Preliminares

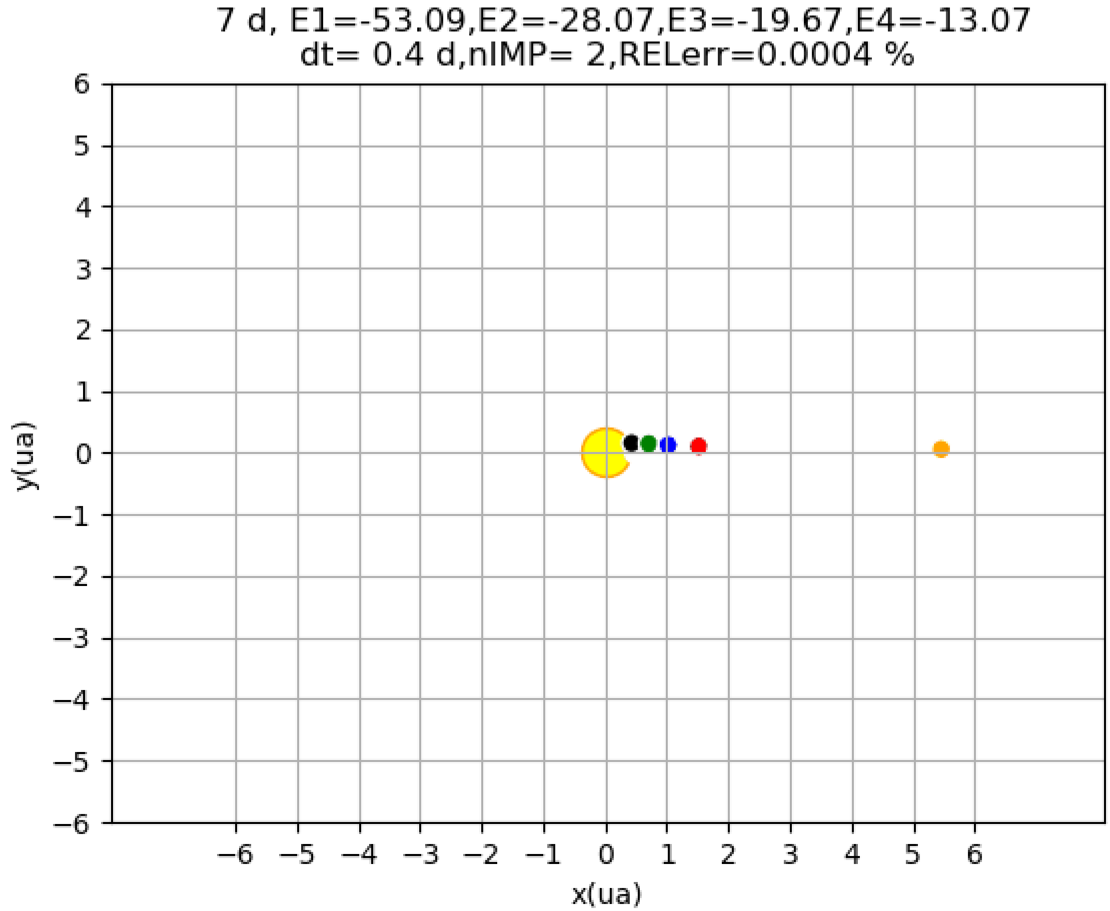
Queremos otimizar as aulas remotas:

- Mandem perguntas/tópicos para a próxima aula
- Mandem sugestões e críticas por mail

Vamos tentar manter os objetivos fixados: o projeto B será aberto após a Páscoa

Aula 11

Equações diferenciais com condições fronteira num ponto



Problemas de valores iniciais

$$\frac{dT}{dt} = -\alpha(T - T_{ar})$$

$$\frac{d^2\vec{r}}{dt^2} = \vec{g} \quad , \qquad \vec{g} = const$$

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x$$

$$\frac{d^2\vec{r}}{dt^2} = \vec{g} = g_x\vec{e}_x + g_y\vec{e}_y, \qquad \vec{g} = -\frac{GM_{Sol}}{d^2}\vec{u}$$

$$\frac{d^2\vec{r}}{dt^2} = \frac{d\vec{v}}{dt} = \vec{g} - 2\vec{\Omega} \times \vec{v}$$

Arrefecimento de um corpo por convecção

Lei de Newton do arrefecimento

$$\frac{dT}{dt} = -\alpha(T - T_{ar})$$

Em que α é o coeficiente de transferência e $T(t = 0) = T_0$ é a condição inicial.

Com $\alpha = const$, existe solução analítica:

$$T = (T_0 - T_{ar})e^{-\alpha t}$$

A solução numérica de uma equação diferencial

Exige a sua transformação num sistema de equações algébricas.

A transformação mais simples consiste na utilização de diferenças finitas.

