PinT Scheme using time as a parameter

Olga Mula

TU Eindhoven

PinT 2022 – July 11-15, 2022

Preliminary thoughts: Setting to study PinT schemes?

Let $\mathbb{U}$ be a Banach space over a domain $\Omega \subset \mathbb{R}^d$,

Problem: find $u \in \mathcal{C}^1([0, T], \mathbb{U})$ solution to

$$\begin{aligned} u'(t) + \mathcal{A}(t, u(t)) = 0, \quad t \in [0, T], \\ u(0) = u_0 \in \mathbb{U} \end{aligned}$$

Task: Build a scheme such that for all $N \geq 1$,

$$\text{Eff}_N := \frac{1}{N} \frac{\text{Sequential Runtime}}{\text{Runtime in parallel with N processors}} \quad \approx 1$$

Accuracy: For a given taget accuracy η, the approximation $\tilde{u}(t)$ must be such that

$$\max_{t \in [0,T]} \|u(t) - \tilde{u}(t)\|_{\mathbb{U}} \leq \eta.$$

Preliminary thoughts: Setting to study PinT schemes?

Different paradigms:

- **Purely PDE-driven:**
  i) Time-stepping schemes (e.g. parareal, MGRIT, PFASST...)
  ii) Preconditionnners (after discretization)
  iii) Exponential integrators (Cauchy integral formula)

  For i), scalability is rather poor, expecially for transport dominated problems.

- **Mixed approaches PDE-Driven + Data-driven:**
  i) Model-Order Reduction
  ii) Deep Learning (PINNs)

  Does the learning phase enter into the count of the parallel efficieny?

Roadmap of the talk

This talk is a discussion about my thoughts/experience regarding:

- The fundamental limits of parareal, and approach i).
- Merits and limitations of alternative promising paradigms?

Roadmap:

1) A fully adaptive parareal algorithm: merits and pitfalls.

2) Time as a parameter:

- The Cauchy integral formula (parabolic problems)
- Reduced Order Models (transport-dominated problems)

Part I:

An adaptive parareal algorithm

[MM20] **An adaptive parareal algorithm**. Y. Maday, O. Mula (JCAM, 2022)

The classical parareal in time algorithm

Let $\mathcal{G}$ and $\mathcal{F}$ be the coarse and fine propagators of an evolution problem.

If $k = 0$,

$$\begin{cases} y_0^N &= \mathcal{G}(T_N, \Delta T, y_0^{N-1}), \ 1 \leq N \leq \underline{N}. \\ y_0^0 &= u(0). \end{cases}$$

If $k \geq 1$,

$$\begin{cases} y_k^N = \mathcal{G}(T_{N-1}, \Delta T, y_k^{N-1}) + \mathcal{F}(T_{N-1}, \Delta T, y_{k-1}^{N-1}) - \mathcal{G}(T_{N-1}, \Delta T, y_{k-1}^{N-1}), \\ y_k^0 = u(0). \end{cases}$$

Two major obstructions

1. Parallel efficiency:
    - **eff** $\approx 1/K$
    - Problem: repeated use of $\mathcal{F}$
2. No online stopping criteria $\longrightarrow$ Need for a posteriori estimators

Setting and notations

Let $\mathbb{U}$ be a Banach space over a domain $\Omega \subset \mathbb{R}^d$,

Problem: find $u \in \mathcal{C}^1([0, T], \mathbb{U})$ solution to

$$\begin{aligned} u'(t) + \mathcal{A}(t, u(t)) &= 0, \quad t \in [0, T], \\ u(0) &= u_0 \in \mathbb{U} \end{aligned}$$

Propagators:

- $\mathcal{E}(t, s, w) = \mathcal{E}(\text{initial time, step, initial condition in } \mathbb{U})$
  $\mathcal{E}(0, t, u_0) = u(t)$
- For any $\zeta > 0$, $[\mathcal{E}(t, s, w); \zeta]$ is an element of $\mathbb{U}$ satisfying

$$\|\mathcal{E}(t, s, w) - [\mathcal{E}(t, s, w); \zeta]\| \leq \zeta\, s\, (1 + \|w\|).$$

- $\mathcal{F} = [\mathcal{E}(), \zeta_{\mathcal{F}}]$, $\mathcal{G} = [\mathcal{E}(); \zeta_{\mathcal{G}}]$.

Discretization in time: $T_0 = 0 < T_1 < \cdots < T_{\underline{N}} = T$.

