

# Computação Científica

## Trabalho Final

Gustavo M. Tonnera, 21/1055272

<sup>1</sup>Escola de Física e Matemática (EFMAT) – Universidade de Brasília (UnB)

gmtonnera@gmail.com

**Abstract.** *As the final work of the Scientific Computing discipline of the EFMAT extension course, the development of four physical simulations was proposed in python 3, each with a different type of force. This report presents the results obtained and the methodology used to produce the simulations.*

**Resumo.** *Como trabalho final da disciplina de Computação Científica do curso de extensão da EFMAT foi proposto o desenvolvimento de quatro simulações físicas em python 3, cada uma com um tipo de força diferente. Este relatório apresenta os resultados obtidos e a metodologia usada para produzir as simulações.*

## 1. Introdução

O trabalho final consistiu na implementação de quatro simulações de colisões 2D de partículas usando a linguagem de programação python 3. Em cada uma das simulações, os estudantes devem implementar um tipo de força específica: a primeira simulação deve simular colisões entre partículas com uma força de resistência, a segunda deve simular colisões com forças atrativas, a terceira com forças repulsivas e a quarta com uma força restauradora. Os estudantes devem produzir um gif de pelos menos 30 segundos e um gráfico representando a energia total do sistema simulado ao longo do tempo para cada simulação, explicando o comportamento dos gráficos de acordo com o comportamento das partículas.

Cada simulação deve simular 30 partículas em um espaço 2D, as quais começarão a simulação com posições, velocidades e massas aleatórias. Para simular as partículas, os estudantes devem implementar uma classe com os seguintes atributos: vetor posição, vetor velocidade, vetor aceleração, massa, raio e energia, sendo o raio igual para todas as partículas. Além disso, o tamanho do espaço 2D pode ser escolhido mediante vontade do estudante, entretanto as partículas devem colidir com as extremidades do espaço, assim como entre si.

## 2. Metodologia

Como citado na Introdução, a linguagem de programação utilizada para produzir as simulações foi python 3, o qual é uma linguagem com uma sintaxe bem simples e com um sistema de debugging simples, portanto, o desenvolvimento de código nessa linguagem é rápido, apesar de sua execução ser lenta em comparação a outras linguagens por ser interpretada.

Como foi necessário realizar cálculos entre matrizes, foi utilizada a biblioteca NumPy para facilitar as contas com matrizes. Ademais, a biblioteca Matplotlib para implementar a parte gráfica da simulação e produção do arquivo .gif.

Além da classe partícula, foi desenvolvida uma classe chamada "Simution", a qual é responsável por armazenar as partículas, bem como os atributos gerais da simulação- como o tamanho do espaço 2D, o número de partículas e o raio das partículas-, e atualizar a simulação de acordo com uma variação de tempo. Essa é a classe mãe das classes "ResistitiveForceSimulation", "AttractiveForceSimulation", "RepulsiveForceSimulation" e "RestorativeForceSimulation", as quais possuem funções específicas para aplicar forças resistivas, atrativas, repulsivas e restauradas, respectivamente.

Além disso, a classe "Simulation" possui um método chamado "make animation" responsável por produzir uma animação e, caso o usuário deseje, salvá-la e um método chamado "make energy graphic", o qual produz o gráfico da energia total do sistema ao longo do tempo e o salva, caso o usuário deseje. Para a produção dos gráficos e animações expostos nesse relatório, foi utilizada a "seed" 10 no módulo random para a criação das partículas.

### 3. Resultados Obtidos

#### 3.1. Simulação com força resistiva

Uma força resistiva é caracterizada por ter uma potência da velocidade multiplicada por menos uma constante. Na simulação produzida, a força resistiva possui o seguinte formato:  $F_r = -0,1.v^2$ , com  $K = 0,1$  sendo a constante,  $v^2$  a potência da velocidade e  $F_r$  a força resistiva. Logo, a aceleração dessa simulação é obtida por meio da equação  $a = F_r/m$ , sendo  $a$  a aceleração da partícula e  $m$  a massa da partícula. A animação produzida para essa simulação está disponível no link [https://youtu.be/va8\\_q5nXzuA](https://youtu.be/va8_q5nXzuA) e o gráfico da energia correspondente está representado pela figura 1.

