Generalized Fast Marching Method and applications

Nicolas Forcadel

Université Paris-Dauphine

Collaborators: E. Carlini, M. Falcone, C. Gout, C. Le Guyader and R. Monneau

Advancing numerical methods for viscosity solutions and applications

14-18 february 2011

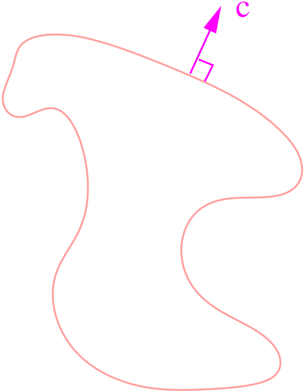
Plan

Plan

Plan

Plan

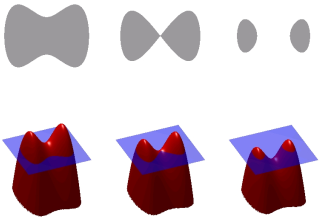
Evolution in the normal direction



$$\frac{d\Gamma_t}{dt} = c\, n_{\Gamma_t}, \quad c = c(x) > 0$$

The Level Set Method

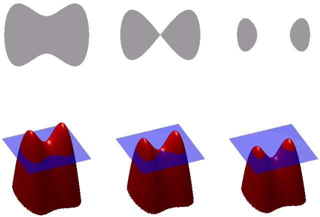
- Level Set Method [Osher, Sethian]: Very popular method for moving front problems
- Main idea: Represent the front implicitly by the zero level set of a function u



- Disadvantage : Add one dimension to the problem

The Level Set Method

- Level Set Method [Osher, Sethian]: Very popular method for moving front problems
- Main idea: Represent the front implicitly by the zero level set of a function u



- Disadvantage : Add one dimension to the problem

The Level Set Method

- Level set formulation :

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

- Formally:

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

with $\frac{\partial \Gamma_t}{\partial t} = c(x) n_{\Gamma_t},\ n_{\Gamma_t} = -\frac{\nabla u}{|\nabla u|}.$

- Level set equation for moving front:

$$u_t = c(x)|\nabla u|.$$

The Level Set Method

- Level set formulation :

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

- Formally:

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

with $\dfrac{\partial \Gamma_t}{\partial t} = c(x) n_{\Gamma_t},\ n_{\Gamma_t} = -\dfrac{\nabla u}{|\nabla u|}.$

- Level set equation for moving front:

$$u_t = c(x)|\nabla u|.$$

The Level Set Method

- Level set formulation :

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

- Formally:

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

with $\dfrac{\partial \Gamma_t}{\partial t} = c(x) n_{\Gamma_t},\ n_{\Gamma_t} = -\dfrac{\nabla u}{|\nabla u|}$.

- Level set equation for moving front:

$$u_t = c(x)|\nabla u|.$$

The Level Set Method

- Level set formulation :

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

- Formally:

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

with $\dfrac{\partial \Gamma_t}{\partial t} = c(x) n_{\Gamma_t},\ n_{\Gamma_t} = -\dfrac{\nabla u}{|\nabla u|}.$

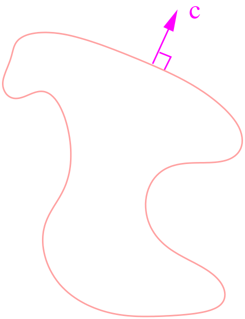
- Level set equation for moving front:

$$u_t = c(x)|\nabla u|.$$

Fast Marching Method

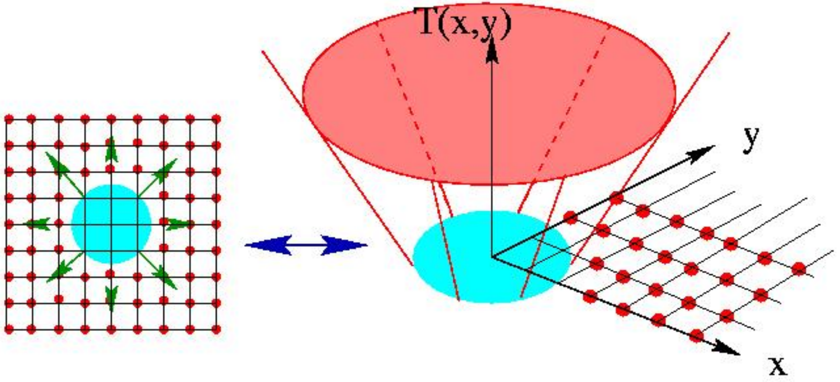
The goal of the Fast Marching Method is to solve in an efficient way the eikonal equation:

$$u_t = c(x)|\nabla u|, \quad c(x) > 0.$$

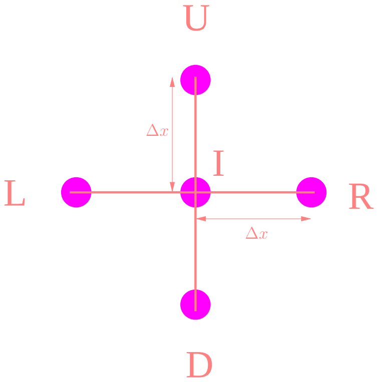


Stationary approach

- We set $u(x,t) = t - T(x) \Rightarrow |\nabla T(x)| = \dfrac{1}{c(x)}$.
- $T(x) =$ time at which the front reach the point x.



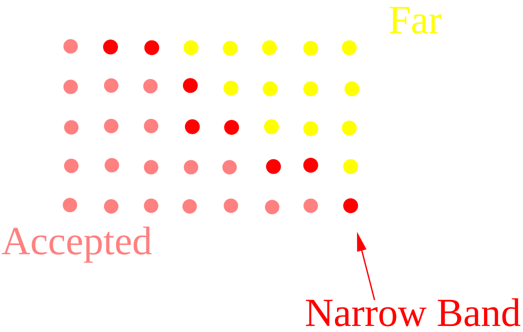
Discretization



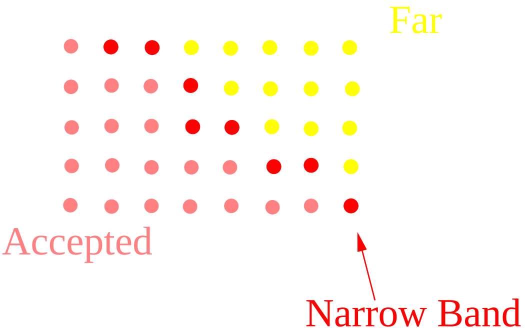
$$\begin{array}{c} \max(T_I - T_L, T_I - T_R, 0)^2 \\ + \\ \max(T_I - T_U, T_I - T_D, 0)^2 \end{array} = \left(\frac{\Delta x}{c_I}\right)^2 \quad (1)$$

Introduction of the Narrow Band

- By computing the nodes in a special order, one can find the solution in just one iteration. [Sethian], [Tsitsiklis]
  This special order corresponds to the increasing value of T
- Idea : Compute the time only near the front.

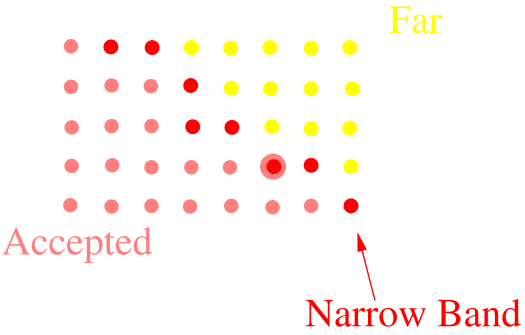


Introduction of the Narrow Band



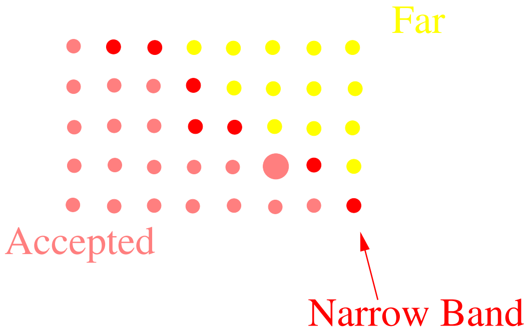
- Compute T_I on NB
- Call t_n the minimum of the time T_I on NB
  and accept at time t_n the points of minimum I
- The new NB is defined as the boundary of the new accepted region

