Hamilton-Jacobi Equations on a Manifold and Applications to Grid Generation or Refinement.

Ph. Hoch and M. Rascle * .

Laboratoire de Mathématiques, U.M.R. C.N.R.S. $n°6621$

Université de Nice, Parc Valrose B.P. 71, F06108 Nice Cedex 2, FRANCE

July 26, 1999

Abstract

In this paper, we consider Hamilton-Jacobi equations on a manifold, typically on the graph of some previously computed function z(x,y) and we show how the corresponding level set method allows to generate and/or to refine a mesh in regions where this function z has large derivatives. Such as it is, the method needs to be strongly improved and accelerated, but the principle is awfully natural, and the method is in principle fully automatic. Similar ideas could also be useful in image processing, in particular for the active contours method.

Keywords: Hamilton-Jacobi equations, viscosity solutions, level set method, adaptative meshes, high order numerical 2D schemes

1 Introduction.

The method we are presenting here is fairly general. However, for concreteness, we restrict ourselves to the following prototype: we consider in $I\!R^3$ a manifold Σ, which we assume to be the graph of a smooth function z: $\Omega \to I\!R$:, $\Sigma := \{(x, z(x)); x = (x_1, x_2) \in \bar{\Omega}\}$, where Ω is a bounded domain in $I\!R^2$. The function z is assumed to be known and smooth, but its derivatives can be very large in some regions of $\bar{\Omega}$. The purpose of this paper is to generate a mesh which is automatically refined in regions where the graph of z -in other words, the manifold Σ- is "steep" (in a sense to be precised later).
Practically, z is the (previously computed) numerical solution of some unspecified PDE, on the domain Ω. If no such underlying function z is available, then we can always choose $z \equiv 0$, in which case we will just deal with the classical eikonal equation.
In this paper, we assume that z and its derivatives are known analytically at every point of $\bar{\Omega}$. Of course, in realistic applications, z is interpolated from its values at points of the mesh on which it has been computed, say by a finite element or a finite volume method.
There exists a huge litterature on grid generation and refinement algorithmes, among many others let us mention for instance [31][32][21][17][33][37][22][11][14][16][24][28][29][51] and [12][15][47] in the case of finite differences methods.
The main idea that we propose here is to generate such an adapted grid by solving the Hamilton-Jacobi

[1](hoch@math.unice.fr, rascle@math.unice.fr)

*The Authors acknowledge the support from the EC TMR network HCL : $n°$ ERB FMRX CT 96 0033.

equation on the graph of z, i.e on the manifold Σ, equipped with a suitable Riemannian metric. Indeed, it is well known that the classical eikonal equation:

$$\partial_t \phi(x,t) + V|\nabla \phi(x,t)| = 0 \tag{1.1}$$

is a powerful tool to move curves at a given speed V in the (Euclidean) normal direction.
In order to generate a mesh which is refined in regions where z has large derivatives, it is perfectly natural to slow down the motion of curves in these regions.
The most natural way to do that is to move curves on the graph of z, at a given speed, in the (Riemannian) normal direction, with a suitable metric. In other words, the curves move with constant speed on a stiff mountain, but their horizontal projections move very slowly, so that the resulting mesh is automatically refined in these regions.
Although the principle is very general, the two natural metrics that we essentialy consider here are based on the positive definite quadratic forms associated either to the matrix:

(i)

$$G_1(x) = \begin{pmatrix} 1 + (\frac{\partial z}{\partial x_1})^2 & \frac{\partial z}{\partial x_1}\frac{\partial z}{\partial x_2} \\ \frac{\partial z}{\partial x_1}\frac{\partial z}{\partial x_2} & 1 + (\frac{\partial z}{\partial x_2})^2 \end{pmatrix}, \tag{1.2}$$

(ii) or to

$$G_2(x) = I +^t H(z)(x) H(z)(x), \qquad \mathrm{H(z)(x)} := (\frac{\partial^2 \mathrm{z}}{\partial \mathrm{x_i} \partial \mathrm{x_j}})(\mathrm{x}) \ : \ \text{Hessian matrix of z} \tag{1.3}$$

Let us briefly comment these two particular (strongly anisotropic) choices:
With the first choice, in regions where $|\nabla z| >> 1$ the Riemannian distance between two neighbor points A and B, see Fig 1, is much bigger than their Euclidean distance, except if vectors $\vec{AB}$ and ∇z are orthogonal, in other words if A and B are essentially on the same level curve of z. Indeed, the eigenvector of $G_1(x)$ associated to the larger eigenvalue $\lambda_2(x) = 1 + |\nabla z(x)|^2$ is $\nabla z(x)$, whereas the eigenvector associated to $\lambda_1(x) = 1$ is $(\nabla z)^{\perp}(x)$.
Consequently, with this first choice, the motion will be much slower:

1. in regions where the (Euclidean) norm $|\nabla z|$ is large,
2. inside these regions, in the direction of ∇z. For instance, locally, we can have as in Figure 1 $d^{G_1}(A,B) = d^{G_1}(A,C)$ if $\nabla z(A) >> 1$.
   In contrast, an isotropic metric defined for instance by $G_3(x) = F(|\nabla z(x)|)I$ see e.g. [51] would lead to $d^{G_3}(A,B) >> d^{G_3}(A,C) \simeq d^{G_3}(A,B')$ which is in principle inappropriate in this context.

With the choice (1.3), the motion is slowed down in regions where the second order derivatives of z are large, regardless of the size of the gradient. The motivation for that is very natural, see e.g. the above-mentionned references [31][32][21][17]. Indeed, for example in the context of P1 finite element approximations, there is no need to refine a mesh in regions where the gradients are large, if the solution is essentially linear, like in region BC in Figure 2 (in the 1D case).
In such a case, the mesh should only be refined in intervals AB and CD. In the multi-dimensional case this simple remark is reflected in the error estimates for the finite element method, see again the above references. In particular, in 2D, these error estimates are optimal when the triangles are almost equilateral with respect to a metric like G_2. For related observations, see e.g. [33].

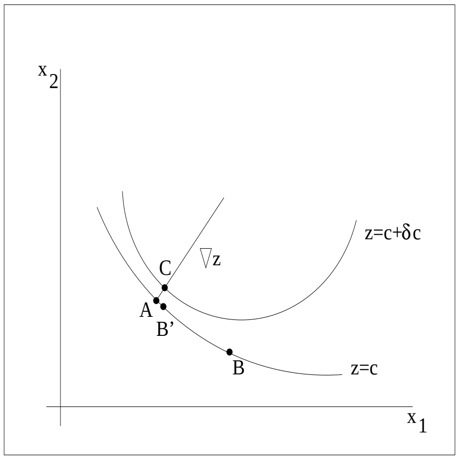


Fig 1

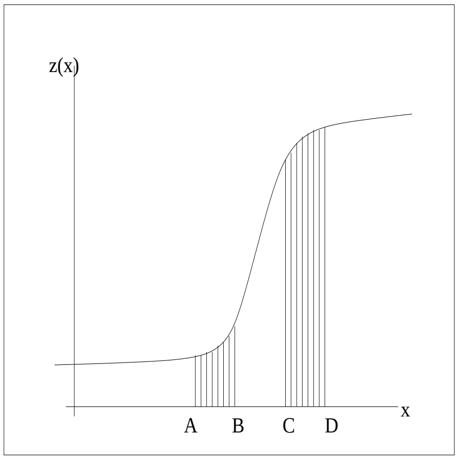


Fig 2

The approach described here in the case of scalar-valued function z of two variables is in principle completely general: z could be any vector or (tensor)-valued function, defined on a domain in $I\!R^3$. This function z could even be defined on a two dimensional manifold $M \in I\!R^3$, say the wing of an airplane, and the goal is then to build a "good" mesh, automatically refined near by singularities of the flow (transonic shocks, geometric singularities of the wing ... etc). In this view, the purpose would be to generate the mesh on the boundary and then to march in time in order to generate the fully 3D mesh.

With these ideas, our method turns out to be a variant of the classical level set methods [45][49][38][44]. Naturally, quite similar problems arise in image processing [3], in particular in active contours (snakes) [20] etc. In the above-mentioned references, several Authors deal with very similar problems, see e.g. [38], but in general the motivation is different. As far as we know, this very simple idea of generating a refined mesh by solving the Hamilton-Jacobi equation on a manifold has never been explicitly expressed.

A last comment is that the results presented here are far from being optimal. As we will see later, there are still many imperfections in generating the grid. Our purpose here is only to show that this strategy is natural, and in principle tractable, even though the first version of the algorithm described here needs to be improved.

The outline of the paper is as follows: In Section 2, we recall a few basic facts on Hamilton-Jacobi (HJ) equations. In Section 3, and 4, we briefly describe the numerical schemes that we have used, and we show on a few examples the level curves of the corresponding viscosity solution. In Sections 5 and 6, we describe the principle of the construction of a triangular mesh based on these level curves, and we show a few numerical results. Finally, we recall the definition of viscosity solutions to HJ equation in Section 7.

