Anisotropic levelset adaptation for accurate interface capturing*

Ducrot V.¹ and Frey P.^{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 numerical simulations, interface capturing and tracking is considered as a very challenging problem and has a strong impact on industrial application. We describe an adaptive scheme based on the definition of an anisotropic metric tensor to control the generation of highly stretched elements near an interface described with a levelset function. The accuracy of the method is verified and various numerical experiments are presented to show its efficiency.

1 Introduction

Since the pionnering works of Dervieux [13] and Osher [28], levelset methods have been successfully used in numerous scientific computing applications. In this context, a very challenging problem 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 computational cost (memory and time resources) of the simulation. In recent years, mesh adaptation and anisotropic features have already proved their efficiency in reducing the number of unknowns while preserving the accuracy of the numerical solution, 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. This issue has been partly addressed by

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

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]. With parameterized interfaces and Lagrangian techniques, accounting for topological changes is almost impossible. On the other hand, levelsets handle quite naturally this situation.

Here, we consider a levelset formulation of an evolution problem. For the sake of simplicity, we consider 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 $u : \mathbb{R}^+ \times \mathbb{R}^d \rightarrow \mathbb{R}$ and the interface Γ is chosen to coincide with the levelset where $u(t, x) = 0$. As suggested by Osher [28], the levelset function is initialized with $u(0, x) = \pm d(x)$, where d is the distance function to the initial interface and sign $\pm$ is negative in Ω^- and positive in Ω^+ .

The motion of the interface is driven by a velocity field $v(t, x) : \mathbb{R}^+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ that may be related to various parameters for instance the geometric properties (normal, curvatures) of the manifold, thus leading to the first-order advection equation posed in $\mathbb{R}^d$, for all $x \in \Omega$, $t \in [0, T]$:

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

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 [11].

From the computational point of view, the domain Ω is covered by an unstructured triangulation T_h and the distance value to the interface is associated with its vertices, thus defining the function u in a discrete manner. High-order schemes (ENO, WENO) have been proposed to resolve the interface evolution on structured grids [29, 26]. However, their extension to arbitrary unstructured triangulations is very difficult and only a few algorithms have been proposed [1]. On the other hand, it is important to control the numerical diffusion, *i.e.* the mass loss or gain, near the interface. In particular, we would like to preserve the interesting property associated with a distance function, $|\nabla u| = 1$ over the domain Ω . This issue is usually addressed via a reinitialisation procedure, that consists in solving for instance the following equation [34]:

$$\partial_t u(t, x) - S(u)(1 - |\nabla u(t, x)|) = 0$$

where $S(u)$ is a sign function taken as 0 at the interface.

The aim of this paper is to present an anisotropic mesh adaptation method, based on a geometric error estimate, for capturing, with a high level of accuracy, an interface described by a levelset function u . We suppose that the function u is known at the vertices of a triangulation T_h and we would like to

generate a triangulation with anisotropic features, *i.e.* with highly stretched elements in the vicinity of the interface. We will show that this can be achieved by defining a Riemannian metric tensor based on the differential properties of the manifold of codimension one corresponding to the interface [16]. Mesh adaptation consists then in creating a quasi-uniform mesh with respect to the metric tensor field, using local mesh modifications. As the interface evolves in time subjected to a velocity flow v , its geometry and topology will then change thus requiring suitable techniques for tracking the interface. In this context, we use a fixed point algorithm that allows us to correctly locate the interface positions at the beginning and at the end of each time step (before and after each advection step). 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. We present here examples of anisotropic mesh adaptation for various levelset functions, for which the interfaces correspond to analytical problems and bi-fluids viscous flow simulations.

Notice also that this approach can be used almost as such with any implicitly defined scalar valued function u . 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 very accurately.

In Section 2, we briefly review the issue of anisotropic mesh adaptation based on Riemannian metric tensor. A complete description of this method can be found in the general purpose book [20] 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 3, we deal with the control of the geometric approximation of an interface and we introduce an anisotropic error estimate for this problem. In Section 4, we briefly review numerical techniques used for solving the advection equation and reinitializing a levelset function. Numerical examples of simulations are given in Section 5 to show the efficiency of this approach.

