Introduction

Stefano Gandolfi

Los Alamos National Laboratory (LANL)

Microscopic Theories of Nuclear Structure, Dynamics and Electroweak Currents
June 12-30, 2017, ECT*, Trento, Italy

Why (quantum) Monte Carlo?

Monte Carlo: solve multi-dimensional integrals by using random numbers.

Quantum Monte Carlo: solve quantum mechanical problems using Monte Carlo methods.

Why (quantum) Monte Carlo?

Monte Carlo: solve multi-dimensional integrals by using random numbers.

Quantum Monte Carlo: solve quantum mechanical problems using Monte Carlo methods.

From Wikipedia, "Monte Carlo algorithm":

> *In computing, a Monte Carlo algorithm is a randomized algorithm whose running time is deterministic, but whose output may be incorrect with a certain (typically small) probability.*
>
> *The name refers to the grand casino in the Principality of Monaco at Monte Carlo, which is well-known around the world as an icon of gambling.*

About these lectures

My contact: Stefano Gandolfi, **stefano@lanl.gov**

- 9 lectures (45 minutes each)
- Main topics:
  - VMC
  - DMC
  - GFMC/AFDMC
- *Quantum Monte Carlo simulations of solids*
  W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Rev. Mod. Phys. 73, 33 (2001)
- *Quantum Monte Carlo methods for nuclear physics*
  J. Carlson, S. Gandolfi, F. Pederiva, Steven C. Pieper, R. Schiavilla, K. E. Schmidt, and R. B. Wiringa, Rev. Mod. Phys. 87, 1067 (2015)

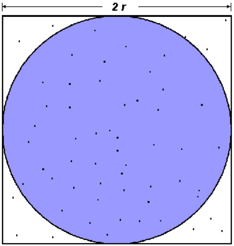
Introduction

What do you need if interested to exercises and examples:

- Basic knowledge of Fortran For your own practice coding, you are very welcome to use any language that you like (but my help will be dependent on that!)
- Know how to compile and run a simple code
- Have your laptop ready to compile Fortran codes. I can help if you use a Linux distribution
- Have basic libraries installed (Lapack, Blas, random numbers generator, ...)
- Install some MPI library. I recommend OpenMPI, https://www.open-mpi.org/ (I can help installing on Linux)
- Have fun while learning!!!

The first, older, example that I found on literature to calculate something by mean of random events, the **dartboard method:**

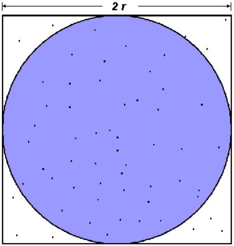
Throw darts, i.e. choose points randomly within a rectangular box with some area inside to integrate on:



$$A_{\rm box} = 4\,r^2\,, \qquad A_{\rm area} = \pi\,r^2\,, \qquad \pi = 4\frac{A_{\rm area}}{A_{\rm box}}$$

$$\frac{A_{\rm area}}{A_{\rm box}} = P(\text{hit inside the area}) = \frac{\#\text{ hits inside the area}}{\#\text{ hits inside the box}}$$

The first example

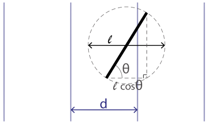


The first code!

```
npt=10000
count=0
do j=1,npt
  x=random1
  y=random2
  if (sqrt(x**2+y**2).lt.radius) count=count+1
enddo
pi=4.0*count/npt
```

Buffon's Needle Problem (1733):

What is the probability that a needle of length l will land on a line, given a floor with equally spaced parallel lines a distance d apart?



Let's define $x = l/d$. For a short needle ($l < d$):

$$P(x) = \int_0^{2\pi} \frac{l|\cos\theta|}{d}\frac{d\,\theta}{2\pi} = \frac{2l}{\pi d} \to \pi \approx \frac{2lN}{dH}$$

where N is needles thrown, and H is the time for a needle to cross a line.

Possible exercise:
throw many needles, or write a MC code for this!

The goal of Monte Carlo integration is to solve multi-dimensional integrals using **random** numbers!

Monte Carlo integration

The goal of Monte Carlo integration is to solve multi-dimensional integrals using **random** numbers!

Why???

Brute force integration

Suppose that we want to calculate some integral of dimension D. The easiest thing to do is a sum over discrete intervals h, something like

$$I = \int_a^b dx_1 \ldots \int_a^b dx_D f(x_1 \ldots x_D) \approx h^D \sum f(x_1 \ldots x_D)$$

Brute force integration

Suppose that we want to calculate some integral of dimension D. The easiest thing to do is a sum over discrete intervals h, something like

$$I = \int_a^b dx_1 \dots \int_a^b dx_D f(x_1 \dots x_D) \approx h^D \sum f(x_1 \dots x_D)$$

