Chapter 2: Monte Carlo Integration

UE Computational statistics

Pierre Pudlo

Aix-Marseille Université / Faculté des Sciences

About this chapter

Objectives:

- Approximate univariate and multivariate integrals using Monte Carlo methods.
- Understant importance sampling (“échantillonnage préférentiel” in French).

Content:

- Role of simulation in computational statistics.
- Simulation of uniform, normal, and other classical distributions (inverse transform method, rejection method, Box-Muller method).
- Introduction to Gaussian vectors: simulation via Cholesky decomposition.

1. The Problem

Determistic methods to compute integrals:

rectangle rule | trapezoidal rule | Simpson’s rule | Runge-Kutta methods | …

$$\Gamma(\lambda) = \int_0^\infty x^{\lambda-1} e^{-x} dx$$

Implementation in Python:

```python
import numpy as np
from scipy.integrate import quad

def f(x, lambd_):
    return x**(lambd_-1) * np.exp(-x)

integral, error = quad(f, 0, np.inf, args=(2,))
print(integral, " +/- ", error)
```

```
0.9999999999999998  +/-  5.901457087046093e-10
```

```python
integral, error = quad(f, 0, np.inf, args=(3,))
print(integral, " +/- ", error)
```

```
2.0  +/-  1.0454242961055758e-10
```

Sometimes, it fails

$$m(\mathbf{x}) = \int_{-\infty}^{\infty} \prod_{i=1}^{10} \frac{1}{\pi} \frac{1}{1+(x_i-\theta)^2} d\theta$$

```python
import scipy.stats as stats
def m(theta, x):
    return np.prod(1/(1+(x-theta)**2)/np.pi)
x = stats.cauchy.rvs(size=10)
integral, error = quad(m, -np.inf, np.inf, args=(np.random.randn(10)))
print(integral, " +/- ", error)
```

```
2.40194783933899e-08  +/-  1.0427256275325777e-11
```

```python
integral, error = quad(m, -200, 200, args=(np.random.randn(10)))
print(integral, " +/- ", error)
```

```
3.4246317094289783e-07  +/-  8.639628776692238e-10
```

```python
integral, error = quad(m, -800, 800, args=(np.random.randn(10)))
print(integral, " +/- ", error)
```

```
1.466229546687716e-10  +/-  2.915312068328523 4e-10
```

2. Classical Monte Carlo integration

Aim

Assume $X \sim f$ with support $\mathscr{X}$. What is the value of

$$\mathbb{E}\Big[h(X)\Big] = \int_{\mathscr{X}} h(x)f(x)dx?$$

Approximation with an iid sample drawn from pdf f:

$$\bar{h}_n = \frac{1}{n}\sum_{i=1}^{n} h(x_i)$$

Error?

By the **Law of Large Numbers**:

$$\bar{h}_n \xrightarrow[n\to\infty]{\text{a.s.}} \mathbb{E}\Big[h(X)\Big]$$

And, if we set $v_n = \dfrac{1}{n-1}\displaystyle\sum_{i=1}^{n}\Big(h(x_i) - \bar{h}_n\Big)^2$ then, by the **Central Limit Theorem**, and the Delta method:

$$\frac{\bar{h}_n - \mathbb{E}\Big[h(X)\Big]}{\sqrt{v_n/n}} \quad \overset{\text{in law}}{\approx} \mathscr{N}(0,1).$$

As a consequence,

- The error is of order $1/\sqrt{n}$
- Whatever the dimension of $\mathscr{X}$, the error is of the same order
- The Standard Error (SE) is $\sqrt{v_n/n}$
- We can get a confidence interval for $\mathbb{E}\left[h(X)\right]$:

$$\bar{h}_n \pm z_{1-\alpha/2}\sqrt{\frac{v_n}{n}}$$

  where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile of the standard normal distribution
- $z_{1-\alpha/2} \approx 1.96 \approx 2$ for $\alpha = 0.05$.