Diferenças finitas de 1ªordem

Série de Taylor

$$y(x+\Delta x) = y(x) + \frac{dy}{dx}\Delta x + \frac{1}{2}\frac{d^2y}{dx^2}\Delta x^2 + \frac{1}{3!}\frac{d^3y}{dx^3}\Delta x^3 + \cdots$$

Desprezando os termos de ordem 2 e superior:

$$y(x+\Delta x) = y(x) + \frac{dy}{dx}\Delta x + E\left(\Delta x^2\right)$$

ou

$$\left(\frac{dy}{dx}\right)_{x=a} \approx \frac{y(a+\Delta x) - y(a)}{\Delta x} + E(\Delta x)$$

Se for $T(t)$, temos a diferença avançada:

$$\left(\frac{dT}{dt}\right)_{t=t} \approx \frac{T(t+\Delta t) - T(t)}{\Delta t} + E(\Delta \mathrm{t})$$

Diferenças finitas centradas (2ª ordem)

Fazendo

$$y(x+\Delta x) = y(x) + \frac{dy}{dx}\Delta x + \frac{1}{2}\frac{d^2y}{dx^2}\Delta x^2 + \frac{1}{3!}\frac{d^3y}{dx^3}\Delta x^3 + \cdots$$

$$y(x-\Delta x) = y(x) - \frac{dy}{dx}\Delta x + \frac{1}{2}\frac{d^2y}{dx^2}\Delta x^2 - \frac{1}{3!}\frac{d^3y}{dx^3}\Delta x^3 + \cdots$$

Subtraindo

$$y(x+\Delta x) - y(x-\Delta x) = 2\frac{dy}{dx}\Delta x + E\left(\Delta x^4\right)$$

ou

$$\left(\frac{dy}{dx}\right)_{x=a} \approx \frac{y(a+\Delta x) - y(a-\Delta x)}{\Delta x} + E\left(\Delta x^2\right)$$

Diferenças finitas

Fazendo

$$y(x+\Delta x) = y(x) + \frac{dy}{dx}\Delta x + \frac{1}{2}\frac{d^2y}{dx^2}\Delta x^2 + \frac{1}{3!}\frac{d^3y}{dx^3}\Delta x^3 + \cdots$$

$$y(x-\Delta x) = y(x) - \frac{dy}{dx}\Delta x + \frac{1}{2}\frac{d^2y}{dx^2}\Delta x^2 - \frac{1}{3!}\frac{d^3y}{dx^3}\Delta x^3 + \cdots$$

Somando

$$y(x+\Delta x) + y(x-\Delta x) = 2y(x) + \frac{d^2y}{dx^2}\Delta x^2 + E\left(\Delta x^4\right)$$

ou

$$\left(\frac{d^2y}{dx^2}\right)_{x=a} \approx \frac{y(a+\Delta x) + y(a-\Delta x) - 2y(a)}{\Delta x^2} + E\left(\Delta x^2\right)$$

Método de Euler para o arrefecimento convectivo

$$\frac{dT}{dt} = -\alpha(T - T_{ar})$$

Usamos diferenças avançadas (1ª ordem)

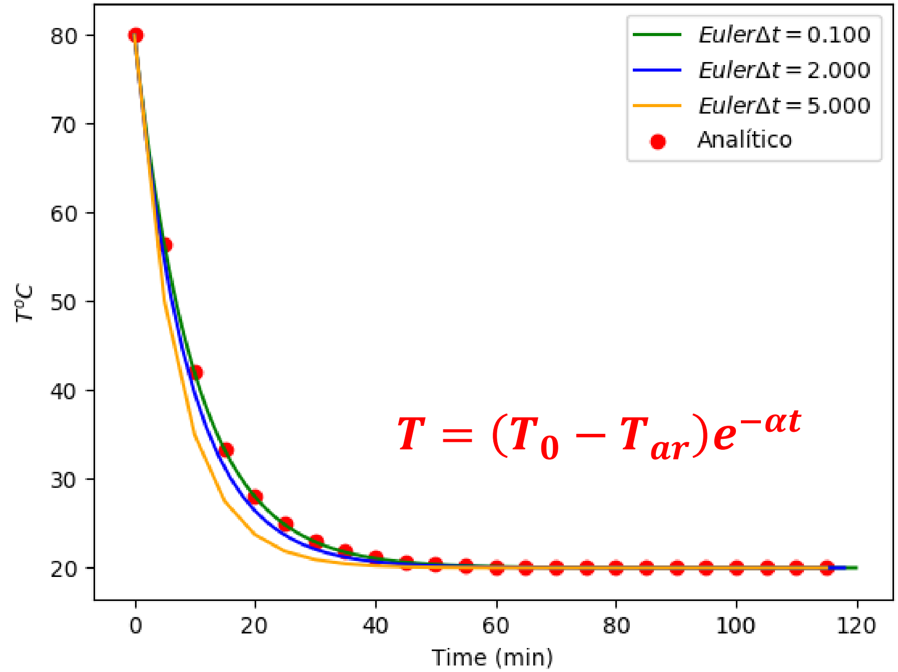
$$\frac{T(t + \Delta t) - T(t)}{\Delta t} = -\alpha(T(t) - T_{ar})$$

