Thick cylinder under pressure

Mohammad Abazari

July 28, 2022

Contents

1 Lame's problem-Thick cylinder subjected to internal pressure

Consider a thick cylinder of inner radius R_i, outer radius R_o, length L subjected to internal pressure P_i and outer pressure $P_o = 0$. Two cases of plane-stress $\sigma_z = 0$ and -strain $\varepsilon_z = 0$ are studied.

1.1 Plane stress

Assuming that both cylinder ends are free thus $\sigma_z = 0$,

$$\frac{\partial \sigma_r}{\partial r} + \frac{\sigma_r - \sigma_\theta}{r} = 0$$

r is the only independent variable in this expression which could be rewriten as below

$$\frac{\mathrm{d}}{\mathrm{d}r}(r\sigma_r) - \sigma_\theta = 0 \tag{1}$$

Following Hooke's law[1],

$$\begin{aligned}
\varepsilon_r =& \frac{1}{E}\left(\sigma_r - \nu\sigma_\theta\right) \\
\varepsilon_\theta =& \frac{1}{E}\left(\sigma_r - \nu\sigma_\theta\right) \\
\sigma_r =& \frac{E}{1-\nu^2}\left(\varepsilon_r + \nu\varepsilon_\theta\right) \\
\sigma_\theta =& \frac{E}{1-\nu^2}\left(\varepsilon_\theta - \nu\varepsilon_r\right)
\end{aligned}$$

Substituting strain equations,

$$\begin{aligned}
\sigma_r =& \frac{E}{1-\nu^2}\left(\frac{\mathrm{d}u_r}{\mathrm{d}r} + \nu\frac{u_r}{r}\right) \\
\sigma_\theta =& \frac{E}{1-\nu^2}\left(\frac{u_r}{r} + \nu\frac{\mathrm{d}u_r}{\mathrm{d}r}\right)
\end{aligned} \tag{2}$$

Substituting the above in eq. (1)

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}r}\left(r\frac{\mathrm{d}u_r}{\mathrm{d}r} + vu_r\right) - \left(\frac{u_r}{r} + v\frac{\mathrm{d}u_r}{\mathrm{d}r}\right) =& 0 \\
\frac{\mathrm{d}u_r}{\mathrm{d}r} + r\frac{\mathrm{d}^2u_r}{\mathrm{d}r^2} + \nu\frac{\mathrm{d}u_r}{\mathrm{d}r} - \frac{u_r}{r} - \nu\frac{\mathrm{d}u_r}{\mathrm{d}r} =& 0 \\
\frac{\mathrm{d}^2u_r}{\mathrm{d}r^2} + \frac{1}{r}\frac{\mathrm{d}u_r}{\mathrm{d}r} - \frac{u_r}{r^2} =& 0 \\
\frac{\mathrm{d}}{\mathrm{d}r}\left[\frac{1}{r}\frac{\mathrm{d}}{\mathrm{d}r}(u,r)\right] =& 0
\end{aligned}$$

Assuming u_r follows the below function

$$u_r = C_1 r + \frac{C_2}{r} \tag{3}$$

Substituting the above in eq. (2),

$$\begin{aligned}
\sigma_r =& \frac{E}{1-\nu^2}\left[C_1(1+\nu) - C_2(1-\nu)\frac{1}{r^2}\right] \\
\sigma_\theta =& \frac{E}{1-\nu^2}\left[C_1(1+\nu) + C_2(1-\nu)\frac{1}{r^2}\right]
\end{aligned} \tag{4}$$

Constants C_1 and C_2 are applying boundary conditions.

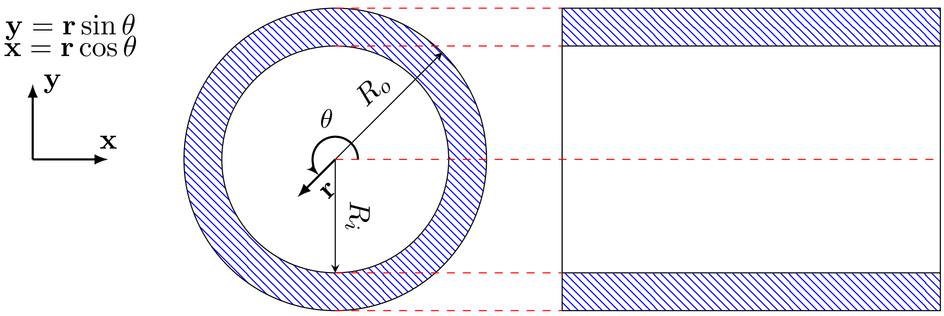


Figure 1.1: Problem geometry.

$$
\begin{aligned}
\sigma_r(r=R_i) &= -P_i = \frac{E}{1-\nu^2}\left[C_1(1+\nu) - C_2(1-\nu)\frac{1}{R_i^2}\right] \\
\sigma_r(r=R_o) &= -P_o = 0 = \frac{E}{1-\nu^2}\left[C_1(1+\nu) - C_2(1-\nu)\frac{1}{R_o^2}\right]
\end{aligned}
$$

Solving for these two constants,

$$
\begin{aligned}
C_1 =& \frac{1-\nu}{E}\frac{P_iR_i^2 - P_oR_o^2}{R_o^2 - R_i^2} \\
C_2 =& \frac{1+\nu}{E}\frac{R_i^2 - R_o^2}{R_o^2 - R_i^2}\left(P_i - P_o\right)
\end{aligned}
$$

Substituting these constants into the above equations

$$
\begin{aligned}
\sigma_r =& \frac{P_iR_i^2 - P_oR_o^2}{R_o^2 - R_i^2} - \frac{R_i^2R_o^2}{r^2}\frac{P_i - P_o}{R_o^2 - R_i^2} \\
\sigma_\theta =& \frac{P_iR_i^2 - P_oR_o^2}{R_o^2 - R_i^2} + \frac{R_i^2R_o^2}{r^2}\frac{P_i - P_o}{R_o^2 - R_i^2}
\end{aligned} \tag{5}
$$

