

MAC110 Introdução à Computação

Primeiro Semestre de 2024

Segundo Exercício-Programa (EP2)

29 de abril de 2024

1 Introdução

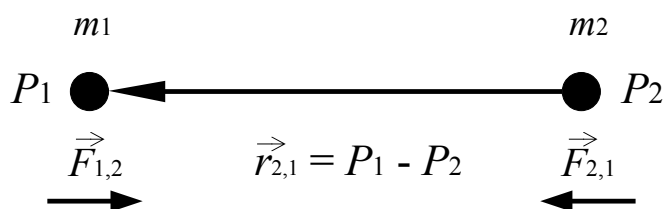
O objetivo deste EP é exercitar os conceitos, vistos em aula, de “funções” e “listas” em linguagem Python. O programa consiste em simular a trajetória de naves sujeitas às forças de atração gravitacional de corpos celestes (ex: estrelas, planetas). Por questões de simplicidade, as posições dos corpos celestes serão consideradas fixas e analisaremos o movimento das naves somente no caso bidimensional.

2 Leis de Newton do movimento

Uma das mais antigas leis físicas conhecida é a *lei da gravitação universal* que Isaac Newton publicou no seu *Principia Mathematica* [1]. Essa lei exprime as forças de interação entre dois corpos, um corpo de massa m_1 em um ponto $P_1 = (x_1, y_1)$ e um corpo de massa m_2 em um ponto $P_2 = (x_2, y_2)$, cujo deslocamento relativo é $\vec{r}_{2,1} = P_2 - P_1 = P_1 - P_2$. Ela diz que:

$$\vec{F}_{2,1} = \frac{G \times m_1 \times m_2}{\|\vec{r}_{2,1}\|^2} \hat{r}_{2,1} = -\vec{F}_{1,2} \quad (1)$$

onde $\|\vec{r}_{2,1}\|$ é a distância entre os centros de massa dos corpos e $\hat{r}_{2,1} = \vec{r}_{2,1}/\|\vec{r}_{2,1}\|$ é o vetor unitário da direção que vai do corpo 2 ao corpo 1, conforme a ilustração abaixo:



Em palavras, (1) diz que a *magnitude* da força gravitacional é diretamente proporcional ao produto das massas dos corpos e inversamente proporcional ao quadrado da distância que os separa. A força

está dirigida ao longo da reta que passa pelos centros de massa dos dois corpos e é *atractiva*, isto é, a força $\vec{F}_{2,1}$ exercida pelo corpo 1 sobre o corpo 2 está dirigida para o corpo 1, em sentido $\vec{r}_{2,1}$.

A constante proporcional G que aparece em (1) é uma constante universal, ou seja, é a mesma para quaisquer corpos. Essa constante é chamada **constante gravitacional**. Seu valor é

$$G \approx 8.65 \times 10^{-13} \text{km}^3 \text{kg}^{-1} \text{h}^{-2} \quad (2)$$

Devido à segunda lei de Newton da mecânica clássica [2], sabemos que

$$\Sigma \vec{F} = m \vec{a} \quad (3)$$

onde $\Sigma \vec{F}$ é a força resultante, m é a massa do corpo e \vec{a} é o vetor aceleração do corpo. A aceleração possui a mesma direção e o mesmo sentido da força.

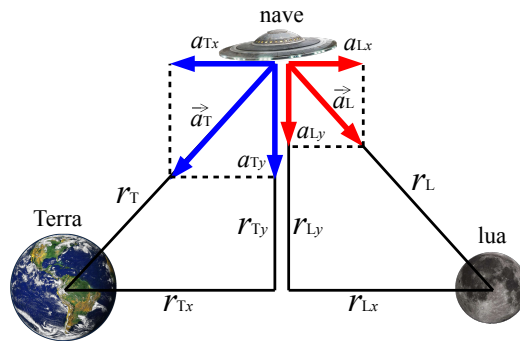
A título de exemplo, vamos ilustrar um caso simples em que uma nave está sujeita à atração gravitacional de dois astros, digamos a Terra e a Lua. Considerando o corpo no ponto P_1 de (1) como sendo a Terra, e o corpo no ponto P_2 de (1) bem como o corpo de (3) como sendo a nave, e combinando (1) e (3) obtemos que

