Deep Learning aided solutions for wave equations

Hieu-Huu Nguyen

University of Basel
Department of Mathematics and Computer Science

Collaborator: Richard Tsai from UT-Austin

PinT 2022 at C.I.R.M
July 11, 2022

Current state-of-the-art Deep Learning models

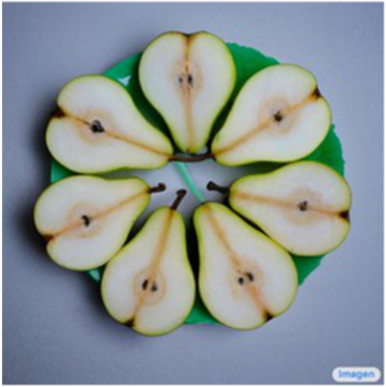
Example text: “A pear cut into seven pieces arranged in a ring”
Output images:

Current state-of-the-art Deep Learning models

Example text: “A pear cut into seven pieces arranged in a ring”
Output images:

Current state-of-the-art Deep Learning models

Example text: “A pear cut into seven pieces arranged in a ring”
Output images:



Current state-of-the-art Deep Learning models

- A text-to-image model called Imagen
- consists of a text encoding model 11B parameters, trained on 807GB of text data
- consists of several U-Net models with another 2.5B parameters, trained on 850MB of text-image data

Current state-of-the-art Deep Learning models

- A text-to-image model called Imagen
- consists of a text encoding model 11B parameters, trained on 807GB of text data
- consists of several U-Net models with another 2.5B parameters, trained on 850MB of text-image data

Reference: Saharia et al. 2022; Ramesh et al. 2022

Current state-of-the-art Deep Learning models

- A text-to-image model called Imagen
- consists of a text encoding model 11B parameters, trained on 807GB of text data
- consists of several U-Net models with another 2.5B parameters, trained on 850MB of text-image data

Reference: Saharia et al. 2022; Ramesh et al. 2022

Other related models: DALL-E 2

Deep Learning models to PDE

$$\begin{aligned}\mathcal{L}_c[u] &= f \text{ in } \Omega \\ u &= g \text{ on } \partial\Omega\end{aligned}$$

- Solve PDE by minimizing “Physics” loss, e.g. PINN, DeepOnet etc. See Song, Alkhalifah, and Waheed 2021; Jagtap and Karniadakis 2021

Deep Learning models to PDE

$$\begin{aligned} \mathcal{L}_c[u] &= f \text{ in } \Omega \\ u &= g \text{ on } \partial\Omega \end{aligned}$$

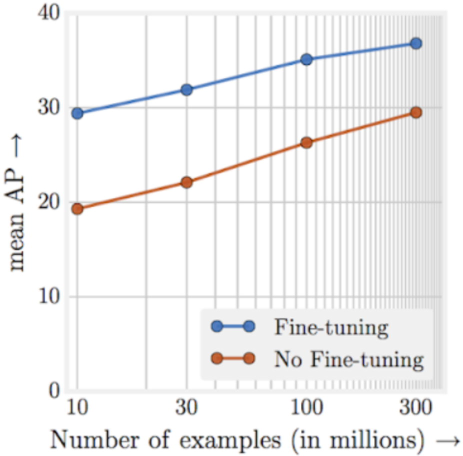
- Solve PDE by minimizing "Physics" loss, e.g. PINN, DeepOnet etc. See Song, Alkhalifah, and Waheed 2021; Jagtap and Karniadakis 2021
- Direct interpolation from pair of parameter-solution data PCA-net, FNO etc. See Z. Li et al. 2020; Kovachki et al. 2021

Deep Learning models to PDE

$$\begin{aligned} \mathcal{L}_c[u] &= f \text{ in } \Omega \\ u &= g \text{ on } \partial\Omega \end{aligned}$$

