Exercício 4

October 11, 2020

1 Exercício PA1-4

Exercício com data de entrega para 14 de outubro de 2020.

Aluno: Noé de Lima Bezerra

noe_lima@id.uff.br

[1]:
```
import numpy as np
import sympy as sp
import matplotlib.pyplot as plt
from IPython.display import display, Math, Image
sp.init_printing(use_latex='mathjax',latex_mode='equation*')
```

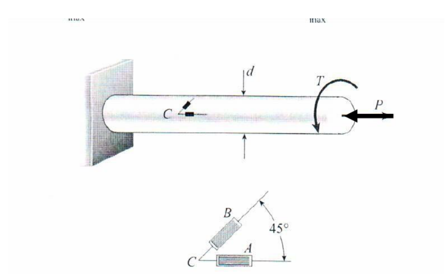
[2]:
```
Image("Figuras/PA1-4.png")
```

[2]:

AVALIAÇÃO 8 – EXERCÍCIO PA1-4

Uma barra circular sólida de diâmetro d=40mm está submetida a uma força axial P e a um torque T, conforme mostrado na figura. Os extensômetros A e B instalados na superfície da barra fornecem as leituras **ε_a = -100 μs e ε_b = -110 μs.** A barra é feita de aço com E=200 GPa e ν=0,29.

a) Determine a força axial P e o torque T;
b) Determine a deformação de cisalhamento máxima γ_{max} e a tensão de cisalhamento máxima τ_{max} na barra.



2 Solução

2.1 Círculo de Mohr

[3]:
```
def mohr(s_x,s_y,t_xy):
    if t_xy == 0:
        t_xy = 1e-10  # Evitar divisão por zero!
    z = np.linspace(0,360,360)
    r = np.sqrt((((s_x-s_y)/2)**2)+t_xy**2)
    s_med = ((s_x+s_y)/2)
    tg2tc = -(s_x-s_y)/(2*t_xy)
    tg2tp = -1/(tg2tc)
    tc = np.degrees(np.arctan(tg2tc)/2)
    tp = np.degrees(np.arctan(tg2tp)/2)
    x = s_med + r*np.cos(np.radians(z))
    y = r*np.sin(np.radians(z))
    a = ([s_x,s_x,s_y,s_y,s_x])
    b = ([0,t_xy,-t_xy,0,0])
    # Plot
    text = '\n'.join((
        r'$\sigma_{min} = %.1f~MPa$' % (s_med-r,),
        r'$\sigma_{max} = %.1f~MPa$' % (s_med+r,),
        r'$\tau_{max} = %.1f~MPa$' % (r,),
        r'$\theta_{p} = %.0f^{o}$' % (tp,),
        r'$\theta_{c} = %.0f^{o}$' % (tc,)
    ))
    props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)
    plt.plot(a,b,x,y)
    plt.xlabel(r"$\sigma$", size=18)
    plt.ylabel(r"$\tau$", size=18)
    plt.title("Círculo de Mohr para Tensões")
    plt.text(s_med-1.6*r,r,text, fontsize=12, verticalalignment='top',␣
 ↪bbox=props)
    plt.axis("equal")
    plt.show()
```

[4]:
```
def mohr_def(e_x,e_y,g_xy):
    if g_xy == 0:
        g_xy = 1e-10  # Evitar divisão por zero!
    z = np.linspace(0,360,360)
    r = np.sqrt((((e_x-e_y)/2)**2)+g_xy**2)
    e_med = ((e_x+e_y)/2)
    tg2tc = -(e_x-e_y)/(2*g_xy)
    tg2tp = -1/(tg2tc)
    tc = np.degrees(np.arctan(tg2tc)/2)
    tp = np.degrees(np.arctan(tg2tp)/2)
    x = e_med + r*np.cos(np.radians(z))
    y = r*np.sin(np.radians(z))
```

```
    a = ([e_x,e_x,e_y,e_y,e_x])
    b = ([0,g_xy,-g_xy,0,0])
    # Plot
    text = '\n'.join((
        r'$\epsilon_{min} = %.1f~\mu s$' % (e_med-r,),
        r'$\epsilon_{max} = %.1f~\mu s$' % (e_med+r,),
        r'$\frac{\gamma_{max}}{2} = %.1f~\mu s$' % (r,),
        r'$\theta_{p} = %.0f^{o}$' % (tp,),
        r'$\theta_{c} = %.0f^{o}$' % (tc,)
    ))
    props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)
    plt.plot(a,b,x,y)
    plt.xlabel(r"$\epsilon$", size=18)
    plt.ylabel(r"$\frac{\gamma}{2}$", size=18)
    plt.title("Círculo de Mohr para Deformações")
    plt.text(e_med-1.6*r,r,text, fontsize=12, verticalalignment='top',␣
↪bbox=props)
    plt.axis("equal")
    plt.show()
```

