Accuracy and convergence of the curvature and normal vector discretizations for 3D static and dynamic front-tracking interfaces

Désir-André Koffi Bi, Mathilde Tavares, Eric Chénier, Stéphane Vincent

Accuracy and convergence of the curvature and normal vector discretizations for 3D static and dynamic front-tracking interfaces

Désir-André Koffi Bi[a,1,*], Mathilde Tavares[a], Eric Chénier[a], Stéphane Vincent[a]

[a]*MSME, Univ Gustave Eiffel, CNRS UMR 8208, Univ Paris Est Creteil, F-77474 Marne-la-Vallée, France*

Abstract

In this work, three classes of numerical methods are investigated to evaluate the mean curvature, the unit normal vector and the surface tension on a front tracking interface encountered in the simulation of multiphase flows with separated phases. The Laplace-Beltrami Operator discretization, the Integral Formulation of the surface tension and the Surface Reconstruction technique are well-known methods used in the literature, but whose accuracy, robustness and convergence properties are seldom studied. In a first step, different variants of these methods are presented and compared against each other on a static analytical surface to measure their sensitivity to the size and regularity of the mesh. Then, to assess the influence of the surface advection scheme and the remeshing procedures, two original and time dependent analytical surfaces have been developed, leading to a ligament formation or the birth of a drop/bubble on a flat surface/liquid film. Comparisons to such dynamical surfaces are especially useful and significant, since they highlight the sensitivity of numerical methods for the interface property calculation to the errors produced by the Lagrangian transport of the front-tracking surface. Finally, the accuracy of the different approximations are correlated to their computation time, so that the user may choose the most appropriate method according to the desired accuracy and cost.

Keywords: Front-Tracking, multiphase flow, surface tension, curvature

*Corresponding author

Email addresses: desir-andre.koffi-bi@univ-eiffel.fr (Désir-André Koffi Bi), mathilde.tavares@univ-eiffel.fr (Mathilde Tavares), eric.chenier@univ-eiffel.fr (Eric Chénier), stephane.vincent@univ-eiffel.fr (Stéphane Vincent)

1. Introduction

Multiphase flows are very common in industry and natural environment. Amongst the numerous examples, it can be mentioned the heat and mass transfer for safety issues in nuclear power plants or for the control of the chemical composition in alloy manufacturing, or also the liquid atomization, either of fuel in engines to reduce pollutants generated by combustion (Canu et al., 2018), or of sea water resulting from the breaking wave to understand the coupling between the ocean and the atmosphere for meteorology purpose. For these flows, the heat and mass transfers result from the energy and momentum balances in each phase, but also from the physics at the interface which is governed by the jump conditions (Delhaye, 1974)

$$[\rho(\mathbf{u} - \mathbf{V}_i)] \cdot \mathbf{n} = \dot{m} \tag{1}$$

$$\left[-\dot{m}\mathbf{u} - p\mathbf{n} + \mu\mathbf{n} \cdot \left(\nabla\mathbf{u} + (\nabla\mathbf{u})^t\right)\right] = \sigma\kappa\mathbf{n} + (\nabla\sigma) \tag{2}$$

$$[\lambda \cdot \nabla T] \cdot \mathbf{n} = \dot{m} L_f \tag{3}$$

where $[\bullet]$ stands for the difference of the quantity $\bullet$ on both sides of the interface, in the direction of the unit normal vector to the interface $\mathbf{n}$, pointing outward. It is worth noticing that the forces and fluxes acting at the interface depend on the one hand, to fluid flow parameters, such as the fluid velocities $\mathbf{u}$, the velocity of the interface $\mathbf{V}_i$, the pressures p, the temperatures T, the densities ρ, the kinematic viscosities μ, the surface tension σ and the latent heat L_f, on the other hand to geometrical properties such as twice the mean curvature κ and the unit normal vector $\mathbf{n}$ to the interface.

Since the shape of the interfaces, allowing the calculation of κ and $\mathbf{n}$, is one of the very important parameters to control the heat and mass transfers, the numerical models must be able to approximate it accurately. However, such a task can turn out to be very difficult, especially when a large range of interfacial scales have to be modelled. For example, in the

atomization process, the cascade of interface scales must be solved, which requires a very expensive mesh refinement: the initial breakup of the continuous liquid phase gives rise to individual liquid parcels before producing a dispersed spray. Even though the establishment of the drop size distribution was successfully described by macro- or meso-scale models with ELSA or CEDRE solvers (Demoulin et al., 2007; Zuzio et al., 2013), these descriptions are only valid once the spray is formed. Therefore, the primary atomization is still poorly understood and needs new modelling strategies. Computing the curvature and more generally the surface properties also arise in boundary integral methods where volumetric problems are transformed into surface problems (De-hao Yu, 2002).

The understanding and the control of the heat and mass transfer in multiphase flows also requires that we distinguish the different interface shapes. Thanks to the databases stemming from CLS-VOF schemes (Vaudor et al., 2017), Massot and co-workers (Canu et al., 2018; Essadki et al., 2019) have been able to propose new ways to analyse the interface dynamics during primary atomization. They have demonstrated that each type of structure (spheres, ligaments or sheets) is characterized by specific distributions of the interface curvature. Therefore, an accurate numerical evaluation of the interface curvature is essential to ensure the reliability of the results.

In our contribution, we are interested in the numerical evaluation of the topological properties of interfaces, namely the mean curvature κ and the unit normal vector $\mathbf{n}$, but also the surface tension $\mathbf{f}$. Depending on the model bieng used, the capillary effects are either introduced in the Navier-Stokes equations (Shin and Juric, 2002), or taken into account through the jump conditions (Eqs. 1-3) by the geometrical components κ and $\mathbf{n}$, or the surface force $\mathbf{f}$. In the first approach, the surface tension contribution is modelled by a volume force depending on the volume fraction derivatives expressed over the fluid mesh (Brackbill, 1992). Another approach relies on the writing of the jump conditions directly onto the interface approximation. This latter method turns out to be more accurate because it does not a-priori suffer from a spreading of the capillary force over the fluid mesh: calculations are carried out precisely at the surface location. The ghost fluid method (Fedkiw

et al., 1999) and the front-tracking approach (Du et al., 2006) belong to these techniques. However, when the interface does not fit to the nodes where the conservation equations for velocity, temperature and pressure are solved, the local capillary information must be distributed on the surrounding fluid nodes, which produces a smoothing effect of the surface tension, again (Fedkiw et al., 1999; Shin and Juric, 2002).

The present work is focused on the numerical evaluation of the mean curvature κ, the unit normal vector $\mathbf{n}$ and the surface tension $\mathbf{f}$, in the framework of a discrete interface made of a triangle tessellation. Coming either from the computational fluid dynamics domain, for example in the front-tracking method framework, or from the computer aided design, only few studies have dealt with their convergence properties and robustness (Li et al., 2015). Moreover, no comparison between a large panel of numerical methods exists in the literature. Lastly, the only available validations concern static surfaces where the nodes of the discrete surface lie exactly on the reference surface.

In our contribution, three large families of methods are studied to get the numerical approximations. In the framework of the computational fluid dynamics, we have developed a original time dependent analytical surface, that qualitatively gives rise to characteristic patterns met in multiphase flows, namely the formation of ligaments or the birth of a drop/bubble on a flat surface/liquid film. These new test cases have allowed us, for the first time, to examine the effect of the advection scheme, with remeshing procedures, onto the accuracy of κ, $\mathbf{n}$ and $\mathbf{f}$ approximations.

The paper is organized as follows. The different numerical approximations of κ, $\mathbf{n}$ and $\mathbf{f}$, which are stemmed from the three families of methods, are first briefly reminded (Sec. 2), and then compared against each other for a complex static analytical surface (Sec. 3). Section 4 is then devoted to comparisons for the time dependent surfaces, first for an exact advection but with a remeshing procedure, and then using the numerical transport scheme. Conclusions and perspectives are finally drawn in the last section.

2. Numerical methods for evaluating the curvature κ, the unit normal vector $\mathbf{n}$ and the capillary force $\mathbf{f}$

The aim of this section is to define the numerical methods used in this paper to approximate the local curvature κ, the unit normal vector $\mathbf{n}$ and the capillary force $\mathbf{f}$. These evaluations rely on a discrete representation of the interface which consists of a set of planar triangular elements, defined by their vertices, and the knowledge of the neighbours through the edges of each triangle. This kind of representation is widely used in the framework of front tracking methods.

Three main families of approximations are identified in the literature. They are based either on

- the discretization of the Laplace-Beltrami Operator,
- an Integral Formulation to get the surface tension over a specific area,
- a reconstruction method of the interface by an analytical surface.

The different methods devoted to the calculation of the interface properties, and which belong to the three families presented here above, are detailed in the following parts. But first, some notations are introduced.

2.1. Main mathematical notations

The discrete approximation of the interface is made up of a family $\mathcal{T}$ of triangles. Let $\mathcal{X}$ be the set of vertices of the whole triangles. The vertices of any $T \in \mathcal{T}$ are simply noted $\delta^2 T \subset \mathcal{X}$. So, we refer to

- the coordinate of any vertex $x \in \mathcal{X}$ by $\mathbf{x}$.
- the surface of any triangle $T \in \mathcal{T}$ by S_T.

We also introduce

- the set whose elements are couples of vertices $\mathcal{X}_2 = \{(x_\alpha, x_\beta) \in \mathcal{X}^2 \mid \exists\ T \in \mathcal{T},\ \{x_\alpha, x_\beta\} \subset \delta^2 T\}$.

- the first and second neighbourhood of any vertex $x \in \mathcal{X}$ by $\mathcal{N}_1(x)$ and $\mathcal{N}_2(x)$, with

$$\mathcal{N}_1(x) = \{x' \in \mathcal{X} \mid (x, x') \in \mathcal{X}_2\} \tag{4}$$

$$\mathcal{N}_2(x) = \{x' \in \mathcal{X} \setminus \mathcal{N}_1(x) \mid \exists\ x'' \in \mathcal{N}_1(x),\ (x', x'') \in \mathcal{X}_2\} \cup \mathcal{N}_1(x) \tag{5}$$

To express the numerical approximations of the curvature, normal vector and capillary force, the endpoints of the triangle edges have to be ordered. This order is closely linked to the notion of the exterior normal vector, noted $\mathbf{n}$. For a closed surface, it is easily defined. As for an open one, a side of the surface is arbitrary chosen as facing "exterior", and $\mathbf{n}$ is set pointing towards this region. Then, we define

- the orientation of any triangle $T \in \mathcal{T}$ by $\mathcal{X}_2(T) \subset \mathcal{X}$

$$\mathcal{X}_2(T) = \{(x_\alpha, x_\beta), (x_\beta, x_\gamma), (x_\gamma, x_\alpha) \mid \delta^2 T = \{x_\alpha, x_\beta, x_\gamma\},\ ((\mathbf{x}_\beta - \mathbf{x}_\alpha) \times (\mathbf{x}_\gamma - \mathbf{x}_\alpha)) \cdot \mathbf{n} > 0\} \tag{6}$$

  with $\times$ the cross product.

- the unit normal vector to any triangle $T \in \mathcal{T}$ by $\mathbf{n}_T$. So, $\forall\ T \in \mathcal{T}$, with $\delta^2 T = \{x_\alpha, x_\beta, x_\gamma\}$ and $\mathcal{X}_2(T) = \{(x_\alpha, x_\beta), (x_\beta, x_\gamma), (x_\gamma, x_\alpha)\}$, then

$$\mathbf{n}_T = \frac{(\mathbf{x}_\beta - \mathbf{x}_\alpha) \times (\mathbf{x}_\gamma - \mathbf{x}_\alpha)}{\|(\mathbf{x}_\beta - \mathbf{x}_\alpha) \times (\mathbf{x}_\gamma - \mathbf{x}_\alpha)\|} \tag{7}$$

- an approximation of the unit normal vector to the surface at any vertex $x \in \mathcal{X}$ by $\mathbf{n}_{\mathcal{N}_1}(x)$:

$$\mathbf{n}_{\mathcal{N}_1}(x) = \frac{\sum_{T \in \mathcal{T} \mid x \in \delta^2 T} S_T \mathbf{n}_T}{\left\| \sum_{T \in \mathcal{T} \mid x \in \delta^2 T} S_T \mathbf{n}_T \right\|} \tag{8}$$

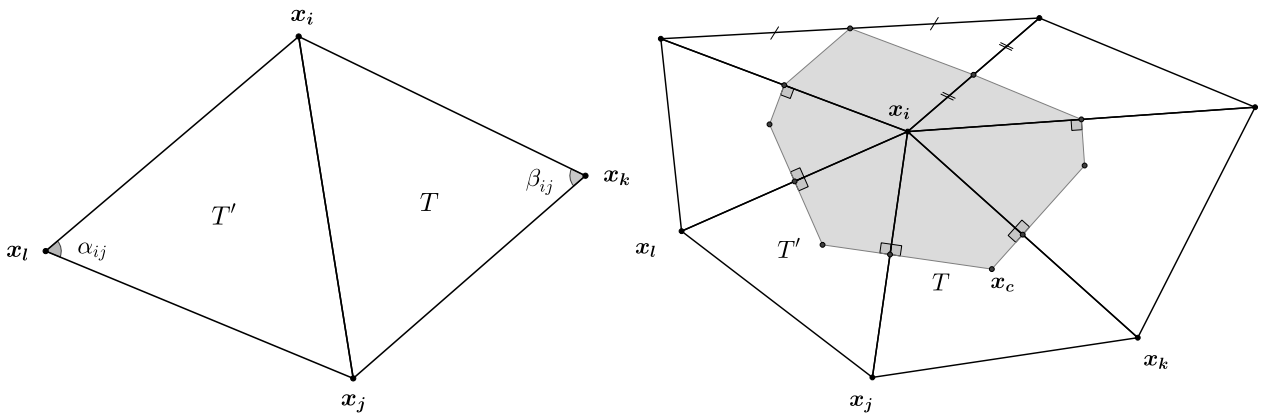
  This expression is simply proportional to the linear weighted combination by the triangle surface S_T of the normal vectors $\mathbf{n}_T$ of all the triangles T sharing the vertex $x \in \mathcal{X}$.

