



The geometric description for FE models

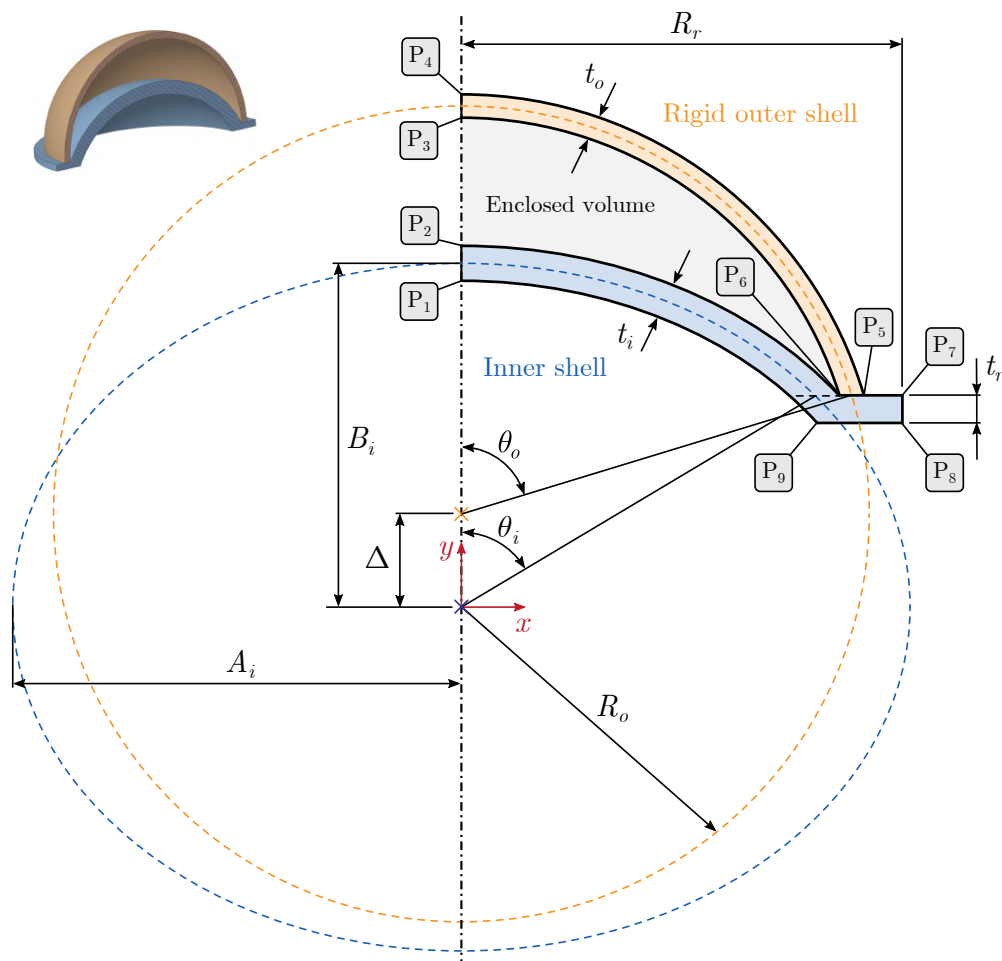


Figure A.1: The geometry and its parameters

The coordinates of each point can be easily obtained in the y direction using the opening angles and the radii, but in the x direction we need to calculate the intersection points of primitive geometric shapes, such as circles, ellipses, and lines. The x coordinate of the first four points is zero because these are laying on the axis of rotation. Their y coordinate can be easily obtained using the parameters B_i , R_o , and the shells' thickness. The first intersection point which needs to be calculated is the x coordinate of the point P_5 , which can be constructed using the intersection (is. in further readings) of a circle and a line, such as:

$$x^2 + (y - \Delta)^2 = \left(R_o + \frac{t_o}{2}\right)^2 \quad \text{circle and the } y = \Delta + R_o \cos \theta_o \quad \text{line is.} \quad (\text{A.1})$$

from which the x coordinate is expressed by substituting the second equation into the first:

$$x_5 = \sqrt{\left(R_o + \frac{t_o}{2}\right)^2 - (R_o \cos \theta_o)^2}. \quad (\text{A.2})$$

