A multi-level fast-marching method for the minimum time problem

Marianne Akian[(1)], Stéphane Gaubert[(1)], Shanqing LIU[(2)]

[(1)] Inria and CMAP, École polytechnique CNRS, IP Paris
Marianne.Akian@inria.fr, Stephane.Gaubert@inria.fr

[(2)] CMAP, École polytechnique CNRS, IP Paris and Inria
Shanqing.Liu@polytechnique.edu.

See arXiv:2303.10705

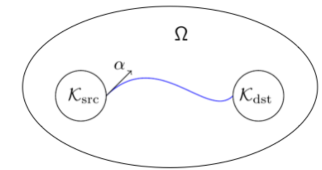
SMAI 2023, Le Gosier, Guadeloupe, 22-26 mai 2023

The Minimum Time Problem

Consider the minimum time optimal control problem:

$$\tau^* = \inf \ \tau$$

$$s.t. \begin{cases} \dot{y}(t) = f(y(t), \alpha(t))\alpha(t), \ \forall t \in [0, \tau], \\ y(0) \in \mathcal{K}_{\mathrm{src}}, \ y(\tau) \in \mathcal{K}_{\mathrm{dst}}, \\ y(t) \in \Omega \subset \mathbb{R}^d \ \forall t \in [0, \tau] \ , \\ \alpha \in \mathcal{A} = \{\alpha : \mathbb{R}^+ \to S_1 : \alpha \text{ is measurable}\} \ . \end{cases}$$

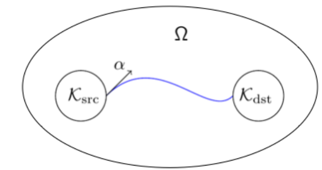


The Minimum Time Problem

Consider the minimum time optimal control problem:

$$\tau^* = \inf \ \tau$$

$$s.t. \begin{cases} \dot{y}(t) = f(y(t), \alpha(t))\alpha(t), \ \forall t \in [0, \tau], \\ y(0) \in \mathcal{K}_{\rm src}, \ y(\tau) \in \mathcal{K}_{\rm dst}, \\ y(t) \in \Omega \subset \mathbb{R}^d \ \forall t \in [0, \tau] \ , \\ \alpha \in \mathcal{A} = \{\alpha : \mathbb{R}^+ \to S_1 : \alpha \text{ is measurable}\} \ . \end{cases}$$



It can be solved using *Eikonal equation*:

$$\min_{\alpha \in S_1} \{(\nabla T(x) \cdot \alpha) f(x, \alpha)\} = 1 \ ,$$

or after the change of variable $v^* = 1 - e^{-\tau^*}$, the *stationary Hamilton-Jacobi Equation*:

$$F(x, v, Dv) = 0 \quad \text{with} \quad F(x, r, p) := -\min_{\alpha \in S_1} \{p \cdot f(x, \alpha)\alpha + 1 - r\} \ .$$

Numerical solution of Hamilton-Jacobi Equations

There are 2 difficulties:

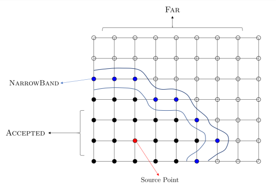
- Standard grid based space discretizations suffer from the *curse of dimensionality*:
  for an error of ϵ, the storage and time complexities of finite difference, finite element or semi-Lagrangian methods is at least in the order of $(1/\epsilon)^{d/2}$.
- For a *stationary equation*, one may need to do a number of value iterations in the order of $1/\epsilon$.

