A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces

Mathieu Coquerelle, Stéphane Glockner

A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces

Mathieu Coquerelle, Stéphane Glockner

Université de Bordeaux, CNRS, UMR 5295 - Bordeaux INP - 16, avenue Pey-Berland, 33607 PESSAC Cedex, France
mathieu.coquerelle@u-bordeaux.fr, glockner@ipb.fr

Abstract

We propose a fourth-order accurate and robust curvature estimation and extension algorithm in a level set framework for the transport of the interface. The method is used to efficiently calculate surface tension forces for two-phase flows, based on the Continuum Surface Force approach. In this framework, accuracy mostly relies on the precise computation of the surface's curvature that we propose to accomplish in a two-step algorithm. First by computing a reliable fourth-order curvature estimation from the level set function. Secondly by extending rigorously this curvature in the viscinity of the surface, following the Closest Point principle. The algorithm is easy to implement and to integrate to existing solvers, furthermore naturally extensible to 3D. We propose a detailed analyze of the geometrical and numerical criteria that are responsible of the appearance of spurious currents, a phenomenon widely explored in the community in various numerical frameworks. We study the response of the novel numerical method on state-of-the-art test cases showing that it permits to greatly reduce parasitic currents, converging at fourth-order regarding spatial discretization, two orders better than common results from the litterature. We also show the necessity of high-order transport methods for the surface by studying the case of the 2D advection of a column at equilibrium proving the robustness of the proposed approach. The algorithm is validated on more complex test cases such as a rising bubble.

Keywords: surface tension, curvature computation, level set method, spurious currents, two phase flow, finite element method, balanced force algorithm, closest point method

1. Introduction

Surface tension is a key mechanical force in multi-phase flows where it plays a particularly important role in the hydrodynamics at small scales. Rising bubbles and water drops are common examples of such flows (see for example Yarin and Weiss [1], Bhaga and Weber [2], Clip et al. [3]). The surface tension force lies at the interface between two phases and is thus singular at that location. This singularity makes it a complicated problem in a continuum mechanics point of view when coupled with the Navier-Stokes equations.

In a numerical point of view, this singular force is challenging as it is based on the precise computation of the surface, its normal as well as its curvature. A variety of methods has been introduced to localize the surface advected by the flow, mainly, in a Lagrangian framework the front-tracking method Tryggvason et al. [4], in a Eulerian framework the Volume of Fluid (VOF) method Gueyffier et al. [5] or the level set (LS) method Sussman et al. [6]. Those three widely used approaches offer advantages and drawbacks such that they are still studied and developed today. We based our work on the level set framework that permits to easily capture topological changes and offers the advantage of usualy simpler algorithms and natural extension from 2D to 3D. However, in comparison to the Lagrangian representation, the precise localization of the surface is lost and the methods usually suffer from more volume loss than VOF methods for example. A wide range of hybrid technics has been proposed to alleviate these difficulties, like the particle level set Enright et al. [7], the CLSVOF methods Sussman and Puckett [8] or high-order semi-Lagrangian particles Cottet et al. [9]. As we will demonstrate in this article, an accurate transport of the surface is mandatory in order to attain high-order computation of the dependant curvature. In a Eulerian framework, the surface is immersed in the fluid through the use of a Dirac's mass and a Heaviside function which characterizes the phases.

The surface tension force is commonly introduced in the Navier-Stokes equations through an external force localized at the interface. The Continuum Surface Force (CSF) model introduced in Brackbill et al. [10] gives an elegant solution based on the spreading of the force in a small neighbourhood around the interface. In the last decade, works have been focused on the reduction of spurious currents, arising in the viscinity of the surface. Those currents are numerical parasitic velocities introduced in the flow by errors on estimation of the surface tension force. Following the observations done by François et al. Francois et al. [11], in the CSF model the creation of such currents are due to:

1. non-coherent numerical differenciation in the solving of the Navier-Stokes equations,

2. non-coherent computations of the surface's position, normal and curvature.

The first criterion has been recently solved by the introduction of the balanced-force algorithm Francois et al. [11] which is based on coherent numerical schemes for the computations of the surface tension force and the associated pressure jump. This technic has been proved to reduce spurious currents down to machine precision when the curvature is prescribed exactly.

The second criterion is still investigated today through the accuracy of the interface's normal and curvature computations, which are direct derivatives of the surface, thus also singular, and need to be treated adequatly in a Eulerian framework. For VOF methods, the height-field Cummins et al. [12] (and further improvements) approach has been proved to be precise and drasticly reduce spurious currents as showed in Popinet [13]. However the algorithm is quite complex to develop in 2D, even more in 3D, and its accuracy is highly dependant on the underneath VOF transport precision, which is also today still a complex problem. For level sets, a curvature's extension, based on a closest point algorithm Macdonald and Ruuth [14], has been efficiently used in Herrmann [15], for unstructured grids, which permits to reduce spurious current much more than previous technics. The implementation of the method is quite simple and is easily extensible in 3D.

Following the CSF model we propose in the present paper an extension of Herrmann's work Herrmann [15] for the curvature computation for level sets, while remaining in a Cartesian framework in our case. Our method permits to attain a fourth-order convergence rates, regarding spatial discretization, for the curvature's errors and for the spurious currents reduction, as long as a sufficiently accurate transport method is employed. An improved closest point algorithm is proposed, which permits to obtain good precision for general surfaces. We study the accuracy of the method first on geometrical cases and then on dynamical simulations where spurious currents are drasticly reduced. We also propose a qualitative analysis of the complex simulation of the bubble rise problem.

2. Governing equations

2.1. Terminology

Before presenting the equations we present the terminology and notations that will be used through the whole document:

Symbol	Definition	Symbol	Definition
ρ	Fluid's density	Γ	A surface
μ	Fluid's dynamic viscosity	$\mathbf{n}$	A surface's normal
$\mathbf{u}$	Fluid's velocity	τ	A surface's tangent
p	Pressure	κ	A surface's curvature
σ	Surface tension coefficient	ϕ, ψ	Level set functions
H	The Heaviside function	$\widetilde{\kappa}$	Discrete representation for any κ
δ	The Dirac mass	κ_{ex}	Exact value for any κ
h	The spatial discretization step	κ_σ	The curvature's standard deviation
Δt	The temporal discretization step	$\kappa_{\sigma,\mathbf{n}}$	The curvature's normal standard deviation
$\mathbf{x}_{ij}$	The discrete value of $\mathbf{x}$ at i, j	RMS	Root Mean Square

Table 1: Terminology.

2.2. Two phases fluid flow equations in presence of surface tension forces

The incompressible Navier-Stokes equations with variable densities and viscosities following the continuum-surface-force (CSF) approach originaly proposed by Brackbill et al. Brackbill et al. [10] can be written:

$$\begin{aligned} \rho(\partial_t \mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u}) &= -\nabla p + 2\nabla\cdot(\mu\mathbf{D}) + \sigma\kappa\delta_\Gamma\mathbf{n} \\ \partial_t\rho + (\mathbf{u}\cdot\nabla)\rho &= 0 \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{1}$$

with $\mathbf{u}$ the fluid velocity, ρ (resp. μ) the fluid variable density (resp. dynamic viscosity), $\mathbf{D}$ the deformation tensor and p the pressure field. The term $\sigma\kappa\delta_\Gamma\mathbf{n}$ is singularly located on the surface Γ separating two phases by the Dirac delta function δ_Γ, oriented in the normal direction $\mathbf{n}$, with curvature κ and physical surface tension coefficient σ. Equation 1 stands for a constant σ along the surface between two phases which is a common hypothesis in most fluid dynamic computations.

2.3. Level set representation for two phases flows

In a level set fashion the phases of the fluid are spatially identified in n dimensions using a function $\phi : \mathbb{R}^n \to \mathbb{R}$ such that each phase stands on one side of a certain level set value ϕ_0. Thus, they are separated by the surface

$$\Gamma = \{\mathbf{x} | \phi(\mathbf{x}) = \phi_0\}. \tag{2}$$

For the sake of simplicity, we choose $\phi_0 = 0$ implying that the first (resp. second) phase stands where ϕ is negative (resp. positive).

The local concentration c_i - or charateristic function - of the i^{th} phase is written

$$c_i(\mathbf{x}, t) = H_i(\phi(\mathbf{x}, t)),$$

where $H_1(y) = H(y) = 1 - H_2(y)$ and $H(y)$ is the Heaviside function such as $H'(y) = \delta(y)$.

The density ρ is expressed in the whole domain as a function of ϕ:

$$\rho(\mathbf{x}, t) = H(\phi(\mathbf{x}, t))\rho_1 + (1 - H(\phi(\mathbf{x}, t)))\rho_2, \tag{3}$$

where ρ_i is the constant density of the i^{th} phase. A similar equation represents the viscosity μ.

As in common level set methods, the function ϕ is passively transported by the fluid through the advection equation:

$$\partial_t \phi + (\mathbf{u} \cdot \nabla)\phi = 0 \tag{4}$$

which permits to implicitely follow the surface's position during time.

The normal and curvature fields can be computed in the whole domain as:

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \tag{5}$$

and

$$\kappa = \nabla \cdot \mathbf{n} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \tag{6}$$

Equations 5 and 6 coincide with the normal and curvature of the surface Γ for points $\mathbf{x}$ such as $\phi(\mathbf{x}) = \phi_0$.

Due to numerical discretization in the level set Eulerian framework we will work with a smooth representation of those quantities where δ_Γ and H are not singular but spread in the vicinity of Γ. Thus we will use the term **extension** for any quantity that physicaly stands on the surface but is numerically spatially extended around it.

2.4. Problems arising with surface tensions forces

As stated in the introduction, a qualitative computation of surface tension forces - ie. minimizing spurious currents - is based on two criteria:

Claim 1. The consistent numerical discretization of the pressure and surface tension terms.

Claim 2. The rational computation of the surface's curvature.

By rational computation we mean a curvature computation that assures the three following aspects:

Proposition 3. *Fast convergence to the exact curvature.*

Proposition 4. *Minimized variation along the surface.*

Proposition 5. *Minimized variation along the surface's normal.*

Several efficient and accurate numerical methods have been developped over the last decade focusing on the curvature and the surface tension term computation. The balanced-force algorithm used in different flow simulation frameworks (eg. based on the Ghost Fluid Method Francois et al. [11], VOF Popinet [13], level set on unstructured grids Herrmann [15]) has been proved to greatly reduce the numerical errors by satisfying claim 1 with adapted methods for the curvature computation to satisfy claim 2.

Our flow solver presented in section 3 is based on the same balanced-force principle and an improved curvature computation that comply with propositions 3, 4 and 5 .

For the sake of comprehension we explain the importance of those 3 points based on the 2D case of the static column equilibrium which has been widely used in the literature as a minimal and representative case.

2.4.1. Fast convergence to the exact curvature

The first criterion is essential to computationally converge toward the physical experiments, as pictured in figure 1a. Even in a stable simulation, a significant error on the the curvature can drasticly change the hydrodynamics. It becomes particularly important when the computational space is refined as high curvatures can arise and often leads the dynamic at those scales.

Hence it is essential to have a precise and fast convergent computation of the curvature, at least at the same order of the flow solver. In today's common solvers the order of errors on the Navier-Stokes equations is $O(h^2)$ where h is the space discretization step. It thus requires an equal or better order for the surface tension forces, hence the curvature.

Errors in the second derivative computation.

Proposition 6. *The error in the second derivative is of order $m-2$ when the error on the surface representation is of order m.*

To prove proposition 6, we first illustrate the impact of numerical errors on the second derivatives of a function. This demonstration can be extended to the general case of surface and curvature computation, whatever the representation of the surface is, Eulerian or Lagrangian.

Let ψ be a 1D function of x, sufficiently differentiable. Let ϕ be a perturbed version of ψ such as $\phi(x,h,m) = \psi(x) + e(x,h,m)$, where $e(x,h,m)$ represents a perturbation of amplitude of order h^m with spatial frequency $1/h$ (where, in a numerical computation, h is the spatial discretization, ie. the smallest variation that can be captured). Then, the n^{th} derivative of ϕ will be a function of the n^{th} derivatives of ψ and $e(x,h,m)$:

$$\frac{\partial^n \phi(x,h,m)}{\partial x^n} = \frac{\partial^n \psi(x)}{\partial x^n} + \frac{\partial^n e(x,h,m)}{\partial x^n}.$$

Then the perturbation on the n^{th} derivative of ϕ will be of the order h^{m-n}. Hence, totaly independentely of the accuracy of the underlying numerical scheme used to discretized the quantities (especialy the curvature), if $m > n$ then the function $\frac{\partial^n \phi}{\partial x^n}$ will converge toward $\frac{\partial^n \psi}{\partial x^n}$ at a positive rate of $O(h^{m-n})$, otherwise it will not converge and lead to errors $> O(1)$ increasing as $h \to 0$.

A detailed example is given for illustration in appendix A.

Errors in the initialization and transport of the surface. When a function ϕ represents the surface, should it be in a Lagrangian or Eulerian fashion, given the previous demonstration, if the error in its initialization and/or transport is less than the order 2, then the errors in the second derivatives of ϕ - hence the curvature - will be at least $O(1)$. This will lead to an increasing error in the flow as the spatial discretization tends to 0.

Hence in order to have at least a second-order error in the computation of the curvature it is mandatory to **initialize** and **transport** the surface with errors less than the order h^4 compared to the exact solution. This is true whatever the errors in the velocity field used to transport the surface as, to be comparable, the exact solution has also to be transported by the perturbed flow.

Examples for second-order precision

- In a lagrangian fashion, the discretized points have to stand closely than h^4 to the exact surface. The velocity interpolation at those points must be at least fourth-order.
- In a level set/signed distance fashion, the distance to the surface has to be computed at least at the order four, which is easily attainable for common geometry. The transport equation must be solved at the same order. Also, reinitialization algorithms must ensure a similar order or precision.
- In a VOF fashion, the fraction of volume occupied by the phase has to be computed at least at the order four, which is not trivialy obtainable even for common geometry. One has to be careful as if a set of discrete small volumes are used inside the cells to integrate their volume, the convergence is not dependent on h but on a fixed number of volumes, hence will not lead to a spatially converging scheme. The transport equation also has to be at least forth order. Until recently, no VOF transport method was sufficiently precise to attain high-order dynamic computation of curvature, recent methods such as DRAC Zhang [16] are fourth-order accurate in space.

2.4.2. Minimized variation along the surface

The overall variation of the curvature can be expressed through the standard deviation. For the 2D static column case, it is evaluated on the surface as:

$$\tilde{\kappa}_\sigma = \sqrt{\int_\Gamma (\tilde{\kappa} - \tilde{\kappa}_{mean})^2} \tag{7}$$

where the tilda denotes discrete measures, the σ symbol is used here to mean standard deviation and $\tilde{\kappa}_{mean}$ is the numerically computed mean curvature:

$$\tilde{\kappa}_{mean} = \int_{\Gamma} \tilde{\kappa}. \tag{8}$$

As stated in Popinet [13], this second criterion is numerically relevant as, in the case of a balanced force algorithm, it permits to reduce the spurious currents toward the machine precision. This statement is true even in the case of a poor evaluation of the absolute physical curvature, ie. even when $|\tilde{\kappa}_{mean} - \kappa_{exact}|$ is high.

For the static column equilibrium problem, the presence of oscillations of the curvature value along the interface will result in the appearence of capilarity waves, thus undesired spurious currents in the whole simulation. It is important to note that this criterion is hardly extensible to the general case where the curvature is not constant along the surface. However, one can picture those small variations in the curvature as a bias in the geometry between the surface's position, normal and curvature that will mislead the flow dynamic through the surface tension errors, as it is depicted in figure 1b.

2.4.3. Minimized variation along the surface's normal

Following the description in §2.3, as the surface is implicitly represented in a Eulerian level set framework, a common strategy to solve PDEs on the phases or surfaces is to smoothy extend the physical properties in a region around the interface. Hence the third criterion, which can be geometrically understood as orthogonal to the previous one, consists in minimizing the errors on curvature of this extension, which will be treated in details in §3.4.4. Here we can do a parallel with the mean curvature motion principle (see Evans et al. [17] for a level set framework study) or any other motion dependent on the extrapolation of the velocity in the normal direction from the surface.

Errors in the normal direction will mislead the dynamic of the flow by introducing forces in the vicinity of the surface that will add undesired perturbations, as depicted in figure 1c.

In the particular case of a level set representation of the surface, it can be easily seen that a simple computation of the curvature as $\kappa = \nabla \cdot \mathbf{n}$ will lead to a high deviation in the normal direction, as depicted in figure 2 representing the level sets of a circle. Moreover, we can see in this example that the curvature will be higher outside the surface and lower inside, compared to the curvature on the surface. This fact is true on any smooth geometry as the spatial discretization tends to zero.

In the presence of surface tension, this last phenomenon will dynamicaly tend to stretch the level sets on one side of the surface while compacting it on the other side. This can lead to more numerical errors as the flow moves the level sets, particularly when using a signed distance function that will be highly distorted, requiring frequent reinitializations of the level set function.

Given the fact that the surface tension forces are spread in the vicinity of the surface, if the variation of the curvature along the normal is null, then the surrounding level sets will move at the same speed hence reducing the numerical errors. In the case of the column equilibrium problem, this will reduce the spurious currents.

As a consequence, in order to reduce parasitic currents, one has to minimize the deviation $\kappa_{\sigma,\mathbf{n}}$ of the curvature along the normal. It can be expressed similarly to eq. 7 given any curvature κ that is extended in the whole domain:

$$\tilde{\kappa}_{\sigma,\mathbf{n}}(\mathbf{x}) = \sqrt{\int_{\mathbf{y_n}} (\tilde{\kappa}(\mathbf{y}) - \tilde{\kappa}(\mathbf{x}))^2 d\mathbf{y}}$$

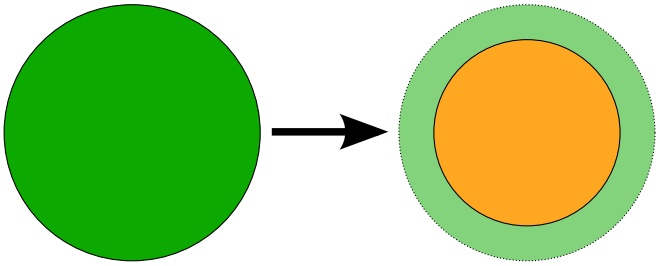
with $\mathbf{x} \in \Gamma$ and where $\mathbf{y}$ varies along the normal in a proximity ϵ of the surface, with $\mathbf{y_n}(\mathbf{x}, \xi) = \mathbf{x} + \xi\mathbf{n}$ a 1D line following the normal at $\mathbf{x}$. This measure is meaningful in the viscinity of Γ, hence on the segment $\{\mathbf{y} \,|\, \mathbf{y} \in \mathbf{y_n}(\xi),\ \xi \in [-\epsilon, \epsilon]\}$, with $\epsilon \to 0$, and for discrete points $\mathbf{x}$ close to the surface.

2.4.4. Conclusion

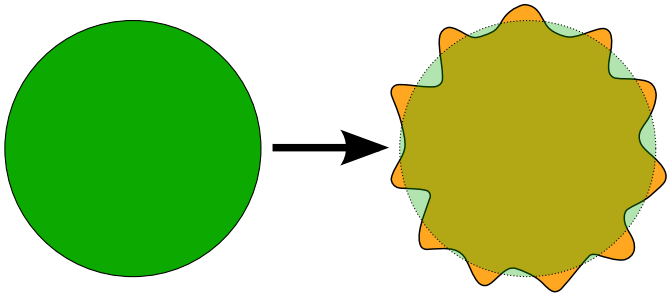
In the next section, we present a novel method which is based on:

1. the balanced force method as developped in Francois et al. [11] and detailed for a level set framework in §3.3,
2. an adaptation of the Closest Point method for curvature extension as first used in Herrmann [15], that we improved as detailed in §3.4.

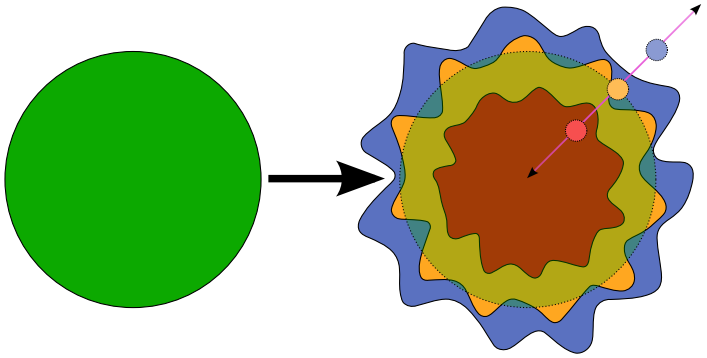
Our approach is focused on the accurate and efficient numerical computation of the curvature in the vicinity of the surface complying with a fast convergence to the exact curvature (prop. 3), a minimized variation along the surface (prop. 4) and along the normal direction (prop.5).



