Unified Framework for Quasi-Monte Carlo Software

Aleksei G. Sorokin
Fred J. Hickernell, Jagadeeswaran Rathinavel, Sou-Cheng T. Choi

Illinois Institute of Technology, Department of Applied Mathematics

June 29, 2023

Outline

- Monte Carlo and Quasi-Monte Carlo formulation and components
- Node generators
  - **LatNetBuilder** for finding generating vectors and matrices
  - **Magic Point Shop** and **QRNG** for generating points from static vectors / matrices
- Distribution transforms
  - **SciPy** for finite dimensional distributions
  - **GaussianRandomFields.jl** for infinite dimensional distributions
- Example problems from **FEniCSx** / **DOLFINx** and **UM-Bridge**
- Adaptive sampling to meet tolerance
  - **GAIL** for finite dimensional distributions
  - **MultilevelEstimators.jl** for infinite dimensional distributions
  - **QMCPy** for vectorized algorithms

Monte Carlo Methods

$$\text{True Mean} = \mu = \mathbb{E}[g(T)] = \mathbb{E}[f(X)] = \int_{[0,1]^d} f(x)\mathrm{d}x$$

- g, the *original integrand* of the *original random variable* T.
- f, the *transformed integrand* of the *transformed random variable* $X \sim \mathcal{U}[0,1]^d$.

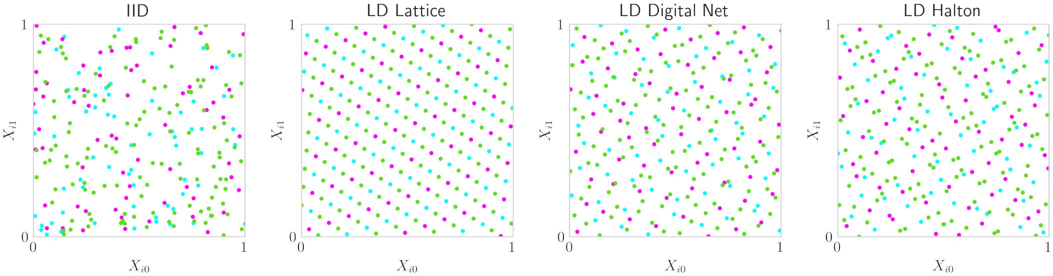
$$\text{Approximate Mean} = \hat{\mu} = \frac{1}{n}\sum_{i=0}^{n-1} f(X_i)$$

	Crude Monte Carlo	**Quasi-Monte Carlo**
$X_0, X_1, \cdots \sim \mathcal{U}[0,1]^d$	independent identically distributed (IID) gaps and clusters	low discrepancy (LD) even coverage
Rate of $\hat{\mu}$ to μ	$\mathcal{O}(n^{-1/2})$	$\mathcal{O}(n^{-1+\delta})$, any $\delta > 0$
Generally	slower	faster
Integrands require	finite variance	low order interactions

Discrte Distribution

Generate sampling locations $X_0, X_1, \dots$

QMCPy [1]: Python. IID generators from **NumPy**[2]. LD Generators from others (next).



[1] https://github.com/QMCSoftware/QMCSoftware

[2] Travis Oliphant. *Guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006. URL: https://ecs.wgtn.ac.nz/foswiki/pub/Support/ManualPagesAndDocumentation/numpybook.pdf.

Construct LD Generating Vectors / Matrices

LatNetBuilder[3]: C++.

Features	Lattice rules		Digital nets
Point set types	Rank-1 ordinary lattice rules	Rank-1 polynomial lattice rules	Sobol' nets Polynomial lattice rules Nets with explicit generating matrices
Figures of merit	P_α, R_α, spectral test	P_α, R	P_α, R, t-value, resolution-gap
Types of weights	projection-dependent, order-dependent, product, product-order-dependent and combined weights		
Exploration methods	evaluation, exhaustive, random, full CBC and random CBC, fast CBC		
Multilevel point sets	Embedded lattices		Digital sequences
Additional features	Extensible lattices, normalizations and filters	Extensible lattices, normalizations and filters, interlaced polynomial lattice rules	Interlaced digital nets

[3] Pierre L'Ecuyer et al. *A Tool for Custom Construction of QMC and RQMC Point Sets*. 2021. arXiv: 2012.10263 [stat.CO].

