

1 MODELLO ELASTICO - PERFETTAMENTE PLASTICO

```
import numpy as np
import sympy as sy
import matplotlib.pyplot as plt

E=1 # Modulo elastico [N/mm2]
A=E*np.sin(np.pi/4) # tensione di snervamento [N/mm2]
t=np.arange( 0.,2.5*np.pi,0.01) # range di tempo [s]
ek=A/E # epslon snervamento
pk=ek-A/E # p snervamento

e=np.sin(t) # funzione epslon armonica
#e=0.01*np.sin(t)*np.exp(-t/10) # funzione epslon armonica

sigma=np.zeros(len(t))
p=np.zeros(len(t))
f=np.zeros(len(t))
f[0]=-A

for i in range(1,(len(t))):
    p[i]=p[i-1]
    sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla
    f[i]=np.abs(sigma[i])-A #funzione di snervamento

    if f[i] < 0: # FASE ELASTICA
        p[i]=p[i-1]
        sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla

    elif f[i]>0: # SCORRIMENTO PLASTICO
        if sigma[i]>0 :
            p[i]=e[i]-A/E
            sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla
            f[i]=(sigma[i])-A #funzione di snervamento

        if sigma[i]<0 :
            p[i]=e[i]+A/E
            sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla
            f[i]=-(sigma[i])-A #funzione di snervamento

exec (open ("LegamipLOT.py") .read ())
plt.suptitle('Modello elasto-plastico perfetto  $\sigma^* = A = \sin(\pi/4)$ ', fontsize=20, color='darkred')

file = open('datimodelloEP.txt', 'w') # Apertura file in scrittura ('w')

for i in range(len(t)):
    # Preparazione stringa da scrivere
    # ho arrotondato con round
    st = str(round(sigma[i], 2)) + '\t' + str(round(e[i], 2)) + '\n'
    file.write(st) # Scrittura su file
file.close() # Chiusura file
```

2 MODELLO ELASTO PLASTICO INCRUDENTE

```
import numpy as np
import sympy as sy
import matplotlib.pyplot as plt

E=1 # Modulo elastico [N/mm2]
A=E*np.sin(np.pi/4) # tensione di snervamento [N/mm2]
t=np.arange( 0.,2.5*np.pi,0.001) # range di tempo [s]
eyk=A/E # epslon snervamento
H=E

e=np.sin(t) # funzione epslon armonica
# e=np.sin(t)*np.exp(-t/100) # funzione epslon armonica

sigma=np.zeros(len(t))
p=np.zeros(len(t))
f=np.zeros(len(t))
f[0]=-A

for i in range(1,(len(t))):
    p[i]=p[i-1]
    sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla
    f[i]=np.abs(sigma[i]-H*p[i])-A #funzione di snervamento

    if f[i] < 0: # FASE ELASTICA
        p[i]=p[i-1]
        sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla

    elif f[i]>0: # SCORRIMENTO PLASTICO
        if sigma[i]-H*p[i]>0 :
            p[i]=(e[i]*E-A)/(E+H)
            sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla
            f[i]=(sigma[i]-H*p[i])-A #funzione di snervamento

        if sigma[i]-H*p[i]<0 :
            p[i]=(e[i]*E+A)/(E+H)
            sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla
            f[i]=-(sigma[i]-H*p[i])-A #funzione di snervamento

exec (open ("LegamipLOT.py") .read ())
plt.suptitle('Modello elasto-plastico incrudente  $\sigma^* = A = \sin(\pi/4)$ ', fontsize=20, color='d
```

3 MODELLO DI TENTATIVO 1

```
import numpy as np
import sympy as sy
import matplotlib.pyplot as plt

E=1 # Modulo elastico [N/mm2]
A=E*np.sin(np.pi/4) # tensione di snervamento [N/mm2]
t=np.arange( 0.,10*np.pi,0.01) # range di tempo [s]
eyk=A/E # epslon snervamento
B=A

e=np.sin(t) # funzione epslon armonica
#e=np.sin(t)*np.exp(-t/20) # funzione epslon armonica

sigma=np.zeros(len(t))
p=np.zeros(len(t))
f=np.zeros(len(t))
f[0]=-A

for i in range(1,(len(t))):
    p[i]=p[i-1]
    sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla
    f[i]=np.abs(sigma[i])-(A+B*(p[i])) #funzione di snervamento

    if f[i] < 0: # FASE ELASTICA
        p[i]=p[i-1]
        sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla

    elif f[i]>0: # SCORRIMENTO PLASTICO
        if sigma[i]>0 :
            p[i]=(e[i]*E-A)/(E+B)
            sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla
            f[i]=sigma[i]-(A+B*p[i]) #funzione di snervamento

        if sigma[i]<0 :
            p[i]=(e[i]*E+A)/(E-B)
            sigma[i]=E*(e[i]-p[i]) # legame costitutivo molla
            f[i]=-sigma[i]-(A+B*(p[i])) #funzione di snervamento

exec (open ("LegamipLOT.py") .read ())
plt.suptitle('Modello elasto-plastico inelastico  $\sigma^*=(A+B \cdot p)$  ', fontsize=20, color='darkred')

file = open('datiscrittiprova3.txt', 'w') # Apertura file in scrittura ('w')

for i in range(len(t)):
    # Preparazione stringa da scrivere
    st = str(round(sigma[i], 2)) + '\t' + str(round(e[i], 2)) + '\n' # ho arrotondato con round
    file.write(st) # Scrittura su file
file.close() # Chiusura file
```