(a) Representation of the dynamic effect of curvature's absolute error. Here the curvature is overestimated hence the dynamic of the surface will be similar to the one of the smaller orange circle.



(b) Representation of the dynamic effect of curvature's variations along the surface. Even though the surface's position may be exact (the translucid green circle with dotted surface), the error on the curvature will mislead the dynamic as if the geometry was erroneous (the virtual orange perturbed surface).



(c) Representation of the dynamic effect of curvature's variations along the surface's normal. Even though the surface's position may be exact (the translucid green circle with dotted surface), the error on the extended curvature in the normal direction will mislead the dynamic as if the geometry was erroneous. The red, orange and blue perturbed forms correspond respectively to curvature computed on three different level set, for example $\phi = \{-h, 0, +h\}$. The normal direction is noted with a purple segment on which three circles have been placed to represent spatial points at a distance of $\{-h, 0, +h\}$ from the exact surface.

Figure 1: Visual representation of the effects of the different errors on curvature following propositions 3, 4, 5.

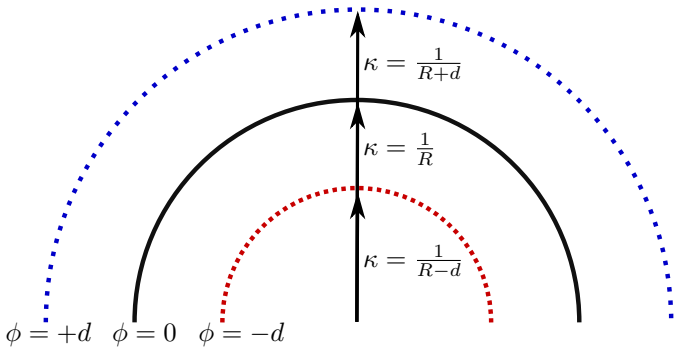


Figure 2: Local evaluation of the curvature based on a signed distance level set function. The curvature varies as $1/(R-\phi)$ on the ϕ's level set, R being the exact radius of the osculatory circle on the surface at $\phi = 0$.

3. Numerical methods

In this section we first describe the general Navier-Stokes flow solver we used, then the level set transport equation algorithm. In the third subsection we present our implementation of the balanced-force surface tension method in a level set framework. We finally present the method that we developped for the accurate computation and extension of the curvature of the surface. The associated numerical results validating the method are presented in section 4 on pure geometric and dynamic cases.

3.1. Flow solver and discretization

For our numerical simulation, we used the Thetis flow solver Poux et al. [18], Lubin and Glockner [19]. Time discretization of the momentum equation is implicit and an Euler scheme is used. The velocity/pressure coupling under the incompressible flow constraint is solved with the time splitting pressure correction method Goda [20]. The equations are discretized on a staggered grid by means of the finite volume method. The space derivatives of the inertial and stress terms are discretized by a second-order centered scheme. Since the phase function is not defined on each grid point where viscosities and densities are needed for the Navier-Stokes discretization, the physical characteristics are interpolated on the staggered grid. We use a linear interpolation to calculate the density on the velocity nodes, whereas a harmonic interpolation is used for the viscosity. Linear systems of the prediction and corrections steps are solved thanks to the direct MUMPS solver Amestoy et al. [21, 22].

3.2. Level set transport

In order to obtain an accurate computation of the curvature, as stated in section 2.4.1, the level set function is advanced in time using a 5^{th} order in space WENO method as first described in Jiang and Shu [23]. A second-order in time Runge Kutta scheme is used based on the advection of ϕ for half a time step:

$$\begin{aligned} \phi^{n+\frac{1}{2}} &= \phi^n - \Delta t\,(\mathbf{u}^n \cdot \nabla)\phi^n \\ \phi^{n+1} &= \phi^n - \Delta t\,(\mathbf{u}^n \cdot \nabla)\phi^{n+\frac{1}{2}}, \end{aligned}$$

with $\phi^n(\mathbf{x}) = \phi(\mathbf{x}, t)$ and $\phi^{n+1}(\mathbf{x}) = \phi(\mathbf{x}, t + \Delta t)$.

This method limits the spatial errors for ϕ around $O(h^5)$ which, with a precise curvature computation, theoretically permits to obtain a third order error in the curvature computation. We will see and explain in section 4 that we reach errors at even higher order in presence of small velocities.

3.3. Balanced force surface tension

In order to satisfy the consistent numerical discretization scheme criterion (cf. §2.4, claim 1), the surface tension term is discretized on the MAC grid at the velocity nodes, ie. on the faces of the control volume, with the same finite difference scheme used for the Poisson pressure equation. As in usual balanced force algorithms, we use the CSF approximation for the surface tension term rewrote relatively to the phase concentration c (we dropped the index for clarity, posing $c = c_1$):

$$\sigma\kappa\delta_{\mathbf{\Gamma}}\mathbf{n} \equiv \sigma\kappa\nabla c,$$

where ∇c is numerically computed at the faces of the cells, and points from phase 2 to phase 1.

As a consequence, recalling that the concentration c changes its value from 0 to 1 on each side of the interface, ∇c is localized on the faces of the cells surrounding Γ depending on the stencil used to compute ∇c and on the smoothness of c.

Remark. Using a sharp fonction for the concentration, ie. c equals to 0 (resp. 1), if ϕ is negative (resp. positive) will lead to an aliased function following the cells's faces. Consequently the pressure profile will straightly follow this aliasing as well as, dynamically, the surface. Moreover, as a cell's level set value can vary with transport from a certain small value $\pm\epsilon$ around zero, the surface tension force (and thus the pressure jump) will undesirably oscillate from one cell to another.

Hence it is necessary to use sufficiently smooth representations of the physical quantities in order to avoid those aliasing problems, even though it may lead to a smoother dynamic behaviour.

A common choice for calculating the concentration is to use a regularized Heaviside function as:

$$c(\mathbf{x}) = H_\epsilon\left(\phi(\mathbf{x})\right)$$

where $\epsilon = O(h)$. In order to guaranty a smooth and regular spatial convergence, we used a value of $\epsilon = 2h$ and

$$H_\epsilon(y) = \begin{cases} 0 & \text{if } y \leq -\epsilon \\ 1 & \text{if } y \geq \epsilon \\ \dfrac{1+y/\epsilon+sin\left(\frac{y\pi}{\epsilon}\right)/\pi}{2} & \text{otherwise} \end{cases} \tag{9}$$

with its associated smooth Dirac mass

$$\delta_\epsilon(y) = \begin{cases} 0 & \text{if } |y| \leq \epsilon \\ \dfrac{1+cos\left(\frac{y\pi}{\epsilon}\right)}{2\epsilon} & \text{otherwise.} \end{cases} \tag{10}$$

Moreover, so as to be sure to reduce the effect of the interface spreading or shrinking when the level set is not a distance function, we use a renormalized representation of the level set as proposed in Cottet and Maitre [24]:

$$c(\mathbf{x}) = H_\epsilon(\phi(\mathbf{x})/|\nabla\phi(\mathbf{x})|).$$

For the discretization of the ∇c, we used the same second-order scheme as for the pressure gradient: $\partial c_{i+\frac{1}{2},j}/\partial x = (c_{i+1,j} - c_{i,j})/\Delta x$. Hence, with $\epsilon = 2h$, the surface tension force is null everywhere but on the faces in the $3 \times 2 = 6$ cells band surrounding Γ.

Consequently, κ has to be computed accurately only at the center of cells around which ∇c is not null. We used interpolation to compute the curvature from the center of cells to the faces, for example here for the face between $\mathbf{x}_{i,j}$ and $\mathbf{x}_{i+1,j}$ noted $\mathbf{x}^f_{i+2,j}$:

$$\kappa^f_{i+\frac{1}{2},j} = \frac{\kappa_{i+1,j} + \kappa_{i,j}}{2}.$$

3.4. Curvature computation in a level set framework

In order to satisfy the rational computation of the surface's curvature criterion (cf. §2.4), one must use numerical methods that prove convergence in space discretization (prop. 3), minimal variation on the surface (prop. 4) and along its normal (prop. 5).

We present here a novel method that permits to attain this goal in 2 steps:

1. The accurate initial estimation of the curvature,
2. The precise extension of the curvature in the vicinity of the interface.

First we describe the Closest Point general principle which is then applied to our surface tension problem. Then we introduce error measure criteria that will help to analyze the numerical behaviour of the method. Finally we detail the novelties of our method which permits to calculate an accurate curvature extension in a level set framework.

3.4.1. The Closest Point method

The Closest Point (CP) method, first introduced in Ruuth and Merriman [25], has been successfully used in Macdonald and Ruuth [14] to solve PDEs defined on a surface by extending the physical quantities in the vicinity of the surface and then solving the equations in Eulerian domain. The CSF principle from Brackbill et al. [10] is based on a parallel point of view for integrating the singular force residing on the surface. While we only use a part of the CP method, ie. the Closest Point search, we briefly present below the overall principle for understanding the context.

Given a PDE for ψ depending on quantities defined on the surface Γ as:

$$\psi_t = F\left(t, \mathbf{x}, \psi, \nabla_S\psi, \nabla_S \cdot \left(\frac{\nabla_S\psi}{|\nabla_S\psi|}\right)\right),$$

$$\psi(0, \mathbf{x}) = \psi_0(\mathbf{x})$$

the CP method transforms the surface differential operators in a Cartesian form in the whole domain and which agrees at the interface:

$$\psi_t = F\left(t, CP(\mathbf{x}), \psi(CP(\mathbf{x})), \nabla\psi(CP(\mathbf{x})), \nabla \cdot \left(\frac{\nabla\psi\left(CP(\mathbf{x})\right)}{|\nabla\psi\left(CP(\mathbf{x})\right)|}\right)\right),$$

$$\psi(0, \mathbf{x}) = \psi_0(CP(\mathbf{x}))$$

where $CP(\mathbf{x})$ is the closest point from $\mathbf{x}$ to the surface Γ defined as:

$$CP(\mathbf{x}) = \min_{\mathbf{y}\in\Gamma}(\|\mathbf{x} - \mathbf{y}\|). \tag{11}$$

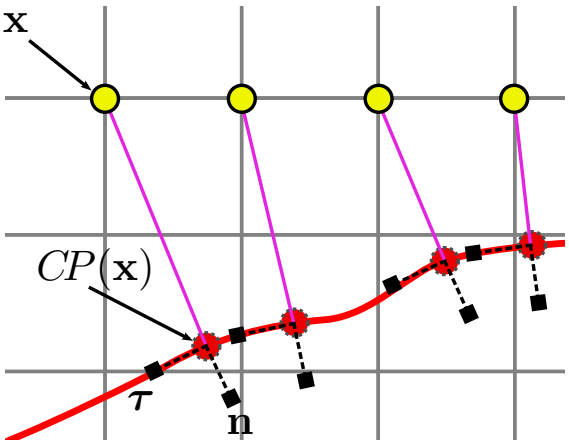


Figure 3: Closest point method: a geometrical representation. Yellow dots are grid points with their corresponding closest point on the surface as red dots. Normals and tangents on the surface are drawn as doted square ended segments. The purple lines show the colinearity between $\overrightarrow{\mathbf{x}\,CP(\mathbf{x})}$ and $\mathbf{n}\,(CP(\mathbf{x}))$.

Claim 7. By construction the vector $\overrightarrow{\mathbf{x}\,CP(\mathbf{x})}$ is orthogonal to the local tangent at $CP(\mathbf{x})$ and hence is colinear with the normal on the surface at $CP(\mathbf{x})$:

$$\overrightarrow{\mathbf{x}\,CP(\mathbf{x})} \cdot \boldsymbol{\tau}\,(CP(\mathbf{x})) = 0$$

where $\boldsymbol{\tau}$ is the unit tangent orthogonal to $\mathbf{n}$ as shown in figure 3.

The use of a CP method to extend curvature in the presence of surface tension forces has first been introduced in a level set framework by Herrmann [15]. That principle can be applied to any physical quantity defined on the surface for which an extension is needed. Our approach follows Hermmann's work by improving the CP research algorithm and enhancing the curvature's extension's smoothness.

3.4.2. Error measures

In order to analyze the accuracy of the numerical methods to compute the curvature, we define four error measurements.

Exact curvature. In order to bound the errors of the numerically computed curvature, we need to define an exact curvature as reference. The curvature is only defined on the surface, in a level set framework as we are searching for an extended representation of the curvature of a surface, we define the exact curvature at a point $\mathbf{x}$ in space as the exact curvature of closest point $CP(\mathbf{x})$:

$$\kappa_{ex}(\mathbf{x}) = \kappa_{ex}(CP(\mathbf{x})). \tag{12}$$

In the case of a circle of radius R, the exact curvature is constant in the whole domain: $\kappa_{ex}(\mathbf{x}) = 1/R$.

To understand how the closest point method permits to extend physical quantities in the whole domain, we define the line $\xi(\mathbf{y}, \lambda)$ that passes by the point $\mathbf{y}$ on the surface and follows the normal at $\mathbf{y}$:

$$\xi(\mathbf{y}, \lambda) = \mathbf{y} + \lambda \mathbf{n}(\mathbf{y}) \tag{13}$$

with $\lambda \in \mathbb{R}$. Based on claim 7, we know that $\mathbf{x}$ stands on the line $\xi(CP(\mathbf{x}), \lambda)$.

We can then define the extended curvature along the normal, ignoring the singularities where two lines for different points on the surface cross each other, by:

$$\kappa_{ex}(\xi(\mathbf{y}, \lambda)) = \kappa_{ex}(\mathbf{y}). \tag{14}$$

with $\mathbf{y} \in \Gamma$. Consequently, this defines a constant curvature in the whole domain along the lines $\xi(\mathbf{y})$ for all the points on the surface and thus equations 12 and 14 are equivalent.

In practice, we only require κ_{ex} in the vicinity of few cells (here N cells) around the interface as required by the CSF approximation (cf. §3.3) and for that purpose we introduce the function $\overline{\delta_\Gamma^N}$ defined by:

$$\overline{\delta_\Gamma^N}(\mathbf{x}_{ij}) = \begin{cases} 1 & \text{if } \overline{\delta_\Gamma^{N-1}}(\mathbf{x}_{i'j'}) = 1 \text{or is next} \\ & \quad \text{to a cell } i'j' \text{such that } \overline{\delta_\Gamma^{N-1}}(\mathbf{x}_{i'j'}) = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

This band around the interface is constructed iteratively based on the cells containing the interface $\overline{\delta_\Gamma^0}$:

$$\overline{\delta_\Gamma^0}(\mathbf{x}_{ij}) = \begin{cases} 1 & \text{if } 0 < c(\mathbf{x}_{ij}) < 1, \\ 0 & \text{otherwise.} \end{cases}$$

In the next sections, for the sake of clarity, we drop the exponent N in $\overline{\delta_\Gamma^N}$.

L_2 and L_∞ norm errors. We define the L_2 and L_∞ normalized errors for the curvature on the overall domain as:

$$L_2 = \frac{1}{\kappa_{ex}} \sqrt{\frac{\sum \overline{\delta_\Gamma}(\tilde{\kappa} - \kappa_{ex})^2}{\sum \overline{\delta_\Gamma}}}, \tag{16}$$

$$L_\infty = \frac{1}{\kappa_{ex}} \max |\overline{\delta_\Gamma}(\tilde{\kappa} - \kappa_{ex})|. \tag{17}$$

Standard deviation on the surface. The (normalized) standard deviation on the surface is only significant for a circle/column where the exact curvature is constant along the surface, hence it can only be studied in that case:

$$\tilde{\kappa}_\sigma = \frac{1}{\tilde{\kappa}_{mean}} \sqrt{\sum \overline{\delta_\Gamma}(\tilde{\kappa} - \tilde{\kappa}_{mean})^2} \tag{18}$$

where

$$\tilde{\kappa}_{mean} = \frac{\sum \overline{\delta_\Gamma} \tilde{\kappa}}{\sum \overline{\delta_\Gamma}}. \tag{19}$$

Standard deviation along the normal. The (normalized) standard deviation along the line $\xi(\mathbf{y}, \lambda)$ defined at point $\mathbf{y}$ is computed as:

$$\kappa_{\sigma,\mathbf{n}}(\mathbf{y}) = \frac{1}{\kappa(\mathbf{y})} \sqrt{\int (\kappa(\xi(\mathbf{y}, \lambda)) - \kappa(\mathbf{y}))^2 \, d\lambda} \tag{20}$$

which is discretized in the M cells neighbourhood of the cell $\mathbf{y}_{i,j}$ as:

$$\tilde{\kappa}_{\sigma,\mathbf{n}}(\mathbf{y}_{i,j}) = \frac{1}{\tilde{\kappa}(\mathbf{y}_{i,j})} \sqrt{\frac{1}{2M+1} \sum_{k=-M}^{+M} (\tilde{\kappa}(\xi(\mathbf{y}_{i,j}, kh)) - \tilde{\kappa}(\mathbf{y}_{i,j}))^2}. \tag{21}$$

In practice, as we are looking at the variation of κ in a small neighbourhood around the interface, we fix $M = 1$. It has to be noted that fourth-order Lagrange interpolation functions are used to compute $\tilde{\kappa}(\xi(\mathbf{y}_{i,j}, kh))$ because $\xi(\mathbf{y}_{i,j}, kh)$ will generally not coincide with the mesh points. The stencil used to compute $\tilde{\kappa}_{\sigma,\mathbf{n}}$ around one cell is presented in figure 4.

The overall L_2 and L_∞ errors for standard deviation along the normal are defined as:

$$L_{2,\tilde{\kappa}_{\sigma,\mathbf{n}}} = \sqrt{\frac{\sum \overline{\delta_\Gamma} \tilde{\kappa}^2_{\sigma,\mathbf{n}}}{\sum \overline{\delta_\Gamma}}}, \tag{22}$$

$$L_{\infty,\tilde{\kappa}_{\sigma,\mathbf{n}}} = \max \overline{\delta_\Gamma} \tilde{\kappa}_{\sigma,\mathbf{n}}. \tag{23}$$

3.4.3. Accurate estimation of the curvature

Because of the Closest Point method principle, we first need to get a precise initial estimation of the curvature at the surface points. Recalling eq. 6, in the Eulerian level set framework, the curvature can be computed on the overall domain by numerically computing:

$$\kappa_{LS} = \nabla \cdot \mathbf{n} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \tag{24}$$

where we have used the index LS to further differentiate this scalar field with the other methods. In order to reduce the numerical errors due to the computation of high-order derivatives, we use a developped formula for the curvature where the second derivative errors are limited to only a part of the calculus:

$$\kappa_{LS} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) = \frac{1}{|\nabla \phi|} \left(\Delta \phi - ([D^2 \phi] \frac{\nabla \phi}{|\nabla \phi|}) \cdot \frac{\nabla \phi}{|\nabla \phi|} \right) \tag{25}$$

where $D^2\phi$ is the Hessian matrix of ϕ: $(D^2\phi)_{i,j} = \frac{\partial^2 \phi}{\partial x_i \partial x_j}$.

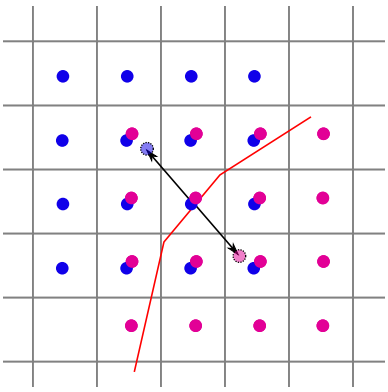


Figure 4: Stencil used for computing the standard deviation along the normal with fourth-order Lagrange interpolation functions. The surface is shown in red, blue (resp. purple) dots are used for $k = 1$ (resp. $k = -1$) in eq. 21 and are spatially shifted for comprehension.

Understanding the curvature field in a level set framework.

Fact 8. *For any function ϕ, as $h \to 0$, the curvature computed by equation 24 on the surface, ie. $\mathbf{x} \in \Gamma$, is the exact curvature of the surface at $\mathbf{x}$.*

We propose to extend this property to all the surfaces captured by ϕ, ie. all iso-values:

Claim 9. For any function ϕ, as $h \to 0$, the curvature computed by equation 24 at a point $\mathbf{x}$ is the exact curvature at $\mathbf{x}$ of the surface defined by the level set $\phi(\mathbf{x})$.

