

Introdução à Física Computacional II (4300318)

Prof. André Vieira
apvieira@if.usp.br
Sala 3120 – Edifício Principal

Aula 11

Amostragem por importância na
mecânica estatística
O método das cadeias de Markov

Simulações de Monte Carlo

- A mecânica estatística lida com processos que envolvem sistemas complexos, quase sempre compostos de muitas partículas, em que ingredientes aleatórios (ou aparentemente aleatórios) desempenham um papel central.
- Esses ingredientes aleatórios podem ser representados por números aleatórios, permitindo a simulação computacional desses sistemas.

Descrição estatística de um sistema físico

- Trocas de energia entre um sistema físico e seu entorno são essenciais em fenômenos térmicos. Por isso, a descrição estatística de um sistema físico passa por caracterizar os diferentes estados de um sistema físico em termos das energias correspondentes a esses estados.
- As interações com o entorno fazem com que o sistema “passeie” entre seus diferentes estados, nos quais não apenas a energia pode variar, mas também outras grandezas tais como volume (em fluidos), a magnetização (em ímãs) etc.

Descrição estatística de um sistema físico

- Por simplicidade, vamos considerar um sistema físico cujos estados tenham energias discretas. Vamos representar o conjunto de estados por $\{j\}$ e o conjunto de energias correspondentes por $\{E_j\}$. Note que mais de um estado pode corresponder à mesma energia.

A distribuição de Boltzmann

- Considere que um sistema físico seja mantido a uma temperatura absoluta constante T . Como se discute em Física 2, no equilíbrio térmico, a probabilidade de observar o sistema em um **estado** i caracterizado pela energia E_i é dada pela **distribuição de Boltzmann**

$$P_i = \frac{e^{-\beta E_i}}{\sum_j e^{-\beta E_j}} \equiv \frac{e^{-\beta E_i}}{Z}, \quad \beta = (k_B T)^{-1}$$

Nas unidades do SI, a constante de Boltzmann k_B vale $1,38 \times 10^{-23}$ J/K. A quantidade representada por Z é chamada de “função de partição”.

A distribuição de Boltzmann

- Se quisermos determinar o valor médio no equilíbrio de uma quantidade X qualquer associada ao sistema, temos que calcular a média ponderada sobre os valores X_j de X em cada estado do sistema:

$$\langle X \rangle = \sum_j X_j P_j = \frac{\sum_j X_j e^{-\beta E_j}}{Z}.$$

- Para calcular o valor médio de X , basta então que computemos as somas acima.

A distribuição de Boltzmann

- Se quisermos determinar o valor médio no equilíbrio de uma quantidade X qualquer associada ao sistema, temos que calcular a média ponderada sobre os valores X_j de X em cada estado do sistema:

$$\langle X \rangle = \sum_j X_j P_j = \frac{\sum_j X_j e^{-\beta E_j}}{Z}.$$

- Analiticamente, esse cálculo pode ser feito em muito poucos casos. (Vejam a disciplina de Mecânica Estatística.)

A distribuição de Boltzmann

- Se quisermos determinar o valor médio no equilíbrio de uma quantidade X qualquer associada ao sistema, temos que calcular a média ponderada sobre os valores X_j de X em cada estado do sistema:

$$\langle X \rangle = \sum_j X_j P_j = \frac{\sum_j X_j e^{-\beta E_j}}{Z}.$$

- Podemos entretanto efetuar o cálculo numericamente. Mas quantas configurações estão envolvidas nas somas?

Sistemas de dois níveis

- Vamos considerar o caso em que o sistema é composto de partículas cada uma das quais pode estar em um de dois estados individuais. Um exemplo é o modelo de Ising, em que cada partícula possui um momento magnético que pode “apontar” apenas para cima ou para baixo. Chamamos essas partículas de *spins*.
- Se o sistema é composto por 2 spins, há 4 possíveis estados coletivos (ou configurações):



Sistemas de dois níveis

- Vamos considerar o caso em que o sistema é composto de partículas cada uma das quais pode estar em um de dois estados individuais. Um exemplo é o modelo de Ising, em que cada partícula possui um momento magnético que pode “apontar” apenas para cima ou para baixo. Chamamos essas partículas de *spins*.
- Com N spins, quantas são as configurações?

$$M_{\text{total}} = \underbrace{2 \times 2 \times \cdots \times 2}_{N \text{ fatores}} = 2^N$$

Sistemas de dois níveis

- Para um sistema cúbico contendo 10 spins por aresta, quantas são as configurações?

$$M_{\text{total}} = 2^{10^3} = \left(2^{10}\right)^{100} \simeq \left(10^3\right)^{100} = 10^{300}$$

- Se fôssemos capazes de processar 10 bilhões de configurações por segundo, levaríamos quantos anos para processar todas as configurações?

Sistemas de dois níveis

- Para um sistema cúbico contendo 10 spins por aresta, quantas são as configurações?

$$M_{\text{total}} = 2^{10^3} = \left(2^{10}\right)^{100} \simeq \left(10^3\right)^{100} = 10^{300}$$

- Se fôssemos capazes de processar 10 bilhões de configurações por segundo, levaríamos quantos anos para processar todas as configurações?

$$\sim 10^{283} \text{ anos!}$$

Estimando a média por Monte Carlo

- Sendo impossível levar em conta todas as configurações, podemos então sortear N estados e estimar as médias a partir de

$$\langle X \rangle \simeq \frac{\sum_{k=1}^N X_k e^{-\beta E_k}}{\sum_{k=1}^N e^{-\beta E_k}} = \frac{\sum_{k=1}^N X_k P_k}{\sum_{k=1}^N P_k}.$$

- Note que agora X_k e E_k referem-se ao k -ésimo estado sorteado.

