

**UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS**

João Victor Alcantara Pimenta

Matrizes Aleatórias e Simulação de Gases de Coulomb

São Carlos

2023

João Victor Alcantara Pimenta

Matrizes Aleatórias e Simulação de Gases de Coulomb

Trabalho de conclusão de curso apresentado ao Programa de Graduação em Física do Instituto de Física de São Carlos, da Universidade de São Paulo, para a obtenção do título de Bacharel em Física Computacional.

Orientador: Prof. Dr. Guilherme Silva

Versão original

**São Carlos
2023**

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTA TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO PARA FINS DE ESTUDO E
PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica revisada pelo Serviço de Biblioteca e Informação Prof. Bernhard Gross, com
os dados fornecidos pelo(a) autor(a)

João Victor Alcantara Pimenta
Matrizes Aleatórias e Simulação de Gases de Coulomb /
João Victor Alcantara Pimenta ; orientador Guilherme Silva.
– São Carlos, 2023.
25 p.

Trabalho de Conclusão de Curso (Graduação em Física
Computacional) – Instituto de Física de São Carlos, Universi-
dade de São Paulo, 2023.

1. Introdução. 2. Simulações e Algoritmos. 3. Implementação
e Resultados 4. Conclusão. I. SILVA, GUILHERME L. F.,
orientador. II. Matrizes Aleatórias e Simulação de Gases de
Coulomb.

RESUMO

PIMENTA, J. V. A. **Matrizes Aleatórias e Simulação de Gases de Coulomb**. 2023. 25p. Monografia (Trabalho de Conclusão de Curso) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2023.

O estudo de Matrizes Aleatórias demonstra aplicabilidade em uma gama diversa de áreas, com destaque no estudo de mecânica estatística, principalmente na simulação de gases. Estudando a densidade espectral de sistemas de matrizes Gaussianas pode-se desenvolver uma analogia que possibilita a simulação de sistemas de gases diversos, como o de Coulomb. Algumas dificuldades surgem na implementação de simulações baseadas nesta teoria, principalmente em escalabilidade do sistema e no tratamento de possíveis singularidades. Para resolver estes problemas, abordou-se na simulação na literatura, dentre outros, o Algoritmo Híbrido de Monte Carlo, de ótimo comportamento numérico. Nosso objetivo é explorar este assunto, as simulações de gases e o algoritmo citado acima além de expandir os potenciais em que foi-se bem documentado o comportamento destas simulações.

Palavras-chave: Gases de Coulomb. Matrizes Aleatórias.

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Distribuição de autovalores	8
1.2	Emsembles Gaussianos	9
1.3	Gases de Coulomb	10
1.4	Medidas de Equilíbrio	11
1.5	Potenciais notáveis	12
2	SIMULAÇÕES E ALGORITMOS	13
2.0.1	Gases de Coulomb	13
2.0.2	Log-Gases	14
2.0.3	Medidas de Equilíbrio	14
2.1	Introdução ao algoritmo	15
2.2	Algoritmo Híbrido de Monte Carlo	16
2.3	Discretização	17
3	IMPLEMENTAÇÃO E RESULTADOS	19
3.1	A implementação	20
3.2	Validação em distribuições conhecidas	21
3.3	Outros Potenciais	22
4	CONCLUSÃO	23
	REFERÊNCIAS	25

1 INTRODUÇÃO

1.1 Distribuição de autovalores

1.2 Ensembles Gaussianos

1.3 Gases de Coulomb

1.4 Medidas de Equilíbrio

1.5 Potenciais notáveis

2 SIMULAÇÕES E ALGORITMOS

Nos referenciaremos aqui aos desenvolvimento do artigo citado em (?). Vamos compilar algum desenvolvimento teórico necessário e explicitar os resultados e métodos do artigo. Vamos esquematizar as notações a serem usadas. O artigo toma um subespaço S de dimensão d em \mathbb{R}^n . O subespaço toma a métrica de Lebesgue, denotado dx . O campo externo é denominado $V : S \mapsto \mathbb{R}$ e a interação entre partículas $W : S \mapsto (-\infty, \infty]$. Para qualquer $N \geq 2$ consideramos P_N em $S^N = S \times \cdots \times S$ definida

