

Introdução à Computação em Física

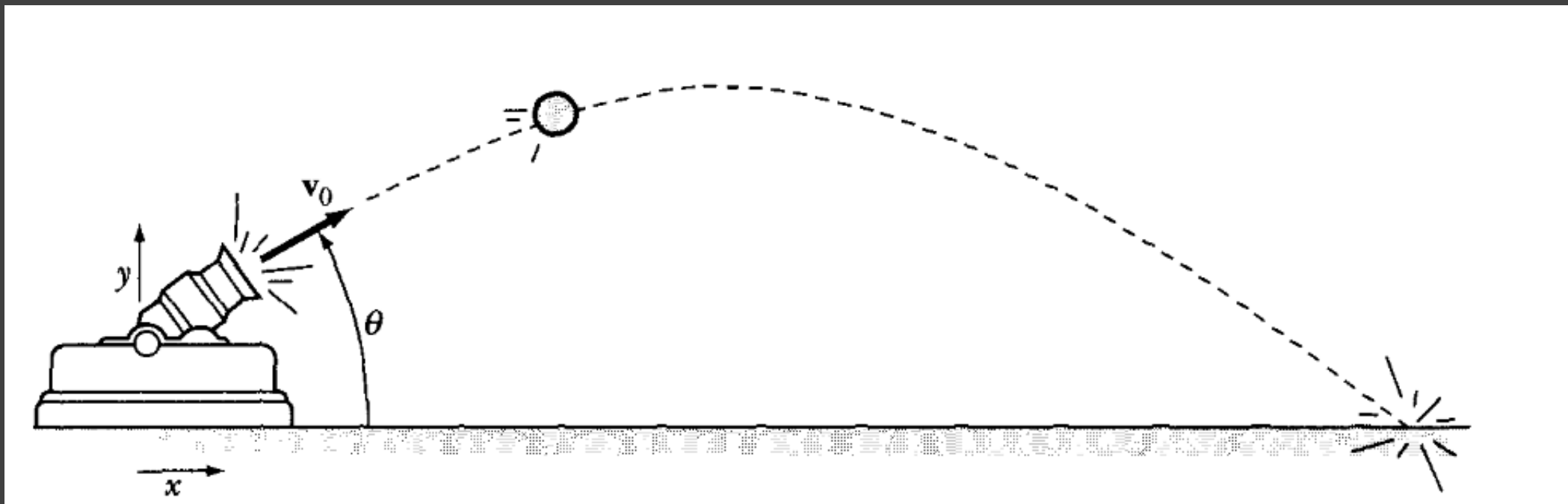
SIMULAÇÕES 3D EM
PYTHON (VPYTHON)

PROF. WALBER

Refs.:

<https://www.glowscript.org/docs/VPythonDocs/index.html>

Aulas anteriores: Lançamento projétil (sem e com resistência ar)



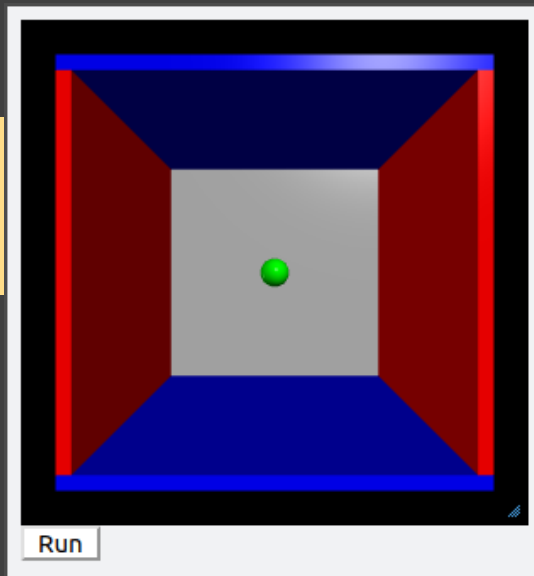
$$v_{0,y} = v_0 \sin \theta$$

$$v_{0,x} = v_0 \cos \theta$$

Seria possível elaborar uma simulação em Python (com objetos 3D) desse lançamento do projétil.