Numerical solution of Hamilton-Jacobi equations: previous improvements

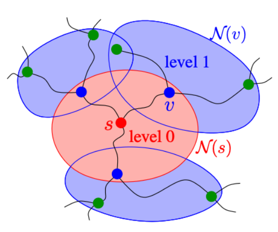
- For eikonal equations: Fast Marching Method introduced by Tsitsiklis (95) Sethian (96) is a "single pass" method.
- Recent developments: Sethian, Vladimirsky, OUMs(03), Cristiani, Falcone, SL-FM (07), Cristiani, BFM(09), Mirebeau, Riemannian FM (18).
- Computational complexity: $O(M \log M)$, where M is the number of discretization points. Feasible only in low dimension.
- Optimization along one or few "optimal" trajectories: Necessary conditions (Pontryagin principle); Direct optimization methods; Stochastic Dual Dynamic Programming method (SDDP) Pereira and Pinto (1991), Shapiro (2011),... for linear-convex problems, DP algorithm on a tree-structure Alla, Falcone, Saluzzi (2019) using Lipschitz continuity; Point based methods for Partially Observable Markov Decision Processes (POMDP) Pineau et al (2003), Kurniawati, Hsu, Lee (2008),...
- tropical/max-plus/idempotent methods: McEneaney (2007), Dower, Zhang (2015), Zheng Qu (2014),...

Idea of Multi-level Fast marching method

- The Fast Marching Method is a variant of the *Dijkstra's Algorithm*, which solves the shortest path problem in discrete time.
- Highway Hierarchies (Sanders, Schulte 12), accelerate the Dijkstra's algorithm (≈3000 times faster) in finding the shortest path between two given points.
- They construct coarse grids, like in the *Algebraic Multigrid Method*.
- For continuous minimum time problems, we shall use rather the ideas of the *Full Geometric Multigrid Method*, and *Highways* will be *optimal trajectories* on coarse grids.

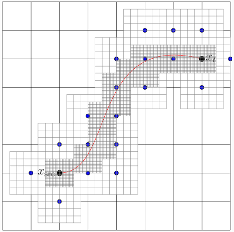


(a) Fast Marching



(b) Highway Hierarchies

⟹



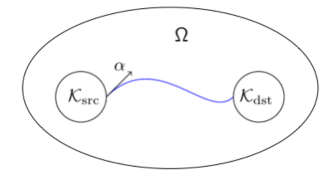
(c) ML Fast Marching

The Minimum Time Problem

Consider the minimum time optimal control problem:

$$\tau^* = \inf \ \tau$$

$$s.t. \begin{cases} \dot{y}(t) = f(y(t), \alpha(t))\alpha(t), \ \forall t \in [0, \tau], \\ y(0) \in \mathcal{K}_{\text{src}}, \ y(\tau) \in \mathcal{K}_{\text{dst}}, \\ y(t) \in \Omega \subset \mathbb{R}^d \ \forall t \in [0, \tau] \ , \\ \alpha \in \mathcal{A} = \{\alpha : \mathbb{R}^+ \to S_1 : \alpha \text{ is measurable}\} \ . \end{cases}$$



and $v^* = 1 - e^{-\tau^*}$,

---

where

- $\Omega \subset \mathbb{R}^d$ is compact, $\partial\Omega$ is C^2;
- $\mathcal{K}_{\text{src}}, \mathcal{K}_{\text{dst}} \subset \Omega$ closed;
- the speed $f > 0$ is continuous, Lipschitz continuous $w.r.t$ x and α, and $\forall x \in \partial\Omega, \alpha \in S_1$:

$$f(x, \alpha)\alpha \cdot n(x) \leq -C < 0.$$

Characterization of value function –”To Destination”

$$v^* = \inf_{x \in \mathcal{K}_{\rm src}} v_{\to\rm t}(x) \ , \quad v_{\to\rm t}(x) := \inf_{\alpha \in \mathcal{A}_{\Omega,x}} \inf_{\tau} \left\{ \int_0^\tau e^{-t} dt \mid y_\alpha(x;\tau) \in \mathcal{K}_{\rm dst} \right\} \ ,$$

where $y_\alpha(x;t)$ denote the solution of

$$\begin{cases} \dot y(t) = f(y(t), \alpha(t))\alpha(t), \ \forall t \geq 0 \ , \\ y(0) = x \ . \end{cases}$$

and

$$\mathcal{A}_{\Omega,x} := \{\alpha \in \mathcal{A} \mid y_\alpha(x;s) \in \overline{\Omega}, \text{for all } s \geq 0\} \ .$$

