Two new Ordered Upwind Methods for Hamilton-Jacobi equations

Emiliano Cristiani
(with S. Cacace and M. Falcone)

Advancing numerical methods for viscosity solutions and applications
Banff International Research Station - Banff, Canada

February 13–18, 2011

Related equation /1

Eikonal equation

$$\begin{cases} c(x)|\nabla u(x)| = 1\,, & x \in \mathbb{R}^n \backslash \Gamma_0 \\ u(x) = 0\,, & x \in \Gamma_0 \end{cases}$$

Kružkov transform: Defining $v = 1 - e^{-u}$ ($v \in [0, 1]$)

$$\begin{cases} v(x) + \max\limits_{a \in B(0,1)} \{c(x) a \cdot \nabla v(x)\} = 1\,, & x \in \mathbb{R}^n \backslash \Gamma_0 \\ v(x) = 0\,, & x \in \Gamma_0 \end{cases}$$

Related equation /1

Eikonal equation

$$\begin{cases} c(x)|\nabla u(x)| = 1\,, & x \in \mathbb{R}^n \backslash \Gamma_0 \\ u(x) = 0\,, & x \in \Gamma_0 \end{cases}$$

Kružkov transform: Defining $v = 1 - e^{-u}$ ($v \in [0,1]$)

$$\begin{cases} v(x) + \max_{a \in B(0,1)} \{c(x) a \cdot \nabla v(x)\} = 1\,, & x \in \mathbb{R}^n \backslash \Gamma_0 \\ v(x) = 0\,, & x \in \Gamma_0 \end{cases}$$

Anisotropic eikonal equation

$$v(x) + \max_{a \in B(0,1)} \{c(x,a) a \cdot \nabla v(x)\} = 1\,, \qquad x \in \mathbb{R}^n \backslash \Gamma_0$$

Related equation /2

Hamilton-Jacobi-Bellman equation

$$v(x) + \max_{a \in A}\{-f(x,a) \cdot \nabla v(x)\} = 1\,, \qquad x \in \mathbb{R}^n \backslash \Gamma_0$$

Related equation /2

Hamilton-Jacobi-Bellman equation

$$v(x) + \max_{a \in A}\{-f(x,a) \cdot \nabla v(x)\} = 1\,, \qquad x \in \mathbb{R}^n \backslash \Gamma_0$$

Hamilton-Jacobi-Isaacs equation

$$v(x) + \min_{b \in B}\max_{a \in A}\{-f(x,a,b) \cdot \nabla v(x)\} = 1\,, \qquad x \in \mathbb{R}^n \backslash \Gamma_0$$

Remarks

All these equations share some properties:

Remarks

All these equations share some properties:

- Information propagates from Γ_0 to the rest of the space along characteristics.

Remarks

All these equations share some properties:

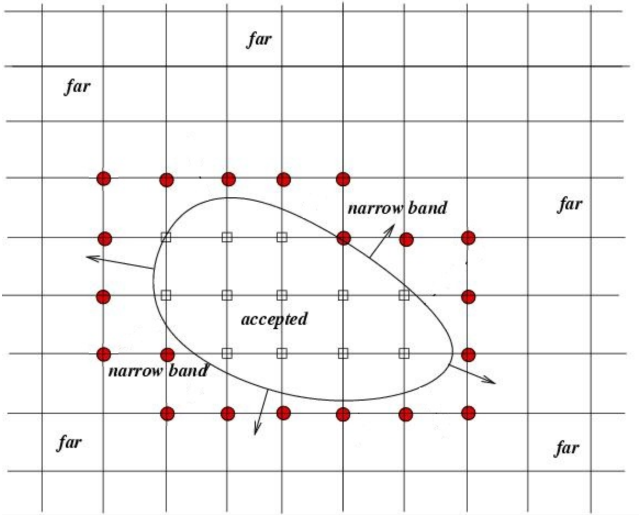
- Information propagates from Γ_0 to the rest of the space along characteristics.
- The solution v (or u) is increasing along characteristics.

