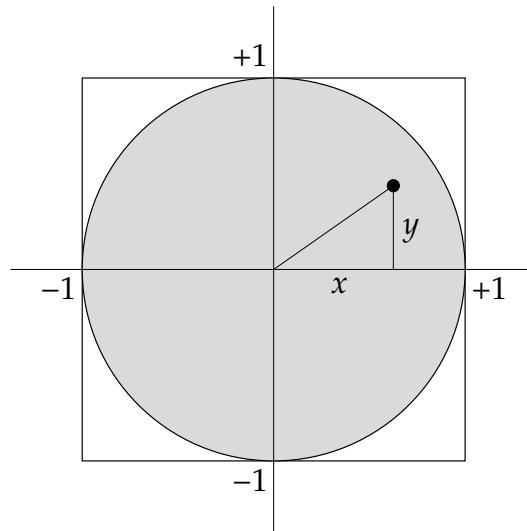


FÍSICA COMPUTACIONAL II

LISTA DE EXERCÍCIOS V - DATA PARA ENTREGA: 26/03/2021 - Valor: 1,5 pt

Problema 1: Volume de uma Hiperesfera

Estime o volume de uma esfera de raio unitário em dez dimensões usando o método de Monte Carlo. Considere o problema equivalente em duas dimensões, a área de um círculo de raio unitário:



A área do círculo, a área sombreada acima, é dada pela integral

$$I = \iint_{-1}^{+1} f(x, y) \, dx \, dy,$$

onde $f(x, y) = 1$ em todos os lugares dentro do círculo e zero em todos os lugares do lado de fora. Em outras palavras,

$$f(x, y) = \begin{cases} 1 & \text{se } x^2 + y^2 \leq 1, \\ 0 & \text{de outra forma.} \end{cases}$$

De maneira que se não conhecermos a área de um círculo, podemos calculá-la usando o método de Monte Carlo. Geramos um conjunto de N pontos aleatórios (x, y) , onde x e y estão no intervalo de -1 até 1 . Então a integral pode ser aproximada por

$$I \simeq \frac{4}{N} \sum_{i=1}^N f(x_i, y_i).$$

Generalize este método para o caso de dez dimensões e escreva um programa para executar o cálculo de Monte Carlo do volume de uma esfera de raio unitário em dez dimensões.

Se tivéssemos de fazer uma integral de dez dimensões da maneira tradicional, levaria muito tempo. Mesmo com apenas 100 pontos ao longo de cada eixo (o que não daria um resultado muito preciso) ainda teríamos $100^{10} = 10^{20}$ pontos para amostrar, o que é impossível em qualquer computador. Mas usando o método de Monte Carlo nós podemos obter um bom resultado com cerca de um milhão de pontos.

Problema 2: Calcule o valor da integral

$$I = \int_0^1 \frac{x^{-1/2}}{e^x + 1} dx,$$

usando a fórmula de amostragem de importância, Eq. (10.42), com $w(x) = x^{-1/2}$, do seguinte modo.

- a) Mostre que a distribuição de probabilidade $p(x)$ a partir do qual os pontos de amostra devem ser sorteados é dada por

$$p(x) = \frac{1}{2\sqrt{x}}$$

e derive a transformação da fórmula para gerar números aleatórios entre 0 e 1 a partir desta distribuição.

- b) Usando sua fórmula, amostre $N = 1\,000\,000$ pontos aleatórios e avalie a integral. Você achará um valor próximo de 0.84.