2.2. Discretization of the Laplace-Beltrami Operator (LBO-methods)

The Laplace-Beltrami Operator (LBO) is the generalization of Laplacian operator from Euclidean spaces to curved manifolds. Commonly used in the domain of computer aided design, it allows to get the geometrical properties of the surfaces, but also to smooth them. Different approximations of the LBO exist in the literature (see Desbrun et al. (1999); Meyer et al. (2003); Polthier et al. (2002); Taubin (1995) to cite but a few). In this paper, the retained method is based on that by Meyer et al. (2003). The approximation of the mean curvature normal vector $\mathbf{K}$ is:

$$\mathbf{K}(x_i) = \frac{1}{2\mathcal{A}_{mixed}} \sum_{(T,T')\in\mathcal{T}_i^\star} (\cot\alpha_{ij} + \cot\beta_{ij})(\mathbf{x}_j - \mathbf{x}_i) = 2\kappa_i\mathbf{n}_i \tag{9}$$

with $\mathcal{T}_i^\star = \{T \in \mathcal{T}, T' \in \mathcal{T} | \delta^2 T = \{x_i, x_l, x_j\}, \delta^2 T' = \{x_i, x_j, x_k\}\}$ and $\alpha_{ij} = \widehat{x_i x_l x_j}$ and $\beta ij = \widehat{x_j x_k x_i}$ (Fig. 1a), κ_i and $\mathbf{n}_i$ are the approximations of the mean curvature and unit



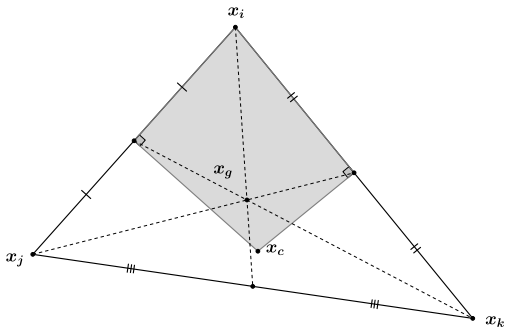
(a) Angles α_{ij} and β_{ij}

(b) Stencil for the mixed area $\mathcal{A}_{mixed}$ (LBO-Meyer method)

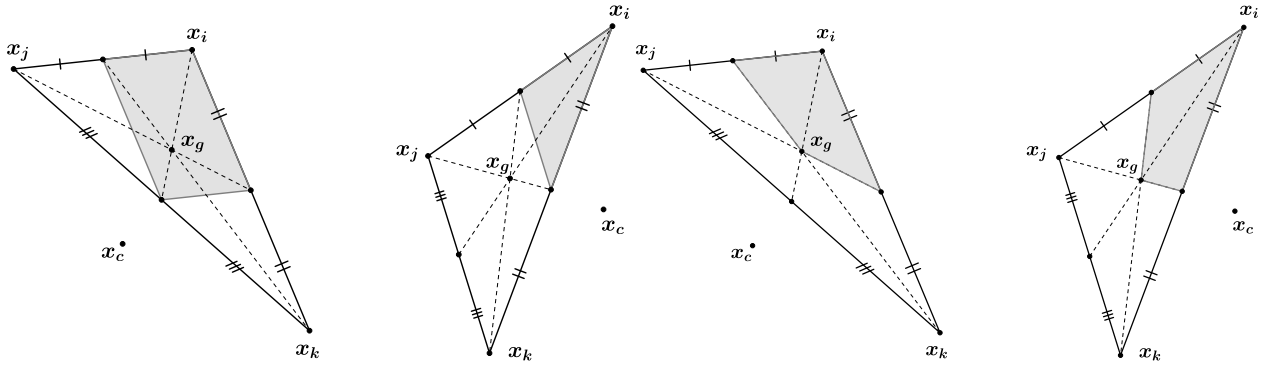
Figure 1: Notations and stencil for the discrete LBO methods (Essadki, 2018; Meyer et al., 2003)

normal vector to the surface at x_i, $\mathcal{A}_{mixed}$ is the measure of the mixed area (Fig. 1b). The shape of the mixed area $\mathcal{A}_{mixed}$ depends on the topology of the triangles sharing the vertex x_i.

LBO-Meyer method (Meyer et al., 2003) In the original method, when the angles of the element are acute, its contribution to $\mathcal{A}_{mixed}$ is the Voronoi area (Fig. 2a).



(a) Area for an acute triangle (LBO methods) Essadki (2018); Meyer et al. (2003)



(b) Areas for obtuse triangles (LBO-Meyer method) Meyer et al. (2003)

(c) Areas for obtuse triangles (LBO-Essad. method) Essadki (2018)

Figure 2: Construction of the mixed area $\mathcal{A}_{mixed}$ of the discrete LBO, according to Meyer et al. (2003) and Essadki (2018)

It corresponds to the area of the quadrangular element surface defined by the vertex x_i, the middle of the two edges adjacent to x_i and the circumcenter x_c of the triangle. For obtuse triangles, the circumcenter is outside the triangle and two cases must be discriminated (Fig. 2b, left). If $\widehat{x_j x_i x_k} > \pi/2$, the circumcenter is replaced by the middle of the opposite edge. Otherwise, the area is that of the triangle defined by the vertex x_i and each middle of the two edges adjacent to x_i (Fig. 2b, right).

LBO-Essad. method (Essadki, 2018) An alternative approach of the mixed area $\mathcal{A}_{mixed}$ can be used. Whereas the formalism does not change for acute triangles (Fig. 2a), for an obtuse angle, the local contribution is based on the Voronoi area for which the circumcenter is replaced by the barycenter x_g (Fig. 2c).

The approximated surface tension, at vertex x_i, is simply expressed by $\mathbf{f}_i = \sigma \mathbf{K}(x_i)$ and

the approximated local curvature and normal vector to surface S, at node x_i, write:

$$\mathbf{n}_i = \text{sign}(\mathbf{f}_i \cdot \mathbf{n}_{\mathcal{N}_1(x_i)}) \frac{\mathbf{f}_i}{\|\mathbf{f}_i\|} \quad , \qquad \kappa_i = \frac{1}{2\sigma} \text{sign}(\mathbf{f}_i \cdot \mathbf{n}_{\mathcal{N}_1(x_i)}) \|\mathbf{f}_i\| \tag{10}$$

2.3. Integral Formulation of the surface tension (IF-methods)

In the Integral Formulation family, the approximation of an average surface tension is directly computed. Initially proposed in the front-tracking framework by Popinet and Zaleski (1999) for 2D cases, this method has been later on extended to 3D cases by Tryggvason et al. (2001). Assuming a constant surface tension σ, the average surface tension over S, $\overline{\mathbf{f}}$, is the circulation of the "pulls" exerted along the border ∂S:

$$\overline{\mathbf{f}} = \frac{\sigma}{S} \int_S \kappa \mathbf{n} dS = \frac{\sigma}{S} \oint_{\partial S} \mathbf{t} \times \mathbf{n} dl \tag{11}$$

where $\mathbf{t}$ is the unit tangent vector to ∂S and $\mathbf{n}$ the unit normal vector to S.

IF-Trygg. method (Tryggvason et al., 2011) There is an earlier version of this method proposed by (Tryggvason et al., 2001) where a quadratic fitting is used to evaluate $\mathbf{t}$ and $\mathbf{n}$ in the integral formulation. However we choice the recent one as it is the most used. The discrete surface is centred around the vertex x_i and relies on a set of adjacent triangles (Fig. 3).

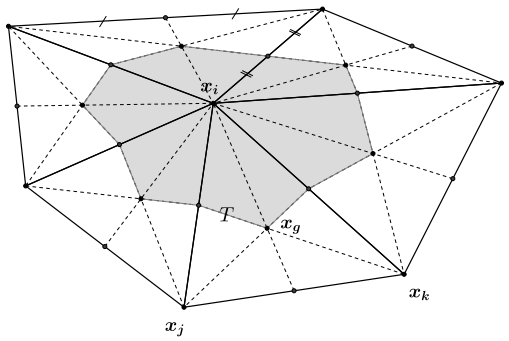


Figure 3: Stencil for the original (Tryggvason et al., 2011) method of the Integral Formulation (IF-Trygg. method)

The surface S is a collection of quadrangular polygons, defined piecewise on each triangle, whose one of the vertices are x_i, the middle of the two edges adjacent to x_i and the barycentre x_g of the triangle. After simplification, it is possible to remove the edge nodes and the barycentre, to get a relation based on the mesh vertices belonging to $\mathcal{N}_1(x)$:

$$\mathbf{f}_i = \frac{\sigma}{2S} \sum_{T \in \mathcal{T}_i^\star} (\mathbf{x}_k - \mathbf{x}_j) \times \mathbf{n}_T \tag{12}$$

with $\mathcal{T}_i^\star = \{T \in \mathcal{T} | \delta^2 T = \{x_i, x_j, x_k\}, (x_j, x_k) \in \mathcal{X}_2(T)\}$, $S = \sum_{T \in \mathcal{T}_i^\star} S_T$, and $\mathbf{n}_T$ the unit normal vector to the triangle T.

IF-triang. method The main difference with the previous method dwells in the shape of the surface. In the triangular version of the Integral Formulation, S covers the whole triangles having x_i as a vertex (Fig. 4).

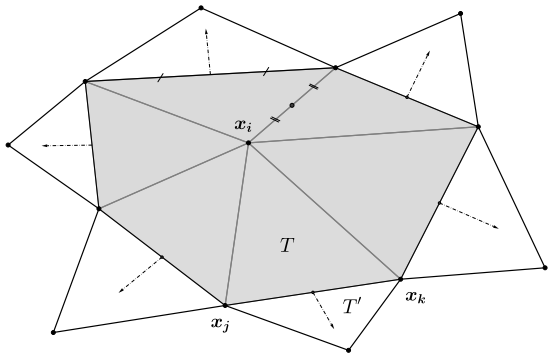


Figure 4: Stencil for the 'triangle version' of the Integral Formulation (IF-triang. method)

The average surface tension is then expressed by

$$\mathbf{f}_i = \frac{\sigma}{S} \sum_{(T,T') \in \mathcal{T}_i^\star} (\mathbf{x}_k - \mathbf{x}_j) \times \frac{S_T \mathbf{n}_{T'} + S_{T'} \mathbf{n}_T}{\| S_T \mathbf{n}_{T'} + S_{T'} \mathbf{n}_T \|} \tag{13}$$

with $\mathcal{T}_i^\star = \{T \in \mathcal{T}, T' \in \mathcal{T} | x_i \in \delta^2 T, x_i \notin \delta^2 T', \{x_j, x_k\} = \delta^2 T \cap \delta^2 T', (x_j, x_k) \in \mathcal{X}_2(T)\}$, S_T and $S_{T'}$ respectively the surfaces of the triangles T and T', $\mathbf{n}_T$ and $\mathbf{n}_{T'}$ their respective unit normal vectors and $S = \sum_{T \in \mathcal{T}_i^\star} S_T$.

Once the surface tension is evaluated, the approximated local curvature and normal vector to the surface S at node x_i are expressed in the same way as for the LBO-methods (see Eq. (10)).

2.4. Surface Reconstruction (SR-methods)

Knowing the analytic equation of a surface enables the exact calculation of its geometrical properties, specifically the mean curvature and the unit normal vector (Goldman, 2005). This equation can be constructed over the entire surface, like what was proposed by Popinet and Zaleski (1999) and Trontin et al. (2012) to approximate 1D interfaces with cubic splines, or defined locally to approximate a part of the interface mesh (Du et al., 2006; Zinchenko et al., 1997). The local approach leads obviously to simpler methods, especially for 2D surfaces and complex interface shapes. Whereas the curvature does not depend on the local coordinates, a transition matrix must be used to proceed to the change from the local to the global basis.

For the local reconstruction methods, the local basis originates at the vertex x_i where the interface properties have to be approximated. Several methods are distinguished in this paper depending on

- the expression of the approximate normal vector to the surface at x_i,
- the stencil employed to reconstruct the local analytical surface,
- the eventual requirement that the node x_i belongs to the reconstructed surface.

Once the approximate normal vector is defined, the two other vectors of the orthogonal basis can be chosen arbitrarily. In this paper, the reconstructed analytical surface has a polynomial expression and the calculation of the coefficients is the result of a least square method. The parameters of the Surface Reconstruction method are now detailed.

Stencil for the reconstructed surface – $\mathcal{N}_1(x_i)$ or $\mathcal{N}_2(x_i)$.

The influence of the stencil on the reconstructed surface is studied. The least square method is defined on mesh vertices that belong to, either the first neighbourhood

$\mathcal{N}_1(x_i)$, or the second neighbourhood $\mathcal{N}_2(x_i)$ of the vertex x_i. Furthermore, some precautions must be taken into account when using the first neighbourhood $\mathcal{N}_1(x_i)$. Indeed, if the number of vertices is not sufficient to use the least square method, a completion is made by choosing arbitrary vertices in the second neighbourhood (Du et al., 2006). In his Phd thesis, D.-A. Koffi Bi (2021) shows that when these additional vertices are aligned with the first neighbourhood, it deteriorates the evaluation of the curvature and normal vector based on the first neighbourhood. Therefore in this peculiar situation, the solution adopted in this work is to automatically switch to a reconstruction based on the second neighbourhood when needed.

Vertex x_i solution (or not) of the reconstructed surface – $a_{00} \neq 0$ or $a_{00} = 0$.

The local surface is approximated by the equation $Z = f(X, Y)$, with the polynomial function $f(X, Y)$ defined by:

$$f(X, Y) = a_{20}X^2 + a_{11}XY + a_{02}Y^2 + a_{10}X + a_{01}Y + a_{00} \tag{14}$$

where a_{ij} are unknown real coefficients to be determined, X, Y, Z are the local coordinates relative to the local basis. Depending on the method, the coefficient a_{00} will be set free or imposed to zero as proposed by Du et al. (2006); Zinchenko et al. (1997). In this latter case, the reconstructed surface passes exactly by vertex x_i.

Construction of the local basis $(\mathbf{e}_X, \mathbf{e}_Y, \mathbf{n}_\alpha)$, with $\alpha \in \{c, m, iter\}$

The local basis is constructed first by defining the local tangent plane $(\mathbf{e}_X, \mathbf{e}_Y)$, which is characterized by its normal vector. Different approaches exist to evaluate this normal vector.

- A first choice is a linear combination of all the unit normal vector $\mathbf{n}_T$ to triangle T, whose vertices belong to the stencil $\mathcal{N}_1(x_i)$:

$$\mathbf{n}_c(x_i) = \frac{\displaystyle\sum_{T\in\mathcal{T}|x_i\in\delta^2 T} S_T\mathbf{n}_T}{\left\|\displaystyle\sum_{T\in\mathcal{T}|x_i\in\delta^2 T} S_T\mathbf{n}_T\right\|} \tag{15}$$

It can be noted that $\mathbf{n_c}(x_i) \equiv \mathbf{n}_{\mathcal{N}_1}(x_i)$ defined in Eq. (8).

- In the works by Du et al. (2006), the authors utilize the formulation proposed by Max (1999):

$$\mathbf{n}_m(x_i) = \frac{\displaystyle\sum_{T\in\mathcal{T}_i^\star} \frac{(\mathbf{x}_j-\mathbf{x}_i)\times(\mathbf{x}_k-\mathbf{x}_i)}{\|\mathbf{x}_j-\mathbf{x}_i\|^2\|\mathbf{x}_k-\mathbf{x}_i\|^2}}{\left\|\displaystyle\sum_{T\in\mathcal{T}_i^\star} \frac{(\mathbf{x}_j-\mathbf{x}_i)\times(\mathbf{x}_k-\mathbf{x}_i)}{\|\mathbf{x}_j-\mathbf{x}_i\|^2\|\mathbf{x}_k-\mathbf{x}_i\|^2}\right\|} \tag{16}$$

  with $\mathcal{T}_i^\star = \{T \in \mathcal{T} | \delta^2 T = \{x_i, x_j, x_k\}, (x_j, x_k) \in \mathcal{X}_2(T)\}$

- Another procedure, proposed in the paper by Zinchenko et al. (1997), is based on an iterative calculation of the unit normal vector $\mathbf{n}_{\text{iter}}(x_i)$ in order that this vector corresponds exactly to the unit normal vector of the reconstructed interface at x_i.

To distinguish the different approaches used in the Surface Reconstruction methods, from now on, they will be called SR-$\mathcal{N}_k$-$\mathbf{n}_\alpha$-β, with $k \in \{1, 2\}$ for the stencil size, $\alpha \in \{c, m, \text{iter}\}$ for the unit normal vector defining the local tangent plane and $\beta \in \{0, a_{00}\}$ to enforce or not x_i to belong to the reconstructed surface. It is worth to point out that the SR-$\mathcal{N}_1$-n_m-0 and SR-$\mathcal{N}_1$-$\mathbf{n}_{\text{iter}}$-0 methods were respectively proposed by Du et al. (2006) and Zinchenko et al. (1997).

