Mathematical Theory of Shells on Elastic Foundations:

An Analysis of Boundary Forms, Constraints, and Applications to Friction and Skin Abrasion

Matlab Codes

Kavinda Jayawardana
zcahe58@ucl.ac.uk
Department of Mathematics
University College London
2016

Contents

1 Kikuchi and Oden's Model for Coulomb's law of Static Friction

In this chapter we present a numerical code for an example of Kikuchi and Oden's [6] model for Coulomb's law of static friction in curvilinear coordinates (see section 2.6 of Jayawardana [4]) implemented in Matlab, i.e. `OdenProb1.m`. Note that to find numerical solutions we employ Newton's method for nonlinear systems (see chapter 10 of Burden *et al.* [3]).

To find a numerical solution consider the map of a rigid semi-prism $(x^1, a\sin(x^2), b\cos(x^2))_{\mathsf{E}}$, where $x^2 \in [-\frac{1}{2}\pi, \frac{1}{2}\pi]$, a is the horizontal radius and b is the vertical radius. Now, assume that an elastic body is over this prism and one is applying a traction τ_0 at $x^2 = -\frac{1}{2}\pi$ and a traction τ_{max} at $x^2 = \frac{1}{2}\pi$. Also, assume that the cylinder is rough and the coefficient of friction between the prism and the body in question is $\frac{1}{2}$. Assume further that the body in question is of thickness h, infinitely long and in contact with an infinitely long semi-prism. This leads to the following map of the unstrained configuration,

$$\bar{\boldsymbol{X}}(x^1, x^2, x^3) = (x^1, a\sin(x^2), b\cos(x^2))_{\mathsf{E}} + x^3(\varphi(x^2))^{-1}(0, b\sin(x^2), a\cos(x^2))_{\mathsf{E}} \ ,$$

where $\varphi(x^2) = (b^2\sin^2(x^2) + a^2\cos^2(x^2))^{\frac{1}{2}}$, $x^1 \in (-\infty, \infty)$, $x^2 \in (-\frac{1}{2}\pi, \frac{1}{2}\pi)$ and $x^3 \in (0, h)$. With some calculations, one finds that the covariant metric tensor is $(g_{ij}) = \mathrm{diag}(1, (\bar{\psi}_2)^2, 1)$ and Christoffel symbols of the second kind are

$$\bar{\Gamma}^2_{22} = (\bar{\psi}_2)^{-1}\partial_2\bar{\psi}_2 \ ,$$
$$\bar{\Gamma}^2_{23} = (\bar{\psi}_2)^{-1}\partial_3\bar{\psi}_2 \ ,$$

where $\bar{\psi}_2 = \varphi(x^2) + x^3 ab(\varphi(x^2))^{-2}$. Now, let $\boldsymbol{v} = (0, v^2(x^2, x^3), v^3(x^2, x^3))$ be the displacement field of the elastic body and let $\delta\boldsymbol{v} = (0, \delta v^2(x^2, x^3), \delta v^3(x^2, x^3))$ be a perturbation of the displacement field. Thus, with some calculations, one finds that covariant derivatives are

$$\bar{\nabla}_2 v^2 = \partial_2 v^2 + \bar{\Gamma}^2_{22}v^2 + \bar{\Gamma}^2_{23}v^3 \ ,$$
$$\bar{\nabla}_2 v^3 = \partial_2 v^3 - (\bar{\psi}_2)^2\bar{\Gamma}^2_{23}v^2 \ ,$$
$$\bar{\nabla}_3 v^2 = \partial_3 v^2 + \bar{\Gamma}^2_{23}v^2 \ ,$$
$$\bar{\nabla}_3 v^3 = \partial_3 v^3 \ .$$

Now, with relative ease, one can express the governing equations as

$$(\lambda + \mu)\partial^2\left(\bar{\nabla}_i v^i\right) + \mu\bar{\Delta}v^2 = 0 \ , \tag{1}$$
$$(\lambda + \mu)\partial^3\left(\bar{\nabla}_i v^i\right) + \mu\bar{\Delta}v^3 = 0 \ , \tag{2}$$
$$(\lambda + \mu)\partial^2\left(\bar{\nabla}_i \delta v^i\right) + \mu\bar{\Delta}\delta v^2 = 0 \ , \tag{3}$$
$$(\lambda + \mu)\partial^3\left(\bar{\nabla}_i \delta v^i\right) + \mu\bar{\Delta}\delta v^3 = 0 \ . \tag{4}$$

Eliminating x^1 dependency one can express the remaining boundaries as

$$\partial\Omega^{\mathsf{New}} = \partial\Omega^{\mathsf{New}}_0 \cup \partial\Omega^{\mathsf{New}}_f \cup \overline{\partial\Omega}^{\mathsf{New}}_{T_0} \cup \overline{\partial\Omega}^{\mathsf{New}}_{T_{\mathsf{max}}} \ ,$$

where

$$\partial\Omega_0^{\text{New}} = \{(-\frac{1}{2}\pi, \frac{1}{2}\pi) \times \{0\}\} \ ,$$
$$\partial\Omega_f^{\text{New}} = \{(-\frac{1}{2}\pi, \frac{1}{2}\pi) \times \{h\}\} \ ,$$
$$\partial\Omega_{T_0}^{\text{New}} = \{\{-\frac{1}{2}\pi\} \times (0, h)\} \ ,$$
$$\partial\Omega_{T_{\max}}^{\text{New}} = \{\{\frac{1}{2}\pi\} \times (0, h)\} \ .$$

Thus, the boundary conditions reduce to

$$v^3|_{\overline{\partial\Omega_0^{\text{New}}}} = 0 \text{ (zero-Dirichlet)} \ , \tag{5}$$

$$\left[(\lambda + 2\mu)\left(\partial_2 v^2 + \bar{\Gamma}_{22}^2 v^2 + \bar{\Gamma}_{23}^2 v^3\right) + \lambda \partial_3 v^3\right]|_{\overline{\partial\Omega_{T_0}^{\text{New}}}} = \tau_0 \text{ (traction)} \ , \tag{6}$$

$$\left[(\lambda + 2\mu)\left(\partial_2 v^2 + \bar{\Gamma}_{22}^2 v^2 + \bar{\Gamma}_{23}^2 v^3\right) + \lambda \partial_3 v^3\right]|_{\overline{\partial\Omega_{T_{\max}}^{\text{New}}}} = \tau_{\max} \text{ (traction)} \ , \tag{7}$$

$$\left[(\bar{\psi}_2)^2 \partial_3 v^2 + \partial_2 v^3\right]|_{\partial\Omega_f^{\text{New}} \cup \partial\Omega_{T_0}^{\text{New}} \cup \partial\Omega_{T_{\max}}^{\text{New}}} = 0 \text{ (zero-Robin)} \ , \tag{8}$$

$$\left[\lambda\left(\partial_2 v^2 + \bar{\Gamma}_{22}^2 v^2 + \bar{\Gamma}_{23}^2 v^3\right) + (\lambda + 2\mu)\partial_3 v^3\right]|_{\overline{\partial\Omega_f^{\text{New}}}} = 0 \text{ (zero-Robin)} \ , \tag{9}$$

$$\delta v^2|_{\overline{\partial\Omega_f^{\text{New}}} \cup \partial\Omega_{T_0}^{\text{New}} \cup \partial\Omega_{T_{\max}}^{\text{New}}} = 0 \ ,$$

$$\delta v^3|_{\partial\Omega^{\text{New}}} = 0 \ .$$

Now, with some more calculations, one can find the fiction laws governing the boundary conditions at the boundary $\overline{\partial\Omega_0^{\text{New}}}$, which are:

If $\bar{\psi}_2 |v^2||_{\partial\Omega_0^{\text{New}}} \geq \epsilon$, then

$$\left[\mu \bar{\psi}_2 \partial_3 v^2 + \nu_F \text{sign}(v^2) T_3^3(\boldsymbol{v})\right]|_{\partial\Omega_0^{\text{New}}} = 0 \ ; \tag{10}$$

If $\bar{\psi}_2 |v^2||_{\partial\Omega_0^{\text{New}}} < \epsilon$, then

$$\begin{aligned} \big[\mu \bar{\psi}_2 \partial_3 \delta v^2 &+ \nu_F \epsilon^{-1} \bar{\psi}_2 v^2 T_3^3(\delta \boldsymbol{v}) + \nu_F \epsilon^{-1} \bar{\psi}_2 \delta v^2 T_3^3(\boldsymbol{v}) \\ &+ \mu \bar{\psi}_2 \partial_3 v^2 + \nu_F \epsilon^{-1} \bar{\psi}_2 v^2 T_3^3(\boldsymbol{v})\big]|_{\partial\Omega_0^{\text{New}}} = 0 \ , \end{aligned} \tag{11}$$

where $T_3^3(\boldsymbol{v}) = \lambda(\partial_2 v^2 + \bar{\Gamma}_{22}^2 v^2 + \bar{\Gamma}_{23}^2 v^3) + (\lambda + 2\mu)\partial_3 v^3$.

Now, we use the second-order-accurate finite-difference method in conjunction with Newton's method for nonlinear systems, i.e. given that $\boldsymbol{v}_m$ and $\delta\boldsymbol{v}_m$ are m^{th} iterative solutions of the problem obtained by the finite-difference method, we assert that $\boldsymbol{v}_{m+1} = \boldsymbol{v}_m + \delta\boldsymbol{v}_m$ is the updated solution, and we follow this iterative scheme such that $\delta\boldsymbol{v}_m$ converges to zero in the limit $m \to \infty$. Another issue we must tackle is the discretisation of the (reduced two-dimensional) domain. As we are dealing with curvilinear coordinates, there is an inherit grid dependence. To be precise, it is approximately $\psi_0 \Delta x^2 \leq \Delta x^3$, $\forall \ \psi_0 \in \{\bar{\psi}_2(x^2, x^3) \mid x^2 \in [-\frac{1}{2}\pi, \frac{1}{2}\pi]$ and $x^3 \in [0, h]\}$, where Δx^j is a small increment in x^j direction. For our purposes we use $\Delta x^2 = \frac{1}{N-1}$ and $\psi_0 = \bar{\psi}_2(\frac{1}{4}\pi, h)$, where $N = 250$. Finally, we must define a terminating condition. For this we terminate our iterating process given that the condition $|1 - (||\boldsymbol{v}_m||_{\ell^2} + ||\delta\boldsymbol{v}_m||_{\ell^2})^{-1}(||\boldsymbol{v}_{m+1}||_{\ell^2} + ||\delta\boldsymbol{v}_{m+1}||_{\ell^2})| < 10^{-10}$ is satisfied, where $|\boldsymbol{v}|_{\ell^2} = ([\text{norm}(v^2, 2)]^2 + [\text{norm}(v^3, 2)]^2)^{\frac{1}{2}}$ and $\text{norm}(\cdot, 2)$ is Matlab 2-norm of matrix [8].

Finally, let `u1`$= v^2$, `u2`$= v^3$, `du1`$= \delta v^2$, `du2`$= \delta v^3$, `a`$= b$, `b`$= a$, `Thickness1`$= h$, `Stress1`$= \tau_0$, `Stress2`$= \tau_{\mathsf{max}}$, `Youngs1`$= E$, `Poisson1`$= \nu$, `NN`$= N$, `Mu`$= \nu_F$ and `epsi`$= \varepsilon$. Thus, we find:

```
function OdenProb1
format long
%% Kikuchi and Oden's Model
% Overlying elastic body on an elastic prism with a variable elliptical cross section
% Contact angle is [0,pi]
% Static friction case

%% INITIAL PARIMITERS
a = 1; % Radius at \theta = 0.5*pi
b = 1; % Radius at \theta = 0

Thickness1 = 0.5; % Thickness of the overlying body

Stress1 = 1; % Applied stress at \theta = 0
Stress2 = 1; % Applied stress at \theta = pi

Youngs1 = 1000; % Young's modulus of the overlying body
Poisson1 = 0.25; % Poisson's ratio of the overlying body

NN = 250; % Azimuthal grid points
error =  10^(-10); % Terminating error

Mu = 0.5; % Coefficient of friction
epsi = 10^(-5); % Regularisation parameter

%% DO NOT CHANGE!
qq1 = sqrt((a^2+b^2)/2) + 2*a*b*Thickness1/(a^2+b^2);

q1 = Thickness1/(qq1*pi);

m = NN; % Azimuthal grid points
n1 = round(q1*NN-q1+1); % Radial grid points of the overlying body

L1  = Poisson1*Youngs1/((1+Poisson1)*(1-2*Poisson1));
M1 = 0.5*Youngs1/(1+Poisson1);

NNN = NN^2;
errr = error*NNN;
p = 2;

dx1 = pi/(m-1);
dx2 = Thickness1/(n1-1);

Idx1 = 1/dx1;
Idx2 = 1/dx2;

Iepsi = 1/epsi;
```

```
Id1 = 0.5*Idx1;
Id11 = Idx1^2;

Id2 = 0.5*Idx2;
Id22 = Idx2^2;

Id12 = Id1*Id2;

aa = zeros(2,2);

u1 = zeros(m,n1); % Azimuthal displacement of the overlying body
u2 = zeros(m,n1); % Radial displacement of the overlying body
du1 = zeros(m,n1); % Perturbed azimuthal displacement of the overlying body
du2 = zeros(m,n1); % Perturbed radial displacement of the overlying body

X = zeros(m,n1);
IX = zeros(m,n1);

L111 = zeros(m,n1);
L112 = zeros(m,n1);

JM11 = zeros(m,n1);
JM12 = zeros(m,n1);
JM21 = zeros(m,n1);
JM22 = zeros(m,n1);

J1S11 = zeros(1,n1);
J1S12 = zeros(1,n1);
J1S21 = zeros(1,n1);
J1S22 = zeros(1,n1);

JmS11 = zeros(1,n1);
JmS12 = zeros(1,n1);
JmS21 = zeros(1,n1);
JmS22 = zeros(1,n1);

JBB11 = zeros(m,1);
JBB12 = zeros(m,1);
JBB21 = zeros(m,1);
JBB22 = zeros(m,1);

A1d22 = zeros(m,n1);
A1d1 = zeros(m,n1);
A1d2 = zeros(m,n1);
A2d1 = zeros(m,n1);

B1d12 = zeros(m,n1);
B2d11 = zeros(m,n1);
B2d22 = zeros(m,n1);
B1d1 = zeros(m,n1);
```

```
B1d2 = zeros(m,n1);
B2d1 = zeros(m,n1);
B2d2 = zeros(m,n1);

Slip = zeros(m,1);

T1 = Stress1;
T2 = Stress2;

A1d11 = (L1+2*M1);
A2d12 = (L1+M1);

%% Curvature terms of the overlying body
for i = 1:m

    x1 = (i-1)*dx1;

    for j = 1:n1

        x2 = (j-1)*dx2;

        alpha2 = (b*sin(x1))^2 + (a*cos(x1))^2;
        alpha = sqrt(alpha2);
        Ialpha = 1/alpha;
        Ialpha2 = 1/alpha2;
        alphad1 = 0.5*(b^2-a^2)*sin(2*x1)*Ialpha;
        alphad11 = ((b^2-a^2)*cos(2*x1) - alphad1^2)*Ialpha;

        X(i,j) = alpha + a*b*Ialpha2*x2;
        IX(i,j) = 1/X(i,j);
        Xd1 = alphad1*(1-2*a*b*x2*Ialpha*Ialpha2);
        Xd11 = alphad11*(1-2*a*b*x2*Ialpha*Ialpha2) + 6*a*b*x2*(alphad1*Ialpha2)^2;
        Xd2 = a*b*Ialpha2;
        Xd12 = -2*a*b*alphad1*Ialpha*Ialpha2;

        L111(i,j) = Xd1*IX(i,j);
        L112(i,j) = Xd2*IX(i,j);
        L1111 = Xd11*IX(i,j)-L111(i,j)^2;
        L1112 = Xd12*IX(i,j)-L111(i,j)*L112(i,j);
        L1122 = -L112(i,j)^2;

        A1d22(i,j) = M1*X(i,j)^2;
        A1d1(i,j) = (L1+2*M1)*L111(i,j);
        A1d2(i,j) = 3*M1*L112(i,j)*X(i,j)^2;
        A2d1(i,j) = (L1+3*M1)*L112(i,j);
        A1 = (L1+2*M1)*L1111;
        A2 = (L1+2*M1)*L1112;

        B1d12(i,j) = (L1+M1)*X(i,j);
        B2d11(i,j) = M1*IX(i,j);
        B2d22(i,j) = (L1+2*M1)*X(i,j);
```

```
        B1d1(i,j) = - 2*M1*L112(i,j)*X(i,j);
        B1d2(i,j) = (L1+M1)*L111(i,j)*X(i,j);
        B2d1(i,j) = - M1*L111(i,j)*IX(i,j);
        B2d2(i,j) = (L1+2*M1)*L112(i,j)*X(i,j);
        B1 = L1*L1112*X(i,j) - 2*M1*L111(i,j)*L112(i,j)*X(i,j);
        B2 = L1*L1122*X(i,j) - 2*M1*L112(i,j)*L112(i,j)*X(i,j);

        aa(1,1) = 2*A1d11*Id11      + 2*A1d22(i,j)*Id22 - A1;
        aa(1,2) = - A2;
        aa(2,1) = - B1;
        aa(2,2) = 2*B2d11(i,j)*Id11 + 2*B2d22(i,j)*Id22 - B2;

        J = inv(aa);

        JM11(i,j) = J(1,1);
        JM12(i,j) = J(1,2);
        JM21(i,j) = J(2,1);
        JM22(i,j) = J(2,2);

    end
end

for i = 1:m

    aa(1,1) = -3*Id2;
    aa(1,2) = 0;
    aa(2,1) = - L1*L111(i,n1);
    aa(2,2) = - L1*L112(i,n1) - 3*Id2*(L1+2*M1);

    J = inv(aa);

    JBB11(i) = J(1,1);
    JBB12(i) = J(1,2);
    JBB21(i) = J(2,1);
    JBB22(i) = J(2,2);

end

for j = 1:n1

    aa(1,1) = 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(1,j);
    aa(1,2) = - (L1+2*M1)*L112(1,j);
    aa(2,1) = 0;
    aa(2,2) = 3*Id1;

    J = inv(aa);

    J1S11(j) = J(1,1);
    J1S12(j) = J(1,2);
    J1S21(j) = J(2,1);
    J1S22(j) = J(2,2);
```

```
    aa(1,1) =  - 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(m,j);
    aa(1,2) = - (L1+2*M1)*L112(m,j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Id1;

    J = inv(aa);

    JmS11(j) = J(1,1);
    JmS12(j) = J(1,2);
    JmS21(j) = J(2,1);
    JmS22(j) = J(2,2);

end

J11BBB11 = 1/( 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(1,1));
Jm1BBB11 = 1/(-3*Id1*(L1+2*M1) - (L1+2*M1)*L111(m,1));

aa(1,1) = 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(1,n1);
aa(1,2) = -3*Id2*L1 - (L1+2*M1)*L112(1,n1);
aa(2,1) = 3*Id1*L1 - L1*L111(1,n1);
aa(2,2) = - 3*(L1+2*M1)*Id2 - L1*L112(1,n1);

J = inv(aa);

J1nBB11 = J(1,1);
J1nBB12 = J(1,2);
J1nBB21 = J(2,1);
J1nBB22 = J(2,2);

aa(1,1) = - 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(m,n1);
aa(1,2) = - 3*Id2*L1 - (L1+2*M1)*L112(m,n1);
aa(2,1) = - 3*Id1*L1 - L1*L111(m,n1);
aa(2,2) = - 3*(L1+2*M1)*Id2 - L1*L112(m,n1);

J = inv(aa);

JmnBB11 = J(1,1);
JmnBB12 = J(1,2);
JmnBB21 = J(2,1);
JmnBB22 = J(2,2);

%% Main code
while errr < p

    pp = norm(u1,2) + norm(u2,2) + norm(du1,2) + norm(du2,2);

    for k = 1:NNN

        %% Corners
        u1d1 = (4*u1(2,1)-u1(3,1))*Id1;
```

```
u2d2 = (4*u2(1,2)-u2(1,3))*Id2;

Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T1; % equation (6)
u1(1,1)  = J11BBB11*Xu1; % equation (8)

u1d1 = -(4*u1(m-1,1)-u1(m-2,1))*Id1;
u2d2 = (4*u2(m,2)-u2(m,3))*Id2;

Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T2; % equation (7)
u1(m,1)  = Jm1BBB11*Xu1; % equation (8)

u1d1 = (4*u1(2,n1)-u1(3,n1))*Id1;
u2d2 = -(4*u2(1,n1-1)-u2(1,n1-2))*Id2;

Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T1; % equation (6)
Xu2 = L1*u1d1 + (L1+2*M1)*u2d2; % equation (8)

u1(1,n1) = J1nBB11*Xu1 + J1nBB12*Xu2;
u2(1,n1) = J1nBB21*Xu1 + J1nBB22*Xu2;

u1d1 = -(4*u1(m-1,n1)-u1(m-2,n1))*Id1;
u2d2 = -(4*u2(m,n1-1)-u2(m,n1-2))*Id2;

Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T2; % equation (7)
Xu2 = L1*u1d1 + (L1+2*M1)*u2d2; % equation (8)

u1(m,n1) = JmnBB11*Xu1 + JmnBB12*Xu2;
u2(m,n1) = JmnBB21*Xu1 + JmnBB22*Xu2;

for i = 2:m-1

    %% Contact region % equation (5) and (9)
    u1d1 = (u1(i+1,1)-u1(i-1,1))*Id1;
    u2d1 = (u2(i+1,1)-u2(i-1,1))*Id1;
    u1d2 = (4*u1(i,2)-u1(i,3))*Id2;
    u2d2 = (4*u2(i,2)-u2(i,3))*Id2;

    du1d1 = (du1(i+1,1)-du1(i-1,1))*Id1;
    du2d1 = (du2(i+1,1)-du2(i-1,1))*Id1;
    du1d2 = (4*du1(i,2)-du1(i,3))*Id2;
    du2d2 = (4*du2(i,2)-du2(i,3))*Id2;

    Xu1 = M1*(X(i,1)*u1d2 + IX(i,1)*u2d1);
    T12u = Xu1 - 3*M1*X(i,1)*u1(i,1)*Id2;
    T22u = L1*(u1d1+L112(i,1)*u2(i,1)) ...
        + (L1+2*M1)*u2d2 - 3*(L1+2*M1)*u2(i,1)*Id2;
    T22  = T22u + L1*L111(i,1)*u1(i,1);

    Xdu1 = M1*(X(i,1)*du1d2 + IX(i,1)*du2d1);
    T22du  = L1*du1d1 + (L1+2*M1)*du2d2;
```

```
    delta =  X(i,1)*u1(i,1);
    absdelta = abs(delta);

    %% Limiting-equilibrium boundary
    if ~(absdelta < epsi)

        XXu1 = (Xu1 + Mu*sign(delta)*T22u); % equation (10)
        B1u1 = 1/(3*X(i,1)*M1*Id2-Mu*sign(delta)*L1*L111(i,1));

        u1(i,1)   = XXu1*B1u1;
        du1(i,1)   = 0;

    end

    %% Bounded boundary
    if absdelta < epsi

        Idu1 =  1/(3*X(i,1)*M1*Id2 - Iepsi*Mu*X(i,1)*T22 ...
            - Iepsi*Mu*delta*L1*L111(i,1));
        Xdu1 = Xdu1 + Iepsi*Mu*delta*T22du ...
            + T12u + Iepsi*Mu*delta*T22; % equation (11)

        du1(i,1)   = Xdu1*Idu1;

    end

    %% Stress-free boundary of the overlying body
    u1d1 = (u1(i+1,n1)-u1(i-1,n1))*Id1;
    u1d2 = -(4*u1(i,n1-1)-u1(i,n1-2))*Id2;
    u2d1 = (u2(i+1,n1)-u2(i-1,n1))*Id1;
    u2d2 = -(4*u2(i,n1-1)-u2(i,n1-2))*Id2;

    Xu1 = u1d2 + (IX(i,n1)^2)*u2d1; % equation (8)
    Xu2 = L1*u1d1 + (L1+2*M1)*u2d2; % equation (9)

    u1(i,n1) = JBB11(i)*Xu1 + JBB12(i)*Xu2;
    u2(i,n1) = JBB21(i)*Xu1 + JBB22(i)*Xu2;

end

%% Stressed boundaries
for j = 2:n1-1

    u1d1 = (4*u1(2,j)-u1(3,j))*Id1;
    u1d2 = (u1(1,j+1)-u1(1,j-1))*Id2;
    u2d1 = (4*u2(2,j)-u2(3,j))*Id1;
    u2d2 = (u2(1,j+1)-u2(1,j-1))*Id2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T1; % equation (6)
    Xu2 = u2d1 + (X(1,j)^2)*u1d2; % equation (8)
```

```
    u1(1,j) = J1S11(j)*Xu1 + J1S12(j)*Xu2;
    u2(1,j) = J1S21(j)*Xu1 + J1S22(j)*Xu2;

    u1d1 = -(4*u1(m-1,j)-u1(m-2,j))*Id1;
    u1d2 = (u1(m,j+1)-u1(m,j-1))*Id2;
    u2d1 = -(4*u2(m-1,j)-u2(m-2,j))*Id1;
    u2d2 = (u2(m,j+1)-u2(m,j-1))*Id2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T2; % equation (7)
    Xu2 = u2d1 + (X(m,j)^2)*u1d2; % equation (8)

    u1(m,j) = JmS11(j)*Xu1 + JmS12(j)*Xu2;
    u2(m,j) = JmS21(j)*Xu1 + JmS22(j)*Xu2;

end

%% Governing equations of the foundation
for i = 2:m-1

    for j = 2:n1-1
        u1d11 = (u1(i+1,j)+u1(i-1,j))*Id11;
        u1d22 = (u1(i,j+1)+u1(i,j-1))*Id22;
        u1d12 = (u1(i+1,j+1)-u1(i+1,j-1)-u1(i-1,j+1)+u1(i-1,j-1))*Id12;
        u1d1 = (u1(i+1,j)-u1(i-1,j))*Id1;
        u1d2 = (u1(i,j+1)-u1(i,j-1))*Id2;

        u2d11 = (u2(i+1,j)+u2(i-1,j))*Id11;
        u2d22 = (u2(i,j+1)+u2(i,j-1))*Id22;
        u2d12 = (u2(i+1,j+1)-u2(i+1,j-1)-u2(i-1,j+1)+u2(i-1,j-1))*Id12;
        u2d1 = (u2(i+1,j)-u2(i-1,j))*Id1;
        u2d2 = (u2(i,j+1)-u2(i,j-1))*Id2;

        Xu1 = A1d11*u1d11      + A1d22(i,j)*u1d22 + A2d12*u2d12      ...
            + A1d1(i,j)*u1d1 + A1d2(i,j)*u1d2 + A2d1(i,j)*u2d1; % equation (1)
        Xu2 = B2d11(i,j)*u2d11 + B2d22(i,j)*u2d22 + B1d12(i,j)*u1d12 ...
            + B1d1(i,j)*u1d1 + B1d2(i,j)*u1d2 ...
            + B2d1(i,j)*u2d1 + B2d2(i,j)*u2d2; % equation (2)

        u1(i,j) = JM11(i,j)*Xu1 + JM12(i,j)*Xu2;
        u2(i,j) = JM21(i,j)*Xu1 + JM22(i,j)*Xu2;

        du1d11 = (du1(i+1,j)+du1(i-1,j))*Id11;
        du1d22 = (du1(i,j+1)+du1(i,j-1))*Id22;
        du1d12 = (du1(i+1,j+1)-du1(i+1,j-1)-du1(i-1,j+1)+du1(i-1,j-1))*Id12;
        du1d1 = (du1(i+1,j)-du1(i-1,j))*Id1;
        du1d2 = (du1(i,j+1)-du1(i,j-1))*Id2;

        du2d11 = (du2(i+1,j)+du2(i-1,j))*Id11;
        du2d22 = (du2(i,j+1)+du2(i,j-1))*Id22;
        du2d12 = (du2(i+1,j+1)-du2(i+1,j-1)-du2(i-1,j+1)+du2(i-1,j-1))*Id12;
        du2d1 = (du2(i+1,j)-du2(i-1,j))*Id1;
```

```
                du2d2 = (du2(i,j+1)-du2(i,j-1))*Id2;

                Xdu1 = A1d11*du1d11      + A1d22(i,j)*du1d22 + A2d12*du2d12      ...
                    + A1d1(i,j)*du1d1 + A1d2(i,j)*du1d2 + A2d1(i,j)*du2d1; % equation (3)
                Xdu2 = B2d11(i,j)*du2d11 + B2d22(i,j)*du2d22 + B1d12(i,j)*du1d12 ...
                    + B1d1(i,j)*du1d1 + B1d2(i,j)*du1d2 ...
                    + B2d1(i,j)*du2d1 + B2d2(i,j)*du2d2; % equation (4)

                du1(i,j) = JM11(i,j)*Xdu1 + JM12(i,j)*Xdu2;
                du2(i,j) = JM21(i,j)*Xdu1 + JM22(i,j)*Xdu2;

            end
        end

        %% Newton's method
        u1(:,:) = du1(:,:) + u1(:,:);
        u2(:,:) = du2(:,:) + u2(:,:);

    end

    %% Terminating condition
    ppp = norm(u1,2) + norm(u2,2) + norm(du1,2) + norm(du2,2);
    p = abs(1- ppp/pp);

end

%% Stick and slip region
for i = 1:m
    delta = X(i,1)*u1(i,1);
    absdelta = abs(delta);

    if ~(absdelta < epsi)
        Slip(i) = sign(delta);
    end
end
slip = sum(Slip)/NN; % Slip

%% Save data file
filename = 'OdenProb1.mat';
save(filename)
```

1.1 Benchmark Model

In this section we present a benchmark numerical code for the model that we introduced in this chapter, i.e. CH2.m. To do so we consider the special case $a = b = 2$, i.e. we consider polar coordinates.

Now, let ww1= v^2, ww2= v^3, dww1= δv^2, dww2= δv^3, h= h, E1= E, Nu1= ν, NN= N, Mu= ν_F, epsi= ε, T0= τ_0 and Tmax= τ_{max}. Thus, we find:

```
function CH2
format long
%% Benchmark Model
% Overlying elastic body on an rigid cylinder: friction case
% Contact angle is [0,pi]

%% INITIAL PARIMITERS
NN = 250; % Azimuthal grid points
error = 10^(-10); % Terminating error

E1 = 1000; % Young's modulus of the overlying body
Nu1 = 0.25; % Poisson's ratio of the overlying bod

h = 0.5; % Thickness of the overlying body
a = 1; % Outer radius

Mu = 0.5; % Coefficient of friction
epsi = 10^(-5); % Regularisation parameter

T0 = 1; % Applied stress at \theta = 0
Tmax = 1; % Applied stress at \theta = pi

%% DO NOT CHANGE!
L1 = Nu1*E1/((1+Nu1)*(1-2*Nu1));
M1 = 0.5*E1/(1+Nu1);

NNN = NN^2;
errr = NNN*error;

ah = a+h;
Iah = 1/ah;
Ia = 1/a;
Iepsi = 1/epsi;

m = NN; % Azimuthal grid points
q2 = h/(ah*pi);
n2 = round(q2*NN-q2+1); % Radial grid points of the overlying body

dx1 = pi/(m-1); % Azimuthal grid spacing
dX2 = h/(n2-1); % Radial grid spacing of the overlying body

Idx1 = 1/(2*dx1);
Idx11 = (1/dx1)^2;

ww1 = zeros(m,n2); % Azimuthal displacement of the overlying body
ww2 = zeros(m,n2); % Radial displacement of the overlying body

dww1 = zeros(m,n2); % Perturbed azimuthal displacement of the overlying body
dww2 = zeros(m,n2); % Perturbed radial displacement of the overlying body

Slip = zeros(m,1);
```

```
aa = zeros(2,2);

rr = zeros(1,n2);
Irr = zeros(1,n2);

aa1d11 = zeros(1,n2);
aa1d22 = zeros(1,n2);
aa1d2 = zeros(1,n2);
aa2d12 = zeros(1,n2);
aa2d1 = zeros(1,n2);

bb1d12 = zeros(1,n2);
bb1d1 = zeros(1,n2);
bb2d11 = zeros(1,n2);
bb2d22 = zeros(1,n2);
bb2d2 = zeros(1,n2);

Iw1 = zeros(1,n2);
Iw2 = zeros(1,n2);

B1w11 = zeros(1,n2);
B1w12 = zeros(1,n2);
B1w21 = zeros(1,n2);
B1w22 = zeros(1,n2);

Bmw11 = zeros(1,n2);
Bmw12 = zeros(1,n2);
Bmw21 = zeros(1,n2);
Bmw22 = zeros(1,n2);

%% Curvature terms
IdX2 = 1/(2*dX2);
IdX22 = (1/dX2)^2;

IdX12 = Idx1*IdX2;

Bn2ww1 = 1/(-3*IdX2*ah^2);
Bn2ww2 = 1/(-3*(L1+2*M1)*IdX2 - L1*Iah);

B1ww1 = 1/(3*a*M1*IdX2);

aa(1,1) = 3*(L1+2*M1)*Idx1;
aa(1,2) = - (L1+2*M1)*Iah - 3*L1*IdX2;
aa(2,1) = 3*L1*Idx1;
aa(2,2) = - L1*Iah - 3*(L1+2*M1)*IdX2;

B1n2ww = inv(aa);

aa(1,1) = - 3*(L1+2*M1)*Idx1;
aa(1,2) = - (L1+2*M1)*Iah - 3*L1*IdX2;
```

```
aa(2,1) = - 3*L1*Idx1;
aa(2,2) = - L1*Iah - 3*(L1+2*M1)*IdX2;

Bmn2ww = inv(aa);

B11ww = 1/(3*(L1+2*M1)*Idx1);
Bm1ww = 1/(- 3*(L1+2*M1)*Idx1);

for j = 1:n2

    rr(j) = a + (j-1)*dX2;
    Irr(j) = 1/rr(j);

    aa1d11(j) = (L1+2*M1);
    aa1d22(j) = M1*rr(j)^2;
    aa1d2(j) = 3*M1*rr(j);
    aa2d12(j) = (L1+M1);
    aa2d1(j) = (L1+3*M1)*Irr(j);

    bb1d12(j) = (L1+M1)*rr(j);
    bb1d1(j) = -2*M1;
    bb2d11(j) = M1*Irr(j);
    bb2d22(j) = (L1+2*M1)*rr(j);
    bb2d2(j) = (L1+2*M1);
    bb2 = -(L1+2*M1)*Irr(j);

    Iw1(j) = 1/(2*aa1d11(j)*Idx11 + 2*aa1d22(j)*IdX22);
    Iw2(j) = 1/(2*bb2d11(j)*Idx11 + 2*bb2d22(j)*IdX22 - bb2);


    aa(1,1) = 3*Idx1*(L1+2*M1);
    aa(1,2) = - (L1+2*M1)*Irr(j);
    aa(2,1) = 0;
    aa(2,2) = 3*Idx1;

    J = inv(aa);

    B1w11(j) = J(1,1);
    B1w12(j) = J(1,2);
    B1w21(j) = J(2,1);
    B1w22(j) = J(2,2);

    aa(1,1) = - 3*Idx1*(L1+2*M1);
    aa(1,2) = - (L1+2*M1)*Irr(j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Idx1;

    J = inv(aa);

    Bmw11(j) = J(1,1);
    Bmw12(j) = J(1,2);
```

```
    Bmw21(j) = J(2,1);
    Bmw22(j) = J(2,2);

end

p = 2;

%% Main code
while errr < p

    pp = norm(ww1,2) + norm(ww2,2) + norm(dww1,2) + norm(dww2,2);

    for k = 1:NNN

        %% Corners
        ww1d1 = (4*ww1(2,n2)-ww1(3,n2))*Idx1;
        ww2d2 = -(4*ww2(1,n2-1)-ww2(1,n2-2))*IdX2;

        Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - T0;
        Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

        ww1(1,n2) = B1n2ww(1,1)*Xww1 + B1n2ww(1,2)*Xww2;
        ww2(1,n2) = B1n2ww(2,1)*Xww1 + B1n2ww(2,2)*Xww2;

        ww1d1 = -(4*ww1(m-1,n2)-ww1(m-2,n2))*Idx1;
        ww2d2 = -(4*ww2(m,n2-1)-ww2(m,n2-2))*IdX2;

        Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - Tmax;
        Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

        ww1(m,n2) = Bmn2ww(1,1)*Xww1 + Bmn2ww(1,2)*Xww2;
        ww2(m,n2) = Bmn2ww(2,1)*Xww1 + Bmn2ww(2,2)*Xww2;

        ww1d1 = (4*ww1(2,1)-ww1(3,1))*Idx1;
        ww2d2 = (4*ww2(1,2)-ww2(1,3))*IdX2;

        Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - T0;
        ww1(1,1) = B11ww*Xww1;

        ww1d1 = -(4*ww1(m-1,1)-ww1(m-2,1))*Idx1;
        ww2d2 = (4*ww2(m,2)-ww2(m,3))*IdX2;

        Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - Tmax;
        ww1(m,1) = Bm1ww*Xww1;

        for i = 2:m-1

            %% Stress-free boundary of the overlying body
            ww1d1 = (ww1(i+1,n2)-ww1(i-1,n2))*Idx1;
            ww2d1 = (ww2(i+1,n2)-ww2(i-1,n2))*Idx1;
            ww1d2 = -(4*ww1(i,n2-1)-ww1(i,n2-2))*IdX2;
```

```
    ww2d2 = -(4*ww2(i,n2-1)-ww2(i,n2-2))*IdX2;

    Xww1 = ww1d2*ah^2 + ww2d1;
    Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

    ww1(i,n2) = Bn2ww1*Xww1;
    ww2(i,n2) = Bn2ww2*Xww2;

    %% Friction boundary
    ww1d1 = (ww1(i+1,1)-ww1(i-1,1))*Idx1;
    ww2d1 = (ww2(i+1,1)-ww2(i-1,1))*Idx1;
    ww1d2 = (4*ww1(i,2)-ww1(i,3))*IdX2;
    ww2d2 = (4*ww2(i,2)-ww2(i,3))*IdX2;

    dww1d1 = (dww1(i+1,1)-dww1(i-1,1))*Idx1;
    dww2d1 = (dww2(i+1,1)-dww2(i-1,1))*Idx1;
    dww1d2 = (4*dww1(i,2)-dww1(i,3))*IdX2;
    dww2d2 = (4*dww2(i,2)-dww2(i,3))*IdX2;

    Xww1 = M1*(a*ww1d2 + Ia*ww2d1);

    T12 = Xww1 - 3*a*M1*ww1(i,1)*IdX2;
    T22 = L1*ww1d1 + (L1+2*M1)*ww2d2;

    Xdww1 = M1*(a*dww1d2 + Ia*dww2d1);
    T22dw  = L1*dww1d1 + (L1+2*M1)*dww2d2;

    delta =  a*ww1(i,1);
    absdelta = abs(delta);

    %% Limiting-equilibrium boundary
    if ~(absdelta < epsi)

        ww1(i,1)  = (Xww1 + Mu*sign(delta)*T22)*B1ww1;
        dww1(i,1)  = 0;

    end

    %% Bounded boundary
    if absdelta < epsi

        Idww1 = 1/(3*a*M1*IdX2 - Iepsi*Mu*a*T22);
        Xdww1 = Xdww1 + Iepsi*Mu*delta*T22dw + T12 + Iepsi*Mu*delta*T22;
        dww1(i,1)  = Xdww1*Idww1;

    end

end

%% Stressed boundary of the overlying body
for j = 2:n2-1
```

```
    ww1d1 = (4*ww1(2,j)-ww1(3,j))*Idx1;
    ww2d1 = (4*ww2(2,j)-ww2(3,j))*Idx1;
    ww1d2 = (ww1(1,j+1)-ww1(1,j-1))*IdX2;
    ww2d2 = (ww2(1,j+1)-ww2(1,j-1))*IdX2;

    Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - T0;
    Xww2 = ww2d1 + (rr(j)^2)*ww1d2;

    ww1(1,j) = B1w11(j)*Xww1 + B1w12(j)*Xww2;
    ww2(1,j) = B1w21(j)*Xww1 + B1w22(j)*Xww2;

    ww1d1 = -(4*ww1(m-1,j)-ww1(m-2,j))*Idx1;
    ww2d1 = -(4*ww2(m-1,j)-ww2(m-2,j))*Idx1;
    ww1d2 = (ww1(m,j+1)-ww1(m,j-1))*IdX2;
    ww2d2 = (ww2(m,j+1)-ww2(m,j-1))*IdX2;

    Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - Tmax;
    Xww2 = ww2d1 + (rr(j)^2)*ww1d2;

    ww1(m,j) = Bmw11(j)*Xww1 + Bmw12(j)*Xww2;
    ww2(m,j) = Bmw21(j)*Xww1 + Bmw22(j)*Xww2;

end

%% Governing equations of the overlying body
for i = 2:m-1

    for j = 2:n2-1

        ww1d1 = (ww1(i+1,j)-ww1(i-1,j))*Idx1;
        ww1d2 = (ww1(i,j+1)-ww1(i,j-1))*IdX2;
        ww2d1 = (ww2(i+1,j)-ww2(i-1,j))*Idx1;
        ww2d2 = (ww2(i,j+1)-ww2(i,j-1))*IdX2;

        ww1d11 = (ww1(i+1,j)+ww1(i-1,j))*Idx11;
        ww1d22 = (ww1(i,j+1)+ww1(i,j-1))*IdX22;
        ww2d11 = (ww2(i+1,j)+ww2(i-1,j))*Idx11;
        ww2d22 = (ww2(i,j+1)+ww2(i,j-1))*IdX22;

        ww1d12 = (ww1(i+1,j+1)-ww1(i-1,j+1)-ww1(i+1,j-1)+ww1(i-1,j-1))*IdX12;
        ww2d12 = (ww2(i+1,j+1)-ww2(i-1,j+1)-ww2(i+1,j-1)+ww2(i-1,j-1))*IdX12;

        Xww1 = aa1d11(j)*ww1d11 + aa1d22(j)*ww1d22 + aa1d2(j)*ww1d2 ...
            + aa2d12(j)*ww2d12 + aa2d1(j)*ww2d1;
        Xww2 = bb1d12(j)*ww1d12 + bb1d1(j)*ww1d1 + bb2d11(j)*ww2d11 ...
            + bb2d22(j)*ww2d22 + bb2d2(j)*ww2d2;

        ww1(i,j) = Xww1*Iw1(j);
        ww2(i,j) = Xww2*Iw2(j);
```

```
                dww1d1 = (dww1(i+1,j)-dww1(i-1,j))*Idx1;
                dww1d2 = (dww1(i,j+1)-dww1(i,j-1))*IdX2;
                dww2d1 = (dww2(i+1,j)-dww2(i-1,j))*Idx1;
                dww2d2 = (dww2(i,j+1)-dww2(i,j-1))*IdX2;

                dww1d11 = (dww1(i+1,j)+dww1(i-1,j))*Idx11;
                dww1d22 = (dww1(i,j+1)+dww1(i,j-1))*IdX22;
                dww2d11 = (dww2(i+1,j)+dww2(i-1,j))*Idx11;
                dww2d22 = (dww2(i,j+1)+dww2(i,j-1))*IdX22;

                dww1d12 = (dww1(i+1,j+1)-dww1(i-1,j+1)-dww1(i+1,j-1)+dww1(i-1,j-1))*IdX12;
                dww2d12 = (dww2(i+1,j+1)-dww2(i-1,j+1)-dww2(i+1,j-1)+dww2(i-1,j-1))*IdX12;

                Xdww1 = aa1d11(j)*dww1d11 + aa1d22(j)*dww1d22 + aa1d2(j)*dww1d2 ...
                    + aa2d12(j)*dww2d12 + aa2d1(j)*dww2d1;
                Xdww2 = bb1d12(j)*dww1d12 + bb1d1(j)*dww1d1 + bb2d11(j)*dww2d11 ...
                    + bb2d22(j)*dww2d22 + bb2d2(j)*dww2d2;

                dww1(i,j) = Xdww1*Iw1(j);
                dww2(i,j) = Xdww2*Iw2(j);

            end
        end

        %% Newton's method
        ww1(:,:) = dww1(:,:) + ww1(:,:);
        ww2(:,:) = dww2(:,:) + ww2(:,:);

    end

    %% Terminating condition
    ppp = norm(ww1,2) + norm(ww2,2) + norm(dww1,2) + norm(dww2,2);
    p = abs(1-ppp/pp);

end

%% Stick and slip region
for i = 1:m
    delta = a*ww1(i,1);
    absdelta = abs(delta);

    if ~(absdelta < epsi)
        Slip(i) = sign(delta);
    end
end
slip = sum(Slip)/NN; % Slip

%% Save data file
filename = 'CH2.mat';
save(filename)
```

2 Shells Supported by Elastic foundations: Bonded Case

In this chapter we present a numerical code for calculating the relative error of our bonded shell model and Baldelli and Bourdin's [2] model with respect to the elastic two-body contact problem (see chapter 3 of Jayawardana [4]) implemented in Matlab, i.e. `BondedCode.m`.

