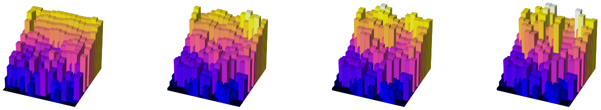
Scientific Machine Learning for Exact Recovery of Nonlinear PDEs



Aleksei G. Sorokin[1,2,3], Fred J. Hickernell[1], Pieterjan Robbe[2], Mohamed Wahib[3], Houman Owhadi[4], Aras Bacho[4], Xianjin Yang[4], Matthieu Darcy[4], Theo Bourdais[4], Edo Calvello[4], Bamdad Hosseini[5], Alexandre Hsu[5]

[1]Illinois Tech, Department of Applied Math, Chicago IL, USA
[2]Sandia National Laboratories, Livermore CA, USA
[3]RIKEN Center for Computational Science, Tokyo Japan
[4]California Institute of Technology, Pasadena, California
[5]University of Washington, Seattle, Washington

Background

PhD Student in Applied Math at Illinois Tech in Chicago IL, advisor Fred J. Hickernell

- (Quasi-)Monte Carlo for fast high dimensional integration
- Gaussian Process Modeling with structure-exploits for scalable UQ
- Scientific Software Development for distribution of efficient algorithms

US DOE SCGSR Fellow[1] at Sandia National Laboratories in Livermore CA

- Multilevel (Quasi-)Monte Carlo for multi-fidelity or infinite-dimensional problems
- Scientific Machine Learning (SciML) for solving nonlinear PDEs

Intern in the RIKEN CCS AI-HPC Research Team in Tokyo Japan

- SciML for Exact PDE Recovery focused on applications to inverse problems
- HPC Compatible Implementations supporting multi-GPU parallelism

[1] Department of Energy Office of Science Graduate Student Research Program

Nonlinear PDEs with Random Coefficients

Applications in Fluid Mechanics, Geophysics, Medical Imaging, ...

Nonlinear PDE $F(v^\star|c) = 0$, solution $v^\star$, random coefficient c

Nonlinear Elliptic PDE

$$F(v|c) = -\Delta v + \kappa v^3 - c$$

Burgers' Equation with an implicit time stepping scheme (step size h_t, constant $\kappa > 0$)

$$F(v|c) = v - h_t(\kappa \Delta v - v \nabla v) - c$$

Nonlinear Darcy flow with forcing term f

$$F(v|c) = -e^c[\nabla c . \nabla v + \Delta v] + \kappa v^3 - f$$

Full Waveform Inversion sets $F(v|c) = P(v) - c$ where P is a numerical forward solver

The Newton–Kantorovich Iteration

A relaxed Gauss–Newton algorithm for root finding in Hilbert spaces of functions [Polyak, 2006]

Nonlinear PDE $F(v^\star|c) = 0$, solution $v^\star$, random coefficient c

Given c and a relaxation parameter $\lambda > 0$, an optimal update $\delta^\star(v)$ is

$$\operatorname*{argmin}_\delta \left(\|F(v+\delta)\|^2 + \lambda\|\delta\|^2\right) \approx -\left[\frac{\partial F^T}{\partial v}(v)\frac{\partial F}{\partial v}(v) + \lambda \mathsf{I}\right]^{-1} \frac{\partial F^T}{\partial v}(v)F(v) =: \delta^\star(v)$$

