HJB-RBF based approach for the control of PDEs

Alessandro Alla
work in collaboration with H. Oliveira, G. Santin

Control in Times of Crisis
Online Seminar

13/05/2021

Outline

Outline

Optimal control and DPP

Dynamical Systems

$$\begin{cases} \dot{y}(t) = f(y(t), u(t)), \ \ t \in (0, \infty), \\ y(0) = x \in \mathbb{R}^d \end{cases}$$

Cost Functional

$$\mathcal{J}_x(y, u) \equiv \int_0^\infty g(y(s), u(s)) e^{-\lambda s} ds$$

Value Function

$$v(x) = \inf_{u \in \mathcal{U}} \mathcal{J}_x(y, u)$$

Optimal control and DPP

Feedback Control

$$u^*(t) = \arg\min_{u \in \mathcal{U}} \{g(x,u) + \nabla v(x) \cdot f(x,u)\}$$

Dynamical Programming Principle (DPP)

$$v(x) = \inf_{u \in \mathcal{U}} \left\{ \int_0^\tau g(y(s), u(s)) e^{-\lambda s} ds + e^{-\lambda \tau}\ v(y_x(\tau)) \right\}, \quad \forall x \in \mathbb{R}^d, \tau > 0$$

Hamilton–Jacobi–Bellman

$$\lambda v(x) + \max_{u \in \mathcal{U}} \{-g(x,u) - \nabla v(x) \cdot f(x,u)\} = 0$$

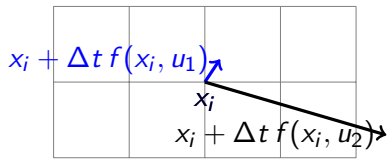
Semi-Lagrangian discretization and Value Iteration

Dynamic Programming Principle

$$v(x) = \min_{u \in \mathcal{U}} \left\{ \int_t^\tau e^{-\lambda s} g(y(s), u(s))\, ds + v\left(y(\tau)\right) e^{-\lambda \tau)} \right\}$$

Semi-Lagrangian scheme

$$V_i^{k+1} = \min_{u \in U} \left\{ \Delta t\, g(x_i, u) + e^{-\lambda \Delta t} \left(V^k(x_i + \Delta t\, f(x_i, u)) \right) \right\}, \; k = 1, 2, \ldots,$$



Discretization: constant Δt for time and N_u controls

Cons of the approach

- $V^n(x_i + \Delta t\, f(x_i, u, t_n))$ needs an interpolation operator
- Requires a numerical domain Ω chosen a priori and selection of BC
- Curse of dimensionality on structured meshes

Semi-Lagrangian discretization and Value Iteration

$$V(x_j) = \min_{u \in U} \left\{ \Delta t\, g(x_j, u) + (1 - \Delta t \lambda) I_1[V](x_j + \Delta t\, f(x_j, u)) \right\}$$

the scheme is a fixed point method

$$V^{k+1} = W(V^k), \quad k = 0, 1, \ldots$$

with

$$[W(V)]_j := \min_{u \in U} \left\{ \Delta t\, g(x_j, u) + (1 - \Delta t \lambda) I_1[V](x_j + \Delta t\, f(x_j, u)) \right\}$$

$W(V)$ is a **contraction**. Convergence is guaranteed for any initial condition.

Feedback reconstruction

Let $u_n^*(x)$ the control at each time interval $[t_n, t_{n+1})$ and $x = y(t_n)$

$$u_n^*(x) = \arg\min_{u \in U} \{ \Delta t\, g(x, u) + (1 - \lambda \Delta t) I_1[V](x + \Delta t f(x, u)) \}$$

How can we compute the value function?

The bottleneck of the DP approach is the computation of the value function, since this requires to **solve a non linear PDE in high-dimension.**
This is a challenging problem due to the **huge number of nodes** involved and to the **singularities** of the solution

- Accelerated iterative schemes
- Domain Decomposition
- Max plus algebra
- Neural Networks
- Model Order Reduction
- Sparse Grids
- Spectral Methods
- Tensor Decomposition
- Tree Structure Algorithm (see L. Saluzzi's talk 17/06/21)

Main Objectives

Literature for this talk

- O. Junge, A. Schreiber. *Dynamic programming using radial basis functions*, 2015
- G. Ferretti, R. Ferretti, O. Junge, A. Scheriber. *An adaptive multilevel radial vasis funcition scheme for HJB equation*, 2017
- C.M. Chilan, B.A. Conway, *Optimal nonlinear control using Hamilton-Jacobi-Bellman viscosity solutions on unstructured grids*, 2020
- A., H. Oliveira, G. Santin, in preparation

What we propose

- An algorithm for Dynamic Programming in **high dimensions** using RBF approximation on unstructured meshes,
- A method to **automatize** the selection of the *shape parameter* used in RBF approximation
- Error estimates
- Feedback for a *class* of initial conditions

Outline

RBF and Shepard's approximation

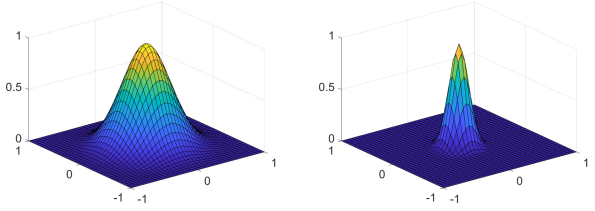
RBF Interpolation

Given a set of nodes $X = \{x_1, x_2, \cdots x_n\} \subset \Omega$ and a bounded function $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$

$$I^\sigma[f](x) = \sum_{i=1}^{n} c_i \varphi(\sigma \|x - x_i\|), \qquad I^\sigma[f](x_j) = f(x_j)$$

where $\varphi^\sigma : [0, \infty) \to \mathbb{R}, \sigma > 0$ is a **shape parameter** that affects the RBF

