

Simulações de Monte Carlo

Aula 1

- O que são simulações de Monte Carlo?
- Estimando o valor de π .
- Probabilidades, a função de partição canônica e amostragem por importância.
- O algoritmo de Metropolis; exemplos.

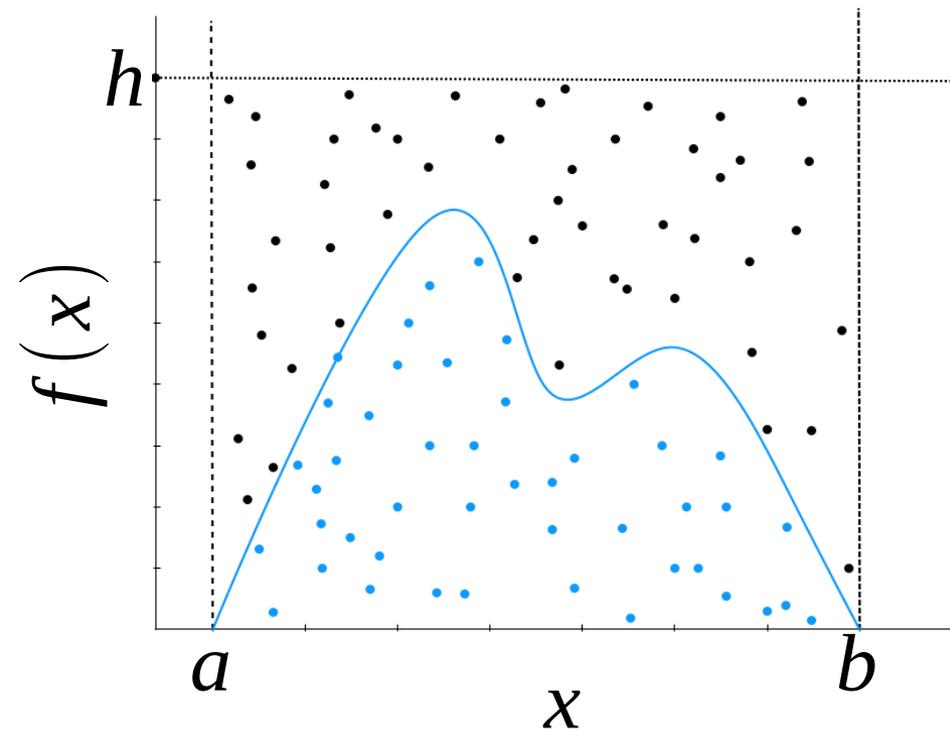
O que são simulações de Monte Carlo?

- Problemas que envolvem variáveis que não controlamos podem ser descritos como processos aleatórios. Simulações de Monte Carlo, que têm por base a produção de números (pseudo)aleatórios, podem ser aplicadas de maneira natural a essa classe de problemas.
- Exemplos: decaimento radioativo, caminhada aleatória, percolação, estática e dinâmica de sistemas termodinâmicos, processos de reação–difusão, propagação de epidemias, modelos de agentes sociais ou econômicos etc.

Outras aplicações

- Como exemplo de uma aplicação diferente, vamos utilizar números aleatórios para estimar o valor de π .
- Essa aplicação é inspirada em ideias formuladas já no século 18. Trata-se do uso de métodos de Monte Carlo para a integração de funções.

Integração por Monte Carlo



- Sorteamos aleatoriamente N pontos no retângulo demarcado. Se N_0 desses pontos situarem-se abaixo da curva da função a ser integrada, uma estimativa para a integral é

$$I_{a \rightarrow b} = \int_a^b f(x) dx$$

$$I_{a \rightarrow b} \simeq \frac{N_0}{N} h (b - a)$$

Estimando π

- Tomamos por base o fato de que a área de um círculo de raio unitário é π . Vamos então trabalhar com a função

