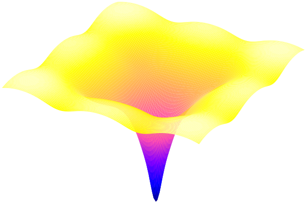
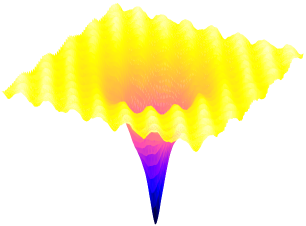
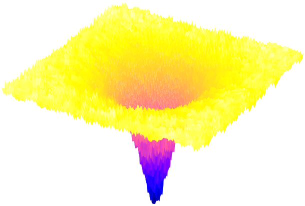
Algorithms and Scientific Software for Quasi-Monte Carlo, Fast Gaussian Process Regression, and Scientific Machine Learning

Aleksei Gregory Sorokin. asorokin@hawk.iit.edu. alegresor.github.io. Advisor Fred J. Hickernell
PhD Defense. Illinois Institute of Technology, Department of Applied Mathematics. 2025.

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







Quasi-Monte Carlo (QMC) Background

Efficient algorithms for high-dimensional numerical integration

Quasi-Monte Carlo (QMC) Background

Efficient algorithms for high-dimensional numerical integration

Motivation

- QMC methods replace independent points with low-discrepancy (LD) points
- LD sequences evenly cover the unit cube in high dimensions
- QMC has faster convergence for "smooth" functions [81, 23, 63, 25, 72, 105, 24]
- Randomizing LD sequences [22, 83, 70, 71, 84, 78, 119] helps to
  - Enable QMC error estimation, e.g., using randomizations of LD points for IID estimates
  - Avoid boundary observations, as randomized LD points lie in $(0, 1)$ almost surely
  - Avoid fooling functions, which are theoretically possible for deterministic LD sequences

Quasi-Monte Carlo (QMC) Background

Efficient algorithms for high-dimensional numerical integration

Motivation

- QMC methods replace independent points with low-discrepancy (LD) points
- LD sequences evenly cover the unit cube in high dimensions
- QMC has faster convergence for "smooth" functions [81, 23, 63, 25, 72, 105, 24]
- Randomizing LD sequences [22, 83, 70, 71, 84, 78, 119] helps to
  - Enable QMC error estimation, e.g., using randomizations of LD points for IID estimates
  - Avoid boundary observations, as randomized LD points lie in $(0,1)$ almost surely
  - Avoid fooling functions, which are theoretically possible for deterministic LD sequences

Applications

- Financial modeling, especially for option pricing [53, 67, 68, 69, 126, 38]
- Solving PDEs (partial differential equations) [41, 65, 66, 42, 64, 101]
- Ray tracing, for simulating light transport in graphics rendering [51, 91, 121]

IID Points versus Randomized Low-Discrepancy (LD) Sequences

IID (independent identically distributed) points have gaps and clusters. LD points have more even coverage.

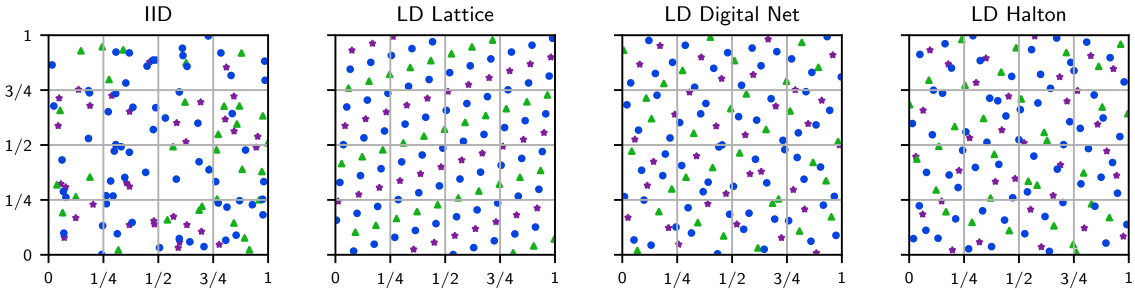


Figure: Purple stars for the first 32 points, green triangles for the next 32, and blue circles for the subsequent 64. Extensible constructions of a lattice with a random shift, a digital net with a nested uniform scramble, and a Halton point set with a linear matrix scramble and permutation scramble.

Quasi-Monte Carlo with LD Points vs IID Monte Carlo

The more even coverage of LD points enables QMC to converge faster for "smooth" functions

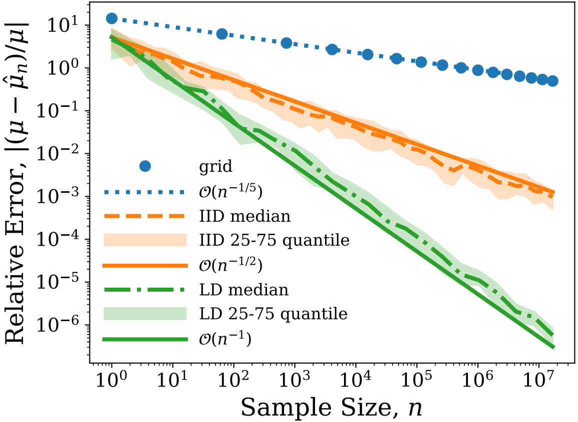


Figure: Approximation of the 6-dimensional Keister function [58].

Gaussian Process (GP) Regression Background

Flexible interpolation models with built-in uncertainty quantification (UQ) [94, 31]

Gaussian Process (GP) Regression Background

Flexible interpolation models with built-in uncertainty quantification (UQ) [94, 31]

Advantages

- UQ from (computable) posterior distribution conditioned on observations
- Equivalent to kernel interpolation in a RKHS (reproducing kernel Hilbert space)

Gaussian Process (GP) Regression Background

Flexible interpolation models with built-in uncertainty quantification (UQ) [94, 31]

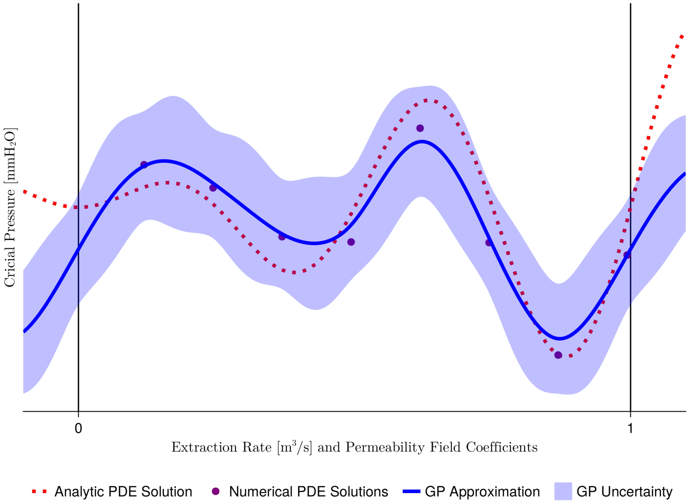
Advantages

- UQ from (computable) posterior distribution conditioned on observations
- Equivalent to kernel interpolation in a RKHS (reproducing kernel Hilbert space)

Applications

- Bayesian optimization, for finding global minima of expensive simulations [32, 106, 125].
- Solving PDEs, with deterministic [18, 14, 15, 6, 75] or random coefficients [54, 55, 5, 114]
- Bayesian cubature, possible since the integral of a GP is still Gaussian [9, 82, 36, 96, 97, 95]
- Reliability analysis, to efficiently predict the probability of system failure [92, 98, 28, 3, 127, 109]

Gaussian Processes (GPs) for Interpolation with Uncertainty Quantification



Multitask Gaussian Processes (MTGPs)

For modeling correlated simulations which capture inter-task and intra-task covariance [7, 10]

Multitask Gaussian Processes (MTGPs)

For modeling correlated simulations which capture inter-task and intra-task covariance [7, 10]

Applications

- Multitask Bayesian optimization, for multi-objective- or meta-learning [118, 60, 49, 20, 112]
- Medical treatment analysis, for modeling correlated treatment effects [1, 29, 33, 13]
- Engineering applications, for performance metrics under design controls [11, 74]

Multitask Gaussian Processes (MTGPs)

For modeling correlated simulations which capture inter-task and intra-task covariance [7, 10]

Applications

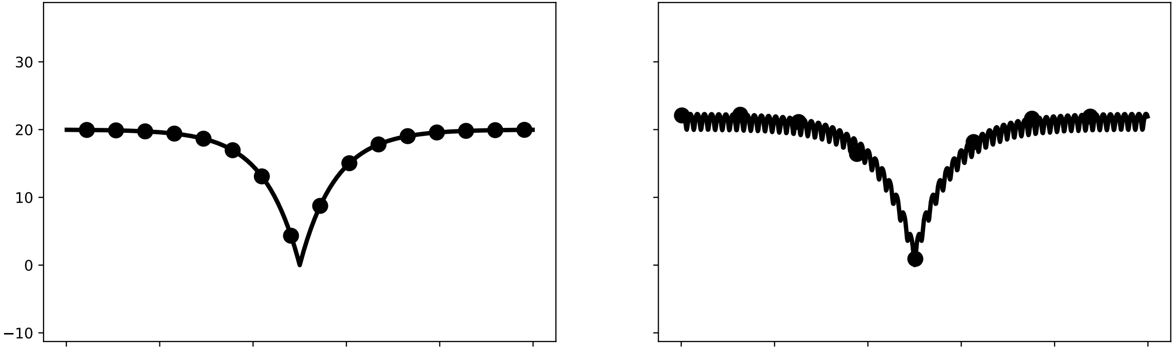
- Multitask Bayesian optimization, for multi-objective- or meta-learning [118, 60, 49, 20, 112]
- Medical treatment analysis, for modeling correlated treatment effects [1, 29, 33, 13]
- Engineering applications, for performance metrics under design controls [11, 74]

Applications of GPs with derivatives observations [94, Chapter 9.4]

- Bayesian optimization, for surface terrain reconstruction [30] and training ML models [89]
- Bayesian cubature, where derivative information enhances GP predictions [124]
- Modeling dynamical systems, with connected function and derivative observations [107]
- Solving PDEs, where the function and derivative observations are optimized to satisfy the governing equation and minimize the RKHS norm [14, 15]

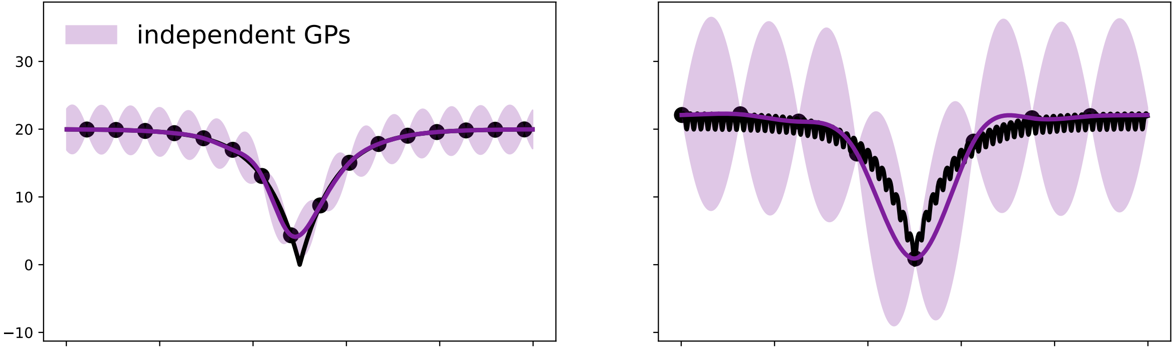
Independent GPs vs a Multitask GP (MTGP)

Low Fidelity Left, High Fidelity Right



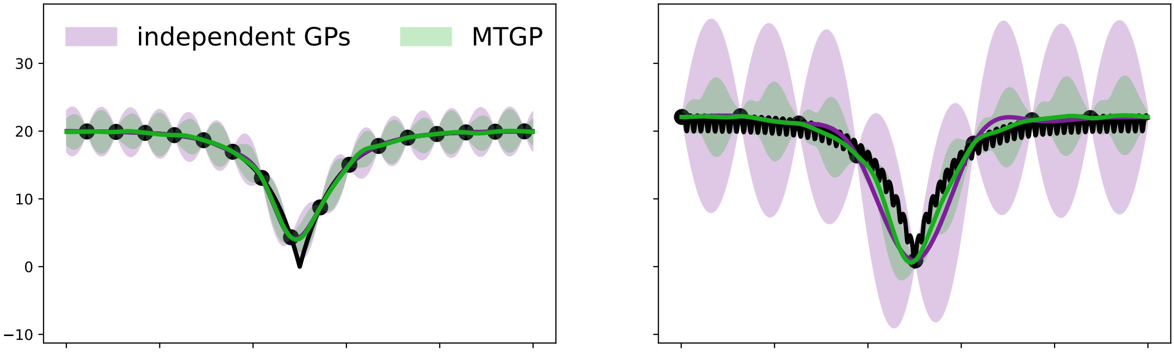
Independent GPs vs a Multitask GP (MTGP)

Low Fidelity Left, High Fidelity Right



Independent GPs vs a Multitask GP (MTGP)

Low Fidelity Left, High Fidelity Right



Accelerated GPs for Controlled Experiments

GP fitting requires solving a linear system in an $n \times n$ Gram matrix which is symmetric-positive-definite (SPD)

Classic GPs fitting and optimization requires $\mathcal{O}(n^3 + n^2 d)$ computations and $\mathcal{O}(n^2)$ storage

Accelerated GPs for Controlled Experiments

GP fitting requires solving a linear system in an $n \times n$ Gram matrix which is symmetric-positive-definite (SPD)

Classic GPs fitting and optimization requires $\mathcal{O}(n^3 + n^2 d)$ computations and $\mathcal{O}(n^2)$ storage

Existing acceleration methods (not explored here)

- Preconditioned conjugate gradient (PCG) [34], which facilitates HPC-GPU implementations
  - Reduces computations to $\mathcal{O}(n^2 d)$
  - Requires well-conditioned Gram matrices

Accelerated GPs for Controlled Experiments

GP fitting requires solving a linear system in an $n \times n$ Gram matrix which is symmetric-positive-definite (SPD)

Classic GPs fitting and optimization requires $\mathcal{O}(n^3 + n^2 d)$ computations and $\mathcal{O}(n^2)$ storage

Existing acceleration methods (not explored here)

- Preconditioned conjugate gradient (PCG) [34], which facilitates HPC-GPU implementations
  - Reduces computations to $\mathcal{O}(n^2 d)$
  - Requires well-conditioned Gram matrices
- Structured kernel interpolation [103, 35, 122, 123], which uses grid points and product kernels
  - Creates Kronecker (or Toeplitz) Gram matrices
  - Reduces storage to $\mathcal{O}(n^{2/d} d)$
  - Reduces computations to $\mathcal{O}(n^{2/d} d^2 + n^{3/d} d)$

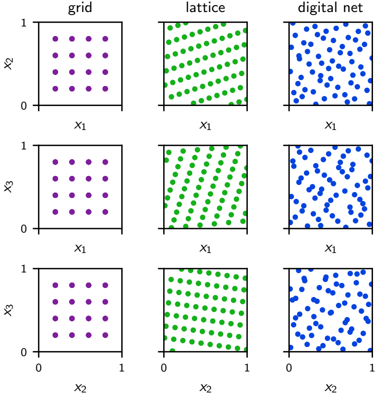


Figure: Projections of $n = 64$ points in $d = 3$ dimensions from a grid, an LD lattice, and an LD digital net.

Fast GPs Pairing Low-Discrepancy (LD) Points with Special Kernels

Reduce high GP costs by forcing structure into the Gram matrix [129, 128]

- $\mathcal{O}(n \log n + nd)$ computations, compared to standard $\mathcal{O}(n^3 + n^2 d)$ requirement, and
- $\mathcal{O}(n)$ memory, compared to standard $\mathcal{O}(n^2)$ requirement

Fast GPs Pairing Low-Discrepancy (LD) Points with Special Kernels

Reduce high GP costs by forcing structure into the Gram matrix [129, 128]

- $\mathcal{O}(n \log n + nd)$ computations, compared to standard $\mathcal{O}(n^3 + n^2 d)$ requirement, and
- $\mathcal{O}(n)$ memory, compared to standard $\mathcal{O}(n^2)$ requirement

Fast GPs exploit Gram matrix structure present when pairing

- Special low-discrepancy (LD) sequences, either lattices or digital nets, to
- Shift-invariant (SI) or digitally-SI (DSI) kernels, of varying smoothness

Fast GPs Pairing Low-Discrepancy (LD) Points with Special Kernels

Reduce high GP costs by forcing structure into the Gram matrix [129, 128]

- $\mathcal{O}(n \log n + nd)$ computations, compared to standard $\mathcal{O}(n^3 + n^2 d)$ requirement, and
- $\mathcal{O}(n)$ memory, compared to standard $\mathcal{O}(n^2)$ requirement

Fast GPs exploit Gram matrix structure present when pairing

- Special low-discrepancy (LD) sequences, either lattices or digital nets, to
- Shift-invariant (SI) or digitally-SI (DSI) kernels, of varying smoothness

Requirements

- Controlling design of experiments, in order to use LD points
- Using uncommon kernels, either the SI or DSI kernels, which are practically performant

Fast GPs Pairing Low-Discrepancy (LD) Points with Special Kernels

Reduce high GP costs by forcing structure into the Gram matrix [129, 128]

- $\mathcal{O}(n \log n + nd)$ computations, compared to standard $\mathcal{O}(n^3 + n^2 d)$ requirement, and
- $\mathcal{O}(n)$ memory, compared to standard $\mathcal{O}(n^2)$ requirement

Fast GPs exploit Gram matrix structure present when pairing

- Special low-discrepancy (LD) sequences, either lattices or digital nets, to
- Shift-invariant (SI) or digitally-SI (DSI) kernels, of varying smoothness

Requirements

- Controlling design of experiments, in order to use LD points
- Using uncommon kernels, either the SI or DSI kernels, which are practically performant

Applications

- Solving PDEs, often with random coefficients sampled at LD locations [54, 55, 114, 116]
- Fast Bayesian cubature, which enables error estimation for QMC [96, 97, 95]
- Discrepancy computation, for terms appearing in cubature error bounds [43, 44]

Fast GPs using LD points and SI/DSI kernels

Maintain accuracy with reduced compute from $\mathcal{O}(n^3+n^2d)$ to $\mathcal{O}(n\log n+nd)$ and storage from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$

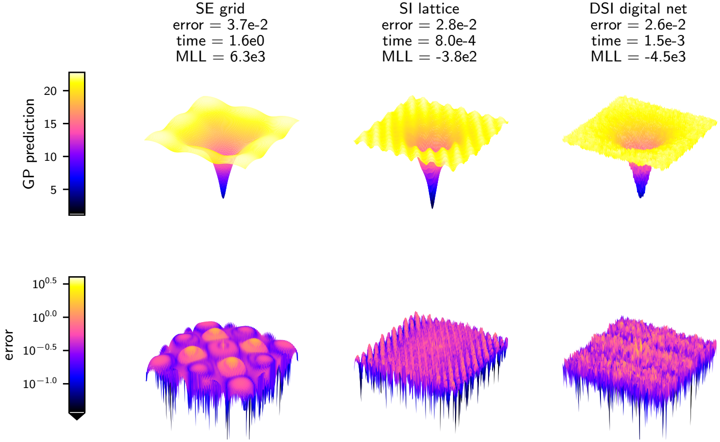


Figure: GP modeling of the 2-dimensional Ackley function with $n = 4096$ points.