Those two properties will be used below to express a first approximation of the extension of the curvature in the whole domain. To illustrate claim 9, under the assumption $\Gamma = \{\mathbf{x}|\phi(\mathbf{x}) = \phi_0 = 0\}$, we distinguish three cases:

1. If ϕ is a signed distance function, ie. $\phi(\mathbf{x}) = sd(\mathbf{x})$, then $\kappa_{LS}(\phi, \mathbf{x})$ is the curvature of the surface represented by the level set $\Gamma_{\phi(\mathbf{x})} = \{\mathbf{y}|\phi(\mathbf{y}) = \phi(\mathbf{x})\}$. The curvature in the whole domain can thus be seen as a stretched/shrinked extension of the curvature at the surface, growing as a function of $sd(\mathbf{x})$, as depicted in figure 2.
2. If ϕ can be defined as the composition of another level set ψ with a scalar function f, ie.$\phi = f(\psi)$, then we can express the curvature by the relationship:

$$\kappa_{LS}(\phi, \cdot) = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) = \nabla \cdot \left(\frac{\nabla f(\psi)}{|\nabla f(\psi)|} \right) = sign\left(f'(\psi)\right) \left(\frac{\nabla \psi}{|\nabla \psi|} \right) = sign\left(f'(\psi)\right) \kappa_{LS}(\psi, \cdot)$$

   which gives $\kappa_{LS}(\phi, \cdot) = \kappa_{LS}(\psi, \cdot)$ when f' is strictly positive. Hence any strictly increasing function f applied to a signed distance brings us back to the first case.
3. Otherwise, outside of Γ where ϕ cannot be interpreted as a function of a signed distance function, κ_{LS} may not be directly compared to the surface's curvature. However, if ϕ is sufficiently smooth, a renormalized form (see Cottet and Maitre [24]) of the level set function $\phi/|\nabla\phi|$ gives a first order approximation of a signed distance function near the interface which brings us back to the first case.

3.4.4. Extension of the surface's curvature

Here we present two different methods to extend the surface's curvature and discuss their advantages and drawbacks. The first one is based on claim 9 and is presented to give a better geometrical understanding of the curvature in a level set framework. The second one is the method that we propose based on the Closest Point method presented in §3.4.1 and following Hermmann's works Herrmann [15].

Osculatory circle approximation. This approximation is based on the fact that, under the assumption that locally, as the discretization tends to zero, the curvature of a surface can be defined as the inverse of the radius of the osculatory circle tangent to Γ at $\mathbf{y}$:

$$\kappa(\mathbf{y}) = 1/R(\mathbf{y}).$$

This assertion is also true in a level set framework as shown in figure 2. Let $d(\mathbf{x})$ be the signed physical distance from the point $\mathbf{x}$ to the surface at level set $\phi_0 = 0$. As $\mathbf{x}$ stands on the line $\mathbf{x} = \xi(\mathbf{y}, \lambda)$ (where ξ is defined by eq. 13), or conversely $\mathbf{y} = CP(\mathbf{x}) \in \Gamma$, we can write $d(\mathbf{x}) = sign(\lambda).\|\mathbf{x} - \mathbf{y}\|$. Then the radius of the osculatory circle at $\mathbf{x}$ is $R(\mathbf{x}) = R(\mathbf{y}) + d(\mathbf{x}) = \kappa^{-1}(\mathbf{y}) + d(\mathbf{x}) = \kappa^{-1}(\mathbf{x})$.

Algorithm 1 $CP_{\odot}$: first order Newton method for the closest point search in a level set framework.

1. Initialize $\hat{\mathbf{y}}^0 = \mathbf{x}$, $i = 0$
2. While $|\phi(\hat{\mathbf{y}}^i)| > \epsilon$ do
   (a) $\hat{\mathbf{y}}^{i+1} = \hat{\mathbf{y}}^i - \Delta_\lambda d(\hat{\mathbf{y}}^i)\mathbf{n}(\hat{\mathbf{y}}^i)$
   (b) $i \leftarrow i + 1$
3. $\widetilde{CP}_{\odot}(\mathbf{x}) = \hat{\mathbf{y}}^i$

Consequently we can defined an extended surface's curvature as:

$$\kappa_{osc}(\mathbf{x}) = \frac{1}{R(\mathbf{y})} = \frac{1}{\kappa^{-1}(\mathbf{x}) - d(\mathbf{x})}$$

which does not require any knowledge of $\mathbf{y}$. In a general level set framework the equation is:

$$\kappa_{osc}(\mathbf{x}) = \frac{1}{\kappa_{LS}^{-1}(\mathbf{x}) - d_\phi(\mathbf{x})} \tag{26}$$

with d_ϕ being the physical signed distance function computed from ϕ which, near the surface, as the discretization step tends to zero, can be approximated by $d_\phi(\mathbf{x}) = \frac{\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}$. In the particular case of a signed distance function $d_\phi(\mathbf{x}) = \phi(\mathbf{x})$.

Limits The limits of this approximation stands for the cases when:

- the local curvature is null, $\kappa(\mathbf{x}) = 0$, then the osculatory circle has an infinite radius which can raise numerical problems. A threshold is thus used and we set $\kappa_{osc}(\mathbf{x}) = 0$ in those cases,
- the radius $R(\mathbf{x}) = R(\mathbf{y}) + d(\mathbf{x}) = 0$, ie. $\mathbf{x}$ is the singular center of the osculatory circle, where we do not extend the curvature. As we only look in the vicinity of the surface, this case will only arise for very high curvatures (resp. small radii), $\kappa = O(1/h)$ for which we propose a threshold for the maximum curvature value $\kappa_{max} = 1/h$, which is coherent with the physical maximum curvature that can be captured at the scale h.

This osculatory circle extension is of low order as it does not take into account the variation of a curvature which is essential in the general case. We present in 4.2 spatial convergence results that show this approximation to be fourth-order in the circle case but only first order in the ellipse case.

Closest point on surface interpolation. As introduced in §3.4.1, another approach to extend the curvature away from the surface consists in finding the closest point to $\mathbf{x}$ on Γ and interpolating the curvature there. This approach has been succesfully implemented in Herrmann [15] in a level set framework. We present in this paragraph a detailed version of the method that will serve as a base for the novel method proposed thereafter.

Without an analytical representation of the surface, finding $\mathbf{y} = CP(\mathbf{x})$ is not as straightforward as finding on which line $\mathbf{x} = \xi(\mathbf{y}, \lambda)$ stands. However, we can get an approximation of the closest point to $\mathbf{x}$ by backtracing the trajectory $\mathbf{y} = \xi^{-1}(\mathbf{x}, \lambda)$ starting at $\mathbf{x}$ and leading to $\mathbf{y}$.

In a level set framework, $\xi^{-1}(\mathbf{x}, \lambda)$ follows the gradient field of ϕ and, in general, is a curve and not a line. Thus, one needs to descend along that curve to find $\hat{\mathbf{y}} = \widetilde{CP}(\mathbf{x})$, a numerical approximation of the exact solution $\mathbf{y} = CP(\mathbf{x})$.

We introduce algorithm 1, noted $CP_{\odot}$, to compute an approximate closest point. $CP_{\odot}$is based on a first-order Newton method with a convergence criterion based on a threshold distance $\epsilon \ll h$ and a with virtual time step Δ_λ. We have used here the local approximation of the distance from the point $\hat{\mathbf{y}}^n$ to the surface: $d_\phi(\hat{\mathbf{y}}^n) = \frac{\phi(\hat{\mathbf{y}}^n)}{|\nabla\phi(\hat{\mathbf{y}}^n)|}$ and of the normal: $\mathbf{n}(\hat{\mathbf{y}}^n) = \frac{\nabla\phi(\hat{\mathbf{y}}^n)}{|\nabla\phi(\hat{\mathbf{y}}^n)|}$. Interpolation is used in order to compute those values from the discrete grid points. Figure 5 shows a schematical view of the algorithm.

In practice, the threshold distance ϵ is set to h^4 as we expect a fourth-order convergence on the CP algorithms and as it will be showed below in the numerical results §4. The virtual time step is set to $\Delta_\lambda = 0.9$ in order to stay as much as possible on the same "side" of the level set, ie. not crossing the zero level set in presence of numerical errors. Setting a lower value for Δ_λ will help to converge more precisely for points far away from the surface, however as we are mostly interested by points in the proximity of Γ, we have found this value to be sufficient.

Algorithm 2 $CP_\perp$: first order Newton method for the closest point search in a level set framework with colinearity property.

1. Initialize $i = 0$,
2. Find a first approximation of the closest point $\hat{\mathbf{y}}^0 = \widetilde{CP_\odot}(\mathbf{x})$
3. While $|\omega(\mathbf{x}, \hat{\mathbf{y}}^i)| > \epsilon$ do
   (a) $\hat{\mathbf{y}}^{i+1} = \widetilde{CP_\odot}(\hat{\mathbf{y}}^i + \omega(\mathbf{x}, \hat{\mathbf{y}}^i)\,\boldsymbol{\tau}(\hat{\mathbf{y}}^i))$
   (b) $i \leftarrow i + 1$
4. $\widetilde{CP}_\perp(\mathbf{x}) = \hat{\mathbf{y}}^i$

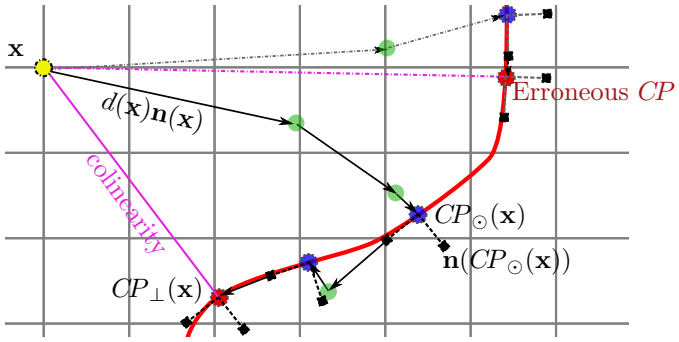


Figure 5: Schematical view of the Closest Point algorithms. Errors on gradient vectors has been exagerated and redundant notations has been omitted for the sake of comprehension. Green dots stand for intermediary points, blue dots for $CP_\odot$ results and the red dot for the final $CP_\perp$ result. Normals on the surface are drawn as doted square ended segments. The top gray dashed paths show an erroneous closest point due to bad approximations of the direction $\nabla\phi$ toward surface.

Once the closest point $\widetilde{CP}(\mathbf{x})$ has been computed we interpolate the curvature at that position leading to the following formula for the extended curvature at the grid point $\mathbf{x}_{i,j}$:

$$\tilde{\kappa}_{CP_\cdot}(\mathbf{x}_{i,j}) = Interpolate(\widetilde{\kappa_{LS}}, \widetilde{CP_\cdot}(\mathbf{x}_{i,j})). \tag{27}$$

where we have used the symbol $\cdot$ replacing $\odot$ because any closest point algorithm can be used.

This method permits to drasticaly reduce the error of curvature along the normal, depending on the κ_{LS} approximation and the interpolation errors, as it is shown in section 4.2.

Colinearity criteria improvement Recalling claim 7, the closest point $CP(\mathbf{x})$ to $\mathbf{x}$ on the surface Γ is colinear to the normal $\mathbf{n}_\Gamma(CP(\mathbf{x}))$ of the surface Γ, or equivalently orthogonal to the tangent:

$$\overrightarrow{\mathbf{x}\,CP(\mathbf{x})} \cdot \boldsymbol{\tau}_\Gamma(CP(\mathbf{x})) = 0$$

Here we have used the notation $\mathbf{n}_\Gamma$ and $\boldsymbol{\tau}_\Gamma$ to distinguish between the normal and tangent defined on the surface Γ and in the whole Eulerian domain. In a level set framework, we can compute the cosine angle between a vector $\overrightarrow{\mathbf{xy}}$ and the tangent at $\mathbf{y}$ with:

$$\omega(\mathbf{x}, \mathbf{y}) = \frac{\overrightarrow{\mathbf{xy}}}{|\overrightarrow{\mathbf{xy}}|} \cdot \boldsymbol{\tau}_\Gamma(\mathbf{y}).$$

The algorithm 1 does not guaranty the colinearity property. We propose algorithm 2, noted $CP_\perp$, that permits to enforce it while still remaining on the surface. This algorithm can be understood as finding a candidate closest point to $\mathbf{x}$ on the interface and then correcting it toward the direction of colinearity. As this new corrected candidate may not be on the interface, we need to project it on the interface. We apply this procedure iteratively until both the distance to the interface and the angle are smaller than a given threshold. A schematical view is proposed in figure 5.

This algorithm works under the assumption that ϕ is a smooth enough function where the initial closest point computed $CP_\odot$ will not bring us next to a local minima ensuring colinearity while not being the closest point on the surface, as shown by the dashed paths in fig. 5. This will be particularly the case when ϕ is very far from a distance function, $\nabla\phi$ points too far away toward the exact closest point and/or for points far away from the surface.

In comparison to the previous algorithm, this one will compute at least as much interpolations. Ensuring the colinearity property greatly improves the accuracy of the Closest Point search and the curvature interpolation. We refer the reader to section 4.2 for numerical results and efficiency comparison.

Reinterpolating The extended field obtained by interpolation at the Closest Point can suffer from important variations of the interpolted field in the few cells around the surface. In order to reduce this perturbation and homogenize even more the extended field, we operate a second interpolation process on the first field's extension, reading:

$$\tilde{\kappa}_{CP^2_\perp}(\mathbf{x}_{i,j}) = Interpolate(\widetilde{\kappa_{CP_\perp}}, \widetilde{CP_\perp}(\mathbf{x}_{i,j})),$$

$$\tilde{\kappa}_{CP^2_\perp}(\mathbf{x}_{i,j}) = Interpolate(Interpolate(\widetilde{\kappa_{LS}}, \widetilde{CP_\perp}(\mathbf{x}_{i,j})), \widetilde{CP_\perp}(\mathbf{x}_{i,j})), \tag{28}$$

which is the effective algorithm that we used in our numerical simulations.

This process only costs one more interpolation per cell as we have already computed the closest point $\widetilde{CP_\perp}(\mathbf{x}_{i,j})$.

We will show in sections 4.2 and 4.3 the numerical results obtained with this method for geometrical cases and in the dynamic simulations.

Remarks and discussion

- In the remainder of the document, if not stated explicitely when noting CP, we assume the reinterpolation algorithm dropping the exponent 2 and the index $\perp$ for simplicity.
- The errors appearing in the numerical closest point computation compared to the exact closest point are due to:
  1. The Newton's descent first order discretization. It has to be noted that this first order descent is optimal in the case of a signed distance function where $|\nabla\phi| = 1$ in the whole domain.
  2. The interpolation functions used.
- In the case where ϕ is a signed distance function, the $CP_\odot$ and $CP_\perp$ will theoreticaly lead to the exact same point on the surface as $\nabla\phi$ points toward the exact closest point. As in many level set applications, reinitialization techniques are used to obtain a signed distance function, $CP_\perp$ will theoreticaly not give better results. However, due to the discretization of the Newton's descent and interpolation errors, the Closest Point computed will not be the same and $CP_\perp$ will guaranty colinearity. Moreover, one has to keep in mind that the use of a reinitialization technique will introduce $O(h^N)$ on the level set's position leading to $O(h^{N-2})$ errors on the curvature ; eg. if a fourth-order method is used to reinitialize ϕ then the curvature error will never be better than second-order.
- The calculation of the Closest Point in the whole domain - or a band around the interface - gives a straightforward result for a reinitialization of the level set. The order of precision of the so reconstructed signed distance function depends on the precision of the CP algorithm used. To attain 4^{th} order precision on the curvature one would need a precision of at least order 6 in the signed distance function. However, as reinitialization is not the topic of the current article, we will not further detail the subject.

4. Numerical results

4.1. Foreword

In all our test cases, the order of convergence for the L_2 and L_∞ norms are the same thus we will only show and discuss the later norm which is the more restrictive. For this study, we used smooth representation of the surface Γ and the phases characteristic function via δ_ϵ and H_ϵ from equations 9 and 10 with $\epsilon = 2h$ where h is the spatial discretization step. This regularization contains approximately 81% of the Dirac mass in the 3 cells around the interface, ie. for $\phi \in [-h, +h]$. Thus, all the error measures are numerically computed around the interface in a 1 cell neighbourhood, fixing $N = 1$ in eq. 15. The thresholds used for the CP algorithms are set to h^4 for the distance and the angle criterion in order to attain fourth-order errors, as shown in the results below.

4.2. Geometrical convergence analysis of the method

In the following paragraphs, we study the Eulerian discretization errors for a fixed geometry by varying the methods used: the level set curvature estimation, the osculatory circle approximation and the Closest Point extension. The numerical convergence analysis for the CP methods are detailed for the ellipse case in §4.2.2 which is the most discriminating.

Method	L_∞	$\frac{\lvert\kappa_{mean}-\kappa_{ex}\rvert}{\kappa_{ex}}$	Std. dev. $\tilde{\kappa}_\sigma$	Normal $L_{\infty,\tilde{\kappa}_{\sigma,\mathbf{n}}}$
κ_{LS} 4^{th} order	1	1	1	1
κ_{osc} (w/ κ_{LS} 4^{th} order)	4	4	4	5
κ_{CP} (w/ κ_{LS} 2^{nd} order)	2	2	2	4
κ_{CP} (w/ κ_{LS} 4^{th} order)	4	4	4	4

Table 2: Circle geometrical test case: approximative order of convergence of error measures for the different methods.

4.2.1. Circle case

A circle of radius $R = 0.4$ is centered at the origin of a unit square. The level set is initialized as the exact signed distance function:

$$\phi(\mathbf{x}) = |\mathbf{x}| - R.$$

Here the relevant parameter is the ratio R/h representing the number of grid cells for a radius.

In this particular case the $CP_\odot$ and $CP_\perp$ methods give the same results hence we will not distinguish them. This is due to the fact that for any point in the domain, by construction, the level set's gradient is always colinear to the normal at its closest point on the surface.

Order 2 and order 4 schemes. First, we show the importance of a high-order numerical computation of κ_{LS} from eq. 25. For that purpose, we have used classical central finite difference schemes of order 2 and 4 for gradients and Laplacian operators. The figure 6 depicts the convergence of the different error measures relatively to spatial discretization, taking $\kappa_{ex} = 1/R = 5$.

The results show a roughly first order convergence for the errors measurements for κ_{LS} ; in that case, the second and fourth-order schemes give qualitativaly the same results. This is expected as the errors in the curvature come from the stretch/shrink effect as $\kappa_{osc} = \frac{1}{\kappa_{ex}^{-1}-d}$ away from the level set at $\phi = 0$, which dominates the error measures.

The CP method acts as expected leading to the same order of convergence than the scheme used for the κ_{LS}. The normal deviation leads to a fourth-order error convergence. This error is due to the interpolation functions used to compute the normal error deviation as it is discussed in the next paragraphs. Concerning the CP reinterpolation method from eq. 28, the error measures are only different from the standard $CP_\perp$ method for the normal deviation: in that case we clearly see that reinterpolation the curvature from a first curvature extension permits to reduce the normal error by a factor of 3. As we will see in the ellipse test case below, this gain only holds for smooth enough curvatures without deteriorating the curvature field in the general case, however it permits to reduce the spurious currents studied in §4.3.

The osculatory approximation based on a fourth-order scheme for κ_{LS} gives very good results: it converges at the same rate as the CP method for all the error measures but for the normal deviation for which it converges more rapidly (around order 5). However, its absolute value is approximatively 100 times higher for $R/h \approx 12$. Despite this good convergence rate, one has to keep in mind that the good properties of this method relies on the first order approximation of the local curvature to the osculatory circle which will be deteriorated in the general case (ie. contrary to this ideal circle case) as it will be shown in the next sections for the ellipse.

The aliasing that can be observed in the mean curvature (figure 6a) is due to the fact that we are using a sharp stencil $\overline{\delta_\Gamma}$ for the error measurements around the interface's position, affecting any measurement contrary to a smooth δ_Γ.

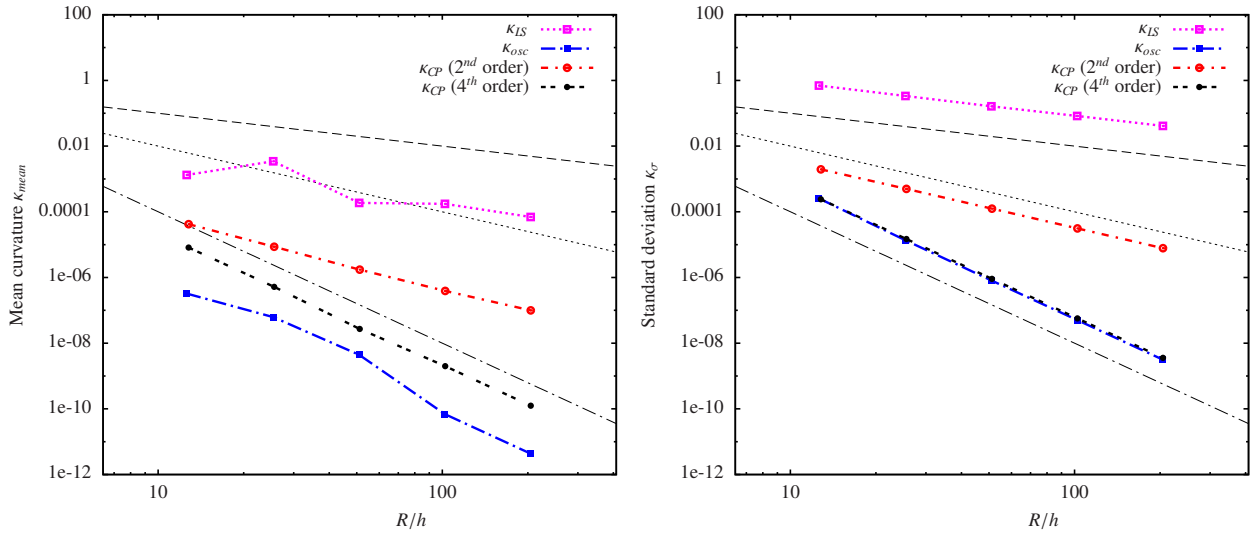
Even though using order 4 scheme for κ_{LS} gives much better results, it is important to note that it has to be avoided for fast varying ϕ, ie. when the local curvature is of the order of h, where it would raise higher errors than a more compact scheme because of high frequencies amplified errors. We chose a sufficiently large R/h ratio to avoid such perturbations. However, in the general case, one could think of mixing different schemes based on a first estimation of the curvature.

