Computational Physics – Lecture 23

(11 May 2020)

Recap

In the previous lecture, we learnt how to test pseudo-random numbers for randomness.

Today's plan

In today's class we want to apply random numbers to certain computational problems.

Monte Carlo methods

It turns out that random numbers are very useful in solving certain numerical problems.

In general, any algorithm that employs random numbers is called a “Monte Carlo method”.

Monte Carlo methods

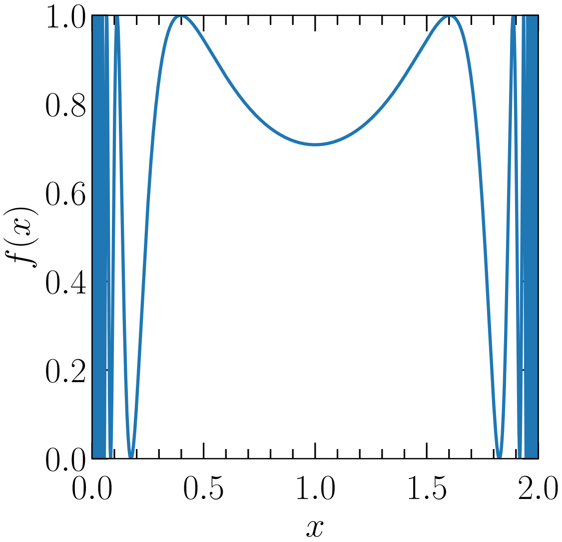
Monte Carlo gambling house in Monaco. Karl Pearson came up with the χ^2 test here in 1892. (cf. Las Vegas algorithms.)

Monte Carlo Integration

Suppose we want to evaluate the integral

$$I = \int_0^2 \sin^2\left[\frac{1}{x(2-x)}\right] dx. \tag{1}$$

Monte Carlo Integration



Monte Carlo Integration

The integrand is well-behaved in the middle part of the range, but varies infinitely near the edges.

So the integral is challenging. Methods like the trapezoidal rule may not work very well.

Note that the function fits in the rectangle of our figure, so the integral should be finite and less than 2.

Monte Carlo Integration

We can compute such integrals using the idea that we adopted in the Rejection Method.

Let the area of the bounding rectangle of the figure be A and the area under the curve be I (our desired integral).

Then if we choose a point uniformly in the area A, the probability of it falling under the curve is $p = I/A$.

Monte Carlo Integration

Now suppose we sample a large number N of random points within the area A.

We then count the number of points that are below the curve. Let this number be k.

Now the fraction k/N should be approximately equal to the probability p.

So $k/N \approx I/A$, or

$$I \approx \frac{kA}{N}. \tag{2}$$

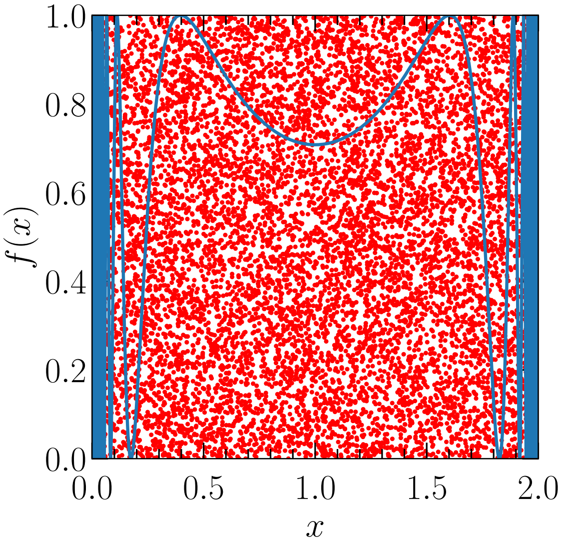
Monte Carlo Integration

Monte Carlo integration is particularly useful in the case of such pathological functions or in higher-dimensional functions.

In our case, we can sample 10,000 points in the area A by doing something like

```
n = 10000
x = np.random.rand(n)*2.0
y = np.random.rand(n)
```

Monte Carlo Integration



Monte Carlo Integration

We count the number of points falling below the curve:

```
k = 0
for i, arg in enumerate(x):
  if y[i] < fn(x[i]):
    k += 1
```

Where I have previously defined

```
def fn(x):
  return (np.sin(1/(x*(2-x))))**2
```

In our case, I get $k = 7183$.

