Compreendendo simulações de Monte Carlo:
Aula 1 - A integral de Monte Carlo

por Matheus Roos

Orientador: Prof. Dr. Mateus Schmidt

Universidade Federal de Santa Maria, Brasil

24 de dezembro de 2023

LabTMC
Laboratório de Teoria da Matéria Condensada

Overview

- Motivação para a criação do curso;

Overview

- Motivação para a criação do curso;
- Estrutura do curso:

Overview

- Motivação para a criação do curso;
- Estrutura do curso:
  - Aula 1: Aspectos gerais da integral de Monte Carlo;

Overview

- Motivação para a criação do curso;
- Estrutura do curso:
  - Aula 1: Aspectos gerais da integral de Monte Carlo;
  - Aula 2: Simulações de computacionais;

Overview

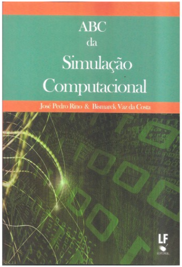
- Motivação para a criação do curso;
- Estrutura do curso:
  - Aula 1: Aspectos gerais da integral de Monte Carlo;
  - Aula 2: Simulações de computacionais;
  - Aula 3: Tópicos de transições de fase.

Overview

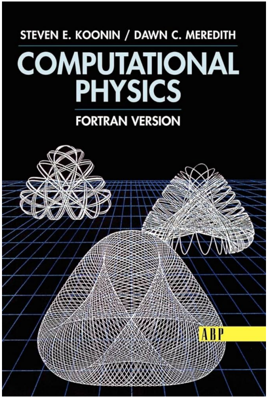
- Motivação para a criação do curso;
- Estrutura do curso:
  - Aula 1: Aspectos gerais da integral de Monte Carlo;
  - Aula 2: Simulações de computacionais;
  - Aula 3: Tópicos de transições de fase.
- Objetivos;

Overview

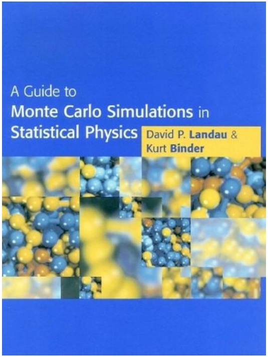
- Motivação para a criação do curso;
- Estrutura do curso:
  - Aula 1: Aspectos gerais da integral de Monte Carlo;
  - Aula 2: Simulações de computacionais;
  - Aula 3: Tópicos de transições de fase.
- Objetivos;
- Bibliografia.



(a) ABC da Simulação Computacional



(b) Computaional Physics



(c) A Guide to Monte Carlo Simulations in Statistical Physics

Introdução

- Rápida introdução histórica;
- O dilema das integrais da física:

$$\mathcal{H} = \frac{p^2}{2m} + U(\vec{r}, t) \ . \tag{1}$$

Considerando $U = U(\vec{r})$:

$$Z_1 = \int\int e^{\beta\mathcal{H}} dp_1 dr_1 \ . \tag{2}$$

Se a função de partição pode ser fatorada, então:

$$Z = Z_1 Z_2 \cdots Z_N = Z_1^N \ . \tag{3}$$

A priori devemos calcular

$$Z = \int\int\cdots\int dr_1 dr_2 \cdots dr_N \int\int\cdots\int dp_1 dp_2 \cdots dp_N e^{\beta\mathcal{H}} \ . \tag{4}$$

Estimativa de Fermi

- Consideremos $N = 20$ partículas;
- O supercomputador Fugaku é capaz de realizar 10^{15} cálculos$/s$;
- A idade do universo é cerca de $10^9 s$.

Construção da bomba atômica

- O problema enfrentado por físicos na II Guerra Mundial;
- Resolvido por: Stanislaw Ulam, John von Neumann e Nicholas Metropolis, dentre outros [1];
- O famoso cassino Monte Carlo, em Mônaco;
- Monte Carlo domina sempre que a dimensão for $D > 3$.