Wendland's RBF $\varphi^\sigma(r) = \max\{0, (1 - \sigma r)^6 (35\sigma^2 r^2 + 18\sigma r + 3)$



Left: $\sigma = 0.8$ (flat)
Right: $\sigma = 2$ (spiky)

RBF and Shepard approximation

Shepard's approximation

$$S^\sigma[f](x) = \sum_{i=1}^{n} f(x_i)\psi_i^\sigma(x), \qquad \psi_i(x) = \frac{\varphi^\sigma(\|x - x_i\|)}{\sum_{j=1}^{n} \varphi^\sigma(\|x - x_j\|)}$$

Approximation versus Interpolation

- Interpolation: Solves linear system
- Shepard approximation: Computes matrix vector multiplication

Properties:

- $\psi_i(x) > 0$ is compactly supported in $B(x_i, 1/\sigma) \subset \Omega$
- $\sum_{i=1}^{n} \psi_i^\sigma(x) = 1$ for all $x \in \Omega_{X,\sigma}$ with $\Omega_{X,\sigma} := \bigcup_{x \in X} B(x, 1/\sigma) \subset \mathbb{R}^d$
- the compact support of the weights leads to a computational advantage and a localization of the method. The distance matrix $D \in \mathbb{R}^{n \times n}$ with $D_{ij} := \|x_i - x_j\|_2$ is sparse such that $\|x_i - x_j\| \leq 1/\sigma$ needs to be computed

RBF and Shepard approximation

Fill Distance

$$h = h_{\Omega,X} := \max_{x\in\Omega} \min_{y\in X} \|x - y\|$$

Separation Distance

$$q = q_X = \min_{x_i \neq x_j \in X} \|x_i - x_j\|$$

Lemma (Junge & Schreiber, 2015)

Let $v : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ be Lipschitz continuous with Lipschitz constant L_v. Let X_k be a sequence of sets of nodes with fill distances h_k and shape parameters $\sigma_k = \frac{\theta}{h_k}$ and $\theta > 0$. Let $\rho > 0$ be such that $B(0,\rho) \supset supp(\varphi)$.Then

$$||v - S_k v||_\infty \leq L_v \frac{\rho}{\theta} h_k$$

Outline

Value Iteration with Shepard Approximation

The Shepard Approximation can be described as an operator

$$S^\sigma : (L^\infty, ||.||_\infty) \to (\mathcal{W}, ||.||_\infty)$$

where $\mathcal{W} = span\{\psi_1, \psi_2, \cdots \psi_n\}$ and $S^\sigma f(x) = \sum_{i=1}^n f(x_i)\psi_i^\sigma(x)$

$$[W_\sigma(V)]_j = \min_{u \in U} \left\{ \Delta t\, g(x_j, u) + (1 - \Delta t \lambda) S^\sigma[V](x_j + \Delta t\, f(x_j, u)) \right\}$$

- **Convergence:** W is a contraction (the operator S_σ has norm 1)
- **Error Estimate:** $\|v - V\|_\infty \leq \frac{L_v}{\theta} \frac{h}{\Delta t} \quad (\sigma = \theta/h)$
- **(Discrete) Feedback reconstruction**

$$u_n^*(x) = \arg\min_{u \in U} \{ g(x, u) + (1 - \lambda \Delta t) S^\sigma[V](x + \Delta t f(x, u)) \}$$

with $x = y(t_n)$

Value Iteration with Shepard Approximation

Comments

- The requirement that h decays to zero is too restrictive for high dimensional problems, since filling the entire Ω may be out of reach
- Shepard's method perform approximations in high dimensions and unstructured grids

Novelties:

- Generation unstructured meshes
- Selection of the shape parameter
- (first) Error estimates
- Control of PDEs

Scattered Mesh

"Standard" ways to generate a mesh

- **equi-distributed grid:** nicely covers the entire space and usually provides accurate results for interpolation problems BUT not feasible for high dimensional problems e.g. $d > 10^3$
- **random set of points:** computationally efficient to generate and to use, BUT the distribution of points can be irregular and the fill distance may decrease only very slowly when increasing the number of points

Remarks

- There is a tradeoff between keeping the grid at a reasonable size and the need to cover the relevant part of the computational domain
- The fill distance for any sequence of points $\{X_n\}_{n\in\mathbb{N}}$ can at most decrease as $h \leq c_\Omega n^{-1/d}$ in $\mathbb{R}^d$

Scattered Mesh

$$[W_\sigma(V)]_j = \min_{u \in U} \left\{ \Delta t \, g(x_j, u) + (1 - \Delta t \lambda) S^\sigma[V](x_j + \Delta t \, f(x_j, u)) \right\}$$

Key fact

The evolution of the system provides itself an indication of the regions of interest within the domain. We propose a discretization method driven by the dynamics of the control problem