2 The Hamilton-Jacobi equations on a manifold.

2.1 Basic facts

Although the method is fairly general, for simplicity we restrict ourselves to the typical situation described in Section 1: Ω is a bounded domain in $I\!R^2$, and Σ is the graph of a smooth function z: $\Omega \to I\!R$, equipped with a Riemannian metric: at every point $(x, z(x))$ of Σ, we are given a positive definite symmetric bilinear form g(x;.,.): in the coordinates system $x = (x^1, x^2)$, the length of dx is defined by:

$$L(x,dx) := |dx|_{G(x)} = (dx, G(x)dx)^{\frac{1}{2}} = (g_{ij}(x)dx^i dx^j)^{\frac{1}{2}} = |G(x)^{\frac{1}{2}}dx|, \tag{2.1}$$

with the Einstein summation convention. Here, $G(x) = (g_{ij}(x))_{1\le i,j\le 2}$, $(u,v) := \delta_{ij}u^i v^j$ is the Euclidean scalar product in $I\!R^2$, and $|u| = (u,u)^{\frac{1}{2}}$. The corresponding length of a curve $\Gamma := \{(\gamma(t), z(\gamma(t))), t \in [0,T]\}$ on Σ is then:

$$\mathrm{L}(\Gamma) = \int_0^T L(\gamma(t), \dot\gamma(t))dt := \int_0^T (\dot\gamma(t), G(\gamma(t))\dot\gamma(t))^{\frac{1}{2}}dt \tag{2.2}$$

Now, let X(x):=(x,z(x)) and X(y) be given points on Σ. A (minimal) geodesic curve on Σ is a curve Γ for which L(Γ) is extremal (minimal) among all curves on Σ with the same endpoints. Under standard assumptions, there exists a minimal C^1 geodesic curve between two arbitrary points X(x) and X(y) on Σ, and this curve is unique if these points are close enough cf e.g. [55].
If the parameter t is the arc length, then minimizing L(Γ) is equivalent to minimizing the energy:

$$\mathrm{E}(\Gamma) := \int_0^T (L(\gamma(t), \dot\gamma(t)))^2 dt \tag{2.3}$$

with the same boundary conditions, see e.g. [27]. Here and often in the paper we allow ourselves the confusion between curves on Σ and their horizontal projections in Ω.

2.2 The variational problem

Now, let $C_0 := \{(x, z(x)), x \in c_0\}$ be a curve on Σ where c_0 is a given (smooth) curve in $\bar\Omega$, parametrized e.g by its arc length s, let x be a nearby point in Ω, and let us consider the problem:

$$\left\{ \begin{array}{lrcl} \text{Find y} \in \text{c}_0 \text{ such that} & d^G(X(x), X(y)) & = & \text{Min}\left\{d^G(X(x), X(y')); y' \in c_0\right\} \\ & & = & d^G(X(x), C_0), \end{array}\right. \tag{2.4}$$

where $d^G(X(x), X(y)) := Inf\left\{\int_0^T L(\gamma(t), \dot\gamma(t))dt; \gamma(0) = y, \gamma(T) = x\right\}$.

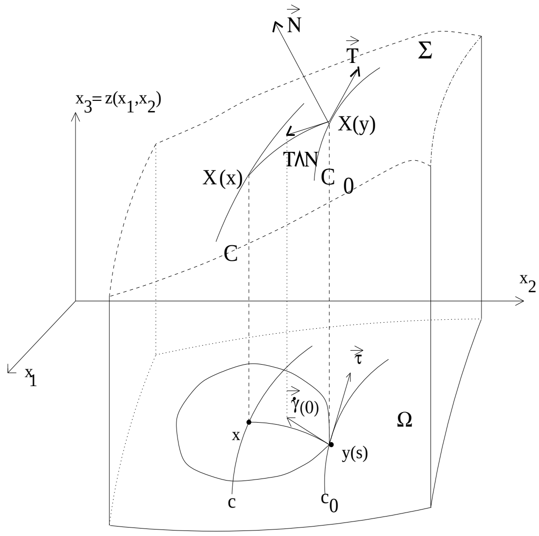


Figure 3

Clearly, cf e.g. [55], for the case of geodesics with fixed endpoints, if C_0 is smooth and if $d(X(x), C_0)$ is small enough, then there exists a unique solution to (2.4), characterized by the Euler-Lagrange equations:

$$-\frac{\mathrm{d}}{\mathrm{d}t}(\frac{\partial L}{\partial \dot{\gamma}}) + \frac{\partial L}{\partial \gamma} = 0, \tag{2.5}$$

$$(\frac{\partial L}{\partial \dot{\gamma}}(\gamma(0), \dot{\gamma}(0))|\tau) = 0, \tag{2.6}$$

where $\tau = \tau(\mathrm{y})$ is the tangent vector to c_0 at $y = \gamma(0)$. Naturally, we can again restrict ourselves here to only considering the horizontal projection.

$$\frac{G(\gamma(0))\dot{\gamma}(0)}{|G(\gamma(0))\dot{\gamma}(0)|} = \pm\mathbf{n} \tag{2.7}$$

where $\mathbf{n} = \mathbf{n}(\gamma(0))$ is one of the two possible unit normal vectors, in $\Omega \subset I\!R^2$. Note that the vector $\dot{\gamma}(0)$ is not normal to c_0, but this vector is the horizontal projection of the *normal geodesic*, cf e.g. [49][38]:

$$\dot{\gamma}(0) = \pm|\dot{\gamma}(0)|\mathbf{T} \wedge \mathbf{N}. \tag{2.8}$$

Naturally, in practical situations, the horizontal projection $\Omega \subset I\!R^2$ could be replaced by a two dimensional manifold $M \subset I\!R^3$, e.g. the wing of an airplane.
Geometrically, (2.6) says that the "ball" of center x and radius $d^G(X(x), X(y))$ is tangent to the curve c_0 at y, see Figure 3.

2.3 The Hamilton-Jacobi equations:

Now, the purpose is to move the curve C_0 on Σ, or equivalently the curve c_0 in $\Omega \in I\!R^2$ at a given Riemannian speed V, by the level set method see e.g. [8] [45] [49][23][34]. For simplicity, we restrict ourselves to the case $V \equiv 1$.
In other words, we define c_0 as the horizontal projection of the zero level curve of a function $\Phi(x, 0) = \Phi_0(x) : \Omega \to I\!R$, and we look for a function $(x, t) \mapsto \Phi(x, t)$ such that, for any $t \geq 0$,

$$c_t := \{x \in \Omega; \Phi(x, t) = 0\} \tag{2.9}$$

A formal first order Taylor expansion gives:

$$0 = \Phi(x, t) = \Phi(y, 0) + t\partial_t\Phi(y, 0) + \nabla\Phi(y, 0).(x - y) + o(|t| + |x - y|). \tag{2.10}$$

Now, at first order,

$$x - y = t\dot{\gamma}(0) + o(t) = \pm t\frac{G^{-1}(y)\mathbf{n}}{|G^{-1}(y)\mathbf{n}|_G} + o(t), \tag{2.11}$$

where the speed V is a given nonnegative scalar function. Since we have assumed that $V \equiv 1$, and since the vector $\nabla\Phi$ is parallel to $\mathbf{n}$, we obtain:

$$x - y = t\dot{\gamma}(0) + o(t) = \pm t\frac{G^{-1}(y)\nabla\Phi}{|G^{-1/2}(y)\nabla\Phi|}, \tag{2.12}$$

where

$$\begin{aligned}|G^{-1/2}(y)\nabla\Phi| &:= (G^{-1}(y)\nabla\Phi, G(y)G^{-1}(y)\nabla\Phi)^{1/2} \\ &= (G^{-1}(y)\nabla\Phi, \nabla\Phi)^{1/2}\end{aligned} \tag{2.13}$$

Finally, we obtain

$$0 = \Phi(x,t) = 0 + t(\partial_t\Phi \pm (G^{-1}(y)\nabla\Phi, \nabla\Phi)^{\frac{1}{2}}) + o(t)$$

Therefore, the function Φ must satisfy :

$$\partial_t\Phi \pm (G^{-1}(y)\nabla\Phi|\nabla\Phi)^{\frac{1}{2}} = \partial_t\Phi \pm |G^{-1/2}(y)\nabla\Phi| = 0 \tag{2.14}$$

Choosing the sign of the initial datum Φ_0 on each side of c_0 imposes the direction of propagation, and therefore suppresses the above ambiguity. From now we choose the positive sign in (2.14). Therefore the Hamiltonian

$$H(x,p) := |G^{-1/2}(x)p| \tag{2.15}$$

is convex with respect to p.

