α a contravariant vector field both defined in S. The metric tensor of S is defined by the expression, Aris,

$$a_{\alpha\beta} = \sum_{i=1}^{3} \frac{\partial y^i}{\partial u^\alpha} \frac{\partial y^i}{\partial u^\beta}, \quad y \in S \tag{62}$$

y^i is the i-th co-ordinate of the current point of S, expressd in the Reimannian space E_3. It is shown that the metric tensor $a_{\alpha\beta}$ is a symmetric covariant second order tensor. We define the short-hand determinant of $a_{\alpha\beta}$ by

$$a = det|a_{\alpha\beta}| = a_{11}a_{22} - (a_{12})^2 \tag{63}$$

The Kronecker delta δ^α_β which is 1 if $\alpha = \beta$ and zero otherwize, permits to define the conjugate metric tensor by

$$a^{\alpha\gamma}a_{\gamma\beta} = a^{\gamma\alpha}a_{\gamma\beta} = a^{\alpha\gamma}a_{\beta\gamma} = \delta^\alpha_\beta \tag{64}$$

1. The surface divergence of A^α is given by

$$\nabla_s \cdot A = A^\alpha_{,\alpha} = \frac{1}{a^{\frac{1}{2}}} \frac{\partial}{\partial u^\alpha} \left(a^{\frac{1}{2}} A^\alpha \right) \tag{65}$$

2. The surface gradient of φ is given by

$$\nabla_s \varphi = \varphi_{,\alpha} = \frac{\partial \varphi}{\partial u^\alpha} \mathbf{e}_\alpha \tag{66}$$

where $\mathbf{e}_\alpha$ is the α-th tangent vector to S.
3. The surface Laplacian of φ is given by

$$\nabla^2_s \varphi = a^{\alpha\beta}\varphi_{,\alpha\beta} = \frac{1}{a^{\frac{1}{2}}} \frac{\partial}{\partial u^\beta} \left(a^{\frac{1}{2}} a^{\alpha\beta} \frac{\partial \varphi}{\partial u^\alpha} \right) \tag{67}$$

4. The surface curl of A^i is given by

$$\nabla_s \times A = \epsilon^{\alpha\beta} A_{\alpha,\beta} \vec{n} = a^{-\frac{1}{2}} \left(\frac{\partial A_2}{\partial u^1} - \frac{\partial A_1}{\partial u^2} \right) \vec{n} \tag{68}$$

$\epsilon^{\alpha\beta}$ is the (two dimensional) permutation tensor defined by

$$\epsilon^{11} = \epsilon^{22} = 0, \ \epsilon^{12} = 1 \ and \ \epsilon^{21} = -1 \tag{69}$$

and $\vec{n}$ is the normal vector to S.
5. The elementary surface is given by

$$ds^2 = a_{\alpha\beta} du^\alpha du^\beta \tag{70}$$

As surface equations are involved, the whole issue is first to choose the surface co-ordinate system that maps the surface S and second to get the associated metric tensor $a_{\alpha\beta}$ in order to express correctly each differential operator according to the mapping co-ordinates as discribed formerly.

Application: axisymmetric surface S

In the case where $\left(u^1, u^2\right) = (s, \phi)$, as shown in figure 1, and considering the axisymmetry of the geometry, then we can write

$$x = r\cos\phi \tag{71}$$
$$y = r\sin\phi \tag{72}$$

Then

$$\begin{aligned}
\frac{\partial x}{\partial s} &= \cos\phi\frac{\partial r}{\partial s} \\
\frac{\partial x}{\partial \phi} &= -r\sin\phi \\
\frac{\partial y}{\partial s} &= \sin\phi\frac{\partial r}{\partial s} \\
\frac{\partial y}{\partial \phi} &= r\cos\phi
\end{aligned}$$