To proceed with this investigation we calculate the error in the energy-norm (see Jayawardana *et al.* [5]) as

$$Err_2(u^i) = \frac{\left(\sum_{\{\Delta x^2\}} ||u^i_{\text{shell}}(\Delta x^2,0) - u^i_{\text{two-body}}(\Delta x^2,0)||^2\right)^{\frac{1}{2}}}{\left(\sum_{\{\Delta x^2\}} ||u^i_{\text{two-body}}(\Delta x^2,0)||^2\right)^{\frac{1}{2}}}~, \tag{12}$$

$$Err_2(w^i) = \frac{\left(\sum_{\{\Delta x^2\}} ||w^i_{\text{Baldelli}}(\Delta x^2) - u^i_{\text{two-body}}(\Delta x^2,0)||^2\right)^{\frac{1}{2}}}{\left(\sum_{\{\Delta x^2\}} ||u^i_{\text{two-body}}(\Delta x^2,0)||^2\right)^{\frac{1}{2}}}~, \tag{13}$$

to calculate the relative error between bonded two-body's displacement field and the approximated displacement fields at the contact region.

Now, let `ShellErrorU1`$= Err_2(u^2)$, `ShellErrorU2`$= Err_2(u^3)$, `BaldelliErrorW1`$= Err_2(w^2)$ and `BaldelliErrorW2`$= Err_2(w^3)$. Thus, we find:

```
function BondedCode
%% Shells Supported by Elastic foundations: Bonded Case
% Contact angle is [0,pi]

%% Input
ar = 2; % Radius at \theta = 0.5*pi
br = 2; % Radius at \theta = 0
Th1 = 0.125; % Thickness of the overlying body
Th2 = 1; % Thickness of the foundation
EE1 = 8000; % Young's modulus of the overlying body
EE2 = 1000; % Young's modulus of the foundation
PP1 = 0.25; % Poisson's ratio of the overlying body
PP2 = 0.25; % Poisson's ratio of the foundation
N = 250; % Azimuthal grid points

%% Output
[W1] = BaldelliProb(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N);
[U1,U2] = BondedShell(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N);
[V1,V2] = TwoBody(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N);

NN = N;

ww1 = zeros(NN,1);
ww2 = zeros(NN,1);
uu1 = zeros(NN,1);
uu2 = zeros(NN,1);
```

```
ww1(:) = W1(:)-V1(:);
ww2(:) = -V2(:);
uu1(:) = U1(:)-V1(:);
uu2(:) = U2(:)-V2(:);

%% Error
BaldelliErrorW1 = norm(ww1,2)/norm(V1,2); % Baldelli azimuthal error % equation (13)
ShellErrorU1    = norm(uu1,2)/norm(V1,2); % Shell azimuthal error % equation (12)
BaldelliErrorW2 = norm(ww2,2)/norm(V2,2); % Baldelli radial error % equation (13)
ShellErrorU2    = norm(uu2,2)/norm(V2,2); % Baldelli radial error % equation (12)

%% Save data file
filename = 'BondedCode.mat';
save(filename)
```

2.1 Bonded Shell Model

In this section we present a numerical code for an example of our bonded shell model (see section 3.5 of Jayawardana [4]) implemented in Matlab, i.e. `BondedShell.m`.

To conduct numerical experiments assume that one is dealing with overlying shell with a thickness h that is bonded to an elastic foundation, where the unstrained configuration of the foundation is an infinitely long annular semi-prism characterised by the diffeomorphism $\bar{\boldsymbol{X}}(x^1,x^2,x^3) = (x^1, a\sin(x^2), b\cos(x^2))_{\mathsf{E}} + x^3(\varphi(x^2))^{-1}(0, b\sin(x^2), a\cos(x^2))_{\mathsf{E}}$, where $\varphi(x^2) = (b^2\sin^2(x^2) + a^2\cos^2(x^2))^{\frac{1}{2}}$, $x^1 \in (-\infty,\infty)$, $x^2 \in (-\frac{1}{2}\pi, \frac{1}{2}\pi)$, $x^3 \in (-H, 0)$, and a is the horizontal radius and b is the vertical radius of the upper surface. With some calculations, one finds that the metric tensor is $\boldsymbol{g} = \mathrm{diag}(1, (\bar{\psi}_2)^2, 1)$, where $\bar{\psi}_2 = \varphi(x^2) + x^3 ab(\varphi(x^2))^{-2}$, and Christoffel symbols are

$$\bar{\Gamma}^2_{22} = (\bar{\psi}_2)^{-1}\partial_2\bar{\psi}_2 \ ,$$
$$\bar{\Gamma}^2_{23} = (\bar{\psi}_2)^{-1}\partial_3\bar{\psi}_2 \ .$$

With few more calculations one finds

$$\bar{\nabla}_2 u^2 = \partial_2 u^2 + \bar{\Gamma}^2_{22}u^2 + \bar{\Gamma}^2_{23}u^3 \ ,$$
$$\bar{\nabla}_2 u^3 = \partial_2 u^3 - (\bar{\psi}_2)^2\bar{\Gamma}^2_{23}u^2 \ ,$$
$$\bar{\nabla}_3 u^2 = \partial_3 u^2 + \bar{\Gamma}^2_{23}u^2 \ ,$$
$$\bar{\nabla}_3 u^3 = \partial_3 u^3 \ ,$$

where $\boldsymbol{u} = (0, u^2(x^2,x^3), u^3(x^2,x^3))$ is the displacement field. Armed with this knowledge, one can express the governing equations of the foundation as

$$(\bar{\lambda}+\bar{\mu})\partial^2\left(\bar{\nabla}_i u^i\right) + \bar{\mu}\bar{\Delta}u^2 = 0 \ , \tag{14}$$
$$(\bar{\lambda}+\bar{\mu})\partial^3\left(\bar{\nabla}_i u^i\right) + \bar{\mu}\bar{\Delta}u^3 = 0 \ . \tag{15}$$

Now, eliminating x^1 dependency one can express the remaining boundaries as

$$\partial\Omega^{\mathsf{New}} = \bar{\omega}^{\mathsf{New}} \cup \partial\Omega^{\mathsf{New}}_0 \cup \partial\Omega^{\mathsf{New}}_f \ ,$$

$$\omega^{\text{New}} = \{(-\frac{1}{2}\pi, \frac{1}{2}\pi) \times \{-0\}\} ,$$
$$\partial\Omega_0^{\text{New}} = \{(-\frac{1}{2}\pi, \frac{1}{2}\pi) \times \{-H\}\} ,$$
$$\partial\Omega_f^{\text{New}} = \{\{-\frac{1}{2}\pi\} \times (-H, 0)\} \cup \{\{\frac{1}{2}\pi\} \times (-H, 0)\} .$$

Thus, the boundary conditions one imposes on the foundation reduce to

$$u^2|_{\overline{\partial\Omega_0^{\text{New}}}} = 0 \text{ (zero-Dirichlet)} ,$$
$$u^3|_{\overline{\partial\Omega_0^{\text{New}}}} = 0 \text{ (zero-Dirichlet)} ,$$
$$\left[(\bar{\psi}_2)^2\partial_3 u^2 + \partial_2 u^3\right]|_{\partial\Omega_f^{\text{New}}} = 0 \text{ (zero-Robin)} , \tag{16}$$
$$\left[(\bar{\lambda} + 2\bar{\mu})\partial_2 u^2 + \bar{\lambda}\left(\partial_3 u^3 + \bar{\Gamma}_{22}^2 u^2 + \bar{\Gamma}_{23}^2 u^3\right)\right]|_{\partial\Omega_f^{\text{New}}} = 0 \text{ (zero-Robin)} . \tag{17}$$

Now, consider overlying shell's unstrained configuration, which is described by the injective immersion $\boldsymbol{\sigma}(x^1, x^2) = (x^1, a\sin(x^2), b\cos(x^2))_{\mathsf{E}}$, where $x^1 \in (-\infty, \infty)$ and $x^2 \in (-\frac{1}{2}\pi, \frac{1}{2}\pi)$. With some calculations one finds that the first fundamental form tensor is $\boldsymbol{F}_{[\mathrm{I}]} = \mathrm{diag}(1, (\psi_2)^2)$, the second fundamental form tensor is $\boldsymbol{F}_{[\mathrm{II}]} = \mathrm{diag}(0, -ab(\varphi(x^2))^{-1})$ and only nonzero Christoffel symbol is $\Gamma_{22}^2 = \psi_2^{-1}\partial_2\psi_2$, where $\psi_2 = \varphi(x^2)$. With few more calculations one further finds

$$\nabla_2 u^2 = \partial_2 u^2 + \Gamma_{22}^2 u^2 ,$$
$$\epsilon_2^2(\boldsymbol{u}) = \nabla_2 u^2 - F_{[\mathrm{II}]2}{}^2 u^3 ,$$
$$\rho_2^2(\boldsymbol{u}) = \Delta u^3 - F_{[\mathrm{II}]2}{}^2 F_{[\mathrm{II}]2}{}^2 u^3 + 2F_{[\mathrm{II}]2}{}^2\nabla_2 u^2 + \partial_2 F_{[\mathrm{II}]2}{}^2 u^2 ,$$

where $\boldsymbol{u}|_{\bar{\omega}^{\text{New}}} = (0, u^2(x^2, 0), u^3(x^2, 0))$ is the displacement field of the shell. Thus, one may express the governing equations of the shell as

$$h\Lambda\partial_2\epsilon_2^2(\boldsymbol{u}) + \frac{1}{3}h^3\Lambda(2F_{[\mathrm{II}]2}{}^2\partial_2\rho_2^2(\boldsymbol{u}) + \partial_2 F_{[\mathrm{II}]2}{}^2\rho_2^2(\boldsymbol{u})) - \mathrm{Tr}(T_2^3(\boldsymbol{u})) = 0 , \tag{18}$$
$$-h\Lambda F_{[\mathrm{II}]2}{}^2\epsilon_2^2(\boldsymbol{u}) + \frac{1}{3}h^3\Lambda(\Delta\rho_2^2(\boldsymbol{u}) - F_{[\mathrm{II}]2}{}^2 F_{[\mathrm{II}]2}{}^2\rho_2^2(\boldsymbol{u})) + \mathrm{Tr}(T_3^3(\boldsymbol{u})) = 0 , \tag{19}$$

where

$$\mathrm{Tr}(T_2^3(\boldsymbol{u})) = \bar{\mu}\left((\bar{\psi}_2)^2\partial_3 u^2 + \partial_2 u^3\right)|_{\omega^{\text{New}}} ,$$
$$\mathrm{Tr}(T_3^3(\boldsymbol{u})) = \left[\bar{\lambda}\left(\partial_2 u^2 + \bar{\Gamma}_{22}^2 u^2 + \bar{\Gamma}_{23}^2 u^3\right) + (\bar{\lambda} + 2\bar{\mu})\partial_3 u^3\right]|_{\omega^{\text{New}}} ,$$

and

$$\Lambda = 4\mu\frac{\lambda + \mu}{\lambda + 2\mu} .$$

Now, eliminating x^1 dependency one can express the remaining boundaries as

$$\partial\omega^{\text{New}} = \partial\omega_{T_0}^{\text{New}} \cup \partial\omega_{T_{\max}}^{\text{New}} ,$$
$$\partial\omega_{T_0}^{\text{New}} = \{0\} ,$$
$$\partial\omega_{T_{\max}}^{\text{New}} = \{\pi\} .$$

Thus, the boundary conditions of the shell reduce to

$$\left[\Lambda\epsilon_2^2(\boldsymbol{u}) + \frac{2}{3}h^2\Lambda F_{[\mathrm{II}]2}{}^2\rho_2^2(\boldsymbol{u})\right]|_{\partial\omega_{T_0}^{\text{New}}} = \tau_0 \text{ (traction)} , \tag{20}$$

$$[\Lambda\epsilon_2^2(\boldsymbol{u}) + \frac{2}{3}h^2\Lambda F_{[\text{II}]2}{}^2\rho_2^2(\boldsymbol{u})]|_{\partial\omega^{\text{New}}_{\tau_{\max}}} = \tau_{\max} \text{ (traction)} , \tag{21}$$

$$\partial_2\rho_2^2(\boldsymbol{u})|_{\partial\omega^{\text{New}}} = 0 \text{ (zero-pressure)} , \tag{22}$$

$$\partial_2 u^3|_{\partial\omega^{\text{New}}} = 0 \text{ (zero-Neumann)} . \tag{23}$$

Now, we use the second-order-accurate finite-difference method, but one issue we must tackle prior is the discretisation of the (reduced two-dimensional) domain. As we are dealing with curvilinear coordinates, there is an inherit grid dependence, and it is approximately $\psi_0\Delta x^2 \leq \Delta x^3$, $\forall\ \psi_0 \in \{\bar{\psi}_2(x^2,x^3) \mid x^2 \in [-\frac{1}{2}\pi, \frac{1}{2}\pi]$ and $x^3 \in [-H, 0]\}$, where Δx^j is a small increment in x^j direction. For our purposes we use $\Delta x^2 = \frac{1}{N-1}$ and $\psi_0 = \bar{\psi}_2(\frac{1}{4}\pi, 0)$, where $N = 250$. Finally, we choose to terminate our iterating process once the condition $|1 - ||\boldsymbol{u}_m||_{\ell^2}^{-1}||\boldsymbol{u}_{m+1}||_{\ell^2}| < 10^{-10}$ is satisfied, where $\boldsymbol{u}_m$ is the m^{th} iterative solution.

Finally, let `v1`$= u^2$, `v2`$= u^3$, `a`$= b$, `b`$= a$, `Thickness1`$= h$, `Thickness2`$= H$, `Stress1`$= \tau_0$, `Stress2`$= \tau_{\max}$, `Youngs1`$= E$, `Youngs2`$= \bar{E}$, `Poisson1`$= \nu$, `Poisson1`$= \bar{\nu}$ and `NN`$= N$. Thus, we find:

```
function [U1,U2] = BondedShell(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N)
format long
%% Bonded Shell Model
% Shell on an elastic prism with a variable elliptical cross section: bonded case
% Contact angle is [0,pi]

%% INITIAL PARIMITERS
a = ar; % Radius at \theta = 0.5*pi
b = br; % Radius at \theta = 0

Thickness1 = Th1; % Thickness of the overlying body
Thickness2 = Th2; % Thickness of the foundation

Stress1 = SS1; % Applied stress at \theta = 0
Stress2 = SS2; % Applied stress at \theta = pi

Youngs1 = EE1; % Young's modulus of the overlying body
Poisson1 = PP1; % Poisson's ratio of the overlying body

Youngs2 = EE2; % Young's modulus of the foundation
Poisson2 = PP2; % Poisson's ratio of the foundation

NN = N; % Azimuthal grid points
error =  10^(-10); % Terminating error

%% DO NOT CHANGE!
qq = sqrt((a^2+b^2)/2);
q = Thickness2/(qq*pi);

m = NN; % Azimuthal grid points
n = round(q*NN-q+1); % Radial grid points of the foundation
```

```
L1  = Poisson1*Youngs1/((1+Poisson1)*(1-2*Poisson1));
M1 = 0.5*Youngs1/(1+Poisson1);
L = 4*M1*(L1+M1)/(L1+2*M1);

L2  = Poisson2*Youngs2/((1+Poisson2)*(1-2*Poisson2));
M2 = 0.5*Youngs2/(1+Poisson2);

NNN = NN^2;
errr = error*NNN;
p = 2;

dx1 = pi/(m-1); % Azimuthal grid spacing
dx2 = Thickness2/(n-1); % Radial grid points of the foundation

Idx1 = 1/dx1;
Idx2 = 1/dx2;

Id1 = 0.5*Idx1;
Id11 = Idx1^2;

Id2 = 0.5*Idx2;
Id22 = Idx2^2;

Id12 = Id1*Id2;

u1 = zeros(m,1);
u2 = zeros(m,1);
U1 = zeros(m,1);
U2 = zeros(m,1);

v1 = zeros(m,n); % Azimuthal displacement of the foundation
v2 = zeros(m,n); % Radial displacement of the foundation

X = zeros(m,1);
IX = zeros(m,1);

K1 = zeros(m,1);
K2 = zeros(m,1);
K11 = zeros(m,1);
K21 = zeros(m,1);

Y = zeros(m,n);
IY = zeros(m,n);

LL111 = zeros(m,n);
LL112 = zeros(m,n);

JF11 = zeros(m,n);
JF12 = zeros(m,n);
JF21 = zeros(m,n);
JF22 = zeros(m,n);
```

```
J1B11 = zeros(1,n);
J1B12 = zeros(1,n);
J1B21 = zeros(1,n);
J1B22 = zeros(1,n);

JmB11 = zeros(1,n);
JmB12 = zeros(1,n);
JmB21 = zeros(1,n);
JmB22 = zeros(1,n);

A1d11 = zeros(m,1);
A1d1 = zeros(m,1);
A1d2 = zeros(m,1);

A2d111 = zeros(m,1);
A2d11 = zeros(m,1);
A2d1 = zeros(m,1);

B1d111 = zeros(m,1);
B1d11 = zeros(m,1);
B1d1 = zeros(m,1);

B2d1111 = zeros(m,1);
B2d111 = zeros(m,1);
B2d11 = zeros(m,1);
B2d1 = zeros(m,1);
B2d2 = zeros(m,1);

JM11 = zeros(m,1);
JM12 = zeros(m,1);
JM21 = zeros(m,1);
JM22 = zeros(m,1);

J1M11 = zeros(m,1);
J1M12 = zeros(m,1);
J1M13 = zeros(m,1);
J1M21 = zeros(m,1);
J1M22 = zeros(m,1);
J1M23 = zeros(m,1);
J1M31 = zeros(m,1);
J1M32 = zeros(m,1);
J1M33 = zeros(m,1);

JmM11 = zeros(m,1);
JmM12 = zeros(m,1);
JmM13 = zeros(m,1);
JmM21 = zeros(m,1);
JmM22 = zeros(m,1);
JmM23 = zeros(m,1);
JmM31 = zeros(m,1);
```

```
JmM32 = zeros(m,1);
JmM33 = zeros(m,1);

aa = zeros(2,2);
aaa = zeros(3,3);

C1d22 = zeros(m,n);
C1d1 = zeros(m,n);
C1d2 = zeros(m,n);
C2d1 = zeros(m,n);

D1d12 = zeros(m,n);
D2d11 = zeros(m,n);
D2d22 = zeros(m,n);
D1d1 = zeros(m,n);
D1d2 = zeros(m,n);
D2d1 = zeros(m,n);
D2d2 = zeros(m,n);

XX = (Thickness1^2)/3;
IXY = 1/(L*Thickness1);

T1 = Stress1/L;
T2 = Stress2/L;

a1d1 = 1;
da1d11 = 1;

C1d11 = (L2+2*M2);
C2d12 = (L2+M2);

%% Curvature terms of the shell
for i = 1:m

    x1 = (i-1)*dx1;

    alpha2 = (b*sin(x1))^2 + (a*cos(x1))^2;
    alpha = sqrt(alpha2);
    Ialpha = 1/alpha;
    alphad1 = 0.5*(b^2-a^2)*sin(2*x1)*Ialpha;
    alphad11 = ((b^2-a^2)*cos(2*x1) - alphad1^2)*Ialpha;
    alphad111 = -(2*(b^2-a^2)*cos(2*x1) + 3*alphad1*alphad11)*Ialpha;

    X(i) = alpha;
    IX(i) = 1/X(i);
    IX2 = IX(i)^2;
    Xd1 = alphad1;
    Xd11 = alphad11;
    Xd111 = alphad111;
    Xd2 = a*b*Ialpha^2;
    Xd12 = -2*a*b*alphad1*Ialpha^3;
```

```
Xd112 = -2*a*b*alphad11*Ialpha^3 + 6*a*b*(alphad1^2)*(Ialpha^4);
Xd1112 = -2*a*b*alphad111*Ialpha^3 + 18*a*b*alphad1*alphad11*(Ialpha^4) ...
    - 24*a*b*(alphad1^3)*(Ialpha^5);

K1(i) = Xd1*IX(i);
K2(i) = Xd2*IX(i);
K11(i) = Xd11*IX(i)-K1(i)^2;
K21(i) = Xd12*IX(i)-K1(i)*K2(i);

K111 = Xd111*IX(i) - 3*Xd11*IX(i)*K1(i) + 2*K1(i)^3;
K211 = Xd112*IX(i) - 2*Xd12*IX(i)*K1(i) - Xd11*IX(i)*K2(i) + 2*K2(i)*K1(i)^2;
K2111 = Xd1112*IX(i) - 3*Xd112*IX(i)*K1(i) - 3*Xd12*Xd11*IX(i)^2 ...
    + 6*Xd12*IX(i)*K1(i)^2 - Xd111*IX(i)*K2(i) ...
    + 6*Xd11*IX(i)*K1(i)*K2(i) - 6*K2(i)*K1(i)^3;

a1 = K1(i);
a2 = K2(i);

da1d1 = K1(i);
da1 = K11(i);
da2d1 = K2(i);
da2 = K21(i);

c1d2 = - M2*(X(i)^2)*IXY;
c2d1 = - M2*IXY;

c1d1 = L2*IXY;
c1 = L2*K1(i)*IXY;
c2d2 = (L2+2*M2)*IXY;
c2 = L2*K2(i)*IXY;

b1d1 = - 2*K2(i);
b1 = - K21(i) - 2*K1(i)*K2(i);
b2d11 = IX2;
b2d1 = - IX2*K1(i);
b2 = - K2(i)^2;

db1d11 = - 2*K2(i);
db1d1 = - 3*K21(i) - 2*K1(i)*K2(i);
db1 = - K211 - 2*K11(i)*K2(i) - 2*K1(i)*K21(i);
db2d111 =  IX2;
db2d11 = - 3*IX2*K1(i);
db2d1 = - K2(i)^2 - IX2*K11(i) + 2*IX2*K1(i)^2;
db2 = - 2*K21(i)*K2(i);

ddb1d111 = - 2*K2(i);
ddb1d11 = - 5*K21(i) - 2*K1(i)*K2(i);
ddb1d1 = - 4*K211 - 4*K11(i)*K2(i) - 4*K21(i)*K1(i);
ddb1 = - K2111 - 2*K111*K2(i) - 4*K11(i)*K21(i) - 2*K211*K1(i);
ddb2d1111 = IX2;
ddb2d111 = - 5*IX2*K1(i);
```

```
ddb2d11 = - K2(i)^2 - 4*IX2*K11(i) + 8*IX2*K1(i)^2;
ddb2d1 = - IX2*K111 + 6*IX2*K11(i)*K1(i) - 4*IX2*K1(i)^3 - 4*K21(i)*K2(i);
ddb2 = - 2*K211*K2(i) - 2*K21(i)^2;

A1d11(i) = da1d11 - 2*K2(i)*XX*db1d11;
A1d1(i) = da1d1 - 2*K2(i)*XX*db1d1 - K21(i)*XX*b1d1;
A1d2(i) = c1d2;
A1 = da1 - 2*K2(i)*XX*db1 - K21(i)*XX*b1 + 3*Id2*c1d2;

A2d111(i) = - 2*K2(i)*XX*db2d111;
A2d11(i) = - 2*K2(i)*XX*db2d11 - K21(i)*XX*b2d11;
A2d1(i) = da2d1 - 2*K2(i)*XX*db2d1 - K21(i)*XX*b2d1 + c2d1;
A2 = da2 - 2*K2(i)*XX*db2 - K21(i)*XX*b2;

B1d111(i) = IX2*XX*ddb1d111;
B1d11(i) = IX2*XX*ddb1d11 - IX2*K1(i)*XX*db1d11;
B1d1(i) = K2(i)*a1d1 + IX2*XX*ddb1d1 - IX2*K1(i)*XX*db1d1 ...
    - (K2(i)^2)*XX*b1d1 + c1d1;
B1 = K2(i)*a1 + IX2*XX*ddb1 - IX2*K1(i)*XX*db1 - (K2(i)^2)*XX*b1 + c1;

B2d1111(i) = IX2*XX*ddb2d1111;
B2d111(i) = IX2*XX*ddb2d111 - IX2*K1(i)*XX*db2d111;
B2d11(i) = IX2*XX*ddb2d11 - IX2*K1(i)*XX*db2d11 - (K2(i)^2)*XX*b2d11;
B2d1(i) = IX2*XX*ddb2d1 - IX2*K1(i)*XX*db2d1 - (K2(i)^2)*XX*b2d1;
B2d2(i) = c2d2;
B2 = K2(i)*a2 + IX2*XX*ddb2 - IX2*K1(i)*XX*db2 - (K2(i)^2)*XX*b2 ...
    + c2 + 3*Id2*c2d2;

aa(1,1) = 2*A1d11(i)*Id11 - A1;
aa(1,2) = 2*A2d11(i)*Id11 - A2;
aa(2,1) = 2*B1d11(i)*Id11 - B1;
aa(2,2) = 2*B2d11(i)*Id11 - B2 - 4*B2d1111(i)*Id11^2;

J = inv(aa);

JM11(i) = J(1,1);
JM12(i) = J(1,2);
JM21(i) = J(2,1);
JM22(i) = J(2,2);

aaa(1,1) = 3*Id1*a1d1 - a1;
aaa(1,2) = - a2;
aaa(1,3) = 0;
aaa(2,1) = 0;
aaa(2,2) = 3*Id1;
aaa(2,3) = 0;
aaa(3,1) = - 2*Id11*db1d11 + 3*Id1*db1d1 - db1;
aaa(3,2) = - 2*Id11*db2d11 - db2;
aaa(3,3) = 3*Id1*db2d111;

J = inv(aaa);
```

```
    J1M11(i) = J(1,1);
    J1M12(i) = J(1,2);
    J1M13(i) = J(1,3);
    J1M21(i) = J(2,1);
    J1M22(i) = J(2,2);
    J1M23(i) = J(2,3);
    J1M31(i) = J(3,1);
    J1M32(i) = J(3,2);
    J1M33(i) = J(3,3);

    aaa(1,1) = - 3*Id1*a1d1 - a1;
    aaa(1,2) = - a2;
    aaa(1,3) = 0;
    aaa(2,1) = 0;
    aaa(2,2) = - 3*Id1;
    aaa(2,3) = 0;
    aaa(3,1) = - 2*Id11*db1d11 - 3*Id1*db1d1 - db1;
    aaa(3,2) = - 2*Id11*db2d11 - db2;
    aaa(3,3) = - 3*Id1*db2d111;

    J = inv(aaa);

    JmM11(i) = J(1,1);
    JmM12(i) = J(1,2);
    JmM13(i) = J(1,3);
    JmM21(i) = J(2,1);
    JmM22(i) = J(2,2);
    JmM23(i) = J(2,3);
    JmM31(i) = J(3,1);
    JmM32(i) = J(3,2);
    JmM33(i) = J(3,3);

end

%% Curvature terms of the foundation
for i = 1:m

    x1 = (i-1)*dx1;

    for j = 1:n

        x2 = (j-1)*dx2 - Thickness2;

        alpha2 = (b*sin(x1))^2 + (a*cos(x1))^2;
        alpha = sqrt(alpha2);
        Ialpha = 1/alpha;
        Ialpha2 = 1/alpha2;
        alphad1 = 0.5*(b^2-a^2)*sin(2*x1)*Ialpha;
        alphad11 = ((b^2-a^2)*cos(2*x1) - alphad1^2)*Ialpha;
```

```
        Y(i,j) = alpha + a*b*Ialpha2*x2;
        IY(i,j) = 1/Y(i,j);
        Yd1 = alphad1*(1-2*a*b*x2*Ialpha*Ialpha2);
        Yd11 = alphad11*(1-2*a*b*x2*Ialpha*Ialpha2) + 6*a*b*x2*(alphad1*Ialpha2)^2;
        Yd2 = a*b*Ialpha2;
        Yd12 = -2*a*b*alphad1*Ialpha*Ialpha2;

        LL111(i,j) = Yd1*IY(i,j);
        LL112(i,j) = Yd2*IY(i,j);
        LL1111 = Yd11*IY(i,j)-LL111(i,j)^2;
        LL1112 = Yd12*IY(i,j)-LL111(i,j)*LL112(i,j);
        LL1122 = -LL112(i,j)^2;

        C1d22(i,j) = M2*Y(i,j)^2;
        C1d1(i,j) = (L2+2*M2)*LL111(i,j);
        C1d2(i,j) = 3*M2*LL112(i,j)*Y(i,j)^2;
        C2d1(i,j) = (L2+3*M2)*LL112(i,j);
        C1 = (L2+2*M2)*LL1111;
        C2 = (L2+2*M2)*LL1112;

        D1d12(i,j) = (L2+M2)*Y(i,j);
        D2d11(i,j) = M2*IY(i,j);
        D2d22(i,j) = (L2+2*M2)*Y(i,j);
        D1d1(i,j) = - 2*M2*LL112(i,j)*Y(i,j);
        D1d2(i,j) = (L2+M2)*LL111(i,j)*Y(i,j);
        D2d1(i,j) = - M2*LL111(i,j)*IY(i,j);
        D2d2(i,j) = (L2+2*M2)*LL112(i,j)*Y(i,j);
        D1 = L2*LL1112*Y(i,j) - 2*M2*LL111(i,j)*LL112(i,j)*Y(i,j);
        D2 = L2*LL1122*Y(i,j) - 2*M2*LL112(i,j)*LL112(i,j)*Y(i,j);

        aa(1,1) = 2*C1d11*Id11      + 2*C1d22(i,j)*Id22 - C1;
        aa(1,2) = - C2;
        aa(2,1) = - D1;
        aa(2,2) = 2*D2d11(i,j)*Id11 + 2*D2d22(i,j)*Id22 - D2;

        J = inv(aa);

        JF11(i,j) = J(1,1);
        JF12(i,j) = J(1,2);
        JF21(i,j) = J(2,1);
        JF22(i,j) = J(2,2);

    end
end

for j = 1:n

    aa(1,1) = 3*Id1*(L2+2*M2) - (L2+2*M2)*LL111(1,j);
    aa(1,2) = - (L2+2*M2)*LL112(1,j);
    aa(2,1) = 0;
    aa(2,2) = 3*Id1;
```

```
    J = inv(aa);

    J1B11(j) = J(1,1);
    J1B12(j) = J(1,2);
    J1B21(j) = J(2,1);
    J1B22(j) = J(2,2);

    aa(1,1) = - 3*Id1*(L2+2*M2) - (L2+2*M2)*LL111(m,j);
    aa(1,2) = - (L2+2*M2)*LL112(m,j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Id1;

    J = inv(aa);

    JmB11(j) = J(1,1);
    JmB12(j) = J(1,2);
    JmB21(j) = J(2,1);
    JmB22(j) = J(2,2);

end

%% Main code
while errr < p

    pp = norm(v1,2)+norm(v2,2);

    for k = 1:NNN

        %% Corners
        u1(1) = (2*v1(1,n)-5*v1(2,n)+4*v1(3,n)-v1(4,n))*Id11;

        v1d1 = (4*v1(2,n)-v1(3,n))*Id1;
        v2d1 = (4*v2(2,n)-v2(3,n))*Id1;
        u2d1 = (4*u2(2)-u2(3))*Id1;
        v1d11 = (-5*v1(2,n)+4*v1(3,n)-v1(4,n))*Id11;
        v2d11 = (-5*v2(2,n)+4*v2(3,n)-v2(4,n))*Id11;

        Xv1 = v1d1 - T1; % equation (21)
        Xv2 = v2d1; % equation (23)
        Xu2 = - 2*K2(1)*v1d11 - (3*K21(1)+2*K1(1)*K2(1))*v1d1 ...
            + (IX(1)^2)*u2d1 - 3*(IX(1)^2)*K1(1)*v2d11; % equation (22)

        v1(1,n) = J1M11(1)*Xv1 + J1M12(1)*Xv2 + J1M13(1)*Xu2;
        v2(1,n) = J1M21(1)*Xv1 + J1M22(1)*Xv2 + J1M23(1)*Xu2;
        u2(1)   = J1M31(1)*Xv1 + J1M32(1)*Xv2 + J1M33(1)*Xu2;

        u1(m) = (2*v1(m,n)-5*v1(m-1,n)+4*v1(m-2,n)-v1(m-3,n))*Id11;

        v1d1 = -(4*v1(m-1,n)-v1(m-2,n))*Id1;
        v2d1 = -(4*v2(m-1,n)-v2(m-2,n))*Id1;
```

```
u2d1 = -(4*u2(m-1)-u2(m-2))*Id1;
v1d11 = (-5*v1(m-1,n)+4*v1(m-2,n)-v1(m-3,n))*Id11;
v2d11 = (-5*v2(m-1,n)+4*v2(m-2,n)-v2(m-3,n))*Id11;

Xv1 = v1d1 - T2; % equation (21)
Xv2 = v2d1; % equation (23)
Xu2 = - 2*K2(m)*v1d11 - (3*K21(m)+2*K1(m)*K2(m))*v1d1 ...
    + (IX(m)^2)*u2d1 - 3*(IX(m)^2)*K1(m)*v2d11; % equation (22)

v1(m,n) = JmM11(m)*Xv1 + JmM12(m)*Xv2 + JmM13(m)*Xu2;
v2(m,n) = JmM21(m)*Xv1 + JmM22(m)*Xv2 + JmM23(m)*Xu2;
u2(m)   = JmM31(m)*Xv1 + JmM32(m)*Xv2 + JmM33(m)*Xu2;

%% Gouerning equations of the shell
for i = 2:m-1

    u1(i) = (v1(i+1,n)-2*v1(i,n)+v1(i-1,n))*Id11;
    u2(i) = (v2(i+1,n)-2*v2(i,n)+v2(i-1,n))*Id11;

    v1d11 = (v1(i+1,n)+v1(i-1,n))*Id11;
    v2d11 = (v2(i+1,n)+v2(i-1,n))*Id11;

    v1d1 = (v1(i+1,n)-v1(i-1,n))*Id1;
    v2d1 = (v2(i+1,n)-v2(i-1,n))*Id1;

    u2d11 = (u2(i+1)+u2(i-1))*Id11;

    u1d1 = (u1(i+1)-u1(i-1))*Id1;
    u2d1 = (u2(i+1)-u2(i-1))*Id1;

    v1d2 = -(4*v1(i,n-1)-v1(i,n-2))*Id2;
    v2d2 = -(4*v2(i,n-1)-v2(i,n-2))*Id2;

    Xv1 = A1d11(i)*v1d11 + A1d1(i)*v1d1   + A1d2(i)*v1d2 ...
        + A2d111(i)*u2d1   + A2d11(i)*v2d11 + A2d1(i)*v2d1; % equation (18)
    Xv2 = B1d111(i)*u1d1 + B1d11(i)*v1d11 + B1d1(i)*v1d1 + B2d1111(i)*u2d11 ...
        + B2d111(i)*u2d1 + (B2d11(i)-2*B2d1111(i)*Id11)*v2d11 ...
        + B2d1(i)*v2d1 + B2d2(i)*v2d2; % equation (19)

    v1(i,n) = JM11(i)*Xv1 + JM12(i)*Xv2;
    v2(i,n) = JM21(i)*Xv1 + JM22(i)*Xv2;

end

%% Stress-free boundary of the foundation
for j = 2:n-1

    v1d1 = (4*v1(2,j)-v1(3,j))*Id1;
    v1d2 = (v1(1,j+1)-v1(1,j-1))*Id2;
    v2d1 = (4*v2(2,j)-v2(3,j))*Id1;
    v2d2 = (v2(1,j+1)-v2(1,j-1))*Id2;
```

```
        Xv1 = (L2+2*M2)*v1d1 + L2*v2d2; % equation (17)
        Xv2 = v2d1 + (Y(1,j)^2)*v1d2; % equation (16)

        v1(1,j) = J1B11(j)*Xv1 + J1B12(j)*Xv2;
        v2(1,j) = J1B21(j)*Xv1 + J1B22(j)*Xv2;

        v1d1 = -(4*v1(m-1,j)-v1(m-2,j))*Id1;
        v1d2 = (v1(m,j+1)-v1(m,j-1))*Id2;
        v2d1 = -(4*v2(m-1,j)-v2(m-2,j))*Id1;
        v2d2 = (v2(m,j+1)-v2(m,j-1))*Id2;

        Xv1 = (L2+2*M2)*v1d1 + L2*v2d2; % equation (17)
        Xv2 = v2d1 + (Y(m,j)^2)*v1d2; % equation (16)

        v1(m,j) = JmB11(j)*Xv1 + JmB12(j)*Xv2;
        v2(m,j) = JmB21(j)*Xv1 + JmB22(j)*Xv2;

    end

    %% Gouerning equations of the foundation
    for i = 2:m-1

        for j = 2:n-1

            v1d11 = (v1(i+1,j)+v1(i-1,j))*Id11;
            v1d22 = (v1(i,j+1)+v1(i,j-1))*Id22;
            v1d12 = (v1(i+1,j+1)-v1(i+1,j-1)-v1(i-1,j+1)+v1(i-1,j-1))*Id12;
            v1d1 = (v1(i+1,j)-v1(i-1,j))*Id1;
            v1d2 = (v1(i,j+1)-v1(i,j-1))*Id2;

            v2d11 = (v2(i+1,j)+v2(i-1,j))*Id11;
            v2d22 = (v2(i,j+1)+v2(i,j-1))*Id22;
            v2d12 = (v2(i+1,j+1)-v2(i+1,j-1)-v2(i-1,j+1)+v2(i-1,j-1))*Id12;
            v2d1 = (v2(i+1,j)-v2(i-1,j))*Id1;
            v2d2 = (v2(i,j+1)-v2(i,j-1))*Id2;

            Xv1 = C1d11*v1d11      + C1d22(i,j)*v1d22 + C2d12*v2d12      ...
                + C1d1(i,j)*v1d1 + C1d2(i,j)*v1d2 + C2d1(i,j)*v2d1; % equation (14)
            Xv2 = D2d11(i,j)*v2d11 + D2d22(i,j)*v2d22 + D1d12(i,j)*v1d12 ...
                + D1d1(i,j)*v1d1 + D1d2(i,j)*v1d2 ...
                + D2d1(i,j)*v2d1 + D2d2(i,j)*v2d2; % equation (15)

            v1(i,j) = JF11(i,j)*Xv1 + JF12(i,j)*Xv2;
            v2(i,j) = JF21(i,j)*Xv1 + JF22(i,j)*Xv2;

        end
    end
end

%% Terminating condition
```

```
    ppp = norm(v1,2)+norm(v2);
    p = abs(1- ppp/pp);

end

%% Contact region
U1(:) = v1(:,n);
U2(:) = v2(:,n);
```

2.2 Baldelli and Bourdin's Model

In this section we present a numerical code for an example of Baldelli and Bourdin's [2] model for membranes bonded to elastic pseudo-foundations (see section 3.6 of Jayawardana [4]) implemented in Matlab, i.e. `BaldelliProb.m`.

To conduct numerical experiments we remain with the framework that we introduced in Section 2.1. Thus, given that $\boldsymbol{w} = (0, w^2(x^2), 0)$ is the displacement field of extended Baldelli and Bourdin's model for a membrane supported by an elastic foundation, one finds

$$\nabla_2 w^2 = \partial_2 w^2 + \Gamma^2_{22} w^2 \ ,$$
$$\epsilon^2_2(\boldsymbol{w}) = \nabla_2 w^2 \ .$$

Thus, one can express the governing equations as

$$\Lambda \partial_2 \epsilon(w)^2_2 - \bar{\mu}(hH)^{-1}(\psi_2)^2 w^2 = 0 \ ,$$

and the boundary conditions as

$$\Lambda \epsilon^2_2(\boldsymbol{w})|_{\partial\omega^{\text{New}}_{T_0}} = \tau_0 \text{ (traction)} \ ,$$
$$\Lambda \epsilon^2_2(\boldsymbol{w})|_{\partial\omega^{\text{New}}_{T_{\text{max}}}} = \tau_{\text{max}} \text{ (traction)} \ .$$

With a little more effort, one can solve this problem explicitly. Thus, given that $\tau_0, \tau_{\text{max}} = 1$, the explicit solution is

$$w^2(x^2) = \frac{\sinh\left(a\alpha E(x^2, e)\right)}{\alpha\Lambda\varphi(x^2)\cosh\left(a\alpha E(e)\right)} \ , \qquad (24)$$

where $E(x^2, e) = \int_0^{x^2} (1 - e^2\sin^2(\theta))^{\frac{1}{2}} \ d\theta$ is the incomplete elliptic integral of the second kind, $E(e) = E(\frac{1}{2}\pi, e)$ is the complete elliptic integral of the second kind, $e = (1 - (b/a)^2)^{\frac{1}{2}}$ is the elliptical modulus (see chapter 17 of Abramowitz *et al.* [1]) and $\alpha = (\bar{\mu}/(hH\Lambda))^{\frac{1}{2}}$.