Scientific Machine Learning (sciML) for Solving PDEs

Scientific Machine Learning (sciML) for Solving PDEs

Advantages

- Easy setup of solvers, with just a few lines of code specifying the PDE
- Flexible parameterizations, from a zoo of neural architectures or GP kernels
- Scalable training, enabled by GPU acceleration libraries such as `PyTorch` [90]
- Rapid inference, thanks to fast and parallelizable evaluation of sciML models; especially evident in operator learning of PDEs with random coefficients [8, 62]

Scientific Machine Learning (sciML) for Solving PDEs

Advantages

- Easy setup of solvers, with just a few lines of code specifying the PDE
- Flexible parameterizations, from a zoo of neural architectures or GP kernels
- Scalable training, enabled by GPU acceleration libraries such as `PyTorch` [90]
- Rapid inference, thanks to fast and parallelizable evaluation of sciML models; especially evident in operator learning of PDEs with random coefficients [8, 62]

Use of low-discrepancy points in sciML

- Collocation points for training sciML models, especially neural architectures [76, 12, 59]
- Solving PDEs with fast GPs, often with random coefficients [54, 55, 114]
- As space filling designs, may be useful when pairing with Sobolev inequalities [27, 26]

Popular SciML Models

Popular SciML Models

Neural network architectures for physics informed ML [57] and operators learning [61]

1. Physics Informed Neural Networks (PINNs), for PDEs with deterministic coefficients [93]
2. Deep Operator Networks (DeepONets), for PDEs with random coefficients [77]
3. Fourier Neural Operators (FNOs), for resolution-invariant operator learning [73]

Popular SciML Models

Neural network architectures for physics informed ML [57] and operators learning [61]

1. Physics Informed Neural Networks (PINNs), for PDEs with deterministic coefficients [93]
2. Deep Operator Networks (DeepONets), for PDEs with random coefficients [77]
3. Fourier Neural Operators (FNOs), for resolution-invariant operator learning [73]

Gaussian process (GP) models, with strong theoretical backing and convergence guarantees [87]

1. GPs methods for PDEs, with deterministic coefficients which optimize function and derivatives observations to match governing equations and minimize RKHS norm [14, 15]
2. GP operator learning, often via vector-valued GPs with resolution-invariance [56, 80, 5]
3. Hybrid GP and neural network operator learning, often using GPs with neural prior means to leverage the theoretical backing of GPs and flexibility of neural architectures [79, 86, 88]

Modeling Radiative Transfer with a SciML DeepONet Neural Architecture

The DeepONet accelerates inference while honoring PDE constraints and overcoming the ray effect

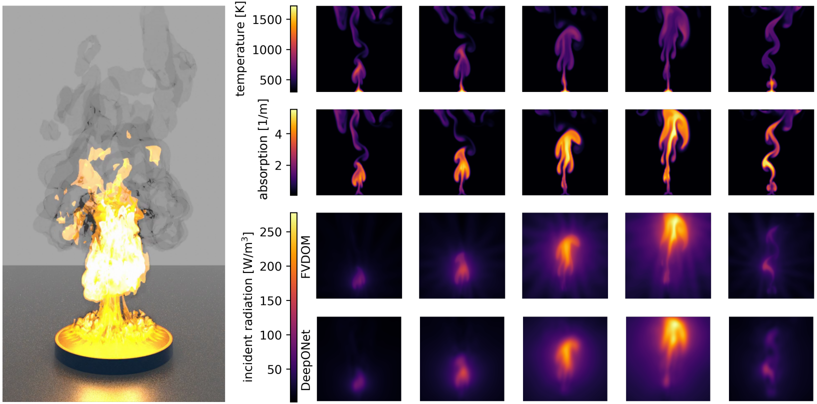


Figure: Predicted incident radiation from unknown temperature and absorption coefficients in a pool fire compared to reference finite volume discrete-ordinate method (FVDOM).

My Papers: Collaborations Across Academia, National Labs, and Industry

My Papers: Collaborations Across Academia, National Labs, and Industry

- [116]: Fast Gaussian Process Regression for High-Dimensional Functions with Derivative Information.
  During my 2025 DOE SCGSR Program fellowship at Sandia National Laboratories.
  Collaboration with Pieterjan M Robbe and Fred J Hickernell.
  Published in the First International Conference on Probabilistic Numerics, PMLR.
- [48]: Quasi-Monte Carlo Methods: What, Why, and How?
  Collaboration with Fred J Hickernell and Nathan Kirk.
  Published in the Monte Carlo and Quasi-Monte Carlo Methods 2024 Proceedings.
- [110]: On Bounding and Approximating Functions of Multiple Expectations Using Quasi-Monte Carlo.
  Collaboration with Jagadeeswaran Rathinavel.
  Published in the Monte Carlo and Quasi-Monte Carlo Methods 2022 Proceedings.

My Papers: Collaborations Across Academia, National Labs, and Industry

- [16]: Challenges in Developing Great Quasi-Monte Carlo Software.
  Collaboration with Sou-Cheng T Choi, Yuhan Ding, Fred J Hickernell, and Jagadeeswaran Rathinavel.
  Published in the Monte Carlo and Quasi-Monte Carlo Methods 2022 Proceedings.
- [17]: Quasi-Monte Carlo Software.
  Collaboration with Sou-Cheng T Choi, Fred J Hickernell, Jagadeeswaran Rathinavel, and Michael J McCourt.
  Published in the Monte Carlo and Quasi-Monte Carlo Methods 2020 Proceedings.
- [111]: (Quasi-)Monte Carlo Importance Sampling with QMCPy.
  Collaboration with Fred J Hickernell, Sou-Cheng T Choi, Michael J McCourt, and Jagadeeswaran Rathinavel.
  Published in the 2021 IIT Undergraduate Research Journal.

My Papers: Collaborations Across Academia, National Labs, and Industry

- [113]: A Neural Surrogate Solver for Radiation Transfer.
  During my 2024 Scientific Machine Learning Researcher appointment at FM.
  Collaboration with Xiaoyi Lu and Yi Wang.
  Published through the NeurIPS 2024 Workshop on Data-Driven and Differentiable Simulations, Surrogates, and Solvers.
- [112]: SigOpt Mulch: An Intelligent System for AutoML of Gradient Boosted Trees.
  During my 2021 Machine Learning Engineer appointment at SigOpt (acquired by Intel).
  Collaboration with Michael McCourt, Xinran Zhu, Eric Hans Lee, and Bolong Cheng.
  Published in a 2023 Knowledge-Based Systems Journal.
- [114]: Computationally Efficient and Error Aware Surrogate Construction for Numerical Solutions of Subsurface Flow Through Porous Media.
  During my 2023 Graduate Internship appointment at Los Alamos National Laboratory.
  Collaboration with Aleksandra Pachalieva, Daniel O'Malley, James M Hyman, Fred J Hickernell, and Nicolas W Hengartner.
  Published in a 2024 Advances in Water Resources Journal.

My Papers: Collaborations Across Academia, National Labs, and Industry

- [39]: GalCEM. I. An Open-Source Detailed Isotopic Chemical Evolution Code and
  [40]: GalCEM: GALactic Chemical Evolution Model.
  Collaboration with Eda Gjergo, Anthony Ruth, Emanuele Spitoni, Francesca Matteucci, Xilong Fan, Jinning Liang, Marco Limongi, Yuta Yamazaki, Motohiko Kusakabe, and Toshitaka Kajino.
  Published in the 2023 Astrophysical Journal Supplement Series and Astrophysics Source Code Library respectively.
- [108]: QMCPy: A Python Software for Randomized Low-Discrepancy Sequences, Quasi-Monte Carlo, and Fast Kernel Methods.
  Submitted to ACM TOMS (Transactions on Mathematical Software).
- [109]: Credible Intervals for Probability of Failure with Gaussian Processes.
  During my 2022 Givens Associate Internship appointment at Argonne National Laboratories.
  Collaboration with Vishwas Rao.
  Currently, an unpublished preprint.

My Papers: Collaborations Across Academia, National Labs, and Industry

- [50]: Empirical Bernstein and Betting Confidence Intervals for Randomized Quasi-Monte Carlo.
  Collaboration with Art Owen, Aadit Jain, and Fred Hickernell.
  Currently, an unpublished preprint.
- [2], Operator Learning at Machine Precision
  During my 2025 DOE SCGSR fellowship at Sandia National Laboratories.
  Collaboration with Houman Owhadi, Aras Bacho, Xianjin Yang, Théo Bourdais, Edoardo Calvello, Matthieu Darcy, Alex Hsu, and Bamdad Hosseini.
  Currently, an unpublished preprint.
- [115], Fast Bayesian Multilevel Quasi-Monte Carlo
  Developed during my 2025 DOE SCGSR fellowship at Sandia National Laboratories.
  Collaboration with Pieterjan M Robbe and Fred J Hickernell.
  Currently, an unpublished preprint.

My Contributions: Novel Algorithms, Implementations, and Applications

My Contributions: Novel Algorithms, Implementations, and Applications

1. Developed `QMCPy` (`https://qmcsoftware.github.io/QMCSoftware/`) [17, 16, 48, 108, 110], a unified open-source Python library for Quasi-Monte Carlo algorithms
2. Built `LDData` (`https://github.com/QMCSoftware/LDData`), which stores lattice generating vectors, digital net generating matrices, and quality low-discrepancy point sets in standardized formats with an interface in `QMCPy`
3. Implemented into `QMCPy` the first interface to certain randomized low-discrepancy sequences, including lattices, higher-order digital nets, higher-order digital net scrambling with either linear matrix scrambling (LMS) or nested uniform scrambling (NUS), and Halton scrambling with either LMS or NUS [108]
4. Developed vectorized Quasi-Monte Carlo stopping criterion to estimate and quantify error for functions of multiple integrals [110], and provided an implementation in `QMCPy`
5. Derived new digitally-shift-invariant (DSI) kernels whose RKHSs contain smooth functions [108] including a new order 4 smoothness kernel whose form has not appeared elsewhere in the literature

My Contributions: Novel Algorithms, Implementations, and Applications

6. Implemented into `QMCPy` the first Python interface to fast kernel method utilities [108] including shift-invariant (SI) and digitally-shift-invariant (DSI) kernels of varying smoothness as well as the fast Fourier transform in bit-reversed order (FFTBR), the inverse FFTBR (IFFTBR), and the fast Walsh–Hadamard transform (FWHT)
7. Developed `FastGPs` (`https://alegresor.github.io/fastgps/`) [116], an open-source Python library for fast Gaussian process regression algorithms pairing low-discrepancy points with (digitally)-shift-invariant kernels
8. Derived and analyzed novel fast multitask GPs and fast derivative-informed GP [116] with an available implementation in the `FastGPs` package
9. Developed numerous applications of Quasi-Monte Carlo, fast Gaussian process regression, and/or scientific machine learning [109, 114, 113, 2, 115, 112, 39, 40] with collaborators from academia, national labs, and industry

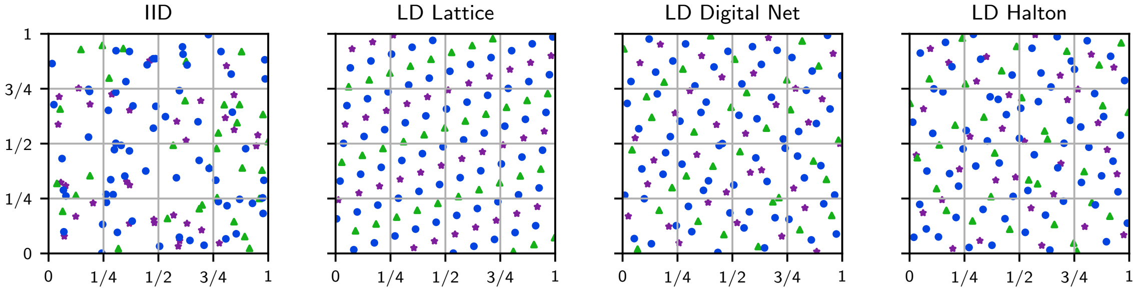
Quasi-Monte Carlo Formulation

QMC is an efficient algorithm for high-dimensional numerical integration

$$\mu = \int_{\mathcal{T}} g(\boldsymbol{t})\lambda(\mathrm{d}\boldsymbol{t}) = \int_{[0,1]^d} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \approx \frac{1}{n}\sum_{i=0}^{n-1} f(\boldsymbol{x}_i), \qquad \boldsymbol{x}_0,\dots,\boldsymbol{x}_{n-1} \in [0,1]^d$$

Classic Monte Carlo has error like $\mathcal{O}(1/\sqrt{n})$ using IID points $(\boldsymbol{x}_i)_{i=0}^{n-1}$

QMC has errors like $\mathcal{O}(1/n)$ using low discrepancy (LD) points with greater uniformity



Adaptive QMC for High-Dimensional Asian Option Pricing

Automatically choose sample size n to meet a user specified error tolerance

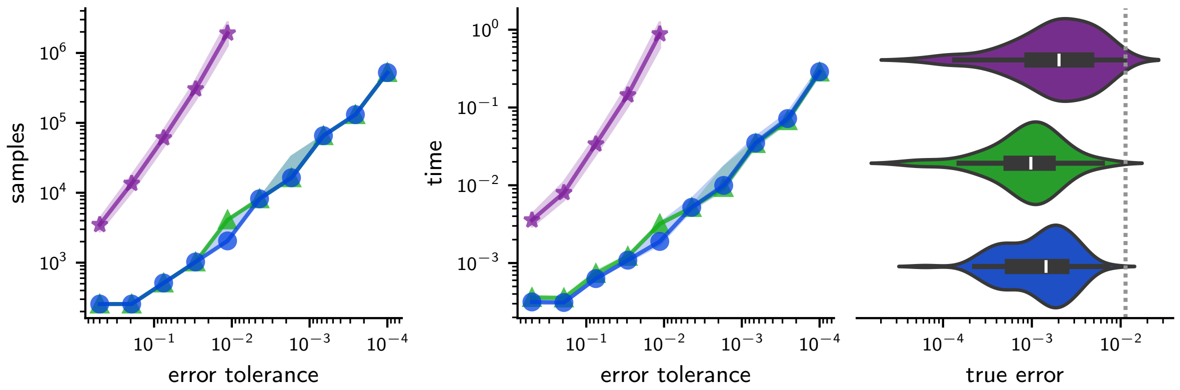


Figure: IID MC vs QMC digital net vs QMC lattice.

QMCPy Software

`pip install qmcpy` or visit qmcsoftware.github.io/QMCSoftware/ [17, 16, 48, 110]

A Python software unifying QMC tools from across the literature

QMCPy Software

`pip install qmcpy` or visit qmcsoftware.github.io/QMCSoftware/ [17, 16, 48, 110]

A Python software unifying QMC tools from across the literature

- Randomized low-discrepancy sequences
  - Lattices: random shifts
  - Digital nets: random digital shifts, LMS, Owen scrambling (NUS), higher-order nets
  - Halton points: digital shifts, permutation scrambles, NUS

QMCPy Software

`pip install qmcpy` or visit qmcsoftware.github.io/QMCSoftware/ [17, 16, 48, 110]

A Python software unifying QMC tools from across the literature

- Randomized low-discrepancy sequences
  - Lattices: random shifts
  - Digital nets: random digital shifts, LMS, Owen scrambling (NUS), higher-order nets
  - Halton points: digital shifts, permutation scrambles, NUS
- Automatic transforms: rewrite problem into $f:[0,1]^d \to \mathbb{R}$ with uniform stochasticity
- Diverse use cases: financial options, parameterized PDEs, sensitivity indices, ...

QMCPy Software

`pip install qmcpy` or visit qmcsoftware.github.io/QMCSoftware/ [17, 16, 48, 110]

A Python software unifying QMC tools from across the literature

- Randomized low-discrepancy sequences
  - Lattices: random shifts
  - Digital nets: random digital shifts, LMS, Owen scrambling (NUS), higher-order nets
  - Halton points: digital shifts, permutation scrambles, NUS
- Automatic transforms: rewrite problem into $f : [0,1]^d \to \mathbb{R}$ with uniform stochasticity
- Diverse use cases: financial options, parameterized PDEs, sensitivity indices, ...
- Adaptive stopping criteria: choosing n to meet user-specified error tolerance ε
  - IID Monte Carlo with guaranteed confidence intervals [46]
  - QMC with replications, see [71] or [85, Chapter 17]
  - QMC via decay tracking of Fourier or Walsh coefficients [47]
  - QMC via Bayesian cubature using fast Gaussian processes [95]
  - Multilevel IID Monte Carlo [37, 100, 102] and Multilevel QMC [38, 101]
  - Vectorized IID Monte Carlo and QMC algorithms for functions of multiple integrands [110]

QMCPy Software

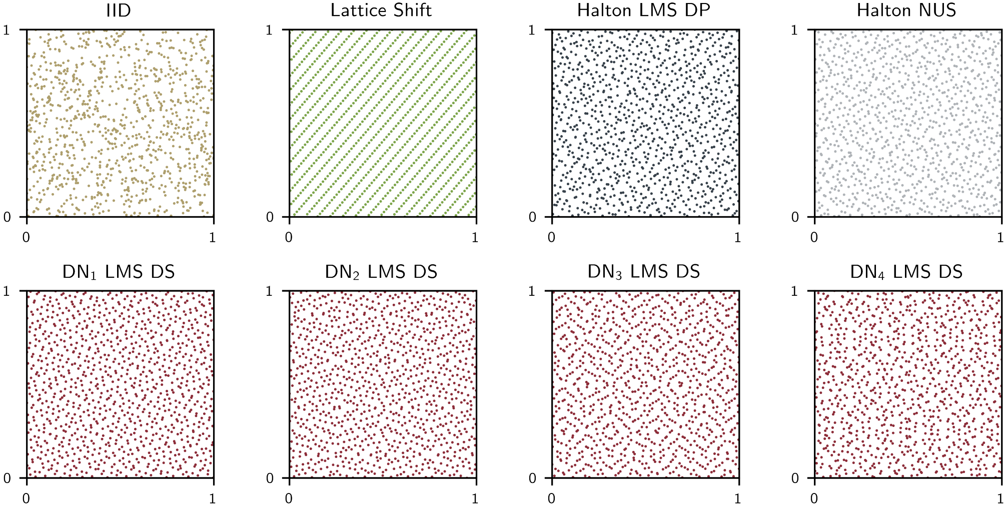
`pip install qmcpy` or visit qmcsoftware.github.io/QMCSoftware/ [17, 16, 48, 110]

A Python software unifying QMC tools from across the literature

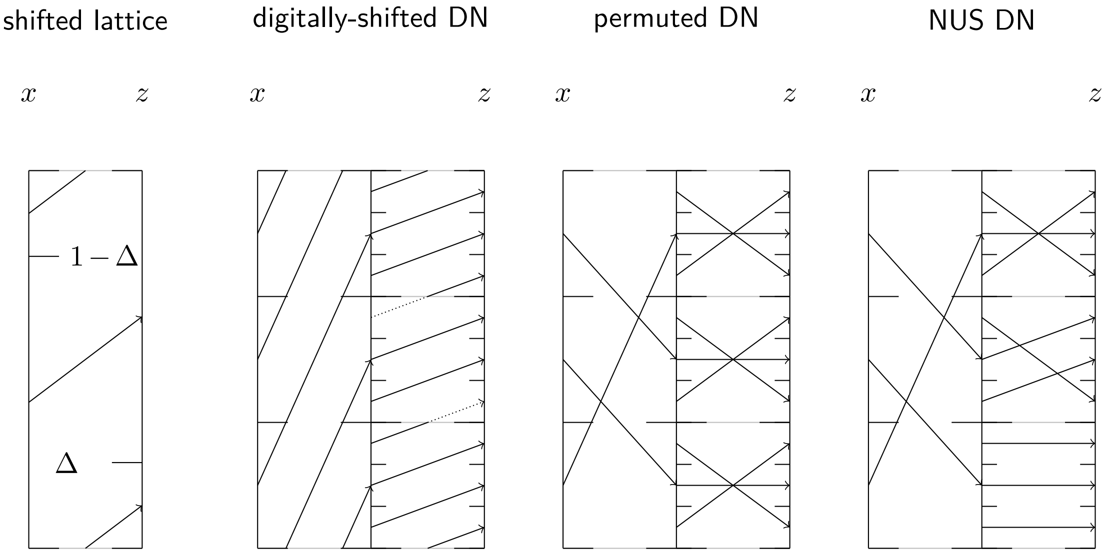
- Randomized low-discrepancy sequences
  - Lattices: random shifts
  - Digital nets: random digital shifts, LMS, Owen scrambling (NUS), higher-order nets
  - Halton points: digital shifts, permutation scrambles, NUS
- Automatic transforms: rewrite problem into $f:[0,1]^d \to \mathbb{R}$ with uniform stochasticity
- Diverse use cases: financial options, parameterized PDEs, sensitivity indices, ...
- Adaptive stopping criteria: choosing n to meet user-specified error tolerance ε
  - IID Monte Carlo with guaranteed confidence intervals [46]
  - QMC with replications, see [71] or [85, Chapter 17]
  - QMC via decay tracking of Fourier or Walsh coefficients [47]
  - QMC via Bayesian cubature using fast Gaussian processes [95]
  - Multilevel IID Monte Carlo [37, 100, 102] and Multilevel QMC [38, 101]
  - Vectorized IID Monte Carlo and QMC algorithms for functions of multiple integrands [110]
- Special kernels and fast transforms: used for fast Gaussian process regression