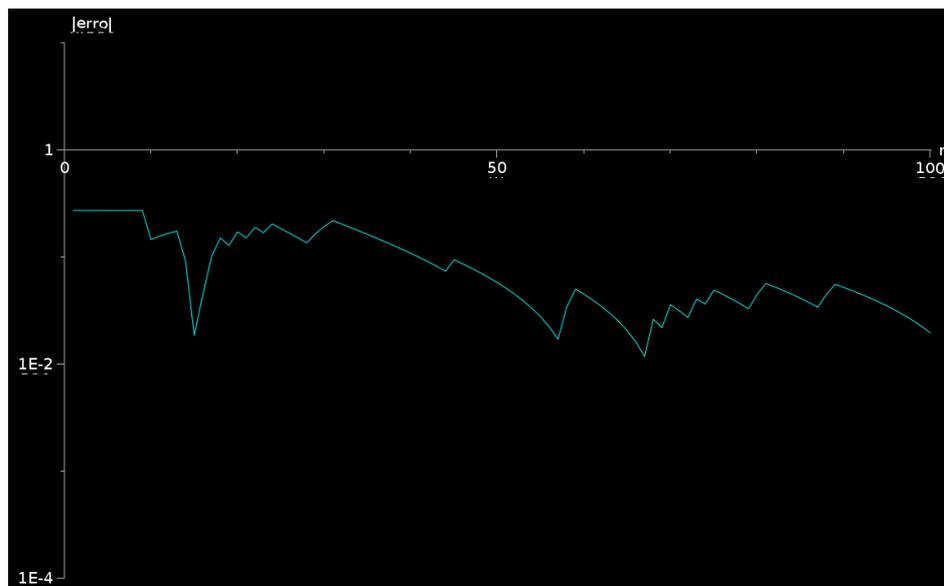
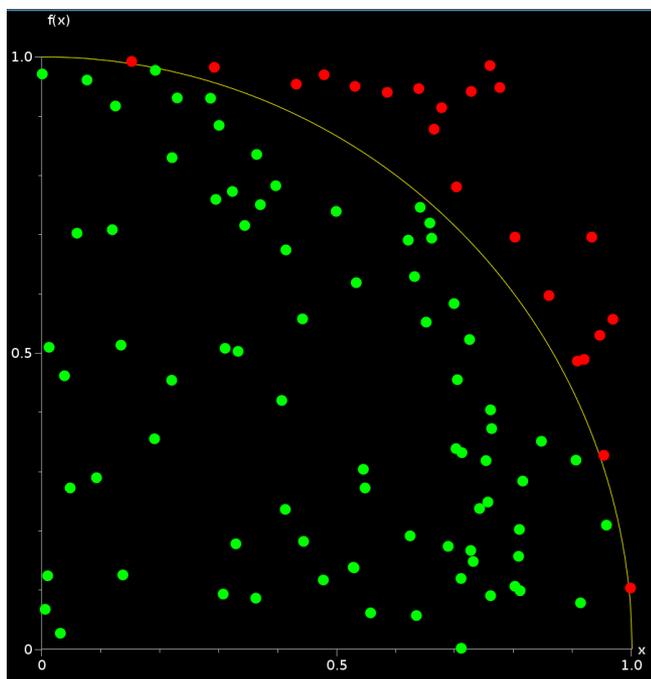
$$f(x) = \sqrt{1 - x^2},$$

de modo que

$$\int_0^1 f(x) dx = \frac{\pi}{4}.$$

Estimando π

- Baixem o programa `estima_pi.py` e o abram no aplicativo VIDLE (disponível na área de trabalho). A execução produzirá gráficos como os mostrados abaixo.



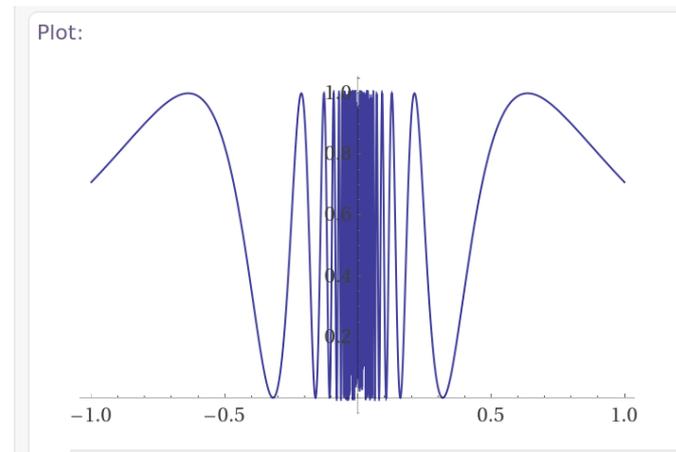
Estimando π

- Examinem o programa até compreenderem o algoritmo.
- Localizem o parâmetro que controla o número de pontos sorteados e variem seu valor de modo a obter um erro relativo estável de 1 milésimo na estimativa de π .