Remarks

All these equations share some properties:

- Information propagates from Γ_0 to the rest of the space along characteristics.
- The solution v (or u) is increasing along characteristics.
- The t-level set $\Gamma_t = \{x \ : \ u(x) = t\}$ can be interpreted as an expanding front at time t.

The Fast Marching method



Limitations of FM method

The FM method accepts the node $X_{min} = \arg\min_{X \in NB} \{u(X)\}$ and enlarges the NB starting from that point.

⇒ The solution is computed following the gradient flow instead of the characteristic flow as required.

⇒ The FM works only for hyperbolic equations such that the gradient and the characteristic flow lie on the same simplex (*f.e.* the eikonal equation).

The FM method fails: an example

Anisotropic eikonal equation in $\mathbb{R}^2$

$$c(x,y,a_1,a_2)=\frac{1}{\sqrt{1+(\lambda a_1+\mu a_2)^2}}, \quad (a_1,a_2)\in B_2(0,1)\,, \quad \Gamma_0=(0,0)$$

Solution: $$u(x,y)=\sqrt{(1+\lambda^2)x^2+(1+\mu^2)y^2+2\lambda\mu xy}$$

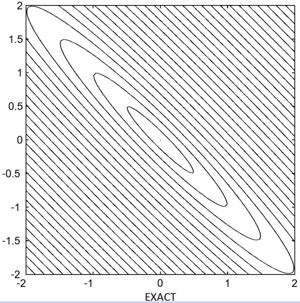
The FM method fails: an example

Anisotropic eikonal equation in $\mathbb{R}^2$

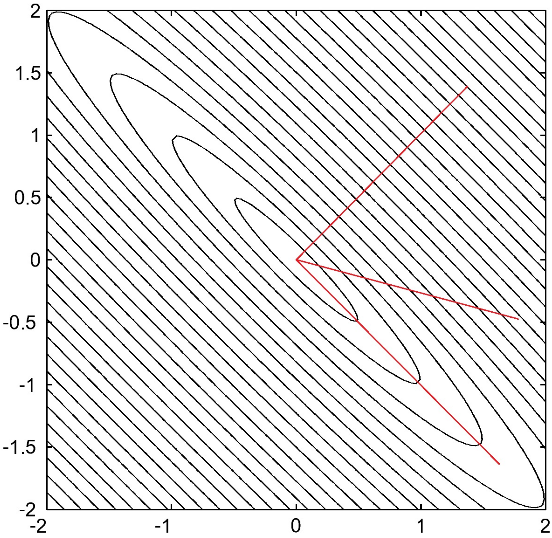
$$c(x,y,a_1,a_2)=\frac{1}{\sqrt{1+(\lambda a_1+\mu a_2)^2}}, \quad (a_1,a_2)\in B_2(0,1)\,, \quad \Gamma_0=(0,0)$$

