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 30, 2025

Outline

High-dimensional integrals

- Many applications involve integrals over $[0,1]^d$:
  - expectations in probability and statistics,
  - option pricing in mathematical finance,
  - posterior averages in Bayesian inference.
- Classical tensor-product quadrature (Gaussian, Newton–Cotes):
  - work grows exponentially with d,
  - the *curse of dimensionality*.
- Need methods whose cost scales more gently with dimension.

Goal of the project

- Compare **plain Monte Carlo (MC)** with **Sobol quasi–Monte Carlo (QMC)** for integrals

$$I(f, d) = \int_{[0,1]^d} f(x)\, dx, \quad d \in \{5, 10, 15, 20\}.$$

- Test on several smooth, high-dimensional integrands with known exact values.
- Metrics:
  - absolute error,
  - empirical convergence rate,
  - **time-to-accuracy**.
- Question: *When does QMC actually beat MC in practice?*

Integral as an expectation

- Let $X = (X_1, \ldots, X_d)$ be uniform on $[0,1]^d$.
- Then

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

- Approximate $I(f,d)$ by a sample mean:

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

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

- This is the basic idea of Monte Carlo integration.

Monte Carlo: error scaling

- Assume $\mathrm{Var}(f(X)) = \sigma^2 < \infty$.
- Then

$$\mathrm{Var}(\widehat{I}_N) = \mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{N} f(X_i)\right) = \frac{\sigma^2}{N}.$$

- Standard deviation (typical error size):

$$\sqrt{\mathrm{Var}(\widehat{I}_N)} = \frac{\sigma}{\sqrt{N}}.$$

- **Key point:**

$$\text{MC error} = O\big(N^{-1/2}\big),$$

with an exponent independent of d (though σ may depend on d).

Quasi-Monte Carlo (QMC)

- Replace random points X_i by a *low-discrepancy sequence* $(x_i)_{i\geq 1} \subset [0,1]^d$:

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

- Sobol sequence: base-2 digital net designed to fill $[0,1]^d$ very uniformly.
- For functions of bounded variation (Hardy–Krause), Koksma–Hlawka inequality gives

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

where D^* is the star-discrepancy.
- For Sobol and related sequences,

$$D^*(x_1,\dots,x_N) = O\left(\frac{(\log N)^d}{N}\right).$$

MC vs QMC: heuristic comparison

- Ignoring $(\log N)^d$:

$$\text{MC: } O(N^{-1/2}), \qquad \text{QMC: } O(N^{-1}).$$

- In practice, QMC often gives

$$\text{much smaller error than MC for the same } N$$

  when
  - f is smooth,
  - the *effective dimension* is moderate.
- But constants and effective dimension matter a lot in high d.
- This is exactly what we explore numerically.

Error metrics

For each method (MC or QMC) we consider:

- **Absolute error**

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

- **Error vs sample size**:
  - log–log plots of E_N vs N,
  - estimate slopes (empirical convergence rates).
- **Error vs wall-clock time**:
  - time-to-accuracy plots E_N vs T_N,
  - compare which method reaches a target tolerance faster.
- For MC, we also look at variability over repeated runs.

Dimensions and sample sizes

- Dimensions:
$$d \in \{5, 10, 15, 20\}.$$
- Sample sizes:
$$N \in \{2^7, 2^8, \ldots, 2^{15}\} = \{128, 256, \ldots, 32768\}.$$
- For each (f, d, N):
  - MC: $R = 30$ independent runs,
  - QMC: single Sobol net of size N (via `random_base2` in SciPy).

Separable test integrands

- We choose $f : [0, 1]^d \to \mathbb{R}$ with closed-form integrals.
- First three integrands are *separable*:

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

Then

$$I(f, d) = \prod_{j=1}^{d} \int_0^1 g_j(x_j)\, dx_j.$$

- This makes exact values cheap to compute while still giving nontrivial high-dimensional behavior.

Integrand A: separable exponential

- Define

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

- Exact integral:

$$I(f_A, d) = \prod_{j=1}^{d} \big(1 - e^{-1}\big) = \big(1 - e^{-1}\big)^d.$$

- Smooth, bounded, symmetric in all coordinates.

Integrand B: rational with arctan closed form

- Define
$$f_B(x) = \prod_{j=1}^{d} \frac{1}{1 + jx_j^2}.$$
- 1D factor:
$$\int_0^1 \frac{1}{1 + jx^2}\, dx = \frac{1}{\sqrt{j}} \arctan(\sqrt{j}).$$
- So
$$I(f_B, d) = \prod_{j=1}^{d} \frac{1}{\sqrt{j}} \arctan(\sqrt{j}).$$
- Smooth, but anisotropic: different behavior in different coordinate directions.