- Solve PDE by minimizing "Physics" loss, e.g. PINN, DeepOnet etc. See Song, Alkhalifah, and Waheed 2021; Jagtap and Karniadakis 2021
- Direct interpolation from pair of parameter-solution data PCA-net, FNO etc. See Z. Li et al. 2020; Kovachki et al. 2021
- Augment standard numerical scheme Um et al. 2020; Siahkoohi, Louboutin, and Herrmann 2019; Kochkov et al. 2021

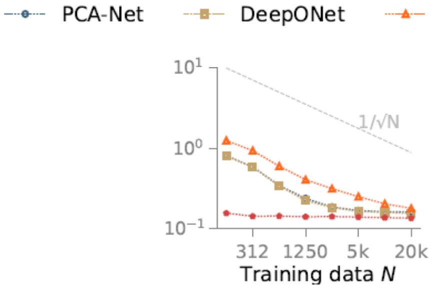
How much data?



Object detection performance when pre-trained on different subsets of JFT-300M from scratch. x-axis is the dataset size in log-scale, y-axis is the detection performance in mAP@[.5,.95] on COCO-minival subset.

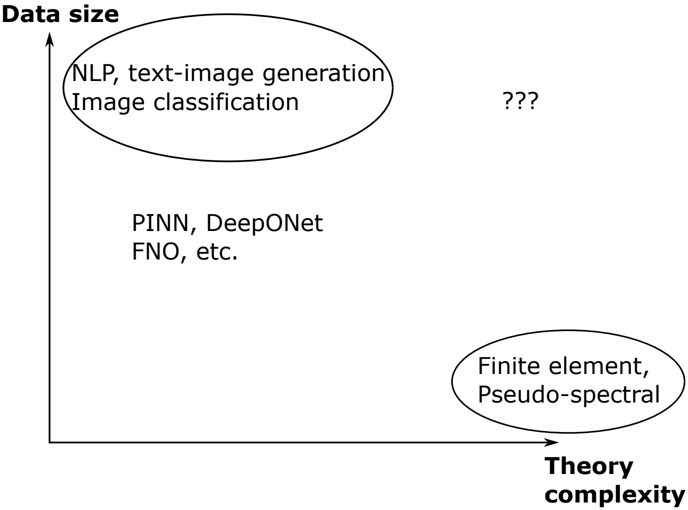
Sun et al. 2017; De Hoop et al. 2022

How much data?



Sun et al. 2017; De Hoop et al. 2022

Theory-driven vs data-driven models



Content

- Wave equation

Content

- Wave equation
- Deep learning aided parareal iteration

Content

- Wave equation
- Deep learning aided parareal iteration
- Numerical results

The second-order Wave Equation

Find solution $u = u(x, t)$ such that

$$\partial_{tt} u = c^2(x)\Delta u, \;\; x \in [-1, 1]^2, 0 \leq t \leq T,$$

The second-order Wave Equation

Find solution $u = u(x, t)$ such that

$$\partial_{tt} u = c^2(x)\Delta u, \;\; x \in [-1, 1]^2, 0 \leq t \leq T,$$

with initial conditions

$$\begin{aligned} u(x, 0) &= u_0(x), \\ \partial_t u(x, 0) &= p_0(x). \end{aligned}$$

The second-order Wave Equation

Find solution $u = u(x,t)$ such that

$$\partial_{tt} u = c^2(x)\Delta u, \;\; x \in [-1,1]^2, 0 \le t \le T,$$

with initial conditions

$$\begin{aligned} u(x,0) &= u_0(x), \\ \partial_t u(x,0) &= p_0(x). \end{aligned}$$

Boundary condition is periodic.