Goal: For a given taget accuracy η, build $\tilde{u}(T_N)$ such that

$$\max_{0 \leq N \leq \underline{N}} \|u(T_N) - \tilde{u}(T_N)\| \leq \eta.$$

Best implementable version of algorithm

Ideal parareal iterations: We build a sequence $(y_k^N)_k$ of approximations of $u(T_N)$ for $0 \leq N \leq \underline{N}$ following the recursive formula

$$\begin{cases} y_0^{N+1} = \mathcal{G}(T_N, \Delta T, y_0^N), & 0 \leq N \leq \underline{N}-1 \\ y_{k+1}^{N+1} = \mathcal{G}(T_N, \Delta T, y_{k+1}^N) & \\ \qquad + \mathcal{E}(T_N, \Delta T, y_k^N) - \mathcal{G}(T_N, \Delta T, y_k^N), & 0 \leq N \leq \underline{N}-1,\ k \geq 0, \\ y_0^0 = u(0). & \end{cases}$$

Feasible parareal iterations: We build a sequence $(\tilde{y}_k^N)_k$ of approximations of $u(T_N)$ for $0 \leq N \leq \underline{N}$ following the recursive formula

$$\begin{cases} \tilde{y}_0^{N+1} = \mathcal{G}(T_N, \Delta T, \tilde{y}_0^N), & 0 \leq N \leq \underline{N}-1 \\ \tilde{y}_{k+1}^{N+1} = \mathcal{G}(T_N, \Delta T, \tilde{y}_{k+1}^N) & \\ \qquad + [\mathcal{E}(T_N, \Delta T, y_k^N), \zeta_k^N] - \mathcal{G}(T_N, \Delta T, \tilde{y}_k^N), & 0 \leq N \leq \underline{N}-1,\ k \geq 0 \\ \tilde{y}_0^0 = u(0). & \end{cases}$$

Question: minimal accuracy ζ_k^N to preserve the convergence rate of ideal scheme?

Parallel efficiency

Lemma 1 (see [MM20])

Let $\zeta_{\mathcal{G}}$ be the accuracy of $\mathcal{G}$, and assume its cost is negligible.
We can find the minimal accuracy for ζ_k^N to guarantee convergence and

$$\boldsymbol{eff}(\eta, [0, T]) = \frac{\boldsymbol{cost}_{AP}(\eta, [0, T])cost}{\boldsymbol{cost}_{seq}(\eta, [0, T])} \approx\sim \frac{1}{(1 + \zeta_{\mathcal{G}})}.$$

and

$$\boldsymbol{speed\text{-}up}(\eta, [0, T]) = \underline{N}\, \boldsymbol{eff}(\eta, [0, T]) \sim \underline{N}\frac{1}{(1 + \zeta_{\mathcal{G}})}.$$

Parallel efficiency

Merits:

- Convergence to exact solution.
- $\forall \eta > 0$, better efficiency than the plain method
- Efficiency independent of the final number of iterations
- Only cost of last fine propagation counts
- Opens the door to adaptive refinements

Obstructions to get full scalability:

- We have solved the issue with the fine solver BUT...
- Cost of coarse solver is in general non negligible.
- This is particularly the case for transport dominated problems.

Connection to other works/approaches

Classical formulation of parareal: We can interpret the fine solver as

$$\mathcal{F}(T_N, \Delta T, w) = [\mathcal{E}(T_N, \Delta T, w), \zeta_{\mathcal{F}}],$$

where $\zeta_{\mathcal{F}}$ is small and kept constant across the parareal iterations.

Improvement of speed-up with info from previous iterations:

- Coupling of the parareal algorithm with spatial domain decomposition (see [MT05, Gue12, ABGM17]).
- Combination of the parareal algorithm with iterative high order methods in time like spectral deferred corrections (see [MWS+08, Min10, MSB+15])
- Solution of internal fixed points initialized with solutions at previous parareal iterations (see [Mul14]).
- In a similar spirit, applications of the parareal algorithm to solve optimal control problems (see [MT05, MST07]).

Improvement of speed-up if we decrease cost of coarse solver.