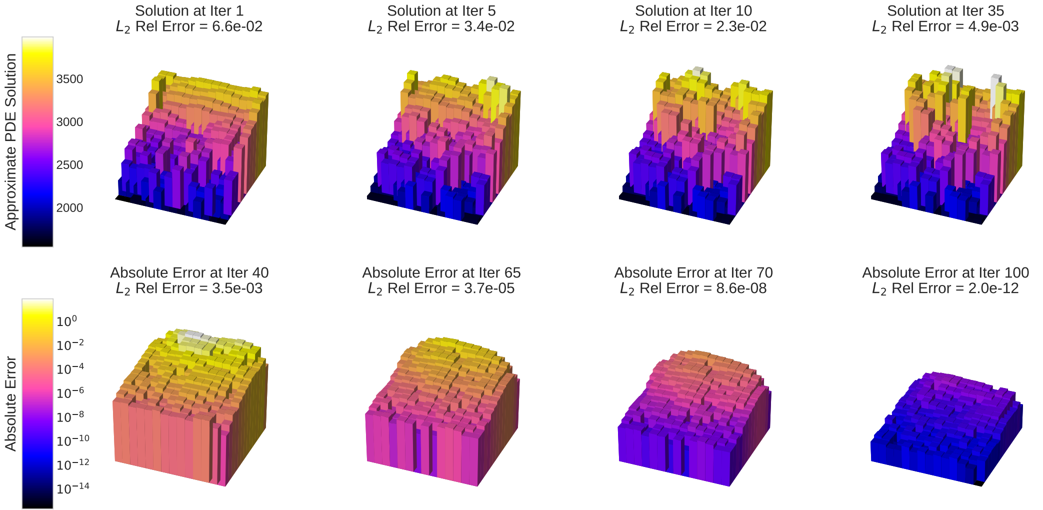
The Newton–Kantorovich iteration with learning rate $\alpha > 0$ sets

$$v_{n+1} = v_n + \alpha\delta^\star(v_n)$$

Relaxation λ interpolates between gradient descent and Gauss–Newton updates

Newton–Kantorovich Iteration for Full Waveform Inversion

Finds subsurface velocity from a surface acoustic signal [Deng et al., 2022, Virieux and Operto, 2009]



SciML for Exact Recovery of Nonlinear PDEs

Learning the approximate Hessian operator enables sciML to converge to machine precision

$$v_{n+1}(c) = v_n(c) - \alpha H^{-1}(v,c,\lambda)\frac{\partial F^T}{\partial v}(v,c)F(v,c)$$

$$H(v,c,\lambda) = L(v,c,\lambda)L^T(v,c,\lambda) = \frac{\partial F^T}{\partial v}(v,c)\frac{\partial F}{\partial v}(v,c) + \lambda \mathsf{I}$$

	Existing sciML Methods	Novel CHONKNORIS Method
Learn to predict	a function (operator learning)	an operator (learning to learn)
Learned Map	$c \mapsto v_S$	$(v,c,\lambda) \mapsto L(v,c,\lambda)$
Input $\mapsto$ Output Sizes	$\mathcal{O}(N) \mapsto \mathcal{O}(N)$	$\mathcal{O}(N) \mapsto \mathcal{O}(N^2)$
Achievable Errors	$\approx \mathcal{O}(10^{-2})$ or $\mathcal{O}(10^{-3})$	machine precision $\mathcal{O}(10^{-16})$

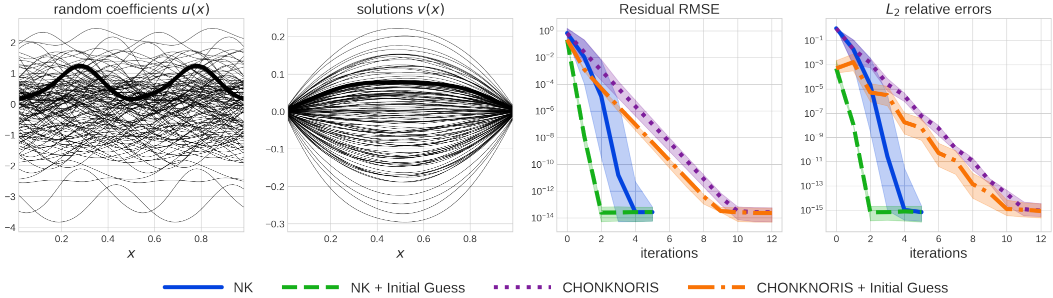
Existing sciML methods

- artificial neural networks [Li et al., 2020, Lu et al., 2021]
- kernel methods [Batlle et al., 2024, Kadri et al., 2016, Nelsen and Stuart, 2021]
- hybrid approaches [Mora et al., 2025, Owhadi, 2023, Owhadi and Yoo, 2019]

