Anisotropic Level Set Adaptation for Accurate Interface Capturing*

V. Ducrot[1] and P. Frey[1,2]

[1] UPMC Univ Paris 06, UMR 7598, Laboratoire J.L. Lions, F-75005 Paris, France
ducrot@ann.jussieu.fr
[2] Universidad de Chile, UMI 2807, Centro de Modelamiento Matemático, Santiago, Chile
frey@ann.jussieu.fr

Summary. In fluid-structure interactions and fluid simulations, like incompressible two-phase flows involving high viscosity and density ratios, interface capturing and tracking is considered as a very challenging problem and has a strong impact on industrial applications. Usually, an adaptive grid is used to resolve the problem as well as to track the interface. In this paper, we describe an adaptive scheme based on the definition of an anisotropic metric tensor to control the generation of highly streched elements near an interface described with a level set function. In our approach, quasi-uniform anisotropic meshes are created with the objective of minimizing the interpolation errors by capturing the interface features using curvature-adapted anisotropic elements. The accuracy of the method is verified and numerical experiments are presented to show its efficiency.

1 Introduction

In this paper, we consider the very challenging problem that consists in capturing and tracking moving fronts and interfaces. This situation arises typically in bi-fluid flow simulations, free surface flows, non linear processes like dendritic growth, for example. Such numerical simulations are considered difficult to carry on because the quality of the numerical solution strongly depends on the accuracy of the interface approximation, which is in principle related to the minimal size of the discretisation. Hence, it seems often tedious to find a right compromise between the desired level of accuracy and the overall computationnal cost (memory and time ressources) of the simulation.

Since the pionnering works of Dervieux [11] and Osher [28], level set methods have been successfully used in numerous scientific computing applications where time-dependent equations and moving interfaces were involved. Until recently, most applications of level sets involved structured grids on which high-order schemes can be implemented and computed easily. On such grids and on related spatial decompositions (e.g., octree), the differential operators and the solutions

* Research supported by the INRIA ARC-LMN and a BQR grant from UPMC Scientific committee.

of time-dependent Hamilton-Jacobi equations can be efficiently computed [15]. The simplicity of Cartesian grids associated with the simplicity of level set formulations compared to Lagrangian or ALE methods seemed to have frozen the situation. For dealing with complex geometries (*i.e.,* domain boundaries and interfaces) however, geometry-adapted unstructured meshes have proved more versatile and accurate in finite element simulations, notably because the number of vertices for a given accuracy is lower than with (graded) Cartesian grids. Moreover, high-order non-oscillatory schemes have been efficiently introduced on triangulations[1]. The next step would be to generate anisotropic curvature-adapted meshes to represent the interface manifold. Then, the numerical schemes may be more expensive to solve, but the substantial reduction of the number of elements will largely compensate this drawback and favorably impacting the overall computational cost of the simulation.

Over the last decade, several anisotropic mesh adaptation techniques have been developed and efficient adaptive algorithms are now available, at least for steady-state problems. For time-dependent problems however, and especially dynamically evolving interfaces, the principal difficulty is to follow the arbitrary evolution of the phenomenon in space and time, including the possible topology modification. Indeed, in these simulations, the geometry is intrinsically a part of the solution of the simulation. Usually, the triangulation is adapted frequently, to maintain a valid representation of the interface with minimal approximation errors. This issue has been partially addressed by using a fixed point algorithm and a metric-based refinement procedure to locally adapt a triangulation within each time step with respect to the solution variations (first and second-order derivatives) [4]. In [25], the triangulation of the interface is directly adapted to the interface curvatures using local mesh modifications thus making it difficult to handle topology changes during the simulation. To overcome this problem, we suggest to use the level set formulation of a time-dependent problem that offers the advantage of embedding the manifold in a higher dimensional space and does not require to discretize it explicitly, and anisotropic mesh adaptation techniques to refine the triangulation in the vicinity of the manifold for approximation error control purposes.

The aim of this paper is to present an anisotropic mesh adaptation method, based on a geometric error estimate, for capturing an interface described by a level set function, with a high level of accuracy. This result is obtained by defining a Riemannian metric tensor based on the differential properties of the manifold of codimension one corresponding to the interface [14]. Mesh adaptation consists then in creating a quasi-uniform mesh with respect to the metric tensor field, using local mesh modifications: edge split, edge collapse, edge flip and vertex relocation. As the interface evolves in time, we rely on a fixed point algorithm to correctly locate the interface positions at the beginning and at the end of each time step (before and after each advection step of the level set function). By intersecting the metric tensors corresponding to these relative positions, we will show that it is possible to substantially reduce the diffusion problem. Moreover, we solve this advection step by a second order scheme based on the method

of characteristics. This approach can be used with any implicitly-defined, scalar valued function. This allows us to envisage other categories of applications, for instance: in computational solid geometry modeling (where objects are commonly defined by an implicit function) and in scientific visualization for representing isovalue curves or surfaces of a numerical finite element solution accurately.

The remainder of this paper is organized as follows. In Section 2, we briefly review the level set formulation of the interface tracking problem and we outline the related numerical issues we have specifically addressed. In Section 3, we introduce the main notions of anisotropic mesh adaptation based on Riemannian metric tensor. A complete description of this method can be found in the general purpose book [18] and the references therein. In particular, we outline the definition and construction of a metric tensor and the fixed point algorithm for time dependent problems. In Section 4, we explain how to control and bound the geometric approximation of an interface and we introduce an anisotropic error estimate for this problem. Numerical examples of simulations are given in Section 5 to show the efficiency of this approach.

2 The Level Set Context

2.1 Level Set Formulation

We consider a level set formulation of an time dependent problem in a computational domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ which is subdivided in two subdomains Ω^+ and Ω^- sharing a common interface Γ, such that $\Omega^+ \cup \Omega^- = \Omega$ and $\Omega^+ \cap \Omega^- = \emptyset$ and $\Gamma = \partial\Omega^+ \cap \partial\Omega^-$, where $\partial\Omega^*$ denotes the boundary of the domain Ω^*. Typically, we introduce a Lipschitz-continuous function $u : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}$ and the interface Γ is chosen to coincide with the level set where $u(t, x) = 0$. As suggested by Osher [28], the level set function u is initialized with the distance function to the initial interface $u(0, x) = \pm d(x)$, the sign $\pm$ is negative in Ω^- and positive in Ω^+.

