

Application of MC methods

Marcin Chrząszcz
mchrzasz@cern.ch



Experimental Methods in Particle Physics,
12 October, 2017

Classical methods of variance reduction

⇒ In Monte Carlo methods the statistical uncertainty is defined as:

$$\sigma = \frac{1}{\sqrt{N}} \sqrt{V(f)}$$

⇒ Obvious conclusion:

- To reduce the uncertainty one needs to increase N .
⇒ Slow convergence. In order to reduce the error by factor of 10 one needs to simulate factor of 100 more points!

⇒ However the other handle ($V(f)$) can be changed! → Lot's of theoretical effort goes into reducing this factor.

⇒ We will discuss **four** classical methods of variance reduction:

1. Stratified sampling.
2. Importance sampling.
3. Control variates.
4. Antithetic variates.

Disadvantages of classical variance reduction methods

- ⇒ All aforementioned methods(beside the Stratified sampling) require knowledge of the integration function!
- ⇒ If you use the method in the incorrect way, you can easily get the opposite effect than intended.
- ⇒ Successful application of them require non negligible effort before running the program.
- ⇒ A natural solution would be that our program is "smart" enough that on his own, he will learn something about our function while he is trying to calculate the integral.
- ⇒ Similar techniques were already created for numerical integration!
- ⇒ Truly adaptive methods are nontrivial to code but are widely available in external packages as we will learn.
- ⇒ Naming conventions:
 - Integration **MC**- software that is able to compute JUST! integrals.
 - Generator **MC**- software that BESIDES! being able to perform the integration is also capable of performing a generation of points accordingly to the integration function.

FOAM algorithm

⇒ S.Jadach (2000), arXiv:physics/9910004, Comp. Phys. Commun. 152 (2003) 55.
Adaptive method with recursive division of the integration domain in cells.

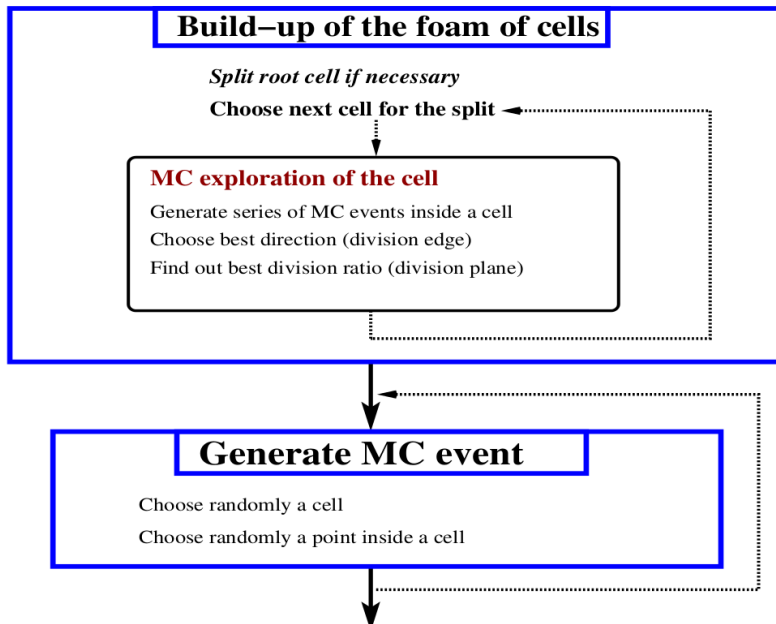
⇒ There are two algorithms in dividing the integration domain:

- Symplectic: Cells are sympleces(hiper-triangles). This method can be applied to not so large number of dimensions. (≤ 5).
- Qubic: Cells are hiper-cubes. This might be applied in higher number dimensions. (≤ 20).

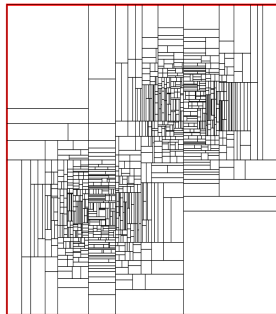
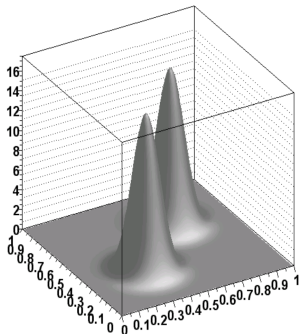
⇒ The algorithm:

- Exploration phase:
The integration domain (hipper-cube) is divided recursively into cells. In each step only one cell is split. The splitting is not event! The procedure is stop when the number of cells reach a certain number that is set by us. One constructs an approximation function and based on this the integral is calculated.
- Generation/Calculation Phase:
We generate random points accordingly to the distribution of approximation function and the integral is calculated using the Importance sampling based on the approximation function.

FOAM algorithm



FOAM algorithm



Neumann-Ulam method

- For example lets try to solve this equation system:

$$\vec{x} = \begin{pmatrix} 1.5 \\ -1.0 \\ 0.7 \end{pmatrix} + \begin{pmatrix} 0.2 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.2 \\ 0.3 & 0.1 & 0.1 \end{pmatrix} \vec{x}$$

- The solution is $\vec{x}_0 = (2.154303, 0.237389, 1.522255)$.
- The propability matrix h_{ij} has the shape:

i/j	1	2	3	0
1	0.2	0.3	0.1	0.4
2	0.4	0.3	0.2	0.1
3	0.3	0.1	0.1	0.5

- An example solution:

```
mchraszcz-ThinkPad-W530% ./mark.x 1 1000000  
2.15625
```

Neumann-Ulam dual method

- Let's try to solve the equation system:

$$\vec{x} = \begin{pmatrix} 1.5 \\ -1.0 \\ 0.7 \end{pmatrix} + \begin{pmatrix} 0.2 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.2 \\ 0.1 & 0.1 & 0.1 \end{pmatrix} \vec{x}$$

- The solution is: $\vec{x}_0 = (2.0, 0.0, 1.0)$.
- Let's put the initial probability as constant:

$$q_1 = q_2 = q_3 = \frac{1}{3}$$

- The propability matrix h_{ij} has the shape:

i/j	1	2	3	4
1	0.2	0.4	0.1	0.3
2	0.3	0.3	0.1	0.3
3	0.1	0.2	0.1	0.6

- An example solution:

```
mchrzasz-ThinkPad-W530% ./mark2.x 1000000  
1.9943 0.001806 1.00267
```


Q & A

Backup