

Numerical integration continued

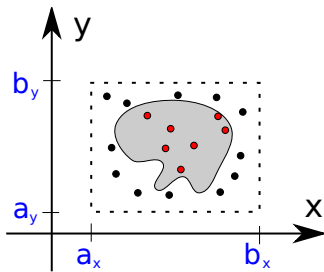
Eugeniy E. Mikhailov

The College of William & Mary



Lecture 10

Toy example - area of the pond



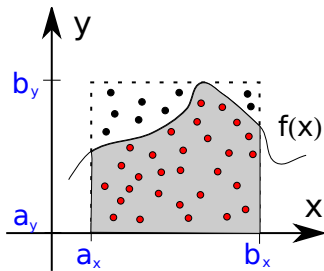
$$A_{pond} = \frac{N_{inside}}{N_{total}} A_{box}$$

where

$$A_{box} = (b_x - a_x)(b_y - a_y)$$

- Points must be **uniformly** and randomly distributed across the area.
- The smaller the enclosing box the better it is.

Naive Monte Carlo integration



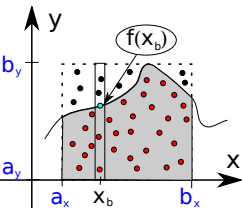
$$\int_{a_x}^{b_x} f(x) dx = \frac{N_{inside}}{N_{total}} A_{box}$$

where

$$A_{box} = (b_x - a_x)(b_y)$$

- Points must be **uniformly** and randomly distributed across the area.
- The smaller the enclosing box the better it is. So $\max(f(x)) \rightarrow b_y$

Monte Carlo integration derived



Notice that if we choose a small stripe around the bin value x_b , then subset of points in that stripe gives an estimate for $f(x_b)$. Thus why bother spreading points around area?

Let's chose a uniform random distribution of points x_i inside $[a_x, b_x]$

$$\int_{a_x}^{b_x} f(x) dx \approx \frac{b_x - a_x}{N} \sum_{i=1}^N f(x_i)$$

Notes

Notes

Notes

Notes

Error estimate for Monte-Carlo method

It can be shown that error of the numerical integration (E) is given by the following expressions

Monte Carlo method

$$E = O\left((b_x - a_x) \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N}}\right)$$

where

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^N f(x_i)$$
$$\langle f^2 \rangle = \frac{1}{N} \sum_{i=1}^N f^2(x_i)$$

The term under $\sqrt{\quad}$ is the estimate of the function standard deviation.

Error estimate for other methods

Rectangle method

$$E = O\left(\frac{(b_x - a_x)h}{2} f'\right) = O\left(\frac{(b_x - a_x)^2}{2N} f'\right)$$

Trapezoidal method

$$E = O\left(\frac{(b_x - a_x)h^2}{12} f''\right) = O\left(\frac{(b_x - a_x)^3}{12N^2} f''\right)$$

Simpson method

$$E = O\left(\frac{(b_x - a_x)h^4}{180} f^{(4)}\right) = O\left(\frac{(b_x - a_x)^5}{180N^4} f^{(4)}\right)$$

Multidimensional integration with interval splitting

$$\int_{a_x}^{b_x} \int_{a_y}^{b_y} f(x, y) dx dy = \int_{a_x}^{b_x} dx \int_{a_y}^{b_y} dy f(x, y)$$

Note that last integral is the function of only x

$$\int_{a_y}^{b_y} dy f(x, y) = F(x)$$

$$\int_{a_x}^{b_x} \int_{a_y}^{b_y} f(x, y) dx dy = \int_{a_x}^{b_x} dx F(x)$$

Thus we replaced multidimensional integral as consequent series of single dimension integrals, which we already know how to do.

3D case would look like this

$$\int_{a_x}^{b_x} \int_{a_y}^{b_y} \int_{a_z}^{b_z} f(x, y, z) dx dy dz = \int_{a_x}^{b_x} dx \int_{a_y}^{b_y} dy \int_{a_z}^{b_z} dz f(x, y, z)$$

Multidimensional integration with Monte Carlo

Note that if we would like to split integration region by N points in every of D dimensions, then evaluation time grows $\sim N^D$, which renders Rectangle, Trapezoidal, Simpson, and alike method useless.

Monte Carlo method is a notable exception, it looks very simple even for multidimensional case and maintains the same $\sim N$ evaluation time.

3D case would look like this

$$\int_{a_x}^{b_x} dx \int_{a_y}^{b_y} dy \int_{a_z}^{b_z} dz f(x, y, z) \approx \frac{(b_x - a_x)(b_y - a_y)(b_z - a_z)}{N} \sum_{i=1}^N f(x_i, y_i, z_i)$$

A general case

$$\int_V dV f(\vec{x}) = \int_V dx_1 dx_2 dx_3 \dots dx_D f(\vec{x}) \approx \frac{V}{N} \sum_{i=1}^N f(\vec{x}_i)$$

where V is the multidimensional volume, \vec{x}_i randomly and uniformly distributed points in the volume V

Notes

Notes

Notes

Notes

Matlab functions for integration

1D integration

- `integral`
- `trapz`
- `quad`

2D and 3D

- `integral2`
- `integral3`

There are many others as well. See Numerical Integration help section.

Matlab's implementations are more powerful than those which we discussed but deep inside they use similar methods.



Notes

Notes

Notes

Notes
