Fast Marching Method

Nicolas Forcadel

INSA de Rouen

Cours GM5 : "Equations de Hamilton-Jacobi et applications"

2015-2016

Plan

Plan

1. Level Set Method
2. La méthode Fast Marching classique

Plan

1. Level Set Method
2. La méthode Fast Marching classique
3. Generalized Fast Marching Method

Plan

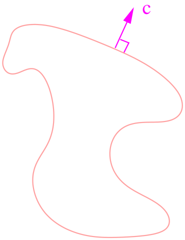
1. Level Set Method
2. La méthode Fast Marching classique
3. Generalized Fast Marching Method
4. Some other extension

Plan

Plan

1. Level Set Method
2. La méthode Fast Marching classique
3. Generalized Fast Marching Method
4. Some other extension
5. Simulations

Evolution d'un front



On s'intéresse à l'évolution d'un front Γ_t dans la **direction normale** et avec une vitesse $c = c(y) > 0, \; y \in \mathbb{R}^d$.

L'équation d'évolution de ce front s'écrit

$$\frac{d\Gamma_t}{dt} = c \, n_{\Gamma_t}$$

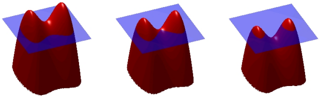
où n_{Γ_t} désigne la **normale unitaire** extérieure au front.

La méthode Level-Set

La méthode Level-Set est une méthode très populaire pour les problèmes de mouvements de fronts.

L'idée principale est de représenter le front par **la ligne de niveau zéro** d'une fonction u:

$$\Gamma_t = \{x,\ u(x,t) = 0\}.$$



La méthode Level-Set

L'inconvénient majeur est d'ajouter **une dimension supplémentaire** au problème, ce qui peut engendrer des coûts de calcul non négligeables.

Soit donc u une fonction telle que

$$\begin{cases} u(x,t) < 0 & \text{si } x \text{ est situé } \textbf{à l'intérieur} \text{ de } \Gamma_t, \\ u(x,t) = 0 & \text{si } x \text{ est situé } \textbf{appartient} \text{ à } \Gamma_t, \\ u(x,t) > 0 & \text{si } x \text{ est situé } \textbf{à l'extérieur} \text{ de } \Gamma_t. \end{cases}$$

On a

$$n_{\Gamma_t} = \frac{\nabla_x u(x,t)}{|\nabla_x u(x,t)|}$$

La méthode Level-Set

Rappelons que

$$\Gamma_t = \{x,\ u(x,t) = 0\}.$$

Formellement:

$$u(\Gamma_t, t) = 0 \Rightarrow u_t(x,t) + \nabla_x u(x,t) \cdot \frac{\partial \Gamma_t}{\partial t} = 0.$$

avec $\dfrac{\partial \Gamma_t}{\partial t} = c(x) n_{\Gamma_t},\ n_{\Gamma_t} = \dfrac{\nabla_x u(x,t)}{|\nabla_x u(x,t)|}$.

- On aboutit à **l'équation eikonale** suivante:

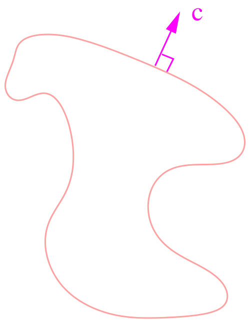
$$u_t(x,t) + c(x)|\nabla_x u(x,t)| = 0$$

Plan

La méthode Fast Marching

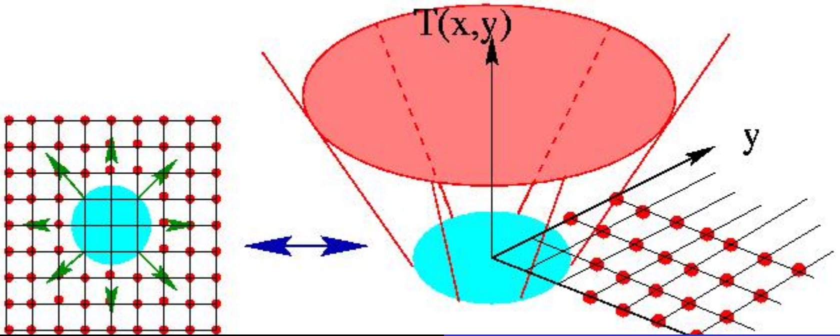
Le but de la méthode Fast Marching est de résoudre efficacement l'équation Eikonale

$$u_t(x,t) + c(x)|\nabla_x u(x,t)| = 0$$



Approche stationnaire

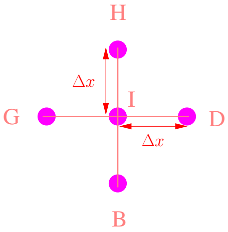
- On utilise une approche stationnaire en posant $u(x,t) = T(x) - t$
  $\Rightarrow |\nabla T(x)| = \dfrac{1}{c(x)}$.
- $T(x)$ représente le temps d'arriver du front au point x.



Description de la méthode

L'objectif est donc de **calculer numériquement** le temps d'arrivée T solution de **l'équation stationnaire**

$$|\nabla T(x)| = \frac{1}{c(x)}.$$



$$\max(T_I - T_G, T_I - T_D, 0)^2 + \max(T_I - T_H, T_I - T_B, 0)^2 = \left(\frac{\Delta x}{c_I}\right)^2 \quad (1)$$

Description de la méthode

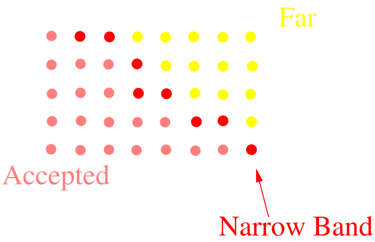
Une première idée serait d'appliquer un schéma itératif pour résoudre ce problème non linéaire et a attendre la convergence de l'algorithme à la précision souhaitée. **Ceci peut être très long!**

Il est en fait possible de calculer les valeurs T_I dans un ordre spécial qui permet d'obtenir la convergence en une seule itération. Cet ordre spécial correspond au **valeur croissante de** T_I.