Point P_6 can be written in terms of the inner and the outer cap's parameters as well. First, the outer cap's circle and The point P_6 can be described by the geometric parameters of both the inner and outer shells. First, considering the outer shell, we look for the intersection of a circle and a line in the form:

$$x^2 + (y - \Delta)^2 = \left(R_o - \frac{t_o}{2}\right)^2 \quad \text{circle and the } y = \Delta + R_o \cos \theta_o \quad \text{line is.} \quad (\text{A.3})$$

from which the x coordinate:

$$x_6^o = \sqrt{\left(R_o - \frac{t_o}{2}\right)^2 - (R_o \cos \theta_o)^2}. \quad (\text{A.4})$$

The same point with respect to the inner cap's parameters as:

$$\frac{x^2}{\left(A_i + \frac{t_i}{2}\right)^2} + \frac{y^2}{\left(B_i + \frac{t_i}{2}\right)^2} = 1 \quad \text{ellipse and the } y = r_e \cos \theta_i \quad \text{line is.} \quad (\text{A.5})$$

from which the x coordinate can be obtained as:

$$x_6^i = \sqrt{\left(1 - \frac{r_e^2 \cos^2 \theta_i}{\left(B_i + \frac{t_i}{2}\right)^2}\right) \cdot \left(A_i + \frac{t_i}{2}\right)^2}, \quad (\text{A.6})$$

where r_e is the length of the line drawn from the origin to an arbitrary point on the ellipse defined by the opening angle θ_i . The formula can be obtained from the polar coded description of the ellipse, given as

$$r_e = \frac{A_i B_i}{\sqrt{A_i^2 \sin^2 \left(\frac{\pi}{2} - \theta_i\right) + B_i^2 \cos^2 \left(\frac{\pi}{2} - \theta_i\right)}}, \quad (\text{A.7})$$

which can be interpreted using the figure below:

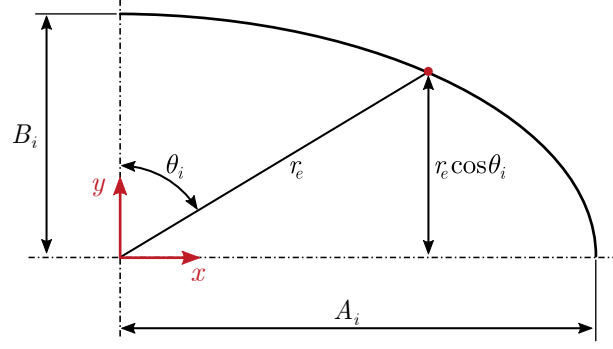


Figure A.2: Interpreting the length of the vector pointing to a given point on the ellipse

The coordinates of points P_7 and P_8 can be easily calculated using the radius and thickness of the ring, respectively, thus they are not discussed in detail. Finally, the point P_9 is defined by

$$\frac{x^2}{\left(A_i - \frac{t_i}{2}\right)^2} + \frac{y^2}{\left(B_i - \frac{t_i}{2}\right)^2} = 1 \quad \text{ellipse and the } y = r_e \cos \theta_i - t_r \quad \text{line is.} \quad (\text{A.8})$$

from which the x coordinate:

$$x_9 = \sqrt{\left(1 - \frac{(r_e \cos \theta_i - t_r)^2}{\left(B_i - \frac{t_i}{2}\right)^2}\right) \cdot \left(A_i - \frac{t_i}{2}\right)^2}. \quad (\text{A.9})$$

