Task2: Python – Program to plot the Mohr's circle of stresses

Write a Python program to compute and plot the Mohr's circle of stresses. Plot the Mohr's circle for the stress states of individual material points and the average of all material points.

Theory: Mohr's circle of stresses is a two-dimensional graphical representation of the stress tensor and is named after the German engineer *Christian Otto Mohr*. Any point on the circle denotes the stress state on a rotated coordinate system. The Mohr's circle is plotted with the normal stresses along the abscissa and the shear stresses along the ordinate (see fig. 1 below).

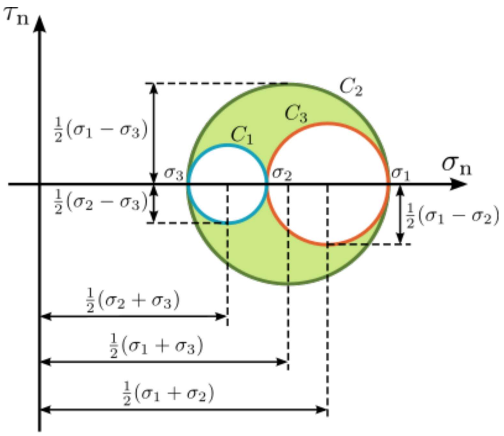

τ_n
C_2
C_3
C_1
$\frac{1}{2}(\sigma_1-\sigma_3)$
σ_3
σ_2
σ_1
σ_n
$\frac{1}{2}(\sigma_2-\sigma_3)$
$\frac{1}{2}(\sigma_1-\sigma_2)$
$\frac{1}{2}(\sigma_2+\sigma_3)$
$\frac{1}{2}(\sigma_1+\sigma_3)$
$\frac{1}{2}(\sigma_1+\sigma_2)$

Fig. 1: Schematic representation of the Mohr's circle. Source: Wikipedia.

The centers and radii of the three circles are given by the equations:

$$C1 = \frac{1}{2}(\sigma_2 + \sigma_3)\,; \quad R1 = \frac{1}{2}(\sigma_2 - \sigma_3)$$

$$C2 = \frac{1}{2}(\sigma_1 + \sigma_3)\,; \quad R2 = \frac{1}{2}(\sigma_1 - \sigma_3)$$

$$C3 = \frac{1}{2}(\sigma_1 + \sigma_2)\,; \quad R3 = \frac{1}{2}(\sigma_1 - \sigma_2)$$

where $\sigma_1, \sigma_2, \sigma_3$ are the three eigenvalues of the stress tensor.

Details of the dataset:

The dataset contains xx number of files. Each file provides the stress state (as the complete tensor, in MPa) and the volume of a material point. The data in the file is organized as follows:

```
# Comment line containing the volume of the material point
Stress_xx   Stress_xy   Stress_xz
Stress_xy   Stress_yy   Stress_yz
Stress_xz   Stress_yz   Stress_zz
```

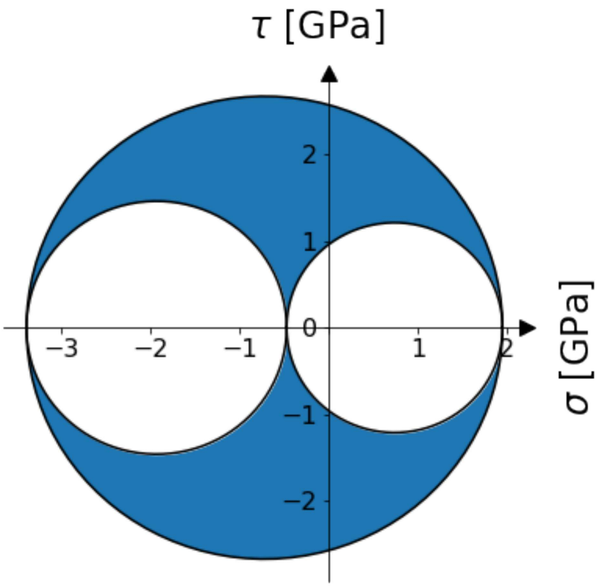
Your task is to write a Python program that plots the Mohr's circle for any given stress state. To demonstrate the working of your program, you will then generate the Mohr's circle for all the ten stress states provided to you. Furthermore, you will generate the Mohr's circle for the stress tensor obtained as the average of all the ten material points; the average being performed both with and without accounting for the volume of the material point