$$\vec{a}_T = \frac{G \times m_T}{r_T^2} \hat{r}_T \quad (4)$$

onde \vec{a}_T é o vetor aceleração da força gravitacional exercida pela Terra sobre a nave, m_T é a massa da Terra, r_T é a distância da nave ao centro da Terra e \hat{r}_T é o vetor unitário da direção que vai da nave à Terra. De maneira similar, temos que

$$\vec{a}_L = \frac{G \times m_L}{r_L^2} \hat{r}_L \quad (5)$$

onde \vec{a}_L é o vetor aceleração da força gravitacional exercida pela Lua sobre a nave, m_L é a massa da Lua, r_L é a distância da nave ao centro da Lua e \hat{r}_L é o vetor unitário da direção que vai da nave à Lua. A ilustração abaixo apresenta a disposição dos astros e da nave destacando as distâncias e as acelerações.



Das equações (4) e (5) concluímos que a aceleração da força resultante é

$$\begin{aligned} \vec{a} &= \vec{a}_T + \vec{a}_L \\ &= \frac{G \times m_T}{r_T^2} \hat{r}_T + \frac{G \times m_L}{r_L^2} \hat{r}_L. \end{aligned} \quad (6)$$

No caso geral, temos um número arbitrário de astros fornecidos em um conjunto A . A aceleração resultante da nave, nesse caso, é calculada como a soma das contribuições exercidas por cada astro c pertencente ao conjunto A :

$$\begin{aligned}\vec{a} &= \sum_{\forall c \in A} \vec{a}_c \\ &= \sum_{\forall c \in A} \left\{ \frac{G \times m_c}{r_c^2} \hat{r}_c \right\}.\end{aligned}\quad (7)$$

Suponha agora que desejemos determinar para cada instante t o vetor posição $\vec{r}(t) = (x(t), y(t))$ de uma nave que está sob a atração gravitacional desse conjunto de corpos celestes. Sabemos que o vetor aceleração instantânea é

$$\vec{a} = \frac{d\vec{v}}{dt} = \frac{d^2\vec{r}}{dt^2}$$

e que seus componentes (a_x, a_y) são tais que

$$a_x = \frac{dv_x}{dt} = \frac{d^2x}{dt^2} \quad \text{e} \quad a_y = \frac{dv_y}{dt} = \frac{d^2y}{dt^2}.\quad (8)$$

As equações de Newton (8) reduzem o problema de determinar o vetor posição $\vec{r}(t) = (x(t), y(t))$ para subproblemas unidimensionais. Na próxima seção mostramos como as equações de Newton podem ser usadas para determinarmos computacionalmente uma aproximação de $x(t)$ e, similarmente, de $y(t)$, $v_x(t)$ e $v_y(t)$.

3 Método de Euler para as equações de Newton

Consideremos a equação da segunda lei de Newton na sua versão unidimensional $F_x = m a_x$ para um objeto de massa m . No nosso caso a força F_x é a gravitacional e pela lei da gravitação universal é função da posição x , de modo que a equação diferencial para a velocidade se escreve:

$$a_x = \frac{dv_x}{dt}.$$

Para obter a posição, recorreremos à relação entre posição e velocidade,

$$v_x = \frac{dx}{dt}.$$

Se conhecermos a posição e a velocidade do objeto num certo instante, essas equações nos permitem determinar $v_x(t)$ e $x(t)$ em qualquer instante t , dado que conhecemos sua aceleração. No entanto, no caso geral, não podemos resolver essas equações analiticamente. Isso significa que temos que recorrer ao computador através de um método numérico. Esse método deverá ser capaz de estimar a velocidade e posição do objeto num certo instante, dadas, por exemplo, a posição e velocidade x_0 e v_0 em um instante inicial t_0 .