Solution: $$u(x,y)=\sqrt{(1+\lambda^2)x^2+(1+\mu^2)y^2+2\lambda\mu xy}$$

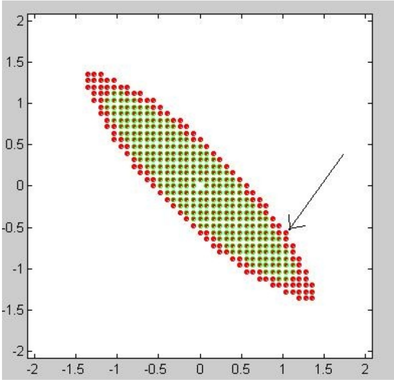
$$\lambda=5\,,\quad \mu=5$$

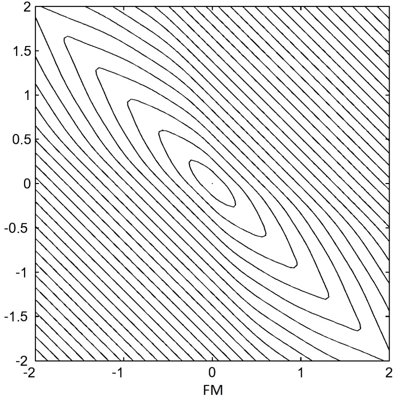


The FM method fails: an example



The FM method fails: an example



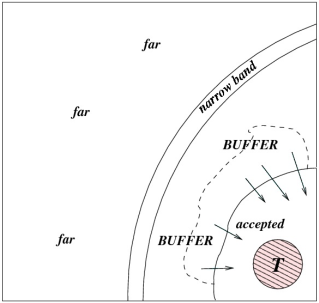


The Buffered Fast Marching method

The Buffered Fast Marching method

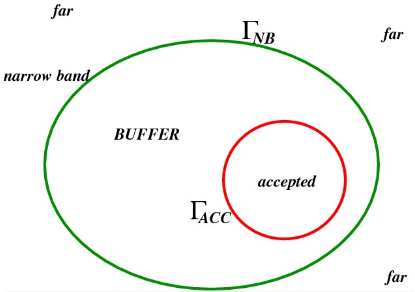
Buffered Fast Marching: main idea

In the BFM method the node in *NB* with the minimum value is not directly accepted but it is moved into a *buffer* region *BUF*. The node exits the buffer only when another accepting condition is satisfied.



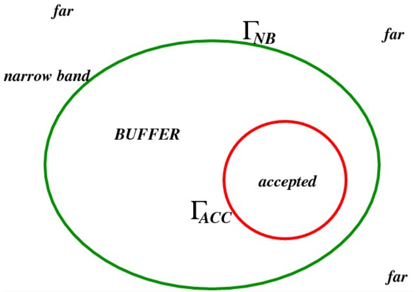
The minimal buffer size needed to accept at least one node depends on the anisotropy of the problem.

New condition to accept nodes



$\longrightarrow v$ with $v(\Gamma_{NB})$ and $v(\Gamma_{ACC})$ unchanged,
$\longrightarrow v_0$ with $v(\Gamma_{NB}) = 0$ and $v(\Gamma_{ACC})$ unchanged,
$\longrightarrow v_1$ with $v(\Gamma_{NB}) = 1$ and $v(\Gamma_{ACC})$ unchanged.

New condition to accept nodes



$$\begin{aligned}
&\longrightarrow v && \text{with } v(\Gamma_{NB}) \text{ and } v(\Gamma_{ACC}) \text{ unchanged,} \\
&\longrightarrow v_0 && \text{with } v(\Gamma_{NB}) = 0 \text{ and } v(\Gamma_{ACC}) \text{ unchanged,} \\
&\longrightarrow v_1 && \text{with } v(\Gamma_{NB}) = 1 \text{ and } v(\Gamma_{ACC}) \text{ unchanged.}
\end{aligned}$$

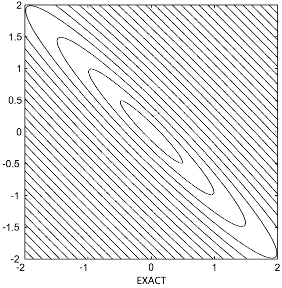
$$\text{new accepted nodes } = \{X \in BUF \ : \ v(X) = v_0(X) = v_1(X)\}.$$

BFM Algorithm

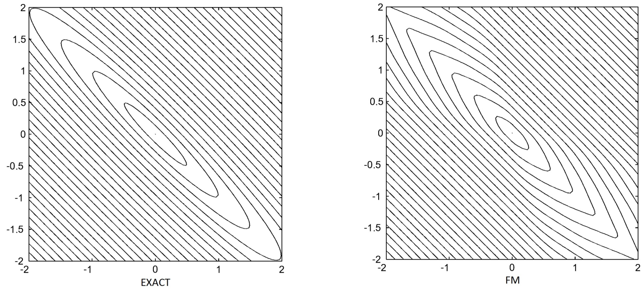
Modifications for the real algorithm

1. We remove from BUF and label as ACC the nodes whose value is changed less than a given tolerance ε.
2. $v = 0$ is substituted by $v_{min} = \min_{NB}\{v\}$.
3. v_1 is not computed.