1.1.1 Plane stress problem with internal pressure

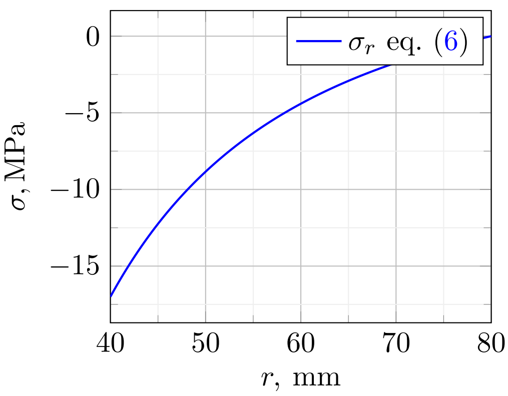
If outside pressure $P_o = 0$, then

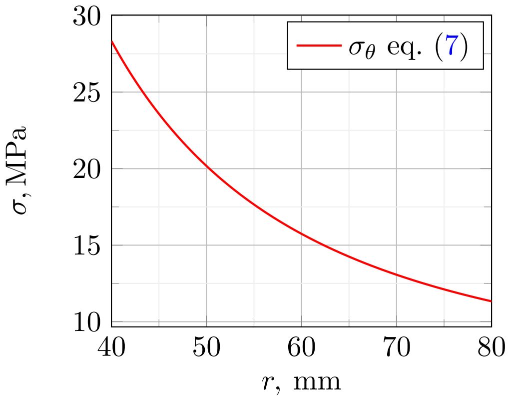
$$
\sigma_r = \frac{P_iR_i^2}{R_o^2 - R_i^2}\left[1 - \frac{R_o^2}{r^2}\right] \tag{6}
$$

$$
\sigma_\theta = \frac{P_iR_i^2}{R_o^2 - R_i^2}\left[1 + \frac{R_o^2}{r^2}\right] \tag{7}
$$

$$
\tag{8}
$$

The above derivation shows that σ_r is compressive through the cylinder thickness while and σ_θ is tensile and positive.





1.2 Plane strain

In the plane strain case, σ_z is assumed constant, eq. (1)

$$\frac{\mathrm{d}}{\mathrm{d}r}(r\sigma_r) - \sigma_\theta = 0$$

From Hooke's law[1]

$$\begin{aligned}
\varepsilon_r =& \frac{1}{E}\left[\sigma_r - v\left(\sigma_\theta + \sigma_z\right)\right] \\
\varepsilon_\theta =& \frac{1}{E}\left[\sigma_\theta - v\left(\sigma_r + \sigma_z\right)\right] \\
\varepsilon_z =& \frac{1}{E}\left[\sigma_r - v\left(\sigma_r + \sigma_\theta\right)\right]
\end{aligned}$$

With $\varepsilon_z = 0$

$$\begin{aligned}
\sigma_z =& v\left(\sigma_r + \sigma_\theta\right) \\
\varepsilon_r =& \frac{1+v}{E}\left[(1-v)\sigma_r - v\sigma_\theta\right] \\
\varepsilon_\theta =& \frac{1+v}{E}\left[(1-v)\sigma_\theta - v\sigma_r\right]
\end{aligned}$$

Solving for stress components

$$\begin{aligned}
\sigma_\theta =& \frac{E}{(1-2v)(1+v)}\left[v\varepsilon_r + (1-v)\varepsilon_\theta\right] \\
\sigma_r =& \frac{E}{(1-2v)(1+v)}\left[(1-v)\varepsilon_r + v\varepsilon_\theta\right]
\end{aligned}$$

Substituting strain

$$\begin{aligned}
\sigma_r =& \frac{E}{(1-2v)(1+v)}\left[(1-v)\frac{du_r}{dr} + v\frac{u_r}{r}\right] \\
\sigma_\theta =& \frac{E}{(1-2v)(1+v)}\left[v\frac{du_r}{dr} + (1-v)\frac{u_r}{r}\right]
\end{aligned} \tag{9}$$

Substituting the above in the equilibrum equations

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}r}\left[(1-v)r\frac{du_r}{dr}+vu_r\right]-v\frac{du_r}{dr}-(1-v)\frac{u_r}{r}=0\\
\frac{du_r}{dr}+r\frac{\mathrm{d}^2u_r}{\mathrm{d}r^2}-\frac{u_r}{r}=0\\
\frac{\mathrm{d}}{\mathrm{d}r}\left(\frac{\mathrm{d}u}{\mathrm{d}r}+\frac{u_r}{r}\right)=0
\end{aligned}$$

Assuming eq. (3) for u_r and substituting into eq. (9)

$$\begin{aligned}
\sigma_\theta =&\frac{E}{(1-2v)(1+v)}\left[C_1+(1-2v)\frac{C_2}{r^2}\right]\\
\sigma_r =&\frac{E}{(1-2v)(1+v)}\left[C_1-(1-2v)\frac{C_2}{r^2}\right]
\end{aligned}$$

Again applying boundary conditions

$$\begin{aligned}
\sigma_r(r=R_i)=&-P_i=\frac{E}{(1-2v)(1+v)}\left[C_1-(1-2v)\frac{C_2}{R_i^2}\right]\\
\sigma_r(r=R_o)=&-P_o=\frac{E}{(1-2v)(1+v)}\left[C_1+(1-2v)\frac{C_2}{R_o^2}\right]
\end{aligned}$$

Thus,

$$\begin{aligned}
C_1 =&\frac{(1-2v)(1+v)}{E}\frac{P_oR_o^2-P_iR_i^2}{R_i^2-R_o^2}\\
C_2 =&\frac{1+v}{E}\frac{(P_o-P_i)\,R_i^2R_o^2}{R_i^2-R_o^2}
\end{aligned}$$

Substituting these constants into the above,

$$\begin{aligned}
\sigma_r =&\frac{P_iR_i^2-P_oR_o^2}{R_o^2-R_i^2}-\frac{R_i^2R_o^2}{r^2}\frac{P_i-P_o}{R_o^2-R_i^2}\\
\sigma_\theta =&\frac{P_iR_i^2-P_oR_o^2}{R_o^2-R_i^2}+\frac{R_i^2R_o^2}{r^2}\frac{P_i-P_o}{R_o^2-R_i^2}
\end{aligned}$$

Which equals eq. (5).

2 MATLAB modeling

A cylinder of inner radius of $R_i = 40$mm outer radius $R_o = 80$mm length $L = 1000$mm inside pressure 17MPa outside pressure 2 was meshed with minimum element dimension of 1mm and maximum element dimension of 10mm.