Usually, the motion of the interface is driven by a velocity field $v(t, x)$ related to the differential properties of the manifold. This leads to a first-order advection equation posed in $\mathbb{R}^d$, for all $x \in \Omega$, $t \in [0, T]$:

$$\partial_t u(t, x) + v_n(t, x)|\nabla u(t, x)| = 0 \tag{1}$$

where v_n denotes the normal component of the velocity to each level set. It is well known that singularities are often encountered in solving this equation and thus appropriate techniques must be developed to select the unique viscosity solution [10].

2.2 Numerical Issues

The computational domain Ω is covered by an anisotropic unstructured triangulation T_h and the distance value to the interface is associated with its vertices,

thus defining the level set function u in a discrete manner. The numerical resolution of Equation (1) raises several issues that we emphasize in the next sections.

Level set advection. We use a finite element technique to solve the level set advection problem on anisotropic unstructured meshes. Suppose the vector field $v(t,x)$ is sufficiently smooth, we recall that a *trajectory* associated to the field v is a solution of the differential problem :

$$\frac{dX}{dt}(t,x_0) = v(t, X(t,x_0))\,, \qquad X(0,x_0) = x_0$$

where the point $X(t,x_0)$ represents the position at time $t \in [0,\infty]$ of a particule initially in x_0 at time $t=0$. Such a trajectory is a parametrized curve $X(t,x)$. The method of characteristics solves this problem by converting the PDE into a system of ODE. Using a finite element scheme to find an interpolation of the desired solution u using piecewise affine continuous functions, the variational problems reads, with the basis functions φ_i:

$$\iint_\Omega u^{n+1}(x,y)\varphi_i(x,y)dxdy = \iint_\Omega u^n \circ X^n(x,y)\varphi_i(x,y)dxdy\,. \tag{2}$$

The linear system $Au^{n+1} = f$ is solved using a second-order scheme thanks to a prediction-correction loop [26]. The main advantage of this formulation is that it accounts for large time stepping as compared with other explicit method.

Reinitialization procedure. An interesting property of the signed distance function is that we have $|\nabla u| = 1$, which greatly simplifies the calculation of the normal $\mathbf{n}$, the mean curvature H and the Gaussian curvature K. Hence, we have:

$$\begin{cases} \mathbf{n(x)} = \dfrac{\nabla u(x)}{|\nabla u(x)|} = \nabla u(x)\,, \\ H(x) = \mathrm{div}(\mathbf{n(x)}) = \Delta u(x) \\ K(x) = det(\Pi^t(x) D^2 u(x) \Pi(x))\,, \end{cases} \tag{3}$$

where Π is the matrix of the projection on the tangent plane at x. Unfortunatly, almost all motion fields, but a constant velocity field v, do not preserve this property. Therefore, in order to recover it and the simple formulas, a *reinitialization* stage is introduced.

Two different approaches have been proposed in the literature. On the first hand, one can solve the non-linear equation at steady state [33]:

$$\partial_t \psi - S(\psi)(1 - |\nabla \psi|) = 0\,, \tag{4}$$

where $S(\psi)$ is a sign function taken as 1 in Ω^+, -1 on Ω^- and 0 on the interface Γ, and $\psi(0,x)$ is taken as the current value of $u(t,x)$. At convergence, the function ψ is substituted back to the function u. A drawback of this methods is related to the regularity of the function ψ that may cause the interface Γ to move incorrectly from its starting position, if ψ is not smooth. Moreover, in

the numerical schemes, the time step is usually bounded by a stability condition related to the minimal size h_{min} of the triangulation T_h. Since we are concerned with anisotropic triangulations with high aspect ratio elements and a minimal size at least three orders of magnitude smaller than the domain size to resolve small features, such technique would lead to a prohibitive computational cost.

On the other hand, a Fast-Marching method is used to evolve a front at constant speed initiated from the interface Γ [34, 31, 13]. It consists in solving the following equation in each element:

$$\sum_{i=1}^{d+1} |\nabla \varphi_i u_i|^2 = 1 \tag{5}$$

where the φ_i are the shape functions in the element and u_i is the value of the level set function at the vertices of the element. The problem is usually solved using a first-order scheme. However, this approach could be improved when considering that the function u is a *signed* distance function to an interface Γ. Since the sign is not modified by the reinitialization stage, it is sufficient to compute the Euclidian distance to Γ in order to determine function φ. Furthermore, we have a bound ε on the distance between the approximation Γ_h and the manifold Γ on T_h using the Riemannian metric tensor field. Hence, function φ can be determined with an accuracy of order ε by computing the distance of the mesh vertices to Γ_h. This can be achieved in a efficient by a parallel algorithm for computing point distance to polyline [21]. The complexity of our approach is then in $(m+n)\log m$, where m represents the number of entities in the reconstructed interface and n denotes the number of mesh vertices, $n \gg m$, in general.

3 Anisotropic Mesh Adaptation

In order to resolve the velocity field accurately, it is mandatory to have a sufficient number of grid points in the vicinity of the interface. For this reason, unstructured triangulations offer a better alternative to Cartesian grids in solving this equation [1]. The approximation error is then strongly related to the quality of the triangulation and to the mesh density. To this end, the definition of a Riemaniann metric tensor helps to control the generation of anisotropic elements close to the interface.

In this section, we briefly present some basic notions and definitions on computational triangulations, metric tensor fields and mesh adaptation. We introduce the following notations: Ω denotes a simply connected open bounded domain in $\mathbb{R}^d$, $\overline{\Omega}$ is a closure of Ω and $|\Omega|$ is the d-dimensional measure or the volume of Ω.

