Community Supported Quasi-Monte Carlo (QMC) Software

Fred Hickernell (hickernell@iit.edu), Sou-Cheng Choi (schoi32@iit.edu), Aleksei Sorokin (asorokin@hawk.iit.edu)

Illinois Instute of Technology

Software Objectives

To provide QMC software that is:

- Comprised of free, open source, tools
- Designed for development and support
- The recognized standard
- Easy to use for non-experts

The QMC Problem

Original Problem

$$\int_T g(t)\lambda(t)dt \tag{1}$$

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

Convenient Form

$$\int_X f(x)\rho(x)dx = \int_X f(x)\hat{\gamma}(dx) \tag{2}$$

γ = well defined probability measure
$\phi : X \to T$ = change of variables
$f : X \to \mathbb{R}$ = integrand after change of variables

(quasi-)Monte Carlo Approximation

$$\sum_{i=1}^{n} f(x_i)w_i \tag{3}$$

$\hat{\gamma}_n \approx \gamma$ = discrete probability distribution
n guarantees

$$\left| \int_X f(x)\gamma(dx) - \sum_{i=1}^{n} f(x_i)w_i \right| \le \epsilon \tag{4}$$

Distribution on the Integration Domain

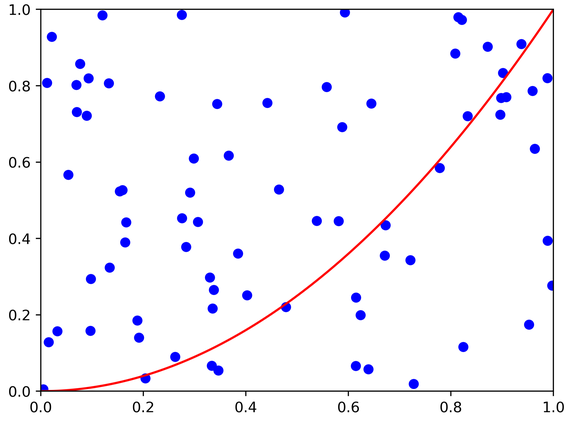


Figure 1: Uniform distribution for estimating $\int_0^1 x^2$

Integrate

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

Arguments

- funObj
- measureObj
- distribObj
- stopcritObj

Function

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

Concrete Classes

- Keister
- Asian Call

Discrete Distribution

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

Concrete Classes

- IID
- Mesh

Stopping Criterion

Finds n such that Equation (4) holds

Concrete Classes

- Central Limit Theorem (IID)
- Mean Variance (Mesh)

Measure

Specify components of a general sampling method

Implemented Functions

- Standard Uniform
- Standard Gaussian
- IID Zero Mean Gaussian
- Brownian Motion
- Lattice base 2
- Sobol base 2

Examples

Keister's Function

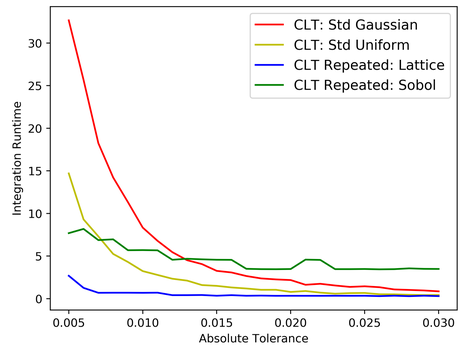
```
dim=3
funObj = KeisterFun()
stopObj = CLTStopping()
measureObj = measure().IIDZMeanGaussian(
    dimension=[dim],
    variance=[1/2])
distribObj = IIDDistribution(trueD =
    measure().stdGaussian(dimension=[dim]))
sol,out = integrate(
    funObj,
    measureObj,
    distribObj,
    stopObj)
```

Multilevel Asian Option Pricing

```
from numpy import arange
stopObj = CLT_Rep(nMax=2**20,absTol=.01)
measureObj = measure().BrownianMotion(
    timeVector = [
        arange(1/4,5/4,1/4),
        arange(1/16,17/16,1/16),
        arange(1/64,65/64,1/64)])
OptionObj = AsianCallFun(measureObj)
distribObj = Lattice(trueD =
    measure().mesh(
        dimension=[4,16,64],
        meshType='lattice'))
sol,out = integrate(
    OptionObj,
    measureObj,
    distribObj,
    stopObj)
```

Results

Integration Time Comparison



Future Work

- Enhance testing and examples library
- Incorporate existing components
- Expand community of contributors

References

1. S.-C. T. Choi, Y. Ding, F. J. Hickernell, L. Jiang, Ll. A. JimÃlnez Rugama, D. Li, R. Jagadeeswaran, X. Tong, K. Zhang,Y. Zhang, and X. Zhou,"GAIL: Guaranteed Automatic,Integration Library (versions 1.0-2.2).",http://gailgithub.github.io/GAIL_Dev/, MATLAB software, 2013-2019.

Acknowledgements

github.com/QMCSoftware/QMCSoftware.git
Other References and Aknowledgements
Larger ∫ and ∑ symbols
Colors and other styling
Better captions on figures
IIT QMC logo