Estimando a média por Monte Carlo

$$\langle X \rangle \simeq \frac{\sum_{k=1}^N X_k e^{-\beta E_k}}{\sum_{k=1}^N e^{-\beta E_k}}.$$

- Essa abordagem, no entanto, não funciona, uma vez que o fator exponencial de Boltzmann é muito pequeno para qualquer estado tal que $E_k \gg k_B T$, condição válida, em geral, para a vasta maioria dos estados (exceto em $T \rightarrow \infty$). É improvável que escolhamos ao acaso estados que efetivamente contribuam para a soma original. O que podemos fazer então?

Amostragem por importância

- Temos que escolher os estados de modo não uniforme, para que tenhamos chances de selecionar aqueles estados importantes, ou seja, que de fato contribuem para a soma.
- Vamos definir a média ponderada

$$\langle g \rangle_w = \frac{\sum_j g_j w_j}{\sum_j w_j},$$

em que g é uma grandeza qualquer e $\{w_j\}$ é um conjunto de pesos definidos para cada estado.

Amostragem por importância

- Escolhendo $g_j = X_j P_j / w_j$, temos

$$\left\langle \frac{X P}{w} \right\rangle_w = \frac{\sum_j (X_j P_j / w_j) w_j}{\sum_j w_j} = \frac{\sum_j X_j P_j}{\sum_j w_j} = \frac{\langle X \rangle}{\sum_j w_j}.$$

$$\langle X \rangle = \left\langle \frac{X P}{w} \right\rangle_w \sum_j w_j \simeq \left(\frac{1}{N} \sum_{k=1}^N \frac{X_k P_k}{w_k} \right) \sum_j w_j.$$

- Na expressão acima, os estados k são sorteados a partir de w e a soma sobre j envolve **todos** os estados. O truque é escolher w para privilegiar os estados importantes.

Amostragem por importância

$$\langle X \rangle = \left\langle \frac{X P}{w} \right\rangle_w \sum_j w_j \simeq \left(\frac{1}{N} \sum_{k=1}^N \frac{X_k P_k}{w_k} \right) \sum_j w_j.$$

- Uma escolha fisicamente intuitiva é $w_k = P_k$. Isso faz com que os estados sejam selecionados segundo seu peso exponencial de Boltzmann, que é como estão presentes no equilíbrio. Ou seja, estamos **simulando** o comportamento térmico. Além disso, a soma sobre j acima é por definição igual a 1. Com essa escolha:

$$\langle X \rangle \simeq \frac{1}{N} \sum_{k=1}^N X_k.$$

O método da cadeia de Markov

- A dificuldade trazida por essa escolha é que não conseguimos escolher os estados a partir da probabilidade P_k , que envolve a função de partição, cujo cálculo não sabemos fazer. (Se soubéssemos, não precisaríamos da simulação!)
- A saída é percorrer uma “cadeia de Markov” de estados, alterando a cada vez a configuração de elementos do sistema, com o auxílio de **taxas de transição** (probabilidades por unidade de tempo) que façam o sistema visitar os estados com probabilidade P_k .

Processos estocásticos markovianos são aqueles em que a probabilidade de um evento depende apenas do estado atual, e não dos estados anteriores. Diz-se que esses processos não “guardam memória”.

O método da cadeia de Markov

- Concretamente, se em um passo da cadeia o estado do sistema é i , o próximo estado é escolhido de tal forma que a probabilidade de transição $\Delta t W_{ij}$ do estado i para o estado j durante aquele passo não dependa do tempo.
- Para avaliarmos se após um grande número de passos os estados visitados na cadeia de Markov seguem a distribuição de Boltzmann, precisamos tratar com as probabilidades de que os diversos estados sejam visitados em cada passo da cadeia.

Aqui Δt é a duração de um passo da cadeia de Markov.

O método da cadeia de Markov

- Concretamente, se em um passo da cadeia o estado do sistema é i , o próximo estado é escolhido de tal forma que a probabilidade de transição $\Delta t W_{ij}$ do estado i para o estado j durante aquele passo não dependa do tempo.
- A probabilidade $p_i(t)$ de que o sistema esteja no estado i no instante t , ao longo da cadeia de Markov, é determinada por três possibilidades:
 - o estado anterior era i e se manteve;
 - o estado anterior era i e mudou para outro;
 - o estado anterior era outro e mudou para i .

O método da cadeia de Markov

- Concretamente, se em um passo da cadeia o estado do sistema é i , o próximo estado é escolhido de tal forma que a probabilidade de transição $\Delta t W_{ij}$ do estado i para o estado j durante aquele passo não dependa do tempo.
- Se $p_i(t)$ é a probabilidade de que o sistema esteja no estado i no instante t , então vale a “equação mestra”

$$p_i(t + \Delta t) = \underbrace{p_i(t)}_{\text{ficando em } i} + \Delta t \sum_{j \neq i} \left[\underbrace{W_{ji} p_j(t)}_{\text{chegando a } i} - \underbrace{W_{ij} p_i(t)}_{\text{saindo de } i} \right].$$

Aqui Δt é a duração de um passo da cadeia de Markov.