2.4 The stationary problem :

Now, Ω is a bounded domain in $I\!R^2$, with a (theoretically) smooth boundary $c_0 := \partial\Omega$. In fact, what we would like to compute is the family of level curves:

$$\tilde{c_t} := \left\{x \in \Omega; d^G(x, \partial\Omega) = t, t \geq 0\right\} \tag{2.16}$$

(we will see below that curves $\tilde{c_t}$ and c_t are the same), or equivalently the family of curves

$$\tilde{C_t} := \{(x, z(x)); x \in \tilde{c_t}\}. \tag{2.17}$$

In the case of the Euclidean metric it is a classical result, see e.g. [6][40][25],that $\chi(x) := d(x, \partial\Omega)$ is a *viscosity solution* to the stationary equation

$$|\nabla\chi(x)| = 1 \qquad \text{in } \Omega, \tag{2.18}$$

and satisfies in the classical sense the Dirichlet boundary condition

$$\chi(x) = 0 \qquad \text{on } \partial\Omega. \tag{2.19}$$

In fact, χ is the unique *viscosity solution* to (2.18) which satisfies (2.19). The definition of viscosity solutions are recalled below, see Section 7. Here, the metric G(x) depends smoothly on x, and it is easy to adapt the above result, to show that:

Proposition 2.1. $\Psi(x) := d^G(x, \partial\Omega) := \inf_{y\in\Omega} d^G(x,y)$ *is the unique viscosity solution to problem:*

$$|G^{-1/2}(x)\nabla\Psi(x)| = 1 \qquad \text{in } \Omega, \tag{2.20}$$

which satisfies in the classical sense the boundary condition

$$\Psi(x) = 0 \qquad \text{on } \partial\Omega. \tag{2.21}$$

2.5 The initial boundary-value problem.

Naturally, the stationary solution to (2.20), (2.21) is unknown, see e.g. [48] for computational aspects. The principle of the *level set method*, [10][8][45] is to compute the family of curves (c_t) defined by (2.16), with suitable initial data and boundary conditions. Indeed, the family of curves (c_t) depends on the initial curve $c_0 = \partial\Omega = \{x \in \bar{\Omega}; \Phi_0(x) = 0\}$, and *not* on the particular solution Φ to:

$$\partial_t \Phi + |G^{-1/2}(x)\nabla\Phi(x)| = 0 \qquad \text{in } Q = \Omega \times (0,T). \tag{2.22}$$

More precisely, we choose an initial datum Φ_0 such that:

$$c_0 := \partial\Omega := \{x \in \bar{\Omega}; \Phi_0(x) = 0\}, \tag{2.23}$$
$$\Omega_0 := \Omega := \{x \in \bar{\Omega}; \Phi_0(x) > 0\}, \tag{2.24}$$

and we impose:

$$|G^{-1/2}(x)\nabla\Phi_0(x)| = 1 \qquad \text{on } \partial\Omega. \tag{2.25}$$

The choice of "optimal" boundary data is a subtle question, which in particular raises in general the issue of discontinuous viscosity solutions see again [6][1][13]. We recall that the family (c_t) is called "regular" if relations similar to (2.23), (2.24) between c_t and Ω_t holds for all $t > 0$, cf [10][8]. Here, using the independence of level curves c_t with respect to the particular solution Φ to (2.22), we follow the simplest choice:

$$\Phi(x,t) = -t \qquad \text{on } \partial\Omega \times (0,t). \tag{2.26}$$

The motivation for this choice is the compatibility with the obvious solution:

$$\tilde{\Phi}(x,t) = \Psi(x) - t = d^G(x,\partial\Omega) - t \tag{2.27}$$

to Problem (2.22),(2.26) with the inital data:

$$\tilde{\Phi}(x,0) = \Psi(x), \tag{2.28}$$

which of course, satisfies conditions (2.23), (2.24),(2.25). With this choice it is clear that curves c_t and $\tilde{c_t}$ are the same. Of course, solving the problem (2.22),(2.26),(2.28) is purely theoretical, since we do not know explicitly the function Ψ.
Therefore, we solve in fact problem (2.22),(2.26),(2.29), with :

$$\Phi(x,0) = \Phi_0(x). \tag{2.29}$$

Adapting [40], prop 11.1, and standard results of uniqueness, [6][40], we obtain:

Proposition 2.2. *For any Lipschitz function Φ_0 satisfying conditions (2.23),(2.24),(2.25), the initial boundary value problem (2.22),(2.26),(2.29) has a unique viscosity solution Φ, whose level curves $\{c_t, t \geq 0\}$ do not depend on the particular choice of boundary conditions and initial data.*

Morevover, consider: $\Omega_0 := \Omega$ and $\Omega_t := \{x \in \Omega; d^G(x,\partial\Omega) > t\}$, then Φ is in principle given by the Oleinik-Lax [43][39][41] or Hopf [35][5][41][7] type of formula:

$$\Phi(x,t) = \begin{cases} \operatorname{Inf}\{\Phi_0(y); d^G(y,x) \leq t\}, & \text{if } x \in \bar{\Omega}_t, \\ d^G(x,\partial\Omega_0) - t, & \text{if } x \in \Omega \backslash \bar{\Omega}_t. \end{cases} \tag{2.30}$$

For related results with the classical eikonal equation, see e.g. [7],[5],[41] and the references therein. For possible pathologies in the case $\Omega = I\!R^n$, see [18] and [53]. We will come back elsewhere to a more precise statement.
We just remark here that, since Ω is bounded, the function Φ given by (2.30) converges in finite time to the stationary solution Ψ defined in Proposition 2.1, and has obviously the same level curves $\{x \in \Omega; d^G(x, \partial\Omega) = t\}$.

Remark: We also recall, see e.g. [30], that (2.25) is necessary to define locally the *bicharacteristics*, if the "initial" condition (2.21) for the stationary problem (2.20) is (only) given on curve $\partial\Omega$.

3 Numerical Schemes

We recall that (2.22) is hyperbolic in $\nabla\Phi = U = (u, v)$. We adopt the Osher-Shu approach [46], in other words, we use a cartesian grid $(x_i, y_j) = (i\Delta x, j\Delta y)$, and we first semi-discretize (2.22) to obtain:

$$\frac{\partial}{\partial t}\Phi_{i,j}(t) = -h^{num}(x_i, y_j, u^-_{ij}, u^+_{ij}, v^-_{ij}, v^+_{ij}) \simeq -H(x_i, y_j, \nabla\Phi(x_i, y_j)), \tag{3.31}$$

where $U^{\pm}_{ij} := (u^{\pm}_{ij}, v^{\pm}_{ij})$ are approximations to $\nabla\Phi(x_i, y_j)$, and h^{num} is simply a two dimensional continuous, monotone Riemann Solver. Let us describe more precisely the three classical steps in the algorithm.

3.1 The interpolation

At point (x_i, y_j), we first construct either an ENO or a TVD polynomial interpolation of Φ. For instance, a TVD interpolation of first order is:

$$\begin{aligned} u^- &= D^{-x}_{ij} = \frac{\Phi_{i,j} - \Phi_{i-1,j}}{\Delta x}, \qquad & u^+ &= D^{+x}_{ij} = \frac{\Phi_{i+1,j} - \Phi_{i,j}}{\Delta x}, \\ v^- &= D^{-y}_{ij} = \frac{\Phi_{i,j} - \Phi_{i,j-1}}{\Delta y}, \qquad & v^+ &= D^{+y}_{ij} = \frac{\Phi_{i,j+1} - \Phi_{i,j}}{\Delta y}. \end{aligned}$$

or [45] a second order TVD interpolation with a minmod limiter is:

$$\begin{aligned} u^- &= D^{-x}_{ij} + \frac{\Delta x}{2}\, m(D^{-x-x}_{ij}, D^{+x-x}_{ij}), \qquad & u^+ &= D^{+x}_{ij} - \frac{\Delta x}{2}\, m(D^{+x+x}_{ij}, D^{+x-x}_{ij}), \\ v^- &= D^{-y}_{ij} + \frac{\Delta y}{2}\, m(D^{-y-y}_{ij}, D^{+y-y}_{ij}), \qquad & v^+ &= D^{+y}_{ij} - \frac{\Delta y}{2}\, m(D^{+y+y}_{ij}, D^{+y-y}_{ij}). \end{aligned}$$

We will essentially use a second order ENO interpolation, see [46].

3.2 Numerical Hamiltonian

Knowing the $(u, v)^{\pm}_{ij} = \nabla\Phi(x_i, y_j) + O(\Delta x^r)$ in regions where the solution is smooth, a way to choose a robust approximation of $H(x_i, y_j, \nabla\Phi_{ij})$ is to evaluate this quantity with a numerical hamiltonian h^{num}:

$$\frac{\partial}{\partial t}\Phi_{i,j}(t) = -h^{num}(x_i, y_j, u^-, u^+, v^-, v^+).$$

We choose the Lax-Friedrichs numerical hamiltonian:

$$h_{LF}^{num}(x,y,u^-,u^+,v^-,v^+) := H(x,y,\frac{u^- + u^+}{2},\frac{v^- + v^+}{2}) - \frac{\alpha}{2}(u^+ - u^-) - \frac{\beta}{2}(v^+ - v^-), \qquad (3.32)$$

where

$$\alpha = \sup_{u,v} |\frac{\partial H(x,y,u,v)}{\partial u}|, \qquad \beta = \sup_{u,v} |\frac{\partial H(x,y,u,v)}{\partial v}|.$$

we note that e.g in (1.2) or (1.3) the eigenvalues of G(x) are larger than 1. Therefore in (2.15), the eigenvalues of $G^{-1/2}$ are smaller than 1, so that it is easy in (3.32) to guarantee the CFL condition $(\Delta t(\frac{1}{\Delta x} + \frac{1}{\Delta y}) \leq 1)$ uniformly in (x,y).