The calculations of the curvature κ and unit normal vector $\mathbf{n}$ are performed on the reconstructed analytic function $Z = f(X, Y)$. The surface tension at node x_i then reads:

$$\mathbf{f}_i = \sigma\kappa(x_i)\mathbf{n}(x_i) \tag{17}$$

Considering the formulae from differential geometry, the unit normal vector and the mean curvature write (Goldman, 2005):

$$\tilde{\mathbf{n}} = \frac{\nabla F}{\|\nabla F\|} \tag{18}$$

and

$$\tilde{\kappa} = -\nabla \cdot \tilde{\mathbf{n}} = \frac{\nabla F \cdot H(F) \cdot \nabla F^T - \|\nabla F\|^2 \mathrm{Trace}(H(F))}{2\|\nabla F\|^3} \tag{19}$$

where $H(F)$ is the Hessian matrix of function $F(X,Y,Z) = Z - f(X,Y)$, with components $H_{\alpha\beta}(F) = \dfrac{\partial^2 F}{\partial\alpha\partial\beta}$ and $(\alpha,\beta) \in \{X,Y,Z\}^2$.

2.5. Discrete surfaces and solution accuracy

The discrete surfaces are constructed into successive steps. First, a planar surface is covered by a set of equilateral triangles of edge size d. Let us note $\tilde{x}$ any vertex of this mesh, with coordinates $\tilde{\mathbf{x}} = (x, y, 0)$. Since the mesh of any interface, for example issued from a front tracking method, is often made up of more or less irregular triangles, a disturbed mesh is build from the regular one by using a mapping $\mathcal{P}(\tilde{x}) = \tilde{\tilde{x}}$ such that

$$\tilde{\tilde{\mathbf{x}}} = \tilde{\mathbf{x}} + r \times p \times d \times (\cos 2\pi\theta, \sin 2\pi\theta, 0) \tag{20}$$

In this expression, (r, θ) is a couple of random variables drawn from a $[0,1]$-uniform distribution and p is a prescribed number driving the magnitude of perturbations ($0 \leq p \leq 0.2$). Moreover for steady problems (static case), the vertices are then projected on the known analytical surface $z = h(x,y)$ to define the discrete counterpart of the continuous surface. It is worth pointing out that, even if $p = 0$, the discrete mesh of the surface is not any more made up of equilateral triangles, since the projection has distorted the shape of all the elements. For dynamics problems, the vertices are simply advected in time by the velocity field. When $p \neq 0$, the error on the curvature and normal vector are evaluated on 10 random draws (r, θ), both for the static and dynamic cases; the retained value is simply the average.

In order to quantify the accuracy of the numerical methods with respect to the exact solutions, two measures of the error are used. Let ϕ be either the scalar variable κ, the unit normal vector $\mathbf{n}$ or the surface tension $\mathbf{f}$. We denote by $\phi(x_i)$ the exact value of ϕ at node x_i, and ϕ_i its numerical approximation. Then, two relative errors are defined over the N vertices of the mesh by:

$$\mathrm{Err}_2^{\mathrm{rel}}(\phi) = \left(\frac{\sum_{j=1}^{N} (\phi_j - \phi(x_j)) \cdot (\phi_j - \phi(x_j))}{\sum_{j=1}^{N} \phi(x_j) \cdot \phi(x_j)} \right)^{1/2} \tag{21}$$

$$\mathrm{Err}_\infty^{\mathrm{rel}}(\phi) = \left(\frac{\max_{j \in [\![1,N]\!]} (\phi_j - \phi(x_j)) \cdot (\phi_j - \phi(x_j))}{\max_{j \in [\![1,N]\!]} \phi(x_j) \cdot \phi(x_j)} \right)^{1/2} \tag{22}$$

where '$\cdot$' denotes either the usual product or the scalar product if ϕ is a vector. Since the studied surfaces are open, the error calculations are performed only for triangles far enough from the boundaries. Notice that this condition depends on numerical methods because they may rely on different stencil.

The numerous methods presented in this paper are mainly compared through curves which present the relative errors (21) (22) as a function of the dimensionless initial mesh size defined as $d \times \max_{j \in [\![1,N]\!]}(\kappa(x_j))$, with $\kappa(x)$ the exact curvature at node x. This latter relation can be re-interpreted as a function of the mean curvature radius $1/\kappa(x_j)$: let us assume that $d \times \max_{j \in [\![1,N]\!]}(\kappa(x_j)) = 10^{-2}$, then the edges of the initial triangles are 100 times smaller than the minimum of mean radius of curvature, $\min_{j \in [\![1,N]\!]}(1/\kappa(x_j))$. To distinguish in the graphs the two norms (21) (22), the curves drawn using L_2 are continuous whereas those with L_∞-norm are discontinuous.

For the sake of brevity, the error curves for the surface tension will not be presented since their behaviour are similar to that of the curvature (same order of convergence and error level).

3. Analysis of a stationary surface

In this section, the static surface proposed by Li et al. (2015) is studied (Fig. 5):

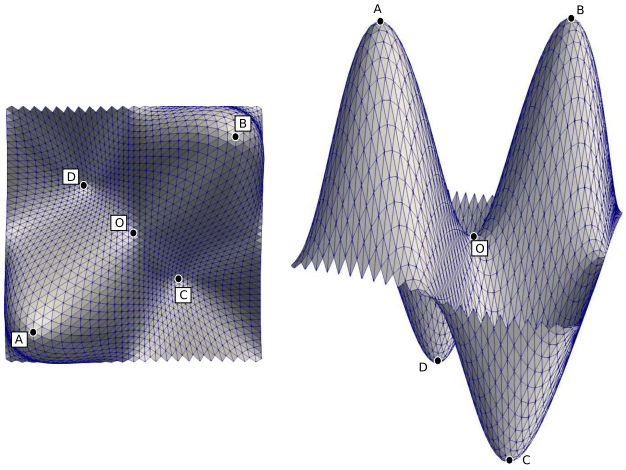


Figure 5: Top (left) and side (right) views of the discrete mesh projected onto the surface (23).

$$h(x,y) = \sin(5x)\sin(5y) \quad \text{with} \quad (x,y) \in [-\frac{\pi}{5},\frac{\pi}{5}] \times [-\frac{\pi}{5},\frac{\pi}{5}] \tag{23}$$

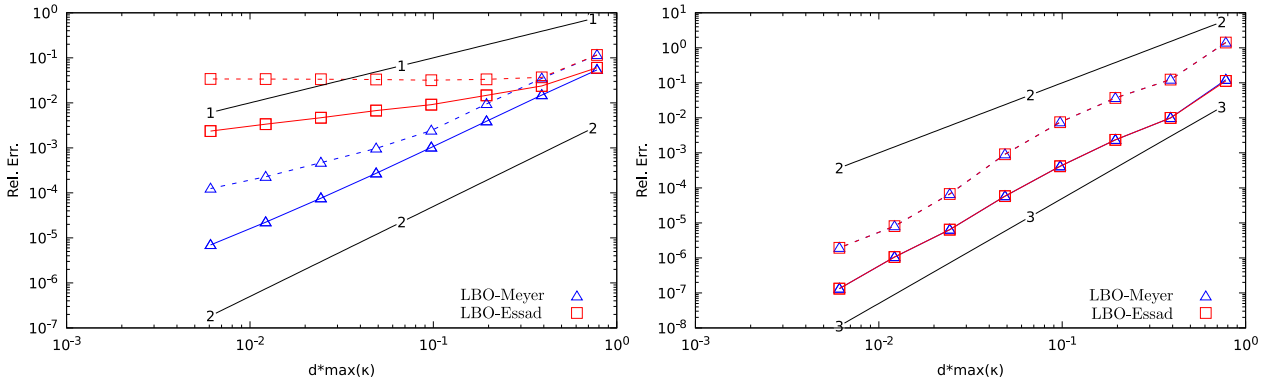
The main interest of this analytical surface is that it is characterized by a large variety of curvatures, sum of the two principal curvatures. Points A, B, C and D have maximal curvatures and point O exhibits a zero mean curvature with maximal but opposite principal curvatures (saddle-node point). To avoid boundary effects, the initial flat surface is defined on a larger domain, $(x,y) \in [-2,2] \times [-2,2]$ and all the vertices in the sub-domain $(x,y) \in [-\pi/5, \pi/5]^2$ are taken into account for the error evaluation (21) (22).

3.1. Results

3.1.1. LBO-methods

The Laplace-Beltrami Operator discretization, LBO-Essad. (Essadki, 2018) and LBO-Meyer (Meyer et al., 2003) methods, are compared in this section as a function of the mesh size, and for a regular mesh ($p = 0$) and a disturbed one ($p = 0.2$).

Without perturbation ($p = 0$), the LBO-Meyer method provides clearly the best results for the curvature (Fig. 6a): it is second order accurate in L_2-norm and tends to be first order



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

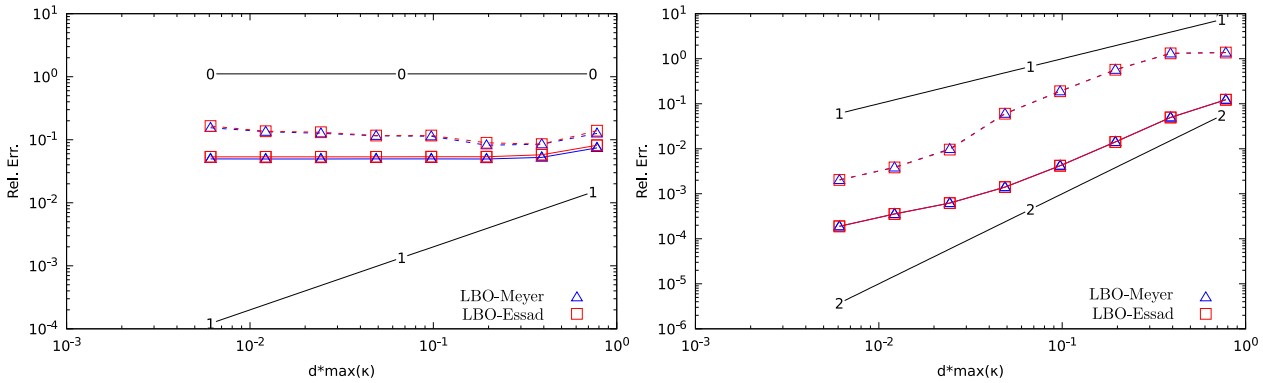
Figure 6: LBO methods - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 0\%$

in L_∞-norm. On the other hand for the LBO-Essad. method, the L_2-order of convergence is much smaller than 1 and the L_∞-norm saturates for $d \times \max_{j\in[\![1,N]\!]}(\kappa(x_j)) \leq 0.2$. This first comparison indicates that the discretization of the mixed area $\mathcal{A}_{mixed}$ for evaluating the discrete Laplace-Beltrami Operator is of paramount importance when the triangles are obtuse. Whereas the curvature calculation is very sensitive to the chosen discrete method, the evaluation of the unit normal vector is not (Fig. 6b). The L_2-order of convergence is nearly 3, like, in a lesser extend, the convergence rate in L_∞-norm.

With disturbances of 20% the initial edge size d, the shape of the triangles wander strongly away from the equilateral structure. Figure 7a shows that both methods saturate at the same level with the grid refinement when the curvature is considered. Despite this lack of convergence, the unit normal vector continues to converge, for both methods, but at the first order instead of the third order measured for regular meshes.

3.1.2. IF-methods

For the Integral Formulation methods, namely the IF-Trygg. (Tryggvason et al., 2011) and IF-triang. methods, the same analysis as the one proposed for the LBO-methods, is

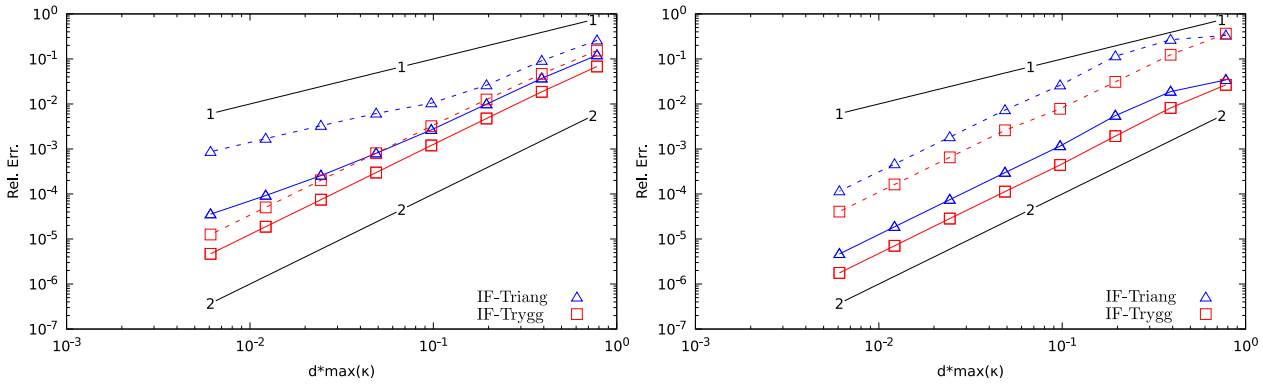


(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure 7: LBO methods - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 20\%$

performed for a regular mesh ($p = 0$) and a disturbed one ($p = 0.2$).

For regular meshes ($p = 0$), the numerical curvatures converge with the decrease in the dimensionless grid size $d \times \max_{j\in[\![1,N]\!]}(\kappa(x_j))$ (Fig. (8a)). The accuracy of the IF-Trygg.



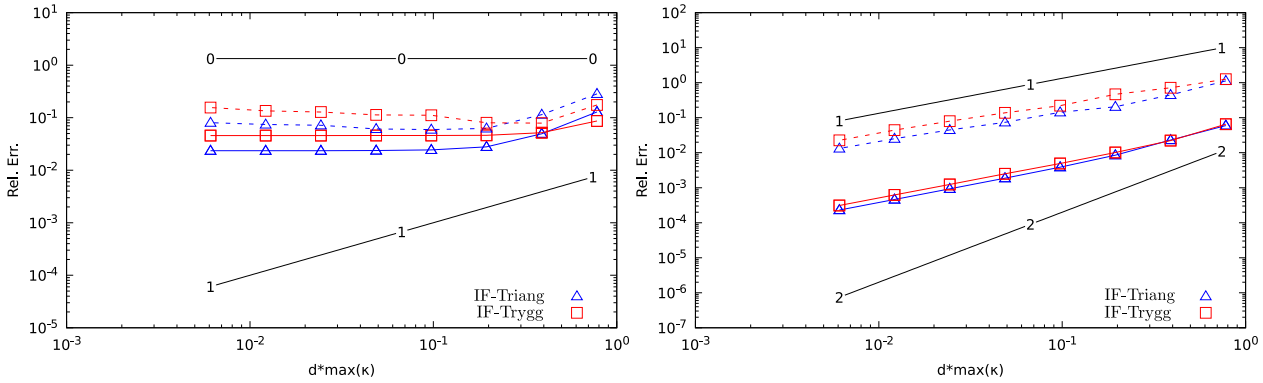
(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure 8: Integral Formulation - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 0\%$

method is second order for both norms. The results are worse for the IF-triang. method. Although the L_2-order of convergence is equal to 2 for coarse grids, it decreases to 1.5 asymptotically. The behaviour of the L_∞-norm, firstly second order, becomes first order for the fine meshes. Thus, for the finest grid the maximal relative error for the IF-triang.

method is larger than 2 decades to that of the IF-Trygg. one. The numerical approximation of the normal unit vector is slightly better for the IF-Trygg. method than for the IF-triang. method (Fig. (8b)). However, they converge both with a second order accuracy for the L_2 and L_∞-norms. This first set of tests seems to indicate that the IF-Trygg. method is more efficient than the IF-triang. method.

