Quasi-Monte Carlo vs Monte Carlo for High-Dimensional Integration on $[0, 1]^d$

Tanner Wagner
MATH 505: Numerical Analysis
University of New Mexico

November 29, 2025

Abstract

This project compares plain Monte Carlo (MC) and quasi-Monte Carlo (QMC) methods based on Sobol sequences for high-dimensional integrals over $[0, 1]^d$. This project looks at several smooth test functions with known exact integrals and compares the methods in terms of absolute error, empirical convergence rate, and practical time-to-accuracy for dimensions $d \in \{5, 10, 15, 20\}$.

Contents

1 Introduction

High-dimensional integrals appear in many applications, including expectations in probability and statistics, option pricing in finance, and posterior averages in Bayesian inference. In these settings one often needs to approximate integrals over a hypercube such as $[0,1]^d$. Classical deterministic quadrature methods (e.g., Gaussian or composite Newton–Cotes rules) are powerful in low dimensions, but their cost grows very rapidly with d if we use tensor-product constructions. This *curse of dimensionality* quickly makes standard quadrature prohibitively expensive.

Monte Carlo (MC) methods offer a dimension-robust alternative. By viewing an integral as the expectation of a function of a random variable, Monte Carlo approximates the integral by the average of N random samples. Under mild conditions, the error decays like $O(N^{-1/2})$, essentially independent of d in the exponent, but this convergence is relatively slow. Quasi–Monte Carlo (QMC) methods seek to improve on this by replacing random samples with deterministic low-discrepancy points (such as Sobol sequences) that are more uniformly distributed over $[0,1]^d$. For smooth integrands with suitable structure, QMC can achieve much smaller errors than MC for the same N, and heuristically exhibits an $O(N^{-1})$-like behavior up to logarithmic factors.

The goal of this project is to compare plain Monte Carlo and Sobol quasi-Monte Carlo for high-dimensional integration on $[0,1]^d$. We focus on three smooth, separable test integrands with known exact values, including a Gaussian-type integrand related to the normal distribution. For dimensions $d \in \{5, 10, 15, 20\}$ and a range of sample sizes N, we measure absolute error, empirical convergence rates, and practical time-to-accuracy for both methods.

The rest of the paper is organized as follows. Section 2 reviews the probabilistic background and formulates MC and QMC as quadrature rules. Section 3 describes the test integrands and experimental setup. Section 4 presents numerical comparisons of MC and QMC, and Section 5 summarizes the main findings and possible extensions.

2 Mathematical Background and Methods

2.1 High-dimensional integration on $[0,1]^d$

We are interested in approximating integrals of the form

$$I(f,d) = \int_{[0,1]^d} f(x)\,dx, \qquad x = (x_1,\dots,x_d) \in [0,1]^d.$$

Here $[0,1]^d$ denotes the d-dimensional unit cube, i.e., each coordinate x_j ranges between 0 and 1. When $d = 1$ this is a single integral over $[0,1]$; when $d = 2$ it is a double integral over the unit square; and in general it is a d-fold integral over the unit cube.

It is often convenient to interpret $I(f,d)$ as an *average value* of f over $[0,1]^d$. This viewpoint naturally leads us to a short review of basic probability concepts.

2.2 Brief probabilistic background

I'll briefly introduce the notions of random variables, expectation, and the law of large numbers that underlie Monte Carlo integration.

Random variables and expectation. A (real-valued) *random variable* Y is a quantity whose value is not fixed, but instead is determined by some random experiment. For example, Y could represent the outcome of rolling a die, or the value of a function evaluated at a randomly chosen point.

The *expectation* (or expected value) of Y, denoted $\mathbb{E}[Y]$, is the average value we would obtain if we repeated the experiment many times and averaged the outcomes. In the continuous setting, if Y has a probability density function $p(y)$, then

$$\mathbb{E}[Y] = \int_{-\infty}^{\infty} y\, p(y)\, dy.$$

More generally, for any function $g(Y)$ we define

$$\mathbb{E}[g(Y)] = \int_{-\infty}^{\infty} g(y)\, p(y)\, dy.$$

In our setting, we'll consider a random vector

$$X = (X_1, \dots, X_d)$$

that is *uniformly distributed* on $[0,1]^d$. This means that every region of $[0,1]^d$ with the same volume is equally likely. The corresponding density $p(x)$ is constant on $[0,1]^d$ and zero outside. In this special case, the expectation of $f(X)$ is

$$\mathbb{E}[f(X)] = \int_{[0,1]^d} f(x)\, dx = I(f,d).$$

Thus the integral $I(f,d)$ we wish to approximate is exactly the expected value of $f(X)$ when X is uniformly distributed on $[0,1]^d$.