O método da cadeia de Markov

$$p_i(t + \Delta t) = p_i(t) + \Delta t \sum_{j \neq i} [W_{ji} p_j(t) - W_{ij} p_i(t)].$$

- No equilíbrio, queremos obter a distribuição de Boltzmann, portanto temos que impor

$$\lim_{t \rightarrow \infty} p_i(t) = P_i = \frac{e^{-\beta E_i}}{Z}.$$

- Substituindo na equação mestra, concluimos que devemos escolher o conjunto $\{W_{ij}\}$ tal que

$$\sum_{j \neq i} (W_{ji} P_j - W_{ij} P_i) = 0.$$

O método da cadeia de Markov

$$\sum_{j \neq i} (W_{ji} P_j - W_{ij} P_i) = 0.$$

- Uma forma de satisfazer a equação acima é tornar nulo cada termo entre parênteses, ou seja, impor para quaisquer i e j que

$$\frac{W_{ij}}{W_{ji}} = \frac{P_j}{P_i} = \frac{e^{-\beta E_j} / Z}{e^{-\beta E_i} / Z} = e^{-\beta (E_j - E_i)},$$

condição chamada de “balanceamento detalhado”. Note o cancelamento da função de partição, que não precisamos conhecer!

O método da cadeia de Markov

$$\frac{W_{ij}}{W_{ji}} = \frac{P_j}{P_i} = \frac{e^{-\beta E_j} / Z}{e^{-\beta E_i} / Z} = e^{-\beta (E_j - E_i)},$$

- O balanceamento detalhado sugere uma escolha para as taxas de transição, mas não demonstramos que, se o respeitarmos, a distribuição de Boltzmann será sempre atingida para tempos longos, qualquer que seja a condição inicial. Para essa demonstração, veja Newman, apêndice D.

A abordagem adotada por Newman é ligeiramente diferente daquela que adotamos aqui. Ele trabalha com probabilidades de transição $T_{ij} = \Delta t W_{ij}$ cuja soma sobre j vale 1. Deve-se escolher Δt com isso em mente.

O algoritmo de Metropolis

- Há muitas formas de escolher as taxas de transição para satisfazer o balanceamento detalhado. A mais simples delas é o **algoritmo de Metropolis**:

$$W_{ij} = \begin{cases} 1, & \text{se } E_j < E_i \\ e^{-\beta(E_j - E_i)}, & \text{se } E_j \geq E_i \end{cases}.$$

- Verifique explicitamente que, com a escolha acima, tanto se $E_j \geq E_i$ quanto se $E_j < E_i$ vale

$$\frac{W_{ij}}{W_{ji}} = e^{-\beta(E_j - E_i)}.$$

Simulações de Monte Carlo

- Eis a receita para implementar uma simulação de Monte Carlo com uma cadeia de Markov.
 1. Escolha um estado inicial aleatório.
 2. Escolha aleatoriamente um movimento a partir de um conjunto de possibilidades (por exemplo, mudar a configuração de uma única partícula).
 3. Calcule a taxa de transição W_{ij} associada.
 4. Aceite o movimento com probabilidade W_{ij} ; em particular, movimentos que diminuem a energia são sempre aceitos.
 5. Meça e acumule as quantidades de interesse.
 6. Repita a partir do passo 2.

Simulações de Monte Carlo

- O conjunto de possíveis movimentos tem que permitir que todos os estados do sistema sejam visitados. (O algoritmo deve ser **ergódico**.)
- Para aceitar um movimento com probabilidade W_{ij} , você pode compará-la com um número aleatório entre 0 e 1.
- As quantidades de interesse devem ser medidas e acumuladas mesmo que um movimento seja rejeitado.
- O número de movimentos para atingir a distribuição de Boltzmann, conhecido como **tempo de equilíbrio**, pode ser longo!

Exemplo: o gás ideal quântico

- As partículas de um gás ideal não interagem diretamente, de modo que a energia total do gás varia apenas quando a energia cinética varia. Em Física 2, discute-se o gás ideal clássico, cuja energia cinética pode variar continuamente. A energia cinética de uma partícula quântica de massa m presa a uma caixa cúbica de lado L , no entanto, somente pode assumir valores discretos dados por

$$E(n_x, n_y, n_z) = \frac{\pi^2 \hbar^2}{2mL^2} (n_x^2 + n_y^2 + n_z^2) \equiv \epsilon (n_x^2 + n_y^2 + n_z^2),$$

em que n_x , n_y e n_z são inteiros quaisquer.

Exemplo: o gás ideal quântico

- Em baixas temperaturas ($k_B T \ll \epsilon$), determinar os possíveis estados de uma coleção de partículas quânticas requer levar em conta eventuais efeitos do princípio da exclusão de Pauli. Esses efeitos tornam-se irrelevantes em altas temperaturas ($k_B T \gg \epsilon$). Vamos supor que as partículas são bósons (ou seja, que não estão sujeitas ao princípio da exclusão).
- A energia cinética de um gás de N partículas é

$$E = \sum_{i=1}^N E(n_x^{(i)}, n_y^{(i)}, n_z^{(i)}) = \epsilon \sum_{i=1}^N \left[\left(n_x^{(i)}\right)^2 + \left(n_y^{(i)}\right)^2 + \left(n_z^{(i)}\right)^2 \right].$$