O método numérico mais simples para resolver equações diferenciais consiste em substituir as derivadas por razões entre variações da função e da respectiva variável. No caso presente temos que

$$\frac{dv_x}{dt} \rightarrow \frac{\Delta v_x}{\Delta t} \quad \Rightarrow \quad \Delta v_x \approx a_x \Delta t$$

A ideia do Δt é substituir a variável contínua t por uma variável *discreta*; ou seja, fazemos o t avançar em intervalos fixos. Assim, dado um instante t , o instante seguinte é $t + \Delta t$. Para intervalos pequenos, podemos considerar a aceleração como constante durante Δt , de modo a obter uma aproximação. Logo, dados a posição x , a velocidade v e a aceleração a num instante t , podemos calcular seus valores x' , v' no instante seguinte $t + \Delta t$ pelas fórmulas:

$$x' = x + v\Delta t + a\Delta t^2/2, \quad (9)$$

$$v' = v + a\Delta t. \quad (10)$$

Para fechar as equações, lembrem que a aceleração no instante t é dada pela equação 7.

O método descrito acima (Equações (9) e (10)) é uma pequena variação do *método de Euler*¹ para o cálculo das posições e velocidades de um corpo sujeito a uma força.

4 Atividade

Neste exercício-programa utilizaremos quilômetros (km) para medir distância, quilogramas (kg) para medir massa e horas (h) para medir tempo.

Vocês devem fazer um programa em linguagem Python para simular a trajetória de naves. O programa deve ler duas listas, uma contendo as informações das naves e outra com os dados dos corpos celestes. Cada nave será descrita por cinco valores reais, as posições x e y , sua velocidade v_x e v_y , e o seu raio r (as naves são OVNI's de formato circular). Cada corpo celeste será descrito por quatro valores reais, suas posições x e y , sua massa m , e seu raio R ². As naves e astros são dispostos sequencialmente em suas respectivas listas, conforme ilustrado abaixo:

$$\text{Naves} = [\underbrace{[[x_1, y_1], [v_{x1}, v_{y1}], r_1]}_{\text{Nave \#1}}, \underbrace{[[x_2, y_2], [v_{x2}, v_{y2}], r_2]}_{\text{Nave \#2}}, \dots, \underbrace{[[x_N, y_N], [v_{xN}, v_{yN}], r_N]}_{\text{Nave \#N}}] \quad (11)$$

$$\text{Astros} = [\underbrace{[[x_1, y_1], m_1, R_1]}_{\text{Astro \#1}}, \underbrace{[[x_2, y_2], m_2, R_2]}_{\text{Astro \#2}}, \dots, \underbrace{[[x_M, y_M], m_M, R_M]}_{\text{Astro \#M}}] \quad (12)$$

Observe que as coordenadas (x e y) e os vetores velocidade (v_x e v_y) são agrupados em listas aninhadas na representação acima.

Vocês devem implementar as seguintes funções:

```
def distancia(P1, P2):
    Calcula a distância entre dois pontos fornecidos P1 = [x1, y1] e P2 = [x2, y2].
```

¹O método de Euler é um dos tópicos de disciplinas de Cálculo Numérico, onde também são considerados métodos mais precisos como o de Runge-Kutta.

²Em geral, as dimensões das naves são desprezíveis em relação aos astros ($r \ll R$), mas aqui vamos considerar naves grandes, como muito usado em jogos de computadores, tal como no jogo Star Control II: https://en.wikipedia.org/wiki/Star_Control_II.

```
def aceleracaoGravitacional(Astro, P):
```

Calcula e devolve a aceleração da atração gravitacional $[a_x, a_y]$ exercida sobre a nave no ponto $P = [x, y]$ pelo astro *Astro*.

```
def aceleracaoResultante(Astros, P):
```

Calcula e devolve a aceleração resultante $[a_x, a_y]$ da soma das contribuições exercidas por cada um dos astros presentes na lista *Astros* sobre a nave no ponto $P = [x, y]$.

```
def deteccaoColisao(Nave, Astros):
```

Verifica se ocorre sobreposição entre a nave fornecida com algum dos astros presentes na lista *Astros*. Devolve **True** se a nave colidiu com algum astro, caso contrário a função retorna **False**.

```
def atualizaNave(Nave, Astros, delta_t):
```

Atualiza a posição e velocidade da nave em *Nave* sujeita às forças de atração gravitacional dos astros presentes na lista *Astros*, após um intervalo de tempo *delta_t*.

```
def distanciaAstroMaisProximo(Nave, Astros):
```

Calcula a distância da nave *Nave* em relação ao astro mais próximo dentre os astros presentes na lista *Astros*. A distância deve ser medida em relação a superfície do astro e da nave. Em caso de nave colidida, a distância deve ser zero.

```
def simulacao(Naves, Astros, niter, delta_t):
```

Recebe uma lista de naves em *Naves*, uma lista de astros em *Astros*, o número de iterações *niter* da simulação e o intervalo de tempo *delta_t*. A função deve calcular as trajetórias das naves em *Naves* sob o efeito da força gravitacional exercida pelos astros em *Astros*.