Nonlinear Elliptic PDE

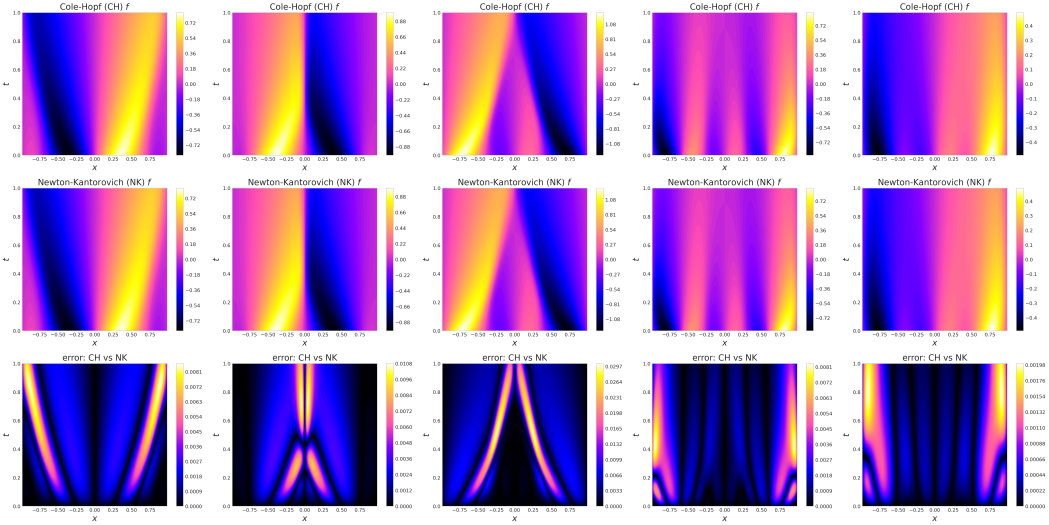
CHONKNORIS achieves machine precision recovery in 12 iterations

$$F(v|c) = -\Delta v + \kappa v^3 - c, \qquad \left[\frac{\partial F}{\partial v}(v)\right](h) = [-\Delta + 3\kappa v^2](h)$$



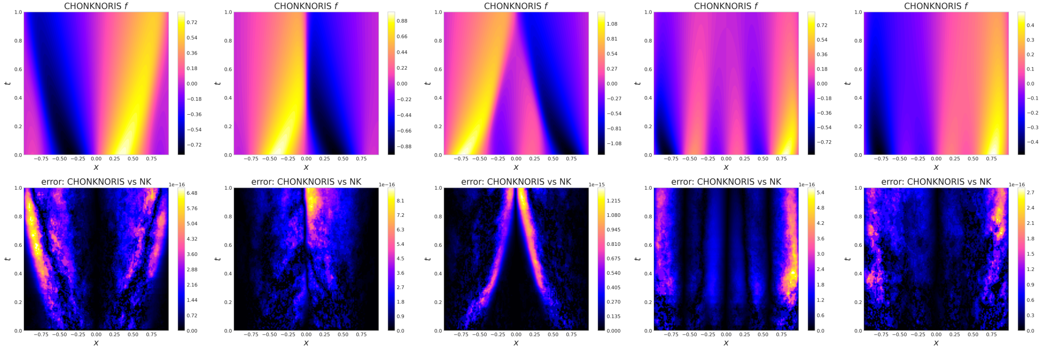
- CHONKNORIS requires more iterations but avoids cubic inversion cost in NK
- Initial guess from existing sciML method can accelerate convergence

Burgers' Equation with Shocks via Implicit Time Stepping



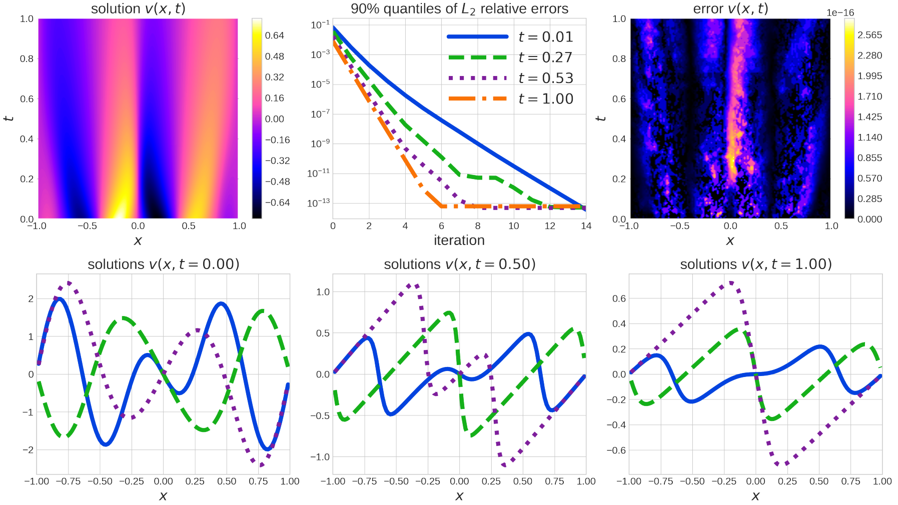
Burgers' Equation with Shocks via Implicit Time Stepping