VPython:

O VPython é uma biblioteca que auxilia na produção de gráficos, objetos 3D, de maneira simplificada :



Instalação em:

<https://vpython.org/presentation2018/install.html>

(Não será feito no jupyter Lab)

```
from vpython import *
```

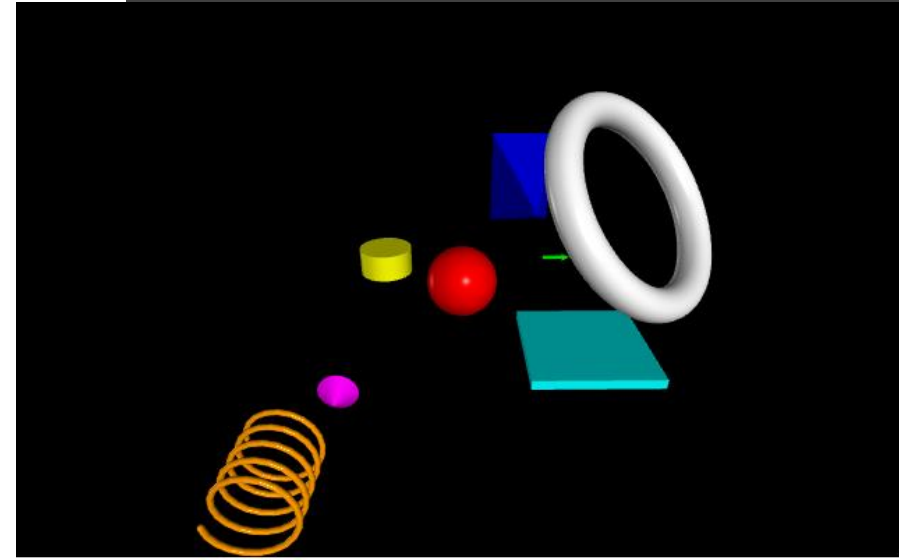
<https://vpython.org/>

(Sugestão para edição online: glowsript.org)
Criar conta e editar no navegador.

Ingredientes básicos (objetos)

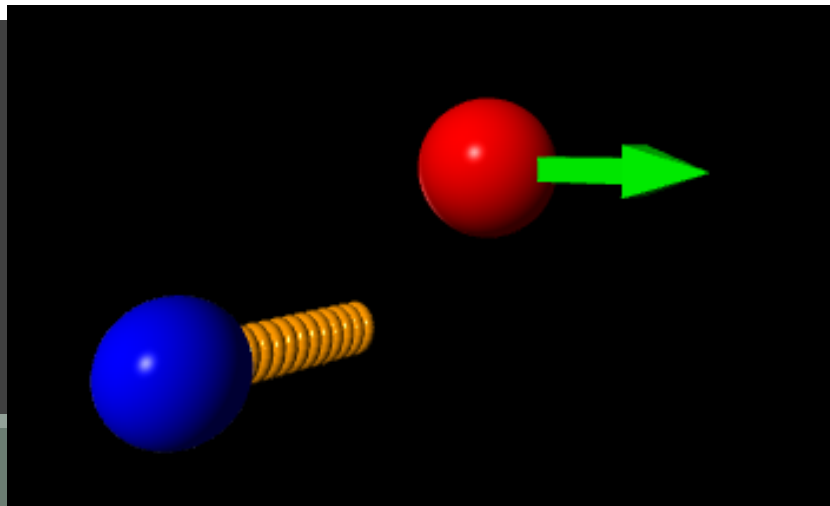
Objetos 3D: podem ser criados com os comandos

```
1 from vpython import *
2
3
4 ## caixa, com centro na posicao (-5,2,2) cartesianas, comprimento 5, largura 5, altura 0.4
5
6 box(pos=vector(5,-2,2), length=5, width=5, height=0.4, color=color.cyan)
7
8 ## esfera com centro na origem, raio 1.5
9
10 sphere(pos=vector(0,0,0), radius=1.5, color=color.red)
11
12 ## seta posicao (3,2,2) eixo direcao (0,0,1)
13
14 arrow(pos=vector(3,2,2), axis=vector(1,0,0), color=color.green)
15
16 ## cilindro
17
18 cylinder(pos=vector(-3,1,1), axis=vector(0,1,0), color=color.yellow)
19
20 ## cone
21
22 cone(pos=vector(-6,-6,0), axis=vector(0,0,1), color=color.magenta)
23
24
25 ## piramide
26
27 pyramid (pos=vector(2,5,2), size=vector(4,3,2), axis=vector(0,0,1), color=color.blue)
28
29
30 ### mola
31
32 helix(pos=vector(-9,-9, 0), axis = vector(0,0,5), radius = 2, thickness = 0.3, color=color.orange)
33
34
35 ## anel
36
37 ring(pos= vector(5,5,5), axis = vector(1,1,1), radius = 3, thickness = 0.5)
38
```