Problema 3: O Modelo de Ising

O modelo de Ising é um modelo teórico para o magnetismo. A magnetização de materiais magnéticos é constituída pela combinação de muitos pequenos dipolos magnéticos espalhados por todo o material. Se esses dipolos apontarem para direções aleatórias, a magnetização geral do sistema será próxima de zero, mas se eles se alinham de modo que todos ou a maioria deles apontem na mesma direção, então o sistema pode adquirir um momento magnético macroscópico - torna-se magnetizado. O modelo Ising é um modelo desse processo em que os momentos individuais são representados por dipolos ou "spins" dispostos em uma grade ou rede:

↑	↓	↑	↑	↓	↑	↓	↓
↑	↓	↑	↓	↓	↓	↑	↓
↓	↑	↓	↑	↓	↓	↑	↑
↑	↑	↑	↓	↑	↓	↓	↑
↓	↑	↑	↓	↑	↑	↓	↓
↓	↓	↑	↑	↑	↑	↓	↓
↓	↓	↑	↓	↑	↓	↑	↓
↑	↑	↓	↑	↓	↑	↑	↓

Neste caso estamos usando uma rede quadrada em duas dimensões, embora o modelo possa ser definido em princípio para qualquer rede em qualquer número de dimensões.

Os spins, neste modelo simples, são restritos apenas em duas direções, para cima e para baixo. Matematicamente os spins são representados pelas variáveis $s_i = \pm 1$ nos pontos da rede, $+1$ para apontamentos para cima e -1 quando o spin apontar para baixo. Dipolos em ímãs reais podem tipicamente apontar em qualquer direção espacial, não apenas para cima ou para baixo, mas o modelo de Ising, com sua restrição a apenas as duas direções, captura uma importante e grande parte da física, sendo significativamente mais simples de entender.

Outra característica importante de muitos materiais magnéticos é que os dipolos individuais no material podem interagir magneticamente de forma a que ele seja energeticamente favorável para se alinhar na mesma direção. A energia potencial magnética devido a interação de dois dipolos é proporcional ao seu produto, mas o modelo de Ising simplifica isso para apenas um produto $s_i s_j$ para spins nos sites i e j da rede, uma vez que as rotações são escalares unidimensionais, não vetores. Então a energia atual da interação é $-J s_i s_j$, onde J é uma constante de interação positiva. O sinal de menos garante que as interações são *ferromagnéticas*, o que significa que a energia é menor quando os dipolos estão alinhados. A interação ferromagnética implica que o material se magnetizará se tiver chance. (Em alguns materiais, a interação tem o sinal oposto de modo que os dipolos preferem ser anti-alinhados. Tal material é chamado *antiferromagnético*, mas não vamos olhar para o caso antiferromagnético aqui.)

Normalmente, é assumido que os spins interagem apenas com aqueles que são imediatamente vizinhos a eles na rede, o que fornece uma energia total para o sistema igual a

$$E = -J \sum_{\langle ij \rangle} s_i s_j,$$

onde a notação $\langle ij \rangle$ indica uma soma em pares i, j que são adjacentes na rede. Na rede quadrada que usamos neste exercício, cada spin tem quatro primeiros vizinhos com os quais interage.

Escreva um programa para executar uma simulação de Monte Carlo do modelo Ising na rede quadrada para um sistema de 20×20 spins. Você irá precisar configurar variáveis para manter o valor ± 1 do spin em cada sítio da rede, provavelmente usando uma matriz de inteiros em duas dimensões, e então siga os seguintes passos.

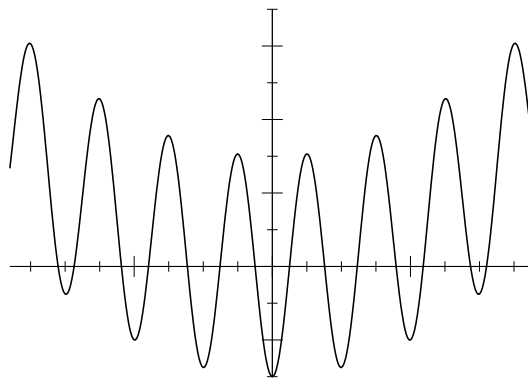
- a) Primeiro escreva a função para calcular a energia total do sistema, como dado pela equação acima. Isto é, para uma dada variedade de valores de spins, passe por cada par de spins adjacentes e some as contribuições, $s_i s_j$ para todos eles, então multiplique por $-J$. Dica 1: Cada par exclusivo de spins adjacentes surge apenas uma vez na soma. Assim, existe um termo $-J s_1 s_2$ se os spins 1 e 2 são adjacentes uns aos outros, mas você não precisa de um termo $-J s_2 s_1$. Dica 2: Para que seu programa final seja executado em um período de tempo razoável, você achará útil se você puder achar uma forma de calcular a energia utilizando a habilidade do Python para fazer aritmética com matrizes inteiras ao mesmo tempo. Se você for calcular usando laços, seu programa será significativamente mais lento.
- b) Agora use sua função como base para uma simulação do tipo Metropolis do modelo de Ising com $J = 1$ e temperatura $T = 1$ em unidades onde a constante de Boltzmann k_B é também 1. Inicialmente configure as variáveis dos spins aleatoriamente para