Part II: Parametric Strategies

The Cauchy integration formula for parabolic problems

Integration of high-dimensional linear parabolic PDEs

Consider the spatial domain

$$\Omega = \Omega_1 \times \cdots \times \Omega_d$$

and the elliptic operator

$$A = \sum_{i=1}^{d} \mathrm{id} \otimes \cdots \otimes \mathrm{id} \otimes A_i \otimes \mathrm{id} \otimes \cdots \otimes \mathrm{id},$$

where each

$$A_i : H_i \to H_i',$$

is elliptic with compact inverse. One example could be

$$H_i = H_0^1(\Omega_i), \qquad A_i = -\partial_{x_i}^2, \qquad A = -\Delta$$

Integration of high-dimensional linear parabolic PDEs

Let $u \in H_1 \times \cdots \times H_d$ be the solution to

$$\partial_t u(t,x) + Au(t,x) = f(t,x), \quad \forall (t,x) \in (0,T) \times \Omega$$
$$u(t,x) = 0, \quad \forall (t,x) \in (0,T) \times \partial\Omega$$

with $u(t=0) = u_0$.

The Cauchy formula gives the solution to this equation in terms of exponential of operators:

$$u(t,\cdot) = e^{-tA} u_0(\cdot) + \int_0^t e^{-(t-s)A} f(s,\cdot) ds.$$

Back-bone of exponential in time integrators (see [SST00, HO10], talks by Mayya Tokmann, Martin Schreiber).

Cauchy integral formula

Let $f : \mathbb{C} \to \mathbb{C}$ be an holomorphic function on some open set $D \subset \mathbb{C}$. The Cauchy integral formula states that

$$f(z) = \frac{1}{2i\pi} \int_\Gamma \frac{f(\zeta)}{\zeta - z} \mathrm{d}\zeta, \quad \forall z \in \mathbb{C}$$

for any rectifiable Jordan curve satisfying some conditions.

Let $\mathcal{H}$ be a Hilbert space, and $A \in \mathcal{L}(\mathcal{H})$. We can define the action of f on A through the Cauchy integral

$$f(T) := \frac{1}{2i\pi} \int_\Gamma f(z)(z - T)^{-1} \mathrm{d}z$$

for a rectifiable curve containing $\sigma(A)$.

Let us apply this to $f : z \mapsto e^{-tz}$ and $A = -\Delta$.

Cauchy integral formula

Let us apply this to our case with $A = -\Delta$:

$$
\begin{aligned}
u(t,\cdot) &= e^{-tA}u_0(\cdot) + \int_0^t e^{-(t-s)A} f(s,\cdot)ds \\
&= \int_\Gamma e^{-tz}(z+\Delta)^{-1}u_0(\cdot)\mathrm{d}z + \int_0^t \int_\Gamma e^{-(t-s)z}(z+\Delta)^{-1} f(,s\cdot)\mathrm{d}z
\end{aligned}
$$

To compute this, select quadrature rules and

$$
\begin{aligned}
\tilde{u}_Q(t,\cdot) = &\sum_{q=-Q}^{Q} \omega_q e^{-tz_q}(z_q+\Delta)^{-1}u_0(\cdot) \\
&+ \int_0^t \sum_{q=-Q}^{Q} \omega_q e^{-(t-s)z_q}(z_q+\Delta)^{-1} f(s,\cdot)\mathrm{d}s
\end{aligned}
$$

Nice consequences:

- Time is a simple parameter.
- Each (static) problem can be computed in parallel.
- ...

Consequences for us

Nice consequences:

- Time is a simple parameter.
- Each (static) problem can be computed in parallel.
- Easy control on accuracy (spectral accuracy on quadrature + a posteriori estimators on elliptic problems)
- Repeated solving of elliptic problems $\to$ Model Reduction.
- Fight against curse of dimensionality:

Suppose $u_0(x_1,\dots,x_d) = \sum_{k=1}^K u_1^k(x_1)\dots u_d^k(x_d)$, then

$$e^{-t\Delta}u_0 = \sum_{k=1}^K \prod_{i=1}^d e^{-t\partial^2_{x_i}} u_i^k(x_i)$$

and

$$e^{-t\partial^2_{x_i}} u_i^k(x_i) \approx \sum_{q=-Q}^{Q} \omega_q e^{-tz_q}(z_q + \partial^2_{x_i})^{-1} u_i^k(x_i)$$

Conclusion about the Cauchy integral formula

Pros:

- Perfectly scalable in the above setting.
- Beats curse of dimensionality in certain settings [GHK02, DDGS15]
- Possibility to use MOR [DAC+22]

Cons:

- As soon as setting becomes more involved, becomes a time-stepping method.
- Use with care for other types of problems.