Integração por Monte Carlo

- Vocês devem ter obtido um valor em torno de 50.000 pontos; não se trata de um método dos mais eficientes, portanto. (Mas torna-se muito mais eficiente para integrais multidimensionais!)
- É possível melhorar o desempenho do método em 1D, mas sua utilidade maior está na integração de funções mal-comportadas, que desafiam os métodos determinísticos. Um exemplo é a função

$$f(x) = \text{sen}^2\left(\frac{1}{x}\right)$$



Integração por Monte Carlo

- Desafio 1 (para depois): estimem o valor da integral da função na página anterior, no intervalo entre -1 e 1 , com uma precisão relativa de um milésimo.
- Desafio 2 (para depois): estimem o volume de uma hiperesfera de raio unitário em 4 dimensões com uma precisão relativa de um milésimo.

Método de Monte Carlo na mecânica estatística

- No equilíbrio térmico, valores esperados são determinados pela distribuição de Boltzmann:

$$\langle Q \rangle = \frac{\sum_{\mu} Q_{\mu} e^{-\beta E_{\mu}}}{\sum_{\mu} e^{-\beta E_{\mu}}} \equiv \sum_{\mu} Q_{\mu} P_{\mu}$$

$$P_{\mu} = \frac{e^{-\beta E_{\mu}}}{\sum_{\nu} e^{-\beta E_{\nu}}} \equiv \frac{e^{-\beta E_{\mu}}}{Z}, \quad \beta = (k_B T)^{-1}$$

Método de Monte Carlo na mecânica estatística

$$\langle Q \rangle = \sum_{\mu} Q_{\mu} P_{\mu}$$

- Logo, para obter exatamente qualquer valor esperado, basta calcular a média ponderada da quantidade correspondente sobre todas as configurações μ .
- Mas quantas são essas configurações (ou **microestados**) para um dado sistema?

Sistemas de dois níveis

- Vamos exemplificar com o modelo de Ising, uma coleção de momentos magnéticos (“spins”) que podem apontar apenas para cima ou para baixo. Se há 2 spins, as configurações podem ser:

↑ ↑ ↑ ↓ ↓ ↑ ↓ ↓

- Com N spins, quantas configurações há?

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

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?

Método de Monte Carlo na mecânica estatística

- Sendo impossível considerar todas as configurações, podemos amostrá-las com uma certa distribuição de probabilidades p_{μ} e estimar os valores esperados segundo

$$\langle Q \rangle_M = \frac{\sum_{i=1}^M Q_{\mu_i} p_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}{\sum_{i=1}^M p_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}$$

- Mas qual p_{μ} escolher?

Amostragem por importância

- Não podemos escolher p_μ uniforme, porque o peso de Boltzmann diz que configurações com energia muito alta raramente são visitadas em um sistema real.
- Eis a chave: devemos escolher p_μ de modo a emular na simulação o que a natureza faz: estados são visitados com probabilidade proporcional ao seu peso de Boltzmann.

$$p_\mu = P_\mu = \frac{e^{-\beta E_\mu}}{Z} \Rightarrow Q_M = \frac{1}{M} \sum_{i=1}^M Q_{\mu_i}$$

Amostragem por importância

$$p_{\mu} = P_{\mu} = \frac{e^{-\beta E_{\mu}}}{Z} \Rightarrow Q_M = \frac{1}{M} \sum_{i=1}^M Q_{\mu_i}$$

- Em vez de escolher ao acaso configurações de acordo com p_{μ} , é mais eficiente percorrer uma cadeia de configurações que reproduza para longo tempo a distribuição de Boltzmann. (Isso pode ser formalizado; veja a bibliografia.)
- A estratégia é passar de uma configuração à seguinte na cadeia pela escolha de probabilidades de transição adequadas.

O algoritmo de Metropolis

- A implementação mais simples dessa ideia é o algoritmo de Metropolis (1953): a probabilidade de transição entre uma configuração e outra é

$$w(\mu \rightarrow \nu) = \begin{cases} 1, & \text{se } E_\nu < E_\mu \\ e^{-\beta(E_\nu - E_\mu)}, & \text{se } E_\nu \geq E_\mu \end{cases}$$

- Com essa escolha, o algoritmo é ergódico e satisfaz o balanço detalhado,

$$p_\mu w(\mu \rightarrow \nu) = p_\nu w(\nu \rightarrow \mu)$$

Exemplo: coleção de sistemas de dois níveis não interagentes

- Formulação alternativa do sistema de dois níveis: estado 0, com energia 0, e estado 1, com energia ϵ .
- A energia total da coleção é

$$E(\{n_i\}) = \sum_{i=1}^N n_i \epsilon \quad n_i \in \{0, 1\}$$

- Como não há interações entre os sistemas que compõem a coleção, a função de partição pode ser calculada exatamente.