As for the LBO-methods, the curvature approximation is deeply worsen as soon as 20% disturbances are applied to the mesh (Fig. 9a). Again, the relative errors remains constant,



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure 9: Integral Formulation - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 20\%$

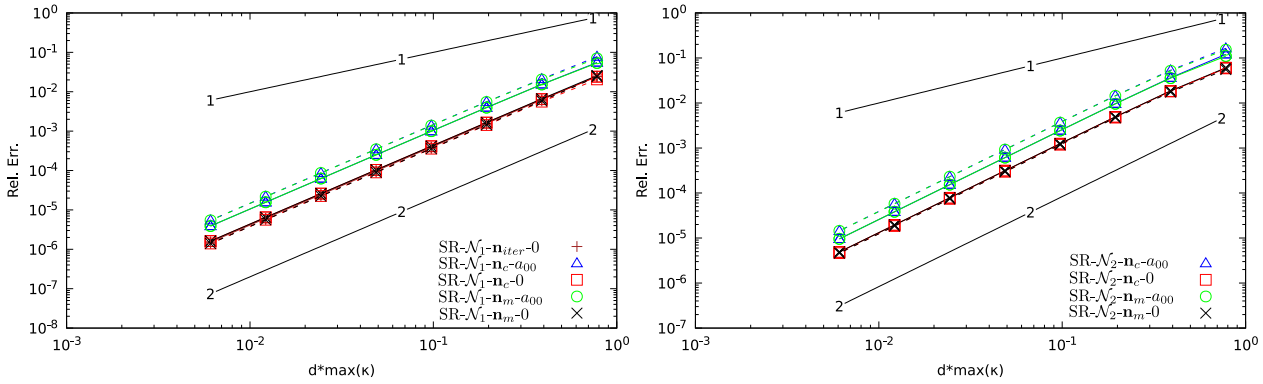
whatever the grid size is. In that case, the IF-triang. method seems to be slightly better, about twice smaller than the IF-Trygg. method. The unit normal vector approximations keep convergent behaviours, with a first order slope, whatever the norm and the method (Fig. 9b). As for the curvature, the IF-triang. method is better, but the difference is not really significant.

3.1.3. SR-methods

The last family of methods concerns the Surface Reconstruction techniques. Several approaches are compared in this section by using the notations presented in Sec. 2.4. Some are based on the first vertex neighbourhood, the SR-$\mathcal{N}_1$-$\mathbf{n}_{\mathrm{iter}}$-0 (Zinchenko et al., 1997), SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0 (Du et al., 2006), SR-$\mathcal{N}_1$-$\mathbf{n}_c$-0, SR-$\mathcal{N}_1$-$\mathbf{n}_m$-a_{00} and SR-$\mathcal{N}_1$-$\mathbf{n}_c$-a_{00} methods, and others

are expressed with the second neighbourhood, the SR-$\mathcal{N}_2$-$\mathbf{n}_m$-0, SR-$\mathcal{N}_2$-$\mathbf{n}_c$-0, SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} and SR-$\mathcal{N}_2$-$\mathbf{n}_c$-a_{00} methods. Following the same methodology as before, the accuracy of the approximations for the curvature and the unit normal vector are carried out as a function of the grid size, in a first step for a regular mesh, and then for a 20% disturbed one.

Figure 10 illustrates the convergence of the curvature κ_i, for a regular grid ($p = 0$), of the

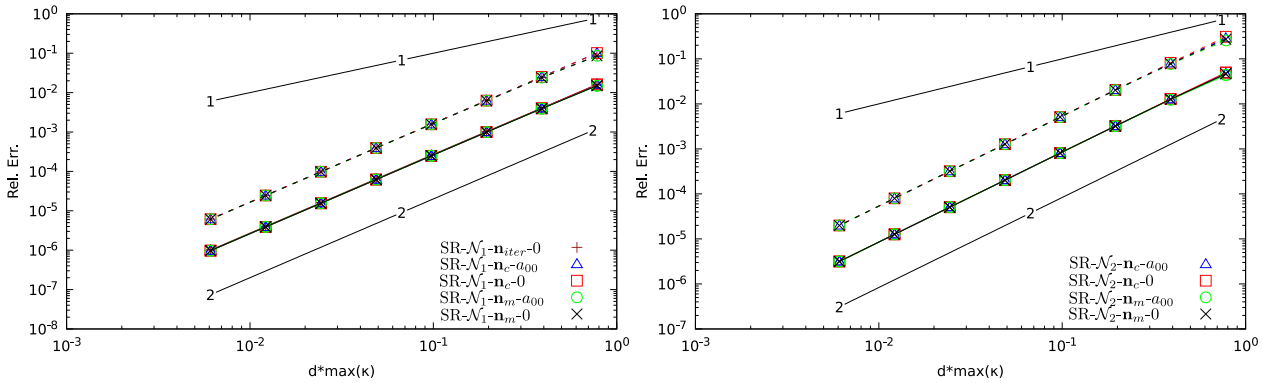


(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ with $\mathcal{N}_1(\mathbf{x_i})$ (b) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ with $\mathcal{N}_2(\mathbf{x_i})$

Figure 10: Surface Reconstruction - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ, $p = 0\%$

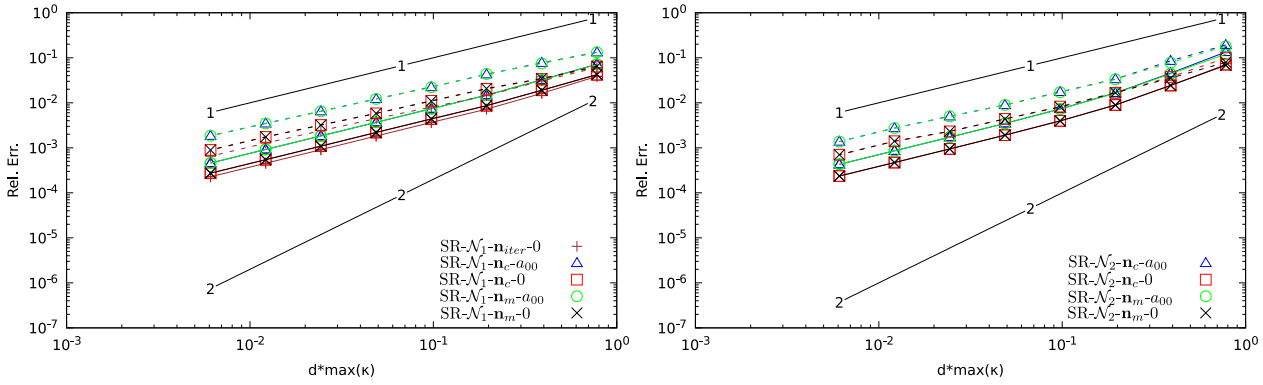
different methods based on the first neighbourhood $\mathcal{N}_1$ (Fig. 10a) and second neighbourhood $\mathcal{N}_2$ (Fig. 10b). Whatever the norm, the methods are second order accurate. However, the best results are obtained with methods that enforce the vertex x_i to belong to the surface (SR-$\mathcal{N}_k$-$\mathbf{n}_\alpha$-0, with $k \in \{1, 2\}$, and $\alpha \in \{c, m, \mathrm{iter}\}$). Likewise, the increase in the stencil from $\mathcal{N}_1$ to $\mathcal{N}_2$ for the reconstructed surface seems to cause a slight deterioration in the approximation. The same remarks apply when the relative errors on the normal vector are studied (Fig. 11).

When the triangular mesh is no longer regular ($p = 20\%$), the curvature convergence is maintained, but with a reduced convergence order equal to one (Fig. 12). Again, enforcing the vertex x_i to lay on the surface provides slightly better results. On the other hand, the relative errors on the normal vectors are not sensitive to the perturbation magnitude



(a) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$ with $\mathcal{N}_1(\mathbf{x_i})$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$ with $\mathcal{N}_2(\mathbf{x_i})$

Figure 11: Surface Reconstruction - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on $\mathbf{n}$, $p = 0\%$



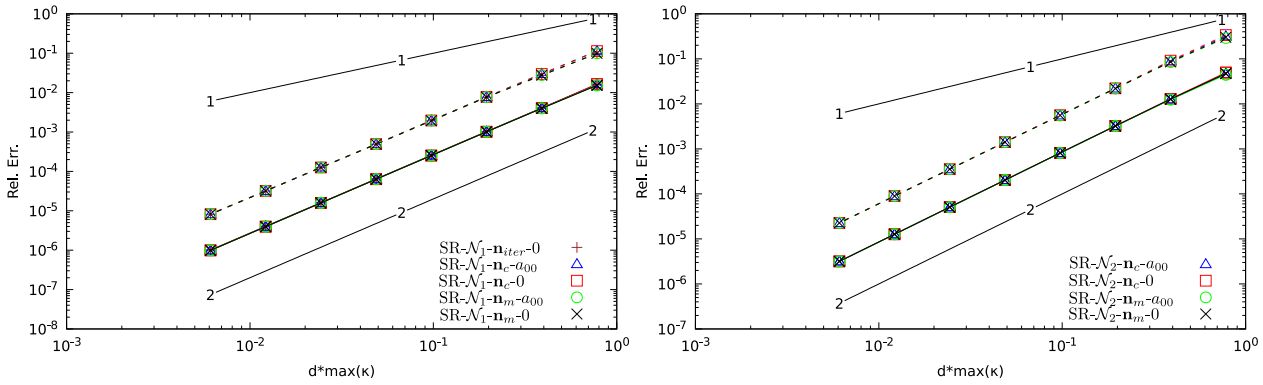
(a) $\mathrm{Err}_2^{\mathrm{rel}}(\kappa)$ with $\mathcal{N}_1(\mathbf{x_i})$ (b) $\mathrm{Err}_2^{\mathrm{rel}}(\kappa)$ with $\mathcal{N}_2(\mathbf{x_i})$

Figure 12: Surface Reconstruction - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ, $p = 20\%$

(Fig. 13). Indeed, both the error levels and convergence orders are preserved for disturbed meshes.

3.2. Comparisons

In this section, the best methods of each family of discretization for the curvature and normal vector are now compared. This comparison step is performed for $p = 20\%$ which leads to the largest errors, but also for $p = 0\%$. Indeed, one may use (or imagine) smoothing procedures to recover a nearly regular mesh which would be close to what is obtained for $p = 0\%$. Thus, for the Laplace-Beltrami-Operator discretization, only the LBO-Meyer



(a) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$ with $\mathcal{N}_1(\mathbf{x_i})$

(b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$ with $\mathcal{N}_2(\mathbf{x_i})$

Figure 13: Surface Reconstruction - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on $\mathbf{n}$, $p = 20\%$

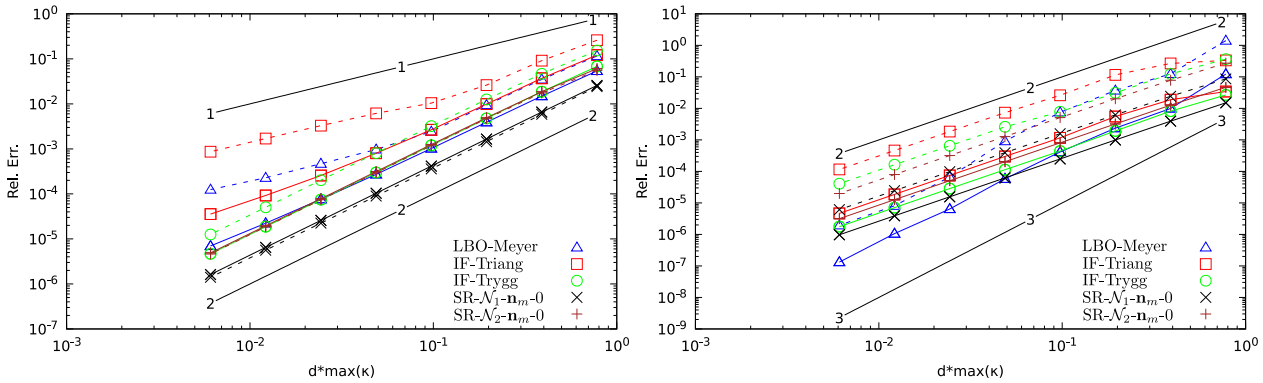
method (Meyer et al., 2003) is considered. Concerning the discrete Integral Formulation, both the IF-Trygg. (Tryggvason et al., 2011) and IF-Triang. are retained. For the last family of methods, the best ones are those satisfying that the vertex x_i, where the curvature and normal vector are calculated, belongs to the reconstructed surface. Thus, both the SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0 (Du et al., 2006) and SR-$\mathcal{N}_2$-$\mathbf{n}_m$-0 methods are kept, depending on the size of the neighbourhood for the calculation of the approximate surface.

3.2.1. Errors on curvature and normal vector

For the curvature calculation on a regular grid, although all the approaches exhibit convergence behaviours, the SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0 method provides the best results up to nearly two order smaller than those obtained with IF-Triang. when the L_2 norm is considered. This gap is increased up to three decades when the L_∞ norm is chosen (Fig. 14a).

The differences between the methods are much smaller when the analysis is focused on the unit normal vector (Fig. 14b). The most interesting result is probably the third order convergence of the LBO-Meyer method, which provides, on the one hand a quite bad approximation for coarse meshes and on the other hand best results for the finest meshes.

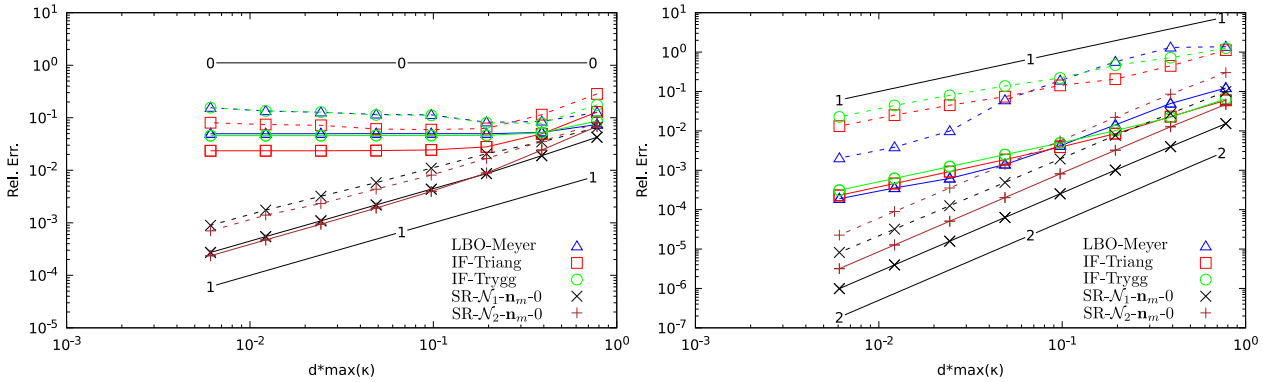
For the curvature calculation with 20%-magnitude perturbations, only the reconstruction methods SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0 and SR-$\mathcal{N}_2$-$\mathbf{n}_m$-0 keep going to converge with the mesh refinement, but



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure 14: Synthesis - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 0\%$

with a rate falling from second to first order for both norms (Fig. 15a) ; the other methods saturate. It is worth noticing that, even for small random perturbations, the quality of the



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure 15: Synthesis - Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 20\%$

curvature approximations is highly deteriorated (Fig. 16). Indeed, for a given dimensionless mesh size $d \times \max(\kappa) = 1.22 \times 10^{-2}$, adding tiny random disturbances of maximal magnitude $p = 5\%$ increases the error induced by Surface Reconstruction methods by at most one decade, nearly two decades for the IF-triang. method and three decades for LBO-Meyer and IF-Trygg. methods The use of irregular meshes for the normal evaluation reduces the order of convergence to one, except for the Surface Reconstruction methods which are quite insensitive to the perturbations (Fig. 15b).