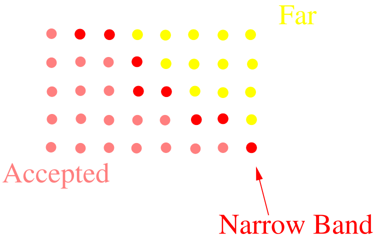
Introduction de la Narrow Band

Pour cela, on va répartir les points en trois régions:

- **Accepted points :** Il s'agit des points qui ont déjà été atteints par le front et pour lesquels on connait déjà la valeur de T_I
- **Narrow band :** Ce sont les points qui n'ont pas encore été atteints par le front mais qui sont sur le point de l'être, c'est à dire ayant un voisin qui a été atteint par le front.
- **Far away :** Ce sont les autres points.

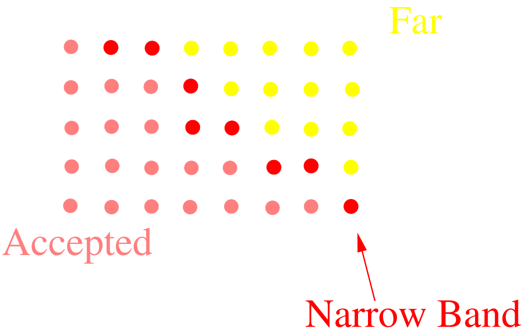


Algorithme de la méthode: initialisation



- On définit la région **accepted** au temps initial comme étant les points à l'intérieur de Γ_t et on initialise T à 0 sur ces points.
- On définit la **Narrow Band** comme étant l'ensemble des points qui ne sont pas accepté mais qui ont un voisin accepté.
- La région **Far away** est le reste des points.

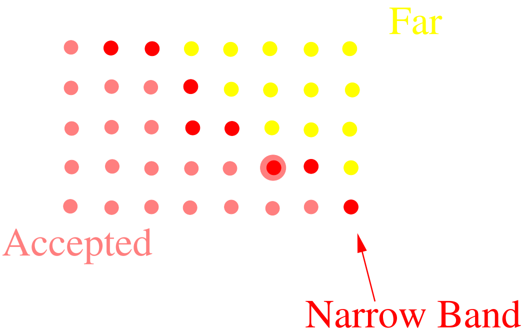
Algorithme de la méthode: boucle principale



- Calculer T_I sur la Narrow Band en résolvant:

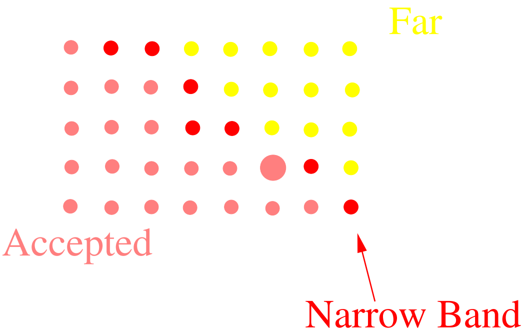
$$\max(T_I - T_G, T_I - T_D, 0)^2 + \max(T_I - T_H, T_I - T_B, 0)^2 = \left(\frac{\Delta x}{c_I}\right)^2 \quad (2)$$

Algorithme de la méthode: boucle principale



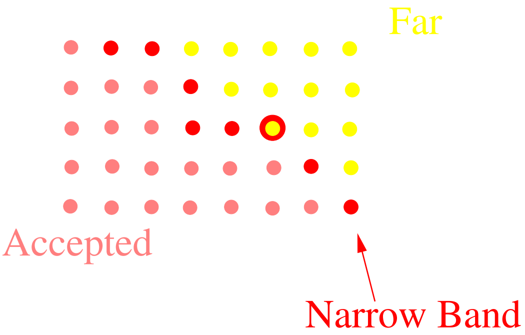
- On cherche la plus petite valeur de T_I sur la Narrow Band.

Algorithme de la méthode: boucle principale



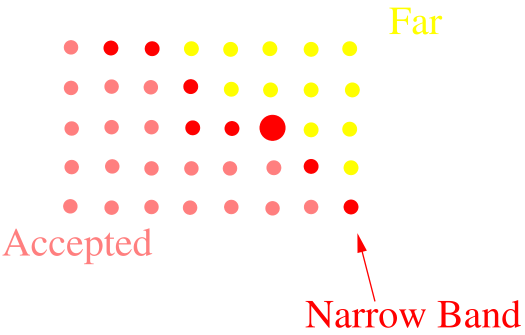
- On cherche la plus petite valeur de T_I sur la Narrow Band. Le point correspondant devient accepté.

Algorithme de la méthode: boucle principale



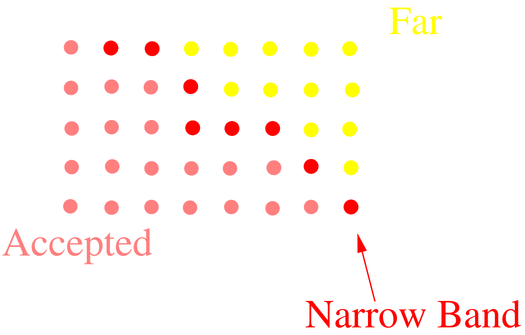
- On cherche la plus petite valeur de T_I sur la Narrow Band. Le point correspondant devient accepté.
- La nouvelle Narrow band est redéfinie comme le bord de la région acceptée.

Algorithme de la méthode: boucle principale



- On cherche la plus petite valeur de T_I sur la Narrow Band. Le point correspondant devient accepté.
- La nouvelle Narrow band est redéfinie comme le bord de la région acceptée.

Algorithme de la méthode: boucle principale



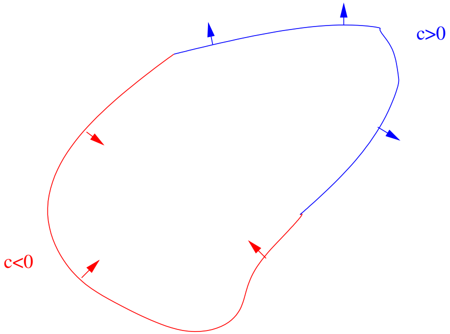
- On cherche la plus petite valeur de T_I sur la Narrow Band. Le point correspondant devient accepté.
- La nouvelle Narrow band est redéfinie comme le bord de la région acceptée.