Dynamics driven grid

Fix a time step $\overline{\Delta t} > 0$, a maximum number $\overline{K} \in \mathbb{N}$ of discrete times and, for $\overline{L}, \overline{M} > 0$, some initial conditions of interest and a discretization of the control space

$$\overline{X} := \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{\overline{L}}\} \subset \Omega, \quad \overline{U} := \{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_{\overline{M}}\} \subset U$$

Scattered Mesh

For a given pair of initial condition $\bar{x}_i \in \overline{X}$ and control $\bar{u}_j \in \overline{U}$ we obtain trajectories

$$x_{i,j}^{k+1} = x_{i,j}^{k} + \overline{\Delta t}\, f(x_{i,j}^{k}, \bar{u}_j), \quad k = 1, \dots, \bar{K} - 1,$$
$$x_{i,j}^{1} = \bar{x}_i$$

Given $(\bar{x}_i, \bar{u}_j)$ we obtain the set $X(\bar{x}_i, \bar{u}_j) := \{x_{i,j}^1, \dots, x_{i,j}^{\bar{K}}\}$ containing the discrete trajectory, and our mesh is defined as

$$X := X(\overline{X}, \overline{U}, \overline{\Delta} t, \overline{K}) := \bigcup_{i=1}^{\bar{L}} \bigcup_{j=1}^{\bar{M}} X(\bar{x}_i, \bar{u}_j)$$

Comments

- We do not aim at filling the space Ω, but provides points along trajectories of interest
- The values of $\overline{X}, \overline{U}, \overline{\Delta} t, \overline{K}$ should be chosen so that X contains points that are suitably close to the points of interest for the solution of the control problem

Scattered Mesh

Property of this mesh I

Let $X := X(\overline{X}, \overline{U}, \overline{\Delta t}, \overline{K})$ be the dynamics-dependent mesh, and assume that f is uniformly bounded i.e., there exists $M_f > 0$ such that

$$\sup_{x \in \Omega, u \in U} \|f(x,u)\| \leq M_f$$

Then for each $x \in X$, $\Delta t > 0$ and $u \in U$ it holds

$$\operatorname{dist}(x + \Delta t\, f(x,u), X) \leq M_f \Delta t$$

Scattered Mesh

Property of this mesh II

Furthermore if f is uniformly Lipschitz continuous in both variables, there exist $L_x, L_u > 0$ s. t.

$$\|f(x,u) - f(x',u)\| \leq L_x \|x - x'\| \quad \forall x, x' \in \Omega, \quad u \in U$$
$$\|f(x,u) - f(x,u')\| \leq L_u \|u - u'\| \quad \forall x \in \Omega, \quad u, u' \in U$$

Then, if $x := x^k(x_0, u, \Delta t) \in \Omega$ is a point on a discrete trajectory with initial point $x_0 \in \Omega$, control $u \in U$, timestep $\Delta t > 0$, and time instant $k \in \mathbb{N}$, $k \leq \bar{K}$, it holds

$$\operatorname{dist}(x, X) \leq \left(|\Delta t - \overline{\Delta t}| \bar{K} M_f + \min_{\bar{x} \in \bar{X}} \|\bar{x} - x_0\| + \bar{K} \overline{\Delta t} L_u \min_{\bar{u} \in \bar{U}} \|\bar{u} - u\| \right) e^{\bar{K} \overline{\Delta t} L_x}$$

Selection of the shape parameter in RBF

- As pointed by Fasshauer (2007) and also by Junge and Schreiber (2015), the ideal choice of shape parameter σ is crucial for accurate approximations
- There is no efficient method defined in the literature for choosing the shape parameter. In general, **a trial and error procedure is necessary**
- Cross validation and maximum likelihood estimation, but they are designed to optimize the value of σ in a fixed approximation setting

Warning

We need to construct an approximant at each iteration k within the value iteration

Selection of the shape parameter in RBF

- $\sigma := \theta / h_{\Omega,X}$ for a given $\theta > 0$
- Since $h_{\Omega,X}$ is difficult to compute or even to estimate in high dimensional problems, we use $\sigma = \theta / q_X$ and we optimize the value of $\theta > 0$

Optimization over θ

Choosing an admissible set of parameters $\mathcal{P} := [\theta_{\min}, \theta_{\max}] \subset \mathbb{R}^+$

$$\bar{\theta} := \operatorname{argmin}_{\theta \in \mathcal{P}} R(\theta / q_X) = \operatorname{argmin}_{\theta \in \mathcal{P}} \| V_{\theta/q_X} - W_{\theta/q_X}(V_{\theta/q_X}) \|_\infty$$

Remark: Optimization problem might be solved by

- comparison over $\{\theta_1, \ldots, \theta_{N_p}\} \subset \mathcal{P}$ computing all the value functions for $\{\theta_i\}_{i=1}^{N_p}$
- gradient method with continuous space $\mathcal{P}$ and

$$R_\theta = \frac{R(\theta + \varepsilon) - R(\theta)}{\varepsilon}, \quad \varepsilon > 0$$

The method relies on a-posteriori criteria (the residual)

Selection of the shape parameter

Compatibility between the mesh and the shape parameter

- The grid X is fixed independently of σ. We need an interval$\mathcal{P} := [\theta_{\min}, \theta_{\max}]$
- $S^\sigma[v](x + \Delta t\, f(x,u))$ with $x \in X$ and $u \in U_M$

$$x + \Delta t f(x,u) \in \Omega_{X,\sigma} = \cup_{x \in X} B(x, 1/\sigma)$$

for all $x \in X$ and $u \in U_M$

$$\text{dist}(x + \Delta t f(x,u), X) \leq M_f \Delta t \leq 1/\sigma = q_X/\theta \Longrightarrow \theta_{\max} \leq \frac{q_X}{M_f \Delta t}$$