Exemplo: coleção de sistemas de dois níveis não interagentes

- A solução exata fornece

$$Z = \sum_{\{n_i\}} e^{-\beta E(\{n_i\})} = (1 + e^{-\beta \epsilon})^N$$

$$U \equiv \langle E(\{n_i\}) \rangle = \frac{\sum_{\{n_i\}} E(\{n_i\}) e^{-\beta E(\{n_i\})}}{\sum_{\{n_i\}} e^{-\beta E(\{n_i\})}}$$

$$U = -\frac{\partial \ln Z}{\partial \beta} = \frac{N \epsilon e^{-\beta \epsilon}}{1 + e^{-\beta \epsilon}}$$

Exemplo: coleção de sistemas de dois níveis não interagentes

- O calor específico é definido por

$$c = \frac{1}{N} \frac{\partial U}{\partial T}$$

- Seu cálculo é numericamente mais conveniente se usarmos as **flutuações** da energia:

$$c = -\frac{1}{N} \frac{\partial^2 \ln Z}{\partial T \partial \beta} = \frac{\langle E^2 \rangle - \langle E \rangle^2}{N k_B T^2}$$

Exemplo: coleção de sistemas de dois níveis não interagentes

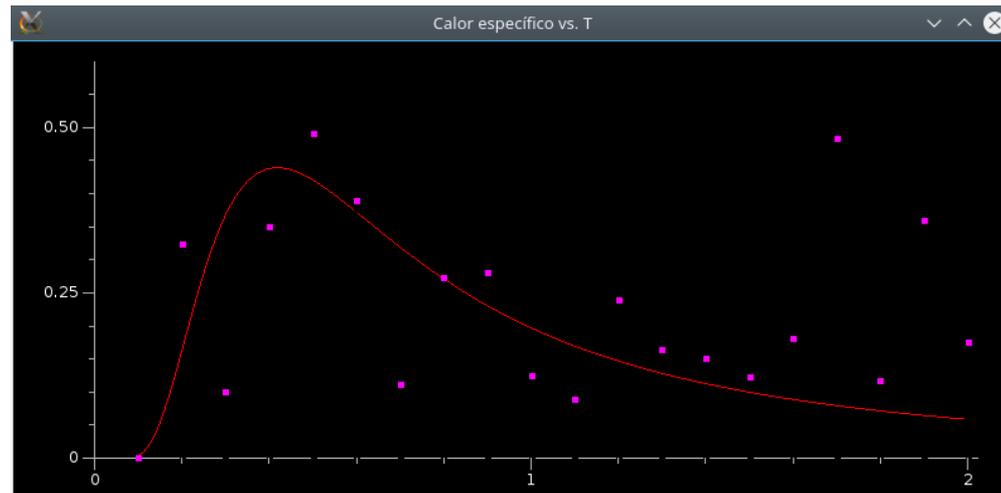
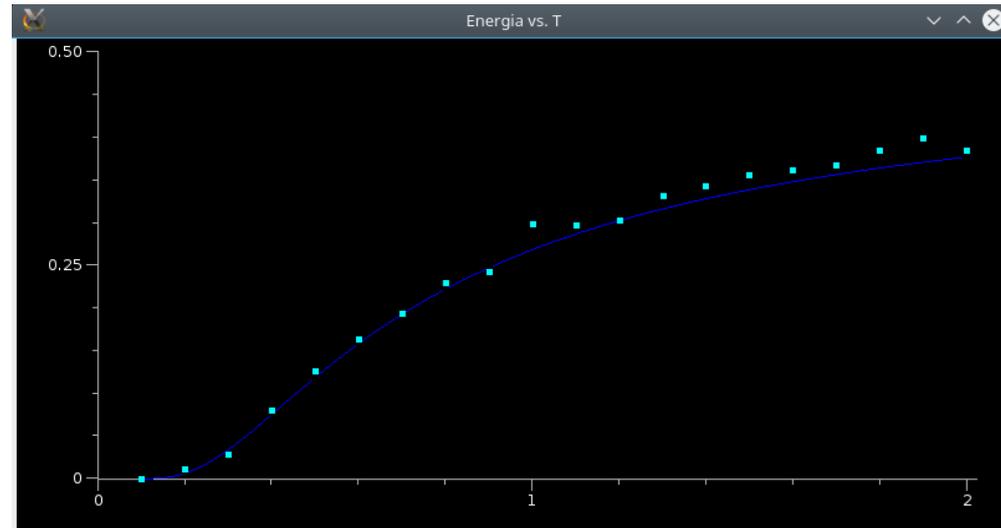
- Um passo do algoritmo de Metropolis consiste nas seguintes etapas:
 - sortear um sistema e propor movê-lo de um nível de energia para o outro;
 - calcular a variação ΔE da energia do sistema entre o novo nível e o antigo nível;
 - determinar a probabilidade de transição segundo a regra de Metropolis;
 - aceitar ou rejeitar o movimento e seguir para o próximo passo.