Using the notations shown in Fig. A.1 and the derivations, the coordinates of each point are summarized below:

$$P_1 = \left(0, B_i - \frac{t_i}{2}\right), \quad (\text{A.10})$$

$$P_2 = \left(0, B_i + \frac{t_i}{2}\right), \quad (\text{A.11})$$

$$P_3 = \left(0, \Delta + B_i - \frac{t_o}{2}\right), \quad (\text{A.12})$$

$$P_4 = \left(0, \Delta + B_i + \frac{t_o}{2}\right), \quad (\text{A.13})$$

$$P_5 = \left(\sqrt{\left(R_o + \frac{t_o}{2}\right)^2 - (R_o \cos \theta_o)^2}, \Delta + R_o \cos \theta_o\right), \quad (\text{A.14})$$

$$P_6^i = \left(\sqrt{\left(1 - \frac{r_e^2 \cos^2 \theta_i}{\left(B_i + \frac{t_i}{2}\right)^2}\right) \cdot \left(A_i + \frac{t_i}{2}\right)^2}, r_e \cos \theta_i\right), \quad (\text{A.15})$$

$$(\text{A.16})$$

$$P_6^o = \left(\sqrt{\left(R_o - \frac{t_o}{2}\right)^2 - (R_o \cos \theta_o)^2}, \Delta + R_o \cos \theta_o \right), \quad (\text{A.17})$$

$$P_7 = (R_r, r_e \cos \theta_i), \quad (\text{A.18})$$

$$P_8 = (R_r, r_e \cos \theta_i - t_r), \quad (\text{A.19})$$

$$P_9 = \left(\sqrt{\left(1 - \frac{(r_e \cos \theta_i - t_r)^2}{\left(B_i - \frac{t_i}{2}\right)^2}\right) \cdot \left(A_i - \frac{t_i}{2}\right)^2}, r_e \cos \theta_i - t_r \right). \quad (\text{A.20})$$

However, an enclosed volume is formed only if the outer and inner shells are in contact at point P_6 , so the following expressions must hold:

$$x_6^i = x_6^o \quad \text{and} \quad y_6^i = y_6^o. \quad (\text{A.21})$$

By comparing the y coordinates of (A.15) and (A.16), the distance between the center of the ellipse and the center of the circle can be expressed as

$$\Delta = r_e \cos(\theta_i) - R_o \cos(\theta_o). \quad (\text{A.22})$$

Similarly, by comparing the x coordinates, we can conclude that the radius of the outer shell can be expressed by the following quadratic equation:

$$R_o = \frac{-B \pm \sqrt{(B^2 - 4AC)}}{2A}, \quad (\text{A.23})$$

where the coefficients are given by

$$A = 1 - \cos^2 \theta_o, \quad (\text{A.24})$$

$$B = -t_o, \quad (\text{A.25})$$

$$C = \frac{t_o^2}{4} - (x_6^i)^2. \quad (\text{A.26})$$

The conclusion is that the distance between the centers of the two shapes and the radius of the outer shell are not independent parameters, they can be described using a function in terms of the other parameters. It is important to note that Eq. (A.23) has two solutions in general. The larger of the two should always be chosen, to ensure that the geometry and the following calculations are correct. To ensure that the inflatable cavity is created, we can use the constraint equations

$$y_2 < y_3 \quad \text{and} \quad \theta_i < \theta_o \quad (\text{A.27})$$

which needs to be satisfied for each combination of the parameters. The coordinates of the points defining the geometry can be easily calculated using a PYTHON script, which first calculates the radius according to Eq. (A.23), which is then used to calculate the center-to-center distance expressed in Eq. (A.22). The program then computes all coordinates, taking into account the constraint equations introduced in Eq. (A.27). Of course, the ring's radius (R_r) and thickness (t_r), the outer shell's opening angle (θ_o) and thickness (t_o) must be defined too, but for the sake of simplicity, I do not treat them as input parameters, instead, I calculate them by creating relations to the rest of the geometry (because their values are not so determinant). The opening angle is chosen to be $\theta_o = 90^\circ$ for all of the simulations. The geometric relations can be expressed such as the radius of the ring must be bigger than the coordinate x_5 , thus