Part II: Parametric Approaches

Model-Order Reduction for conservation laws

Motivation: Reduced Order Modeling of parametric PDEs

Parametric PDE: Consider

$$\mathcal{P}_x(y)(r) = 0, \quad \forall r \in \Omega \subset \mathbb{R}^s, \tag{1}$$

where

- $\mathcal{P}_x$: differential operator depending on a parameter x
- $y \in \mathcal{Y}$: solution to the PDE in a metric space $\mathcal{Y}$

Parameter-to-solution map:

- $x \in \mathrm{X}$ compact set of Hilbert space $\mathcal{X}$.
- Mapping:

$$\begin{aligned} y : \mathrm{X} &\to \mathcal{Y} \\ x &\mapsto y(x) \text{ solution to (1)} \end{aligned}$$

- Solution Set: $\mathcal{M} := \{y(x) \in \mathcal{Y} \ : \ x \in \mathrm{X}\}$.

Goal: Find a quick approximation to the parameter-to-solution map

$$\begin{aligned} A : \mathrm{X} &\to \mathcal{Y} \\ x &\mapsto A(x) \approx y(x) \end{aligned}$$

Motivation: Reduced Order Modeling of parametric PDEs

Easy setting: If the PDE is elliptic/parabolic:

- Work in a Hilbert space $\mathcal{Y}$.
- Approximate $\mathcal{M}$ with linear subspaces V_n:

$$\begin{aligned} A : \mathrm{X} &\to V_n \subseteq \mathcal{Y} \\ x &\mapsto A(x) \approx y(x) \end{aligned}$$

- Kolmogorov n-width of $\mathcal{M}$ decays fast with n. (see e.g. [CD16])

Problem: Nonlinear strategies for transport-dominated PDEs?

Our focus: Conservation Laws for which we can work with

$$\mathcal{Y} = W_2(\Omega)$$

Roadmap of the rest of the talk

1. Definition of $W_2(\Omega)$ and why this choice for MOR?
2. Sparse, adaptive interpolation of measures.
3. Some tests for a 2D Burgers equation.

Part I:

Definition of $W_2(\Omega)$ and why this choice for MOR?

The L_2-Wasserstein space $W_2(\Omega)$

Let $(\Omega, \|\cdot\|)$ be a compact metric space (e.g. $\Omega \subset \mathbb{R}^s$).

Let $\mathcal{P}_2(\Omega)$ be the set of probability measures on Ω with finite second order moments.

The L^2-Wasserstein distance is defined by

$$W_2^2(u, v) := \inf_{\pi \in \Pi(u,v)} \int_{\Omega \times \Omega} \|x - y\|^2 \, \mathrm{d}\pi(x, y), \quad \forall (u, v) \in \mathcal{P}_2(\Omega) \times \mathcal{P}_2(\Omega).$$

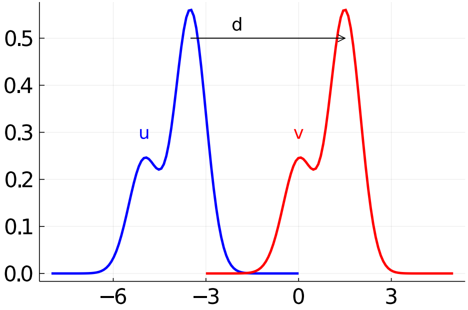
The infimum is taken over all transport plans

$$\Pi(u, v) := \{\pi \in \mathcal{P}_2(\Omega \times \Omega) \ : \ \int_y \mathrm{d}\pi(x, y) = \mathrm{d}u(x), \ \int_x \mathrm{d}\pi(x, y) = \mathrm{d}v(x)\}$$

The space $\mathcal{P}_2(\Omega)$ with the distance W_2 is a metric space.

Interest of working in W_2 for MOR?

- Solutions of conservation laws, and gradient flows can be seen as members of W_2.
- The metric W_2 encodes translations and can locate shocks.
- It is a metric space so approximations must be nonlinear.



$$W_2(u,v) = d,$$
$$L_1(u,v) = 2 \quad \text{as soon as } \mathrm{supp}(u) \cap \mathrm{supp}(v) = \emptyset$$

Part II:

Adaptive, Sparse interpolation of Measures in $W_2(\Omega)$.