Finally, let `v1`$= w^2$, `a`$= b$, `b`$= a$, `Thickness1`$= h$, `Thickness2`$= H$, `Stress1`$= 1$, `Stress2`$= 1$, `Youngs1`$= E$, `Youngs2`$= \bar{E}$, `Poisson1`$= \nu$, `Poisson1`$= \bar{\nu}$ and `NN`$= N$. Thus, we find:

```
function [W1] = BaldelliProb(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N)
format long
%% Baldelli and and Bourdin's Model
```

```
% Membrane bonded to an elastic prism with a variable elliptical cross section
% Contact angle is [0,pi]

%% INITIAL PARIMITERS
a = ar; % Radius at \theta = 0.5*pi
b = br; % Radius at \theta = 0

Thickness1 = Th1; % Thickness of the foundation
Thickness2 = Th2; % Thickness of the overlying body

% Stress1 = 1; % Applied stress at \theta = 0
% Stress2 = 1; % Applied stress at \theta = pi

Youngs1 = EE1; % Young's modulus of the overlying body
Poisson1 = PP1; % Poisson's ratio of the overlying body

Youngs2 = EE2; % Young's modulus of the foundation
Poisson2 = PP2; % Poisson's ratio of the foundation

NN = N; % Azimuthal grid points

%% DO NOT CHANGE!
m = NN; % Azimuthal grid points

L1  = Poisson1*Youngs1/((1+Poisson1)*(1-2*Poisson1));
M1 = 0.5*Youngs1/(1+Poisson1);
L = 4*M1*(L1+M1)/(L1+2*M1);

M2 = 0.5*Youngs2/(1+Poisson2);

dx1 = pi/(m-1);

v1 = zeros(m,1); %% Azimuthal displacement at the boundary

LM = M2/(L*Thickness1*Thickness2);
LMI2 = LM^(0.5);
aIb = a/b;
e = (1-aIb^2);
xv = LMI2*L*cosh(LMI2*b*ellipticE(e));
Iv = 1/xv;

%% Main code
for i = 1:m

    x1 = (i-1)*dx1 - 0.5*pi;

    alpha2 = (a*sin(x1))^2 + (b*cos(x1))^2;
    alpha = sqrt(alpha2);
    Ialpha = 1/alpha;

    v1(i) = sinh(b*LMI2*ellipticE(x1,e))*Iv*Ialpha; % equation (24)
```

```
end

%% Contact region
W1(:) = v1(:);
```

2.3 Two-Body Model

In this section we present a numerical code for an example of the two-body model for two bonded elastic bodies (see section 3.6 of Jayawardana [4]) implemented in Matlab, i.e. `TwoBody.m`.

To conduct numerical experiments we numerically model the overlying body as a three-dimensional body and we do not approximate this body as a shell or otherwise. Thus, the displacement at the contact region with this approach is the displacement field at the contact region of the bonded two-body elastic problem, whose solution is obtained by the use of the stranded equilibrium equations in the liner elasticity theory.

In accordance with the framework that is introduced in Section 2.1, the overlying body is restricted to the region $x^3 \in (0,h)$. Thus, with some calculations one finds

$$\begin{aligned}
\bar{\nabla}_2 v^2 &= \partial_2 v^2 + \bar{\Gamma}^2_{22} v^2 + \bar{\Gamma}^2_{23} v^3 \ , \\
\bar{\nabla}_2 v^3 &= \partial_2 v^3 - (\bar{\psi}_2)^2 \bar{\Gamma}^2_{23} v^2 \ , \\
\bar{\nabla}_3 v^2 &= \partial_3 v^2 + \bar{\Gamma}^2_{23} v^2 \ , \\
\bar{\nabla}_3 v^3 &= \partial_3 v^3 \ ,
\end{aligned}$$

where $\boldsymbol{v} = (0, v^2(x^2,x^3), v^3(x^2,x^3))$ is the displacement field of the overlying body. With relative ease, one can express the governing equations of the overlying body as

$$(\lambda+\mu)\partial^2\left(\bar{\nabla}_i v^i\right) + \mu\bar{\Delta} v^2 = 0 \ , \tag{25}$$

$$(\lambda+\mu)\partial^3\left(\bar{\nabla}_i v^i\right) + \mu\bar{\Delta} v^3 = 0 \ , \tag{26}$$

and the boundary conditions of the overlying body as

$$\left[(\lambda+2\mu)\partial_2 v^2 + \lambda\left(\partial_3 v^3 + \bar{\Gamma}^2_{22} v^2 + \bar{\Gamma}^2_{23} v^3\right)\right]|_{\{\partial\omega^{\text{New}}_{T_0}\times[0,h]\}} = \tau_0 \text{ (traction)} \ , \tag{27}$$

$$\left[(\lambda+2\mu)\partial_2 v^2 + \lambda\left(\partial_3 v^3 + \bar{\Gamma}^2_{22} v^2 + \bar{\Gamma}^2_{23} v^3\right)\right]|_{\{\partial\omega^{\text{New}}_{T\text{max}}\times[0,h]\}} = \tau_{\text{max}} \text{ (traction)} \ , \tag{28}$$

$$\left[\lambda\left(\partial_2 v^2 + \bar{\Gamma}^2_{22} v^2 + \bar{\Gamma}^2_{23} v^3\right) + (\lambda+2\mu)\partial_3 v^3\right]|_{\{(-\frac{1}{2}\pi,\frac{1}{2}\pi)\times\{h\}\}} = 0 \text{ (zero-Robin)} \ , \tag{29}$$

$$\left[(\bar{\psi}_2)^2\partial_3 v^2 + \partial_2 v^3\right]|_{\{\partial\omega^{\text{New}}\times[0,h]\}\cup\{(-\frac{1}{2}\pi,\frac{1}{2}\pi)\times\{h\}\}} = 0 \text{ (zero-Robin)} \ , \tag{30}$$

with following equations characterising the bonding of the overlying body to the foundation

$$\left[u^2 - v^2\right]|_{\omega^{\text{New}}} = 0 \text{ (continuous azimuthal displacement)} \ , \tag{31}$$

$$\left[u^3 - v^3\right]|_{\omega^{\text{New}}} = 0 \text{ (continuous radial displacement)} \ , \tag{32}$$

$$\text{Tr}(T^2_3(\boldsymbol{u})) - \mu\left((\bar{\psi}_2)^2\partial_3 v^2 + \partial_2 v^3\right)|_{\omega^{\text{New}}} = 0 \text{ (continuous azimuthal stress)} \ , \tag{33}$$

$$\text{Tr}(T^3_3(\boldsymbol{u})) - \left[\lambda\left(\partial_2 v^2 + \bar{\Gamma}^2_{22} v^2 + \bar{\Gamma}^2_{23} v^3\right) + (\lambda+2\mu)\partial_3 v^3\right]|_{\omega^{\text{New}}} = 0 \text{ (continuous radial stress)} \ . \tag{34}$$

Note that the grid dependence of the overlying body is approximately $\psi_0 \Delta x^2 \leq \Delta x^3$, $\forall\ \psi_0 \in \{\bar{\psi}_2(x^2, x^3) \mid x^2 \in [-\frac{1}{2}\pi, \frac{1}{2}\pi]$ and $x^3 \in [0, h]\}$, where Δx^j is a small increment in x^j direction. For our purposes, we use $\Delta x^2 = \frac{1}{N-1}$ and $\psi_0 = \bar{\psi}_2(\frac{1}{4}\pi, h)$, where $N = 250$. Furthermore, we choose to terminate our iterating process once the condition $|1-(||\boldsymbol{u}_m||_{\ell^2}+||\boldsymbol{v}_m||_{\ell^2})^{-1}(||\boldsymbol{u}_{m+1}||_{\ell^2}+||\boldsymbol{v}_{m+1}||_{\ell^2})| < 10^{-10}$ is satisfied, where $\boldsymbol{u}_m$ and $\boldsymbol{v}_m$ are the m^{th} iterative solutions of the bonded two-body model.

Finally, let `v1`$= u^2$, `v2`$= u^3$, `u1`$= v^2$, `u2`$= v^3$, `a`$= b$, `b`$= a$, `Thickness1`$= h$, `Thickness2`$= H$, `Stress1`$= \tau_0$, `Stress2`$= \tau_{\text{max}}$, `Youngs1`$= E$, `Youngs2`$= \bar{E}$, `Poisson1`$= \nu$, `Poisson1`$= \bar{\nu}$ and `NN`$= N$. Thus, we find:

```
function [V1,V2] = TwoBody(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N)
format long
%% Two-Body Model
% Overlying elastic body on an elastic prism with a variable elliptical cross section
% Contact angle is [0,pi]
% Bonded case

%% INITIAL PARAMETERS
a = ar; % Radius at \theta = 0.5*pi
b = br; % Radius at \theta = 0

Thickness1 = Th1; % Thickness of the overlying body
Thickness2 = Th2; % Thickness of the foundation

Stress1 = 1; % Applied stress at \theta = 0
Stress2 = 1; % Applied stress at \theta = pi

Youngs1 = EE1; % Young's modulus of the overlying body
Poisson1 = PP1; % Poisson's ratio of the overlying body

Youngs2 = EE2; % Young's modulus of the foundation
Poisson2 = PP2; % Poisson's ratio of the foundation

NN = N; % Azimuthal grid points
error =  10^(-10); % Terminating error

%% DO NOT CHANGE!
qq1 = sqrt((a^2+b^2)/2) + 2*a*b*Thickness1/(a^2+b^2);
qq2 = sqrt((a^2+b^2)/2);

q1 = Thickness1/(qq1*pi);
q2 = Thickness2/(qq2*pi);

m = NN; % Azimuthal grid points
n1 = round(q1*NN-q1+1); % Radial grid points of the overlying body
n2 = round(q2*NN-q2+1); % Radial grid points of the foundation

L1  = Poisson1*Youngs1/((1+Poisson1)*(1-2*Poisson1));
```

```
M1 = 0.5*Youngs1/(1+Poisson1);

L2  = Poisson2*Youngs2/((1+Poisson2)*(1-2*Poisson2));
M2 = 0.5*Youngs2/(1+Poisson2);

NNN = NN^2;
errr = error*NNN;
p = 2;

dx1 = pi/(m-1); % Azimuthal grid spacing
dx2 = Thickness1/(n1-1); % Radial grid spacing of the overlying body
dX2 = Thickness2/(n2-1); % Radial grid spacing of the foundation

Idx1 = 1/dx1;
Idx2 = 1/dx2;
IdX2 = 1/dX2;

Id1 = 0.5*Idx1;
Id11 = Idx1^2;

Id2 = 0.5*Idx2;
Id22 = Idx2^2;

Id12 = Id1*Id2;

ID2 = 0.5*IdX2;
ID22 = IdX2^2;

ID12 = Id1*ID2;

aa = zeros(2,2);

u1 = zeros(m,n1); % Azimuthal displacement of the overlying body
u2 = zeros(m,n1); % Radial displacement of the overlying body

V1 = zeros(m,1);
V2 = zeros(m,1);

v1 = zeros(m,n2); % Azimuthal displacement of the foundation
v2 = zeros(m,n2); % Radial displacement of the foundation

X = zeros(m,n1);
IX = zeros(m,n1);

L111 = zeros(m,n1);
L112 = zeros(m,n1);

Y = zeros(m,n2);
IY = zeros(m,n2);

LL111 = zeros(m,n2);
```

```
LL112 = zeros(m,n2);

JM11 = zeros(m,n1);
JM12 = zeros(m,n1);
JM21 = zeros(m,n1);
JM22 = zeros(m,n1);

JF11 = zeros(m,n2);
JF12 = zeros(m,n2);
JF21 = zeros(m,n2);
JF22 = zeros(m,n2);

J1S11 = zeros(1,n1);
J1S12 = zeros(1,n1);
J1S21 = zeros(1,n1);
J1S22 = zeros(1,n1);

JmS11 = zeros(1,n1);
JmS12 = zeros(1,n1);
JmS21 = zeros(1,n1);
JmS22 = zeros(1,n1);

J1B11 = zeros(1,n2);
J1B12 = zeros(1,n2);
J1B21 = zeros(1,n2);
J1B22 = zeros(1,n2);

JmB11 = zeros(1,n2);
JmB12 = zeros(1,n2);
JmB21 = zeros(1,n2);
JmB22 = zeros(1,n2);

JnB11 = zeros(m,1);
JnB12 = zeros(m,1);
JnB21 = zeros(m,1);
JnB22 = zeros(m,1);

JBB11 = zeros(m,1);
JBB12 = zeros(m,1);
JBB21 = zeros(m,1);
JBB22 = zeros(m,1);

JBBB11 = zeros(m,1);
JBBB12 = zeros(m,1);
JBBB21 = zeros(m,1);
JBBB22 = zeros(m,1);

A1d22 = zeros(m,n1);
A1d1 = zeros(m,n1);
A1d2 = zeros(m,n1);
A2d1 = zeros(m,n1);
```

```
B1d12 = zeros(m,n1);
B2d11 = zeros(m,n1);
B2d22 = zeros(m,n1);
B1d1 = zeros(m,n1);
B1d2 = zeros(m,n1);
B2d1 = zeros(m,n1);
B2d2 = zeros(m,n1);

C1d22 = zeros(m,n2);
C1d1 = zeros(m,n2);
C1d2 = zeros(m,n2);
C2d1 = zeros(m,n2);

D1d12 = zeros(m,n2);
D2d11 = zeros(m,n2);
D2d22 = zeros(m,n2);
D1d1 = zeros(m,n2);
D1d2 = zeros(m,n2);
D2d1 = zeros(m,n2);
D2d2 = zeros(m,n2);

T1 = Stress1;
T2 = Stress2;

A1d11 = (L1+2*M1);
A2d12 = (L1+M1);

C1d11 = (L2+2*M2);
C2d12 = (L2+M2);

%% Curvature terms of the overlying body
for i = 1:m

    x1 = (i-1)*dx1;

    for j = 1:n1

        x2 = (j-1)*dx2;

        alpha2 = (b*sin(x1))^2 + (a*cos(x1))^2;
        alpha = sqrt(alpha2);
        Ialpha = 1/alpha;
        Ialpha2 = 1/alpha2;
        alphad1 = 0.5*(b^2-a^2)*sin(2*x1)*Ialpha;
        alphad11 = ((b^2-a^2)*cos(2*x1) - alphad1^2)*Ialpha;

        X(i,j) = alpha + a*b*Ialpha2*x2;
        IX(i,j) = 1/X(i,j);
        Xd1 = alphad1*(1-2*a*b*x2*Ialpha*Ialpha2);
        Xd11 = alphad11*(1-2*a*b*x2*Ialpha*Ialpha2) + 6*a*b*x2*(alphad1*Ialpha2)^2;
```

```
        Xd2 = a*b*Ialpha2;
        Xd12 = -2*a*b*alphad1*Ialpha*Ialpha2;

        L111(i,j) = Xd1*IX(i,j);
        L112(i,j) = Xd2*IX(i,j);
        L1111 = Xd11*IX(i,j)-L111(i,j)^2;
        L1112 = Xd12*IX(i,j)-L111(i,j)*L112(i,j);
        L1122 = -L112(i,j)^2;

        A1d22(i,j) = M1*X(i,j)^2;
        A1d1(i,j) = (L1+2*M1)*L111(i,j);
        A1d2(i,j) = 3*M1*L112(i,j)*X(i,j)^2;
        A2d1(i,j) = (L1+3*M1)*L112(i,j);
        A1 = (L1+2*M1)*L1111;
        A2 = (L1+2*M1)*L1112;

        B1d12(i,j) = (L1+M1)*X(i,j);
        B2d11(i,j) = M1*IX(i,j);
        B2d22(i,j) = (L1+2*M1)*X(i,j);
        B1d1(i,j) = - 2*M1*L112(i,j)*X(i,j);
        B1d2(i,j) = (L1+M1)*L111(i,j)*X(i,j);
        B2d1(i,j) = - M1*L111(i,j)*IX(i,j);
        B2d2(i,j) = (L1+2*M1)*L112(i,j)*X(i,j);
        B1 = L1*L1112*X(i,j) - 2*M1*L111(i,j)*L112(i,j)*X(i,j);
        B2 = L1*L1122*X(i,j) - 2*M1*L112(i,j)*L112(i,j)*X(i,j);

        aa(1,1) = 2*A1d11*Id11      + 2*A1d22(i,j)*Id22 - A1;
        aa(1,2) = - A2;
        aa(2,1) = - B1;
        aa(2,2) = 2*B2d11(i,j)*Id11 + 2*B2d22(i,j)*Id22 - B2;

        J = inv(aa);

        JM11(i,j) = J(1,1);
        JM12(i,j) = J(1,2);
        JM21(i,j) = J(2,1);
        JM22(i,j) = J(2,2);

    end
end

for i = 1:m

    aa(1,1) = 3*Id2*M1 + 3*ID2*M2;
    aa(1,2) = 0;
    aa(2,1) = - L1*L111(i,1) + L2*L111(i,1);
    aa(2,2) = - L1*L112(i,1) + L2*L112(i,1) + 3*Id2*(L1+2*M1) + 3*ID2*(L2+2*M2);

    J = inv(aa);

    JBBB11(i) = J(1,1);
```

```
    JBBB12(i) = J(1,2);
    JBBB21(i) = J(2,1);
    JBBB22(i) = J(2,2);

    aa(1,1) = -3*Id2;
    aa(1,2) = 0;
    aa(2,1) = - L1*L111(i,n1);
    aa(2,2) = - L1*L112(i,n1) - 3*Id2*(L1+2*M1);

    J = inv(aa);

    JBB11(i) = J(1,1);
    JBB12(i) = J(1,2);
    JBB21(i) = J(2,1);
    JBB22(i) = J(2,2);

end

for j = 1:n1

    aa(1,1) = 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(1,j);
    aa(1,2) = - (L1+2*M1)*L112(1,j);
    aa(2,1) = 0;
    aa(2,2) = 3*Id1;

    J = inv(aa);

    J1S11(j) = J(1,1);
    J1S12(j) = J(1,2);
    J1S21(j) = J(2,1);
    J1S22(j) = J(2,2);

    aa(1,1) =  - 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(m,j);
    aa(1,2) = - (L1+2*M1)*L112(m,j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Id1;

    J = inv(aa);

    JmS11(j) = J(1,1);
    JmS12(j) = J(1,2);
    JmS21(j) = J(2,1);
    JmS22(j) = J(2,2);

end

%% Curvature terms of the foundation
for i = 1:m

    x1 = (i-1)*dx1;
```

```
for j = 1:n2

    x2 = (j-1)*dX2 - Thickness2;

    alpha2 = (b*sin(x1))^2 + (a*cos(x1))^2;
    alpha = sqrt(alpha2);
    Ialpha = 1/alpha;
    Ialpha2 = 1/alpha2;
    alphad1 = 0.5*(b^2-a^2)*sin(2*x1)*Ialpha;
    alphad11 = ((b^2-a^2)*cos(2*x1) - alphad1^2)*Ialpha;

    Y(i,j) = alpha + a*b*Ialpha2*x2;
    IY(i,j) = 1/Y(i,j);
    Yd1 = alphad1*(1-2*a*b*x2*Ialpha*Ialpha2);
    Yd11 = alphad11*(1-2*a*b*x2*Ialpha*Ialpha2) + 6*a*b*x2*(alphad1*Ialpha2)^2;
    Yd2 = a*b*Ialpha2;
    Yd12 = -2*a*b*alphad1*Ialpha*Ialpha2;

    LL111(i,j) = Yd1*IY(i,j);
    LL112(i,j) = Yd2*IY(i,j);
    LL1111 = Yd11*IY(i,j)-LL111(i,j)^2;
    LL1112 = Yd12*IY(i,j)-LL111(i,j)*LL112(i,j);
    LL1122 = -LL112(i,j)^2;

    C1d22(i,j) = M2*Y(i,j)^2;
    C1d1(i,j) = (L2+2*M2)*LL111(i,j);
    C1d2(i,j) = 3*M2*LL112(i,j)*Y(i,j)^2;
    C2d1(i,j) = (L2+3*M2)*LL112(i,j);
    C1 = (L2+2*M2)*LL1111;
    C2 = (L2+2*M2)*LL1112;

    D1d12(i,j) = (L2+M2)*Y(i,j);
    D2d11(i,j) = M2*IY(i,j);
    D2d22(i,j) = (L2+2*M2)*Y(i,j);
    D1d1(i,j) = - 2*M2*LL112(i,j)*Y(i,j);
    D1d2(i,j) = (L2+M2)*LL111(i,j)*Y(i,j);
    D2d1(i,j) = - M2*LL111(i,j)*IY(i,j);
    D2d2(i,j) = (L2+2*M2)*LL112(i,j)*Y(i,j);
    D1 = L2*LL1112*Y(i,j) - 2*M2*LL111(i,j)*LL112(i,j)*Y(i,j);
    D2 = L2*LL1122*Y(i,j) - 2*M2*LL112(i,j)*LL112(i,j)*Y(i,j);

    aa(1,1) = 2*C1d11*Id11      + 2*C1d22(i,j)*ID22 - C1;
    aa(1,2) = - C2;
    aa(2,1) = - D1;
    aa(2,2) = 2*D2d11(i,j)*Id11 + 2*D2d22(i,j)*ID22 - D2;

    J = inv(aa);

    JF11(i,j) = J(1,1);
    JF12(i,j) = J(1,2);
    JF21(i,j) = J(2,1);
```

```
            JF22(i,j) = J(2,2);

        end
    end

    for j = 1:n2

        aa(1,1) = 3*Id1*(L2+2*M2) - (L2+2*M2)*LL111(1,j);
        aa(1,2) = - (L2+2*M2)*LL112(1,j);
        aa(2,1) = 0;
        aa(2,2) = 3*Id1;

        J = inv(aa);

        J1B11(j) = J(1,1);
        J1B12(j) = J(1,2);
        J1B21(j) = J(2,1);
        J1B22(j) = J(2,2);

        aa(1,1) = - 3*Id1*(L2+2*M2) - (L2+2*M2)*LL111(m,j);
        aa(1,2) = - (L2+2*M2)*LL112(m,j);
        aa(2,1) = 0;
        aa(2,2) = - 3*Id1;

        J = inv(aa);

        JmB11(j) = J(1,1);
        JmB12(j) = J(1,2);
        JmB21(j) = J(2,1);
        JmB22(j) = J(2,2);

    end


    for i = 1:m

        aa(1,1) = -3*Id2;
        aa(1,2) = 0;
        aa(2,1) = - L2*LL111(i,n2);
        aa(2,2) = - L2*LL112(i,n2) - 3*(L2+2*M2)*ID2;

        J = inv(aa);

        JnB11(i) = J(1,1);
        JnB12(i) = J(1,2);
        JnB21(i) = J(2,1);
        JnB22(i) = J(2,2);

    end

    aa(1,1) = 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(1,1);
```

```
aa(1,2) = 3*Id2*L1 - (L1+2*M1)*L112(1,1);
aa(2,1) = 3*Id1*(L2+2*M2) - (L2+2*M2)*L111(1,1);
aa(2,2) = - 3*ID2*L2 - (L2+2*M2)*L112(1,1);

J = inv(aa);

J11BBB11 = J(1,1);
J11BBB12 = J(1,2);
J11BBB21 = J(2,1);
J11BBB22 = J(2,2);

aa(1,1) = - 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(m,1);
aa(1,2) = 3*Id2*L1 - (L1+2*M1)*L112(m,1);
aa(2,1) = - 3*Id1*(L2+2*M2) - (L2+2*M2)*L111(m,1);
aa(2,2) = - 3*ID2*L2 - (L2+2*M2)*L112(m,1);

J = inv(aa);

Jm1BBB11 = J(1,1);
Jm1BBB12 = J(1,2);
Jm1BBB21 = J(2,1);
Jm1BBB22 = J(2,2);

aa(1,1) = 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(1,n1);
aa(1,2) = -3*Id2*L1 - (L1+2*M1)*L112(1,n1);
aa(2,1) = 3*Id1*L1 - L1*L111(1,n1);
aa(2,2) = - 3*(L1+2*M1)*Id2 - L1*L112(1,n1);

J = inv(aa);

J1nBB11 = J(1,1);
J1nBB12 = J(1,2);
J1nBB21 = J(2,1);
J1nBB22 = J(2,2);

aa(1,1) = - 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(m,n1);
aa(1,2) = - 3*Id2*L1 - (L1+2*M1)*L112(m,n1);
aa(2,1) = - 3*Id1*L1 - L1*L111(m,n1);
aa(2,2) = - 3*(L1+2*M1)*Id2 - L1*L112(m,n1);

J = inv(aa);

JmnBB11 = J(1,1);
JmnBB12 = J(1,2);
JmnBB21 = J(2,1);
JmnBB22 = J(2,2);

%% Main code
while errr < p

    pp = norm(v1,2) + norm(v2,2);
```

```
for k = 1:NNN

    %% Corners
    u1d1 = (4*u1(2,1)-u1(3,1))*Id1;
    u2d2 = (4*u2(1,2)-u2(1,3))*Id2;
    v1d1 = (4*v1(2,n2)-v1(3,n2))*Id1;
    v2d2 = -(4*v2(1,n2-1)-v2(1,n2-2))*ID2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T1; % equation (24)
    Xu2 = (L2+2*M2)*v1d1 + L2*v2d2; % equation (27)

    u1(1,1) = J11BBB11*Xu1 + J11BBB12*Xu2;
    u2(1,1) = J11BBB21*Xu1 + J11BBB22*Xu2;

    v1(1,n2) = u1(1,1);
    v2(1,n2) = u2(1,1);

    u1d1 = -(4*u1(m-1,1)-u1(m-2,1))*Id1;
    u2d2 = (4*u2(m,2)-u2(m,3))*Id2;
    v1d1 = -(4*v1(m-1,n2)-v1(m-2,n2))*Id1;
    v2d2 = -(4*v2(m,n2-1)-v2(m,n2-2))*ID2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T2; % equation (25)
    Xu2 = (L2+2*M2)*v1d1 + L2*v2d2; % equation (27)

    u1(m,1) = Jm1BBB11*Xu1 + Jm1BBB12*Xu2;
    u2(m,1) = Jm1BBB21*Xu1 + Jm1BBB22*Xu2;

    v1(m,n2) = u1(m,1);
    v2(m,n2) = u2(m,1);

    u1d1 = (4*u1(2,n1)-u1(3,n1))*Id1;
    u2d2 = -(4*u2(1,n1-1)-u2(1,n1-2))*Id2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T1; % equation (24)
    Xu2 = L1*u1d1 + (L1+2*M1)*u2d2; % equation (27)

    u1(1,n1) = J1nBB11*Xu1 + J1nBB12*Xu2;
    u2(1,n1) = J1nBB21*Xu1 + J1nBB22*Xu2;

    u1d1 = -(4*u1(m-1,n1)-u1(m-2,n1))*Id1;
    u2d2 = -(4*u2(m,n1-1)-u2(m,n1-2))*Id2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T2; % equation (25)
    Xu2 = L1*u1d1 + (L1+2*M1)*u2d2; % equation (27)

    u1(m,n1) = JmnBB11*Xu1 + JmnBB12*Xu2;
    u2(m,n1) = JmnBB21*Xu1 + JmnBB22*Xu2;

    %% Bonded boundary
```

```
for i = 2:m-1 % equation (31),(32), (33) and (34)

    u1d1 = (u1(i+1,1)-u1(i-1,1))*Id1;
    u2d1 = (u2(i+1,1)-u2(i-1,1))*Id1;
    u1d2 = (4*u1(i,2)-u1(i,3))*Id2;
    u2d2 = (4*u2(i,2)-u2(i,3))*Id2;

    v1d1 = (v1(i+1,n2)-v1(i-1,n2))*Id1;
    v2d1 = (v2(i+1,n2)-v2(i-1,n2))*Id1;
    v1d2 = -(4*v1(i,n2-1)-v1(i,n2-2))*ID2;
    v2d2 = -(4*v2(i,n2-1)-v2(i,n2-2))*ID2;

    Xu1 = M1*(u1d2 + (IX(i,1)^2)*u2d1);
    Xu2 = L1*u1d1 + (L1+2*M1)*u2d2;
    Xv1 = M2*(v1d2 + (IX(i,1)^2)*v2d1);
    Xv2 = L2*v1d1 + (L2+2*M2)*v2d2;

    u1(i,1)  = JBBB11(i)*(Xu1-Xv1) + JBBB12(i)*(Xu2-Xv2);
    u2(i,1)  = JBBB21(i)*(Xu1-Xv1) + JBBB22(i)*(Xu2-Xv2);

    v1(i,n2) = u1(i,1);
    v2(i,n2) = u2(i,1);

    u1d1 = (u1(i+1,n1)-u1(i-1,n1))*Id1;
    u1d2 = -(4*u1(i,n1-1)-u1(i,n1-2))*Id2;
    u2d1 = (u2(i+1,n1)-u2(i-1,n1))*Id1;
    u2d2 = -(4*u2(i,n1-1)-u2(i,n1-2))*Id2;

    Xu1 = u1d2 + (IX(i,n1)^2)*u2d1;
    Xu2 = L1*u1d1 + (L1+2*M1)*u2d2;

    u1(i,n1) = JBB11(i)*Xu1 + JBB12(i)*Xu2;
    u2(i,n1) = JBB21(i)*Xu1 + JBB22(i)*Xu2;

end

%% Stressed boundaries of the overlying body
for j = 2:n1-1

    u1d1 = (4*u1(2,j)-u1(3,j))*Id1;
    u1d2 = (u1(1,j+1)-u1(1,j-1))*Id2;
    u2d1 = (4*u2(2,j)-u2(3,j))*Id1;
    u2d2 = (u2(1,j+1)-u2(1,j-1))*Id2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T1; % equation (27)
    Xu2 = u2d1 + (X(1,j)^2)*u1d2; % equation (30)

    u1(1,j) = J1S11(j)*Xu1 + J1S12(j)*Xu2;
    u2(1,j) = J1S21(j)*Xu1 + J1S22(j)*Xu2;

    u1d1 = -(4*u1(m-1,j)-u1(m-2,j))*Id1;
```

```
    u1d2 = (u1(m,j+1)-u1(m,j-1))*Id2;
    u2d1 = -(4*u2(m-1,j)-u2(m-2,j))*Id1;
    u2d2 = (u2(m,j+1)-u2(m,j-1))*Id2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T2; % equation (28)
    Xu2 = u2d1 + (X(m,j)^2)*u1d2; % equation (30)

    u1(m,j) = JmS11(j)*Xu1 + JmS12(j)*Xu2;
    u2(m,j) = JmS21(j)*Xu1 + JmS22(j)*Xu2;

end

%% Stress-free boundary of the overlying body
for j = 2:n2-1

    v1d1 = (4*v1(2,j)-v1(3,j))*Id1;
    v2d1 = (4*v2(2,j)-v2(3,j))*Id1;
    v1d2 = (v1(1,j+1)-v1(1,j-1))*ID2;
    v2d2 = (v2(1,j+1)-v2(1,j-1))*ID2;

    Xv1 = (L2+2*M2)*v1d1 + L2*v2d2; % equation (29)
    Xv2 = v2d1 + (Y(1,j)^2)*v1d2; % equation (30)

    v1(1,j) = J1B11(j)*Xv1 + J1B12(j)*Xv2;
    v2(1,j) = J1B21(j)*Xv1 + J1B22(j)*Xv2;

    v1d1 = -(4*v1(m-1,j)-v1(m-2,j))*Id1;
    v2d1 = -(4*v2(m-1,j)-v2(m-2,j))*Id1;
    v1d2 = (v1(m,j+1)-v1(m,j-1))*ID2;
    v2d2 = (v2(m,j+1)-v2(m,j-1))*ID2;

    Xv1 = (L2+2*M2)*v1d1 + L2*v2d2; % equation (29)
    Xv2 = v2d1 + (Y(m,j)^2)*v1d2; % equation (30)

    v1(m,j) = JmB11(j)*Xv1 + JmB12(j)*Xv2;
    v2(m,j) = JmB21(j)*Xv1 + JmB22(j)*Xv2;

end

%% Governing equations of the overlying body
for i = 2:m-1

    for j = 2:n1-1
        u1d11 = (u1(i+1,j)+u1(i-1,j))*Id11;
        u1d22 = (u1(i,j+1)+u1(i,j-1))*Id22;
        u1d12 = (u1(i+1,j+1)-u1(i+1,j-1)-u1(i-1,j+1)+u1(i-1,j-1))*Id12;
        u1d1 = (u1(i+1,j)-u1(i-1,j))*Id1;
        u1d2 = (u1(i,j+1)-u1(i,j-1))*Id2;

        u2d11 = (u2(i+1,j)+u2(i-1,j))*Id11;
        u2d22 = (u2(i,j+1)+u2(i,j-1))*Id22;
```

```
                u2d12 = (u2(i+1,j+1)-u2(i+1,j-1)-u2(i-1,j+1)+u2(i-1,j-1))*Id12;
                u2d1 = (u2(i+1,j)-u2(i-1,j))*Id1;
                u2d2 = (u2(i,j+1)-u2(i,j-1))*Id2;

                Xu1 = A1d11*u1d11      + A1d22(i,j)*u1d22 + A2d12*u2d12 ...
                    + A1d1(i,j)*u1d1 + A1d2(i,j)*u1d2 + A2d1(i,j)*u2d1; % equation (25)
                Xu2 = B2d11(i,j)*u2d11 + B2d22(i,j)*u2d22 + B1d12(i,j)*u1d12 ...
                    + B1d1(i,j)*u1d1 + B1d2(i,j)*u1d2 ...
                    + B2d1(i,j)*u2d1 + B2d2(i,j)*u2d2; % equation (26)

                u1(i,j) = JM11(i,j)*Xu1 + JM12(i,j)*Xu2;
                u2(i,j) = JM21(i,j)*Xu1 + JM22(i,j)*Xu2;

            end
        end

        %% Governing equations of the foundation
        for i = 2:m-1

            for j = 2:n2-1

                v1d11 = (v1(i+1,j)+v1(i-1,j))*Id11;
                v1d22 = (v1(i,j+1)+v1(i,j-1))*ID22;
                v1d12 = (v1(i+1,j+1)-v1(i+1,j-1)-v1(i-1,j+1)+v1(i-1,j-1))*ID12;
                v1d1 = (v1(i+1,j)-v1(i-1,j))*Id1;
                v1d2 = (v1(i,j+1)-v1(i,j-1))*ID2;

                v2d11 = (v2(i+1,j)+v2(i-1,j))*Id11;
                v2d22 = (v2(i,j+1)+v2(i,j-1))*ID22;
                v2d12 = (v2(i+1,j+1)-v2(i+1,j-1)-v2(i-1,j+1)+v2(i-1,j-1))*ID12;
                v2d1 = (v2(i+1,j)-v2(i-1,j))*Id1;
                v2d2 = (v2(i,j+1)-v2(i,j-1))*ID2;

                Xv1 = C1d11*v1d11      + C1d22(i,j)*v1d22 + C2d12*v2d12      ...
                    + C1d1(i,j)*v1d1 + C1d2(i,j)*v1d2 + C2d1(i,j)*v2d1;
                Xv2 = D2d11(i,j)*v2d11 + D2d22(i,j)*v2d22 + D1d12(i,j)*v1d12 ...
                    + D1d1(i,j)*v1d1 + D1d2(i,j)*v1d2 + D2d1(i,j)*v2d1 + D2d2(i,j)*v2d2;

                v1(i,j) = JF11(i,j)*Xv1 + JF12(i,j)*Xv2;
                v2(i,j) = JF21(i,j)*Xv1 + JF22(i,j)*Xv2;

            end
        end
    end

    %% Terminating condition
    ppp = norm(v1,2) + norm(v2,2);
    p = abs(1- ppp/pp);

end
```

```
%% Contact region
V1(:) = v1(:,n2);
V2(:) = v2(:,n2);
```

2.4 Benchmark Model

In this section we present a benchmark numerical code for the model that we introduced in this chapter, i.e. `CH3.m`. To do so we consider the special case $a = b = 2$ (and $= \tau_0 = \tau_{\text{max}} = 1$), i.e. we consider polar coordinates.