Generate LD Points from Static Generating Vectors / Matrices

Magic Point Shop[4]: C++, MATLAB, Python, Julia[5].

- Rank 1 Lattice in base 2.
- Digital nets and sequences in base 2, including higher order versions.

QRNG[6]: R

- Rank 1 lattice in base 2.
- Sobol', special cases of Digital nets in base 2.
- Halton, optionally generalized.

[4] 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". In: *Foundations of Computational Mathematics* 16 (2016), pp. 1631–1696.

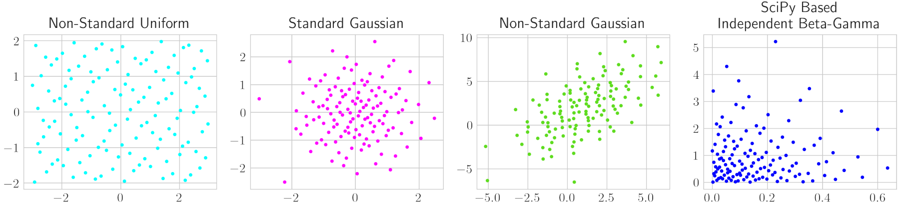
[5] https://github.com/alegresor/QMCGenerators.jl

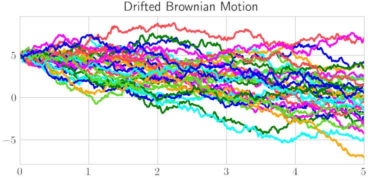
[6] Marius Hofert and Christiane Lemieux. *qrng: (Randomized) Quasi-Random Number Generators*. R package version 0.0-7. 2023. URL: https://CRAN.R-project.org/package=qrng.

True Measures

Define original distribution T and facilitate automatically transform so $\mathbb{E}[g(T)] = \mathbb{E}[f(X)]$

SciPy[7]: Univariate and multivariate distributions. Python.



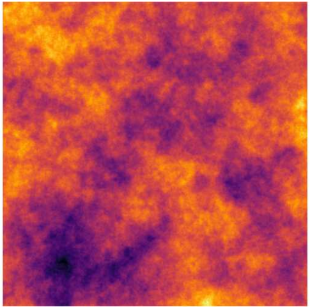


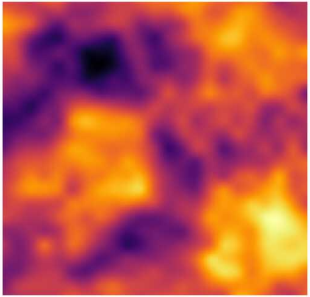
[7] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: https://doi.org/10.1038/s41592-019-0686-2.

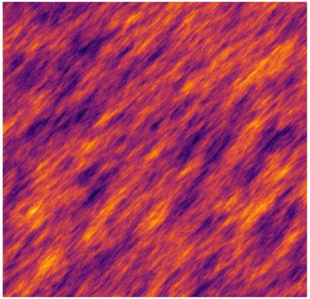
Infinite Dimensional Distributions Truncated to Finite Dimensions

GaussianRandomFields.jl [8]: Julia.

- Isotropic and anisotropic kernels with support for custom kernels.
- Implemented standard kernels e.g. Gaussian, Exponential, and Matérn.
- Generate with Cholesky, PCA, KL expansion, circulant embedding.
- Generate on a Finite Element mesh.