3.3 Time discretization

We now discretize the ordinary differential equation: $\frac{\partial}{\partial t}U = L(U)$. Here, we have used the Total Variation Diminishing Runge-Kutta (TVD) second order method of Osher, Shu [46].

$$\begin{cases} U^{(0)} = U^n, \\ U^{(i)} = \sum_{j=0}^{i-1} \alpha_{ij}\, U^{(j)} + \beta_{ij}\, \Delta t\, L(U^{(j)}) \qquad i = 1...s = 2, \\ U^{n+1} = U^{(s)}. \end{cases} \qquad (3.33)$$

We recall the principle of this approximation: if any spatial stability is satisfied for the Euler time approximation scheme, e.g. $TV(U^n + \Delta t L^x(U^n)) \leq TV(U^n)$, then the same stability holds for (3.33), modulo $C\Delta t$. An easy way to obtain such a scheme is to impose $\forall(i,j)$: $(\alpha_{ij}, \beta_{ij}) \geq 0$, since then (3.33) is a convex combination of Euler steps, with Δt replaced by

$$\left(\min_{i,j}(\beta_{ij}\alpha_{ij}^{-1})\right) \Delta t.$$

3.4 The boundary conditions

As in Section 2, we impose the boundary condition (3.35), and for second order schemes, we add near by $\partial\Omega$ one layer of ghost points (x_i, y_j) out of Ω, at a distance of order Δx, and we define the values $\Phi_{i,j}^n$ at these ghost points at time $t_n = n\Delta t$ in order to satisfy a discrete approximation of (2.27). In fact, let $x = x_{i,j}$ be such a ghost point. We first define the signed distance:

$$\tilde{\Psi}(x) := \begin{cases} \Psi(x) = d^G(x,\partial\Omega) & \text{in } \Omega, \\ -d^G(x,\partial\Omega) & \text{out of } \Omega. \end{cases}$$

Now, let y be (an approximation of) the normal projection of x on $\partial\Omega$:

$$\text{x-y} = \text{- d(x,}\partial\Omega\text{)}\ \mathbf{n}(\text{y}) = \text{- d}\ \mathbf{n}(\text{y}),$$

were d is the Euclidean distance. Since $\mathbf{n} = \mathbf{n}(y) = \frac{\nabla\Psi(y)}{|\nabla\Psi(y)|} = \frac{x-y}{|x-y|}$ is the inward normal vector to $\partial\Omega$, and since the Hamiltonian H in (2.15) is homogenous of order one, we have:

$$|\nabla\Psi(y)| = (H(y,\mathbf{n}(y)))^{-1}.$$

We approximate $\Psi(x)$ by:

$$\Psi_h(x) \;=\; \Psi(y) + \nabla\Psi(y).(x-y) = -d\,|\nabla\Psi(y)| = -\frac{d(x,\partial\Omega)}{H(y,\mathbf{n}(y))}. \qquad (3.34)$$

In view of (2.27) we define the approximate boundary data:

$$\Phi_h(x, t_n) := \quad \Psi_h(x) - t_n. \tag{3.35}$$

Remark:
As we already said, this choice is not completely satisfactory, especially if $\partial\Omega$ is only piecewise smooth. However, we already noted that the family of c_t is independent from this choice. For a discussion of more refined boundary conditions, we refer to [9][1]. In fact, at time t, we are only interested in solving the problem in $\Omega_t = \{x \in \Omega, d^G(x, \partial\Omega) \geq t\}$, and for this problem the boundary condition at this time is:

$$\Phi(x, t) = \Psi(x) - t = 0 \qquad \text{on} \;\; \partial\Omega_t. \tag{3.36}$$

This remark is the basis of the fast marching method developped by Sethian *et al*, see [49] [50] etc ...

3.5 The initial data

In the same spirit, we define the initial condition $\Phi(x, 0)$ for x in Ω by the approximation:

$$\Phi_h(x, 0) := \Psi_h(x) = +\frac{d(x, \partial\Omega)}{H(y, \mathbf{n}(y))}, \tag{3.37}$$

Where $d(x, \partial\Omega)$ is again the Euclidean distance of x to $\partial\Omega$, which in our case is easy to calculate. In order to illustrate the role of the denominator in (3.37), we show in Fig 4a,4b the successive level curves of the solution when the metric G(x,.) involves a boundary layer. In Fig 4b, where we have dropped the denominator in (3.37), the level curves almost do not detect the boundary layer.

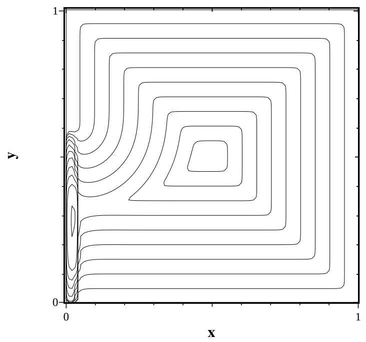


Figure 3a

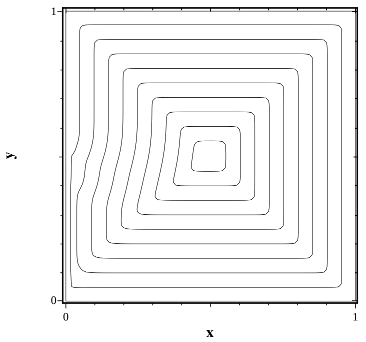


Figure 3b

4 Level curves

We have tested the above-mentioned algorithm on several examples:

(i) Ω is either a square or a circle. In the first case, note the shocks -sometimes smoothed out by the numerical viscosity- due to the corners of the square. Of course this difficulty disappears in the case of a circle.

(ii) The involved Riemannian metric is associated to the matrix: $G(x) = V^{-2}(x)G_i(x)$, i=1,2, so that the speed is multiplied by V(x). Here, V(x) is a given positive scalar speed, adjusted to impose a minimal speed Vmin if necessary, say V(x)= Max(h(x),Vmin) where

$$h(x) = \begin{cases} h1(x) := (\mathrm{detG(x)})^{-1/2} & \text{or} \\ h2(x) := \exp(1 - (\mathrm{detG(x)})^{10}), \end{cases} \tag{4.38}$$

and

$$G_1(x) = I + \nabla z \otimes \nabla z \qquad ; \ G_2(x) = I + {}^tH(z)H(z), \tag{4.39}$$

(iii) This function z is a given (combination of) smooth real valued function(s), with large first and second order derivatives, which is analytically known. Typically, see Figures 4a,4b,4e,4h below,

$$z(x) = z(x_1, x_2) = K_1 \, atan \left(K_2 f(x_1, x_2)\right), \tag{4.40}$$

where K_1 and/or K_2 are constants, adjusted to produce sharp fronts near by the curve $\{f(x_1, x_2) = 0\}$. In other cases, see Fig 4f,4g, $z(x) = z(x_1, x_2)$ is a regularized delta-function:

$$z(x) = \delta_\epsilon(|x - x_0|) = \frac{C}{\epsilon}\delta_1(\frac{|x - x_0|}{\epsilon}), \tag{4.41}$$

or a tensor-product $z = z_1 \otimes z_2$,

$$z_i(x) = C_i \delta_{\epsilon_i}(a_i x_1 + b_i x_2 + c_i), \tag{4.42}$$

see Fig 3a,4d,4e. In (4.41), $\delta_1(s) \geq 0$, $\int_{I\!R} \delta_1(s)ds = 1$ and $\delta_1(s) \equiv 0$ for $|s| \geq 1$.
The last two examples are chosen to simulate a sort of boundary layer or some oblic singularity near by the boundary.

(iv) We have used either first or second order ENO-TVD methods described in Section 3. Typically, the CFL condition is about 0.5.

(v) In all cases, we have used a second order ENO interpolation with second order TVD time discretisation with nx=ny=101, and at the top of each figure, we have noted the corresponding metric G, speed h and the minimum speed Vmin.