Now, let `v1`$= u^2_{\text{shell}}$, `v2`$= u^3_{\text{shell}}$, `vv1`$= u^2_{\text{two-body}}$, `vv2`$= u^3_{\text{two-body}}$, `ww1`$= v^2$, `ww2`$= v^3$, `h`$= h$, `H`$= H$, `E1`$= E$, `E2`$= \bar{E}$, `Nu1`$= \nu$, `Nu2`$= \bar{\nu}$ and `NN`$= N$. Thus, we find:

```
function CH3
format long
%% Benchmark Model
% Overlying elastic body on an elastic cylinder: bonded case
% Contact angle is [0,pi]

%% INITIAL PARIMITERS
NN = 250; % Azimuthal grid points
error = 10^(-10); % Terminating error

E1 = 8000; % Young's modulus of the overlying body
E2 = 1000; % Young's modulus of the foundation

Nu1 = 0.25; % Poisson's ratio of the overlying body
Nu2 = 0.25; % Poisson's ratio of the foundation

h = 0.125; % Thickness of the overlying body
a = 2; % Outer radius
b = 1; % Inner radius

H = a-b; % Thickness of the foundation

%% DO NOT CHANGE!
L1 = Nu1*E1/((1+Nu1)*(1-2*Nu1));
L2 = Nu2*E2/((1+Nu2)*(1-2*Nu2));

M1 = 0.5*E1/(1+Nu1);
M2 = 0.5*E2/(1+Nu2);

LL = 4*M1*(M1+L1)/(L1+2*M1);
IL = 1/LL;

NNN = NN^2;
errr = NNN*error;

ah = a+h;
```

```
Iah = 1/ah;

m = NN; % Azimuthal grid points
q1 = H/(a*pi);
q2 = h/(ah*pi);
n1 = round(q1*NN-q1+1); % Radial grid points of the foundation
n2 = round(q2*NN-q2+1); % Radial grid points of the overlying body

dx1 = pi/(m-1); % Azimuthal grid spacing
dx2 = H/(n1-1); % Radial grid spacing of the foundation
dX2 = h/(n2-1); % Radial grid spacing of the overlying body

Ia = 1/a;

Idx1 = 1/(2*dx1);
Idx11 = (1/dx1)^2;

Idx2 = 1/(2*dx2);
Idx22 = (1/dx2)^2;

Idx12 = Idx1*Idx2;

ww = zeros(m,1); % Azimuthal displacement at the boundary (BALDELLI)

w1 = zeros(m,1);
w2 = zeros(m,1);

v1 = zeros(m,n1); % Azimuthal displacement of the foundation (SHELL)
v2 = zeros(m,n1); % Radial displacement of the foundation (SHELL)

vv1 = zeros(m,n1); % Azimuthal displacement of the foundation (TwoBody)
vv2 = zeros(m,n1); % Radial displacement of the foundation (TwoBody)

ww1 = zeros(m,n2); % Azimuthal displacement of the overlying body (TwoBody)
ww2 = zeros(m,n2); % Radial displacement of the overlying body (TwoBody)

aa = zeros(2,2);
aaa = zeros(3,3);

r = zeros(1,n1);
Ir = zeros(1,n1);

a1d11 = zeros(1,n1);
a1d22 = zeros(1,n1);
a1d2 = zeros(1,n1);
a2d12 = zeros(1,n1);
a2d1 = zeros(1,n1);

b1d12 = zeros(1,n1);
b1d1 = zeros(1,n1);
b2d11 = zeros(1,n1);
```

```
b2d22 = zeros(1,n1);
b2d2 = zeros(1,n1);

Iv1 = zeros(1,n1);
Iv2 = zeros(1,n1);

B1v11 = zeros(1,n1);
B1v12 = zeros(1,n1);
B1v21 = zeros(1,n1);
B1v22 = zeros(1,n1);

Bmv11 = zeros(1,n1);
Bmv12 = zeros(1,n1);
Bmv21 = zeros(1,n1);
Bmv22 = zeros(1,n1);

rr = zeros(1,n2);
Irr = zeros(1,n2);

aa1d11 = zeros(1,n2);
aa1d22 = zeros(1,n2);
aa1d2 = zeros(1,n2);
aa2d12 = zeros(1,n2);
aa2d1 = zeros(1,n2);

bb1d12 = zeros(1,n2);
bb1d1 = zeros(1,n2);
bb2d11 = zeros(1,n2);
bb2d22 = zeros(1,n2);
bb2d2 = zeros(1,n2);

Iw1 = zeros(1,n2);
Iw2 = zeros(1,n2);

B1w11 = zeros(1,n2);
B1w12 = zeros(1,n2);
B1w21 = zeros(1,n2);
B1w22 = zeros(1,n2);

Bmw11 = zeros(1,n2);
Bmw12 = zeros(1,n2);
Bmw21 = zeros(1,n2);
Bmw22 = zeros(1,n2);

Baldelli1 = zeros(m,1);
Baldelli2 = zeros(m,1);
Shell1 = zeros(m,1);
Shell2 = zeros(m,1);

%% Baldelli and and Bourdin's Model
alpha2 = M2*(a^2)/(LL*H*h);
```

```
alpha = (alpha2)^(0.5);
C = 1/(LL*alpha*cosh(0.5*alpha*pi));

for i = 1:m
    x1 = - 0.5*pi + dx1*(i-1);
    ww(i) = C*sinh(alpha*x1);
end


%% Bonded Shell Model
A1d11 = 1+(4/3)*(h/a)^2;
A1d2 = - M2*IL*(a^2)/h;
A2d111 = - (2/3)*(h^2)*a^(-3);
A2d1 = (1/a) + (2/3)*(h^2)*a^(-3) - M2*IL/h;

B2d1111 = (1/3)*(h^2)*a^(-3);
B2d11 = - (2/3)*(h^2)*a^(-3);
B2 = (1/a) + (1/3)*(h^2)*a^(-3) + L2*IL/h;
B2d2 = (L2+2*M2)*IL*(a/h);
B1d111 = -(2/3)*(h/a)^2;
B1d1 = 1 + (2/3)*(h/a)^2 + L2*IL*(a/h);

Iu1 = 1/(2*A1d11*Idx11 - 3*A1d2*Idx2);
Iu2 = 1/(2*(B2d11-2*B2d1111*Idx11)*Idx11 - B2 - 3*B2d2*Idx2);

%% Curvature terms of the foundation
for j = 1:n1

    r(j) = b + (j-1)*dx2;
    Ir(j) = 1/r(j);

    a1d11(j) = (L2+2*M2);
    a1d22(j) = M2*r(j)^2;
    a1d2(j) = 3*M2*r(j);
    a2d12(j) = (L2+M2);
    a2d1(j) = (L2+3*M2)*Ir(j);

    b1d12(j) = (L2+M2)*r(j);
    b1d1(j) = -2*M2;
    b2d11(j) = M2*Ir(j);
    b2d22(j) = (L2+2*M2)*r(j);
    b2d2(j) = (L2+2*M2);
    b2 = -(L2+2*M2)*Ir(j);

    Iv1(j) = 1/(2*a1d11(j)*Idx11 + 2*a1d22(j)*Idx22);
    Iv2(j) = 1/(2*b2d11(j)*Idx11 + 2*b2d22(j)*Idx22 - b2);

    aa(1,1) = 3*Idx1*(L2+2*M2);
    aa(1,2) = - (L2+2*M2)*Ir(j);
    aa(2,1) = 0;
    aa(2,2) = 3*Idx1;
```

```
    J = inv(aa);

    B1v11(j) = J(1,1);
    B1v12(j) = J(1,2);
    B1v21(j) = J(2,1);
    B1v22(j) = J(2,2);

    aa(1,1) = - 3*Idx1*(L2+2*M2);
    aa(1,2) = - (L2+2*M2)*Ir(j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Idx1;

    J = inv(aa);

    Bmv11(j) = J(1,1);
    Bmv12(j) = J(1,2);
    Bmv21(j) = J(2,1);
    Bmv22(j) = J(2,2);

end

aaa(1,1) = 3*Idx1;
aaa(1,2) = - Ia;
aaa(1,3) = 0;
aaa(2,1) = 0;
aaa(2,2) = 3*Idx1;
aaa(2,3) = 0;
aaa(3,1) = 4*Idx11;
aaa(3,2) = 0;
aaa(3,3) = 3*Ia*Idx1;

JSB1 = inv(aaa);

aaa(1,1) = -3*Idx1;
aaa(1,2) = - Ia;
aaa(1,3) = 0;
aaa(2,1) = 0;
aaa(2,2) = -3*Idx1;
aaa(2,3) = 0;
aaa(3,1) = 4*Idx11;
aaa(3,2) = 0;
aaa(3,3) = -3*Ia*Idx1;

JSBm = inv(aaa);

p = 2;

%% Main code
while errr < p
```

```
pp = norm(v1,2) + norm(v2,2);

for k = 1:NNN

    %% Boundary conditions of the shell
    w1(1) = (2*v1(1,n1)-5*v1(2,n1)+4*v1(3,n1)-v1(4,n1))*Idx11;

    v1d1 = (4*v1(2,n1)-v1(3,n1))*Idx1;
    v2d1 = (4*v2(2,n1)-v2(3,n1))*Idx1;
    w2d1 = (4*w2(2)-w2(3))*Idx1;

    Xv1 = v1d1 - IL;
    Xv2 = v2d1;
    Xw2 = - 2*(-5*v1(2,n1)+4*v1(3,n1)-v1(4,n1))*Idx11 + Ia*w2d1;

    v1(1,n1) = Xv1*JSB1(1,1) + Xv2*JSB1(1,2) + Xw2*JSB1(1,3);
    v2(1,n1) = Xv1*JSB1(2,1) + Xv2*JSB1(2,2) + Xw2*JSB1(2,3);
    w2(1) = Xv1*JSB1(3,1) + Xv2*JSB1(3,2) + Xw2*JSB1(3,3);

    v1d1 = -(4*v1(m-1,n1)-v1(m-2,n1))*Idx1;
    v2d1 = -(4*v2(m-1,n1)-v2(m-2,n1))*Idx1;
    w2d1 = -(4*w2(m-1)-w2(m-2))*Idx1;

    w1(m) = (2*v1(m,n1)-5*v1(m-1,n1)+4*v1(m-2,n1)-v1(m-3,n1))*Idx11;

    Xv1 = v1d1 - IL;
    Xv2 = v2d1;
    Xw2 = - 2*(-5*v1(m-1,n1)+4*v1(m-2,n1)-v1(m-3,n1))*Idx11 + Ia*w2d1;

    v1(m,n1) = Xv1*JSBm(1,1) + Xv2*JSBm(1,2) + Xw2*JSBm(1,3);
    v2(m,n1) = Xv1*JSBm(2,1) + Xv2*JSBm(2,2) + Xw2*JSBm(2,3);
    w2(m) = Xv1*JSBm(3,1) + Xv2*JSBm(3,2) + Xw2*JSBm(3,3);

    %% Governing equations of the shell
    for i = 2:m-1

        w1(i) = (v1(i+1,n1)-2*v1(i,n1)+v1(i-1,n1))*Idx11;
        w2(i) = (v2(i+1,n1)-2*v2(i,n1)+v2(i-1,n1))*Idx11;

        v1d1 = (v1(i+1,n1)-v1(i-1,n1))*Idx1;
        v2d1 = (v2(i+1,n1)-v2(i-1,n1))*Idx1;

        v1d11 = (v1(i+1,n1)+v1(i-1,n1))*Idx11;
        v2d11 = (v2(i+1,n1)+v2(i-1,n1))*Idx11;

        w1d1 = (w1(i+1)-w1(i-1))*Idx1;
        w2d1 = (w2(i+1)-w2(i-1))*Idx1;

        w2d11 = (w2(i+1)+w2(i-1))*Idx11;

        v1d2 = -(4*v1(i,n1-1)-v1(i,n1-2))*Idx2;
```

```
    v2d2 = -(4*v2(i,n1-1)-v2(i,n1-2))*Idx2;

    Xv1 = A1d11*v1d11 + A1d2*v1d2 + A2d111*w2d1 + A2d1*v2d1;
    Xv2 = B2d1111*w2d11 + (B2d11-2*B2d1111*Idx11)*v2d11 + B2d2*v2d2 ...
        + B1d111*w1d1 + B1d1*v1d1;

    v1(i,n1) = Xv1*Iu1;
    v2(i,n1) = Xv2*Iu2;

end

%% Stress-free boundary of the foundation
for j = 2:n1-1

    v1d1 = (4*v1(2,j)-v1(3,j))*Idx1;
    v2d1 = (4*v2(2,j)-v2(3,j))*Idx1;
    v1d2 = (v1(1,j+1)-v1(1,j-1))*Idx2;
    v2d2 = (v2(1,j+1)-v2(1,j-1))*Idx2;

    Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
    Xv2 = v2d1 + (r(j)^2)*v1d2;

    v1(1,j) = B1v11(j)*Xv1 + B1v12(j)*Xv2;
    v2(1,j) = B1v21(j)*Xv1 + B1v22(j)*Xv2;

    v1d1 = -(4*v1(m-1,j)-v1(m-2,j))*Idx1;
    v2d1 = -(4*v2(m-1,j)-v2(m-2,j))*Idx1;
    v1d2 = (v1(m,j+1)-v1(m,j-1))*Idx2;
    v2d2 = (v2(m,j+1)-v2(m,j-1))*Idx2;

    Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
    Xv2 = v2d1 + (r(j)^2)*v1d2;

    v1(m,j) = Bmv11(j)*Xv1 + Bmv12(j)*Xv2;
    v2(m,j) = Bmv21(j)*Xv1 + Bmv22(j)*Xv2;

end

%% Governing equations of the foundation
for i = 2:m-1

    for j = 2:n1-1

        v1d1 = (v1(i+1,j)-v1(i-1,j))*Idx1;
        v1d2 = (v1(i,j+1)-v1(i,j-1))*Idx2;
        v2d1 = (v2(i+1,j)-v2(i-1,j))*Idx1;
        v2d2 = (v2(i,j+1)-v2(i,j-1))*Idx2;

        v1d11 = (v1(i+1,j)+v1(i-1,j))*Idx11;
        v1d22 = (v1(i,j+1)+v1(i,j-1))*Idx22;
        v2d11 = (v2(i+1,j)+v2(i-1,j))*Idx11;
```

```
                v2d22 = (v2(i,j+1)+v2(i,j-1))*Idx22;

                v1d12 = (v1(i+1,j+1)-v1(i-1,j+1)-v1(i+1,j-1)+v1(i-1,j-1))*Idx12;
                v2d12 = (v2(i+1,j+1)-v2(i-1,j+1)-v2(i+1,j-1)+v2(i-1,j-1))*Idx12;

                Xv1 = a1d11(j)*v1d11 + a1d22(j)*v1d22 + a1d2(j)*v1d2 ...
                    + a2d12(j)*v2d12 + a2d1(j)*v2d1;
                Xv2 = b1d12(j)*v1d12 + b1d1(j)*v1d1 + b2d11(j)*v2d11 ...
                    + b2d22(j)*v2d22 + b2d2(j)*v2d2;

                v1(i,j) = Xv1*Iv1(j);
                v2(i,j) = Xv2*Iv2(j);

            end
        end
    end

    %% Terminating condition
    ppp = norm(v1,2) + norm(v2,2);
    p = abs(1-ppp/pp);

end

%% Two-Body Model
IdX2 = 1/(2*dX2);
IdX22 = (1/dX2)^2;

IdX12 = Idx1*IdX2;

Bn2ww1 = 1/(-3*IdX2*ah^2);
Bn2ww2 = 1/(-3*(L1+2*M1)*IdX2 - L1*Iah);

B1ww1 = 1/(3*M1*IdX2*a^2 + 3*M2*Idx2*a^2);
B1ww2 = 1/(3*(L1+2*M1)*IdX2 + 3*(L2+2*M2)*Idx2 - L1*Ia + L2*Ia);

aa(1,1) = 3*(L1+2*M1)*Idx1;
aa(1,2) = - (L1+2*M1)*Iah - 3*L1*IdX2;
aa(2,1) = 3*L1*Idx1;
aa(2,2) = - L1*Iah - 3*(L1+2*M1)*IdX2;

B1n2ww = inv(aa);

aa(1,1) = - 3*(L1+2*M1)*Idx1;
aa(1,2) = - (L1+2*M1)*Iah - 3*L1*IdX2;
aa(2,1) = - 3*L1*Idx1;
aa(2,2) = - L1*Iah - 3*(L1+2*M1)*IdX2;

Bmn2ww = inv(aa);

aa(1,1) = 3*(L1+2*M1)*Idx1;
aa(1,2) = - (L1+2*M1)*Ia + 3*L1*IdX2;
```

```
aa(2,1) = 3*(L2+2*M2)*Idx1;
aa(2,2) = - (L2+2*M2)*Ia - 3*L2*Idx2;

B11ww = inv(aa);

aa(1,1) = - 3*(L1+2*M1)*Idx1;
aa(1,2) = - (L1+2*M1)*Ia + 3*L1*IdX2;
aa(2,1) = - 3*(L2+2*M2)*Idx1;
aa(2,2) = - (L2+2*M2)*Ia - 3*L2*Idx2;

Bm1ww = inv(aa);

%% Curvature terms of the overlying body
for j = 1:n2

    rr(j) = a + (j-1)*dX2;
    Irr(j) = 1/rr(j);

    aa1d11(j) = (L1+2*M1);
    aa1d22(j) = M1*rr(j)^2;
    aa1d2(j) = 3*M1*rr(j);
    aa2d12(j) = (L1+M1);
    aa2d1(j) = (L1+3*M1)*Irr(j);

    bb1d12(j) = (L1+M1)*rr(j);
    bb1d1(j) = -2*M1;
    bb2d11(j) = M1*Irr(j);
    bb2d22(j) = (L1+2*M1)*rr(j);
    bb2d2(j) = (L1+2*M1);
    bb2 = -(L1+2*M1)*Irr(j);

    Iw1(j) = 1/(2*aa1d11(j)*Idx11 + 2*aa1d22(j)*IdX22);
    Iw2(j) = 1/(2*bb2d11(j)*Idx11 + 2*bb2d22(j)*IdX22 - bb2);

    aa(1,1) = 3*Idx1*(L1+2*M1);
    aa(1,2) = - (L1+2*M1)*Irr(j);
    aa(2,1) = 0;
    aa(2,2) = 3*Idx1;

    J = inv(aa);

    B1w11(j) = J(1,1);
    B1w12(j) = J(1,2);
    B1w21(j) = J(2,1);
    B1w22(j) = J(2,2);

    aa(1,1) = - 3*Idx1*(L1+2*M1);
    aa(1,2) = - (L1+2*M1)*Irr(j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Idx1;
```

```
    J = inv(aa);

    Bmw11(j) = J(1,1);
    Bmw12(j) = J(1,2);
    Bmw21(j) = J(2,1);
    Bmw22(j) = J(2,2);

end

p = 2;

%% Main code
while errr < p

    pp = norm(vv1,2) + norm(vv2,2);

    for k = 1:NNN

        %% Corners
        ww1d1 = (4*ww1(2,n2)-ww1(3,n2))*Idx1;
        ww2d2 = -(4*ww2(1,n2-1)-ww2(1,n2-2))*IdX2;

        Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - 1;
        Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

        ww1(1,n2) = B1n2ww(1,1)*Xww1 + B1n2ww(1,2)*Xww2;
        ww2(1,n2) = B1n2ww(2,1)*Xww1 + B1n2ww(2,2)*Xww2;

        ww1d1 = -(4*ww1(m-1,n2)-ww1(m-2,n2))*Idx1;
        ww2d2 = -(4*ww2(m,n2-1)-ww2(m,n2-2))*IdX2;

        Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - 1;
        Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

        ww1(m,n2) = Bmn2ww(1,1)*Xww1 + Bmn2ww(1,2)*Xww2;
        ww2(m,n2) = Bmn2ww(2,1)*Xww1 + Bmn2ww(2,2)*Xww2;

        ww1d1 = (4*ww1(2,1)-ww1(3,1))*Idx1;
        ww2d2 = (4*ww2(1,2)-ww2(1,3))*IdX2;
        vv1d1 = (4*vv1(2,n1)-vv1(3,n1))*Idx1;
        vv2d2 = -(4*vv2(1,n1-1)-vv2(1,n1-2))*Idx2;

        Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - 1;
        Xww2 = (L2+2*M2)*vv1d1 + L2*vv2d2;

        ww1(1,1) = B11ww(1,1)*Xww1 + B11ww(1,2)*Xww2;
        ww2(1,1) = B11ww(2,1)*Xww1 + B11ww(2,2)*Xww2;

        vv1(1,n1) = ww1(1,1);
        vv2(1,n1) = ww2(1,1);
```

```
ww1d1 = -(4*ww1(m-1,1)-ww1(m-2,1))*Idx1;
ww2d2 = (4*ww2(m,2)-ww2(m,3))*IdX2;
vv1d1 = -(4*vv1(m-1,n1)-vv1(m-2,n1))*Idx1;
vv2d2 = -(4*vv2(m,n1-1)-vv2(m,n1-2))*Idx2;

Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - 1;
Xww2 = (L2+2*M2)*vv1d1 + L2*vv2d2;

ww1(m,1) = Bm1ww(1,1)*Xww1 + Bm1ww(1,2)*Xww2;
ww2(m,1) = Bm1ww(2,1)*Xww1 + Bm1ww(2,2)*Xww2;

vv1(m,n1) = ww1(m,1);
vv2(m,n1) = ww2(m,1);

for i = 2:m-1

    %% Stress-free boundary of the overlying body
    ww1d1 = (ww1(i+1,n2)-ww1(i-1,n2))*Idx1;
    ww2d1 = (ww2(i+1,n2)-ww2(i-1,n2))*Idx1;
    ww1d2 = -(4*ww1(i,n2-1)-ww1(i,n2-2))*IdX2;
    ww2d2 = -(4*ww2(i,n2-1)-ww2(i,n2-2))*IdX2;

    Xww1 = ww1d2*ah^2 + ww2d1;
    Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

    ww1(i,n2) = Bn2ww1*Xww1;
    ww2(i,n2) = Bn2ww2*Xww2;

    %% Bonded boundary
    ww1d1 = (ww1(i+1,1)-ww1(i-1,1))*Idx1;
    ww2d1 = (ww2(i+1,1)-ww2(i-1,1))*Idx1;
    ww1d2 = (4*ww1(i,2)-ww1(i,3))*IdX2;
    ww2d2 = (4*ww2(i,2)-ww2(i,3))*IdX2;

    Xww1 = M1*(ww1d2*a^2 + ww2d1);
    Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

    vv1d1 = (vv1(i+1,n1)-vv1(i-1,n1))*Idx1;
    vv2d1 = (vv2(i+1,n1)-vv2(i-1,n1))*Idx1;
    vv1d2 = -(4*vv1(i,n1-1)-vv1(i,n1-2))*Idx2;
    vv2d2 = -(4*vv2(i,n1-1)-vv2(i,n1-2))*Idx2;

    Xvv1 = M2*(vv1d2*a^2 + vv2d1);
    Xvv2 = L2*vv1d1 + (L2+2*M2)*vv2d2;

    ww1(i,1) = B1ww1*(Xww1 - Xvv1);
    ww2(i,1) = B1ww2*(Xww2 - Xvv2);

    vv1(i,n1) = ww1(i,1);
    vv2(i,n1) = ww2(i,1);
```

```
end

%% Stressed boundary of the overlying body
for j = 2:n2-1

    ww1d1 = (4*ww1(2,j)-ww1(3,j))*Idx1;
    ww2d1 = (4*ww2(2,j)-ww2(3,j))*Idx1;
    ww1d2 = (ww1(1,j+1)-ww1(1,j-1))*IdX2;
    ww2d2 = (ww2(1,j+1)-ww2(1,j-1))*IdX2;

    Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - 1;
    Xww2 = ww2d1 + (rr(j)^2)*ww1d2;

    ww1(1,j) = B1w11(j)*Xww1 + B1w12(j)*Xww2;
    ww2(1,j) = B1w21(j)*Xww1 + B1w22(j)*Xww2;

    ww1d1 = -(4*ww1(m-1,j)-ww1(m-2,j))*Idx1;
    ww2d1 = -(4*ww2(m-1,j)-ww2(m-2,j))*Idx1;
    ww1d2 = (ww1(m,j+1)-ww1(m,j-1))*IdX2;
    ww2d2 = (ww2(m,j+1)-ww2(m,j-1))*IdX2;

    Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - 1;
    Xww2 = ww2d1 + (rr(j)^2)*ww1d2;

    ww1(m,j) = Bmw11(j)*Xww1 + Bmw12(j)*Xww2;
    ww2(m,j) = Bmw21(j)*Xww1 + Bmw22(j)*Xww2;

end

%% Stress-free boundary of the foundation
for j = 2:n1-1

    vv1d1 = (4*vv1(2,j)-vv1(3,j))*Idx1;
    vv2d1 = (4*vv2(2,j)-vv2(3,j))*Idx1;
    vv1d2 = (vv1(1,j+1)-vv1(1,j-1))*Idx2;
    vv2d2 = (vv2(1,j+1)-vv2(1,j-1))*Idx2;

    Xvv1 = (L2+2*M2)*vv1d1 + L2*vv2d2;
    Xvv2 = vv2d1 + (r(j)^2)*vv1d2;

    vv1(1,j) = B1v11(j)*Xvv1 + B1v12(j)*Xvv2;
    vv2(1,j) = B1v21(j)*Xvv1 + B1v22(j)*Xvv2;

    vv1d1 = -(4*vv1(m-1,j)-vv1(m-2,j))*Idx1;
    vv2d1 = -(4*vv2(m-1,j)-vv2(m-2,j))*Idx1;
    vv1d2 = (vv1(m,j+1)-vv1(m,j-1))*Idx2;
    vv2d2 = (vv2(m,j+1)-vv2(m,j-1))*Idx2;

    Xvv1 = (L2+2*M2)*vv1d1 + L2*vv2d2;
    Xvv2 = vv2d1 + (r(j)^2)*vv1d2;
```

```
    vv1(m,j) = Bmv11(j)*Xvv1 + Bmv12(j)*Xvv2;
    vv2(m,j) = Bmv21(j)*Xvv1 + Bmv22(j)*Xvv2;

end

%% Governing equations of the overlying body
for i = 2:m-1

    for j = 2:n2-1

        ww1d1 = (ww1(i+1,j)-ww1(i-1,j))*Idx1;
        ww1d2 = (ww1(i,j+1)-ww1(i,j-1))*IdX2;
        ww2d1 = (ww2(i+1,j)-ww2(i-1,j))*Idx1;
        ww2d2 = (ww2(i,j+1)-ww2(i,j-1))*IdX2;

        ww1d11 = (ww1(i+1,j)+ww1(i-1,j))*Idx11;
        ww1d22 = (ww1(i,j+1)+ww1(i,j-1))*IdX22;
        ww2d11 = (ww2(i+1,j)+ww2(i-1,j))*Idx11;
        ww2d22 = (ww2(i,j+1)+ww2(i,j-1))*IdX22;

        ww1d12 = (ww1(i+1,j+1)-ww1(i-1,j+1)-ww1(i+1,j-1)+ww1(i-1,j-1))*IdX12;
        ww2d12 = (ww2(i+1,j+1)-ww2(i-1,j+1)-ww2(i+1,j-1)+ww2(i-1,j-1))*IdX12;

        Xww1 = aa1d11(j)*ww1d11 + aa1d22(j)*ww1d22 + aa1d2(j)*ww1d2 ...
            + aa2d12(j)*ww2d12 + aa2d1(j)*ww2d1;
        Xww2 = bb1d12(j)*ww1d12 + bb1d1(j)*ww1d1 + bb2d11(j)*ww2d11 ...
            + bb2d22(j)*ww2d22 + bb2d2(j)*ww2d2;

        ww1(i,j) = Xww1*Iw1(j);
        ww2(i,j) = Xww2*Iw2(j);

    end

    %% Governing equations of the foundation
    for j = 2:n1-1

        vv1d1 = (vv1(i+1,j)-vv1(i-1,j))*Idx1;
        vv1d2 = (vv1(i,j+1)-vv1(i,j-1))*Idx2;
        vv2d1 = (vv2(i+1,j)-vv2(i-1,j))*Idx1;
        vv2d2 = (vv2(i,j+1)-vv2(i,j-1))*Idx2;

        vv1d11 = (vv1(i+1,j)+vv1(i-1,j))*Idx11;
        vv1d22 = (vv1(i,j+1)+vv1(i,j-1))*Idx22;
        vv2d11 = (vv2(i+1,j)+vv2(i-1,j))*Idx11;
        vv2d22 = (vv2(i,j+1)+vv2(i,j-1))*Idx22;

        vv1d12 = (vv1(i+1,j+1)-vv1(i-1,j+1)-vv1(i+1,j-1)+vv1(i-1,j-1))*Idx12;
        vv2d12 = (vv2(i+1,j+1)-vv2(i-1,j+1)-vv2(i+1,j-1)+vv2(i-1,j-1))*Idx12;

        Xvv1 = a1d11(j)*vv1d11 + a1d22(j)*vv1d22 + a1d2(j)*vv1d2 ...
            + a2d12(j)*vv2d12 + a2d1(j)*vv2d1;
```

```
                Xvv2 = b1d12(j)*vv1d12 + b1d1(j)*vv1d1 + b2d11(j)*vv2d11 ...
                    + b2d22(j)*vv2d22 + b2d2(j)*vv2d2;

                vv1(i,j) = Xvv1*Iv1(j);
                vv2(i,j) = Xvv2*Iv2(j);

            end
        end
    end

    %% Terminating condition
    ppp = norm(vv1,2) + norm(vv2,2);
    p = abs(1-ppp/pp);

end

%% Error
normvv1  = norm(vv1(:,n1),2);
normvv2  = norm(vv2(:,n1),2);

Baldelli1(:) = abs(ww(:)-vv1(:,n1));
Baldelli2(:) = abs(-vv2(:,n1));
Shell1(:) = abs(v1(:,n1)-vv1(:,n1));
Shell2(:) = abs(v2(:,n1)-vv2(:,n1));

w1Norm = norm(Baldelli1,2)/normvv1; % Baldelli azimuthal error
v1Norm = norm(Shell1,2)/normvv1; % Shell azimuthal error

w2Norm = norm(Baldelli2,2)/normvv2; % Baldelli radial error
v2Norm = norm(Shell2,2)/normvv2; % Shell radial error

%% Save data file
filename = 'CH3.mat';
save(filename)
```

3 Shells Supported by Elastic Foundations: Friction Case

In this chapter we present a numerical code for calculating the relative error of our shell model with friction with respect to Kikuchi and Oden's [6] (see chapter 4 of Jayawardana [4]) implemented in Matlab, i.e. `FrictionCode.m`.

We calculate the relative error between the displacement field of the foundation predicted by shell model with friction and extended Kikuchi and Oden's model by

$$\text{Relative Error}(u^i) = \frac{\left(\sum_{\Delta x^2,\Delta x^3} ||u^i_{\text{Shell}}(\Delta x^2,\Delta x^3) - u^i_{\text{Kikuchi}}(\Delta x^2,\Delta x^3)||^2\right)^{\frac{1}{2}}}{\left(\sum_{\Delta x^2,\Delta x^3} ||u^i_{\text{Shell}}(\Delta x^2,\Delta x^3) + u^i_{\text{Kikuchi}}(\Delta x^2,\Delta x^3)||^2\right)^{\frac{1}{2}}} . \quad (35)$$

Now, let `ShellErrorU1`$=$ $\text{Relative Error}(u^2)$ and `ShellErrorU2`$=$ $\text{Relative Error}(u^3)$. Thus, we find:

```
function FrictionCode
%% Shells Supported by Elastic Foundations: Friction Case
% Contact angle is [0,pi]

%% Input
ar = 2; % Radius at \theta = 0.5*pi
br = 2; % Radius at \theta = 0
Th1 = 0.125; % Thickness of the overlying body
Th2 = 1; % Thickness of the foundation
EE1 = 8000; % Young's modulus of the overlying body
EE2 = 1000; % Young's modulus of the foundation
PP1 = 0.25; % Poisson's ratio of the overlying body
PP2 = 0.25; % Poisson's ratio of the foundation
N = 250; % Azimuthal grid points

SS1 = 1; % Applied stress at \theta = 0
SS2 = 1.5; % Applied stress at \theta = pi

%% Output
[U1,U2] = FrictionShell(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N,SS1,SS2);
[V1,V2] = OdenProb2(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N,SS1,SS2);

%% Error
ShellErrorU1 = norm(U1(:,:)-V1(:,:),2)/norm(U1(:,:)+V1(:,:),2); % Azimuthal error % equation (35)
ShellErrorU2 = norm(U2(:,:)-V2(:,:),2)/norm(U2(:,:)+V2(:,:),2); % Radial error % equation (35)

%% Save data file
filename = 'FrictionCode.mat';
save(filename)
```

3.1 Shell Model with Friction

In this section we present a numerical code for an example of our shell model with friction (see section 4.5 of Jayawardana [4]) implemented in Matlab, i.e. `FrictionShell.m`.

To conduct numerical experiments we remain with the framework that we introduced in Chapter 2 (see Section 2.1). Thus, we may express the governing equations of the shell as:

If $[2\nu_F u^3 + \psi_2|u^2|]|_{\omega^{\text{New}}} < 0$, then

$$h\Lambda\partial_2\epsilon^2_2(\boldsymbol{u}) + \frac{1}{3}h^3\Lambda(2F_{[\text{II}]2}^{\ 2}\partial_2\rho^2_2(\boldsymbol{u}) + \partial_2 F_{[\text{II}]2}^{\ 2}\rho^2_2(\boldsymbol{u})) - \text{Tr}(T^3_2(\boldsymbol{u})) = 0 \ , \tag{36}$$

$$-h\Lambda F_{[\text{II}]2}^{\ 2}\epsilon^2_2(\boldsymbol{u}) + \frac{1}{3}h^3\Lambda(\Delta\rho^2_2(\boldsymbol{u}) - F_{[\text{II}]2}^{\ 2}F_{[\text{II}]2}^{\ 2}\rho^2_2(\boldsymbol{u})) + \text{Tr}(T^3_3(\boldsymbol{u})) = 0 \ ; \tag{37}$$

If $[2\nu_F u^3 + \psi_2|u^2|]|_{\omega^{\text{New}}} = 0$, then

$$\begin{aligned}\nu_F h\Lambda\partial_2\epsilon^2_2(\bar{\boldsymbol{u}}) &- \frac{1}{2}h\Lambda\psi_2\text{sign}(u^2)F_{[\text{II}]2}^{\ 2}\epsilon^2_2(\bar{\boldsymbol{u}}) \\ &+ \frac{1}{3}\nu_F h^3\Lambda(2F_{[\text{II}]2}^{\ 2}\partial_2\rho^2_2(\bar{\boldsymbol{u}}) + \partial_2 F_{[\text{II}]2}^{\ 2}\rho^2_2(\bar{\boldsymbol{u}})) \\ &+ \frac{1}{6}h^3\Lambda\psi_2\text{sign}(u^2)(\Delta\rho^2_2(\bar{\boldsymbol{u}}) - F_{[\text{II}]2}^{\ 2}F_{[\text{II}]2}^{\ 2}\rho^2_2(\bar{\boldsymbol{u}})) \\ &- \nu_F\text{Tr}(T^3_2(\bar{\boldsymbol{u}})) + \frac{1}{2}\psi_2\text{sign}(u^2)\text{Tr}(T^3_3(\bar{\boldsymbol{u}})) = 0 \ ,\end{aligned} \tag{38}$$

where $\bar{\boldsymbol{u}}|_{\omega^{\text{New}}} = (0, u^2, -\frac{1}{2}\nu_F^{-1}\psi_2|u^2|)|_{\omega^{\text{New}}}$ and $(0, \partial_3\bar{u}^2, \partial_3\bar{u}^3)|_{\omega^{\text{New}}} = (0, \partial_3 u^2, \partial_3 u^3)|_{\omega^{\text{New}}}$. Note that

$$\text{Tr}(T^3_2(\boldsymbol{u})) = \bar{\mu}\left((\bar{\psi}_2)^2\partial_3 u^2 + \partial_2 u^3\right)|_{\omega^{\text{New}}} \ ,$$

$$\text{Tr}(T^3_3(\boldsymbol{u})) = \left[\bar{\lambda}\left(\partial_2 u^2 + \bar{\Gamma}^2_{22}u^2 + \bar{\Gamma}^2_{23}u^3\right) + (\bar{\lambda} + 2\bar{\mu})\partial_3 u^3\right]|_{\omega^{\text{New}}} \ .$$

Note that to conduct numerical experiments we use the second-order-accurate finite-difference method.