Complexité numérique

- Calcul du temps T_I : au plus 4 fois pour chaque noeud.
- Recherche du minimum: en utilisant un arbre binaire, le coût est en $O(\ln(N_{NB}))$.
- Le coût global est donc en $O(N \ln(N))$ (N représente le nombre total de point sur la grille).

General case: no stationnary representation

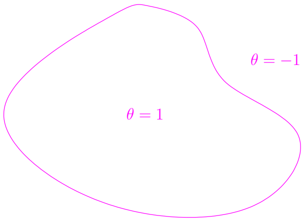
The notion of “Accepted” points is not adapted to generalize the algorithm.



Rewriting the classical FMM

Introduction of a field θ to represent the front:

$$\begin{cases} \theta = 1 & \text{inside} \\ \theta = -1 & \text{outside} \end{cases}$$



Rewriting the classical FMM

- $V(I) = \{J \in \mathbb{Z}^N,\ |I - J| \leq 1\} \backslash \{I\}$

Rewriting the classical FMM

- $V(I) = \{J \in \mathbb{Z}^N,\ |I - J| \leq 1\} \backslash \{I\}$
- "Accepted" points at step $n = \{I,\ \theta_I^n = 1\}$

Rewriting the classical FMM

- $V(I) = \{J \in \mathbb{Z}^N,\ |I - J| \leq 1\} \backslash \{I\}$
- "Accepted" points at step $n = \{I,\ \theta_I^n = 1\}$
- Narrow Band at step $n = \{I,\ \exists\ J \in V(I), \theta_J^n = -\theta_I^n = 1\} = NB^n$

Rewriting the classical FMM

- $V(I) = \{J \in \mathbb{Z}^N,\ |I - J| \leq 1\} \backslash \{I\}$
- "Accepted" points at step $n = \{I,\ \theta_I^n = 1\}$
- Narrow Band at step $n = \{I,\ \exists\ J \in V(I), \theta_J^n = -\theta_I^n = 1\} = NB^n$
- Useful points at step n : if $I \in NB^n$,

$$\mathcal{U}^n(I) = \{J \in V(I),\ \theta_J^n = 1\}, \quad \mathcal{U}^n = \cup_{I \in NB^n} \mathcal{U}^n(I)$$

Algorithm

Initialization

1 $\theta_I^0 = \begin{cases} 1 & \text{if } x_I \in \Omega_0 \\ -1 & \text{otherwise} \end{cases}$

Algorithm

Initialization

1. $\theta_I^0 = \begin{cases} 1 & \text{if } x_I \in \Omega_0 \\ -1 & \text{otherwise} \end{cases}$
2. $t_0 = 0$

   $T_I^0 = \begin{cases} 0 & \text{if } I \in \mathcal{U}^0 \\ +\infty & \text{otherwise} \end{cases}$

   $n = 1$

Algorithm

Loop

3 Computation of the candidate time $\tilde{T}_I$ for $I \in NB^{n-1}$:

$$\begin{array}{c} \max(\tilde{T}_I^{n-1} - T_G^{n-1}, \tilde{T}_I^{n-1} - T_D^{n-1}, 0)^2 \\ + \\ \max(\tilde{T}_I^{n-1} - T_H^{n-1}, \tilde{T}_I^{n-1} - T_B^{n-1}, 0)^2 \end{array} = \frac{(\Delta x)^2}{|c_I|^2}$$

Algorithm

Loop

3 Computation of the candidate time $\tilde{T}_I$ for $I \in NB^{n-1}$:

$$\begin{array}{c} \max(\tilde{T}_I^{n-1} - T_G^{n-1}, \tilde{T}_I^{n-1} - T_D^{n-1}, 0)^2 \\ + \\ \max(\tilde{T}_I^{n-1} - T_H^{n-1}, \tilde{T}_I^{n-1} - T_B^{n-1}, 0)^2 \end{array} = \frac{(\Delta x)^2}{|c_I|^2}$$

4 $t_n = \inf_{I \in NB^{n-1}} \tilde{T}_I^{n-1}$

Algorithm

5 $NA^n = \{I,\ \tilde{T}_I^{n-1} = t_n\}$

Algorithm

5 $NA^n = \{I,\ \tilde{T}_I^{n-1} = t_n\}$

6 $\theta_I^n = \begin{cases} 1 & \text{if } I \in NA^n \\ \theta_I^{n-1} & \text{otherwise} \end{cases}$

Algorithm

5 $NA^n = \{I,\ \tilde{T}_I^{n-1} = t_n\}$

6 $\theta_I^n = \begin{cases} 1 & \text{if } I \in NA^n \\ \theta_I^{n-1} & \text{otherwise} \end{cases}$

7 $T_I^n = \begin{cases} t_n & \text{if } I \in NA^n \text{ and } I \in \mathcal{U}^n \\ T_I^{n-1} & \text{if } I \in \mathcal{U}^{n-1} \backslash NA^n \text{ and } I \in \mathcal{U}^n \\ +\infty & \text{otherwise} \end{cases}$

Algorithm

5 $NA^n = \{I,\ \tilde{T}_I^{n-1} = t_n\}$

6 $\theta_I^n = \begin{cases} 1 & \text{if } I \in NA^n \\ \theta_I^{n-1} & \text{otherwise} \end{cases}$

7 $T_I^n = \begin{cases} t_n & \text{if } I \in NA^n \text{ and } I \in \mathcal{U}^n \\ T_I^{n-1} & \text{if } I \in \mathcal{U}^{n-1} \backslash NA^n \text{ and } I \in \mathcal{U}^n \\ +\infty & \text{otherwise} \end{cases}$

8 $n := n + 1$

Convergence result

We define

$$\theta^\varepsilon(x,t) = \theta_I^n \text{ if } x \in [x_I, x_I + \Delta x[,\ t \in [t_n, t_{n+1}[.$$

Theorem (Carlini, Falcone, F., Monneau)