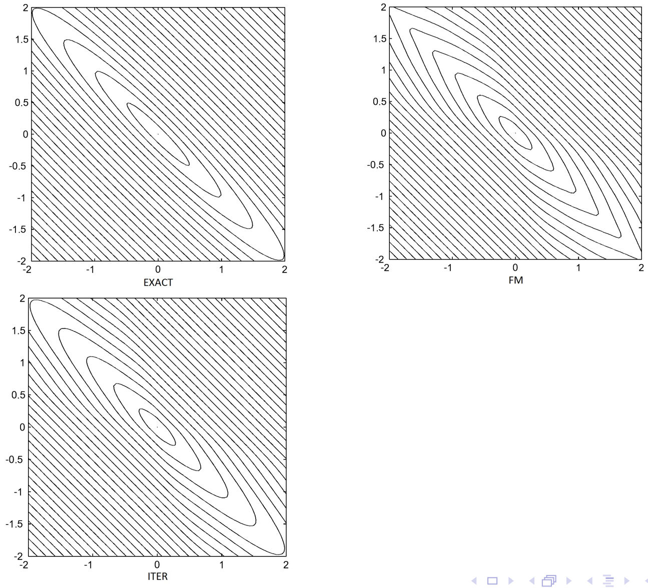
Test 1: Anisotropic front propagation



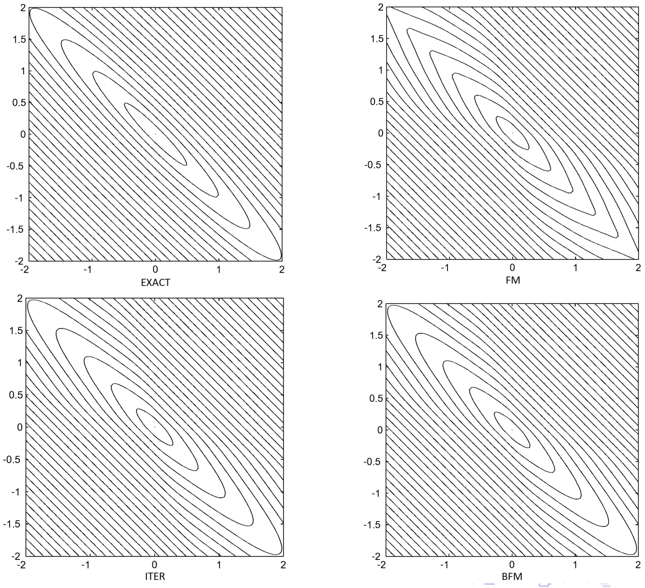
Test 1: Anisotropic front propagation



Test 1: Anisotropic front propagation



Test 1: Anisotropic front propagation



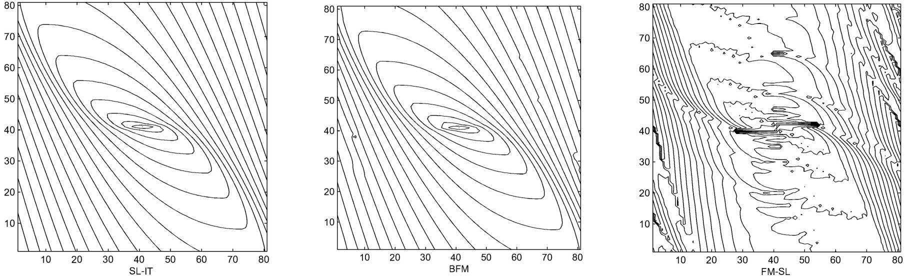
Test 1: Anisotropic front propagation

The L^1 error is computed with respect to the solution of the iterative algorithm accelerated by the Fast Sweeping method.