State constrained HJ Equation (in the viscosity sense, Soner) :

$$\begin{cases} F(x, v_{\to\rm t}(x), Dv_{\to\rm t}(x)) = 0, & x \in \Omega \setminus \mathcal{K}_{\rm dst}, \\ F(x, v_{\to\rm t}(x), Dv_{\to\rm t}(x)) \geq 0, & x \in \partial\Omega, \\ v_{\to\rm t}(x) = 0, & x \in \mathcal{K}_{\rm dst}; \end{cases} \tag{HJ$_{\rm t}$}$$

where $F(x,r,p) := -\min_{\alpha \in S_1}\{p \cdot f(x,\alpha)\alpha + 1 - r$.

Characterization of value function –"From Source"

$$v^* = \inf_{x \in \mathcal{K}_{\text{dst}}} v_{s\to}(x) \;, \quad v_{s\to}(x) := \inf_{\tilde{\alpha} \in \tilde{\mathcal{A}}_{\Omega,x}} \inf_{\tau} \left\{ \int_0^\tau e^{-t} dt \mid \tilde{y}_{\tilde{\alpha}}(x;\tau) \in \mathcal{K}_{\text{src}} \right\} \;,$$

where $\tilde{y}_{\tilde{\alpha}}(x;t)$ denote the solution of

$$\begin{cases} \dot{\tilde{y}}(t) = -f(\tilde{y}(t), \tilde{\alpha}(t))\tilde{\alpha}(t), \ \forall t \geq 0 \;, \\ \tilde{y}(0) = x \;. \end{cases}$$

and

$$\tilde{\mathcal{A}}_{\Omega,x} = \{\tilde{\alpha} \in \mathcal{A} \mid \tilde{y}_{\tilde{\alpha}}(x;s) \in \overline{\Omega}, \text{for all } s \geq 0\}.$$

State constrained HJ equation (in the viscosity sense, Soner) :

$$\begin{cases} F(x, v_{s\to}(x), -Dv_{s\to}(x)) = 0, & x \in \Omega \setminus \mathcal{K}_{\text{src}}, \\ F(x, v_{s\to}(x), -Dv_{s\to}(x)) \geq 0, & x \in \partial\Omega, \\ v_{s\to}(x) = 0, & x \in \mathcal{K}_{\text{src}}; \end{cases} \tag{HJ$_s$}$$

where $F(x, r, p) := -\min_{\alpha \in S_1}\{p \cdot f(x,\alpha)\alpha + 1 - r$.

The Optimal Trajectories

Denote the set of optimal points in $\mathcal{K}_{\mathrm{src}}, \mathcal{K}_{\mathrm{dst}}$:

$$\mathcal{X}_{\mathrm{src}} = \mathrm{Argmin}_{x \in \mathcal{K}_{\mathrm{src}}} v_{\rightarrow \mathrm{t}}(x), \ \mathcal{X}_{\mathrm{dst}} = \mathrm{Argmin}_{x \in \mathcal{K}_{\mathrm{dst}}} v_{\mathrm{s} \rightarrow}(x) \ .$$

Denote Γ_x^*, $\tilde{\Gamma}_x^*$ the set of geodesic points (points of optimal trajectories) for the two directions' problems.