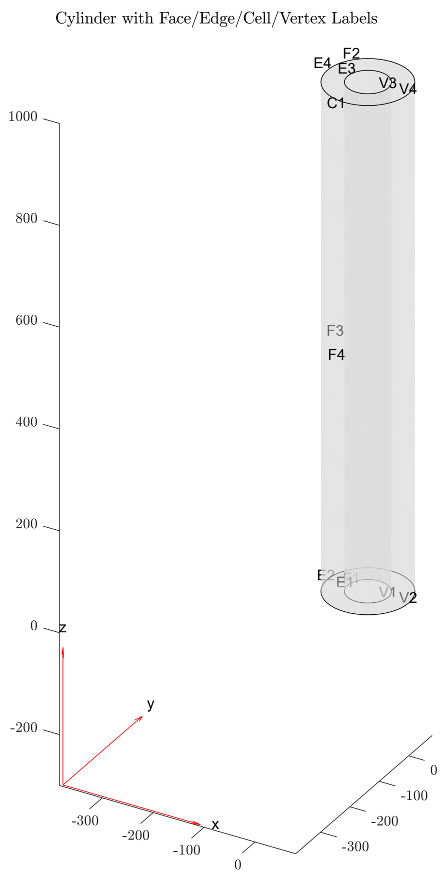


Figure 2.1: 3D model geometry.

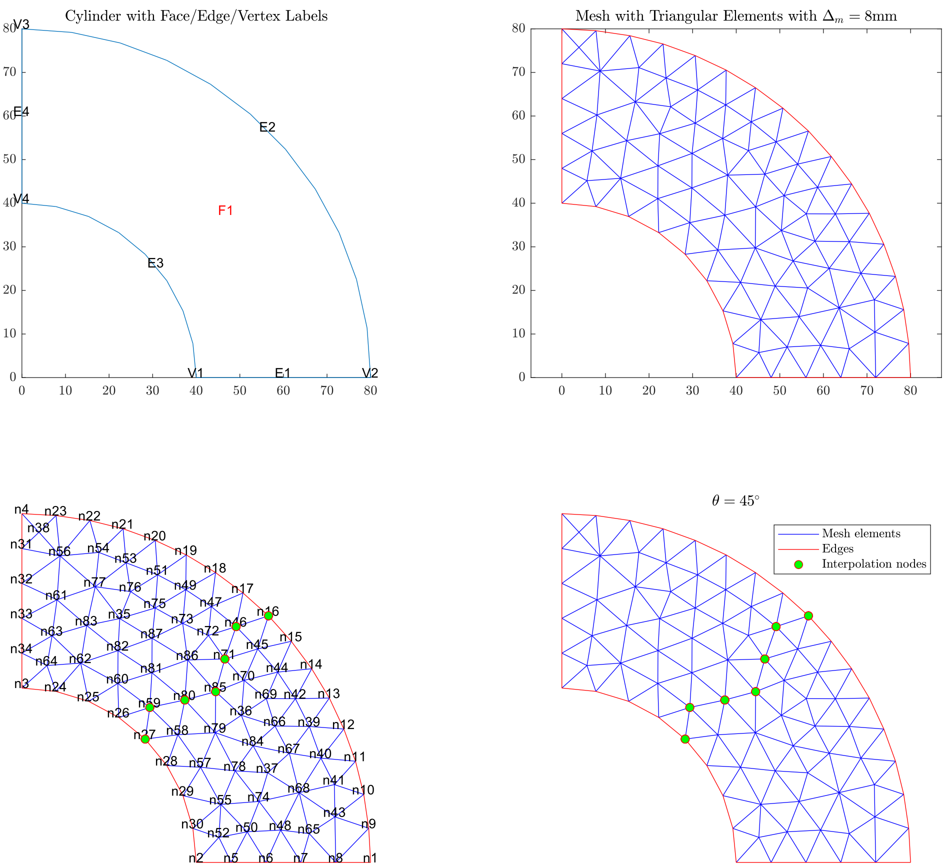


Figure 2.2: 2D model and mesh.