Variance and standard deviation. The *variance* of a random variable Y is defined by

$$\mathrm{Var}(Y) = \mathbb{E}\big[(Y - \mathbb{E}[Y])^2\big],$$

and measures the spread of Y around its mean. The *standard deviation* is $\sqrt{\mathrm{Var}(Y)}$ and has the same units as Y. Intuitively, the standard deviation is a measure of the typical size of the deviation $Y - \mathbb{E}[Y]$.

Law of large numbers (LLN). Suppose $Y_1, Y_2, \ldots$ are independent copies of the same random variable Y (we write Y_i are i.i.d.). Consider the sample average

$$\overline{Y}_N = \frac{1}{N}\sum_{i=1}^{N} Y_i.$$

The *law of large numbers* states that, under mild assumptions,

$$\overline{Y}_N \to \mathbb{E}[Y] \quad \text{as } N \to \infty,$$

i.e., the sample average converges to the expected value. Intuitively, if we take more and more samples, their average stabilizes near the true mean.

In our application, we'll take $Y_i = f(X_i)$, where X_i are random points in $[0,1]^d$ chosen independently and uniformly. The law of large numbers then tells us that the average of the values $f(X_i)$ converges to the integral $I(f,d)$.

Fluctuations and the $N^{-1/2}$ scaling. We are not only interested in convergence, but also in *how fast* the sample average approaches its mean. A key quantity is the fluctuation

$$\overline{Y}_N - \mathbb{E}[Y],$$

which measures how far the sample average wanders away from the true mean for a typical sample.

A simple variance calculation already reveals the $N^{-1/2}$ scaling. If $\mathrm{Var}(Y) = \sigma^2 < \infty$ and Y_i are i.i.d., then

$$\mathrm{Var}(\overline{Y}_N) = \mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{N} Y_i\right) = \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{Var}(Y_i) = \frac{1}{N^2}\cdot N\sigma^2 = \frac{\sigma^2}{N}.$$

Therefore the standard deviation of the sample average is

$$\sqrt{\mathrm{Var}(\overline{Y}_N)} = \frac{\sigma}{\sqrt{N}}.$$

Since the standard deviation is a measure of the typical size of $\overline{Y}_N - \mathbb{E}[Y]$, we see that

$$\left|\overline{Y}_N - \mathbb{E}[Y]\right| \text{ is typically of size } O(N^{-1/2}).$$

Central limit theorem (CLT). The *central limit theorem* refines this by describing the shape of the distribution of the error. Under mild conditions,

$$\sqrt{N}\left(\overline{Y}_N - \mathbb{E}[Y]\right) \Rightarrow \mathcal{N}(0,\sigma^2),$$

where $\Rightarrow$ denotes convergence in distribution and $\mathcal{N}(0, \sigma^2)$ is a normal (Gaussian) distribution with mean zero and variance σ^2. This again shows that the natural scale of the error $\overline{Y}_N - \mathbb{E}[Y]$ is $1/\sqrt{N}$, consistent with the variance calculation above.

In the Monte Carlo setting we will take $Y_i = f(X_i)$, so the Monte Carlo error is exactly a sample average minus a mean, and its typical size is of order $N^{-1/2}$.

2.3 Quadrature rules and Monte Carlo integration

Any numerical integration method can be viewed as replacing the continuous integral by a finite weighted sum of function values. More precisely, a *quadrature rule* with N points has the form

$$Q_N(f) = \sum_{i=1}^{N} w_i \, f(x_i),$$

where $x_i \in [0, 1]^d$ are the nodes and $w_i \in \mathbb{R}$ are weights. Classical deterministic quadrature rules (e.g., trapezoidal or Gaussian quadrature in one dimension, and tensor-product versions in low dimensions) correspond to carefully chosen deterministic nodes and weights.