2.2 Deformações

Na seção que contém o ponto C, temos as seguintes deformações:

$$\begin{cases} \epsilon_a & = & -100 \ \ \mu s \\ \epsilon_b & = & -110 \ \ \mu s \end{cases}$$

[5]:
```
mm,N,MPa = sp.symbols("mm,N,MPa")
kN = 1000*N
m = 1000*mm
GPa = 1e3*MPa
```

[6]:
```
d = 40*mm
A = np.pi*(d**2)/4
J = np.pi*(d**4)/32
E = 200*GPa
nu = 0.29
G = E/(2*(1+nu))
```

2.3 Equação de Deformação

Temos a seguinte função que relaciona a deformação ao ângulo em relação aos eixos locais #xy#:

$$\epsilon\left(\theta\right) = \frac{\epsilon_x + \epsilon_y}{2} + \frac{\epsilon_x - \epsilon_y}{2}\cos\left(2\theta\right) + \gamma_{xy}\sin\left(2\theta\right)$$

[7]:
```
epsilon,theta,epsilon_x,epsilon_y,gamma_xy = sp.
↪symbols('epsilon,theta,epsilon_x,epsilon_y,gamma_xy')
```

[8]:
```
eqext = sp.Eq(epsilon,((epsilon_x+epsilon_y)/2) + ((epsilon_x-epsilon_y)/2)*sp.
↪cos(2*theta) + (gamma_xy/2)*sp.sin(2*theta))
display(eqext)
```

$$\epsilon = \frac{\epsilon_x}{2} + \frac{\epsilon_y}{2} + \frac{\gamma_{xy}\sin(2\theta)}{2} + \left(\frac{\epsilon_x}{2} - \frac{\epsilon_y}{2}\right)\cos(2\theta)$$

2.4 a)

[9]:
```
display(eqext.subs(epsilon,-100e-6).subs(theta,0))
display(eqext.subs(epsilon,-110e-6).subs(theta,sp.pi/4))
display(eqext.subs(epsilon,(-100e-6)*(-nu)).subs(theta,sp.pi/2))
```

$$-0.0001 = \epsilon_x$$

$$-0.00011 = \frac{\epsilon_x}{2} + \frac{\epsilon_y}{2} + \frac{\gamma_{xy}}{2}$$

$$2.9 \cdot 10^{-5} = \epsilon_y$$

[10]:
```
eqgamma = eqext.subs(epsilon,-110e-6).subs(theta,sp.pi/4).subs(epsilon_x,-1e-4).
↪subs(epsilon_y,2.9e-5)
display(eqgamma)
2*(-0.00011+3.55e-5)
```

$$-0.00011 = \frac{\gamma_{xy}}{2} - 3.55 \cdot 10^{-5}$$

[10]:

$$-0.000149$$

Assim, temos:

$$\begin{cases} \epsilon_x & = & -100 \;\; \mu s \\ \epsilon_y & = & 29 \;\; \mu s \\ \gamma_{xy} & = & -149 \;\; \mu s \end{cases}$$

[11]:
```
epsilon_x_c = -100e-6
epsilon_y_c = 29e-6
gamma_xy_c = -149e-6
```

[12]:
```
sigma_x = E*epsilon_x_c
sigma_y = 0
tau_xy = G*gamma_xy_c
display(sigma_x,sigma_y,tau_xy)
```