[8] https://github.com/PieterjanRobbe/GaussianRandomFields.jl

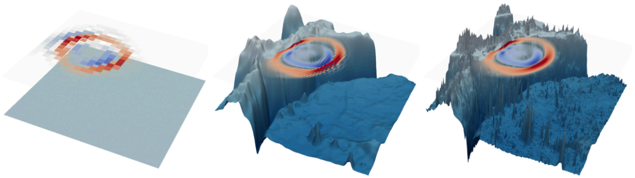
Integrands

Define original g, stores transformed f

FEniCSx[9] / **DOLFINx**[10]: Finite Element methods for PDEs. Python, C++.

UM-Bridge[11]: Clients in C++, MATLAB, Python, R.

- Containerize complex models for easy evaluation.
- Distribute models written in different languages.
- Support evaluation on HPC systems seamlessly.



[9] M. W. Scroggs et al. "Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes". In: *ACM Transactions on Mathematical Software* (2022). To appear. DOI: 10.1145/3524456.

[10] A. Logg, G. N. Wells, and J. Hake. "DOLFIN: a C++/Python Finite Element Library". In: *Automated Solution of Differential Equations by the Finite Element Method*. Ed. by A. Logg, K.-A. Mardal, and G. N. Wells. Vol. 84. Lecture Notes in Computational Science and Engineering. Springer, 2012. Chap. 10.

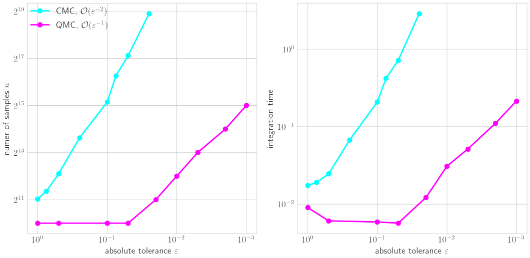
[11] Linus Seelinger et al. "UM-Bridge: Uncertainty quantification and modeling bridge". In: *Journal of Open Source Software* 8.83 (2023), p. 4748.

Stopping Criteria

Adaptively increase n until error $|\hat{\mu}-\mu| \leq \epsilon$ tolerance

GAIL: Guaranteed Automatic Integration Library[12]: MATLAB.

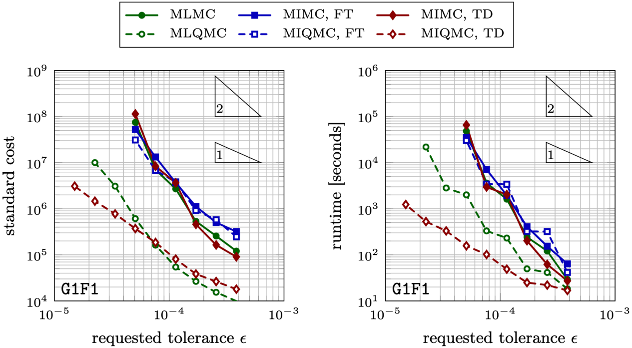
- Crude Monte Carlo by Berry-Esseen inequality, accounts for variance estimation.
- Quasi-Monte Carlo by IID randomizations.
- Quasi-Monte Carlo by decay of fourier coefficients.
- Quasi-Monte Carlo by Kriging with circulant embedding.



[12] Xin Tong et al. "Guaranteed Automatic Integration Library (GAIL): An Open-Source MATLAB Library for Function Approximation, Optimization, and Integration". In: *Journal of Open Research Software* 10.1 (2022).

MultilevelEstimators.jl[13]: For PDEs with random coefficients. Julia.

- Multilevel (Quasi-)Monte Carlo for 1D spacial domain.
- Multi-index (Quasi-)Monte Carlo[14] for > 1D spacial domain, see figure below.
- Adaptive, unbiased versions of multi-index (Quasi-)Monte Carlo.