Finally, let `v1`$= u^2$, `v2`$= u^3$, `a`$= b$, `b`$= a$, `Thickness1`$= h$, `Thickness2`$= H$, `Stress1`$= \tau_0$, `Stress2`$= \tau_{\text{max}}$, `Youngs1`$= E$, `Youngs2`$= \bar{E}$, `Poisson1`$= \nu$, `Poisson1`$= \bar{\nu}$, `NN`$= N$ and `Mu`$= \nu_F$. Thus, we find:

```
function [U1,U2] = FrictionShell(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N,SS1,SS2)
format long
%% Shell Model with Friction
% Shell on an elastic prism with a variable elliptical cross section: friction case
% Contact angle is [0,pi]

%% INITIAL PARIMITERS
a = ar; % Radius at \theta = 0.5*pi
b = br; % Radius at \theta = 0

Thickness1 = Th1; % Thickness of the overlying body
Thickness2 = Th2; % Thickness of the foundation

Stress1 = SS1; % Applied stress at \theta = 0
Stress2 = SS2; % Applied stress at \theta = pi
```

```
Youngs1 = EE1; % Young's modulus of the overlying body
Poisson1 = PP1; % Poisson's ratio of the overlying body

Youngs2 = EE2; % Young's modulus of the foundation
Poisson2 = PP2; % Poisson's ratio of the foundation

NN = N; % Azimuthal grid points
error = 10^(-10); % Terminating error

Mu = 0.5; % Coefficient of friction

%% DO NOT CHANGE!
qq = sqrt((a^2+b^2)/2);
q = Thickness2/(qq*pi);

IMu = 1/(2*Mu);

m = NN; % Azimuthal grid points
n = round(q*NN-q+1); % Radial grid points of the foundation

L1  = Poisson1*Youngs1/((1+Poisson1)*(1-2*Poisson1));
M1 = 0.5*Youngs1/(1+Poisson1);
L = 4*M1*(L1+M1)/(L1+2*M1);

L2  = Poisson2*Youngs2/((1+Poisson2)*(1-2*Poisson2));
M2 = 0.5*Youngs2/(1+Poisson2);

NNN = NN^2;
errr = error*NNN;
p = 2;

dx1 = pi/(m-1); % Azimuthal grid spacing
dx2 = Thickness2/(n-1); % Radial grid spacing of the foundation

Idx1 = 1/dx1;
Idx2 = 1/dx2;

Id1 = 0.5*Idx1;
Id11 = Idx1^2;

Id2 = 0.5*Idx2;
Id22 = Idx2^2;

Id12 = Id1*Id2;

u1 = zeros(m,1);
u2 = zeros(m,1);
U1 = zeros(m,n);
U2 = zeros(m,n);
```

```
v1 = zeros(m,n); % Azimuthal displacement of the foundation
v2 = zeros(m,n); % Radial displacement of the foundation

X = zeros(m,1);
IX = zeros(m,1);

K1 = zeros(m,1);
K2 = zeros(m,1);
K11 = zeros(m,1);
K21 = zeros(m,1);

Y = zeros(m,n);
IY = zeros(m,n);

LL111 = zeros(m,n);
LL112 = zeros(m,n);

JF11 = zeros(m,n);
JF12 = zeros(m,n);
JF21 = zeros(m,n);
JF22 = zeros(m,n);

J1B11 = zeros(1,n);
J1B12 = zeros(1,n);
J1B21 = zeros(1,n);
J1B22 = zeros(1,n);

JmB11 = zeros(1,n);
JmB12 = zeros(1,n);
JmB21 = zeros(1,n);
JmB22 = zeros(1,n);

A1d11 = zeros(m,1);
A1d1 = zeros(m,1);
A1d2 = zeros(m,1);

A2d111 = zeros(m,1);
A2d11 = zeros(m,1);
A2d1 = zeros(m,1);

B1d111 = zeros(m,1);
B1d11 = zeros(m,1);
B1d1 = zeros(m,1);

B2d1111 = zeros(m,1);
B2d111 = zeros(m,1);
B2d11 = zeros(m,1);
B2d1 = zeros(m,1);
B2d2 = zeros(m,1);

JM11 = zeros(m,1);
```

```
JM12 = zeros(m,1);
JM21 = zeros(m,1);
JM22 = zeros(m,1);

J1M11 = zeros(m,1);
J1M12 = zeros(m,1);
J1M13 = zeros(m,1);
J1M21 = zeros(m,1);
J1M22 = zeros(m,1);
J1M23 = zeros(m,1);
J1M31 = zeros(m,1);
J1M32 = zeros(m,1);
J1M33 = zeros(m,1);

JmM11 = zeros(m,1);
JmM12 = zeros(m,1);
JmM13 = zeros(m,1);
JmM21 = zeros(m,1);
JmM22 = zeros(m,1);
JmM23 = zeros(m,1);
JmM31 = zeros(m,1);
JmM32 = zeros(m,1);
JmM33 = zeros(m,1);

aa = zeros(2,2);
aaa = zeros(3,3);

C1d22 = zeros(m,n);
C1d1 = zeros(m,n);
C1d2 = zeros(m,n);
C2d1 = zeros(m,n);

D1d12 = zeros(m,n);
D2d11 = zeros(m,n);
D2d22 = zeros(m,n);
D1d1 = zeros(m,n);
D1d2 = zeros(m,n);
D2d1 = zeros(m,n);
D2d2 = zeros(m,n);

SX11 = zeros(m,1);
SX12 = zeros(m,1);
SX21 = zeros(m,1);
SX22 = zeros(m,1);

XX = (Thickness1^2)/3;
IXY = 1/(L*Thickness1);

T1 = Stress1/L;
T2 = Stress2/L;
```

```
a1d1 = 1;
da1d11 = 1;

C1d11 = (L2+2*M2);
C2d12 = (L2+M2);

%% Curvature terms of the shell
for i = 1:m

    x1 = (i-1)*dx1;

    alpha2 = (b*sin(x1))^2 + (a*cos(x1))^2;
    alpha = sqrt(alpha2);
    Ialpha = 1/alpha;
    alphad1 = 0.5*(b^2-a^2)*sin(2*x1)*Ialpha;
    alphad11 = ((b^2-a^2)*cos(2*x1) - alphad1^2)*Ialpha;
    alphad111 = -(2*(b^2-a^2)*cos(2*x1) + 3*alphad1*alphad11)*Ialpha;

    X(i) = alpha;
    IX(i) = 1/X(i);
    IX2 = IX(i)^2;
    Xd1 = alphad1;
    Xd11 = alphad11;
    Xd111 = alphad111;
    Xd2 = a*b*Ialpha^2;
    Xd12 = -2*a*b*alphad1*Ialpha^3;
    Xd112 = -2*a*b*alphad11*Ialpha^3 + 6*a*b*(alphad1^2)*(Ialpha^4);
    Xd1112 = -2*a*b*alphad111*Ialpha^3 + 18*a*b*alphad1*alphad11*(Ialpha^4) ...
        - 24*a*b*(alphad1^3)*(Ialpha^5);

    K1(i) = Xd1*IX(i);
    K2(i) = Xd2*IX(i);
    K11(i) = Xd11*IX(i)-K1(i)^2;
    K21(i) = Xd12*IX(i)-K1(i)*K2(i);

    K111 = Xd111*IX(i) - 3*Xd11*IX(i)*K1(i) + 2*K1(i)^3;
    K211 = Xd112*IX(i) - 2*Xd12*IX(i)*K1(i) - Xd11*IX(i)*K2(i) + 2*K2(i)*K1(i)^2;
    K2111 = Xd1112*IX(i) - 3*Xd112*IX(i)*K1(i) - 3*Xd12*Xd11*IX(i)^2 ...
        + 6*Xd12*IX(i)*K1(i)^2 - Xd111*IX(i)*K2(i) ...
        + 6*Xd11*IX(i)*K1(i)*K2(i) - 6*K2(i)*K1(i)^3;

    a1 = K1(i);
    a2 = K2(i);

    da1d1 = K1(i);
    da1 = K11(i);
    da2d1 = K2(i);
    da2 = K21(i);

    c1d2 = - M2*(X(i)^2)*IXY;
    c2d1 = - M2*IXY;
```

```
c1d1 = L2*IXY;
c1 = L2*K1(i)*IXY;
c2d2 = (L2+2*M2)*IXY;
c2 = L2*K2(i)*IXY;

b1d1 = - 2*K2(i);
b1 = - K21(i) - 2*K1(i)*K2(i);
b2d11 = IX2;
b2d1 = - IX2*K1(i);
b2 = - K2(i)^2;

db1d11 = - 2*K2(i);
db1d1 = - 3*K21(i) - 2*K1(i)*K2(i);
db1 = - K211 - 2*K11(i)*K2(i) - 2*K1(i)*K21(i);
db2d111 =  IX2;
db2d11 = - 3*IX2*K1(i);
db2d1 = - K2(i)^2 - IX2*K11(i) + 2*IX2*K1(i)^2;
db2 = - 2*K21(i)*K2(i);

ddb1d111 = - 2*K2(i);
ddb1d11 = - 5*K21(i) - 2*K1(i)*K2(i);
ddb1d1 = - 4*K211 - 4*K11(i)*K2(i) - 4*K21(i)*K1(i);
ddb1 = - K2111 - 2*K111*K2(i) - 4*K11(i)*K21(i) - 2*K211*K1(i);
ddb2d1111 = IX2;
ddb2d111 = - 5*IX2*K1(i);
ddb2d11 = - K2(i)^2 - 4*IX2*K11(i) + 8*IX2*K1(i)^2;
ddb2d1 = - IX2*K111 + 6*IX2*K11(i)*K1(i) - 4*IX2*K1(i)^3 - 4*K21(i)*K2(i);
ddb2 = - 2*K211*K2(i) - 2*K21(i)^2;

A1d11(i) = da1d11 - 2*K2(i)*XX*db1d11;
A1d1(i) = da1d1 - 2*K2(i)*XX*db1d1 - K21(i)*XX*b1d1;
A1d2(i) = c1d2;
A1 = da1 - 2*K2(i)*XX*db1 - K21(i)*XX*b1 + 3*Id2*c1d2;

A2d111(i) = - 2*K2(i)*XX*db2d111;
A2d11(i) = - 2*K2(i)*XX*db2d11 - K21(i)*XX*b2d11;
A2d1(i) = da2d1 - 2*K2(i)*XX*db2d1 - K21(i)*XX*b2d1 + c2d1;
A2 = da2 - 2*K2(i)*XX*db2 - K21(i)*XX*b2;

B1d111(i) = IX2*XX*ddb1d111;
B1d11(i) = IX2*XX*ddb1d11 - IX2*K1(i)*XX*db1d11;
B1d1(i) = K2(i)*a1d1 + IX2*XX*ddb1d1 - IX2*K1(i)*XX*db1d1 ...
    - (K2(i)^2)*XX*b1d1 + c1d1;
B1 = K2(i)*a1 + IX2*XX*ddb1 - IX2*K1(i)*XX*db1 - (K2(i)^2)*XX*b1 + c1;

B2d1111(i) = IX2*XX*ddb2d1111;
B2d111(i) = IX2*XX*ddb2d111 - IX2*K1(i)*XX*db2d111;
B2d11(i) = IX2*XX*ddb2d11 - IX2*K1(i)*XX*db2d11 - (K2(i)^2)*XX*b2d11;
B2d1(i) = IX2*XX*ddb2d1 - IX2*K1(i)*XX*db2d1 - (K2(i)^2)*XX*b2d1;
B2d2(i) = c2d2;
```

```
B2 = K2(i)*a2 + IX2*XX*ddb2 - IX2*K1(i)*XX*db2 - (K2(i)^2)*XX*b2 + c2 + 3*Id2*c2d2;

aa(1,1) = 2*A1d11(i)*Id11 - A1;
aa(1,2) = 2*A2d11(i)*Id11 - A2;
aa(2,1) = 2*B1d11(i)*Id11 - B1;
aa(2,2) = 2*B2d11(i)*Id11 - B2 - 4*B2d1111(i)*Id11^2;

SX11(i) = aa(1,1);
SX12(i) = aa(1,2);
SX21(i) = aa(2,1);
SX22(i) = aa(2,2);

J = inv(aa);

JM11(i) = J(1,1);
JM12(i) = J(1,2);
JM21(i) = J(2,1);
JM22(i) = J(2,2);

aaa(1,1) = 3*Id1*a1d1 - a1;
aaa(1,2) = - a2;
aaa(1,3) = 0;
aaa(2,1) = 0;
aaa(2,2) = 3*Id1;
aaa(2,3) = 0;
aaa(3,1) = - 2*Id11*db1d11 + 3*Id1*db1d1 - db1;
aaa(3,2) = - 2*Id11*db2d11 - db2;
aaa(3,3) = 3*Id1*db2d111;

J = inv(aaa);

J1M11(i) = J(1,1);
J1M12(i) = J(1,2);
J1M13(i) = J(1,3);
J1M21(i) = J(2,1);
J1M22(i) = J(2,2);
J1M23(i) = J(2,3);
J1M31(i) = J(3,1);
J1M32(i) = J(3,2);
J1M33(i) = J(3,3);

aaa(1,1) = - 3*Id1*a1d1 - a1;
aaa(1,2) = - a2;
aaa(1,3) = 0;
aaa(2,1) = 0;
aaa(2,2) = - 3*Id1;
aaa(2,3) = 0;
aaa(3,1) = - 2*Id11*db1d11 - 3*Id1*db1d1 - db1;
aaa(3,2) = - 2*Id11*db2d11 - db2;
aaa(3,3) = - 3*Id1*db2d111;
```

```
    J = inv(aaa);

    JmM11(i) = J(1,1);
    JmM12(i) = J(1,2);
    JmM13(i) = J(1,3);
    JmM21(i) = J(2,1);
    JmM22(i) = J(2,2);
    JmM23(i) = J(2,3);
    JmM31(i) = J(3,1);
    JmM32(i) = J(3,2);
    JmM33(i) = J(3,3);

end

%% Curvature terms of the foundation
for i = 1:m

    x1 = (i-1)*dx1;

    for j = 1:n

        x2 = (j-1)*dx2 - Thickness2;

        alpha2 = (b*sin(x1))^2 + (a*cos(x1))^2;
        alpha = sqrt(alpha2);
        Ialpha = 1/alpha;
        Ialpha2 = 1/alpha2;
        alphad1 = 0.5*(b^2-a^2)*sin(2*x1)*Ialpha;
        alphad11 = ((b^2-a^2)*cos(2*x1) - alphad1^2)*Ialpha;

        Y(i,j) = alpha + a*b*Ialpha2*x2;
        IY(i,j) = 1/Y(i,j);
        Yd1 = alphad1*(1-2*a*b*x2*Ialpha*Ialpha2);
        Yd11 = alphad11*(1-2*a*b*x2*Ialpha*Ialpha2) + 6*a*b*x2*(alphad1*Ialpha2)^2;
        Yd2 = a*b*Ialpha2;
        Yd12 = -2*a*b*alphad1*Ialpha*Ialpha2;

        LL111(i,j) = Yd1*IY(i,j);
        LL112(i,j) = Yd2*IY(i,j);
        LL1111 = Yd11*IY(i,j)-LL111(i,j)^2;
        LL1112 = Yd12*IY(i,j)-LL111(i,j)*LL112(i,j);
        LL1122 = -LL112(i,j)^2;

        C1d22(i,j) = M2*Y(i,j)^2;
        C1d1(i,j) = (L2+2*M2)*LL111(i,j);
        C1d2(i,j) = 3*M2*LL112(i,j)*Y(i,j)^2;
        C2d1(i,j) = (L2+3*M2)*LL112(i,j);
        C1 = (L2+2*M2)*LL1111;
        C2 = (L2+2*M2)*LL1112;

        D1d12(i,j) = (L2+M2)*Y(i,j);
```

```
        D2d11(i,j) = M2*IY(i,j);
        D2d22(i,j) = (L2+2*M2)*Y(i,j);
        D1d1(i,j) = - 2*M2*LL112(i,j)*Y(i,j);
        D1d2(i,j) = (L2+M2)*LL111(i,j)*Y(i,j);
        D2d1(i,j) = - M2*LL111(i,j)*IY(i,j);
        D2d2(i,j) = (L2+2*M2)*LL112(i,j)*Y(i,j);
        D1 = L2*LL1112*Y(i,j) - 2*M2*LL111(i,j)*LL112(i,j)*Y(i,j);
        D2 = L2*LL1122*Y(i,j) - 2*M2*LL112(i,j)*LL112(i,j)*Y(i,j);

        aa(1,1) = 2*C1d11*Id11      + 2*C1d22(i,j)*Id22 - C1;
        aa(1,2) = - C2;
        aa(2,1) = - D1;
        aa(2,2) = 2*D2d11(i,j)*Id11 + 2*D2d22(i,j)*Id22 - D2;

        J = inv(aa);

        JF11(i,j) = J(1,1);
        JF12(i,j) = J(1,2);
        JF21(i,j) = J(2,1);
        JF22(i,j) = J(2,2);

    end
end

for j = 1:n

    aa(1,1) = 3*Id1*(L2+2*M2) - (L2+2*M2)*LL111(1,j);
    aa(1,2) = - (L2+2*M2)*LL112(1,j);
    aa(2,1) = 0;
    aa(2,2) = 3*Id1;

    J = inv(aa);

    J1B11(j) = J(1,1);
    J1B12(j) = J(1,2);
    J1B21(j) = J(2,1);
    J1B22(j) = J(2,2);

    aa(1,1) = - 3*Id1*(L2+2*M2) - (L2+2*M2)*LL111(m,j);
    aa(1,2) = - (L2+2*M2)*LL112(m,j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Id1;

    J = inv(aa);

    JmB11(j) = J(1,1);
    JmB12(j) = J(1,2);
    JmB21(j) = J(2,1);
    JmB22(j) = J(2,2);

end
```

```
%% Main code
while errr < p

    pp = norm(v1,2) + norm(v2,2);

    for k = 1:NNN

        %% Corners
        u1(1) = (2*v1(1,n)-5*v1(2,n)+4*v1(3,n)-v1(4,n))*Id11;

        v1d1 = (4*v1(2,n)-v1(3,n))*Id1;
        v2d1 = (4*v2(2,n)-v2(3,n))*Id1;
        u2d1 = (4*u2(2)-u2(3))*Id1;
        v1d11 = (-5*v1(2,n)+4*v1(3,n)-v1(4,n))*Id11;
        v2d11 = (-5*v2(2,n)+4*v2(3,n)-v2(4,n))*Id11;

        Xv1 = v1d1 - T1;
        Xv2 = v2d1;
        Xu2 = - 2*K2(1)*v1d11 - (3*K21(1)+2*K1(1)*K2(1))*v1d1 ...
            + (IX(1)^2)*u2d1 - 3*(IX(1)^2)*K1(1)*v2d11;

        v1(1,n) = J1M11(1)*Xv1 + J1M12(1)*Xv2 + J1M13(1)*Xu2;
        v2(1,n) = J1M21(1)*Xv1 + J1M22(1)*Xv2 + J1M23(1)*Xu2;
        u2(1)   = J1M31(1)*Xv1 + J1M32(1)*Xv2 + J1M33(1)*Xu2;

        u1(m) = (2*v1(m,n)-5*v1(m-1,n)+4*v1(m-2,n)-v1(m-3,n))*Id11;

        v1d1 = -(4*v1(m-1,n)-v1(m-2,n))*Id1;
        v2d1 = -(4*v2(m-1,n)-v2(m-2,n))*Id1;
        u2d1 = -(4*u2(m-1)-u2(m-2))*Id1;
        v1d11 = (-5*v1(m-1,n)+4*v1(m-2,n)-v1(m-3,n))*Id11;
        v2d11 = (-5*v2(m-1,n)+4*v2(m-2,n)-v2(m-3,n))*Id11;

        Xv1 = v1d1 - T2;
        Xv2 = v2d1;
        Xu2 = - 2*K2(m)*v1d11 - (3*K21(m)+2*K1(m)*K2(m))*v1d1 ...
            + (IX(m)^2)*u2d1 - 3*(IX(m)^2)*K1(m)*v2d11;

        v1(m,n) = JmM11(m)*Xv1 + JmM12(m)*Xv2 + JmM13(m)*Xu2;
        v2(m,n) = JmM21(m)*Xv1 + JmM22(m)*Xv2 + JmM23(m)*Xu2;
        u2(m)   = JmM31(m)*Xv1 + JmM32(m)*Xv2 + JmM33(m)*Xu2;

        %% Gouerning equations of the shell
        for i = 2:m-1

            u1(i) = (v1(i+1,n)-2*v1(i,n)+v1(i-1,n))*Id11;
            u2(i) = (v2(i+1,n)-2*v2(i,n)+v2(i-1,n))*Id11;

            v1d11 = (v1(i+1,n)+v1(i-1,n))*Id11;
            v2d11 = (v2(i+1,n)+v2(i-1,n))*Id11;
```

```
    v1d1 = (v1(i+1,n)-v1(i-1,n))*Id1;
    v2d1 = (v2(i+1,n)-v2(i-1,n))*Id1;

    u2d11 = (u2(i+1)+u2(i-1))*Id11;

    u1d1 = (u1(i+1)-u1(i-1))*Id1;
    u2d1 = (u2(i+1)-u2(i-1))*Id1;

    v1d2 = -(4*v1(i,n-1)-v1(i,n-2))*Id2;
    v2d2 = -(4*v2(i,n-1)-v2(i,n-2))*Id2;

    %% Bounded boundary
    Xv1 = A1d11(i)*v1d11 + A1d1(i)*v1d1 + A1d2(i)*v1d2 + A2d111(i)*u2d1 ...
        + A2d11(i)*v2d11 + A2d1(i)*v2d1; % equation (36)
    Xv2 = B1d111(i)*u1d1 + B1d11(i)*v1d11 + B1d1(i)*v1d1 + B2d1111(i)*u2d11 ...
        + B2d111(i)*u2d1 + (B2d11(i)-2*B2d1111(i)*Id11)*v2d11 ...
        + B2d1(i)*v2d1 + B2d2(i)*v2d2; % equation (37)

    v1(i,n) = JM11(i)*Xv1 + JM12(i)*Xv2;
    v2(i,n) = JM21(i)*Xv1 + JM22(i)*Xv2;

    delta = 2*Mu*v2(i,n) + X(i)*abs(v1(i,n));

    %% Limiting-equilibrium boundary
    if ~(delta < 0)

        v2(i,n) = - IMu*X(i)*abs(v1(i,n));

        XXv1  = Xv1+IMu*X(i)*sign(v1(i,n))*Xv2; % equation (38)
        IXv1 = 1/(SX11(i) + IMu*X(i)*sign(v1(i,n))*(-SX12(i)+SX21(i)) ...
            - SX22(i)*(X(i)*IMu)^2);

        v1(i,n) = XXv1*IXv1;

    end
end

%% Stress-free Boundary of the foundation
for j = 2:n-1

    v1d1 = (4*v1(2,j)-v1(3,j))*Id1;
    v1d2 = (v1(1,j+1)-v1(1,j-1))*Id2;
    v2d1 = (4*v2(2,j)-v2(3,j))*Id1;
    v2d2 = (v2(1,j+1)-v2(1,j-1))*Id2;

    Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
    Xv2 = v2d1 + (Y(1,j)^2)*v1d2;

    v1(1,j) = J1B11(j)*Xv1 + J1B12(j)*Xv2;
    v2(1,j) = J1B21(j)*Xv1 + J1B22(j)*Xv2;
```

```
            v1d1 = -(4*v1(m-1,j)-v1(m-2,j))*Id1;
            v1d2 = (v1(m,j+1)-v1(m,j-1))*Id2;
            v2d1 = -(4*v2(m-1,j)-v2(m-2,j))*Id1;
            v2d2 = (v2(m,j+1)-v2(m,j-1))*Id2;

            Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
            Xv2 = v2d1 + (Y(m,j)^2)*v1d2;

            v1(m,j) = JmB11(j)*Xv1 + JmB12(j)*Xv2;
            v2(m,j) = JmB21(j)*Xv1 + JmB22(j)*Xv2;

        end

        %% Gouerning equations of the foundation
        for i = 2:m-1

            for j = 2:n-1

                v1d11 = (v1(i+1,j)+v1(i-1,j))*Id11;
                v1d22 = (v1(i,j+1)+v1(i,j-1))*Id22;
                v1d12 = (v1(i+1,j+1)-v1(i+1,j-1)-v1(i-1,j+1)+v1(i-1,j-1))*Id12;
                v1d1 = (v1(i+1,j)-v1(i-1,j))*Id1;
                v1d2 = (v1(i,j+1)-v1(i,j-1))*Id2;

                v2d11 = (v2(i+1,j)+v2(i-1,j))*Id11;
                v2d22 = (v2(i,j+1)+v2(i,j-1))*Id22;
                v2d12 = (v2(i+1,j+1)-v2(i+1,j-1)-v2(i-1,j+1)+v2(i-1,j-1))*Id12;
                v2d1 = (v2(i+1,j)-v2(i-1,j))*Id1;
                v2d2 = (v2(i,j+1)-v2(i,j-1))*Id2;

                Xv1 = C1d11*v1d11      + C1d22(i,j)*v1d22 + C2d12*v2d12      ...
                    + C1d1(i,j)*v1d1 + C1d2(i,j)*v1d2 + C2d1(i,j)*v2d1;
                Xv2 = D2d11(i,j)*v2d11 + D2d22(i,j)*v2d22 + D1d12(i,j)*v1d12 ...
                    + D1d1(i,j)*v1d1 + D1d2(i,j)*v1d2 + D2d1(i,j)*v2d1 + D2d2(i,j)*v2d2;

                v1(i,j) = JF11(i,j)*Xv1 + JF12(i,j)*Xv2;
                v2(i,j) = JF21(i,j)*Xv1 + JF22(i,j)*Xv2;

            end
        end
    end

    %% Terminating condition
    ppp = norm(v1,2) + norm(v2);
    p = abs(1- ppp/pp);

end

%% Foundation
U1(:,:) = v1(:,:);
```

```
U2(:,:) = v2(:,:);
```

3.2 Two-Body Kikuchi and Oden's Model

In this section we present a numerical code for an example of Kikuchi and Oden's [6] model for two-body friction problems (see section 4.6 of Jayawardana [4]) implemented in Matlab, i.e. `OdenProb2.m`. Note that to find numerical solutions we employ the Newton's method for nonlinear systems (see chapter 10 of Burden *et al.* [3]).

To conduct numerical experiments consider the following. In accordance with the framework that is introduced in section 2.1, the overlying body is restricted to the region $x^3 \in (0, h)$. Thus, with some calculations, one finds that the perturbed governing equations of the overlying body are

$$(\lambda + \mu)\partial^2 \left(\bar{\nabla}_i \delta v^i\right) + \mu \bar{\Delta} \delta v^2 = 0 \ , \tag{39}$$

$$(\lambda + \mu)\partial^3 \left(\bar{\nabla}_i \delta v^i\right) + \mu \bar{\Delta} \delta v^3 = 0 \ , \tag{40}$$

where

$$\begin{aligned}
\bar{\nabla}_2 \delta v^2 &= \partial_2 \delta v^2 + \bar{\Gamma}^2_{22} \delta v^2 + \bar{\Gamma}^2_{23} \delta v^3 \ , \\
\bar{\nabla}_2 \delta v^3 &= \partial_2 \delta v^3 - (\bar{\psi}_2)^2 \bar{\Gamma}^2_{23} \delta v^2 \ , \\
\bar{\nabla}_3 \delta v^2 &= \partial_3 \delta v^2 + \bar{\Gamma}^2_{23} \delta v^2 \ , \\
\bar{\nabla}_3 \delta v^3 &= \partial_3 \delta v^3 \ ,
\end{aligned}$$

and $\delta \boldsymbol{v} = (0, \delta v^2(x^2, x^3), \delta v^3(x^2, x^3))$ is a small perturbation of the displacement field of the overlying body. With relative ease, one finds that the perturbed governing equations of the foundation are

$$(\bar{\lambda} + \bar{\mu})\partial^2 \left(\bar{\nabla}_i \delta u^i\right) + \bar{\mu} \bar{\Delta} \delta u^2 = 0 \ , \tag{41}$$

$$(\bar{\lambda} + \bar{\mu})\partial^3 \left(\bar{\nabla}_i \delta u^i\right) + \bar{\mu} \bar{\Delta} \delta u^3 = 0 \ , \tag{42}$$

where $\delta \boldsymbol{u} = (0, \delta u^2(x^2, x^3), \delta u^3(x^2, x^3))$ is the perturbation of the displacement field of the foundation. With some more calculations, one finds the following boundary conditions to the displacement fields,

$$\left[v^3 - u^3\right]|_{\omega^{\text{New}}} = 0 \text{ (continuous radial displacement)} \ , \tag{43}$$

$$\left[T^3_3(\boldsymbol{v}) - T^3_3(\boldsymbol{u})\right]|_{\omega^{\text{New}}} = 0 \text{ (continuous radial stress)} \ , \tag{44}$$

and the boundary conditions for the perturbations

$$\begin{aligned}
\delta u^2|_{\partial\Omega_0^{\text{New}} \cup \overline{\partial\Omega}_f^{\text{New}}} &= 0 \ , \\
\delta u^3|_{\overline{\partial\Omega}^{\text{New}}} &= 0 \ , \\
\delta v^2|_{\{\partial\omega^{\text{New}} \times (0,h)\} \cup \{[-\frac{1}{2}\pi, \frac{1}{2}\pi] \times \{h\}\}} &= 0 \ , \\
\delta v^3|_{\bar{\omega}^{\text{New}} \cup \{\partial\omega^{\text{New}} \times (0,h)\} \cup \{[-\frac{1}{2}\pi, \frac{1}{2}\pi] \times \{h\}\}} &= 0 \ .
\end{aligned}$$

Thus, the equations characterising the frictionally coupling of the overlying body to the foundation can be expressed as:

If $\bar{\psi}_2|v^2 - u^2||_{\omega^{\text{New}}} \geq \epsilon$, then

$$\left[\mu\left(\bar{\psi}_2\partial_3 v^2 + (\bar{\psi}_2)^{-1}\partial_2 v^3\right) + \nu_F \text{sign}(v^2 - u^2)T_3^3(\boldsymbol{v})\right]|_{\omega^{\text{New}}} = 0\ , \tag{45}$$

$$\left[\bar{\mu}\left(\bar{\psi}_2\partial_3 u^2 + (\bar{\psi}_2)^{-1}\partial_2 u^3\right) + \nu_F \text{sign}(v^2 - u^2)T_3^3(\boldsymbol{u})\right]|_{\omega^{\text{New}}} = 0\ ; \tag{46}$$

If $\bar{\psi}_2|v^2 - u^2||_{\omega^{\text{New}}} < \epsilon$, then

$$\begin{aligned}&\left[\mu\left(\bar{\psi}_2\partial_3 \delta v^2\right) + \nu_F\epsilon^{-1}\bar{\psi}_2(v^2 - u^2)T_3^3(\delta\boldsymbol{v}) + \nu_F\epsilon^{-1}\bar{\psi}_2(\delta v^2 - \delta u^2)T_3^3(\boldsymbol{v})\right.\\ &\quad\left.+\,\mu\left(\bar{\psi}_2\partial_3 v^2 + (\bar{\psi}_2)^{-1}\partial_2 v^3\right) + \nu_F\epsilon^{-1}\bar{\psi}_2(v^2 - u^2)T_3^3(\boldsymbol{v})\right]|_{\omega^{\text{New}}} = 0\ ,\end{aligned} \tag{47}$$

$$\begin{aligned}&\left[\bar{\mu}\left(\bar{\psi}_2\partial_3 \delta u^2\right) + \nu_F\epsilon^{-1}\bar{\psi}_2(v^2 - u^2)T_3^3(\delta\boldsymbol{u}) + \nu_F\epsilon^{-1}\bar{\psi}_2(\delta v^2 - \delta u^2)T_3^3(\boldsymbol{u})\right.\\ &\quad\left.+\,\bar{\mu}\left(\bar{\psi}_2\partial_3 u^2 + (\bar{\psi}_2)^{-1}\partial_2 u^3\right) + \nu_F\epsilon^{-1}\bar{\psi}_2(v^2 - u^2)T_3^3(\boldsymbol{u})\right]|_{\omega^{\text{New}}} = 0\ ,\end{aligned} \tag{48}$$

where

$$T_3^3(\boldsymbol{v}) = \lambda\left(\partial_2 v^2 + \bar{\Gamma}_{22}^2 v^2 + \bar{\Gamma}_{23}^2 v^3\right) + (\lambda + 2\mu)\partial_3 v^3\ ,$$

$$T_3^3(\boldsymbol{u}) = \bar{\lambda}\left(\partial_2 u^2 + \bar{\Gamma}_{22}^2 u^2 + \bar{\Gamma}_{23}^2 u^3\right) + (\bar{\lambda} + 2\bar{\mu})\partial_3 u^3\ .$$

We choose to terminate our iterating process once the condition $|1-(||\boldsymbol{u}_m||_{\ell^2}+||\boldsymbol{v}_m||_{\ell^2}+||\delta\boldsymbol{u}_m||_{\ell^2}+||\delta\boldsymbol{v}_m||_{\ell^2})^{-1}(||\boldsymbol{u}_{m+1}||_{\ell^2}+||\boldsymbol{v}_{m+1}||_{\ell^2}+||\delta\boldsymbol{u}_{m+1}||_{\ell^2}+||\delta\boldsymbol{v}_{m+1}||_{\ell^2})| < 10^{-10}$ is satisfied, where $\boldsymbol{u}_m$, $\boldsymbol{v}_m$, $\delta\boldsymbol{u}_m$ and $\delta\boldsymbol{v}_m$ are the m^{th} iterative solutions of extended Kikuchi and Oden's model model.

Finally, let `v1`$= u^2$, `v2`$= u^3$, `dv1`$= \delta u^2$, `dv2`$= \delta u^3$, `u1`$= v^2$, `u2`$= v^3$, `du1`$= \delta v^2$, `du2`$= \delta v^3$, `a`$= b$, `b`$= a$, `Thickness1`$= h$, `Thickness2`$= H$, `Stress1`$= \tau_0$, `Stress2`$= \tau_{\text{max}}$, `Youngs1`$= E$, `Youngs2`$= \bar{E}$, `Poisson1`$= \nu$, `Poisson1`$= \bar{\nu}$, `NN`$= N$ and `Mu`$= \nu_F$. Thus, we find:

```
function [V1,V2] = OdenProb2(ar,br,Th1,Th2,EE1,EE2,PP1,PP2,N,SS1,SS2)
format long
%% Two-Body Kikuchi and Oden's Model
% Overlying elastic body on an elastic prism with a variable elliptical cross section
% Contact angle is [0,pi]
% Static friction case

%% INITIAL PARIMITERS
a = ar; % Radius at \theta = 0.5*pi
b = br; % Radius at \theta = 0

Thickness1 = Th1; % Thickness of the overlying body
Thickness2 = Th2; % Thickness of the foundation

Stress1 = SS1; % Applied stress at \theta = 0
Stress2 = SS2; % Applied stress at \theta = pi

Youngs1 = EE1; % Young's modulus of the overlying body
Poisson1 = PP1; % Poisson's ratio of the overlying body

Youngs2 = EE2; % Young's modulus of the foundation
Poisson2 = PP2; % Poisson's ratio of the foundation
```

```
NN = N; % Azimuthal grid points
error =  10^(-10); % Terminating error

Mu = 0.5; % Coefficient of friction
epsi = 10^(-10); % Regularisation parameter

%% DO NOT CHANGE!
qq1 = sqrt((a^2+b^2)/2) + 2*a*b*Thickness1/(a^2+b^2);
qq2 = sqrt((a^2+b^2)/2);

q1 = Thickness1/(qq1*pi);
q2 = Thickness2/(qq2*pi);

m = NN; % Azimuthal grid points
n1 = round(q1*NN-q1+1); % Radial grid points of the overlying body
n2 = round(q2*NN-q2+1); % Radial grid points of the foundation

L1  = Poisson1*Youngs1/((1+Poisson1)*(1-2*Poisson1));
M1 = 0.5*Youngs1/(1+Poisson1);

L2  = Poisson2*Youngs2/((1+Poisson2)*(1-2*Poisson2));
M2 = 0.5*Youngs2/(1+Poisson2);

NNN = NN^2;
errr = error*NNN;
p = 2;

dx1 = pi/(m-1); % Azimuthal grid spacing
dx2 = Thickness1/(n1-1); % Radial grid spacing of the overlying body
dX2 = Thickness2/(n2-1); % Radial grid spacing of the foundation

Idx1 = 1/dx1;
Idx2 = 1/dx2;
IdX2 = 1/dX2;

Iepsi = 1/epsi;

Id1 = 0.5*Idx1;
Id11 = Idx1^2;

Id2 = 0.5*Idx2;
Id22 = Idx2^2;

Id12 = Id1*Id2;

ID2 = 0.5*IdX2;
ID22 = IdX2^2;

ID12 = Id1*ID2;

aa = zeros(2,2);
```

```
u1 = zeros(m,n1); % Azimuthal displacement of the overlying body
u2 = zeros(m,n1); % Radial displacement of the overlying body
du1 = zeros(m,n1); % Perturbed azimuthal displacement of the overlying body
du2 = zeros(m,n1); % Perturbed radial displacement of the overlying body

v1 = zeros(m,n2); % Azimuthal displacement of the foundation
v2 = zeros(m,n2); % Radial displacement of the foundation
dv1 = zeros(m,n2); % Perturbed azimuthal displacement of the foundation
dv2 = zeros(m,n2); % Perturbed radial displacement of the foundation

V1 = zeros(m,n2);
V2 = zeros(m,n2);

X = zeros(m,n1);
IX = zeros(m,n1);

L111 = zeros(m,n1);
L112 = zeros(m,n1);

Y = zeros(m,n2);
IY = zeros(m,n2);

LL111 = zeros(m,n2);
LL112 = zeros(m,n2);

JM11 = zeros(m,n1);
JM12 = zeros(m,n1);
JM21 = zeros(m,n1);
JM22 = zeros(m,n1);

JF11 = zeros(m,n2);
JF12 = zeros(m,n2);
JF21 = zeros(m,n2);
JF22 = zeros(m,n2);

J1S11 = zeros(1,n1);
J1S12 = zeros(1,n1);
J1S21 = zeros(1,n1);
J1S22 = zeros(1,n1);

JmS11 = zeros(1,n1);
JmS12 = zeros(1,n1);
JmS21 = zeros(1,n1);
JmS22 = zeros(1,n1);

J1B11 = zeros(1,n2);
J1B12 = zeros(1,n2);
J1B21 = zeros(1,n2);
J1B22 = zeros(1,n2);
```

```
JmB11 = zeros(1,n2);
JmB12 = zeros(1,n2);
JmB21 = zeros(1,n2);
JmB22 = zeros(1,n2);

JnB11 = zeros(m,1);
JnB12 = zeros(m,1);
JnB21 = zeros(m,1);
JnB22 = zeros(m,1);

JBB11 = zeros(m,1);
JBB12 = zeros(m,1);
JBB21 = zeros(m,1);
JBB22 = zeros(m,1);

JBBB11 = zeros(m,1);
JBBB22 = zeros(m,1);

A1d22 = zeros(m,n1);
A1d1 = zeros(m,n1);
A1d2 = zeros(m,n1);
A2d1 = zeros(m,n1);

B1d12 = zeros(m,n1);
B2d11 = zeros(m,n1);
B2d22 = zeros(m,n1);
B1d1 = zeros(m,n1);
B1d2 = zeros(m,n1);
B2d1 = zeros(m,n1);
B2d2 = zeros(m,n1);

C1d22 = zeros(m,n2);
C1d1 = zeros(m,n2);
C1d2 = zeros(m,n2);
C2d1 = zeros(m,n2);

D1d12 = zeros(m,n2);
D2d11 = zeros(m,n2);
D2d22 = zeros(m,n2);
D1d1 = zeros(m,n2);
D1d2 = zeros(m,n2);
D2d1 = zeros(m,n2);
D2d2 = zeros(m,n2);

T1 = Stress1;
T2 = Stress2;

A1d11 = (L1+2*M1);
A2d12 = (L1+M1);

C1d11 = (L2+2*M2);
```

```
C2d12 = (L2+M2);

%% Curvature terms of the overlying body
for i = 1:m

    x1 = (i-1)*dx1;

    for j = 1:n1

        x2 = (j-1)*dx2;

        alpha2 = (b*sin(x1))^2 + (a*cos(x1))^2;
        alpha = sqrt(alpha2);
        Ialpha = 1/alpha;
        Ialpha2 = 1/alpha2;
        alphad1 = 0.5*(b^2-a^2)*sin(2*x1)*Ialpha;
        alphad11 = ((b^2-a^2)*cos(2*x1) - alphad1^2)*Ialpha;

        X(i,j) = alpha + a*b*Ialpha2*x2;
        IX(i,j) = 1/X(i,j);
        Xd1 = alphad1*(1-2*a*b*x2*Ialpha*Ialpha2);
        Xd11 = alphad11*(1-2*a*b*x2*Ialpha*Ialpha2) + 6*a*b*x2*(alphad1*Ialpha2)^2;
        Xd2 = a*b*Ialpha2;
        Xd12 = -2*a*b*alphad1*Ialpha*Ialpha2;

        L111(i,j) = Xd1*IX(i,j);
        L112(i,j) = Xd2*IX(i,j);
        L1111 = Xd11*IX(i,j)-L111(i,j)^2;
        L1112 = Xd12*IX(i,j)-L111(i,j)*L112(i,j);
        L1122 = -L112(i,j)^2;

        A1d22(i,j) = M1*X(i,j)^2;
        A1d1(i,j) = (L1+2*M1)*L111(i,j);
        A1d2(i,j) = 3*M1*L112(i,j)*X(i,j)^2;
        A2d1(i,j) = (L1+3*M1)*L112(i,j);
        A1 = (L1+2*M1)*L1111;
        A2 = (L1+2*M1)*L1112;

        B1d12(i,j) = (L1+M1)*X(i,j);
        B2d11(i,j) = M1*IX(i,j);
        B2d22(i,j) = (L1+2*M1)*X(i,j);
        B1d1(i,j) = - 2*M1*L112(i,j)*X(i,j);
        B1d2(i,j) = (L1+M1)*L111(i,j)*X(i,j);
        B2d1(i,j) = - M1*L111(i,j)*IX(i,j);
        B2d2(i,j) = (L1+2*M1)*L112(i,j)*X(i,j);
        B1 = L1*L1112*X(i,j) - 2*M1*L111(i,j)*L112(i,j)*X(i,j);
        B2 = L1*L1122*X(i,j) - 2*M1*L112(i,j)*L112(i,j)*X(i,j);

        aa(1,1) = 2*A1d11*Id11      + 2*A1d22(i,j)*Id22 - A1;
        aa(1,2) = - A2;
        aa(2,1) = - B1;
```

```
        aa(2,2) = 2*B2d11(i,j)*Id11 + 2*B2d22(i,j)*Id22 - B2;

        J = inv(aa);

        JM11(i,j) = J(1,1);
        JM12(i,j) = J(1,2);
        JM21(i,j) = J(2,1);
        JM22(i,j) = J(2,2);

    end
end

for i = 1:m

    JBBB11(i) = 1/(3*X(i,1)*M1*Id2 + 3*X(i,1)*M2*ID2);
    JBBB22(i) = 1/(- L1*L112(i,1) + L2*L112(i,1) + 3*Id2*(L1+2*M1) + 3*ID2*(L2+2*M2));

    aa(1,1) = -3*Id2;
    aa(1,2) = 0;
    aa(2,1) = - L1*L111(i,n1);
    aa(2,2) = - L1*L112(i,n1) - 3*Id2*(L1+2*M1);

    J = inv(aa);

    JBB11(i) = J(1,1);
    JBB12(i) = J(1,2);
    JBB21(i) = J(2,1);
    JBB22(i) = J(2,2);

end

for j = 1:n1

    aa(1,1) = 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(1,j);
    aa(1,2) = - (L1+2*M1)*L112(1,j);
    aa(2,1) = 0;
    aa(2,2) = 3*Id1;

    J = inv(aa);

    J1S11(j) = J(1,1);
    J1S12(j) = J(1,2);
    J1S21(j) = J(2,1);
    J1S22(j) = J(2,2);

    aa(1,1) =  - 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(m,j);
    aa(1,2) = - (L1+2*M1)*L112(m,j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Id1;

    J = inv(aa);
```

```
    JmS11(j) = J(1,1);
    JmS12(j) = J(1,2);
    JmS21(j) = J(2,1);
    JmS22(j) = J(2,2);

end

%% Curvature terms of the foundation
for i = 1:m

    x1 = (i-1)*dx1;

    for j = 1:n2

        x2 = (j-1)*dX2 - Thickness2;

        alpha2 = (b*sin(x1))^2 + (a*cos(x1))^2;
        alpha = sqrt(alpha2);
        Ialpha = 1/alpha;
        Ialpha2 = 1/alpha2;
        alphad1 = 0.5*(b^2-a^2)*sin(2*x1)*Ialpha;
        alphad11 = ((b^2-a^2)*cos(2*x1) - alphad1^2)*Ialpha;

        Y(i,j) = alpha + a*b*Ialpha2*x2;
        IY(i,j) = 1/Y(i,j);
        Yd1 = alphad1*(1-2*a*b*x2*Ialpha*Ialpha2);
        Yd11 = alphad11*(1-2*a*b*x2*Ialpha*Ialpha2) + 6*a*b*x2*(alphad1*Ialpha2)^2;
        Yd2 = a*b*Ialpha2;
        Yd12 = -2*a*b*alphad1*Ialpha*Ialpha2;

        LL111(i,j) = Yd1*IY(i,j);
        LL112(i,j) = Yd2*IY(i,j);
        LL1111 = Yd11*IY(i,j)-LL111(i,j)^2;
        LL1112 = Yd12*IY(i,j)-LL111(i,j)*LL112(i,j);
        LL1122 = -LL112(i,j)^2;

        C1d22(i,j) = M2*Y(i,j)^2;
        C1d1(i,j) = (L2+2*M2)*LL111(i,j);
        C1d2(i,j) = 3*M2*LL112(i,j)*Y(i,j)^2;
        C2d1(i,j) = (L2+3*M2)*LL112(i,j);
        C1 = (L2+2*M2)*LL1111;
        C2 = (L2+2*M2)*LL1112;

        D1d12(i,j) = (L2+M2)*Y(i,j);
        D2d11(i,j) = M2*IY(i,j);
        D2d22(i,j) = (L2+2*M2)*Y(i,j);
        D1d1(i,j) = - 2*M2*LL112(i,j)*Y(i,j);
        D1d2(i,j) = (L2+M2)*LL111(i,j)*Y(i,j);
        D2d1(i,j) = - M2*LL111(i,j)*IY(i,j);
        D2d2(i,j) = (L2+2*M2)*LL112(i,j)*Y(i,j);
```

```
        D1 = L2*LL1112*Y(i,j) - 2*M2*LL111(i,j)*LL112(i,j)*Y(i,j);
        D2 = L2*LL1122*Y(i,j) - 2*M2*LL112(i,j)*LL112(i,j)*Y(i,j);

        aa(1,1) = 2*C1d11*Id11      + 2*C1d22(i,j)*ID22 - C1;
        aa(1,2) = - C2;
        aa(2,1) = - D1;
        aa(2,2) = 2*D2d11(i,j)*Id11 + 2*D2d22(i,j)*ID22 - D2;

        J = inv(aa);

        JF11(i,j) = J(1,1);
        JF12(i,j) = J(1,2);
        JF21(i,j) = J(2,1);
        JF22(i,j) = J(2,2);

    end
end

for j = 1:n2

    aa(1,1) = 3*Id1*(L2+2*M2) - (L2+2*M2)*LL111(1,j);
    aa(1,2) = - (L2+2*M2)*LL112(1,j);
    aa(2,1) = 0;
    aa(2,2) = 3*Id1;

    J = inv(aa);

    J1B11(j) = J(1,1);
    J1B12(j) = J(1,2);
    J1B21(j) = J(2,1);
    J1B22(j) = J(2,2);

    aa(1,1) = - 3*Id1*(L2+2*M2) - (L2+2*M2)*LL111(m,j);
    aa(1,2) = - (L2+2*M2)*LL112(m,j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Id1;

    J = inv(aa);

    JmB11(j) = J(1,1);
    JmB12(j) = J(1,2);
    JmB21(j) = J(2,1);
    JmB22(j) = J(2,2);

end

for i = 1:m

    aa(1,1) = -3*Id2;
    aa(1,2) = 0;
    aa(2,1) = - L2*LL111(i,n2);
```

```
    aa(2,2) = - L2*LL112(i,n2) - 3*(L2+2*M2)*ID2;

    J = inv(aa);

    JnB11(i) = J(1,1);
    JnB12(i) = J(1,2);
    JnB21(i) = J(2,1);
    JnB22(i) = J(2,2);

end

aaa(1,1) = 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(1,1);
aaa(1,2) = 3*Id2*L1 - (L1+2*M1)*L112(1,1);
aaa(1,3) = 0;
aaa(2,1) = 3*Id1*L1 - L1*L111(1,1);
aaa(2,2) = 3*Id2*(L1+2*M1) + 3*ID2*(L2+2*M2) - (L1+2*M1)*L112(1,1) + (L2+2*M2)*L112(1,1);
aaa(2,3) = - 3*Id1*L2 + L2*L111(1,1);
aaa(3,1) = 0;
aaa(3,2) = - 3*ID2*L2 - (L2+2*M2)*L112(1,1);
aaa(3,3) = 3*Id1*(L2+2*M2) - (L2+2*M2)*L111(1,1);

J = inv(aaa);

J11BBB11 = J(1,1);
J11BBB12 = J(1,2);
J11BBB13 = J(1,3);
J11BBB21 = J(2,1);
J11BBB22 = J(2,2);
J11BBB23 = J(2,3);
J11BBB31 = J(3,1);
J11BBB32 = J(3,2);
J11BBB33 = J(3,3);

aaa(1,1) = - 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(m,1);
aaa(1,2) = 3*Id2*L1 - (L1+2*M1)*L112(m,1);
aaa(1,3) = 0;
aaa(2,1) = - 3*Id1*L1 - L1*L111(m,1);
aaa(2,2) = 3*Id2*(L1+2*M1) + 3*ID2*(L2+2*M2) - (L1+2*M1)*L112(m,1) + (L2+2*M2)*L112(m,1);
aaa(2,3) = 3*Id1*L2 + L2*L111(m,1);
aaa(3,1) = 0;
aaa(3,2) = - 3*ID2*L2 - (L2+2*M2)*L112(m,1);
aaa(3,3) = - 3*Id1*(L2+2*M2) - (L2+2*M2)*L111(m,1);

J = inv(aaa);

Jm1BBB11 = J(1,1);
Jm1BBB12 = J(1,2);
Jm1BBB13 = J(1,3);
Jm1BBB21 = J(2,1);
Jm1BBB22 = J(2,2);
Jm1BBB23 = J(2,3);
```

```
Jm1BBB31 = J(3,1);
Jm1BBB32 = J(3,2);
Jm1BBB33 = J(3,3);

aa(1,1) = 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(1,n1);
aa(1,2) = -3*Id2*L1 - (L1+2*M1)*L112(1,n1);
aa(2,1) = 3*Id1*L1 - L1*L111(1,n1);
aa(2,2) = - 3*(L1+2*M1)*Id2 - L1*L112(1,n1);

J = inv(aa);

J1nBB11 = J(1,1);
J1nBB12 = J(1,2);
J1nBB21 = J(2,1);
J1nBB22 = J(2,2);

aa(1,1) = - 3*Id1*(L1+2*M1) - (L1+2*M1)*L111(m,n1);
aa(1,2) = - 3*Id2*L1 - (L1+2*M1)*L112(m,n1);
aa(2,1) = - 3*Id1*L1 - L1*L111(m,n1);
aa(2,2) = - 3*(L1+2*M1)*Id2 - L1*L112(m,n1);

J = inv(aa);

JmnBB11 = J(1,1);
JmnBB12 = J(1,2);
JmnBB21 = J(2,1);
JmnBB22 = J(2,2);

%% Main code
while errr < p

    pp = norm(u1,2) + norm(u2,2) + norm(v1,2) + norm(v2,2) ...
        + norm(du1,2) + norm(du2,2) + norm(dv1,2) + norm(dv2,2);

    for k = 1:NNN

        %% Corners
        u1d1 = (4*u1(2,1)-u1(3,1))*Id1;
        u2d2 = (4*u2(1,2)-u2(1,3))*Id2;
        v1d1 = (4*v1(2,n2)-v1(3,n2))*Id1;
        v2d2 = -(4*v2(1,n2-1)-v2(1,n2-2))*ID2;

        Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T1;
        Xu2 = L1*u1d1 + (L1+2*M1)*u2d2;
        Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
        Xv2 = L2*v1d1 + (L2+2*M2)*v2d2;

        u1(1,1)  = J11BBB11*Xu1 + J11BBB12*(Xu2-Xv2) + J11BBB13*Xv1;
        u2(1,1)  = J11BBB21*Xu1 + J11BBB22*(Xu2-Xv2) + J11BBB23*Xv1;
        v1(1,n2) = J11BBB31*Xu1 + J11BBB32*(Xu2-Xv2) + J11BBB33*Xv1;
```

```
v2(1,n2) = u2(1,1);

u1d1 = -(4*u1(m-1,1)-u1(m-2,1))*Id1;
u2d2 = (4*u2(m,2)-u2(m,3))*Id2;
v1d1 = -(4*v1(m-1,n2)-v1(m-2,n2))*Id1;
v2d2 = -(4*v2(m,n2-1)-v2(m,n2-2))*ID2;

Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T2;
Xu2 = L1*u1d1 + (L1+2*M1)*u2d2;
Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
Xv2 = L2*v1d1 + (L2+2*M2)*v2d2;

u1(m,1)  = Jm1BBB11*Xu1 + Jm1BBB12*(Xu2-Xv2) + Jm1BBB13*Xv1;
u2(m,1)  = Jm1BBB21*Xu1 + Jm1BBB22*(Xu2-Xv2) + Jm1BBB23*Xv1;
v1(m,n2) = Jm1BBB31*Xu1 + Jm1BBB32*(Xu2-Xv2) + Jm1BBB33*Xv1;

v2(m,n2) = u2(m,1);

u1d1 = (4*u1(2,n1)-u1(3,n1))*Id1;
u2d2 = -(4*u2(1,n1-1)-u2(1,n1-2))*Id2;

Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T1;
Xu2 = L1*u1d1 + (L1+2*M1)*u2d2;

u1(1,n1) = J1nBB11*Xu1 + J1nBB12*Xu2;
u2(1,n1) = J1nBB21*Xu1 + J1nBB22*Xu2;

u1d1 = -(4*u1(m-1,n1)-u1(m-2,n1))*Id1;
u2d2 = -(4*u2(m,n1-1)-u2(m,n1-2))*Id2;

Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T2;
Xu2 = L1*u1d1 + (L1+2*M1)*u2d2;

u1(m,n1) = JmnBB11*Xu1 + JmnBB12*Xu2;
u2(m,n1) = JmnBB21*Xu1 + JmnBB22*Xu2;

for i = 2:m-1

    %% Contact region % equations (43) and (44)
    u1d1 = (u1(i+1,1)-u1(i-1,1))*Id1;
    u2d1 = (u2(i+1,1)-u2(i-1,1))*Id1;
    u1d2 = (4*u1(i,2)-u1(i,3))*Id2;
    u2d2 = (4*u2(i,2)-u2(i,3))*Id2;

    v1d1 = (v1(i+1,n2)-v1(i-1,n2))*Id1;
    v2d1 = (v2(i+1,n2)-v2(i-1,n2))*Id1;
    v1d2 = -(4*v1(i,n2-1)-v1(i,n2-2))*ID2;
    v2d2 = -(4*v2(i,n2-1)-v2(i,n2-2))*ID2;

    du1d1 = (du1(i+1,1)-du1(i-1,1))*Id1;
    du2d1 = (du2(i+1,1)-du2(i-1,1))*Id1;
```

```
du1d2 = (4*du1(i,2)-du1(i,3))*Id2;
du2d2 = (4*du2(i,2)-du2(i,3))*Id2;

dv1d1 = (dv1(i+1,n2)-dv1(i-1,n2))*Id1;
dv2d1 = (dv2(i+1,n2)-dv2(i-1,n2))*Id1;
dv1d2 = -(4*dv1(i,n2-1)-dv1(i,n2-2))*ID2;
dv2d2 = -(4*dv2(i,n2-1)-dv2(i,n2-2))*ID2;

Xu1 = M1*(X(i,1)*u1d2 + IX(i,1)*u2d1);
Xu2 = L1*(u1d1+L111(i,1)*u1(i,1))  + (L1+2*M1)*u2d2;

Xv1 = M2*(X(i,1)*v1d2 + IX(i,1)*v2d1);
Xv2 = L2*(v1d1+L111(i,1)*v1(i,n2)) + (L2+2*M2)*v2d2;

u2(i,1)  = JBBB22(i)*(Xu2-Xv2);
v2(i,n2) = u2(i,1);

T12u = Xu1 - 3*M1*X(i,1)*u1(i,1)*Id2;
T22u = L1*(u1d1+L112(i,1)*u2(i,1))  + (L1+2*M1)*u2d2 - 3*(L1+2*M1)*u2(i,1)*Id2;

T12v = Xv1 + 3*M2*X(i,1)*v1(i,n2)*ID2;
T22v = L2*(v1d1+L112(i,1)*v2(i,n2)) + (L2+2*M2)*v2d2 + 3*(L2+2*M2)*v2(i,n2)*ID2;

T22 = T22u + L1*L111(i,1)*u1(i,1);

Xdu1 = M1*(X(i,1)*du1d2 + IX(i,1)*du2d1);
Xdv1 = M2*(X(i,1)*dv1d2 + IX(i,1)*dv2d1);

T22du  = L1*du1d1 + (L1+2*M1)*du2d2;
T22dv  = L2*dv1d1 + (L2+2*M2)*dv2d2;

delta =  X(i,1)*(u1(i,1)-v1(i,n2));
absdelta = abs(delta);

%% Limiting-equilibrium boundary
if ~(absdelta < epsi)

    XXu1 = (Xu1 + Mu*sign(delta)*T22u); % equation (45)
    XXv1 = (Xv1 + Mu*sign(delta)*T22v); % equation (46)

    B1u1 = 1/( 3*X(i,1)*M1*Id2-Mu*sign(delta)*L1*L111(i,1));
    B1v1 = 1/(-3*X(i,1)*M2*ID2-Mu*sign(delta)*L2*L111(i,1));

    u1(i,1)  = XXu1*B1u1;
    v1(i,n2) = XXv1*B1v1;

    du1(i,1)  = 0;
    dv1(i,n2) = 0;

end
```

```
    %% Bounded boundary
    if absdelta < epsi

        u1(i,1)  = (Xu1 - Xv1)*JBBB11(i); % Initial guess: FOR LARGE epsi COMMENT THIS LINE
        v1(i,n2) = u1(i,1); % Initial guess: FOR LARGE epsi COMMENT THIS LINE

        aa(1,1) =   3*X(i,1)*M1*Id2 - Iepsi*Mu*X(i,1)*T22 ...
            - Iepsi*Mu*delta*L1*L111(i,1);
        aa(1,2) =   Iepsi*Mu*X(i,1)*T22;
        aa(2,1) = - Iepsi*Mu*X(i,1)*T22;
        aa(2,2) = - 3*X(i,1)*M2*ID2 + Iepsi*Mu*X(i,1)*T22 ...
            - Iepsi*Mu*delta*L2*L111(i,1);

        J = inv(aa);

        Xdu1 = Xdu1 + Iepsi*Mu*delta*T22du + T12u + Iepsi*Mu*delta*T22; % equation (47)
        Xdv1 = Xdv1 + Iepsi*Mu*delta*T22dv + T12v + Iepsi*Mu*delta*T22; % equation (48)

        du1(i,1)  = Xdu1*J(1,1) + Xdv1*J(1,2);
        dv1(i,n2) = Xdu1*J(2,1) + Xdv1*J(2,2);

    end

    %% Stress-free boundary of the overlying body
    u1d1 = (u1(i+1,n1)-u1(i-1,n1))*Id1;
    u1d2 = -(4*u1(i,n1-1)-u1(i,n1-2))*Id2;
    u2d1 = (u2(i+1,n1)-u2(i-1,n1))*Id1;
    u2d2 = -(4*u2(i,n1-1)-u2(i,n1-2))*Id2;

    Xu1 = u1d2 + (IX(i,n1)^2)*u2d1;
    Xu2 = L1*u1d1 + (L1+2*M1)*u2d2;

    u1(i,n1) = JBB11(i)*Xu1 + JBB12(i)*Xu2;
    u2(i,n1) = JBB21(i)*Xu1 + JBB22(i)*Xu2;

end

%% Stressed boundaries of the overlying body
for j = 2:n1-1

    u1d1 = (4*u1(2,j)-u1(3,j))*Id1;
    u1d2 = (u1(1,j+1)-u1(1,j-1))*Id2;
    u2d1 = (4*u2(2,j)-u2(3,j))*Id1;
    u2d2 = (u2(1,j+1)-u2(1,j-1))*Id2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T1;
    Xu2 = u2d1 + (X(1,j)^2)*u1d2;

    u1(1,j) = J1S11(j)*Xu1 + J1S12(j)*Xu2;
    u2(1,j) = J1S21(j)*Xu1 + J1S22(j)*Xu2;
```

```matlab
    u1d1 = -(4*u1(m-1,j)-u1(m-2,j))*Id1;
    u1d2 = (u1(m,j+1)-u1(m,j-1))*Id2;
    u2d1 = -(4*u2(m-1,j)-u2(m-2,j))*Id1;
    u2d2 = (u2(m,j+1)-u2(m,j-1))*Id2;

    Xu1 = (L1+2*M1)*u1d1 + L1*u2d2 - T2;
    Xu2 = u2d1 + (X(m,j)^2)*u1d2;

    u1(m,j) = JmS11(j)*Xu1 + JmS12(j)*Xu2;
    u2(m,j) = JmS21(j)*Xu1 + JmS22(j)*Xu2;

end

%% Stress-free boundary of the foundation
for j = 2:n2-1

    v1d1 = (4*v1(2,j)-v1(3,j))*Id1;
    v2d1 = (4*v2(2,j)-v2(3,j))*Id1;
    v1d2 = (v1(1,j+1)-v1(1,j-1))*ID2;
    v2d2 = (v2(1,j+1)-v2(1,j-1))*ID2;

    Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
    Xv2 = v2d1 + (Y(1,j)^2)*v1d2;

    v1(1,j) = J1B11(j)*Xv1 + J1B12(j)*Xv2;
    v2(1,j) = J1B21(j)*Xv1 + J1B22(j)*Xv2;

    v1d1 = -(4*v1(m-1,j)-v1(m-2,j))*Id1;
    v2d1 = -(4*v2(m-1,j)-v2(m-2,j))*Id1;
    v1d2 = (v1(m,j+1)-v1(m,j-1))*ID2;
    v2d2 = (v2(m,j+1)-v2(m,j-1))*ID2;

    Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
    Xv2 = v2d1 + (Y(m,j)^2)*v1d2;

    v1(m,j) = JmB11(j)*Xv1 + JmB12(j)*Xv2;
    v2(m,j) = JmB21(j)*Xv1 + JmB22(j)*Xv2;

end

%% Governing equations of the overlying body
for i = 2:m-1

    for j = 2:n1-1

        u1d11 = (u1(i+1,j)+u1(i-1,j))*Id11;
        u1d22 = (u1(i,j+1)+u1(i,j-1))*Id22;
        u1d12 = (u1(i+1,j+1)-u1(i+1,j-1)-u1(i-1,j+1)+u1(i-1,j-1))*Id12;
        u1d1 = (u1(i+1,j)-u1(i-1,j))*Id1;
        u1d2 = (u1(i,j+1)-u1(i,j-1))*Id2;
```

```
        u2d11 = (u2(i+1,j)+u2(i-1,j))*Id11;
        u2d22 = (u2(i,j+1)+u2(i,j-1))*Id22;
        u2d12 = (u2(i+1,j+1)-u2(i+1,j-1)-u2(i-1,j+1)+u2(i-1,j-1))*Id12;
        u2d1 = (u2(i+1,j)-u2(i-1,j))*Id1;
        u2d2 = (u2(i,j+1)-u2(i,j-1))*Id2;

        Xu1 = A1d11*u1d11      + A1d22(i,j)*u1d22 + A2d12*u2d12      ...
            + A1d1(i,j)*u1d1 + A1d2(i,j)*u1d2 + A2d1(i,j)*u2d1;
        Xu2 = B2d11(i,j)*u2d11 + B2d22(i,j)*u2d22 + B1d12(i,j)*u1d12 ...
            + B1d1(i,j)*u1d1 + B1d2(i,j)*u1d2 + B2d1(i,j)*u2d1 + B2d2(i,j)*u2d2;

        u1(i,j)  = JM11(i,j)*Xu1 + JM12(i,j)*Xu2;
        u2(i,j)  = JM21(i,j)*Xu1 + JM22(i,j)*Xu2;

        du1d11 = (du1(i+1,j)+du1(i-1,j))*Id11;
        du1d22 = (du1(i,j+1)+du1(i,j-1))*Id22;
        du1d12 = (du1(i+1,j+1)-du1(i+1,j-1)-du1(i-1,j+1)+du1(i-1,j-1))*Id12;
        du1d1 = (du1(i+1,j)-du1(i-1,j))*Id1;
        du1d2 = (du1(i,j+1)-du1(i,j-1))*Id2;

        du2d11 = (du2(i+1,j)+du2(i-1,j))*Id11;
        du2d22 = (du2(i,j+1)+du2(i,j-1))*Id22;
        du2d12 = (du2(i+1,j+1)-du2(i+1,j-1)-du2(i-1,j+1)+du2(i-1,j-1))*Id12;
        du2d1 = (du2(i+1,j)-du2(i-1,j))*Id1;
        du2d2 = (du2(i,j+1)-du2(i,j-1))*Id2;

        Xdu1 = A1d11*du1d11      + A1d22(i,j)*du1d22 + A2d12*du2d12      ...
            + A1d1(i,j)*du1d1 + A1d2(i,j)*du1d2 + A2d1(i,j)*du2d1; % equation (39)
        Xdu2 = B2d11(i,j)*du2d11 + B2d22(i,j)*du2d22 + B1d12(i,j)*du1d12 ...
            + B1d1(i,j)*du1d1 + B1d2(i,j)*du1d2 ...
            + B2d1(i,j)*du2d1 + B2d2(i,j)*du2d2; % equation (40)

        du1(i,j)  = JM11(i,j)*Xdu1 + JM12(i,j)*Xdu2;
        du2(i,j)  = JM21(i,j)*Xdu1 + JM22(i,j)*Xdu2;

    end
end

%% Governing equations of the foundation
for i = 2:m-1

    for j = 2:n2-1

        v1d11 = (v1(i+1,j)+v1(i-1,j))*Id11;
        v1d22 = (v1(i,j+1)+v1(i,j-1))*ID22;
        v1d12 = (v1(i+1,j+1)-v1(i+1,j-1)-v1(i-1,j+1)+v1(i-1,j-1))*ID12;
        v1d1 = (v1(i+1,j)-v1(i-1,j))*Id1;
        v1d2 = (v1(i,j+1)-v1(i,j-1))*ID2;

        v2d11 = (v2(i+1,j)+v2(i-1,j))*Id11;
        v2d22 = (v2(i,j+1)+v2(i,j-1))*ID22;
```

```
                v2d12 = (v2(i+1,j+1)-v2(i+1,j-1)-v2(i-1,j+1)+v2(i-1,j-1))*ID12;
                v2d1 = (v2(i+1,j)-v2(i-1,j))*Id1;
                v2d2 = (v2(i,j+1)-v2(i,j-1))*ID2;

                Xv1 = C1d11*v1d11      + C1d22(i,j)*v1d22 + C2d12*v2d12      ...
                    + C1d1(i,j)*v1d1 + C1d2(i,j)*v1d2 + C2d1(i,j)*v2d1;
                Xv2 = D2d11(i,j)*v2d11 + D2d22(i,j)*v2d22 + D1d12(i,j)*v1d12 ...
                    + D1d1(i,j)*v1d1 + D1d2(i,j)*v1d2 + D2d1(i,j)*v2d1 + D2d2(i,j)*v2d2;

                v1(i,j) = JF11(i,j)*Xv1 + JF12(i,j)*Xv2;
                v2(i,j) = JF21(i,j)*Xv1 + JF22(i,j)*Xv2;

                dv1d11 = (dv1(i+1,j)+dv1(i-1,j))*Id11;
                dv1d22 = (dv1(i,j+1)+dv1(i,j-1))*ID22;
                dv1d12 = (dv1(i+1,j+1)-dv1(i+1,j-1)-dv1(i-1,j+1)+dv1(i-1,j-1))*ID12;
                dv1d1 = (dv1(i+1,j)-dv1(i-1,j))*Id1;
                dv1d2 = (dv1(i,j+1)-dv1(i,j-1))*ID2;

                dv2d11 = (dv2(i+1,j)+dv2(i-1,j))*Id11;
                dv2d22 = (dv2(i,j+1)+dv2(i,j-1))*ID22;
                dv2d12 = (dv2(i+1,j+1)-dv2(i+1,j-1)-dv2(i-1,j+1)+dv2(i-1,j-1))*ID12;
                dv2d1 = (dv2(i+1,j)-dv2(i-1,j))*Id1;
                dv2d2 = (dv2(i,j+1)-dv2(i,j-1))*ID2;

                Xdv1 = C1d11*dv1d11      + C1d22(i,j)*dv1d22 + C2d12*dv2d12      ...
                    + C1d1(i,j)*dv1d1 + C1d2(i,j)*dv1d2 + C2d1(i,j)*dv2d1; % equation (41)
                Xdv2 = D2d11(i,j)*dv2d11 + D2d22(i,j)*dv2d22 + D1d12(i,j)*dv1d12 ...
                    + D1d1(i,j)*dv1d1 + D1d2(i,j)*dv1d2 ...
                    + D2d1(i,j)*dv2d1 + D2d2(i,j)*dv2d2; % equation (42)

                dv1(i,j) = JF11(i,j)*Xdv1 + JF12(i,j)*Xdv2;
                dv2(i,j) = JF21(i,j)*Xdv1 + JF22(i,j)*Xdv2;

            end
        end

        %% Newton's method
        u1(:,:) = du1(:,:) + u1(:,:);
        u2(:,:) = du2(:,:) + u2(:,:);
        v1(:,:) = dv1(:,:) + v1(:,:);
        v2(:,:) = dv2(:,:) + v2(:,:);

    end

    %% Terminating condition
    ppp = norm(u1,2) + norm(u2,2) + norm(v1,2) + norm(v2,2) ...
        + norm(du1,2) + norm(du2,2) + norm(dv1,2) + norm(dv2,2);
    p = abs(1- ppp/pp);

end
```

```
%% Foundation
V1(:,:) = v1(:,:);
V2(:,:) = v2(:,:);
```