Parallel-in-time method

- Pareareal iteration

$$u_{n+1}^{k+1} = \mathcal{G}u_n^{k+1} + (\mathcal{F}u_n^k - \mathcal{G}u_n^k)$$

for k is the iteration and n is the time snapshot

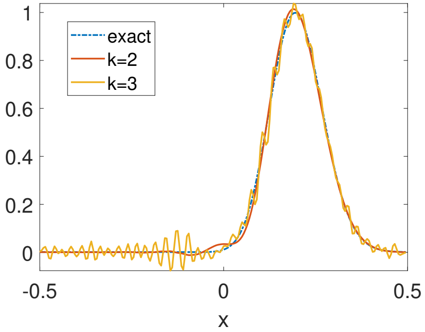
Parallel-in-time method

- Parareal iteration

$$u_{n+1}^{k+1} = \mathcal{G}u_n^{k+1} + (\mathcal{F}u_n^k - \mathcal{G}u_n^k)$$

for k is the iteration and n is the time snapshot

- *Unstable iteration for solving hyperbolic equations*!



DL-aided parareal

- Introduce a neural network $\mathcal{H}_\theta$ to postprocess coarse solution

$$u_{n+1}^{k+1} = \mathcal{H}_\theta \mathcal{G} u_n^{k+1} + (\mathcal{F} u_n^k - \mathcal{H}_\theta \mathcal{G} u_n^k)$$

DL-aided parareal

- Introduce a neural network $\mathcal{H}_\theta$ to postprocess coarse solution

$$u_{n+1}^{k+1} = \mathcal{H}_\theta \mathcal{G} u_n^{k+1} + (\mathcal{F} u_n^k - \mathcal{H}_\theta \mathcal{G} u_n^k)$$

  - Network architecture

DL-aided parareal

- Introduce a neural network $\mathcal{H}_\theta$ to postprocess coarse solution

$$u_{n+1}^{k+1} = \mathcal{H}_\theta \mathcal{G} u_n^{k+1} + (\mathcal{F} u_n^k - \mathcal{H}_\theta \mathcal{G} u_n^k)$$

  - Network architecture
  - Generate training data

DL-aided parareal

- Introduce a neural network $\mathcal{H}_\theta$ to postprocess coarse solution

$$u_{n+1}^{k+1} = \mathcal{H}_\theta \mathcal{G} u_n^{k+1} + (\mathcal{F} u_n^k - \mathcal{H}_\theta \mathcal{G} u_n^k)$$

  - Network architecture
  - Generate training data

- Operate in the energy component

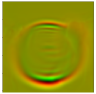
$$\|\nabla u\|_2^2 + \left\| \frac{\partial_t u}{c} \right\|_2^2 = constant$$