A função deve devolver:

- Uma lista $[T_1, T_2, \dots, T_N]$ das trajetórias de cada nave, sendo N o número total de naves. Cada trajetória T_i é por sua vez uma lista das posições da i -ésima nave ao longo da simulação.
- Uma lista $[D_1, D_2, \dots, D_N]$, em que N é o número total de naves e D_i representa uma lista com as distâncias da i -ésima nave em relação ao seu astro mais próximo a cada iteração.

O instante inicial da simulação é zero e a cada passo da simulação o tempo de simulação é acrescido de *delta_t*. As posições das naves serão atualizadas em cada passo da simulação. Uma nave será desativada assim que colidir com algum astro. Neste caso a nave conserva a posição em que ocorreu a colisão ao longo das próximas iterações. Não serão consideradas colisões entre naves. A simulação termina quando o número de iterações executadas atingir *niter*.

Esta função DEVE usar as funções 'atualizaNave', 'deteccaoColisao' e 'distanciaAstroMaisProximo'.

A função principal visando testar as demais funções é fornecida pronta e **NÃO** deve ser modificada. Ela lê o número máximo de iterações, o valor de Δt e as listas de naves (contendo as velocidades iniciais) e de astros. Depois é chamada a função de *simulacao*. A cada iteração da simulação, devem

ser atualizadas as posições e velocidades das naves que ainda estiverem ativas, sendo verificado a cada atualização a ocorrência de colisões entre naves e astros. Naves que sofreram colisão se tornam inativas e devem conservar a posição em que ocorreu a colisão ao longo das próximas iterações. A simulação é interrompida quando for atingido o número máximo de iterações.

Ao final é impresso um relatório ao longo das iterações para cada nave contendo, sua posição (x, y) (posição quando aconteceu a colisão no caso de naves inativas) e sua distância em relação ao astro mais próximo, conforme mostrado no exemplo a seguir. Os valores em vermelho foram digitados pelo usuário.

```
Número máximo de iterações: 12
Delta t: 5
Número de naves: 3
*** Nave 1 ***
Digite a posição (x,y): 150000 150000
Digite a velocidade inicial (vx,vy): -1000 1000
Digite o raio: 10000
*** Nave 2 ***
Digite a posição (x,y): -140000 -160000
Digite a velocidade inicial (vx,vy): -1000 1000
Digite o raio: 20000
*** Nave 3 ***
Digite a posição (x,y): -50000 -50000
Digite a velocidade inicial (vx,vy): 500 500
Digite o raio: 5000
Número de astros: 1
*** Astro 1 ***
Digite a posição (x,y): 0 0
Digite a massa: 57.35e+22
Digite o raio: 30000
***** iteração 1 *****
*** Nave 1 ***
Posição: (144902.561,154902.561)
Distância ao astro mais próximo: 172112.130
*** Nave 2 ***
Posição: (-144909.660,-154896.754)
Distância ao astro mais próximo: 162112.739
*** Nave 3 ***
Posição: (-46623.051,-46623.051)
Distância ao astro mais próximo: 30934.950
***** iteração 2 *****
*** Nave 1 ***
Posição: (139613.529,159607.032)
Distância ao astro mais próximo: 172052.687
*** Nave 2 ***
Posição: (-149634.822,-149589.616)
Distância ao astro mais próximo: 161583.632
*** Nave 3 ***
Posição: (-41360.565,-41360.565)
```