2 Anisotropic mesh adaptation

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 .

2.1 Definitions

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 $h_K = \text{diam}(K)$ the diameter of K , the size $h = \max_{K \in T_h} h_K$ is given by the relation:

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

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

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

where, for any open ball $B_i \subset K$, $\rho_K = \sup_i \{\text{diam}(B_i)\}$ is the in-diameter of K and (ii) the variation of h is bounded by a constant.

2.2 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 this context, *isotropic* mesh adaptation attempts to adjust only the elements size in order to create regular mesh elements almost everywhere. Usually, an error estimate prescribes the desired element size at any location of the computational domain [19]. In many applications however, 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.

Recently, the theory of error estimates as been largely investigated by various research groups [5, 8, 14, 18, 21] 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].

2.3 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:

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

which corresponds to the discrete formulation

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

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 $\gamma(t) : [0, 1] \rightarrow \mathbb{R}^d$ is a parametrization of γ . As pointed out, the metric M_K associated with an element K is represented by its *unit sphere* $B(K)$ centered at O and defined as [22] :

$$B(K) = \{P \in \mathbb{R}^d, |\mathbf{OP}|_{M_K} = \sqrt{\langle \mathbf{OP}, M_K \mathbf{OP} \rangle} = 1\}.$$

Since the metric tensor $M(x)$ is supposed a symmetric positive definite matrix, the spectral decomposition theorem allows to decompose M as:

$$M = R \Lambda R^t = \sum_{i=1}^d \lambda_i \mathbf{e}_i \mathbf{e}_i^t,$$

where the normalized eigenvectors of M are the columns of matrix $R = [\mathbf{e}_1, \dots, \mathbf{e}_d]$ such that $R R^t = R^t P = I_d$ and Λ is the diagonal matrix of the eigenvalues λ_i . It is obvious to see that the matrix P prescribes the orientation and the matrix Λ prescribes the size and shape of any element K .

2.4 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 assesement 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 edges 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.

2.5 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 case, anisotropic meshes are created using a Delaunay-based method in which a metric tensor prescribing the size, stretching and orientation of the element is associated with the mesh vertices [20]. Moreover for unsteady physical problems, several metric tensor fields must be combined to overcome the numerical diffusion associated with the numerical scheme [4].

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 be 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} . Figure 1 shows the anisotropic mesh used to resolve the internal fixed-point loop.

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 [22].

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 .*

3 Geometric approximation

In the kind of problem we consider, an interface is defined as a manifold of codimension one embedded in $\mathbb{R}^d$. Here, we introduce an implicit levelset 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 .

3.1 Error bound

We aim at controlling the 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 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 [16]. 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:

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

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

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

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

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

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.

3.2 Metric tensor definition

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 minimum and maximum values of the two local principal curvatures of Γ in K .

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:

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

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 Numerical issues

As mentioned in the Introduction, the motion of the interface Γ is driven by a velocity field $v(t, x)$ related to the geometric properties of the manifold or to physical parameters of the problem, through the advection equation (1). Here, we describe the numerical scheme used to solve this Hamilton-Jacobi equation in $\mathbb{R}^d$ and we address the problem of reinitializing the levelset function u .

4.1 Levelset advection

Generally, given a velocity field $v(t, x)$, high-order non oscillatory upwind schemes are employed for solving the Hamilton-Jacobi equation (1), as suggested by [29, 26]. In this context, the levelset function u is used both to represent and to evolve the interface Γ , using an Eulerian formulation. The implementation of this equation may be slightly difficult on a Cartesian grid, if the velocity v is prescribed at the interface only. Although several authors advocate that it is usually sufficient to have v defined on a band surrounding the interface Γ [2], here we will consider the field v to be defined everywhere in the domain Ω .

In order to resolve the velocity field, 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]. In all these approaches, the approximation error is strongly related to the quality of the triangulation and to the mesh density. To this end, the definition of a Riemannian metric tensor (section 3) helps to control the generation of anisotropic elements close to the interface.