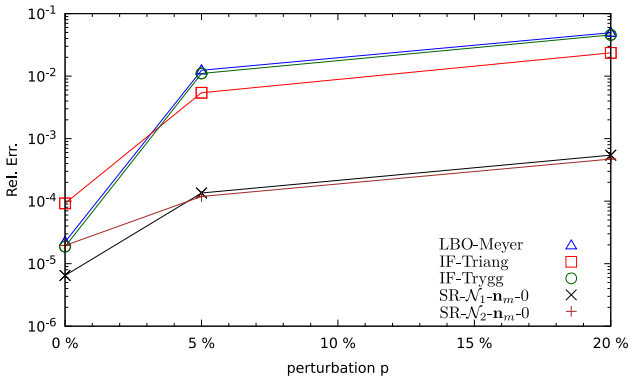


Figure 16: Variation of the relative errors as a function of the maximum amplitude p of the perturbations

A partial conclusion on the accuracy of the different numerical approximations can now be drawn. Contrary to the LBO and Integral Formulation methods, the Surface Reconstruction seems promising for the curvature calculation since it preserves convergence properties, even for disturbed meshes. Indeed, such meshes are very common: they can for example be met in front tracking methods for which the interface between two immiscible fluids are discretized by a set of triangles whose vertices are advected by the flow field. Although the accuracy of the normal vector approximation is better for the Surface Reconstruction methods, the LBO and Integral Formulation remains attractive, despite much larger errors, especially with the L_∞-norm.

3.2.2. CPU-cost and efficiency

Although the Surface Reconstruction methods provide very often the best approximations, their computational times are 3 times larger than that stemming from the LBO-Meyer approximation, and up to 10 times larger when compared to the IF-Trygg. method for the finest meshes. Thus, to perform pertinent comparisons, the accuracy of the different numerical methods must be studied in regards of the overall CPU time required to get them. The machine used is an Intel(R) Xeon (R) CPU E7-8890 v4 @ 2,20Ghz.

Whereas the Laplace Beltrami Operator and Integral Formulation methods compute the

surface tension, from which the curvature and the unit normal vector are directly deduced, the Surface Reconstruction method needs at first the evaluation of an analytical approximation of the discrete surface, then the independent calculation of the curvature and the normal vector to finally evaluate the surface tension. The expensive part of these algorithms lies in the surface tension calculation for the LBO and IF methods, and in the Surface Reconstruction for the SR method. Therefore, the computation times for the evaluations of the curvature, the unit normal vector, the surface tension or the three together are similar.

Figures 17 illustrates the relative errors committed on the curvature approximations as a function of the CPU time spent to evaluate κ_i, $\mathbf{n}_i$ and $\mathbf{f}_i$ over the whole vertices of the mesh. It must be noticed that the increase in the CPU time corresponds obviously to the decrease in the mesh size. For $p = 0\%$, the most efficient methods are IF-Trygg. and SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0 (Fig. 17a). Assuming a relative error of 1%, the aforementioned methods take about 0.02 s

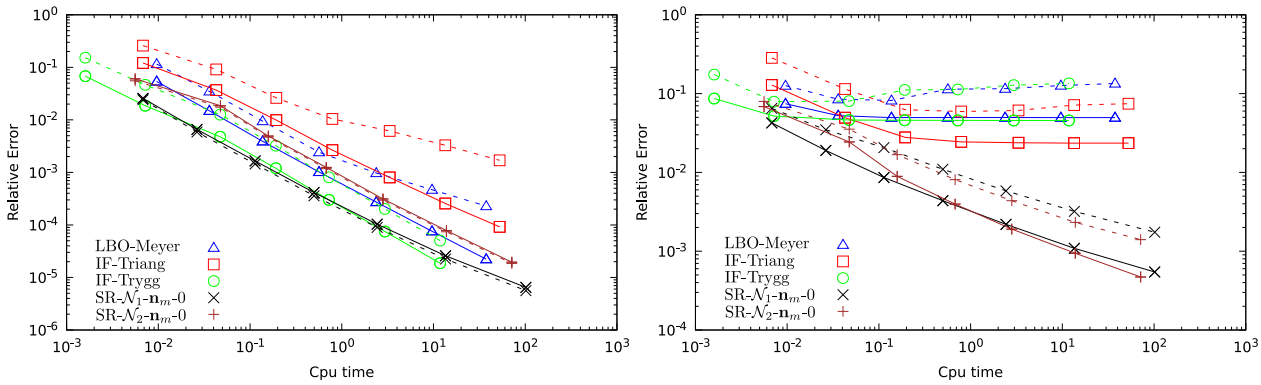


Figure 17: Relative errors on κ in function of the time in L_2 (continuous line) and L_∞ (dash line) norms, with (a) $p = 0\%$ and (b) $p = 20\%$.

and require a mesh size $d \times \max_i(\kappa(x_i)) = 0.3$ for IF-Trygg and 0.5 for SR-$\mathcal{N}_1$-$\mathbf{n}_m$. But for $p = 20\%$, this accuracy can only be reached by the Surface Reconstruction methods, for a CPU time about 5 times larger than with $p = 0\%$ and a grid size $d \times \max_i(\kappa(x_i)) = 0.2$ (Fig. 17b) .

3.3. Summary

For stationary surfaces, the evaluation of the curvature, which behaves as the surface tension, is the most sensitive step. Only the Surface Reconstruction methods prove to be convergent and accurate with the mesh refinement, whatever the regularity the surface mesh and the applied norm. The Laplace-Beltrami Operator and Integral Formulation methods are not robust regarding the mesh regularity. And even for regular grids where they provide good results, their efficiencies are never better than the Surface Reconstruction methods.

These tests for static surfaces give us a first glimpse of the accuracy and robustness of the different approaches to get the surface properties. The next section is focused on the coupling between the surface advection and the computation of the curvature and normal vector.

4. Analysis of advected surfaces

In the previous section, the calculation of the curvature and normal vector estimates are carried out for a fixed discrete surface: the resulting errors are then only governed by the mesh size, the regularity of the triangles and the accuracy of the approximation methods. Such an approach has already been used in literature (Li et al., 2015; Wardetzky, 2008; Xu, 2004). The only drawback of this static approach is that it does not take into account all the processes involved in the simulation of moving or deforming interfaces.

The aim of this part is to study, in what extend the advection of the interface mesh with all associated numerical procedures (advection, reseeding, swapping) may alter the results, both in the convergence process and in the accuracy.

4.1. Problem presentation

To address this problem, an analytical flow field acting on a moving interface, with deformations, is proposed in the first part of this section. Thereafter, the mechanisms used to advect and manage the interface mesh deformation are presented. These supplementary

processes must also be taken into account for the definition of the reference value, which forms the third part. Later on, the methodology of the numerical tests is presented.

4.1.1. Velocity field

In order to design a dynamic interface with deformation, a divergence-free flow field defined in spherical coordinate frame is first considered

$$\mathbf{v} = \frac{Q}{r^2}\mathbf{e_r}, \qquad r > 0 \tag{24}$$

with $4\pi Q$ the algebraic volume flow rate through the sphere of radius r. Starting from a discrete surface located at $z = z_0$, each vertex x_i, initially at coordinates $\mathbf{x}_i(t = 0) = (r(t = 0), \theta, \varphi)$, is advected in the radial direction according to equation:

$$r(t) = \sqrt[3]{3Qt + \frac{z_0^3}{\cos^3\theta}} \tag{25}$$

Depending on the sign of parameter Q, the origin of coordinates corresponds to a sink or a source. If $Q < 0$, the surface is stretched to the singularity, forming an elongated ligament characterized by two principal curvatures with opposite signs. On the contrary, for $Q > 0$, the surface inflates, producing an increasing bubble with a sharp lateral edge, similar to an impact on a liquid film. The interface topology in this configuration also looks like a droplet impinging a flat liquid film. These two cases are treated in separated subsections, as they clearly define quite different interface deformation dynamics, with curvature singularities of liquid finger type for the case $Q < 0$ whereas the other case provides an abrupt curvature evolution of liquid impact type in the most varying interface geometry zones.

4.1.2. Surface advection

The mesh of the initial surface is made up of equilateral triangles with an edge size d. The vertices of these elements are then randomly displaced with a maximal disturbance p, according to relation (20). Advection from $t = n\Delta t$ to $(n + 1)\Delta t$ of each vertex x_i of the mesh is the result of the numerical integration of equation $d\mathbf{x}_i/dt = \mathbf{V}_i$ by a second order Runge-Kutta scheme (Heun's method). The vertex velocity $\mathbf{V}_i$ is spatially interpolated by

the PERM method (McDermott and Pope, 2008) which relies on the knowledge of the exact velocity components (Eq. 24) on the staggered Eulerian grids. The fix Eulerian grid size is always twice the edge length d of the equilateral triangles, before applying any perturbation. The time step Δt is based on the Courant number equal to 0.5. Using the maximal instantaneous velocity magnitude over all the vertices x_i, namely $\max_i(Q/r_i^2) = Q(3Qt + z_0^3)^{-2/3}$, and the Eulerian grid size $2d$, the instantaneous time step writes $\Delta t = (3Qt + z_0^3)^{2/3} \times d/Q$.

During the transport, the edges of the triangles are stretched or compressed, leading eventually to undesirable elements. To overcome this issue and in order to keep triangle edges in the range $[d/2, 2d]$, several procedures are implemented (Roghair et al., 2016). A new vertex may be inserted at the middle of a triangle edge, if its size exceeds $2d$. Conversely, if the triangle edge is less than $d/2$, it is suppressed by merging its two vertices at the center of the former edge. At last, to improve the quality of the triangles, edges may be flipped between two flatten triangles sharing their longest edge. All these procedures, which aim to keep a good approximation of the real interface, are performed at each time step (see Appendix B for more details). Concerning the transport of the interface mesh, both the exact advection (the edge velocity is not interpolated but uses the analytical flow field Eq. 25) and the discrete advection scheme described before, as in real multiphase flow simulations, are performed. Indeed, considering an exact advection is interesting here in order to separate several numerical effects linked to interface management, advection or geometrical properties.

The error calculations for the moving surface are restricted to $\theta \leq \theta_0$, with $\theta_0 = \arctan(L_0/z_0)$, $L_0 = 1$, $z_0 = 1$ for $Q < 0$ and $z_0 = 0.01$ for $Q > 0$. The extension of the initial flat surface is large enough to avoid edge effects during curvature and normal vector computations.

4.1.3. Reference surface

In Sec. 3, the mesh vertices were exactly located on the stationary surface. For advected surfaces, the time integration, as well as the different procedure developed to improve the mesh quality, do not ensure that the vertices belong to the analytical surface any more after exact or discrete advection. Therefore, to perform comparisons and error measurements (21) (22), a reference surface must be defined; this issue is addressed below.

Two different cases are met, depending on the exact or approximative nature of the vertex transport.

For an exact transport, the advected vertices are located on the analytical surface (25). Due to the mesh improvement procedures, the modified or new nodes are projected on the reference surface in order that all the vertices keep an exact position as a function of time. Therefore, reference values are simply calculated with relations (18) and (19).

On the other hand, the advection of the vertices by a numerical scheme (Sec. 4.1.2) does not allow to stay on the analytical surface at any time. To compare the curvature and the normal vector to reference solutions, each vertex $x_i(t^{(n)})$ of the mesh must be associated to a point $x_i^{\text{ref}}(n\Delta t)$ located on the analytical surface $S^{\text{ref}}(n\Delta t)$. This point, defined as the intersection between this surface and the streamline stemming from the considered vertex $x_i(t^{(n)})$, is obtained as follows (Fig. 18). The vertex $x_i(t^{(n)})$ is first advected with the exact transport equation backward in time to give a point $x_i^{\text{ref}}(0)$ belonging to the initial flat surface at $z = z_0$. Since the exact and approximative trajectories differ, the coordinates $\mathbf{x}_i^{\text{ref}}(0)$ and $\mathbf{x}_i(t^{(0)})$ (assuming the node x_i existed at $t = 0$ in the mesh) are different from each other. Therefore, the time required to move the vertex x_i from $\mathbf{x}_i(t^{(n)})$ to $\mathbf{x}_i^{\text{ref}}(0)$, is not $n\Delta t$. The reference point $x_i^{\text{ref}}(n\Delta t)$ is then simply obtained by advecting forward the point $x_i^{\text{ref}}(0)$ with the exact transport equation over the time interval $n\Delta t$. Finally, the reference curvature and the normal vector are computed at coordinates $\mathbf{x}_i^{\text{ref}}(n\Delta t)$ with relations (18) and (19) applied to the analytical surface $S^{\text{ref}}(n\Delta t)$.

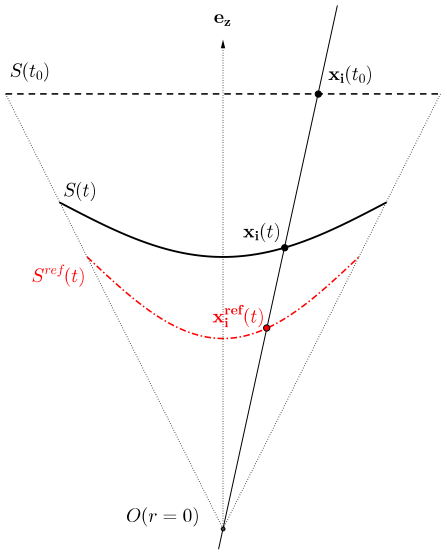


Figure 18: Choice of the reference value to compute the relative errors.

4.1.4. Methodology of the numerical tests

Following the same methodology as in the previous section, the LBO-Meyer (Meyer et al., 2003), IF-Trygg. (Tryggvason et al., 2011), IF-Triang., and the SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0 (Du et al., 2006) methods are considered. In addition, variants of the Surface Reconstruction method are studied for larger stencils ($\mathcal{N}_2$ instead of $\mathcal{N}_1$) or by keeping the degree of freedom a_{00} in the evaluation of the approximated surface $Z = f(X, Y)$ (Eq. 14). For all these methods, the tracking of the interface as a function of time is carried out for an initial mesh made up of, either perfect equilateral triangles, or irregular patterns (see. Eq. 20).

To point out the role played by the advection scheme on the curvature and normal vector evaluation, simulations are carried out into two steps.

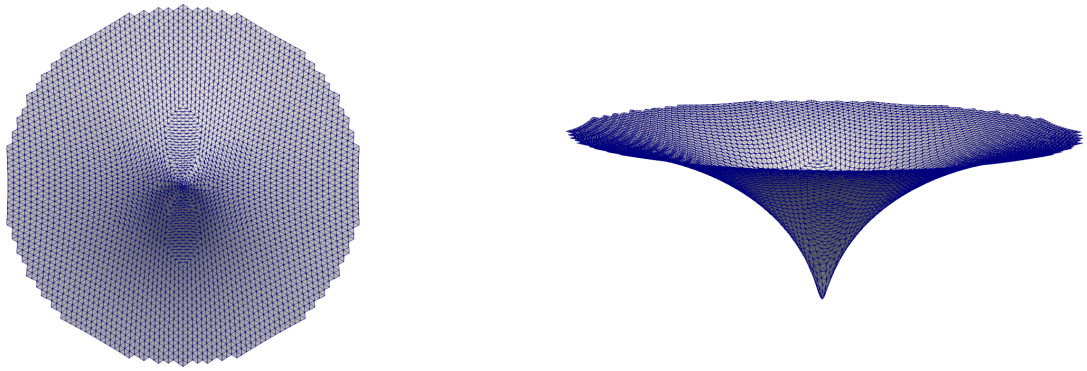
First, the displacement of the vertices is performed in an exact way in order to get rid of errors associated to the transport scheme. Thus, the difference between the numerical and exact solutions can only be due to the mesh structure, namely the spatial distribution of the triangles and their departure from the ideal equilateral shape. Contrary to the stationary surfaces, whose vertices always belong to six triangles, the addition or suppression of vertices/triangles during the advection process modify the number of neighbours.

In a second step, the transport scheme is accounted for, in order to estimate how much it reduces the accuracy of the numerical approximations for the curvature and normal vector.