Introduction of the Narrow Band



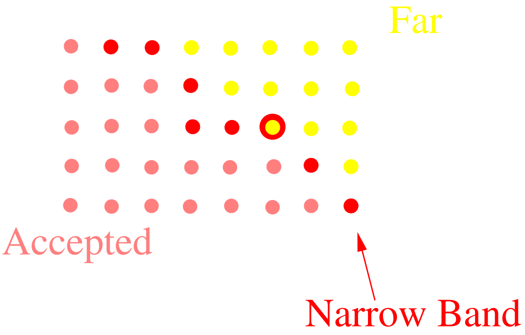
- Compute T_I on NB
- Call t_n the minimum of the time T_I on NB
  and accept at time t_n the points of minimum I
- The new NB is defined as the boundary of the new accepted region

Introduction of the Narrow Band



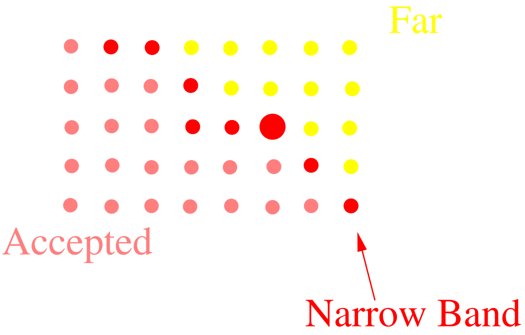
- Compute T_I on NB
- Call t_n the minimum of the time T_I on NB
  and accept at time t_n the points of minimum I
- The new NB is defined as the boundary of the new accepted region

Introduction of the Narrow Band



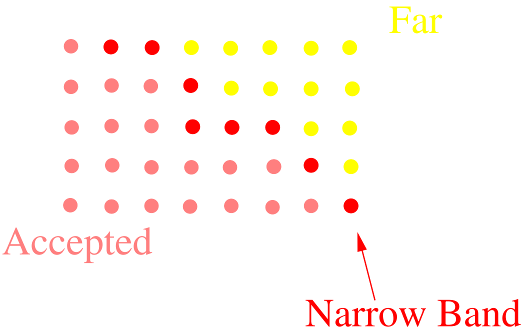
- Compute T_I on NB
- Call t_n the minimum of the time T_I on NB
  and accept at time t_n the points of minimum I
- The new NB is defined as the boundary of the new accepted region

Introduction of the Narrow Band



- Compute T_I on NB
- Call t_n the minimum of the time T_I on NB
  and accept at time t_n the points of minimum I
- The new NB is defined as the boundary of the new accepted region

Introduction of the Narrow Band



- Compute T_I on NB
- Call t_n the minimum of the time T_I on NB
  and accept at time t_n the points of minimum I
- The new NB is defined as the boundary of the new accepted region

Numerical complexity

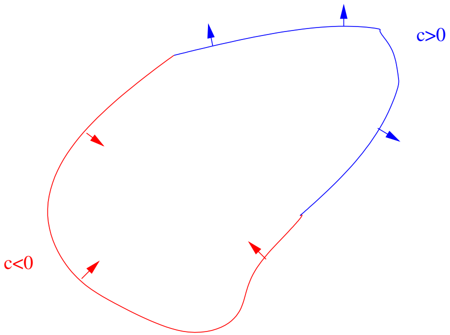
- Computation of the tentative value : at most 4 time for each node.
- Search for the minimum : Using a heap sort method this search costs $O(\ln(N_{NB}))$.
- The global cost is dominated by $O(N\ln(N))$ (N represents the total number of nodes in the grid).

References

- [Sethian]: http://math.berkeley.edu/~sethian/
- [Vladimirsky] : case $c = c(x,t) > 0$
- [Cristiani, Falcone] : Proof of convergence

General case: no stationnary representation

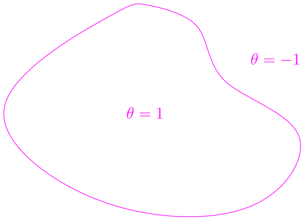
The notion of “Accepted” points is not adapted to generalize the algorithm.