Exemplo: coleção de sistemas de dois níveis não interagentes

- Baixem o programa `dois_niveis.py` e o abram no aplicativo VIDLE. Ele implementa a simulação de uma coleção de $N=100$ sistemas de dois níveis.
- Examinem o programa para identificar as etapas do algoritmo, e em seguida o executem sem alterar qualquer parâmetro.

Exemplo: coleção de sistemas de dois níveis não interagentes

- Após 10 passos de Monte Carlo por sistema, o programa exibe gráficos como aqueles ao lado, comparando resultado exato (curvas) e simulação (pontos).
- O que chama a atenção neles?

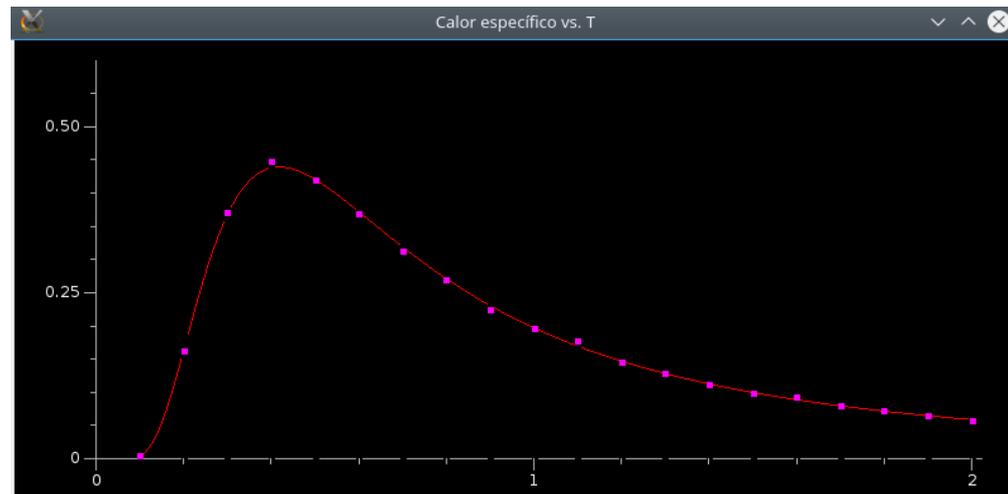
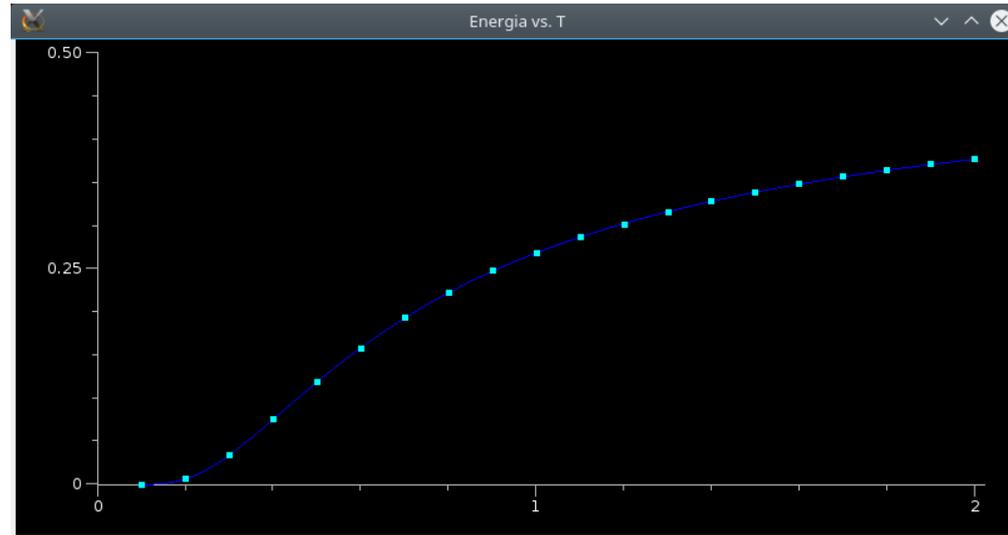


Exemplo: coleção de sistemas de dois níveis não interagentes

- Aumentem progressivamente o número de passos de Monte Carlo por sistema para 100, 500, 1000 e 5000. Acompanhem a evolução do erro médio tanto para a energia quanto para o calor específico. Procurem quantificar como esse erro varia com o número de passos de Monte Carlo.