IID and Low-Discrepancy Points

Including higher-order α digital nets (DN_α) and higher-order scramblings

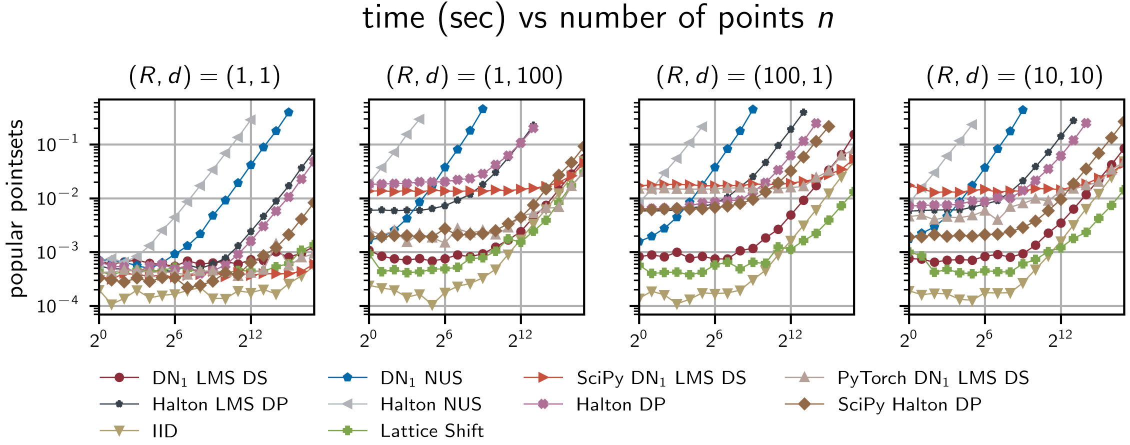


Randomizations of Low-Discrepancy (LD) Point Sets



Speed of Generating IID and Low-Discrepancy (LD) Points

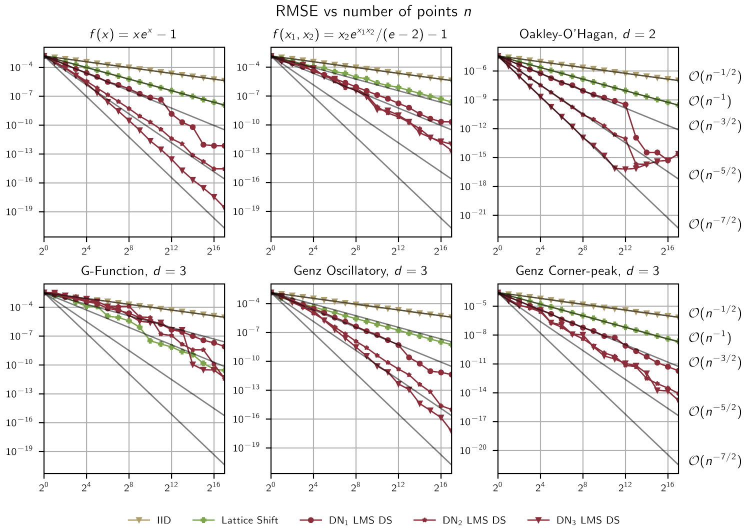
Randomized LD points are as fast to generate as IID points



R randomizations of n points in d dimensions

Root Mean Squared Error (RMSE) of Randomized QMC

Higher-order digital nets achieve higher-order convergence for low-dimensional functions



Automatic Single-Level Stopping Criteria for Asian Option Pricing

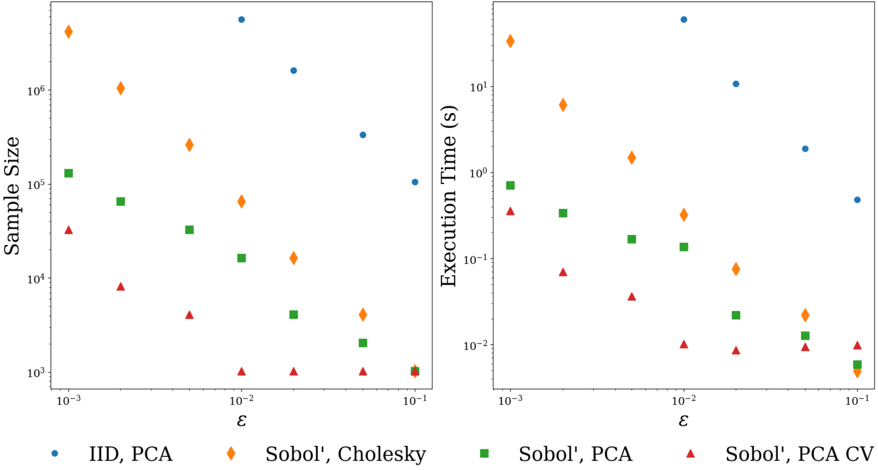


Figure: Using the principal component analysis (PCA) decomposition of the Brownian motion provides substantial savings compared to the Cholesky decomposition. Control variates provide additional savings.

Comparison of Automatic Stopping Criteria in QMCPy

QMCPy Class	Point Sets	Guaranteed	Vectorized	Multilevel	References
CubMCCLT	IID				[46]
CubMCCLTVec	IID		✓		[46]
CubMCG	IID	✓			[46]
CubQMCRepStudentT	LD		✓		[71]
CubQMCNetG	DigitalNetB2	✓	✓		[45]
CubQMCLatticeG	Lattice	✓	✓		[52]
CubQMCBayesNetG	DigitalNetB2	✓	✓		[97]
CubQMCBayesLatticeG	Lattice	✓	✓		[96]
CubMLMC	IID			✓	[37]
CubMLMCCont	IID			✓	[19]
CubMLQMC	LD			✓	[38]
CubMLQMCCont	LD			✓	[99]

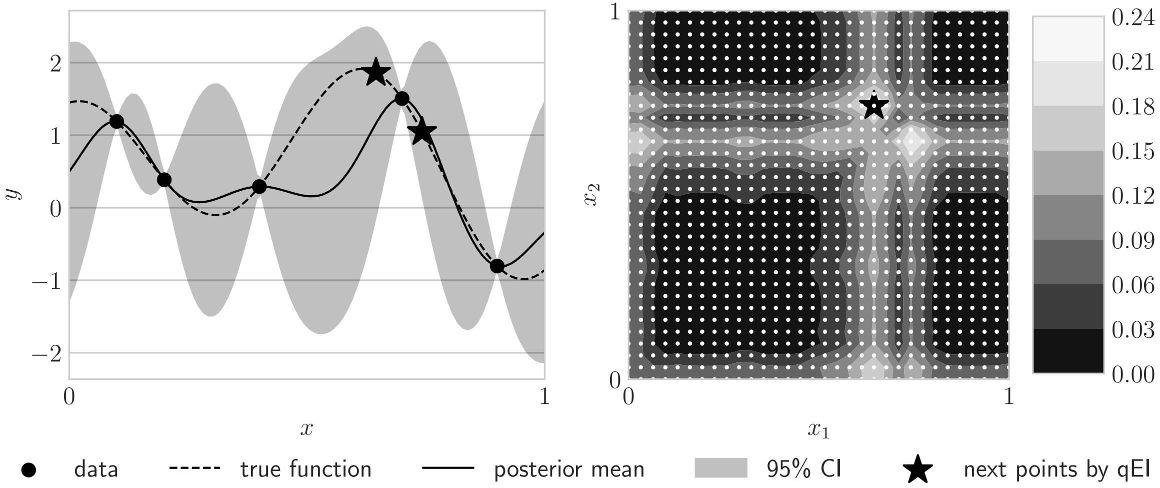
New (Q)MC Stopping Criteria for Functions of Multiple Expectations

Vectorized (Q)MC algorithms propagate bounds using interval arithmetic [110]

$s = C(\boldsymbol{\mu})$	$s^- = C^-(\boldsymbol{\mu}^-, \boldsymbol{\mu}^+)$	$s^+ = C^+(\boldsymbol{\mu}^-, \boldsymbol{\mu}^+)$
$\mu_1 + \mu_2$	$\mu_1^- + \mu_2^-$	$\mu_1^+ + \mu_2^+$
$\mu_1 - \mu_2$	$\mu_1^- - \mu_2^+$	$\mu_1^+ - \mu_2^-$
$\mu_1 \cdot \mu_2$	$\min(\mu_1^-\mu_2^-, \mu_1^-\mu_2^+, \mu_1^+\mu_2^-, \mu_1^+\mu_2^+)$	$\max(\mu_1^-\mu_2^-, \mu_1^-\mu_2^+, \mu_1^+\mu_2^-, \mu_1^+\mu_2^+)$
μ_1 / μ_2	$\begin{cases} -\infty, & 0 \in [\mu_2^-, \mu_2^+] \\ \min\left(\frac{\mu_1^-}{\mu_2^-}, \frac{\mu_1^+}{\mu_2^-}, \frac{\mu_1^-}{\mu_2^+}, \frac{\mu_1^+}{\mu_2^+}\right), & 0 \notin [\mu_2^-, \mu_2^+] \end{cases}$	$\begin{cases} \infty, & 0 \in [\mu_2^-, \mu_2^+] \\ \max\left(\frac{\mu_1^-}{\mu_2^-}, \frac{\mu_1^+}{\mu_2^-}, \frac{\mu_1^-}{\mu_2^+}, \frac{\mu_1^+}{\mu_2^+}\right), & 0 \notin [\mu_2^-, \mu_2^+] \end{cases}$
$\min(\mu_1, \mu_2)$	$\min(\mu_1^-, \mu_2^-)$	$\min(\mu_1^+, \mu_2^+)$
$\max(\mu_1, \mu_2)$	$\max(\mu_1^-, \mu_2^-)$	$\max(\mu_1^+, \mu_2^+)$

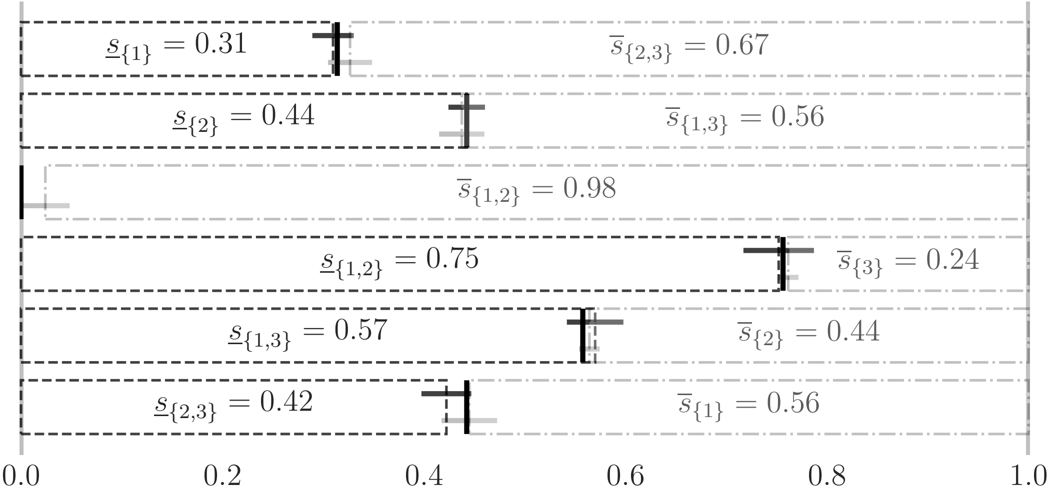
Bayesian Optimization with a Vectorized Acquisition Function

The utility of sampling at each grid point on the right is an expectation to be approximated with QMC



Approximating Sensitivity for the Ishigami function

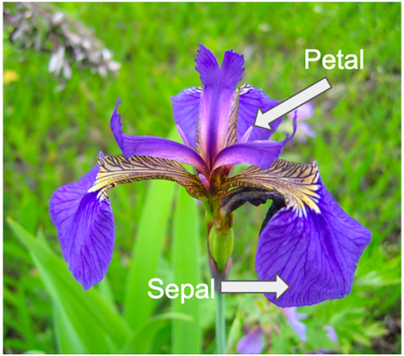
Each sensitivity index may be written as a ratio involving three expectations



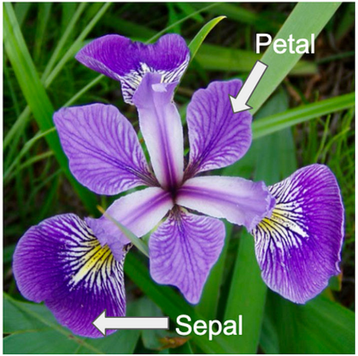
Iris Species Classification by Attributes

Which iris flower attributes are most important to determining its species (setosa, versicolor, or virginica)?

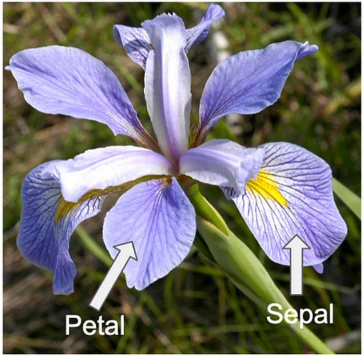
Iris setosa



Iris versicolor



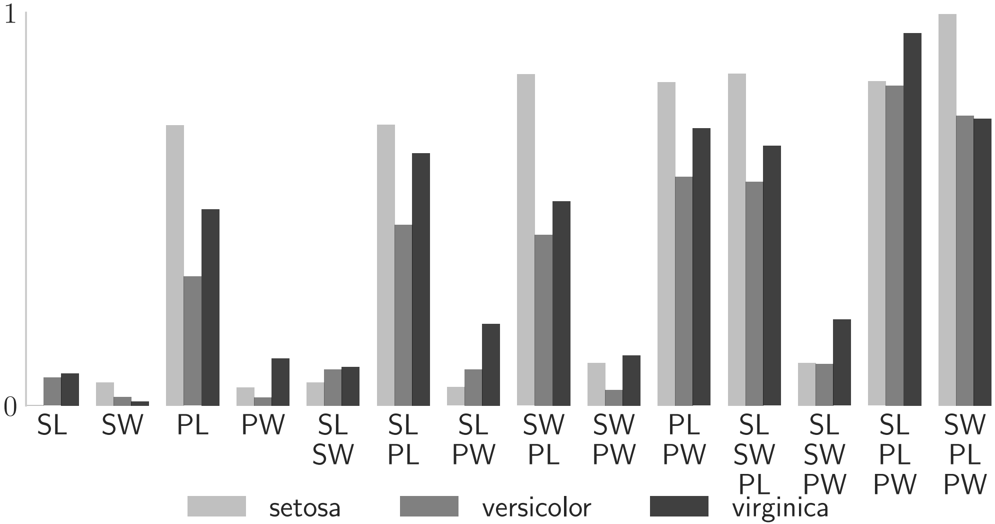
Iris virginica



Available attributes: petal length (PL), petal width (PW), sepal length (SL), sepal width (SW)

Sensitivity Indices for a Neural Network

Quantify the importance of subsets of inputs to classifying Iris species



Gaussian Processes Formulation

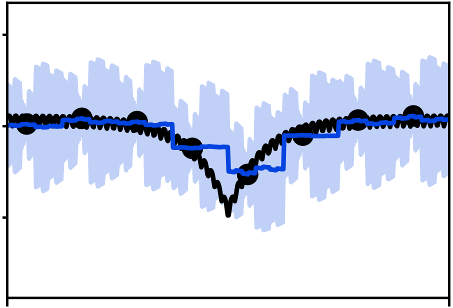
Flexible interpolation models with built-in UQ [94, 31]

$f \sim \mathrm{GP}(0,K)$ with posterior mean and covariance

$$\mathbb{E}[f(\boldsymbol{x})|\boldsymbol{f}] = \boldsymbol{K}(\boldsymbol{x})\mathsf{K}^{-1}\boldsymbol{f}$$

$$\mathbb{V}[f(\boldsymbol{x})|\boldsymbol{f}] = K(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{K}(\boldsymbol{x})^T\mathsf{K}^{-1}\boldsymbol{K}(\boldsymbol{x})$$

- SPD $K:[0,1]^d \times [0,1]^d \to \mathbb{R}$
- Sampling locations $\mathsf{X} = \{\boldsymbol{x}_i\}_{i=0}^{n-1}$
- Sample values $\boldsymbol{f} = \{f(\boldsymbol{x}_i)\}_{i=0}^{n-1}$
- Kernel vector $\boldsymbol{K}(\boldsymbol{x}) = \{K(\boldsymbol{x},\boldsymbol{x}_i)\}_{i=0}^{n-1}$
- Gram matrix $\mathsf{K} = \{K(\boldsymbol{x}_i,\boldsymbol{x}_{i'})\}_{i,i'=0}^{n-1}$



Typically requires $\mathcal{O}(n^2)$ storage and $\mathcal{O}(n^3+n^2d)$ computations

Fast GPs Pairing LD Points with Special Kernels

Fast GPs Pairing LD Points with Special Kernels

1. **LD Lattices + Shift-Invariant (SI) Kernels**
   - Give circulant Gram matrices $\mathsf{K} = \{K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})\}_{i,i'=0}^{n-1}$

   $\therefore$ Eigendecomp $\mathsf{K} = \mathsf{V}\Lambda\overline{\mathsf{V}}$ where $\overline{\mathsf{V}}$ is the DFT matrix $\to$ FFT in $\mathcal{O}(n \log n)$

Fast GPs Pairing LD Points with Special Kernels

1. **LD Lattices + Shift-Invariant (SI) Kernels**
   - Give circulant Gram matrices $\mathsf{K} = \{K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})\}_{i,i'=0}^{n-1}$

   $\therefore$ Eigendecomp $\mathsf{K} = \mathsf{V}\Lambda\overline{\mathsf{V}}$ where $\overline{\mathsf{V}}$ is the DFT matrix $\to$ FFT in $\mathcal{O}(n \log n)$
2. **LD Digital Nets + Digitally-Shift-Invariant (DSI) Kernels**
   - Give Recursive Symmetric Block Toeplitz (RSBT) Gram matrices K

   $\therefore$ Eigendecomp $\mathsf{K} = \mathsf{V}\Lambda\overline{\mathsf{V}}$ where $\overline{\mathsf{V}}$ is the Hadamard matrix $\to$ FWHT in $\mathcal{O}(n \log n)$