It's easy to show that if we want an error of order ϵ, we need to sum a number of points

$$N = \epsilon^{-D}$$

Brute force integration

Suppose that we want to calculate some integral of dimension D. The easiest thing to do is a sum over discrete intervals h, something like

$$I = \int_a^b dx_1 \ldots \int_a^b dx_D f(x_1 \ldots x_D) \approx h^D \sum f(x_1 \ldots x_D)$$

It's easy to show that if we want an error of order ϵ, we need to sum a number of points

$$N = \epsilon^{-D}$$

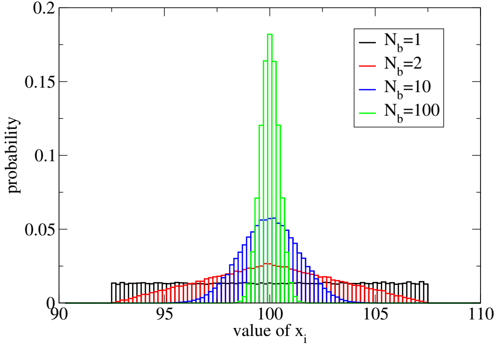
so, for $\epsilon = 0.1$ and a system with 20 particles ($D = 60$) we have to sum $N = 10^{60}$ points.

With the best available supercomputers the time needed is greater than the age of the universe!

- Assume independent samples X_i
- Compute averages by "blocks": $\bar{X}_i = \frac{1}{N_b} \sum_i^{j+N_b} X_i$

Central limit theorem: For $N_b \to \infty$ the Distribution of averages over blocks converges to a Gaussian, whose width gives the statistical error.

Histogram of 100,000 blocks, for different values of N_b:



Note how the width reduces by increasing N_b!

Statistical errors from Gaussian distributions

- Assume independent samples with Gaussian distribution of "measurements"
- Compute average $\bar{X} = \frac{1}{N_b} \sum_i X_i$
- Compute standard error $\sigma = \sqrt{\frac{\sum_i (X_i - \bar{X})^2 / N_b}{N_b}}$

The probabilities to have the average $\bar{X}$ within 1,2, or 3 σ within the exact integral are

- $\mathrm{P}(\bar{X} - \ \sigma \leq x \leq \bar{X} + \ \sigma) \approx 0.6827$
- $\mathrm{P}(\bar{X} - 2\,\sigma \leq x \leq \bar{X} + 2\,\sigma) \approx 0.9545$
- $\mathrm{P}(\bar{X} - 3\,\sigma \leq x \leq \bar{X} + 3\,\sigma) \approx 0.9973$

Monte Carlo integration

Monte Carlo methods provide the tools to solve multi-dimensional integrals!

Monte Carlo methods allow us to solve for integrals:

$$I = \int dx F(x) = \int dx W(x) \frac{F(x)}{W(x)} \,,$$

where we sample points x_i distributed as $W(x)$, and we evaluate $F(x_i)/W(x_i)$. In the limit of large points sampled, we have:

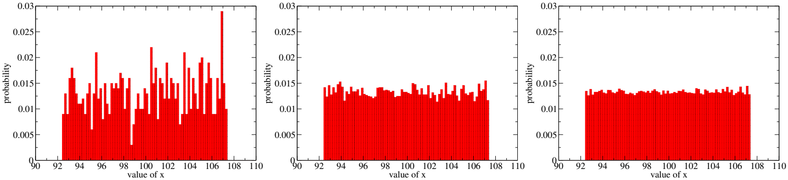
$$X = \frac{1}{N} \sum_{n=1}^{N} \frac{F(x_n)}{W(x_n)} \to I \,.$$

Central limit theorem:

Sample many X_i above. As explained before, the average of them is a "measure" of I, with a (statistical) error that goes like $1/\sqrt{N_b}$ and depends critically on the choice of $W(x)$.

Monte Carlo sampling: the first example

Sampling the distribution $x_i = 100 + 15 * (\xi - 0.5)$ with ξ uniformly distributed from 0 to 1:



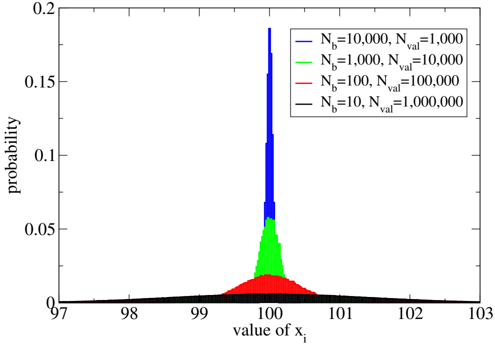
100 samples 1000 samples 1,000,000 samples

Caveat: samples must be independent!

→ calculate averages using several independent samples.

Block size *vs* number of blocks

The total number of samples is given by $N_{\rm tot} = N_b \times N_{\rm val}$. What happens by increasing N_b by keeping constant $N_{\rm tot}$?



The distribution is more peaked, but it becomes "less" Gaussian for small values of $N_{\rm val}$. Need to find a compromise.

Possible exercise: try to solve some simple integral

Monte Carlo integration

Questions???

Monte Carlo integration

Questions???

... for now :-)