Exemplo: coleção de sistemas de dois níveis não interagentes

- Após 5000 passos de Monte Carlo por sistema, o programa exibe gráficos como aqueles ao lado.



Exemplo: coleção de sistemas de dois níveis não interagentes

- Desafio (para depois): procurem formular uma maneira de estimar os erros (para uma dada temperatura) sem levar em conta a solução exata. Isso será essencial em problemas cuja solução exata é desconhecida!

Exemplo: fluido de Lennard–Jones

- O potencial de Lennard–Jones é

$$U(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

- Uma configuração é definida pelas posições e velocidades de todos os átomos. A energia correspondente é

$$E(\{\vec{r}_i, \vec{v}_i\}) = K(\{\vec{v}_i\}) + \sum_{\text{pares } ij} U(r_{ij})$$

- A energia cinética contribui para a energia proporcionalmente à temperatura, mas de resto pode ser ignorada.

Exemplo: fluido de Lennard–Jones

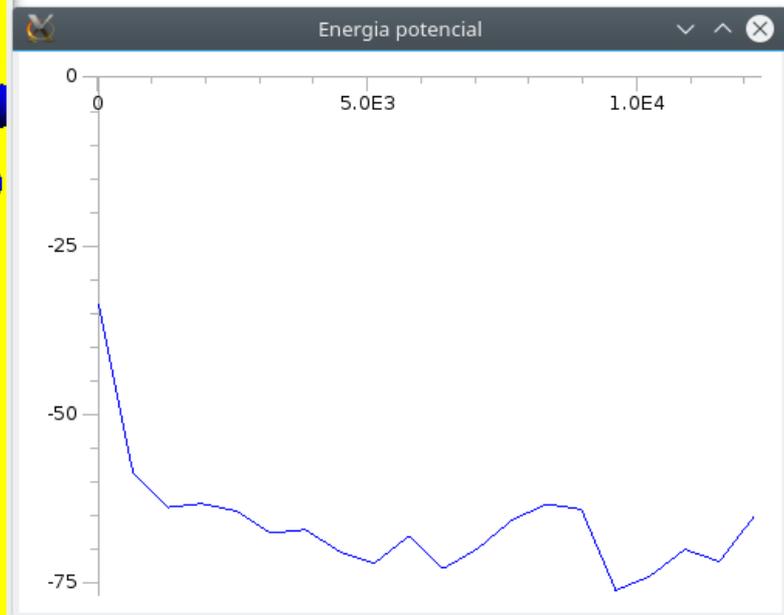
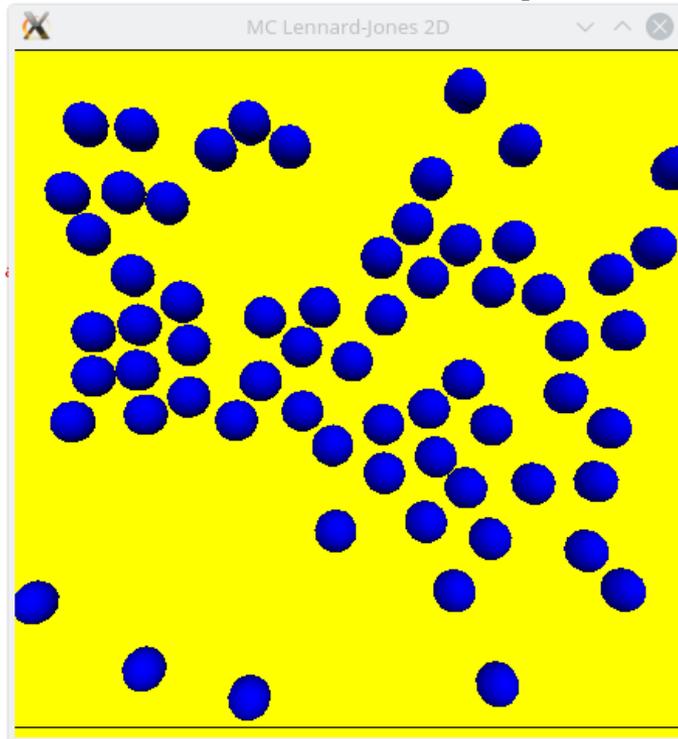
- Um passo do algoritmo de Metropolis consiste nas seguintes etapas:
 - sortear um átomo e propor movê-lo para uma nova posição nas vizinhanças (em um volume que otimiza a probabilidade de transição);
 - calcular a variação ΔU da energia potencial entre a nova posição e a antiga posição;
 - determinar a probabilidade de transição segundo a regra de Metropolis com $\Delta E = \Delta U$;
 - aceitar ou rejeitar o movimento e seguir para o próximo passo.