method	nodes	Δx	ε	L^1 error	CPU time (sec)
IT (FS)	100^2	0.04	–	–	2.49
BFM	100^2	0.04	10^{-3}	0.01	0.45
FM	100^2	0.04	–	1.02	0.09
IT (FS)	200^2	0.02	–	–	13.55
BFM	200^2	0.02	10^{-3}	0.02	1.67
FM	200^2	0.02	–	1.01	0.4

Test 2: Lunar landing

$$\Gamma_0 = (0,0)\,, \qquad f(x,y,a) = (y,a)\,, \quad a \in \{-1,1\}$$



ITER BFM FM

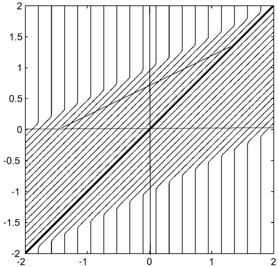
Test 2: Lunar landing

The L^1 error is computed with respect to the solution of the iterative algorithm accelerated by the Fast Sweeping method.

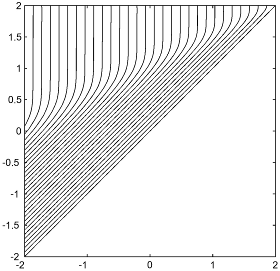
method	nodes	Δx	ε	L^1 error	CPU time (sec)
IT (FS)	100^2	0.1	–	–	0.67
BFM	100^2	0.1	10^{-4}	0.07	0.15
FM	100^2	0.1	–	3.21	0.02
IT (FS)	200^2	0.05	–	–	3.91
BFM	200^2	0.05	10^{-5}	0.05	2.05
FM	200^2	0.05	–	6.11	0.11

Test 3: Differential games with state constraints

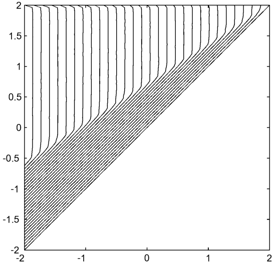
$$f(x, y, a, b) = (2a, b), \quad a \in [-1, 1], \quad b \in [-1, 1]$$



EXACT



BFM



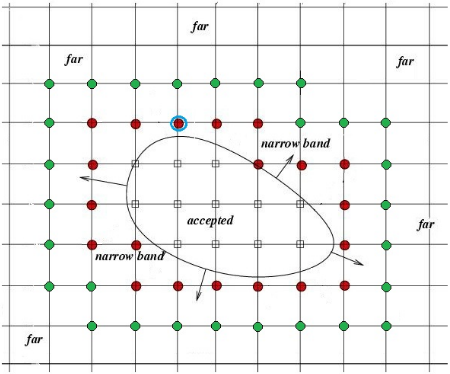
FM

The Progressive Fast Marching method

The Progressive Fast Marching method

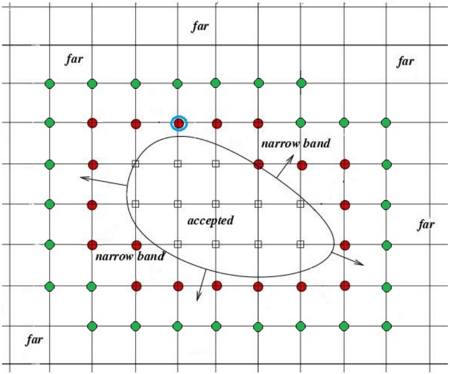
Progressive Fast Marching: main idea

The PFM method is inspired by the BFM, but **it is kept local**. The node to be accepted is found by means of computations which involve only the nodes in *NB* and in NB's first neighbours *NBN*.