From the numerical point of view the resolution of equation (1) raises several issues. Here, we use a finite element technique to solve this problem on

highly anisotropic unstructured meshes. Suppose the vector field $v(t, x)$ is sufficiently smooth, we recall that a *trajectory* (or characteristic curve) associated to the field v is a solution of the differential problem :

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

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 $X(t, x)$. For divergence-free convecting fields, the quantity u is constant along each trajectory. The method of characteristics solves this problem by converting the PDE into a system of ODE. With a finite element scheme, we find an interpolation of the desired solution u using piecewise affine continuous functions. To this end, we introduce the functional approximation space

$$V_h = \{v_h \in C^0(\overline{\Omega}) / v_h|_K \in \mathbb{P}_1, \forall K \in T_h\}.$$

At each time step $t_n = n\Delta t, n \in \mathbb{N}$, we look for the value $u^n = u(t_n, \cdot)$. We can show that:

$$u^{n+1}(x_i) = u^n \circ X^n(x_i) = u^n(X^n(x_i)), \forall i,$$

where x_i is a vertex of T_h and $X^n(x_i)$ is an approximation of $X(t^n, x_i)$. For example, in two dimensions the variational problem reads:

$$\iint_{\Omega} u^{n+1}(x, y) \varphi_i(x, y) dx dy = \iint_{\Omega} u^n \circ X^n(x, y) \varphi_i(x, y) dx dy, \quad (3)$$

with the basis functions φ_i . This leads to solving a linear system of the form: $Au^{n+1} = f$. The latter is solved using a fourth-order Runge-Kutta scheme and a conjugate gradient algorithm.

The main advantage of this approach is that it accounts for large time stepping as compared with other explicit method, since there is no stability condition (*i.e.* no CFL condition) to satisfy. This point will be emphasized in the numerical examples provided in the last section of this paper (see also Figure 1).

4.2 Reinitialization procedure

An interesting property of the 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:

$$\begin{cases} \mathbf{n}(\mathbf{x}) = \frac{\nabla u(x)}{|\nabla u(x)|} = \nabla u(x), \\ H(x) = \operatorname{div}(\mathbf{n}(\mathbf{x})) = \Delta u(x) \\ K(x) = \det(\Pi^t(x) D^2 u(x) \Pi(x)), \end{cases} \quad (4)$$

where Π is the matrix of the projection on the tangent plane at x . Unfortunately, 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.

To this purpose, Sussman et al. [34] propose to solve a non-linear equation at steady state:

$$\partial_t \psi - S(\psi)(1 - |\nabla \psi|) = 0, \quad (5)$$

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, Sethian [32] suggested to use a Fast-Marching method to evolve a front at constant speed initiated from the interface Γ . The algorithm was initially proposed by Tsitsiklis [35] and can be seen as an extension of Dijkstra's algorithm [15] to compute shortest paths between points in Ω . It consists in solving the following equation in each element:

$$\sum_{i=1}^{d+1} |\nabla \phi_i u_i|^2 = 1 \quad (6)$$

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.

Schematically, the corresponding algorithm can be described as follows. Suppose we have three groups of vertices: (A)cccepted, (C)onsidered and (F)ar. Then, proceed as follows:

1. Compute the distance values for the vertices of the elements intersected by Γ , now considered as (A);
2. Put the neighbors of all these points in the group (C) and compute the distance values using Equation (6).
3. Loop over the set (C) until it is empty:
 - Take $p \in (C)$ with the smallest absolute distance value. Move it in set (A);
 - Transfer all neighbors of p from (F) to (C);
 - Update the values of all neighbors of p in (C) using (6).

This is a first order scheme when a linear approximation is used to solve Equation (6).

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 previous 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 very efficient and easily parallelizable manner by an algorithm for computing point distance to polyline, like the one suggested by [23]. 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.

4.3 Hessian approximation

As pointed out, the geometric metric tensor definition involves the Hessian matrix of the levelset 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. In practice, the solution $\tilde{u}$ is not necessary only the Hessian matrix is needed.

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 $A^t Ay = A^t b$, where y is a $\frac{d(d+1)}{2}$ vector of the Hessian coefficients.

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 [37] and we consider here a slightly modified version suggested by Rudman [31] 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).

Figure 2 shows the evolution of the rigid body in the velocity field corresponding to angles of rotation $\theta = k\pi/2$, $k \in \{1, \dots, 4\}$. 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 6500 vertices whereas the corresponding isotropic mesh 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 classical laptop.