Proposition

$\cup_{x \in \mathcal{X}_{\mathrm{src}}} \{\Gamma_x^*\} = \cup_{x \in \mathcal{X}_{\mathrm{dst}}} \{\tilde{\Gamma}_x^*\} := \Gamma^*$, geodesic points from $\mathcal{K}_{\mathrm{src}}$ to $\mathcal{K}_{\mathrm{dst}}$.

$$\begin{aligned} \inf_{x \in \mathcal{K}_{\mathrm{src}}} v_{\rightarrow \mathrm{t}}(x) &= \inf_{x \in \mathcal{K}_{\mathrm{dst}}} v_{\mathrm{s} \rightarrow}(x) := v^* \\ &\leq \mathcal{F}_v(x) := \{v_{\mathrm{s} \rightarrow}(x) + v_{\rightarrow \mathrm{t}}(x) - v_{\mathrm{s} \rightarrow}(x) v_{\rightarrow \mathrm{t}}(x)\} \ . \end{aligned}$$

$\mathcal{F}_v(x) = v^* \Leftrightarrow x \in \Gamma^*$.

The above formula is similar to:

$$\tau^* \leq T_{\mathrm{s} \rightarrow}(x) + T_{\rightarrow \mathrm{t}}(x) \ .$$

The Restricted State Constraint

We then can define two families of neighborhoods of optimal trajectories:

$$\mathcal{O}_\eta = \{x \in (\Omega \setminus (\mathcal{K}_{\mathrm{src}} \cup \mathcal{K}_{\mathrm{dst}})) \mid \mathcal{F}_v(x) < \inf_{y \in \Omega} \mathcal{F}_v(y) + \eta \} \ .$$

$$\Gamma^\delta := \text{the set of } \delta\text{−geodesic points from } \mathcal{K}_{\mathrm{src}} \text{ to } \mathcal{K}_{\mathrm{dst}}.$$

Proposition

For every $0 < \delta < \eta$, and $\delta' > 0$, we have: $\Gamma^* \subseteq \Gamma^\delta \subseteq \overline{\mathcal{O}}_\eta \subseteq \Gamma^{\eta+\delta'}$.

Denote $v^\eta_{\mathrm{s}\to}$, $v^\eta_{\to\mathrm{t}}$ the value function of the problem in $\mathcal{O}_\eta$ instead of Ω, then:

Theorem

For every $\delta < \eta$, and $x \in \Gamma^\delta$: $v^\eta_{\mathrm{s}\to}(x) = v_{\mathrm{s}\to}(x), \qquad v^\eta_{\to\mathrm{t}}(x) = v_{\to\mathrm{t}}(x)$.

The Restricted State Constraint

We then can define two families of neighborhoods of optimal trajectories:

$$\mathcal{O}_\eta = \{x \in (\Omega \setminus (\mathcal{K}_{\mathrm{src}} \cup \mathcal{K}_{\mathrm{dst}})) \mid \mathcal{F}_v(x) < \inf_{y \in \Omega} \mathcal{F}_v(y) + \eta \} .$$

$$\Gamma^\delta := \text{the set of } \delta\text{−geodesic points from } \mathcal{K}_{\mathrm{src}} \text{ to } \mathcal{K}_{\mathrm{dst}}.$$

Proposition

For every $0 < \delta < \eta$, and $\delta' > 0$, we have: $\Gamma^* \subseteq \Gamma^\delta \subseteq \overline{\mathcal{O}}_\eta \subseteq \Gamma^{\eta+\delta'}$.

Denote $v^\eta_{\mathrm{s}\to}$, $v^\eta_{\to\mathrm{t}}$ the value function of the problem in $\mathcal{O}_\eta$ instead of Ω, then:

Theorem

For every $\delta < \eta$, and $x \in \Gamma^\delta$: $v^\eta_{\mathrm{s}\to}(x) = v_{\mathrm{s}\to}(x), \qquad v^\eta_{\to\mathrm{t}}(x) = v_{\to\mathrm{t}}(x)$.