JNet

similar to U-Net c.f. Ronneberger, Fischer, and Brox 2015

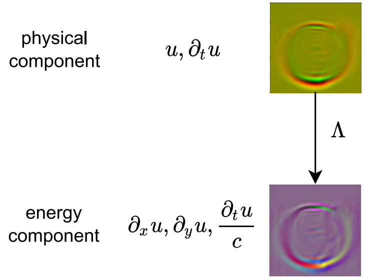
physical component

$u, \partial_t u$



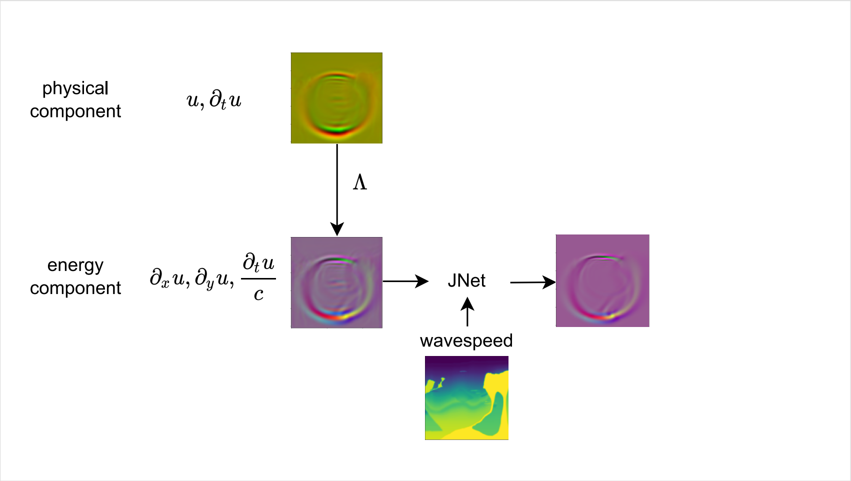
JNet

similar to U-Net c.f. Ronneberger, Fischer, and Brox 2015



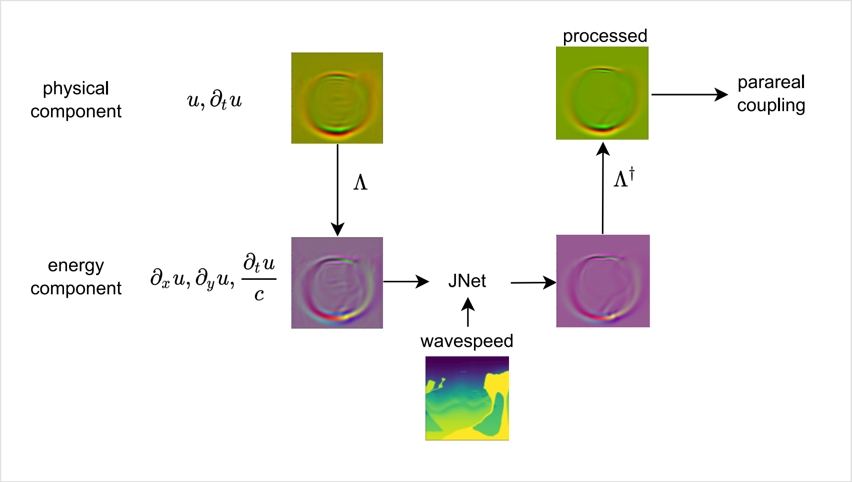
JNet

similar to U-Net c.f. Ronneberger, Fischer, and Brox 2015

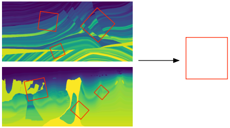


JNet

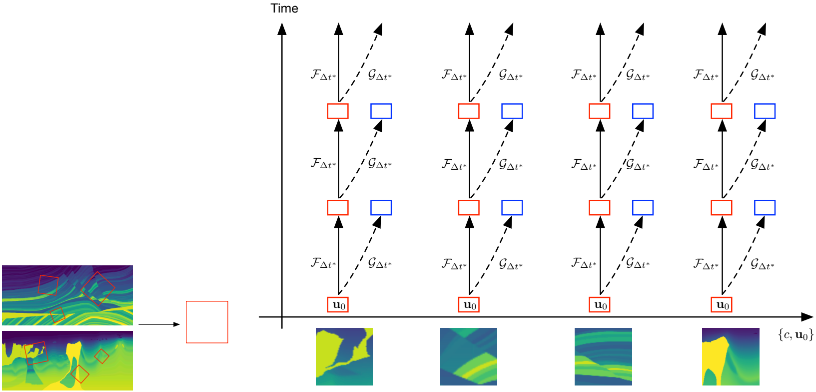
similar to U-Net c.f. Ronneberger, Fischer, and Brox 2015