CHONKNORIS (at each time step) recovers solutions f to within 10^{-16} error



Burgers' Equation with Shocks via Implicit Time Stepping

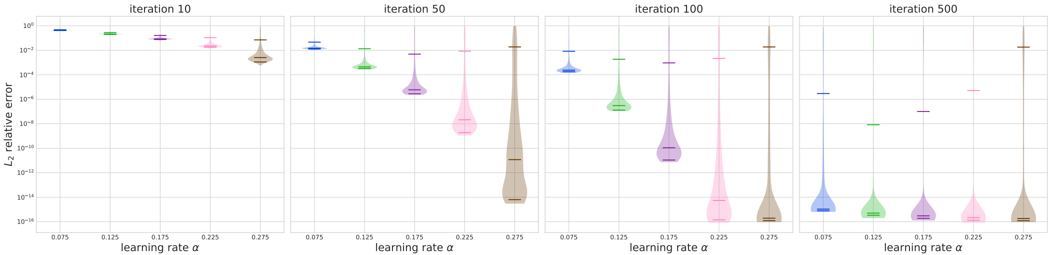
CHONKNORIS (at each time step) recovers solutions f to within 10^{-16} error



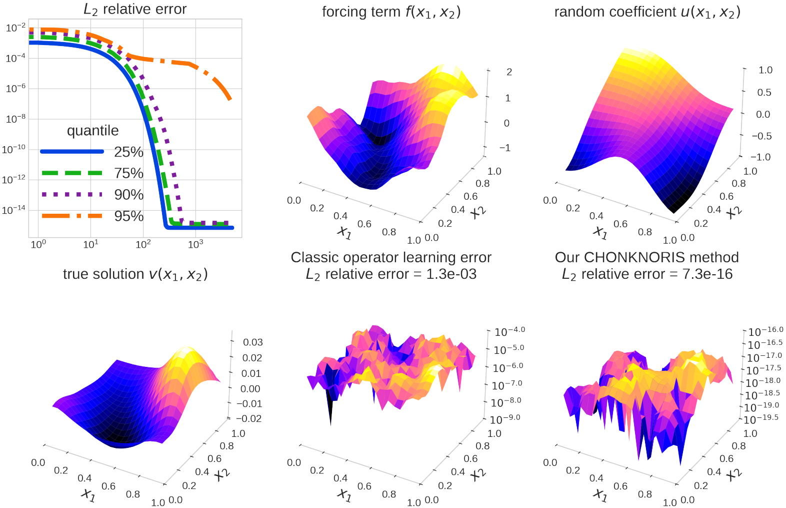
CHONKNORIS Requires a Scalable Implementation and Careful Tuning

Darcy equation with 15×15 discretization requires predicting over > 25k outputs
Automatic selection of (λ, α) improve stability, e.g., using line search

- Relaxation λ must be adaptively decreased for convergence to machine precision
- Learning rate α trades off convergence speed and robustness



Darcy Flow in Two Dimensions



FONKNORIS: Foundation Modeling for Common Jacobian Structures

$$\left[\frac{\partial F}{\partial v}(v,c)\right](h) = \left[A(v,c)\partial_{xx} + B(v,c)\partial_x + C(v,c)\right]h.$$