Back-bone of many strategies for Hilbert spaces $\mathcal{Y}$:

- Training set: for $N \gg 1$, compute
$$\mathrm{X}_N^{\text{train}} := \{x_i\}_{i=1}^N, \quad \mathrm{Y}_N^{\text{train}} := \{y(x_i)\}_{i=1}^N$$
- For a fixed $1 \leq n \ll N$, find parameters
$$\mathrm{X}_n := \{x_1^*, \dots, x_n^*\} \subset \mathrm{X}_n^{\text{train}}.$$
with their associated solution snapshots
$$\mathrm{Y}_n := \{y(x_1^*), \dots, y(x_n^*)\} \subset \mathrm{Y}_n^{\text{train}}.$$
and build the approximation class
$$V_n := \text{span}\{y(x_1^*), \dots, y(x_n^*))\} \subset \mathcal{Y}$$
- Approximate with
$$\begin{aligned} A : \mathrm{X} &\to V_n \subseteq \mathcal{Y} \\ x &\mapsto A(x) \approx y(x) \end{aligned}$$
- Interpolation property:
$$A(x) = y(x), \quad \forall x \in \mathrm{X}_n.$$

Our strategy in $\mathcal{Y} = W_2(\Omega)$ (ideal version):

- For every $x \in \mathrm{X}$:
  - Find appropriate parameters (with a procedure yet to be defined)

    $$\mathrm{X}_n^x = \{x_1^*(x), \dots, x_n^*(x)\} \subset \mathrm{X}_N^{\mathrm{train}}$$

    and set

    $$\mathrm{Y}_n^x = \{y(x) \; : \; x \in \mathrm{X}_n^x\}_{i=1}^n \subset \mathrm{Y}_N^{\mathrm{train}}.$$
  - Given weights

    $$\Lambda_n = (\lambda_1, \dots, \lambda_n) \in \Sigma_n$$

    from the n-dimensional simplex

    $$\Sigma_n := \{\Lambda_n \in \mathbb{R}^n \; : \; \sum_{i=1}^n \lambda_i = 1, \; \lambda_i \geq 0\},$$

    we can define define the barycenter

    $$\mathrm{Bar}(\Lambda_n, \mathrm{Y}_n^x) \in \operatorname*{arg\,min}_{v \in \mathcal{P}_2(\Omega)} \sum_{y \in \mathrm{Y}_n^x}^n \lambda_i W_2^2(v, y)$$

    Our approximation class is now the set of barycenters

    $$V_n^x := \mathrm{Bar}(\Sigma_n, \mathrm{Y}_n^x) = \{\mathrm{Bar}(\Lambda_n, \mathrm{Y}_n^x) \; : \; \Lambda_n \in \Sigma_n\}$$

Our strategy in $\mathcal{Y} = W_2(\Omega)$ (continued):

- For every $x \in \mathrm{X}$:
  - Find $b \in \mathrm{Bar}(\Sigma_n, \mathrm{Y}_n^x)$ such that

$$b \in \arg\min_{b \in V_n^x} W_2^2(y(x), b)$$

    This is equivalent to finding the optimal weights

$$\Lambda_n^x \in \arg\min_{\Lambda_n \in \Sigma_n} W_2^2(y(x), \mathrm{Bar}(\Lambda_n, \mathrm{Y}_n^x))$$

  - Approximate with

$$A^x : \mathrm{X} \to V_n^x := \mathrm{Bar}(\Sigma_n, \mathrm{Y}_n^x) \subset \mathcal{Y}$$
$$z \mapsto A^x(z) = \mathrm{Bar}(\Lambda_n^z, V_n^x) \approx y(z).$$

    In particular,

$$A^x(x) = \mathrm{Bar}(\Lambda_n^x, \mathrm{Y}_n^x)$$

- Interpolation properties:
  - Local: for all $x \in \mathrm{X}$,

$$A^x(z) = y(z), \quad \forall z \in \mathrm{X}_n^x$$

  - Global:

$$A^x(x) = y(x), \quad \forall x \in \mathrm{X}_N^{\text{train}}$$

Questions/Challenges

Questions:

- For a given $x \in \mathrm{X}$, what are the best $\mathrm{X}_n^x, \Lambda_n^x$?
- Algorithm to compute them?
- Numerical cost and feasibility?

Limitations:

- Cost of computing a barycenter with state of the art Sinkhorn-type algorithms:
$$\mathcal{O}(n\mathcal{N}_{\mathrm{dof}} \log \mathcal{N}_{\mathrm{dof}})$$
$\Rightarrow$ Not as cheap as classical ROM setting but provides nevertheless fast computations thanks to GPU architectures.