Integrand C: Gaussian-type

- Define

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

- Use the error function:

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

- Hence

$$I(f_C, d) = \left(\frac{\sqrt{\pi}}{2}\,\mathrm{erf}(1)\right)^d.$$

- Smooth, strongly related to Gaussian integrals.

Integrand D: low effective dimension

- Define, for $d \geq 5$,

$$f_D(x) = \left(\frac{1}{5} \sum_{j=1}^{5} x_j \right)^2 .$$

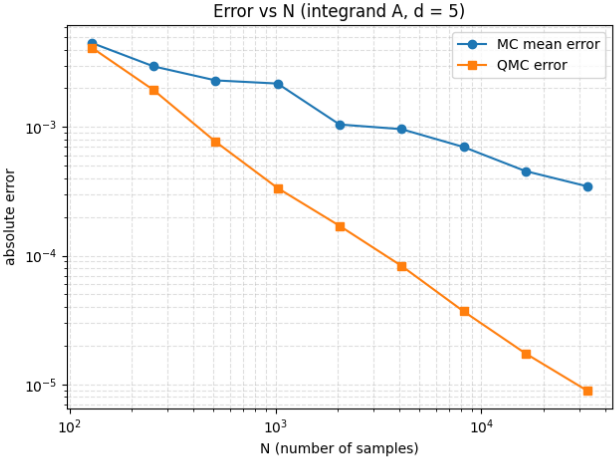
- Depends only on first five coordinates $\Rightarrow$ *low effective dimension*.
- Let $X_1, \ldots, X_5 \sim \mathrm{Unif}(0,1)$ independent, $S = X_1 + \cdots + X_5$.
- Then

$$I(f_D, d) = \mathbb{E}\left[\left(\frac{S}{5} \right)^2 \right] = \frac{1}{25} \left(\mathrm{Var}(S) + (\mathbb{E}S)^2 \right) = \frac{4}{15},$$

  independent of d.
- Designed to be especially favorable to QMC.

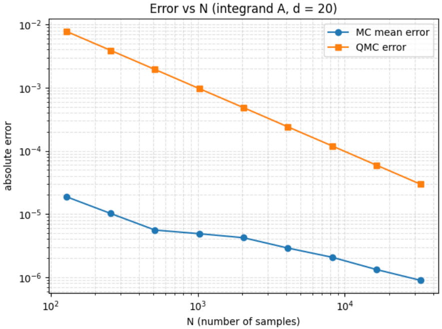
Error vs N: moderate dimension ($d = 5$)

- Example: f_A in $d = 5$.

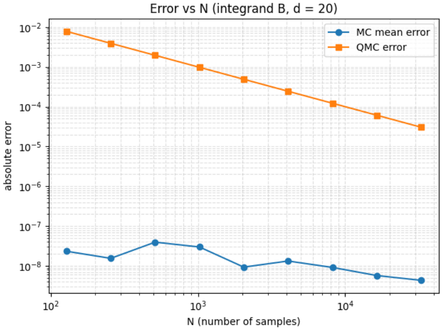


- QMC curve lies below MC mean error for moderate and large N.
- On log–log scale:

Error vs N: high dimension ($d = 20$)



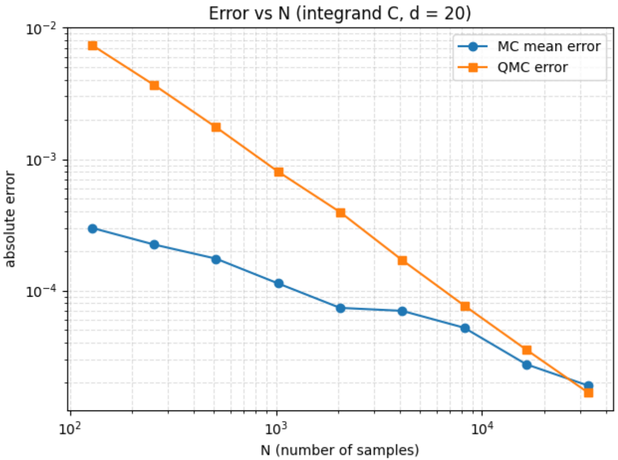
f_A in $d = 20$



f_B in $d = 20$

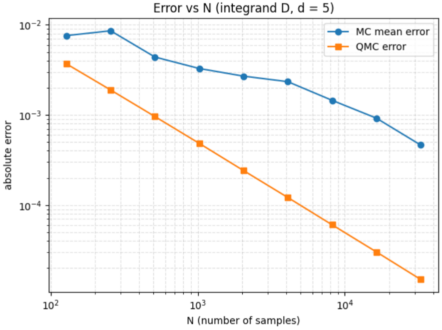
- QMC still shows $\approx N^{-1}$-like decay but with a *large constant*.
- MC errors start much smaller and stay below QMC for all N tested.