Under regularity assumptions on Ω_0 and c,we have

$$\theta^\varepsilon \to \theta$$

solution of

$$\begin{cases} \theta_t = c(x)|\nabla\theta| \\ \theta(t=0,\cdot) = 1_{\Omega_0} - 1_{\Omega_0^c} \end{cases}$$

Case $c(x,t) > 0$

Loop

3 Computation of the candidate time $\tilde{T}_I$ for $I \in NB^{n-1}$:

$$\begin{array}{c} \max(\tilde{T}_I^{n-1} - T_G^{n-1}, \tilde{T}_I^{n-1} - T_D^{n-1}, 0)^2 \\ + \\ \max(\tilde{T}_I^{n-1} - T_H^{n-1}, \tilde{T}_I^{n-1} - T_B^{n-1}, 0)^2 \end{array} = \frac{(\Delta x)^2}{|c_I \quad|^2}$$

4 $t_n = \inf_{I \in NB^{n-1}} \tilde{T}_I^{n-1}$

Case $c(x,t) > 0$

Loop

3 Computation of the candidate time $\tilde{T}_I$ for $I \in NB^{n-1}$:

$$\begin{array}{c} \max(\tilde{T}_I^{n-1} - T_G^{n-1}, \tilde{T}_I^{n-1} - T_D^{n-1}, 0)^2 \\ + \\ \max(\tilde{T}_I^{n-1} - T_H^{n-1}, \tilde{T}_I^{n-1} - T_B^{n-1}, 0)^2 \end{array} = \frac{(\Delta x)^2}{|c_I^{n-1}|^2}$$

4 $t_n = \inf_{I \in NB^{n-1}} \tilde{T}_I^{n-1}$

Case $c(x,t) > 0$

Loop

3 Computation of the candidate time $\tilde{T}_I$ for $I \in NB^{n-1}$:

$$\begin{array}{cc} & \max(\tilde{T}_I^{n-1} - T_G^{n-1}, \tilde{T}_I^{n-1} - T_D^{n-1}, 0)^2 \\ + & \\ & \max(\tilde{T}_I^{n-1} - T_H^{n-1}, \tilde{T}_I^{n-1} - T_B^{n-1}, 0)^2 \end{array} = \frac{(\Delta x)^2}{|c_I^{n-1}|^2}$$

4 $\tilde{t}_n = \inf_{I \in NB^{n-1}} \tilde{T}_I^{n-1}$

Case $c(x,t) > 0$

Loop

3 Computation of the candidate time $\tilde{T}_I$ for $I \in NB^{n-1}$:

$$\begin{array}{c} \max(\tilde{T}_I^{n-1} - T_G^{n-1}, \tilde{T}_I^{n-1} - T_D^{n-1}, 0)^2 \\ + \\ \max(\tilde{T}_I^{n-1} - T_H^{n-1}, \tilde{T}_I^{n-1} - T_B^{n-1}, 0)^2 \end{array} = \frac{(\Delta x)^2}{|c_I^{n-1}|^2}$$

4 $\tilde{t}_n = \inf_{I \in NB^{n-1}} \tilde{T}_I^{n-1}$

4bis Truncature of $\tilde{t}_n$: $t_n = \max(t_{n-1}, \min(\tilde{t}_n, t_{n-1} + \Delta t))$
If $t_n = t_{n-1} + \Delta t < \tilde{t}_n$, then go to 3 with $n := n+1$.

Case $c(x,t) > 0$

5 $NA^n = \{I,\ \tilde{T}_I^{n-1} = t_n\}$

6 $\theta_I^n = \begin{cases} 1 & \text{if } I \in NA^n \\ \theta_I^{n-1} & \text{otherwise} \end{cases}$

7 $T_I^n = \begin{cases} t_n & \text{if } I \in NA^n \text{ and } I \in \mathcal{U}^n \\ T_I^{n-1} & \text{if } I \in \mathcal{U}^{n-1} \backslash NA^n \text{ and } I \in \mathcal{U}^n \\ +\infty & \text{otherwise} \end{cases}$

8 $n := n+1$

Case $c(x,t) > 0$

5 $NA^n = \{I,\ \tilde{T}_I^{n-1} = \tilde{t}_n\}$

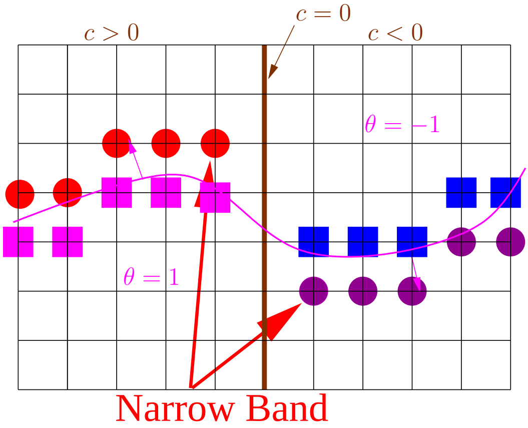
6 $\theta_I^n = \begin{cases} 1 & \text{if } I \in NA^n \\ \theta_I^{n-1} & \text{otherwise} \end{cases}$

7 $T_I^n = \begin{cases} t_n & \text{if } I \in NA^n \text{ and } I \in \mathcal{U}^n \\ T_I^{n-1} & \text{if } I \in \mathcal{U}^{n-1} \backslash NA^n \text{ and } I \in \mathcal{U}^n \\ +\infty & \text{otherwise} \end{cases}$