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

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

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

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 $I \in NB^n$,

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

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 $I \in NB^n$,

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

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

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

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 $I \in NB^{n-1}$:

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

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

Algorithm

Loop

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

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

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

Algorithm

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

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 $n := n + 1$

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 $n := n + 1$

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 $n := n + 1$

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

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

Dependence in time of the velocity

- The time is given implicitly by the algorithm!

$$t_n = \min_{I \in NB} T_I$$

- Two difficulties:
  - t_n can be smaller than t_{n-1}:

$$t_n := t_{n-1}$$

  - t_n can be very large:

$$\Rightarrow \text{ Introduction of a time step } \Delta t:$$

$$\text{if } t_n > t_{n-1} + \Delta t \Rightarrow t_n := t_{n-1} + \Delta t$$

Dependence in time of the velocity

- The time is given implicitly by the algorithm!

$$t_n = \min_{I \in NB} T_I$$

- Two difficulties:
  - t_n can be smaller than t_{n-1}:

$$t_n := t_{n-1}$$

  - t_n can be very large:

$$\Rightarrow \text{ Introduction of a time step } \Delta t :$$

$$\text{if } t_n \geq t_{n-1} + \Delta t \Rightarrow t_n := t_{n-1} + \Delta t$$

Dependence in time of the velocity

- The time is given implicitly by the algorithm!

$$t_n = \min_{I \in NB} T_I$$

- Two difficulties:
  - t_n can be smaller than t_{n-1}:

$$t_n := t_{n-1}$$

  - t_n can be very large:

$$\Rightarrow \text{ Introduction of a time step } \Delta t :$$

$$\text{if } t_n \geq t_{n-1} + \Delta t \Rightarrow t_n := t_{n-1} + \Delta t$$

Dependence in time of the velocity

- The time is given implicitly by the algorithm!

$$t_n = \min_{I \in NB} T_I$$

- Two difficulties:
  - t_n can be smaller than t_{n-1}:

$$t_n := t_{n-1}$$

  - t_n can be very large:

$$\Rightarrow \text{ Introduction of a time step } \Delta t :$$

$$\text{if } t_n \geq t_{n-1} + \Delta t \Rightarrow t_n := t_{n-1} + \Delta t$$

Case $c(x,t) > 0$

Loop

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

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

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 $I \in NB^{n-1}$:

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

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 $I \in NB^{n-1}$:

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

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 $I \in NB^{n-1}$:

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

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

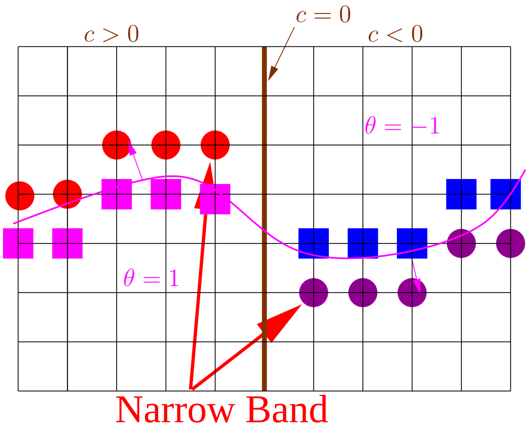
6 $\theta_I^n = \begin{cases} 1 & \text{if } I \in NA^n \\ \theta_I^{n-1} & \text{otherwise} \end{cases}$