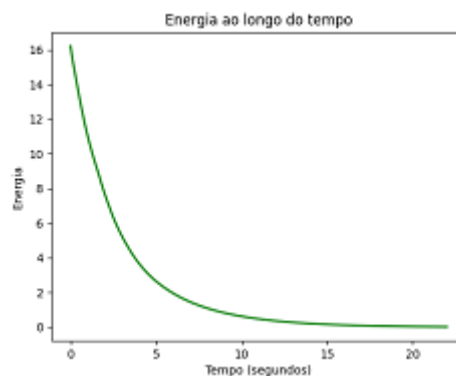


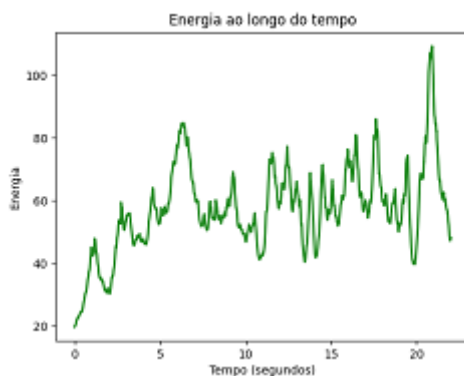
Figura 1. Gráfico da energia ao longo do tempo da simulação com força resistiva

Analisando o gráfico da figura 1, observamos que a energia total do sistema diminui rapidamente, o que se dá pelo fato de a aceleração das partículas ser no sentido oposto a velocidade das mesmas e proporcional a sua velocidade, ou seja, quanto maior a velocidade de uma partícula maior será sua desaceleração. Como a energia cinética depende da

velocidade e velocidade das partículas é cada vez menor com o passar do tempo, a energia cinética de cada partícula e a energia total do sistema diminuem ao longo do tempo. Por isso, o gráfico apresenta o comportamento observado.

### 3.2. Simulação com força atrativa

Uma força atrativa entre duas partículas é caracterizada pela fórmula  $F_a = -\frac{K}{d^2} \cdot \hat{r}$ , com  $F_a$  sendo a força atrativa entre as partículas,  $K$  uma constante real,  $d$  a distância entre as duas partículas e  $\hat{r}$  o vetor que indica a direção da força, o qual é calculado a partir da seguinte equação:  $\hat{r} = \frac{p_1 - p_2}{d}$ , com a posição da partícula sofrendo a força sendo  $p_1$ , a posição da partícula exercendo a força sendo  $p_2$  e  $d$  a distância entre essas partículas. Assim, a aceleração de uma partícula é calculada pela divisão entre a soma de todas as forças atrativas que ela sofre pela sua massa. A animação produzida para essa simulação pode ser acessada pelo link <https://youtu.be/WX2xgCREz0Y> e o gráfico da energia correspondente está representado na figura 2. Nesse caso, foi utilizado a contante  $K = 1$ .



**Figura 2. Gráfico da energia ao longo do tempo da simulação com força atrativa**

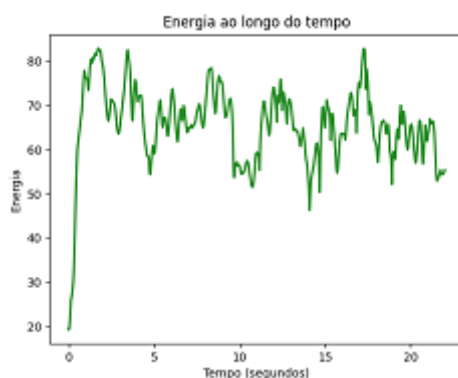
Analisando o gráfico da figura 2, observa-se que no primeiro momento a energia do sistema tem um aumento significativo e depois apresenta um comportamento oscilatório, diminuindo e aumentando com o tempo. Isso acontece porque a força atrativa é inversamente proporcional ao quadrado da distância entre partículas, ou seja, quanto mais distantes as partículas estão umas das outras menor a força de atração entre elas.

No momento inicial da simulação, as partículas estão espalhadas aleatoriamente pelo espaço 2D e, com o passar do tempo, elas vão ficando mais próximas uma das outras, devido às forças atrativas, o que explica o aumento significativo da energia do sistema no início. Depois disso, as partículas colidem e, por isso, se afastam uma das outras, mas como algumas partículas bem próximas, o nível de energia não volta ao inicial, mas sofre uma pequena queda. E esse comportamento acontece repetidas vezes: as partículas colidem, se afastam e, depois, voltam a se aproximar, o que pode ser observado no gráfico pelos picos de energia, os quais são os momentos em que as partículas estão mais próximas.

Devido a essa interação, as partículas tendem a ficar juntas em um ponto do espaço. Quando isso acontecer, a energia do sistema será constante, já que a distância entre as partículas não sofrerá mudanças nesse instante da simulação.

### 3.3. Simulação com força repulsiva

Similarmente à força atrativa, uma força repulsiva entre duas partículas é caracterizada pela fórmula  $F_{re} = \frac{K}{d^2} \hat{r}$ , sendo  $K$  uma constante real,  $d$  a distância entre as duas partículas e  $\hat{r}$  o vetor que indica a direção da força que é calculado da mesma maneira que na força atrativa. Interessante notar que a única diferença da força atrativa para a repulsiva é o sinal do vetor, ou seja, as forças atuam em sentidos diferentes. Dessa maneira, a aceleração de uma partícula é calculada pela divisão entre a soma das forças repulsivas que ela sofre e a massa dessa partícula. A animação produzida para essa simulação pode ser visualizada pelo link <https://youtu.be/A2Q0yxdpE5Y> e o gráfico da energia gerado é representado pela figura 3. Nesse caso, foi usado a constante  $K = 5$ .