Moreover, $v^\eta_{\to\mathrm{t}}$ is solution of

$$\begin{cases} F(x, v, Dv(x)) = 0, & x \in \mathcal{O}_\eta, \\ F(x, v(x), Dv(x)) \geq 0, & x \in \partial\mathcal{O}_\eta \setminus \mathcal{K}_{\mathrm{dst}}, \\ v(x) = 0, & x \in \partial\mathcal{O}_\eta \cap \partial\mathcal{K}_{\mathrm{dst}}. \end{cases} \qquad (\mathrm{HJ}^\eta_{\mathrm{t}})$$

Idea of The Multilevel Algorithm

- Solve the $(\mathrm{HJ_s})$ and $(\mathrm{HJ_t})$ in COARSE-GRID.
- Approximate O_η using the approximate value function: $V_{\mathrm{s}\to}, V_{\to\mathrm{t}}$.
- Build FINE-GRID in $\mathcal{O}_\eta$ only, solve $(\mathrm{HJ_s^\eta})$, $(\mathrm{HJ_t^\eta})$ in FINE-GRID.
- Repeat from level to level ...

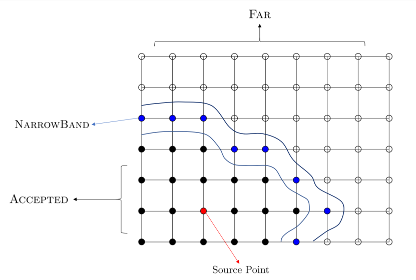
Idea of The Multilevel Algorithm

- Solve the (HJ_s) and (HJ_t) in COARSE-GRID.
- Approximate O_η using the approximate value function: $V_{\mathrm{s}\to}$, $V_{\to\mathrm{t}}$.
- Build FINE-GRID in $\mathcal{O}_\eta$ only, solve $(\mathrm{HJ}_\mathrm{s}^\eta)$, $(\mathrm{HJ}_\mathrm{t}^\eta)$ in FINE-GRID.
- Repeat from level to level ...

And we associate this idea with Fast Marching Algorithm.

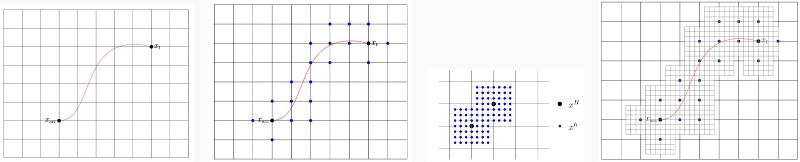
The Fast Marching Method

- An efficient **single-pass** method to solve stationary HJ PDEs.
- Need a full discretization (finite differences, semi-Lagrangian scheme,...) in the form of the fixed point equation of an *update operator* $\mathcal{U}$: $v = \mathcal{U}(v)$.
- The nodes are divided by FAR, ACCEPTED , NARROWBAND .
- At each step, one node x from NARROWBAND with minimum value $v(x)$ will be added to ACCEPTED , and the NARROWBAND will be updated.
- The computational complexity is $O(M \log M)$, with $M =$ number of nodes.
- *Partial Fast Marching* stops once ACCEPTED contains the points of interest ($\mathcal{K}_{\text{dst}}$ or $\mathcal{K}_{\text{src}}$).



Two Level Fast Marching Method (2LFM)

- In coarse grid:
  i. Do a two direction partial fast marching in the grid X^H $\longrightarrow$ $V^H_{s\to}$ and $V^H_{\to t}$.
  ii. Select *Active* nodes based on the two approximate value functions, and store them $\longrightarrow$ $O^H_\eta := \{x \in X^H \mid \mathcal{F}_{V^H}(x) \leq \min_{x^H \in X^H}\left(\mathcal{F}_{V^H}(x^H)\right) + \eta_H\}$.
- In fine grid:
  i. Construct fine grid nodes based on *Active* set O^H_η $\longrightarrow$ $G^h_\eta = \{x \in X^h \mid \exists x^H \in O^H_\eta : \|x - x^H\| \leq max((H-h), h)\}$.
  ii. Do fast marching starting from one direction in fine grid nodes only $\longrightarrow$ $V^{h,2}_{s\to}$ or $V^{h,2}_{\to t}$.