[13] https://github.com/PieterjanRobbe/MultilevelEstimators.jl

[14] Pieterjan Robbe, Dirk Nuyens, and Stefan Vandewalle. “A multi-index quasi–Monte Carlo algorithm for lognormal diffusion problems”. In: *SIAM Journal on Scientific Computing* 39.5 (2017), S851–S872.

QMCPy[15]: Vectorized stopping criterion. Python.

QOIs $s \in \mathbb{R}^\eta$ are combination $C : \mathbb{R}^\rho \to \mathbb{R}^\eta$ of multiple expectations $(\mu_1, \dots, \mu_\rho)$

- Vector of expectations

$$s = (\mu_1, \dots, \mu_\rho)$$

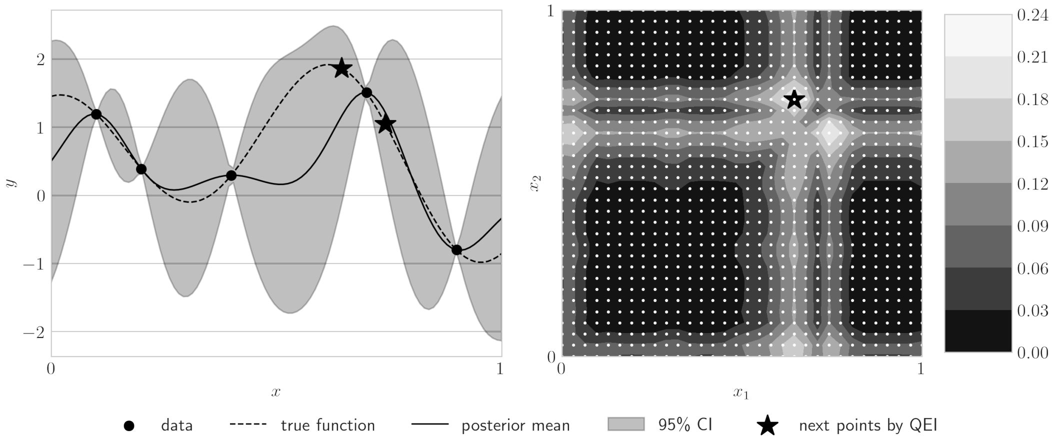
- Bayesian posterior mean of $\Theta \in \mathbb{R}^\eta$ given $(y_1, \dots, y_n)$

$$s_\ell = \mathbb{E}\left[\Theta_\ell \prod_{i=1}^{N} \rho(y_i \mid \Theta)\right] / \mathbb{E}\left[\prod_{i=1}^{N} \rho(y_i \mid \Theta)\right]$$

- Sensitivity indices (normalized Sobol' indices) for quantifying importance of function inputs

[15] Aleksei G. Sorokin and Rathinavel Jagadeeswaran. "Monte Carlo for Vector Functions of Integrals". In preparation for the 2022 Monte Carlo and Quasi-Monte Carlo Methods Conference Proceedings.

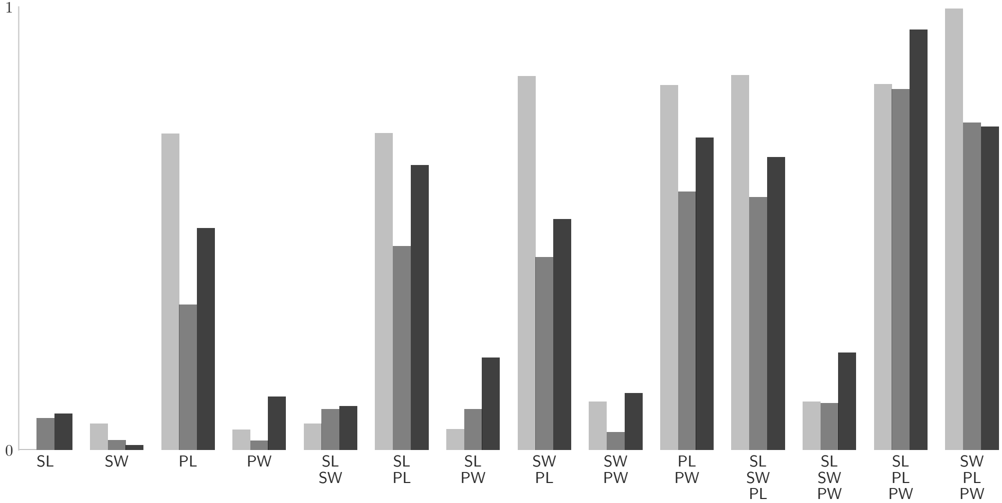
Bayesian Optimization: Vector of utility function evaluations (expectations)