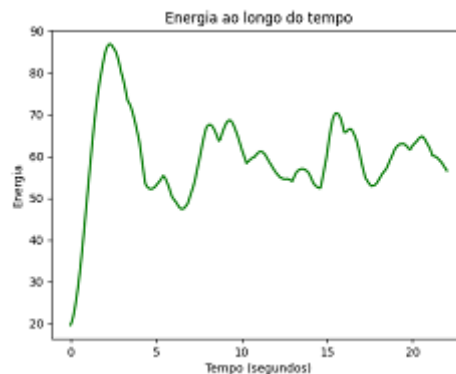
**Figura 3. Gráfico da energia ao longo do tempo da simulação com força repulsiva**

Analisando o gráfico, notamos um comportamento similar ao da simulação anterior: no início a energia aumenta generalizadamente e, depois, oscila, posto que a força a repulsiva também é inversamente proporcional ao quadrado da distância entre as partículas. Logo, no início, quando as partículas estão dispostas aleatoriamente no espaço, as partículas aceleram para se distanciar uma das outras, conseqüente, há uma aumento na energia do sistema e, com o passar do tempo, essa energia oscila devido às colisões entre as próprias partículas e às colisões com as extremidades do espaço vetorial, as quais fazem com que algumas partículas se aproximem e acelerem novamente.

Esse comportamento indica que as partículas tendem a se organizar de maneira a ficarem distantes entre si. Em algum momento, as partículas acham um ponto de equilíbrio, de modo que a distância entre elas permaneça constante e, como resultado, a energia do sistema permanecerá constante.

### 3.4. Simulação com força restauradora

Uma força restauradora é caracterizada pela fórmula  $F_{res} = -x.K$ , sendo  $F_{res}$  a força restauradora,  $x$  a distância entre o ponto de equilíbrio da partícula e sua posição e  $K$  uma constante real. Assim, a aceleração de uma partícula pode ser calculada por  $a = \frac{F_{res}}{m}$ , sendo  $m$  a massa da partícula. A animação correspondente a essa simulação pode ser acessada pelo link <https://youtu.be/dmR-xjWo0PQ> e o gráfico da energia do sistema está representado na figura 4. A simulação foi configurada de maneira que todas as partículas possuíssem o mesmo ponto de equilíbrio: ponto com coordenadas  $x = 10$  e  $y = 10$  e constante  $K = 0.1$ .



**Figura 4. Gráfico da energia ao longo do tempo da simulação com força restauradora**

Analisando o gráfico da figura 4, observamos que ele possui um comportamento similar aos gráficos anteriores: aumento da energia no início e oscilação depois desse aumento. Entretanto, esse comportamento ocorre não por causa das distâncias entre as partículas, mas sim porque a força restauradora é diretamente proporcional à distância entre a partícula e seu ponto de equilíbrio. Nesse caso, como a maioria das partículas começam em posições distantes do ponto de equilíbrio, as partículas são aceleradas de maneira a se aproximarem desse ponto. Quanto mais próximas as partículas estão do ponto de equilíbrio, maiores serão suas velocidades, logo, suas energias cinéticas serão maiores, e por isso a energia do sistema aumenta. Ao passo que as partículas se aproximam do ponto de equilíbrio, elas começam a colidir entre si e, conseqüente, algumas partículas se afastam do ponto de equilíbrio. Essas partículas são progressivamente desalinhadas pela força restauradora, assim, a energia do sistema diminui. Porém, quando elas chegam no valor máximo de  $x$ , no qual a aceleração da partícula é máxima, elas voltam a acelerar no sentido do ponto de equilíbrio e assim sucessivamente.

#### 4. Conclusão

Dessa maneira, utilizando a linguagem de programação python 3 e suas bibliotecas NumPy e Matplotlib é possível simular sistemas físicos bem simples, os quais simulam colisões entre partículas em um espaço 2D que sofrem a ação de quatro tipos de força: restauradora, atrativa, repulsiva e restauradora. Observamos que, com excessão da força resistiva- na qual a energia do sistema torna-se cada vez menor ao longo do tempo-, a energia dos sistemas sofrem um significativo aumento devido a aceleração inicial das partículas e, após esse momento, a energia do sistema oscila, devido às colisões entre as partículas e às colisões das partículas com as extremidades do espaço 2D.

#### 5. Referências

- [1] Aulas da EFMAT;
- [2] Anotações da aulas da EFMAT;
- [3] HALLIDAY, David; RESNICK, Robert; WALKER, Jearl. Fundamentos de física - MECÂNICA. 8. ed. Rio de Janeiro, RJ: LTC, c2009 vol 1.

[4] THOMAS, George; WEIR, Maurice; HASS, Joel. Cálculo. Volume 1. 12. ed. São Paulo: Pearson, 2014.