- The value $\theta_{\min} > 0$ can instead be chosen freely, since a smaller parameter corresponds to a wider RBF, and thus to a larger support

Error estimates

Mesh estimate $\sigma := \theta / h_{X,\widetilde{\Omega}}$

$$h_{X,\widetilde{\Omega}} := \operatorname{dist}(\widetilde{\Omega}, X) = \max_{x \in \widetilde{\Omega}} \operatorname{dist}(x, X)$$
$$\leq C(\bar{K}, \overline{\Delta t}, L_x, M_f, L_u) \left(\max_{\widetilde{\Delta t} \in \widetilde{T}} |\widetilde{\Delta t} - \overline{\Delta t}| + \max_{\widetilde{x} \in \widetilde{X}} \min_{\bar{x} \in \bar{X}} \|\bar{x} - x_0\| + \max_{\widetilde{u} \in \widetilde{U}} \min_{\bar{u} \in \bar{U}} \|\bar{u} - u\| \right)$$

Value Function estimate

$$\|v - V\|_{\infty,\widetilde{\Omega}} := \max_{x \in \widetilde{\Omega}} |v(x) - V(x)| \leq \frac{L_v}{\theta \widetilde{\Delta t}} h_{X,\widetilde{\Omega}}$$
$$\leq \frac{L_v C}{\theta \widetilde{\Delta t}} \left(\max_{\widetilde{\Delta t} \in \widetilde{T}} |\widetilde{\Delta t} - \overline{\Delta t}| + \max_{\widetilde{x} \in \widetilde{X}} \min_{\bar{x} \in \bar{X}} \|\bar{x} - x_0\| + \max_{\widetilde{u} \in \widetilde{U}} \min_{\bar{u} \in \bar{U}} \|\bar{u} - u\| \right)$$

$h_{X,\widetilde{\Omega}}$ and q_X can be computed

Algorithm 1: Value Iteration with shape parameter selection

INPUT: $\Omega, \Delta t, U, \mathcal{P}$ parameter range, tolerance, RBF and system dynamics f, flag
initialization;
Generate Mesh
if flag $==$ Comparison **then**
  for $\theta \in P$ **do**
    Compute V_θ;
    $R(\theta) = ||V_\theta - W(V_\theta)||_\infty$
  end for
  $\bar{\theta} = \arg\min_{\theta \in P} R(\theta)$;
else
  $R_\theta = 1, \theta = \theta_0$, tol, ε
  while $\|R_\theta\| > tol$ **do**
    Compute V_θ and $V_{\theta+\varepsilon}$
    Evaluate $R(V_\theta)$ and $R(V_{\theta+\varepsilon})$
    $R_\theta = \dfrac{R(V_{\theta+\varepsilon}) - R(V_\theta)}{\varepsilon}$
    $\theta = \theta - R_\theta$
  end while
  $\bar{\theta} = \theta, V_{\bar{\theta}} = V_\theta$
end if

Outline

Eikonal equation in 2D

Problem data

We consider a two dimensional problem in $[-1,1]^2$ with dynamics

$$f(x,u) = \begin{pmatrix} \cos(u) \\ \sin(u) \end{pmatrix}$$

control space $U = [0, 2\pi]$, target $\mathcal{T} = (0,0)$ and a cost functional $\mathcal{J}_x(y,u) = \int_0^{t(x,u)} e^{-\lambda s} ds$ where

$$t(x,u) := \begin{cases} \inf_s\{s \in \mathbb{R}_+ : y_x(s,u) \in \mathcal{T}\}, \text{ if } y_x(s,u) \in \mathcal{T} \text{ for some } s \\ +\infty, \text{ otherwise.} \end{cases}$$

.

Eikonal equation in 2D

Details

- The distance function is the exact solution
- The RBF used in all cases is the C^2 Compactly Supported Wendland's function $\varphi^\sigma(r) = \max\{0, (1-\sigma r)^6(35\sigma r^2 + 18\sigma r + 3)\}$
- We compute the error with $\|\cdot\|_\infty$

Errors

- *Relative error:* $\frac{||V_{\bar{\theta}} - V^*||_\infty}{||V^*||_\infty}$ where V^* is the exact solution and $\bar{\theta} = \arg\min_{\theta \in P} R(\theta)$
- *Minimum error:* For each fill distance the minimum error is $\frac{||V_{\theta^*} - V^*||_\infty}{||V^*||_\infty}$ where $\theta^* = \arg\min_{\theta \in P} \frac{||V_\theta - V^*||_\infty}{||V^*||_\infty}$

Eikonal equation in 2D - Residuals - Equidistant Grid

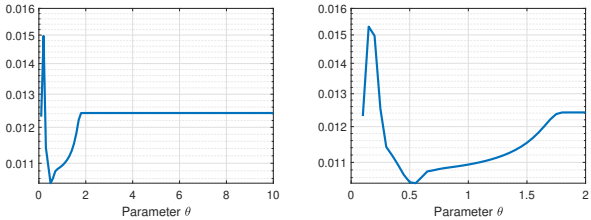


Figure: Left: $R(V_\theta)$ in $\mathcal{P}_1 = [0.1, 10]$ with step size of 0.1. Right: $R(V_\theta)$ in $\mathcal{P}_2 = [0.1, 2]$ with step size of 0.05.