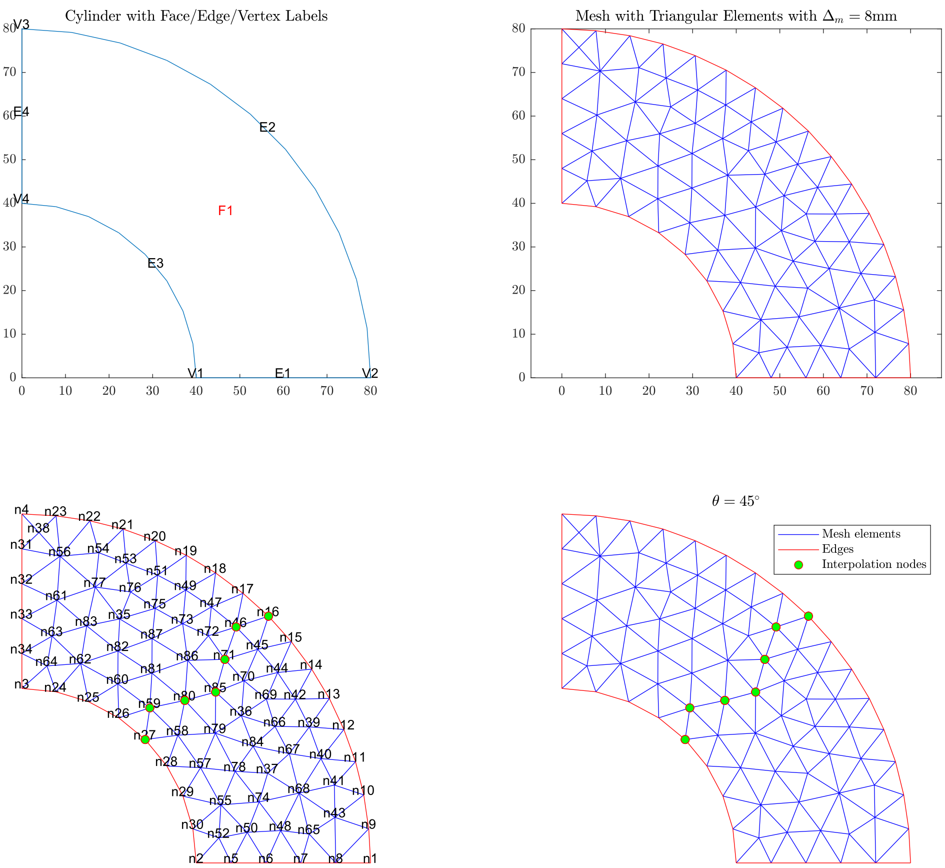
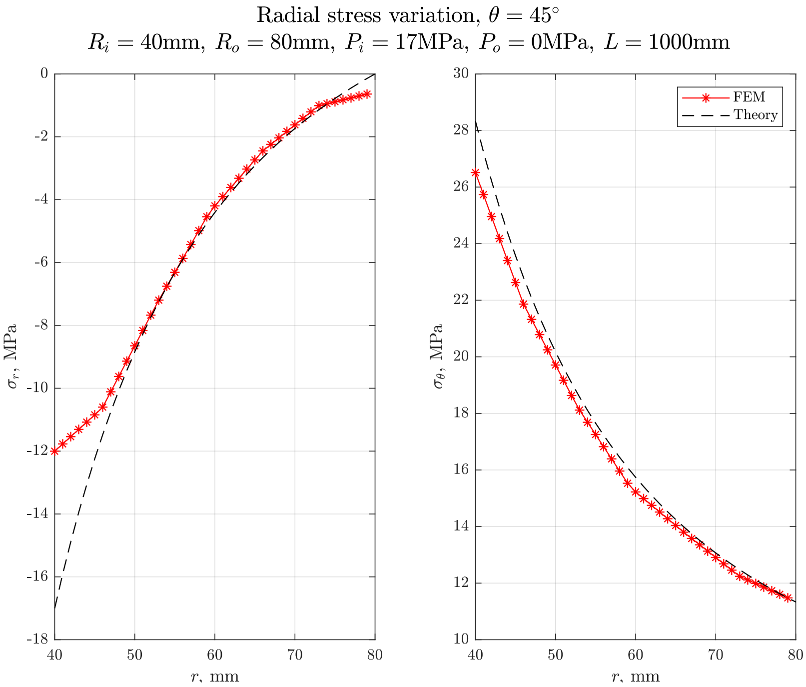
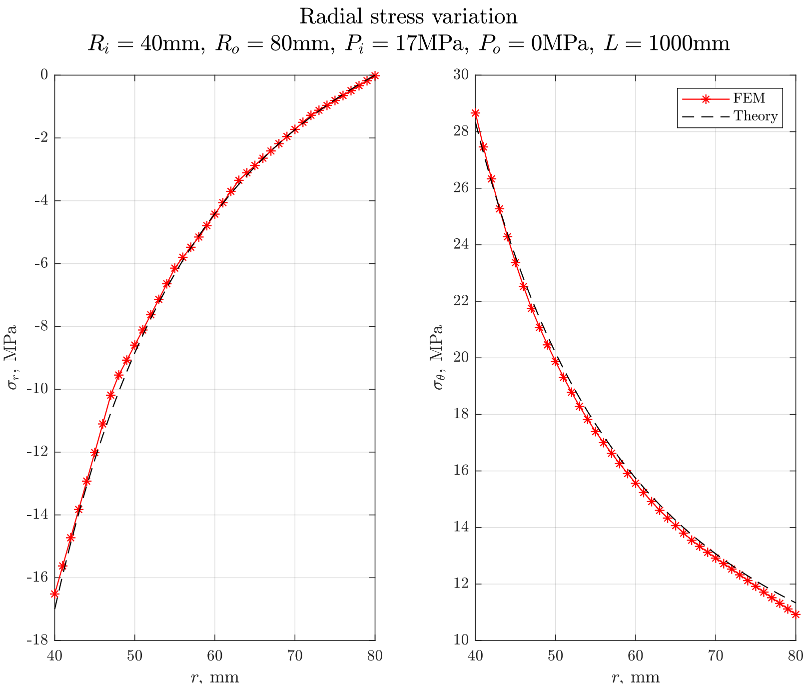


Figure 2.3: 3D model and mesh.



(a) 2D results.



(b) 3D results, Δ.

Figure 2.4: Radial variation of stress components.

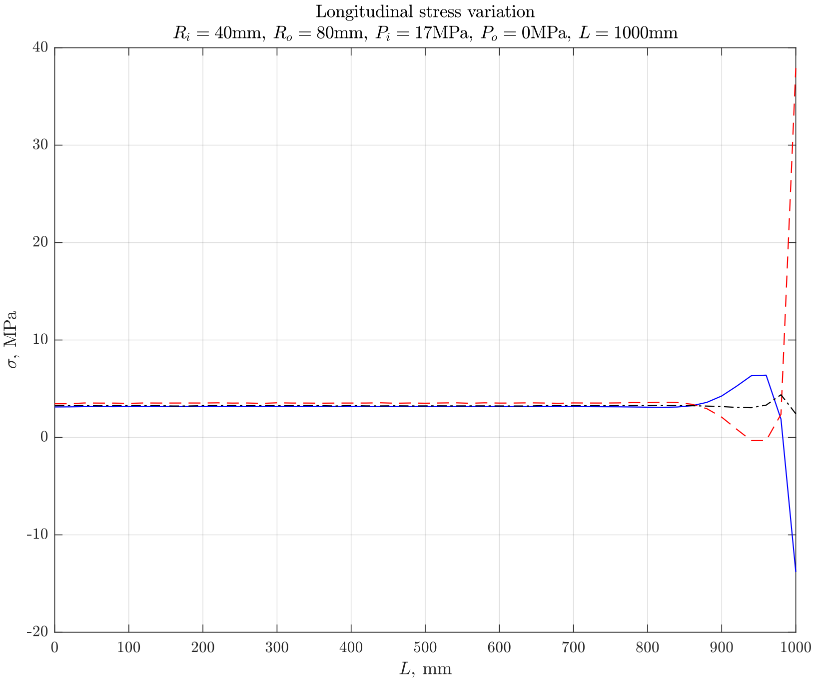


Figure 2.5: Stress variation in length.

3 Conclusions

- Both 2D and 3D numerical models (§ 2) agree with theoretical derivation (§ 1) (fig. 2.4).
- Based on both numerical and theoretical results radial (σ_r) and angular stresses (σ_θ) reduce radially(fig. 2.4).
- 3D numerical results shows no significant change in longitudinal stress (σ_z) which sattisfies the plane stress and strain assumptions(fig. 2.5).
- Longitudinal stress oscilations in the cylinder length (fig. 2.5) showcases model sensitivity to proper boundary conditions which is dealt with through finer mesh sizes.

