Questao2

October 6, 2019

[1]:
```
from vpython import vector, sphere, rate, cylinder, color, canvas
from math import cos, sin, pi
from numpy import arange
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.Javascript object>
```

1 Padrao de cores mostrado no video

[2]:
```
color_map = [
    color.red, color.orange, color.yellow, color.green, color.blue, color.
 ↪purple,
    color.red, color.orange, color.yellow, color.green, color.blue, color.
 ↪purple,
    color.red, color.orange, color.yellow
]
```

2 Calculos para o comprimento de cada pendulo

A equacao diferencial que representa o movimento de um pendulo simples é

$$\frac{d^2\theta}{dt^2} + \frac{g}{l}\sin\theta = 0 \tag{1}$$

onde g e a aceleracao da gravidade, l e o cumprimento do pendulo, e θ é o deslocamento angular.

A equacao 1 nao e facil de resolver, e nao tem solucao em termos de funcoes elementares. Nao obstante, adiconando uma restricao a amplitudo do oscilador resulta numa forma cuja solucao pode ser obtida facilmente. Se for assumido que o angulo e muito pequeno, entao substituindo por $\sin\theta$ na equacao 1 usando a aproximacao de pequenos angulos,

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

proporciona a equacao do oscilador harmonico

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

Dadas as condicoes iniciais $\theta(0) = \Theta$ e e $\frac{d\theta}{dt}(0) = 0$, a solucao para o n-esimo pendulo se torna

$$\theta_n(t) = \Theta \cos(\omega_n t)$$

com

$$\omega_n = \sqrt{\frac{g}{l_n}} \tag{2}$$

dado que

$$\omega_n = 2\pi freq_n \tag{3}$$

[3]:
```
per = 60 # segundos
freq_ls = [osc/per for osc in range(51, 66)]

# Constantes
R_esfera = 1e-2
R_fio = 1e-3
g = 9.8
omega = [2*pi*f for f in freq_ls] #eq. 3
L_fio = [g/w**2 for w in omega] # eq. 2
tmax = per
fps = 30
dt = 1/fps

# Objetos
A = 6*pi/180
theta = [A for _ in range(15)]
x = [ L_fio[i]*sin(theta[i]) for i in range(15)]
y = [-L_fio[i]*cos(theta[i]) for i in range(15)]

scene = canvas()
scene.autoscale = False

esferas = [
    sphere(
        canvas=scene,
        radius=R_esfera,
        color=color_map[i],
        pos=vector(x[i],y[i],i)
    ) for i in range(15)
]
fio = [cylinder(
    canvas=scene,
    radius=R_fio,
```

```python
    axis=esfera.pos
) for esfera in esferas]

# Laco da animacao
for t in arange(0, tmax, dt):
    rate(fps)

    for i in range(15):
        theta[i] = A*cos(omega[i]*t)
        x[i], y[i] = L_fio[i]*sin(theta[i]), -L_fio[i]*cos(theta[i])
        esferas[i].pos = vector(x[i],y[i],i*R_esfera*3)
        fio[i].pos = vector(0,0,i*R_esfera*3)
        fio[i].axis = vector(x[i],y[i],0)

    # angulo semelhante
    scene.camera.pos = vector(0, -L_fio[0], 0.55)
    scene.camera.axis = vector(0, .5, -1)
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

3 Modificacoes para periodo do video

Para o script proposto acima, basta alterar o valor da variavel *per* para modificar a duracao do periodo da danca em segundos.

4 Outras combinacoes que produzem a danca

O aluno entende que o fenomeno da “danca”, trata-se da manifestacao de batimento em osciladores de frequencias comensuraveis, que determinam harmonicos da frequencia do sistema (no caso, 60 segundos, ou 100 no video). Sendo assim, eh possivel dizer que ha outras configuracoes que compoem uma “danca”, a denpender das harmonia entre as frequencias que participam do sistema.