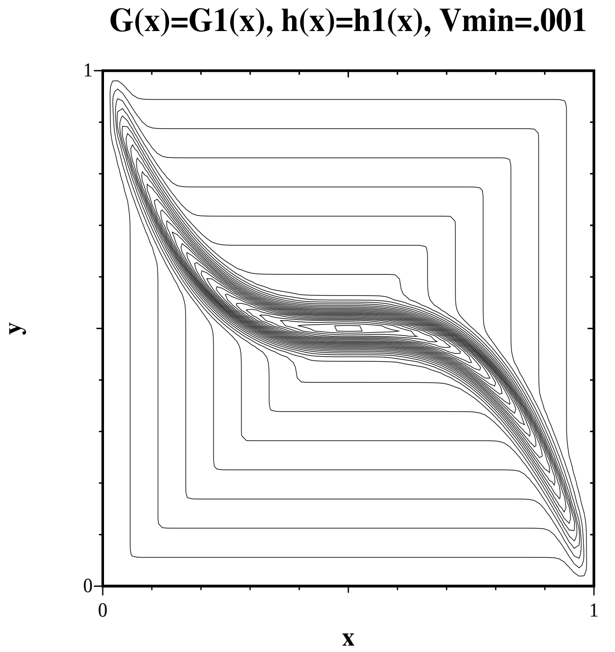


Fig 4a

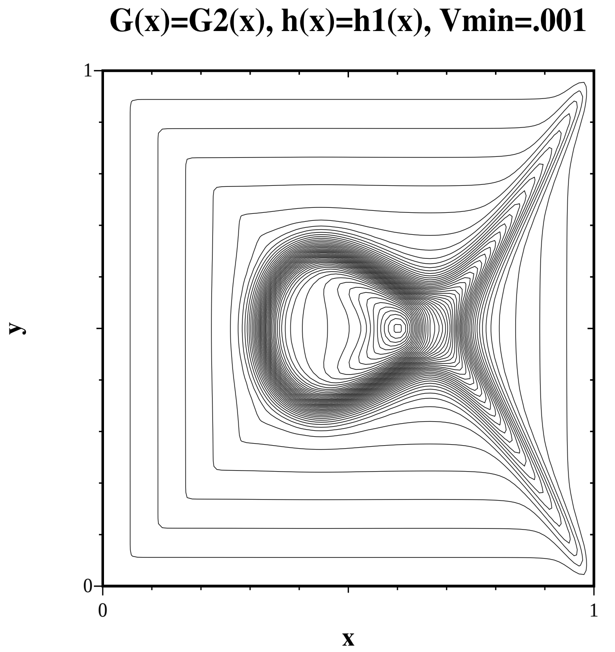


Fig 4b

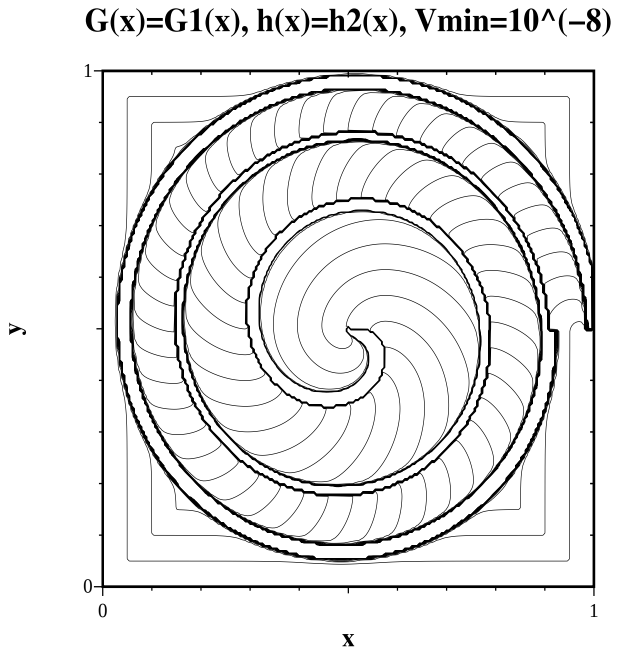


Fig 4c

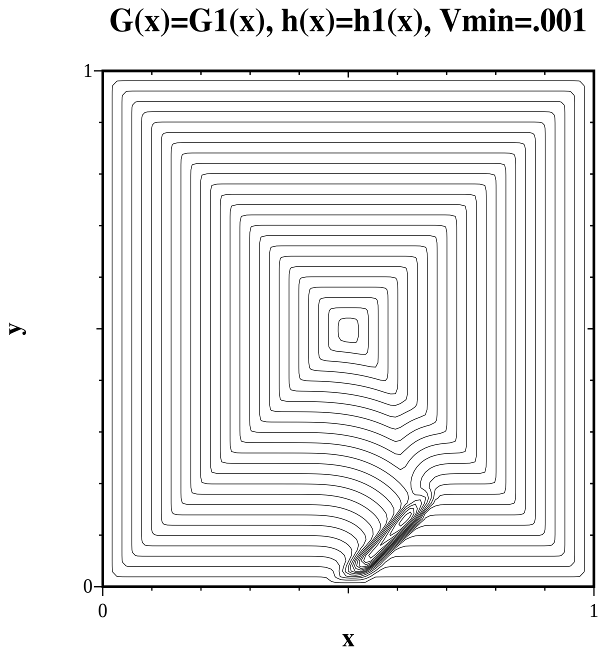


Fig 4d

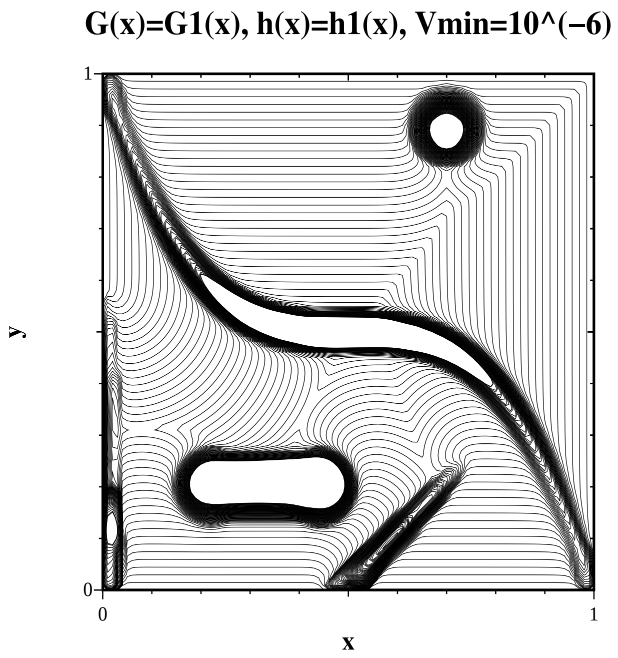


Fig 4e

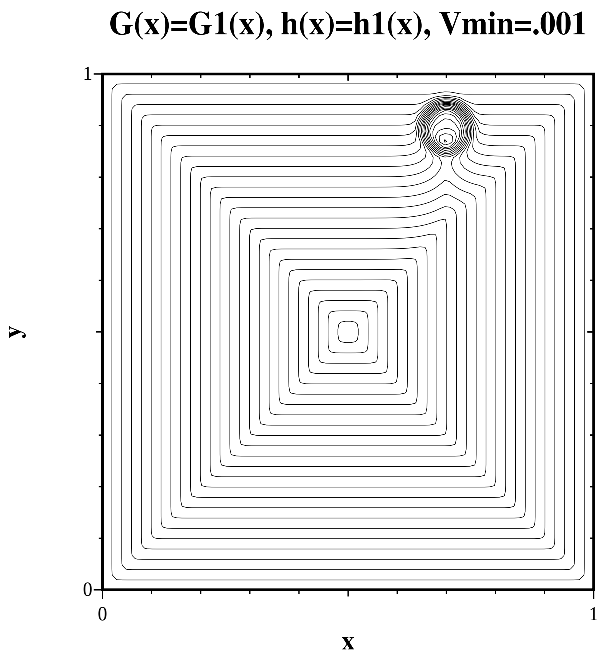


Fig 4f

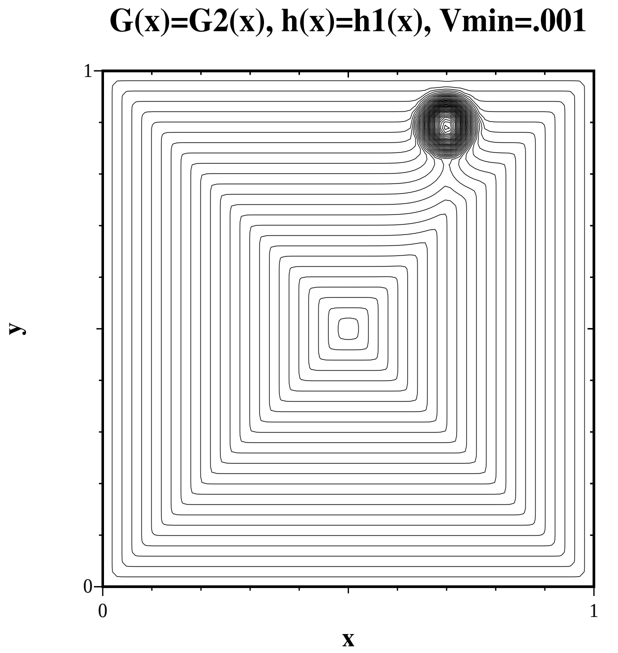


Fig 4g

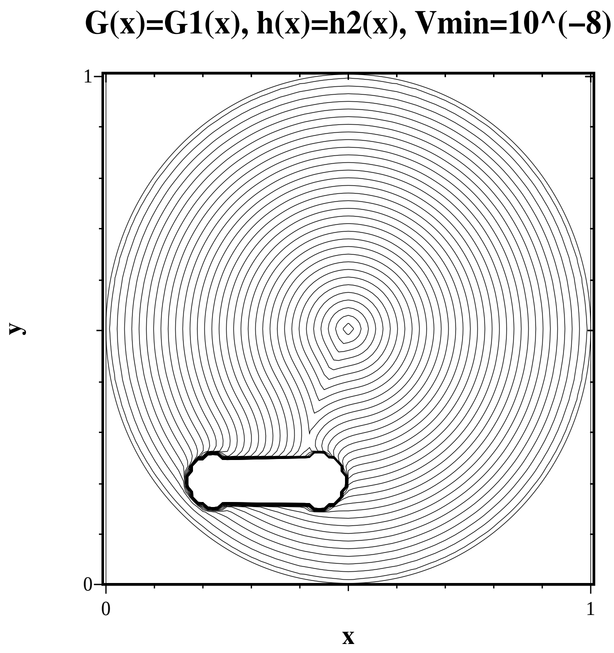


Fig 4h

A few comments are in order:

(i) Note the changes of topology of the level curves after having passed an obstacle, e.g in Fig 3a, and Fig 4d to 4h.

(ii) Note that the minimal speed is too small in Fig 4e or 4h, so that the level curves do not advance fast enough. This fact is a severe draw-back if the purpose is to generate a mesh, as in Section 5. On the contrary it is an advantage if the purpose is to detect contours in image processing (snakes).

(iii) In pictures 4f and 4g, the involved function z is the same, given by (4.41), but Fig 4f and 4g respectively correspond to the metrics G_1 and G_2. Here, the Hessian metric G_2 gives a much better location of the smoothed delta function. It would be interesting to check if this advantage of the Hessian metric is general.

5 Applications to grid generation

We want to generate a mesh whose vertices are located on successive level curves of the viscosity solution Φ to problem (2.22), (2.26),(3.36), (3.35). The level curves of Φ allow to generate a mesh as soon as we are able to connect points between two fronts. The elements that we generate are triangles, or in some experiments mixed (quadrangles or triangles).

First, we recall that the ODE associated to (2.22) is:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = V(x)\frac{G^{-1}(x)\nabla\Phi(x,t)}{|G^{-1/2}(x)\ \nabla\Phi(x,t)|} = \nabla_p H(x, \nabla\Phi). \tag{5.43}$$

These curves are the horizontal projections of the bicharacteristics, given (if V$\equiv$ 1) by:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = \nabla_p H(x,p), \\ \frac{\mathrm{d}p}{\mathrm{d}t} = -\nabla_x H(x,p), \quad p = \nabla\Phi(x,t). \end{cases} \tag{5.44}$$

At each time step, having numerically solved equation (2.22) provides an approximation of $p = \nabla\Phi(x_i, y_j)$. We then interpolate p to locally solve numerically (5.43). We detect the appearance of *shocks* in step 5 below. We restrict ourselves to the description of the case of triangles:

5.1 Principle of the construction of a triangular mesh

The principle is the following:

1. Define the domain Ω, and define Φ_0 by (3.37). In our numerical experiments, Ω is either a square or a circle.

2. Define a finite number of points $\left\{\tilde{M_j^0}, 1 \le j \le \tilde{N}_0\right\}$, with $\tilde{M_{N_0}^0} = \tilde{M_0^0}$, compute an approximation of the Riemannian distance $d^0 := d^G(\tilde{M_j^0}, \tilde{M_{j+1}^0})$ and replace the $\tilde{M_j^0}$ by new points M_j^0, which are approximately equidistant with respect to the distance d^G.

3. n:=0

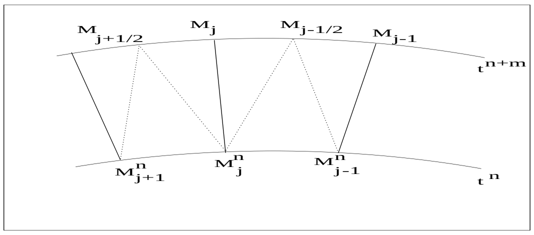
4. Choose $m = m_n = [Kd^n/\Delta t]$, where [x] is the integer part of x and K is a constant so that the average triangles will be "approximately" equilateral with respect to d^G, in the Euclidean case, $K = \sqrt{3}/2$.

5. For k=1..m,
   (i) update Φ from t_{n+k-1} to t_{n+k}, using the algorithm defined in Section 3.
   (ii) then update the points M_j according to the ODE (5.43).
   (iii) check the Riemannian distances $d_j^{n+k} := d^G(M_j^{n+k}, M_{j+1}^{n+k})$.

      . if $d_j^{n+k} < 0.5 d^n$, then suppress one point: $M_j^{n+k} := (M_j^{n+k} + M_{j+1}^{n+k})/2$
      . if $d_j^{n+k} > 1.5 d^n$, then add one point: $M_{j+1/2}^{n+k}$ between M_j^{n+k} and M_{j+1}^{n+k};

   (iv) If necessary, relabel points M_j^{n+k}.

6. At time t_{n+m},
   (i) define $M_{j+1/2}^{n+m} \simeq$ the Riemannian midpoint between M_j^{n+m} and M_{j+1}^{n+m}.



   (ii) check whether the triangles $M_j^n M_{j+1}^n M_{j+1/2}^{n+m}$ and $M_{j-1/2}^{n+m} M_j^n M_{j+1/2}^{n+m}$ are approximately equilateral with respect to the distance d^G. If necessary, add or remove one point at time t_{n+m}.
   (iii) In any case, relabel the points $M_j^{n+m} := M_{j+1/2}^{n+m}$; evaluate the average Riemannian distance d^{n+m}.
   (iv) Project these new points M_j^{n+m} on the level curve $\{\Phi(x, t^{n+m}) = 0\}$, using say two or three steps of the Newton method.

7. In the case where the above iteration fails for some points M_j^{n+m}, freeze these points, use some "cuisine" to detect the underlying change of topologies in the level curves of $\Phi(., t_{n+m})$, and to generate the corresponding local part of the mesh.

8. Go back to step 4, with n:=n+m, until all the points M_j^n are frozen.

Remark:
The most difficult parts are steps 6 and 7, which require a more detailed analysis, and will be addressed in a forthcoming work [4]. Note that such changes of topologies essentially appear in two types of situations:

(i) when there is a "pass" in the graph of the stationary viscosity solution $\Psi(x) = d^G(x, \partial\Omega)$, see e.g. Fig 4d to 4h.

(ii) when this stationary solution Ψ presents a local maximum, i.e when its level curves of level t are (locally) about to disappear, i.e. when the corresponding part of the mesh is *almost* finished. Not surprisingly, this step is one of the most difficult to handle, and should be handled by a local Delaunay algorithm, adapted to the Riemannian metric, see e.g. [32] .