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.

We provide two measures of the accuracy of the level set approximation. First, we evaluated the perimeter of the interface during the simulation and the values are ranging from 4.3275 (initial length) to 4.2985 (final length), thus corresponding to a shrinkage of less than one percent of the interface curve. On the other hand, the surface of the internal domain delineated by

Fig. 3. *Zalesak disk test. Advection of the levelset corresponding to rotation angles $\theta = 0$, $\theta = \frac{3\pi}{4}$, $\theta = 2\pi$.*

the curve is initially equal to 0.713 and at completion of one revolution its value is 0.706. Again, the area (mass) loss is of the order of one percent. This result is quite good as no specific mass loss control has been added to the advection procedure.

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 sub-domains Ω_i sharing a common interface:

$$\begin{cases} -\mu^i \Delta \mathbf{u}^i + \nabla p^i = \rho^i \mathbf{f}^i \\ \operatorname{div} \mathbf{u}^i = 0 \end{cases} \quad (7)$$

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 :

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

endowed with Dirichlet or Neumann boundary conditions:

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

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 $k \rightarrow \infty$ [10]. In 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 $\mu^1 = 1$, $\mu^2 = 10^{-6}$ 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.

Fig. 4. Mesh of the computational domain (notice the anisotropy in the vicinity of the interface) and 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 embedded in a unit sphere defined in spherical coordinates as:

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

Here, the accuracy level is controlled by the parameter $\varepsilon = 10^{-3}$, leading to a mesh containing $242887 < 3 \cdot 10^5$ 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, left-hand side).

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.

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 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 successful 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 [17]. 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. R. Abgrall, *Numerical discretisation of boundary conditions for first order Hamilton Jacobi equation on triangular meshes*, Comm. Pure Appl. Maths., **49**

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

- (1996), 1339-1373.
2. D. Adalsteinsson and J.A. Sethian, *A fast level set method for propagating interfaces*, Journal of Computational Physics, textbf118(2), (1995), 269-277.
3. F. Alauzet et P. Frey, *Estimateur d'erreur géométrique et métriques anisotropes pour l'adaptation de maillage. Partie I: aspects théoriques*, preprint INRIA, RR-4759, (2003).
4. F. Alauzet, P. Frey, P.L. George and B. Mohammadi, *3D transient fixed-point mesh adaptation for time dependent problems : application to CFD simulations*, J. Comp. Phys., **222** (2007), 592-623.
5. T. Apel, *Anisotropic finite elements: local estimates and applications*, B.G. Teubner Stuttgart, Leipzig, Series of Advances in Numerical Mathematics, 1999.
6. T. Apel, M. Berzins, P.K. Jimack, G. Kunert, A. Plaks, I. Tsukerman, and M. Walkley, *Mesh shape and anisotropic elements: theory and practice*, in The Mathematics of Finite Elements and Applications X, (ed. J.R. Whiteman), 367376, Elsevier, Oxford, 2000.
7. E.F. D'Azevedo and R. B. Simpson, *On optimal triangular meshes for minimizing the gradient error*, Numer. Math., **59** (1991), 321348.
8. I. Babuška and A. K. Aziz, *On the angle condition in the finite element method*, SIAM J. Numer. Anal., **13** (1976), 214-226.
9. F.J. Bossen and P.S. Heckbert, *A pliant method for anisotropic mesh generation*, Proc. 5th Int. Meshing Roundtable (1996), 6374.
10. T.T.C. Bui, p. Frey and B. Maury, *Méthode du second membre modifié pour la gestion de rapports de viscosité importants dans le problème de Stokes bifluide* C.R. Acad. Sci., Paris, Srie I, (2008), 524-529.
11. M. Crandall and J.L. Lions, *Two approximations of solutions of Hamilton-Jacobi equations*, Math. Comp., **43** (1984), 1-19.
12. B.N. Delaunay, *Sur la sphère vide*, Bul. Acad. Sci. URSS, Class. Sci. Nat. (1934), 793-800.