(d) Level-0 (e) Active Nodes (f) Fine grid (g) Level-1

2LFM: Proof of Correctness

link the fine and coarse grids as follows:

- Extend the approximate solutions from X^H to Ω by linear interpolation $\longrightarrow \quad V^{H,I}_{\mathrm{s}\to}$ or $V^{H,I}_{\to\mathrm{t}}$.
- Define

$$O^{H,I}_{\eta} = \{x \in (\Omega \setminus (\mathcal{K}_{\mathrm{src}} \cup \mathcal{K}_{\mathrm{dst}})) \mid \mathcal{F}_{V^{H,I}}(x) \leq \min_{x^H \in X^H} \mathcal{F}_{V^H}(x^H) + \eta_H \} \supset O^H_{\eta} \ .$$

Theorem

Assume $\|v^h_{\mathrm{s}\to} - v_{\mathrm{s}\to}\| \leq Ch^{\gamma}$.

There exists a constant $C \geq 0$ *such that for every* $\eta_H \geq CH^{\gamma}$, $\overline{O}^{H,I}_{\eta}$ *contains the optimal trajectories* Γ^* *of the continuous problem.*

Applying the results in continuous time and space, we obtain:

Theorem

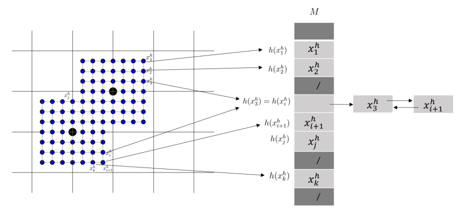
There exist $\delta < \eta_H$ such that, for every $x \in G^h_{\eta} \cap \Gamma_{\delta}$, $V^{h,2}_{\mathrm{s}\to}(x) = V^h_{\mathrm{s}\to}(x)$.

Multi-level Fast Marching Method (MLFM)

- The above algorithm can be extended to multi-level case.
- The parameter: H_l, η_l for every $l \in \{1, 2, ..., N\}$
- The fine grid in current level will be the coarse grid in next level.
- Same analysis in each level: proof of correctness.

Data Storage

- For the algorithm to be efficient, we need to avoid storing full grids.
- We use a hash table, which has space complexity $O(M)$, and computational complexity $O(1)$ for:
  - Checking if one node already exists in the grid;
  - If not, insert this new node into the existing grid;
  - After the grid has been constructed, checking neighborhood's information for each node.
- Point $\Rightarrow$(Hash Function)$\Rightarrow$Pointer$\Longrightarrow$Point+Value Function+...



Analysis of Computational Complexity (2 Level Case)

Given the mesh step h of fine grid, two parameters need to be chosen:

i. The mesh step of the coarse grid H.
ii. The parameter η_H to select the active nodes in coarse grid.

Space Complexity

Assume there exists a finite number of optimal trajectories, that the distance between Γ^* and $\mathcal{O}_\eta$ is in the order of η^β, and $\eta_H \geq C_\gamma H^\gamma$. Then, the space complexity of 2LFM is:

$$\mathcal{C}_{spa}(H,h) = \widetilde{O}\Big(C^d\Big(\frac{1}{H^d} + \frac{(\eta_H)^{\beta(d-1)}}{h^d}\Big)\Big) \ ,$$

where C depends in particular on the Euclidean distance D between $\mathcal{K}_{\mathrm{src}}$ and $\mathcal{K}_{\mathrm{dst}}$.

For semilagrangian schemes, the same estimation holds for time complexity. One can obtain by induction a similar result for several levels, and then optimize the mesh steps.