Errors in the curvature estimation in the presence of fictitious perturbations. In order to see how the method responds in presence of numerical errors (that will arise with the transport of the level set), we introduced small perturbations to the initial level set as:

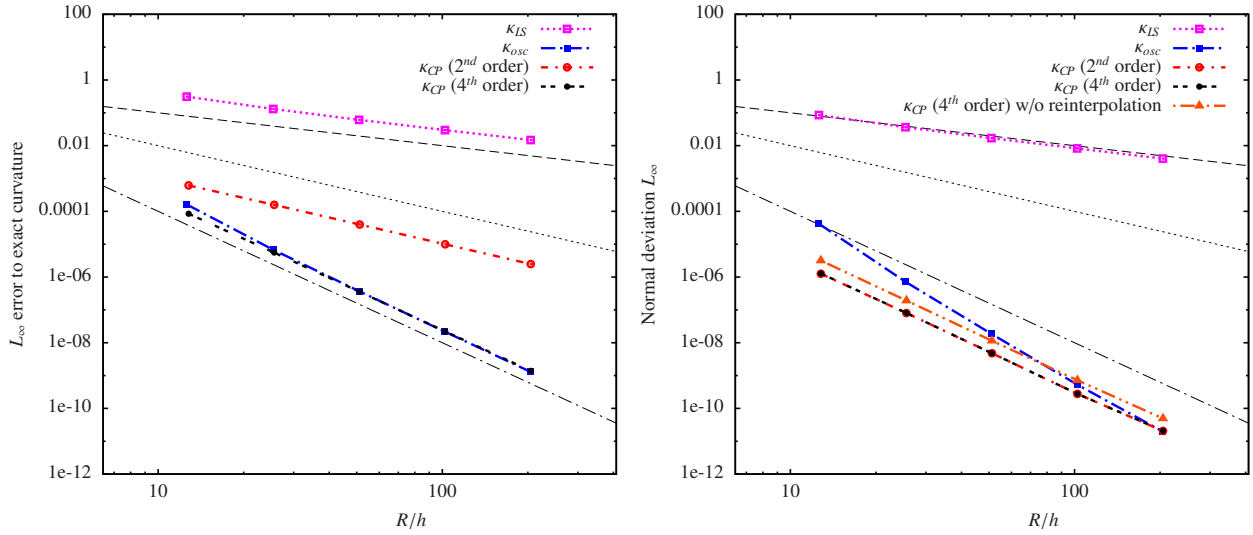
$$\hat{\phi}(\mathbf{x}) = \phi(\mathbf{x}) + e(h^M) \tag{29}$$

where $\phi(\mathbf{x})$ is the signed distance function to the circle and $e(h^M)$ is a random function that takes its values in the interval $[-h^M, +h^M]$. One has to note that if $e(h^M) = h^M$ is constant then $\hat{\phi}(\mathbf{x}) = \phi(\mathbf{x}) + h^M$ and the relative error for the level set function is

$$\frac{\hat{\phi}(\mathbf{x}) - \phi(\mathbf{x})}{\phi(\mathbf{x})} = \frac{h^M}{\phi(\mathbf{x})} = O(h^{M-1})$$

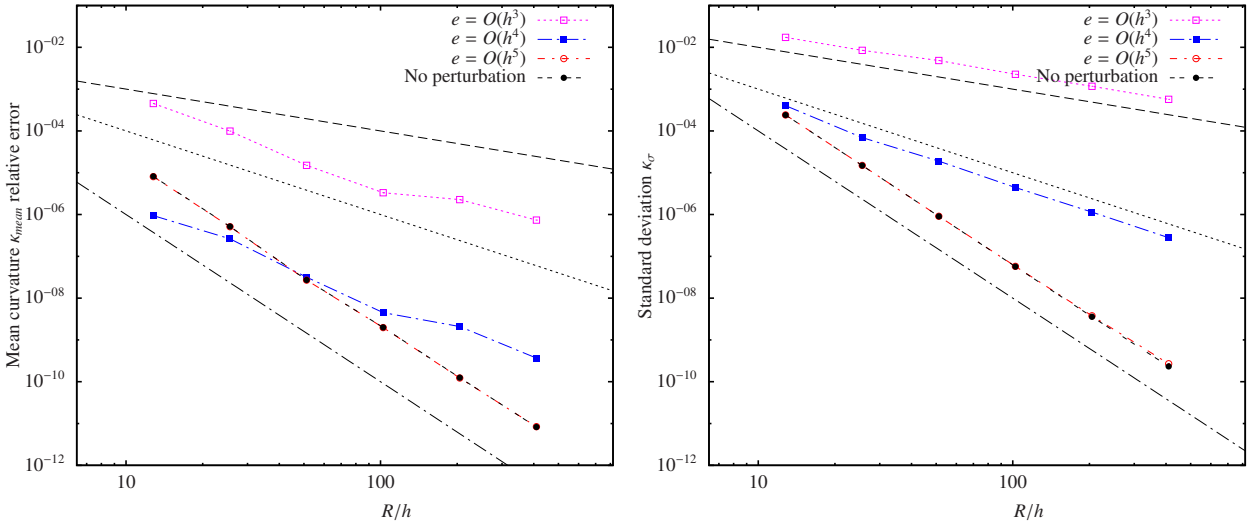


(a) Mean curvature error and standard deviation.

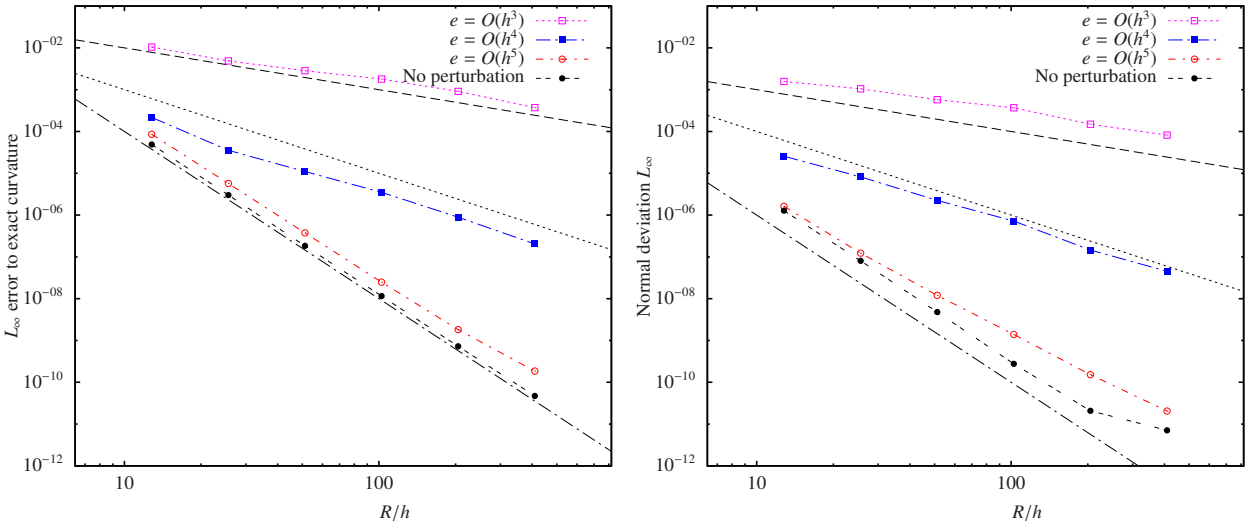


(b) L_∞ error to exact curvature and L_∞ normal deviation.

Figure 6: Circle geometrical test case: spatial convergence as a function of R/h for the 4^{th} order κ_{LS}, the osculatory circle correction and the CP method with order 2 and order 4 κ_{LS}. κ_{LS} errors are only shown for $O(h^4)$ as the results with the $O(h^2)$ scheme are almost identical.



(a) Mean curvature relative error and standard deviation.



(b) L_∞ error to exact curvature and L_∞ normal deviation. The last normal deviation measure for the calculus without perturbation is decreasing less rapidly because of real number double precision limits at such small scales ($h = 1/2048$ and errors of the order 10^{-11}).

Figure 7: Circle geometrical test case: convergence of the different error measures as a function of R/h in presence of small perturbations for the fourth-order *CP* method.

near the interface. Thus if we add a perturbation of the order 3 then the expected perturbation on the curvature computation should be of the order $3 - 1 - 2 = 0$. However in the following results, as $e(h^M) \in [-h^M, +h^M]$ is a random function, we do expect a smaller deterioration on the order of convergence.

Here we used a fourth-order scheme for κ_{LS}. Figure 7 shows that the introduction of a $O(h^3)$ (resp. $O(h^4)$) perturbation approximately reduces the order of convergence from 4 to 1 (resp. from 4 to 2), as expected, as discussed in §2.4.1.

4.2.2. Ellipse case

In order to test the robustness of the method in a more general case, we applied the same spatial convergence study for an ellipse. Particularly, we will show the effect of the colinear closest point method compared to the standard method. The surface is implicitely defined by the level set:

$$\phi(\mathbf{x}) = \sqrt{\left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2} - R$$

with $a = 1.2$, $b = 0.8$ and $R = 0.2$, or in a parametric form:

$$\Gamma(\theta) = (aR\,cos(\theta), bR\,sin(\theta))\,.$$

One has to note that the level set function ϕ is not a signed distance function which one is not trivial to compute in that case. The exact curvature extension that was constant for the circle case is here less trivial. Recall the curvature extrapolation as:

$$\kappa_{ex}(\mathbf{x}) = \kappa_{ex}\left(CP_{ex}(\mathbf{x})\right) = \kappa_{\Gamma,ex}\left(\theta_{CP_{ex}}(\mathbf{x})\right),$$

where $\theta_{CP}(\mathbf{x})$ is the angular parameter for the closest point to $\mathbf{x}$ on the surface. We thus need to compute the exact closest point $CP_{ex}(\mathbf{x})$ on the ellipse which is obtained by solving the equation

$$\overrightarrow{\mathbf{x}\,CP_{ex}(\mathbf{x})} \cdot \Gamma(\theta) = 0$$

for θ, as the line from $\mathbf{x}$ to its closest point on Γ is colinear to the normal at that point. This non linear equation has in the general case 2 solutions but only one in the nearest quadrant (known by the position of $\mathbf{x}$ compared to the ellipse's parameters). To solve it we used the Newton's method with the search reduced to the nearest quadrant. The curvature at that point $\Gamma(\theta)$ is then computed with:

$$\kappa_{\Gamma,ex}(\theta) = \frac{ab}{\left(b^2 cos^2(\theta) + a^2 sin^2(\theta)\right)^{3/2}}.$$

One has to notice that this "exact" extended curvature is subjected to discretization errors due to this Newton's descent. In consequence the errors that we present below can be biased by this additional perturbation. However we expect this last to be negligible.

Also, as stated in §2.4.2 there is no meaning in comparing the mean curvature and the standard deviation as the curvature varies along the surface, hence we removed those two error measures for this ellipse test case.

Closest point spatial convergence. In this paragraph, we study the spatial convergence of the two different CP algorithms. For this purpose, we look at the error in the distance map defined as the distance from $\mathbf{x}$ to its computed closest point on the surface $CP(\mathbf{x})$, and the distance between $CP(\mathbf{x})$ and the exact closest point $CP_{ex}(\mathbf{x})$. Those two errors have different meaning: while the error in the distance to the surface is meaningful, as we aim to interpolate values on the interface, the error in the closest point search is in our case more relevant.

The L_2 and L_∞ closest point error measures are defined as

$$L_{2,CP.} = \sqrt{\int \left|CP.(\mathbf{x}) - CP_{ex}(\mathbf{x})\right|^2}$$

and

$$L_{\infty,CP.} = \max \left|CP.(\mathbf{x}) - CP_{ex}(\mathbf{x})\right|$$

where the dot index is either $\odot$ or $\perp$ standing for the two different CP algorithms. We also define the distance map error measures as

$$L_{2,d.} = \sqrt{\int \left(d(\mathbf{x}, CP.(\mathbf{x})) - d(\mathbf{x}, CP_{ex}(\mathbf{x}))\right)^2}$$

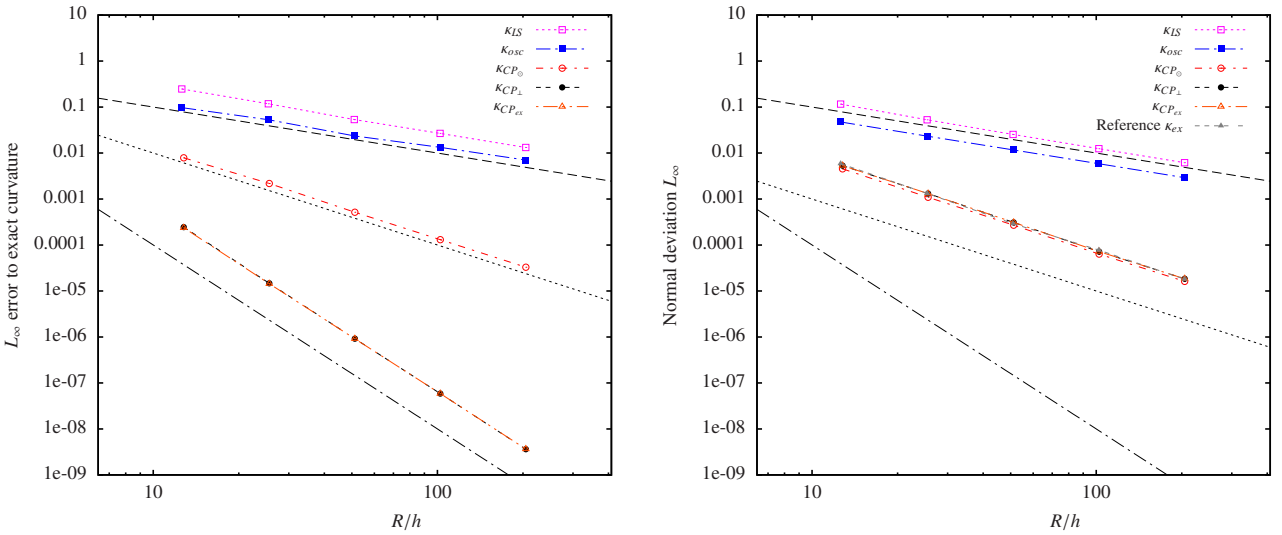
and

$$L_{\infty,d.} = \max \left|d(\mathbf{x}, CP.(\mathbf{x})) - d(\mathbf{x}, CP_{ex}(\mathbf{x}))\right|$$

where $d(\mathbf{x}, \mathbf{y})$ is here the Euclidian distance from $\mathbf{x}$ to $\mathbf{y}$. In a similar fashion as in §3.4.2 those error measures are numerically evaluated in a band around the surface.

Curvature spatial convergence. Figure 8 shows the spatial convergence for the computed curvature as a fonction of R/h for the L_∞ error measures (L_2 errors are qualitatively equivalent) and table 3 presents a synthesis. As expected the osculatory circle method is decreased to first order while the CP method without colinearity converges at second-order and the use of $CP_\perp$ converges at fourth-order showing the clear advantage of using an accurate Closest Point search. Moreover, this last method approximately equals the reference results when using CP_{ex}. In this ellipse case, the error measures obtained with and without reinterpolation of the level set (eq. 28) are approximately equal and hence not shown. We can conclude that this process does not deteriorate the solution in the general case, while it has a clear benefit for smooth surfaces localy homogeneous to a circle.

In this test case the L_∞ normal error are decreased to second-order for both CP methods. This loss of precision compared to the circle case is due to the variation of curvature perceived by the interpolations from equation 23. This can be observed in figure 8b where we have also plotted the normal deviation error based on the interpolation of the exact curvature computation $\kappa_{ex}(\mathbf{x})$ at $CP_{ex}(\mathbf{x})$, which is the numerical lower bound



(a) L_∞ errors to exact curvature. The last three curves stand for the three closest point estimates: $CP_\odot$, $CP_\perp$ and CP_{ex}. The use of $CP_\perp$ permits to gain 2 orders of convergence and give similar results as CP_{ex} for reference.

(b) Normal deviation with the lower bound reference curve using κ_{ex} which equals all the CP methods.

Figure 8: Ellipse geometrical test case: spatial convergence as a function of R/h for κ_{LS} with an order 4 scheme, the osculatory circle correction and the $CP_\odot$ and $CP_\perp$ methods.

Method	L_∞	Normal $L_{\infty,\tilde{\kappa}_{\sigma,\mathbf{n}}}$
κ_{LS}	1	1
κ_{osc}	1	1
$CP_\odot$	2	2
$CP_\perp$	4	2
CP_{ex}	4	2

Table 3: Ellipse geometrical test case: approximative order of convergence of error measures for the different methods, a 4^{th} order κ_{LS} computation is used.

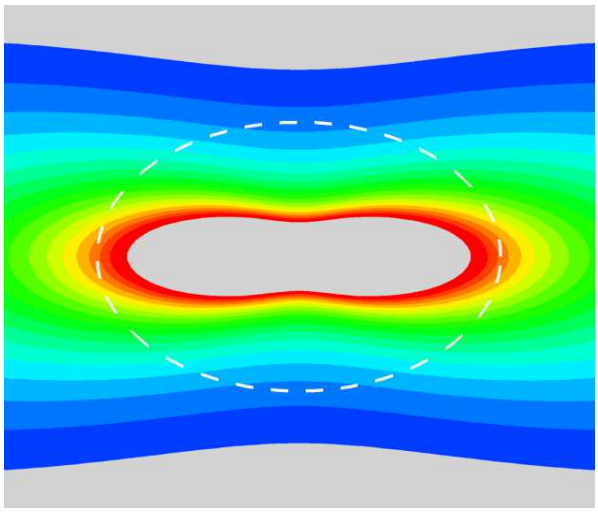
that can be attained by any method. All the results using any of the CP method approximately equals this reference curve for $L_{\infty,\tilde{\kappa}_{\sigma,\mathbf{n}}}$.

Figure 9 shows a spatial representation of the curvature in the domain for the different methods for $R/h = 25.4$. We can clearly see the effect of the CP algorithm on the extension of the curvature along the normal. Looking closely, the simple $CP_\odot$ method induce non rectilinear iso-values for the curvature while the $CP_\perp$ method correct this problem. The exact curvature profile obtained with CP_{ex} is not shown as it is undistinguishable from the $CP_\perp$ method.

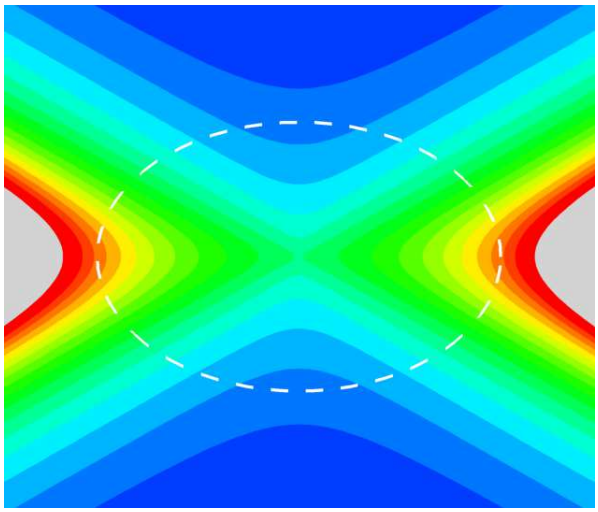
Numerical convergence in presence of perturbations. Figure 10 shows the spatial convergence of the closest point error and the distance map as a function of R/h with varying perturbations applied to the level set as presented in eq. 29. Summarized in table 4, we observe a maximum order 2 for the closest point error measure in the case of the $CP_\odot$ algorithm while it increases to fourth-order for the $CP_\perp$method. The $CP_\odot$ method's distance map has a maximum third-order convergence, while the $CP_\perp$one is up to fourth-order. We believe that the convergence rate is limited because of the errors on the computation of $\nabla\phi$, on the numerical thresholds used but more importantly because of the fourth-order interpolation functions used. However this fourth-order convergence rate is sufficient in our case and for the aimed applications.