5.2 Results

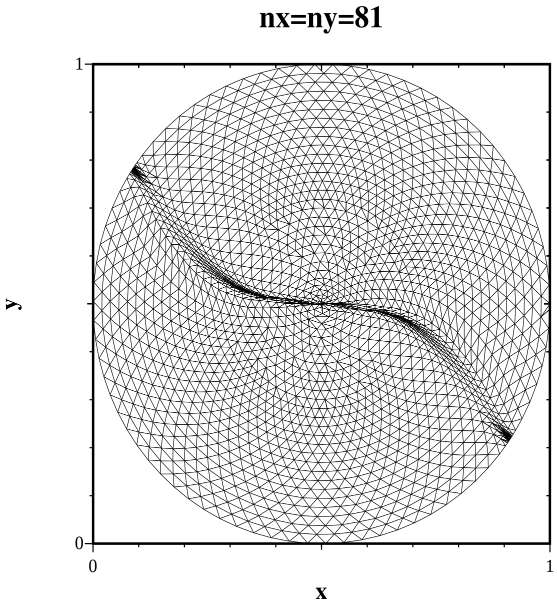


Fig 5a

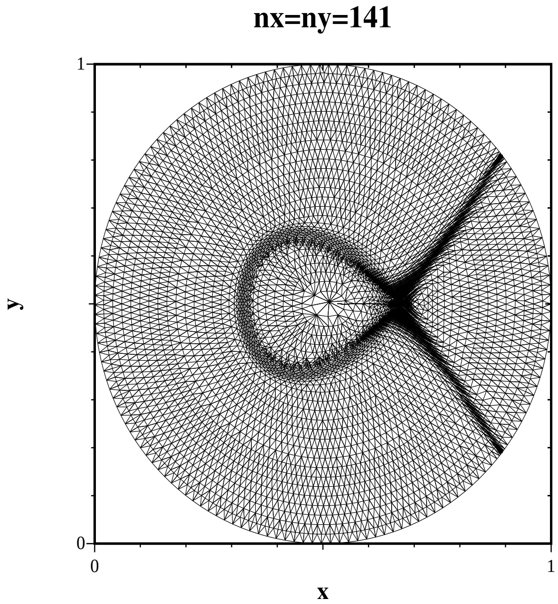


Fig 5b

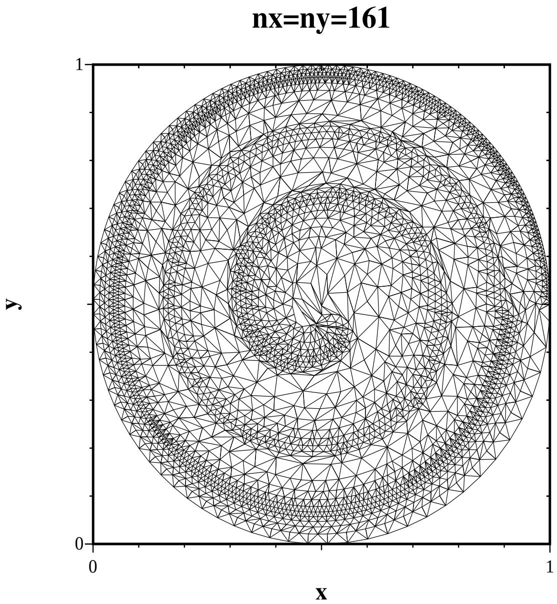


Fig 5c

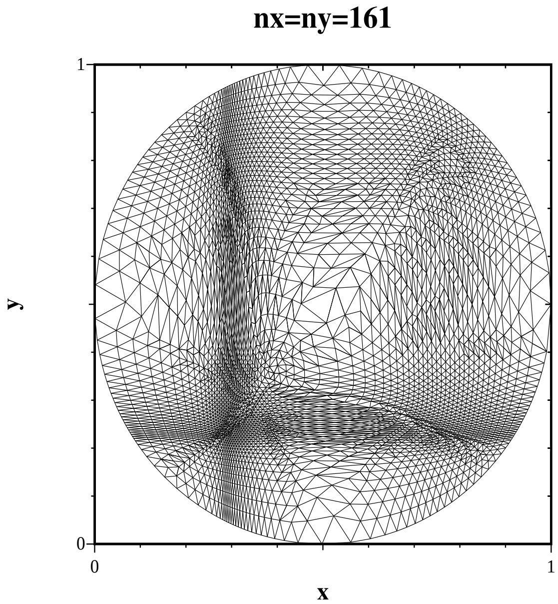


Fig 5d

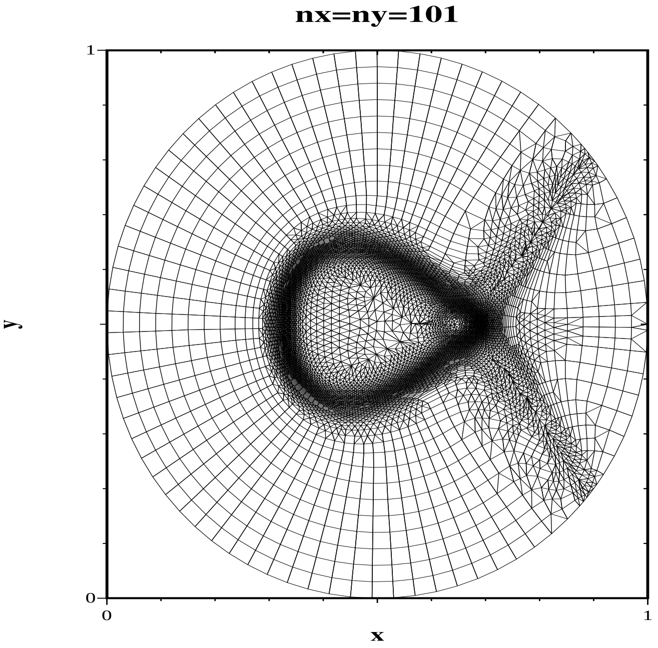


Fig 5e

Comments:

(i) All these simulations have been performed with G(x)=G1(x) and V(x)≡1.

(ii) Figures 5c and 5d, are motivated by [32]. In the case of Figure 5c, the underlying siff function z is defined implicitely, which creates some trouble: there is a jump in the angle at y=0.5.

(iii) Clearly, the generation of some of the triangles in the above pictures needs to be improved !

6 Conclusion

In this paper we have introduced the simple and natural idea of moving curves with respect to Hamilton-Jacobi equation on a manifold, or more precisely according to a given Riemannian metric, in order to generate and/or refine an anisotropic mesh. This idea is clearly in the line of the level set method, and several papers have used similar concepts, in a different context and with different motivations, see e.g. [38][49], but as far as we know, this idea had never been explicitly stated in this way, especially in this context. Clearly, our algorithms are far from being optimal, and should be improved in several respects. For instance, one could:

. use a fast-marching method, based on the stationary problem, [2][52][50], see also [42].

. use an algorithm of optimal control [48][19].

. use a relaxation approximation [36] which in the so called characteristic case induces some curvature-dependent regularization.

. improve the boundary conditions [1][13].

. strongly improve steps 5, 6, and 7 in the mesh generation, in particular combine step 7 with a local Delaunay algorithm based on a Riemannian metric, in the spirit of [32].

. more generally, use the present method in a local way and combine it with other algorithms in regions where the latter are faster, or more efficient.

. in a 3D complex geometry, use this method on a manifold to generate the "skin", and then march in time to generate a full 3D mesh.

We will address some of these questions in a forthcoming work [4].
We also note that our approach could have natural applications in image processing, not only in image enhancement [54], but possibly in the active contours method (snakes).

7 Appendix

For convenience, we recall the definition of viscosity solution, and refer e.g. to [6][25] for more details. Let us consider the evolution equation:

$$\partial_t u + H(x, \nabla u) = 0 \tag{7.45}$$

Definition 7.1. *$u \in C(\Omega)$ is said to be a viscosity solution to equation (7.45), iff for all C^1 test functions v, u satisfies simultaneously:*

(i) u is a viscosity sub-solution, i.e. if (u-v) has a local maximum at a point (t_0, x_0) in $(0,t) \times \Omega$, then

$$\partial_t v(t_0, x_0) + H(x_0, \nabla v(t_0, x_0)) \leq 0, \tag{7.46}$$

(ii) u is a viscosity super-solution, i.e. if (u-v) has a local minimum at a point (t_0, x_0) in $(0,t) \times \Omega$, then

$$\partial_t v(t_0, x_0) + H(x_0, \nabla v(t_0, x_0)) \geq 0. \tag{7.47}$$

The definition of viscosity solution in the stationary case is quite similar. As to the initial boundary value problem, in the general case the formulation involves *discontinuous* viscosity solutions, which take into account the possibility that sometimes boundary conditions cannot be satisfied in the classical sense on the whole boundary, e.g. in the case of the problem:

$$|\phi'(x)| - 1 = 0 \text{ on } (0,1), \;\; \phi(0) = 0 \,, \; \phi(1) = 2.$$

In this paper, the solution Ψ to the stationary problem (2.20), (2.21), as well as the solution Φ to problem (2.22) (2.26) (3.37), satisfy the boundary condition in the classical sense.

Figure Captions

Fig 3a. and 3b. The function z(x) here mimics a boundary layer located at a part of {x=0}.

Fig 4a. and 5a. The function $f(x_1, x_2) = 0$ is the curve:
$$x_2 - (a * x_1^3) - 1.5 * a * x_1^2 + (a/2 - 1) * x_1 + 1, \text{ where a=-4.}$$

Fig 4b., 5b. and 5e. The function $f(x_1, x_2) = 0$ is the "alpha-curve":
$$x_1^3 - 5 * x_1^2/3 + 8 * x_1/9 - 4/27 - 8 * (x_2 - .5)^2/27.$$

Fig 4c. and 5c. The function $f(x_1, x_2) = 0$ is the spiral:
$r^2 - (0.25 - \theta/8/\pi/3)$ where $r^2 = (x - .5)^2 + (y - .5)^2$, and $tan\,\theta = (\frac{y-.5}{x-.5})$ with $\theta \in (0, 6\,\pi)$.

Fig 4d. The function z(x) here is an approximation of a product of oblic Dirac masses.

Fig 4e. This is a combination of the above singularities.

Fig 4f. and 4g. The function z(x) here is an approximation of a delta-function.

Fig 4h. The function $f(x_1, x_2) = 0$ is the dumbell.

Fig 5d. The function is taken in [17].

Fig 5e. Example of a triangulation with mixed (triangles and quadrangles) elements.

References

[1] Abgrall R., Numerical Discretisation of Boundary Conditions for first order Hamilton-Jacobi equations, preprint.

[2] Adalsteinsson D., Sethian A., A fast level set method for propagating interfaces, Jour. of Comp. Physics, vol 118,p 269-277, 1995.

[3] Alvarez L., Lions P.L., Morel J.M., Image selective smoothing and edge detection by nonlinear diffusion, Siam Jour. of num. anal., vol 29, pp 845-866, 1992.