Monte Carlo and quasi–Monte Carlo methods also fit this template, but with different choices of nodes. In particular, Monte Carlo uses *random* nodes, while quasi–Monte Carlo uses *deterministic* nodes that are designed to be evenly spread over the integration domain.

2.4 Plain Monte Carlo integration

Plain Monte Carlo (MC) integration applies the ideas from the previous subsection directly. We draw random points

$$X_1, \ldots, X_N \sim \mathrm{Unif}([0, 1]^d)$$

independently and identically distributed, and approximate the integral by the sample average

$$\widehat{I}_N^{\mathrm{MC}}(f, d) = \frac{1}{N} \sum_{i=1}^{N} f(X_i).$$

This is a quadrature rule of the form

$$Q_N(f) = \sum_{i=1}^{N} w_i f(x_i),$$

with random nodes $x_i = X_i$ and equal weights $w_i = 1/N$.

Because $\mathbb{E}[f(X_i)] = I(f, d)$, the estimator is unbiased:

$$\mathbb{E}\big[\widehat{I}_N^{\mathrm{MC}}(f, d)\big] = I(f, d).$$

Moreover, the law of large numbers implies that

$$\widehat{I}_N^{\mathrm{MC}}(f,d) \to I(f,d) \qquad \text{as } N \to \infty,$$

so the Monte Carlo approximation converges to the true integral when we take more and more random points.

The variance calculation from the previous subsection can be applied with $Y_i = f(X_i)$ and $\overline{Y}_N = \widehat{I}_N^{\mathrm{MC}}(f,d)$. If $\mathrm{Var}(f(X)) = \sigma^2 < \infty$, then

$$\mathrm{Var}\big(\widehat{I}_N^{\mathrm{MC}}(f,d)\big) = \frac{\sigma^2}{N},$$

and the standard deviation (a measure of the typical size of the error) is

$$\sqrt{\mathrm{Var}\big(\widehat{I}_N^{\mathrm{MC}}(f,d)\big)} = \frac{\sigma}{\sqrt{N}}.$$

Thus the Monte Carlo error

$$\widehat{I}_N^{\mathrm{MC}}(f,d) - I(f,d)$$

has typical magnitude of order $N^{-1/2}$. In the language of big-O notation, the root mean square error (RMSE) satisfies

$$\sqrt{\mathbb{E}\big[(\widehat{I}_N^{\mathrm{MC}} - I)^2\big]} = O(N^{-1/2}),$$

with a constant that depends on the variance of $f(X)$ and may grow with the dimension d, but the exponent $-1/2$ itself does not depend on d. This $N^{-1/2}$ scaling is a hallmark of Monte Carlo methods.

2.5 Quasi-Monte Carlo with Sobol sequences

Quasi-Monte Carlo (QMC) methods replace the random nodes of Monte Carlo with a deterministic sequence of points that is designed to fill the unit cube $[0,1]^d$ as uniformly as possible. Let $(x_i)_{i\geq 1}$ be a *low-discrepancy sequence* in $[0,1]^d$, such as a Sobol sequence. The corresponding QMC estimator is

$$\widehat{I}_N^{\mathrm{QMC}}(f,d) = \frac{1}{N}\sum_{i=1}^{N} f(x_i).$$

This again has the quadrature form $Q_N(f) = \frac{1}{N}\sum f(x_i)$, but now the nodes x_i are carefully constructed rather than random.

The key quantity in QMC analysis is the (star) discrepancy $D^*(x_1,\ldots,x_N)$ of the first

N points, which measures how far the empirical distribution of the points deviates from the uniform distribution on $[0,1]^d$. Lower discrepancy means the points are more uniformly spread out.

For functions of bounded variation in the sense of Hardy–Krause, the Koksma-Hlawka inequality states that

$$\left|\widehat{I}_N^{\mathrm{QMC}}(f,d) - I(f,d)\right| \le V_{\mathrm{HK}}(f)\, D^*(x_1,\ldots,x_N),$$

where $V_{\mathrm{HK}}(f)$ is the (Hardy-Krause) variation of f. For many low-discrepancy sequences (including Sobol), the discrepancy satisfies

$$D^*(x_1,\ldots,x_N) = O\left(\frac{(\log N)^d}{N}\right) \qquad \text{as } N \to \infty.$$

Ignoring the logarithmic factor $(\log N)^d$, this suggests that QMC can achieve an error that behaves like $O(N^{-1})$ for sufficiently smooth integrands with "low effective dimension," which is asymptotically better than the Monte Carlo rate $O(N^{-1/2})$.