Error vs N: Gaussian-type f_C in $d = 20$

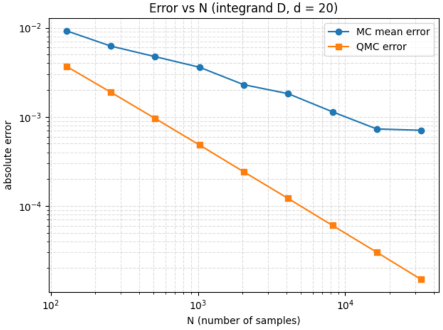


- For small N: MC error $<$ QMC error (pre-asymptotic regime).
- As N grows: QMC decays faster, eventually catching up and slightly overtaking MC.

Error vs N: low effective dimension f_D



f_D in $d = 5$



f_D in $d = 20$

- QMC clearly dominates MC in both $d = 5$ and $d = 20$.
- QMC error curves in $d = 5$ and $d = 20$ are very similar: effective dimension ≈ 5.
- Textbook example of QMC's $O(N^{-1})$-type behavior.

Error vs time: representative examples

- For each configuration, we measure wall-clock time T_N.
- Plots of E_N vs T_N mirror the E_N vs N behavior:
  - When QMC has smaller error at fixed N (e.g. f_A in $d = 5$, f_D), it also reaches a target tolerance faster.
  - When MC dominates in error (e.g. f_A and f_B in $d = 20$), MC is also more efficient in time.
- Cost per sample is similar for MC and QMC, so accuracy differences translate directly into time-to-accuracy.

Summary of findings

- **Moderate dimension ($d = 5$):**
  - QMC outperforms MC on all three separable integrands f_A, f_B, f_C.
  - Empirical slopes: QMC ≈ -1, MC $\approx -1/2$.
- **High dimension ($d = 20$):**
  - For f_A and f_B (very small exact integrals), MC has smaller error than QMC for all N.
  - For Gaussian-type f_C, MC is better for small N, QMC catches up at larger N.
- **Low effective dimension (f_D):**
  - QMC clearly wins in both $d = 5$ and $d = 20$.
  - Error curves essentially independent of the nominal dimension.

Rates, constants, and effective dimension

- Asymptotic statements

$$\text{MC } O(N^{-1/2}), \quad \text{QMC } O(N^{-1})$$

  hide important constants.
- For f_A and f_B in $d = 20$:
  - exact integrals are extremely small,
  - MC errors inherit this small scale via the variance,
  - QMC behaves more like C/N with $C \approx O(1)$.
- Effective dimension matters:
  - when most variation is in a few coordinates (like f_D), QMC works extremely well,
  - when variation is spread across many coordinates, the benefit can be muted or delayed.

Practical guidelines

- Use Sobol QMC when:
  - the integrand is smooth,
  - effective dimension is moderate or low,
  - you can afford moderately large N.
- Plain MC can be competitive or better when:
  - dimension is very high,
  - exact integrals / variances are extremely small,
  - you are restricted to relatively small N.
- Overall message:

"QMC is great, but problem structure really matters."

Python implementation (Code/)

- `integrands.py`
  - definitions of f_A, f_B, f_C, f_D,
  - closed-form formulas for $I(f, d)$.
- `mc_qmc.py`
  - plain MC estimator using NumPy random sampling,
  - Sobol QMC estimator using `scipy.stats.qmc.Sobol`.
- `run_experiments.py`
  - loops over f, d, N,
  - performs $R = 30$ MC runs and 1 QMC run,
  - writes results to `results_mc_qmc.csv`.
- `plot_results.py`
  - reads `results_mc_qmc.csv`,
  - generates the error-vs-N and error-vs-time plots.

References

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

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

A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, 2nd ed., Springer, 2007.

A. B. Owen, *Monte Carlo Theory, Methods and Examples*, 2013. Available online.

Questions?