Exemplo: fluido de Lennard–Jones

- Baixem o programa [LJ2dMC.py](#) e o abram no aplicativo VIDLE. Trata-se de uma versão bidimensional do fluido de Lennard–Jones, com um corte em distâncias além de $2,5\sigma$.
- Examinem o programa para identificar as etapas do algoritmo, e em seguida o executem sem alterar qualquer parâmetro (64 átomos, caixa com comprimento 15σ , 200 passos de MC por átomo, temperatura reduzida de 0,6).

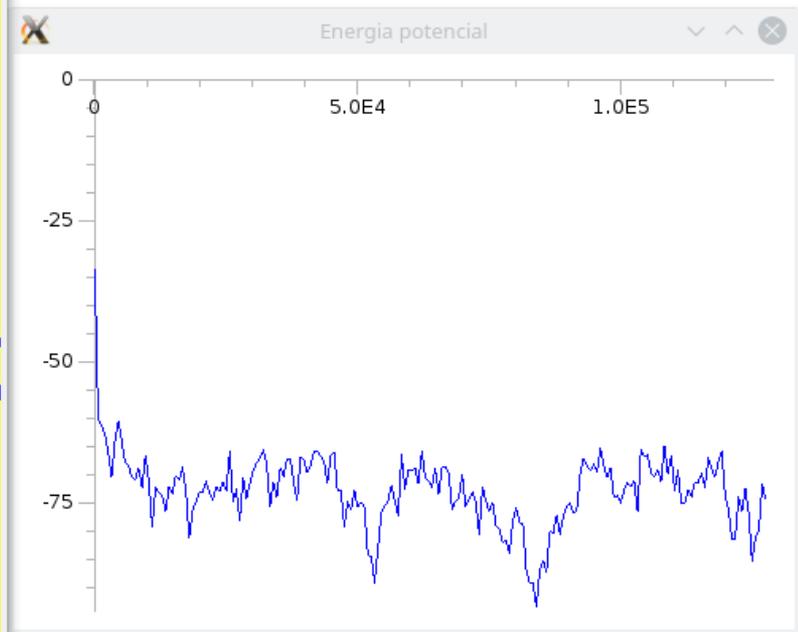
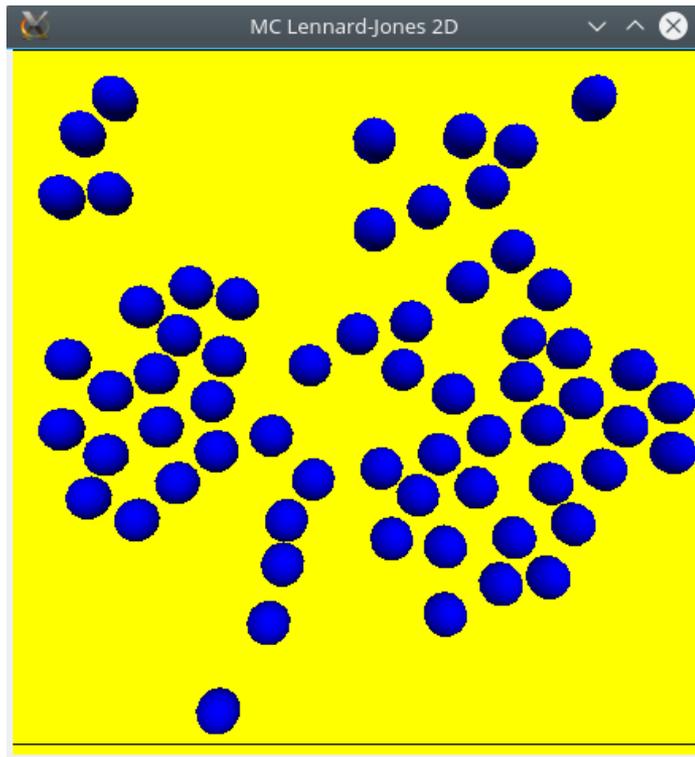
Exemplo: fluido de Lennard–Jones

- Após 200 passos por átomo, o programa exibe gráficos como os abaixo. Podemos afirmar que a energia potencial já flutua em torno do equilíbrio?