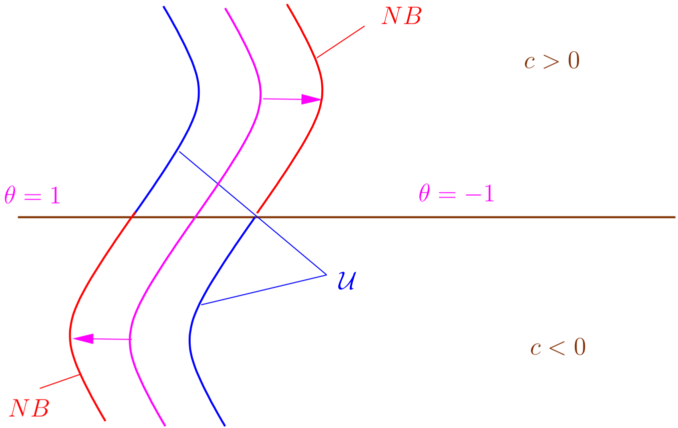
8 $n := n + 1$

Plan

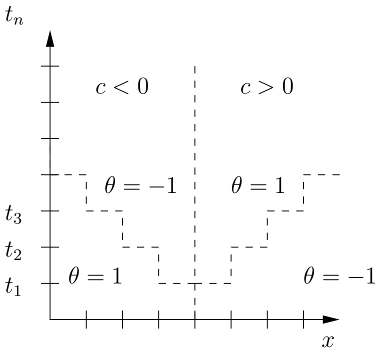
Idea of the GFMM



Schematic representation



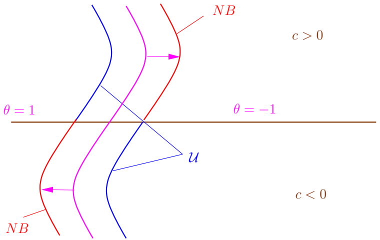
Regularisation of the speed



Definition

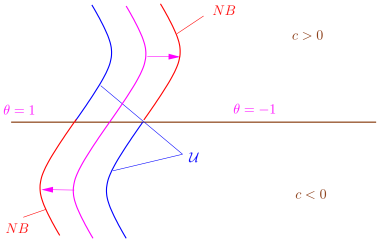
$$\hat{c}_I^n = \begin{cases} 0 & \text{if } \exists\, J \in V(I) \text{ t.q. } c_I^n c_J^n < 0 \text{ et } |c_I^n| < |c_J^n| \\ c_I^n & \text{otherwise} \end{cases}$$

Definition



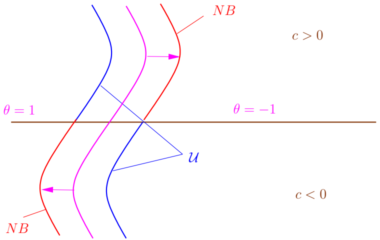
- $NB^n = \{I,\ \exists\ J \in V(I),\ \theta_I^n = -\theta_J^n \text{ and } \hat{c}_I^n \theta_I^n < 0\}$

Definition



- $NB^n = \{I, \ \exists\ J \in V(I), \ \theta_I^n = -\theta_J^n \text{ and } \hat{c}_I^n \theta_I^n < 0\}$
- If $I \in NB^n$, $\mathcal{U}^n(I) = \{J \in V(I), \ \theta_J^n = -\theta_I^n\}$
  $\mathcal{U}^n = \cup_{I \in NB^n} \mathcal{U}^n(I)$

Definition



- $NB^n = \{I,\ \exists\ J \in V(I),\ \theta_I^n = -\theta_J^n \text{ and } \hat{c}_I^n \theta_I^n < 0\}$
- If $I \in NB^n$, $\mathcal{U}^n(I) = \{J \in V(I),\ \theta_J^n = -\theta_I^n\}$
  $\mathcal{U}^n = \cup_{I \in NB^n} \mathcal{U}^n(I)$
- $T_{J \to I}^n = \begin{cases} T_J^n & \text{if } J \in \mathcal{U}^n(I) \\ +\infty & \text{otherwise} \end{cases}$

Algorithm

Loop

3 Computation of the candidate time $\tilde{T}_I$ for $I \in NB^{n-1}$:

$$\begin{array}{c} \max(\tilde{T}_I^{n-1} - T_{G\to I}^{n-1}, \tilde{T}_I^{n-1} - T_{D\to I}^{n-1}, 0)^2 \\ + \\ \max(\tilde{T}_I^{n-1} - T_{H\to I}^{n-1}, \tilde{T}_I^{n-1} - T_{B\to I}^{n-1}, 0)^2 \end{array} = \frac{(\Delta x)^2}{|c_I^{n-1}|^2}$$

4 $\tilde{t}_n = \inf_{I \in NB^{n-1}} \tilde{T}_I^{n-1}$

4bis Truncature of $\tilde{t}_n$: $t_n = \max(t_{n-1}, \min(\tilde{t}_n, t_{n-1} + \Delta t))$
If $t_n = t_{n-1} + \Delta t < \tilde{t}_n$, then go to 3 with $n := n + 1$.

Algorithm

5 $NA^n = \{I,\ \tilde{T}_I^{n-1} = \tilde{t}_n\}$

6 $\theta_I^n = \begin{cases} -\theta_I^{n-1} & \text{if } I \in NA^n \\ \theta_I^{n-1} & \text{otherwise} \end{cases}$

7 $T_I^n = \begin{cases} t_n & \text{if } I \in NA^n \cup (\mathcal{U}^{n-1})^c \text{ and } I \in \mathcal{U}^n \\ T_I^{n-1} & \text{if } I \in \mathcal{U}^{n-1} \backslash NA^n \text{ and } I \in \mathcal{U}^n \\ +\infty & \text{otherwise} \end{cases}$

8 $n := n + 1$

Convergence result

- Increasing sequence of time: $(t_{n_k})_{k\in\mathbb{N}}$ s.t.

$$t_{n_k} = t_{n_k+1} = ... = t_{n_{k+1}-1} < t_{n_{k+1}}$$

Convergence result

- Increasing sequence of time: $(t_{n_k})_{k\in\mathbb{N}}$ s.t.

$$t_{n_k} = t_{n_k+1} = ... = t_{n_{k+1}-1} < t_{n_{k+1}}$$

- We define

$$\theta^\varepsilon(x,t) = \theta_I^{n_{k+1}-1} \text{ if } x \in [x_I, x_I + \Delta x[,\ t \in [t_{n_k}, t_{n_{k+1}}[.$$

Convergence result

- Increasing sequence of time: $(t_{n_k})_{k\in\mathbb{N}}$ s.t.

$$t_{n_k} = t_{n_k+1} = \ldots = t_{n_{k+1}-1} < t_{n_{k+1}}$$

- We define

$$\theta^\varepsilon(x,t) = \theta_I^{n_{k+1}-1} \text{ if } x \in [x_I, x_I + \Delta x[,\ t \in [t_{n_k}, t_{n_{k+1}}[.$$