Exemplo: o gás ideal quântico

- Vamos utilizar uma simulação de Monte Carlo para estimar a energia cinética desse gás em altas temperaturas.
 - Vamos partir do estado fundamental, em que todos os n s são iguais a 0.
 - Admitiremos movimentos que alterem um n ao acaso por uma quantidade ± 1 . (Essa escolha é ergódica?) Por exemplo, alterar um único n_x significa alterar a energia segundo

$$\begin{aligned}\Delta E &= \epsilon \left[(n_x \pm 1)^2 + n_y^2 + n_z^2 \right] - \epsilon \left[n_x^2 + n_y^2 + n_z^2 \right] \\ &= \epsilon \left[(n_x^2 \pm 2n_x + 1) + n_y^2 + n_z^2 \right] - \epsilon \left[n_x^2 + n_y^2 + n_z^2 \right] \\ &= \epsilon (\pm 2n_x + 1).\end{aligned}$$

Exemplo: o gás ideal quântico

```
# Parâmetros e constantes da simulação
N = 10**5                                # Número de átomos
L = 1e-6                                  # Lado do cubo
h = 1.055e-34                             # Valor da constante de Planck dividida por 2*pi
m = 6.645e-27                             # Massa de um átomo de hélio-4
epsilon = (pi*h/L)**2/(2*m)               # Escala de energia
kB = 1.381e-23                             # Constante de Boltzmann
T = 10*epsilon/kB                         # Temperatura
Passos = 300                              # Número de passos de Monte Carlo da simulação
kBT = kB*T

# Matriz que armazena os números quânticos
n = zeros([N,3],int)                      # Partimos do estado fundamental

# Laço principal
E_lista = []                              # Lista para armazenar a energia
E = 0.0
for i in range(N):
    E += n[i,0]**2 + n[i,1]**2 + n[i,2]**2
E *= epsilon
for k in range(Passos):                   # Percorremos os passos de Monte Carlo
    # Escolhemos um número quântico que tentaremos alterar
    for l in range(N):                   # Cada partícula em média tem 1 chance de mudar
        i = randrange(N)                # Partícula
        j = randrange(3)                # Número quântico da partícula
        if random() < 0.5:              # Com probabilidade 1/2 propomos n->n+1
            dn = 1
            dE = (2*n[i,j]+1)*epsilon    # Variação da energia
        else:                            # Caso contrário propomos n->n-1
            dn = -1
            dE = (-2*n[i,j]+1)*epsilon
        if random() < exp(-dE/kBT):      # Decidimos se aceitamos a alteração
            n[i,j] += dn
            E += dE
    # Registramos a energia a cada passo de Monte Carlo
    E_lista.append(E/N/kBT)              # Energia por partícula (medida em kB*T)
```

Exemplo: o gás ideal quântico

```
# Parâmetros e constantes da simulação
N = 10**5                      # Número de átomos
L = 1e-6                       # Lado do cubo
h = 1.055e-34                  # Valor da constante de Planck dividida por 2*pi
m = 6.645e-27                  # Massa de um átomo de hélio-4
epsilon = (pi*h/L)**2/(2*m)     # Escala de energia
kB = 1.381e-23                  # Constante de Boltzmann
T = 10*epsilon/kB               # Temperatura
Passos = 300                    # Número de passos de Monte Carlo da simulação
kBT = kB*T
```

```
# Matriz que armazena os números quânticos
```

Como o fator de Boltzmann envolve a razão entre a energia e $k_B T$, é conveniente definir a temperatura como um múltiplo de ϵ/k_B .

```
for i in range(N):
    E += n[i,0]**2 + n[i,1]**2 + n[i,2]**2
E *= epsilon
for k in range(Passos):        # Percorremos os passos de Monte Carlo
    # Escolhemos um número quântico que tentaremos alterar
    for l in range(N):         # Cada partícula em média tem 1 chance de mudar
        i = randrange(N)      # Partícula
        j = randrange(3)      # Número quântico da partícula
        if random() < 0.5:     # Com probabilidade 1/2 propomos n->n+1
            dn = 1
            dE = (2*n[i,j]+1)*epsilon # Variação da energia
        else:                  # Caso contrário propomos n->n-1
            dn = -1
            dE = (-2*n[i,j]+1)*epsilon
        if random() < exp(-dE/kBT): # Decidimos se aceitamos a alteração
            n[i,j] += dn
            E += dE

    # Registramos a energia a cada passo de Monte Carlo
    E_lista.append(E/N/kBT)      # Energia por partícula (medida em kB*T)
```


Exemplo: o gás ideal quântico

```
# Parâmetros e constantes da simulação
N = 10**5                # Número de átomos
L = 1e-6                 # Lado do cubo
h = 1.055e-34            # Valor da constante de Planck dividida por 2*pi
m = 6.645e-27            # Massa de um átomo de hélio-4
epsilon = (pi*h/L)**2/(2*m) # Escala de energia
kB = 1.381e-23           # Constante de Boltzmann
T = 10*epsilon/kB        # Temperatura
Passos = 300             # Número de passos de Monte Carlo da simulação
kBT = kB*T
```

```
# Matriz que armazena os números quânticos
```