3.1 Uniform Triangulations

Suppose given a family of triangulations T_h on the domain Ω, h representing the characteristic element size. Each element $K \in T_h$ is a closed subdomain of Ω and we assume that $\overline{\Omega} \subset \bigcup_{K \in T_h} K$ and that the usual finite element requirements

are verified (*i.e.* elements do not overlap). Then, a *uniform* mesh T_h of Ω is a mesh in which all element are equally sized and regular. In such case, if $|T_h|$ represents the number of mesh elements and $h_K = diam(K)$ the diameter of K, the size $h = \max_{K \in T_h} h_K$ is given by the relation:

$$h_K \approx \left(\frac{|\Omega|}{|T_h|} \right)^{1/d} \qquad \forall\, K \in T_h\,.$$

A *quasi-uniform* mesh is a mesh for which (i) there exists a constant τ such that

$$\frac{h_K}{\rho_K} \leq \tau \qquad \forall K \in \bigcup_h T_h\,,$$

where, for any open ball $B_i \subset K$, $\rho_K = \sup_i\{diam(B_i)\}$ is the in-diameter of K and (ii) the variation of h is bounded by a constant. This does not mean a constant mesh size over the domain.

3.2 Anisotropic Mesh Adaptation Scheme

Nowadays, mesh adaptation is widely used to improve the efficiency of the numerical schemes as well as the accuracy of numerical solutions in computational simulations. It consists in concentrating a maximum number of nodes in regions of large solution variations and a minimal number of nodes in other regions of the computational domain. Consequently, the overall number of nodes required to achieved a desired accuracy can be reduced by a substantial amount, thus impacting favorably the computational cost of the simulation. In many applications, solutions exhibit a large gradient variation in some regions, often combined with highly anisotropic features (shock waves or boundary layers in fluid dynamics, for instance), and elements with high aspect ratio are needed. In this case, isotropic triangulations would contain too many elements. It is thus desirable to adjust the element size as well as the element shape and orientation to better match the solution variations. In the past decade, the theory of error estimates as been largely investigated by various research groups [5, 8, 12, 16, 19] and has provided the concept of *optimal* triangles with respect to a so-called *metric tensor* field based on the gradient and Hessian of the solution [7].

3.3 Metric Tensor Field

From the geometric viewpoint, the metric defining the element size, shape and orientation is represented by an ellipsoid. Hence these notions are related to its volume, the lengths of its semi-axes and its principal axes vectors, respectively. The metric tensor denoted as $M(x)$ is used to generate a quasi-uniform mesh in the metric related to M. More precisely, the volume of an element $K \in T_h$ is unitary:

$$\int_K \sqrt{det(M(x))}\, dx = 1\,, \qquad \forall K \in T_h$$

which corresponds to the discrete formulation

$$|K|\sqrt{det(M_K)} = 1\,, \qquad \forall K \in T_h$$

where M_K is an average of $M(x)$ on K. By extension, the length of a curve γ in a metric given by $M(x)$ for any $x \in \Omega$ is defined as:

$$|\gamma|_M = \int_0^1 \sqrt{\langle \gamma'(t), M(\gamma(t))\gamma'(t)\rangle}dt\,,$$

where $\gamma(t) : [0,1] \to \mathbb{R}^d$ is a parametrization of γ. Since the metric tensor $M(x)$ is a symmetric positive definite matrix, the spectral decomposition theorem allows to decompose M as:

$$M = P\,\Lambda\,P^t \;=\; \sum_{i=1}^{d} \lambda_i \mathbf{e_i}\mathbf{e_i}^t\,,$$

where the normalized eigenvectors of M are the columns of matrix $P = [\mathbf{e_1}, \ldots, \mathbf{e_d}]$ such that $P\,P^t = I_d$ and Λ is the diagonal matrix of the eigenvalues λ_i. Notice that the matrix P prescribes the orientation and the matrix Λ prescribes the size and shape of any element K.

3.4 Mesh Quality

As we have numerical simulations in mind, it is important to introduce quality measures in order to evaluate how close the metric prescriptions are satisfied at any x in the domain. Mesh quality assessement has been an active area of research for isotropic meshes (see [6] for a survey). For practical reasons, we will consider a single measure to evaluate the quality of an element K:

$$Q_{ani}(K) = \alpha_d \frac{\left(\sum_{i=1}^{k} \langle \mathbf{e_i}, M_K \mathbf{e_i}\rangle\right)^d}{|K|\sqrt{det(M_K)}}\,,$$

where $\mathbf{e_i}$ represents here any of the k egdes of K and α_d is a normalisation constant such that $Q_{ani}(K) = 1$ for a regular element. Notice that $Q_{ani} \geq 1$ for all $K \in T_h$ and thus the larger $\max_K Q_{ani}(K)$ is, the more the triangulation T_h deviates from the metric prescriptions.

3.5 Anisotropic Mesh Adaptation

Traditionally, mesh adaptation algorithm are based on either r-methods or h-methods. In the first approach, the number of mesh vertices is kept constant, only the vertex locations are changed throughout the simulation. Algorithms in the second category proceed by local mesh modifications like edge flipping, edge contraction, vertex addition and vertex relocation. In our approach, anisotropic meshes are created using a Delaunay-based method in which a metric tensor

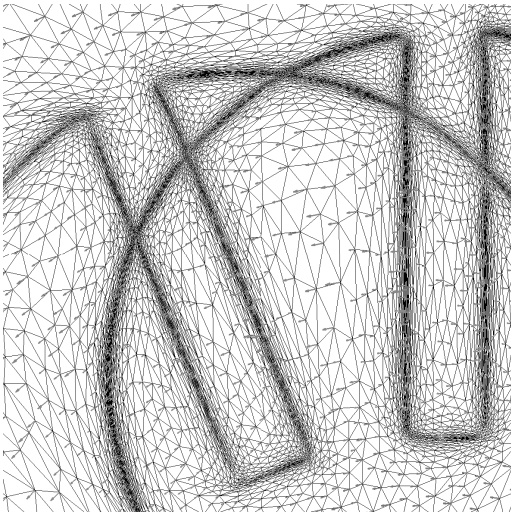


Fig. 1. Anisotropic mesh for the transient fixed-point problem. Adapted to the solutions u_n and u_{n+1}, and the corresponding velocity field at time t_n.

prescribing the size, stretching and orientation of the element is associated with the mesh vertices [18]. Moreover for unsteady physical problems, several metric tensor fields must be combined to overcome the numerical diffusion associated with the numerical scheme [4, 20].