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

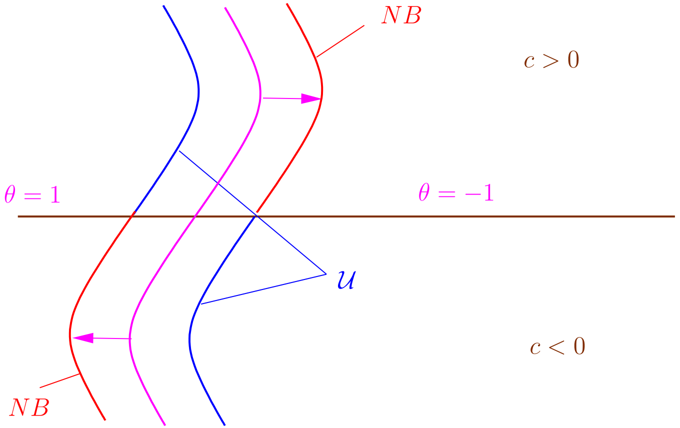
8 $n := n + 1$

Plan

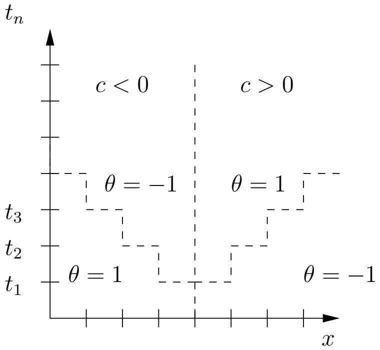
Idea of the GFMM



Schematic representation



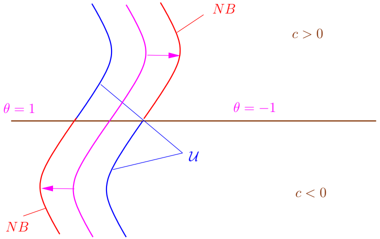
Regularisation of the speed



Definition

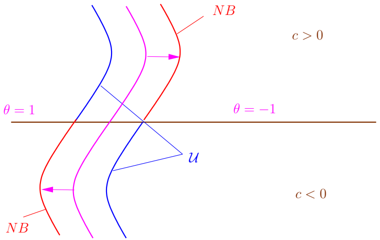
$$\hat{c}_I^n = \begin{cases} 0 & \text{if } \exists\, J \in V(I) \text{ t.q. } c_I^n c_J^n < 0 \text{ et } |c_I^n| < |c_J^n| \\ c_I^n & \text{otherwise} \end{cases}$$

Definition



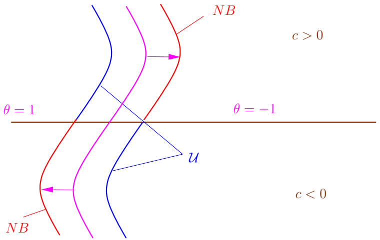
- $NB^n = \{I,\ \exists\ J \in V(I),\ \theta_I^n = -\theta_J^n \text{ and } \hat{c}_I^n \theta_I^n < 0\}$
- If $I \in NB^n$, $\mathcal{U}^n(I) = \{J \in V(I),\ \theta_J^n = -\theta_I^n\}$
  $\mathcal{U}^n = \cup_{I \in NB^n} \mathcal{U}^n(I)$
- $T_{J \to I}^n = \begin{cases} T_J^n & \text{if } J \in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

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

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 $I \in NB^{n-1}$:

$$\begin{array}{l} \max(\tilde{T}_I^{n-1} - T_{L\to I}^{n-1}, \tilde{T}_I^{n-1} - T_{R\to I}^{n-1}, 0)^2 \\ + \\ \max(\tilde{T}_I^{n-1} - T_{U\to I}^{n-1}, \tilde{T}_I^{n-1} - T_{D\to I}^{n-1}, 0)^2 \end{array} = \frac{(\Delta x)^2}{|c_I^{n-1}|^2}$$

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

8 $n := n + 1$

Convergence result

- For simplicity, assume that $t_{n+1} > t_n$
- We define

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

- 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

- For simplicity, assume that $t_{n+1} > t_n$
- We define

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

- 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

- For simplicity, assume that $t_{n+1} > t_n$
- We define

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

- 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)

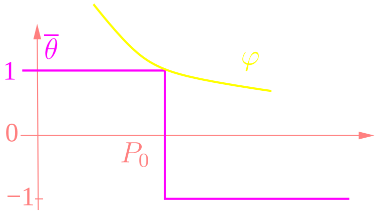
Under regularity assumptions on Ω_0 and c, we have that $\overline{\theta}$ is a sub solution and $\underline{\theta}$ is a super solution of

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

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:

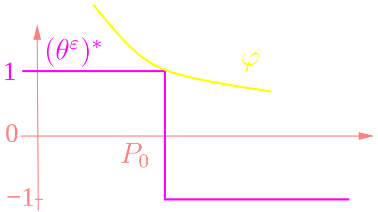


with

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

Idea of the proof

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

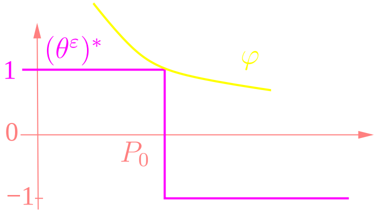


- Define ψ by

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

Idea of the proof

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



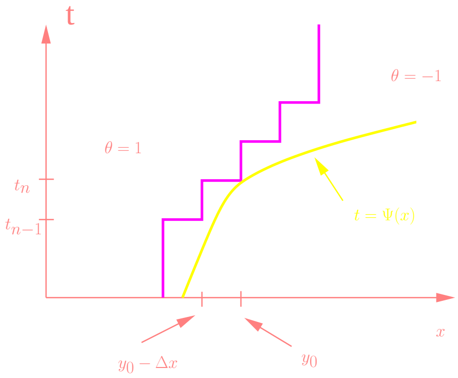
- Define ψ by

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

Idea of the proof

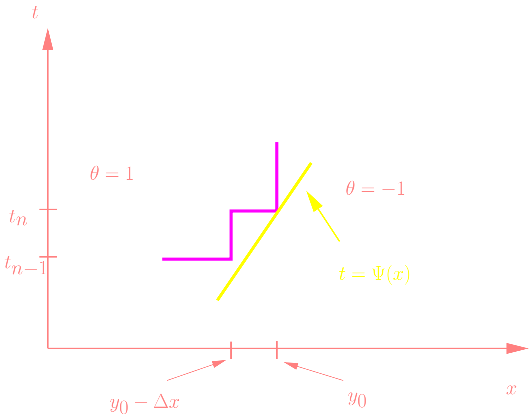
- To simplify, assume in 1D that

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

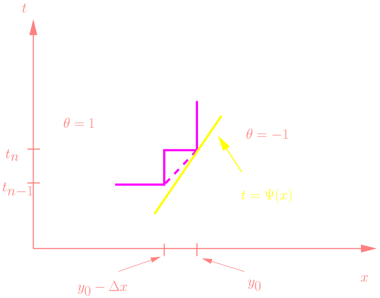


Idea of the proof

- To simplify, assume that φ is linear



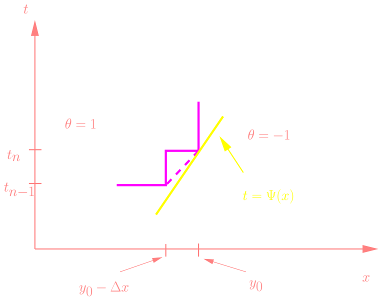
Idea of the proof



Then

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

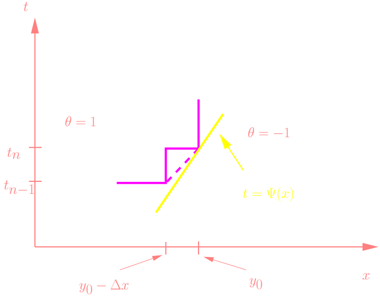
Idea of the proof



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

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

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*

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

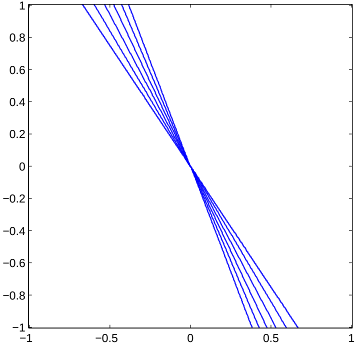
If $\Omega_u^0 \subset \Omega_v^0$ *then*

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

Plan

A straight line

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



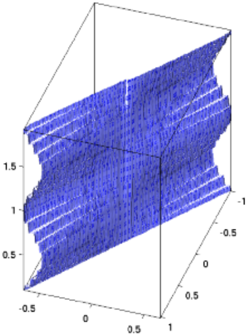
A straight line

	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)

Application to dislocation dynamics

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

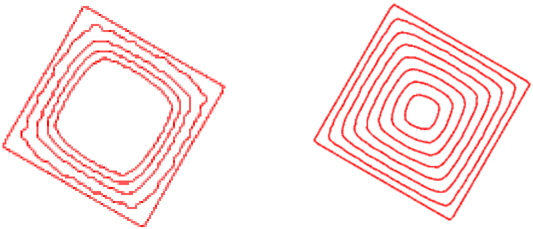


Figure: Finite difference (left) versus GFMM (right).

Application to image segmentation

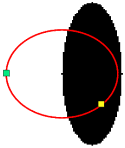
Chan-Vese model

We define the quantities

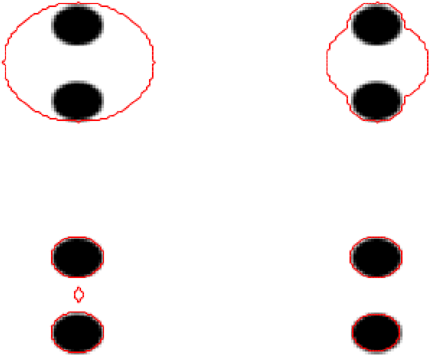
$$c_1(t) = \frac{\int_\Omega I(x)\frac{\theta(x,t)+1}{2}}{\int_\Omega \frac{\theta(x,t)+1}{2}} \quad c_2(t) = \frac{\int_\Omega I(x)\frac{1-\theta(x,t)}{2}}{\int_\Omega \frac{1-\theta(x,t)}{2}}$$

