Homeworks for SM-I course

Important:

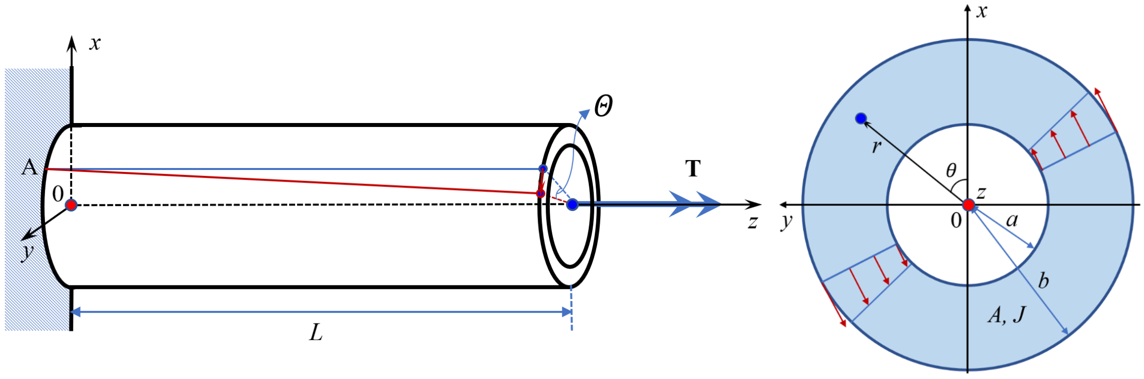
1). Please download the latest grcodes, images, and related chapters before working on the homework.

2). Both pdf files and the source codes must be submitted, or the work will not be marked.

Homework 7:

Question: Shafts with hollow cross-section subject to a torque

Consider a uniform bar made of Grade-304 stainless steel with shear modulus $G = 85$ GPa. The cross-section is hollow circular with inner radius of $a = 48$ mm, outer radius of $b = 50$ mm. The length of the bar is $L = 0.7m$. It is fixed at $z = 0$, and is subjected to a torque of $T = 900$ Nm at $z = L$, as shown below.



Stress on a hollow cylindrical cross-section of a uniform bar.

Using the exact formulas calculate:

1. The maximum shear stress on the cross-section in the shaft.
2. The twist angle at $x = L$.
3. The stress components on the cross-section and by the lateral surface in the Cartesian coordinates on the x-axis.
4. The stress components on the cross-section and by the lateral surface in the Cartesian coordinates at 30° from the x-axis.
5. The stress components on the cross-section and by the lateral surface in the Cartesian coordinates on the y-axis.
6. Compute the principle stresses in the shaft.
7. Compute the maximum shear stress in the shaft.
8. Assume the wall of the shaft is thin, using the formulas for thin-wall cross-section, repeat items 1 and 2. Compare the results obtained with those obtained using the exact formulas.
9. Using the same amount of material, but create a square cross-section with the same thickness. Repeat items 1 and 2. Compare the results obtained with those obtained in item 8.

In [ ]:
```python
# Place curse in this cell, and press Ctrl+Enter to import dependences.
import sys                              # for accessing the computer system
sys.path.append('../grbin/')  # Change to the directory in your system

from commonImports import *       # Import dependences from '../grbin/'
import grcodes as gr                 # Import the module of the author
#importlib.reload(gr)             # When grcodes is modified, reload it

from continuum_mechanics import vector
init_printing(use_unicode=True)       # For latex-like quality printing
np.set_printoptions(precision=4,suppress=True,
       formatter={'float': '{:0.4e}'.format})   # Digits in print-outs
```

1.

In [ ]:
```python
T = 900 # Nm, torque
a = 0.048; b = 0.05 # m, radii
J = np.pi * (b**4 - a**4)/2

print(f"The polar second moment of area J = {J:.4e} (m^4)")

maxshear = T*b/J

print(f"The maximum shear stress on the cross-section = {maxshear*1.0e-6:.4e} (MPa)
```

```
The polar second moment of area J = 1.4790e-06 (m^4)
The maximum shear stress on the cross-section = 3.0425e+01 (MPa)
```

2.

In [ ]:
```python
G = 85e9 # Pa
L = .7 # m
theta_t = T/(G*J)

print(f"Twist angle at x = L = {(theta_t*L):.4e} rad or {np.degrees(theta_t*L):.4e}
```

```
Twist angle at x = L = 5.0112e-03 rad or 2.8712e-01°
```

3.

In [ ]:
```python
theta3 = 0.0 # On the x-axis, θ=0
x = b*np.cos(np.deg2rad(theta3))
y = b*np.sin(np.deg2rad(theta3))
szx3 = -theta_t*G*y
szy3 = theta_t*G*x

print(f"On x-axis: σzx={szx3*1.0e-6:.4e}; σzy={szy3*1.0e-6:.4e} (MPa)")
```

```
On x-axis: σzx=-0.0000e+00; σzy=3.0425e+01 (MPa)
```

4.