In practice, QMC methods often show substantially smaller errors than MC for the same N on smooth, well-behaved integrands, especially in moderate dimensions. When the dimension is very high or the integrand has significant discontinuities or strong localized features, the advantage of QMC can diminish.

2.6 Error metrics and comparison criteria

For any method (MC or QMC) that produces an approximation $\widehat{I}_N(f,d)$ to the integral $I(f,d)$, we define the absolute error

$$E_N(f,d) = \left|\widehat{I}_N(f,d) - I(f,d)\right|,$$

where $I(f,d)$ is the exact integral, known in closed form for our chosen test functions. We will compare MC and QMC using three related criteria:

1. **Error versus sample size:** we will plot $E_N(f,d)$ versus N on logarithmic scales and estimate empirical convergence rates by fitting a slope to $\log_{10}(E_N)$ versus $\log_{10}(N)$ for each method.

2. **Error versus wall-clock time:** we will record the computation time required to obtain each approximation and compare time-to-accuracy by plotting $E_N(f,d)$ against wall-clock time.

3. **Variance and stability:** when repeated runs are performed (for Monte Carlo and, optionally, randomized QMC), we will summarize the distribution of errors (for example, via mean and standard deviation) to assess the stability of the estimators.

This framework allows us to quantify both the asymptotic behavior (convergence rates) and the practical performance (time-to-accuracy) of plain Monte Carlo and Sobol quasi-Monte Carlo integration on high-dimensional test problems.

3 Test Integrands and Experimental Setup

In this section we describe the specific test problems and experimental parameters used to compare Monte Carlo and quasi–Monte Carlo integration on high-dimensional integrals over $[0,1]^d$.

3.1 Dimensions and general problem statement

For each experiment we fix a dimension

$$d \in \{5, 10, 15, 20\},$$

and consider integrals of the form

$$I(f,d) = \int_{[0,1]^d} f(x)\,dx, \qquad x = (x_1, \ldots, x_d) \in [0,1]^d.$$

The goal is to approximate $I(f,d)$ using both plain Monte Carlo (MC) and Sobol quasi–Monte Carlo (QMC), and to measure the resulting errors and computation times.

3.2 Test integrands with known exact values

To evaluate the accuracy of MC and QMC, we choose test functions $f : [0,1]^d \to \mathbb{R}$ for which the integral $I(f,d)$ can be computed in closed form (possibly involving special functions). This allows us to compute the *true error* $\left|\widehat{I}_N(f,d) - I(f,d)\right|$ for each method and each configuration.

All of the test integrands we use are *separable* in the sense that they factor into a product of one-dimensional functions,

$$f(x) = \prod_{j=1}^{d} g_j(x_j),$$

so that the d-dimensional integral reduces to a product of one-dimensional integrals,

$$I(f,d) = \int_{[0,1]^d} f(x)\,dx = \prod_{j=1}^{d} \int_0^1 g_j(x_j)\,dx_j.$$

This structure makes it straightforward to obtain exact values while still providing a variety of behaviors for the numerical methods.

We work with three test integrands, denoted f_A, f_B, and f_C.

Integrand A (separable exponential). Our first test integrand is a symmetric exponential:

$$f_A(x) = \exp\Bigl(-\sum_{j=1}^{d} x_j\Bigr) = \prod_{j=1}^{d} e^{-x_j}.$$

The exact integral is

$$I(f_A, d) = \int_{[0,1]^d} f_A(x)\,dx = \prod_{j=1}^{d} \int_0^1 e^{-x_j}\,dx_j = \prod_{j=1}^{d} \bigl(1 - e^{-1}\bigr) = \bigl(1 - e^{-1}\bigr)^d.$$

This integrand is smooth, bounded, and treats all coordinates symmetrically.