13. A. Dervieux and F. Thomasset, *Multifluid incompressible flows by a finite element method*, Lecture Notes in Physics, **11** (1981), 158-163.
14. J. Dompierre, M.-G. Vallet, Y. Bourgault, M. Fortin, and W. G. Habashi, *Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part III: unstructured meshes*, Int. J. Numer. Meth. Fluids, **39** (2002), 675-702.
15. N. Dijkstra *A note on two problems in connexion with graphs*, Numerische Mathematik, **1**(1), (1959), 269-271.
16. V. Ducrot, P. Frey, *Contrôle de l'approximation géométrique d'une interface par une métrique anisotrope*, C.R. Acad. Sci. Paris, **345**, Série I (2007), 537-542.
17. D. Enright and R. Fedkiw, *Robust treatment of interfaces for fluid flows and computer graphics*, Hyperbolic Problems: Theory, Numerics, Applications, T. Hou and E. Tadmor (eds), Springer-Verlag, New York, (2003), 153-164.
18. L. Formaggia and S. Perotto, *New anisotropic a priori error estimates*, Numer. Math., **89** (2001), 641-667.
19. M. Fortin, *Etude numérique d'estimations d'erreur a posteriori*, REEF, **9**, (2000), 467-486.
20. P.J. Frey and P.L. George, *Mesh generation. Application to finite elements*, 2nd edition, Wiley, 2008.
21. P. Frey and F. Alauzet, *Anisotropic mesh adaptation for CFD computations*, Comput. Methods Appl. Mech. Engrg., **194** (2005), 5068-5082.
22. P.J. Frey, *A differential approach to mesh generation*, in Series in Contemporary Applied Mathematics, (eds. P.G. Ciarlet, T. Li), World Scientific, 2008.
23. A. Guèziec *"Meshsweeper": Dynamic Point-to-Polygonal-Mesh Distance and Applications*, IEEE Trans. Visualization and Computer Graphics, **7**(1), (2001).
24. Ph. Hoch and M. Rascle, *Hamilton-Jacobi equations on a manifold and applications to grid generation or refinement*, SIAM J. Sci. Comput., **23**(6), (2002), 2055-2073.
25. W. Huang, *Metric tensors for anisotropic mesh generation*, J. Comput. Phys., **204** (2005), 633-665.
26. G.S. Jiang and D. Peng, *Weighted ENO schemes for Hamilton-Jacobi equations*, SIAM J. Sci. Comput., **21** (2000), 2126-2143.
27. S. Nagrath, K.E. Jansen and R.T. Lahey, *Computation of incompressible bubble dynamics with a stabilized finite element level set method*, Comput. Methods Appl. Mech. Engrg., **194** (2005), 4565-4587.
28. S. Osher, J.A. Sethian, *Fronts propagating with curvature-dependent speed : Algorithms based on Hamilton-Jacobi formulations*, J. Comp. Phys., **79** (1988), 12-49.
29. S. Osher and C.W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SIAM J. Numer. Anal., **28** (1991), 907-922.
30. O. Pironneau, "Méthodes d'éléments finis pour les fluides", Masson, Paris, 1988.
31. M. Rudman, *Volume tracking methods for interfacial flow calculations*, Int. Journal for Numerical Methods in Fluids, **24**(7), (1997), 671-691.
32. J. Sethian, *Level set methods and fast marching methods*, Cambridge University Press Cambridge, (1999).
33. P. Smereka, *Semi-implicit level set method for curvature and surface diffusion motion*, Journal of Scientific Computing, **19** (2003), 439-456.
34. M. Sussman, P. Smereka, and S. Osher, *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys., **114** (1994), 146159.

- 35. J.N. Tsitsiklis, *Efficient algorithms for globally optimal trajectories*, IEEE Trans. on Automatic Control, **40**(9), (1995), 1528-1538.
- 36. S. Yamakawa and K. Shimada, *High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing*, Proc. 9th Int. Meshing Roundtable (2000), 263273.
- 37. S.T. Zalesak, *Fully multidimensional flux-corrected transport algorithms for fluids*, Journal of Computational Physics, **31**(3), (1979), 335-362.
- 38. J.Z. Zhu and O.C. Zienkiewicz, *Superconvergence recovery technique and a posteriori error estimators*, Inter. J. Numer. Meth. Engrg., **30** (1990), 1321-1339.