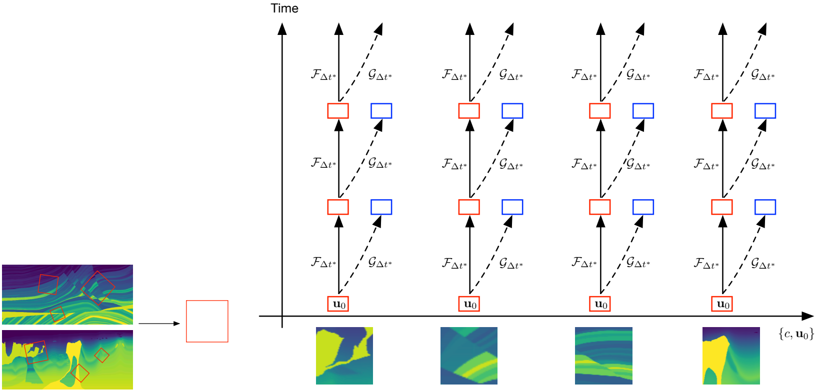
Data generation



Data generation

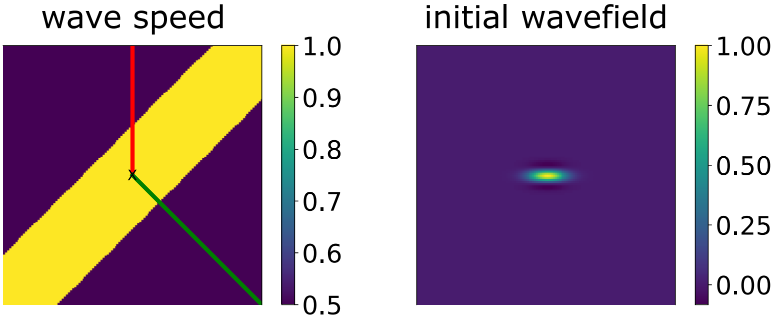


Data generation

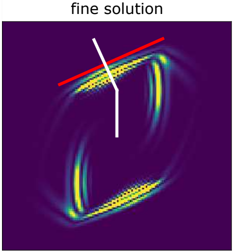


Using the Procrustes parareal c.f. Nguyen and Tsai 2020 to generate data

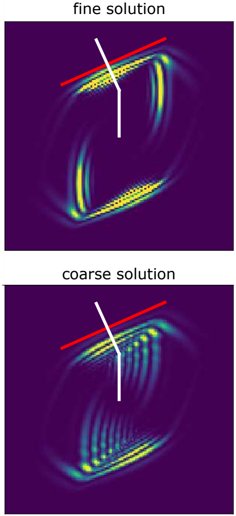
Numerical results - Refraction



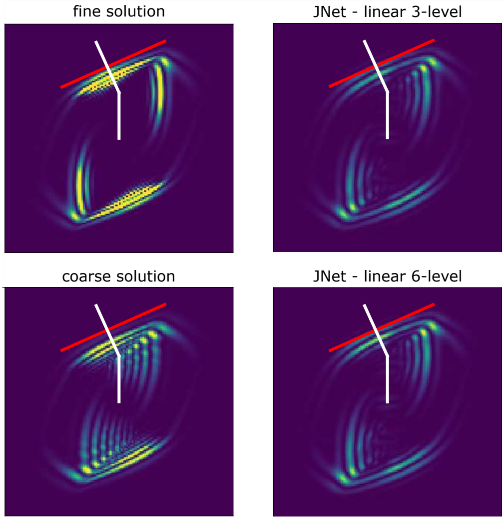
Numerical results - Refraction sequential correction



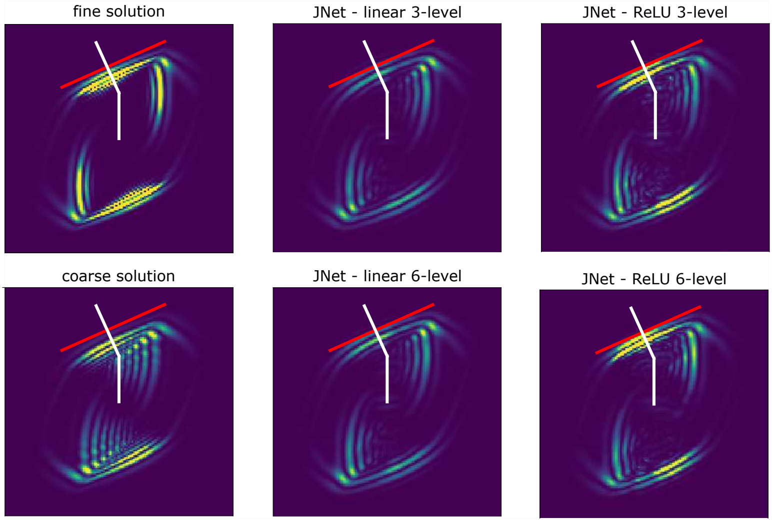
Numerical results - Refraction sequential correction