Monte Carlo Integration

In our case, $A = 2 \times 1 = 2$, so we have

$$\begin{aligned} I &= \frac{kA}{N} \\ &= \frac{7183 \times 2}{10000} \\ &= 1.444 \end{aligned} \tag{3}$$

which is close to the correct answer.
This is called Monte Carlo Integration.

Monte Carlo Integration

How does this integration method compare with classical methods such as the Trapezoidal?

The probability of a single point to fall below the curve is $p = I/A$. The probability that it falled above the curve is $1 - p$.

The probability that a particular k points fall below the curve and the rest above the curve is therefore $p^k(1-p)^{N-k}$.

But there are $\binom{N}{k}$ ways to choose k points out of N so the total probability of having k points below the curve is

$$P(k) = \binom{N}{k} p^k (1-p)^{N-k}. \tag{4}$$

Monte Carlo Integration

The variance of this distribution is

$$\text{var } k = Np(1-p) = N\frac{I}{A}\left(1 - \frac{I}{A}\right). \tag{5}$$

The expected variation in the value of k is the standard deviation $\sqrt{\text{var } k}$.

We can now propagate this error to compute the error in the integration as

$$\sigma = \sqrt{\text{var } k}\frac{A}{N} = \frac{\sqrt{I(A-I)}}{\sqrt{N}}. \tag{6}$$

So the accuracy of our integral varies as $N^{-1/2}$.

Monte Carlo Integration

We can compare this performance with that of, e.g., the Trapezoidal rule.

The total integration error in the Trapezoidal Method goes as h^2, where h is the step size.

But $h = (b - a)/N$ so the error in the Trapezoidal method goes as N^{-2}, which is much better than the $N^{-1/2}$ of the Monte Carlo method.

So, when they work, classical methods will perform better than Monte Carlo methods.

Monte Carlo Integration: Mean Value method

A slight modification of the above method, called Mean Value method, is also available.

Suppose we want to calculate

$$I = \int_a^b dx f(x). \tag{7}$$

The mean value of f is given by

$$\langle f \rangle = \frac{1}{b-a} \int_a^b dx f(x) = \frac{I}{b-a}. \tag{8}$$

So we have new formula $I = (b-a)\langle f \rangle$. If we estimate $\langle f \rangle$ we can estimate I.

Monte Carlo Integration: Mean Value method

But we can simply estimate $\langle f \rangle$ by taking N random points $x_1, x_2, \ldots, x_N$ between a and b and doing

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i). \tag{9}$$

Then we have

$$I \approx \frac{b-a}{N} \sum_{i=1}^{N} f(x_i). \tag{10}$$

This is the Mean Value Method.

Monte Carlo Integration: Mean Value method

How accurate is the mean value method?

It can be shown that in this case

$$\sigma = (b-a)\frac{\sqrt{\text{var } f}}{\sqrt{N}}. \tag{11}$$

So the error again goes as $1/\sqrt{N}$ but the leading constant is often smaller in this case, so the mean value method performs a bit better.

Monte Carlo Integration: Higher dimensions

For integration in higher dimensions, Monte Carlo is usually the best method.

The number of mesh points required by classical methods become exponentially large in higher dimensions. If we use $N = 100$ in each direction, Trapzoidal integration in 4 dimensions will need 100 million points.

The Monte Carlo integration formula in higher dimensions can be written as

$$I \approx \frac{V}{N} \sum_{i=1}^{N} f(\mathbf{r}_i), \tag{12}$$

which is a generalisation of the mean value method formula.

Monte Carlo Integration: Importance Sampling

There are functions for which Monte Carlo integration does not work well.

An example is functions that have divergences.

Consider the integral

$$I = \int_0^1 dx \frac{x^{-1/2}}{e^x + 1} \tag{13}$$

in which the integrand diverges at $x = 0$ but the integrand is finite.

Monte Carlo Integration: Importance Sampling

If we try to do this integration using the mean value method, we have problems.

Occassionally, we will get an x_i that is very close to 0 and therefore $f(x_i)$ has a very large value.

In other words, the error on I, viz. σ, is very large in this case.

This problem is solved using a technique called importance sampling.

Monte Carlo Integration: Importance Sampling

For a general function $g(x)$, we can define a weighted average over the integral from a to b as

$$\langle g \rangle_w = \frac{\int_a^b w(x) g(x) dx}{\int_a^b w(x) dx}, \tag{14}$$

where $w(x)$ is some other function.

