QMCPy: A Quasi-Monte Carlo (QMC) Software in Python 3

Aleksei G. Sorokin[1], Sou-Cheng T. Choi[1,2],
Fred J. Hickernell[1], Michael J. McCourt[3], Jagadeeswaran Rathinavel[1]

[1]Illinois Institute of Technology, [2]Kamakura Corporation, [3]SigOpt

Kamakura Corporation SIGOPT

The QMC Problem

Original Form

$$\mu = \int_{\mathcal{T}} g(\boldsymbol{t})\lambda(\boldsymbol{t})\mathrm{d}\boldsymbol{t}$$

$g : \mathcal{T} \to \mathbb{R}$ = original integrand
λ = true measure weight

Convenient Form

$$\mu = \int_{\mathcal{T}} g(\boldsymbol{t})\lambda(\boldsymbol{t})\mathrm{d}\boldsymbol{t} = \int_{[0,1]^d} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

$\boldsymbol{T} : [0,1]^d \to \mathcal{T}$ = change of variables
$f : [0,1]^d \to \mathbb{R}$ = integrand after change of variables

(Quasi-)Monte Carlo Approximation

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i) \approx \int_{[0,1]^d} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \mu$$

discrete distribution $= \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots\} \sim \mathcal{U}[0,1]^d$

QMCPy Sources

- Article for MCQMC2020 Proceedings [1]
- Package Distribution with PyPI: `pypi.org/project/qmcpy`
- Open Source Code on GitHub `github.com/QMCSoftware/QMCSoftware`
- Documentation on Read the Docs: `qmcpy.readthedocs.io/en/latest`
- QMC Blogs Posts Website: `qmcpy.org`
- Updates from QMC Software Google Group: `qmc-software@googlegroups.com`