± 1 , de modo que, em média, metade deles estão para cima e a outra metade para baixo, dando uma magnetização total de aproximadamente zero. Em seguida, escolha um spin aleatório, gire-o e calcule a energia nova depois que ele é virado e, portanto, também a mudança de energia como resultado do giro. Então, decida se deve aceitar o giro usando a fórmula de aceitação Metropolis. Se o movimento for rejeitado, você terá que girar o spin de volta para onde estava. Caso contrário, você mantém a spin invertido. Agora, repita esse processo para muitos movimentos.

- c) Faça um gráfico da magnetização total $M = \sum_i s_i$ do sistema em função do tempo para um milhão de passos de Monte Carlo. Você deve ver que o sistema desenvolve uma "magnetização espontânea", com valores diferentes de zero para a magnetização geral. Dica: Enquanto você está trabalhando com seu programa, faça corridas mais curtas, tente dez mil passos de cada vez. Uma vez que o programa esteja funcionando corretamente, faça um longo período de um milhão de passos para obter os resultados finais.
- d) Execute seu programa várias vezes e observe o sinal da magnetização que se desenvolve, positivo ou negativo. Descreva o que você encontrou e dê uma breve explicação sobre o que está acontecendo.
- e) (Extra:+0.2) Faça uma segunda versão do seu programa que produza uma animação do sistema usando o pacote vpython, com esferas ou quadrados de duas cores, em uma grade regular, para representar os spins para cima e para baixo. Rode com temperatura $T = 1$ e observe o comportamento do sistema. Em seguida, rode mais duas vezes para as temperaturas $T = 2$ e $T = 3$. Explique brevemente o que você vê nas três execuções. Como e por que o comportamento do sistema muda a medida que aumenta a temperatura?

Problema 4: O mínimo global de uma função

Considere a função $f(x) = x^2 - \cos 4\pi x$, que tem a seguinte aparência:



Claramente o mínimo global desta função está em $x = 0$.

- a) Escreva um programa para confirmar este fato usando recozimento simulado, começando em $x = 2$, com movimentos de Monte Carlo da forma $x \rightarrow x + \delta$ onde δ é um número aleatório sorteado de uma distribuição Gaussiana com média zero e desvio padrão de um. (Veja a seção 10.1.6 do Newman para lembrar de como gerar números aleatórios

Gaussianos). Use um esquema de resfriamento exponencial e ajuste as temperaturas iniciais e finais, bem como a constante exponencial, até que você ache valores que dão boas respostas em um tempo razoável. Faça o seu programa plotar os valores de x em função do tempo durante sua simulação e faça ele imprimir o valor final de x no final. Você vai achar o gráfico mais fácil de interpretar se você usar pontos ao invés de linhas, com um comando da forma `plot(x, ". ")` ou similar.

- b) Agora adapte seu programa para achar o mínimo de uma função mais complicada, $f(x) = \cos x + \cos \sqrt{2}x + \cos \sqrt{3}x$ no intervalo $0 < x < 50$.

Dica: A resposta correta para a parte (b) está em torno de $x = 16$, mas há mínimos que competem em torno de $x = 2$ e $x = 42$, que seu programa pode achar. Em simulações do mundo real, muitas vezes basta achar qualquer solução razoável para um problema, não necessariamente a que é melhor absolutamente. Então o fato de o programa as vezes convergir para estas outras soluções não é algo necessariamente ruim.