__The Python program must include the following functions:__

- Function `readData(fname)`: To read in data from a single file
  `fname` → Name of the file from which data is to be read
  This function should return the stress tensor as well as the volume of the material point (stored in the first line)
- Function `AvgStress(n,SigArr,VolumeArr)`: To average stresses
  `n` → Number of points to average. NOTE: Although `n` is 10 in the current scenario (`n` is the number of material points), the function must be general. Hence use the value provided to the function and do not hard code `n`!
  `SigArr` → An array of stress tensors that are to be averaged
  `VolumeArr` → An array of material point volumes of the corresponding stress tensors
  This function should perform a volumetric average of *n* stress tensors as follows:

  $$\bar{\sigma} = \frac{\sum_{i=1}^{n} \sigma_i V_i}{\sum_{i_1}^{n} V_i}$$

  where σ_i and V_i denote the stress and volume, respectively, of the ith material point.
- Function `EigVal(Sigma)`: To compute eigenvalues of a given stress tensor, `Sigma`. The function must return the sorted eigenvalues. You may use available functionality in numpy to calculate the eigenvalues.
- Func `plotMohrsCircle(SigmaEigval, foutNamePNG)`: To plot the Mohr's circle(s) of stresses for the given eigenValues.
  `SigmaEigval` → Eigenvalues of the stress tensor
  `foutNamePNG` → filename for the output PNG file
  In this function, you should calculate the center and radius of all three circles. Plot the Mohr's circle(s) and shade the different regions as shown in the figure below. Note that the circumference of the circle must be clearly demarcated.



- Annotate the plot appropriately, i.e., ensure proper axes labels (with units) and ticks with the right fontsize.
- Save the plot as a PNG file. An example pattern of the PNG filename can be: `MohrCirc_<PrefixStringOfDataset>.PNG`, where `<PrefixStringOfDataset>` is the file name of the corresponding dataset without the extension.

  Example: `Filename -Gr122_MatPt10.dat`
  `<PrefixStringOfDataset> - Gr122_MatPt10;`
  `PNG filename – MohrCirc_Gr122_MatPt10.PNG`
- For the average stress tensor, use only the grain number appended with `_VolAvg`.

  Example: `Filename - Gr122_MatPt10.dat;`
  `PNG filename – MohrCirc_Gr122_VolAvg.PNG`

NOTE: You may choose the name of the PNG file. It is important that all required information is present in the filename, i.e., Grain number, material point number or the averaging procedure.

<u>The main program must deal with the following:</u>

- Obtain the prefix string of the filename(s) containing the dataset, and the number of material points as user input.
  Hint: You may use `sys.argv` for this purpose.
- Read the stress tensors and volumes of all material points using the function `readData`. How you choose to store this dataset is left to your discretion. You may store it as an array (i.e. array of arrays) or as a dictionary or simply as a list.
- Obtain the average stress tensor. Here, perform averaging with and without the material point volume, i.e. in the latter case choose an identical volume for all material points. Ensure that your function `AvgStress` works not only for the entire array of stress tensors, but also for a subset of it (choose a value of n which is lesser than 10!).
- Obtain the sorted eigenvalues of individual stress tensors and the average stress tensor(s) by calling the function `EigVal`.
- Plot the Mohr's circle for individual stress tensors and the average stress tensor(s) by calling the function `plotMohrsCircle`. Make sure that you are passing the right output filenames for each case!

<u>Some hints:</u>

- To plot the circle, there are two possible methods:
  - `plt.Circle(center, radius)`
  - `plt.plot (xx,yy)`

  The former requires you to add a patch via `ax.add_patch`. The latter is a simple plot that we have also done in the lecture/exercise, with `xx`, `yy` the arrays of `x` and `y` coordinates of points on the circumference on the circle (Divide the circle into segments and calculate the points on the circumference).
- Scale the stress values from MPa to GPa. This will make your axis ticks easier to read!

<u>Execution:</u>

The Python program should be executable as follows:

```
$> python MohrCirc.py <FilePrefixStr> <MatPtStr> <FileExtension> <NumMaterialPoints>
```

An example of a specific command could be:

```
$> python MohrCirc.py Gr1 MatPt dat 10
```

Further points to keep in mind:

- Ensure you are using the right environment by activating your virtual environment.
- You may choose to program your tasks in a Jupyter notebook or in the ipython interpreter. However, your final submission must by a python script that is directly executable from the command prompt.
- Whilst programming, make sure that your code is well commented. Put in appropriate comments wherever necessary, particularly where you need to explain the logic of your program. Avoid trivial comments like: “Here we add two numbers”.
- Remember the basic principle of programming – You program a certain logic so as to use it repeatedly. To this end, your program must be generic and usable with other datasets also. Adhering to the given design will help you in this regard.