Integrand B (rational integrand with arctan closed form). The second test integrand is a smooth rational function. For $j = 1, \dots, d$ define

$$g_j(x_j) = \frac{1}{1 + j x_j^2},$$

and set

$$f_B(x) = \prod_{j=1}^{d} g_j(x_j) = \prod_{j=1}^{d} \frac{1}{1 + j x_j^2}.$$

Each one-dimensional factor has an explicit antiderivative:

$$\int_0^1 \frac{1}{1 + j x_j^2}\,dx_j = \frac{1}{\sqrt{j}} \arctan\bigl(\sqrt{j}\,x_j\bigr)\Big|_{x_j=0}^{x_j=1} = \frac{1}{\sqrt{j}} \arctan\bigl(\sqrt{j}\bigr),$$

so that

$$I(f_B, d) = \prod_{j=1}^{d} \frac{1}{\sqrt{j}} \arctan\bigl(\sqrt{j}\bigr).$$

This integrand is smooth but has a different shape from the exponential example: it is close to 1 near $x_j = 0$ and decays as x_j increases, with the rate depending on j. The dependence on j introduces a form of anisotropy, since directions corresponding to larger j are damped more strongly.

Integrand C (Gaussian-type integrand). The third test integrand is inspired by the Gaussian (normal) distribution. We define

$$f_C(x) = \exp\Bigl(-\sum_{j=1}^{d} x_j^2\Bigr) = \prod_{j=1}^{d} e^{-x_j^2}.$$

The corresponding one-dimensional integral

$$\int_0^1 e^{-x^2}\,dx$$

does not have an elementary antiderivative, but it is a classical integral that can be expressed in terms of the *error function*, defined by

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2}\, dt.$$

From this definition we obtain

$$\int_0^1 e^{-x^2}\, dx = \frac{\sqrt{\pi}}{2}\operatorname{erf}(1).$$

Therefore, by separability,

$$I(f_C, d) = \int_{[0,1]^d} f_C(x)\, dx = \prod_{j=1}^{d} \int_0^1 e^{-x_j^2}\, dx_j = \left(\int_0^1 e^{-x^2}\, dx\right)^d = \left(\frac{\sqrt{\pi}}{2}\operatorname{erf}(1)\right)^d.$$

This integrand is smooth and strongly related to the Gaussian integral that underlies the normal distribution, providing a natural probabilistic connection in our test set.

Together, f_A, f_B, and f_C provide a small but diverse collection of high-dimensional test problems. All three are smooth and separable, but they differ in symmetry, anisotropy, and decay behavior.

3.3 Sample sizes and repetitions

For each integrand $f \in \{f_A, f_B, f_C\}$, each dimension $d \in \{5, 10, 15, 20\}$, and each numerical method (MC or QMC), we consider a sequence of sample sizes

$$N \in \{2^7, 2^8, \ldots, 2^{15}\} = \{128, 256, 512, \ldots, 32768\}.$$

This range allows us to observe the convergence behavior over several orders of magnitude in N.

For *plain Monte Carlo*, the estimator $\widehat{I}_N^{\mathrm{MC}}(f, d)$ is random, so we perform

$$R = 30$$

independent runs for each choice of (f, d, N). This yields an ensemble of errors, from which we can compute statistics such as the mean error and standard deviation.

For *Sobol quasi–Monte Carlo*, the estimator $\widehat{I}_N^{\mathrm{QMC}}(f, d)$ is deterministic once f, d, and N are fixed, so in the simplest version we perform a single run for each configuration. Our framework also allows for *randomized* Sobol sequences (scrambled QMC), in which case we could again take R repeated runs to estimate a variance; however, in this project we primarily focus on the deterministic Sobol case.

3.4 Error computation

For each method, integrand, dimension, and sample size we obtain an approximation $\widehat{I}_N(f,d)$ to the exact integral $I(f,d)$. The *absolute error* is defined by

$$E_N(f,d) = \left|\widehat{I}_N(f,d) - I(f,d)\right|.$$

In the Monte Carlo case, the estimator $\widehat{I}_N^{\mathrm{MC}}(f,d)$ is random, so repeated runs with different random seeds produce different approximations and hence different errors. For a fixed configuration (f,d,N) we perform R independent runs, yielding a collection of error values

$$E_N^{(1)}(f,d),\ E_N^{(2)}(f,d),\ \ldots,\ E_N^{(R)}(f,d).$$

It is natural to think of these as R samples from an underlying random variable "error" at sample size N.