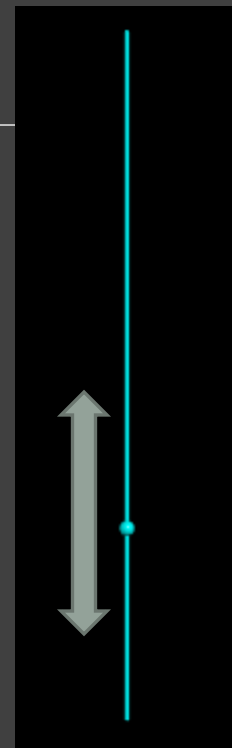
Acoplando objetos:

```
1 from vpython import *
2
3
4 ## esfera com centro na origem, raio 1.5
5
6 sphere(pos=vector(0,0,0), radius=1.5, color=color.red)
7
8 ## seta posicao (0,0,0) eixo direcao (5,0,0)
9
10 arrow(pos=vector(0,0,0), axis=vector(5,0,0), color=color.green)
11
12 ### mola
13
14 helix(pos=vector(-4,-4, 0), axis = vector(0,0,5), radius = 0.5, thickness = 0.3, coils=15 ,color=color.orange)
15
16
17 sphere(pos=vector(-4,-4,5), radius=1.5, color=color.blue)
18
```



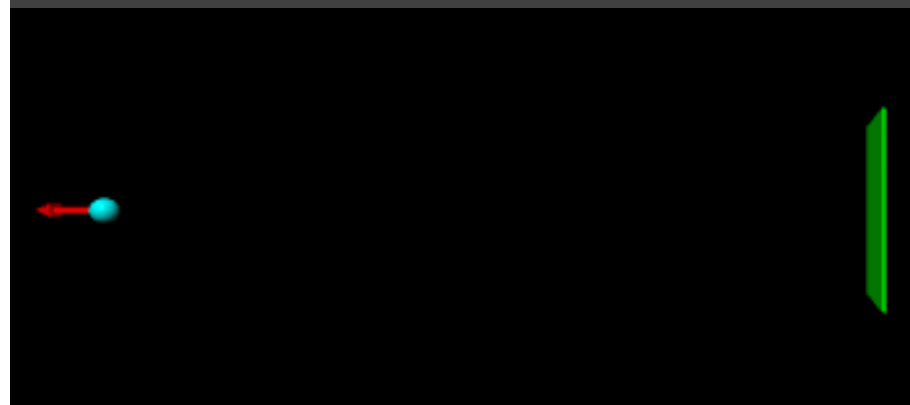
Exemplo 1: Atualizando posição objeto (dinâmica)

```
1 from vpython import *
2 import numpy as np
3
4
5 ### Bola em MHS
6
7 bola = sphere(pos=vector(0,0,0), radius = 4.5, color=color.cyan, make_trail = True)
8
9 ### posicao da particula
10
11 deltat = 0.010
12
13 t = 0
14
15 tfinal = 100
16
17 while t < tfinal:
18
19     —» rate(100)    ### while nao ira executar mais que 100 vezes por segundo
20     —» bola.pos = vector(bola.pos) + vector(0,10*np.cos(5*t),0)
21     —» t += deltat
```