Part III:

Best n-term barycentric approximation

Best n-term barycentric approximation

For a given $x \in \mathrm{X}$, the best $(\mathrm{X}_n^x, \Lambda_n^x)$ are characterized as follows:

$$\Lambda_n^x \in \underset{\Lambda_N \in \Sigma_N \cap \Delta_N^n}{\arg\min} W_2^2 \left(y(x), \mathrm{Bar}(\Lambda_N, \mathrm{Y}_N^{\mathrm{train}}) \right)$$

where

$$\begin{cases} \Delta_N^n := \{ v \in \mathbb{R}^N \ : \ \|v\|_0 \leq n \} \\ \Sigma_N := \{ v \in \mathbb{R}^N \ : \ v_i \geq 0, \ \sum_{i=1}^N v_i = 1 \} \end{cases}$$

We obtain a sparse vector

$$\Lambda_n^x = (0, \dots, 0, \lambda_{i_1^x}, 0, \dots, 0, \lambda_{i_n^x}, 0, \dots, 0)^T \in \mathbb{R}^N$$

with non-zero entries at coordinates

$$\{ i_1^x, \dots, i_n^x \}.$$

We define

$$\mathrm{X}_n^x := \{ x_{i_1^x}, \dots, x_{i_n^x} \} \subset \mathrm{X}_N^{\mathrm{train}}, \quad \text{and} \quad \mathrm{Y}_n^x := \{ y(x_{i_1^x}), \dots, y(x_{i_n^x}) \} \subset \mathrm{Y}_N^{\mathrm{train}}$$

The best n-term barycentric approximation of $y(x)$ is

$$\mathrm{Bar}(\Lambda_n^x, \mathrm{Y}_n^x)$$

Best n-term barycentric approximation

Some remarks on the optimization problem:

$$\Lambda_n^x \in \operatorname*{arg\,min}_{\Lambda_N \in \Sigma_N \cap \Delta_N^n} W_2^2 \left(y(x), \mathrm{Bar}(\Lambda_N, \mathrm{Y}_N^{\mathrm{train}}) \right)$$

- Vaguely reminiscent of Compressed Sensing problems posed on $\mathcal{Y} = \mathbb{R}^N$ of the type:

$$\min_{z \in \mathbb{R}^N} \|z\|_0 \quad \text{s.t. } Az = y^{\mathrm{obs}}$$

- Difficult optimization problem:
  - Nonconvex
  - Possibly plenty of local minima, non-unique minimizer.
  - ℓ_1-regularization not possible because in conflict with Σ_N.
- We have built a minimization algorithm that delivers satisfactory results.

Algorithm 1: Sparse projection on the simplex

Data: Target $y \in \mathcal{P}_2(\Omega)$, Training set $\mathrm{Y}_N^{\mathrm{train}}$, sparsity degree n

Result: $\Lambda_N^y \in \Sigma_N \cap \Delta_N^n$.

Initialize $\Lambda \in \Sigma_N$;

repeat

Gradient step: $\Lambda \leftarrow \Lambda - \tau \nabla_\Lambda W_2^2(y, \mathrm{Bar}(\Lambda, \mathrm{Y}_N^{\mathrm{train}}))$;

Projection into $\Delta_N^n \cap \Sigma_N$: $\Lambda \leftarrow P_{\Delta_N^n \cap \Sigma_N}(\Lambda)$

until *convergence*;

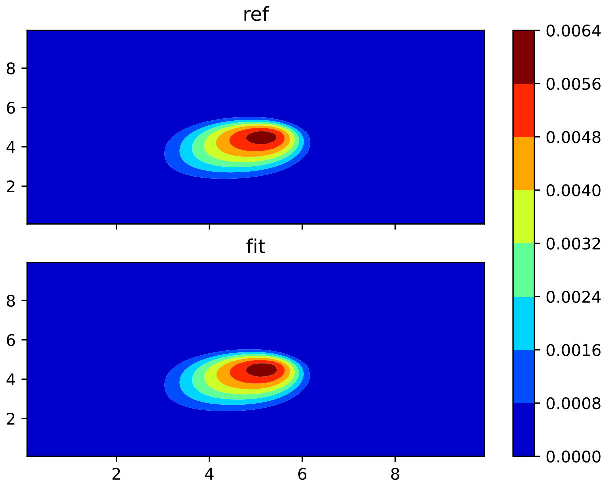
Algorithm is a generalization of CoSamp and refinements such as GSSP (see [NT09, KBCK13, BRB13]).

We can build variants in which we learn adaptively the sparsity degree n.

Rigorous convergence analysis seems out of reach due to nonconvexity.