$$R_r = x_5 + 0,05 \cdot x_5 = 1,05 \cdot x_5. \quad (\text{A.28})$$

The thicknesses are defined by the following expression:

$$t_r = t_o = \frac{t_i}{2}. \quad (\text{A.29})$$

Using these assumptions, the axisymmetric geometry is defined using 4 input parameters, which are used to describe the inner elliptical shell in its entirety. These parameters are the semi-axes (A_i and B_i), the opening angle (θ_i) and the shell's thickness (t_i).

B

Finite element modelling

B.1 Mesh independence study

A mesh independence study was carried out to ensure that the discretization is fine enough. The used model does not differ from the ones that were introduced, only the mesh size was changed. The model can be seen on the figure below, which has a geometric properties of:

$$A_i = 22 \text{ mm}, \quad B_i = 18 \text{ mm}, \quad t_i = 2.5 \text{ mm} \quad \theta_i = 65^\circ. \quad (\text{B.1})$$

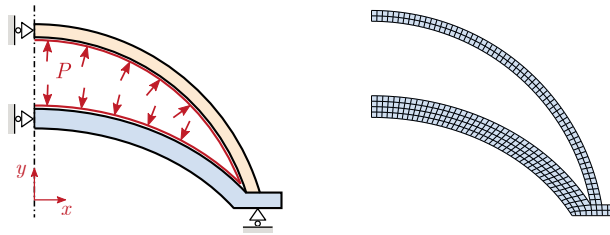


Figure B.1: The finite element model (left) with a coarse mesh (right)

The mesh size is automatically calculated by a program code, with the assumption, that the outer shell (which has a thickness of $t_o = t_i/2$) is divided into M number of elements ($M \in \mathbb{N}^+$). Three cases were investigated, starting from $M = 2$, then $M = 4$ and eventually reaching $M = 8$.

The resulting pressure - volume curve from the model discretized with the finest element size ($M = 8$) can be seen on the left of Fig. B.2. If we plot the relative error between the

$M8 - M4$ and the $M8 - M2$ case, we can create a diagram which can be seen on the right hand side of Fig. B.2. Taking a closer look, we can see that even the biggest difference was around 2.2% between the $M8$ and $M4$ case (even in the case of $M8$ and $M2$ this maxima is around 3.1%), which is negligible in the case of the original pressure range.

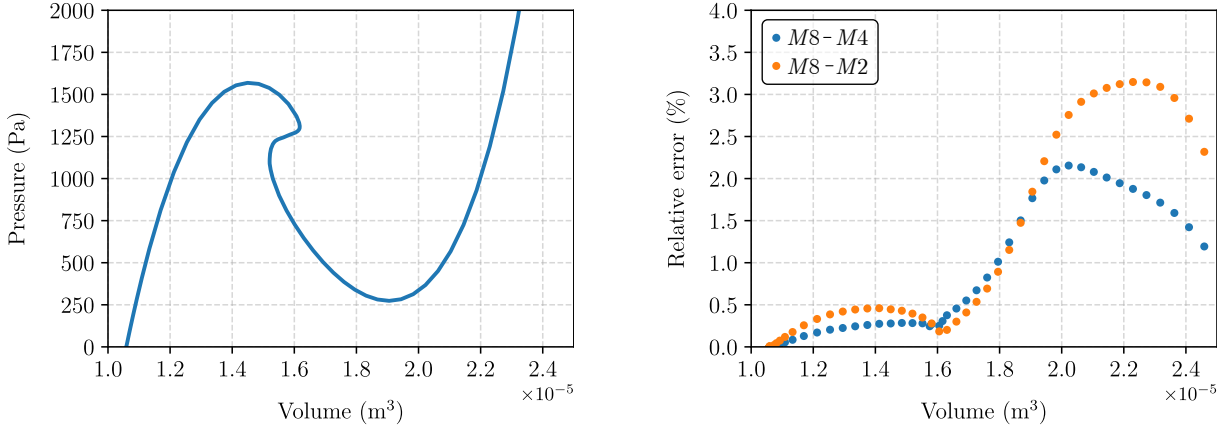


Figure B.2: The resulting pressure-volume curve from the finest mesh (left) and the relative errors between the meshes (right)

As a consequence of this, we can be sure that our numerical calculations will be independent of the mesh size, if we use the $M = 4$ division rule for all of our models.