Solution prediction. To deal with unsteady phenomena, we use a fixed-point mesh adaptation algorithm to predict the solution evolution in the domain Ω. This algorithm solves a transient fixed-point problem for the couple (mesh, solution) at each iteration of the classical adaptation scheme. Suppose that the triangulation T_h is adapted to the behavior of the solution $u_n = u(t_n, x)$, for all x in Ω, at the beginning of a time period $[t_n, t_{n+1}]$, the aim of this scheme is to predict the solution behavior at t_{n+1} and to refine the mesh accordingly. Therefore, within this time period at least two anisotropic metric tensors must combined together, producing a mesh that is refined in regions of large gradient variations of both u_n and u_{n+1}. This procedure can be seen as an internal loop within a classical adaptation scheme. In each internal iteration j, the solution u_{n+1}^j converges toward the solution u_{n+1}. This can be checked numerically using the L^1-norm of the difference between the solutions u_{n+1}^j and u_{n+1}^{j-1}.

When two solutions are considered their relative metric tensors must be reduced to a single one for mesh generation purpose. To this end, a metric intersection procedure is used, based on the simultaneous reduction of the quadratic forms associated with the tensors [20]. Figure 1 shows the anisotropic mesh used to resolve the internal fixed-point loop.

Local mesh adaptation. In our approach, we rely on local topological and geometrical mesh modifications (edge split, edge collapse, vertex relocation, edge flip). At each step, the current triangulation is modified to generate quasi-uniform triangulations with respect to the metric tensor field M. The method is based on the analysis of the mesh edges: all edges must belong to the interval $[l_{min}, l_{max}]$ and the mesh elements quality must be close to the optimal unit value. Long edges $l_M(\mathbf{e}) > l_{max}$ are splitted into unit subsegments with respect to M and the new vertices are introduced using a modification of the Deaunay kernel to account for anisotropic elements. Short edges $l_M(\mathbf{e}) < l_{min}$ are collapsed by merging their two endpoints at one of the extremity. A mesh optimization stage involving vertex relocation procedure is then applied to improve the overall mesh quality [18].

4 Geometric Approximation

In the kind of problem we consider, an interface is defined as a manifold of codimension one embedded in $\mathbb{R}^d$. We introduce an implicit level set interface representation, where the interface Γ is the zero isocontour of a Lipschitz-continuous function u. We choose to have a $V_h = \mathbb{P}_1(T_h)$ Lagrange approximation space, and we denote u_h the projection of u on V_h.

4.1 Error Bound

We aim at controlling the approximation error between the exact (unknown) curve Γ and an approximate curve Γ_h defined as a zero isocontour of the function u_h:

$$\Gamma_h = \{x \in \Omega\,,\ u_h(x) = 0\}\,.$$

This control can be obtained by introducing a bound on the Hausdorff distance between Γ and Γ_h, *i.e.*, by prescribing a value of ε sufficiently small such that:

$$\max(\max_{x\in\Gamma}\min_{y\in\Gamma_h}|x-y|, \max_{y\in\Gamma_h}\min_{x\in\Gamma}|x-y|) < \varepsilon\,.$$

Actually, we established the following result that provides an indirect yet practical control on the Hausdorff distance [14]. Suppose u is an implicit real-valued function defined on a domain Ω such that ∇u exists ($|\nabla u| = n$, $n > 0$) and such that its derivatives are k-lipschitz ($k \geq 1$) on an open set. Then, we have the theoretical bounds:

Theorem 1. *Under the previous hypothesis, if we denote the set of elements intersected by Γ as $\mathcal{E} = \{x \in K, K \in T_h$ and $K \cap \Gamma \neq \emptyset\}$, then for all $x \in \Gamma_h$:*

$$d(x,\Gamma) \leq \frac{1}{\min\limits_{p\in\mathcal{E}}|\nabla u(p)|}\left|\sum_{j=1}^{d+1}\lambda_j\left({}^t(x-x_j)(D^2u(x_j)-D^2u(x))(x-x_j)\right)\right|\,,$$

and conversely, for all $x \in \Gamma$,

$$d(x,\Gamma_h) \leq \frac{1}{|\nabla u_h(K)|}\left|\sum_{j=1}^{d+1}\lambda_j\left({}^t(x-x_j)(D^2u(x_j)-D^2u(x))(x-x_j)\right)\right|.$$

where λ_j denotes here the barycentric coordinate of a point x associated with the vertex x_j of K and D^2u represents the Hessian of the function u (the gradient of u_h is piecewise constant per element).

The complete proof of this result can be found in [14] and relies on the analysis of a Taylor expansion with integral rest in any simplex intersected by the desired level set. In the next section, we will show how this result can be used to define a metric tensor field to prescribe appropriate element sizes and shapes to minimize the approximation error in a triangulation.

4.2 A Geometric Metric Tensor

Let us consider the set $\mathcal{E}$ of mesh simplices intersected by the zero level set manifold Γ, *i.e.* such that the sign of the function u is not constant on the element. The following results relate the size of the elements of the set $\mathcal{E}$ to the local curvature(s) of the manifold in two and three dimensions and provide a bound on the distance between the manifold and its piecewise affine approximation.

Corollary 1. *In two dimensions, we have:*

$$d(\Gamma_h,\Gamma) \leq \max_{K\in\mathcal{E}}\frac{l^2\kappa}{1-h\kappa-h^2\kappa^2}$$

where h (resp. l) represents the size of the element K in the normal (resp. tangential) direction to the curve Γ, κ is the maximum of the absolute value of the local curvature to Γ in K.

Corollary 2. *Similarly, in three dimensions, we have:*

$$d(\Gamma_h,\Gamma) \leq \max_{K\in\mathcal{E}}\frac{l_1^2\kappa_1}{1-h\kappa_1-h^2\kappa_1^2}+\frac{l_2^2\kappa_2}{1-h\kappa_2-h^2\kappa_2^2}$$

where h (resp. l_1, l_2) represents the size of the element K in the direction normal (resp. of the principal curvatures) to the manifold and κ_1, κ_2 are the minumum and maximum values of the two local principal curvatures of Γ in K.