Theorem (Space or time computational complexity)

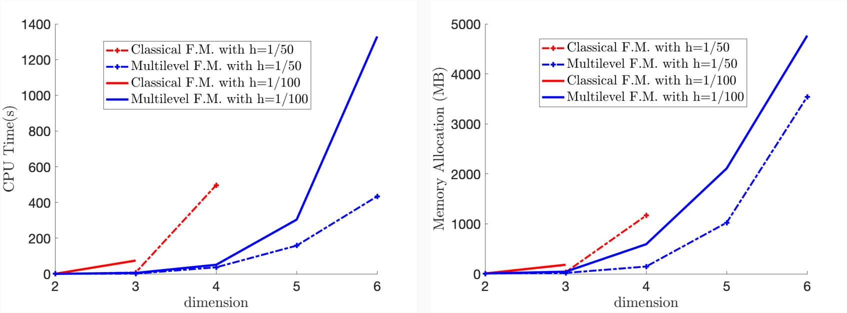
Assume $d \geq 2$, and let $\nu := \gamma\beta(1 - \frac{1}{d}) < 1$. Let $\varepsilon > 0$, and choose $h = (C_\gamma^{-1}\varepsilon)^{\frac{1}{\gamma}}$. Then, in order to obtain an error bound on the value of the minimum time problem $\leq \varepsilon$, one can use one of the following methods:

1. *2LFM (two-level fast marching method) with $\eta_H = CH^\gamma$, and $H = h^{\frac{1}{\nu+1}}$. Then, the computational complexity is $\widetilde{O}(C^d(\frac{1}{\varepsilon})^{\frac{d}{\gamma(1+\nu)}})$.*
2. *The $N-$level MLFM (fast marching method) with $\eta_l = CH_l^\gamma$ and $H_l = h^{\frac{1-\nu^l}{1-\nu^N}}$. Then, the computational complexity is $\widetilde{O}(NC^d(\frac{1}{\varepsilon})^{\frac{1-\nu}{1-\nu^N}\frac{d}{\gamma}})$.*
3. *The $N-$level MLFM with $N = \lfloor \frac{d}{\gamma}\log(\frac{1}{\varepsilon})\rfloor$, and $\eta_l = CH_l^\gamma$ and $H_l = h^{\frac{l}{N}}$. Then, the computational complexity is $\widetilde{O}(C^d(\frac{1}{\varepsilon})^{(1-\nu)\frac{d}{\gamma}})$. When $\gamma = \beta = 1$, it reduces to $\widetilde{O}(C^d\frac{1}{\varepsilon})$.*

- The computational complexity of Fast Marching is $\widetilde{O}(C^d(\frac{1}{\varepsilon})^{\frac{d}{\gamma}})$.
- $\gamma = 1/2$ in general and $\gamma = 1$ when f, $v_{\rightarrow\mathrm{t}}$ and $v_{\mathrm{s}\rightarrow}$ are semi-concave.
- $\beta = 1/2$ for $\mathcal{C}^2$ value functions but $\beta = 1$ for some Lipschitz cases.

Some Numerical Results

- FM and MLFM are implemented in C++, and executed on a single core of a Quad Core IntelCore I7 at 2.3Gh with 16Gb of RAM, and will be on gitlab.inria.fr soon.
- They have been tested on several minimum time problems. An easy problem with $f \equiv 1$ (Problem 1), a problem with discontinuous speed (Problem 2), a Poincaré Model (Problem 3), a problem for which $\beta = 1$ (Problem 4),... See arXiv:2303.10705.
- Up to dimension 6 for MLFM.



(h) CPU time

(i) Memory allocation

Figure 1: CPU time and memory allocation as a function of the dimension, for a fixed finest mesh step h.