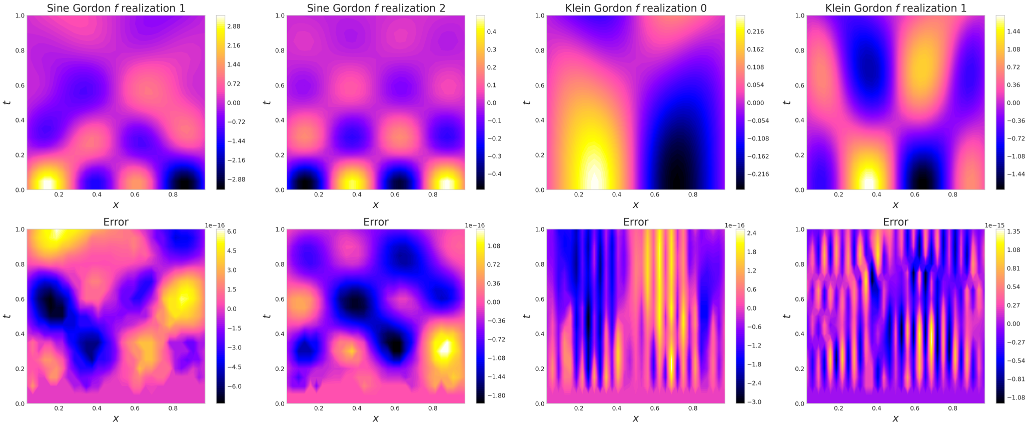
$$H(A,B,C,\lambda) = L(A,B,C,\lambda)L^T(A,B,C,\lambda) = \frac{\partial F^T}{\partial v}(A,B,C)\frac{\partial F}{\partial v}(A,B,C) + \lambda \mathrm{I}$$

CHONKNORIS to learn map $(A,B,C,\lambda) \mapsto L(A,B,C,\lambda)$

FONKNORIS	Partial differential equation	A	B	C
training PDEs	Nonlinear Elliptic	-1	0	$3\kappa v^2$
	Burgers'	$-h_t\kappa$	$h_t v$	$1+h_t\nabla v$
	Nonlinear Darcy Flow	$-e^c$	$-e^c\nabla c$	$3\kappa v^2$
testing PDEs	Sine–Gordon	$-\kappa_1 h_t^2$	0	$1+\kappa_2 h_t^2\cos(v)$
	Klein–Gordon	$-\kappa_1 h_t^2$	0	$1+3\kappa_2 h_t^2 v^2$

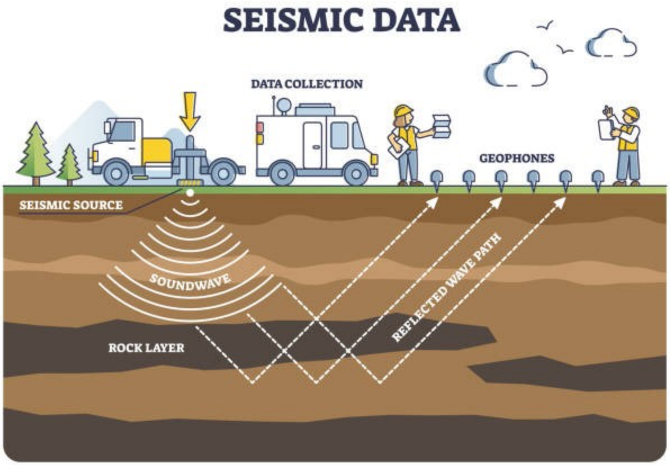
FONKNORIS for Machine Precision Foundation Modeling

CHONKNORIS trained on 1D PDEs can exactly recover unseen Sine–Gordon and Klein–Gordon PDEs



Seismic Imaging Full Waveform Inversion (FWI) Problem

Diagram from `https://www.linkedin.com/pulse/seismic-survey-planning-pmp-framework-himanshu-bhardwaj-mzdaf/`