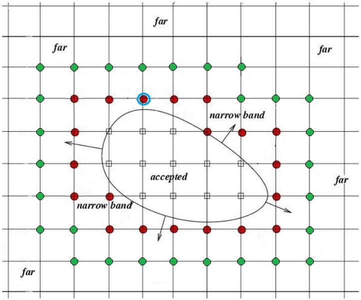
The algorithm /1

1. Solve the equation in *NB* iteratively until all values stabilize ($\Rightarrow$ at least one node has the "exact" value).
2. Find $v_{min} = \min_{X \in NB}\{v(X)\}$.
3. The value v_{min} is assigned to the nodes in *NBN*.



The algorithm /2

4. Re-solve the equation in NB and compare new and old values.
5. If $v_{new}(X) \neq v_{old}(X)$ for all $X \in NB$ it means that now all the values of the nodes in NB do not depend on ACC zone, and this is impossible because of step 1. Then, we slightly increment the value v_{min} and repeat the procedure until a node $Y \in NB$ satisfies $v_{new}(Y) = v_{old}(Y)$.
6. The node Y is labelled as ACC.



Some comments

- To our knowledge PFM method is the only one able to find the correct order of acceptance, keeping the computation local.

Some comments

- To our knowledge PFM method is the only one able to find the correct order of acceptance, keeping the computation local.
- PFM method recovers standard FM method when solving the eikonal equation.

Some comments

- To our knowledge PFM method is the only one able to find the correct order of acceptance, keeping the computation local.
- PFM method recovers standard FM method when solving the eikonal equation.
- More than one node per iteration can be accepted, as in Characteristic FM and Group FM methods.

Test 1: anisotropic front propagation

Anisotropic eikonal equation in $\mathbb{R}^2$

$$c(x, y, a_1, a_2) = \frac{1}{\sqrt{1 + (\lambda a_1 + \mu a_2)^2}}, \quad (a_1, a_2) \in B_2(0, 1), \quad \Gamma_0 = (0, 0)$$

Solution: $u(x, y) = \sqrt{(1 + \lambda^2)x^2 + (1 + \mu^2)y^2 + 2\lambda\mu xy}$

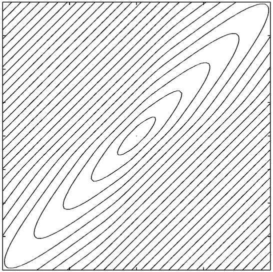
Test 1: anisotropic front propagation

Anisotropic eikonal equation in $\mathbb{R}^2$

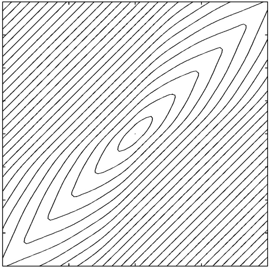
$$c(x,y,a_1,a_2)=\frac{1}{\sqrt{1+(\lambda a_1+\mu a_2)^2}},\qquad (a_1,a_2)\in B_2(0,1)\,,\qquad \Gamma_0=(0,0)$$

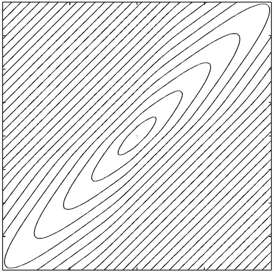
Solution: $$u(x,y)=\sqrt{(1+\lambda^2)x^2+(1+\mu^2)y^2+2\lambda\mu xy}$$