The $CP_\perp$ algorithm we propose in this paper shows a clear benefit compared to the $CP_\odot$ used in the litterature that will have an important impact on the order of convergence of the κ_{CP} methods studied in the next paragraph.

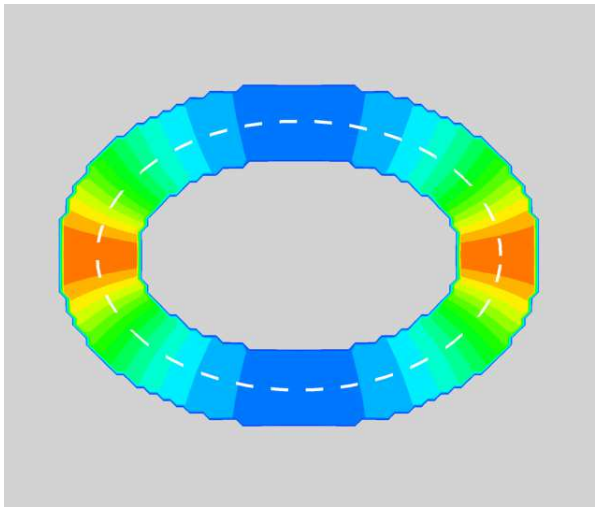
Computational cost of the closest point algorithms. Figure 11 shows the mean number of iterations per grid points for the $CP_\odot$ and $CP_\perp$ methods. For coarse grids with more errors on the computation of the normal and interpolations, the $CP_\perp$ algorithm requires much more iterations than the simple $CP_\odot$ algorithm (around three times for $R/h = 12.8$). This ratio reduces to around 1.6 for a finer mesh. This behaviour is expected as



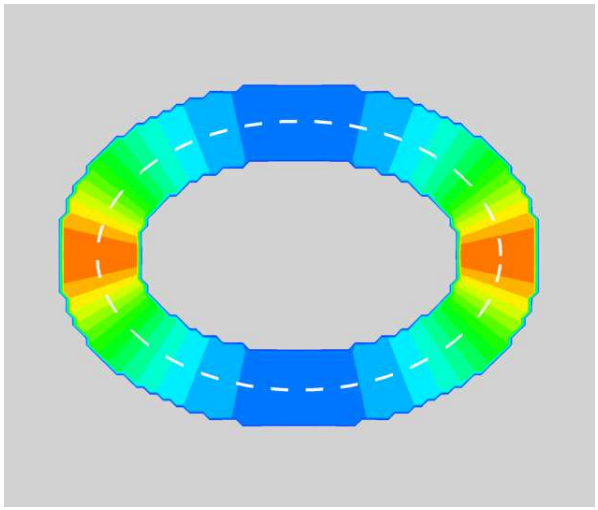
(a) κ_{LS}.



(b) κ_{osc}.



(c) $CP_{\odot}$ method.



(d) $CP_{\perp}$ method.

Figure 9: Ellipse geometrical test case: curvature profiles for the different methods. The position of the ellipse is drawn with white dashed lines. The color variation goes from blue ($\kappa = 2.5$) to red ($\kappa = 10$), gray zones represent curvature values outside that interval or outside the computation band for the CP methods. The $CP_{\perp}$ method and the exact curvature profiles are undistinguishable hence this last one is not shown.

Method	Perturb.	$L_{\infty,CP}$	$L_{\infty,d}$
$CP_{\odot}$	$O(h^1)$	0	0
$CP_{\odot}$	$O(h^2)$	2	2
$CP_{\odot}$	$O(h^3)$	2	3
$CP_{\odot}$	$O(h^4)$	2	3

Method	Perturb.	$L_{\infty,CP}$	$L_{\infty,d}$
$CP_{\perp}$	$O(h^1)$	0	0
$CP_{\perp}$	$O(h^2)$	2	2
$CP_{\perp}$	$O(h^3)$	3	3
$CP_{\perp}$	$O(h^4)$	4	4

Table 4: Ellipse geometrical case: approximative orders of convergence of error measures for the two different CP methods and ϕ perturbations.

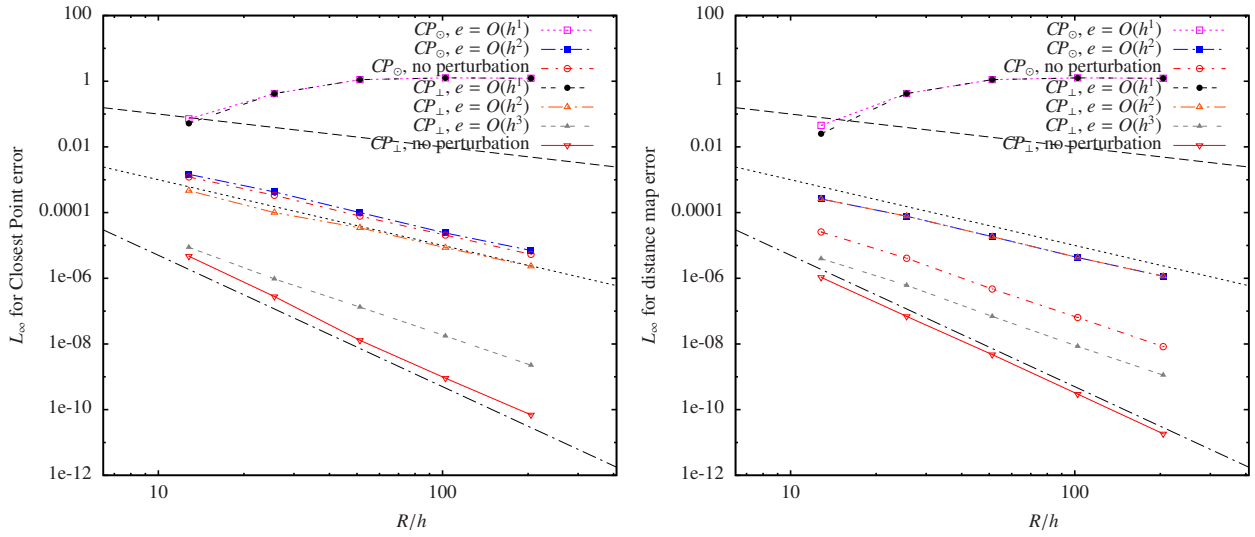


Figure 10: Ellipse geometrical test case: convergence of the $L_{\infty,CP.}$ closest point error (left) and $L_{\infty,d.}$ distance map error (right) as a function of R/h in presence of small perturbations for $CP_\odot$ and the $CP_\perp$ methods. Results for $CP_\odot$ with $O(h^3)$ and $O(h^4)$ errors are almost equal to the ones with no perturbation thus not shown. Similarly, results for $CP_\perp$ with $O(h^4)$ perturbations are not shown.

the higher the curvature to spatial step ratio, the farther away the first $CP_\odot$ evaluation will brings us from the exact closest point.

As the number of grid points linearly grows with the spatial discretization, the use of the $CP_\perp$ is highly recommanded while having a low impact on the computational time.

4.2.3. Conclusion of the geometrical study

Through this geometrical study on the circle and ellipse cases, we have shown that the CP method attain high (up to 4) order spatial convergence on the different error measures permitting to guaranty numerical accuracy for the 3 propositions of §2.4 and thus accuracy in the surface tension driven simulations. The use of a fourth-order κ_{LS} computation is also a key aspect in order to attain this precision and will be demonstrated to be very important in the dynamic numerical simulations in the next sections.

Compared to the $CP_\odot$ method, the $CP_\perp$ method permits to gain 2 orders of convergence for the curvature's L_2 and L_∞ error in the general (ellipse) case while only requiring twice more interpolations. The reinterpolation $CP_\perp^2$ method ameliorates the error in the normal deviation for the circle, which will help stabilize the static column equilibrium case studied below.

4.3. Static viscous column equilibrium with constant density

The problem of the equilibrium of a 2D column subjected to surface tension forces has been widely used as a reference in the litterature, for example Denner et al. [26], Zahedi et al. [27], Fuster et al. [28] propose numerical results and analysis with various methods in VOF or level set frameworks. From the Laplace law, without discretization errors, the surface tension forces should not create any current - the fluid should be at rest - as they are at equilibrium with the pressure jump between the two phases:

$$\nabla p = \frac{1}{We}\kappa \mathbf{n}\delta_\Gamma \tag{30}$$

giving a pressure inside the drop of $p_{drop} = \sigma\kappa$ when the reference pressure outside is 0.

The Weber number is given as

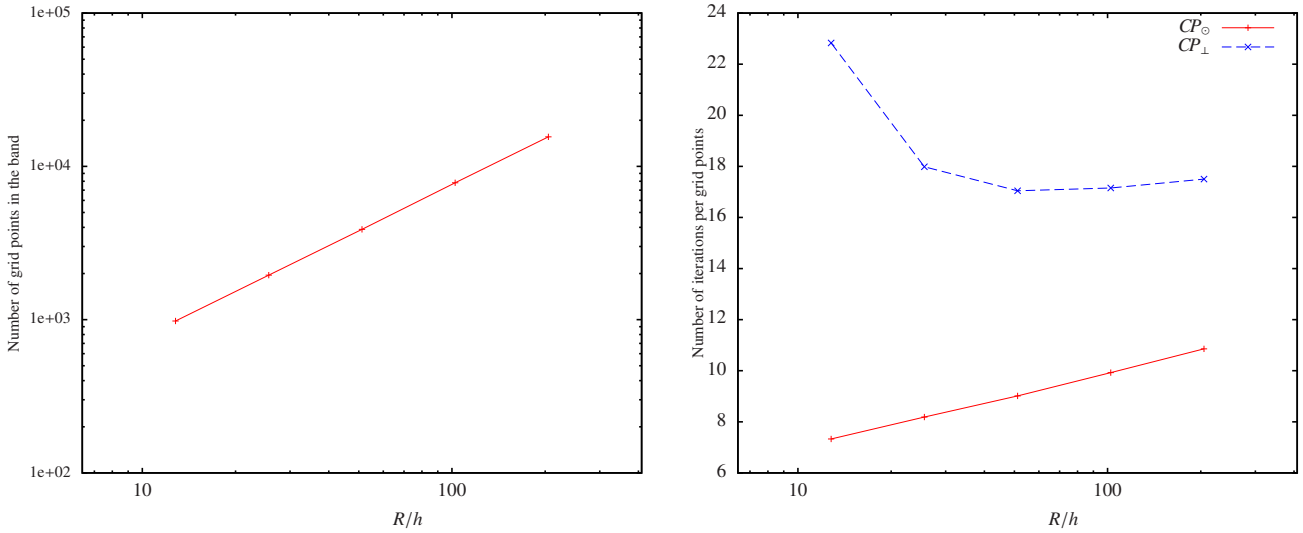
$$We = Re \cdot Ca = \frac{\rho U L}{\mu} \cdot \frac{\mu U}{\sigma} = \frac{\rho U^2 L}{\sigma}, \tag{31}$$

with U (resp. L) a reference velocity (resp. lenght) characterizing the problem. In our problem, $L = D$ the diameter of the column.

Two other relevant non-dimensional numbers are the Laplace and the Ohnesorge number relating the surface tension to the viscous forces:

$$La = \frac{\sigma\rho L}{\mu^2} = Oh^{-2} = \frac{Re^2}{We}. \tag{32}$$

The Laplace number characterizes the numerical behaviour of the flow computed due to numerical errors on the discretization of the surface tension forces, ie. the errors in the localization δ_Γ of the column's interface, its



(a) Linear increase of the number of points in the $6h$ band with respect to R/h.

(b) Mean number of iterations per grid points in the $6h$ band for the closest point algorithms $CP_{\odot}$ and $CP_{\perp}$.

Figure 11: Efficiency of the closest point algorithms.

normal $\mathbf{n}$ and its curvature κ will create capillarity waves thus currents that will propagate in the fluids through the viscous term. The more viscous the fluid is the more smoothly the problem will reach equilibrium. At high Laplace numbers, we expect the interface between the two phases to oscilate/vibrate around its equilibrium state more rapidly and for a longer time.

In the following sections, for clarity, we call "velocity" the root mean square of the fluid velocity as:

$$v_{rms} = \sqrt{\int_{\Omega} |\mathbf{v}(\mathbf{x})|^2 d\mathbf{x}} \tag{33}$$

and the capillary number is defined in absolute value as:

$$Ca = \frac{\mu |\mathbf{v}|_{max}}{\sigma}. \tag{34}$$

If not stated differently, v_{rms} is given in the T_σ and U_σ reference frames as defined below.

As it has been shown in Francois et al. [11], applying a constant curvature κ_{exact} for the surface tension force in a balanced force CSF framework permits to obtain instantly the numerical equilibrium, ie. v_{rms} is of the order of the machine precision. We study below the effect of errors on the computation of the curvature.

Non-dimensional reference frames. All the velocities are given in the velocity scale for the inviscid problem:

$$U_\sigma = \sqrt{\frac{\sigma}{\rho D}}$$

as well as the corresponding time scale:

$$T_\sigma = \sqrt{\frac{\rho D^3}{\sigma}}$$

which is proportional to the period of the capillary waves. We also define the viscous time scale as:

$$T_\mu = \frac{D^2}{\mu}.$$

Time step restriction. As it was clearly demonstrated in Galusinski and Vigneaux [29], the stability of flows subjected to surface tension is waranty by the following condition on the time step:

$$\Delta t \leq \frac{1}{2}\left(c_2 \frac{\mu}{\sigma} h + \sqrt{\left(c_2 \frac{\mu}{\sigma} h\right)^2 + 4 c_1 \frac{\rho}{\sigma} h^3} \right)$$

for sufficiently small Reynolds numbers, where c_1 and c_2 do not depend on the physical and discretization data of the problem. In most simulations studied below, we have respected this time step constraint however we have observed that for smooth enough problems, larger time steps can be used without introducing instabilities in the computations.

4.3.1. General configuration

We consider the physical problem of a column of diameter $D = 2R = 0.4$ centered at the origin of a unit square. The density and the viscosity are set to unity: $\rho = \mu = 1$. The Laplace number is thus varied by changing the surface tension coefficient σ.

The level set is initialized exactly as

$$\phi(\mathbf{x}) = |\mathbf{x}| - R.$$

No symetry property is used, we compute the solutions in the whole domain. No-slip conditions are applied to all boundaries.

The pressure error, considering the reference pressure outside the drop to be zero, is computed as:

$$E(\Delta p_{mean}) = \frac{|p_{2,mean} - p_{1,mean} - p_{drop}|}{p_{exact}}$$

where $p_{i,mean}$ is the mean pressure calculated in the i^{th} fluid, with $i = 2$ for the column, by integrating numerically in the whole domain:

$$p_{i,mean} = \frac{\sum p \cdot H_{\epsilon,i}(\phi/|\nabla\phi|)}{\sum H_{\epsilon,i}(\phi/|\nabla\phi|)}$$

where $H_{\epsilon,i}$ is the regularized characteristic function associated to the i^{th} phase.

4.3.2. Order 2 and order 4 κ_{LS} calculation

We study the dynamic effect of spatial errors due to the κ_{LS} computation precision on the $CP_\perp$ method by varying the scheme used and the number of grid points. The Laplace number is 120, corresponding to $\sigma = 300$, and the time step is set to $\Delta t = 3 \times 10^{-5}$. The curvature errors and the impact on the fluid velocity are shown in figure 12 ; as time evolution for the pressure, the mean curvature and its normal deviation are small we have not reported the plots. The numerical evaluations are reported in table 11 and summarized for the finest grid resolution in table 5.

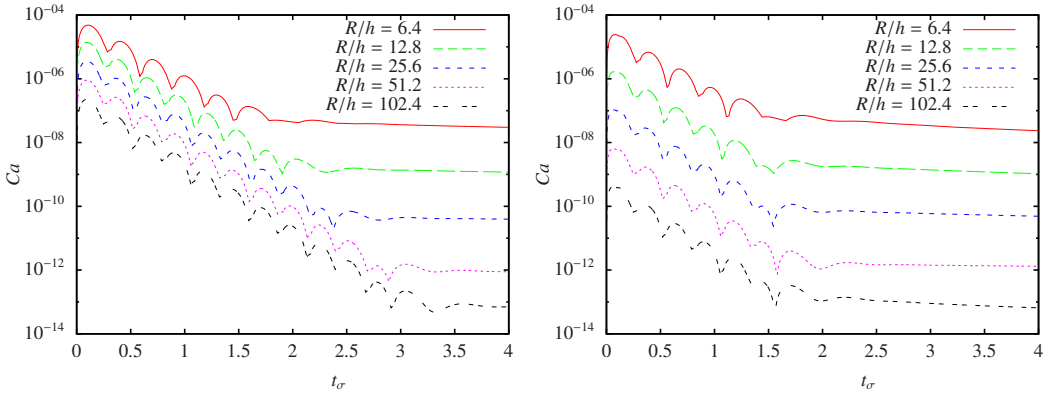
The results clearly show that, compared to the second-order scheme, the fourth-order computation of the κ_{LS} permits to greatly reduce the capillary number in the first time steps at the order 4 as well as the maximum spurious currents. When the approximate numerical equilibrium is reached for all the simulations, around $t_\sigma = 30$ (though after that time the velocity still declines but much less rapidly), the pressure error also converges at fourth-order, as expected from eq. 30 and from the balanced force CSF principle because κ_{mean} is more accurate.

The capillary number (the RMS velocity follows qualitatively the same behaviour thus we do not show it here) converges toward machine precision at even a higher order when the spatial discretization tends to zero. It is noticeable that the second and fourth-order schemes converge toward the approximate same equilibrium Ca and RMS velocity residual. This is due to the physical dynamic of the test case that searches a numerical equilibrium state, which geometrically depends on h, through the level set function displacement by minimizing the standard deviation on the curvature. We see in fig. 12c that, for a same spatial step h, the approximate minimal κ_σ is qualitatively independant on the κ_{LS} accuracy. However, this minimum is reached around $t_\sigma = 1.5$ for any discretization of the 4^{th} order scheme, whereas that time is stretching with decreasing h for the second-order scheme. As with the later, more numerical errors are introduced in the early stages, the numerical equilibrium is reached later implying more oscillations of the interface.

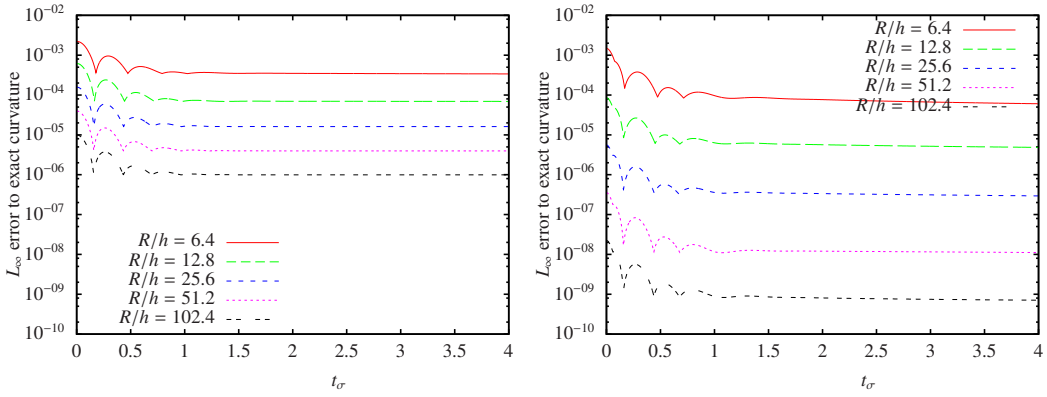
Surprisingly, we observe a faster convergence rate on the capillary number at $t_\sigma = 30$ for the second-order κ_{LS} computation: we believe that at very low resolution such small velocities are close to machine precision and thus very sensitive to numerical perturbations, which are not negligible at those scales. For instance, the maximum velocities for the $R/h = 102.4$ discretization approximately equals 5×10^{-12} which is of the same order of magnitude as the maximum velocity introduced in the flow when applying the constant curvature $\kappa_{ex} = 1/R$.

However, as the curvature and the surface are in better geometrical balance for high-order schemes for κ_{LS} (refer to 4.2 for the geometrical convergence analysis), the errors induced in the first time steps and on the maximum capillary number are much smaller which will lead to reduced errors in the general case. We see in figures 12c and 12a the proportional link between the curvature's standard deviation and the spurious currents that arise mostly because of κ_σ. The normal deviation for this column case is, as expected, equal for κ_{LS} at order 2 and 4 because of the regularity of the particular circular surface as it was shown in §4.2. When the standard deviation becomes stable after 4 half-oscillations, the surface oscillates one more time until parasitic currents are damped by viscosity and the numerical equilibrium is reached.