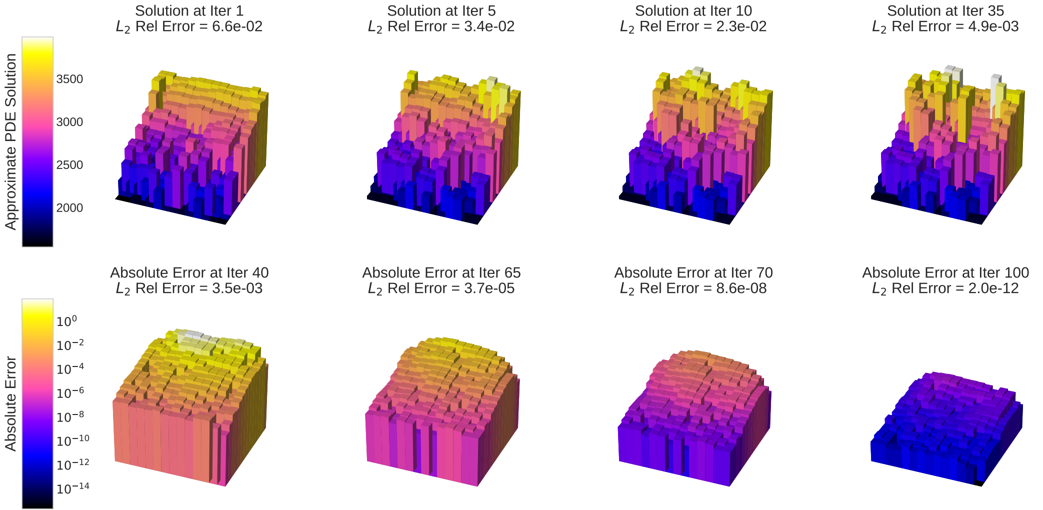


FWI necessitates

- Evaluated Jacobians with millions of entries
- multi-GPU acceleration: forward solves required 5 hours running in parallel on 5 NVIDIA A100 80 GB GPUs
- Training large scale AI models

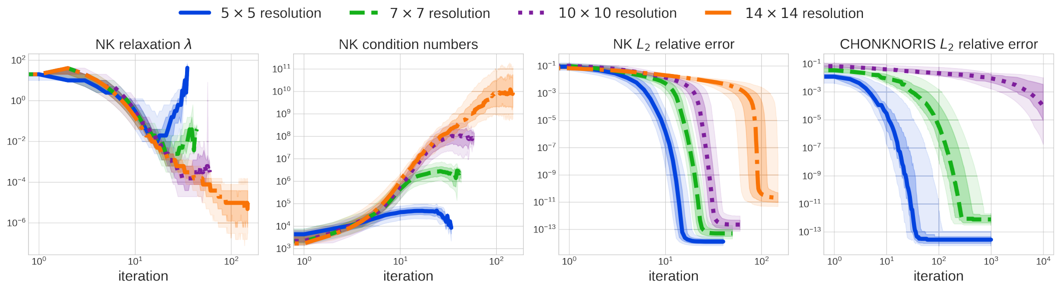
Newton–Kantorovich Iteration for Full Waveform Inversion

Finds subsurface velocity from a surface acoustic signal [Deng et al., 2022, Virieux and Operto, 2009]



Full Waveform Inversion

CHONKNORIS can exactly recover rough solutions at low resolutions

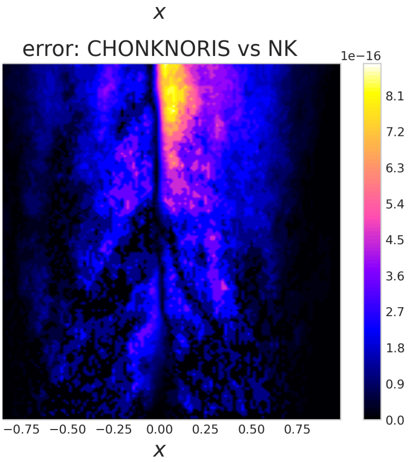


CHONKNORIS struggles to predict the ill conditioned matrices from

- fine discretizations
- rough coefficients
- near convergence perturbations

Summary

- CHONKNORIS for exact recovery of parameterized nonlinear PDEs
- CHONKNORIS for forward problems
  - Nonlinear elliptic PDE
  - Burgers' equation
  - Nonlinear Darcy flow
  - Sine–Gordon
  - Klein–Gordon
- FONKNORIS for machine precision foundation modeling
- CHONKNORIS for inverse problems
  - Seismic imaging FWI



Thank you for listening! Connect or follow my research at alegresor.github.io

References I

Pau Batlle, Matthieu Darcy, Bamdad Hosseini, and Houman Owhadi. Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496: 112549, 2024.

Sou-Cheng T. Choi, Fred J. Hickernell, Rathinavel Jagadeeswaran, Michael J. McCourt, and Aleksei G. Sorokin. Quasi-Monte Carlo software. In Alexander Keller, editor, *Monte Carlo and Quasi-Monte Carlo Methods*, pages 23–47, Cham, 2022. Springer International Publishing. ISBN 978-3-030-98319-2.

