

Community Supported Quasi-Monte Carlo (QMC) Software

Aleksei Sorokin (asorokin@hawk.iit.edu)¹, Sou-Cheng Choi^{1,2},
Michael McCourt³, Jagadeeswaran Rathinavel¹, Fred Hickernell (hickernell@iit.edu)¹
¹Illinois Institute of Technology, ²Kamakura Corporation, ³SigOpt



Software Objectives

To provide QMC software that is:

- Comprised of free open source tools
- Easy to use for non-experts
- The recognized standard

The QMC Problem

Original Form

$$\mu = \int_T g(t) \lambda(dt)$$

$g : T \rightarrow \mathbb{R}$ = original integrand

λ = original measure

Convenient Form

$$\mu = \int_X f(x) \rho(x) dx = \int_X f(x) \nu(dx)$$

ν = well defined probability measure

$\phi : X \rightarrow T$ = change of variables

$f : X \rightarrow \mathbb{R}$ = integrand after change of variables

(Quasi-)Monte Carlo Approximation

$$\hat{\mu}_n = a_n \sum_{i=1}^n f(x_i) w_i = \int_X f(x) \hat{\nu}(dx)$$

$$\nu \approx \hat{\nu}_n = a_n \sum_{i=1}^n w_i \delta_{\hat{x}_i}(\cdot)$$

= discrete probability measure

Design Challenges

- Atomize Monte Carlo method into objects
- Define abstract methods and properties
- Unify existing components into framework
- Expand framework to allow multi-level problems
- Develop thorough documentation
- Ensure reproducibility

Keister Example

```
>>> from qmcpy import *
>>> CubLattice_g(
...     Keister(
...         Gaussian(
...             Lattice(dimension=2),
...             covariance = 1/2)),
...     abs_tol = .01).integrate()
```

Stopping Criterion

Determine n such that $|\mu - \hat{\mu}_n| \leq \epsilon$

- Central Limit Theorem (CLT)
- CLT Repeated
- Mean Monte Carlo (Guaranteed)
- Lattice Cubature (Guaranteed)

Integrand

Specify and generate values $f(\hat{x})$ for $\hat{x} \in \hat{\nu}$

- Keister
- Asian Call

True Measure

Specify components of a general sampling method

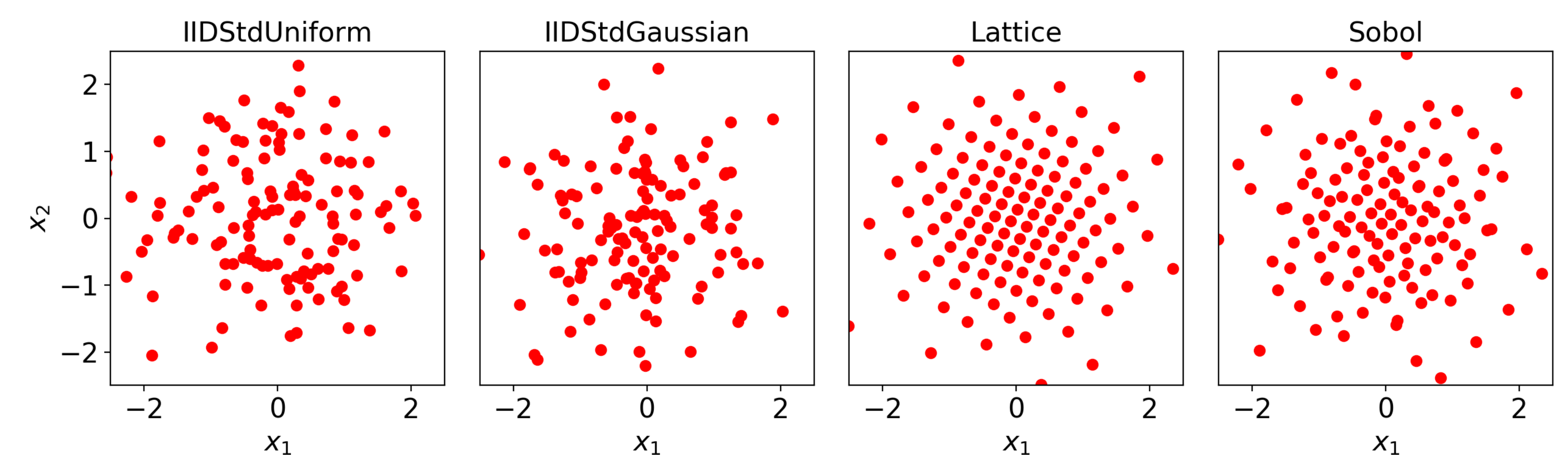
- Uniform
- Gaussian
- Brownian Motion
- Lebesgue

