Nome	Nº	P	G

Exame 1

Identifique-se nas páginas 1 e 3.

1. As funções seguintes contêm (~3) erros fatais. Corrija-as. Admita que, em todos os casos se fez `import numpy as np`

(a) Cálculo de (x, y, z)

$$\begin{cases} x - y + 3z = 0 \\ 2x - y + 4z = 0 \\ 2x + 6y + 5 = 0 \end{cases}$$

```
M=np.array([[1,-1,3],[2,-1,4],[2,6,5]])
b=np.zeros((3))
MI=np.linalg.inv(M)
X=MI*b
X=X[0];y=X[1];z=X[2]
```

(b) Cálculo do traço de uma matriz

```
def traco(M):
    for k in range(M):
        T=T+M(k,k)
    return T
```

(c) Fórmula resolvente da eq. do 2º grau:

$$y = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
def y(a,b,c):
    y1=-b+sqrt(b^2-4*a*c)/2*a
    y2=-b-sqrt(b^2-4*a*c)/2*a
    return y
```

(d) Cálculo de $\int_a^b f(x)\,dx$, pelo método do trapézio

```
def intTRAP(f,a,b,n):
    dx=(b-a)/n
    x=np.linspace(a,b,n)
    integral=(f(a)+f(b))/2
    for k in range(n):
        integral=integral+f[k]
    integral=integral*dx
    return integral
```

(a) corrigido

```
M=np.array([[1,-1,3],[2,-1,4],[2,6,0]])
b=np.array([0,0,-5])
MI=np.linalg.inv(M)
X=np.matmul(MI,b)
X=X[0];y=X[1];z=X[2]
```

(b) corrigido

```
def traco(M):
    T=0.
    for k in range(len(M)):
        T=T+M[k,k]
    return T
```

(c) corrigido

```
def y(a,b,c):
    y1=(-b+sqrt(b**2-4*a*c))/2*a
    y2=(-b-sqrt(b**2-4*a*c))/2*a
    return y1,y2
```

(d) corrigido

```
def intTRAP(f,a,b,n):
    dx=(b-a)/n
    x=np.linspace(a,b,n+1)
    integral=(f(a)+f(b))/2
    for k in range(1,n+1):
        integral=integral+f(x[k])
    integral=integral*dx
    return integral
```

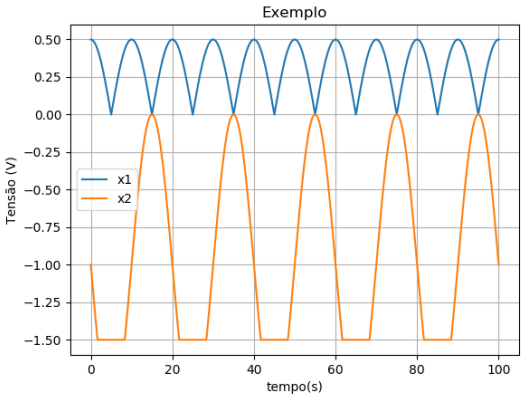
2. Dada uma série T de temperaturas horárias no período de n anos sucessivos, sem falhas, escreva uma função python **datas=fd(T)** que dados a série e a data inicial, calcule e devolva a série de datas correspondente em formato datetime. A função d=datetime.datetime(ano,mes,dia,hora) define um objeto datetime. A função i=datetime.timedelta(dias,segundos) define um intervalo de tempo, d2=d+i define um novo objeto datetime.

```
import datetime
import numpy as np
def datas=fd(T):
    data=datetime.datetime(ano,mês,dia,hora) #data inicial
    deltah=datetime.timedelta(0,3600)
    n=len(T)
    dataser=[data]
    for k in range(1,n):
        data=data+deltah
        dataser.append(data)
    return dataser
```

3. Escreva um script que produza a figura, incluindo anotações, com uma resolução no eixo das abcissas de 0.01 s (Nota: as partes curvas das linhas são sinusóides).

```
import numpy as np
import matplotlib.pyplot as plt
plt.close('all')

t=np.linspace(0,100,1001)
s=np.abs(0.5*np.cos(2.*np.pi/20*t))
plt.plot(t,s,label='x1')
s2=-np.sin(2*np.pi*t/20)-1
for k in range(len(s2)):
    s2[k]=max(s2[k],-1.5)
plt.plot(t,s2,label='x2')
plt.legend()
plt.xlabel('tempo(s)')
plt.ylabel('Tensão (V)')
plt.title('Exemplo')
plt.grid()
```



Nome	Nº	P	G

4. A função $f(x) = x^4 - e^{-x} - 300$ tem uma raiz na vizinhança de $x = 5$. Escreva um script python capaz de localizar essa raiz, usando o método da Newton, iterando até que até o método saltar entre interações menos de 10^{-6}, em não mais de 20 iterações. Lembre a recursão de Newton: $x^{n+1} = x^n - \frac{f(x^n)}{f'(x^n)}$.

```
import numpy as np
import matplotlib.pyplot as plt #opcional não pedido
def f(x):
    fun=x**4-np.exp(-x)-300
    return fun
def fprime(x):
    fp=4*x**3-np.exp(x)
    return fp
x=np.linspace(0,10,1001)
plt.plot(x,f(x)) #opcional não pedido
plt.grid()#opcional não pedido
maxerr=1e-6
xmin=0;xmax=5.
raiz=5
err=2*maxerr
nit=0
while err>maxerr and nit<100:
    nit=nit+1
    newraiz=raiz-f(raiz)/fprime(raiz)
    err=np.abs(raiz-newraiz)
    raiz=newraiz
plt.scatter(raiz,f(raiz))
```

5. A função `np.convolve(s,h,mode='same')` calcula a convolução entre a série `s` e a série `h`, devolvendo uma série com o mesmo número de pontos de s, e.g. $y_k = \sum_{n=0}^{N} s_n h_{k-n}$. Utilizando essa função, escreva uma função python a usar a forma `Tm24=m24(T)` que dada uma série temporal de valores horários da temperatura, devolva uma série de valores horários da média móvel de 24h.

```
def m24(T):
    h=np.ones(24)/24
    Tm24=np.convolve(T,h,'same')
    return Tm24
```

6. As equações

$$\frac{d\omega}{dt} = -\frac{g}{L}\sin\theta$$

$$\frac{d\theta}{dt} = \omega$$

regem o movimento de um pêndulo gravítico de comprimento L, com aceleração da gravidade g. Escreva uma função python a usar na forma t**heta,omega=pend(theta0,omega0,t,L,g),** que dados os parâmetros (**L,g),** a posição e a velocidade angular iniciais **(theta0,omega0)** e um vetor de tempos regularmente espaçados (**t**), calcule as séries temporais da posição (**theta**) e da velocidade angular (**omega**) ao longo desse tempo. Utilize o método de Euler, com uma iteração para se aproximar do método do ponto médio.

```
def pend(theta0,omega0,t,L,g):
    n=len(t)
    dt=t[1]-t[0]
    theta=np.zeros((n))
    omega=np.zeros((n))
    theta[0]=theta0
    omega[0]=omega0
    for k in range(1,n):
        omega[k]=omega[k-1]-g/L*np.sin(theta[k-1])
        omegaH=0.5*(omega[k]+omega[k-1])
        theta[k]=theta[k-1]+omegaH*dt
        thetaH=0.5*(theta[k]+theta[k-1])
        omega[k]=omega[k-1]-g/L*np.sin(thetaH)
        omegaH=0.5*(omega[k]+omega[k-1])
        theta[k]=theta[k-1]+omegaH*dt
    return theta,omega
```