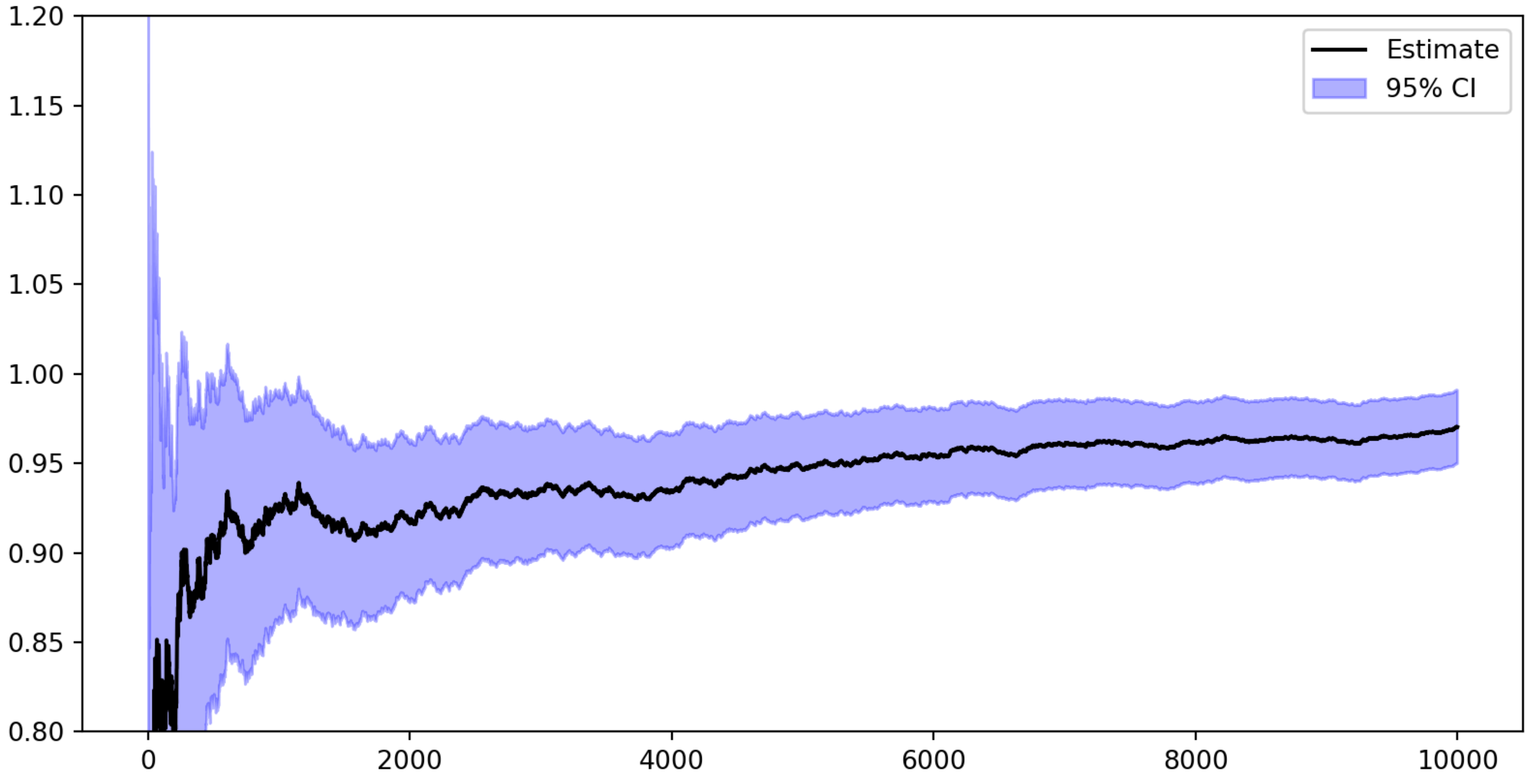
Example

Approximate $\int_0^1 \Big(\cos(50x) + \sin(20x)\Big)^2 dx$

```python
def h(x):
    return (np.cos(50*x) + np.sin(20*x))**2
N = 10000
u = stats.uniform.rvs(size=N)
bar_h = np.cumsum(h(u))/np.arange(1, N+1)
v = np.cumsum((h(u))**2)/np.arange(1, N+1) - bar_h**2
v = v[1:]*np.arange(2, N+1) / np.arange(1, N)
se = np.sqrt(v/np.arange(1, N))
print(bar_h[-1], " +/- ", 1.96*se[-1])
```

```
0.9703145294440477  +/-  0.020474738731358952
```

```
(0.8, 1.2)
```



Evolution of the estimate and the 95% confidence interval as a function of the sample size.

⚠ **Warning!**

- The Law of Large Numbers can be apply only if $h(X) \in L^1$
- The Central Limit Theorem can be apply only if $h(X) \in L^2$
- We have assume that v_n is a proper estimate of the variance of $\bar{h}_n$

If these assumptions are not satisfied,

- the estimates can be divergent or the convergence slower,
- the confidence interval can be too narrow, etc.