Exemplo: fluido de Lennard–Jones

- Com 2000 passos por átomo, o programa exibe gráficos como os abaixo. Podemos agora supor que a energia potencial já flutua em torno do equilíbrio?



Exemplo: fluido de Lennard–Jones

- O cálculo da pressão em d dimensões (mesmo sem paredes!) pode ser realizado com o auxílio do **teorema do virial**.

$$P = \frac{Nk_B T}{V} + \frac{1}{d \times V} \left\langle \sum_{\text{pares } ij} \vec{r}_{ij} \cdot \vec{F}_{\text{int}}(\vec{r}_{ij}) \right\rangle$$

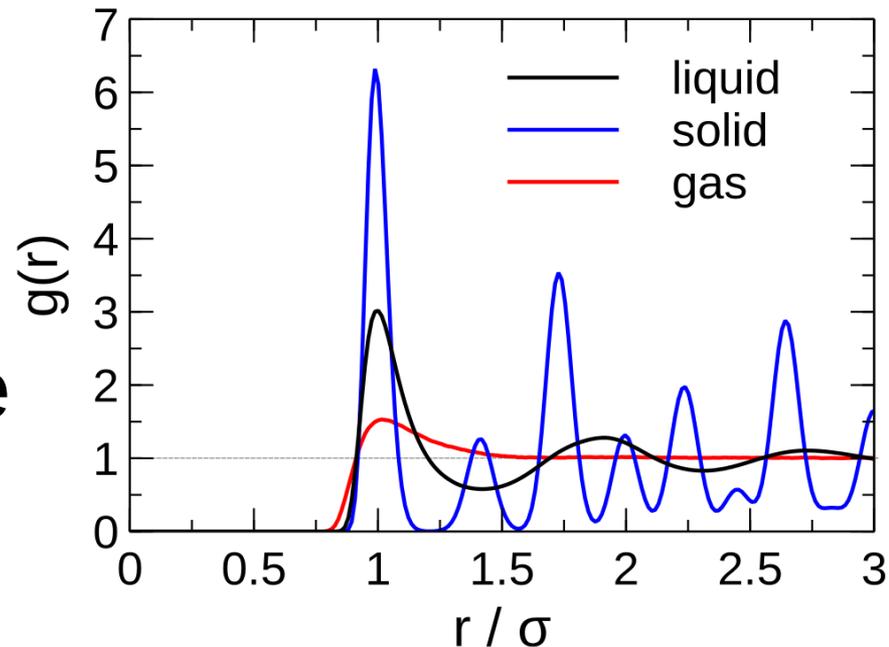
- A média tem que ser tomada no equilíbrio.

Exemplo: fluido de Lennard–Jones

- Executem a simulação (com 2000 passos por átomo) diminuindo a temperatura reduzida em 0,1 por vez até atingir 0,3.
 - Em qual temperatura aproximada ocorre a transição de gás para líquido?
 - Diminuem a temperatura reduzida para 0,2. Há algo estranho com a pressão? O que poderia ser a causa disso?

Exemplo: fluido de Lennard–Jones

- Desafio (para depois): incorporem ao código o cálculo da função distribuição radial, que permite quantificar o ordenamento e identificar a fase do sistema.



$$g(r) = \frac{1}{n^2} \left\langle \sum_{\text{pares } ij} \delta(\vec{r}_i) \delta(\vec{r}_j - \vec{r}) \right\rangle$$

Comentários

- Diferentes aspectos do comportamento termodinâmico de fluidos são melhor explorados por mudanças de ensemble. Aqui fizemos uso do ensemble *NVT*. A bibliografia discute outros casos.
- Apesar de sua grande aplicabilidade, o algoritmo de Metropolis está longe de ser o mais eficiente. Algoritmos muito mais eficientes podem ser definidos caso a caso, dependendo do sistema e do problema a ser simulado. Vejam a bibliografia.
- Para aplicações mais longas, é essencial utilizar geradores de números pseudo-aleatórios confiáveis. Vejam a bibliografia.

Bibliografia

- D. P. Landau e K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*.
- M. E. J. Newman e G. T. Barkema, *Monte Carlo Methods in Statistical Physics*.
- L. M. Sander, *Equilibrium Statistical Physics with Computer Simulation in Python*.