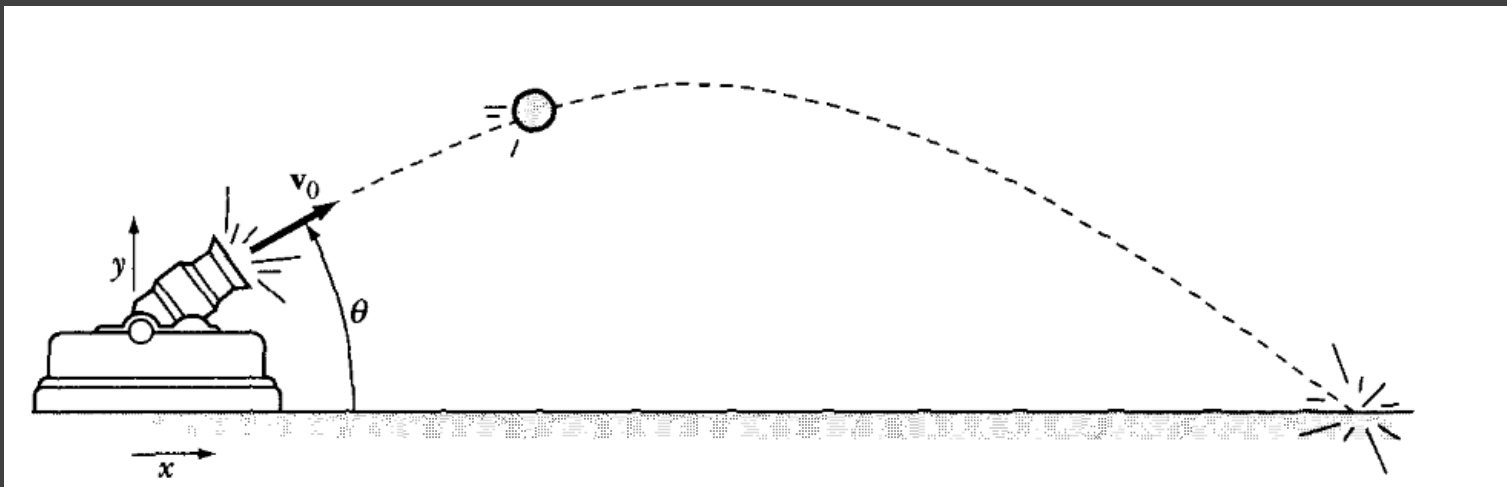


Exemplo 2: Partícula colidindo com parede

```
1 from vpython import *
2
3
4 ### Bola
5
6 bola = sphere(pos=vector(-20,0,0), radius = 0.8, color=color.cyan)
7
8 ### parede
9
10 paredeDir = box(pos=vector(5,0,0), length=0.2, width=12, height=12, color=color.green)
11
12 ### velocidade
13
14 ### velocidade inicial bola na direcao x
15 bola.veini = vector(15,0,0)
16
17
18 deltat = 0.010
19 t = 0
20 tfinal = 5
21
22
23 arscale = 0.3
24 viarr = arrow(pos=bola.pos, axis=arscale*bola.veini, color=color.red)
25
26
27 while t < tfinal:
28     rate(100)
29
30
31     if bola.pos.x > paredeDir.pos.x:
32         bola.veini.x = (-1)*bola.veini.x    ### bola eh refletida ao se chocar na parede
33
34
35     bola.pos = bola.pos + bola.veini*deltat    ### atualiza posicao da bola
36     viarr.pos = bola.pos    ### atualiza posicao da seta
37     viarr.axis = arscale*bola.veini    ### atualiza direcao seta
38
39     t += deltat
```