Um passo de Monte Carlo corresponde a 1 tentativa de movimento para cada partícula do sistema. Vamos registrar a energia somente após finalizar cada passo de Monte Carlo.

```
E += n[i,0]**2 + n[i,1]**2 + n[i,2]**2
E *= epsilon
for k in range(Passos):    # Percorremos os passos de Monte Carlo
    # Escolhemos um número quântico que tentaremos alterar
    for l in range(N):     # Cada partícula em média tem 1 chance de mudar
        i = randrange(N)   # Partícula
        j = randrange(3)   # Número quântico da partícula
        if random() < 0.5: # Com probabilidade 1/2 propomos n->n+1
            dn = 1
            dE = (2*n[i,j]+1)*epsilon # Variação da energia
        else:              # Caso contrário propomos n->n-1
            dn = -1
            dE = (-2*n[i,j]+1)*epsilon
        if random() < exp(-dE/kBT): # Decidimos se aceitamos a alteração
            n[i,j] += dn
            E += dE

    # Registramos a energia a cada passo de Monte Carlo
    E_lista.append(E/N/kBT)    # Energia por partícula (medida em kB*T)
```

Exemplo: o gás ideal quântico

```
# Parâmetros e constantes da simulação
N = 10**5                                # Número de átomos
L = 1e-6                                  # Lado do cubo
h = 1.055e-34                             # Valor da constante de Planck dividida por 2*pi
m = 6.645e-27                             # Massa de um átomo de hélio-4
epsilon = (pi*h/L)**2/(2*m)               # Escala de energia
kB = 1.381e-23                             # Constante de Boltzmann
T = 10*epsilon/kB                         # Temperatura
Passos = 300                              # Número de passos de Monte Carlo da simulação
kBT = kB*T
```

```
# Matriz que armazena os números quânticos
n = zeros([N,3],int)                     # Partimos do estado fundamental
```

```
# Laço principal
E_lista = []                             # Lista para armazenar a energia
```

Especificamos a configuração (ou estado) inicial, que aqui será o estado de mais baixa energia, em que todos os números quânticos são nulos.

```
for i in range(N):                        # Cada partícula em media tem 1 chance de mudar
    i = randrange(N)                      # Partícula
    j = randrange(3)                      # Número quântico da partícula
    if random() < 0.5:                    # Com probabilidade 1/2 propomos n->n+1
        dn = 1
        dE = (2*n[i,j]+1)*epsilon        # Variação da energia
    else:                                 # Caso contrário propomos n->n-1
        dn = -1
        dE = (-2*n[i,j]+1)*epsilon
    if random() < exp(-dE/kBT):            # Decidimos se aceitamos a alteração
        n[i,j] += dn
        E += dE

# Registramos a energia a cada passo de Monte Carlo
E_lista.append(E/N/kBT)                  # Energia por partícula (medida em kB*T)
```

Exemplo: o gás ideal quântico

```
# Parâmetros e constantes da simulação
N = 10**5                                # Número de átomos
L = 1e-6                                  # Lado do cubo
h = 1.055e-34                             # Valor da constante de Planck dividida por 2*pi
m = 6.645e-27                             # Massa de um átomo de hélio-4
epsilon = (pi*h/L)**2/(2*m)               # Escala de energia
kB = 1.381e-23                             # Constante de Boltzmann
T = 10*epsilon/kB                         # Temperatura
Passos = 300                              # Número de passos de Monte Carlo da simulação
kBT = kB*T

# Matriz que armazena os números quânticos
n = zeros([N,3],int)                      # Partimos do estado fundamental
```

A cada tentativa, sorteamos um número quântico de alguma partícula propomos modificá-lo.

```
for i in range(N):
    E += n[i,0]**2 + n[i,1]**2 + n[i,2]**2
E *= epsilon
for k in range(Passos):                  # Percorremos os passos de Monte Carlo
    # Escolhemos um número quântico que tentaremos alterar
    for l in range(N):                  # Cada partícula em média tem 1 chance de mudar
        i = randrange(N)                # Partícula
        j = randrange(3)                # Número quântico da partícula
        if random() < 0.5:              # Com probabilidade 1/2 propomos n->n+1
            dn = 1
            dE = (2*n[i,j]+1)*epsilon    # Variação da energia
        else:                            # Caso contrário propomos n->n-1
            dn = -1
            dE = (-2*n[i,j]+1)*epsilon
        if random() < exp(-dE/kBT):      # Decidimos se aceitamos a alteração
            n[i,j] += dn
            E += dE
    # Registramos a energia a cada passo de Monte Carlo
    E_lista.append(E/N/kBT)              # Energia por partícula (medida em kB*T)
```

Exemplo: o gás ideal quântico

```
# Parâmetros e constantes da simulação
N = 10**5                                # Número de átomos
L = 1e-6                                  # Lado do cubo
h = 1.055e-34                             # Valor da constante de Planck dividida por 2*pi
m = 6.645e-27                             # Massa de um átomo de hélio-4
epsilon = (pi*h/L)**2/(2*m)                # Escala de energia
kB = 1.381e-23                             # Constante de Boltzmann
T = 10*epsilon/kB                         # Temperatura
Passos = 300                              # Número de passos de Monte Carlo da simulação
kBT = kB*T

# Matriz que armazena os números quânticos
n = zeros([N,3],int)                      # Partimos do estado fundamental

# Laço principal
E_lista = []                              # Lista para armazenar a energia
E = 0.0
for i in range(N):
```

Damos chances iguais para que o número quântico aumente ou diminua, calculando a variação correspondente da energia.

```
# Escolhemos um número quântico que tentaremos alterar.
for l in range(N):                        # Cada partícula em média tem 1 chance de mudar
    i = randrange(N)                      # Partícula
    j = randrange(3)                      # Número quântico da partícula
    if random() < 0.5:                     # Com probabilidade 1/2 propomos n->n+1
        dn = 1
        dE = (2*n[i,j]+1)*epsilon        # Variação da energia
    else:                                  # Caso contrário propomos n->n-1
        dn = -1
        dE = (-2*n[i,j]+1)*epsilon
    if random() < exp(-dE/kBT):             # Decidimos se aceitamos a alteração
        n[i,j] += dn
        E += dE

# Registramos a energia a cada passo de Monte Carlo
E_lista.append(E/N/kBT)                  # Energia por partícula (medida em kB*T)
```