$$P_N(dx) = \frac{e^{-\beta_N H_N(x_1, \dots, x_N)}}{Z_N} dx_1 \cdots dx_N$$

Onde $\beta_n > 0$ é uma constante e Z_N é contante de normalização. Por último,

$$H_N(x_1, \dots, x_N) = \frac{1}{N} \sum_{i=1}^N V(x_i) + \frac{1}{2N^2} \sum_{i \neq j} W(x_i - x_j)$$

Note que P_N é invariante por permutação e que H_N depende somente da medida empírica

$$\mu_N = \frac{1}{N} \sum_{i=1}^N \delta_{x_i}$$

Note que as partículas vivem em S^N de dimensão dN .

2.0.1 Gases de Coulomb

Eu não entendo Gases de Coulomb. O importante aqui é notar que tomando o subespaço S como um condutor em \mathbb{R}^n e $W = g$, onde g é o kernel de Coulomb ou função de Green em \mathbb{R}^n onde

$$g(x) = \begin{cases} \log \frac{1}{|x|} & \text{sen} = 2 \\ \frac{1}{|x|^{n-2}} & \text{sen} \geq 2 \end{cases}$$

Em termos de física, interpretamos $H_N(x_1, x_2, \dots, x_N)$ é energia eletrostática da configuração dos N elétrons em \mathbb{R}^n contidos em S nas posições x_1, x_2, \dots, x_N em um campo externo de potencial V . g expressa a repulsão de coulomb da interação entre dois corpos. P_N pode ser visto como a medida de Boltzmann-Gibbs, com β_N sendo o inverso da temperatura. P_N é o denominado gás de Coulomb.

2.0.2 Log-Gases

Log-Gases são caracterizados pela escolha de $n = d$ e W tal que

$$W(x) = \log \frac{1}{|x|} = -\frac{1}{2} \log(x_1^2 + \cdots + x_d^2)$$

note que os gases coincidem quando $n = d = 2$.

2.0.3 Medidas de Equilíbrio

É sabido que a medida empírica μ_N tende, quando $N \rightarrow \infty$ para uma medida de probabilidade não aleatória

$$\mu^* = \arg \inf \epsilon$$

sendo o minimizante único da função 'energia' ϵ convexa e semi continua definida por

$$\mu \mapsto \epsilon(\mu) = \int V d\mu + \int \int W(x - y) \mu(dx) \mu(dy)$$

Se $W = g$ for o kernel de Coulomb, $\epsilon(\mu)$ é a energia eletrostática da distribuição de cargas μ .

2.1 Introdução ao algoritmo

A ideia explorada é que P_N é medida de probabilidade invariante reversível do processo de difusão de Markov $(X_t)_{t>0}$ solução de

$$dX_t = -\alpha_N \nabla H_N(X_t) dt + \sqrt{2 \frac{\alpha_N}{\beta_N}} dB_t.$$

Sob algumas condições em β_N e V , podemos afirmar que

$$X_t \xrightarrow[t \rightarrow \infty]{Law} P_N.$$

Discretizado, tomamos o processo

$$x_{k+1} = x_k - \nabla H_N(x_k) \alpha_N \Delta t + \sqrt{2 \frac{\alpha_N}{\beta_N}} \Delta t G_k,$$

onde G_k é a família de variáveis gaussianas usuais. Uma forma de contornar o viés embutido é amenizar a dinâmica com a forma

$$x_{k+1} = x_k - \frac{\nabla H_N(x_k) \alpha_N \Delta t}{1 + |\nabla H_N(x_k)| \alpha_N \Delta t} + \sqrt{2 \frac{\alpha_N}{\beta_N}} \Delta t G_k.$$