- half relaxed limits:

$$\overline{\theta}(x,t) = \limsup_{\varepsilon\to 0, y\to x, s\to t} \theta^\varepsilon(y,s), \quad \underline{\theta}(x,t) = \liminf_{\varepsilon\to 0, y\to x, s\to t} \theta^\varepsilon(y,s),$$

Convergence result

Theorem (Carlini, Falcone, F., Monneau)

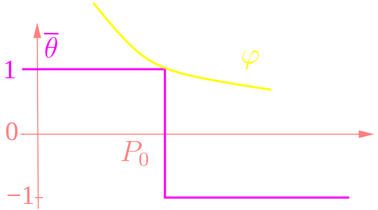
Under regularity assumptions on Ω_0 and c, we have that $\overline{\theta}$ is a sub solution and $\underline{\theta}$ is a super solution of

$$\begin{cases} \theta_t = c(x,t)|\nabla\theta| \\ \theta(t=0,\cdot) = 1_{\Omega_0} - 1_{\Omega_0^c} \end{cases} \tag{2}$$

In particular, if (2) satisfies a comparison principle, then $\overline{\theta} = (\underline{\theta})^*$ is the unique usc solution of (2)

Idea of the proof

- Assume that $\overline{\theta}^0$ is not a subsolution at P_0:

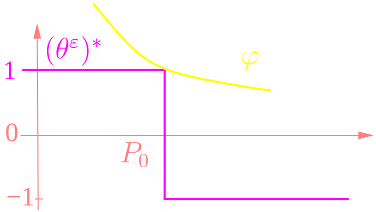


with

$$\varphi_t(P_0) = \bar{c}\,|\nabla\varphi(P_0)| > 0 \quad \text{and} \quad \bar{c} > c(P_0)$$

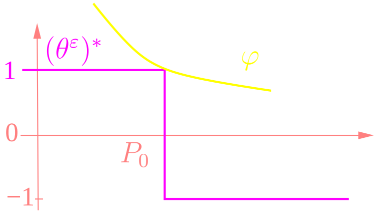
Idea of the proof

- Deduce at the ε-level that (with $P_\varepsilon = P_0$ to simplify)



Idea of the proof

- Deduce at the ε-level that (with $P_\varepsilon = P_0$ to simplify)



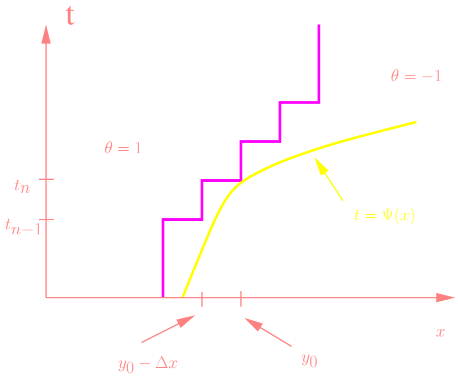
- Define ψ by

$$\varphi(x, \psi(x)) = 1$$

Idea of the proof

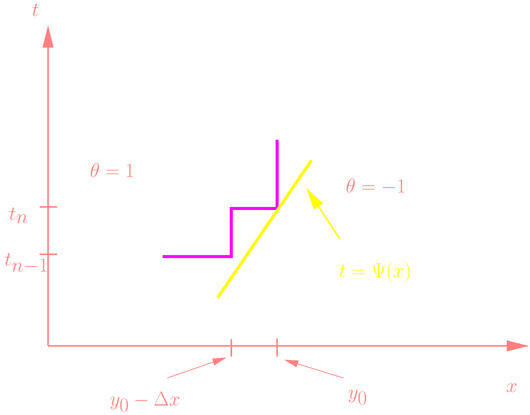
- To simplify, assume in 1D that

$$P_0 = (y_0, t_n) \quad \text{with} \quad \frac{y_0}{\Delta x} \in \mathbb{Z}$$

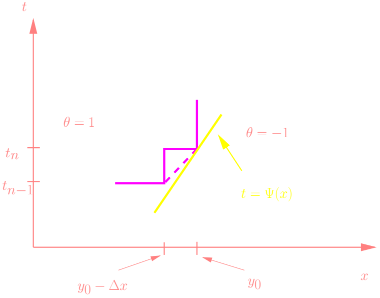


Idea of the proof

- To simplify, assume that φ is linear



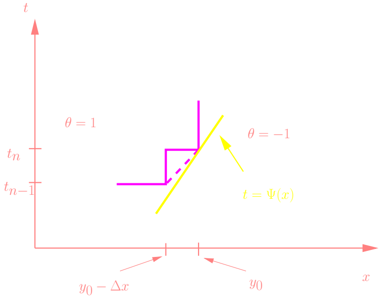
Idea of the proof



Then

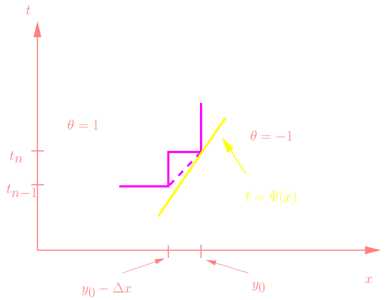
$$\psi' \geq \frac{t_n - t_{n-1}}{\Delta x}$$

Idea of the proof



$$\frac{1}{\overline{c}} = \psi' \geq \frac{t_n - t_{n-1}}{\Delta x} = \frac{1}{c(y_0 - \Delta x, t_{n-1})} = \frac{1}{c(P_0)} + o(1)$$

Idea of the proof



$$\frac{1}{\overline{c}} = \psi' \geq \frac{t_n - t_{n-1}}{\Delta x} = \frac{1}{c(y_0 - \Delta x, t_{n-1})} = \frac{1}{c(P_0)} + o(1)$$

Contradiction because $\overline{c} > c(P_0) > 0$

Numerical complexity

Assume that the velocity is constant in each intervall $[k\Delta T, (k+1)\Delta T)$ for some $\Delta T > 0$.

1. Constant in time velocity ($\Delta T = +\infty$): $O(N \ln N)$.
2. $O(\frac{1}{\sqrt{N}}) \leq \Delta T < +\infty$: $O(N \ln N)$.
3. $0 \leq \Delta T < O(\frac{1}{\sqrt{N}})$: $O(N^{\frac{3}{2}})$.