Exemplo: o gás ideal quântico

```
# Parâmetros e constantes da simulação
N = 10**5                                # Número de átomos
L = 1e-6                                  # Lado do cubo
h = 1.055e-34                             # Valor da constante de Planck dividida por 2*pi
m = 6.645e-27                             # Massa de um átomo de hélio-4
epsilon = (pi*h/L)**2/(2*m)                # Escala de energia
kB = 1.381e-23                             # Constante de Boltzmann
T = 10*epsilon/kB                          # Temperatura
Passos = 300                              # Número de passos de Monte Carlo da simulação
kBT = kB*T

# Matriz que armazena os números quânticos
n = zeros([N,3],int)                      # Partimos do estado fundamental

# Laço principal
E_lista = []                              # Lista para armazenar a energia
E = 0.0
for i in range(N):
    E += n[i,0]**2 + n[i,1]**2 + n[i,2]**2
E *= epsilon
for k in range(Passos):                    # Percorremos os passos de Monte Carlo
    # Escolhemos um número quântico que tentaremos alterar
```

Definida a alteração, determinamos agora se a aceitamos. Note que se a energia diminuir a condição será sempre satisfeita, logo a alteração será sempre aceita.

```
    dE = (-2*n[i,j]+1)*epsilon             # Variação da energia
    else:
        dn = -1
        dE = (-2*n[i,j]+1)*epsilon
        if random() < exp(-dE/kBT):         # Decidimos se aceitamos a alteração
            n[i,j] += dn
            E += dE
    # Registramos a energia a cada passo de Monte Carlo
    E_lista.append(E/N/kBT)                 # Energia por partícula (medida em kB*T)
```


Exemplo: o gás ideal quântico

```
# Parâmetros e constantes da simulação
N = 10**5                                # Número de átomos
L = 1e-6                                  # Lado do cubo
h = 1.055e-34                             # Valor da constante de Planck dividida por 2*pi
m = 6.645e-27                             # Massa de um átomo de hélio-4
epsilon = (pi*h/L)**2/(2*m)               # Escala de energia
kB = 1.381e-23                             # Constante de Boltzmann
T = 10*epsilon/kB                         # Temperatura
Passos = 300                              # Número de passos de Monte Carlo da simulação
kBT = kB*T

# Matriz que armazena os números quânticos
n = zeros([N,3],int)                      # Partimos do estado fundamental

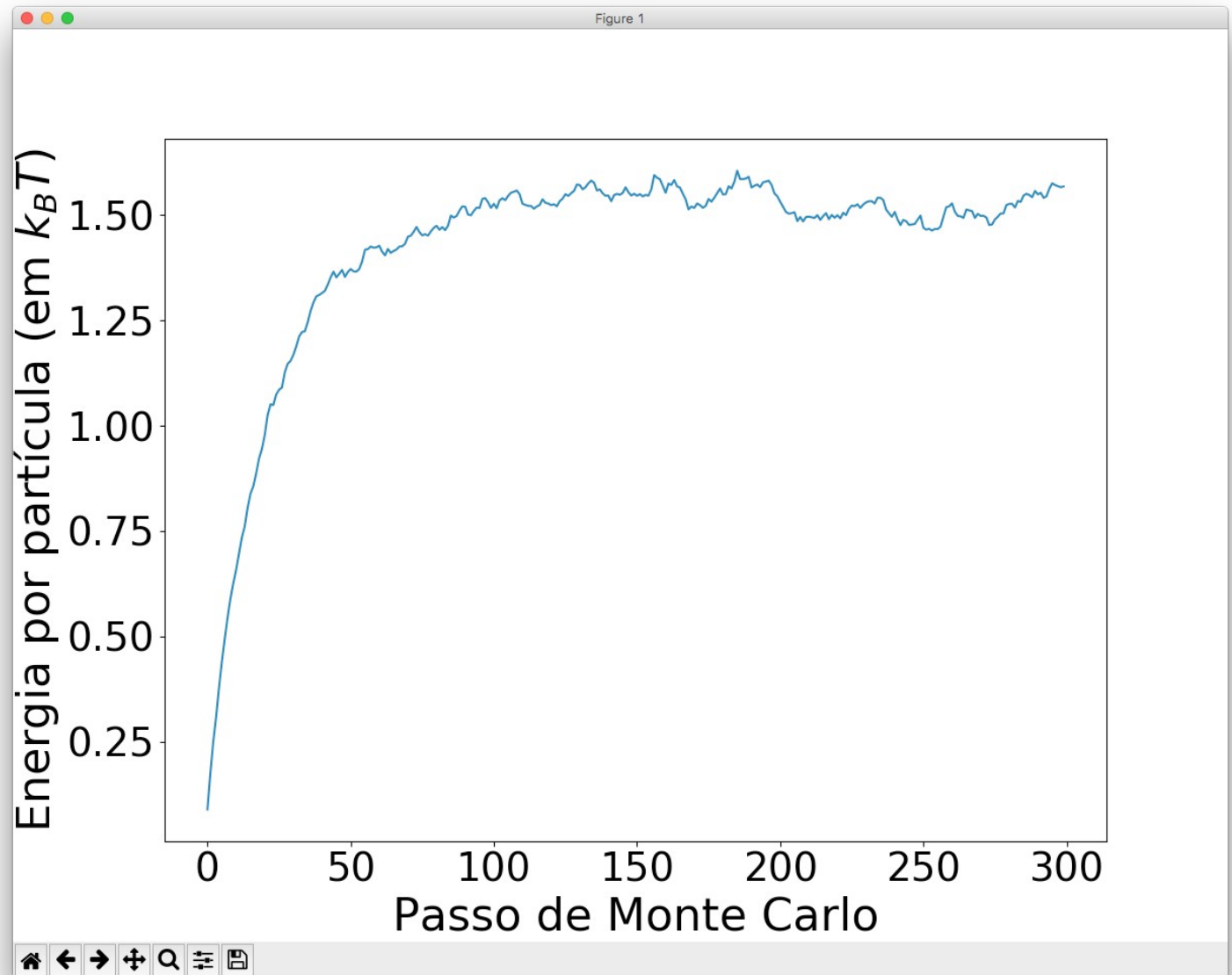
# Laço principal
E_lista = []                              # Lista para armazenar a energia
E = 0.0
for i in range(N):
    E += n[i,0]**2 + n[i,1]**2 + n[i,2]**2
E *= epsilon
for k in range(Passos):                   # Percorremos os passos de Monte Carlo
    # Escolhemos um número quântico que tentaremos alterar
    for l in range(N):                    # Cada partícula em média tem 1 chance de mudar
        i = randrange(N)                  # Partícula
        j = randrange(3)                  # Número quântico da partícula
        if random() < 0.5:                 # Com probabilidade 1/2 propomos n->n+1
            dn = 1
            dE = (2*n[i,j]+1)*epsilon      # Variação da energia
            n[i,j] += dn
            E += dE

# Registramos a energia a cada passo de Monte Carlo
E_lista.append(E/N/kBT)                   # Energia por partícula (medida em kB*T)
```