Ainda assim, precisamos otimizar o processo. A ideia do algoritmo de Metropolis é adicionar um processo de seleção para evitar passos irrelevantes, algo do tipo:

- defina \tilde{x}_{k+1} de acordo com o kernel $K(x_k, \cdot)$ gaussiano;
- defina p_k

$$p_k = 1 \wedge \frac{K(\tilde{x}_{k+1}, x_k) e^{\beta_N H_N(\tilde{x}_{k+1})}}{K(x_k, \tilde{x}_{k+1}) e^{\beta_N H_N(x_k)}};$$

- defina

$$x_{k+1} = \begin{cases} \tilde{x}_{k+1} & \text{com probabilidade } p_k, \\ x_k & \text{com probabilidade } 1 - p_k. \end{cases}$$

2.2 Algoritmo Híbrido de Monte Carlo

O algoritmo híbrido de Monte Carlo é baseado no algoritmo anterior mas adicionando uma variável de momento para melhor explorar o espaço. Defina $E = \mathbb{R}^{dN}$ e deixe $U_N : E \rightarrow \mathbb{R}$ ser suave para que $e^{-\beta_N U_N}$ seja Lebesgue integrável. Seja ainda $(X_t, Y_t)_{t \geq 0}$ o processo de difusão em $E \times E$ solução de

$$\begin{cases} dX_t = \alpha_N \nabla U_N(Y_t) dt, \\ dY_t = \alpha_N \nabla H_N(X_t) dt - \gamma_N \alpha_N \nabla U_N(Y_t) dt + \sqrt{2 \frac{\gamma_N \alpha_N}{\beta_N}} dB_t, \end{cases}$$

onde $(B_t)_{t \geq 0}$ é o movimento browniano em E e $\gamma_N > 0$ parâmetro representando atrito.

Quando $U_N(y) = \frac{1}{2}|y|^2$ temos $Y_t = dX_t/dt$ e teremos que X_t e Y_t poderão ser interpretados como posição e velocidade do sistema de N pontos em S no tempo t . Nesse caso, U_n é energia cinética

2.3 Discretização

Descrevemos agora o algoritmo discretizado. Inicie de uma configuração (x_0, y_0) e para todo $k \geq 0$ faça

1. atualize as velocidades com

$$\tilde{y}_k = \eta y_k + \sqrt{\frac{1 - \eta^2}{\beta_N}} G_k, \quad \eta = e^{-\gamma_N \alpha_N \Delta t};$$

2. calcule os termos

$$\begin{cases} \tilde{y}_{k+\frac{1}{2}} = \tilde{y}_k - \nabla H_N(x_k) \alpha_N \frac{\Delta t}{2}, \\ \tilde{x}_{k+1} = x_k + \tilde{y}_{k+\frac{1}{2}} \alpha_N \Delta t, \\ \tilde{y}_{k+1} = \tilde{y}_{k+\frac{1}{2}} - \nabla H_N(x_{k+1}) \alpha_N \frac{\Delta t}{2}; \end{cases}$$

3. definir p_k

$$p_k = 1 \wedge \exp \left\{ \left[-\beta_N \left(H_N(\tilde{x}_{k+1}) + \frac{\tilde{y}_{k+1}^2}{2} - H_N(x_k) - \frac{\tilde{y}_k^2}{2} \right) \right] \right\};$$

4. defina

$$(x_{k+1}, y_{k+1}) = \begin{cases} (\tilde{x}_{k+1}, \tilde{y}_{k+1}) \text{ com probabilidade } p_k, \\ (x_k, -\tilde{y}_k) \text{ com probabilidade } 1 - p_k; \end{cases}$$

3 IMPLEMENTAÇÃO E RESULTADOS

3.1 A implementação

3.2 Validação em distribuições conhecidas

3.3 Outros Potenciais

4 CONCLUSÃO

REFERÊNCIAS