Fast GPs Pairing LD Points with Special Kernels

1. **LD Lattices + Shift-Invariant (SI) Kernels**
   - Give circulant Gram matrices $\mathsf{K} = \{K(\boldsymbol{x}_i, \boldsymbol{x}_{i'})\}_{i,i'=0}^{n-1}$

   $\therefore$ Eigendecomp $\mathsf{K} = \mathsf{V}\Lambda\overline{\mathsf{V}}$ where $\overline{\mathsf{V}}$ is the DFT matrix $\to$ FFT in $\mathcal{O}(n \log n)$
2. **LD Digital Nets + Digitally-Shift-Invariant (DSI) Kernels**
   - Give Recursive Symmetric Block Toeplitz (RSBT) Gram matrices K

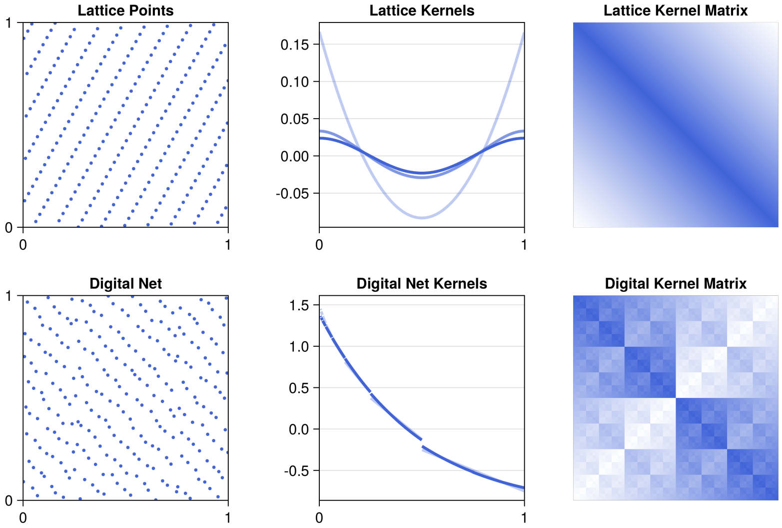
   $\therefore$ Eigendecomp $\mathsf{K} = \mathsf{V}\Lambda\overline{\mathsf{V}}$ where $\overline{\mathsf{V}}$ is the Hadamard matrix $\to$ FWHT in $\mathcal{O}(n \log n)$

Let $\boldsymbol{v}_1 = \mathbf{1}/\sqrt{n}$ and $\boldsymbol{k}_1$ be the first columns of $\overline{\boldsymbol{V}}$ and K respectively:

$$\boldsymbol{\lambda} := \Lambda \mathbf{1} = \sqrt{n}\Lambda\overline{\boldsymbol{v}}_1 = \sqrt{n}\overline{\mathsf{V}}\mathsf{V}\Lambda\overline{\boldsymbol{v}}_1 = \sqrt{n}\overline{\mathsf{V}}\boldsymbol{k}_1$$

- $\mathsf{K}\boldsymbol{a}$, $\mathsf{K}^{-1}\boldsymbol{a}$, and $|\mathsf{K}|$ can all be computed in $\mathcal{O}(n \log n + nd)$ computations
- Only requires evaluating and storing the first column of K

Lattices + SI K = Circulant K and Digital Nets + DSI K = RSBT K



Standard GPs vs Fast GPs using LD points and (D)SI Kernels

Comparison of methods and requirements for both storage and computations

$\{\boldsymbol{x}_i\}_{i=0}^{n-1}$ points	K structure	factor K method	K storage	form K cost
any	general SPD	Cholesky	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2 d)$
rank-1 lattice	SPD SI	(I)FFTBR	$\mathcal{O}(n)$	$\mathcal{O}(nd)$
base 2 digital net	SPD DSI	FWHT	$\mathcal{O}(n)$	$\mathcal{O}(nd)$

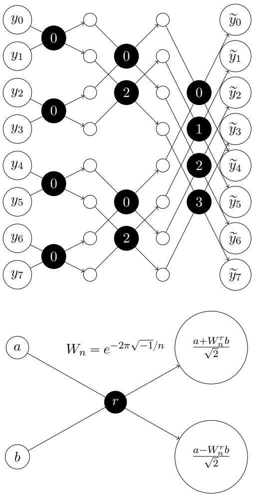
$\{\boldsymbol{x}_i\}_{i=0}^{n-1}$	factor K cost	K$\boldsymbol{y}$ cost	K$^{-1}\boldsymbol{y}$ cost	|K| cost
any	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
rank-1 lattice	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\log n)$
base 2 digital net	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\log n)$

Table: When evaluating (forming) K, we assume the cost of evaluating the kernel is $\mathcal{O}(d)$. Factorization of the SPD Gram matrix K is the cost of computing the eigendecomposition or Cholesky decomposition. The costs of matrix-vector multiplication K$\boldsymbol{y}$, solving a linear system K$^{-1}\boldsymbol{y}$ (when K is symmetric positive definite), and computing the determinant $|$K$|$ are the costs after performing the decomposition.

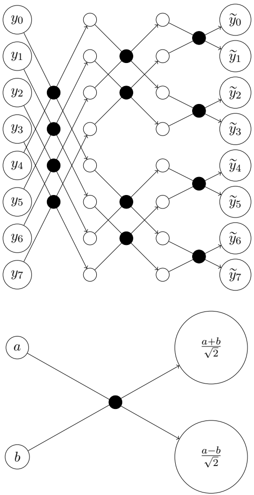
Fast Transforms for Decomposing Fast GP Gram Matrices

Fast Fourier Transform with bit-reversal (FFTBR) and the Fast Walsh–Hadamard Transform (FWHT)

FFTBR



FWHT



Product Kernels that are Shift-Invariant (SI) and Periodic

Product kernel

$$K(\boldsymbol{x},\boldsymbol{x}') = \gamma \prod_{j=1}^d \left(1+\eta_j K_{\alpha_j}(x_j,x_j')\right) \qquad \forall \boldsymbol{x},\boldsymbol{x}' \in [0,1)^d.$$

For smoothness parameters $\boldsymbol{\alpha} \in \mathbb{R}^d_{>1/2}$, our univariate SI kernels take the form

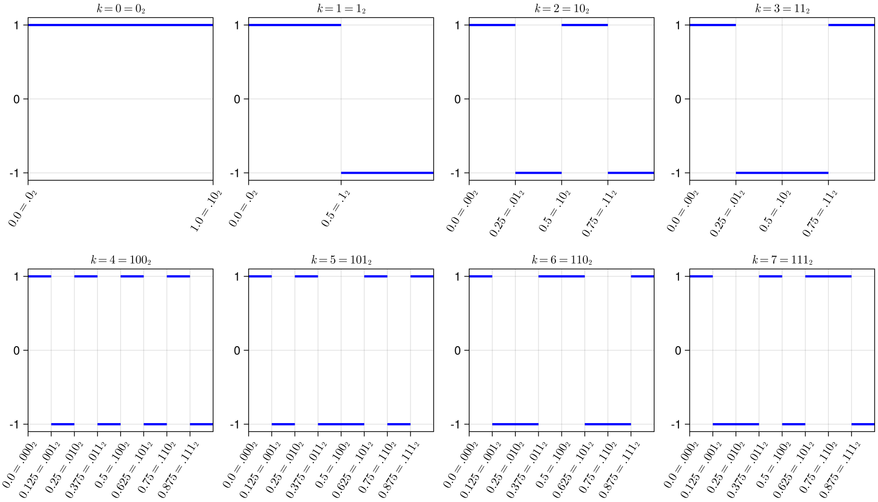
$$K_\alpha(x,x') = \frac{(-1)^{\alpha+1}(2\pi)^{2\alpha}}{(2\alpha)!} B_{2\alpha}((x-x') \mod 1)$$

with B_p is the p^{th} Bernoulli polynomial. May be written in terms of a decaying Fourier series. The inner product of the univariate SI kernel's RKHS is

$$\langle f,g\rangle_{K_\alpha} \propto \int_0^1 f^{(\alpha)}(x) g^{(\alpha)}(x) \mathrm{d}x.$$

Orthonormal Walsh Function Basis for L_2

Digitally-shift-invariant kernels are written in terms of Walsh Functions wal_k for $k \in \mathbb{N}_0$



New Higher-Order Digitally-Shift-Invariant Product Kernels in Base $b=2$

DSI kernels are *not* periodic, but they are discontinuous; yet their RKHSs contain smooth functions

For $x,x' \in [0,1)$, let $\beta(x) := -\lfloor \log_2(x) \rfloor$ and for $\nu \in \mathbb{N}$ let $t_\nu(x) = 2^{-\nu\beta(x)}$.
Write $x \oplus x' := \sum_{k \in \mathbb{N}} ((\mathsf{x}_k + \mathsf{x}'_k) \mod 2) 2^{-k}$, the XOR of $x = \sum_{k\in\mathbb{N}} \mathsf{x}_k 2^{-k}$ and $x' = \sum_{k\in\mathbb{N}} \mathsf{x}'_k 2^{-k}$.

$$K_\alpha(x,x') := \ddot{K}_{\alpha,0,0}(x \oplus x', 0)$$

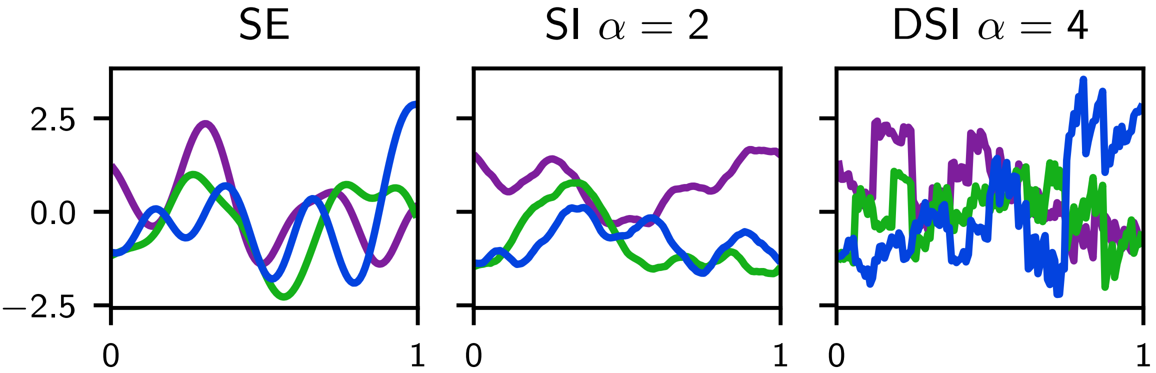
$$\ddot{K}_{\alpha,0,0}(x,0) = \begin{cases} -1 + -\beta(x)x + \frac{5}{2}\left[1 - t_1(x)\right], & \alpha = 2 \\ -1 + \beta(x)x^2 - 5\left[1-t_1(x)\right]x + \frac{43}{18}\left[1-t_2(x)\right], & \alpha = 3 \\ -1 - \frac{2}{3}\beta(x)x^3 + 5\left[1-t_1(x)\right]x^2 - \frac{43}{9}\left[1-t_2(x)\right]x & \\ \quad + \frac{701}{294}\left[1-t_3(x)\right] + \beta(x)\left[\frac{1}{48}\sum_{a=0}^\infty \frac{\mathrm{wal}_{2^a}(x)}{2^{3a}} - \frac{1}{42}\right], & \alpha = 4 \end{cases}.$$

For any $\alpha \in \mathbb{N}_{>2}$, one may compute $\ddot{K}_{\alpha,0,0}(x,x')$ using the algorithm in [4, Theorem 2].
The RKHSs of these kernels contain smooth functions. Specifically, $H(Q_\alpha) \subseteq H(K_\alpha)$ where

$$\langle f,g \rangle_{Q_\alpha} = \sum_{l=1}^{\alpha-1} \int_0^1 f^{(l)}(x)\mathrm{d}x \int_0^1 g^{(l)}(x)\mathrm{d}x + \int_0^1 f^{(\alpha)}(x) g^{(\alpha)}(x)\mathrm{d}x.$$

Prior Draws from GPs with Different Kernels

Squared exponential (SE) kernel, the $\alpha = 2$ SI kernel, and the $\alpha = 4$ DSI kernel



FastGPs Software

`pip install fastgps` or visit alegresor.github.io/fastgps/ [116]

A scalable Python software for fast GPs regression requiring only $\mathcal{O}(n \log n)$ computations

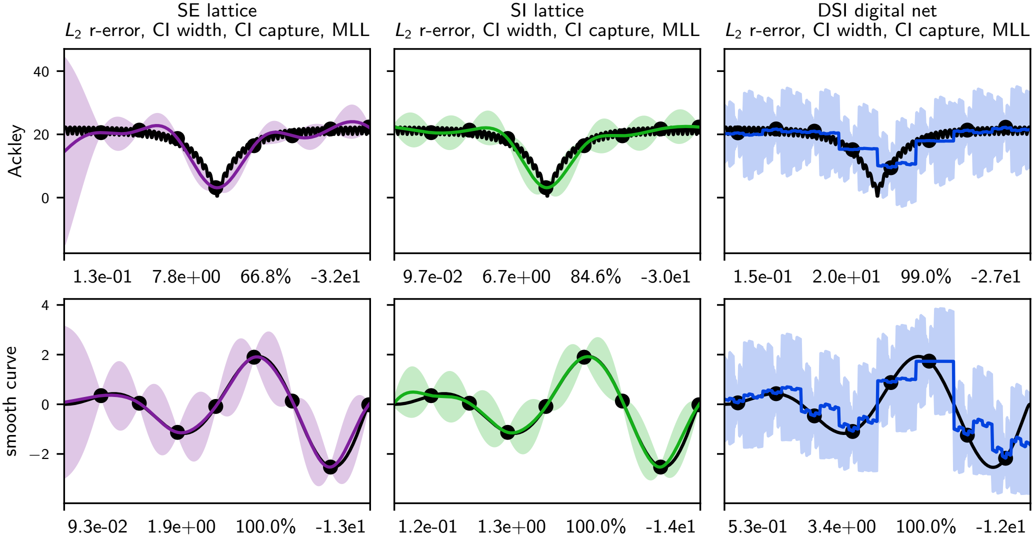
FastGPs Software

`pip install fastgps` or visit alegresor.github.io/fastgps/ [116]

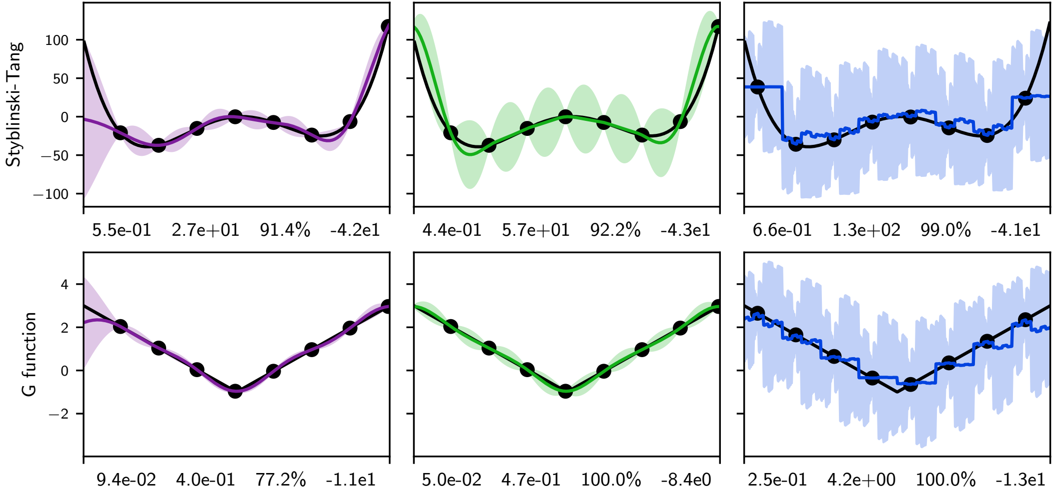
A scalable Python software for fast GPs regression requiring only $\mathcal{O}(n \log n)$ computations

1. Kernel hyperparameter optimization of marginal log likelihood (MLL) or generalized cross validation (GCV) loss
2. Fast Bayesian cubature for uncertainty quantification in Quasi-Monte Carlo
3. Fast multitask GPs with support for different sample sizes for each task; potentially useful for multi-fidelity simulations and Multilevel Monte Carlo
4. Batched GPs for simultaneously modeling vector-output simulations
5. GPU support enabled by the `PyTorch` stack
6. Flexible LD sequences and SI/DSI kernels with various LD randomizations and kernels of higher-order smoothness using the implementation in `QMCPy`
7. Fast Derivative Informed GPs for simulations coupled with automatic differentiation
8. Efficient variance projections for non-greedy Bayesian optimization in multilevel methods

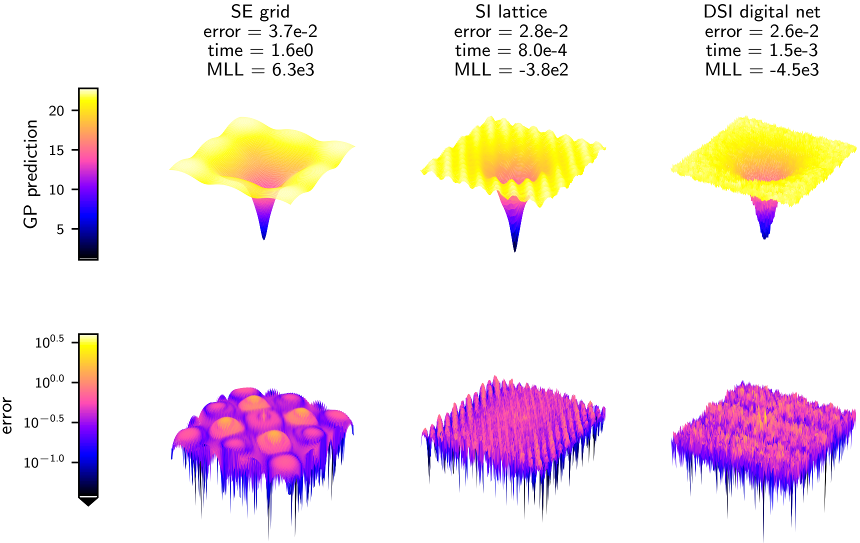
Fast GPR Examples in One Dimension [117], $n = 8$



Fast GPR Examples in One Dimension [117], $n = 8$



Fast GPR for the Ackley Function in Two Dimensions [117], $n = 4096$



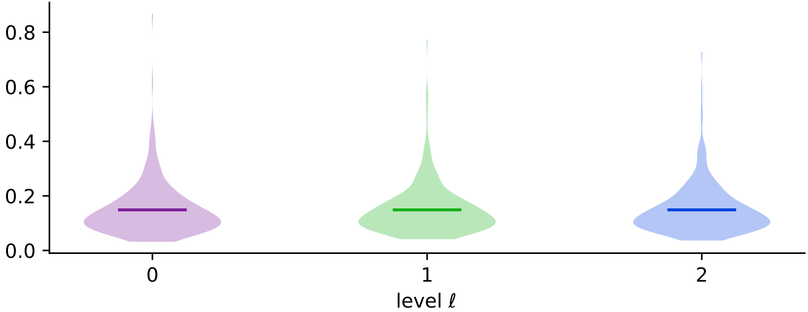
Multilevel (Multitask) Modeling

Given a multilevel simulation

$$f : \{1,\dots,L\} \times [0,1]^d \to \mathbb{R},$$

we want to model $f(L,\cdot)$, the true (maximum-fidelity) simulation.