Exemplo 3: Lançamento projétil (sem resistência ar)



$$v_{0,x} = v_0 \cos \theta$$

$$v_{0,y} = v_0 \sin \theta$$

$$\vec{F}_{res} = m\vec{g}$$

$$\vec{p}_f = \vec{p}_i + \vec{F}_{res} \Delta t$$

$$\vec{r}_f = \vec{r}_i + \frac{\vec{p}}{m} \Delta t$$

Serão utilizadas para implementação de nossa simulação

```

1 from vpython import *
2 import numpy as np
3
4 scene.title='Movimento Projétil'
5 scene.width=scene.height= 800      # torna a janela um pouco maior
6
7 ##### condicoes iniciais
8
9 theta = 70.0          ##### angulo inicial em graus
10 thetar = theta*np.pi/180.0
11 vini = 120.0          ##### modulo velocidade inicial
12
13
14 ### aceleracao gravidade
15
16 g = vector(0.0,-9.8,0.0)
17
18 ##### Desenhando o plano horizontal (xz):'
19 box(pos=vector(0,0.0,0), size=vector(1000.0,10,1000.0), color=color.green)
20
21 ### projétil (esfera)
22 raio = 30.0
23
24 proj = sphere(pos=vector(-500,raio,0), radius = raio, color=color.white, make_trail = True)
25
26 ### velocidade inicial projétil
27
28 projvini= vector(vini*np.cos(thetar), vini*np.sin(thetar),0)
29
30 projm = 0.5
31
32 ### momento linear
33
34 projp = projm*projvini
35
36
37 ### forca resultante
38
39 Fres = projm*g
40
41 print("Forca resultante = ", Fres)

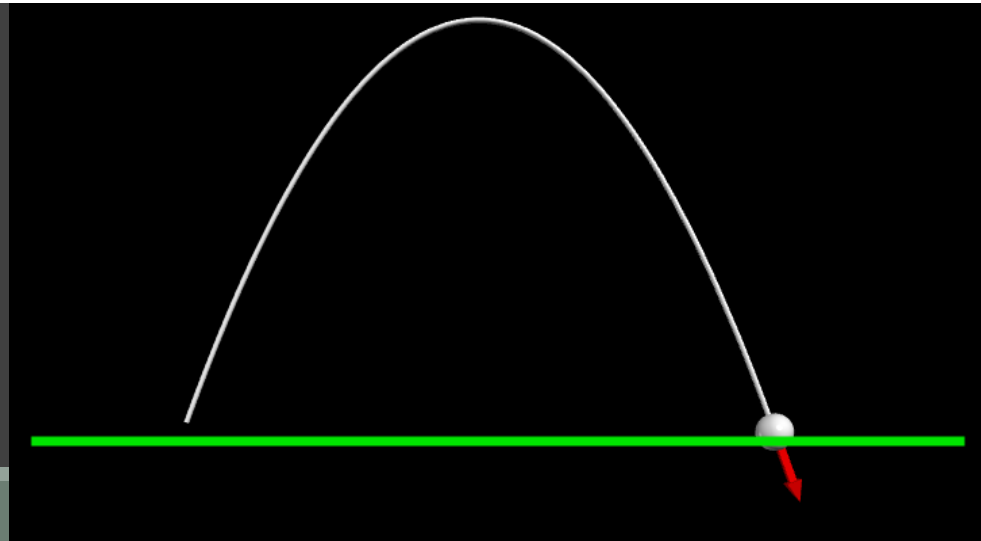
```

Exemplo 3: Lançame nto projétil

```

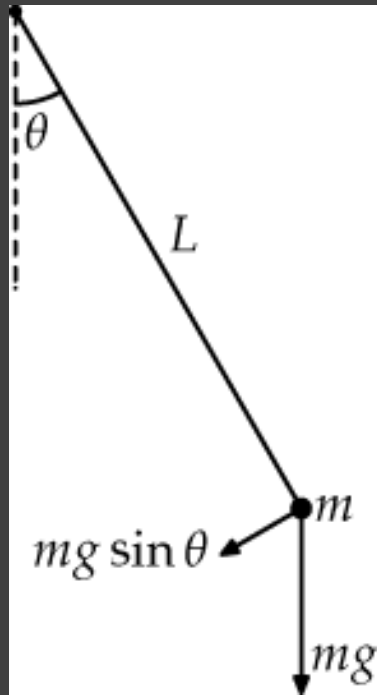
42
43 t = 0.0
44 dt = 0.001
45
46 arscale = 2
47 parr = arrow(pos=proj.pos, axis=arscale*projp, color=color.red)
48
49 while (proj.pos.y >= raio):
50
51     rate(50)
52
53
54     projp = projp + (Fres)*t    ### momento linear projetil
55
56     proj.pos = proj.pos + (projp/projm)*t    #### posicao projetil
57
58
59     parr.pos = proj.pos
60     parr.axis = arscale*projp
61
62     t = t + dt
63