Discrete Distribution

Specify and generate $a_n \sum_{i=1}^n w_i \delta_{\hat{x}_i}(\cdot)$

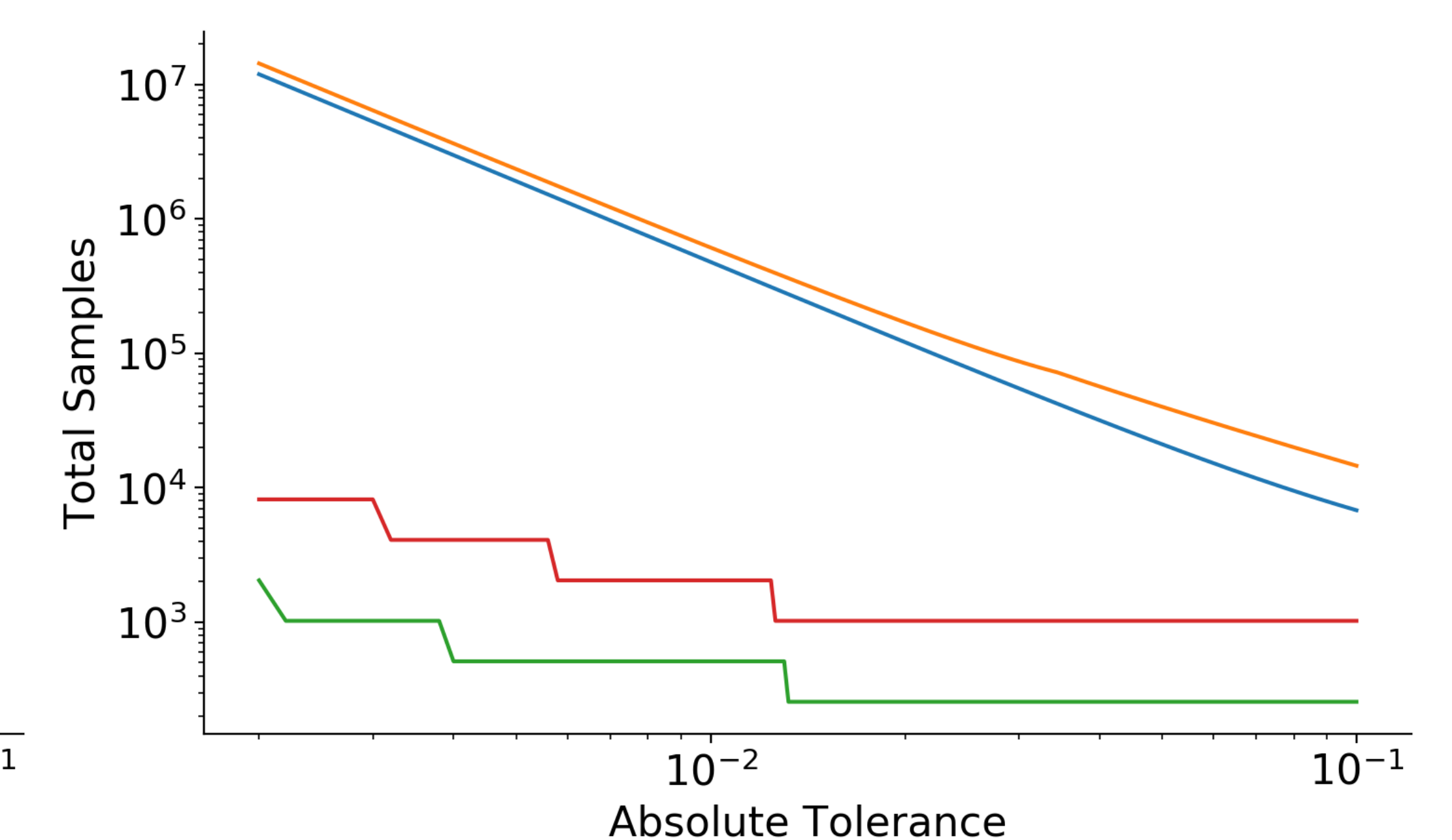
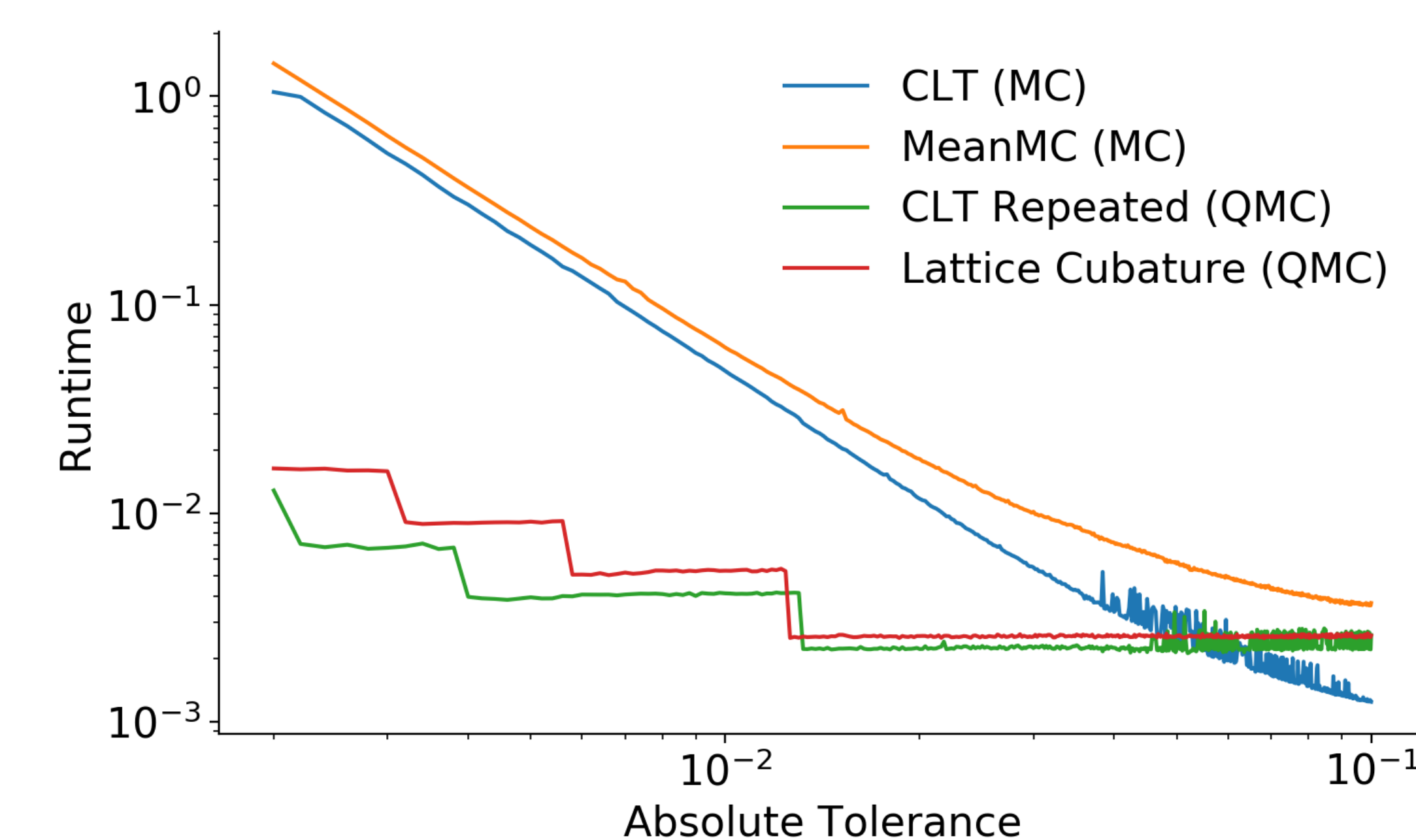
- IID Standard Uniform
- IID Standard Gaussian
- Lattice
- Sobol

Discrete Distributions Transformed to Mimic $\mathcal{N}_2(0, 1)$



```
>>> Gaussian(Sobol(dimension=2)).gen_samples(n_min=0, n_max=128)
```

Stopping Criterion Comparison on Keister Example



Future Work

- Expand library of examples
- Incorporate established research packages
- Grow community of contributors
- Utilize community feedback to improve software

Acknowledgements

- Thank you to SigOpt for continued funding and development support.
- Thank you to Dirk Nuyens for providing lattice and Sobol generators.

References

- <https://github.com/QMCSoftware/QMCSoftware>
- S.-C. T. Choi, Y. Ding, F. J. Hickernell, L. Jiang, Ll. A. Jiménez Rugama, D. Li, R. Jagadeeswaran, X. Tong, K. Zhang, Y. Zhang, and X. Zhou, "GAIL: Guaranteed Automatic Integration Library (versions 1.0-2.3)," MATLAB software, 2013-2019. http://gailgithub.github.io/GAIL_Dev/
- F. Y. Kuo and D. 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.