Chengyuan Deng, Shihang Feng, Hanchen Wang, Xitong Zhang, Peng Jin, Yinan Feng, Qili Zeng, Yinpeng Chen, and Youzuo Lin. Openfwi: Large-scale multi-structural benchmark datasets for full waveform inversion. *Advances in Neural Information Processing Systems*, 35:6007–6020, 2022.

Fred J. Hickernell, Nathan Kirk, and Aleksei G. Sorokin. Quasi-monte carlo methods: What, why, and how?, 2025. URL https://arxiv.org/abs/2502.03644.

References II

Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17(20):1–54, 2016.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Carlos Mora, Amin Yousefpour, Shirin Hosseinmardi, Houman Owhadi, and Ramin Bostanabad. Operator learning with gaussian processes. *Computer Methods in Applied Mechanics and Engineering*, 434:117581, 2025.

References III

Nicholas H Nelsen and Andrew M Stuart. The random feature model for input-output maps between banach spaces. *SIAM Journal on Scientific Computing*, 43(5): A3212–A3243, 2021.

Houman Owhadi. Do ideas have shape? idea registration as the continuous limit of artificial neural networks. *Physica D: Nonlinear Phenomena*, 444:133592, 2023.

Houman Owhadi and Gene Ryan Yoo. Kernel flows: From learning kernels from data into the abyss. *Journal of Computational Physics*, 389:22–47, 2019.

Boris Teodorovich Polyak. Newton-kantorovich method and its global convergence. *Journal of Mathematical Sciences*, 133:1513–1523, 2006.

Jagadeeswaran Rathinavel and Fred J. Hickernell. Fast automatic bayesian cubature using lattice sampling. *Statistics and Computing*, 29(6):1215–1229, Sep 2019. ISSN 1573-1375. doi: 10.1007/s11222-019-09895-9. URL http://dx.doi.org/10.1007/s11222-019-09895-9.

References IV

Jagadeeswaran Rathinavel and Fred J Hickernell. Fast automatic bayesian cubature using sobol' sampling. In *Advances in Modeling and Simulation: Festschrift for Pierre L'Ecuyer*, pages 301–318. Springer, 2022.

Aleksei G Sorokin and Jagadeeswaran Rathinavel. On bounding and approximating functions of multiple expectations using quasi-monte carlo. In *International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 583–599. Springer, 2022.

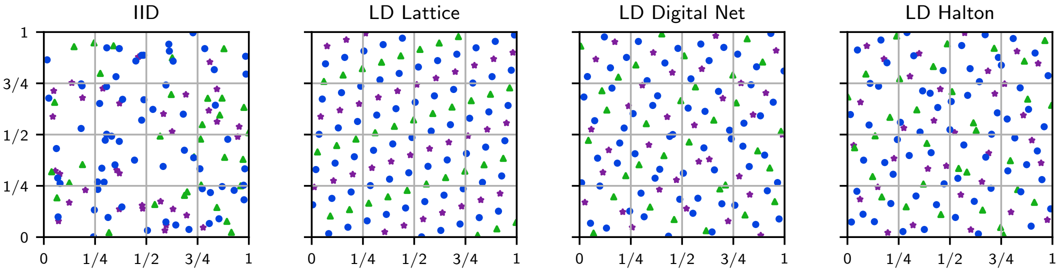
Jean Virieux and Stéphane Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009.

Quasi-Monte Carlo Methods

Efficient algorithms for high dimensional numerical integration [Choi et al., 2022, Hickernell et al., 2025]

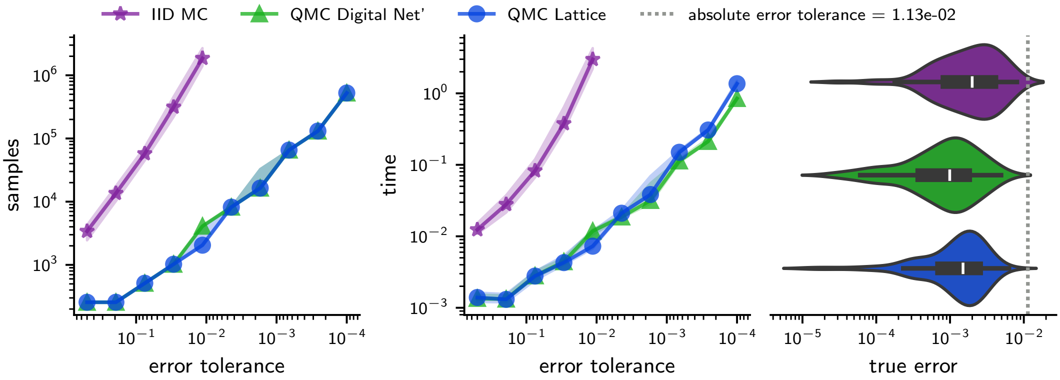
$$\mu = \int_{\mathcal{T}} g(\boldsymbol{t})\lambda(\mathrm{d}\boldsymbol{t}) = \int_{[0,1]^d} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \approx \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i), \qquad \boldsymbol{x}_1,\dots,\boldsymbol{x}_n \in [0,1]^d$$