4.2. Ligament formation – Case $Q < 0$

For a sink located at the origin $O(r = 0)$, the parameter Q is negative. The front tracking surface, initially at $z = z_0$, is sucked all the more that the vertices are close to the singularity. The resulting surface exhibits a thin tail coming closer and closer to O as time runs out. It is interesting to notice that, as can be observed in a two-phase flow, the evolution of the surface looks like a long liquid ligament which becomes thinner and thinner as a function of time. In real two-phase flow, the surface tension instabilities would potentially break up the structure to produce droplets.

In the next subsections, the initial flat surface is located at $z_0 = 1$ and the volume flow rate is set to $4\pi Q = -0.4\pi/3$. The calculations are carried out until time $t = 9\,s$. Figure 19 illustrates the shape of the surface at the final instant, for vertices advected in an exact way.



(a) Bottom view

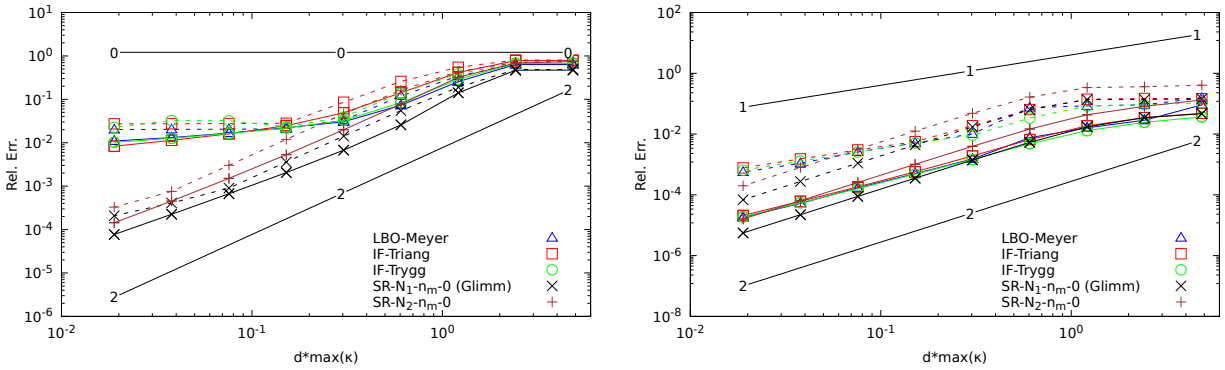
(b) Side view

Figure 19: Bottom and side views of the discrete mesh transported in a spherical velocity field, $Q < 0$

4.2.1. Exact transport – Ideal case

In the ideal case, vertices added by the remeshing process are projected back on the analytical surface and simulations are carried out with the exact transport model, starting

from the surface at $z = z_0$ discretized by equilateral triangles (see Eq. 25).



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure 20: Relative errors in L_2 (continuous line) and L_∞ (dash line) norms for κ and $\mathbf{n}$, $p = 0\%$ with exact transport - $Q < 0$

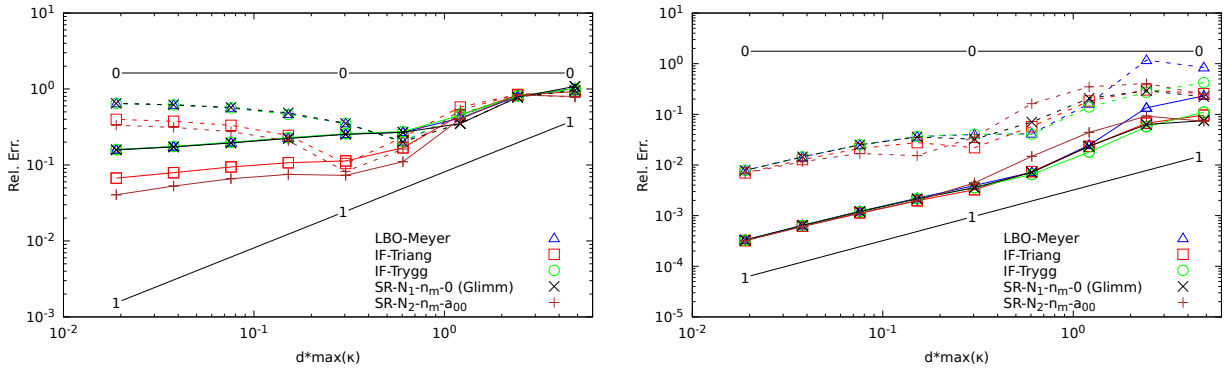
The behaviors of the curvature error as a function of the reduced mesh size (Fig. 20) show convergences in L_2-norm, but with very different orders depending whether IF or LBO methods are used (0.3, 0.34, 0.47 respectively for LBO-Meyer, IF-Trygg. and IF-Triang.) or the SR approximations are considered (nearly second order). This distinction is also true for the L_∞-norm: whereas the asymptotic convergence is close to 2 for the SR-methods, a saturation is observed for the two other classes of methods. Concerning the normal vector, all the methods are convergent with a second order slope in L_2-norm, order that is preserved in L_∞-norm with the SR methods and reduced to first order for the IF and LBO-methods.

Introducing 20%-perturbations in the initial mesh destroys the convergence of the curvature (Fig. A.27 in Appendix A) for the LBO and IF approximations in L_2-norm, and even exhibits a divergence behavior in L_∞-norm. This deterioration of accuracy is also measured for the SR solutions where the order of convergence falls from 2 to 1, whatever the norm considered. The normal vector approximations are also getting worst for the LBO and IF methods, decreasing from 2 to 1, whereas their orders are unchanged for the SR approximations.

4.2.2. Numerical scheme for the advection

Unlike the ideal case, the new vertices from the remeshing process are not projected back on the analytical surface and the advection is a second order Runge-Kutta scheme (Heun's method). The vertex velocity is interpolated by the PERM method (McDermott and Pope, 2008).

Figure 21 gives the relative errors for the curvature and the normal vector, when the initial flat surface is made of equilateral triangles and an advection of the vertices is carried out by the numerical scheme. For the curvature evaluation, the substitution of the exact



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure 21: Relative errors in L_2 (continuous line) and L_∞ (dash line) norms for κ and $\mathbf{n}$, $p = 0\%$ with Lagrangian transport - $Q < 0$

transport by a numerical scheme increases the level of the errors for all methods (Figs. 20a and 21a). Moreover, the SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0 (Du et al., 2006) approximation loses precision to become as accurate as the LBO-Meyer (Meyer et al., 2003) solution, which is much cheaper in computation time than SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0. To recover more accurate solutions, the number of neighbouring triangles utilized to define the reconstructed surface is increased from $\mathcal{N}_1$ to $\mathcal{N}_2$, and the degree of freedom a_{00} of the analytical surface $Z = f(X, Y)$ (Eq. 14) is let free in the fitting process. These changes lead to the SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} method. The use of the numerical scheme for the vertex advection also deteriorates the order of convergence. As discussed here-above, the improvement in the quality of the approximations is the result of the increase in the stencil size with, for the Surface Reconstruction methods, the addition of one degree

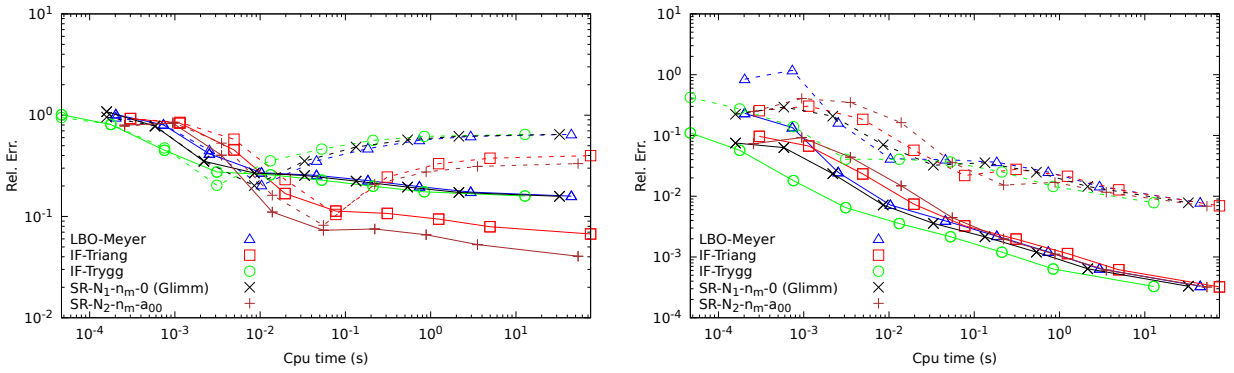
of freedom to compute the surface fitting. These two conditions ensure a smoothing effect of the surface irregularities, which originates from local remeshing performed to preserve a good mesh quality (see Sec. 4.1.2 for details).

For the normal vector calculation, the level of the errors is higher than the one obtained with the exact transport and the order of convergence is 1 (Figs. 20b and 21b).

A quick insight on results initialized with the flat surface made of equilateral triangles, whose vertices are randomly perturbed with maximum magnitude equal to a 20% edge length (see Fig. A.28 in Appendix A), shows that no significant difference appears between solutions obtained with $p = 0\%$ and $p = 20\%$. Thus, contrary to the exact advection case where the orders of convergence were substantially reduced with the loss of mesh regularity, this is not observed here because the transport scheme introduces errors that dominate those produced by the lack of mesh regularity.

4.2.3. CPU-cost and efficiency

The choice of a numerical method results from an equilibrium between the accuracy it provides and the computation time required to get the solution. Figure 22 shows the relative



(a) (b)

Figure 22: Relative errors on κ (a) and $\mathbf{n}$ (b) according to CPU time in L_2 (continuous line) and L_∞ (dash line) norms, $p = 0\%$ - $Q < 0$ and numerical transport.

errors for the curvature and normal vector as a function of the CPU-cost, for an initial flat surface made up of undisturbed ($p = 0\%$) triangles advected by the numerical scheme. It has

to be noticed that, for a given method and mesh, the computation time is identical between the curvature and the normal vector because both approximations are highly linked together (see eq. 10 for LBO- and IF-methods, and Eqs. 18 and 19 for the SR-methods).

For the curvature evaluation, the SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} and IF-Triang. methods are the most CPU-time efficient as soon as the relative error in L2-norm is less than 20%. In that case, the aforementioned methods are 2, even 3 times more accurate for the same CPU cost. Their superiority is less clear when the normal vector has to be approximated (Fig. 22b). Indeed, unlike for the curvature, no method is really distinguishable from the others, except maybe the IF-Trygg. method (Tryggvason et al., 2011) which goes a little bit faster, mainly when a coarse approximation is required. The computation times for disturbed initial meshes with $p = 20\%$ provide similar conclusions (Appendix A, Fig. A.29).

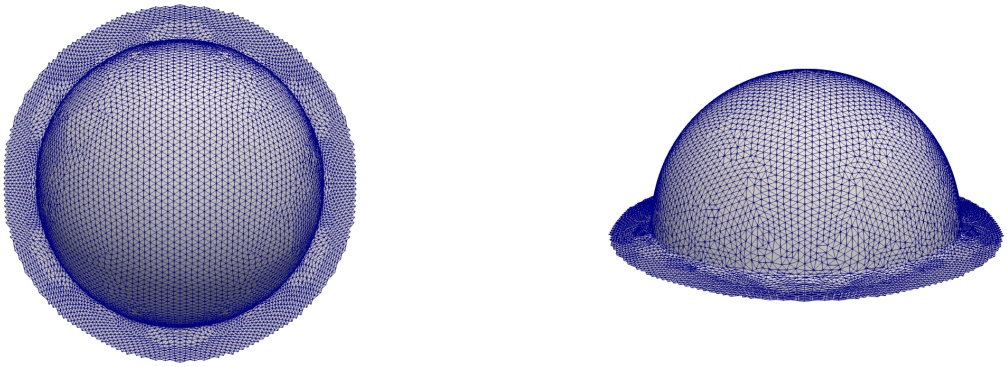
4.3. Bubble shape – Case $Q > 0$

When the flow emerges from the singularity $O(r = 0)$, the initial surface at $z = z_0$ is inflated in the radial direction, giving rise to a hemispherical shape connected on its boundary to a nearly flat surface. Like for $Q < 0$, this peculiar shape may be related to interfaces observed in two-phase fluid flows like droplet impinging onto a liquid film on a wall. A last interpretation could be the rising of a gas bubble which emerges through a gas/liquid interface. However, this case may be more critical than for $Q < 0$ because the region of rapid variation of cuvature is much wider.

In this part, the initial surface is located at $z_0 = 0.01$ and the volume flow rate is equal to $4\pi Q = +0.4\pi/3$. Again, the surface advection is carried out until time $t = 9\,s$. Figure 23 illustrates the shape of the surface at the final instant, for vertices advected in an exact way.

4.3.1. Exact transport – Ideal case

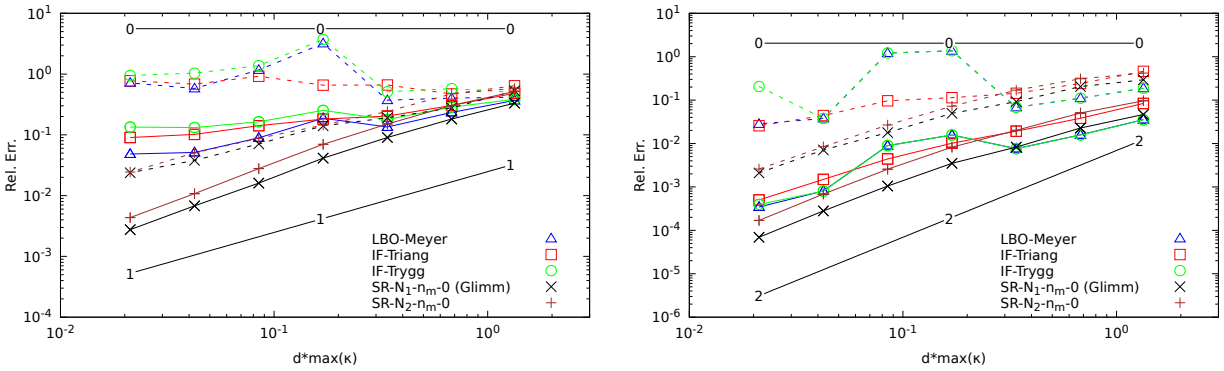
Like in Sec. 4.2.1 for $Q < 0$, the exact advection of the initial surface meshed by equilateral triangles is first studied, where the additional nodes, stemming from the refinement step, are projected on the analytical surface to ensure that all vertices lie on this surface.



(a) Bottom view (b) Side view

Figure 23: Top and side views of the discrete mesh transported in a spherical velocity field, $Q > 0$

Figure 24a exhibits that the errors on the curvature evaluation, calculated with the Laplace-



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure 24: Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 0\%$ with exact transport - $Q > 0$

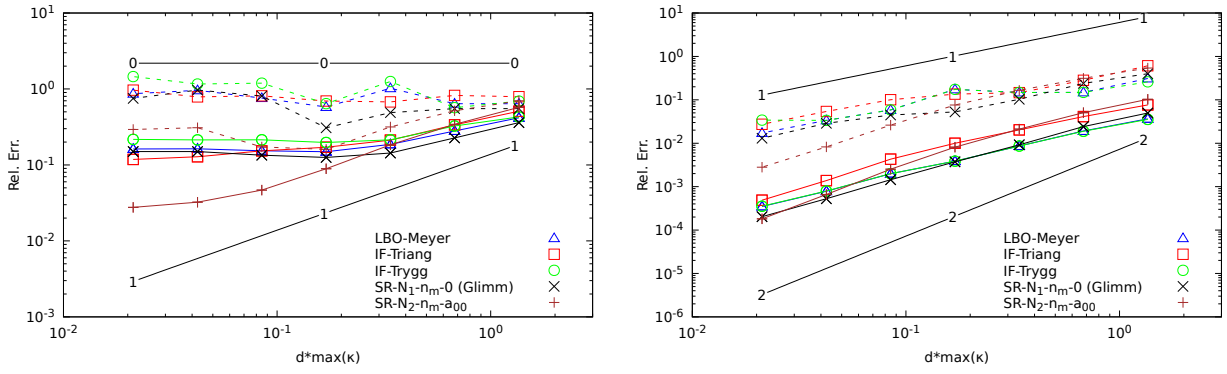
Beltrami Operator and Integral Formulations methods, stagnate when the mesh size decreases. The Surface Reconstruction methods SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0 and SR-$\mathcal{N}_2$-$\mathbf{n}_m$-0 converge with a slope slightly better than 1.

