Numerical Integration over the Unit Ball in $\mathbb{R}^3$

Deterministic, Monte Carlo, and Symmetrized Monte Carlo Methods

Minou Emmad

December 10, 2025

Abstract

This report goes over the numerical methods for computing integrals over the d-dimensional unit ball, with emphasis on the three-dimensional case required by the project. I implement and compare: (1) deterministic spherical-coordinate integration, (2) standard Monte Carlo (MC) sampling, (3) rejection sampling MC, and (4) a symmetrized Monte Carlo estimator using all 2^d sign-flip symmetries.

Experiments are performed for $d = 2, 3, 4$, with detailed error analysis for the 3D project integrand $f(x, y, z) = (1 + x^2 + y^2)e^z - x/(1 + z^2)$. All results are reproducible, with full user documentation, developer documentation, and tested code included separately.

1 Introduction

Computing high-dimensional integrals is challenging because classical numerical quadrature rules scale exponentially with dimension. Monte Carlo (MC) methods avoid this and converge at the dimension-independent rate $O(N^{-1/2})$, with N the number of random samples.

This project focuses on integration over the unit ball

$$B = \{x \in \mathbb{R}^3 : \|x\| \leq 1\}.$$

2 Task 1: Monte Carlo Expectation and Variance

Let X be a point chosen uniformly at random from the unit ball $B \subset \mathbb{R}^3$. This means every region of equal volume inside the ball is equally likely.

The uniform probability measure on B is

$$d\pi(x) = \frac{1}{\mathrm{vol}(B)}\, dx, \qquad \mathrm{vol}(B) = \frac{4\pi}{3}.$$

Expectation

The expected value of $f(X)$ is just the average value of f over the ball:

$$\mathbb{E}[f(X)] = \frac{1}{\mathrm{vol}(B)} \int_B f(x)\,dx.$$

Multiplying both sides by the volume of the ball shows that

$$\int_B f(x)\,dx = \mathrm{vol}(B)\,\mathbb{E}[f(X)].$$

Variance

The variance of $f(X)$ measures how much f fluctuates inside the ball:

$$\mathrm{Var}[f(X)] = \mathbb{E}\left[(f(X) - \mathbb{E}[f(X)])^2\right].$$

Monte Carlo estimator

If we draw N independent uniform samples $X_1, \ldots, X_N$ in the ball, the Monte Carlo estimator for the full integral is

$$S_N = \mathrm{vol}(B)\,\frac{1}{N}\sum_{i=1}^{N} f(X_i).$$

Expectation

Using linearity of expectation,

$$\mathbb{E}[S_N] = \mathrm{vol}(B)\,\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[f(X_i)] = \mathrm{vol}(B)\,\mathbb{E}[f(X)].$$

Since

$$\mathbb{E}[f(X)] = \frac{1}{\mathrm{vol}(B)} \int_B f(x)\,dx,$$

This implies that

$$\mathbb{E}[S_N] = \int_B f(x)\,dx.$$

Thus the estimator is unbiased.

Variance

Assume

$$A = \mathrm{vol}(B), \qquad Y_i = f(X_i).$$

Then

$$S_N = A \cdot \frac{1}{N}\sum_{i=1}^{N} Y_i.$$

Using the rule $\mathrm{Var}(cZ) = c^2\,\mathrm{Var}(Z)$,

$$\mathrm{Var}(S_N) = A^2\ \mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{N} Y_i\right).$$

Because the Y_i are independent and identically distributed,

$$\mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{N} Y_i\right) = \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{Var}(Y_i) = \frac{1}{N^2}\big(N\ \mathrm{Var}(f(X))\big) = \frac{1}{N}\,\mathrm{Var}(f(X)).$$

Putting everything together,

$$\mathrm{Var}(S_N) = A^2 \cdot \frac{1}{N}\,\mathrm{Var}(f(X)) = \frac{\mathrm{vol}(B)^2}{N}\,\mathrm{Var}(f(X)).$$

This shows explicitly that the Monte Carlo estimator has variance proportional to $1/N$, so its error decreases at the standard rate

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

Thus the Monte Carlo standard error scales like

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

which is the dimension-independent convergence rate characteristic of Monte Carlo methods.

My deterministic integrator uses a uniform $m \times m \times m$ product grid, implemented in `spherical_grid_integral_equal_m`. For the 3D project integrand, I use $m = 80$ to generate a high-accuracy reference value:

$$I_{\mathrm{ref}}(d = 3) \approx 6.628895.$$

3 Task 3: Monte Carlo Sampling Methods

Both methods for generating uniform samples inside the unit d-ball were tested for $d = 2, 3, 4$ and the results agree with the theoretical sampling distributions.