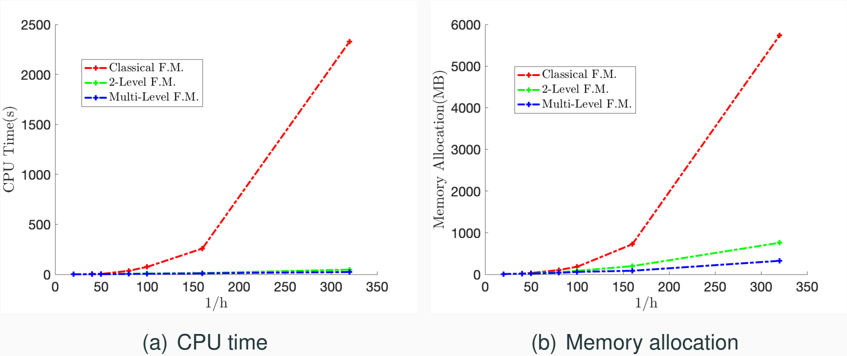
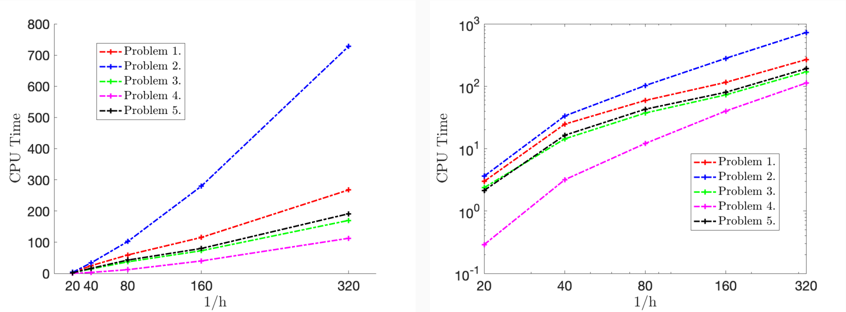


Figure 2: CPU time and memory allocation for several values of the finest mesh step h, in dimension 3.



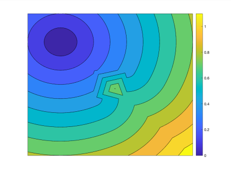
(a) Linear scale.

(b) Log-log scale.

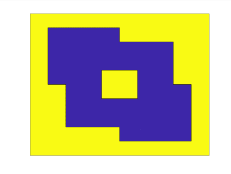
Figure 3: Growth of CPU time w.r.t. mesh steps in dimension 4.

Several other cases to show how the algorithm works

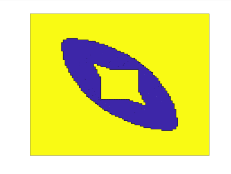
Optimal trajectories with obstacles:



(a) Classical F.M.

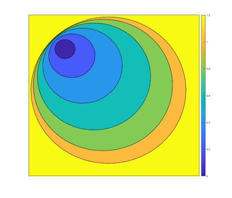


(b) Level-1

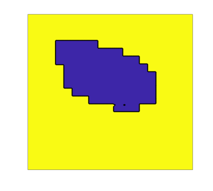


(c) Level-2

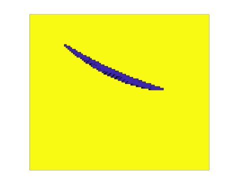
Poincare Disk:



(d) Classical F.M.



(e) Level-1



(f) Level-2

Conclusion and Perspectives

- New numerical method using *multilevel discretizations* and a characterization of optimal trajectories based on two directions value functions.
- Time and space *complexity* to obtain an ε-error is reduced to $\widetilde{O}(\varepsilon^{-\frac{(1-\nu)d}{\gamma}})$, $\nu := \gamma\beta(1-\frac{1}{d})$, compared with $\widetilde{O}(\varepsilon^{-\frac{d}{\gamma}})$ for ordinary grid-based methods.
- When $\gamma = \beta = 1$ (for instance for a semi-concave value function which is stiff around optimal trajectories), this complexity becomes in $\widetilde{O}(\varepsilon^{-1})$.
- *Numerical experiments* have been done up to dimension 7 on a laptop.

- Finite horizon deterministic control problems and *tropical numerical methods* (arxiv:2304.10342, will be presented at IFAC 2023 by Shanqing Liu).
- *Infinite horizon* discounted problems with value iteration?
- The method need to guess the constants (γ, β and C in $\eta_H = CH^\gamma$) or to tune the parameters H_l and η_l. *Adaptive construction* ?
- Extension to *stochastic control problems* ?