In [ ]:
```
theta4 = 30
rad = theta4*np.pi/180.
x = b*np.cos(np.deg2rad(theta4))
y = b*np.sin(np.deg2rad(theta4))
szx4 =-theta_t*G*y
szy4 = theta_t*G*x
print(f"at 30°: σzx = {szx4*1.0e-6:.4e}; σzy = {szy4*1.0e-6:.4e} (MPa)")
```

```
at 30°: σzx = -1.5213e+01; σzy = 2.6349e+01 (MPa)
```

5.

In [ ]:
```
theta5 = 90.0 # On the y-axis, θ=90
x = b*np.cos(np.deg2rad(theta5))
y = b*np.sin(np.deg2rad(theta5))
szx5 = -theta_t*G*y
szy5 = theta_t*G*x

print(f"On y-axis: σzx={szx5*1.0e-6:.4e}; σzy={szy5*1.0e-6:.4e} (MPa)")
```

```
On y-axis: σzx=-3.0425e+01; σzy=1.8630e-15 (MPa)
```

6.

In [ ]:
```
def principalS(S):
    '''Compute the principal stresses and their directions.
    inputs:
       S: given stress tensor, numpy array
    return:
       principal stresses (eigenValues), their direction cosines
       (eigenVectors) ranked by its values. Right-hand-rule is enforced
    '''
    eigenValues, eigenVectors = lg.eig(S)

    #Sort in order
    idx = eigenValues.argsort()[::-1]
    eigenValues = eigenValues[idx]
    eigenVectors = eigenVectors[:,idx]
    print('Pricipal stress (Eigenvalues):\n',eigenValues,'\n')

    # make the first element in the first vector positive (optional):
    #eigenVectors[0,:] = eigenVectors[0,:]/np.sign(eigenVectors[0,0])

    # Determine the sign for given eigenVector-1 and eigenVector-3
    eigenVectors[:,2] = np.cross(eigenVectors[:,0], eigenVectors[:,1])

    angle = np.arccos(eigenVectors[0,0])*180/np.pi       # in degree
    print(f'Principal stress directions:\n{eigenVectors}\n')
    print(f"Possible angles (n1,x)={angle}∘ or {180-angle}∘")

    return eigenValues, eigenVectors
```

In [ ]:

```
theta6 = 90.0 # use the stress components on suface-y
x = b*np.cos(np.deg2rad(theta6)); y = b*np.sin(np.deg2rad(theta6))
szx6 = -theta_t*G*y
szy6 = theta_t*G*x
print(f"at 90°: σzx = {szx6:.4e}; σzy = {szy6:.4e} (Pa) \n")
s_np = np.array([[0,0,szx6],[0,0,szy6],[szx6,szy6,0]], dtype = float)
eigs, eigvs = principalS(s_np)
print(f"\n τ_max = {(eigs[0]-eigs[-1])/2:.4e}(Pa)")
```

```
at 90°: σzx = -3.0425e+07; σzy = 1.8630e-09 (Pa)

Pricipal stress (Eigenvalues):
 [3.0425e+07 5.2380e-74 -3.0425e+07]

Principal stress directions:
[[7.0711e-01 6.1232e-17 7.0711e-01]
 [-4.3298e-17 1.0000e+00 -4.3298e-17]
 [-7.0711e-01 7.0373e-50 7.0711e-01]]

Possible angles (n1,x)=45.00000000000001∘ or 135.0∘

 τ_max = 3.0425e+07(Pa)
```

7.

In [ ]:

```
T7, _ = gr.transferM(45., about = 'y') # rotation axis, angle
T7@s_np@T7.T
```

Out[ ]:

```
array([[3.0425e+07, -1.3173e-09, 1.7993e-10],
       [-1.3173e-09, 0.0000e+00, 1.3173e-09],
       [-1.7993e-10, 1.3173e-09, -3.0425e+07]])
```

8.

In [ ]:

```
t = b-a
R = (b-a)/2
J = 2*np.pi*(R**3)*t

print(f"The polar second moment of area J = {J:.4e} (m^4)")

maxshear = T*R/J

print(f"The maximum shear stress on the cross-section = {maxshear*1.0e-6:.4e} (MPa)
```

```
The polar second moment of area J = 1.2566e-11 (m^4)
The maximum shear stress on the cross-section = 7.1620e+04 (MPa)
```

In [ ]:

```
G = 85e9 # Pa
L = .7 # m
theta_t = T/(G*J)

print(f"Twist angle at x = L = {(theta_t*L):.4e} rad or {np.degrees(theta_t*L):.4e}
```

```
Twist angle at x = L = 5.8981e+02 rad or 3.3794e+04°
```

9.

In [ ]:
```
J = 4*(R**4)/((4*R)/t)

print(f"The polar second moment of area J = {J:.4e} (m^4)")

maxshear = T*R/J

print(f"The maximum shear stress on the cross-section = {maxshear*1.0e-6:.4e} (MPa)
```

```
The polar second moment of area J = 2.0000e-12 (m^4)
The maximum shear stress on the cross-section = 4.5000e+05 (MPa)
```

In [ ]:
```
G = 85e9 # Pa
L = .7 # m
theta_t = T/(G*J)

print(f"Twist angle at x = L = {(theta_t*L):.4e} rad or {np.degrees(theta_t*L):.4e}
```

```
Twist angle at x = L = 3.7059e+03 rad or 2.1233e+05°
```