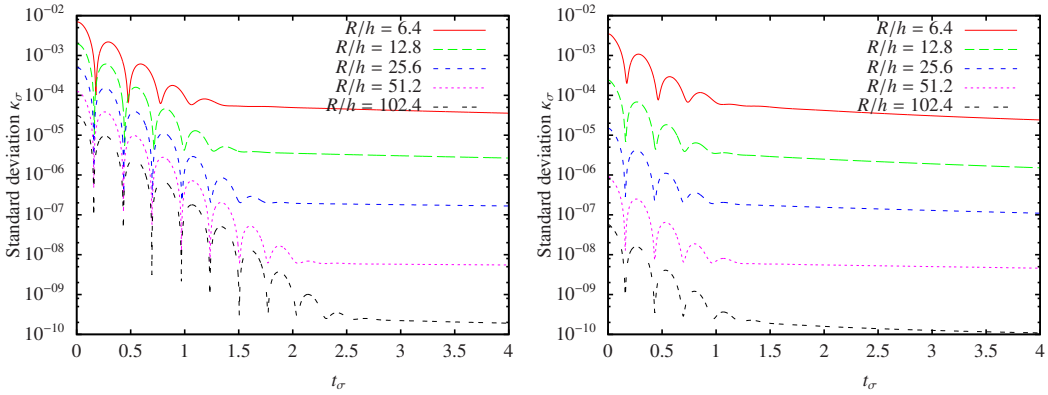
The use of a fifth order WENO scheme for the advection of the level set does not deteriorate this convergence rate as it is expected (down to order $5 - 2 = 3$). We believe that this is due to the smoothness of the level set function and to the very small velocities, which introduce errors in ϕ much less than $O(h^5)$, added to the dynamic of the case that searches to minimize the geometrical errors. We show in §4.5 how the advection of the level set impacts the appearance of spurious currents.



(a) RMS velocity.



(b) L_∞ error to exact curvature.



(c) Standard deviation to the mean curvature.

Figure 12: Static column case: convergence study for the RMS velocity, L_∞ curvature error and standard deviation using the $CP_\perp$ method with 2 iterations and order 2 (left) and order 4 (right) κ_{LS} calculation, $La = 120$.

κ_{LS}	R/h	Ca at $t = \Delta t$	O	Ca at $t_\sigma = 30$	O	$max\, Ca$	O	$E(\Delta p_{mean})$ at $t_\sigma = 30$	O
Order 2 scheme	102.4	1.19×10^{-8}	1.98	1.34×10^{-14}	4.93	2.27×10^{-7}	1.99	9.91×10^{-7}	2.00
Order 4 scheme	102.4	3.74×10^{-11}	4.05	2.08×10^{-14}	4.30	4.13×10^{-10}	3.97	4.24×10^{-10}	3.94

(a) Velocity error measures and pressure error.

κ_{LS}	R/h	$\lvert\frac{\kappa_{mean}-\kappa_{ex}}{\kappa_{ex}}\rvert$	O	L_∞ curvature error	O	κ_σ	O	L_∞ normal dev.	O
Order 2 scheme	102.4	9.91×10^{-7}	2.00	9.91×10^{-7}	2.00	6.31×10^{-11}	5.28	1.68×10^{-11}	5.09
Order 4 scheme	102.4	4.24×10^{-10}	3.94	4.47×10^{-10}	3.95	2.30×10^{-11}	4.93	1.27×10^{-11}	4.57

(b) Curvature error measures at $t_\sigma = 30$.

Table 5: Static column case: numerical results for the finest grid ($R/h = 102.4$) and approximative global order of convergence for the $CP_\perp$ method with 2 iterations, order 2 and order 4 κ_{LS} calculation, $La = 120$. Detailed numerical results are given for varying R/h in table 11.

4.3.3. Varying the Laplace number

We study the effect of a varying Laplace number by changing the surface tension coefficient σ at fixed density and viscosity. Augmenting La will make the interface response to numerical errors "stiffer" thus reaching a stable state after a longer time in the T_σ reference frame.

Here we used the $CP_\perp$ method with 2 iterations with a 4^{th} order κ_{LS} and the spatial discretization is $R/h = 6.4$. The time step is adapted because of the varying surface tension coefficient as $\Delta t = \{3, 1, 0.3, 0.1\} \times 10^{-4}$ for respective $La = \{120, 1200, 12\,000, 120\,000\}$.

We have summarized and compared the numerical results to the ones from the litterature in table 6 where we can see that the spurious currents we obtained are much smaller for the ratio $R/h = 6.4$. Moreover, the convergence rate with respect to h is also much higher. One has to note that in those references Ca is computed at $t_{cap} = t\sigma/(D\mu) = 250$, a time for which the simulation has not converged to the equilibrium velocity which, as we have observed in our simulations (see fig. 13a), is around $t_{cap} = 25,000$ for $La = 120,000$. Thus the velocity at that time is still highly oscillating toward equilibrium and not minimum as it is shown in fig. 13b. Consequently, we have given a new table of Ca number for different $t_\sigma = t/T_\sigma$ (when the velocity is stabilized), which we believe are more representative.

We believe that the results obtained for $R/h = 6.4$ suffer from a too coarse spatial discretization as the regularization width is $\epsilon = 2h \simeq 0.3\,R$. Thus we propose to study the $R/h = 12.8$ case for which we show more complete numerical results in figure 13, where Δt has been divided by 3. The results are qualitatively equivalent to the one obtained by Popinet [13] in a VOF framework for the same discretization. However, the velocity RMS is approximately 20 times smaller in our case thanks to the higher order of convergence of the curvature error of the $CP_\perp$ method. This difference will increase rapidly with finer discretization. As expected, increasing the Laplace number makes the interface stiffer and takes more time to reach equilibrium, where the oscillations observed converge toward an approximate period of $0.44\,T_\sigma$.

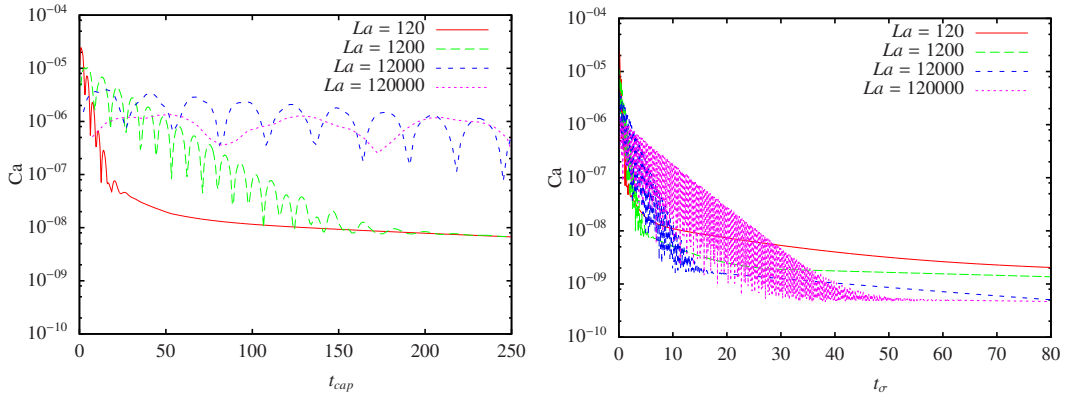
We observe in figures 14c and 14d a reduction of the curvature error measures when augmenting La. The Ca curve follows the standard and normal deviations showing the importance of those two criteria in the reductions of spurious currents. This is consistent with the idea that a higher surface tension coefficient makes the surface's response stiffer thus impacting the level set's movement in the viscinity of Γ making it numerically more homogeneous.

4.4. Static viscous column equilibrium with variable density

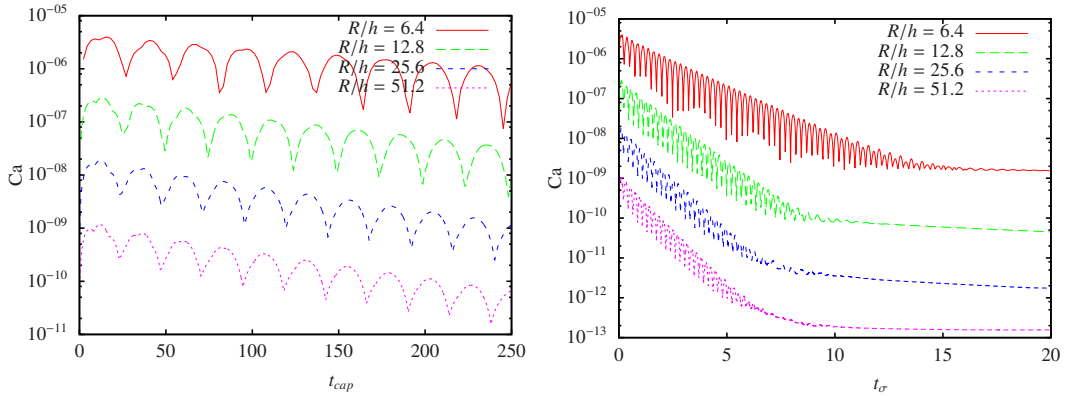
In order to study how the method deals in general fluids simulations, we study the effect of surface tension on the column problem with a variable density.

4.4.1. General configuration

We take the exact same configuration as in §4.3 but change the density ρ_1 of the fluid inside the column while keeping the density outside ρ_2 fixed and σ constant. This has the effect to increase the Laplace number. As ρ_2 does not change, the time step is restricted by this lower bound and we fixed $\Delta t = 1 \times 10^{-4}$. However,



(a) Varying the Laplace number for $R/h = 6.4$.



(b) Varying the spatial discretization for $La = 12,000$.

Figure 13: Static column case: convergence study using the $CP_\perp$ method with 2 iterations and order 4 κ_{LS} calculation in the T_{cap} (left) and T_σ (right) reference frames.

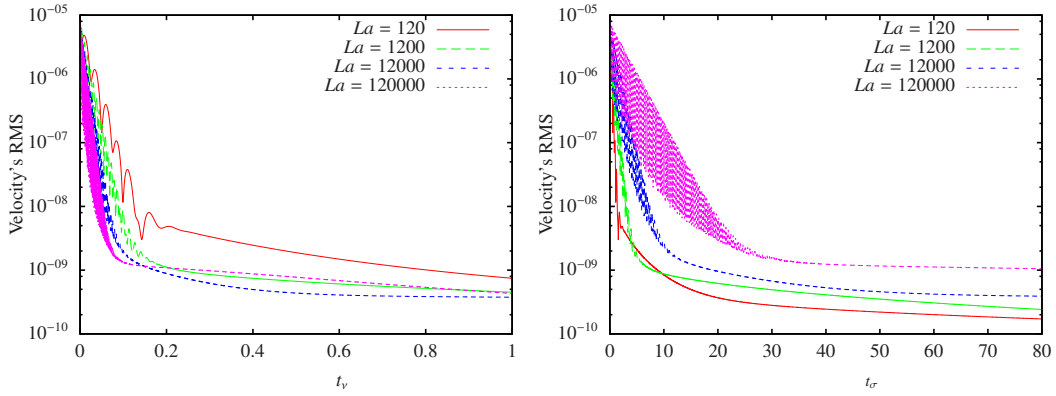
R/h	Ca		La			
			120	1200	12 000	120 000
6.4	At $t_{cap} = 250$	Proposed method	6.74×10^{-9}	6.82×10^{-9}	5.42×10^{-7}	2.68×10^{-7}
		Owkes and Desjardins [30]	4.59×10^{-7}	7.80×10^{-7}	2.18×10^{-6}	3.00×10^{-6}
		Herrmann [15] cartesian	1.10×10^{-7}	1.20×10^{-7}	1.44×10^{-6}	3.09×10^{-6}
		Shin et al. [31]	2.18×10^{-6}	2.18×10^{-6}	2.22×10^{-6}	-
		Popinet and Zaleski [32]	5.71×10^{-6}	5.99×10^{-6}	8.76×10^{-6}	-
	At $t_\sigma = 80$	Proposed method	2.04×10^{-9}	1.37×10^{-9}	5.02×10^{-10}	4.68×10^{-10}
12.8	At $t_\sigma = 80$	Proposed method	9.16×10^{-11}	3.58×10^{-11}	1.78×10^{-11}	1.03×10^{-11}

(a) Varying Laplace number at $R/h = 6.4$.

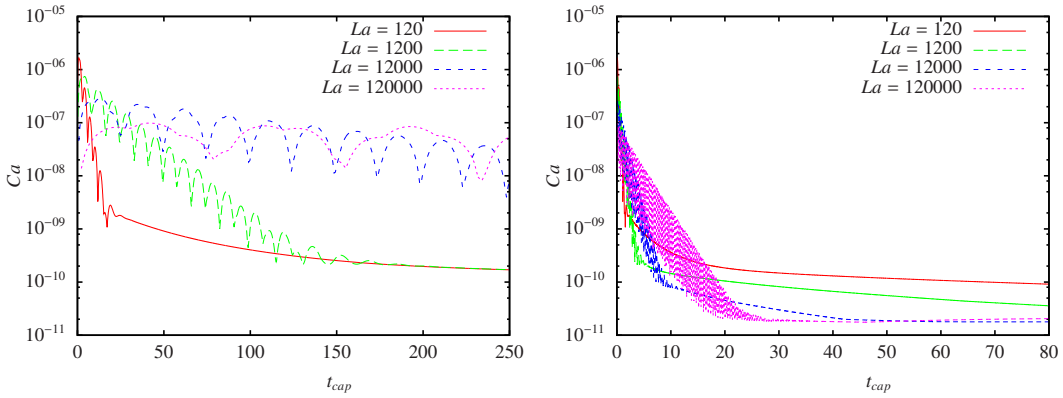
Ca		R/h			
		6.4	12.8	25.6	51.2
At $t_{cap} = 250$	Proposed method	5.42×10^{-7}	9.21×10^{-9}	1.13×10^{-9}	6.53×10^{-11}
	Owkes and Desjardins [30]	2.18×10^{-6}	2.32×10^{-7}	5.91×10^{-8}	-
	Herrmann [15] cartesian	1.44×10^{-6}	3.40×10^{-7}	5.00×10^{-8}	-
	Popinet and Zaleski [32]	6.68×10^{-6}	1.07×10^{-6}	1.20×10^{-7}	-
At $t_\sigma = 20$	Proposed method	1.52×10^{-9}	4.49×10^{-11}	1.72×10^{-12}	1.55×10^{-13}

(b) Spatial convergence for $La = 12\,000$ compared to results from the litterature.

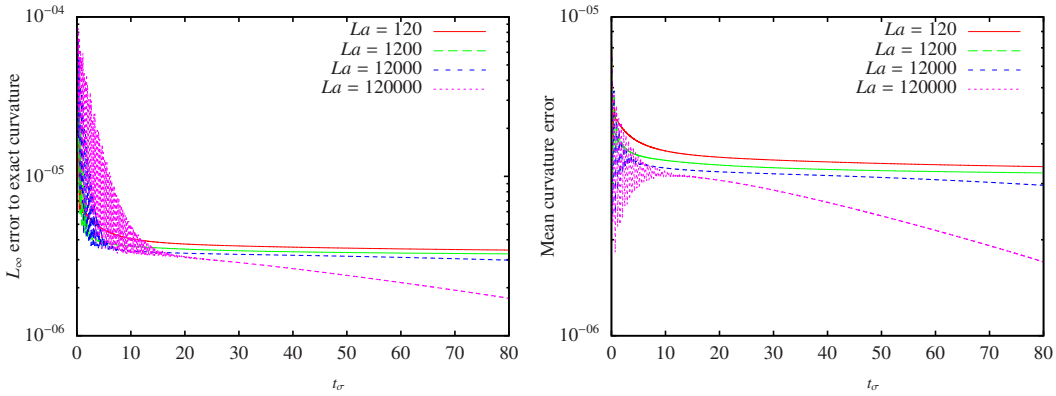
Table 6: Static column case: convergence of capillary number in function of the Laplace number and R/h with the proposed method compared to results from Herrmann [15], Popinet and Zaleski [32], Shin et al. [31], Owkes and Desjardins [30].



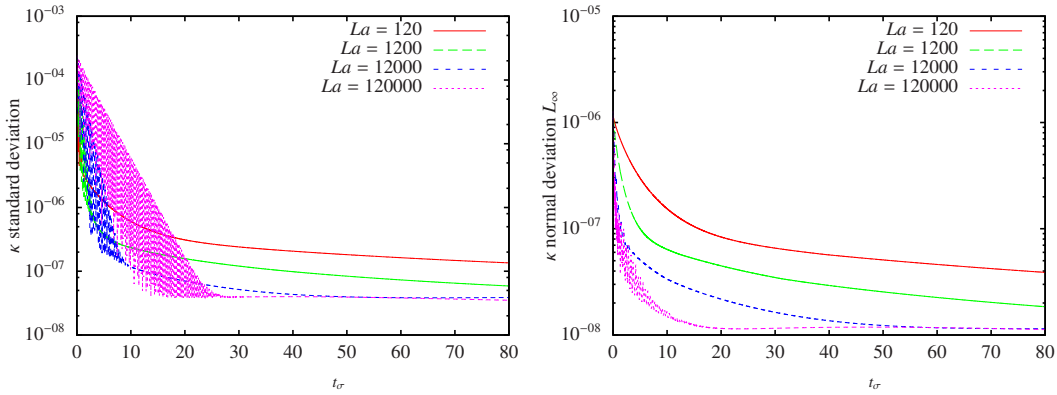
(a) Velocity's RMS for different time scales: T_ν and T_σ.



(b) Capillary number for different time scales: T_{cap} and T_σ.



(c) L_∞ error to exact curvature and error on the mean curvature.



(d) Standard deviation and L_∞ normal deviation.

Figure 14: Static column case: convergence study by varying the Laplace number using the $CP_\perp$ method with 2 iterations and order 4 κ_{LS} calculation with $R/h = 12.8$.

for increasing density, we have observed in the first time steps a high error in the velocity field that we believe due to the quick diffusion of the initial errors on the surface tension term by the reduced cinematic viscosity $\nu = \mu/\rho$. To counter this phenomenon, we reduced Δt to 1×10^{-6} for the first hundred computational steps, leaving time for the simulation to stabilize, and smoothly increased it to 1×10^{-4}.

4.4.2. Varying the density

The surface tension coefficient is set to 300 while the density is varied as $\rho_1 = \{1, 10, 100, 1000\}$ with corresponding Laplace numbers $La = \{120, 660, 6060, 60\,060\}$, calculated as:

$$La_{\rho_1,\rho_2} = \frac{\sigma \overline{\rho} L}{\mu^2} \tag{35}$$

with

$$\overline{\rho} = \frac{\rho_1 + \rho_2}{2}.$$

The reference scales T_σ and U_σ are defined with that same mean density.

Figure 15 shows coherent results with the constant density test in §4.3 that permits us to conclude the stability and accuracy of the method in the presence of variable density.

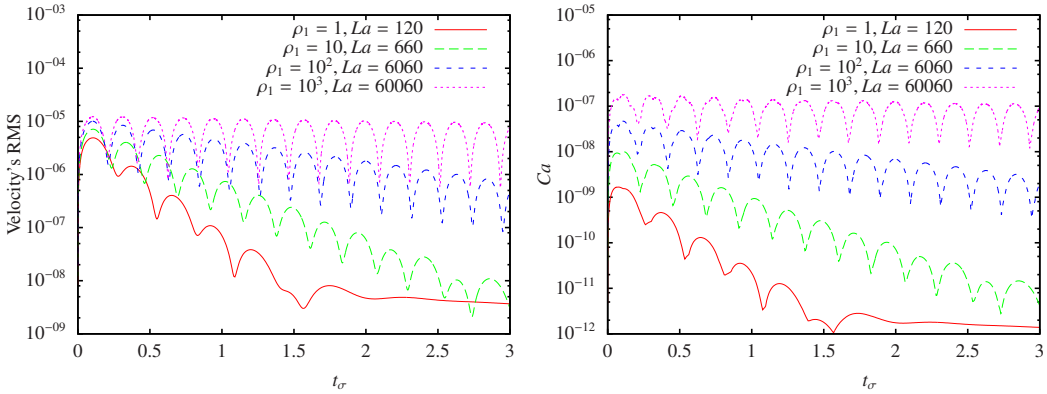


Figure 15: Static column case with variable density: convergence study for the velocity's RMS and Ca number by varying the density inside the column using the $CP_\perp$ method with 2 iterations and order 4 κ_{LS} with $R/h = 12.8$.