Again, the full proof of these results has been given in [14]. These bounds on the approximation error can be conveniently used to define a suitable discrete metric tensor field at each mesh vertex of every element in the set $\mathcal{E}$. Indeed, setting the level of accuracy of the piecewise affine approximation to an arbitrary value ε is equivalent to imposing $d(\Gamma_h,\Gamma) < \varepsilon$. Hence, we propose to define the metric tensor at each vertex as follows:

$$M = \frac{\nabla u \nabla u^t}{h_{min}^2} + \frac{(D^2u)}{\varepsilon} \tag{6}$$

where h_{min} represents the smallest (user-defined) element size in T_h. With this definition, the elements will be aligned with the principal directions of curvatures and with the normal to Γ and their relative sizes will be related to the local curvatures by setting:

$$l_i = \sqrt{\frac{\varepsilon}{\kappa_i}}\,.$$

4.3 Numerical Issues

The geometric metric tensor definition involves the Hessian matrix of the level set function u. Practically, the solution u_h is piecewise affine and only known at the vertices of triangulation T_h. In order to compute the gradient and the Hessian matrices, we propose to reconstruct a high-order solution $\tilde{u} \in C^2(\Omega)$ from u_h in order to write:

$$|u - u_h|_X \approx |\tilde{u} - u_h|_X$$

where $|\cdot|_X$ is the norm related to the approximation space. Notice that in our approach, the solution $\tilde{u}$ does not need to be known, as only the Hessian matrix is required to define the metric tensor.

We propose a least-squares approximation of the second derivative of $\tilde{u}$ based on a Taylor expansion of the solution $\tilde{u}$ at vertex x, following [3]:

$$\tilde{u}(x_i) = \tilde{u}(x) + \mathbf{x}\mathbf{x}_i \cdot \nabla\tilde{u}(x) + \frac{1}{2}\langle \mathbf{x}\mathbf{x}_i, M(x)\mathbf{x}\mathbf{x}_i\rangle$$

where $\tilde{u}(x_i)$ is the solution associated with the vertex x_i connected to the vertex x. This leads to a usually overdetermined linear system $Ay = b$, that is solved using a least square approximation $A^tAy = A^tb$, where y is a $\frac{d(d+1)}{2}$ vector of the Hessian coefficients. Solving it directly allows us to avoid all problems related with ill-conditioned matrices.

We have also investigated the Hessian reconstruction using a dual L^2 projection scheme of the numerical solution u [18]. This robust technique, did not provide better results for our purposes in the numerical simulations, thus we have used the least-square approximations in all following examples.

5 Numerical Examples

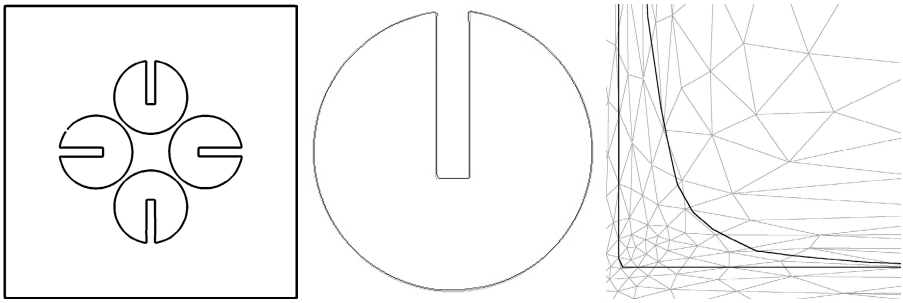
The authors and colleagues have implemented all the numerical techniques described in this paper (the error estimate, the adaptive mesh generation algorithm, the advection and the Stokes flow solvers) in two and three dimensions.

5.1 Numerical Examples in Two Dimensions

In this section, we investigate the efficiency and the accuracy of our interface tracking mesh adaptation method on numerical simulations of Zalesak rigid body rotation and on a viscous flow problem.

Zalesak rigid body

This test case has been originally proposed by Zalesak [36] and we consider here a slightly modified version suggested by Rudman [30] and commonly used for volume integrity checking. The computational domain Ω is here a $[-2,2]\times[-2,2]$ square, the circular body of unit radius is centered at position $(0, 0.75)$ and the width of the slot is 0.12. We consider a steady advection problem where the disk is rotated with a constant angular velocity of 1 time unit, which constitutes a good test of how efficiently the interface is advected in case of high curvatures. The body should return to the initial position after a revolution of 2π, without any modification of the interface shape and volume (Figure 2). To overcome numerical dissipation problems when solving the advection equation, we have used a simple mass-correction procedure. Assuming that the error is constant over the domain, we use an optimization scheme to determine a constant value by which the level set is normalized so as to preserve the total mass.



(a) (b) (c)

Fig. 2. Evolution of the rigid body Zalesak disk: (a) four positions of the rotated body, (b) superposition of the initial and final (after one complete revolution) discretized level set curves , (c) zoom near a singularity

Figure 2 shows the evolution of the rigid body in the velocity field corresponding to angles of rotation $\theta = k\pi/2$, $k \in \{1, \ldots, 4\}$. At each time step, the solution is computed using a high-resolution adapted mesh, for which the smallest prescribed edge length is 0.001 and the maximal prescribed element stretching is given by the ratio between the maximal and the minimal edge length, $0.2/0.001 = 200$. Each anisotropic adapted mesh contains about 6 500 vertices whereas the corresponding isotropic graded mesh (*i.e.*, non uniform) would contain about 10 times more vertices. The generation of each anisotropic mesh corresponding to a rotation angle of $\pi/8$ is obtained using a fixed point algorithm involving 5 metric intersections in time and requires about 46 seconds while the advection time requires 4.3 seconds on a laptop.