So, according to (62)

$$\begin{aligned}
a_{ss} &= \left(\frac{\partial x}{\partial s}\right)^2 + \left(\frac{\partial y}{\partial s}\right)^2 + \left(\frac{\partial z}{\partial s}\right)^2 \\
&= \left(\frac{\partial r}{\partial s}\right)^2 + \left(\frac{\partial z}{\partial s}\right)^2
\end{aligned} \tag{73}$$

$$\begin{aligned}
a_{s\phi} &= a_{\phi s} = \frac{\partial x}{\partial s}\frac{\partial x}{\partial \phi} + \frac{\partial y}{\partial s}\frac{\partial y}{\partial \phi} + \frac{\partial z}{\partial s}\frac{\partial z}{\partial \phi} \\
&= -\frac{\partial r}{\partial s} r\cos\phi\sin\phi + \frac{\partial r}{\partial s} r\cos\phi\sin\phi = 0
\end{aligned} \tag{74}$$

$$a_{\phi\phi} = \left(\frac{\partial x}{\partial \phi}\right)^2 + \left(\frac{\partial y}{\partial \phi}\right)^2 + \left(\frac{\partial z}{\partial \phi}\right)^2 = r^2 \tag{75}$$

and

$$a = \det\left(a_{\alpha\beta}\right) = r^2\left[\left(\frac{\partial r}{\partial s}\right)^2 + \left(\frac{\partial z}{\partial s}\right)^2\right] \tag{76}$$

Thus

$$\begin{aligned}\nabla_s \cdot A &= A^{\alpha}_{,\alpha} = \frac{1}{a^{\frac{1}{2}}}\frac{\partial}{\partial u^{\alpha}}\left(a^{\frac{1}{2}}A^{\alpha}\right) \\ &= \frac{1}{r\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{1}{2}}}\frac{\partial}{\partial s}\left(r\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{1}{2}}A^s\right) \end{aligned} \tag{77}$$

Whereas

$$\nabla_s\varphi = \varphi_{,\alpha} = \frac{\partial\varphi}{\partial u^{\alpha}}\vec{e}_{\alpha} = \frac{\partial\varphi}{\partial s}\vec{e}_s \tag{78}$$

And

$$\nabla_s^2\varphi = \frac{1}{r\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{1}{2}}}\frac{\partial}{\partial s}\left(r\left[\left(\frac{\partial r}{\partial s}\right)^2+\left(\frac{\partial z}{\partial s}\right)^2\right]^{\frac{3}{2}}\frac{\partial\varphi}{\partial s}\right) \tag{79}$$

B Short presentation of Comsol Multiphysics 3.2 software

This section is dedicated to describe the numerical tool used to implement and simulate the triple line dynamics, to name COMSOL Multiphysics 3.2b. COMSOL Multiphysics is an interactive environment for modeling and solving many kinds of scientific and engineering problems based on partial differential equations (PDEs). One accesses the power of COMSOL Multiphysics as a standalone product through a flexible graphical user interface, or by script programming in the COMSOL Script language or in the MATLAB language. As noted, the underlying mathematical structure in COMSOL Multiphysics is a system of partial differential equations. Three ways of describing PDEs are provided through the following mathematical application modes:

1. Coefficient form, suitable for linear or nearly linear models;
2. General form, suitable for nonlinear models;
3. Weak form, for models with PDEs on boundaries, edges, or points, or for models using terms with mixed space and time derivatives.

Using these application modes, one can perform various types of analysis including:

1. Stationary and time-dependent analysis
2. Linear and nonlinear analysis
3. Eigenfrequency and modal analysis

When solving the PDEs, COMSOL Multiphysics uses the finite element method (FEM). The software runs the finite element analysis together with adaptive meshing and error control using a variety of numerical solvers. PDEs form the basis for the laws of science and provide the foundation for modeling a wide range of scientific and engineering phenomena. Therefore one can use COMSOL Multiphysics in many application areas.