and the velocity

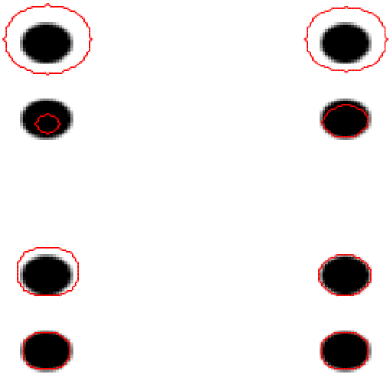
$$c(x,t) = (I(x) - c_2(t))^2 - (I(x) - c_1(t))^2$$



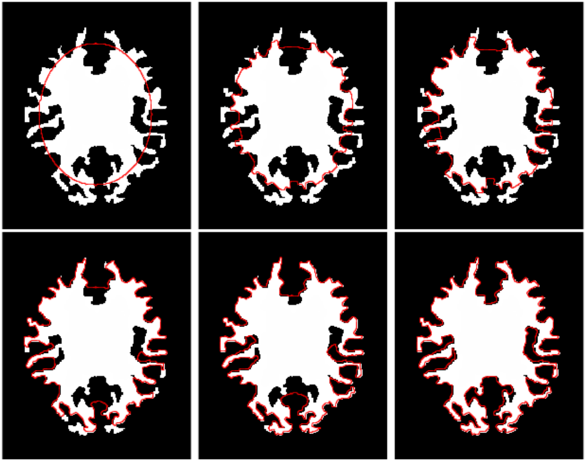
Segmentation



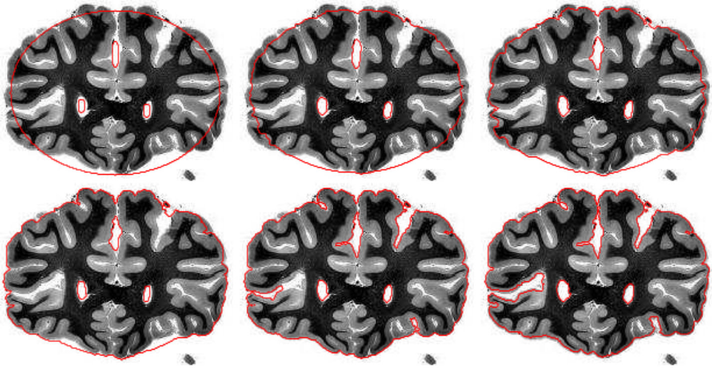
Segmentation



Segmentation



Segmentation



Medical data

Medical data

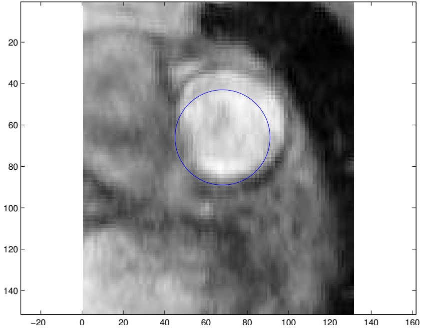


Figure: Initial data

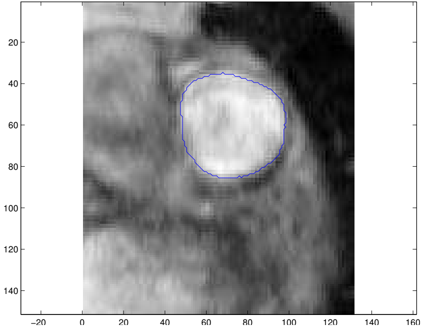


Figure: Final result

Computation of the arterial flow

Main open problems

- Error estimate for the GFMM
- (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*, SIAM J. Num. Anal.
- E. Carlini, N. Forcadel et R. Monneau, *Generalized Fast Marching Method for dislocation dynamics*, Preprint.
- N. Forcadel, *Comparison principle for the Generalized Fast Marching Method*, SIAM J. Num. Anal.
- N. Forcadel, C. Gout et C. Le Guyader, *Generalized Fast Marching Method: Applications to Image Segmentation*, Numerical Algorithms.
- 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.