4.5. Advected viscous column equilibrium

In this test cases, we aim to study the impact of the errors due to the advection of the surface - ie. through the level set advection - by an initial constant velocity field. Theoreticaly, in the reference frame of the transported column, the spurious currents should tend to zero. But because of the grid aliasing effect and advection errors we expect the interface to reach equilibrium with more difficulty, while still creating minimized spurious currents with the proposed method.

4.5.1. General configuration

The fluid parameters used are the same as in §4.3: $\rho = \mu = 1$ in the two phases. The domain is a rectangle of size $[1, 2]$ with symetry conditions for the top and bottom boundaries, and periodic in the horizontal direction. The column has a radius of $R = 0.2$. We initialized the x-velocity to $U_x = 2$ in the whole domain so that, idealy, the velocity field should remain $\mathbf{U} = (U_x, 0)$ through time. The spurious currents are measured in the column reference frame and calculated as:

$$v_{rms} = \sqrt{\int_\Omega |\mathbf{v}(\mathbf{x}) - \mathbf{U}|^2 d\mathbf{x}}$$

and

$$Ca = \frac{\mu |\mathbf{v} - \mathbf{U}|_{max}}{\sigma}.$$

All the velocities are given in the velocity scale for the advection problem:

$$U_{adv} = \sqrt{U_\sigma \cdot U_x}$$

as well as the corresponding time scale:

$$T_{adv} = D/U_x.$$

The CFL involved here will be much larger than the time step restriction due to surface tension: $\Delta t_{CFL} \gg \Delta t_\sigma$.

La	R/h	Mean v_{rms}	O	Mean Ca	O	$max\,Ca$	O	$\frac{max\,Ca}{mean\,Ca}$
1200	6.4	7.83×10^{-5}	-	1.23×10^{-6}	-	1.05×10^{-5}	-	8.53
	12.8	2.02×10^{-6}	5.28	3.73×10^{-8}	5.04	7.57×10^{-7}	3.79	20.27
	25.6	6.94×10^{-8}	4.86	1.56×10^{-9}	4.58	4.87×10^{-8}	3.96	31.20
	51.2	3.65×10^{-9}	4.25	1.10×10^{-10}	3.83	3.07×10^{-9}	3.99	28.02
	102.4	1.17×10^{-10}	4.96	5.00×10^{-12}	4.46	1.91×10^{-10}	4.01	38.27
12 000	6.4	3.86×10^{-4}	-	1.10×10^{-6}	-	3.93×10^{-6}	-	3.56
	12.8	5.38×10^{-6}	6.16	1.55×10^{-8}	6.16	2.89×10^{-7}	3.77	18.66
	25.6	3.11×10^{-7}	4.11	9.90×10^{-10}	3.97	1.90×10^{-8}	3.92	19.23
	51.2	2.81×10^{-8}	3.47	1.02×10^{-10}	3.27	1.19×10^{-9}	4.00	11.60
	102.4	7.35×10^{-10}	5.26	3.02×10^{-12}	5.08	7.57×10^{-11}	3.97	25.08

Table 7: Advected column case: numerical results for varying R/h and approximative order of convergence for the $CP_{\perp}$ method with 2 iterations and order 4 κ_{LS} calculation. Mean measures are taken in the intervals $t_{adv} \in [0.06, 0.2]$ for $La = 1200$ and $t_{adv} \in [0.03, 0.1]$ for $La = 12\,000$, v_{rms} is given in the U_{adv} reference frame.

4.5.2. Spatial convergence

We study the impact of the advection error of the level set coupled with the errors due to the surface representation and curvature calculation by varying the spatial discretization in the same maner as in §4.3. This case was first introduced by Popinet et al. in Popinet [13] in a VOF method with Height Field curvature framework. We have set $\Delta t = 3 \times 10^{-6}$ for the $La = 1200$ simulations and $\Delta t = 1 \times 10^{-6}$ for $La = 12\,000$, except for the $R/h = 102.4$ discretization where we have divided Δt by three to avoid instabilities.

As shown in figure 16, for early times, we observe the same exact numerical results as in §4.3 as the error dominating here are due to the fact that the interface is not at equilibrium. After time $t_{adv} = 0.04$ (resp. $t_{adv} = 0.02$) for $La = 1200$ (resp. $La = 12\,000$) we clearly observe the effects of the errors on the advection perturbing the interface and the curvature in small oscillations (repeating patterns) around the equilibrium state. The period of those perturbations of scale $1/h$ in the T_{adv} reference frame. This result is due to aliasing where the interface - ie. the level set - suffers from approximately the same numerical discretization aliasing every time it has been translated by one cell. This observation coincides with the remarks given in Popinet [13]. Yet as our method for advection and curvature calculation is more precise as the errors in the spurious currents are much smaller (by a factor around 3×10^{-4} for the $R/h = 12.8$ discretization) as it is shown in figure 16 and table 7.

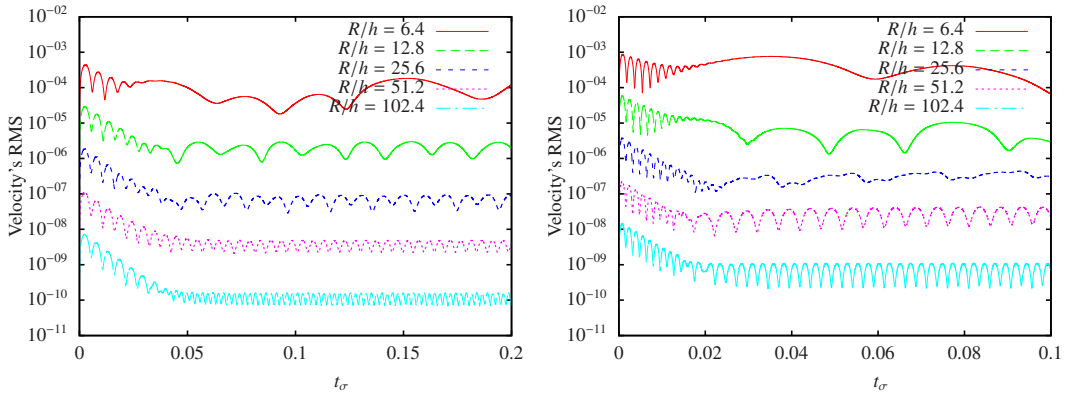
As the simulations never reach a non oscillating equilibrium, we computed mean values for the velocity's RMS and Ca in the interval $t_{adv} = [0.06, 0.2]$ for $La = 1200$ and $t_{adv} = [0.03, 0.1]$ for $La = 12\,000$. The mean capillary number computed at equilibrium is much higher than the one of the static case, the ratio $\frac{max\,Ca}{mean\,Ca}$ gives a good indication of how the method efficiently manages to reduce the spurious currents counteracting the errors constrantly introduced on the level set. The curvature error measures presented in fig. 16c were also computed as a mean in the interval $t_{adv} = [0.06, 0.2]$ for $La = 1200$. The orders of convergence are approximately equal to 4 for the L_∞ and standard deviation, and between 3.5 and 4 for the normal deviation, a reduction for the later that is due to the 5^{th} order WENO advection scheme as expected by proposition 6.

We observe in fig. 16a two different regimes: a first one where the non-equilibrate geometrical circle tends to find its equilibrium, followed by a second one where the errors in the solving of the ϕ transport equation become dominant and avoid to reach the converged equilibrium as studied in §4.3. The peak capillary number converges at the order 4 as observed previously in §12 while the mean v_{rms} and the mean Ca converge between the order 4 and 5, which is notably higher than the results obtained for the static case in §4.3. We observe in fig. 16a higher amplitudes for Ca in the transport error regime (compared to the initial geometry equilibrium search regime) for coarser resolutions where the grid aliasing is bigger and the order 2 Navier-Stokes errors are more present. Hence the errors on the velocities get more reduced as $h \to 0$, converging faster toward the "optimum" reference values fixed by the static case results thus showing an order of convergence higher than 4.

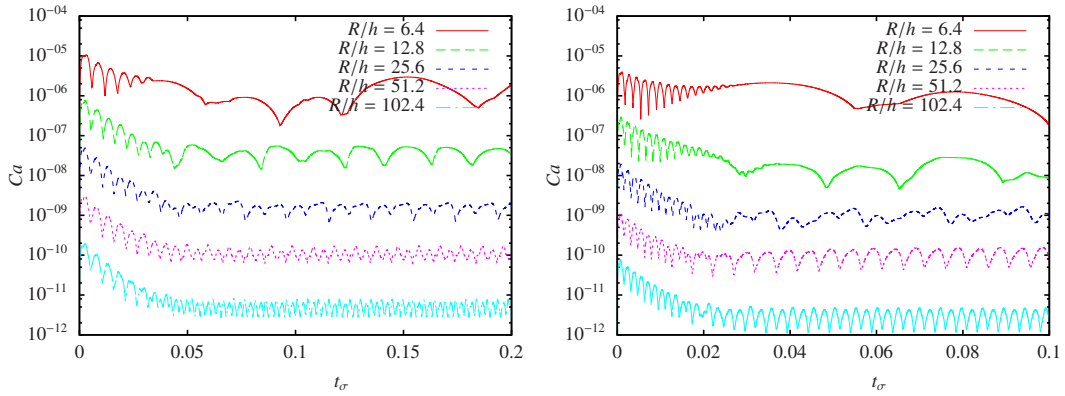
As shown in figure 16c, the spatial convergences for the capillary number (similarly to the RMS velocity) is at least order 4 which is 3 orders better than the L_2 norm presented in Popinet [13] while their L_∞ measures do not show convergence while ours - through the Ca number - is better than fourth-order. This clearly demonstrates that the use of a high-order advection scheme coupled with a high-order curvature extension is necessary in order to reduce the spurious currents in general flow simulations.

4.6. Zero gravity drop oscillation

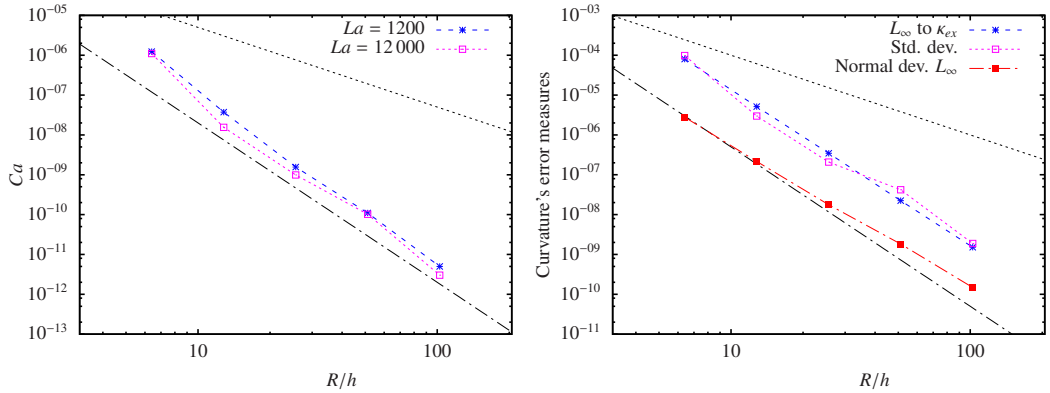
In order to study the dynamic behaviour of our method in presence of a non constant curvature, we have measured the oscillation period of a non circular drop. The analytical results proposed in Lamb [33] have been numerical studied in Herrmann [15], Tryggvason et al. [4], Shepel and Smith [34] and Torres and Brackbill [35] in various frameworks.



(a) Velocity RMS for varying R/h, $La = 1200$ (left) and $La = 12\,000$ (right).



(b) Capillary number for varying R/h, $La = 1200$ (left) and $La = 12\,000$ (right).



(c) Left: mean Ca for $La = 1200$ and $La = 12\,000$. Right: mean curvature error measures for $La = 12\,000$. Spatial convergence with respect to R/h for mean values computed in the intevals $t_{adv} \in [0.06, 0.2]$ for $La = 1200$ and $t_{adv} \in [0.03, 0.1]$ for $La = 12\,000$.

Figure 16: Advected column case: convergence study for the RMS velocity, Ca and curvature error measures for the $CP_\perp$ method with 2 iterations and order 4 κ_{LS} computation, in the U_{adv} and T_{adv} reference frames.

E_ω	R/h					
	6.4	O	12.8	O	25.6	O
Herrmann [15] cartesian	4.04×10^{-2}	-	1.05×10^{-2}	1.94	0.37×10^{-2}	1.5
Torres and Brackbill [35]	13.2×10^{-2}	-	6.1×10^{-2}	1.11	1.5×10^{-2}	2.02
Proposed method	7.4×10^{-2}	-	1.54×10^{-2}	2.28	0.53×10^{-2}	1.54

T_{calc}	R/h			
	6.4	12.8	25.6	∞
Proposed method	7.83	7.40	7.33	7.31
Analytical Lamb [33]	-	-	-	7.29138

Table 8: Spatial convergence of the oscillation error for the zero gravity drop oscillation. The value for T_{calc} at $R/h = \infty$ with our method has been computed with a Richardson's extrapolation .

A 2D circular droplet is perturbed by a cosine function of mode n and amplitude A_0. The theorical oscillation period ω as given by Lamb [33] is:

$$\omega^2 = \frac{n(n^2-1)\sigma}{(\rho_1+\rho_2)R_0^3}$$

where ρ_1 (resp. ρ_2) is the density of the fluid inside the droplet (resp. outside).

4.6.1. General configuration

An initial column of radius $R_0 = 2$ is placed in the center of a $[-10, 10]$ square box with slip boundary conditions. The surface tension coefficient is fixed to $\sigma = 1$ and the physical parameters $\rho_1 = 1$, $\rho_2 = 0.01$, $\mu_1 = 0.01$ and $\mu_2 = 1 \cdot 10^{-4}$ which from eq. 35 corresponds to $La \approx 80\,000$. The column is perturbed by a cosine function which we translated in the level set representation as:

$$\phi(x) = |\mathbf{x}| - (R_0 + A_0 cos(n\theta))$$

where we set $A_0 = 0.01\, R_0$ and $n = 2$. The time step was chosen as $\Delta_t = 3 \cdot 10^{-3}$.

4.6.2. Spatial convergence

Table 8 shows a comparison of the results in the litterature compared to the results obtained with our method. The oscillation error is calculated as $E_\omega = \left|\frac{T_{calc}\omega}{2\pi} - 1\right|$ with T_{calc} measured as between the picks in the surface minimum x position, averaged over the 3 first periods. The results show better results by a factor of 3 Torres and Brackbill [35] while higher error by a factor of 1.5 than Herrmann [15]. We see an approximative second-order of convergence.

As the oscillation error measures the dynamic behaviour of the drop, we do not expect orders of convergence as high as the error on spurious currents. Moreover, the theory is based on a linear regime study in an infinite fluid thus the complex numerical conditions for the simulation may never converge toward the analytical result but more toward a numerical equilibrium. The oscillation period extrapolated with Richardson's method on our measures is 7.31 which yields to a better than second-order spatial convergence.

4.7. 2D bubble rise

4.7.1. General configuration

For this study we used the $CP_\perp$ algorithm with 2 iterations and order 4 κ_{LS} calculation.

We consider the rise of a 2D bubble under the effect of buoyancy following Hysing et al. [36], Zahedi et al. [27] numerical studies. A disc of diameter $D = 0.5$ lies at the position $[0.5, 0.5]$ at $t = 0$, inside a domain of size $[0, 0] \times [1, 2]$ with no-slip conditions on top and bottom walls and free-slip on left and right boundaries. The densities and viscosities inside (resp. outside) the bubble are ρ_2 and μ_2 (resp. ρ_1 and μ_1) and the gravity is set to 0.98.

The reference length is $L_{rise} = D = 0.5$ and the associated time scale is $T_{rise} = L_{rise}/U_{rise}$ with $U_{rise} = \sqrt{gD}$.

We study the evolution of the center of mass of the bubble numerically computed in the whole domain as:

$$(x_c, y_c) = \frac{\sum \mathbf{x}\, H_\epsilon(\phi/|\nabla\phi|)}{\sum H_\epsilon(\phi/|\nabla\phi|)}.$$

The rising bubble velocity is computed in a similar fashion as:

$$(u_c, v_c) = \frac{\sum \mathbf{u}\, H_\epsilon(\phi/|\nabla\phi|)}{\sum H_\epsilon(\phi/|\nabla\phi|)}.$$

Case	ρ_1	ρ_2	μ_1	μ_2	σ
A	1000	100	10	1	24.5
B	1000	1	10	0.1	1.96

Table 9: 2D bubble rise case: physical parameters.

The circularity shows information on the deformation of the bubble surface an is writen:

$$c = \frac{\pi D}{perimeter} \tag{36}$$

where the perimeter is numerically calculated with:

$$perimeter = h^2 \sum \delta_\epsilon(\phi/|\nabla\phi|). \tag{37}$$

The time step used in all simulations is $\Delta t = 3 \times 10^{-4}$.

4.7.2. Spatial convergence

As in Zahedi et al. [27] in a finite element and level set framework, we propose to study the accuracy of the method in comparison to the numerical benchmark presented in Hysing et al. [36] with various numerical methods. The physical parameters of the two reference cases are presented in table 9.

The numerical results for test case A are illustrated in figure 17 and numerically summarized and compared to the litterature in table 10. They are in good agreement with the reference benchmark permitting us to validate our method for this rising bubble case. The maximum velocity of the bubble and the time it arises both fit in the reference values and converge at the order 2. However, we observe a more important difference in the the bubble circularity and center of mass between our results and the benhmark than in Zahedi et al. [27] for the same discretization. First, the circularity is computed in a level set framework from equations 36 and 37 which is hard as in a Eulerian framework it involves the regularized singular Dirac's mass thus bringing oscillations because of the aliasing due to the grid. Hence the computation of the value c_{min} and the measure of the time it is reached becomes more sensitive. We also observe a lower center of mass at $t = 3$.

We believe that those small deviations come from a better precision in the curvature computation at small scales with our method, particularly on the bottom left and right corners where the curvature becomes high, thus keeping the bubble more circular hence showing more resistance to the vertical motion.

Figure 18 shows qualitative results for test case B through the plots of the bubble's surface, simulated with a $R/h = 80$ discretization. In the early times, $t < 2.2$, we see a similar behaviour of the bubble to the one presented in Hysing et al. [36]. After $t = 2.2$ we observe a wider width of the dragged filaments that are kept "alive" whereas in Hysing et al. [36] they disappear in the flow as residual small bubbles with the TP2D (finite elements and level set with reinitialization) method and a very thin filament with MoonNMD (finite elements with a Lagrangian surface representation) and Fluent. We believe that the filaments we observe are due to:

1. The use of a regularized surface of width $\epsilon = 2h$ that smoothes the blending zone thus the dynamic of the bubble. This zone gets finer as $h \to 0$, though it does not disappear.
2. The more accurate computation of κ in high curvature areas, ie. where the filaments arise, that computes a more precise surface tension force preventing an early decomposition of the surface.

5. Conclusion

In this article we have presented an improved method for the accurate computation of curvature in a level set framework. A complementary predicate (prop. 5) concerning the normal deviation of the curvature in the CSF approach has been introduced and analyzed. We have studied the effect of perturbations on the surface's representation showing that it is necessary to use high-order transport technics to calculate accurate and steady surface tension forces in a dynamic flow.

We have illustrated the curvature calculus in a level set representation to understand the necessity of an adequate curvature extension. The proposed $CP_\perp$ method with reinterpolation yields to a fourth-order precision closest point approximation, enough to obtain fourth-order precision for the curvature error and standard deviation, and second-order for the normal error for general geometry. The algorithm is easy to implement, compared for example to Height Field methods in a VOF framework Cummins et al. [12] and later works, straightforwardly extensible to three dimensions while not being too computationaly costly.

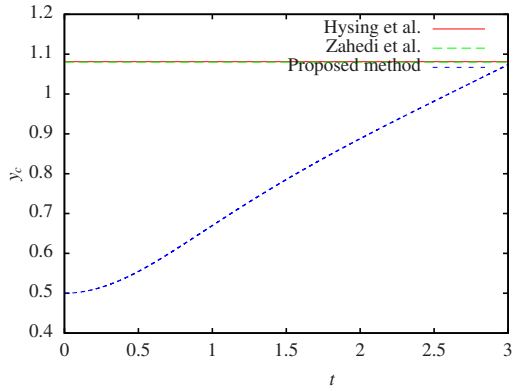
Measure	Proposed method	Zahedi et al. [27] coupled method	Ref. Hysing et al. [36]
c_{min}	0.914	0.89955	0.9012 ± 0.0001
$t\|c = c_{min}$	1.817	1.909	1.9
$v_{c,max}$	0.2421	0.24190	0.2419 ± 0.0002
$t\|v = v_{c,max}$	0.9222	0.929	$0.921 \leq t \leq 0.932$
$y_c\|t = 3$	1.0761	1.07989	1.081 ± 0.001

(a) Comparative results with $CP_\perp$ and order 4 κ_{LS}, for $R/h = 80$.