Monte Carlo Integration: Importance Sampling

Now if we have

$$I = \int_a^b f(x)dx, \tag{15}$$

setting $g(x) = f(x)/w(x)$ we have

$$\left\langle \frac{f(x)}{w(x)} \right\rangle_w = \frac{\int_a^b w(x)f(x)/w(x)dx}{\int_a^b w(x)dx} \tag{16}$$

$$= \frac{\int_a^b f(x)dx}{\int_a^b w(x)dx} \tag{17}$$

$$= \frac{I}{\int_a^b w(x)dx} \tag{18}$$

Monte Carlo Integration: Importance Sampling

So we have

$$I = \left\langle \frac{f(x)}{w(x)} \right\rangle_w \int_a^b w(x)dx \tag{19}$$

We just need to now calculate the weighted average to get I.

Monte Carlo Integration: Importance Sampling

Consider the normalised PDF

$$p(x) = \frac{w(x)}{\int_a^b w(x)dx}. \tag{20}$$

If we sample N random numbers x_i from this PDF, the average number of points that will fall between x and $x + dx$ will be $Np(x)dx$.

So for any function $g(x)$

$$\sum_{i=1}^{N} g(x_i) = \int_a^b Np(x)g(x)dx. \tag{21}$$

Monte Carlo Integration: Importance Sampling

So we can now write the general weighted average of $g(x)$ as

$$\langle g\rangle_w = \frac{\int_a^b w(x)g(x)dx}{\int_a^b w(x)dx} \tag{22}$$

$$= \int_a^b p(x)g(x)dx \tag{23}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} g(x_i), \tag{24}$$

where x_i are random numbers sampled from $p(x) \propto w(x)$.

Monte Carlo Integration: Importance Sampling

We saw above that

$$I = \left\langle \frac{f(x)}{w(x)} \right\rangle_w \int_a^b w(x)dx \tag{25}$$

So now we have

$$I = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{w(x_i)} \int_a^b w(x)dx. \tag{26}$$

We now choose $w(x)$ so that there are no divergences.

Monte Carlo Integration: Importance Sampling

Importance Sampling is a generalisation of the mean value method.

We recover the mean value method if $w(x) = 1$.

Importance sampling allows us to calculate I from the sum $\sum_i f(x_i)/w(x_i)$ instead of the sum $\sum_i f(x_i)$.

This can be used to get rid of any pathologies in $f(x)$. The extra work now is to sample random numbers of the non-uniform distribution $p(x) \propto w(x)$.

Monte Carlo Integration: Importance Sampling

We can estimate the error on this integral also.

It can be shown that this is given by

$$\sigma = \frac{\sqrt{\mathrm{var}_w(f/w)}}{\sqrt{N}} \int_a^b w(x)dx, \tag{27}$$

where $\mathrm{var}_w g = \langle g^2 \rangle_w - \langle g \rangle_w^2$.

Monte Carlo Integration: Importance Sampling

In our original example

$$I = \int_0^1 dx \frac{x^{-1/2}}{e^x + 1} \tag{28}$$

if we choose $w(x) = x^{-1/2}$, we get

$$\frac{f(x)}{w(x)} = (e^x + 1)^{-1}, \tag{29}$$

which is finite and well-behaved.

All we need is to use the transformation metod to get random numbers from

$$p(x) = \frac{x^{-1/2}}{\int_0^1 x^{-1/2} dx} = \frac{1}{2\sqrt{x}}. \tag{30}$$

Monte Carlo Integration: Importance Sampling

Importance sampling can also be used to evaluate integrals over infinite domains.

If we have to do an integral from 0 to ∞, we cannot choose uniform random numbers in this domain.

In this case, we could use a function $w(x)$ that also goes from 0 to ∞ but is integrable (such as the exponential distribution).

This is very valuable for pathological functions or higher-dimensional integrals.

Activity

The area of a circle with unit radius is given by

$$I = \int \int_{-1}^{1} f(x, y) dx dy, \tag{31}$$

where

$$f(x) \begin{cases} 1 & \text{if } x^2 + y^2 \leq 1 \\ 0, & \text{otherwise,} \end{cases} \tag{32}$$

Calculate this area using Monte Carlo integration.

Then calculate the volume of a ten-dimensional unit sphere.