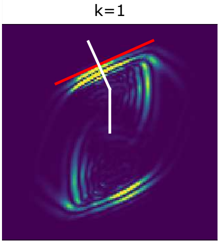
Numerical results - Refraction sequential correction



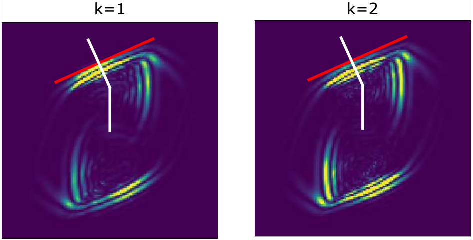
Numerical results - Refraction sequential correction



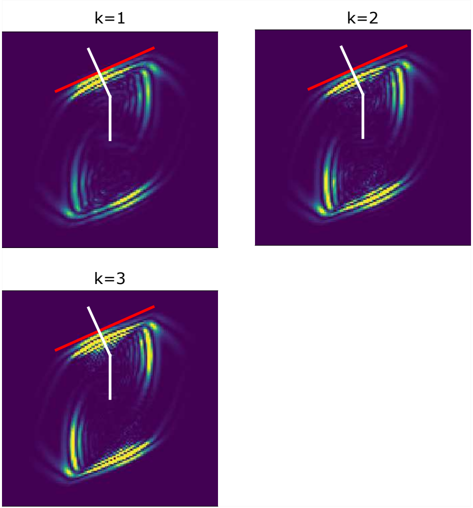
Numerical results - Refraction parareal correction



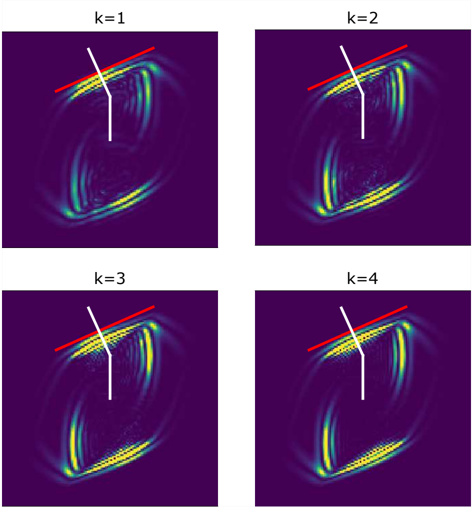
Numerical results - Refraction parareal correction



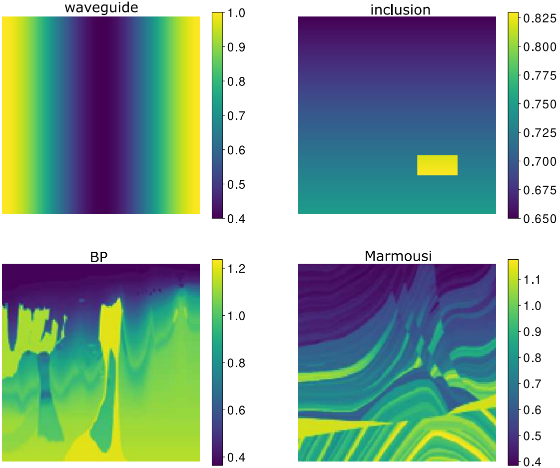
Numerical results - Refraction parareal correction



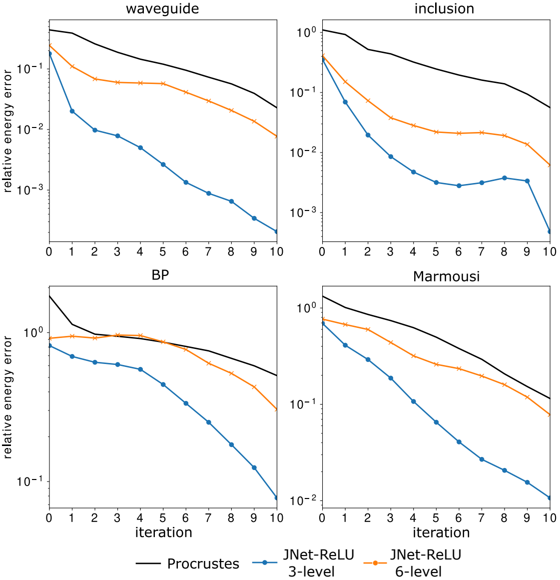
Numerical results - Refraction parareal correction