[1] N. Metropolis e S. Ulam (1949). «The Monte Carlo Method». Em: *Journal of the American Statistical Association* 44.247, pp. 97–111

A integral de Monte Carlo: caso 1D

- Consideremos o cálculo da integral

$$I = \int_a^b f(x)dx \; . \tag{5}$$

A integral de Monte Carlo: caso 1D

- Consideremos o cálculo da integral

$$I = \int_a^b f(x)dx \ . \tag{5}$$

- Ao invés de subdividirmos o intervalo, tomamos a média:

$$I \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i) \ , \tag{6}$$

A integral de Monte Carlo: caso 1D

- Consideremos o cálculo da integral

$$I = \int_a^b f(x)dx \ . \tag{5}$$

- Ao invés de subdividirmos o intervalo, tomamos a média:

$$I \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i) \ , \tag{6}$$

- Para um longo número N, segue a seguinte lei estatística:

$$\sigma_I^2 \approx \frac{1}{N}\sigma_f^2 = \frac{1}{N}\left[\frac{1}{N}\sum_{i=1}^{N} f_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} f_i\right)^2\right] \ . \tag{7}$$

Script para calcular uma integral via Monte Carlo [2].

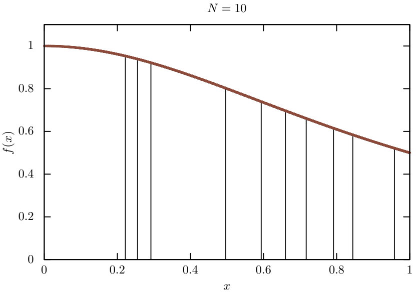
```fortran
program Ex1
implicit none
!seed for random number generator
integer :: seed = 987654321
real :: exact = .78540 !exact answer
integer :: N, iX
real :: sumF, sumF2, fx
real :: f_ave, f2_ave, sigma

do
  print *, 'Enter number of points &
  & (0 to stop)'
  read *, N
  if ( N == 0 ) STOP

  sumF = 0.    !zero sums
  sumF2 = 0.

  !loop over samples
  do iX = 1, n
    !integrand
    fx = fun(ran(seed))
    !add contributions to sums
    sumF = sumF + fX
    sumF2 = sumF2 + fX**2
  end do

  f_ave = sumF/N
  f2_ave = sumF2/N

  !incerteza
  sigma = sqrt((f2_ave - f_ave**2) / N)

  print *, 'integral=',f_ave,'+/-', &
  & sigma,'error=',exact-f_ave
end do
contains
real function fun(x)
  implicit none
  real, intent(in) :: x
  fun = 1./(1. + x**2)
end function
end program Ex1
```

[2] S. E. Koonin e D. C. Meredith (1998). *Computational Physics-Fortran Version*.

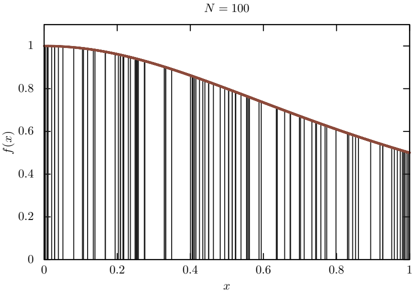
Exemplo 1:

Consideremos o cálculo da integral

$$I = \int_0^1 \frac{dx}{1+x^2} = \frac{\pi}{4} = 0.7854 \ . \tag{8}$$

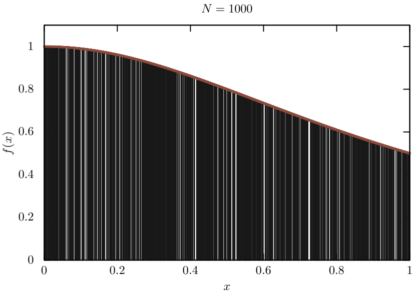


(a) $I = 0.74325 \pm 4.66 \times 10^{-2}$, $\mathrm{E} = 10^{-2}$.