Another example

$$\text{Estimate } \Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$$

with an iid sample from $\mathscr{N}(0,1)$ and

$$\widehat{\Phi}_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{Z_i \leq t\}$$

Compute the estimates for various values of t and n.

```python
import pandas as pd
def Phi(t, Z):
  return np.mean(Z <= t)
t = [0.0, 0.67, 0.84, 1.28, 1.65, 1.32, 1.58, 3.09, 3.72]
N = [10**2, 10**3, 10**4, 10**5, 10**6, 10**7]
results = pd.DataFrame(index=N, columns=t)
Z = stats.norm.rvs(size=max(N))
for n in N:
  for threshold in t:
    results.loc[n, threshold] = Phi(threshold, Z[:n])
results.loc['scipy.stats'] = [stats.norm.cdf(threshold) for threshold in t]
results = results.applymap(lambda x: round(x, 3))
print(results)
```

```
              0.00   0.67   0.84   1.28   1.65   1.32   1.58   3.09  3.72
100          0.580  0.730  0.830  0.900  0.930  0.910  0.930  1.000   1.0
1000         0.533  0.753  0.807  0.896  0.948  0.908  0.940  1.000   1.0
10000        0.504  0.750  0.799  0.898  0.950  0.906  0.942  0.999   1.0
100000       0.502  0.749  0.801  0.902  0.952  0.909  0.945  0.999   1.0
1000000      0.501  0.749  0.800  0.900  0.951  0.907  0.943  0.999   1.0
10000000     0.500  0.749  0.800  0.900  0.951  0.907  0.943  0.999   1.0
scipy.stats  0.500  0.749  0.800  0.900  0.951  0.907  0.943  0.999   1.0
```

3. Importance sampling

3.1 The idea

So far, the integral $\int_{\mathscr{X}} h(x)f(x)dx$ was seen as $\mathbb{E}(h(X))$ where $X \sim f$

Assume that g is such that, for all $x \in \mathscr{X}$, $g(x) > 0$. Then, we can write

$$\int_{\mathscr{X}} h(x)f(x)dx = \int_{\mathscr{X}} h(x)\frac{f(x)}{g(x)}g(x)dx = \mathbb{E}\left[h(Y)\frac{f(Y)}{g(Y)}\right]$$

where $Y \sim g$.

Importance weight

$$\mathbb{E}(h(X)) = \mathbb{E}\left[h(Y)\frac{f(Y)}{g(Y)}\right], \quad X \sim f,\, Y \sim g$$

- The ratio $f(Y)/g(Y)$ is called the **importance weight**
- f is the **target distribution** and g is the **sampling distribution**

Note

The importance weight corrects the bias of the Monte Carlo estimate drawn from g instead of f

- The IS weights $f(y)/g(y)$ does not depend on the h-function we want to integrate
- Hence, the same iid sample from g, with the same weights
  - can be used to estimate $\mathbb{E}(h(X))$ for different h-functions

Importance sampling estimate

The **crude Monte Carlo** estimate is

$$\widehat{\mathrm{MC}}_n = \frac{1}{n}\sum_{i=1}^{n} h(X_i)\ 1$$

where $X_1, \ldots, X_n$ is an iid sample from f. $\implies$ weights are all equal to 1

The **importance sampling** estimate is

$$\widehat{\mathrm{IS}}_n = \frac{1}{n}\sum_{i=1}^{n} h(Y_i)\ \frac{f(Y_i)}{g(Y_i)}$$

where $Y_1, \ldots, Y_n$ is an iid sample from g.

An example

Let $Z \sim \mathcal{N}(0,1)$. We want to estimate

$$\mathbb{P}(Z > 4.5) = \int_{4.5}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$$

According to scipy, this probability is approximately

```python
print(stats.norm.sf(4.5))
```

```
3.3976731247300543e-06
```

If we use an iid sample from $\mathcal{N}(0,1)$, what is happening to

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{Z_i > 4.5\}?$$

- Almost all $\mathbf{1}\{Z_i > 4.5\}$ are equal to 0
- There is a high chance that the estimate is 0, which is wrong
- The empirical variance is also zero very bad!

$$\mathbb{P}(Z > 4.5) = \mathbb{E}(h(Z)), \quad \text{where } h(z) = \mathbf{1}\{z > 4.5\}$$

Analysis:

- The target distribution f draws points z_i where the h-function is zero
- The Monte Carlo estimate is then zero
- We need a sampling distribution g that draw points where the h-function is not zero, i.e., on $(4.5, \infty)$
- We can use a distribution on the half line $(0, \infty)$ and shift it to $(4.5, \infty)$, by adding the constant 4.5 to the random variates from the half-line distribution

Let us use an iid sample from $\mathrm{Exp}(1, \mathrm{shift} = 4.5)$, shifted at 4.5, with pdf:

$$g(x) = e^{4.5-x}\mathbf{1}\{x > 4.5\}$$

We have

$$\begin{aligned}\mathbb{P}(Z > 4.5) &= \int_{4.5}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz \\ &= \int_{4.5}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2+y-4.5}\, e^{4.5-y} dy\end{aligned}$$