4 MODELLO DI TENTATIVO 2

```
import numpy as np
import sympy as sy
import matplotlib.pyplot as plt

E=1 # Modulo elastico [N/mm2]
A=E*np.sin(np.pi/4) # tensione di snervamento [N/mm2]
t=np.arange( 0.,2.5*np.pi,0.0001) # range di tempo [s]
eyk=A/E # epslon snervamento
B=E

e=np.sin(t) # funzione epslon armonica
# e=np.sin(t)*np.exp(t/20) # funzione epslon armonica

sigma=np.zeros(len(t))
p=np.zeros(len(t))
f=np.zeros(len(t))
f[0]=-A

def sigmaf(e,p):
    return E*(e-p)

for i in range(1,(len(t))):
    p[i]=p[i-1]
    sigma[i]=sigmaf(e[i],p[i]) # legame costitutivo molla
    f[i]=np.abs(sigmaf(e[i],p[i]))-(A+B*np.abs(p[i])) #funzione di snervamento

    if f[i] <= 0: # FASE ELASTICA
        p[i]=p[i-1]
        sigma[i]=sigmaf(e[i],p[i])

    elif f[i]>0 and sigma[i]>=0 : # SCORRIMENTO PLASTICO 1 #and e[i]>eyk

        p[i]=(e[i]*E-A)/(E+B)

        sigma[i]=sigmaf(e[i],p[i]) # legame costitutivo molla

        f[i]=sigma[i]-(A+B*(p[i])) #funzione di snervamento

    elif f[i]>0 and sigma[i]<=0 : # SCORRIMENTO PLASTICO 2 #e[i] < -eyk

        p[i]=(e[i]*E+A)/(E+B)

        sigma[i]=sigmaf(e[i],p[i]) # legame costitutivo molla

        f[i]=-(sigma[i])-(A+B*(-p[i])) #funzione di snervamento

exec (open ("LegamipLOT.py") .read ())
plt.suptitle('Modello elasto-plastico inelastico  $\sigma^*=(A+B\cdot|p|)$ ', fontsize=20, color='darkred')
```

```

file = open('datiscritti_prova4.txt', 'w') # Apertura file in scrittura ('w')

for i in range(len(t)):
    # Preparazione stringa da scrivere
    st = str(round(sigma[i], 2)) + '\t' + str(round(e[i], 2)) + '\n' # ho arrotondato con round
    file.write(st) # Scrittura su file
file.close() # Chiusura file

```

5 MODELLO ELASTO PLASTICO ISOTROPO INCRUDENTE

```
import numpy as np
import sympy as sy
import matplotlib.pyplot as plt

E=1 # Modulo elastico [N/mm2]
A=E*np.sin(np.pi/4) # tensione di snervamento [N/mm2]
t=np.arange( 0.,10*np.pi,0.01) # range di tempo [s]
ek=A/E # epslon snervamento
B=A/10 # e necessario dividere A altrimenti e come se la plasticit smorzasse

#e=np.sin(t) # funzione epslon armonica
e=np.sin(t)*np.exp(t/20) # funzione epslon armonica

sigma=np.zeros(len(t))
p=np.zeros(len(t))
K=np.zeros(len(t))
f=np.zeros(len(t))
f[0]=-A
P=np.zeros(len(t))

def sigmaf(e,p):
    return E*(e-p)

for i in range(1,(len(t))):

    p[i]=p[i-1]

    K[i]=np.abs(p[i]-p[i-1])

    P[i]=sum(K)

    sigma[i]=sigmaf(e[i],p[i]) # legame costitutivo molla

    f[i]=np.abs(sigmaf(e[i],p[i]))-(A+B*(P[i]+np.abs(p[i]-p[i-1])))) #funzione di snervamento

    if f[i] <= 0 : # FASE ELASTICA

        p[i]=p[i-1]

        K[i]=np.abs(p[i]-p[i-1])

        sigma[i]=sigmaf(e[i],p[i])

    elif f[i]>0 and sigma[i]>=0 : # SCORRIMENTO PLASTICO 1 #and e[i]>eyk

        p[i]=(e[i]*E-A-B*P[i]+B*p[i-1])/(E+B)

        K[i]=np.abs(p[i]-p[i-1])
```

```

sigma[i]=sigmaf(e[i],p[i]) # legame costitutivo molla

f[i]=sigma[i]-(A+B*(P[i]+(p[i]-p[i-1])))) #funzione di snervamento

elif f[i]>0 and sigma[i]<=0 : # SCORRIMENTO PLASTICO 2 #e[i] < -eyk

    p[i]=(e[i]*E+A+B*P[i]+B*p[i-1])/(E+B)
    #p[i]=(e[i]*E+A+B*P[i]-B*p[i-1])/(E-B)

    K[i]=np.abs(p[i]-p[i-1])

    sigma[i]=sigmaf(e[i],p[i]) # legame costitutivo molla

    f[i]=-sigma[i]-(A+B*(P[i]-(p[i]-p[i-1])))) #funzione di snervamento


exec (open ("LegamipLOT.py") .read ())
plt.suptitle('Modello elasto-plastico isotropo incrudente  $\sigma^*=(A+B\cdot \int{|p|})$ ',fontsi

#Plot K in funzione di t
axs[2, 0].plot(t,P,'-g',linewidth=1)


file = open('datiscritti_prova4.txt', 'w') # Apertura file in scrittura ('w')

for i in range(len(t)):
    # Preparazione stringa da scrivere
    st = str(round(sigma[i], 2)) + '\t' + str(round(e[i], 2)) + '\n' # ho arrotondato con round
    file.write(st) # Scrittura su file
file.close() # Chiusura file

```