$$-20.0MPa$$

$$0$$

$$-11.5503875968992MPa$$

[13]:
```
P = A*sigma_x
display(P)
T = (J/(d/2))*tau_xy
display(T)
```

$$-25132.7412287183MPamm^2$$

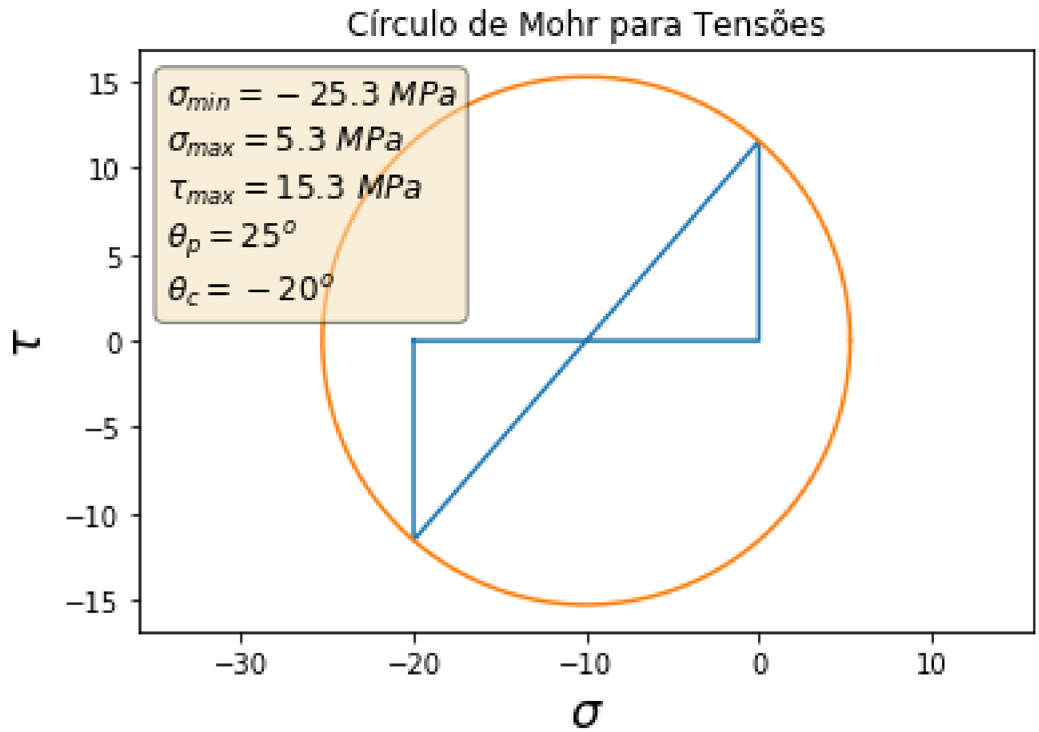
$$-145146.451282133MPamm^3$$

Portanto,

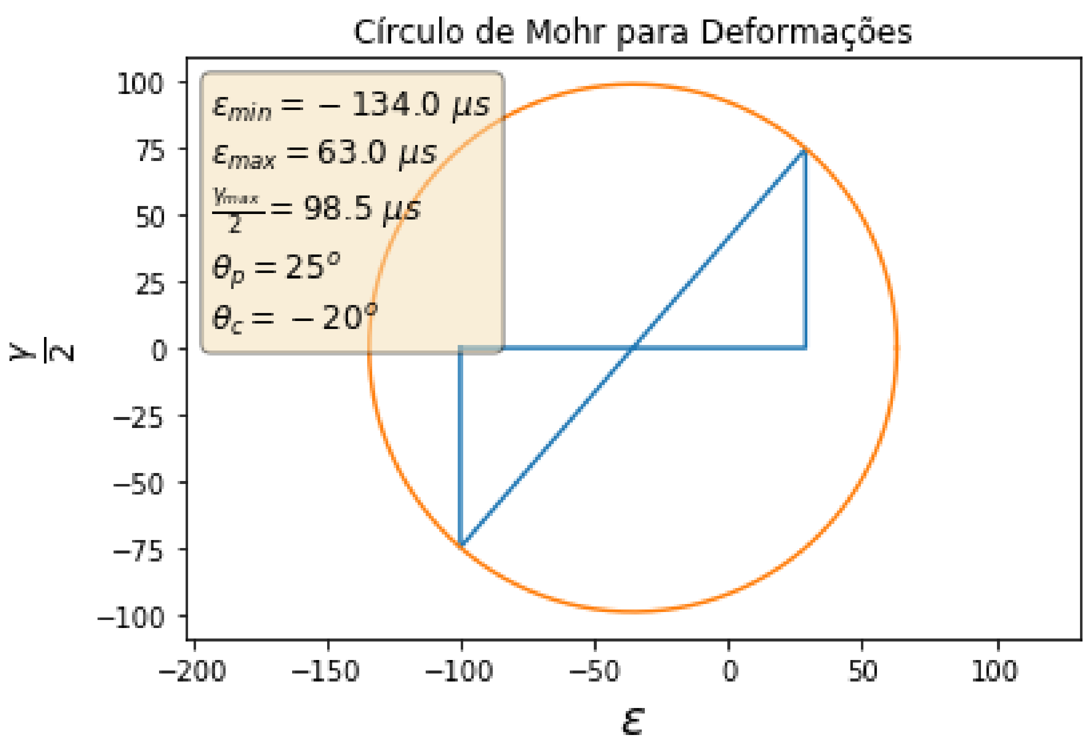
$$\begin{cases} P &= & -25,13 \ \ kN \\ T &= & -145,15 \ \ kN \cdot mm \end{cases}$$

2.5 b)

[14]:
```
mohr(float(sigma_x.subs(MPa,1)),sigma_y,float(tau_xy.subs(MPa,1)))
```



[15]:
```
mohr_def(-100,29,-149/2)
```



[16]:
```
(15.3*MPa)/G
```

[16]:

$$0.00019737$$

Assim, pelo círculo de Mohr, temos $\sigma_{min} = -25,3\ MPa < 0$ e $\sigma_{max} = 5,3\ MPa > 0$ e, portanto, o τ_{max} (e, consequentemente, o γ_{max}) ocorre neste plano.

Temos, portanto:

$$\begin{cases} \gamma_{xy_{max}} & = & 197\ \ \mu s \\ \tau_{xy_{max}} & = & 15,3\ \ MPa \end{cases}$$