3.3 Benchmark Model

In this section we present a benchmark numerical code for the model that we introduced in this chapter, i.e. CH4.m. To do so we consider the special case $a = b = 2$, i.e. we consider polar coordinates.

Now, let v1= u^2_{shell}, v2= u^3_{shell}, vv1= u^2_{Kikuchi}, vv2= u^3_{Kikuchi}, dvv1= δu^2, dvv2= δu^3, ww1= v^2, ww2= v^3, dww1= δv^2, dww2= δv^3, h= h, H= H, T0= τ_0, Tmax= τ_{max}, E1= E, E2= $\bar{E}$, Nu1= ν, Nu2= $\bar{\nu}$, NN= N and Mu= ν_F. Thus, we find:

```
function CH4
format long
%% Benchmark Model
% Overlying elastic body on an elastic cylinder: friction case
% Contact angle is [0,pi]

%% INITIAL PARIMITERS
NN = 250; % Azimuthal grid points
error = 10^(-10); % Terminating error

E1 = 8000; % Young's modulus of the overlying body
E2 = 1000; % Young's modulus of the foundation

Nu1 = 0.25; % Poisson's ratio of the overlying body
Nu2 = 0.25; % Poisson's ratio of the foundation

h = 0.125; % Thickness of the overlying body
a = 2; % Outer radius
b = 1; % Inner radius

H = a-b; % Thickness of the foundation

Mu = 0.5; % Coefficient of friction
epsi = 10^(-10); % Regularisation parameter

T0 = 1; % Applied stress at \theta = 0
Tmax = 1; % Applied stress at \theta = pi

%% DO NOT CHANGE!
L1 = Nu1*E1/((1+Nu1)*(1-2*Nu1));
L2 = Nu2*E2/((1+Nu2)*(1-2*Nu2));

M1 = 0.5*E1/(1+Nu1);
M2 = 0.5*E2/(1+Nu2);
```

```
LL = 4*M1*(M1+L1)/(L1+2*M1);
IL = 1/LL;

NNN = NN^2;
errr = NNN*error;

ILT0 = T0*IL;
ILTmax = Tmax*IL;

ah = a+h;
Iah = 1/ah;
Ia = 1/a;
Iepsi = 1/epsi;

IMu = 0.5/Mu;

m = NN;  % Azimuthal grid points
q1 = H/(a*pi);
q2 = h/(ah*pi);
n1 = round(q1*NN-q1+1); % Radial grid points of the foundation
n2 = round(q2*NN-q2+1); % Radial grid points of the overlying body

dx1 = pi/(m-1); % Azimuthal grid spacing
dx2 = H/(n1-1); % Radial grid spacing of the foundation
dX2 = h/(n2-1); % Radial grid spacing of the overlying body

Idx1 = 1/(2*dx1);
Idx11 = (1/dx1)^2;

Idx2 = 1/(2*dx2);
Idx22 = (1/dx2)^2;

Idx12 = Idx1*Idx2;

w1 = zeros(m,1);
w2 = zeros(m,1);

v1 = zeros(m,n1); % Azimuthal displacement of the foundation (SHELL)
v2 = zeros(m,n1); % Radial displacement of the foundation (SHELL)

vv1 = zeros(m,n1); % Azimuthal displacement of the foundation (KIKUCHI)
vv2 = zeros(m,n1); % Radial displacement of the foundation (KIKUCHI)

dvv1 = zeros(m,n1);  % Perturbed azimuthal displacement of the foundation (KIKUCHI)
dvv2 = zeros(m,n1);  % Perturbed azimuthal displacement of the foundation (KIKUCHI)

ww1 = zeros(m,n2); % Azimuthal displacement of the overlying body (KIKUCHI)
ww2 = zeros(m,n2); % Radial displacement of the overlying body (KIKUCHI)

dww1 = zeros(m,n2);  % Perturbed azimuthal displacement of the overlying body (KIKUCHI)
```

```
dww2 = zeros(m,n2); % Perturbed radial displacement of the overlying body (KIKUCHI)

aa = zeros(2,2);
aaa = zeros(3,3);

r = zeros(1,n1);
Ir = zeros(1,n1);

a1d11 = zeros(1,n1);
a1d22 = zeros(1,n1);
a1d2 = zeros(1,n1);
a2d12 = zeros(1,n1);
a2d1 = zeros(1,n1);

b1d12 = zeros(1,n1);
b1d1 = zeros(1,n1);
b2d11 = zeros(1,n1);
b2d22 = zeros(1,n1);
b2d2 = zeros(1,n1);

Iv1 = zeros(1,n1);
Iv2 = zeros(1,n1);

B1v11 = zeros(1,n1);
B1v12 = zeros(1,n1);
B1v21 = zeros(1,n1);
B1v22 = zeros(1,n1);

Bmv11 = zeros(1,n1);
Bmv12 = zeros(1,n1);
Bmv21 = zeros(1,n1);
Bmv22 = zeros(1,n1);

rr = zeros(1,n2);
Irr = zeros(1,n2);

aa1d11 = zeros(1,n2);
aa1d22 = zeros(1,n2);
aa1d2 = zeros(1,n2);
aa2d12 = zeros(1,n2);
aa2d1 = zeros(1,n2);

bb1d12 = zeros(1,n2);
bb1d1 = zeros(1,n2);
bb2d11 = zeros(1,n2);
bb2d22 = zeros(1,n2);
bb2d2 = zeros(1,n2);

Iw1 = zeros(1,n2);
Iw2 = zeros(1,n2);
```

```
B1w11 = zeros(1,n2);
B1w12 = zeros(1,n2);
B1w21 = zeros(1,n2);
B1w22 = zeros(1,n2);

Bmw11 = zeros(1,n2);
Bmw12 = zeros(1,n2);
Bmw21 = zeros(1,n2);
Bmw22 = zeros(1,n2);

%% Shell Model with Friction
A1d11 = 1+(4/3)*(h/a)^2;
A1d2 = - M2*IL*(a^2)/h;
A2d111 = - (2/3)*(h^2)*a^(-3);
A2d1 = (1/a) + (2/3)*(h^2)*a^(-3) - M2*IL/h;

B2d1111 = (1/3)*(h^2)*a^(-3);
B2d11 = - (2/3)*(h^2)*a^(-3);
B2 = (1/a) + (1/3)*(h^2)*a^(-3) + L2*IL/h;
B2d2 = (L2+2*M2)*IL*(a/h);
B1d111 = -(2/3)*(h/a)^2;
B1d1 = 1 + (2/3)*(h/a)^2 + L2*IL*(a/h);

Iu1 = 1/(2*A1d11*Idx11 - 3*A1d2*Idx2);
Iu2 = 1/(2*(B2d11-2*B2d1111*Idx11)*Idx11 - B2 - 3*B2d2*Idx2);

IXv1 = 1/(1/Iu1 - ((a*IMu)^2)/Iu2);

%% Curvature terms of the foundation
for j = 1:n1

    r(j) = b + (j-1)*dx2;
    Ir(j) = 1/r(j);

    a1d11(j) = (L2+2*M2);
    a1d22(j) = M2*r(j)^2;
    a1d2(j) = 3*M2*r(j);
    a2d12(j) = (L2+M2);
    a2d1(j) = (L2+3*M2)*Ir(j);

    b1d12(j) = (L2+M2)*r(j);
    b1d1(j) = -2*M2;
    b2d11(j) = M2*Ir(j);
    b2d22(j) = (L2+2*M2)*r(j);
    b2d2(j) = (L2+2*M2);
    b2 = -(L2+2*M2)*Ir(j);

    Iv1(j) = 1/(2*a1d11(j)*Idx11 + 2*a1d22(j)*Idx22);
    Iv2(j) = 1/(2*b2d11(j)*Idx11 + 2*b2d22(j)*Idx22 - b2);

    aa(1,1) = 3*Idx1*(L2+2*M2);
```

```
    aa(1,2) = - (L2+2*M2)*Ir(j);
    aa(2,1) = 0;
    aa(2,2) = 3*Idx1;

    J = inv(aa);

    B1v11(j) = J(1,1);
    B1v12(j) = J(1,2);
    B1v21(j) = J(2,1);
    B1v22(j) = J(2,2);

    aa(1,1) = - 3*Idx1*(L2+2*M2);
    aa(1,2) = - (L2+2*M2)*Ir(j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Idx1;

    J = inv(aa);

    Bmv11(j) = J(1,1);
    Bmv12(j) = J(1,2);
    Bmv21(j) = J(2,1);
    Bmv22(j) = J(2,2);

end

aaa(1,1) = 3*Idx1;
aaa(1,2) = - Ia;
aaa(1,3) = 0;
aaa(2,1) = 0;
aaa(2,2) = 3*Idx1;
aaa(2,3) = 0;
aaa(3,1) = 4*Idx11;
aaa(3,2) = 0;
aaa(3,3) = 3*Ia*Idx1;

JSB1 = inv(aaa);

aaa(1,1) = -3*Idx1;
aaa(1,2) = - Ia;
aaa(1,3) = 0;
aaa(2,1) = 0;
aaa(2,2) = -3*Idx1;
aaa(2,3) = 0;
aaa(3,1) = 4*Idx11;
aaa(3,2) = 0;
aaa(3,3) = -3*Ia*Idx1;

JSBm = inv(aaa);

p = 2;
```

```matlab
%% Main code
while errr < p

    pp = norm(v1,2) + norm(v2,2);

    for k = 1:NNN

        %% Boundary conditions of the shell
        w1(1) = (2*v1(1,n1)-5*v1(2,n1)+4*v1(3,n1)-v1(4,n1))*Idx11;

        v1d1 = (4*v1(2,n1)-v1(3,n1))*Idx1;
        v2d1 = (4*v2(2,n1)-v2(3,n1))*Idx1;
        w2d1 = (4*w2(2)-w2(3))*Idx1;

        Xv1 = v1d1 - ILT0;
        Xv2 = v2d1;
        Xw2 = - 2*(-5*v1(2,n1)+4*v1(3,n1)-v1(4,n1))*Idx11 + Ia*w2d1;

        v1(1,n1) = Xv1*JSB1(1,1) + Xv2*JSB1(1,2) + Xw2*JSB1(1,3);
        v2(1,n1) = Xv1*JSB1(2,1) + Xv2*JSB1(2,2) + Xw2*JSB1(2,3);
        w2(1) = Xv1*JSB1(3,1) + Xv2*JSB1(3,2) + Xw2*JSB1(3,3);

        v1d1 = -(4*v1(m-1,n1)-v1(m-2,n1))*Idx1;
        v2d1 = -(4*v2(m-1,n1)-v2(m-2,n1))*Idx1;
        w2d1 = -(4*w2(m-1)-w2(m-2))*Idx1;

        w1(m) = (2*v1(m,n1)-5*v1(m-1,n1)+4*v1(m-2,n1)-v1(m-3,n1))*Idx11;

        Xv1 = v1d1 - ILTmax;
        Xv2 = v2d1;
        Xw2 = - 2*(-5*v1(m-1,n1)+4*v1(m-2,n1)-v1(m-3,n1))*Idx11 + Ia*w2d1;

        v1(m,n1) = Xv1*JSBm(1,1) + Xv2*JSBm(1,2) + Xw2*JSBm(1,3);
        v2(m,n1) = Xv1*JSBm(2,1) + Xv2*JSBm(2,2) + Xw2*JSBm(2,3);
        w2(m) = Xv1*JSBm(3,1) + Xv2*JSBm(3,2) + Xw2*JSBm(3,3);

        %% Governing equations of the shell
        for i = 2:m-1

            w1(i) = (v1(i+1,n1)-2*v1(i,n1)+v1(i-1,n1))*Idx11;
            w2(i) = (v2(i+1,n1)-2*v2(i,n1)+v2(i-1,n1))*Idx11;

            v1d1 = (v1(i+1,n1)-v1(i-1,n1))*Idx1;
            v2d1 = (v2(i+1,n1)-v2(i-1,n1))*Idx1;

            v1d11 = (v1(i+1,n1)+v1(i-1,n1))*Idx11;
            v2d11 = (v2(i+1,n1)+v2(i-1,n1))*Idx11;

            w1d1 = (w1(i+1)-w1(i-1))*Idx1;
            w2d1 = (w2(i+1)-w2(i-1))*Idx1;
```

```
    w2d11 = (w2(i+1)+w2(i-1))*Idx11;

    v1d2 = -(4*v1(i,n1-1)-v1(i,n1-2))*Idx2;
    v2d2 = -(4*v2(i,n1-1)-v2(i,n1-2))*Idx2;

    Xv1 = A1d11*v1d11 + A1d2*v1d2 + A2d111*w2d1 + A2d1*v2d1;
    Xv2 = B2d1111*w2d11 + (B2d11-2*B2d1111*Idx11)*v2d11 + B2d2*v2d2 ...
        + B1d111*w1d1 + B1d1*v1d1;

    %% Bounded boundary
    v1(i,n1) = Xv1*Iu1;
    v2(i,n1) = Xv2*Iu2;

    delta = 2*Mu*v2(i,n1) + a*abs(v1(i,n1));

    %% Limiting-equilibrium boundary
    if ~(delta < 0)

        v2(i,n1) = - IMu*a*abs(v1(i,n1));
        v1(i,n1) = (Xv1+a*IMu*sign(v1(i,n1))*Xv2)*IXv1;

    end
end

%% Stress free boundary of the foundation
for j = 2:n1-1

    v1d1 = (4*v1(2,j)-v1(3,j))*Idx1;
    v2d1 = (4*v2(2,j)-v2(3,j))*Idx1;
    v1d2 = (v1(1,j+1)-v1(1,j-1))*Idx2;
    v2d2 = (v2(1,j+1)-v2(1,j-1))*Idx2;

    Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
    Xv2 = v2d1 + (r(j)^2)*v1d2;

    v1(1,j) = B1v11(j)*Xv1 + B1v12(j)*Xv2;
    v2(1,j) = B1v21(j)*Xv1 + B1v22(j)*Xv2;

    v1d1 = -(4*v1(m-1,j)-v1(m-2,j))*Idx1;
    v2d1 = -(4*v2(m-1,j)-v2(m-2,j))*Idx1;
    v1d2 = (v1(m,j+1)-v1(m,j-1))*Idx2;
    v2d2 = (v2(m,j+1)-v2(m,j-1))*Idx2;

    Xv1 = (L2+2*M2)*v1d1 + L2*v2d2;
    Xv2 = v2d1 + (r(j)^2)*v1d2;

    v1(m,j) = Bmv11(j)*Xv1 + Bmv12(j)*Xv2;
    v2(m,j) = Bmv21(j)*Xv1 + Bmv22(j)*Xv2;

end
```

```
        %% Governing equations of the foundation
        for i = 2:m-1

            for j = 2:n1-1

                v1d1 = (v1(i+1,j)-v1(i-1,j))*Idx1;
                v1d2 = (v1(i,j+1)-v1(i,j-1))*Idx2;
                v2d1 = (v2(i+1,j)-v2(i-1,j))*Idx1;
                v2d2 = (v2(i,j+1)-v2(i,j-1))*Idx2;

                v1d11 = (v1(i+1,j)+v1(i-1,j))*Idx11;
                v1d22 = (v1(i,j+1)+v1(i,j-1))*Idx22;
                v2d11 = (v2(i+1,j)+v2(i-1,j))*Idx11;
                v2d22 = (v2(i,j+1)+v2(i,j-1))*Idx22;

                v1d12 = (v1(i+1,j+1)-v1(i-1,j+1)-v1(i+1,j-1)+v1(i-1,j-1))*Idx12;
                v2d12 = (v2(i+1,j+1)-v2(i-1,j+1)-v2(i+1,j-1)+v2(i-1,j-1))*Idx12;

                Xv1 = a1d11(j)*v1d11 + a1d22(j)*v1d22 + a1d2(j)*v1d2 ...
                    + a2d12(j)*v2d12 + a2d1(j)*v2d1;
                Xv2 = b1d12(j)*v1d12 + b1d1(j)*v1d1 + b2d11(j)*v2d11 ...
                    + b2d22(j)*v2d22 + b2d2(j)*v2d2;

                v1(i,j) = Xv1*Iv1(j);
                v2(i,j) = Xv2*Iv2(j);

            end
        end
    end

    %% Terminating condition
    ppp = norm(v1,2) + norm(v2,2);
    p = abs(1-ppp/pp);

end

%% Kikuchi and Oden's Model
IdX2 = 1/(2*dX2);
IdX22 = (1/dX2)^2;

IdX12 = Idx1*IdX2;

Bn2ww1 = 1/(-3*IdX2*ah^2);
Bn2ww2 = 1/(-3*(L1+2*M1)*IdX2 - L1*Iah);

B1ww1 = 1/(3*a*M1*IdX2);
B1vv1 = 1/(-3*a*M2*Idx2);

X1ww1 = 1/(3*a*M1*IdX2 + 3*a*M2*Idx2);
B1ww2 = 1/(3*(L1+2*M1)*IdX2 + 3*(L2+2*M2)*Idx2 - L1*Ia + L2*Ia);
```

```
aa(1,1) = 3*(L1+2*M1)*Idx1;
aa(1,2) = - (L1+2*M1)*Iah - 3*L1*IdX2;
aa(2,1) = 3*L1*Idx1;
aa(2,2) = - L1*Iah - 3*(L1+2*M1)*IdX2;

B1n2ww = inv(aa);

aa(1,1) = - 3*(L1+2*M1)*Idx1;
aa(1,2) = - (L1+2*M1)*Iah - 3*L1*IdX2;
aa(2,1) = - 3*L1*Idx1;
aa(2,2) = - L1*Iah - 3*(L1+2*M1)*IdX2;

Bmn2ww = inv(aa);

aaa(1,1) = 3*(L1+2*M1)*Idx1;
aaa(1,2) = - (L1+2*M1)*Ia + 3*L1*IdX2;
aaa(1,3) = 0;
aaa(2,1) = 3*L1*Idx1;
aaa(2,2) = - L1*Ia + L2*Ia + 3*(L1+2*M1)*IdX2 + 3*(L2+2*M2)*Idx2;
aaa(2,3) = - 3*L2*Idx1;
aaa(3,1) = 0;
aaa(3,2) = - (L2+2*M2)*Ia - 3*L2*Idx2;
aaa(3,3) = 3*(L2+2*M2)*Idx1;

B11ww = inv(aaa);

aaa(1,1) = - 3*(L1+2*M1)*Idx1;
aaa(1,2) = - (L1+2*M1)*Ia + 3*L1*IdX2;
aaa(1,3) = 0;
aaa(2,1) = - 3*L1*Idx1;
aaa(2,2) = - L1*Ia + L2*Ia + 3*(L1+2*M1)*IdX2 + 3*(L2+2*M2)*Idx2;
aaa(2,3) = 3*L2*Idx1;
aaa(3,1) = 0;
aaa(3,2) = - (L2+2*M2)*Ia - 3*L2*Idx2;
aaa(3,3) = - 3*(L2+2*M2)*Idx1;

Bm1ww = inv(aaa);

%% Curvature terms of the overlying body
for j = 1:n2

    rr(j) = a + (j-1)*dX2;
    Irr(j) = 1/rr(j);

    aa1d11(j) = (L1+2*M1);
    aa1d22(j) = M1*rr(j)^2;
    aa1d2(j) = 3*M1*rr(j);
    aa2d12(j) = (L1+M1);
    aa2d1(j) = (L1+3*M1)*Irr(j);

    bb1d12(j) = (L1+M1)*rr(j);
```

```
    bb1d1(j) = -2*M1;
    bb2d11(j) = M1*Irr(j);
    bb2d22(j) = (L1+2*M1)*rr(j);
    bb2d2(j) = (L1+2*M1);
    bb2 = -(L1+2*M1)*Irr(j);

    Iw1(j) = 1/(2*aa1d11(j)*Idx11 + 2*aa1d22(j)*IdX22);
    Iw2(j) = 1/(2*bb2d11(j)*Idx11 + 2*bb2d22(j)*IdX22 - bb2);

    aa(1,1) = 3*Idx1*(L1+2*M1);
    aa(1,2) = - (L1+2*M1)*Irr(j);
    aa(2,1) = 0;
    aa(2,2) = 3*Idx1;

    J = inv(aa);

    B1w11(j) = J(1,1);
    B1w12(j) = J(1,2);
    B1w21(j) = J(2,1);
    B1w22(j) = J(2,2);

    aa(1,1) = - 3*Idx1*(L1+2*M1);
    aa(1,2) = - (L1+2*M1)*Irr(j);
    aa(2,1) = 0;
    aa(2,2) = - 3*Idx1;

    J = inv(aa);

    Bmw11(j) = J(1,1);
    Bmw12(j) = J(1,2);
    Bmw21(j) = J(2,1);
    Bmw22(j) = J(2,2);

end

p = 2;

%% Main code
while errr < p

    pp = norm(ww1,2) + norm(ww2,2) + norm(dww1,2) + norm(dww2,2) ...
        + norm(vv1,2) + norm(vv2,2) + norm(dvv1,2) + norm(dvv2,2);

    for k = 1:NNN

        %% Corners
        ww1d1 = (4*ww1(2,n2)-ww1(3,n2))*Idx1;
        ww2d2 = -(4*ww2(1,n2-1)-ww2(1,n2-2))*IdX2;

        Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - T0;
        Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;
```

```
ww1(1,n2) = B1n2ww(1,1)*Xww1 + B1n2ww(1,2)*Xww2;
ww2(1,n2) = B1n2ww(2,1)*Xww1 + B1n2ww(2,2)*Xww2;

ww1d1 = -(4*ww1(m-1,n2)-ww1(m-2,n2))*Idx1;
ww2d2 = -(4*ww2(m,n2-1)-ww2(m,n2-2))*IdX2;

Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - Tmax;
Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

ww1(m,n2) = Bmn2ww(1,1)*Xww1 + Bmn2ww(1,2)*Xww2;
ww2(m,n2) = Bmn2ww(2,1)*Xww1 + Bmn2ww(2,2)*Xww2;

ww1d1 = (4*ww1(2,1)-ww1(3,1))*Idx1;
ww2d2 = (4*ww2(1,2)-ww2(1,3))*IdX2;
vv1d1 = (4*vv1(2,n1)-vv1(3,n1))*Idx1;
vv2d2 = -(4*vv2(1,n1-1)-vv2(1,n1-2))*Idx2;

Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - T0;
Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;
Xvv1 = (L2+2*M2)*vv1d1 + L2*vv2d2;
Xvv2 = L2*vv1d1 + (L2+2*M2)*vv2d2;

ww1(1,1)  = B11ww(1,1)*Xww1 + B11ww(1,2)*(Xww2-Xvv2) + B11ww(1,3)*Xvv1;
ww2(1,1)  = B11ww(2,1)*Xww1 + B11ww(2,2)*(Xww2-Xvv2) + B11ww(2,3)*Xvv1;
vv1(1,n1) = B11ww(3,1)*Xww1 + B11ww(3,2)*(Xww2-Xvv2) + B11ww(3,3)*Xvv1;
vv2(1,n1) = ww2(1,1);

ww1d1 = -(4*ww1(m-1,1)-ww1(m-2,1))*Idx1;
ww2d2 = (4*ww2(m,2)-ww2(m,3))*IdX2;
vv1d1 = -(4*vv1(m-1,n1)-vv1(m-2,n1))*Idx1;
vv2d2 = -(4*vv2(m,n1-1)-vv2(m,n1-2))*Idx2;

Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - Tmax;
Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;
Xvv1 = (L2+2*M2)*vv1d1 + L2*vv2d2;
Xvv2 = L2*vv1d1 + (L2+2*M2)*vv2d2;

ww1(m,1)  = Bm1ww(1,1)*Xww1 + Bm1ww(1,2)*(Xww2-Xvv2) + Bm1ww(1,3)*Xvv1;
ww2(m,1)  = Bm1ww(2,1)*Xww1 + Bm1ww(2,2)*(Xww2-Xvv2) + Bm1ww(2,3)*Xvv1;
vv1(m,n1) = Bm1ww(3,1)*Xww1 + Bm1ww(3,2)*(Xww2-Xvv2) + Bm1ww(3,3)*Xvv1;
vv2(m,n1) = ww2(m,1);

for i = 2:m-1

    %% Stress-free boundary of the overlying body
    ww1d1 = (ww1(i+1,n2)-ww1(i-1,n2))*Idx1;
    ww2d1 = (ww2(i+1,n2)-ww2(i-1,n2))*Idx1;
    ww1d2 = -(4*ww1(i,n2-1)-ww1(i,n2-2))*IdX2;
    ww2d2 = -(4*ww2(i,n2-1)-ww2(i,n2-2))*IdX2;
```

```
Xww1 = ww1d2*ah^2 + ww2d1;
Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

ww1(i,n2) = Bn2ww1*Xww1;
ww2(i,n2) = Bn2ww2*Xww2;

%% Friction boundary
ww1d1 = (ww1(i+1,1)-ww1(i-1,1))*Idx1;
ww2d1 = (ww2(i+1,1)-ww2(i-1,1))*Idx1;
ww1d2 = (4*ww1(i,2)-ww1(i,3))*IdX2;
ww2d2 = (4*ww2(i,2)-ww2(i,3))*IdX2;

vv1d1 = (vv1(i+1,n1)-vv1(i-1,n1))*Idx1;
vv2d1 = (vv2(i+1,n1)-vv2(i-1,n1))*Idx1;
vv1d2 = -(4*vv1(i,n1-1)-vv1(i,n1-2))*Idx2;
vv2d2 = -(4*vv2(i,n1-1)-vv2(i,n1-2))*Idx2;

dww1d1 = (dww1(i+1,1)-dww1(i-1,1))*Idx1;
dww2d1 = (dww2(i+1,1)-dww2(i-1,1))*Idx1;
dww1d2 = (4*dww1(i,2)-dww1(i,3))*IdX2;
dww2d2 = (4*dww2(i,2)-dww2(i,3))*IdX2;

dvv1d1 = (dvv1(i+1,n1)-dvv1(i-1,n1))*Idx1;
dvv2d1 = (dvv2(i+1,n1)-dvv2(i-1,n1))*Idx1;
dvv1d2 = -(4*dvv1(i,n1-1)-dvv1(i,n1-2))*Idx2;
dvv2d2 = -(4*dvv2(i,n1-1)-dvv2(i,n1-2))*Idx2;

Xww1 = M1*(a*ww1d2 + Ia*ww2d1);
Xww2 = L1*ww1d1 + (L1+2*M1)*ww2d2;

Xvv1 = M2*(a*vv1d2 + Ia*vv2d1);
Xvv2 = L2*vv1d1 + (L2+2*M2)*vv2d2;

ww2(i,1) = B1ww2*(Xww2 - Xvv2);
vv2(i,n1) = ww2(i,1);

T12w = Xww1 - 3*a*M1*ww1(i,1)*IdX2;
T12v = Xvv1 + 3*a*M2*vv1(i,n1)*Idx2;
T22  = Xww2 - 3*(L1+2*M1)*ww2(i,1)*IdX2 + Ia*L1*ww2(i,1);

Xdww1 = M1*(a*dww1d2 + Ia*dww2d1);
Xdvv1 = M2*(a*dvv1d2 + Ia*dvv2d1);

T22dw  = L1*dww1d1 + (L1+2*M1)*dww2d2;
T22dv  = L2*dvv1d1 + (L2+2*M2)*dvv2d2;

delta =  a*(ww1(i,1)-vv1(i,n1));
absdelta = abs(delta);

%% Limiting-equilibrium boundary
if ~(absdelta < epsi)
```

```
        ww1(i,1)  = (Xww1 + Mu*sign(delta)*T22)*B1ww1;
        vv1(i,n1) = (Xvv1 + Mu*sign(delta)*T22)*B1vv1;

        dww1(i,1)  = 0;
        dvv1(i,n1) = 0;

    end

    %% Bounded boundary
    if absdelta < epsi

        ww1(i,1)  = X1ww1*(Xww1 - Xvv1); % Initial guess: FOR LARGE epsi COMMENT THIS LINE
        vv1(i,n1) = ww1(i,1); % Initial guess: FOR LARGE epsi COMMENT THIS LINE

        aa(1,1) =   3*a*M1*IdX2 - Iepsi*Mu*a*T22;
        aa(1,2) =   Iepsi*Mu*a*T22;
        aa(1,2) = - Iepsi*Mu*a*T22;
        aa(2,2) = - 3*a*M2*Idx2 + Iepsi*Mu*a*T22;

        J = inv(aa);

        Xdww1 = Xdww1 + Iepsi*Mu*delta*T22dw + T12w + Iepsi*Mu*delta*T22;
        Xdvv1 = Xdvv1 + Iepsi*Mu*delta*T22dv + T12v + Iepsi*Mu*delta*T22;

        dww1(i,1)  = Xdww1*J(1,1) + Xdvv1*J(1,2);
        dvv1(i,n1) = Xdww1*J(2,1) + Xdvv1*J(2,2);

    end
end

%% Stressed boundary of the overlying body
for j = 2:n2-1

    ww1d1 = (4*ww1(2,j)-ww1(3,j))*Idx1;
    ww2d1 = (4*ww2(2,j)-ww2(3,j))*Idx1;
    ww1d2 = (ww1(1,j+1)-ww1(1,j-1))*IdX2;
    ww2d2 = (ww2(1,j+1)-ww2(1,j-1))*IdX2;

    Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - T0;
    Xww2 = ww2d1 + (rr(j)^2)*ww1d2;

    ww1(1,j)  = B1w11(j)*Xww1 + B1w12(j)*Xww2;
    ww2(1,j)  = B1w21(j)*Xww1 + B1w22(j)*Xww2;

    ww1d1 = -(4*ww1(m-1,j)-ww1(m-2,j))*Idx1;
    ww2d1 = -(4*ww2(m-1,j)-ww2(m-2,j))*Idx1;
    ww1d2 = (ww1(m,j+1)-ww1(m,j-1))*IdX2;
    ww2d2 = (ww2(m,j+1)-ww2(m,j-1))*IdX2;

    Xww1 = (L1+2*M1)*ww1d1 + L1*ww2d2 - Tmax;
```

```
    Xww2 = ww2d1 + (rr(j)^2)*ww1d2;

    ww1(m,j) = Bmw11(j)*Xww1 + Bmw12(j)*Xww2;
    ww2(m,j) = Bmw21(j)*Xww1 + Bmw22(j)*Xww2;

end

%% Stress-free boundary of the foundation
for j = 2:n1-1

    vv1d1 = (4*vv1(2,j)-vv1(3,j))*Idx1;
    vv2d1 = (4*vv2(2,j)-vv2(3,j))*Idx1;
    vv1d2 = (vv1(1,j+1)-vv1(1,j-1))*Idx2;
    vv2d2 = (vv2(1,j+1)-vv2(1,j-1))*Idx2;

    Xvv1 = (L2+2*M2)*vv1d1 + L2*vv2d2;
    Xvv2 = vv2d1 + (r(j)^2)*vv1d2;

    vv1(1,j) = B1v11(j)*Xvv1 + B1v12(j)*Xvv2;
    vv2(1,j) = B1v21(j)*Xvv1 + B1v22(j)*Xvv2;

    vv1d1 = -(4*vv1(m-1,j)-vv1(m-2,j))*Idx1;
    vv2d1 = -(4*vv2(m-1,j)-vv2(m-2,j))*Idx1;
    vv1d2 = (vv1(m,j+1)-vv1(m,j-1))*Idx2;
    vv2d2 = (vv2(m,j+1)-vv2(m,j-1))*Idx2;

    Xvv1 = (L2+2*M2)*vv1d1 + L2*vv2d2;
    Xvv2 = vv2d1 + (r(j)^2)*vv1d2;

    vv1(m,j) = Bmv11(j)*Xvv1 + Bmv12(j)*Xvv2;
    vv2(m,j) = Bmv21(j)*Xvv1 + Bmv22(j)*Xvv2;

end

%% Governing equations of the overlying body
for i = 2:m-1

    for j = 2:n2-1

        ww1d1 = (ww1(i+1,j)-ww1(i-1,j))*Idx1;
        ww1d2 = (ww1(i,j+1)-ww1(i,j-1))*IdX2;
        ww2d1 = (ww2(i+1,j)-ww2(i-1,j))*Idx1;
        ww2d2 = (ww2(i,j+1)-ww2(i,j-1))*IdX2;

        ww1d11 = (ww1(i+1,j)+ww1(i-1,j))*Idx11;
        ww1d22 = (ww1(i,j+1)+ww1(i,j-1))*IdX22;
        ww2d11 = (ww2(i+1,j)+ww2(i-1,j))*Idx11;
        ww2d22 = (ww2(i,j+1)+ww2(i,j-1))*IdX22;

        ww1d12 = (ww1(i+1,j+1)-ww1(i-1,j+1)-ww1(i+1,j-1)+ww1(i-1,j-1))*IdX12;
        ww2d12 = (ww2(i+1,j+1)-ww2(i-1,j+1)-ww2(i+1,j-1)+ww2(i-1,j-1))*IdX12;
```

```
    Xww1 = aa1d11(j)*ww1d11 + aa1d22(j)*ww1d22 + aa1d2(j)*ww1d2 ...
        + aa2d12(j)*ww2d12 + aa2d1(j)*ww2d1;
    Xww2 = bb1d12(j)*ww1d12 + bb1d1(j)*ww1d1 + bb2d11(j)*ww2d11 ...
        + bb2d22(j)*ww2d22 + bb2d2(j)*ww2d2;

    ww1(i,j) = Xww1*Iw1(j);
    ww2(i,j) = Xww2*Iw2(j);

    dww1d1 = (dww1(i+1,j)-dww1(i-1,j))*Idx1;
    dww1d2 = (dww1(i,j+1)-dww1(i,j-1))*IdX2;
    dww2d1 = (dww2(i+1,j)-dww2(i-1,j))*Idx1;
    dww2d2 = (dww2(i,j+1)-dww2(i,j-1))*IdX2;

    dww1d11 = (dww1(i+1,j)+dww1(i-1,j))*Idx11;
    dww1d22 = (dww1(i,j+1)+dww1(i,j-1))*IdX22;
    dww2d11 = (dww2(i+1,j)+dww2(i-1,j))*Idx11;
    dww2d22 = (dww2(i,j+1)+dww2(i,j-1))*IdX22;

    dww1d12 = (dww1(i+1,j+1)-dww1(i-1,j+1)-dww1(i+1,j-1)+dww1(i-1,j-1))*IdX12;
    dww2d12 = (dww2(i+1,j+1)-dww2(i-1,j+1)-dww2(i+1,j-1)+dww2(i-1,j-1))*IdX12;

    Xdww1 = aa1d11(j)*dww1d11 + aa1d22(j)*dww1d22 + aa1d2(j)*dww1d2 ...
        + aa2d12(j)*dww2d12 + aa2d1(j)*dww2d1;
    Xdww2 = bb1d12(j)*dww1d12 + bb1d1(j)*dww1d1 + bb2d11(j)*dww2d11 ...
        + bb2d22(j)*dww2d22 + bb2d2(j)*dww2d2;

    dww1(i,j) = Xdww1*Iw1(j);
    dww2(i,j) = Xdww2*Iw2(j);

end

%% Governing equations of the foundation
for j = 2:n1-1

    vv1d1 = (vv1(i+1,j)-vv1(i-1,j))*Idx1;
    vv1d2 = (vv1(i,j+1)-vv1(i,j-1))*Idx2;
    vv2d1 = (vv2(i+1,j)-vv2(i-1,j))*Idx1;
    vv2d2 = (vv2(i,j+1)-vv2(i,j-1))*Idx2;

    vv1d11 = (vv1(i+1,j)+vv1(i-1,j))*Idx11;
    vv1d22 = (vv1(i,j+1)+vv1(i,j-1))*Idx22;
    vv2d11 = (vv2(i+1,j)+vv2(i-1,j))*Idx11;
    vv2d22 = (vv2(i,j+1)+vv2(i,j-1))*Idx22;

    vv1d12 = (vv1(i+1,j+1)-vv1(i-1,j+1)-vv1(i+1,j-1)+vv1(i-1,j-1))*Idx12;
    vv2d12 = (vv2(i+1,j+1)-vv2(i-1,j+1)-vv2(i+1,j-1)+vv2(i-1,j-1))*Idx12;

    Xvv1 = a1d11(j)*vv1d11 + a1d22(j)*vv1d22 + a1d2(j)*vv1d2 ....
        + a2d12(j)*vv2d12 + a2d1(j)*vv2d1;
    Xvv2 = b1d12(j)*vv1d12 + b1d1(j)*vv1d1 + b2d11(j)*vv2d11 ...
```

```
                    + b2d22(j)*vv2d22 + b2d2(j)*vv2d2;

                vv1(i,j) = Xvv1*Iv1(j);
                vv2(i,j) = Xvv2*Iv2(j);

                dvv1d1 = (dvv1(i+1,j)-dvv1(i-1,j))*Idx1;
                dvv1d2 = (dvv1(i,j+1)-dvv1(i,j-1))*Idx2;
                dvv2d1 = (dvv2(i+1,j)-dvv2(i-1,j))*Idx1;
                dvv2d2 = (dvv2(i,j+1)-dvv2(i,j-1))*Idx2;

                dvv1d11 = (dvv1(i+1,j)+dvv1(i-1,j))*Idx11;
                dvv1d22 = (dvv1(i,j+1)+dvv1(i,j-1))*Idx22;
                dvv2d11 = (dvv2(i+1,j)+dvv2(i-1,j))*Idx11;
                dvv2d22 = (dvv2(i,j+1)+dvv2(i,j-1))*Idx22;

                dvv1d12 = (dvv1(i+1,j+1)-dvv1(i-1,j+1)-dvv1(i+1,j-1)+dvv1(i-1,j-1))*Idx12;
                dvv2d12 = (dvv2(i+1,j+1)-dvv2(i-1,j+1)-dvv2(i+1,j-1)+dvv2(i-1,j-1))*Idx12;

                Xdvv1 = a1d11(j)*dvv1d11 + a1d22(j)*dvv1d22 + a1d2(j)*dvv1d2 ...
                    + a2d12(j)*dvv2d12 + a2d1(j)*dvv2d1;
                Xdvv2 = b1d12(j)*dvv1d12 + b1d1(j)*dvv1d1 + b2d11(j)*dvv2d11 ...
                    + b2d22(j)*dvv2d22 + b2d2(j)*dvv2d2;

                dvv1(i,j) = Xdvv1*Iv1(j);
                dvv2(i,j) = Xdvv2*Iv2(j);

            end
        end

        %% Newton's method
        ww1(:,:) = dww1(:,:) + ww1(:,:);
        ww2(:,:) = dww2(:,:) + ww2(:,:);
        vv1(:,:) = dvv1(:,:) + vv1(:,:);
        vv2(:,:) = dvv2(:,:) + vv2(:,:);

    end

    %% Terminating condition
    ppp = norm(ww1,2) + norm(ww2,2) + norm(dww1,2) + norm(dww2,2) ...
        + norm(vv1,2) + norm(vv2,2) + norm(dvv1,2) + norm(dvv2,2);
    p = abs(1-ppp/pp);

end

%% Error
normV1 = norm(v1(:,:)-vv1(:,:),2)/norm(v1(:,:)+vv1(:,:),2); % Azimuthal error
normV2 = norm(v2(:,:)-vv2(:,:),2)/norm(v2(:,:)+vv2(:,:),2); % Radial error

%% Save data file
filename = 'CH4.mat';
save(filename)
```