$\lambda=5\,,\quad \mu=5$

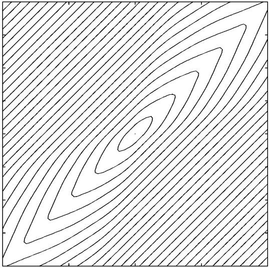


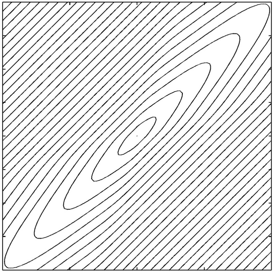
Test 1: anisotropic front propagation



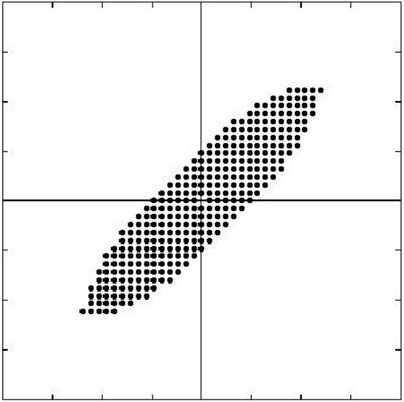


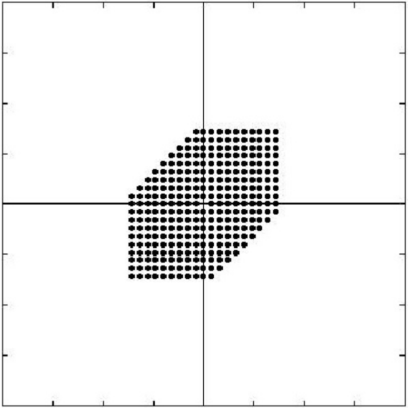
Test 1: anisotropic front propagation





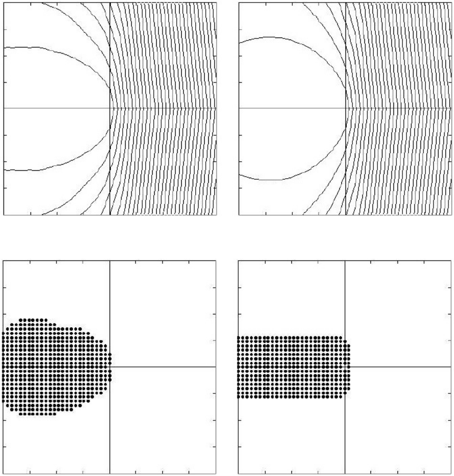
Test 1: anisotropic front propagation





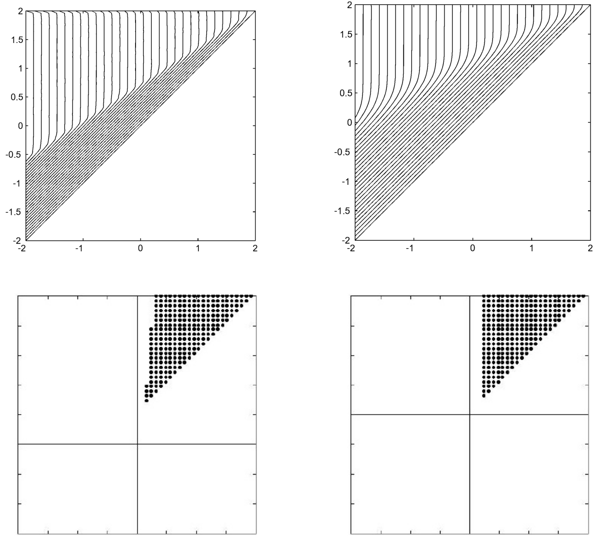
Test 2: Zermelo navigation problem

$$\Gamma_0 = (0,0)\,, \qquad f(x,y,a) = 2.1a + (2,0)\,, \quad a \in B_2(0,1)$$



Test 3: differential games with state constraints

$$f(x, y, a, b) = (2a, b), \quad a \in [-1, 1], \quad b \in [-1, 1]$$



References

1. E. Cristiani, *A fast marching method for Hamilton-Jacobi equations modeling monotone front propagations*, J. Sci. Comput., 39 (2009), 189–205.
2. S. Cacace, E. Cristiani, M. Falcone, *A local Ordered Upwind Method for Hamilton-Jacobi and Isaacs equations*, submitted to Proceedings of IFAC 2011.
3. J. A. Sethian, A. Vladimirsky, *Ordered upwind methods for static Hamilton-Jacobi equations: theory and algorithms*, SIAM J. Numer. Anal., 41 (2003), 325–363.