[4] Bagnerini P., Hoch P., Rascle M., In preparation.

[5] Bardi M., Evans L.C., On Hopf's formulas for solutions of Hamilton-Jacobi Equations, Nonlinear Ana., vol 8, No 11, pp 1373-1381, 1984.

[6] Barles G., Solutions de viscosité des équations de Hamilton-Jacobi, Math. & Appli., Springer-Verlag.

[7] Barles G., Uniqueness for First-Order Hamilton-Jacobi Equations and Hopf Formula, Jour. Diff. Equa., 69, pp 346-367, 1987.

[8] Barles G., Souganidis P.E., A New Approach to Front Propagation Problems:Theory and Applications, Arch. Ratio. Mech. Anal., vol 141, p 237-296, 1998.

[9] Barles G., Souganidis P.E., Convergence of approximation schemes for fully nonlinear second order equations, Asympt. Anal. 4 (1991), p271-283.

[10] Barles G., Soner H.M., Souganidis P.E., Front propagation and phase field theory, SIAM J. Control and Optimization, vol 31 No.2,p 439-469, March 1993.

[11] Beck R., Hege H-C., Seebass M., Wust P., Deuflhard P., Felix R., Adaptive finite element codes for numerical calculations in hyperthermia treatment planning, Proc. of 7th Int. Cong on Hyperthermic Onco., Rome, Italy, vol 2, pp 549-551, April 1996.

[12] Bell J., Berger M., Saltzman J., Welcome M., Three dimensional adaptative mesh refinement for hyperbolic conservation laws. SIAM J. Sci. Comput., vol 15, no1, pp 127-138, 1994.

[13] Benamou J.D., Multivalued solution and viscosity solutions of the Eikonal equation, rapport de recherche INRIA no 3281, Octobre 1997.

[14] Beraudo C., Fortin A., Coupez T., Demay Y., Vergnes B., Agassant J.F., A Finite Element Method for Computing the Flow of Multi-Mode Viscoelastic Fluids: Comparison with Experiments, Journal of Non-Newtonian Fluid Mechanics, vol. 75, pp.1-23, 1998.

[15] Berger M., Leveque R., Cartesian meshes and adaptative refinement for hyperbolic partial differential equations. Third Int. Conf. on Hyperbolic Prob., vol I,II, 67-73, Studentlitteratur, Lund, 1991.

[16] Bernardi C., Maday Y., Raffinement de maillage en éléments finis par la méthode des joints. (French) [Finite-element mesh refinement by the mortar method] C. R. Acad. Sci. Paris Sér. I Math. 320, no.3, 373–377, 1995,

[17] Borouchaki H., George P.L., Hecht F., Laug P., Mohammadi B., Saltel E., Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie II: Applications, rapport de recherche INRIA no 2760, Decembre 1995.

[18] Camilli F., Siconolfi A., Discontinuous solutions of a Hamilton-Jacobi equation with infinite speed of propagation, Siam J. Math. Anal., Vol 28, no 6, pp 1420-1445, Nov. 1997.

[19] Cardaliaguet P., Saint-Pierre P., Quincampoix M., Set-valued numerical analysis for optimal control and differential games, Stochastic and differential games", Annals of the International Society of Dynamic Games, Eds. M. Bardi, T.E.S Raghavan, T. Parthasarathy, chez Birhauser.

[20] Caselles V., Kimmel R., Sapiro G., Geodesic active contours, Interna. Journ. of Compu. Vision., vol 22, pp 61-79, 1997.

[21] Castro Diaz M.J., Hecht F., Mohammadi B., New progress in Anisotropic Grid Adaptation for Inviscid and Viscous Flows Simulations, rapport de recherche INRIA no 2671, Octobre 95.

[22] Castro Diaz M.J., Hecht F., Mohammadi B., Pironneau O., Anisotropic unstructured mesh adaptation for flows simulations, Intern. J. Numer. Methods Fluids, vol 25, no 4, pp 475-491, 1997.

[23] Chang Y.C., Hou T.Y., Merriman B., Osher S., A level set formulation of Eulerian interface capturing methods for incompressible fluid flows., J. Comp. Phys., vol 124, no2, pp 449-464, 1996.

[24] Coupez Th., Fourment L., Chenot J.L., Adaptive solutions in industrial forming process simulation, in Advances in adaptive computational methods in mechanics (Cachan), 1997, pp 365-381, Stud. Appl. Mech. , 47, Elsevier, Amsterdam, 1998.

[25] Crandall M., Lions P.L., Viscosity solutions of Hamilton-Jacobi equations, Trans of AMS, 277 May 1983.

[26] Crandall M., Ishii H., Lions P.L., User's guide to viscosity solutions of second order partial equations, Bull. of AMS, vol 27 N1, July 1992.

[27] Demazure M., Calcul différentiel, Cours de l'école polytechnique ed. 1979.

[28] Désidéri J.A., Control of hyperbolic grid-generation systems, Proc. of the Ninth International Conference on Finite Elements in Fluids, New Trends and Applications, Venice, Italy, 15-21 October 1995, M. Morandi Cecchi, K. Morgan, J. Périaux, B.A. Schreffer, O.C. Zienkiewicz Eds., (Universita di Padova Publ.).

[29] Erdmann B., Frei M., Hoppe R.H.W., Kornhuber R., Wiest U., Adaptative finite element methods for variational inequalities, East-West J. Numer. Math., inequalities', East-West J. Numer. Math. 1, pp 165-197, 1993.

[30] Fritz J., Partial Differential Equations, Applied Mathe. Scien., Springer Verlag.

[31] George P.L., Génération automatique de maillages, applications aux méthodes d'éléments finis, Masson.

[32] George P.L., Borouchaki H., Triangulation de Delaunay et maillage, applications aux éléments finis, Hermes.

[33] Goodman J., Samuelsson K., Szepessy A., Anisotropic refinement algorithms for finite elements, preprint.

[34] Harabetian E., Osher S., Regularization of ill-posed problems via the level set approach, SIAM J. Appl. Math, vol 58, no 6, pp 1689-1706, 1998.

[35] Hopf E., Generalized solutions of non-linear equations of first-order, Jou. Math. Mech., 14, pp 951-973, 1965.

[36] Jin S., Katsoulakis M.A., Relaxation approximations to front propagation, Jour. Diffe. Equa. 138, pp 380-387, 1997.

[37] Johnson C., Szepessy A., Adaptative finite element methods for conservation laws based on a posteriori error estimates, Comm. Pure Appl. Math., 98, pp 199-234, 1995.

[38] Kimmel R., Curve evolution on surfaces, PhD Thesis 1995.

[39] Lax P.D., Hyperbolic systems of conservations laws II, Comm. pure appl. math., 10, pp 537-566, 1957.

[40] Lions P.L., Generalized solutions of Hamilton-Jacobi equations, Pitman, Boston, 1982.

[41] Lions P.L., Rochet J.C., Hopf formula and multitime Hamilton-Jacobi equations., proc. of AMS, vol 96, No1, Jan. 1996.

[42] Milne R.B., An Adaptative Level Set Method, PhD Thesis 1995.

[43] Oleinik O.A., Discontinuous solutions of non-linear differential equations, Am. math. Soc. Transl., 2, pp 95-172, 1963.

[44] Osher S., A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations, SIAM J. Math. Anal., vol 24, no5, pp 1145-1152, 1993.

[45] Osher S., Sethian J.A., Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, Jour. of Comp. Physics, vol 79, pp 12-49, 1998.

[46] Osher S., Shu C.W., High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations, SIAM J.Numer. Ana., vol 28, no 4, pp 907-922, August 1991.

[47] Pember R., Bell J., Colella Ph., Crutchfield W., Welcome M., An adaptative cartesian grid method for unsteady compressible flow in irregular regions, J. Comput. Phys., vol 120 no2, pp 278-304, 1995.

[48] Rouy E., Tourin A., A viscosity solutions approach to shape-from-shading, SIAM J.Numer.Anal., vol 29, No3, pp 867-884, June 1992.

[49] Sethian J.A., Level Set Methods, Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press 1996.

[50] Sethian J.A., Level Set Methods and Fast Marching Methods, Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press 1999.

[51] Sethian J.A., Curvature Flow and Entropy Conditions Applied to Grid Generation, Jour. of Comp. Physics, vol 115,p 440-454, 1994.

[52] Sethian J.A., A fast Marching level set method for monotonically advancing fronts, Proc. Nat. Acad. Sci, vol 93,p 1591-1595, February 1996.

[53] Siconolfi A., Metric properties of Hamilton-Jacobi equations in the nonconvex case, int. conf. in memory of S.N Kruzhkov, Besancon France 1999.

[54] Sochen N., Kimmel R. Malladi R., A geometrical framework for low level vision, Ieee trans. on image proc., 7(3), pp 310-318, 1998.

[55] Spivak M., A comprehensive introduction to Differential Geometry, vol 1, Brandeis University.