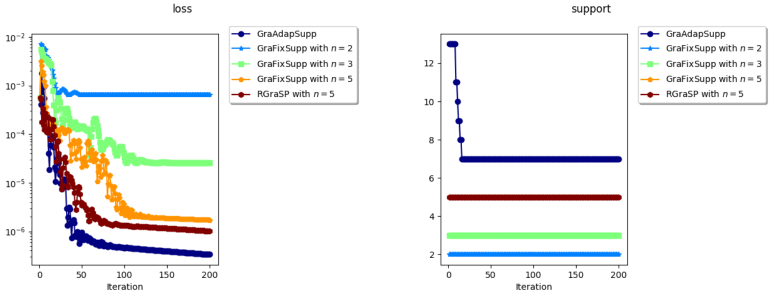
Experiment with functions from a Burgers' equation

- $N = 100$ and $\mathrm{Y}_N^{\mathrm{train}} = \{y(x_i)\}_{i=1}^N$ contains solutions of a Burgers' equation.
- Target function is a barycenter with support 2 from training set.

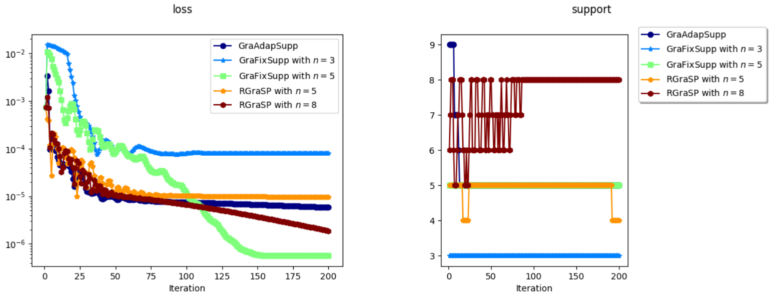


Experiment with functions from a Burgers' equation

Target function is a barycenter with support 2 from training set.



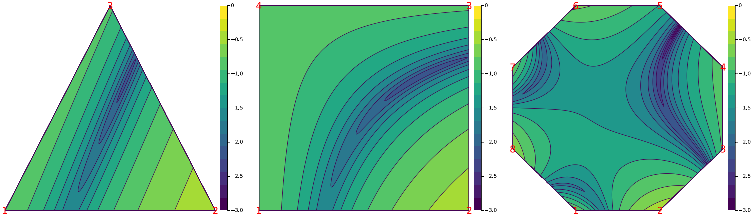
Target function is a barycenter with support 10 from training set.



Some functions of the target barycenter are probably redundant.

Error Landscape $\Lambda_n \to W_2(y, \mathrm{Bar}(\Lambda_n, Y_n))$

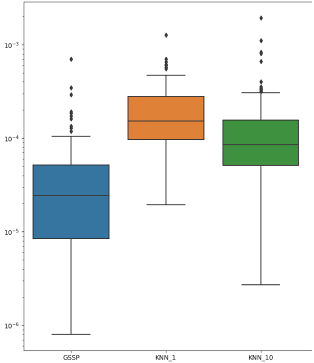
As n increases, there are more and more local minima.



Ref: "Wasserstein model reduction approach for parametrized flow problems in porous media"
(T. Blickham, B. Battisti, G. Enchery, V. Ehrlacher, D. Lombardi, O. Mula, submitted to ESAIM Proc.)

Experiment with functions from a Burgers' equation

How do we perform with respect to nearest neighbors?



Part IV:

How to use the best n-term barycentric approximation for model reduction?

Back to Model Reduction

- For a given $x \in \mathrm{X}$, computing X_n^x and Λ_n^x is based on solving

$$\Lambda_n^x \in \underset{\Lambda_N \in \Sigma_N \cap \Delta_N^n}{\arg\min} \; W_2^2 \left(y(x), \mathrm{Bar}(\Lambda_N, \mathrm{Y}_N^{\text{train}}) \right)$$

- The computation requires $y(x)$ so we can only use it for $x \in \mathrm{X}_N^{\text{train}}$.
- For a general $x \in \mathrm{X}$, we cannot assume that $y(x)$ is given. We need an extra approximation step.