Applying 20%-perturbations to the initial mesh does not change the behaviors of the errors with the mesh refinement: the levels of the errors are only slightly increased for the curvature and the normal vector approximations (Fig. A.30 in Appendix A) .

4.3.2. Numerical scheme for the advection

As a reminder, the advection is performed with a second order Runge-Kutta scheme (Heun's method) and the vertex velocity is interpolated by the PERM method of McDermott and Pope (2008).

The accounting of the advection scheme for the initial surface meshed by equilateral triangles increases unsurprisingly the error levels, especially for the curvature variable (Fig. 25a versus Fig. 24a). On the finest mesh, the saturation of the LBO- and IF-methods is found



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure 25: Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 0\%$ with Lagrangian transport - $Q > 0$

again in L_2-norm and L_∞-norm. Only the SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} method sticks out of the others and gives the smallest level of errors on the curvature. The order of convergence, beginning from 1 on coarse meshes, becomes smaller than 1 when refining the mesh.

As usual, the error levels for the normal vector calculation are quite low (Fig. 25b), compared to those obtained for the curvature. The approximations based on the smallest stencil provide the best results (LBO-Meyer, IF-Trygg. and SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0). The two other methods, namely IF-Triang. and SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00}, give similar errors, slightly larger than the previous ones.

The addition of 20% random perturbations to the initial discretization of the flat surface increases slightly the error levels (Fig. A.31 in Appendix A). For the curvature evaluation

(Fig. A.31a), the SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} method seems converging with a first order accuracy, except for the two finest meshes for which the error tends to stagnate. This increase in the relative error is found for all the methods, since it is related to a bad approximation of the surface by the discrete mesh, as discussed here-above for $p = 0\%$, and not to the evaluation method of the curvature itself.

Concerning the approximation of the normal vector (Fig. A.31b), the same comments given for $p = 0\%$ apply: except the finest meshes, for which SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} is the most accurate, the methods based on the smallest stencils (LBO-Meyer, IF-Trygg. and SR-$\mathcal{N}_1$-$\mathbf{n}_m$-0) provide usually the lower errors.

4.3.3. CPU-cost and efficiency

The same analysis, as the one carried out for $Q < 0$ on the relative accuracy versus the CPU-time (Sec. 4.2.3), is now considered (Fig. 26). The data of the relative error are

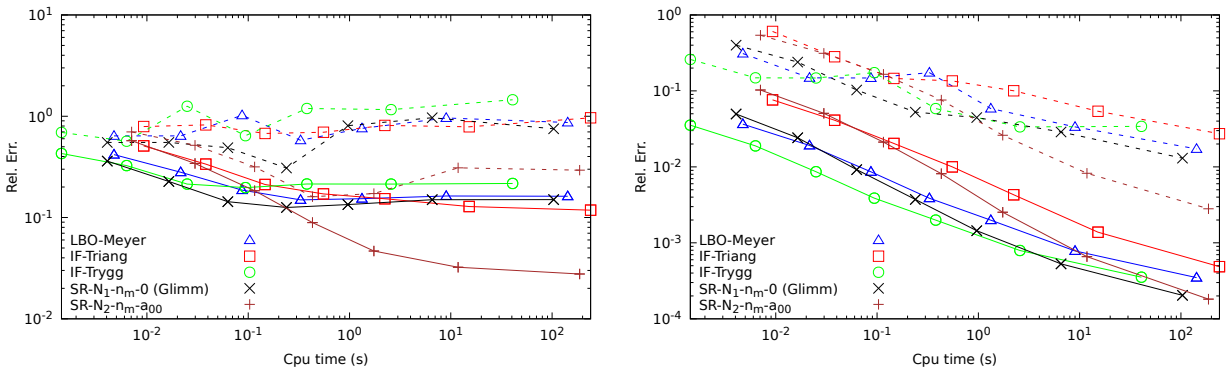


Figure 26: Relative errors on κ (a) and $\mathbf{n}$ (b) according to CPU time in L_2 (continuous line) and L_∞ (dash line) norms, $p = 0\%$ - $Q > 0$.

extracted from Fig. 25. The SR-$\mathcal{N}_2$-$\mathbf{n}_m$-0 method, which was clearly better than the others for estimating the curvature when $Q < 0$, turns out to be also the best choice, even if its efficiency becomes less obvious (Fig. 26a). However, this is the only one capable of providing approximations lower than 10% in L_2-norm, with a L_∞-norm often very much lower than those given by the other methods.

On the whole, the IF-Trygg. method (Tryggvason et al., 2011) is the most efficient for the calculation of the normal vector (Fig. 26b). It is more than ten times less expensive than SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} method for coarse meshes, but this gap is closed for the finest surface discretizations. To ensure that the normal vectors are well approximated on all the mesh vertices, the L_∞-norm must be considered. In that case, only the SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} method allows getting relative errors lower than 1%. The computation times for disturbed initial meshes with $p = 20\%$ provide similar conclusions (Appendix A, Fig. A.32).

4.4. Summary

The robustness of the Surface Reconstruction methods has already been proved for stationary surfaces, regarding the initial disturbances. The key point to get a good approximation remains the evaluation of the curvature of the surface which requires the most restrictive condition on the mesh size. The accounting of the mesh vertex advection by a numerical scheme deteriorates the relative accuracy of the solution whatever the methods to estimate the surface properties. However, the SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} method still remains the most competitive. In a more general way, methods using a larger stencil (SR-$\mathcal{N}_2$-$\mathbf{n}_m$-a_{00} and IF-Triang. methods) are the most accurate in the dynamic case with Lagrangian transport.

5. Conclusion

Three families of numerical methods have been investigated to evaluate the local curvature and the unit normal vector to a surface. They were compared on steady and unsteady interfaces, the later being newly proposed in the present work. To this end, an original analytical solution of a time dependent surface has been developed. Starting from a flat surface and depending on the direction of the flow rate, a long thin ligament or a hemispherical shape bubble joined at its boundary to a flat surface grows with time. These dynamic surface evaluation problems are interesting as they allow to recover geometrical trends found in real two-phase flows.

A first general conclusion, valid for both static and dynamic surfaces, is that there is no major difficulty to get an accurate approximation of the normal vector. The main issue lies in the curvature calculation. Indeed, its analytical expression is based on second order derivatives and then requires regularity. The Laplace-Beltrami-Operator approximations and the Integral Formulation methods are very accurate and efficient for regular meshes. However, as soon as perturbations are introduced, for example on the equilateral triangle mesh for the stationary surface or through the remeshing procedures (dynamic surfaces) that are used during the time integration to track the surface, the curvature evaluation does not converge any more with the decrease of the mesh size. The third approach, the Surface Reconstruction method, is basically more expensive than the previous popular methods. However, it is more robust and provides better results as soon as the surface mesh evolves or is perturbed. Nonetheless, special cares must be taken for SR-$\mathcal{N}_1$ methods as it is done in this paper. Their neighbourhood must be extended to the second one when the degree of freedom for reconstructing the surface is not reached. Generally, even if the SR methods are less accurate on coarse meshes, they become more robust with mesh refinement. So, the Surface Reconstruction methods using the first neighbourhood excel in the dynamic case with exact transport, that is the initial and additional vertices brought by the mesh management process are projected back on the exact surface. Whereas the SR-$\mathcal{N}_2$ methods shine in a more segregate case with Lagrangian transport and mesh perturbation. Although expensive, the Surface Reconstruction methods turn out to be the most efficient methods when the surface is advected: for a given relative accuracy, the CPU-time is often the lowest compared to other methods.

This thoroughly study on the surface geometric properties has emphasized the fact that the remeshing process plays a fundamental role to get a correct evaluation of the curvature. An expected improvement could be to make use of the fitted surface to preserve a good representation of the theoretical one. The new or moved vertices, stemming from the refinement or coarsening step of the discrete mesh, could be projected on the local reconstructed surface in order to preserve a high quality approximation of the analytical surface.

Moreover, the approaches used in the Surface Reconstruction methods are the most used and also among the simplest. Other techniques exist (especially in the domain of computer aided graphics that could give better results (Cazals and Pouget, 2005; Pottmann et al., 2007; Rusinkiewicz, 2004). However, it has to be noted that all these approaches have to be coupled with Navier-Stokes resolution. Hence, the precision brought by these methods must be compared in light of those used in the Navier-Stokes discretisation.

Appendix A. Errors for 20%-disturbed meshes

As described in the body of the present article, when the initial mesh was made up of equilateral triangles, the remeshing process has destroyed the regular structure of the discrete surface. As a consequence, the measured errors were much larger than those recorded for the stationary surface, especially for the curvature. Therefore, introducing random disturbances on the initial location of the vertices does not significantly modify, neither the behaviour, nor the level of the errors. This remark is valid whatever the Q-value, as well for the exact advection (Eq. 25):

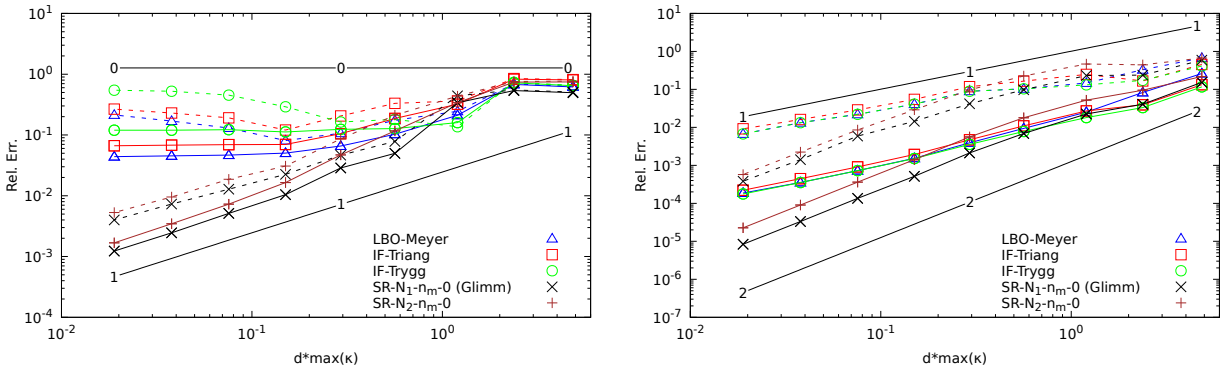
- see Fig. A.27 versus Fig. 20 for $Q < 0$,
- see Fig. A.30 versus Fig. 24 for $Q > 0$

as for the transport numerical scheme:

- see Fig. A.28 versus Fig. 21 for $Q < 0$,
- see Fig. A.31 versus Fig. 25 for $Q > 0$.

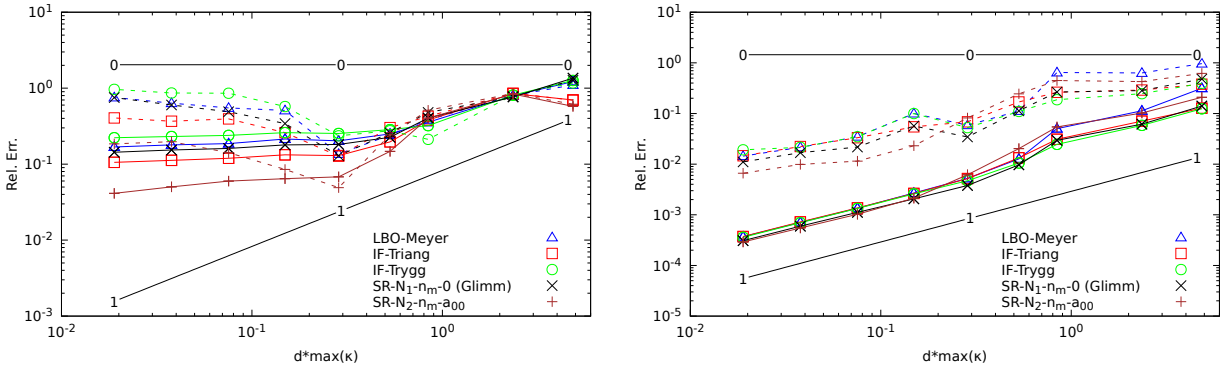
and for the CPU-cost and efficiency:

- see Fig. A.29 versus Fig. 22 for $Q < 0$,
- see Fig. A.32 versus Fig. 26 for $Q > 0$.



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure A.27: Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 20\%$ with exact transport - $Q < 0$



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

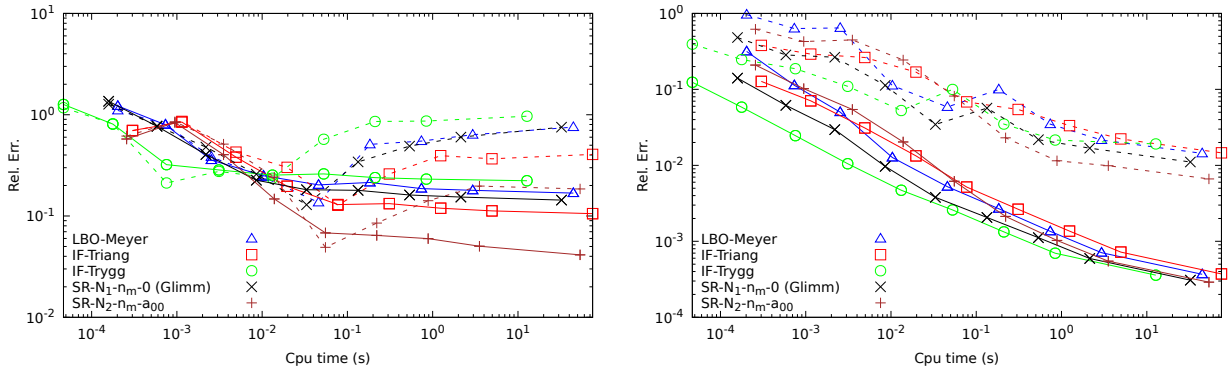
Figure A.28: Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 20\%$ with Lagrangian transport - $Q < 0$

Appendix B. Remeshing of the surface

In order to have an adequate description of a surface during its evolution in a multiphase flow simulation, different process are implemented and described below.

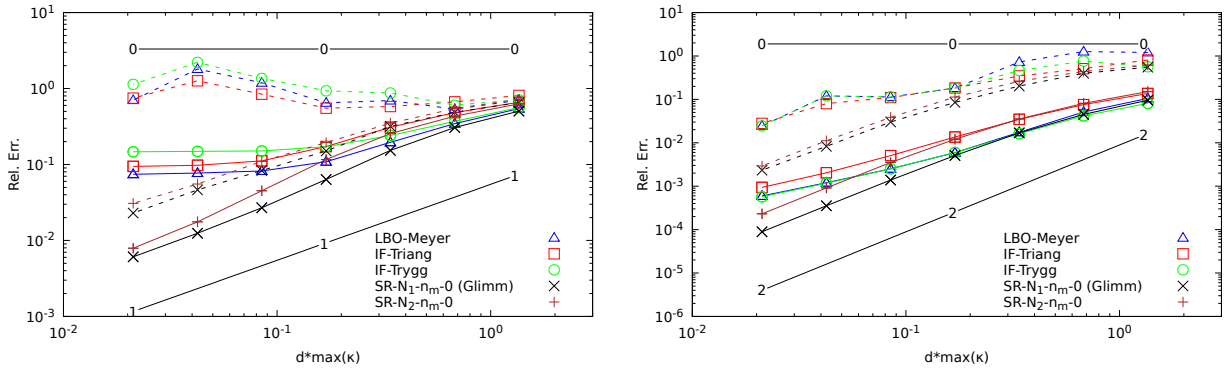
Edge splitting.