The accuracy of the level set approximation can be evaluated by two measures. First, we evaluated the perimeter of the interface and the area of the domain

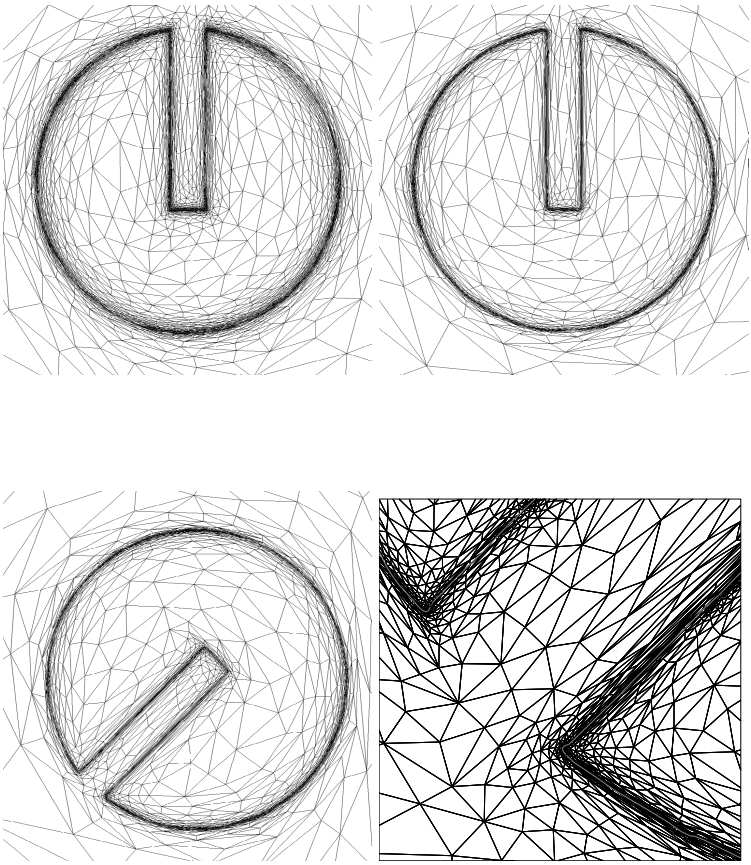


Fig. 3. Zalesak disk test. Advection of the level set corresponding to rotation angles $\theta = 0$, $\theta = 2\pi$ (top) and $\theta = \frac{3\pi}{4}$ with a local enlargement close to the singularities (bootom).

delineated by of the interface during the simulation without and with the mass correction scheme. Without mass correction, the perimeter values are ranging from 4.3275 (initial length) to 4.2985 (final length), corresponding to a shrinkage of less than one percent of the interface curve. The area of the internal domain is equal to 0.71316 (initial) and to 0.706 (final), the area (mass) loss is of the order of one percent. With the mass correction scheme, the final surface is 0.7131677, less than 0.01 percent difference wih the initial surface,but the final perimeter value is 4.450584 (Figure 3).

A viscous flow example

The next example concerns a viscous flow computation. Here, we attempt to solve the Stokes problem (7) describing an incompressible flow of two immiscible

fluids with very different viscosities μ_i on a domain Ω subdivided in two subdomains Ω_i sharing a common interface:

$$\begin{cases} -\mu^i \Delta \mathbf{u}^i + \nabla p^i = \rho^i \, \mathbf{f}^i \\ \text{div } \mathbf{u}^i = 0 \end{cases} \tag{7}$$

where ρ_i and p_i represent the volumic mass and pressure, respectively and with the continuity condition of the velocity field and the equilibrium of the normal constraints with the surface tension at the interface Γ:

$$\begin{cases} \mathbf{u}^1 - \mathbf{u}^2 = \mathbf{0} \\ (\boldsymbol{\sigma}^1 - \boldsymbol{\sigma}^2) \cdot \mathbf{n}^1 = -K \, \mathbf{n}^1 \end{cases} \tag{8}$$

endowed with Dirichlet or Neumann boundary conditions:

$$\mathbf{u} = \mathbf{u}_D \text{ on } \Gamma_D, \quad \sigma \cdot \mathbf{n} = \mathbf{s}_N \text{ on } \Gamma_N, \quad \Gamma_D \cup \Gamma_N = \partial\Omega, \quad \Gamma_D \cap \Gamma_N = \emptyset \tag{9}$$

In the variational formulation, we consider as approximation space of the velocity $\mathbf{u}$ (resp. of the pressure p) the space of Lagrange finite elements $\mathbb{P}_1$-bubble (resp. $\mathbb{P}_1$). Uzawa's method is then used to solve the resulting linear system, in which several equations of the form $AX = G$, A being the stiffness matrix, are solved using a conjugate gradient method. In our approach, the condition number of A depends only on the mesh size h of the triangulation T_h covering the domain Ω. The Uzawa solution $(\mathbf{u}, p)$ of the Stokes problem is now obtained as the limit of a sequence of solutions $(\mathbf{u}_k, p_k)$ when $k \to \infty$ [9]. If this example, the critical part is related to the anisotropic discretization of the interface.

Here, we present the simulation of a Couette flow in a unit square $\Omega = [0,1] \times [0,1]$, a disk of radius 0.1 centered inside the domain corresponds to the domain Ω^1. We considered the following parameter values: viscosities $\mu^1 = 1$, $\mu^2 = 10^{-6}$