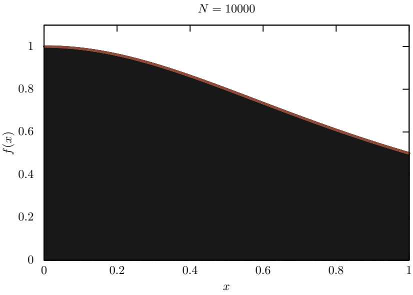


(b) $I = 0.79313 \pm 1.64 \times 10^{-2}$, $\mathrm{E} = 10^{3}$.

Mais alguns resultados.



(a) $I = 0.78127 \pm 4.99 \times 10^{-3}$, $\mathrm{E} = 10^{-3}$.



(b) $I = 0.78922 \pm 1.61 \times 10^{-3}$ $\mathrm{E} = 10^{-3}$.

Peso estatístico

- Podemos reduzir a variância ao multiplicar o integrando por um peso $w(x)$ normalizado,

$$\int_a^b w(x)dx = 1 \ .$$

Peso estatístico

- Podemos reduzir a variância ao multiplicar o integrando por um peso $w(x)$ normalizado,

$$\int_a^b w(x)dx = 1 \ .$$

- A integral pode então ser reescrita como

$$I = \int_a^b dx \frac{f(x)}{w(x)} w(x) \ .$$

Peso estatístico

- Podemos reduzir a variância ao multiplicar o integrando por um peso $w(x)$ normalizado,

$$\int_a^b w(x)dx = 1 \ .$$

- A integral pode então ser reescrita como

$$I = \int_a^b dx \frac{f(x)}{w(x)} w(x) \ .$$

- se fizermos uma mudança de variável, obtemos (detalhes em Koonin e Meredith)

$$I = \int_0^1 dy \frac{f(x(y))}{w(x(y))} \ .$$

Peso estatístico

- A abordagem em MC é avaliar esta integral como

$$I \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(x(y_i))}{w(x(y_i))} \ .$$

- A função do peso estatístico na estimativa de valores.

Exemplo 2:

Consideremos o caso do cálculo da integral na Eq.(8), mas agora com um peso

$$w(x) = (4 - 2x)/3 \ ,$$

com a nova variável de integração sendo

$$y = \frac{x(4 - x)}{3} \ ,$$

que pode ser invertida para $x = 2 - \sqrt{4 - 3y}$.

Script para o cálculo da integral com um peso estatístico.

```fortran
module Ex2_mod
contains
real function fun(x)
  implicit none
  real, intent(in) :: x
  fun = 1./(1. + x**2)
end function

real function weight(x)
  implicit none
  real, intent(in) :: x
  weight = (4. - 2*x)/3.
end function

real function xx(y)
  !x as a function of y
  implicit none
  real, intent(in) :: y
  xx = 2. - sqrt(4. - 3*y)
end function
end module Ex2_mod

program Ex2
use Ex2_mod !Import module
implicit none
integer :: seed = 987654321
real, parameter :: exact = .78540
integer :: N, iX
real :: sumF, sumF2, fX, f_ave, f2_ave
real :: sigma, x, y
```

```fortran
do
  print *, 'Enter number of points &
        & (0 to stop)'
  read *, N
  if ( N == 0 ) STOP

  sumF = 0.
  sumF2 = 0.
  do iX = 1, n
    y = ran(seed)
    x = xx(y)
    fx = fun(x) / weight(x) !integrand

    sumF = sumF + fX
    sumF2 = sumF2 + fX**2
  end do
  f_ave = sumF/N
  f2_ave = sumF2/N

  sigma = sqrt((f2_ave - f_ave**2) / N)

  print *, 'integral =', f_ave, '+/-', &
  & sigma, 'error =', exact - f_ave
end do
end program Ex2
```