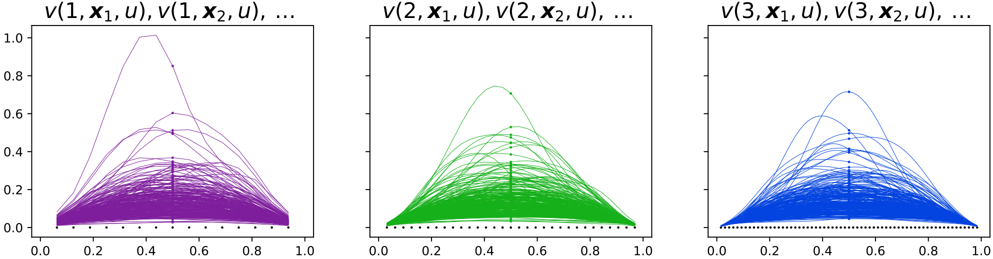
- $f(\ell,\boldsymbol{x})$ simulates at level $\ell \in \{1,\dots,L\}$ and with parameters $\boldsymbol{x} \in [0,1]^d$
- Cost C_ℓ typically greater on higher levels
- $f(1,\cdot), f(2,\cdot),\dots,f(L,\cdot)$ are typically highly correlated



Numerical Solutions of PDEs with Random Coefficients

$$f(\ell, \boldsymbol{x}) = \mathcal{F}(v(\ell, \boldsymbol{x}, \cdot))$$

- $v : \{1, \dots, L\} \times [0,1]^d \times \Omega$ is the numerical solution to the PDE
- $\boldsymbol{x}$ represent random coefficients, e.g. coefficients in a Karhunen–Loève expansion
- ℓ controls the fidelity of the numerical solver e.g. the mesh width is $2^{-\ell}$
- $\mathcal{F}$ is a (possibly non-linear) functional of the PDE solution, e.g.,
  - $\mathcal{F}(v(\ell, \boldsymbol{x}, \cdot)) = \mathbb{E}[v(\ell, \boldsymbol{x}, \boldsymbol{U})]$ where $\boldsymbol{U} \sim \mathcal{U}(\Omega)$, or
  - $\mathcal{F}(v(\ell, \boldsymbol{x}, \cdot)) = v(\ell, \boldsymbol{x}, u)$ for some $u \in \Omega$, e.g., $u = 1/2$ shown below



Fast Bayesian Multilevel Quasi-Monte Carlo Without Replications [115]

$$\mathbb{E}[f(L,\boldsymbol{X})] = \sum_{\ell=1}^{L} \mathbb{E}[\underbrace{f(\ell,\boldsymbol{X}) - f(\ell-1,\boldsymbol{X})}_{\Delta_\ell(\boldsymbol{X})}], \quad \boldsymbol{X} \sim \mathcal{U}[0,1]^d, \qquad f(0,\cdot) = 0$$

Fast Bayesian Multilevel Quasi-Monte Carlo Without Replications [115]

$$\mathbb{E}[f(L,\boldsymbol{X})] = \sum_{\ell=1}^{L} \mathbb{E}[\underbrace{f(\ell,\boldsymbol{X}) - f(\ell-1,\boldsymbol{X})}_{\Delta_\ell(\boldsymbol{X})}], \quad \boldsymbol{X} \sim \mathcal{U}[0,1]^d, \qquad f(0,\cdot) = 0$$

MLMC $\mathbb{E}[\Delta_\ell(\boldsymbol{X})] \approx \frac{1}{n_\ell} \sum_{i=0}^{n_\ell-1} \Delta_\ell(\boldsymbol{x}_i^\ell)$ for IID $\boldsymbol{x}_0^\ell, \dots, \boldsymbol{x}_{n_\ell-1}^\ell \sim \mathcal{U}[0,1]^d$

- $n_\ell \propto \sqrt{\mathbb{V}[\Delta_\ell(\boldsymbol{X})]/C_\ell}$ chosen to optimally minimize error [37]

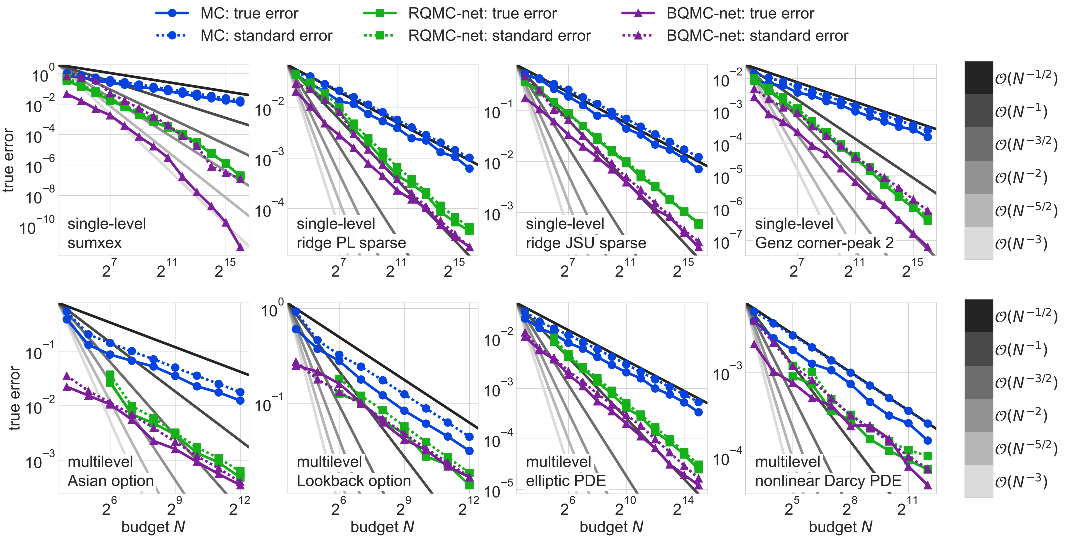
RMLQMC with replications $\mathbb{E}[\Delta_\ell(\boldsymbol{X})] \approx \frac{1}{R} \sum_{r=1}^{R} \frac{1}{n_\ell} \sum_{i=0}^{n_\ell-1} \Delta_\ell(\boldsymbol{x}_{ri}^\ell)$ [38]

- IID randomizations of low-discrepancy point $\{\boldsymbol{x}_{1i}^\ell\}_{i=0}^{n_\ell-1}, \dots, \{\boldsymbol{x}_{Ri}^\ell\}_{i=0}^{n_\ell-1}$.
- n_ℓ greedily doubled on level where $\mathbb{V}[\Delta_\ell(\boldsymbol{X})]/(n_\ell C_\ell)$ the largest
- Variance approximated by sample variance across R sub-means

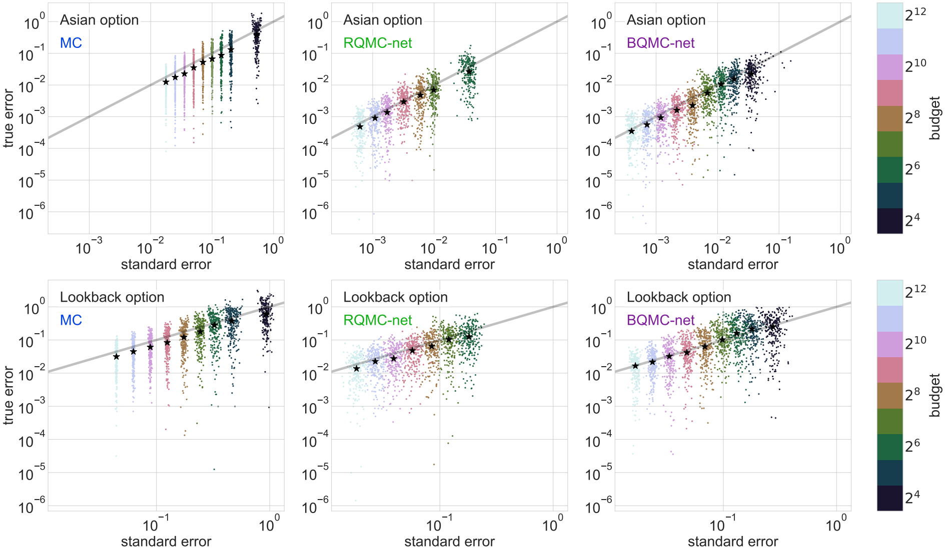
New BMLQMC Bayesian MLQMC without replications $\mathbb{E}[\Delta_\ell(\boldsymbol{X})] \approx \frac{1}{n_\ell} \sum_{i=0}^{n_\ell-1} \Delta_\ell(\boldsymbol{x}_i^\ell)$

- $\{\boldsymbol{x}_i^\ell\}_{i=0}^{n_\ell-1}$ a single randomized low-discrepancy point set
- n_ℓ greedily doubled on level with maximum variance reduction for the cost
- Variance taken to be posterior variance with $\Delta_\ell \sim \mathrm{GP}(\tau, K_\ell)$ (independent GPs)

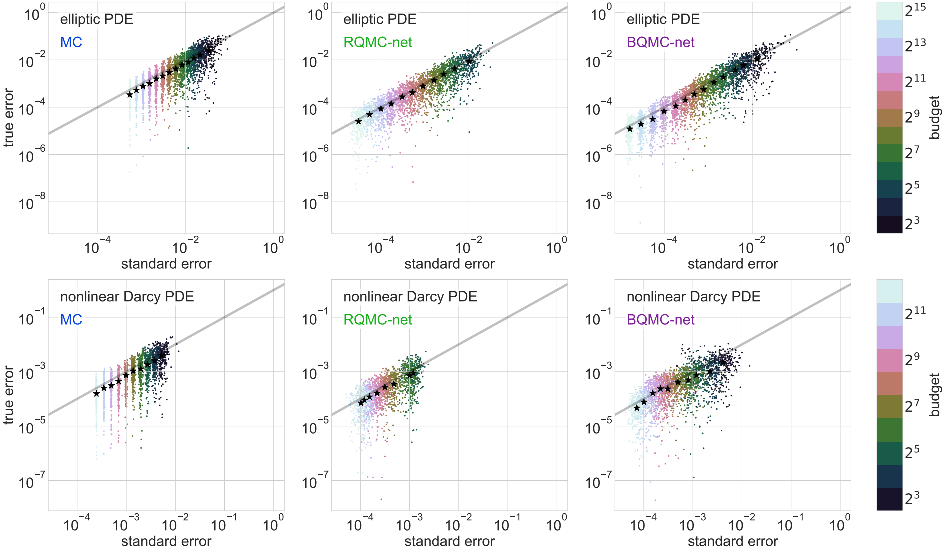
Convergence with Increasing Budgets: LD Digital Nets



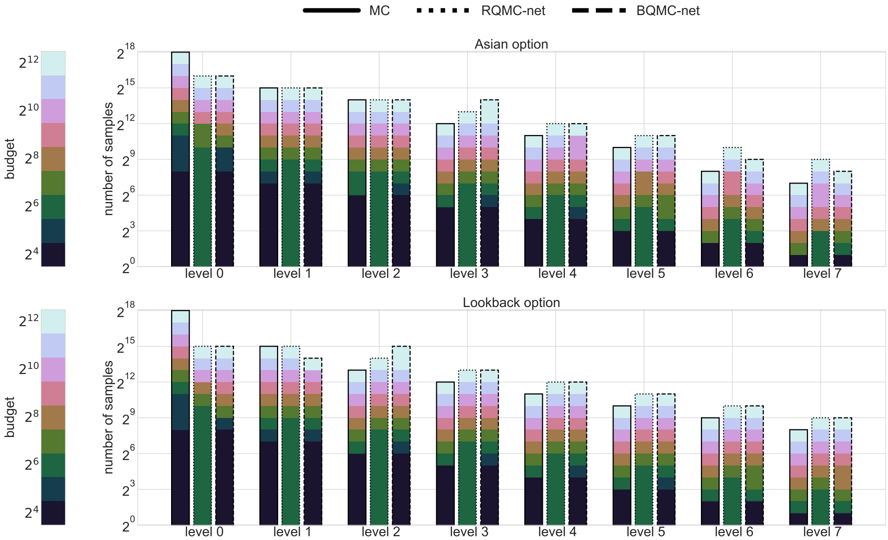
Multilevel Error Estimate Accuracy: LD Digital Nets



Multilevel Error Estimate Accuracy: LD Digital Nets



Multilevel Sample Allocation: LD Digital Nets



Multitask Gaussian Processes

$$f(\ell,\boldsymbol{x}) \sim \mathrm{GP}(0,K)$$

$N = n_1 + \cdots + n_L$ sampling locations $\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_L$ where $\mathcal{D}_\ell = \left\{\left(\ell,\boldsymbol{x}_i^\ell\right)\right\}_{i=0}^{n_\ell-1}$. Posterior mean and covariance

$$\mathbb{E}[f(\ell,\boldsymbol{x})] = \boldsymbol{K}^T(\ell,\boldsymbol{x})\mathsf{K}^{-1}\boldsymbol{f}$$
$$\mathbb{V}[f(\ell,\boldsymbol{x})] = K((\ell,\boldsymbol{x}),(\ell,\boldsymbol{x})) - \boldsymbol{K}^T(\ell',\boldsymbol{x}')\mathsf{K}^{-1}\boldsymbol{K}(\ell,\boldsymbol{x})$$

- $\boldsymbol{K}(\ell,\boldsymbol{x}) = K(\mathcal{D},(\ell,\boldsymbol{x}))$ and $\boldsymbol{f} = f(\mathcal{D})$ are length N vectors
- $\mathsf{K} = K(\mathcal{D},\mathcal{D}^T)$ is the $N \times N$ Gram matrix

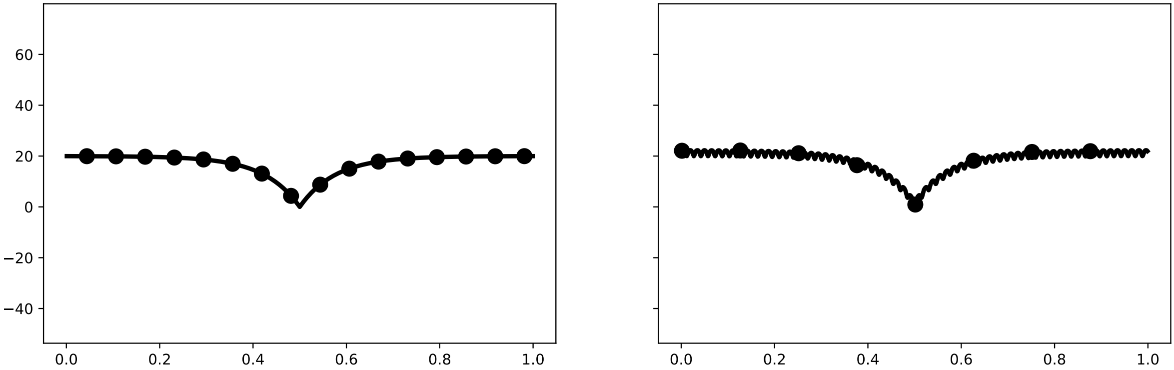
Kernel K depends on hyperparameters $\boldsymbol{\theta}$ e.g. global scale, lengthscale, etc.
Hyperparameters $\boldsymbol{\theta}$ often chosen to minimize negative marginal log likelihood (NMLL)

$$\mathrm{NMLL} \propto \boldsymbol{f}^T\mathsf{K}^{-1}\boldsymbol{f} + \log|\mathsf{K}| + \log(2\pi)N$$

$\therefore$ Multitask GPs require computing $\mathsf{K}^{-1}\boldsymbol{f}$ and $\log|\mathsf{K}|$ $\implies$ standard cost $\mathcal{O}(N^3)$

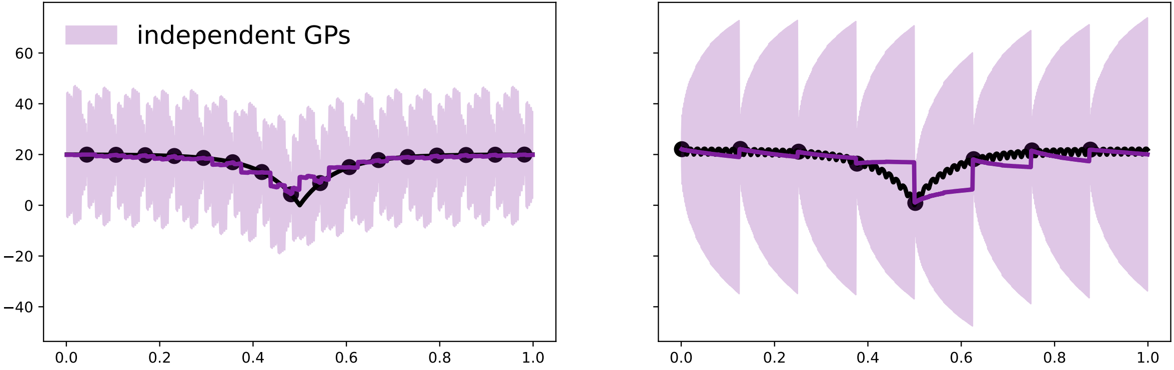
Independent GPs vs a Multitask GP (MTGP)

Low Fidelity Left, High Fidelity Right



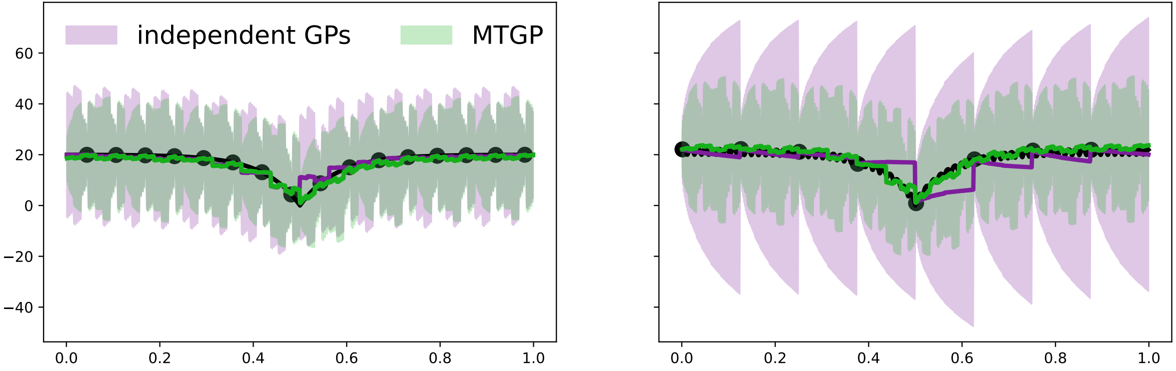
Independent GPs vs a Multitask GP (MTGP)

Low Fidelity Left, High Fidelity Right



Independent GPs vs a Multitask GP (MTGP)

Low Fidelity Left, High Fidelity Right



Product Kernels for Multitask GPs

Common to assume

$$K((\ell,\boldsymbol{x}),(\ell',\boldsymbol{x}')) = R(\ell,\ell')Q(\boldsymbol{x},\boldsymbol{x}')$$

- $R : \{1,\dots,L\} \times \{1,\dots,L\} \to \mathbb{R}$ an SPD kernel over levels e.g.

$$\mathsf{R} = \{R(\ell,\ell')\}_{\ell,\ell'=1}^L = \mathsf{B}\mathsf{B}^T + \mathrm{diag}(\boldsymbol{\nu}), \quad \boldsymbol{\nu} \in \mathbb{R}_+^L, \quad \mathsf{B} \in \mathbb{R}^{L\times r}, \quad \text{rank } r \leqslant L$$

- $Q : [0,1]^d \times [0,1]^d \to \mathbb{R}$ an SPD kernel over parameters

$$\mathsf{K} = \begin{pmatrix} \mathsf{K}_{11} & \cdots & \mathsf{K}_{1L} \\ \vdots & \ddots & \vdots \\ \overline{\mathsf{K}_{1L}} & \cdots & \mathsf{K}_{LL} \end{pmatrix} = \begin{pmatrix} R_{11}\mathsf{Q}_{11} & \cdots & R_{1L}\mathsf{Q}_{1L} \\ \vdots & \ddots & \vdots \\ \overline{R_{1L}\mathsf{Q}_{1L}} & \cdots & R_{LL}\mathsf{Q}_{LL} \end{pmatrix}$$