From these R samples we compute two basic summary statistics:

- the *sample mean error*

  $$\overline{E}_N(f,d) = \frac{1}{R}\sum_{r=1}^{R} E_N^{(r)}(f,d),$$

  which is simply the average of the observed errors and serves as an estimate of the typical error size at that N;

- the *sample standard deviation* of the error

  $$\mathrm{std}_N(f,d) = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}\left(E_N^{(r)}(f,d) - \overline{E}_N(f,d)\right)^2}.$$

The quantity inside the square root is the *sample variance* with denominator $R-1$ rather than R; this choice (sometimes called the Bessel correction) makes the sample variance an unbiased estimator of the true variance of the error when the $E_N^{(r)}$ are independent samples. The sample standard deviation $\mathrm{std}_N(f,d)$ therefore provides an estimate of the typical fluctuation of the Monte Carlo error around its mean.

For deterministic QMC, the estimator $\widehat{I}_N^{\mathrm{QMC}}(f,d)$ (and hence the error $E_N(f,d)$) is fully determined by f, d, and N, so we simply report the single absolute error for each configuration.

3.5 Timing measurements

In addition to accuracy, we are interested in the practical cost of each method. For each run we measure the wall-clock time required to *generate the sample points* and *evaluate the integrand* on those points. We denote this time by $T_N(f,d)$.

By plotting the error $E_N(f,d)$ against the corresponding time $T_N(f,d)$, we obtain *time-to-accuracy* plots that show, for example, which method reaches a target error (e.g. 10^{-3}) in less time for a given integrand and dimension.

3.6 Summary of comparison metrics

Summarizing, for each integrand f, dimension d, method (MC or QMC), and sample size N, we collect:

- absolute errors $E_N(f, d)$ (and, for MC, averages and standard deviations over repeated runs),
- corresponding wall-clock times $T_N(f, d)$.

We will use these data to construct:

1. log–log plots of $E_N(f, d)$ versus N to estimate empirical convergence rates for MC and QMC,
2. plots of $E_N(f, d)$ versus $T_N(f, d)$ to compare time-to-accuracy,
3. (optionally) error bars or standard deviations for Monte Carlo to illustrate the variability of the estimator.

These diagnostics allow us to evaluate both the asymptotic behavior and the practical performance of plain Monte Carlo and Sobol quasi–Monte Carlo integration on the chosen high-dimensional test problems.

4 Numerical Results

Here you will insert tables and figures generated from your Python code.

4.1 Error versus sample size

For each integrand and dimension, show plots of $\log_{10}(E_N)$ vs $\log_{10}(N)$ for MC and QMC. Discuss:

- Observed slopes (roughly $-1/2$ for MC, closer to -1 for QMC in favorable cases).
- How dimension affects performance.

4.2 Error versus time (time-to-accuracy)

Show plots of error vs wall-clock time for MC and QMC. Comment on which method reaches a given target accuracy (e.g. 10^{-3}) faster and how this changes with d.

4.3 Variance and stability (if applicable)

If you run repeated experiments, summarize the spread of errors (e.g., error bars or standard deviations) for MC and randomized QMC.

5 Discussion and Conclusions

Summarize the main findings:

- When and why QMC outperforms MC for your test problems.
- How the observed rates compare with theoretical expectations.
- How performance changes as d increases.

Mention limitations and possible extensions:

- Non-smooth integrands, discontinuities, or very high dimensions.
- Other low-discrepancy sequences (Halton, lattice rules, etc.).
- Importance sampling or variance reduction techniques.

Acknowledgments

(Optional) Acknowledge your professor, classmates, or any resources that were particularly helpful.

References

[1] H. Niederreiter, *Random Number Generation and Quasi–Monte Carlo Methods*, SIAM, 1992.

[2] C. Lemieux, *Monte Carlo and Quasi–Monte Carlo Sampling*, Springer, 2009.

A Python Code

Include your main scripts here (e.g., integrand definitions, MC/QMC implementations, and plotting routines).

B Additional Tables and Figures

Any extra numerical tables, convergence plots, or diagnostics that do not fit in the main text.