This operations consists on adding a new vertex on an edge of a triangular mesh. It happens when this selected edge is longer than a specific length. In our study, we decide that any edge of length d greater than the Eulerian grid size h is splitting in two new edges. So, if a triangle has 1, 2 or 3 large edges, it results in the formation



(a) (b)

Figure A.29: Relative errors on κ (a) and $\mathbf{n}$ (b) according to CPU time in L_2 (continuous line) and L_∞ (dash line) norms, $p = 20\%$ - $Q < 0$ and numerical transport.



(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

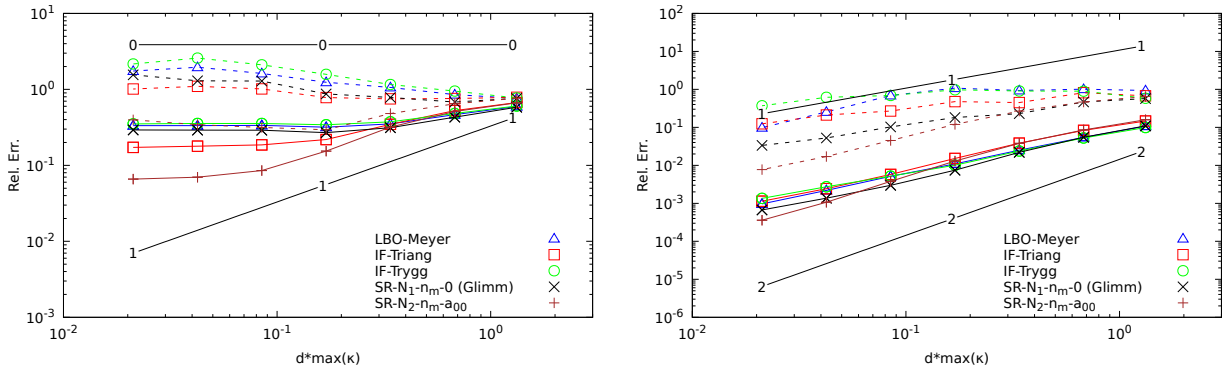
Figure A.30: Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 20\%$ with exact transport - $Q > 0$

of respectively 2, 3 and 4 small triangles B.33. And the new vertices are added in the middle of their respective edges.

Edge collapsing.

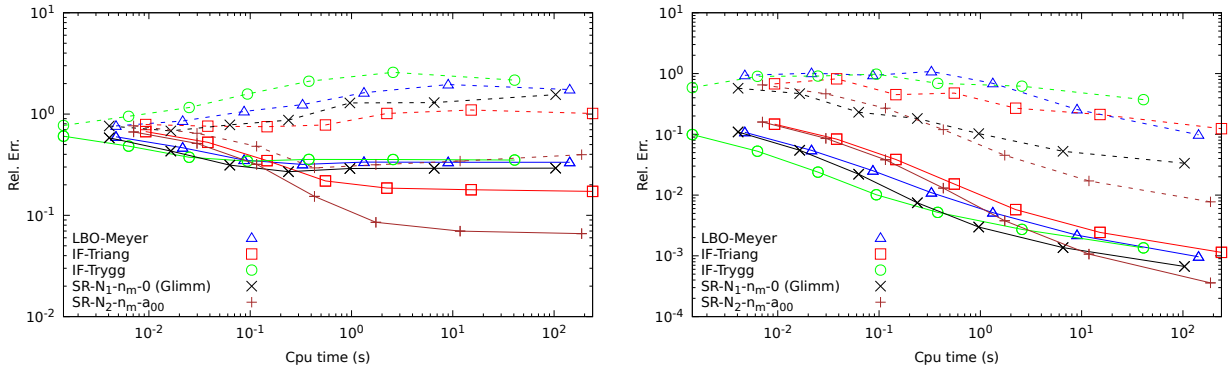
It is the opposite of the edge splitting process. In this case, any edge of length d smaller than $h/2$, with h the Eulerian grid size, is removed. This results in the suppression of two triangles, and the collapsing of the edges vertices onto their middle point B.34.

Edge swapping.



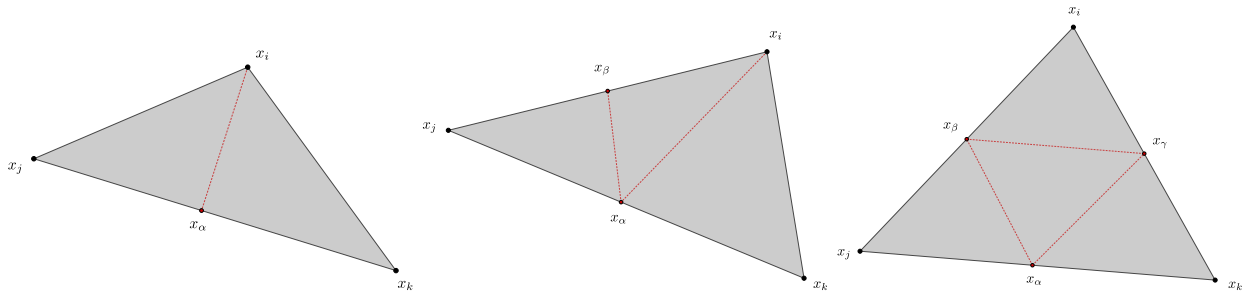
(a) $\mathrm{Err}^{\mathrm{rel}}(\kappa)$ (b) $\mathrm{Err}^{\mathrm{rel}}(\mathbf{n})$

Figure A.31: Relative errors in L_2 (continuous line) and L_∞ (dash line) norms on κ and $\mathbf{n}$, $p = 20\%$ with Lagrangian transport - $Q > 0$



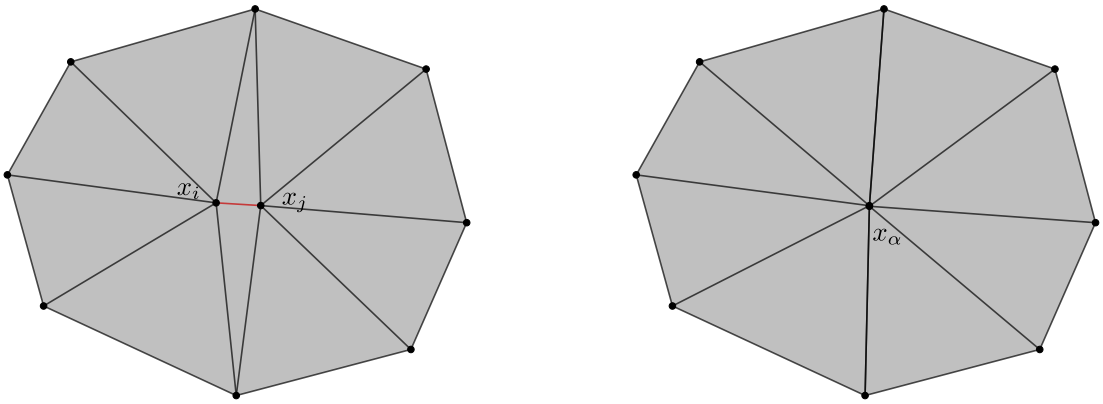
(a) (b)

Figure A.32: Relative errors on κ (a) and $\mathbf{n}$ (b) according to CPU time in L_2 (continuous line) and L_∞ (dash line) norms, $p = 20\%$ - $Q > 0$ and numerical transport.



(a) one edge splitting case (b) two edges splitting case (c) three edges splitting case

Figure B.33: Different edge splitting cases

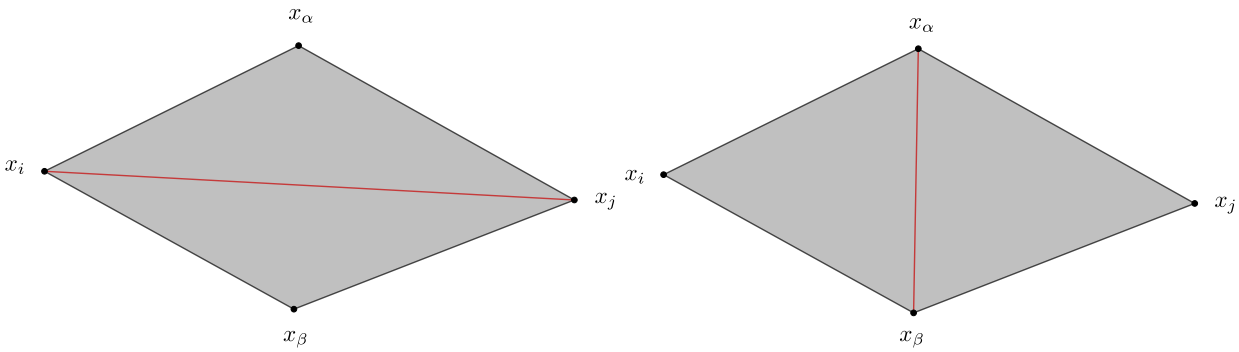


(a) Edge $x_i x_j$ to collapse

(b) After collapsing

Figure B.34: Before and after collpasing the edge $x_i x_j$ onto their middle vertex x_α.

Contrary to the two previous methods that aim to have a appropriate mesh resolution of the surface troughout its evolution, this method deals with the quality of the mesh. So, it swaps the edge shared by two triangles in order to construct more equilateral triangles, as described in figure B.35.



(a) Edge $x_i x_j$ to swap

(b) After swapping

Figure B.35: Edge $x_i x_j$ is swapped to create the edge $x_\alpha x_\beta$.

References

Brackbill, J.U., Kothe, D.B. & Zemach, C., 1992, A continuum method for modeling surface tension, *Journal of Computational Physics*, 100, 335–354

Canu, R., Puggelli, S., Essadki, M., Duret, B., Ménard, T., Massot, M., Reveillon, J. & Demoulin, F.X., 2018, Where does the droplet size distribution come from?, *International Journal of Multiphase Flow*, 107, 230–245

Delhaye, J.M., 1974, Jump conditions and entropy sources in two-phase systems. Local instant formulation, *International Journal of Multiphase Flow*, 1, 395–409

Demoulin, F.X., Beau, P.A., Blokkeel, G., Mura, A. & Borghi, R., 2007, A new model for turbulent flows with large density fluctuations: Application to liquid atomization, *Atomization and Sprays*, 17, 315–345

Desbrun, M., Meyer, M., Schroder, P. & Barr, A.H., 1999, A new model for turbulent flows with large density fluctuations: Application to liquid atomization, *SIGGRAPH99*, 317–324

Du, J., Fix, B., Glimm, J., Jia, X., Li, X., Li, Y. & Wu, L., 2006, A simple package for front tracking, *Journal of Computational Physics*, 213, 613–628

Essadki, M., Drui, F., de Chaisemartin, S., Larat, A., Ménard, T. & Massot, M., 2019, Statistical modeling of the gas-liquid interface using geometrical variables: toward a unified description of the disperse and separated phases, *International Journal on Multiphase Flows*

Essadki, M., 2018, *Contribution to a unified Eulerian modeling of fuel injection: from dense liquid to polydisperse evaporating spray*, Universite Paris-Saclay Ph.D. thesis

Fedkiw, R., Aslam, T., Merriman, B. & Osher, S., 1999, A non-oscillatory Eulerian approach to interfaces in multi-material flows (the Ghost Fluid method), *Journal of Computational Physics*, 152, 457–492

Goldman, R., 2005, Curvature formulas for implicit curves and surfaces, *Computer Aided Geometric Design*, 22, 632–658

Li, X., Xu, G. & Zhang, J., 2015, Localized discrete Laplace-Beltrami operator over triangular mesh, *Computer Aided Geometric Design*, 39, 67–82

Max, N., 1999, Weights for computing vertex normals from facets normals, *Journal of Graphics Tools*, 4, 1–6

McDermott, R. & Pope, S.B., 2008, The parabolic edge reconstruction method (PERM) for Lagrangian particle advection, *Journal of Computational Physics*, 227, 5447–5491

Meyer, M., Desbrun, M., Schroder, P. & Barr, A.H., 2003, Discrete Differential-Geometry Operators for Triangulated 2-Manifolds, *Visualization and Mathematics III*, 35–57

Polthier, K., Hass, J., Hoffman, D., Jaffe, A., Rosenberg, H., Schoen, R. & Wolf, M., 2002, Computational aspects of discrete minimal surfaces, *Proc. of the Clay Summer school on Global Theory of Minimal Surfaces*

Popinet, S. & Zaleski, S., 1999, A front-tracking algorithm for accurate representation of surface tension, *International Journal for Numerical Methods in Fluids*, 30, 775–793

Roghair, I., Van Sint Annaland, M. & Kuipers, J.A.M., 2016, An improved Front-Tracking technique for the simulation of mass transfer in dense bubbly flows, *Chemical Engineering Science*, 152, 351–369

Shin, S. & Juric, D., 2002, Modeling Three-Dimensional Multiphase Flow Using a Level Contour Reconstruction Method for Front Tracking without Connectivity, *Journal of Computational Physics*, 180, 427–470

Taubin, G., 1995, A signal processing approach to fair surface design, *SIGGRAPH '95*, 351–358

Trontin, P., Vincent, S., Estivalezes, J.-L. & Caltagirone, J.P., 2012, A subgrid computation of the curvature by a particle/level-set method. Application to a front-tracking/ghost-fluid method for incompressible flows, *Journal of Computational Physics*, 231, 6990–7010

Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S. & Jan, Y.-J., 2001, A Front-Tracking Method for the Computations of Multiphase Flow, *Journal of Computational Physics*, 169, 708–759

Tryggvason, G., Scardovelli, R. & Zaleski, S., 2011, *Direct Numerical Simulations of Gas-Liquid Multiphase Flows* Cambridge University Press

Vaudor, G., Ménard, T., Aniszewski, W., Doring, M. & Berlemont, A., 2017, A consistent mass and momentum flux computation method for two phase flows. Application to atomization process, *Computers and Fluids*, 152, 204–216

Wardetzky, M., 2008, Convergence of the Cotangent Formula: An Overview, *Discrete Differential Goemetry*, 38, 275–286

Xu, G., 2004, Discrete Laplace–Beltrami operators and their convergence, *Computer Aided Geometric Design*, 21, 767–784

Zinchenko, A.Z., Rother, M.A. & Davis, R.H., 1997, A novel boundary-integral algorithm for viscous interaction of deformable drops, *Physics of Fluids*, 9

Zuzio, D., Estivalezes, J.-L., Villedieu, P. & Blanchard, G., 2013, Numerical simulation of primary and secondary atomization, *Comptes Rendus Mecanique*, 341, 15–25

Cazals, F. & Pouget, M., 2005, Estimating differential quantities using polynomial fitting of osculating jets, *Computer Aided Geometric Design*, 22, 121–146

Pottmann, H., Wallner, J., Yang, Y. L. , Lai, Y. K.& Hu, S. M., 2007, Principal curvatures from the integral invariant viewpoint, *Computer Aided Geometric Design*, 24, 428–442

Wardetzky, M., 2004, Estimating Curvatures and Their Derivatives on Triangle Meshes, *Symposium on 3D Data Processing, Visualization, and Transmission, September 2004*

Yu, D-H., 2002, *Natural Boundary Integral Method and Its Applications* Springer Netherlands 539

Koffi Bi, D.-A., 2021, *Numerical characterisation of curvatures and normal vectors at multiphase flows interfaces* PhD thesis Université Gustave Eiffel