4 Appendices

4.1 Stress transformation from cartesian to cylindrical coordinates

Converting Tensors from Cartesian to Cylindrical

November 4, 2019

From `http://solidmechanics.org/text/AppendixD/AppendixD.htm` we get:

$$\begin{bmatrix} S_{rr} & S_{r\theta} & S_{rz} \\ S_{\theta r} & S_{\theta\theta} & S_{\theta z} \\ S_{zr} & S_{\theta z} & S_{zz} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{\theta\theta} & S_{\theta z} \\ S_{zr} & S_{\theta z} & S_{zz} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Take the first two matrices and multiply them:
Note: to ease space constraints, the following translations have been defined:

$$\cos\theta = \mathbf{c}_\theta \tag{2}$$

$$\sin\theta = \mathbf{s}_\theta \tag{3}$$

$$\begin{aligned} &\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{\theta\theta} & S_{\theta z} \\ S_{zr} & S_{\theta z} & S_{zz} \end{bmatrix} \\ &\implies \begin{bmatrix} \mathbf{c}_\theta S_{xx} + \mathbf{s}_\theta S_{yx} + 0 & \mathbf{c}_\theta S_{xy} + \mathbf{s}_\theta S_{yy} + 0 & \mathbf{c}_\theta S_{xz} + \mathbf{s}_\theta S_{yz} + 0 \\ -\mathbf{s}_\theta S_{xx} + \mathbf{c}_\theta S_{yx} + 0 & -\mathbf{s}_\theta S_{xy} + \mathbf{c}_\theta S_{yy} + 0 & -\mathbf{s}_\theta S_{xz} + \mathbf{c}_\theta S_{yz} + 0 \\ 0 + 0 + S_{zx} & 0 + 0 + S_{zy} & 0 + 0 + S_{zz} \end{bmatrix} \\ &\implies \begin{bmatrix} \mathbf{c}_\theta S_{xx} + \mathbf{s}_\theta S_{yx} & \mathbf{c}_\theta S_{xy} + \mathbf{s}_\theta S_{yy} & \mathbf{c}_\theta S_{xz} + \mathbf{s}_\theta S_{yz} \\ -\mathbf{s}_\theta S_{xx} + \mathbf{c}_\theta S_{yx} & -\mathbf{s}_\theta S_{xy} + \mathbf{c}_\theta S_{yy} & -\mathbf{s}_\theta S_{xz} + \mathbf{c}_\theta S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \end{aligned} \tag{4}$$

Multiply the result by the third matrix:

$$\begin{aligned} &\begin{bmatrix} \mathbf{c}_\theta S_{xx} + \mathbf{s}_\theta S_{yx} & \mathbf{c}_\theta S_{xy} + \mathbf{s}_\theta S_{yy} & \mathbf{c}_\theta S_{xz} + \mathbf{s}_\theta S_{yz} \\ -\mathbf{s}_\theta S_{xx} + \mathbf{c}_\theta S_{yx} & -\mathbf{s}_\theta S_{xy} + \mathbf{c}_\theta S_{yy} & -\mathbf{s}_\theta S_{xz} + \mathbf{c}_\theta S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ &\begin{bmatrix} \mathbf{c}_\theta^2 S_{xx} + \mathbf{s}_\theta \mathbf{c}_\theta S_{yx} + \mathbf{c}_\theta \mathbf{s}_\theta S_{xy} + \mathbf{s}_\theta^2 S_{yy} & -\mathbf{c}_\theta \mathbf{s}_\theta S_{xx} - \mathbf{s}_\theta^2 S_{yx} + \mathbf{c}_\theta^2 S_{xy} + \mathbf{c}_\theta \mathbf{s}_\theta S_{yy} & \mathbf{c}_\theta S_{xy} + \mathbf{s}_\theta S_{yz} \\ -\mathbf{s}_\theta \mathbf{c}_\theta S_{xx} + \mathbf{c}_\theta^2 S_{yx} - \mathbf{s}_\theta^2 S_{xy} + \mathbf{c}_\theta \mathbf{s}_\theta S_{yy} & \mathbf{s}_\theta^2 S_{xx} - \mathbf{c}_\theta \mathbf{s}_\theta S_{yx} - \mathbf{c}_\theta \mathbf{s}_\theta S_{xy} + \mathbf{c}_\theta^2 S_{yy} & -\mathbf{s}_\theta S_{xz} + \mathbf{c}_\theta S_{yz} \\ \mathbf{c}_\theta S_{zx} + \mathbf{s}_\theta S_{zy} & -\mathbf{s}_\theta S_{zx} + \mathbf{c}_\theta S_{zy} & S_{zz} \end{bmatrix} \end{aligned} \tag{5}$$

Combine like terms, assuming that the tensor is symmetric (ie. $S_{ij} = S_{ji}$):

$$(5) \implies \begin{bmatrix} \mathbf{c}_\theta^2 S_{xx} + 2\mathbf{c}_\theta \mathbf{s}_\theta S_{xy} + \mathbf{s}_\theta^2 S_{yy} & -\mathbf{c}_\theta \mathbf{s}_\theta S_{xx} + (\mathbf{c}_\theta^2 - \mathbf{s}_\theta^2) S_{xy} + \mathbf{c}_\theta \mathbf{s}_\theta S_{yy} & \mathbf{c}_\theta S_{xz} + \mathbf{s}_\theta S_{yz} \\ -\mathbf{c}_\theta \mathbf{s}_\theta S_{xx} + (\mathbf{c}_\theta^2 - \mathbf{s}_\theta^2) S_{xy} + \mathbf{c}_\theta \mathbf{s}_\theta S_{yy} & \mathbf{s}_\theta^2 S_{xx} - 2\mathbf{c}_\theta \mathbf{s}_\theta S_{xy} + \mathbf{c}_\theta^2 S_{yy} & -\mathbf{s}_\theta S_{xz} + \mathbf{c}_\theta S_{yz} \\ \mathbf{c}_\theta S_{xz} + \mathbf{s}_\theta S_{yz} & -\mathbf{s}_\theta S_{xz} + \mathbf{c}_\theta S_{yz} & S_{zz} \end{bmatrix} \tag{6}$$