Comparison principle

For a GFMM algorithm slightly different, we have the following theorem :

Theorem (F.)

We consider 2 GFMM with speed c_u (θ_u) and c_v (θ_v). Assume that

$$\inf_{s\in[t-\Delta t,t]} c_v(x,s) \geq \sup_{s\in[t-\Delta t,t]} c_u(x,s).$$

If $\Omega_u^0 \subset \Omega_v^0$ then

$$\theta_u^\varepsilon(x,t) \leq \theta_v^\varepsilon(x,t).$$

Plan

Group Marching Method

- **Idea**: accept a group of points at every iteration [Kim]
- **Advantages**: $O(N)$ complexity.
- **Disadvantages**: introduction of some small errors and specifically design for the eikonal equations

Buffered Fast Marching Method

- **Anisotropic evolution**: a value T can depend on values greater than T
- **Idea**: do not accept the node with the minimum value BUT put it in a buffer. All the node of the buffer are recomputed until that their value is stabilized [Cristiani]
- **Advantages**: anisotropic evolution
- **Disadvantages**: very long and only for monotone evolution

GFMM for non local velocity

- GFMM for the dislocation dynamics:

$$c(x,t) = c_0 \star 1_{\Omega_t}.$$

- Convergence result obtained using the comparison principle for the GFMM [Carlini, F., Monneau]

Dislocation line dynamics

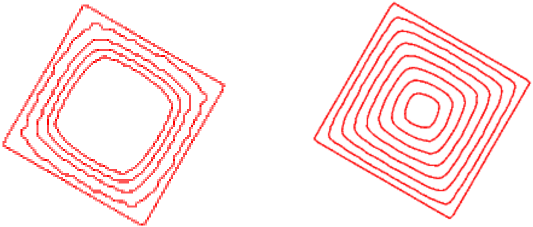
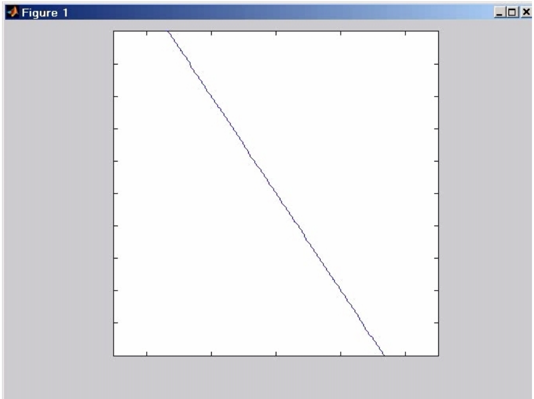


Figure : Finite difference (left) versus GFMM (right).

Plan

A straight line

$$c(x,t) = x_1$$



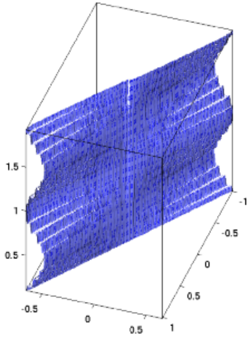
A straight line

	GFMM		**FD**	
Δx	$\mathcal{H}(\mathcal{C},\tilde{\mathcal{C}})$	CPU	$\mathcal{H}(\mathcal{C},\tilde{\mathcal{C}})$	CPU
0.04	$5.08 \cdot 10^{-2}$	$0.19s$	$4.10 \cdot 10^{-2}$	$1.82s$
0.02	$2.72 \cdot 10^{-2}$	$0.73s$	$2.05 \cdot 10^{-2}$	$13.2s$
0.01	$1.35 \cdot 10^{-2}$	$3.98s$	$1.03 \cdot 10^{-2}$	$102s$
0.005	$6.80 \cdot 10^{-3}$	$76s$	$2.60 \cdot 10^{-3}$	$810s$

Table : Hausdorff distance: GFMM versus Finite difference (FD)

A straight line with a velocity depended on time

$$c(x,t) = \sin(2\pi t)x_1$$



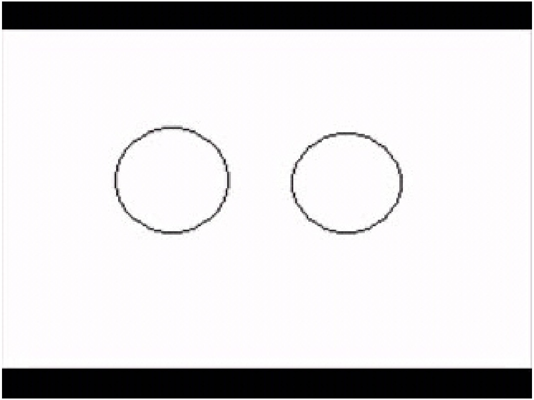
A straight line with a velocity depended on time

	GFMM		**FD**	
Δx	$\mathcal{H}(\mathcal{C},\tilde{\mathcal{C}})$	CPU	$\mathcal{H}(\mathcal{C},\tilde{\mathcal{C}})$	CPU
0.04	$5.21 \cdot 10^{-2}$	$0.52s$	$4.82 \cdot 10^{-2}$	$1.82s$
0.02	$3.07 \cdot 10^{-2}$	$1.71s$	$2.46 \cdot 10^{-2}$	$13.3s$
0.01	$1.54 \cdot 10^{-2}$	$10.5s$	$1.35 \cdot 10^{-2}$	$102s$
0.005	$9.00 \cdot 10^{-3}$	$130s$	$7.00 \cdot 10^{-3}$	$842s$

Table : Hausdorff distance: GFMM versus Finite difference (FD)