- $R_{\ell\ell'} = R(\ell,\ell')$ is a scalar
- $\mathsf{Q}_{\ell\ell'} = Q(\mathcal{D}_\ell, \mathcal{D}_{\ell'}^T)$ is an $n_\ell \times N_{\ell'}$ Gram matrix

New Fast Multitask GPs

$$\mathsf{K} = \begin{pmatrix} R_{11}\mathsf{Q}_{11} & \cdots & R_{1L}\mathsf{Q}_{1L} \\ \vdots & \ddots & \vdots \\ \overline{R_{1L}\mathsf{Q}_{1L}} & \cdots & R_{LL}\mathsf{Q}_{LL} \end{pmatrix}$$

Idea: Force "nice" structure in $\mathsf{Q}_{\ell\ell'}$ through special pairings of $\mathsf{X}_\ell = \{\boldsymbol{x}_i^\ell\}_{i=1}^{n_\ell}$ and Q

1. X_ℓ a lattice and Q a shift-invariant (SI) kernel $\implies$ $\mathsf{Q}_{\ell\ell'}$ block circulant
2. X_ℓ a (base 2) digital net and Q a digitally-SI (DSI) kernel $\implies$ $\mathsf{Q}_{\ell\ell}$ block RSBT

Technicalities

- Lattices and digital nets require sample sizes $n_\ell = 2^{m_\ell}$
- Lattices $\mathsf{X}_1,\dots,\mathsf{X}_L$: same generating vector, possibly different random shifts
- Circulant matrices diagonalizable by FFT
- Digital nets $\mathsf{X}_1,\dots,\mathsf{X}_L$: same generating matrices, possibly different digital shifts
- RSBT matrices diagonalizable by FWHT

Fast Multitask GPs Continued

$$\mathsf{K}_{\ell\ell'} = R_{\ell\ell'}\mathsf{Q}_{\ell\ell'} = \mathsf{V}_{m_\ell}\Sigma_{\ell\ell'}\overline{\mathsf{V}_{m_{\ell'}}}$$

- $\overline{\mathsf{V}_m}$ a $2^m \times 2^m$ *fast transform matrix*
  1. Lattice X_ℓ with SI Q makes $\overline{\mathsf{V}_{m_\ell}}$ the Fast Fourier Transform
  2. Digital Net X_ℓ with DSI Q makes $\overline{\mathsf{V}_{m_\ell}}$ the Fast Walsh Hadamard Transform
- $\mathsf{V}_m\boldsymbol{a}$ and $\overline{\mathsf{V}_m}\boldsymbol{a}$ both cost only $\mathcal{O}(m2^m)$ to compute
- The first column of $\overline{\mathsf{V}_m}$ is $\mathbf{1}_m/\sqrt{2^m}$
- $\Sigma_{\ell\ell'}$ a diagonal block matrix characterized by

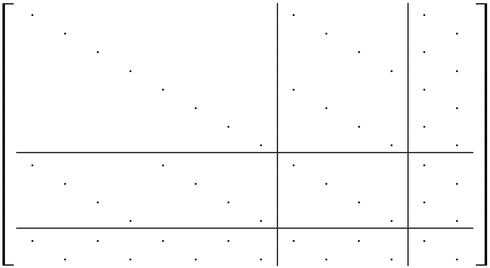
$$\boldsymbol{\sigma}_{\ell\ell'} = \Sigma_{\ell\ell'}\mathbf{1}_{m_{\ell'}} = \sqrt{2^{m_{\ell'}}}\,\overline{\mathsf{V}_{m_\ell}}\boldsymbol{k}_{\ell\ell',1}$$

where $\boldsymbol{k}_{\ell\ell',1}$ is the first column of $\mathsf{K}_{\ell\ell'}$, and we assume $m_\ell \geqslant m_{\ell'}$

Fast Multitask GPs Gram Matrix Structure

$$\mathsf{K} = \begin{pmatrix} \mathsf{V}_{m_1} & & \\ & \ddots & \\ & & \mathsf{V}_{m_L} \end{pmatrix} \begin{pmatrix} \Sigma_{11} & \cdots & \Sigma_{1L} \\ \vdots & \ddots & \vdots \\ \overline{\Sigma_{1L}} & \cdots & \Sigma_{LL} \end{pmatrix} \begin{pmatrix} \overline{\mathsf{V}_{m_1}} & & \\ & \ddots & \\ & & \overline{\mathsf{V}_{m_L}} \end{pmatrix} =: \mathsf{V}\Sigma\overline{\mathsf{V}}$$

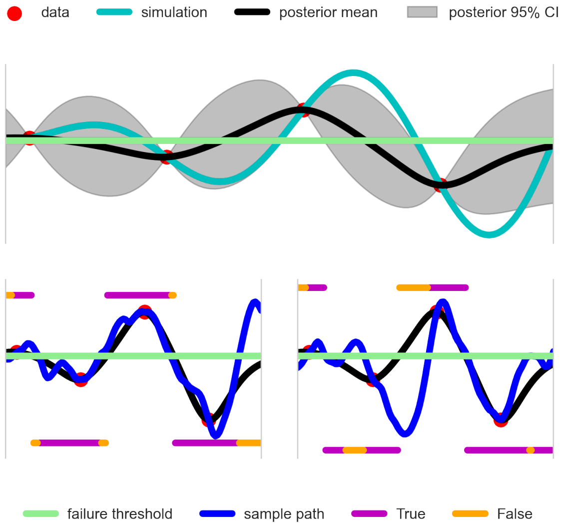
For example, if $n_1 = 8$, $n_2 = 4$, and $n_3 = 2$ then Σ has the following structure

$\Sigma =$ 

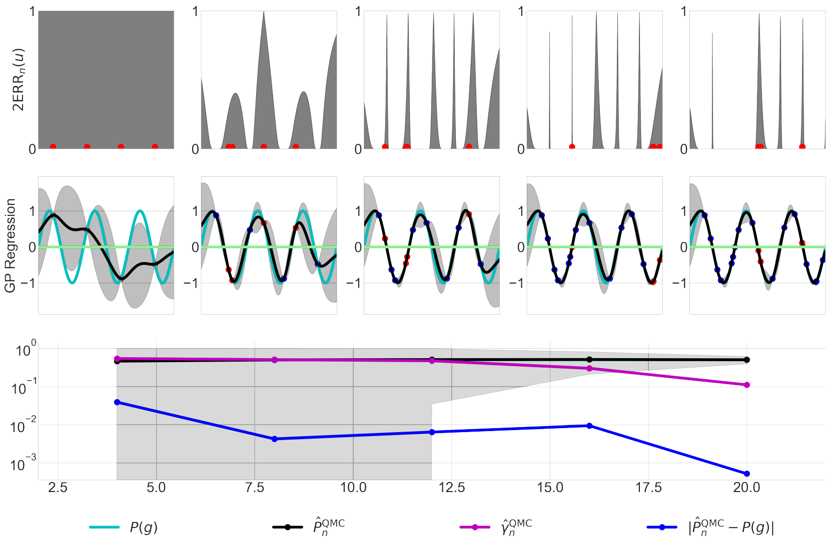
Thesis contains algorithm for efficiently computing $\Sigma^{-1}\widehat{f}$ and $\log\lvert\Sigma\rvert$ with cost & storage analysis

New Gaussian Processes for Reliability Analysis [109]

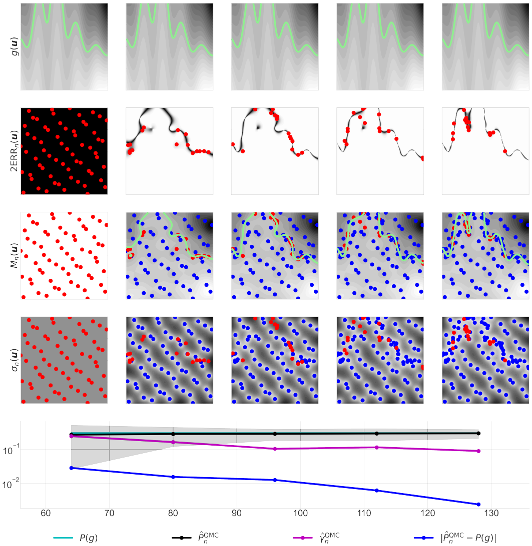
Guaranteed credible intervals for probability of failure estimation with GPs



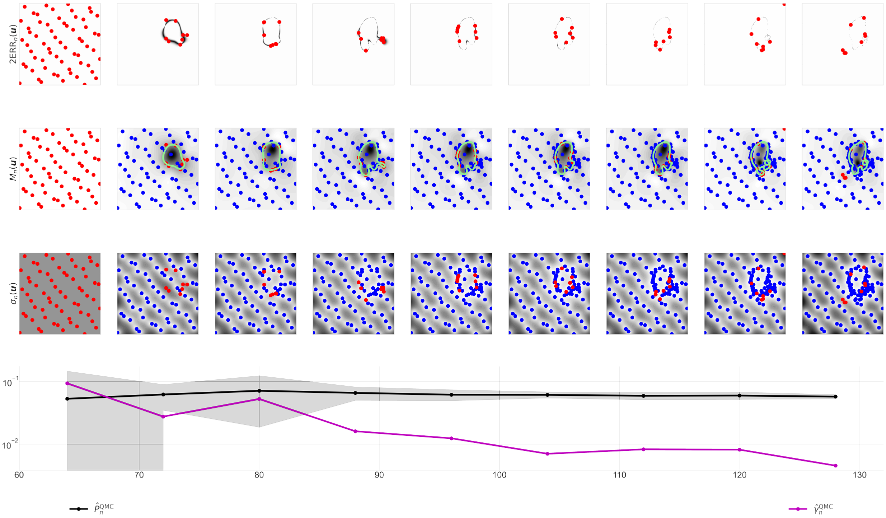
Sine Function in $d = 1$ Dimension



Multimodal Function in $d = 2$ Dimensions

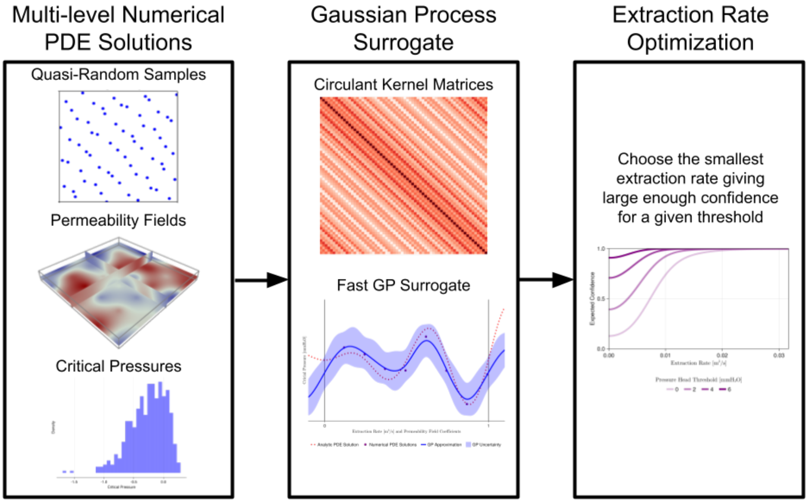


Tsunami Model from UM-Bridge [104] in $d=2$ Dimensions

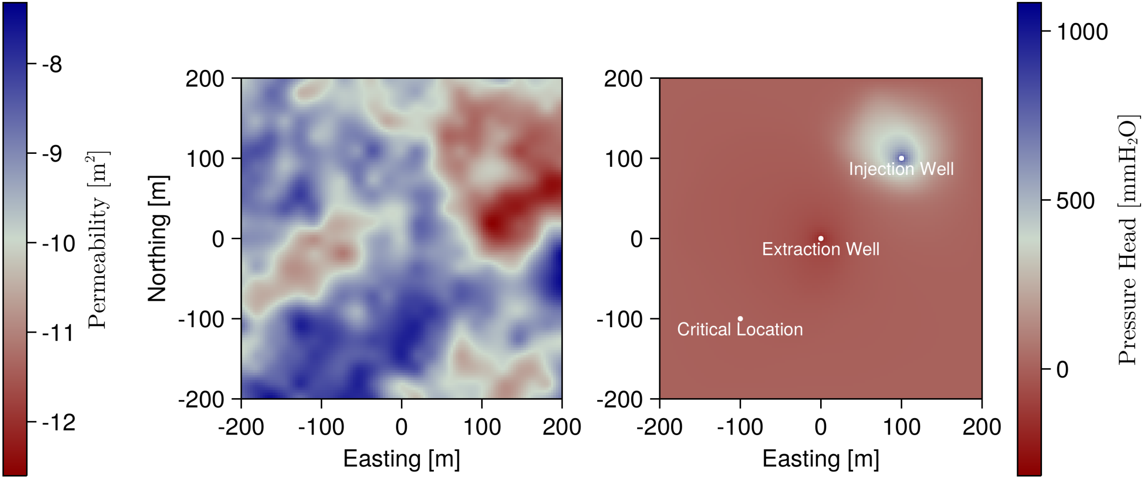


New Fast Gaussian Processes (GPs) for the Darcy Equation [114]

Error-aware multilevel GPs for modeling subsurface flow through porous media

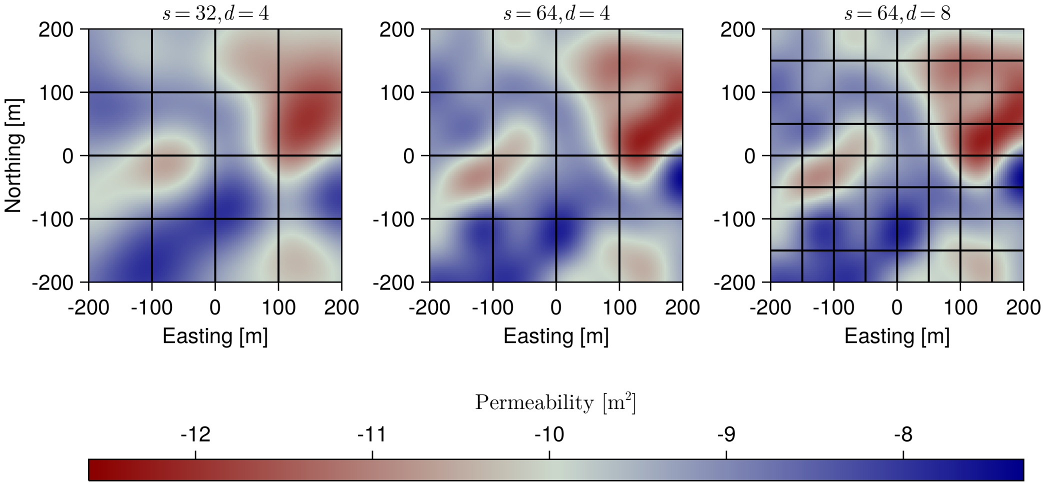


Permeability Field and Pressure Field from Darcy's Equation



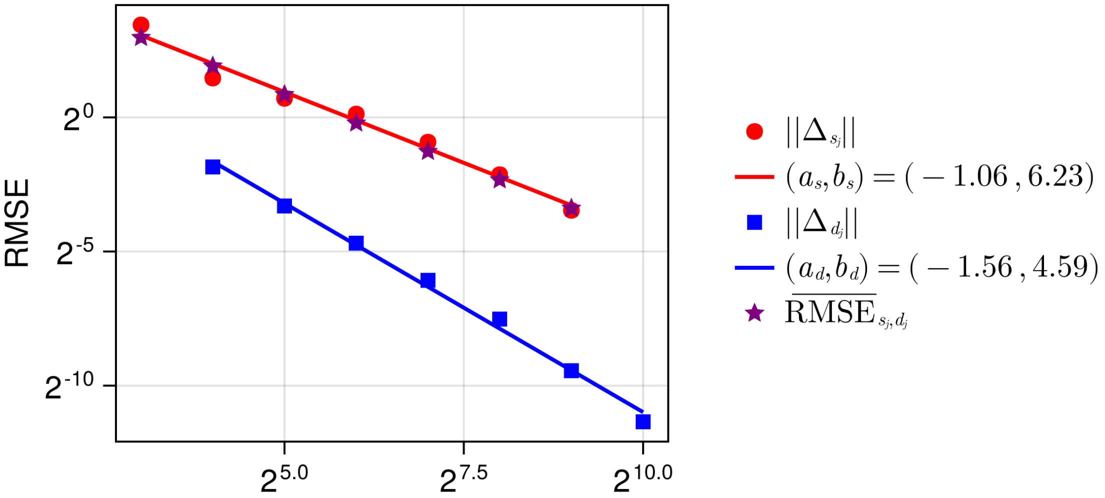
Multi-Resolution Permeability Field Decomposition

Discretized in both the number of KL terms s and the number of mesh points d



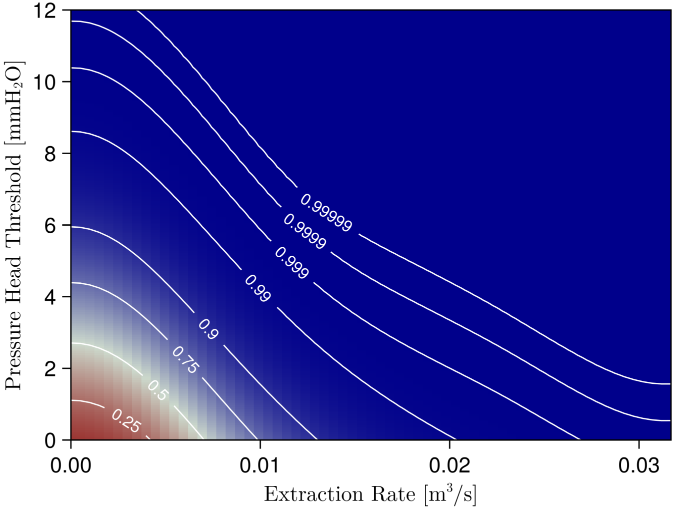
Tracking Convergence of Both Discretization Errors

Two-dimensional subsurface left and three-dimensional subsurface right



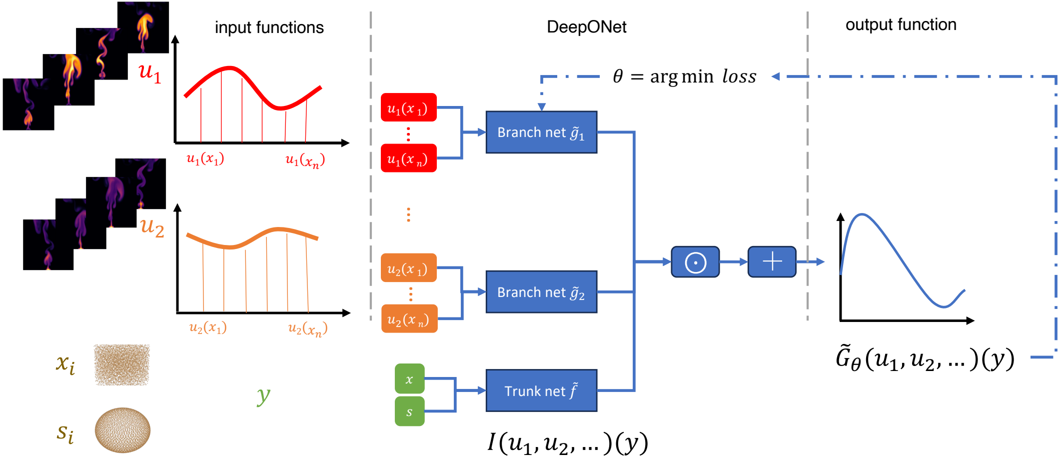
Confidence Pressure Remains Below Critical Threshold