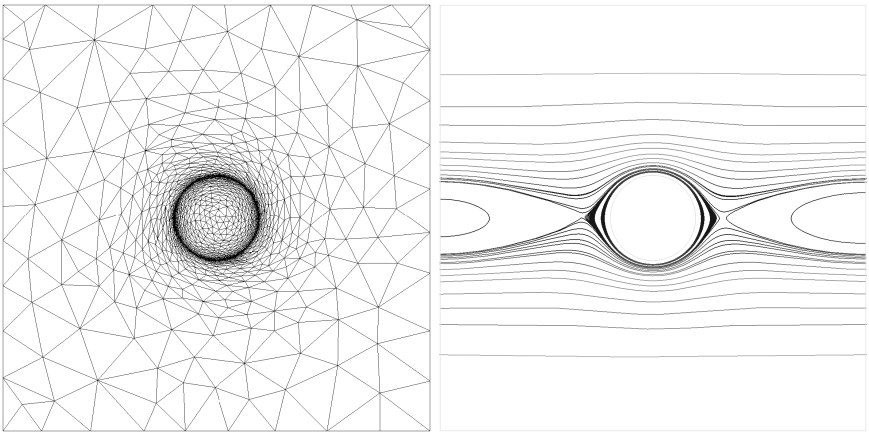
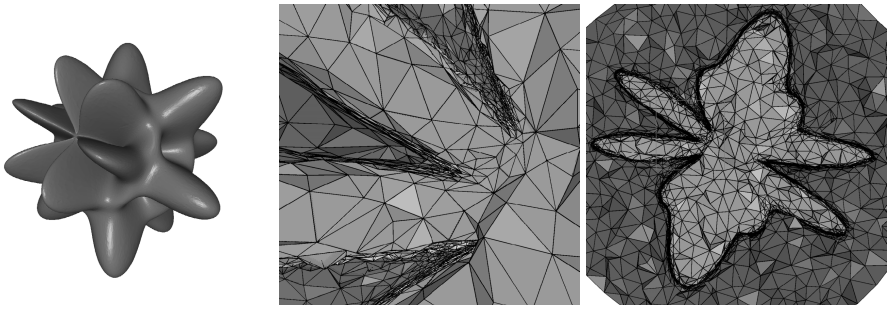


Fig. 4. Mesh of the computational domain (notice the anisotropy in the vicinity of the interface) and streamlines of the velocity field



(a) (b) (c)

Fig. 5. Example of mesh adaptation of a surface embedded in the unit sphere. (a) the reconstructed piecewise affine surface, (b) cutting plane through the tetrahedral mesh and (c) local enlargement of the cutting plane.

and volumic mass $\rho^1 = \rho^2 = 1$. Dirichlet boundary conditions have been prescribed on the external boundaries of Ω: $\mathbf{u}_x = 2(y - 0.5)$, $\mathbf{u}_y = 0$. Figure 4 shows the anisotropic mesh used to perform the simulation and the streamlines of the velocity field.

5.2 A Numerical Example in Three Dimensions

This approach has been implemented in three dimensions for dealing with interface capturing problems. In order to illustrate this feature, Figure 5 shows an anisotropic mesh adapted to a slightly complex although regular analytical surface embbeded in a unit sphere defined in spherical coordinates as:

$$\rho = 0.45 + 0.3\cos(6\theta)\cos^2(3\phi) \qquad \theta \in [0, 2\pi]\,, \phi \in [0, 2\pi]\,.$$

Here, the accuracy level is controlled by the parameter $\varepsilon = 10^{-3}$, leading to a mesh containing $242\,887 < 3.10^5$ vertices. Notice that an isotropic mesh corresponding to the same level of accuracy would contain about 10^9 vertices, 3 to 4 orders of magnitude more vertices than with an anisotropic mesh. The apparent irregularity of the mesh Figure 5 (middle and right-hand side) is due to the cutting plane through tetrahedral elements. Actually, the piecewise affine discretization of the manifold is surprisingly smooth (Figure 5).

6 Conclusions and Perspectives

In this paper, we have presented an efficient method for obtaining very accurate piecewise affine approximations of manifolds of co-dimension one based on the definition of a metric tensor field used to govern the generation of anisotropic meshes. The results obtained so far in numerical simulations are more accurate than similar results obtained using isotropic meshes. The next stage will be to

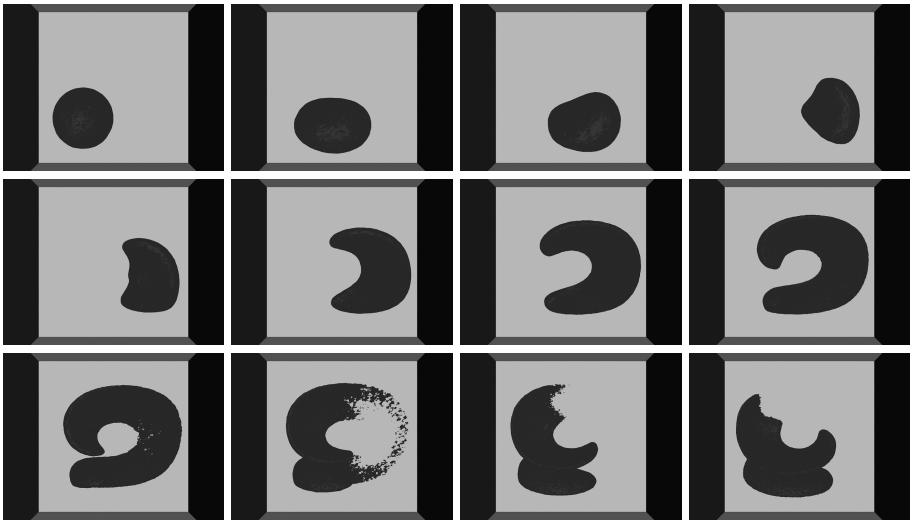


Fig. 6. Preliminary results for capturing accurately a dynamically evolving manifold in three dimensions

handle dynamically evolving interfaces in time-dependent simulations. Figure 6 shows the preliminary results obtained on the deformation of a unit sphere under a deformation field. This approach seems quite successfull in preserving the volume of the level set when the interface undergoes large amount of deformation and stretching induced by an incompressible flow field and can very well compete with methods like particle level sets [15]. By nature, this approach is well suited to maintain regions of high-curvature and the thin surface that developed during the deformation. Less than 10^5 vertices are required to resolve this test case, a number that can be compared with the 100^3 grid generally used in other approaches.