Classic Monte Carlo has error like $\mathcal{O}(1/\sqrt{n})$ using IID points $[\boldsymbol{x}_i]_{i=1}^n$
QMC has errors like $\mathcal{O}(1/n)$ using low discrepancy (LD) points with greater uniformity



Adaptive QMC Stopping Criterion

Automatically choose sample size n to meet user specified error tolerances [Sorokin and Rathinavel, 2022]



`pip install qmcpy`: QMC Python software qmcsoftware.github.io/QMCSoftware

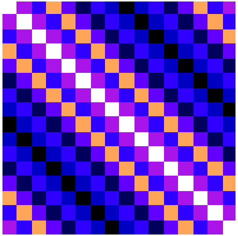
Gaussian Process Regression with Structure-Exploits

Scalable kernel interpolation with built-in uncertainty quantification (UQ) [Rathinavel and Hickernell, 2019, 2022]

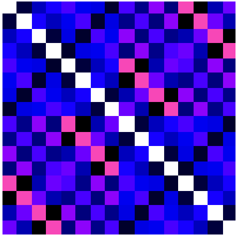
$$f(\boldsymbol{x}) \approx \boldsymbol{K}(\boldsymbol{x})^T \mathsf{K}^{-1} \boldsymbol{f}$$

- SPD kernel $K(\cdot,\cdot)$
- Points $[\boldsymbol{x}_i]_{i=1}^n$
- Values $\boldsymbol{f} = [f(\boldsymbol{x}_i)]_{i=1}^n$
- Basis $\boldsymbol{K}(\cdot) = [K(\cdot,\boldsymbol{x}_i)]_{i=1}^n$
- Gram matrix $\mathsf{K} = [K(\boldsymbol{x}_i,\boldsymbol{x}_j)]_{i,j=1}^N$

SI Lattice K



DSI Digital Net K



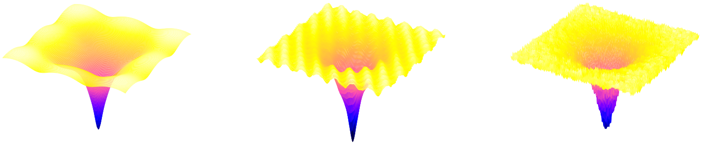
Classic GPR costs $\mathcal{O}(n^3)$ as we must compute the matrix inverse K^{-1}

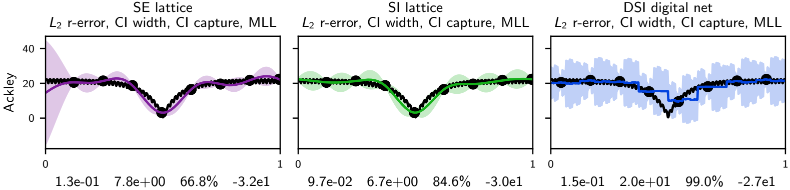
Fast GPR cost only $\mathcal{O}(n \log n)$ by pairing (SI/DSI) kernels K to LD points $[\boldsymbol{x}_i]_{i=1}^n$

- Exploit structure in the Gram matrix K: diagonalizable by fast Fourier transforms

Fast Gaussian Process Regression

SE grid	SI lattice	DSI digital net
error = 3.7e-2	error = 2.8e-2	error = 2.6e-2
time = 1.6e0	time = 8.0e-4	time = 1.5e-3
MLL = 6.3e3	MLL = -3.8e2	MLL = -4.5e3





`pip install fastgps`: Scalable GPR Python software alegresor.github.io/fastgps