Two-dimensional subsurface left and three-dimensional subsurface right



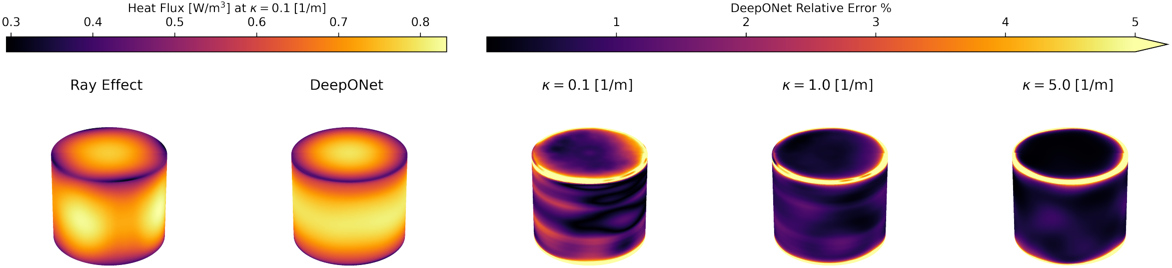
New Neural Radiative Transfer [113]

Using Deep Operator Network scientific machine learning models to accelerate radiative transfer simulations



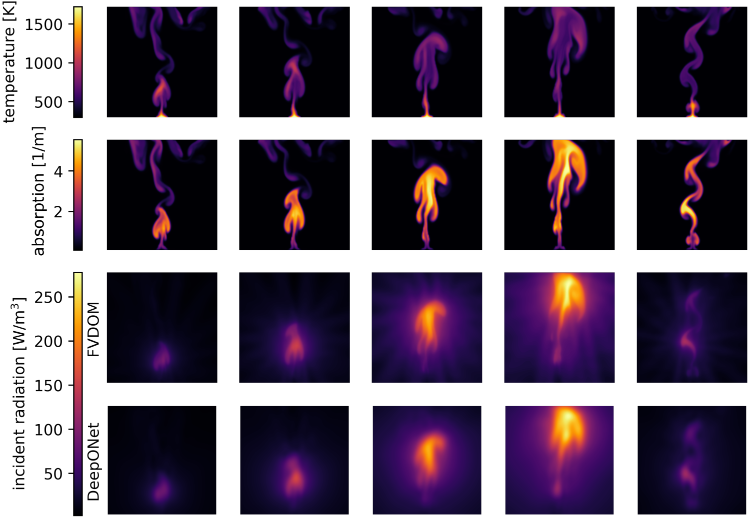
Modeling Gray Gas in a Cylinder

DeepONets honor radiative transfer physics and overcome the ray effect in traditional numerical PDE solvers



Modeling a Two-Dimensional Pool Fire

Compared against reference finite volume discrete-ordinate method (FVDOM) in `FireFOAM`



New SciML for Exact Recovery of Nonlinear PDEs [2]

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

Newton–Kantorovich solution trajectories of PDE solutions $\{v_n(c)\}_{n=0}^S$ for random coefficients c

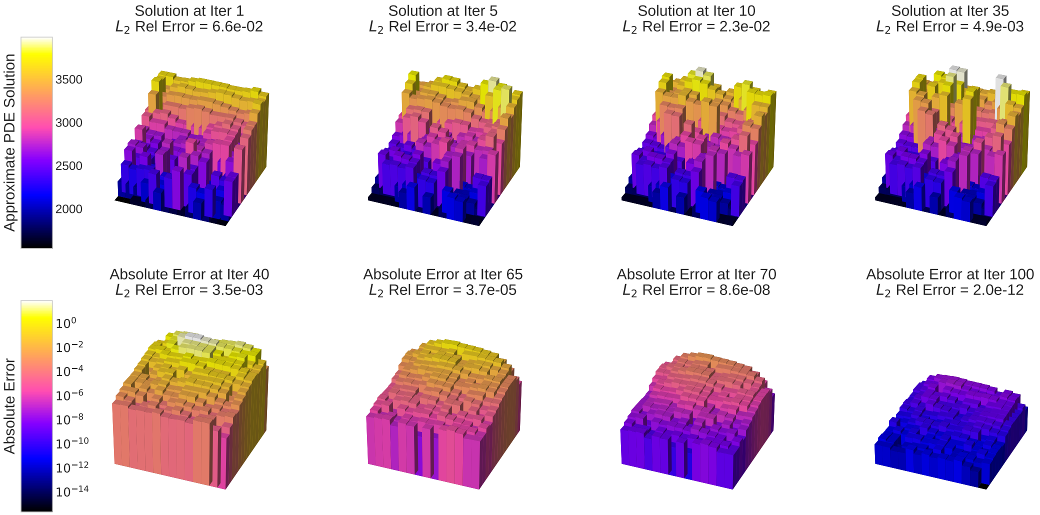
$$v_{n+1}(c) = v_n(c) - \alpha H^{-1}(v,c,\lambda)\frac{\partial F^T}{\partial v}(v,c)F(v,c)$$

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

	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 $\mathcal{O}(10^{-16})$

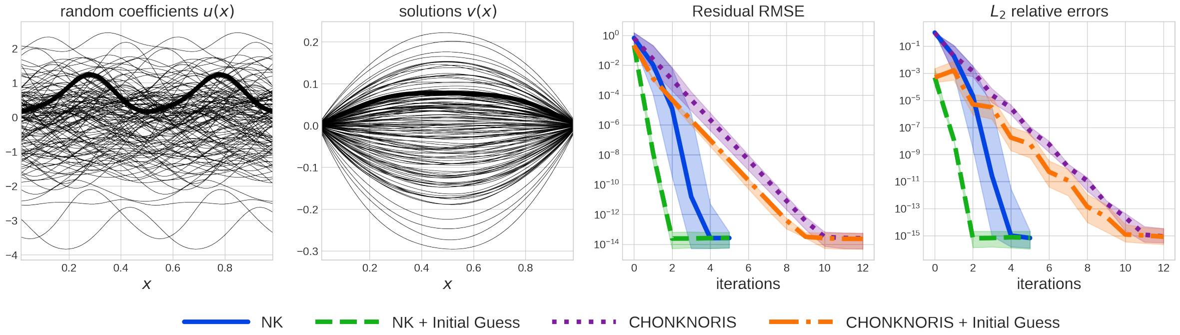
Newton–Kantorovich (NK) Iteration for Seismic Imaging

Finds subsurface velocity from a surface acoustic signal [120, 21]



Nonlinear Elliptic PDE

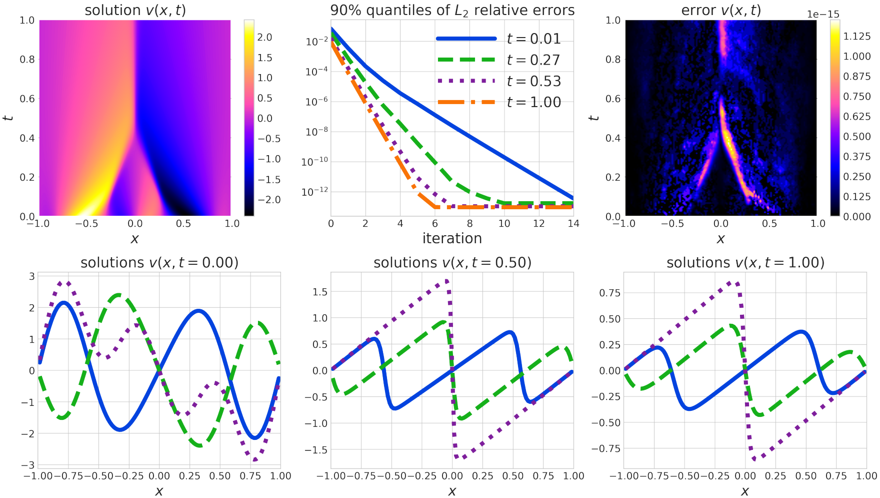
CHONKNORIS achieves machine precision recovery in 12 iterations



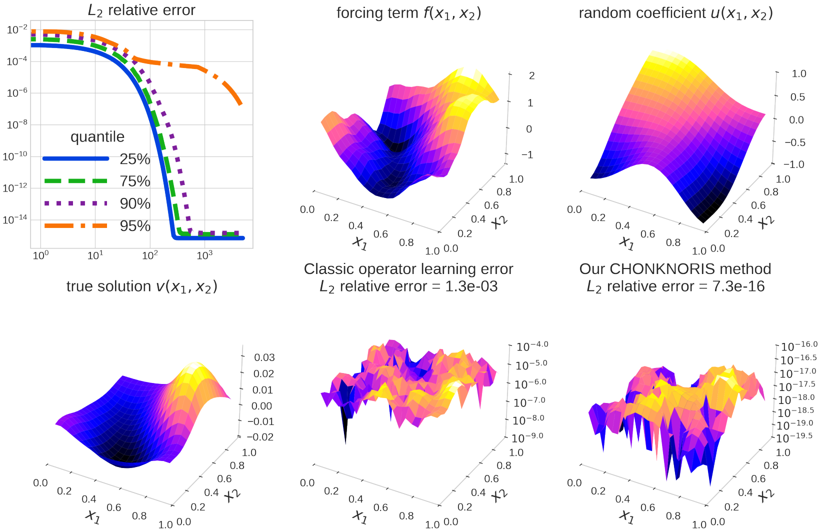
- CHONKNORIS requires more iterations but avoids cubic inversion cost in NK method
- Initial guess from existing sciML method can accelerate convergence

Burgers' Equation with Shocks via Implicit Time Stepping

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

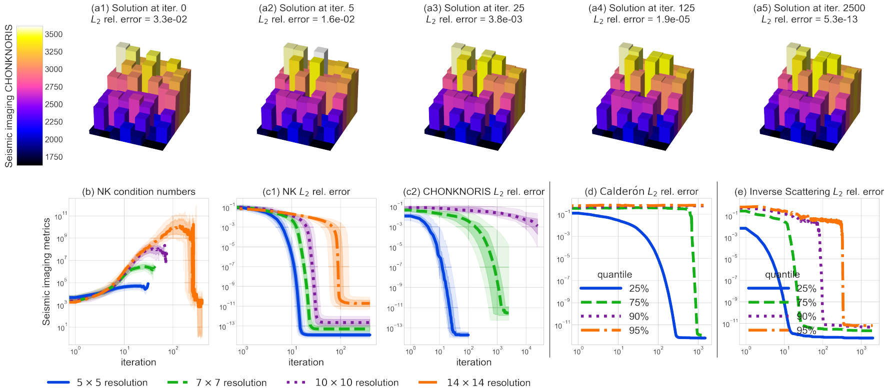


Darcy Flow in Two Dimensions



Inverse Problems

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

`pip install qmcpy`: qmcsoftware.github.io/QMCSoftware/ for Quasi-Monte Carlo

- Low-discrepancy sequences with randomizations
- Automatic transforms, to rewrite problem into $f : [0,1]^d \to \mathbb{R}$ with uniform stochasticity
- Diverse use cases, including financial options, parameterized PDEs, sensitivity indices, ...
- Adaptive stopping criteria, and vectorized algorithms for functions of multiple expectations
- (Digitally)-shift-invariant kernels, of varying smoothness

`pip install fastgps`: alegresor.github.io/fastgps/ for fast Gaussian process

- Single task and multitask fast GPs as well as baseline standard GPs
- Kernel hyperparameter optimization via MLL or GCV losses
- Fast Bayesian cubature, including for multitask GPs
- Efficient batched GPs with GPU support

Applications of QMC, fast GPs, sciML, and more

References I

[1] Ahmed M. Alaa and Mihaela Van Der Schaar. Bayesian inference of individualized treatment effects using multi-task Gaussian processes. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 3424–3432. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/6a508a60aa3bf9510ea6acb021c94b48-Abstract.html`.

[2] Aras Bacho, Aleksei G. Sorokin, Xianjin Yang, Théo Bourdais, Edoardo Calvello, Matthieu Darcy, Alexander Hsu, Bamdad Hosseini, and Houman Owhadi. Operator learning at machine precision. *ArXiv preprint*, abs/2511.19980, 2025. URL `https://arxiv.org/abs/2511.19980`.

[3] Sangjune Bae, Chanyoung Park, and Nam H. Kim. Estimating effect of additional sample on uncertainty reduction in reliability analysis using Gaussian process. *Journal of Mechanical Design*, 142(11):111706, 2020.

References II

[4] Jan Baldeaux, Josef Dick, Gunther Leobacher, Dirk Nuyens, and Friedrich Pillichshammer. Efficient calculation of the worst-case error and (fast) component-by-component construction of higher order polynomial lattice rules. *Numerical Algorithms*, 59(3): 403–431, 2012.

[5] Pau Batlle, Matthieu Darcy, Bamdad Hosseini, and Houman Owhadi. Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496:112549, 2024. ISSN 0021-9991. doi: 10.1016/j.jcp.2023.112549. URL `https://www.sciencedirect.com/science/article/pii/S0021999123006447`.

[6] Pau Batlle, Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Error analysis of kernel/GP methods for nonlinear and parametric PDEs. *Journal of Computational Physics*, 520:113488, 2025. ISSN 0021-9991. doi: 10.1016/j.jcp.2024.113488. URL `https://www.sciencedirect.com/science/article/pii/S0021999124007368`.

References III

[7] Edwin V. Bonilla, Kian Ming Adam Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 153–160. Curran Associates, Inc., 2007. URL `https://proceedings.neurips.cc/paper/2007/hash/66368270ffd51418ec58bd793f2d9b1b-Abstract.html`.

[8] Nicolas Boullé and Alex Townsend. A mathematical guide to operator learning. In *Handbook of Numerical Analysis*, volume 25, pages 83–125. Elsevier, 2024.

[9] François-Xavier Briol, Chris J. Oates, Mark Girolami, Michael A. Osborne, and Dino Sejdinovic. Probabilistic integration. *Statistical Science*, 34(1):1–22, 2019.

[10] Kian Ming Chai. *Multi-task learning with Gaussian processes*. PhD thesis, The University of Edinburgh, 2010.

References IV

[11] Kian Ming Adam Chai, Christopher K. I. Williams, Stefan Klanke, and Sethu Vijayakumar. Multi-task Gaussian process learning of robot inverse dynamics. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21, pages 265–272. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/hash/15d4e891d784977cacbfcbb00c48f133-Abstract.html.

[12] Jingrun Chen, Rui Du, Panchi Li, and Liyao Lyu. Quasi-Monte Carlo sampling for solving partial differential equations by deep neural networks. *Numerical Mathematics: Theory, Methods & Applications*, 14(2), 2021.

References V

[13] Yehu Chen, Annamaria Prati, Jacob Montgomery, and Roman Garnett. A multi-task Gaussian process model for inferring time-varying treatment effects in panel data. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 4068–4088. PMLR, 2023. URL https://proceedings.mlr.press/v206/chen23d.html.

[14] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447: 110668, 2021. ISSN 0021-9991. doi: 10.1016/j.jcp.2021.110668. URL https://www.sciencedirect.com/science/article/pii/S0021999121005635.

[15] Yifan Chen, Houman Owhadi, and Florian Schäfer. Sparse Cholesky factorization for solving nonlinear PDEs via Gaussian processes. *Mathematics of Computation*, 94(353): 1235–1280, 2025.

References VI

[16] Sou-Cheng T. Choi, Yuhan Ding, Fred J. Hickernell, Jagadeeswaran Rathinavel, and Aleksei G. Sorokin. Challenges in developing great quasi-Monte Carlo software. In Aicke Hinrichs, Peter Kritzer, and Friedrich Pillichshammer, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2022*, pages 209–222. Springer, 2022.

[17] Sou-Cheng T. Choi, Fred J. Hickernell, Jagadeeswaran Rathinavel, Michael J. McCourt, and Aleksei G. Sorokin. Quasi-Monte Carlo software. In Alexander Keller, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2020*, pages 23–47, Cham, 2022. Springer International Publishing. ISBN 978-3-030-98319-2.

[18] Jon Cockayne, Chris J. Oates, Tim Sullivan, and Mark Girolami. Probabilistic numerical methods for PDE-constrained Bayesian inverse problems. In *AIP Conference Proceedings*, volume 1853, page 060001. AIP Publishing LLC, 2017.

[19] Nathan Collier, Abdul-Lateef Haji-Ali, Fabio Nobile, Erik Von Schwerin, and Raúl Tempone. A continuation multilevel Monte Carlo algorithm. *BIT Numerical Mathematics*, 55(2):399–432, 2015.

References VII

[20] Sihui Dai, Jialin Song, and Yisong Yue. Multi-task Bayesian optimization via Gaussian process upper confidence bound. In *ICML 2020 workshop on real world experiment design and active learning*, volume 60, page 61, 2020.

[21] 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. URL https://proceedings.neurips.cc/paper/2022/hash/27d3ef263c7cb8d542c4f9815a49b69b-Abstract.html.

[22] Josef Dick. Higher order scrambled digital nets achieve the optimal rate of the root mean square error for smooth integrands. *The Annals of Statistics*, 39(3):1372 – 1398, 2011. doi: 10.1214/11-AOS880. URL 10.1214/11-AOS880.

[23] Josef Dick and Friedrich Pillichshammer. *Digital nets and sequences: discrepancy theory and quasi–Monte Carlo integration*. Cambridge University Press, 2010.

References VIII

[24] Josef Dick, Frances Y. Kuo, and Ian H. Sloan. High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013.

[25] Josef Dick, Peter Kritzer, and Friedrich Pillichshammer. *Lattice rules*. Springer, 2022.

[26] Josef Dick, Takashi Goda, Gerhard Larcher, Friedrich Pillichshammer, and Kosuke Suzuki. On the quasi-uniformity properties of quasi-Monte Carlo lattice point sets and sequences. *ArXiv preprint*, abs/2502.06202, 2025. URL `https://arxiv.org/abs/2502.06202`.

[27] Josef Dick, Takashi Goda, and Kosuke Suzuki. On the quasi-uniformity properties of quasi-Monte Carlo digital nets and sequences. *ArXiv preprint*, abs/2501.18226, 2025. URL `https://arxiv.org/abs/2501.18226`.

[28] Vincent Dubourg, Bruno Sudret, and Franois Deheeger. Metamodel-based importance sampling for structural reliability analysis. *Probabilistic Engineering Mechanics*, 33:47–57, 2013.

References IX

[29] Robert Dürichen, Marco A. F. Pimentel, Lei Clifton, Achim Schweikard, and David A. Clifton. Multi-task Gaussian process models for biomedical applications. In *IEEE-EMBS international Conference on Biomedical and health informatics (BHI)*, pages 492–495. IEEE, 2014.

[30] David Eriksson, Kun Dong, Eric Hans Lee, David Bindel, and Andrew Gordon Wilson. Scaling Gaussian process regression with derivatives. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 6868–6878. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/c2f32522a84d5e6357e6abac087f1b0b-Abstract.html`.

[31] Gregory E. Fasshauer. *Meshfree approximation methods with MATLAB*, volume 6. World Scientific, 2007.

[32] Peter I. Frazier. A tutorial on Bayesian optimization. *ArXiv preprint*, abs/1807.02811, 2018. URL `https://arxiv.org/abs/1807.02811`.

References X

[33] Joseph Futoma, Sanjay Hariharan, and Katherine A. Heller. Learning to detect sepsis with a multitask Gaussian process RNN classifier. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1174–1182. PMLR, 2017. URL http://proceedings.mlr.press/v70/futoma17a.html.

[34] Jacob R. Gardner, Geoff Pleiss, Kilian Q. Weinberger, David Bindel, and Andrew Gordon Wilson. GPyTorch: blackbox matrix-matrix Gaussian process inference with GPU acceleration. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 7587–7597. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/27e8e17134dd7083b050476733207ea1-Abstract.html.