Resolvendo para o futuro temos uma recursão

$$T(t + \Delta t) = T(t) - \alpha\Delta t(T(t) - T_{ar})$$

A solução depende da condição inicial $T(t = 0) = T_0$ e dos parâmetros T_{ar}, α.

$$\frac{dT}{dt} = -\alpha(T - T_{ar})$$



$$T(t + \Delta t) = T(t) - \alpha \Delta t (T(t) - T_{ar})$$

```
import numpy as np
import matplotlib.pyplot as plt
Tar=20; Tinicial=80; alpha=0.1
cores=['green','blue','orange']; kc=0
for dt in[0.1,2,5]:
    tempo=np.arange(0.,120.,dt)
    n=len(tempo)
    T=np.zeros(tempo.shape)
    T[0]=Tinicial
    for k in range(1,n): #Euler
        T[k]=T[k-1]-alpha*(T[k-1]-Tar)*dt
    plt.plot(tempo,T,color=cores[kc],label=r'$Euler \Delta t=%6.3f$'%(dt))
    plt.xlabel('Time (min)'); plt.ylabel(r'$T ^{o} C$');    kc=kc+1
tempoAn=np.arange(0.,120.,5)
Tan=Tar+(Tinicial-Tar)*np.exp(-alpha*tempoAn)
plt.scatter(tempoAn,Tan,color='red',label='Analítico')
plt.legend()
```

Movimento balístico $(g = const)$

$$\vec{F} = m\vec{a} \Longrightarrow \frac{d\vec{v}}{dt} = \frac{d^2\vec{r}}{dt^2} = \vec{g} = -g\vec{k}$$

Com $\vec{r} = x\vec{\imath} + y\vec{\jmath} + z\vec{k}$

Trata-se de uma equação (vetorial) de segunda ordem para a posição.

Vamos escrever como um sistema de 6 equações de 1ª ordem: de facto 2 são triviais $(u, v = const)$.

$$\begin{cases} \dfrac{du}{dt} = 0 \\ \dfrac{dv}{dt} = 0 \\ \dfrac{dw}{dt} = -g \\ \dfrac{dx}{dt} = u \\ \dfrac{dy}{dt} = v \\ \dfrac{dz}{dt} = w \end{cases}$$

Método de Euler

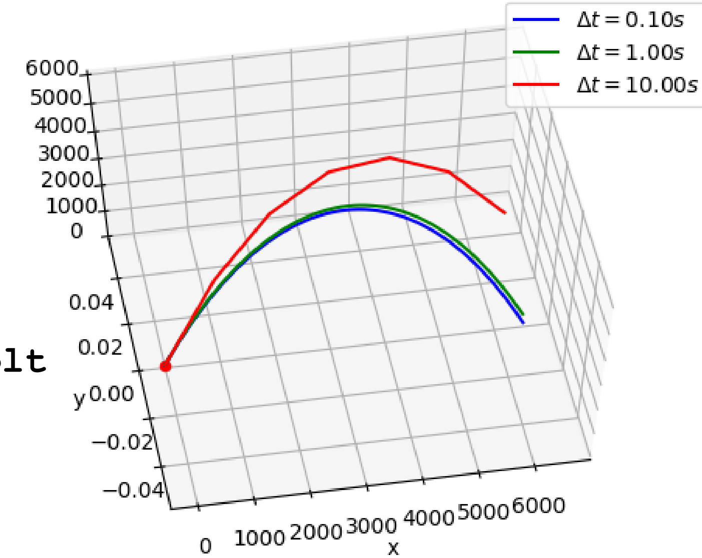
$$
\begin{cases}
\dfrac{du}{dt} = 0 \\
\dfrac{dv}{dt} = 0 \\
\dfrac{dw}{dt} = -g \\
\dfrac{dx}{dt} = u \\
\dfrac{dy}{dt} = v \\
\dfrac{dz}{dt} = w
\end{cases}
$$

$$
\begin{gathered}
w(t+\Delta t) = w(t) - g\Delta \mathrm{t} \\
x(t+\Delta t) = x(t) + u\Delta t \\
y(t+\Delta t) = y(t) + v\Delta t \\
z(t+\Delta t) = z(t) + w(t)\Delta t
\end{gathered}
$$

Com condições iniciais

$$
\begin{gathered}
x(0) = x_0 \\
y(0) = y_0 \\
z(0) = z_0 \\
u(0) = u \\
v(0) = v \\
w(0) = w_0
\end{gathered}
$$