Thus,

$$\mathbb{P}(Z > 4.5) = \mathbb{E}\left[\frac{1}{\sqrt{2\pi}} e^{-Y^2/2+Y-4.5}\right], \quad Y \sim \mathrm{Exp}(1, \mathrm{shift} = 4.5)$$

```python
print("True value: ", stats.norm.sf(4.5))
```

```
True value:  3.3976731247300543e-06
```

```python
N = 10**5
Z = stats.norm.rvs(size=N)
print("Crude Monte-Carlo: ", np.mean(Z > 4.5))
```

```
Crude Monte-Carlo:  0.0
```

```python
print("Crude Monte-Carlo SE: ", np.std(Z > 4.5)/np.sqrt(N))
```

```
Crude Monte-Carlo SE:  0.0
```

```python
Y = stats.expon.rvs(size=N) + 4.5
hY = stats.norm.pdf(Y)/stats.expon.pdf(Y-4.5, scale=1)
print("Importance sampling: ", np.mean(hY))
```

```
Importance sampling:  3.3777381054528004e-06
```

```python
print("Importance sampling SE: ", np.std(hY)/np.sqrt(N))
```

```
Importance sampling SE:  1.3867946727179822e-08
```

- Important sampling is much better
- The sampling distribution is $\mathrm{Exp}(1)$, shifted at 4.5
- What is happening with $\mathrm{Exp}(\lambda, \mathrm{shift} = 4.5)$ with other values of λ?

```python
print("True value: ", stats.norm.sf(4.5))
```

```
True value:  3.3976731247300543e-06
```

```python
N = 10**5
Z = stats.norm.rvs(size=N)
print("Crude Monte-Carlo: ", np.mean(Z > 4.5))
```

```
Crude Monte-Carlo:  0.0
```

```python
print("Crude Monte-Carlo SE: ", np.std(Z > 4.5)/np.sqrt(N))
```

```
Crude Monte-Carlo SE:  0.0
```

```python
lambda_exp = 4.8
Y = stats.expon.rvs(size=N, scale = 1./lambda_exp) + 4.5
hY = stats.norm.pdf(Y)/stats.expon.pdf(Y-4.5, scale=1/lambda_exp)
print("Importance sampling: ", np.mean(hY))
```

```
Importance sampling:  3.3967453013170363e-06
```

```python
print("Importance sampling SE: ", np.std(hY)/np.sqrt(N))
```

```
Importance sampling SE:  4.256979688916116e-10
```

3.2 Best sampling distribution?

Aim: estimate $I = \int_{\mathscr{X}} h(x) f(x) dx$

If $h(x) > 0$ on $\mathscr{X}$, then $I > 0$, and

$$g(x) = h(x) f(x) \Big/ I$$

is a proper pdf on $\mathscr{X}$.

Importance sampling consider then the estimate

$$\frac{1}{n} \sum_{i=1}^{n} \frac{h(X_i) f(X_i)}{g(X_i)} = \frac{1}{n} \sum_{i=1}^{n} I = I$$

When the sampling distribution is $g(x) \propto h(x)f(x)$,

- The estimate is **exact**, and the Monte Carlo error is zero.
- But the estimate **cannot be computed** in practice:
  - we cannot compute $g(X_i)$ since I is unknown
  - it may be difficult to sample from g

When $h(X)$ can take both positive and negative values, we can use the following sampling distribution:

$$g^\star(x) = \frac{|h(x)|f(x)}{\int_{\mathscr{X}} |h(x)|f(x)dx}$$

The IS estimate is then

$$\frac{1}{n}\sum_{i=1}^{n} \frac{h(X_i)f(X_i)}{g^\star(X_i)} = \int_{\mathscr{X}} |h(x)|f(x)dx \; \frac{1}{n}\sum_{i=1}^{n} \mathrm{sign}(h(X_i))$$

where $X_1, \ldots, X_n$ is an iid sample from $g^\star$.

Likewise, it is possible to compute it explicitly because

- it depends on the unknwon quantity $\int_{\mathscr{X}} |h(x)| f(x) dx$
- it may be difficult to sample from $g^\star$

Nevertheless, $g^\star(y) \propto |h(y)| f(y)$ draws random variates :

- where $|h|$ is large and where f is large

Exercise

1. Fix any sampling distribution g. Use Jensen's inequality to show that, when $Y \sim g$,

$$\mathbb{E}\left[\frac{h^2(Y)f^2(Y)}{g^2(Y)}\right] \geq \left(\mathbb{E}\left[\frac{|h(Y)|f(Y)}{g(Y)}\right]\right)^2$$

2. Deduce that $g^\star$ is the best sampling distribution in terms of variance of the IS estimate.

$\mathbb{P}(Z > 4.5)$ revisited, $Z \sim \mathcal{N}(0,1)$

- The h-function is $\mathbf{1}\{z > 4.5\} \quad \Longrightarrow \quad$ a sampling dist. with support $(4.5, \infty)$
- Given $Z > 4.5$, where would the target $\mathcal{N}(0,1)$ distribution draw random variates?
  - Not far away from 4.5

Hence, a sampling distribution g on $(4.5, \infty)$ that draws random variates close to 4.5 is a good choice. That does explain $\mathrm{Exp}(\lambda, \mathrm{shift} = 4.5)$

How to tune λ?