B.2 Deformation of the measurement container

This section of the appendix contains a short study of the measurement container which is made out of ABS, to prove that the occurring pressures does not deform it significantly.

The boundary conditions (fixed support at the bottom) and the loading by the pressure ($P = 20$ kPa, which is beyond the expected range for the shells) can be seen on the figure below. The mesh size was appropriately fine enough, and the results stated that the maximum deformation of the container is around 0.09 mm.

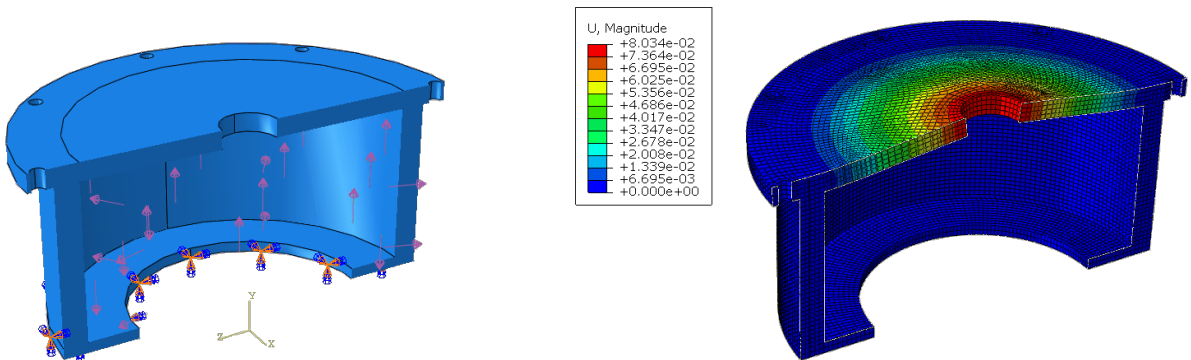


Figure B.3: The boundary conditions and the results (deformation magnified by 50)

The material properties for the ABS was determined as $E = 2.04$ GPa, $\nu = 0.37$ and $\rho = 1040$ kg/m³ based on *MatWeb*.

C

Characteristic of the used diaphragm pump

The characteristic of the diaphragm pump is defined by a voltage-volume flow rate curve. We can create a simple measurement setup, to deduct this relationship as follows:

1. Set up a range of voltage inputs: 5, 6, ... 12 (V), note that the pump cant operate below 5 V of input voltage.
2. Operate the pump for one minute for each input, and measure the volume of the fluid at the destination vessel.
3. Graph the correlation between these.