This can be further simplified by combinging the $\mathbf{c}_\theta \mathbf{s}_\theta$ groups:

$$\begin{bmatrix} S_{rr} & S_{r\theta} & S_{rz} \\ S_{\theta r} & S_{\theta\theta} & S_{\theta z} \\ S_{zr} & S_{\theta z} & S_{zz} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_\theta^2 S_{xx} + 2\mathbf{c}_\theta\mathbf{s}_\theta S_{xy} + \mathbf{s}_\theta^2 S_{yy} & \mathbf{c}_\theta\mathbf{s}_\theta(S_{yy} - S_{xx}) + (\mathbf{c}_\theta^2 - \mathbf{s}_\theta^2)S_{xy} & \mathbf{c}_\theta S_{xz} + \mathbf{s}_\theta S_{yz} \\ \mathbf{c}_\theta\mathbf{s}_\theta(S_{yy} - S_{xx}) + (\mathbf{c}_\theta^2 - \mathbf{s}_\theta^2)S_{xy} & \mathbf{s}_\theta^2 S_{xx} - 2\mathbf{c}_\theta\mathbf{s}_\theta S_{xy} + \mathbf{c}_\theta^2 S_{yy} & -\mathbf{s}_\theta S_{xz} + \mathbf{c}_\theta S_{yz} \\ \mathbf{c}_\theta S_{xz} + \mathbf{s}_\theta S_{yz} & -\mathbf{s}_\theta S_{xz} + \mathbf{c}_\theta S_{yz} & S_{zz} \end{bmatrix} \tag{7}$$

To list off the unique components:

$$S_{rr} = \mathbf{c}_\theta^2 S_{xx} + 2\mathbf{c}_\theta\mathbf{s}_\theta S_{xy} + \mathbf{s}_\theta^2 S_{yy} \tag{8}$$

$$S_{r\theta} = \mathbf{c}_\theta\mathbf{s}_\theta(S_{yy} - S_{xx}) + (\mathbf{c}_\theta^2 - \mathbf{s}_\theta^2)S_{xy} \tag{9}$$

$$S_{rz} = \mathbf{c}_\theta S_{xz} + \mathbf{s}_\theta S_{yz} \tag{10}$$

$$S_{\theta\theta} = \mathbf{s}_\theta^2 S_{xx} - 2\mathbf{c}_\theta\mathbf{s}_\theta S_{xy} + \mathbf{c}_\theta^2 S_{yy} \tag{11}$$

$$S_{\theta z} = -\mathbf{s}_\theta S_{xz} + \mathbf{c}_\theta S_{yz} \tag{12}$$

$$S_{zz} = S_{zz} \tag{13}$$

4.2 3D model

```
%% Preamble
clc; clear;
% startup
set(groot,'DefaultTextInterpreter','latex')
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
set(findall(gcf,'-property','FontSize'),'FontSize',14);
format compact;
% % set(groot, 'DefaultFigureRenderer', 'painters');
close all;

% % Problem statement
   =====================================================
% Geometry description
   ===================================================
Ri = 40; % Inner radius Ri, mm
Ro = 80; % Outer radius Ro, mm
L = 1000; % Length L, mm
Pi = 17; % Inside pressure (radial pressure), N/mm2
Po = 0; % Outside pressure (radial pressure), N/mm2
meshsize_max = 20; % maximum mesh dimension, mm
meshsize_min = 5; % minimum mesh dimension, mm
mesh_order = 'linear'; % or quadratic
% Material Properties
   ====================================================
E = 210e3; % Modulus of elasticity E, N/mm2
nu = 0.3; % Poisson's ratio \nu
%% Model creation
   =========================================================
model = createpde('structural','static-solid');
% Introduce geometry
   =====================================================
gm = multicylinder([Ri Ro],L,'Void',[true,false]);
model.Geometry = gm;
figure
pdegplot(model,'CellLabels','on','FaceLabels','on','
   EdgeLabels',...
    'on','VertexLabels','on','FaceAlpha',0.5)
view(30,30);
title('Cylinder with Face/Edge/Cell/Vertex Labels')
```

```
% Assign Material Values
   ================================================
structuralProperties(model,'YoungsModulus',E, ...
                           'PoissonsRatio',nu);
% Applying boundary conditions
   ==========================================
structuralBC(model,'Face',1,'Constraint','symmetric');
structuralBC(model,'Face',2,'Constraint','symmetric');

% Apply pressure loads in and out
   =======================================
structuralBoundaryLoad(model,"Face",3,"Pressure",Pi);
structuralBoundaryLoad(model,"Face",4,"Pressure",Po);

% Mesh description, Generate mesh
   =======================================
generateMesh(model,'Hmax',meshsize_max,'Hmin',meshsize_min,
   ...
    'GeometricOrder',mesh_order);
figure;
pdeplot3D(model);
title(['Mesh with Quadratic Tetrahedral Elements with $\
   Delta_m=',...
    num2str(meshsize_max),'$mm']);

% Calculate solution
   ====================================================
result = solve(model);

%% Evaluate Results
   ======================================================
figure
pdeplot3D(model,'ColorMapData',result.VonMisesStress)
title('Von-Mises Stress, MPa')
colormap('jet')
figure
pdeplot3D(model,'ColorMapData',result.Stress.yy)
title('Von-Mises Stress, MPa')
colormap('jet')
save(['ResultsM',num2str(meshsize_max),'m',...
    num2str(meshsize_min),mesh_order,'.mat'],'result','model
       ')
%% Stress values by radius
```

```
   ===============================================
% Theory
st = @(r) (Pi*Ri^2-Po*Ro^2)/(Ro^2-Ri^2) ...
    + ((Ri^2*Ro^2)./r.^2)*(Pi-Po)/((Ro^2-Ri^2));
sr = @(r) (Pi*Ri^2-Po*Ro^2)/(Ro^2-Ri^2) ...
    - ((Ri^2*Ro^2)./r.^2)*(Pi-Po)/((Ro^2-Ri^2));
% Based on cartesian to cylindrical transformation if Cos(
   theta) =
% Sine(theta) = 0 then Sxx = Syy and Stt = Syy as in the
   present case.
% http://solidmechanics.org/text/AppendixD/AppendixD.htm

% Practice and FEM
   =======================================================
R = Ri:Ro;

S_theta = []; S_r = [];

for r = R
    si = interpolateStress(result,r,0,L/2);
    S_theta = [S_theta,si.syy];
    S_r = [S_r,si.sxx];
end
%% Longitudinal stress values
   ============================================
figure;
Sz_Ro = [];
Sz_Ri = [];
Sz_Rm = [];
Rm = (Ro+Ri)/2;

l = 0:meshsize_max:L;

for li = l
    si_Ro = interpolateStress(result,Ro,0,li);
    si_Ri = interpolateStress(result,Ri,0,li);
    si_Rm = interpolateStress(result,Rm,0,li);
    Sz_Ro = [Sz_Ro,si_Ro.szz];
    Sz_Ri = [Sz_Ri,si_Ri.szz];
    Sz_Rm = [Sz_Rm,si_Rm.szz];
end

%% Visualize results
```

```
  ====================================================
f5 = figure;
subplot(1,2,1);
plot(R,S_r,'b','LineWidth',0.7); hold on;
plot(R,sr(R),'--k','LineWidth',0.7);
xlabel('$r,\:\mathrm{mm}$');
ylabel('$\sigma_r,\:\mathrm{MPa}$');grid on;
subplot(1,2,2);
plot(R,S_theta,'b','LineWidth',0.7); hold on;
plot(R,st(R),'--k','LineWidth',0.7);
xlabel('$r,\:\mathrm{mm}$');
ylabel('$\sigma_\theta,\:\mathrm{MPa}$'); grid on;
legend('FEM','Theory');
sgtitle({'Radial stress variation',...
    ['$R_i',num2str(Ri),'\mathrm{mm},\:R_o=',num2str(Ro),...
    '\mathrm{mm},\:P_i=',num2str(Pi),'\mathrm{MPa},\:P_o=',
       num2str(Po),...
    '\mathrm{MPa},\:L=',num2str(L),'\mathrm{mm}$']});
f6 = figure;
plot(l,Sz_Ro,'-b','LineWidth',0.7); hold on;
plot(l,Sz_Ri,'--r','LineWidth',0.7);
plot(l,Sz_Rm,'-.k','LineWidth',0.7);
xlabel('$L,\:\mathrm{mm}$');
ylabel('$\sigma,\:\mathrm{MPa}$');grid on;
title({'Longitudinal stress variation',...
    ['$R_i',num2str(Ri),'\mathrm{mm},\:R_o=',num2str(Ro),...
    '\mathrm{mm},\:P_i=',num2str(Pi),'\mathrm{MPa},\:P_o=',
       num2str(Po),...
    '\mathrm{MPa},\:L=',num2str(L),'\mathrm{mm}$']});
```