```



Exemplo 3: Lançamento projétil

Exemplo 4: Pêndulo simples



$$y = L - L \cos \theta$$

$$x = L \sin \theta$$

$$\frac{d^2 \theta}{dt^2} + \frac{g}{L} \sin \theta = 0$$

$$\sin \theta \approx \theta$$

$$\frac{d^2 \theta}{dt^2} + \omega_0^2 \theta = 0$$

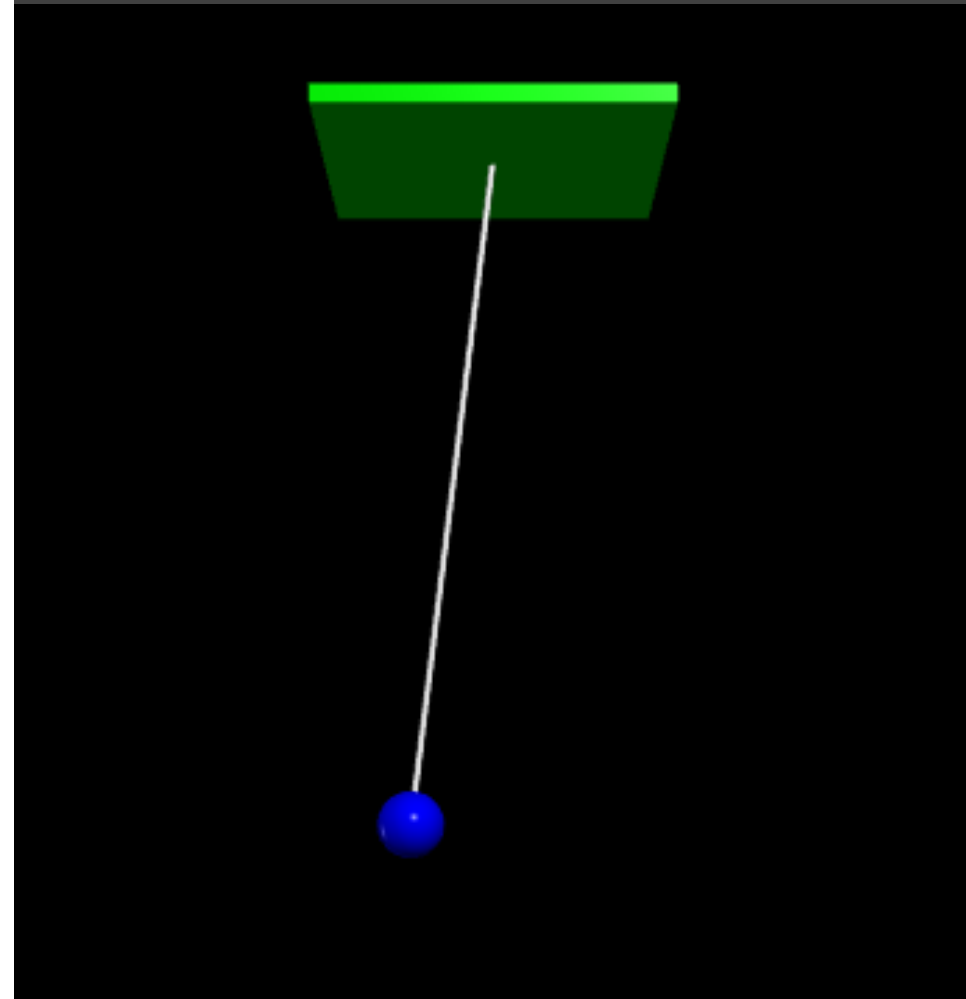


$$\theta(t) = B \cos(\omega_0 t - \varphi)$$

$$\omega_0^2 = \frac{g}{L}$$

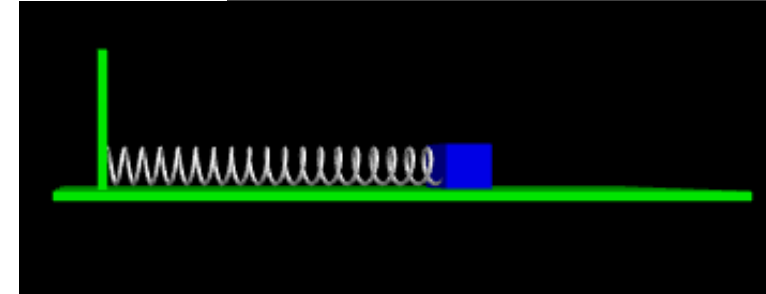
Exemplo 4: Pêndulo simples

```
1 from vpython import *
2 import numpy as np
3
4 scene.width=scene.height= 800
5
6 ## ponto conexao
7 conex = vector(0,20,0)
8
9 box(pos=conex, size=vector(10.0,0.5,10.0), color=color.green)
10
11 ### esfera
12
13 esf = sphere(pos=vector(2,2,0), radius=1, color=color.blue)
14
15 ## fio
16
17 fio=cylinder(pos=conex, axis=esf.pos-conex, radius=0.1, color=color.white)
18
19 ## comprimento do fio
20
21 L = mag(esf.pos-conex)    ### modulo vetor esf.pos-conex
22
23 cost = (conex.y-esf.pos.y)/L    ### calculo cos(theta)
24
25 thetaini = acos(cost)
26
27 print ("Angulo inicial = ", thetaini)
28
29 ##### dinamica
30
31 t = 0.0
32 dt = 0.01
33 tf = 100.00
34
35 g = 9.8
36 omg = np.sqrt(g/L)
37 B = 0.5
38
39 while (t < tf):
40     rate(100)
41     theta = B*np.cos(omg*t)    ### theta(t)
42
43     esf.pos = vector(L*np.sin(theta), L-L*np.cos(theta),0)
44     fio.axis = esf.pos-fio.pos
45     t = t + dt
```