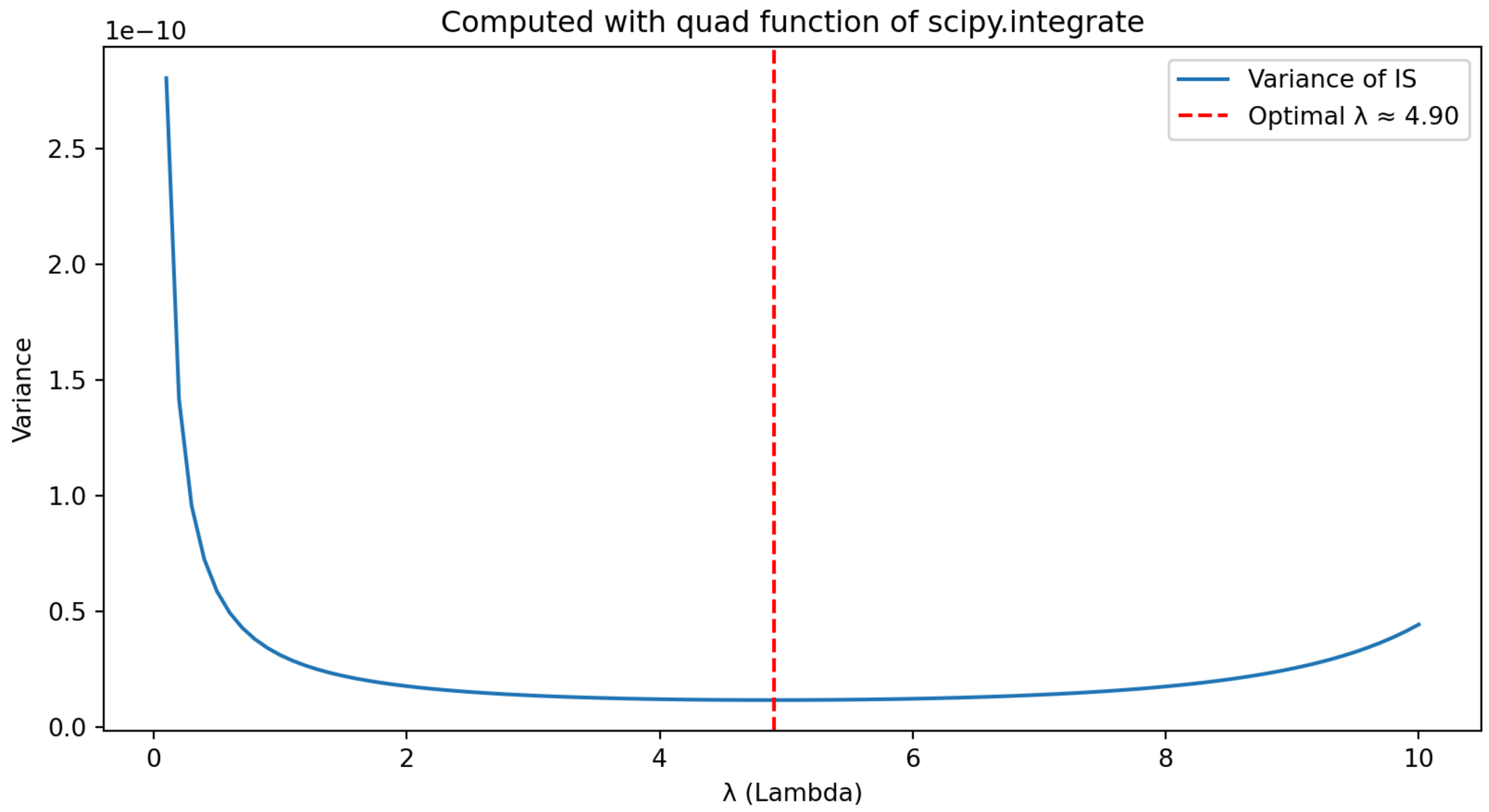
The variance is

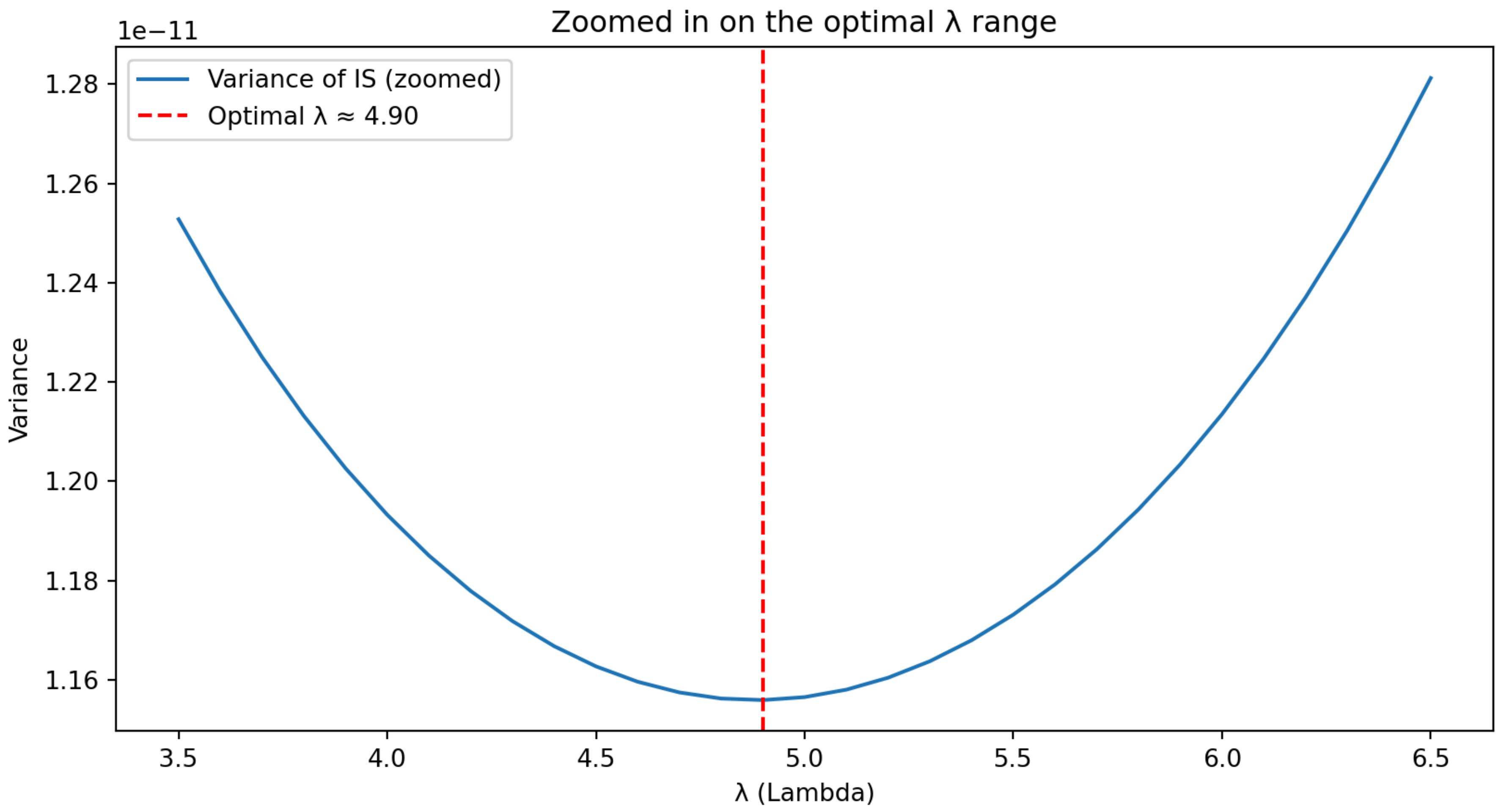
$$\mathbb{E}\left[h^2(Y)\frac{f^2(Y)}{g_\lambda(Y)}\right] - I^2, \quad Y \sim g_\lambda = \mathrm{Exp}(\lambda, \mathrm{shift} = 4.5)$$

The first integral is

$$\int_{4.5}^{\infty} \frac{1}{\lambda^2 2\pi} e^{-y^2+\lambda(y-4.5)} \lambda e^{-\lambda(y-4.5)} dy = \frac{1}{2\pi\lambda} \int_{4.5}^{\infty} e^{-y^2+\lambda(y-4.5)} dy$$

We can plot this, as a function of λ


Computed with quad function of scipy.integrate
1e−10
2.5
2.0
1.5
1.0
0.5
0.0
Variance
0
2
4
6
8
10
λ (Lambda)
Variance of IS
Optimal λ ≈ 4.90


Zoomed in on the optimal λ range
1e−11
Variance of IS (zoomed)
Optimal λ ≈ 4.90
Variance
1.16
1.18
1.20
1.22
1.24
1.26
1.28
3.5
4.0
4.5
5.0
5.5
6.0
6.5
λ (Lambda)

Selection of the sampling distribution

The variance of the IS estimate is

$$\frac{1}{n}\mathbb{V}\left[h(X)\frac{f(X)}{g(X)}\right], \quad X \sim g$$

It is finite if and only if

$$\mathbb{E}\left[h^2(X)\frac{f^2(X)}{g^2(X)}\right] < \infty.$$

Since $X \sim g$, it is equivalent to

$$\int_{\mathscr{X}} h^2(x)\frac{f^2(x)}{g(x)}dx < \infty.$$

Example

Assume we want to estimate

$$I = \int_{-\infty}^{\infty} \frac{1}{1+x^2} dx$$

and introduce the pdf $\varphi(x)$ of the $\mathcal{N}(0,1)$ distribution.

