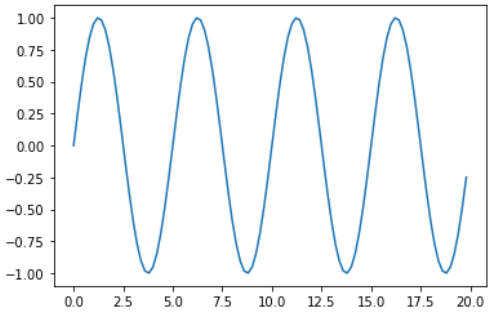
Ciências
ULisboa

Modelação Numérica 2022, Exame 2 (27 de Junho)

1. Considere o fragmento de código:

```
T=5
dt=0.2;N=100;t=np.arange(0,N*dt,dt);y=np.sin(2*np.pi*t/T)
Y=np.abs(np.fft.fft(y))
```

(a) Esquematize o gráfico `plt.plot(t,y)`



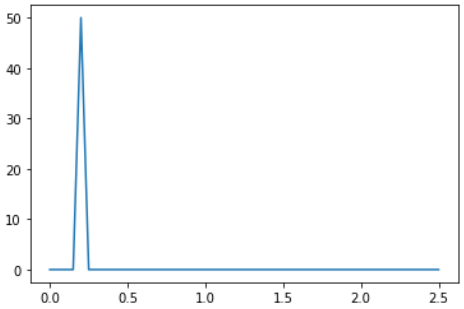
(b) O que representa `Y`?

`Y` é o espetro de amplitude da série `y`. Trata-se de um array float com 100 elementos.

(c) No gráfico `plt.plot(f,Y[0:N//2+1])` como calcularia `f`?

```
fNyq=1/(2*dt)
df=fNyq/(N//2)
f=np.arange(0,fNyq+df/2,df)
```

(d) Esquematize o gráfico (c), incluindo a escala do eixo dos xx.



2. A advecção de temperatura num fluido com velocidade constante pode representar-se pela equação:

$$\frac{\partial T}{\partial t} = -u\frac{\partial T}{\partial x} - v\frac{\partial T}{\partial y}$$

(a) Discretize a equação num método explícito utilizando aproximações de segunda ordem no espaço e no tempo.

$$\frac{T_{i,j}^{n+1} - T_{i,j}^{n-1}}{2\Delta t} = -u\frac{T_{i+1,j}^{n} - T_{i-1,j}^{n}}{2\Delta x} - v\frac{T_{i,j+1}^{n} - T_{i,j-1}^{n}}{2\Delta y}$$

(b) Escreva uma função python na forma Tnt=adv(T,x,y,u,v,dt,nt), onde T,x,y são arrays 2D e u,v são as componentes da velocidade, dt o passo de tempo, nt o número de passos de tempo de integração e Tnt a distribuição final de temperatura. A função deve (i) determinar nx,ny

(dimensões da malha espacial); (ii) inicializar as variáveis auxiliares; (iii) resolver com o método (a); (iv) usar condições fronteira cíclicas.

```
def adv(T,x,y,u,v,dt,nt)
    nx,ny=T.shape
    TM=np.copy(T)
    TP=np.copy(T)
    for it in range(nt):
        for ix in range(nx):
             ixp=ix+1
             ixm=ix-1
             if ixp>nx-1:
                 ixp=0
             elif ixm<0:
                 ixm=nx-1
            for iy in range(ny):
                    iyp=iy+1
                    iym=iy-1
                    if iyp>ny-1:
                        iyp=0
                    elif iym<0:
                        iym=ny-1
                    if it==1:
                        TP[ix,iy]=T[ix,iy]-u*dt/(2*dx)*(T[ixp,iy]-T[ixm,iy])\
                                  -v*dt/(2*dy)*(T[ix,iyp]-T[ix,iym])

                    else:
                        TP[ix,iy]=TM[ix,iy]-u*dt/dx*(T[ixp,iy]-T[ixm,iy])\
                                  -v*dt/dy*(T[ix,iyp]-T[ix,iym])
        TM=np.copy(T)
        T=np.copy(TP)

    return TP
```

(c) Como calcularia um valor razoável para dt, dados os restantes parâmetros? Justifique.
Deveria garantir que o número de Courant fosse inferior a 1, no sentido de produzir um método estável. C=u*dt/dx ou C=v*dt/dy

3. Escreva uma função de custo (em python) que poderia ser utilizada para calcular os coeficientes de um polinómio $y = ax^3 + bx^2 + cx + d$, dadas N observações $[x_i, y_i, i = 0, \dots N-1]$.

```
def cost(a,b,c,d,xi,yi):
    y=a*xi**3+b*xi**2+c*xi+d
    custo=np.mean((y-yi)**2)
    return custo
```