

Lista 4 - Números Aleatórios na Física Computacional
Física Computacional 2 - 92444 (B) - 2/2024
Prof. Matheus Paes Lima

Observações:

- Todos os exercícios desta lista devem ser resolvidos utilizando programas escritos na linguagem de programação Python 3. Recomendamos o uso do Google Colab, pois ele facilita a integração com o Google Sala de Aula.
- Após finalizar os exercícios, faça o download do arquivo no formato `.ipynb` e envie este arquivo para correção. **Atenção: listas enviadas em outros formatos não serão corrigidas.**

Exercício 1 - Testando geradores de números aleatórios

Implemente o método congruencial linear para geração de números aleatórios uniformes (entre 0 e 1) e compare com o gerador de número aleatórios do pacote `numpy`. Elabore os seguintes gráficos em função da quantidade de números gerados de 1 à 100000 com passos de 10000:

- (a) o valor médio dos números gerados;
- (b) O desvio padrão médio dos números gerados.

Nesse exercício, use os seguintes parâmetros: $a = 1664525$, $c = 1013904223$, $m = 2^{32}$, para o método congruencial linear e $seed = 42$ para ambos os métodos. *Dica: para gerar número uniformes com o python, é possível usar o comando: `numpy.random.uniform(0, 1, N)`.*

Exercício 2 - Integração de Monte Carlo

Implemente o método de integração de Monte Carlo e calcule as seguintes integrais. Compare os valores obtidos com um outro método da sua escolha, isto é método de integração do python e/ou pelas integrais analíticas e/ou fórmulas conhecidas. para todas as integrais, utilize $N_{points} = 10^8$. *Dica: A integração de uma função no python pode ser feita importando o pacote: `from scipy.integrate import quad`.*

- (a) $\int_0^1 x dx$;
- (b) $\int_0^1 e^{-x} dx$;
- (c) $\int_0^\pi \sin(x) dx$;
- (d) $\int_0^1 x^2 + x^3 dx$;
- (e) Volume do elipsoide: $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} \leq 1$, com $a = 1$, $b = 2$ e $c = 3$.

Exercício 3 - O passo do bêbado

Implemente o passo do bêbado em 3 Dimensões. Elabore um gráfico dos deslocamentos em x, y e z em função do número de passos. realize um total de 1000 passos nas três direções e defina a seed padrão do python usando o comando: `numpy.random.seed(42)`. Com base nesse gráfico, responda as seguintes questões:

- (a) Qual o deslocamento lateral máximo em x, y e z;
- (b) Qual o deslocamento lateral médio em x, y e z;

Exercício 4 - Simulação de Monte Carlo do Modelo de Ising 2D

Faça uma simulação do modelo de Ising em duas dimensões usando o algoritmo de Metropolis incluindo condições periódicas de contorno e interações entre primeiros vizinhos. Defina o sistema de spins como uma rede quadrada regular de dimensões 10×10 . Para esse tipo de rede, as configurações de spin podem ser armazenadas em uma matriz, como exemplificado na figura abaixo.

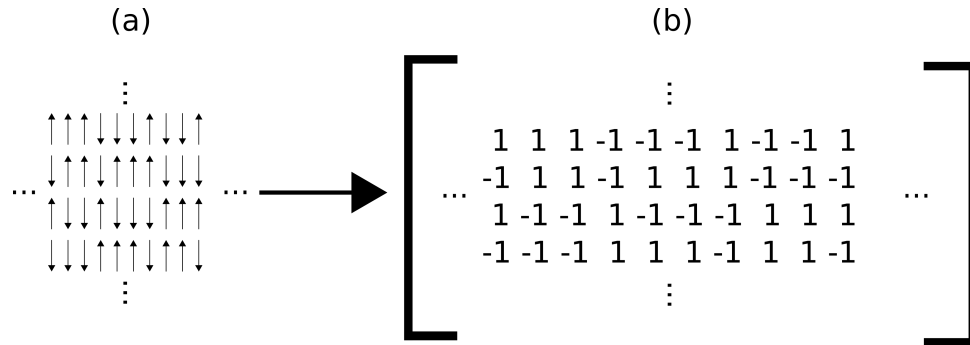


Figura 1: Modelo de Ising em 2 dimensões (a) e representação matricial (b).

Usando os parâmetros:

- $N_{\text{steps}} = 100000$;
- $J = 1$;
- $\mu = 1$;
- $B = 0.1$,

faça um gráfico da variação da energia, magnetização e calor específico médios em função da temperatura para T variando de 1.6 à 4 em passos de 0.1. Além disso, determine a temperatura crítica do sistema. *Dica: Para execução mais rápida do código, use o pacote numba com compilação just-in-time: <https://numba.pydata.org/numba-doc/dev/user/jit.html>*