References

1. Abgrall, R.: Numerical discretisation of boundary conditions for first order Hamilton Jacobi equation n triangular meshes. Comm. Pure Appl. Maths. 49, 1339–1373 (1996)
2. Adalsteinsson, D., Sethian, J.A.: A fast level set method for propagating interfaces. Journal of Computational Physics 118(2), 269–277 (1995)
3. Alauzet, F., Frey, P.: Estimateur d'erreur géométrique et métriques anisotropes pour l'adaptation de maillage. Partie I: aspects théoriques, preprint INRIA, RR-4759 (2003)
4. Alauzet, F., Frey, P., George, P.L., Mohammadi, B.: 3D transient fixed-point mesh adaptation for time dependent problems: application to CFD simulations. J. Comp. Phys. 222, 592–623 (2007)

5. Apel, T.: Anisotropic finite elements: local estimates and applications, B.G. Teubner Stuttgart, Leipzig, Series of Advances in Numerical Mathematics (1999)
6. Apel, T., Berzins, M., Jimack, P.K., Kunert, G., Plaks, A., Tsukerman, I., Walkley, M.: Mesh shape and anisotropic elements: theory and practice. In: Whiteman, J.R. (ed.) The Mathematics of Finite Elements and Applications X, pp. 367–376. Elsevier, Oxford (2000)
7. DAzevedo, E.F., Simpson, R.B.: On optimal triangular meshes for minimizing the gradient error. Numer. Math. 59, 321–348 (1991)
8. Babuška, I., Aziz, A.K.: On the angle condition in the finite element method. SIAM J. Numer. Anal. 13, 214–226 (1976)
9. Bui, T.T.C., Frey, P., Maury, B.: Méthode du second membre modifié pour la gestion de rapports de viscosité importants dans le problme de Stokes bifluide. C.R. Acad. Sci. Série I, 524–529 (2008)
10. Crandall, M., Lions, J.L.: Two approximations of solutions of Hamilton-Jacobi equations. Math. Comp. 43, 1–19 (1984)
11. Dervieux, A., Thomasset, F.: Multifluid incompressible flows by a finite element method. Lecture Notes in Physics 11, 158–163 (1981)
12. Dompierre, J., Vallet, M.-G., Bourgault, Y., Fortin, M., Habashi, W.G.: Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part III: unstructured meshes. Int. J. Numer. Meth. Fluids 39, 675–702 (2002)
13. Dijkstra, N.: A note on two problems in connexion with graphs. Numerische Mathematik 1(1), 269–271 (1959)
14. Ducrot, V., Frey, P.: Contrôle de l'approximation géométrique d'une interface par une métrique anisotrope. C.R. Acad. Sci. 345, Série I, 537–542 (2007)
15. Enright, D., Fedkiw, R.: Robust treatment of interfaces for fluid flows and computer graphics. In: Hou, T., Tadmor, E. (eds.) Hyperbolic Problems: Theory, Numerics, Applications, pp. 153–164. Springer, New York (2003)
16. Formaggia, L., Perotto, S.: New anisotropic a priori error estimates. Numer. Math. 89, 641–667 (2001)
17. Fortin, M.: Etude numérique d'estimations d'erreur a posteriori. REEF 9, 467–486 (2000)
18. Frey, P.J., George, P.L.: Mesh generation. Application to finite elements, 2nd edn. Wiley, Chichester (2008)
19. Frey, P., Alauzet, F.: Anisotropic mesh adaptation for CFD computations. Comput. Methods Appl. Mech. Engrg. 194, 5068–5082 (2005)
20. Frey, P.J.: A differential approach to mesh generation. In: Ciarlet, P.G., Li, T. (eds.). Series in Contemporary Applied Mathematics. World Scientific, Singapore (2008)
21. Guèziec, A.: Meshsweeper: Dynamic Point-to-Polygonal-Mesh Distance and Applications. IEEE Trans. Visualization and Computer Graphics 7(1) (2001)
22. Hoch, P., Rascle, M.: Hamilton-Jacobi equations on a manifold and applications to grid generation or refinement. SIAM J. Sci. Comput. 23(6), 2055–2073 (2002)
23. Huang, W.: Metric tensors for anisotropic mesh generation. J. Comput. Phys. 204, 633–665 (2005)
24. Jiang, G.S., Peng, D.: Weighted ENO schemes for Hamilton-Jacobi equations. SIAM J. Sci. Comput. 21, 2126–2143 (2000)
25. Jiao, X., Colombi, A., Ni, X., Hart, J.C.: Anisotropic mesh adaptation for evolving triangulated surfaces. In: Proc. 15th Int. meshing Roundtable, pp. 173–190 (2006)

26. Kim, B., Liu, Y., Llamas, I., Rossignac, J.: Advections with significantly reduced dissipation and diffusion. IEEE Transactions on Visualization and Computer Graphics (2006)
27. Nagrath, S., Jansen, K.E., Lahey, R.T.: Computation of incompressible bubble dynamics with a stabilized finite element level set method. Comput. Methods Appl. Mech. Engrg. 194, 4565–4587 (2005)
28. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. J. Comp. Phys. 79, 12–49 (1988)
29. Osher, S., Shu, C.W.: High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. SIAM J. Numer. Anal. 28, 907–922 (1991)
30. Rudman, M.: Volume tracking methods for interfacial flow calculations. Int. Journal for Numerical Methods in Fluids 24(7), 671–691 (1997)
31. Sethian, J.: Level set methods and fast marching methods. Cambridge University Press, Cambridge (1999)
32. Smereka, P.: Semi-implicit level set method for curvature and surface diffusion motion. Journal of Scientific Computing 19, 439–456 (2003)
33. Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible two-phase flow. J. Comput. Phys. 114, 146–159 (1994)
34. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. IEEE Trans. on Automatic Control 40(9), 1528–1538 (1995)
35. Yamakawa, S., Shimada, K.: High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing. In: Proc. 9th Int. Meshing Roundtable, pp. 263–273 (2000)
36. Zalesak, S.T.: Fully multidimensional flux-corrected transport algorithms for fluids. Journal of Computational Physics 31(3), 335–362 (1979)
37. Zheng, X., Lowengrub, J., Anderson, A., Cristini, V.: Adaptive unstructured volume remshing: II: application to two and three dimensional level-set simulations of multiphase flow. J. Comput. Phys. 208, 626–650 (2005)
38. Zhu, J.Z., Zienkiewicz, O.C.: Superconvergence recovery technique and a posteriori error estimators. Inter. J. Numer. Meth. Engrg. 30, 1321–1339 (1990)