3.1 Direct (Non-Rejecting) Sampling in B_d

To generate truly uniform points in the ball, I use the standard method:

1. Sample $Z \sim N(0, I_d)$.
2. Normalize to obtain a uniform direction

$$u = \frac{Z}{\|Z\|} \quad \in S^{d-1}.$$

3. Sample a radius
$$r = U^{1/d}, \quad U \sim \mathrm{Uniform}(0, 1).$$
This accounts for the fact that the density in a ball grows like r^{d-1}.

4. Return the point
$$x = r\,u.$$

This method never rejects samples and always returns uniform points in B_d. It is computationally efficient and works well for all tested dimensions.

3.2 Rejection Sampling from the Cube

The second method draws points uniformly from the hypercube
$$[-1, 1]^d,$$
and accepts a sample x only if $\|x\| \leq 1$.

The acceptance probability is exactly the geometric ratio
$$p = \frac{\mathrm{vol}(B_d)}{(2)^d}.$$

Using
$$\mathrm{vol}(B_d) = \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2} + 1\right)},$$
we obtain:
$$d = 2: \quad p \approx \frac{\pi}{4} \approx 0.785,$$
$$d = 3: \quad p \approx \frac{4\pi/3}{8} \approx 0.524,$$
$$d = 4: \quad p \approx \frac{\pi^2/2}{16} \approx 0.308.$$

My experiments confirm these values: the observed acceptance fraction matches the theoretical prediction for each dimension.

As d increases, p decreases rapidly, which is why rejection sampling becomes inefficient for d=4. Direct sampling does not suffer from this limitation.

4 Task 4: Error Analysis for the Project Integrand

The given integrand is
$$f(x, y, z) = (1 + x^2 + y^2)e^z - \frac{x}{1 + z^2}.$$

I compute MC estimates for $N = 500, 1000, 2000, 5000$ using both direct and rejection sampling, with 10 repeats each. All absolute and relative errors reference the deterministic spherical value $I_{\mathrm{ref}} \approx 6.628895$.

5 Task 5: Symmetrized Monte Carlo

For each sample x, I evaluate

$$\overline{f}(x) = \frac{1}{2^d} \sum_{s \in \{\pm 1\}^d} f(s \odot x).$$

This cancels all odd terms of f exactly. For linear test functions $f(x) = x_i$ my implementation (tests included) achieves machine-precision zero.

For the project integrand, the odd component is small, so variance reduction is present but the absolute error may not always decrease.

6 Experimental Results for d = 2,3,4

I ran the script `run_full_error_report.py`, generating `outputs/full_error_report.csv`. Representative results are summarized below.

6.1 Dimension d=2 (Radial Integrand $f(x) = \|x\|^2$)

- Analytic reference: $I_{\text{ref}} = \pi/2 \approx 1.570796$.
- Standard MC achieves very small errors: as low as 2×10^{-5} for $N = 1000$.
- Symmetrized MC does not help (the integrand is even), sometimes increasing error.

Example rows:

- ($N = 1000$): standard MC mean estimate 1.5707766, abs. error 2×10^{-5}.
- ($N = 5000$): standard MC abs. error 4.6×10^{-6}.

6.2 Dimension d=3 (Project Integrand)

- Deterministic reference: $I_{\text{ref}} \approx 6.628895$.
- Standard MC (direct sampling) errors: 0.18–0.22.
- Symmetrized MC has much smaller variance but similar mean error (as expected).
- Rejection sampling: slightly worse accuracy than direct sampling.

Representative values:

- $N = 500$: standard MC error 0.1812; symmetrized error 0.2180.
- $N = 5000$: standard MC error 0.2155; symmetrized error 0.2126.

6.3 Dimension d=4 (Radial Integrand)

- Analytic reference: $I_{\mathrm{ref}} \approx 3.289868$.
- Standard MC errors around 0.003–0.01.
- Symmetrized MC again reduces variance but does not consistently reduce error.

7 Discussion

The experiments verify several expected behaviors:

1. MC variance scales like $N^{-1/2}$.
2. Direct sampling outperforms rejection sampling in 3D.
3. Symmetrized MC dramatically reduces variance when f has odd components.
4. For integrands dominated by even terms, symmetrization may not improve absolute error, although variance still decreases.
5. The deterministic spherical grid provides a stable 3D reference.

8 Conclusion

This project implemented and analyzed deterministic and Monte Carlo integration methods for the 3D unit ball. Sampling methods, variance formulas, deterministic spherical integration, and a symmetrized MC estimator were all implemented. Experiments across $d = 2, 3, 4$ demonstrate consistency with theoretical expectations.

User documentation, developer documentation, a complete codebase, and automated experiment scripts satisfy all submission requirements.