Distância ao astro mais próximo: 23492.672
***** iteração 3 *****
*** Nave 1 ***
Posição: (134139.550,164107.055)
Distância ao astro mais próximo: 171954.109
*** Nave 2 ***
Posição: (-154167.867,-144083.900)
Distância ao astro mais próximo: 161016.354
*** Nave 3 ***
Posição: (-33807.925,-33807.925)
Distância ao astro mais próximo: 12811.627
***** iteração 4 *****
*** Nave 1 ***
Posição: (128487.421,168396.410)
Distância ao astro mais próximo: 171816.827
*** Nave 2 ***
Posição: (-158501.210,-138385.167)
Distância ao astro mais próximo: 160411.710
*** Nave 3 ***
Posição: (-23055.590,-23055.590)
Distância ao astro mais próximo: 0.000
***** iteração 5 *****
*** Nave 1 ***
Posição: (122664.099,172469.016)
Distância ao astro mais próximo: 171641.307
*** Nave 2 ***
Posição: (-162627.302,-132499.228)
Distância ao astro mais próximo: 159770.553
*** Nave 3 ***
Posição: (-23055.590,-23055.590)
Distância ao astro mais próximo: 0.000
***** iteração 6 *****
*** Nave 1 ***
Posição: (116676.702,176318.929)
Distância ao astro mais próximo: 171428.043
*** Nave 2 ***
Posição: (-166538.637,-126432.163)
Distância ao astro mais próximo: 159093.781
*** Nave 3 ***
Posição: (-23055.590,-23055.590)
Distância ao astro mais próximo: 0.000
***** iteração 7 *****
*** Nave 1 ***
Posição: (110532.516,179940.342)
Distância ao astro mais próximo: 171177.565
*** Nave 2 ***
Posição: (-170227.756,-120190.325)
Distância ao astro mais próximo: 158382.348

```

*** Nave 3 ***
Posição: (-23055.590,-23055.590)
Distância ao astro mais próximo: 0.000
***** iteração 8 *****
*** Nave 1 ***
Posição: (104239.000,183327.592)
Distância ao astro mais próximo: 170890.434
*** Nave 2 ***
Posição: (-173687.252,-113780.360)
Distância ao astro mais próximo: 157637.260
*** Nave 3 ***
Posição: (-23055.590,-23055.590)
Distância ao astro mais próximo: 0.000
***** iteração 9 *****
*** Nave 1 ***
Posição: (97803.788,186475.159)
Distância ao astro mais próximo: 170567.248
*** Nave 2 ***
Posição: (-176909.779,-107209.214)
Distância ao astro mais próximo: 156859.579
*** Nave 3 ***
Posição: (-23055.590,-23055.590)
Distância ao astro mais próximo: 0.000
***** iteração 10 *****
*** Nave 1 ***
Posição: (91234.701,189377.668)
Distância ao astro mais próximo: 170208.639
*** Nave 2 ***
Posição: (-179888.061,-100484.148)
Distância ao astro mais próximo: 156050.428
*** Nave 3 ***
Posição: (-23055.590,-23055.590)
Distância ao astro mais próximo: 0.000
***** iteração 11 *****
*** Nave 1 ***
Posição: (84539.747,192029.897)
Distância ao astro mais próximo: 169815.277
*** Nave 2 ***
Posição: (-182614.902,-93612.753)
Distância ao astro mais próximo: 155210.989
*** Nave 3 ***
Posição: (-23055.590,-23055.590)
Distância ao astro mais próximo: 0.000
***** iteração 12 *****
*** Nave 1 ***
Posição: (77727.131,194426.782)
Distância ao astro mais próximo: 169387.871
*** Nave 2 ***

```


Posição: (-185083.197, -86602.959)
Distância ao astro mais próximo: 154342.512
*** Nave 3 ***
Posição: (-23055.590, -23055.590)
Distância ao astro mais próximo: 0.000

Observação importante:

- O protótipo das funções (tipo de valor de retorno, nome da função, tipo e ordem dos parâmetros) **NÃO** deve ser alterado. Vocês podem escrever mais funções auxiliares se acharem conveniente e podem usar qualquer função do módulo math se julgarem necessário.

Referências

- [1] Herch Moysés Nussenzveig, *Curso de Física Básica: 1-Mecânica*, Edgard Blücher, 1981.
- [2] Hugh D. Young e Roger A. Freedman, *Sears e Zemansky's Física I: Mecânica*, Addison Wesley, 2003.