4.3 2D model

```
%% Preamble
clc; clear;
% startup
set(groot,'DefaultTextInterpreter','latex')
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
set(findall(gcf,'-property','FontSize'),'FontSize',14);
format compact;
% % set(groot, 'DefaultFigureRenderer', 'painters');
close all;

% % Problem statement
```

```
% Geometry description
   ==================================================
Ri = 40; % Inner radius Ri, mm
Ro = 80; % Outer radius Ro, mm
L = 1000; % Length L, mm
Pi = 17; % Inside pressure (radial pressure), N/mm2
Po = 0; % Outside pressure (radial pressure), N/mm2
meshsize_max = 8; % maximum mesh dimension, mm
meshsize_min = 1; % minimum mesh dimension, mm
mesh_order = 'linear'; % or quadratic

% Material Properties
   ===================================================
E = 210e3; % Modulus of elasticity E, N/mm2
nu = 0.3; % Poisson's ratio \nu
% rho = 8000; % Mass density \rho, kg/m3 irrelevent for
   static model

% Measurement
theta = pi/4;

%% Model creation
   ========================================================
model = createpde('structural','static-planestress');
% Introduce geometry
   ====================================================
importGeometry(model,'mesh2d.stl');

f1 = figure('Position',[100,40,1200,1000],'Renderer','
   painters');
subplot(2,2,1);
pdegplot(model,'EdgeLabels','on','FaceLabels','on','
   VertexLabels','on');
title('Cylinder with Face/Edge/Vertex Labels')

% Assign Material Values
structuralProperties(model,'YoungsModulus',E, ...
                           'PoissonsRatio',nu);
% Applying boundary conditions
structuralBC(model,'Edge',1,'Constraint','symmetric');
structuralBC(model,'Edge',4,'Constraint','symmetric');
%
% % Apply pressure loads in and out
```

```
structuralBoundaryLoad(model,"Edge",3,"Pressure",Pi);
structuralBoundaryLoad(model,"Edge",2,"Pressure",Po);
%
% % Mesh description, Generate mesh
mesh = generateMesh(model,'Hmax',meshsize_max,'Hmin',
   meshsize_min,'GeometricOrder',mesh_order);
subplot(2,2,2);
pdeplot(model);
title(['Mesh with Triangular Elements with $\Delta_m=',
   num2str(meshsize_max),'$mm']);

% Specify line of nodes considered for analysis
R = ((Ri):(Ro)); n_ids = zeros(size(R));
for i = 1:length(R)
    r = R(i);
    point = r*[cos(theta);sin(theta)];
    n_ids(i) = findNodes(mesh,'nearest',point);
end
subplot(2,2,3);
pdemesh(model,'NodeLabels','on'); hold on;
plot(mesh.Nodes(1,n_ids),mesh.Nodes(2,n_ids),'or','
   MarkerFaceColor','g')
axis off;
subplot(2,2,4);
pdemesh(model); hold on
plot(mesh.Nodes(1,n_ids),mesh.Nodes(2,n_ids),'or','
   MarkerFaceColor','g')
axis off;

title(['$\theta=',num2str(rad2deg(theta)),'^\circ$'])
legend('Mesh elements','Edges','Interpolation nodes');

% print('-f',['Figures/fig01m',num2str(meshsize_max)],'-dsvg
   ')
%% Calculate solution
result = solve(model);
%% Visualize
figure('Position',[100,100,1900,600],'Renderer','painters');
subplot(1,4,1);
pdeplot(model,'XYData',result.Stress.sxx,'ColorMap','jet')
title('Stress $\sigma_{xx}$');
axis equal;
subplot(1,4,2);
```

```
pdeplot(model,'XYData',result.Stress.syy,'ColorMap','jet')
title('Stress $\sigma_{yy}$');
axis equal;
subplot(1,4,3);
pdeplot(model,'XYData',result.Stress.sxy,'ColorMap','jet')
title('Stress $\sigma_{xy}=\sigma_{yx}$');
axis equal;
subplot(1,4,4);
pdeplot(model,'XYData',result.VonMisesStress,'ColorMap','jet
    ');
title('Von-Mises stress');
axis equal;
print('-f',['Figures/fig02m',num2str(meshsize_max)],'-dsvg')
% save(['Results2dM',num2str(meshsize_max),'m',num2str(
    meshsize_min),mesh_order,'.mat'],'result','model')
%% More
st = @(r) (Pi*Ri^2-Po*Ro^2)/(Ro^2-Ri^2) + ((Ri^2*Ro^2)./r
    .^2)*(Pi-Po)/((Ro^2-Ri^2));
sr = @(r) (Pi*Ri^2-Po*Ro^2)/(Ro^2-Ri^2) - ((Ri^2*Ro^2)./r
    .^2)*(Pi-Po)/((Ro^2-Ri^2));
R = ((Ri):(Ro));

S_theta = []; S_r = [];
% figure();
for r = R
    Ct = cos(theta); St = sin(theta);
    si = interpolateStress(result,r*Ct,r*St);
    S_theta = [S_theta,si.syy*Ct^2 - 2*Ct*St*si.sxy + si.sxx
        *St^2];
    S_r = [S_r,si.sxx*Ct^2 + 2*Ct*St*si.sxy + si.syy*St^2];
end
f5 = figure('Position',[100,80,800,600],'Renderer','painters
    ');
subplot(1,2,1);
plot(R,S_r,'-*r','LineWidth',0.7); hold on;
plot(R,sr(R),'--k','LineWidth',0.7);
xlabel('$r,\:\mathrm{mm}$');
ylabel('$\sigma_r,\:\mathrm{MPa}$');grid on;
subplot(1,2,2);
plot(R,S_theta,'-*r','LineWidth',0.7); hold on;
plot(R,st(R),'--k','LineWidth',0.7);
xlabel('$r,\:\mathrm{mm}$');
ylabel('$\sigma_\theta,\:\mathrm{MPa}$'); grid on;
```

```
legend('FEM','Theory');
sgtitle({['Radial stress variation, $\theta=',num2str(
   rad2deg(theta)),'^\circ$'],...
    ['$R_i=',num2str(Ri),'\mathrm{mm},\:R_o=',num2str(Ro),...
    '\mathrm{mm},\:P_i=',num2str(Pi),'\mathrm{MPa},\:P_o=',
       num2str(Po),...
    '\mathrm{MPa},\:L=',num2str(L),'\mathrm{mm}$']});
% print('-f',['Figures/fig03m',num2str(meshsize_max)],'-dsvg
   ')
```