Then

$$\widehat{I}_n = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{1+X_i^2}\frac{1}{\varphi(X_i)} = \frac{1}{n}\sum_{i=1}^{n} \frac{\sqrt{2\pi}e^{X_i^2/2}}{1+X_i^2}$$

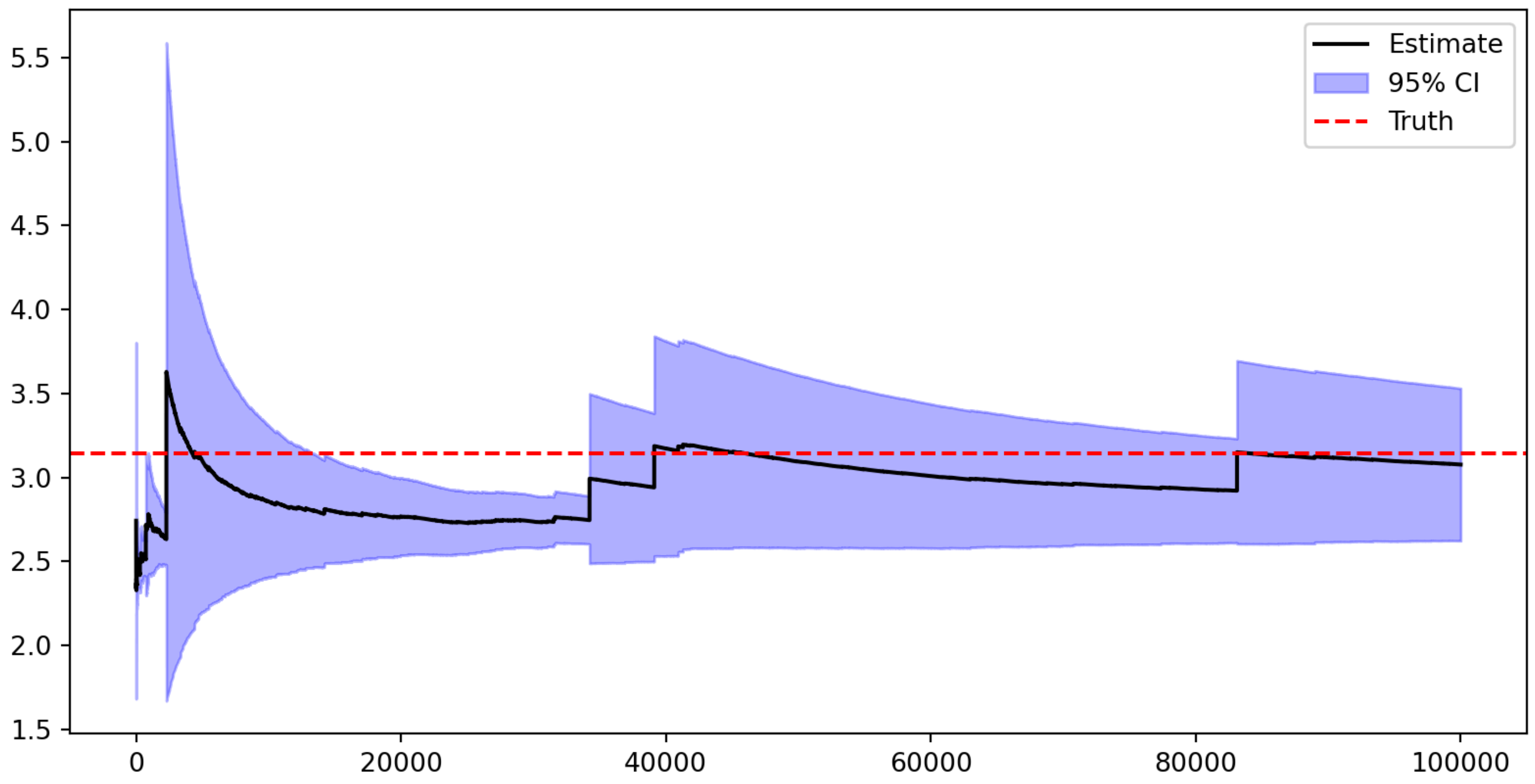
where $X_1, \dots, X_n$ is an iid sample from φ.

```python
N = 10**5
X = stats.norm.rvs(size=N)
hX = np.sqrt(2*np.pi)*np.exp(X**2/2)/(1+X**2)
IS = np.cumsum(hX)/np.arange(1, N+1)
print("Estimate: ", IS[-1])
```

```
Estimate:  3.0767041718743835
```