Eikonal equation in 2D - Residuals - Equidistant Grid

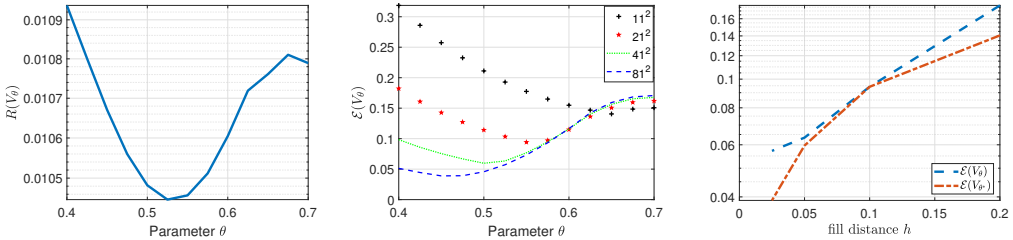


Figure: Left: Residual in case of 81^2 points. Middle: V_θ error. Right: relative and minimum error

h	Points	CPU time	$\bar{\theta}$	θ^*	$\mathcal{E}(V_{\bar{\theta}})$	$\mathcal{E}(V_{\theta^*})$	$\frac{h}{\bar{\theta}}$	$\frac{h}{\theta^*}$
0.2	11^2	5.1	0.55	0.65	0.1775	0.1405	0.3636	0.3077
0.1	21^2	14.5	0.55	0.55	0.0942	0.0942	0.1818	0.1818
0.05	41^2	242	0.525	0.5	0.0634	0.0596	0.0952	0.1002
0.025	81^2	6.44e+3	0.525	0.45	0.0572	0.0389	0.0476	0.0556

Eikonal equation in 2D - Value Functions - Equidistant Grid

Regular Case: Domain discretized in 41^2 equally spaced points. Parameters: $\Delta t = 0.5\Delta x$, $\lambda = 1$, $\sigma = 0.475/h$ and $U = [0, 2\pi]$ discretized in 16 points.

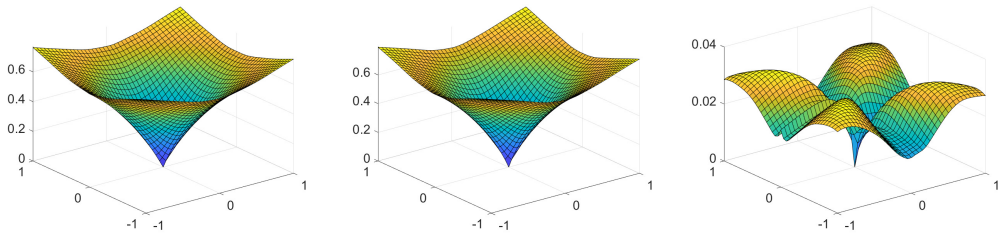


Figure: Left: Value Functions obtained by VI and Linear Interpolation. Right: Solution obtained by VI and Shepard approximation

Eikonal equation in 2D - Residuals - Equidistant Grid

Points	CPU time	$\bar{\theta}$	$\mathcal{E}(V_{\bar{\theta}})$
121	0.31	0.55	0.1774
441	15	0.54	0.0984
1681	561	0.53	0.0649
6561	2.17e+4	0.518	0.0536

Table: Results using gradient method.

Eikonal equation in 2D - Meshes - Scattered Case

Meshes generated by a set of random points clustered using k-means algorithm

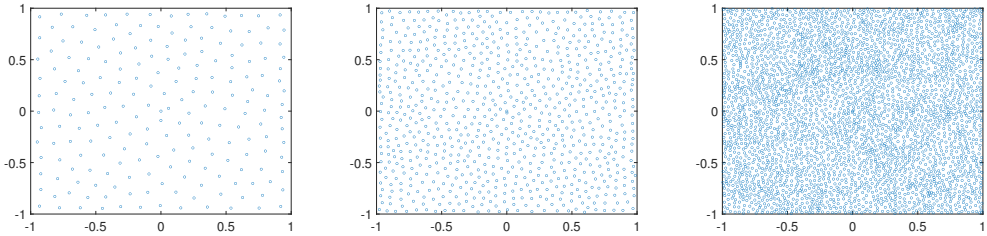


Figure: Left: 200 points and fill distance 0.1618. Center: 800 points and fill distance 0.0846. Right: 3200 points and fill distance 0.0461.

Eikonal equation in 2D - Residuals - Scattered case

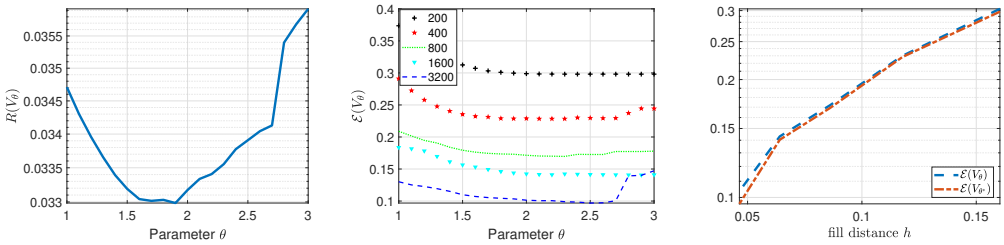


Figure: Left: Average residual. Middle: Average V_θ error. Right: Average Relative and minimum error