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




```python
import numpy as np;import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
plt.close('all');g=9.8065
x0=0;y0=0;z0=0 #posição inicial
u=100;v=0;w0=320 #velocidade inicial
timeInt=2*w0/g #tempo máximo de integração
fig=plt.figure();ax=fig.add_subplot(111, projection='3d')
c=['blue','green','red'];kc=-1;ax.scatter(x0,y0,z0,color='red')
for dt in[0.1,1,10]:
    kc=kc+1;tempo=np.arange(0.,timeInt,dt);n=len(tempo)
    X=np.zeros((n));Y=np.copy(X);Z=np.copy(X)
    X[0]=x0;Y[0]=y0;Z[0]=z0;w=w0
    for kt in range(1,n):
        X[kt]=X[kt-1]+u*dt; Y[kt]=Y[kt-1]+v*dt; Z[kt]=Z[kt-1]+w*dt
        w=w-g*dt
    ax.plot(xs=X,ys=Y,zs=Z,color=c[kc],label=r'$\Delta t=%6.2f s $'%(dt))
plt.legend()
ax.set_zlim(0,6000);ax.set_xlabel('x');ax.set_ylabel('y')
```

Funciona com Δt pequeno

Oscilador harmónico (lei de Hooke)

Oscilador 1D: 2 equações diferenciais de 1ª ordem:

$$\frac{dv}{dt} = \frac{F}{m} = -\frac{k}{m}x$$

$$\frac{dx}{dt} = v$$

A aceleração é variável (mais complicado que o caso $g = const$), depende só da posição.

(Tal como no movimento balístico) o sistema deve conservar energia mecânica:

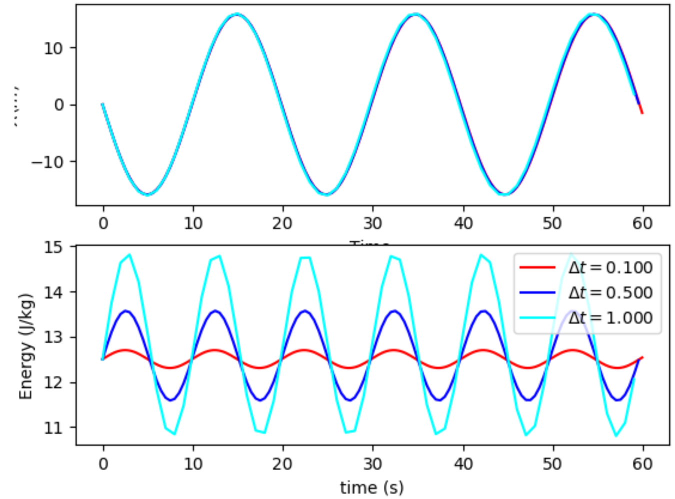
$$\frac{E_M}{m} = \frac{1}{2}\, v^2 + \frac{kx^2}{2} = const$$

Existe solução analítica:

$$x = A\cos(\omega t + \phi)$$

Onde a Amplitude (A) e a fase inicial (ϕ) dependem das condições iniciais, e $\omega = \frac{2\pi}{T} = \sqrt{\frac{k}{m}}$

Oscilador: método de Euler




```
import numpy as np
import matplotlib.pyplot as plt
xinicial=0; vinicial=-5; kapa=0.1 # k/m (lei de Hooke)
cores=['red','blue','cyan','green']; kc=0 #line color
for dt in[0.1,0.5,1.]:
    tempo=np.arange(0.,60.,dt); n=len(tempo)
    X=np.zeros(tempo.shape); V=np.zeros(tempo.shape)
    X[0]=xinicial; V[0]=vinicial
    for k in range(1,n):
        V[k]=V[k-1]-kapa*X[k-1]*dt
        X[k]=X[k-1]+V[k]*dt
    plt.figure(2); plt.subplot(2,1,1)
    plt.plot(tempo,X,color=cores[kc],label=r'$ \Delta t=%6.3f$' %(dt))
    plt.xlabel('Time '); plt.ylabel(r'$X (m)$')
    EM=V**2/2+kapa*X**2/2
    plt.subplot(2,1,2)
    plt.plot(tempo,EM,color=cores[kc],label=r'$ \Delta t=%6.3f$' %(dt))
    plt.ylabel('Energy (J/kg)'); plt.xlabel('time (s)'); plt.legend()
    kc=kc+1
```

Energia varia
2% com $\Delta t = 0.1s$
20% com $\Delta t = 1s$

Comentários

O método de Euler é muito simples. Converge para a solução analítica no $\lim_{\Delta t \to 0}$. Mas é pouco preciso e pode não ser estável. É possível obter métodos melhores com um pequeno custo adicional.

Vamos desenvolver um método mais preciso no próximo exemplo.