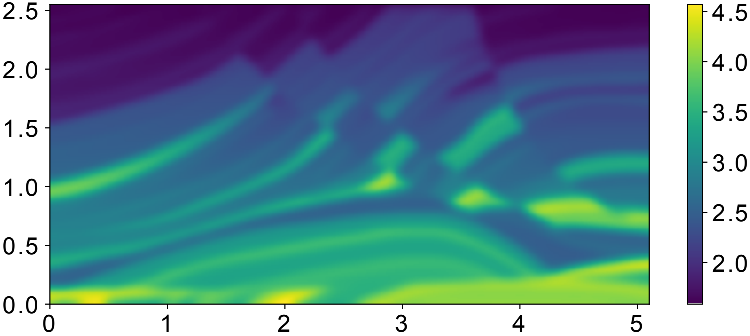
Numerical results - Parareal simulations



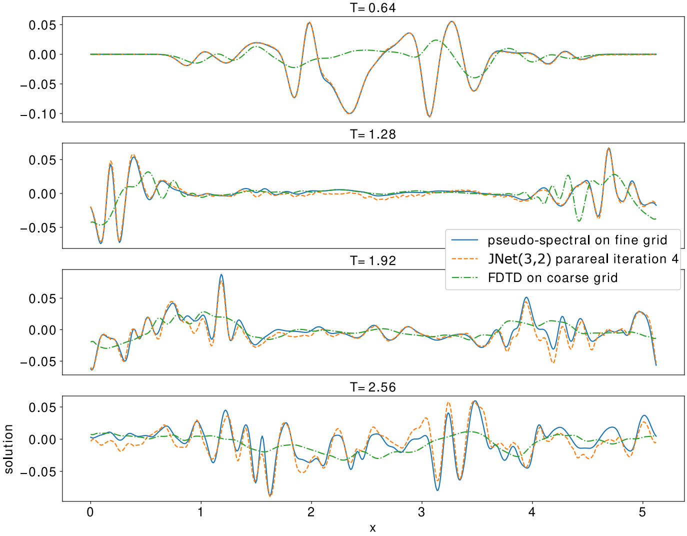
Numerical results - Parareal simulations



Numerical results - Parareal simulations on Marmousi model

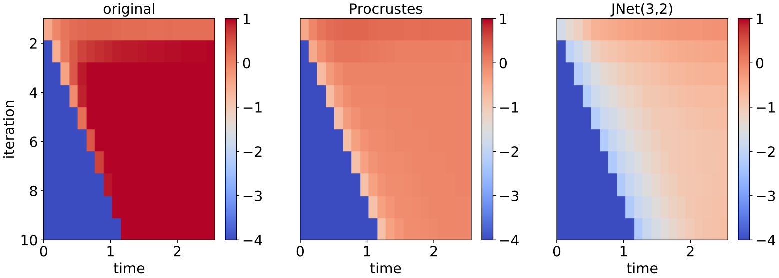


Numerical results - Parareal simulations on Marmousi model



Numerical results - Parareal simulations on Marmousi model

Energy error of solutions in log-10 scale



To sum up

- A deep neural network can stabilize parareal iteration for the wave equation
- Data must consist of stable solutions to train accurate models
- Agnostic to chosen fine/coarse numerical schemes
- Solution accuracy is gained thanks to the parareal coupling

Acknowledgement

- European Research Council Grant - SWING
- National Science Foundation Grant DMS-1913209.
- Texas Advanced Computing Center