h	Points	CPU time	$\bar\theta$	θ^*	$\mathcal{E}(V_{\bar\theta})$	$\mathcal{E}(V_{\theta^*})$	$\frac{h}{\bar\theta}$	$\frac{h}{\theta^*}$
0.1603	200	9.8	1.91	2.16	0.3031	0.2981	0.0839	0.0742
0.1177	400	14.6	1.86	2.06	0.23	0.2284	0.0633	0.0572
0.0861	800	31.8	1.92	2.21	0.172	0.1697	0.0448	0.0389
0.0641	1600	115	2.04	2.42	0.1432	0.1407	0.0314	0.0265
0.0464	3200	504	1.76	2.06	0.1037	0.0969	0.0264	0.0225

Eikonal equation in 2D - Value Functions - Scattered Case

Scattered Case: Domain populated by 3200 randomly selected points. Parameters: $\Delta t = h$, $\lambda = 1$, $\sigma = 1.88/h$ and $U = [0, 2\pi]$ discretized in 16 points.

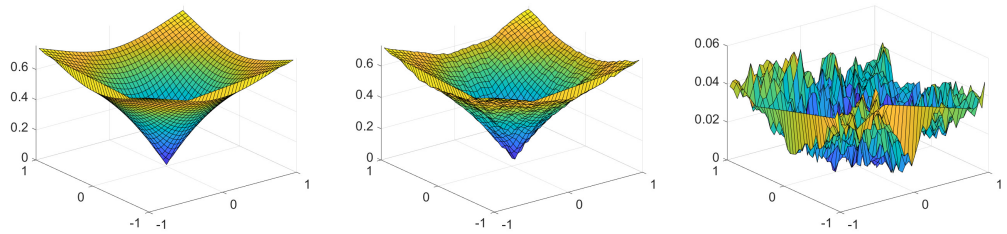


Figure: Value Functions generated in a Random Unstructured Grid formed by 3200 points. Left: Exact solution. Center: Solution obtained by VI and Shepard approximation. Right: Absolute error of exact solution and value function obtained by Shepard approximation Value Iteration.

Eikonal equation in 2D - Meshes - Dynamic Grid

- Meshes generated using the dynamics of the problem. 16 controls, $\Delta t = 0.05$ and the Δt used to select points were respectvely 0.1, 0.05 and 0.025.
- Left: 4 initial conditions. Center: 8 initial conditions. Right: 16 initial conditions. All selected using **k-means** algorithm.

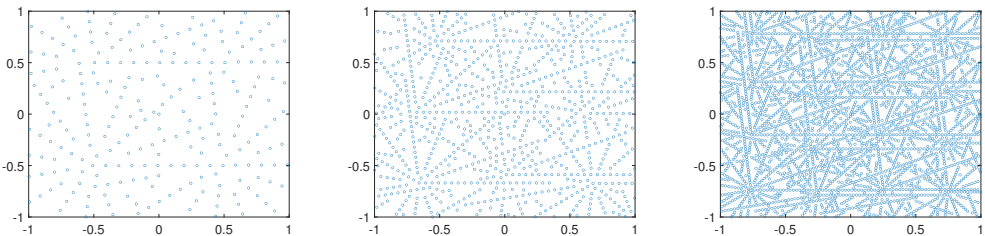


Figure: Left: 246 points and fill distance 0.1436. Upper Right: 909 points and fill distance 0.0882. Center: 3457 points and fill distance 0.0439.

Eikonal equation in 2D - Residuals - Dynamic Grid

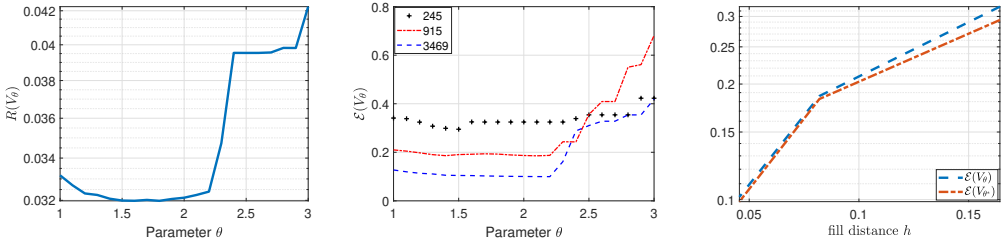
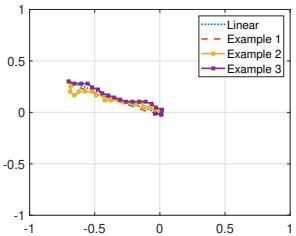


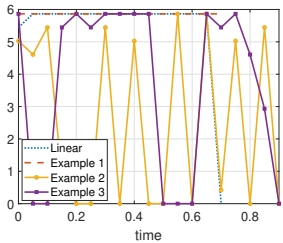
Figure: Left: Average residual to case with 3483 points. Middle: Average V_θ error, Right: Average Relative Error and Average Minimum Error against fill distance

h	Points	CPU time	θ	θ^*	$\mathcal{E}(V_{\bar{\theta}})$	$\mathcal{E}(V_{\theta^*})$	$\frac{h}{\bar{\theta}}$	$\frac{h}{\theta^*}$
0.1642	245	8.5	1.58	1.82	0.3182	0.2949	0.1040	0.0902
0.0820	915	55.6	1.66	1.76	0.1861	0.1855	0.0494	0.0466
0.0455	3469	654	1.7	1.82	0.1016	0.0997	0.0268	0.0250

Eikonal equation in 2D - Feedback Reconstruction

$x = (-0.7, 0.3)$





x	Linear	*Example 1*	*Example 2*	*Example 3*
$(0.5, 0.75)$	0.6287	0.6287	0.7724	0.6664
$(-0.7, 0.3)$	0.5646	0.5646	0.6481	0.6481

Table: Evaluation of the cost functional for different methods and initial conditions x.