4 Belt-Friction Model

In this chapter we present a numerical code for an example of a membrane on a rigid foundation with dynamic friction (see section 6.3 of Jayawardana [4]) implemented in Matlab, i.e. `FrictionCode.m`.

To conduct numerical experiments assume a surface of revolution case, where the both contact surface and the unstressed true-membrane parameterised by the same immersion (i.e. an aximembrane case). Let this immersion be $\boldsymbol{\sigma}(x^1,x^2)=(x^1,\varphi(x^1)\sin(x^2),\varphi(x^1)\cos(x^2))_{\mathsf{E}}$, where $x^1\in(0,l)$ and $x^2\in(-\frac{1}{2}\pi,0)$. To keep the contact area as a surface of positive mean-curvature, assert that $\varphi(x^1)=r_0-16c(l^{-1}x^1-\frac{1}{2})^4$, where $c<r_0$. Note that l, c and r_0 are some positive constants that one can specify later. With some calculations, one finds that the first fundamental form tensor is $\boldsymbol{F}_{[\mathrm{I}]}=\mathrm{diag}((\psi_1)^2,(\psi_2)^2)$, where $\psi_1=(1+(\varphi'(x^1))^2)^{\frac{1}{2}}$ and $\psi_2=\varphi(x^1)$. With a few more calculations, one finds

$$\Gamma^1_{11}=(\psi_1)^{-1}\partial_1\psi_1\ ,\qquad F_{[\mathrm{II}]}{}^1_1=(\psi_1)^{-1}\varphi''(x^1)\left(1+(\varphi'(x^1))^2\right)^{-1}\ ,$$
$$\Gamma^2_{21}=(\psi_2)^{-1}\partial_1\psi_2\ ,\qquad F_{[\mathrm{II}]}{}^2_2=-(\psi_2)^{-1}\left(1+(\varphi'(x^1))^2\right)^{-\frac{1}{2}}\ ,$$

where $\Gamma^\gamma_{\alpha\beta}$ are Christoffel symbols of the second kind and $\boldsymbol{F}_{[\mathrm{II}]}$ is the second fundamental form tensor. Now, given that $\boldsymbol{w}=(w^1(x^1,x^2),w^2(x^1,x^2))$ is the displacement field, one finds that the covariant derivatives are

$$\begin{aligned}
\nabla_1 w^1&=\partial_1 w^1+\Gamma^1_{11}w^1\ ,\\
\nabla_1 w^2&=\partial_1 w^2+\Gamma^2_{21}w^2\ ,\\
\nabla_2 w^1&=\partial_2 w^1-(\psi_1)^{-2}(\psi_2)^2\Gamma^2_{21}w^2\ ,\\
\nabla_2 w^2&=\partial_2 w^2+\Gamma^2_{22}w^2\ .
\end{aligned}$$

Now, assume that the membrane is subjected to the acceleration of gravity, i.e. subject to the field $(0,0,-g)_{\mathsf{E}}$. With a coordinate transform, from Euclidean to curvilinear, one may re-expresses acceleration due to gravity in curvilinear coordinates as $g\boldsymbol{J}$, where

$$\boldsymbol{J}=\left(-\varphi'(x^1)(\psi_1)^{-2}\cos(x^2),(\varphi(x^1))^{-1}\sin(x^2),-(\psi_1)^{-1}\cos(x^2)\right)\ ,$$

and x^2 is the acute angle that the vector $(0,0,1)_{\mathsf{E}}$ makes with the vector $(0,\psi_2,0)$.

Now, given that ϱ is the mass density, $(0,0,0)$ is the acceleration field and $(0,(\psi_2)^{-1}V,0)$ is the velocity field of the membrane, one can express the governing equations of the membrane as

$$(\Lambda-\mu)\partial^1\left(\nabla_\alpha w^\alpha\right)+\mu\Delta w^1+\varrho gJ^1+f^1_r(\boldsymbol{w})=-\varrho(\psi_1)^{-2}\Gamma^2_{21}V^2\ ,\tag{49}$$
$$(\Lambda-\mu)\partial^2\left(\nabla_\alpha w^\alpha\right)+\mu\Delta w^2+\varrho gJ^2+f^2_r(\boldsymbol{w})=0\ ,\tag{50}$$

and

$$\begin{aligned}
&\left(\Lambda\left(\partial_1 w^1+\Gamma^1_{11}w^1\right)+(\Lambda-2\mu)\left(\partial_2 w^2+\Gamma^2_{21}w^1\right)\right)F_{[\mathrm{II}]}{}^1_1\\
&\quad+\left((\Lambda-2\mu)\left(\partial_1 w^1+\Gamma^1_{11}w^1\right)+\Lambda\left(\partial_2 w^2+\Gamma^2_{21}w^1\right)\right)F_{[\mathrm{II}]}{}^2_2+\varrho gJ^3+f^3_r(\boldsymbol{w})=\varrho F_{[\mathrm{II}]}{}^2_2V^2\ ,
\end{aligned}\tag{51}$$

where $\Lambda = 4(\lambda+2\mu)^{-1}\mu(\lambda+\mu)$ and λ and μ are first and second Lamé's parameters respectively.

Assume that the contact area is rough, and thus, one obtains a final governing equation to solve the problem, which is

$$f_r^\beta(\boldsymbol{w}) + \nu_F (w_\alpha w^\alpha)^{-\frac{1}{2}} w^\beta f_r^3(\boldsymbol{w}) = 0 \ , \tag{52}$$

where the coefficient of friction, ν_F, is considered to be an unknown. Now, divide the boundary into sub-boundaries as

$$\begin{aligned} \partial\omega_f &= \{\{0\}\times(-\frac{1}{2}\pi,0)\}\cup\{\{l\}\times(-\frac{1}{2}\pi,0)\} \\ \partial\omega_{T_0} &= \{[0,l]\times\{-\frac{1}{2}\pi\}\} \ , \\ \partial\omega_{T_{\text{max}}} &= \{\{[0,l]\times\{0\}\} \ , \end{aligned}$$

and assert that the boundary conditions are

$$\begin{aligned} \left[(\Lambda-2\mu)\left(\partial_1 w^1+\Gamma_{11}^1 w^1\right)+\Lambda\left(\partial_2 w^2+\Gamma_{21}^2 w^1\right)\right]|_{\partial\omega_{T_0}} &= \tau_0 \text{ (traction)} \ , \\ \left[(\Lambda-2\mu)\left(\partial_1 w^1+\Gamma_{11}^1 w^1\right)+\Lambda\left(\partial_2 w^2+\Gamma_{21}^2 w^1\right)\right]|_{\partial\omega_{T_{\text{max}}}} &= \tau_{\text{max}} \text{ (traction)} \ , && (53) \\ \left[\Lambda\left(\partial_1 w^1+\Gamma_{11}^1 w^1\right)+(\Lambda-2\mu)\left(\partial_2 w^2+\Gamma_{21}^2 w^1\right)\right]|_{\partial\omega_f} &= 0 \text{ (zero-Robin)} \ , && (54) \\ \left[(\psi_1)^2\partial_2 w^1+(\psi_2)^2\partial_1 w^2\right]|_{\partial\omega} &= 0 \text{ (zero-Robin)} \ , && (55) \end{aligned}$$

where $\tau_{\text{max}} > \tau_0$ are positive constants.

Now, assume $\boldsymbol{u} = (\psi_1,\psi_2)$ and $(w_\gamma w^\gamma)^{-\frac{1}{2}} w^\alpha u_\alpha = (u_\gamma u^\gamma)^{-\frac{1}{2}} u^\alpha u_\alpha$ in the compatibility condition to obtain

$$\begin{aligned} \sqrt{2}\,\nu_F \int_\omega f_r^3(\boldsymbol{w})\psi_1\psi_2 \, dx^1 dx^2 =& (\tau_{\text{max}}-\tau_0)\int_0^l \psi_1\psi_2 \, dx^1 \\ &+ \varrho \int_\omega \left(gJ^\alpha\psi_\alpha + (\psi_1)^{-1}\Gamma_{21}^2 V^2\right)\psi_1\psi_2 \, dx^1 dx^2 \ . \end{aligned} \tag{56}$$

Note that for our experiments we keep the values $\tau_0 = 1$, $r_0 = 1$ and $g = 9.81$ fixed. Also, we employ the second-order-accurate finite-difference method in conjunction with Newton's method for nonlinear systems. As we are dealing with curvilinear coordinates, there is a inherit grid dependence, and it is approximately $\Delta x^2 \leq \psi_0 \Delta x^1$, $\forall\ \psi_0 \in \{(\psi_2)^{-1}\psi_1 \mid x^1 \in [0,l]\}$, where Δx^β is a small increment in x^β direction. For our purposes we use $\Delta x^2 = \frac{1}{N-1}$ and $\psi_0 = (\psi_2)^{-1}\psi_1|_{x^1=\frac{1}{2}l}$, where $N = 250$. Finally, we must define a terminating condition. For this, we choose to terminate our iterating process once the condition $|1-(\nu_{Fm})^{-1}\nu_{Fm+1}| < 10^{-8}$ is satisfied, where ν_{Fm} is the m^{th} iterative solution for the coefficient of friction. Further note that to numerically model equation (56) we use the prismoidal formula [7]. Furthermore, as this is a pure-traction problem, the solution is highly unstable. By construction we have $w^2|_{\partial\omega_{T_0}} \leq 0$, and thus, whenever this condition is violated we enforce the condition $w^2|_{\partial\omega_{T_0}} = 0$ to keep the solution from diverging out of control.

Finally, let `u1`$= w^1$, `u2`$= w^2$, `du1`$= \delta w^1$, `du2`$= \delta w^2$, `width`$= l/r_0$, `radius`$= 1$, `radius2`$= c/r_0$, `speed`$= V$, `density`$= \varrho/r_0$, `gravity`$= g$, `f`$= \varphi/r_0$, `Stress1`$= 1$, `Stress2`$= \tau_{\text{max}}/\tau_0$, `Youngs`$= E$, `Poisson`$= \nu$, `NN`$= N$ and `Mu`$= \nu_F$. Thus, we find:

```
function BeltFriction
format long
%% Belt-Friction Model
% Overlying elastic membrane on an rigid foundation with a variable Gaussian curvature
% Contact angle is [0,0.5*pi]
% Dynamic friction case
% Coefficient friction is calculated as a part of the solution

%% INITIAL PARIMITERS
width = 1; %  = Width/Radius
radius = 1; % DO NOT CHANGE!
radius2 = 0; % Difference between the maximum and the minimum radius, i.e. 1/curvature
speed = 0.01;
density = 0.01; % = MassDensity/T0
gravity = 9.81;

%% RADIAL FUNCTION
alpha = 2^4;
f = @(u)   radius - alpha*radius2*(u/width-1/2)^4;
f1 = @(u)  -4*(alpha*radius2*(u/width-1/2)^3)/width; % f'
f11 = @(u)  -12*(alpha*radius2*(u/width-1/2)^2)/(width^2); % f''
f111 = @(u)  -24*alpha*radius2*(u/width-1/2)/(width^3); % f'''

Stress1 = 1; % DO NOT CHANGE!
Stress2 = 1.5; % = Tmax/T0

Youngs = 1000; % Young's modulus of the membrane
Poisson = 0.25; % Poisson's' ratio of the overlying body

NN = 250; % Azimuthal grid points
error = 10^(-8); % Terminating error

%% DO NOT CHANGE!
q1 = 2*width((1+f1(0.5*width)^2)^(0.5))/(f(0.5*width)*pi);

m = round(q1*NN-q1+1); % Azimuthal grid points
n = NN; % Azimuthal grid points

dx1 = width/(m-1); % Axial grid spacing
dx2 = 0.5*pi/(n-1); % Azimuthal grid spacing

LL  = Poisson*Youngs/((1+Poisson)*(1-2*Poisson));
MM = 0.5*Youngs/(1+Poisson);

L = 2*LL*MM/(LL+2*MM);
M = MM;

cff = 2*log(Stress2/Stress1)/pi; % Initial guess for coefficient of friction

NNN = NN^2;
errr = NNN*error;
```

```
Idx1 = 1/dx1;
Idx2 = 1/dx2;

Id1 = 0.5*Idx1;
Id11 = Idx1^2;

Id2 = 0.5*Idx2;
Id22 = Idx2^2;

Id12 = Id1*Id2;

ax = zeros(2,2);

u1 = zeros(m,n);
u2 = zeros(m,n);
du1 = zeros(m,n);
du2 = zeros(m,n);
F3 = zeros(m,n);

G1 = zeros(m,1);
G2 = zeros(m,1);

IG1 = zeros(m,1);
IG2 = zeros(m,1);

L111 = zeros(m,1);
L113 = zeros(m,1);

L221 = zeros(m,1);
L223 = zeros(m,1);

AA1d11 = zeros(m,1);
AA1d22 = zeros(m,1);
AA2d12 = zeros(m,1);
AA1d1 = zeros(m,1);
AA2d2 = zeros(m,1);
AA1 = zeros(m,1);

BB1d12 = zeros(m,1);
BB2d11 = zeros(m,1);
BB2d22 = zeros(m,1);
BB1d2 = zeros(m,1);
BB2d1 = zeros(m,1);

CC1d1 = zeros(m,1);
CC2d2 = zeros(m,1);
CC1 = zeros(m,1);

DD1d1 = zeros(m,1);
DD2d2 = zeros(m,1);
```

```
DD1 = zeros(m,1);

V1 = zeros(m,1);
V3 = zeros(m,1);

J1 = zeros(m,n);
J2 = zeros(m,n);
J3 = zeros(m,n);

JMi1B11 = zeros(m,1);
JMi1B12 = zeros(m,1);
JMi1B21 = zeros(m,1);
JMi1B22 = zeros(m,1);

JMinB11 = zeros(m,1);
JMinB12 = zeros(m,1);
JMinB21 = zeros(m,1);
JMinB22 = zeros(m,1);

JIG11 = zeros(m,1);
JIG12 = zeros(m,1);
JIG21 = zeros(m,1);
JIG22 = zeros(m,1);

SS2 = zeros(m,1);

SS1 = zeros(m,n);
SS3 = zeros(m,n);

%% Curvature terms of the membrane
for i = 1:m

    x = dx1*(i-1);

    F = f(x);
    F1 = f1(x);
    F11 = f11(x);
    F111 = f111(x);
    D = 1+F1^2;

    G1(i)  = D^(1/2);
    G2(i)  = F;

    IG1(i) = 1/G1(i);
    IG2(i) = 1/G2(i);

    X1 = F1*F11*IG1(i);
    X11 = (F1*F111 + F11^2)*IG1(i) - ((F1*F11)^2)*IG1(i)^3;

    X3 = - F11*IG1(i)^2;
```

```
Y1 = F1;
Y11 = F11;

Y3 = IG1(i);

L111(i) = X1*IG1(i);
L113(i) = X3*IG1(i);

L1111 = X11*IG1(i)-L111(i)^2;

L221(i) = Y1*IG2(i);
L223(i) = Y3*IG2(i);

L2211 = Y11*IG2(i)-L221(i)^2;

AA1d11(i) = L+2*M;
AA1d22(i) = M*(G1(i)*IG2(i))^2;
AA2d12(i) = L+M;
AA1d1(i)  = (L+2*M)*(L111(i)+L221(i));
AA2d2(i)  = - 2*M*L221(i);
AA1(i)    = (L+2*M)*L1111 + L*L2211 + 2*M*L221(i)*(L111(i)-L221(i)) ;

BB1d12(i) = L+M;
BB2d11(i) = M*(G2(i)*IG1(i))^2;
BB2d22(i) = L+2*M;
BB1d2(i)  = (L+2*M)*L111(i) + (L+3*M)*L221(i);
BB2d1(i)  = M*(3*L221(i)-L111(i))*(G2(i)*IG1(i))^2;

CC1d1(i) = L+2*M;
CC2d2(i) = L;
CC1(i)   = (L+2*M)*L111(i) + L*L221(i);

DD1d1(i) = L;
DD2d2(i) = L+2*M;
DD1(i)   = L*L111(i) + (L+2*M)*L221(i);

V1(i) = - density*L221(i)*(IG1(i)*speed)^2;
V3(i) = - density*L223(i)*speed^2;

ax(1,1) = 3*Id2*G1(i)^2;
ax(1,2) = 0;
ax(2,1) = - L*L111(i) - (L+2*M)*L221(i);
ax(2,2) = 3*Id2*(L+2*M);

J = inv(ax);

JMi1B11(i) = J(1,1);
JMi1B12(i) = J(1,2);
JMi1B21(i) = J(2,1);
JMi1B22(i) = J(2,2);
```

```
    ax(1,1) = -3*Id2*G1(i)^2;
    ax(1,2) = 0;
    ax(2,1) = - L*L111(i) - (L+2*M)*L221(i);
    ax(2,2) = -3*Id2*(L+2*M);

    J = inv(ax);

    JMinB11(i) = J(1,1);
    JMinB12(i) = J(1,2);
    JMinB21(i) = J(2,1);
    JMinB22(i) = J(2,2);

    SS2(i) = G1(i)*G2(i);

    for j = 1:n

        J1(i,j) = - density*gravity*sin(dx2*(j-1))*F1*IG1(i)^2;
        J2(i,j) = - density*gravity*cos(dx2*(j-1))*IG2(i);
        J3(i,j) = - density*gravity*sin(dx2*(j-1))*IG1(i);

        SS1(i,j) =  ((J1(i,j)-V1(i))*G1(i) + J2(i,j)*G2(i))*G1(i)*G2(i);
        SS3(i,j) = - (J3(i,j)-V3(i))*G1(i)*G2(i);

    end

    ax(1,1) = 2*AA1d11(i)*Id11 + 2*AA1d22(i)*Id22 - AA1(i);
    ax(1,2) = 0;
    ax(2,1) = 0;
    ax(2,2) = 2*BB2d11(i)*Id11 + 2*BB2d22(i)*Id22;

    J = inv(ax);

    JIG11(i) = J(1,1);
    JIG12(i) = J(1,2);
    JIG21(i) = J(2,1);
    JIG22(i) = J(2,2);

end

ax(1,1) = 3*Id1*(L+2*M) - (L+2*M)*L111(1) - L*L221(1);
ax(1,2) = 0;
ax(2,1) = 0;
ax(2,2) = 3*Id1*G2(1)^2;

J = inv(ax);

JM1jB11 = J(1,1);
JM1jB12 = J(1,2);
JM1jB21 = J(2,1);
JM1jB22 = J(2,2);
```

```
ax(1,1) = -3*Id1*(L+2*M) - (L+2*M)*L111(m) - L*L221(m);
ax(1,2) = 0;
ax(2,1) = 0;
ax(2,2) = -3*Id1*G2(m)^2;

J = inv(ax);

JMmjB11 = J(1,1);
JMmjB12 = J(1,2);
JMmjB21 = J(2,1);
JMmjB22 = J(2,2);

ax(1,1) = 3*Id1*(L+2*M) - (L+2*M)*L111(1) - L*L221(1);
ax(1,2) = - 3*Id2*L;
ax(2,1) = 3*Id1*L - L*L111(1) - (L+2*M)*L221(1);
ax(2,2) = - 3*Id2*(L+2*M);

J = inv(ax);

JM1nB11 = J(1,1);
JM1nB12 = J(1,2);
JM1nB21 = J(2,1);
JM1nB22 = J(2,2);

ax(1,1) = -3*Id1*(L+2*M) - (L+2*M)*L111(m) - L*L221(m);
ax(1,2) = -3*Id2*L;
ax(2,1) = -3*Id1*L - L*L111(m) - (L+2*M)*L221(m);
ax(2,2) = -3*Id2*(L+2*M);

J = inv(ax);

JMmnB11 = J(1,1);
JMmnB12 = J(1,2);
JMmnB21 = J(2,1);
JMmnB22 = J(2,2);

SS  = (Stress2-Stress1)*sum(SS2)*dx1*(1-1/m) + sum(sum(SS1))*dx1*dx2*(1-1/m)*(1-1/n);
S3  = sum(sum(SS3))*dx1*dx2*(1-1/m)*(1-1/n);

JMi1B11X = 1/(3*Id2);

p1 = 2;

%% Initial guess
while errr < p1

    pp = norm(u1,2) + norm(u2,2);

    for kk = 1:NNN

        %% Corners
```

```
u1d1 = (4*u1(2,n)-u1(3,n))*Id1;
u2d2 = -(4*u2(1,n-1)-u2(1,n-2))*Id2;

Xu1 = (L+2*M)*u1d1 + L*u2d2;
Xu2 = L*u1d1 + (L+2*M)*u2d2 - Stress2;

u1(1,n) = JM1nB11*Xu1 + JM1nB12*Xu2;
u2(1,n) = JM1nB21*Xu1 + JM1nB22*Xu2;

u1d1 = -(4*u1(m-1,n)-u1(m-2,n))*Id1;
u2d2 = -(4*u2(m,n-1)-u2(m,n-2))*Id2;

Xu1 = (L+2*M)*u1d1 + L*u2d2;
Xu2 = L*u1d1 + (L+2*M)*u2d2 - Stress2;

u1(m,n) = JMmnB11*Xu1 + JMmnB12*Xu2;
u2(m,n) = JMmnB21*Xu1 + JMmnB22*Xu2;

u1d2 = (4*u1(1,2)-u1(1,3))*Id2;
u1(1,1) = JMi1B11X*u1d2;

u1d2 = (4*u1(m,2)-u1(m,3))*Id2;
u1(m,1) = JMi1B11X*u1d2;

%% Stressed boundaries
for i = 2:m-1

    u1d2 = (4*u1(i,2)-u1(i,3))*Id2;

    u1(i,1) = JMi1B11X*u1d2;

    u1d1 = (u1(i+1,n)-u1(i-1,n))*Id1;
    u1d2 = -(4*u1(i,n-1)-u1(i,n-2))*Id2;

    u2d1 = (u2(i+1,n)-u2(i-1,n))*Id1;
    u2d2 = -(4*u2(i,n-1)-u2(i,n-2))*Id2;

    Xu1 = (G2(i)^2)*u2d1 + (G1(i)^2)*u1d2;
    Xu2 = L*u1d1 + (L+2*M)*u2d2 - Stress2;

    u1(i,n) = JMinB11(i)*Xu1 + JMinB12(i)*Xu2;
    u2(i,n) = JMinB21(i)*Xu1 + JMinB22(i)*Xu2;

end

%% Stress-free boundaries
for j = 2:n-1

    u1d1 = (4*u1(2,j)-u1(3,j))*Id1;
    u1d2 = (u1(1,j+1)-u1(1,j-1))*Id2;
```

```matlab
    u2d1 = (4*u2(2,j)-u2(3,j))*Id1;
    u2d2 = (u2(1,j+1)-u2(1,j-1))*Id2;

    Xu1 = (L+2*M)*u1d1 + L*u2d2;
    Xu2 = (G2(1)^2)*u2d1 + (G1(1)^2)*u1d2;

    u1(1,j) = JM1jB11*Xu1 + JM1jB12*Xu2;
    u2(1,j) = JM1jB21*Xu1 + JM1jB22*Xu2;

    u1d1 = -(4*u1(m-1,j)-u1(m-2,j))*Id1;
    u1d2 = (u1(m,j+1)-u1(m,j-1))*Id2;

    u2d1 = -(4*u2(m-1,j)-u2(m-2,j))*Id1;
    u2d2 = (u2(m,j+1)-u2(m,j-1))*Id2;

    Xu1 = (L+2*M)*u1d1 + L*u2d2;
    Xu2 = (G2(m)^2)*u2d1 + (G1(m)^2)*u1d2;

    u1(m,j) = JMmjB11*Xu1 + JMmjB12*Xu2;
    u2(m,j) = JMmjB21*Xu1 + JMmjB22*Xu2;

end

%% Governing equations of the membrane
for i = 2:m-1

    for j = 2:n-1

        u1d11 = (u1(i+1,j)+u1(i-1,j))*Id11;
        u1d22 = (u1(i,j+1)+u1(i,j-1))*Id22;

        u2d11 = (u2(i+1,j)+u2(i-1,j))*Id11;
        u2d22 = (u2(i,j+1)+u2(i,j-1))*Id22;

        u1d12 = (u1(i+1,j+1)-u1(i+1,j-1)-u1(i-1,j+1)+u1(i-1,j-1))*Id12;
        u2d12 = (u2(i+1,j+1)-u2(i+1,j-1)-u2(i-1,j+1)+u2(i-1,j-1))*Id12;

        u1d1 = (u1(i+1,j)-u1(i-1,j))*Id1;
        u1d2 = (u1(i,j+1)-u1(i,j-1))*Id2;

        u2d1 = (u2(i+1,j)-u2(i-1,j))*Id1;
        u2d2 = (u2(i,j+1)-u2(i,j-1))*Id2;

        Xu1 = AA1d11(i)*u1d11 + AA1d22(i)*u1d22 + AA2d12(i)*u2d12 ...
            + AA1d1(i)*u1d1 + AA2d2(i)*u2d2;
        Xu2 = BB1d12(i)*u1d12 + BB2d11(i)*u2d11 + BB2d22(i)*u2d22 ...
            + BB1d2(i)*u1d2 + BB2d1(i)*u2d1;

        u1(i,j) = JIG11(i)*Xu1 + JIG12(i)*Xu2;
        u2(i,j) = JIG21(i)*Xu1 + JIG22(i)*Xu2;
```

```
            end
        end
    end

    %% Terminating condition
    ppp = norm(u1,2) + norm(u2,2);
    p1 = abs(1-ppp/pp);

end

p1 = 2;

%% Main code
while errr < p1

    pp = cff;

    for kk = 1:NNN

        %% Corners
        u1d1 = (4*u1(2,n)-u1(3,n))*Id1;
        u2d2 = -(4*u2(1,n-1)-u2(1,n-2))*Id2;

        Xu1 = (L+2*M)*u1d1 + L*u2d2; % equation (54)
        Xu2 = L*u1d1 + (L+2*M)*u2d2 - Stress2; % equation (53)

        u1(1,n) = JM1nB11*Xu1 + JM1nB12*Xu2;
        u2(1,n) = JM1nB21*Xu1 + JM1nB22*Xu2;

        u1d1 = -(4*u1(m-1,n)-u1(m-2,n))*Id1;
        u2d2 = -(4*u2(m,n-1)-u2(m,n-2))*Id2;

        Xu1 = (L+2*M)*u1d1 + L*u2d2; % equation (54)
        Xu2 = L*u1d1 + (L+2*M)*u2d2 - Stress2; % equation (53)

        u1(m,n) = JMmnB11*Xu1 + JMmnB12*Xu2;
        u2(m,n) = JMmnB21*Xu1 + JMmnB22*Xu2;

        u1d2 = (4*u1(1,2)-u1(1,3))*Id2;
        u1(1,1) = JMi1B11X*u1d2;

        u1d2 = (4*u1(m,2)-u1(m,3))*Id2;
        u1(m,1) = JMi1B11X*u1d2;

        %% Stressed boundaries
        for i = 2:m-1

            u1d2 = (4*u1(i,2)-u1(i,3))*Id2;

            u1(i,1) = JMi1B11X*u1d2;
```

```
    u1d1 = (u1(i+1,n)-u1(i-1,n))*Id1;
    u1d2 = -(4*u1(i,n-1)-u1(i,n-2))*Id2;

    u2d1 = (u2(i+1,n)-u2(i-1,n))*Id1;
    u2d2 = -(4*u2(i,n-1)-u2(i,n-2))*Id2;

    Xu1 = (G2(i)^2)*u2d1 + (G1(i)^2)*u1d2; % equation (55)
    Xu2 = L*u1d1 + (L+2*M)*u2d2 - Stress2; % equation (53)

    u1(i,n) = JMinB11(i)*Xu1 + JMinB12(i)*Xu2;
    u2(i,n) = JMinB21(i)*Xu1 + JMinB22(i)*Xu2;

end

%% Stress-free boundaries
for j = 2:n-1

    u1d1 = (4*u1(2,j)-u1(3,j))*Id1;
    u1d2 = (u1(1,j+1)-u1(1,j-1))*Id2;

    u2d1 = (4*u2(2,j)-u2(3,j))*Id1;
    u2d2 = (u2(1,j+1)-u2(1,j-1))*Id2;

    Xu1 = (L+2*M)*u1d1 + L*u2d2; % equation (54)
    Xu2 = (G2(1)^2)*u2d1 + (G1(1)^2)*u1d2; % equation (55)

    u1(1,j) = JM1jB11*Xu1 + JM1jB12*Xu2;
    u2(1,j) = JM1jB21*Xu1 + JM1jB22*Xu2;

    u1d1 = -(4*u1(m-1,j)-u1(m-2,j))*Id1;
    u1d2 = (u1(m,j+1)-u1(m,j-1))*Id2;

    u2d1 = -(4*u2(m-1,j)-u2(m-2,j))*Id1;
    u2d2 = (u2(m,j+1)-u2(m,j-1))*Id2;

    Xu1 = (L+2*M)*u1d1 + L*u2d2; % equation (54)
    Xu2 = (G2(m)^2)*u2d1 + (G1(m)^2)*u1d2; % equation (55)

    u1(m,j) = JMmjB11*Xu1 + JMmjB12*Xu2;
    u2(m,j) = JMmjB21*Xu1 + JMmjB22*Xu2;

end

%% Governing equations of the membrane % equations (49), (50), (51) and (52)
for i = 2:m-1

    for j = 2:n-1

        uu = (G1(i)*u1(i,j))^2+(G2(i)*u2(i,j))^2;
        Iu = 1/sqrt(uu);
```

```
if uu == 0
    Iu = 0;
end

xu1 = u1(i,j)*Iu;
xu2 = u2(i,j)*Iu;

u1d11 = (u1(i+1,j)-2*u1(i,j)+u1(i-1,j))*Id11;
u1d22 = (u1(i,j+1)-2*u1(i,j)+u1(i,j-1))*Id22;

u2d11 = (u2(i+1,j)-2*u2(i,j)+u2(i-1,j))*Id11;
u2d22 = (u2(i,j+1)-2*u2(i,j)+u2(i,j-1))*Id22;

u1d12 = (u1(i+1,j+1)-u1(i+1,j-1)-u1(i-1,j+1)+u1(i-1,j-1))*Id12;
u2d12 = (u2(i+1,j+1)-u2(i+1,j-1)-u2(i-1,j+1)+u2(i-1,j-1))*Id12;

u1d1 = (u1(i+1,j)-u1(i-1,j))*Id1;
u1d2 = (u1(i,j+1)-u1(i,j-1))*Id2;

u2d1 = (u2(i+1,j)-u2(i-1,j))*Id1;
u2d2 = (u2(i,j+1)-u2(i,j-1))*Id2;

du1d11 = (du1(i+1,j)+du1(i-1,j))*Id11;
du1d22 = (du1(i,j+1)+du1(i,j-1))*Id22;

du2d11 = (du2(i+1,j)+du2(i-1,j))*Id11;
du2d22 = (du2(i,j+1)+du2(i,j-1))*Id22;

du1d12 = (du1(i+1,j+1)-du1(i+1,j-1)-du1(i-1,j+1)+du1(i-1,j-1))*Id12;
du2d12 = (du2(i+1,j+1)-du2(i+1,j-1)-du2(i-1,j+1)+du2(i-1,j-1))*Id12;

du1d1 = (du1(i+1,j)-du1(i-1,j))*Id1;
du1d2 = (du1(i,j+1)-du1(i,j-1))*Id2;

du2d1 = (du2(i+1,j)-du2(i-1,j))*Id1;
du2d2 = (du2(i,j+1)-du2(i,j-1))*Id2;

Tu1 = AA1d11(i)*u1d11 + AA1d22(i)*u1d22 + AA2d12(i)*u2d12 ...
    + AA1d1(i)*u1d1 + AA2d2(i)*u2d2;
Tu2 = BB1d12(i)*u1d12 + BB2d11(i)*u2d11 + BB2d22(i)*u2d22 ...
    + BB1d2(i)*u1d2 + BB2d1(i)*u2d1;

Tdu1 = AA1d11(i)*du1d11 + AA1d22(i)*du1d22 + AA2d12(i)*du2d12 ...
    + AA1d1(i)*du1d1 + AA2d2(i)*du2d2;
Tdu2 = BB1d12(i)*du1d12 + BB2d11(i)*du2d11 + BB2d22(i)*du2d22 ...
    + BB1d2(i)*du1d2 + BB2d1(i)*du2d1;

T1u = CC1d1(i)*u1d1 + CC2d2(i)*u2d2 + CC1(i)*u1(i,j);
T2u = DD1d1(i)*u1d1 + DD2d2(i)*u2d2 + DD1(i)*u1(i,j);
Fu  = L113(i)*T1u + L223(i)*T2u + SS3(i,j);
```

```
            T1du = CC1d1(i)*du1d1 + CC2d2(i)*du2d2;
            T2du = DD1d1(i)*du1d1 + DD2d2(i)*du2d2;
            Fdu  = L113(i)*T1du + L223(i)*T2du;

            Xdu1 = Tdu1 - cff*Fdu*xu1 + Tu1 - cff*Fu*xu1 + J1(i,j) - V1(i);
            Xdu2 = Tdu2 - cff*Fdu*xu2 + Tu2 - cff*Fu*xu2 + J2(i,j);

            ax(1,1) = 2*AA1d11(i)*Id11 + 2*AA1d22(i)*Id22 - AA1(i) + cff*Fu*Iu ...
                - cff*Fu*xu1*xu1*Iu + cff*(CC1(i)*L113(i) + DD1(i)*L223(i))*xu1;
            ax(1,2) = - cff*Fu*xu1*xu2*Iu;
            ax(2,1) = - cff*Fu*xu1*xu2*Iu;
            ax(2,2) = 2*BB2d11(i)*Id11 + 2*BB2d22(i)*Id22          ...
                + cff*Fu*Iu - cff*Fu*xu2*xu2*Iu;

            J = inv(ax);

            JM11 = J(1,1);
            JM12 = J(1,2);
            JM21 = J(2,1);
            JM22 = J(2,2);

            du1(i,j) = JM11*Xdu1 + JM12*Xdu2;
            du2(i,j) = JM21*Xdu1 + JM22*Xdu2;

        end
    end

    %% Newton's method
    u1(:,:) = u1(:,:) + du1(:,:);
    u2(:,:) = u2(:,:) + du2(:,:);

    %% Coefficient of friction % equation (56)
    for i = 2:m-1

        u1d1 = (u1(i+1,1)-u1(i-1,1))*Id1;
        u2d2 = (-3*u2(i,1)+4*u2(i,2)-u2(i,3))*Id2;

        T1u = CC1d1(i)*u1d1 + CC2d2(i)*u2d2 + CC1(i)*u1(i,1);
        T2u = DD1d1(i)*u1d1 + DD2d2(i)*u2d2 + DD1(i)*u1(i,1);
        Fu  = L113(i)*T1u + L223(i)*T2u;
        F3(i,1) = Fu*dx1*dx2*(1-1/m)*(1-1/n);

        u1d1 = (u1(i+1,n)-u1(i-1,n))*Id1;
        u2d2 = (3*u2(i,n)-4*u2(i,n-1)+u2(i,n-2))*Id2;

        T1u = CC1d1(i)*u1d1 + CC2d2(i)*u2d2 + CC1(i)*u1(i,n);
        T2u = DD1d1(i)*u1d1 + DD2d2(i)*u2d2 + DD1(i)*u1(i,n);
        Fu  = L113(i)*T1u + L223(i)*T2u;
        F3(i,n) = Fu*dx1*dx2*(1-1/m)*(1-1/n);

        for j = 2:n-1
```

```
        u1d1 = (u1(i+1,j)-u1(i-1,j))*Id1;
        u2d2 = (u2(i,j+1)-u2(i,j-1))*Id2;

        T1u = CC1d1(i)*u1d1 + CC2d2(i)*u2d2 + CC1(i)*u1(i,j);
        T2u = DD1d1(i)*u1d1 + DD2d2(i)*u2d2 + DD1(i)*u1(i,j);
        Fu  = L113(i)*T1u + L223(i)*T2u;
        F3(i,j) = Fu*dx1*dx2*(1-1/m)*(1-1/n);

    end
end

for j = 2:n-1

    u1d1 = (-3*u1(1,j)+4*u1(2,j)-u1(3,j))*Id1;
    u2d2 = (u2(1,j+1)-u2(1,j-1))*Id2;

    T1u = CC1d1(1)*u1d1 + CC2d2(1)*u2d2 + CC1(1)*u1(1,j);
    T2u = DD1d1(1)*u1d1 + DD2d2(1)*u2d2 + DD1(1)*u1(1,j);
    Fu  = L113(1)*T1u + L223(1)*T2u;
    F3(1,j) = Fu*dx1*dx2*(1-1/m)*(1-1/n);

    u1d1 = (3*u1(m,j)-4*u1(m-1,j)+u1(m-2,j))*Id1;
    u2d2 = (u2(m,j+1)-u2(m,j-1))*Id2;

    T1u = CC1d1(m)*u1d1 + CC2d2(m)*u2d2 + CC1(m)*u1(m,j);
    T2u = DD1d1(m)*u1d1 + DD2d2(m)*u2d2 + DD1(i)*u1(m,j);
    Fu  = L113(m)*T1u + L223(m)*T2u;
    F3(m,j) = Fu*dx1*dx2*(1-1/m)*(1-1/n);

end

u1d1 = (-3*u1(1,1)+4*u1(2,1)-u1(3,1))*Id1;
u2d2 = (-3*u2(1,1)+4*u2(1,2)-u2(1,3))*Id2;

T1u = CC1d1(1)*u1d1 + CC2d2(1)*u2d2 + CC1(1)*u1(1,1);
T2u = DD1d1(1)*u1d1 + DD2d2(1)*u2d2 + DD1(1)*u1(1,1);
Fu  = L113(1)*T1u + L223(1)*T2u;
F3(1,1) = Fu*dx1*dx2*(1-1/m)*(1-1/n);

u1d1 = (-3*u1(1,n)+4*u1(2,n)-u1(3,n))*Id1;
u2d2 = (3*u2(1,n)-4*u2(1,n-1)+u2(1,n-2))*Id2;

T1u = CC1d1(1)*u1d1 + CC2d2(1)*u2d2 + CC1(1)*u1(1,n);
T2u = DD1d1(1)*u1d1 + DD2d2(1)*u2d2 + DD1(1)*u1(1,n);
Fu  = L113(1)*T1u + L223(1)*T2u;
F3(1,n) = Fu*dx1*dx2*(1-1/m)*(1-1/n);

u1d1 = (3*u1(m,1)-4*u1(m-1,1)+u1(m-2,1))*Id1;
u2d2 = (-3*u2(m,1)+4*u2(m,2)-u2(m,3))*Id2;
```

```
        T1u = CC1d1(m)*u1d1 + CC2d2(m)*u2d2 + CC1(m)*u1(m,1);
        T2u = DD1d1(m)*u1d1 + DD2d2(m)*u2d2 + DD1(m)*u1(m,1);
        Fu  = L113(m)*T1u + L223(m)*T2u;
        F3(m,1) = Fu*dx1*dx2*(1-1/m)*(1-1/n);

        u1d1 = (3*u1(m,n)-4*u1(m-1,n)+u1(m-2,n))*Id1;
        u2d2 = (3*u2(m,n)-4*u2(m,n-1)+u2(m,n-2))*Id2;

        T1u = CC1d1(m)*u1d1 + CC2d2(m)*u2d2 + CC1(m)*u1(m,n);
        T2u = DD1d1(m)*u1d1 + DD2d2(m)*u2d2 + DD1(m)*u1(m,n);
        Fu  = L113(m)*T1u + L223(m)*T2u;
        F3(m,n) = Fu*dx1*dx2*(1-1/m)*(1-1/n);

        cff = SS/(sum(sum(F3)) + S3); % Coefficient of friction


    end

    %% Terminating condition
    p1 = abs(1-cff/pp);

end

%% Save data file
filename = 'BeltFriction.mat';
save(filename)
```