4.4 2D geometry generation with ABAQUS

```
# -*- coding: mbcs -*-
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=200
                                         .0)
mdb.models['Model-1'].sketches['__profile__'].CircleByCenterPerimeter(
                                         center=(
    0.0, 0.0), point1=(80.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].CircleByCenterPerimeter(
                                         center=(
    0.0, 0.0), point1=(40.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(0.0, 80.0),
                                         point2=(
    0.0, 40.0))
mdb.models['Model-1'].sketches['__profile__'].VerticalConstraint(
                                         addUndoState=
    False, entity=mdb.models['Model-1'].sketches['__profile__'].geometry[
                                             4])
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
    addUndoState=False, entity1=
    mdb.models['Model-1'].sketches['__profile__'].geometry[2], entity2=
    mdb.models['Model-1'].sketches['__profile__'].geometry[4])
mdb.models['Model-1'].sketches['__profile__'].CoincidentConstraint(
    addUndoState=False, entity1=
    mdb.models['Model-1'].sketches['__profile__'].vertices[3], entity2=
    mdb.models['Model-1'].sketches['__profile__'].geometry[2])
mdb.models['Model-1'].sketches['__profile__'].CoincidentConstraint(
    addUndoState=False, entity1=
    mdb.models['Model-1'].sketches['__profile__'].vertices[4], entity2=
```

```
    mdb.models['Model-1'].sketches['__profile__'].geometry[3])
mdb.models['Model-1'].sketches['__profile__'].Line(point1=(40.0, 0.0),
                                       point2=(
    80.0, 0.0))
mdb.models['Model-1'].sketches['__profile__'].HorizontalConstraint(
    addUndoState=False, entity=
    mdb.models['Model-1'].sketches['__profile__'].geometry[5])
mdb.models['Model-1'].sketches['__profile__'].PerpendicularConstraint(
    addUndoState=False, entity1=
    mdb.models['Model-1'].sketches['__profile__'].geometry[3], entity2=
    mdb.models['Model-1'].sketches['__profile__'].geometry[5])
mdb.models['Model-1'].sketches['__profile__'].autoTrimCurve(curve1=
    mdb.models['Model-1'].sketches['__profile__'].geometry[3], point1=(
    -28.4243316650391, 22.0929641723633))
mdb.models['Model-1'].sketches['__profile__'].autoTrimCurve(curve1=
    mdb.models['Model-1'].sketches['__profile__'].geometry[2], point1=(
    -48.0591735839844, 64.5711441040039))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='Part-1',
                                       type=
    DEFORMABLE_BODY)
mdb.models['Model-1'].parts['Part-1'].BaseShell(sketch=
    mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']
mdb.models['Model-1'].parts['Part-1'].seedPart(deviationFactor=0.1,
    minSizeFactor=0.1, size=11.0)
mdb.models['Model-1'].parts['Part-1'].generateMesh()
mdb.models['Model-1'].parts['Part-1'].deleteMesh()
mdb.models['Model-1'].parts['Part-1'].seedPart(deviationFactor=0.1,
    minSizeFactor=0.1, size=2.0)
mdb.models['Model-1'].parts['Part-1'].generateMesh()
mdb.models['Model-1'].parts['Part-1'].deleteMesh(regions=
    mdb.models['Model-1'].parts['Part-1'].faces.getSequenceFromMask(('[#1
                                                ]',
    ), ))
mdb.models['Model-1'].parts['Part-1'].setMeshControls(elemShape=TRI,
                                       regions=
    mdb.models['Model-1'].parts['Part-1'].faces.getSequenceFromMask(('[#1
                                                ]',
    ), ))
mdb.models['Model-1'].parts['Part-1'].generateMesh()
```

References

[1] R Hooke. De potentia restitutiva, or of spring explaining the power of springing bodies. *London, UK: John Martyn*, 23, 1678. 2, 4