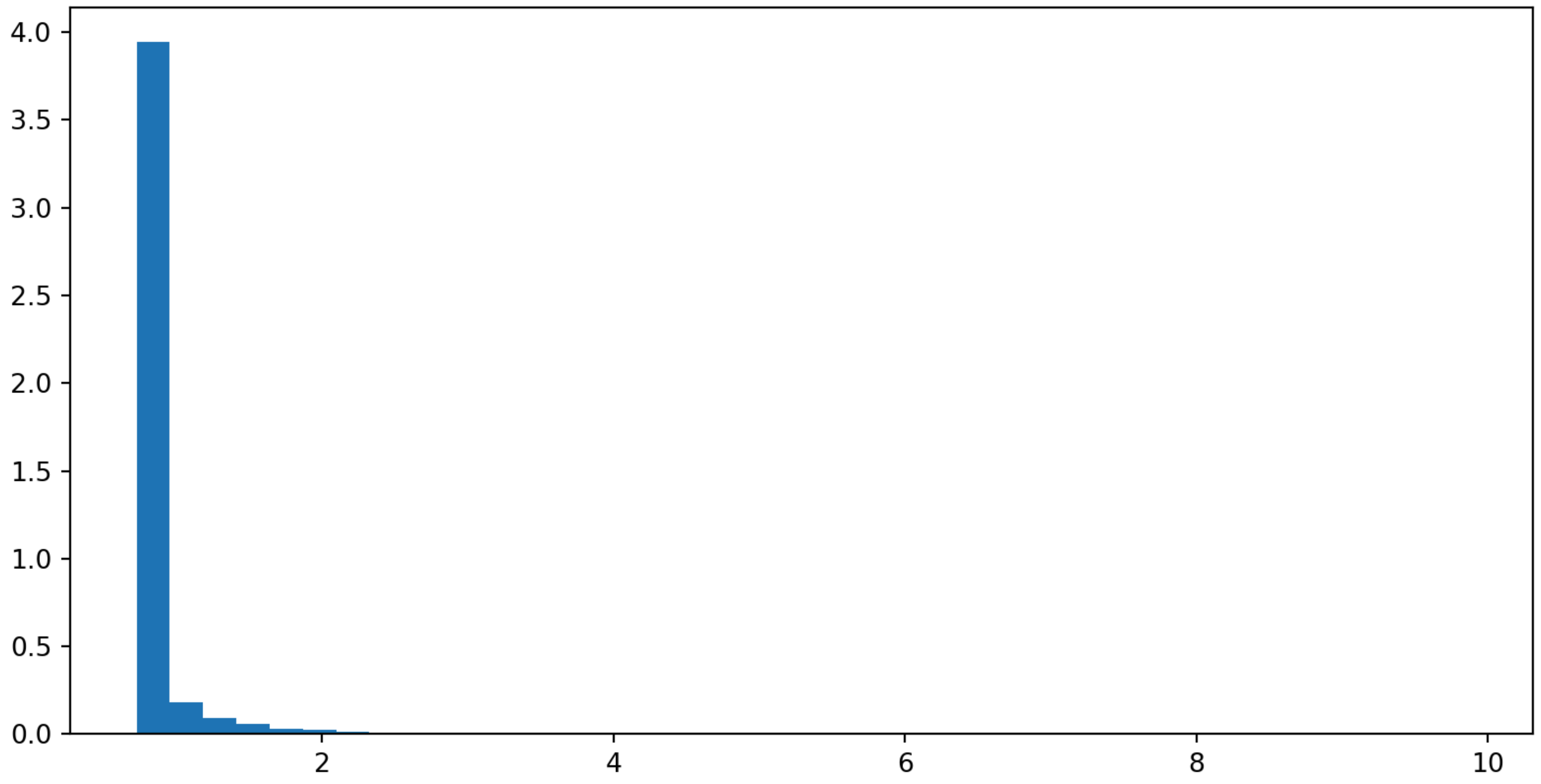
```python
VAR = np.cumsum(hX**2)/np.arange(1, N+1) - IS**2
IS = IS[1:]
VAR = VAR[1:]*np.arange(2, N+1)/np.arange(1, N)
print("SE: ", np.sqrt(VAR[-1]/N))
```

```
SE:  0.23148110061200364
```


Estimate
95% CI
Truth
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
0
20000
40000
60000
80000
100000

- The variance is infinite
- The estimated value of the variance is not reliable (far from being infinite)

The empirical distribution of $\log(f(X_i)/g(X_i))$ is



- A few values of $f(X_i)/g(X_i)$ are very large
- Most of them are very small
- Actually, $f(x)/g(x)$ is an unbounded function of x: it tends to ∞ when $|x| \to \infty$

Tip

Choose the sampling distribution g such that the IS weight $f(x)/g(x)$ is bounded.

In this case, if $\int_{\mathscr{X}} h^2(x)f(x)dx < \infty$ then the variance of the IS estimate is finite.

Exercise: prove it!