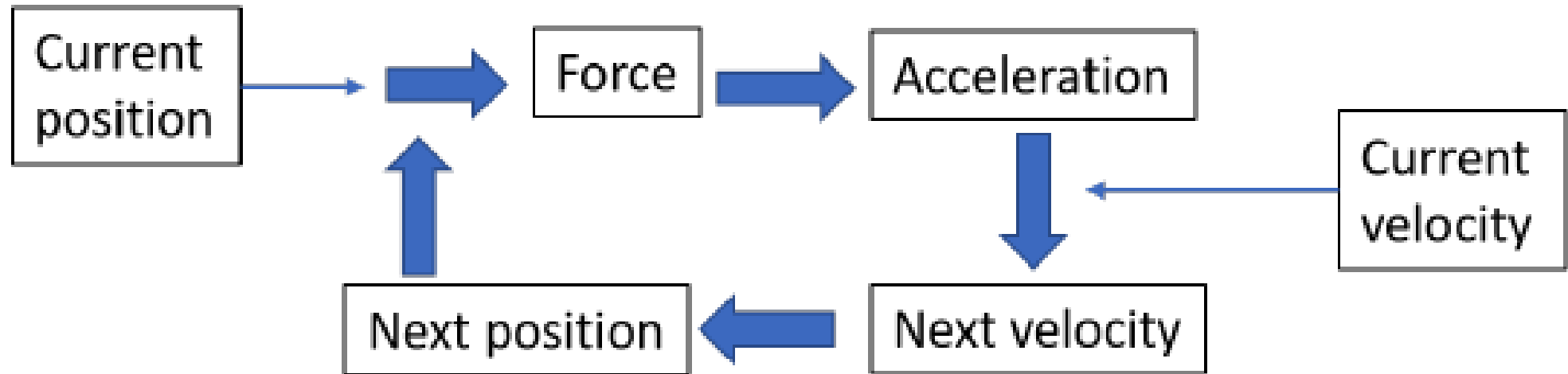


Exemplo 5: Sistema Massa Mola

```
1 from vpython import *
2 import numpy as np
3
4 ##### Objetos 3D da simulacao
5
6 parede = box(pos=vector(0,1,0), size=vector(0.2, 3,2), color=color.green)
7
8 piso = box(pos=vector(6,-0.6,0), size=vector(14,0.2, 4), color=color.green)
9
10 ### bloco massa 1 kg
11
12 bloco = box(pos=vector(12,0,0), size=vector(1,1,1), mass = 1.0, color=color.blue)
13
14 ## ponto conexao
15 conex = vector(0,0,0)
16
17
18 mola = helix(pos=conex, axis=bloco.pos-conex, radius=0.4, constant= 3 , thickness = 0.1, coils=20, color=color.white)
19
20 ### posicao equilibrio
21
22 equi = vector(9,0,0)
23
24
25 t = 0.0
26 dt = 0.01
27
28 tf = 100
29
30 omg = np.sqrt(mola.constant/bloco.mass)
31 B = 1.0
32
33 while (t < tf):
34     rate(100)
35
36     bloco.pos = vector(equi.x+B*cos(omg*t),0,0)
37     mola.axis = bloco.pos - mola.pos
38     t = t + dt
```



Em muitos casos a ideia pode ser resumida como



Movimento de objetos em 3D;

Atividades práticas:

1. Escreva um programa que simule o lançamento de um projétil (esfera) com resistência do ar:

$$\vec{F}_{ar} = -\frac{1}{2}\rho ACv^2\hat{v}$$

densidade do ar (ρ) = 0.2

$$A = \pi r^2$$

área de referência da esfera (r = raio)

Dados adicionais
(Sugestões):

$C = 0.001$

Coeficiente de arrasto da esfera

$V_{ini} = 600 \text{ m/s}$

Ang. Inicial = 70 graus

Raio = 5.0 m

$g = 9.8 \text{ m/s}^2$

$M = 1.0 \text{ Kg}$

Altere os valores do raio e veja a influência no alcance do projétil

Atividades práticas:

2. Escreva um programa que simule o lançamento de uma bolinha, sob ação da resistência do ar, e que vai quicando até parar. Siga os seguintes passos:

(a). Desenhe o piso e um conjunto de eixos (x,y,z) (com setas nas pontas) indicando a direção dos mesmos. Para isso utilize cilindros e cones (cones serão as pontas dos eixos). Utilize o comando `label(pos= ..., text='NOME EIXO', box=0,opacity=0,color=...)` para colocar as letras nos eixos.

(b). Considere que a bolinha se desloca sob ação de uma força de resistência do ar:

$$\vec{F}_{ar} = -Cv\vec{v}$$

Onde $C = 0.0008$
 v é a velocidade

Atividades práticas:

(c). Considere os seguintes dados adicionais (sugestões):

Posição inicial (0,0,0)

$V_{ini}(x) = 10 \text{ m/s}$

$V_{ini}(y) = 20 \text{ m/s}$

Raio = 1 m

$g = 9.8 \text{ m/s}^2$

$M = 0.143 \text{ Kg}$

(d). A bola deve "quicar" ao encontrar o piso. A trajetória da bolinha deve ser desenhada.

(e). Escreva na tela a posição final da partícula após 50 s.

Atividades práticas:

3. Escreva um programa em Python que simule o movimento de um pêndulo cônico, como mostrado na figura abaixo. Considere o comprimento do fio $L = 20$ m, velocidade angular $\omega = 2$ rad/s e ângulos iniciais $\theta = \pi/6$ e $\phi = \pi/4$. Note que o ângulo ϕ está relacionado a posição do corpo no plano horizontal.