Test 2: Advection Equation

Dynamics

$$\begin{cases} \tilde{y}_t(\xi,t) + v \cdot \nabla_\xi \tilde{y}(\xi,t) = u(t)\tilde{y}(\xi,t) & (\xi,t) \in \Omega \times [0,T] \\ \tilde{y}(\xi,t) = 0 & \xi \in \partial\Omega \times [0,T] \\ \tilde{y}(\xi,0) = \tilde{y}_0(\xi) & \xi \in \Omega \end{cases}$$

$\Omega = [0,5]^2,\ v = 1,\ T = 2.5,\ U = [-2,0],\ d = 10121$

Cost functional (after semi-discretization)

$$\mathcal{J}_x(y,u) \equiv \int_0^\infty (\|y(s)\|_2^2 + 10^{-5}|u(s)|^2)e^{-\lambda s}ds$$

Parameters

Grid: 11 controls $\bar{\Delta}t = 0.1$ $k = \{0.5, 1\}$ $\mathcal{C} := \left\{k \sin(\pi\xi_1)\sin(\pi\xi_2)\chi_{[0,1]^2}\right\}$
VI 21 controls, $\Delta t = 0.05, \mathcal{P} = [0.4, 0.7]$ with step 0.05, CPU time = 583s

Test 2: Advection Equation (Parameters Traj: 81 controls, $\Delta t = 0.05$) NO NEED TO UPDATE THE VF

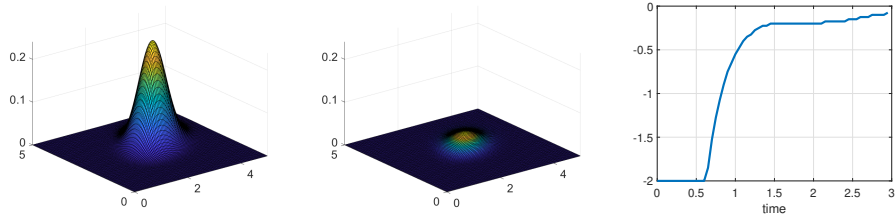
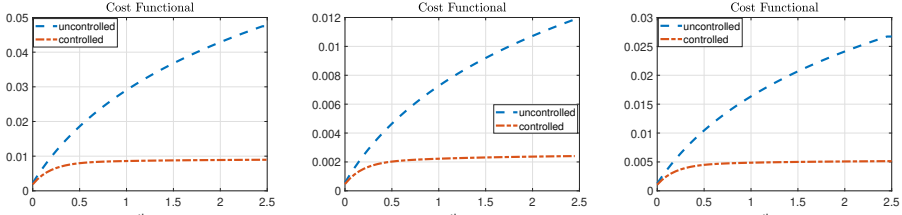


Figure: Initial condition $y(x,0) = 0.75 sin(\pi x_1) sin(\pi x_2) \chi_{[0,1]^2}$. Left-Right: uncontrolled solution, controlled solution, optimal control



Test 3: Nonlinear Heat Equation

Dynamics

$$\begin{cases} \tilde{y}_t(x,t) = \alpha\Delta\tilde{y}(x,t) + \beta(\tilde{y}^2(x,t) - \tilde{y}^3(x,t)) + u(t)\tilde{y}_0(x) & (x,t) \in \Omega \times [0,\infty) \\ \partial_n\tilde{y}(x,t) = 0 & x \in \partial\Omega \times [0,\infty) \\ \tilde{y}_0(x) = \tilde{y}(x,0) & x \in \Omega \end{cases}$$

with $\Omega = [0,1] \times [0,1], \alpha = \frac{1}{100}, \beta = 6,\ d = 961$

Cost functional (after semi-discretization)

$$\mathcal{J}_x(y,u) \equiv \int_0^\infty (\|y(s)\|_2^2 + 10^{-3}|u(s)|^2)e^{-\lambda s}ds$$

Parameters

Grid: 11 controls $\overline{\Delta t} = 0.1$ $k = \{0.5, 1\}$ $\mathcal{C} := \left\{k\sin(\pi\xi_1)\sin(\pi\xi_2)\chi_{[0,1]^2}\right\}$
VI: 21 controls, $\Delta t = 0.05, \mathcal{P} = [1.8, 2.2]$ with step 0.05 CPU: 1.64e+04