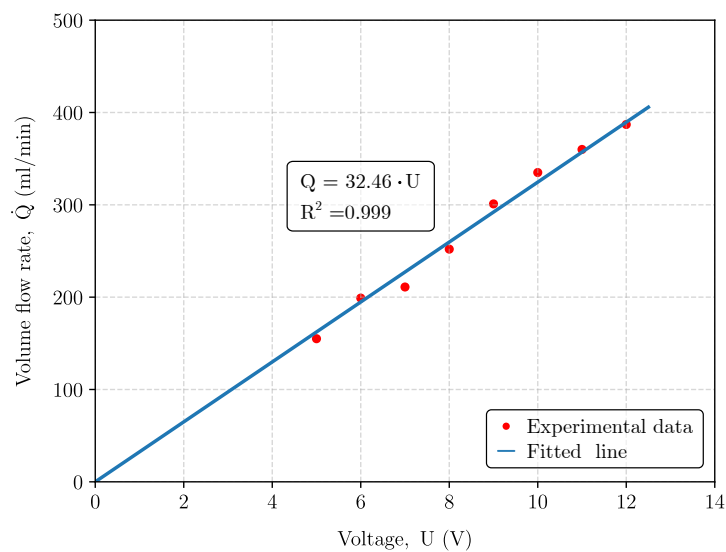


Figure C.1: The linear pump characteristic



Optimization script for the equivalent circle

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.optimize import curve_fit
4
5 # Parameters of the elliptical segment
6 A_ell = 22.0 # Semi - major axis
7 B_ell = 18.0 # Semi - minor axis
8 Q_ell = np.radians(60.0) # Opening angle
9
10 # Function to calculate the points of the elliptical segment
11 def ELLIPSEpoints(A,B,Q):
12     t = np.linspace(np.pi/2-Q, np.pi/2+Q, 1000)
13
14     x = A * np.cos(t) # x coordinate
15     y = B * np.sin(t) # y coordinate
16
17     return x, y
18
19 # Function to calculate the radius of ellipse at a point defined by Q
20 def ELLIPSEradius(A,B,Q):
21     S = np.sin(np.pi/2-Q)
22     C = np.cos(np.pi/2-Q)
23
24     r = A*B/np.sqrt(A**2*S**2 + B**2*C**2)
25
26     return r
27
28 # Function to calculate the points of the circular segment
29 def CIRCLEpoints(ellipse_x,delta,R):
30     x = ellipse_x
31     y = []
32
33     for i in range(len(x)):
34         yc = delta + np.sqrt(R**2 - x[i]**2)
35         y.append(yc)
36
37     return y
```

```

38
39 # Function to calculate the points of the circular segment with the obtained values
40 def CIRCLEobtained(R,delta,Q):
41     t = np.linspace(np.pi/2-Q, np.pi/2+Q, 1000)
42     x = R * np.cos(t)
43     y = R * np.sin(t) + delta
44
45     return x, y
46
47 # The elliptical segment
48 x_ell, y_ell = ELLIPSEpoints(A_ell, B_ell, Q_ell)
49
50 # Find the best fitting circle with the initial guess of p0 = [delta, R]
51 constants = curve_fit(CIRCLEpoints, x_ell, y_ell, p0=[-2, 22])
52 delta_cir = round(constants[0][0], 5)
53 R_cir = round(constants[0][1], 5)
54
55 # Find the opening angle of the circle by the law of sines
56 R_ell = ELLIPSEradius(A_ell, B_ell, Q_ell)
57 Q_cir = np.arcsin(R_ell/R_cir*np.sin(np.pi-Q_ell))
58
59 # Print the obtained values
60 print(f'The obtained parameters of the equivalent circle:\n* R = {R_cir} [mm],\n*
61       Q_cir = {round(np.degrees(Q_cir), 3)} deg.')
62
63 # The circular segment
64 x_cir, y_cir = CIRCLEobtained(R_cir,delta_cir,Q_cir)
65
66 # Plot the ellipse and the circle
67 plt.figure()
68
69 plt.plot(x_ell, y_ell, linewidth=2)
70 plt.plot(x_cir, y_cir, linewidth=2)
71 plt.scatter(0, 0, s=10, color='C0')
72 plt.scatter(0, delta_cir, s=10, color='C1')
73
74 plt.legend(labels=['Ellipse', 'Circle'])
75 plt.xlim(-A_ell*1.5, A_ell*1.5)
76 plt.ylim(delta_cir*1.5, B_ell*1.5)
77 plt.gca().set_aspect('equal', adjustable='box')
78 plt.title('Comparison')
79 plt.xlabel('X')
80 plt.ylabel('Y')
81 plt.grid()
82 plt.savefig('test.pdf')
83 plt.show()

```