Vamos exibir a energia por partícula, em unidades de $k_B T$, para que possamos compará-la com a previsão para o gás ideal clássico.

Exemplo: o gás ideal quântico

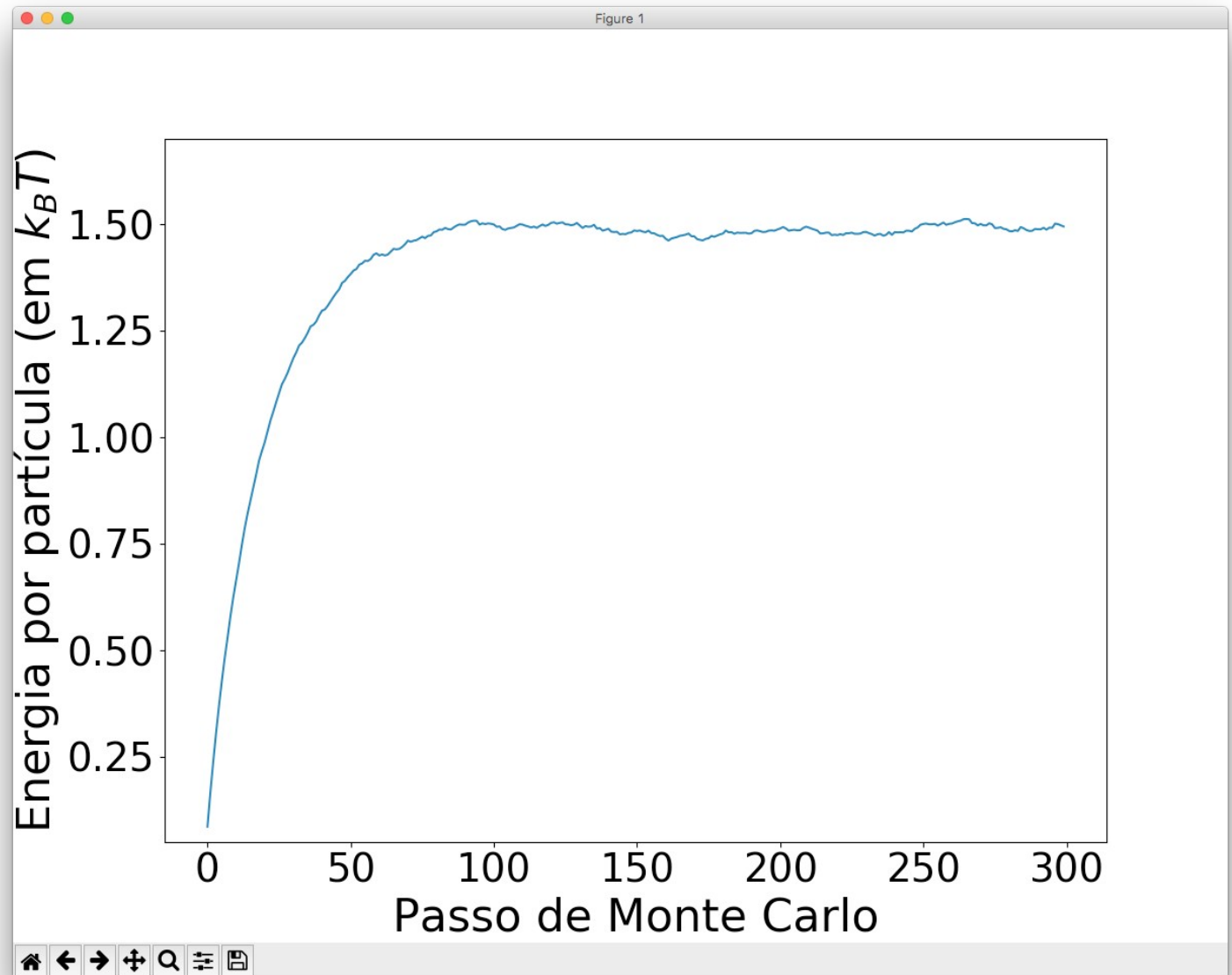
$$N = 10^3$$
$$k_B T / \epsilon = 10$$



**Quantos passos de MC são necessários para a equilibração?
Em torno de que valor flutua a energia após a equilibração?**

Exemplo: o gás ideal quântico

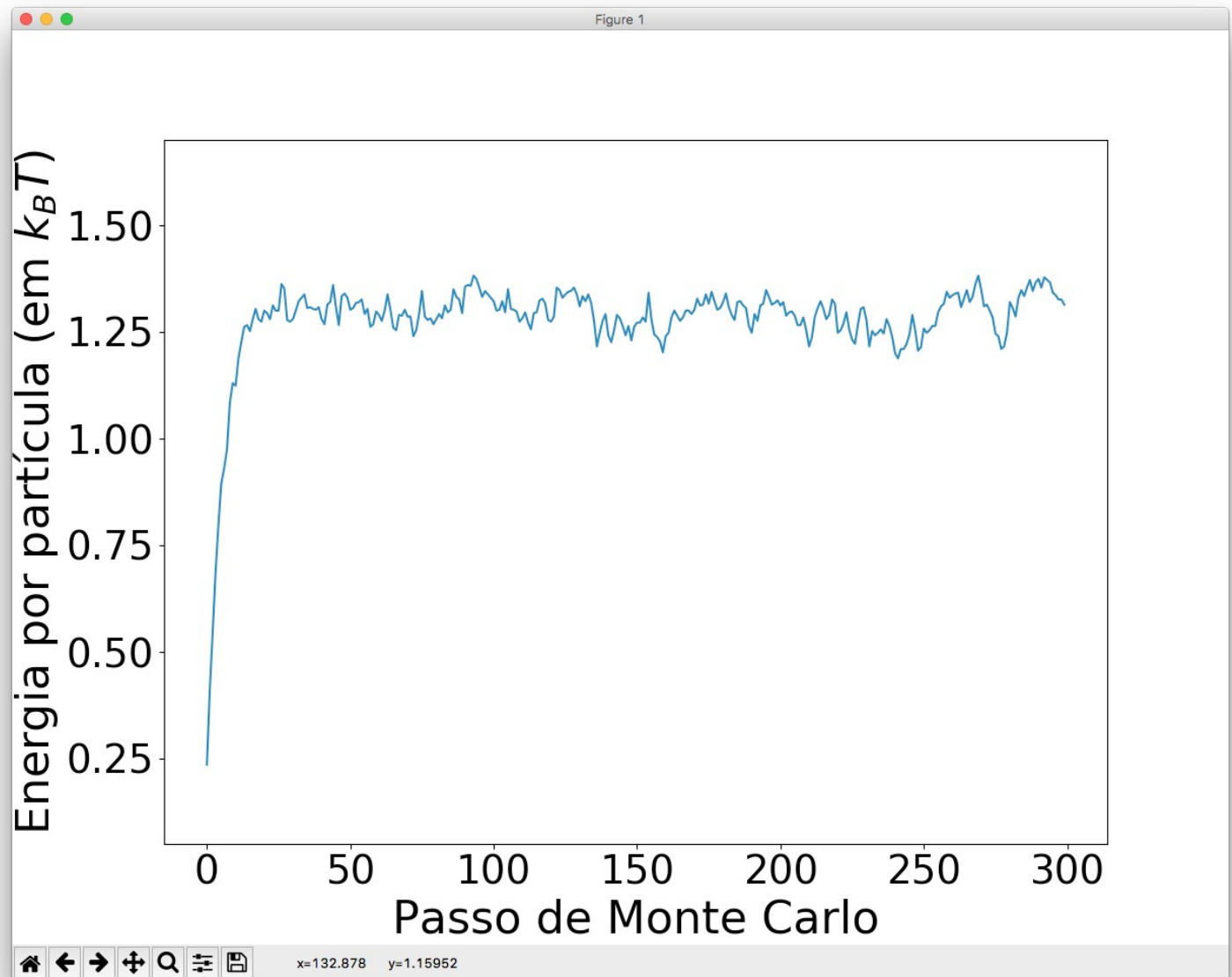
$$N = 10^4$$
$$k_B T / \epsilon = 10$$



Em torno de que valor flutua a energia após a equilibração? O que ocorreu com as flutuações da energia em torno desse valor?

Exemplo: o gás ideal quântico

$$N = 10^3$$
$$k_B T / \epsilon = 1/2$$



Em torno de que valor flutua a energia após a equilibração?
Esse valor é compatível com o de um gás ideal clássico?

Exercício no moodle

Há um único exercício, explorando o conteúdo da aula de hoje, e que pode ser feito com base nos programas dos exemplos desta aula e da aula anterior, disponíveis no moodle. O exercício deve ser entregue até o dia **3 de junho**.