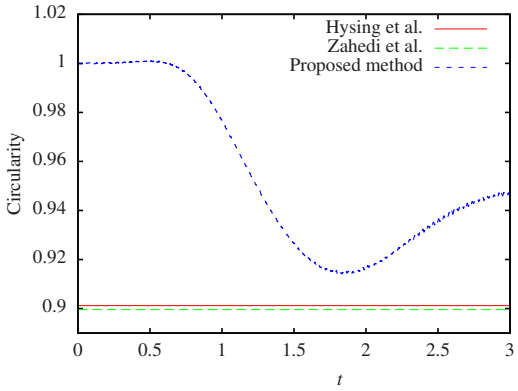
Measure	R/h				Richardson's extrapolation	O
	10	20	40	80		
$CP_\perp$ and order 4 κ_{LS}						
c_{min}	0.976	0.931	0.924	0.914	0.910	1.8
$t\|c = c_{min}$	1.769	1.934	1.806	1.817	1.826	1.2
$v_{c,max}$	0.2385	0.2411	0.2419	0.2421	0.2422	2.0
$t\|v = v_{c,max}$	0.9435	0.9282	0.9237	0.9222	0.9217	1.9
$y_c\|t = 3$	1.0587	1.0635	1.0708	1.0746	1.0761	1.8

(b) Results with the proposed method and the simple κ_{LS} method with no curvature extension, varying R/h. The order of convergence is computed by comparing the numerical results for the two finest discretizations comparatively to the Richardson's extrapolation.

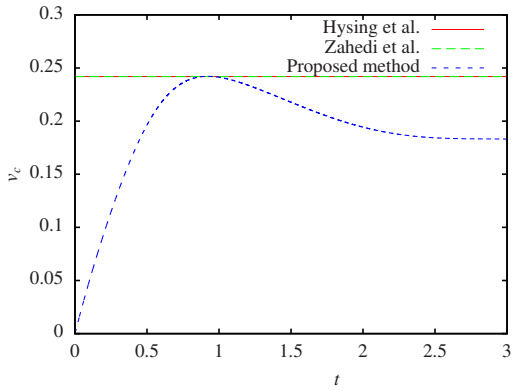
Table 10: 2D bubble rise case: numerical results compared to Hysing et al. [36], Zahedi et al. [27].



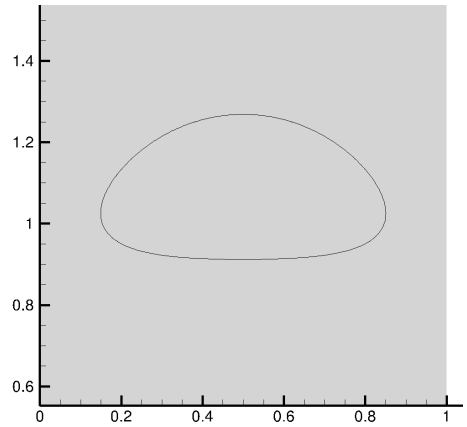
(a) Center of mass y_c.



(b) Center of mass y_c.

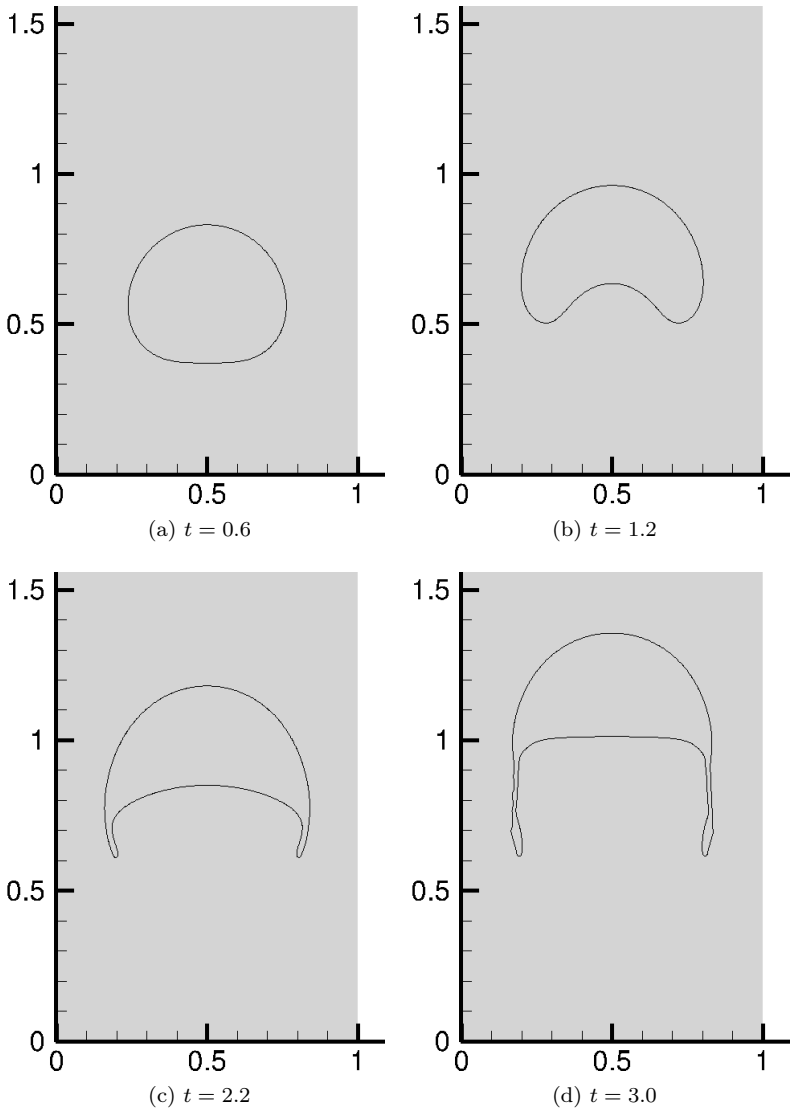


(c) Center of mass y_c.



(d) Center of mass y_c.

Figure 17: 2D bubble rise case A: center of mass, circularity, rise velocity and bubble's shape at $t = 3$ for $R/h = 80$. Extremum results from Hysing et al. [36], Zahedi et al. [27] are drawn as lines.



(a) $t = 0.6$ (b) $t = 1.2$

(c) $t = 2.2$ (d) $t = 3.0$

Figure 18: 2D bubble rise case B: bubble's shape for different times, for $R/h = 80$.

We have validated our approach on geometrical, static, translating and complex dynamical cases in comparison to the litterature. Particularly we have showed that a fourth-order accurate curvature and surface tension forces computation can greatly reduce the spurious currents, even when the surface is transported, although the flow solver remains second-order accurate. We also observed a good agreement of our numerical results with the state of the art in more complex simulations like the rise of a bubble.

A. Appendix

A.1. Second derivative error amplitude

In this paragraph we study the effect of numerical errors on derivatives through the introduction of perturbations in an analytical smooth function ψ. Let ψ, e and ϕ three $\mathbb{R} \to \mathbb{R}$ functions defined as:

$$\begin{aligned} \psi(x) &= cos(2\pi x), \\ e(x,h,m) &= h^m cos(2\pi x/h), \\ \phi(x,h,m) &= cos(2\pi x) + h^m cos(2\pi x/h), \end{aligned}$$

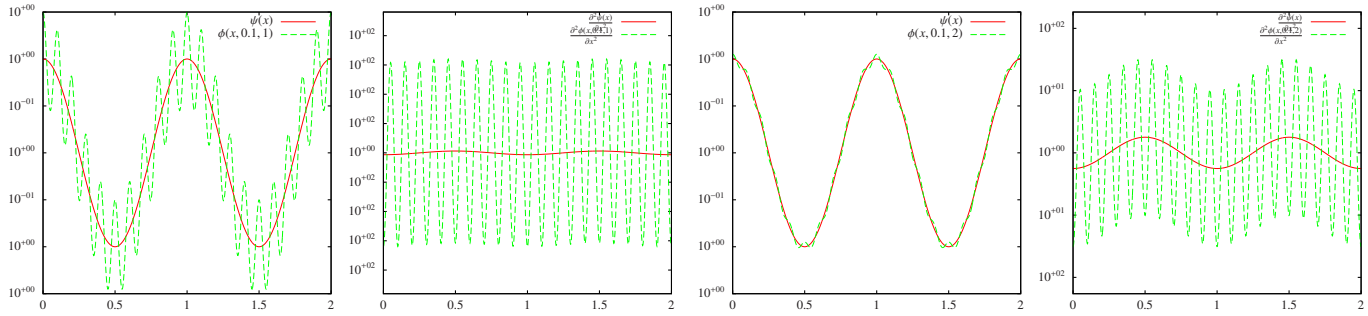
as represented in figure 19. We can easily calculate the second derivative of ϕ:

$$\frac{\partial^2 \phi}{\partial x^2} = -4\pi^2 (sin(2\pi x) + h^{m-2} sin(2\pi x/h))$$

which is equal to $\frac{\partial^2 \psi}{\partial x^2}$ perturbed at the scale h with an amplitude of the order h^{m-2}.
The error in the second derivative is measured by:

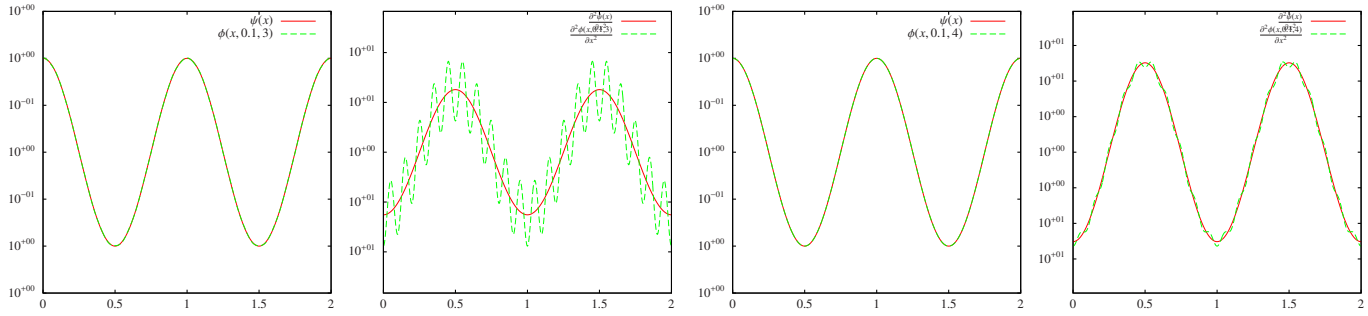
$$error(x,h,m) = \left| \frac{\partial^2 \phi/\partial x^2 - \partial^2 \psi/\partial x^2}{\partial^2 \psi/\partial x^2} \right|$$

when $\partial^2 \psi/\partial x^2 \neq 0$ and is plotted in figure 19 for different perturbations scales. We clearly see that perturbations bigger that h^2 lead to errors superior to 1 on the second derivative which is unpropitious in numerical computations.



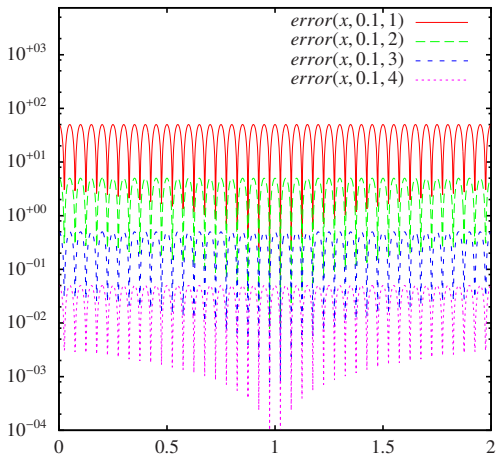
(a) $O(h^1)$ perturbation.

(b) $O(h^2)$ perturbation.



(c) $O(h^3)$ perturbation.

(d) $O(h^4)$ perturbation.



(e) Relative error summary.

Figure 19: Second derivative errors in presence of perturbations.

A.2. Detailed numerical results

κ_{LS}	R/h	Ca at $t = \Delta t$	O	Ca at $t_\sigma = 30$	O	$max\, Ca$	O	$E(\Delta p_{mean})$ at $t_\sigma = 30$	O
Order 2 scheme	6.4	2.02×10^{-6}	-	6.59×10^{-9}	-	4.83×10^{-5}	-	3.05×10^{-4}	-
	12.8	6.60×10^{-7}	1.62	1.96×10^{-10}	5.07	1.38×10^{-5}	1.81	6.71×10^{-5}	2.18
	25.6	1.82×10^{-7}	1.86	9.73×10^{-12}	4.33	3.60×10^{-6}	1.94	1.60×10^{-5}	2.07
	51.2	4.72×10^{-8}	1.95	4.09×10^{-13}	4.57	9.03×10^{-7}	1.99	3.96×10^{-6}	2.01
	102.4	1.19×10^{-8}	1.98	1.34×10^{-14}	4.93	2.27×10^{-7}	1.99	9.91×10^{-7}	2.00
Order 4 scheme	6.4	1.50×10^{-6}	-	5.26×10^{-9}	-	2.43×10^{-5}	-	2.63×10^{-5}	-
	12.8	1.30×10^{-7}	3.53	1.47×10^{-10}	5.16	1.67×10^{-6}	3.87	3.55×10^{-6}	2.89
	25.6	9.29×10^{-9}	3.81	7.58×10^{-12}	4.28	1.05×10^{-7}	3.98	1.75×10^{-7}	4.35
	51.2	6.20×10^{-10}	3.90	4.11×10^{-13}	4.21	6.47×10^{-9}	4.03	6.52×10^{-9}	4.74
	102.4	3.74×10^{-11}	4.05	2.08×10^{-14}	4.30	4.13×10^{-10}	3.97	4.24×10^{-10}	3.94

(a) Velocity errors measures and pressure error.

κ_{LS}	R/h	$\lvert\frac{\kappa_{mean}-\kappa_{ex}}{\kappa_{ex}}\rvert$	O	L_∞ curvature error	O	κ_σ	O	L_∞ normal dev.	O
Order 2 scheme	6.4	3.05×10^{-4}	-	3.10×10^{-4}	-	1.01×10^{-5}	-	2.65×10^{-6}	-
	12.8	6.71×10^{-5}	2.19	6.73×10^{-5}	2.20	4.60×10^{-7}	4.45	1.18×10^{-7}	4.49
	25.6	1.60×10^{-5}	2.07	1.60×10^{-5}	2.07	4.54×10^{-8}	3.34	1.14×10^{-8}	3.36
	51.2	3.96×10^{-6}	2.01	3.96×10^{-6}	2.01	2.45×10^{-9}	4.21	5.71×10^{-10}	4.32
	102.4	9.91×10^{-7}	2.00	9.91×10^{-7}	2.00	6.31×10^{-11}	5.28	1.68×10^{-11}	5.09
Order 4 scheme	6.4	2.66×10^{-5}	-	2.97×10^{-5}	-	6.92×10^{-6}	-	1.85×10^{-6}	-
	12.8	3.56×10^{-6}	2.90	3.66×10^{-6}	3.02	2.39×10^{-7}	4.86	6.52×10^{-8}	4.82
	25.6	1.75×10^{-7}	4.35	1.83×10^{-7}	4.32	1.80×10^{-8}	3.73	6.96×10^{-9}	3.23
	51.2	6.51×10^{-9}	4.74	6.90×10^{-9}	4.73	7.02×10^{-10}	4.68	3.02×10^{-10}	4.53
	102.4	4.24×10^{-10}	3.94	4.47×10^{-10}	3.95	2.30×10^{-11}	4.93	1.27×10^{-11}	4.57

(b) Curvature error measures at $t_\sigma = 30$.

Table 11: Static column case: numerical results in function of R/h and order of convergence for the $CP_\perp$ method with 2 iterations, order 2 and order 4 κ_{LS} calculation, $La = 120$.

[1] A. Yarin, D. Weiss, Impact of drops on solid surfaces: self-similar capillary waves, and splashing as a new type of kinematic discontinuity, Journal of Fluid Mechanics 283 (1995) 141–173.

[2] D. Bhaga, M. Weber, Bubbles in viscous liquids: shapes, wakes and velocities, Journal of Fluid Mechanics 105 (1981) 61–85.

[3] R. Clip, J. Grace, M. Weber, Bubbles, Drops, and Particles .

[4] G. Tryggvason, B. Bunner, O. Ebrat, W. Tauber, Computations of multiphase flows by a finite difference/front tracking method. I. Multi-fluid flows, Lecture Series-von Karman Institute For Fluid Dynamics (1998) 7–7.

[5] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, Journal of Computational Physics 152 (2) (1999) 423–456.

[6] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational physics 114 (1) (1994) 146–159.

[7] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, Journal of Computational Physics 183 (1) (2002) 83–116.

[8] M. Sussman, E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, Journal of Computational Physics 162 (2) (2000) 301–337.

[9] G.-H. Cottet, J.-M. Etancelin, F. Pérignon, C. Picard, et al., High order Semi-Lagrangian particle methods for transport equations: numerical analysis and implementation issues, ESAIM: Mathematical Modelling and Numerical Analysis .

[10] J. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, Journal of computational physics 100 (2) (1992) 335–354.

[11] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, M. W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, Journal of Computational Physics 213 (1) (2006) 141–173.

[12] S. J. Cummins, M. M. Francois, D. B. Kothe, Estimating curvature from volume fractions, Computers & structures 83 (6) (2005) 425–434.

[13] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, Journal of Computational Physics 228 (16) (2009) 5838–5866.

[14] C. B. Macdonald, S. J. Ruuth, Level set equations on surfaces via the Closest Point Method, Journal of Scientific Computing 35 (2-3) (2008) 219–240.

[15] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, Journal of Computational Physics 227 (4) (2008) 2674–2706.

[16] Q. Zhang, On a family of unsplit advection algorithms for volume-of-fluid methods, SIAM Journal on Numerical Analysis 51 (5) (2013) 2822–2850.

[17] L. C. Evans, J. Spruck, et al., Motion of level sets by mean curvature I, J. Diff. Geom 33 (3) (1991) 635–681.

[18] A. Poux, S. Glockner, M. Azaïez, Improvements on open and traction boundary conditions for Navier–Stokes time-splitting methods, Journal of Computational Physics 230 (10) (2011) 4011–4027.

[19] P. Lubin, S. Glockner, Numerical simulations of three-dimensional breaking waves: aeration and turbulent structures, Journal of Fluid Mechanics Under review.

[20] K. Goda, A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows, Journal of Computational Physics 30 (1) (1979) 76–95.

[21] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, J. Koster, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM Journal on Matrix Analysis and Applications 23 (1) (2001) 15–41.

[22] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, Parallel computing 32 (2) (2006) 136–156.

[23] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of computational physics 126 (1) (1996) 202–228.

[24] G.-H. Cottet, E. Maitre, A level set method for fluid-structure interactions with immersed surfaces, Mathematical models and methods in applied sciences 16 (3) (2006) 415–438.

[25] S. J. Ruuth, B. Merriman, A simple embedding method for solving partial differential equations on surfaces, Journal of Computational Physics 227 (3) (2008) 1943–1961.

[26] F. Denner, D. R. van der Heul, G. T. Oud, M. M. Villar, A. da Silveira Neto, B. G. van Wachem, Comparative study of mass-conserving interface capturing frameworks for two-phase flows with surface tension, International Journal of Multiphase Flow 61 (0) (2014) 37 – 47, ISSN 0301-9322.

[27] S. Zahedi, M. Kronbichler, G. Kreiss, Spurious currents in finite element based level set methods for two-phase flow, International Journal for Numerical Methods in Fluids 69 (9) (2012) 1433–1456.

[28] D. Fuster, G. Agbaglah, C. Josserand, S. Popinet, S. Zaleski, Numerical simulation of droplets, bubbles and waves: state of the art, Fluid dynamics research 41 (6) (2009) 065001.

[29] C. Galusinski, P. Vigneaux, On stability condition for bifluid flows with surface tension: Application to microfluidics, Journal of Computational Physics 227 (12) (2008) 6140–6164.

[30] M. Owkes, O. Desjardins, A mesh-decoupled height function method for computing interface curvature, Journal of Computational Physics .

[31] S. Shin, S. Abdel-Khalik, V. Daru, D. Juric, Accurate representation of surface tension using the level contour reconstruction method, Journal of Computational Physics 203 (2) (2005) 493–516.

[32] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, International Journal for Numerical Methods in Fluids 30 (6) (1999) 775–793.

[33] S. H. Lamb, Hydrodynamics, Dover Books, 1945.

[34] S. V. Shepel, B. L. Smith, On surface tension modelling using the level set method, International journal for numerical methods in fluids 59 (2) (2009) 147–171.

[35] D. Torres, J. Brackbill, The point-set method: front-tracking without connectivity, Journal of Computational Physics 165 (2) (2000) 620–644.

[36] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, International Journal for Numerical Methods in Fluids 60 (11) (2009) 1259–1288.