Current strategy

Offline: For $x \in \mathrm{X}_N^{\text{train}}$:

- Compute $\mathrm{X}_n^x, \mathrm{Y}_n^x, \Lambda_n^x$.
- Find local Euclidean embedding:

$$M(x) \in \arg\min_{M \geq 0} \sum_{y(z) \in \mathcal{N}_n(y(x))} |(x-z)^T M (x-z) - W_2^2(y(x), y(z))|^2$$

Online: Given $x \in \mathrm{X}$:

- Find the n nearest neighbors of $y(x)$ by using the Euclidean embedding. Evaluate

$$(z-x)^T M(z)(z-x) \approx W_2^2(y(x), y(z)), \quad \forall z \in \mathrm{X}_N^{\text{train}}$$

  and pick the n smallest values to define the neighbors.
- For the neighbors $y(z) \in \mathcal{N}_n(y(x))$, we have $\Lambda_n^z, \mathrm{X}_n^z, \mathrm{Y}_n^z$.
- We want to use this information to define an interpolation strategy for the weights. [ongoing step]

Interpolation strategy

$$\sum_{y(z)\in\mathcal{N}_n(y(x))} |W_2^2(y(x),y(z)) - W_2^2(b,y(z))|^2$$
$$\approx \sum_{y(z)\in\mathcal{N}_n(y(x))} |(x-z)^T M(z)(x-z) - W_2^2(b,y(z))|^2$$

So we look for

$$\min_{\Lambda_n\in\Sigma_n} \sum_{y(z)\in\mathcal{N}_n(y(x))} |(x-z)^T M(z)(x-z) - W_2^2(\mathrm{Bar}(\Lambda_n,\mathcal{N}_n(y(x))),y(z))|^2$$

References I

- S. Aouadi, D. Q. Bui, R. Guetat, and Y. Maday, *Convergence analysis of the coupled parareal-schwarz waveform relaxation method*, 2017, in preparation.
- S. Bahmani, B. Raj, and P. T. Boufounos, *Greedy sparsity-constrained optimization*, Journal of Machine Learning Research **14** (2013), no. Mar, 807–841.
- A. Cohen and R. DeVore, *Kolmogorov widths under holomorphic mappings*, IMA Journal of Numerical Analysis **36** (2016), no. 1, 1–12.
- Huy Dinh, Harbir Antil, Yanlai Chen, Elena Cherkaev, and Akil Narayan, *Model reduction for fractional elliptic problems using kato's formula*, arXiv preprint arXiv:1904.09332 (2022).
- W. Dahmen, R. DeVore, L. Grasedyck, and E. Süli, *Tensor-sparsity of solutions to high-dimensional elliptic partial differential equations*, Foundations of Computational Mathematics (2015), 1–62.

References II

- I. P. Gavrilyuk, W. Hackbusch, and B. N. Khoromskij, *H-matrix approximation for the operator exponential with applications*, Numerische Mathematik **92** (2002), no. 1, 83–111.
- R. Guetat, *Méthode de parallélisation en temps: Application aux méthodes de décomposition de domaine*, Ph.D. thesis, Paris VI, 2012.
- M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numerica **19** (2010), 209–286.
- A. Kyrillidis, S. Becker, V. Cevher, and C. Koch, *Sparse projections onto the simplex*, International Conference on Machine Learning, PMLR, 2013, pp. 235–243.
- M. Minion, *A hybrid parareal spectral deferred corrections method*, Comm. App. Math. and Comp. Sci. **5** (2010), no. 2.
- Y. Maday and O. Mula, *An adaptive parareal algorithm*, Journal of Computational and Applied Mathematics (2020).

References III

M. L. Minion, R. Speck, M. Bolten, M. Emmett, and D. Ruprecht, *Interweaving PFASST and parallel multigrid*, SIAM Journal on Scientific Computing **37** (2015), no. 5, S244–S263.

Y. Maday, J. Salomon, and G. Turinici, *Monotonic parareal control for quantum systems*, SIAM Journal on Numerical Analysis **45** (2007), no. 6, 2468–2482.

Y. Maday and G. Turinici, *The Parareal in Time Iterative Solver: a Further Direction to Parallel Implementation*, Domain Decomposition Methods in Science and Engineering, Springer Berlin Heidelberg, 2005, pp. 441–448.

O. Mula, *Some contributions towards the parallel simulation of time dependent neutron transport and the integration of observed data in real time*, Ph.D. thesis, Université Pierre et Marie Curie-Paris VI, 2014.

References IV

- M. L. Minion, A. Williams, T. E. Simos, G. Psihoyios, and C. Tsitouras, *Parareal and spectral deferred corrections*, AIP Conference Proceedings, vol. 1048, 2008, p. 388.
- D. Needell and J. A. Tropp, *Cosamp: Iterative signal recovery from incomplete and inaccurate samples*, Applied and computational harmonic analysis **26** (2009), no. 3, 301–321.
- D. Sheen, I. Sloan, and V. Thomée, *A parallel method for time-discretization of parabolic problems based on contour integral representation and quadrature*, Mathematics of Computation **69** (2000), no. 229, 177–195.