References XI

[35] Jacob R. Gardner, Geoff Pleiss, Ruihan Wu, Kilian Q. Weinberger, and Andrew Gordon Wilson. Product kernel interpolation for scalable Gaussian processes. In Amos J. Storkey and Fernando Pérez-Cruz, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 1407–1416. PMLR, 2018. URL `http://proceedings.mlr.press/v84/gardner18a.html`.

[36] Zoubin Ghahramani and Carl Edward Rasmussen. Bayesian Monte Carlo. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 489–496. MIT Press, 2002. URL `https://proceedings.neurips.cc/paper/2002/hash/24917db15c4e37e421866448c9ab23d8-Abstract.html`.

[37] Michael B. Giles. Multilevel Monte Carlo path simulation. *Operations research*, 56(3): 607–617, 2008.

References XII

[38] Michael B. Giles and Benjamin J. Waterhouse. Multilevel quasi-Monte Carlo path simulation. *Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics*, 8:165–181, 2009.

[39] Eda Gjergo, Aleksei G. Sorokin, Anthony Ruth, Emanuele Spitoni, Francesca Matteucci, Xilong Fan, Jinning Liang, Marco Limongi, Yuta Yamazaki, Motohiko Kusakabe, and Toshitaka Kajino. GalCEM. I. An open-source detailed isotopic chemical evolution code, 2023. URL `https://dx.doi.org/10.3847/1538-4365/aca7c7`.

[40] Eda Gjergo, Aleksei G. Sorokin, Anthony Ruth, Emanuele Spitoni, Francesca Matteucci, Xilong Fan, Jinning Liang, Marco Limongi, Yuta Yamazaki, Motohiko Kusakabe, et al. GalCEM: galactic chemical evolution model. *Astrophysics Source Code Library*, pages ascl–2301, 2023.

References XIII

[41] Ivan G. Graham, Frances Y. Kuo, Dirk Nuyens, Robert Scheichl, and Ian H. Sloan. Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *Journal of Computational Physics*, 230(10):3668–3694, 2011. ISSN 0021-9991. doi: 10.1016/j.jcp.2011.01.023. URL https://www.sciencedirect.com/science/article/pii/S0021999111000489.

[42] Ivan G. Graham, Frances Y. Kuo, James A. Nichols, Robert Scheichl, Christoph Schwab, and Ian H. Sloan. Quasi-Monte Carlo finite element methods for elliptic PDEs with lognormal random coefficients. *Numerische Mathematik*, 131(2):329–368, 2015.

[43] Fred J. Hickernell. A generalized discrepancy and quadrature error bound. *Mathematics of computation*, 67(221):299–322, 1998.

[44] Fred J. Hickernell. Lattice rules: how well do they measure up? In *Random and quasi-random point sets*, pages 109–166. Springer, 1998.

References XIV

[45] Fred J. Hickernell and Lluís Antoni Jiménez Rugama. Reliable adaptive cubature using digital sequences. *ArXiv preprint*, abs/1410.8615, 2014. URL https://arxiv.org/abs/1410.8615.

[46] Fred J. Hickernell, Lan Jiang, Yuewei Liu, and Art B. Owen. Guaranteed conservative fixed width confidence intervals via Monte Carlo sampling. In Josef Dick, Frances Y. Kuo, Gareth W. Peters, and Ian H. Sload, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2012*, pages 105–128. Springer, 2013.

[47] Fred J. Hickernell, Lluís Antoni Jiménez Rugama, and Da Li. Adaptive quasi-Monte Carlo methods for cubature. *ArXiv preprint*, abs/1702.01491, 2017. URL https://arxiv.org/abs/1702.01491.

[48] Fred J. Hickernell, Nathan Kirk, and Aleksei G. Sorokin. Quasi-Monte Carlo methods: what, why, and how? *ArXiv preprint*, abs/2502.03644, 2025. URL https://arxiv.org/abs/2502.03644.

References XV

[49] Jiangli Huang, Shuhan Zhang, Cong Tao, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Bayesian optimization approach for analog circuit design using multi-task Gaussian process. In *2021 IEEE international symposium on circuits and systems (ISCAS)*, pages 1–5. IEEE, 2021.

[50] Aadit Jain, Fred J. Hickernell, Art B. Owen, and Aleksei G. Sorokin. Empirical Bernstein and betting confidence intervals for randomized quasi-Monte Carlo. *ArXiv preprint*, abs/2504.18677, 2025. URL `https://arxiv.org/abs/2504.18677`.

[51] Henrik Wann Jensen, James Arvo, Phil Dutre, Alexander Keller, Art B. Owen, Matt Pharr, and Peter Shirley. Monte Carlo ray tracing. In *ACM SIGGRAPH*, volume 5, page 340769537, 2003.

[52] Lluís Antoni Jiménez Rugama and Fred J. Hickernell. Adaptive multidimensional integration based on rank-1 lattices. *ArXiv preprint*, abs/1411.1966, 2014. URL `https://arxiv.org/abs/1411.1966`.

References XVI

[53] Corwin Joy, Phelim P. Boyle, and Ken Seng Tan. Quasi-Monte Carlo methods in numerical finance. *Management science*, 42(6):926–938, 1996.

[54] Vesa Kaarnioja, Yoshihito Kazashi, Frances Y. Kuo, Fabio Nobile, and Ian H. Sloan. Fast approximation by periodic kernel-based lattice-point interpolation with application in uncertainty quantification. *Numerische Mathematik*, pages 1–45, 2022.

[55] Vesa Kaarnioja, Frances Y. Kuo, and Ian H. Sloan. Lattice-based kernel approximation and serendipitous weights for parametric PDEs in very high dimensions. *ArXiv preprint*, abs/2303.17755, 2023. URL `https://arxiv.org/abs/2303.17755`.

[56] 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.

[57] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

References XVII

[58] Bradley D. Keister. Multidimensional quadrature algorithms. *Computers in Physics*, 10: 119–122, 1996. doi: 10.1063/1.168565.

[59] Alexander Keller, Frances Y. Kuo, Dirk Nuyens, and Ian H. Sloan. Regularity and tailored regularization of deep neural networks, with application to parametric PDEs in uncertainty quantification. *ArXiv preprint*, abs/2502.12496, 2025. URL https://arxiv.org/abs/2502.12496.

[60] Danial Khatamsaz, Brent Vela, and Raymundo Arróyave. Multi-objective Bayesian alloy design using multi-task Gaussian processes. *Materials Letters*, 351:135067, 2023.

[61] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[62] Nikola B. Kovachki, Samuel Lanthaler, and Andrew M. Stuart. Operator learning: algorithms and analysis. *Handbook of Numerical Analysis*, 25:419–467, 2024.

References XVIII

[63] Dirk P. Kroese, Thomas Taimre, and Zdravko I. Botev. *Handbook of Monte Carlo methods*. John Wiley & Sons, 2013.

[64] Frances Y. Kuo and Dirk Nuyens. Application of quasi-Monte Carlo methods to elliptic PDEs with random diffusion coefficients - a survey of analysis and implementation. *Foundations of Computational Mathematics*, 16(6):1631–1696, 2016. ISSN 1615-3375. doi: 10.1007/s10208-016-9329-5. URL `https://doi.org/10.1007/s10208-016-9329-5`.

[65] Frances Y. Kuo, Christoph Schwab, and Ian H. Sloan. Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM Journal on Numerical Analysis*, 50(6):3351–3374, 2012.

[66] Frances Y. Kuo, Christoph Schwab, and Ian H. Sloan. Multi-level quasi-Monte Carlo finite element methods for a class of elliptic PDEs with random coefficients. *Foundations of Computational Mathematics*, 15(2):411–449, 2015.

References XIX

[67] Yongzeng Lai and Jerome Spanier. Applications of Monte Carlo/quasi-Monte Carlo methods in finance: option pricing. In Harald Niederreiter and Jerome Spanier, editors, *Monte-Carlo and Quasi-Monte Carlo Methods 1998*, pages 284–295. Springer, 1998.

[68] Pierre L'Ecuyer. Quasi-Monte Carlo methods in finance. In *Proceedings of the Winter Simulation Conference 2004*, volume 2, pages 1645–1655. IEEE, 2004. ISBN 0780387864.

[69] Pierre L'Ecuyer. Quasi-Monte Carlo methods with applications in finance. *Finance and Stochastics*, 13(3):307–349, 2009.

[70] Pierre L'Ecuyer. Randomized quasi-Monte Carlo: an introduction for practitioners. In Art B. Owen and Peter W. Glynn, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2016*, pages 29–52. Springer, 2016.

[71] Pierre L'Ecuyer, Marvin K. Nakayama, Art B. Owen, and Bruno Tuffin. Confidence intervals for randomized quasi-Monte Carlo estimators. In *Proceedings of the Winter Simulation Conference 2023*, pages 445–456. IEEE, 2023.

References XX

[72] Christiane Lemieux. *Monte Carlo and quasi-Monte Carlo sampling*, volume 20. Springer, 2009.

[73] 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*, abs/2010.08895, 2020. URL `https://arxiv.org/abs/2010.08895`.

[74] Haitao Liu, Kai Wu, Yew-Soon Ong, Chao Bian, Xiaomo Jiang, and Xiaofang Wang. Learning multitask Gaussian process over heterogeneous input domains. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(10):6232–6244, 2023.

[75] Da Long, Nicole Mrvaljević, Shandian Zhe, and Bamdad Hosseini. A kernel framework for learning differential equations and their solution operators. *Physica D: Nonlinear Phenomena*, 460:134095, 2024.

References XXI

[76] Marcello Longo, Siddhartha Mishra, T. Konstantin Rusch, and Christoph Schwab. Higher-order quasi-Monte Carlo training of deep neural networks. *SIAM Journal on Scientific Computing*, 43(6):A3938–A3966, 2021.

[77] 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. *Nat. Mach. Intell.*, 3(3):218–229, 2021.

[78] Jiří Matoušek. On the L2-discrepancy for anchored boxes. *Journal of Complexity*, 14(4): 527–556, 1998. ISSN 0885-064X. doi: 10.1006/jcom.1998.0489. URL https://www.sciencedirect.com/science/article/pii/S0885064X98904897.

[79] 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 XXII

[80] 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.

[81] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, 1992.

[82] Anthony O'Hagan. Bayes–Hermite quadrature. *Journal of statistical planning and inference*, 29(3):245–260, 1991.

[83] Art B. Owen. Randomly permuted (t, m, s)-nets and (t, s)-sequences. In Harald Niederreiter and Peter J. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods 1994*, pages 299–317. Springer, 1995.

[84] Art B. Owen. Variance and discrepancy with alternative scramblings. *ACM Transactions of Modeling and Computer Simulation*, 13(4), 2003.

References XXIII

[85] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013. URL https://artowen.su.domains/mc/.

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

[87] Houman Owhadi and Clint Scovel. *Operator-adapted wavelets, fast solvers, and numerical homogenization: from a game theoretic approach to numerical approximation and algorithm design*, volume 35. Cambridge University Press, 2019.

[88] Houman Owhadi and Gene Ryan Yoo. Kernel flows: from learning kernels from data into the abyss. *Journal of Computational Physics*, 389:22–47, 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.03.040. URL https://www.sciencedirect.com/science/article/pii/S0021999119302232.

References XXIV

[89] Misha Padidar, Xinran Zhu, Leo Huang, Jacob Gardner, and David Bindel. Scaling Gaussian processes with derivative information using variational inference. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann Dauphin, Percy S. Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6442–6453. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/32bbf7b2bc4ed14eb1e9c2580056a989-Abstract.html.

[90] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: an imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035.

References XXV

Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[91] Matthias Raab, Daniel Seibert, and Alexander Keller. Unbiased global illumination with participating media. In Alexander Keller, Stefan Heinrich, and Harald Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 591–605. Springer, 2006.

[92] Rüdiger Rackwitz. Reliability analysis–a review and some perspectives. *Structural safety*, 23(4):365–395, 2001.

[93] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

References XXVI

[94] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, Cambridge, Massachusetts, 2006. URL http://www.gaussianprocess.org/gpml/.

[95] Jagadeeswaran Rathinavel. *Fast automatic Bayesian cubature using matching kernels and designs*. PhD thesis, Illinois Institute of Technology, 2019.

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

[97] 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.

References XXVII

[98] S. Ashwin Renganathan, Vishwas Rao, and Ionel M. Navon. CAMERA: a method for cost-aware, adaptive, multifidelity, efficient reliability analysis. *Journal of Computational Physics*, 472:111698, 2023. ISSN 0021-9991. doi: 10.1016/J.JCP.2022.111698. URL https://www.sciencedirect.com/science/article/pii/S0021999122007616.

[99] Pieterjan Robbe. *Multilevel uncertainty quantification methods for robust design of industrial applications*. PhD thesis, KU Leuven, 2019.

[100] Pieterjan Robbe, Dirk Nuyens, and Stefan Vandewalle. A dimension-adaptive multi-index Monte Carlo method applied to a model of a heat exchanger. In Art B. Owen and Peter W. Glynn, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2016*, pages 429–445. Springer, 2016.

[101] Pieterjan Robbe, Dirk Nuyens, and Stefan Vandewalle. A multi-index quasi–Monte Carlo algorithm for lognormal diffusion problems. *SIAM Journal on Scientific Computing*, 39(5): S851–S872, 2017.

References XXVIII

[102] Pieterjan Robbe, Dirk Nuyens, and Stefan Vandewalle. Recycling samples in the multigrid multilevel (quasi-) Monte Carlo method. *SIAM Journal on Scientific Computing*, 41(5): S37–S60, 2019.

[103] Yunus Saatçi. *Scalable inference for structured Gaussian process models*. PhD thesis, Citeseer, 2012.

[104] Linus Seelinger, Vivian Cheng-Seelinger, Andrew Davis, Matthew Parno, and Anne Reinarz. UM-Bridge: uncertainty quantification and modeling bridge. *Journal of Open Source Software*, 8(83):4748, 2023.

[105] Ian H. Sloan and Stephen Joe. *Lattice methods for multiple integration*. Oxford University Press, 1994.

References XXIX

[106] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 2960–2968. Curran Associates, Inc., 2012. URL `https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html`.

[107] Ercan Solak, Roderick Murray-Smith, William E. Leithead, Douglas J. Leith, and Carl Edward Rasmussen. Derivative observations in Gaussian process models of dynamic systems. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 1033–1040. MIT Press, 2002. URL `https://proceedings.neurips.cc/paper/2002/hash/5b8e4fd39d9786228649a8a8bec4e008-Abstract.html`.

References XXX

[108] Aleksei G. Sorokin. QMCPy: a Python software for randomized low-discrepancy sequences, quasi-Monte Carlo, and fast kernel methods. *ArXiv preprint*, abs/2502.14256, 2025. URL `https://arxiv.org/abs/2502.14256`.

[109] Aleksei G. Sorokin and Vishwas Rao. Credible intervals for probability of failure with Gaussian processes. *ArXiv preprint*, abs/2311.07733, 2023. URL `https://arxiv.org/abs/2311.07733`.

[110] Aleksei G. Sorokin and Jagadeeswaran Rathinavel. On bounding and approximating functions of multiple expectations using quasi-Monte Carlo. In Aicke Hinrichs, Peter Kritzer, and Friedrich Pillichshammer, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2022*, pages 583–599. Springer, 2022.

[111] Aleksei G. Sorokin, Fred J. Hickernell, Sou-Cheng T. Choi, Michael J. McCourt, and Jagadeeswaran Rathinavel. (Quasi-)Monte Carlo importance sampling with QMCPy. *IIT Undergraduate Research Journal*, pages 49–54, 2021. URL `http://urj.library.iit.edu/index.php/urj/article/view/48`.

References XXXI

[112] Aleksei G. Sorokin, Xinran Zhu, Eric Hans Lee, and Bolong Cheng. SigOpt Mulch: an intelligent system for AutoML of gradient boosted trees. *Knowledge-Based Systems*, page 110604, 2023. ISSN 0950-7051. doi: 10.1016/j.knosys.2023.110604. URL `https://www.sciencedirect.com/science/article/pii/S0950705123003544`.

[113] Aleksei G. Sorokin, Xiaoyi Lu, and Yi Wang. A neural surrogate solver for radiation transfer. In *NeurIPS 2024 Workshop on Data-Driven and Differentiable Simulations, Surrogates, and Solvers*, 2024. URL `https://openreview.net/forum?id=SHidR8UMKo`.

[114] Aleksei G. Sorokin, Aleksandra Pachalieva, Daniel O'Malley, James M. Hyman, Fred J. Hickernell, and Nicolas W. Hengartner. Computationally efficient and error aware surrogate construction for numerical solutions of subsurface flow through porous media. *Advances in Water Resources*, 193:104836, 2024. ISSN 0309-1708. doi: 10.1016/j.advwatres.2024.104836. URL `https://www.sciencedirect.com/science/article/pii/S0309170824002239`.

References XXXII

[115] Aleksei G. Sorokin, Pieterjan Robbe, Gianluca Geraci, Michael S. Eldred, and Fred J. Hickernell. Fast Bayesian multilevel quasi-Monte Carlo. *ArXiv preprint*, abs/2510.24604, 2025. URL `https://arxiv.org/abs/2510.24604`.

[116] Aleksei G. Sorokin, Pieterjan Robbe, and Fred J. Hickernell. Fast Gaussian process regression for high dimensional functions with derivative information. In Motonobu Kanagawa, Jon Cockayne, Alexandra Gessner, and Philipp Hennig, editors, *Proceedings of the First International Conference on Probabilistic Numerics*, volume 271 of *Proceedings of Machine Learning Research*, pages 35–49. PMLR, 2025. URL `https://proceedings.mlr.press/v271/sorokin25a.html`.

[117] Sonja Surjanovic and Derek Bingham. Virtual library of simulation experiments: test functions and datasets, 2013. URL `http://www.sfu.ca/~ssurjano`.

References XXXIII

[118] Kevin Swersky, Jasper Snoek, and Ryan Prescott Adams. Multi-task Bayesian optimization. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 2004–2012. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper/2013/hash/f33ba15effa5c10e873bf3842afb46a6-Abstract.html`.

[119] Shu Tezuka. On randomization of generalized Faure sequences. Technical report, Tech. Rep. RT0494, IBM Tokyo Research Laboratory, 2002.

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

[121] Carsten Waechter and Alexander Keller. Quasi-Monte Carlo light transport simulation by efficient ray tracing, 2011. US Patent 7,952,583.

References XXXIV

[122] Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1775–1784. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/wilson15.html.

[123] Andrew Gordon Wilson, Elad Gilboa, John P. Cunningham, and Arye Nehorai. Fast kernel learning for multidimensional pattern extrapolation. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 3626–3634. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/hash/77369e37b2aa1404f416275183ab055f-Abstract.html.

[124] Anqi Wu, Mikio C. Aoi, and Jonathan W. Pillow. Exploiting gradients and Hessians in Bayesian optimization and Bayesian quadrature. *ArXiv preprint*, abs/1704.00060, 2017. URL https://arxiv.org/abs/1704.00060.

References XXXV

[125] Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, and Andrew Gordon Wilson. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 788–798. AUAI Press, 2019. URL http://proceedings.mlr.press/v115/wu20a.html.

[126] Linlin Xu and Giray Ökten. High-performance financial simulation using randomized quasi-Monte Carlo methods. *Quantitative Finance*, 15(8):1425–1436, 2015.

[127] Andrea Zanette, Junzi Zhang, and Mykel J. Kochenderfer. Robust super-level set estimation using Gaussian processes. *ArXiv preprint*, abs/1811.09977, 2018. URL https://arxiv.org/abs/1811.09977.

[128] Xiaoyan Zeng, King-Tai Leung, and Fred J. Hickernell. *Error analysis of splines for periodic problems using lattice designs*. Springer, 2006.

References XXXVI

[129] Xiaoyan Zeng, Peter Kritzer, and Fred J. Hickernell. Spline methods using integration lattices and digital nets. *Constructive Approximation*, 30:529–555, 2009.