Two circles

$$c(x,t) = 1 - t$$



Application to image segmentation

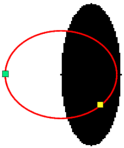
Chan-Vese model

We define the quantities

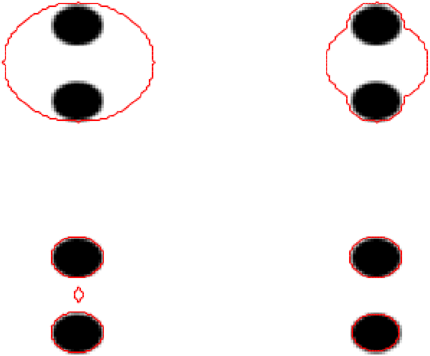
$$c_1(t) = \frac{\int_\Omega I(x)\frac{\theta(x,t)+1}{2}}{\int_\Omega \frac{\theta(x,t)+1}{2}} \quad c_2(t) = \frac{\int_\Omega I(x)\frac{1-\theta(x,t)}{2}}{\int_\Omega \frac{1-\theta(x,t)}{2}}$$

and the velocity

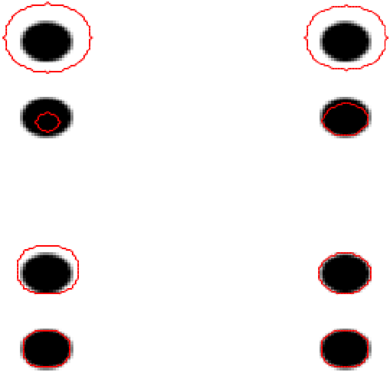
$$c(x,t) = (I(x) - c_2(t))^2 - (I(x) - c_1)^2$$



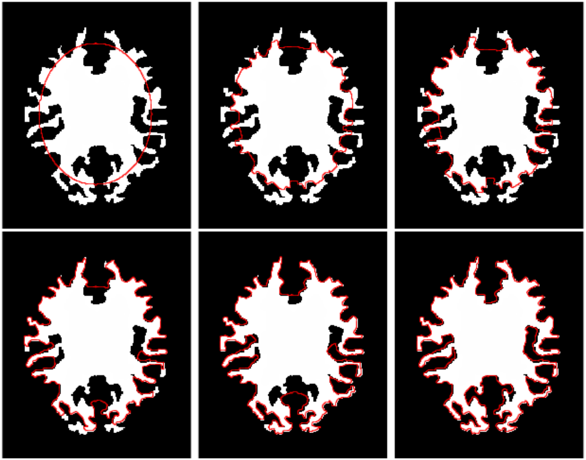
Segmentation



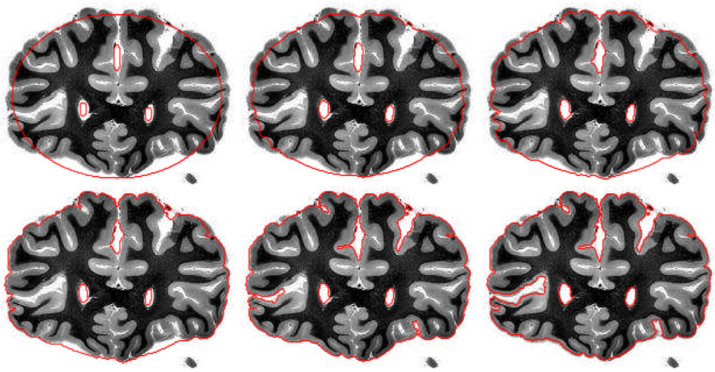
Segmentation



Segmentation



Segmentation



Medical data

Medical data

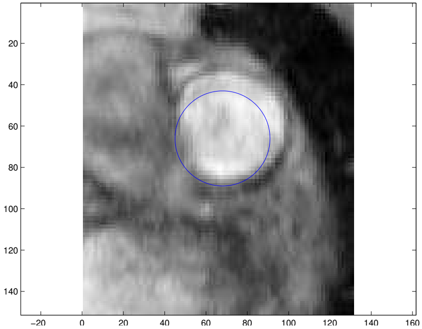


Figure : Initial data

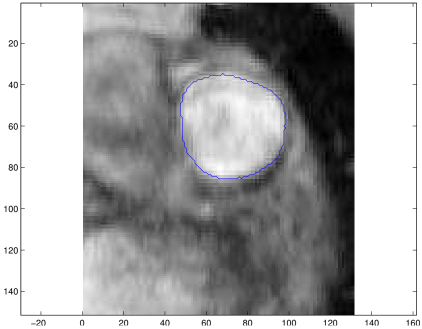
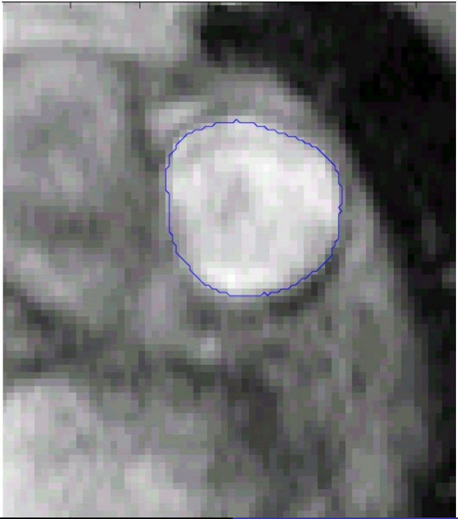


Figure : Final result

Calcul du débit d'une arthère



Main open problems

- (Non-monotone) anisotropic evolution
- Transport equation
- Mean curvature motion
- More general equations

Références

- E. Carlini, M. Falcone, N. Forcadel et R. Monneau, *Convergence of a Generalized Fast Marching Method for an Eikonal equation with a Velocity Changing Sign*, SINUM.
- E. Carlini, N. Forcadel et R. Monneau, *Generalized Fast Marching Method for dislocation dynamics*, (en cours).
- E. Cristiani, M. Falcone, *Fast semi-Lagrangian schemes for the Eikonal equation and applications*, SIAM J. Num. Anal.
- N. Forcadel, *Comparison principle for the Generalized Fast Marching Method*.
- N. Forcadel, C. Gout et C. Le Guyader, *Generalized Fast Marching Method: Applications to Image Segmentation*, Numerical Algorithms.
- S. Kim, *An O(N) level set method for eikonal equations*, SIAM J. Sci. Comput.
- J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. USA.
- J. N. Tsitsiklis, *Efficient algorithms for globally optimal trajectories*, IEEE Tran. Automatic. Control.
- A. Vladimirsky, *Static PDEs for time-dependent control problems*, Interfaces and Free Boundaries.