Test 3: Nonlinear Heat Equation

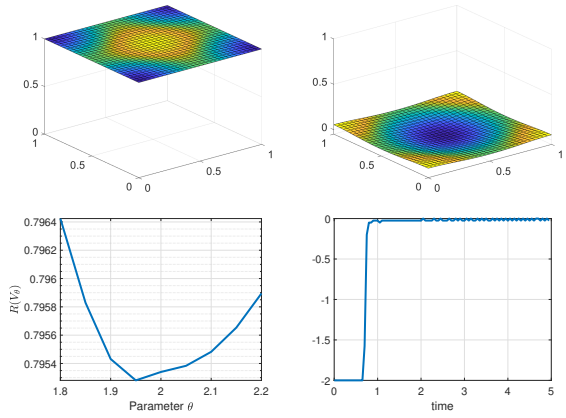


Figure: Top. Initial condition $y(x,0) = 0.75 sin(\pi x_1) sin(\pi x_2)$. Left: uncontrolled solution at time $t = 5$. Right: controlled solution at time $t = 5$. Bottom. Left: residual, Right: optimal control

Test 3: Nonlinear Heat Equation

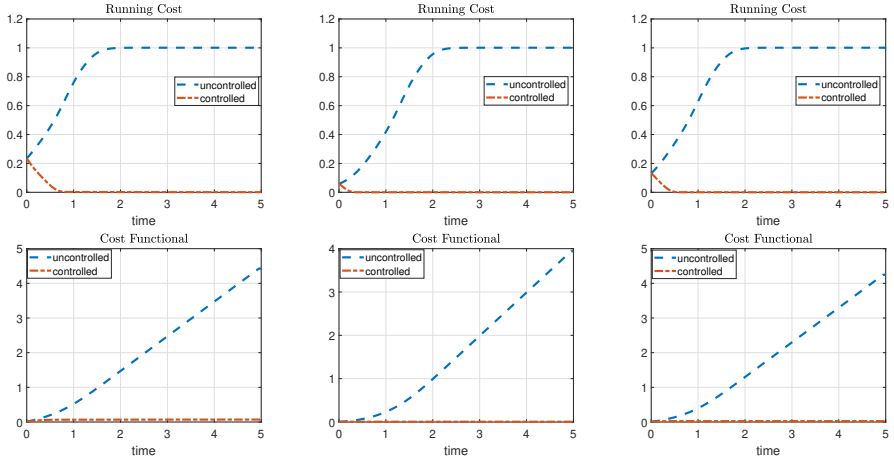


Figure: Running Cost and cost functional with initial condition $y(x,0) = ksin(\pi x_1)sin(\pi x_2)$, $k = 1, 0.5, 0.75$ (left to right)

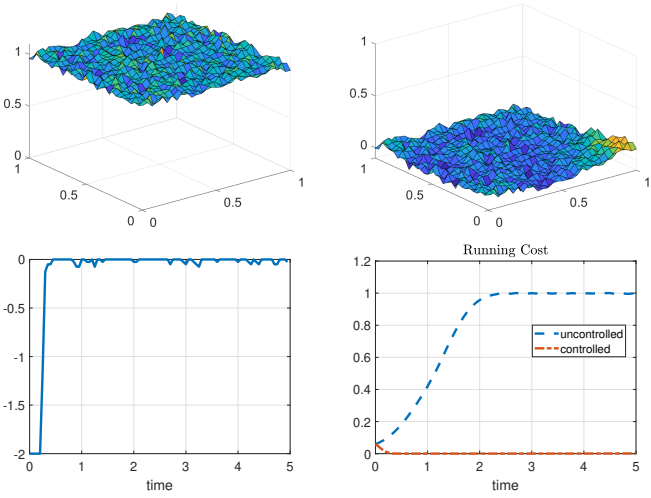


Figure: Initial condition $y(x,0) = 0.75sin(\pi x_1)sin(\pi x_2) + \mathcal{N}(0, 0.025)$. Left: uncontrolled solution. Middle: controlled solution. Right: optimal control. Running Cost with initial condition $y(x,0) = 0.75sin(\pi x_1)sin(\pi x_2) + \mathcal{N}(0, 0.025)$

Conclusions and Future Works

Conclusions

- RBF and Shepard's approximation are useful and computationally efficient to work in high dimensional control problems
- A new algorithm to simultaneously solve the Value Iteration algorithm and select the shape parameter
- A method that uses unstructured meshes driven by the dynamics

Outlook

- Adapt this framework to Policy Iteration algorithm
- Extend this method to semi-Lagrangian schemes
- Use of model reduction to speed up the computation and to ease the interpolation

References

- A. Alla, M. Falcone, and D. Kalise. *An efficient policy iteration algorithm for dynamic programming equations*, SIAM J. Sci. Comput., 2015.
- A. Alla. M. Falcone. L. Saluzzi. *An efficient DP algorithm on a tree-structure for finite horizon optimal control problems* SIAM J. Sci. Comput., 2019.
- A. Alla, H. Oliveira, G. Santin. *HJB-RBF based approach for the control of PDEs,* in preparation.
- M. Bardi and I. Capuzzo-Dolcetta. Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations, 1997.
- C.M. Chilan, B.A. Conway, *Optimal nonlinear control using Hamilton-Jacobi-Bellman viscosity solutions on unstructured grids,* Journal of Guidance, Control, and Dynamics, 2020.
- M. Falcone and R. Ferretti. Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi equations, SIAM, 2013.
- G. F. Fasshauer. Meshfree Approximation Methods with MATLAB, 2007.
- L. Grüne *An adaptive grid scheme for the discrete Hamilton-Jacobi-Bellman equation,* Numerische Mathematik, 1997.
- O. Junge, A. Schreiber. *Dynamic programming using radial basis functions* Discrete and Continuous Dynamical Systems- Series A, 2015.

Thank you for you attention