Sensitivity Indices of Neural Network Classifier

Classify Iris species from Sepal Length, Sepal Width, Petal Length, Petal Width.

How important are subsets of attributes to each class?



Conclusions

- Unify (Q)MC software into common framework.
  - *Discrete Distributions* to generating sampling nodes
  - *True Measures* to automatically transform between distributions
  - *Integrands* to define user problems
  - *Stopping Criteria* to adaptively sample to meet tolerance
- Additional details and libraries: Sou-Cheng T. Choi et al. "Quasi-Monte Carlo Software". In: *Monte Carlo and Quasi-Monte Carlo Methods*. Ed. by Alexander Keller. Cham: Springer International Publishing, 2022, pp. 23–47. ISBN: 978-3-030-98319-2

Thank you for listening!

References I

[1] Sou-Cheng T. Choi et al. "Quasi-Monte Carlo Software". In: *Monte Carlo and Quasi-Monte Carlo Methods*. Ed. by Alexander Keller. Cham: Springer International Publishing, 2022, pp. 23–47. ISBN: 978-3-030-98319-2.

[2] Marius Hofert and Christiane Lemieux. *qrng: (Randomized) Quasi-Random Number Generators*. R package version 0.0-7. 2023. URL: https://CRAN.R-project.org/package=qrng.

[3] 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". In: *Foundations of Computational Mathematics* 16 (2016), pp. 1631–1696.

[4] Pierre L'Ecuyer et al. *A Tool for Custom Construction of QMC and RQMC Point Sets*. 2021. arXiv: 2012.10263 [stat.CO].

References II

[5] A. Logg, G. N. Wells, and J. Hake. “DOLFIN: a C++/Python Finite Element Library”. In: *Automated Solution of Differential Equations by the Finite Element Method*. Ed. by A. Logg, K.-A. Mardal, and G. N. Wells. Vol. 84. Lecture Notes in Computational Science and Engineering. Springer, 2012. Chap. 10.

[6] Travis Oliphant. *Guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006. URL: https://ecs.wgtn.ac.nz/foswiki/pub/Support/ManualPagesAndDocumentation/numpybook.pdf.

[7] Pieterjan Robbe, Dirk Nuyens, and Stefan Vandewalle. “A multi-index quasi–Monte Carlo algorithm for lognormal diffusion problems”. In: *SIAM Journal on Scientific Computing* 39.5 (2017), S851–S872.

References III

[8] M. W. Scroggs et al. "Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes". In: *ACM Transactions on Mathematical Software* (2022). To appear. DOI: 10.1145/3524456.

[9] Linus Seelinger et al. "UM-Bridge: Uncertainty quantification and modeling bridge". In: *Journal of Open Source Software* 8.83 (2023), p. 4748.

[10] Aleksei G. Sorokin and Rathinavel Jagadeeswaran. "Monte Carlo for Vector Functions of Integrals". In preparation for the 2022 Monte Carlo and Quasi-Monte Carlo Methods Conference Proceedings.

[11] Xin Tong et al. "Guaranteed Automatic Integration Library (GAIL): An Open-Source MATLAB Library for Function Approximation, Optimization, and Integration". In: *Journal of Open Research Software* 10.1 (2022).

References IV

[12] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: https://doi.org/10.1038/s41592-019-0686-2.