Installation

To install QMCPy with Python run the command

```
pip install qmcpy
```

Acknowledgements

NSF Grant DMS-1522687, SIAM Travel Award, SigOpt

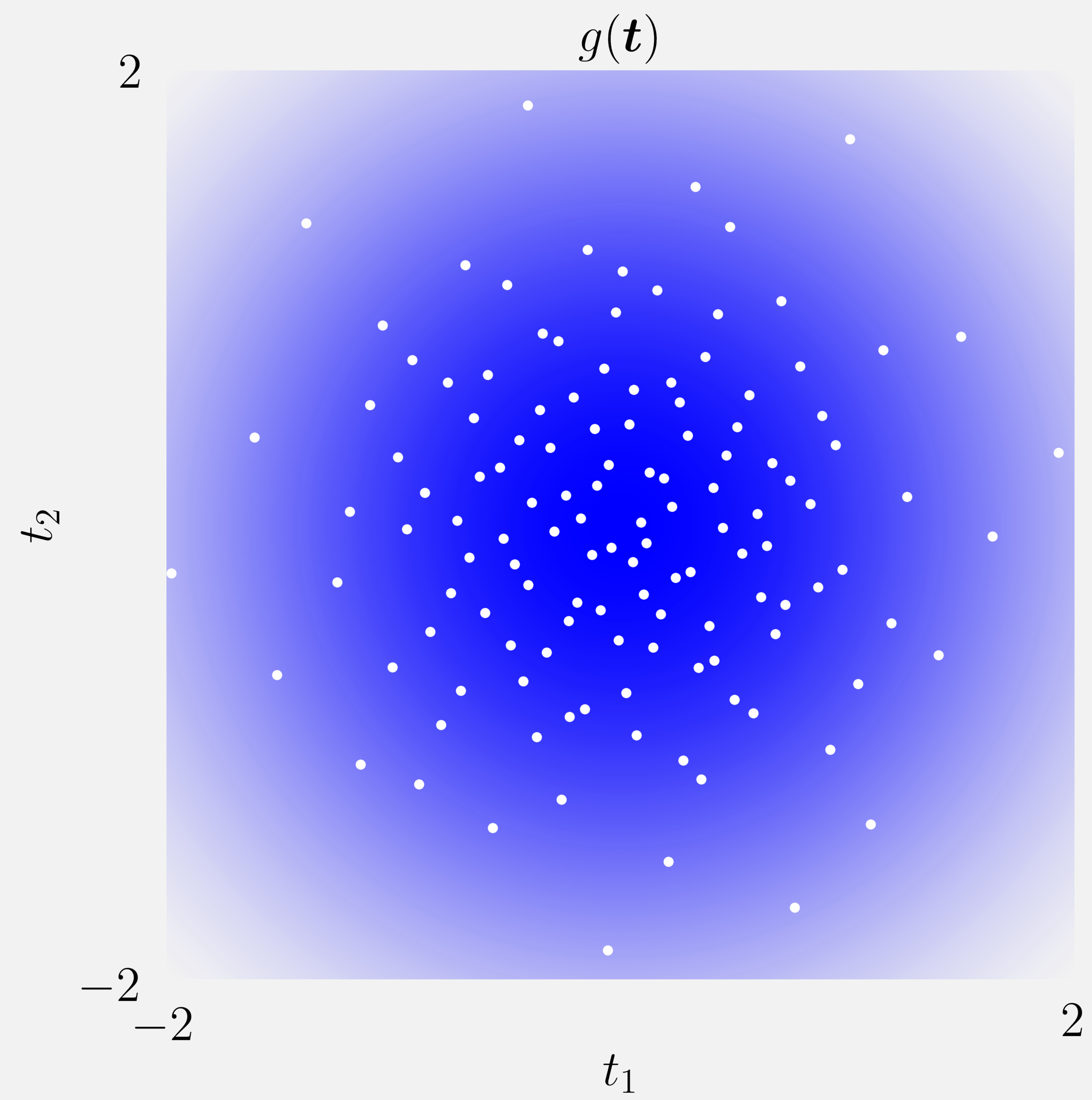
Keister Example

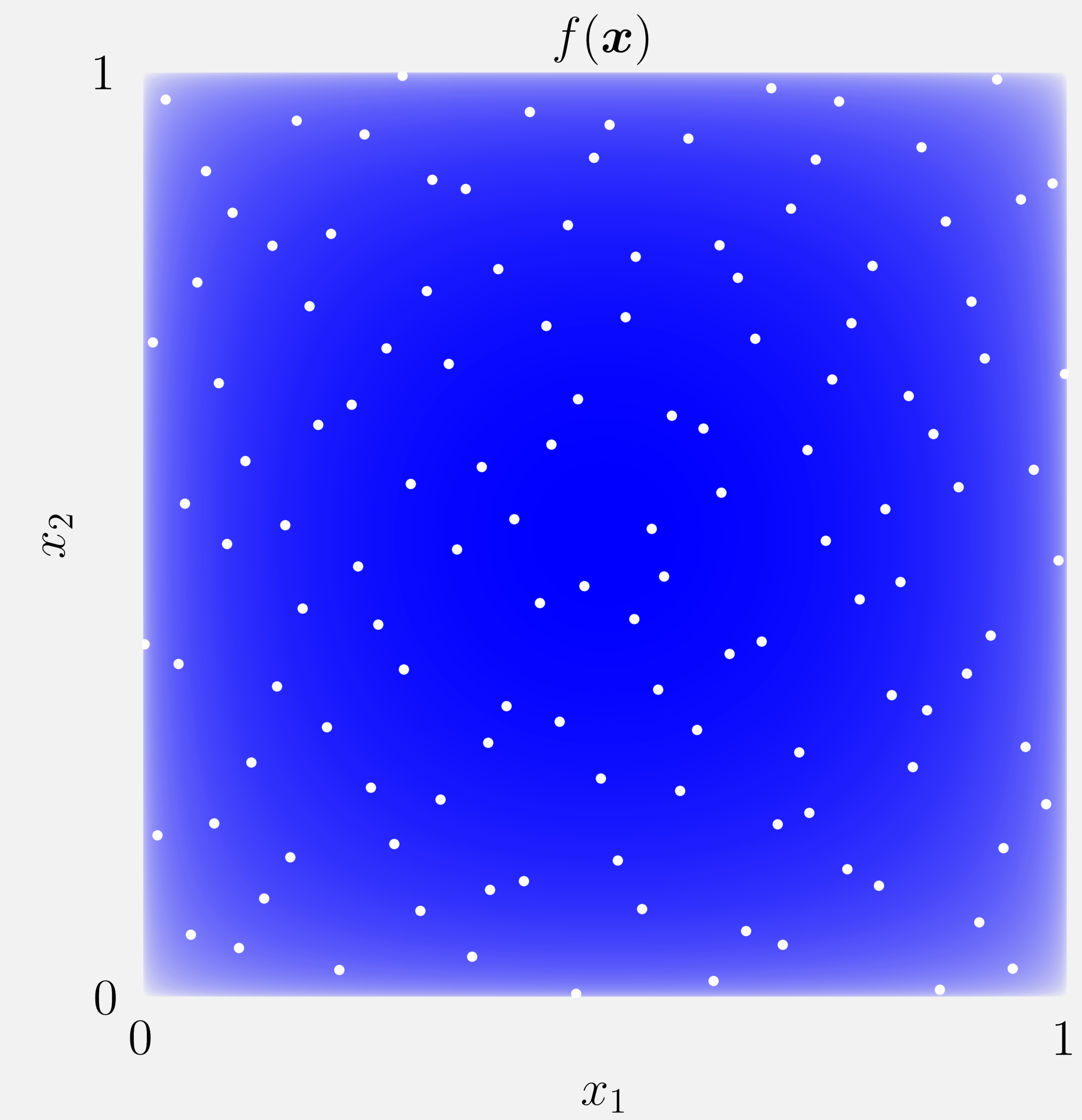
Original integrand [2]: $g(\boldsymbol{t}) = \pi^{d/2}\cos(||\boldsymbol{t}||)$
True measure: PDF λ for $\mathcal{N}(\boldsymbol{0}, \mathsf{I}/2)$
Discrete distribution: Sobol'

$$\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{\Phi}^{-1}(\boldsymbol{x})/2$$