The time-dependent solver As mentioned before, COMSOL Multiphysics proposes many solver modes, such stationary linear, stationary nonlinear, eigenvalue and time-dependent solvers. Each solver uses a specific algorithm. The time-dependent solver algorithm, which we are burdened with, and the setting parameters are presented in this section. To use the time-dependent solver, one has to go to the **Solver Parameters** dialog box, find the **Solver** list, and select **Time dependent**.

Specifying output times. In the **Time stepping** area, one has to enter the output in the **Times** edit field, which contains a vector of times at which one wants the solution to the time-dependent model. The matlab syntax `0:0.1:1` represents a vector of times starting at `0` steps of `0.1` up to `1`. The relevant time span depends on the model's dynamics.

Setting tolerances. To specify the accuracy and sometimes influence the performance, one has to set absolute and relative tolerance parameters for the time-dependent solver. In the **Relative tolerance** edit field, one can enter a positive number (default = `0.01`). In the **Absolute tolerance** edit field, one can enter a single positive number (default = `0.001`) or a space-separated list whose entries alternate between degree-of-freedom names and positive scalars. More precisely, let U be the solution vector corresponding to the solution at a certain time step, and let $\vec{E}$ be the solver's estimate of the local error in U commited during the current time step. The step is accepted if

$$\left(\frac{1}{N} \sum_i \left(\frac{\left| \vec{E}_i \right|}{A_i + R\,|U_i|} \right)^2 \right)^{1/2} < 1$$

where A_i is the absolute tolerance, R is the relative tolerance and N is the number of degrees of freedom.

The time-dependent solver algorithm. The finite element discretization of the time-dependent system of partial differential equations problem is

$$0 = L\left(U, \dot{U}, \ddot{U}, t\right) - N\left(U, t\right)^t \Lambda$$

$$0 = M\left(U, t\right)$$

which is reffered to as the *method of lines*. Before solving this system, the algorithm eliminates the Lagrange multipliers Λ. If the constraints $0 = M$ are linear and time independent, the algorithm also eliminates them from the system. otherwise it keeps the constraints, leading to a algebraic-differential system. To solve the above ordinary/algebraic -differential

equations system, COMSOL Multiphysics uses a version of the Differential Algebraic equation Solver Package, DASPK, originally proposed by L. Petzolds. This algorithm uses variable-order variable-stepsize backward differential formulae. Thus the solver is an implicit time-stepping scheme, which implies that it must solve a possibly nonlinear system of equations at each time step. It solves the nonlinear system using Newton iteration, and it then solves the resulting sytems with an arbitrary COMSOL Multiphysics linear system solver. The linearization of the above system used in the Newton iteration is

$$E\ddot{V} + D\dot{V} + KV = L - N^t\Lambda$$

$$NV = M$$

where $E = -\partial L/\partial\ddot{U}$ is the mass matrix, $D = -\partial L/\partial\dot{U}$ is the damping matrix, $K = -\partial L/\partial U$ is the stiffness matrix. When $E = 0$, D is called the mass matrix.

C Short presentation of our home-developed software Micralef

Micralefe is a C++ finite element software (home-developed) solving 2D axisymmetric Navier-Stokes flow with free surface dynamics and curvature forces. It treats of a droplet impact on a solid substrate. The dynamics of the free surface is described using an ALE formulation (Arbitrary Lagrangian Eulerian), while the dynamic triple line modelling is based on the Shikhmurzaev's theory.The code is based on the Rheolef C++ finite element library. Micralefe capabilities include the dynamics of the contact angle, the triple point dynamics (liquid-gas-solid), variable surface tension coefficients, slip type boundary conditions and automatic mesh refinement. It has been developed in order to obtain numerical results presented in the present article.
The code modularity allows us to plan further developments such as coupling with a surfactant model or thermal model.
Micralef is avalaible under request , see URL: http://ljk.imag.fr/membres/Jerome.Monnier. The C++ source codes are commented while a short user's documentation is available. Potential users must be expert in computer sciences (C++ programming, compilers, Linux and open-source libraries).

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399