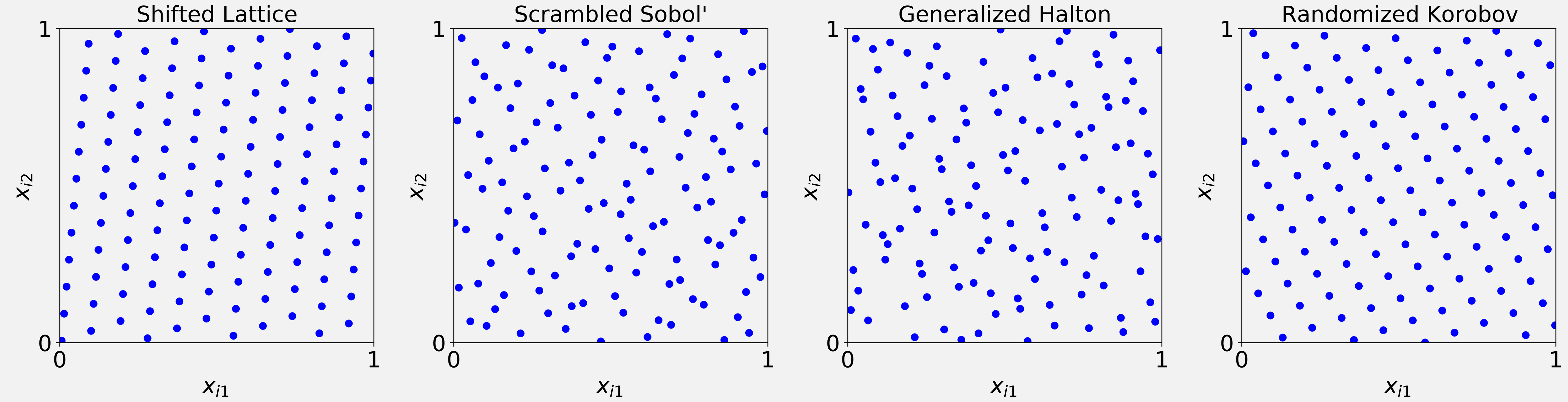
$$\mu = \int_{\mathbb{R}^d} g(\boldsymbol{t})\,\pi^{-d/2}\exp(-||\boldsymbol{t}||^2)\,\mathrm{d}\boldsymbol{t} = \int_{[0,1]^d} \underbrace{\pi^{d/2}\cos\left(||\boldsymbol{T}(\boldsymbol{x})||\right)}_{f(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x}$$

```
import qmcpy as qp
from numpy import *
d = 2
S = qp.Sobol(d)
G = qp.Gaussian(S,covariance=1/2)
K = qp.CustomFun(G,lambda t:
pi**(d/2)*cos(linalg.norm(t,1)))
CS = qp.CubQMCSobolG(K,abs_tol=1e-3)
solution,data = CS.integrate()
```





Low Discrepancy Discrete Distributions




```
>>> qp.Lattice(dimension=2, randomize=True).gen_samples(n=2**7)
```

Contributing Projects

- Guaranteed Automatic Integration Library [3]
- Quasi-Random Number Generators [4]
- P. Robbe's Multilevel MC and QMC [5]
- M. Giles' Multilevel MC and QMC [6,7]
- A. Owen's Halton Generator [8]
- LatNet Builder Generating Vectors [9]

References

1. S.-C. T. Choi, F. J. Hickernell, R. Jagadeeswaran, M. J. McCourt, A. G. Sorokin, "Quasi-Monte Carlo Software," Monte Carlo and Quasi-Monte Caro Methods 2020.
2. B. D. Keister, "Multidimensional Quadrature Algorithms," Computers in Physics 10, (1996).
3. S.-C. T. Choi, Y. Ding, F. J. Hickernell, L. Jiang, D. Li, R. Jagadeeswaran, L.-A. Jimenez Rugama, X. Tong, K. Zhang, Y. Zhang, and X. Zhou, "GAIL: Guaranteed Automatic Integration Library" (Version 2.3.1), MATLAB Software, 2020.
4. M. Hofert and C. Lemieux, "qrng: (Randomized) Quasi-Random Number Generators" R package version 0.0-7, (2019).
5. P. Robbe. "Multilevel Uncertainty Quantification Methods for Robust Design of Industrial Applications," PhD thesis, KU Leuven, 2019.
6. M. B. Giles. "Improved multilevel Monte Carlo convergence using the Milstein scheme," pp.343-358, in Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer, 2008.
7. M. B. Giles and B. J. Waterhouse. "Multilevel quasi-Monte Carlo path simulation," pp.165-181 in Advanced Financial Modelling, in Radon Series on Computational and Applied Mathematics, de Gruyter, 2009.
8. A. B. Owen. "A randomized Halton algorithm in R2017." arXiv:1706.02808 [stat.CO]
9. P. L'Ecuyer, P. Marion, M. Godin, and F. Puchhammer, "A Tool for Custom Construction of QMC and RQMC Point Sets," Monte Carlo and Quasi-Monte Carlo Methods 2020.