5 Shell-Membrane Model

In this chapter we present a numerical code for an example of a shell-membrane on an elastic foundation with static friction (see section 6.6 of Jayawardana [4]) implemented in Matlab, i.e. `FrictionCode.m`.

To conduct numerical experiments assume a shell-membrane with a thickness h, supported by an elastic foundation, where the unstrained configuration of the foundation is an annular cylindrical which is characterised by the diffeomorphism $\bar{X}(x,\theta,r) = (x, r\sin(\theta),\ r\cos(\theta))_{\mathsf{E}}$, where $(x^1,x^2,x^3) = (x,\theta,r)$, $x \in (-L,L)$, $\theta \in (-\pi,\pi]$ and $r \in (a_0,a)$, and assume that the contact region is defined by $x \in (-\ell,\ell)$, $\theta \in (-\frac{1}{2}\pi,0)$, where $0 < \ell < L$. Let the sufficiently smooth field $\boldsymbol{u} = (u^1(x,\theta,r), u^2(x,\theta,r), u^3(x,\theta,r))$ be the displacement field of the foundation. With some calculations one finds that the metric tensor is $\boldsymbol{g} = \mathrm{diag}(1,r^2,1)$ and the covariant derivatives are

$$\begin{aligned}
&\bar{\nabla}_1 u^1 = \partial_1 u^1\ , && \bar{\nabla}_1 u^2 = \partial_1 u^2\ , && \bar{\nabla}_1 u^3 = \partial_1 u^3\ ,\\
&\bar{\nabla}_2 u^1 = \partial_2 u^1\ , && \bar{\nabla}_2 u^2 = \partial_2 u^2 + r^{-1}u^3\ , && \bar{\nabla}_2 u^3 = \partial_2 u^3 - r u^2\ ,\\
&\bar{\nabla}_3 u^1 = \partial_3 u^1\ , && \bar{\nabla}_3 u^2 = \partial_3 u^2 + r^{-1}u^2\ , && \bar{\nabla}_3 u^3 = \partial_3 u^3\ .
\end{aligned}$$

With further calculations, one can express the governing equations of the foundation as

$$(\bar{\lambda}+\bar{\mu})\partial^1(\bar{\nabla}_i u^i) + \bar{\mu}\bar{\Delta}u^1 = 0\ , \tag{57}$$

$$(\bar{\lambda}+\bar{\mu})\partial^2(\bar{\nabla}_i u^i) + \bar{\mu}\bar{\Delta}u^2 = 0\ , \tag{58}$$

$$(\bar{\lambda}+\bar{\mu})\partial^3(\bar{\nabla}_i u^i) + \bar{\mu}\bar{\Delta}u^3 = 0\ . \tag{59}$$

The boundary of the foundation can be decomposed into sub-boundaries as

$$\begin{aligned}
\partial\Omega &= \bar{\omega} \cup \partial\Omega_0 \cup \partial\Omega_f\ ,\\
\omega &= \{a\} \times (-\frac{1}{2}\pi, 0) \times (-\ell,\ell)\ ,\\
\partial\Omega_0 &= \{\{a_0\} \times (-\pi,\pi] \times [-L,L]\} \cup \{(a_0,a] \times (-\pi,\pi] \times \{\{-L\}\cup\{L\}\}\}\ ,\\
\partial\Omega_f &= \{\{a\} \times (-\pi,\pi] \times (-L,L)\} \setminus \bar{\omega}\ .
\end{aligned}$$

Thus, one can express the boundary conditions of the foundation as

$$\boldsymbol{u}|_{\partial\Omega_0} = \boldsymbol{0} \text{ (zero-Dirichlet)}\ ,$$

$$\left[\partial_3 u^1 + \partial_1 u^3\right]|_{\partial\Omega_f} = 0 \text{ (zero-Robin)}\ , \tag{60}$$

$$\left[r^2\partial_3 u^2 + \partial_2 u^3\right]|_{\partial\Omega_f} = 0 \text{ (zero-Robin)}\ , \tag{61}$$

$$\left[\bar{\lambda}(\partial_1 u^1 + \partial_2 u^2 + r^{-1}u^3) + (\bar{\lambda}+2\bar{\mu})\partial_3 u^3\right]|_{\partial\Omega_f} = 0 \text{ (zero-Robin)}\ . \tag{62}$$

Let $\boldsymbol{u}|_\omega = (u^1(x,\theta,a), u^2(x,\theta,a), u^3(x,\theta,a))$ be the displacement field of the shell-membrane. With some calculations, one finds that the first fundamental form tensor is $\boldsymbol{F}_{[\mathrm{I}]} = \mathrm{diag}(1,a^2)$ and the covariant derivatives are

$$\begin{aligned}
&\nabla_1 u^1 = \partial_1 u^1\ , && \nabla_1 u^2 = \partial_1 u^2\ ,\\
&\nabla_2 u^1 = \partial_2 u^1\ , && \nabla_2 u^2 = \partial_2 u^2\ .
\end{aligned}$$

With further calculations one can express the governing equations of the shell-membrane as:

If $[2\nu_F u^3 + (u^1u^1 + r^2u^2u^2)^{\frac{1}{2}}]|_\omega < 0$, then

$$(\Lambda-\mu)\partial^1(\nabla_\alpha u^\alpha) + \mu\Delta u^1 + \frac{1}{a}(\Lambda-2\mu)\partial^1 u^3 - \frac{1}{h}\mathrm{Tr}(T_3^1(\boldsymbol{u})) = 0\ , \tag{63}$$

$$(\Lambda-\mu)\partial^2(\nabla_\alpha u^\alpha) + \mu\Delta u^2 + \frac{1}{a}\Lambda\partial^2 u^3 - \frac{1}{h}\mathrm{Tr}(T_3^2(\boldsymbol{u})) = 0\ , \tag{64}$$

$$(\Lambda-2\mu)\partial_1 u^1 + \Lambda\left(\partial_2 u^2 + \frac{1}{a}u^3\right) + \frac{a}{h}\mathrm{Tr}(T_3^3(\boldsymbol{u})) = 0\ , \tag{65}$$

where $\Lambda = 2\lambda\mu(\lambda+2\mu)^{-1}$;

If $[2\nu_F u^3 + (u^1u^1 + r^2u^2u^2)^{\frac{1}{2}}]|_\omega = 0$, then

$$(\Lambda-\mu)\partial^1(\nabla_\alpha u^\alpha) + \mu\Delta u^1 - \frac{(\Lambda-2\mu)}{2a\nu_F}\partial^1(u^1u^1+a^2u^2u^2)^{\frac{1}{2}} - \frac{1}{h}\mathrm{Tr}(T_3^1(\boldsymbol{u})) - \frac{(\Lambda-\mu)}{4a\nu_F^2}u^1 - \frac{(\bar{\lambda}+\bar{\mu})}{4h\nu_F^2}u^1$$
$$+\frac{1}{2\nu_F}\frac{u^1}{(u^1u^1+a^2u^2u^2)^{\frac{1}{2}}}\Big((\Lambda-2\mu)\partial_1u^1 + \Lambda\partial_2u^2 + \frac{a}{h}\left(\bar{\lambda}(\partial_1u^1+\partial_2u^2) + (\bar{\lambda}+2\bar{\mu})\partial_3u^3\right)\Big) = 0\ , \tag{66}$$

$$(\Lambda-\mu)\partial^2(\nabla_\alpha u^\alpha) + \mu\Delta u^2 - \frac{\Lambda}{2a\nu_F}\partial^2(u^1u^1+a^2u^2u^2)^{\frac{1}{2}} - \frac{1}{h}\mathrm{Tr}(T_3^2(\boldsymbol{u})) - \frac{(\Lambda-\mu)}{4a\nu_F^2}u^2 - \frac{(\bar{\lambda}+\bar{\mu})}{4h\nu_F^2}u^2$$
$$+\frac{1}{2\nu_F}\frac{u^2}{(u^1u^1+a^2u^2u^2)^{\frac{1}{2}}}\Big((\Lambda-2\mu)\partial_1u^1 + \Lambda\partial_2u^2 + \frac{a}{h}\left(\bar{\lambda}(\partial_1u^1+\partial_2u^2) + (\bar{\lambda}+2\bar{\mu})\partial_3u^3\right)\Big) = 0\ . \tag{67}$$

The boundary of the shell-membrane can be decomposed into sub-boundaries as

$$\begin{aligned}\partial\omega &= \partial\omega_{T_0} \cup \partial\omega_{T_{\text{max}}} \cup \partial\omega_f\ ,\\ \partial\omega_{T_0} &= [-\ell,\ell]\times\{-\frac{1}{2}\pi\}\ ,\\ \partial\omega_{T_{\text{max}}} &= [-\ell,\ell]\times\{0\}\ ,\\ \partial\omega_f &= \{\{-\ell\}\cup\{\ell\}\}\cup(-\frac{1}{2}\pi,0)\ .\end{aligned}$$

Thus, one can express the boundary conditions of the shell-membranes as

$$\left[(\Lambda-2\mu)\partial_1u^1 + \Lambda(\partial_2u^2 + a^{-1}u^3)\right]|_{\partial\omega_{T_0}} = \tau_0 \text{ (traction)}\ , \tag{68}$$

$$\left[(\Lambda-2\mu)\partial_1u^1 + \Lambda(\partial_2u^2 + a^{-1}u^3)\right]|_{\partial\omega_{T_{\text{max}}}} = \tau_{\text{max}} \text{ (traction)}\ , \tag{69}$$

$$\left[\Lambda\partial_1u^1 + (\Lambda-2\mu)(\partial_2u^2 + a^{-1}u^3)\right]|_{\partial\omega_f} = 0 \text{ (zero-Robin)}\ , \tag{70}$$

$$\left[\partial_2u^1 + a^2\partial_1u^2\right]|_{\partial\omega_f} = 0 \text{ (zero-Robin)}\ . \tag{71}$$

Note that we employ the second-order-accurate finite-difference method in conjunction with Newton's method for nonlinear systems. As we are dealing with curvilinear coordinates, there is a inherit grid dependence, and it is approximately $r\Delta x^2 \leq \Delta x^1$ and $r\Delta x^2 \leq \Delta x^2$, for all $a_0 \leq r \leq a$, where Δx^j is a small increment in x^j direction. For our purposes we use $\Delta x^2 = \frac{1}{N-1}$ and $r = a$, where $N = 250$. Furthermore, we must define a terminating condition. For this we choose to terminate our iterating process given that the condition $|1 - ||\boldsymbol{u}_m||_{\ell^2}^{-1}||\boldsymbol{u}_{m+1}||_{\ell^2}| < 10^{-10}$ is satisfied.

Finally, let `v1`$= u^1$, `v2`$= u^2$, `v3`$= u^3$, `dv1`$= \delta u^1$, `dv2`$= \delta u^2$, `dv3`$= \delta u^3$, `a`$= a$, `b`$= a_0$, `h`$= h$, `H`$= H$, `T0`$= \tau_0$, `Tmax`$= \tau_{\text{max}}$, `E1`$= E$, `E2`$= \bar{E}$, `Nu1`$= \nu$, `Nu2`$= \bar{\nu}$, `NN`$= N$ and `Mu`$= \nu_F$. Thus, we find:

```
function ShellMembrane
format long
%% Shell-Membrane Model
% Overlying elastic membrane on an elastic cylinder: friction case
% Contact angle is [0.5*pi,pi]

%% INITIAL PARIMITERS
NN = 250;
error = 10^(-10);

E1 = 1000; % Young's modules of the membrane
E2 = 100; % Young's modules of the foundation

Nu1 = 0.45; % Poisson's ratio of the membrane
Nu2 = 0; % Poisson's ratio of the foundation

h = 0.001; % Thickness of the membrane
a = 1; % width of the foundation
b = 0.25; % Inner radius

H = 1-b; % Thickness of the foundation

Mu = 1; % Coefficient of friction

T0 = 1; % Applied stress at \theta = 0.5*pi
Tmax = 2; % Applied stress at \theta = pi

LL = Nu1*E1/((1+Nu1)*(1-2*Nu1));
L2 = Nu2*E2/((1+Nu2)*(1-2*Nu2));

M1 = 0.5*E1/(1+Nu1);
M2 = 0.5*E2/(1+Nu2);

L1 = 2*M1*LL/(LL+2*M1);

%% DO NOT CHANGE!
NNN = NN^2;
errr = NNN*error;

IMu = 0.5/Mu;
IMu2 = IMu^2;
Ih = 1/h;

q1 = 0.25*a/(2*pi);
q3 = H/(2*pi);

l = round(q1*NN-q1+1);
m = round(0.25*NN);
n = round(q3*NN-q3+1);

l1 = l;
```

```
l3 = 3*l;
l4 = 4*l;

m1 = m;
m2 = 2*m;
m4 = 4*m;

dx1 = a/(l4-1); % Axial grid spacing
dx2 = 2*pi/(m4-1); % Azimuthal grid spacing
dx3 = H/(n-1); % Radial grid spacing

Idx1 = 1/(2*dx1);
Idx11 = (1/dx1)^2;

Idx2 = 1/(2*dx2);
Idx22 = (1/dx2)^2;

Idx3 = 1/(2*dx3);
Idx33 = (1/dx3)^2;

Idx12 = Idx1*Idx2;
Idx13 = Idx1*Idx3;
Idx23 = Idx2*Idx3;

v1 = zeros(l4,m4,n); % Axial displacement of the foundation
v2 = zeros(l4,m4,n); % Azimuthal displacement of the foundation
v3 = zeros(l4,m4,n); % Radial displacement of the foundation

dv1 = zeros(l4,m4,n); % Perturbed axial displacement of the foundation
dv2 = zeros(l4,m4,n); % Perturbed azimuthal displacement of the foundation
dv3 = zeros(l4,m4,n); % Perturbed radial displacement of the foundation

r = zeros(1,n);

a1d11 = zeros(1,n);
a1d22 = zeros(1,n);
a1d33 = zeros(1,n);
a2d12 = zeros(1,n);
a3d13 = zeros(1,n);
a1d3 = zeros(1,n);
a3d1 = zeros(1,n);

b1d12 = zeros(1,n);
b2d11 = zeros(1,n);
b2d22 = zeros(1,n);
b2d33 = zeros(1,n);
b3d23 = zeros(1,n);
b2d3 = zeros(1,n);
b3d2 = zeros(1,n);

c1d13 = zeros(1,n);
```

```
c2d23 = zeros(1,n);
c3d11 = zeros(1,n);
c3d22 = zeros(1,n);
c3d33 = zeros(1,n);
c2d2 = zeros(1,n);
c3d3 = zeros(1,n);

Iv1 = zeros(1,n);
Iv2 = zeros(1,n);
Iv3 = zeros(1,n);

PX = zeros(1,n);

tx2 = zeros(2*l,m2);

%% Curvature terms of the membrane
A1d11 = L1 + 2*M1;
A1d22 = M1;
A2d12 = L1 + M1;
A3d1 = L1 - M2*Ih;
A1d3 = - M2*Ih;

B1d12 = L1 + M1;
B2d11 = M1;
B2d22 = L1 + 2*M1;
B3d2 = L1 + 2*M1 - M2*Ih;
B2d3 = - M2*Ih;

C1d1 = L1 + L2*Ih;
C2d2 = L1 + 2*M1 + L2*Ih;
C3d3 = (L2 + 2*M2)*Ih;
C3 = L1 + L2*Ih;

Xu1 = 2*A1d11*Idx11 + 2*A1d22*Idx22 - 3*A1d3*Idx3;
Xu2 = 2*B2d11*Idx11 + 2*B2d22*Idx22 - 3*B2d3*Idx3;
Xu3 = - C3 - 3*C3d3*Idx3;

Iu1 = 1/Xu1;
Iu2 = 1/Xu2;
Iu3 = 1/Xu3;

%% Curvature terms of the foundation
for k = 1:n

    r(k) = b + (k-1)*dx2;
    Ir = 1/r(k);

    a1d11(k) = (L2+2*M2);
    a1d22(k) = M2*Ir^2;
    a1d33(k) = M2;
    a2d12(k) = (L2+M2);
```

```
    a3d13(k) = (L2+M2);
    a1d3(k)  = M2*Ir;
    a3d1(k)  = (L2+M2)*Ir;

    b1d12(k) = (L2+M2);
    b2d11(k) = M2*r(k)^2;
    b2d22(k) = (L2+2*M2);
    b2d33(k) = M2*r(k)^2;
    b3d23(k) = (L2+M2);
    b2d3(k)  = 3*M2*r(k);
    b3d2(k)  = (L2+3*M2)*Ir;

    c1d13(k) = (L2+M2);
    c2d23(k) = (L2+M2);
    c3d11(k) = M2;
    c3d22(k) = M2*Ir^2;
    c3d33(k) = (L2+2*M2);
    c2d2(k)  = - 2*M2*Ir;
    c3d3(k)  = (L2+2*M2)*Ir;
    c3       = - (L2+2*M2)*Ir^2;

    Iv1(k) = 1/(2*a1d11(k)*Idx11 + 2*a1d22(k)*Idx22 + 2*a1d33(k)*Idx33);
    Iv2(k) = 1/(2*b2d11(k)*Idx11 + 2*b2d22(k)*Idx22 + 2*b2d33(k)*Idx33);
    Iv3(k) = 1/(2*c3d11(k)*Idx11 + 2*c3d22(k)*Idx22 + 2*c3d33(k)*Idx33 - c3);

end

Ibv1 = 1/(-3*Idx3);
Ibv2 = 1/(-3*Idx3);
Ibv3 = 1/(-3*(L2+2*M2)*Idx3 - L2);

JM1jB11 = 1/(3*(L1+2*M1)*Idx1);
JM1jB22 = 1/(3*Idx1);

JMljB11 = -1/(3*(L1+2*M1)*Idx1);
JMljB22 = -1/(3*Idx1);

JMi1B11 = 1/(3*Idx2);
JMi1B22 = 1/(3*(L1+2*M1)*Idx2);

JMimB11 = -1/(3*Idx2);
JMimB22 = -1/(3*(L1+2*M1)*Idx2);

ax(1,1) = 3*Idx1*(L1+2*M1);
ax(1,2) = 3*Idx2*L1;
ax(2,1) = 3*Idx1*L1;
ax(2,2) = 3*Idx2*(L1+2*M1);

J = inv(ax);

JM11B11 = J(1,1);
```

```
JM11B12 = J(1,2);
JM11B21 = J(2,1);
JM11B22 = J(2,2);

ax(1,1) = 3*Idx1*(L1+2*M1);
ax(1,2) = - 3*Idx2*L1;
ax(2,1) = 3*Idx1*L1;
ax(2,2) = - 3*Idx2*(L1+2*M1);

J = inv(ax);

JM1mB11 = J(1,1);
JM1mB12 = J(1,2);
JM1mB21 = J(2,1);
JM1mB22 = J(2,2);

ax(1,1) = -3*Idx1*(L1+2*M1);
ax(1,2) = 3*Idx2*L1;
ax(2,1) = -3*Idx1*L1;
ax(2,2) = 3*Idx2*(L1+2*M1);

J = inv(ax);

JMl1B11 = J(1,1);
JMl1B12 = J(1,2);
JMl1B21 = J(2,1);
JMl1B22 = J(2,2);

ax(1,1) = -3*Idx1*(L1+2*M1);
ax(1,2) = -3*Idx2*L1;
ax(2,1) = -3*Idx1*L1;
ax(2,2) = -3*Idx2*(L1+2*M1);

J = inv(ax);

JMlmB11 = J(1,1);
JMlmB12 = J(1,2);
JMlmB21 = J(2,1);
JMlmB22 = J(2,2);

p = 2;
pp = 10^10;

%% Main code
while errr < p

    for kk = 1:NNN

        %% Corners of the membrane
        v1d1 = (4*v1(l1+2,m1+1,n)-v1(l1+3,m1+1,n))*Idx1;
        v2d2 = (4*v2(l1+1,m1+2,n)-v2(l1+1,m1+3,n))*Idx2;
```

```
Xv1 = (L1+2*M1)*v1d1 + L1*(v2d2+v3(l1+1,m1+1,n)); % equation (71)
Xv2 = L1*v1d1 + (L1+2*M1)*(v2d2+v3(l1+1,m1+1,n)) - T0; % equation (68)

v1(l1+1,m1+1,n) = JM11B11*Xv1 + JM11B12*Xv2;
v2(l1+1,m1+1,n) = JM11B21*Xv1 + JM11B22*Xv2;

v1d1 = (4*v1(l1+2,m2,n)-v1(l1+3,m2,n))*Idx1;
v2d2 = -(4*v2(l1+1,m2-1,n)-v2(l1+1,m2-2,n))*Idx2;

Xv1 = (L1+2*M1)*v1d1 + L1*(v2d2+v3(l1+1,m2,n)); % equation (71)
Xv2 = L1*v1d1 + (L1+2*M1)*(v2d2+v3(l1+1,m2,n)) - Tmax; % equation (69)

v1(l1+1,m2,n) = JM1mB11*Xv1 + JM1mB12*Xv2;
v2(l1+1,m2,n) = JM1mB21*Xv1 + JM1mB22*Xv2;

v1d1 = -(4*v1(l3-1,m1+1,n)-v1(l3-2,m1+1,n))*Idx1;
v2d2 = (4*v2(l3,m1+2,n)-v2(l3,m1+3,n))*Idx2;

Xv1 = (L1+2*M1)*v1d1 + L1*(v2d2+v3(l3,m1+1,n)); % equation (71)
Xv2 = L1*v1d1 + (L1+2*M1)*(v2d2+v3(l3,m1+1,n)) - T0; % equation (68)

v1(l3,m1+1,n) = JMl1B11*Xv1 + JMl1B12*Xv2;
v2(l3,m1+1,n) = JMl1B21*Xv1 + JMl1B22*Xv2;

v1d1 = -(4*v1(l3-1,m2,n)-v1(l3-2,m2,n))*Idx1;
v2d2 = -(4*v2(l3,m2-1,n)-v2(l3,m2-2,n))*Idx2;

Xv1 = (L1+2*M1)*v1d1 + L1*(v2d2+v3(l3,m2,n)); % equation (71)
Xv2 = L1*v1d1 + (L1+2*M1)*(v2d2+v3(l3,m2,n)) - Tmax; % equation (69)

v1(l3,m2,n) = JMlmB11*Xv1 + JMlmB12*Xv2;
v2(l3,m2,n) = JMlmB21*Xv1 + JMlmB22*Xv2;

%% Stessed boundaries of the membrane
for i = l1+2:l3-1

    v1d1 = (v1(i+1,m1+1,n)-v1(i-1,m1+1,n))*Idx1;
    v2d1 = (v2(i+1,m1+1,n)-v2(i-1,m1+1,n))*Idx1;

    v1d2 = (4*v1(i,m1+2,n)-v1(i,m1+3,n))*Idx2;
    v2d2 = (4*v2(i,m1+2,n)-v2(i,m1+3,n))*Idx2;

    Xv1 = v2d1 + v1d2; % equation (71)
    Xv2 = L1*v1d1 + (L1+2*M1)*(v2d2+v3(i,m1+1,n)) - T0; % equation (68)

    v1(i,m1+1,n) = JMi1B11*Xv1;
    v2(i,m1+1,n) = JMi1B22*Xv2;

    v1d1 = (v1(i+1,m2,n)-v1(i-1,m2,n))*Idx1;
    v2d1 = (v2(i+1,m2,n)-v2(i-1,m2,n))*Idx1;
```

```
        v1d2 = -(4*v1(i,m2-1,n)-v1(i,m2-2,n))*Idx2;
        v2d2 = -(4*v2(i,m2-1,n)-v2(i,m2-2,n))*Idx2;

        Xv1 = v2d1 + v1d2; % equation (71)
        Xv2 = L1*v1d1 + (L1+2*M1)*(v2d2+v3(i,m2,n)) - Tmax; % equation (69)

        v1(i,m2,n) = JMimB11*Xv1;
        v2(i,m2,n) = JMimB22*Xv2;

    end

    %% Stress-free boundaries of the membrane
    for j = m1+2:m2-1

        v1d1 = (4*v1(l1+2,j,n)-v1(l1+3,j,n))*Idx1;
        v2d1 = (4*v2(l1+2,j,n)-v2(l1+3,j,n))*Idx1;

        v1d2 = (v1(l1+1,j+1,n)-v1(l1+1,j-1,n))*Idx2;
        v2d2 = (v2(l1+1,j+1,n)-v2(l1+1,j-1,n))*Idx2;

        Xv1 = (L1+2*M1)*v1d1 + L1*(v2d2+v3(l1+1,j,n)); % equation (70)
        Xv2 = v2d1 + v1d2; % equation (71)

        v1(l1+1,j,n) = JM1jB11*Xv1;
        v2(l1+1,j,n) = JM1jB22*Xv2;

        v1d1 = -(4*v1(l3-1,j,n)-v1(l3-2,j,n))*Idx1;
        v2d1 = -(4*v2(l3-1,j,n)-v2(l3-2,j,n))*Idx1;

        v1d2 = (v1(l3,j+1,n)-v1(l3,j-1,n))*Idx2;
        v2d2 = (v2(l3,j+1,n)-v2(l3,j-1,n))*Idx2;

        Xv1 = (L1+2*M1)*v1d1 + L1*(v2d2+v3(l3,j,n)); % equation (70)
        Xv2 = v2d1 + v1d2; % equation (71)

        v1(l3,j,n) = JMljB11*Xv1;
        v2(l3,j,n) = JMljB22*Xv2;

    end

    %% Governing equations of the membrane
    for i = l1+2:l3-1

        for j = m1+2:m2-1

            v1d1 = (v1(i+1,j,n)-v1(i-1,j,n))*Idx1;
            v3d1 = (v3(i+1,j,n)-v3(i-1,j,n))*Idx1;

            v2d2 = (v2(i,j+1,n)-v2(i,j-1,n))*Idx2;
            v3d2 = (v3(i,j+1,n)-v3(i,j-1,n))*Idx2;
```

```
v1d3 = -(4*v1(i,j,n-1)-v1(i,j,n-2))*Idx3;
v2d3 = -(4*v2(i,j,n-1)-v2(i,j,n-2))*Idx3;
v3d3 = -(4*v3(i,j,n-1)-v3(i,j,n-2))*Idx3;

v1d11 = (v1(i+1,j,n)+v1(i-1,j,n))*Idx11;
v2d11 = (v2(i+1,j,n)+v2(i-1,j,n))*Idx11;

v1d22 = (v1(i,j+1,n)+v1(i,j-1,n))*Idx22;
v2d22 = (v2(i,j+1,n)+v2(i,j-1,n))*Idx22;

v1d12 = (v1(i+1,j+1,n)-v1(i-1,j+1,n)...
    -v1(i+1,j-1,n)+v1(i-1,j-1,n))*Idx12;
v2d12 = (v2(i+1,j+1,n)-v2(i-1,j+1,n)...
    -v2(i+1,j-1,n)+v2(i-1,j-1,n))*Idx12;

%% Bounded boundary
Xv1 = A1d11*v1d11 + A1d22*v1d22 + A2d12*v2d12 ...
    + A3d1*v3d1 + A1d3*v1d3; % equation (63)
Xv2 = B1d12*v1d12 + B2d11*v2d11 + B2d22*v2d22 ...
    + B3d2*v3d2 + B2d3*v2d3; % equation (64)
Xv3 = C1d1*v1d1 + C2d2*v2d2 + C3d3*v3d3; % equation (65)

v1(i,j,n) = Xv1*Iu1;
v2(i,j,n) = Xv2*Iu2;
v3(i,j,n) = Xv3*Iu3;

alpha = sqrt(v1(i,j,n)^2 + v2(i,j,n)^2);
beta  = 1/alpha;
beta1 = v1(i,j,n)*beta;
beta2 = v2(i,j,n)*beta;
delta = v3(i,j,n) + IMu*alpha;

%% Limiting-equilibrium boundary
if ~(delta < 0)

    v3(i,j,n) = - IMu*alpha;

    ax(1,1) = (Xu1-IMu2*Xu3) - IMu*Xv3*beta + IMu*Xv3*beta*beta1^2;
    ax(1,2) = IMu*Xv3*beta*beta1*beta2;
    ax(2,1) = IMu*Xv3*beta*beta1*beta2;
    ax(2,2) = (Xu2-IMu2*Xu3) - IMu*Xv3*beta + IMu*Xv3*beta*beta2^2;

    J = inv(ax);

    Xdv1 = Xv1 - (Xu1-IMu2*Xu3)*v1(i,j,n) + IMu*Xv3*beta1; % equation (66)
    Xdv2 = Xv2 - (Xu2-IMu2*Xu3)*v2(i,j,n) + IMu*Xv3*beta2; % equation (67)

    dv1(i,j,n) = J(1,1)*Xdv1 + J(1,2)*Xdv2;
    dv2(i,j,n) = J(2,1)*Xdv1 + J(2,2)*Xdv2;
```

```
        end
    end
end

%% Stress-free boundary of the foundation
for i = l1+1:l3

    v1d1 = (v1(i+1,m1+1,n)-v1(i-1,m1+1,n))*Idx1;
    v2d2 = (v2(i,m1+2,n)-v2(i,m1,n))*Idx2;
    v3d3 = -(4*v3(i,m1+1,n-1)-v3(i,m1+1,n-2))*Idx3;

    Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)
    v3(i,m1+1,n) = Xv3*Ibv3;

    v1d1 = (v1(i+1,m2,n)-v1(i-1,m2,n))*Idx1;
    v2d2 = (v2(i,m2+1,n)-v2(i,m2-1,n))*Idx2;
    v3d3 = -(4*v3(i,m2,n-1)-v3(i,m2,n-2))*Idx3;

    Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)
    v3(i,m2,n) = Xv3*Ibv3;

end

for j = m1+2:m2-1

    v1d1 = (v1(l1+2,j,n)-v1(l1,j,n))*Idx1;
    v2d2 = (v2(l1+1,j+1,n)-v2(l1+1,j-1,n))*Idx2;
    v3d3 = -(4*v3(l1+1,j,n-1)-v3(l1+1,j,n-2))*Idx3;

    Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)
    v3(l1+1,j,n) = Xv3*Ibv3;

    v1d1 = (v1(l3+1,j,n)-v1(l3-1,j,n))*Idx1;
    v2d2 = (v2(l3,j+1,n)-v2(l3,j-1,n))*Idx2;
    v3d3 = -(4*v3(l3,j,n-1)-v3(l3,j,n-2))*Idx3;

    Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)
    v3(l3,j,n) = Xv3*Ibv3;

end

for i = 2:l4-1

    v1d1 = (v1(i+1,1,n)-v1(i-1,1,n))*Idx1;
    v3d1 = (v3(i+1,1,n)-v3(i-1,1,n))*Idx1;

    v2d2 = (v2(i,2,n)-v2(i,m4,n))*Idx2;
    v3d2 = (v3(i,2,n)-v3(i,m4,n))*Idx2;

    v1d3 = -(4*v1(i,1,n-1)-v1(i,1,n-2))*Idx3;
    v2d3 = -(4*v2(i,1,n-1)-v2(i,1,n-2))*Idx3;
```

```
    v3d3 = -(4*v3(i,1,n-1)-v3(i,1,n-2))*Idx3;

    Xv1 = v1d3 + v3d1; % equation (60)
    Xv2 = v2d3 + v3d2; % equation (61)
    Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)

    v1(i,1,n) = Xv1*Ibv1;
    v2(i,1,n) = Xv2*Ibv2;
    v3(i,1,n) = Xv3*Ibv3;

    for j = 2:m1

        v1d1 = (v1(i+1,j,n)-v1(i-1,j,n))*Idx1;
        v3d1 = (v3(i+1,j,n)-v3(i-1,j,n))*Idx1;

        v2d2 = (v2(i,j+1,n)-v2(i,j-1,n))*Idx2;
        v3d2 = (v3(i,j+1,n)-v3(i,j-1,n))*Idx2;

        v1d3 = -(4*v1(i,j,n-1)-v1(i,j,n-2))*Idx3;
        v2d3 = -(4*v2(i,j,n-1)-v2(i,j,n-2))*Idx3;
        v3d3 = -(4*v3(i,j,n-1)-v3(i,j,n-2))*Idx3;

        Xv1 = v1d3 + v3d1; % equation (60)
        Xv2 = v2d3 + v3d2; % equation (61)
        Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)

        v1(i,j,n) = Xv1*Ibv1;
        v2(i,j,n) = Xv2*Ibv2;
        v3(i,j,n) = Xv3*Ibv3;

    end
end

for i = 2:l4-1

    v1d1 = (v1(i+1,m4,n)-v1(i-1,m4,n))*Idx1;
    v3d1 = (v3(i+1,m4,n)-v3(i-1,m4,n))*Idx1;

    v2d2 = (v2(i,1,n)-v2(i,m4-1,n))*Idx2;
    v3d2 = (v3(i,1,n)-v3(i,m4-1,n))*Idx2;

    v1d3 = -(4*v1(i,m4,n-1)-v1(i,m4,n-2))*Idx3;
    v2d3 = -(4*v2(i,m4,n-1)-v2(i,m4,n-2))*Idx3;
    v3d3 = -(4*v3(i,m4,n-1)-v3(i,m4,n-2))*Idx3;

    Xv1 = v1d3 + v3d1; % equation (60)
    Xv2 = v2d3 + v3d2; % equation (61)
    Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)

    v1(i,m4,n) = Xv1*Ibv1;
    v2(i,m4,n) = Xv2*Ibv2;
```

```
        v3(i,m4,n) = Xv3*Ibv3;

        for j = m2+1:m4-1

            v1d1 = (v1(i+1,j,n)-v1(i-1,j,n))*Idx1;
            v3d1 = (v3(i+1,j,n)-v3(i-1,j,n))*Idx1;

            v2d2 = (v2(i,j+1,n)-v2(i,j-1,n))*Idx2;
            v3d2 = (v3(i,j+1,n)-v3(i,j-1,n))*Idx2;

            v1d3 = -(4*v1(i,j,n-1)-v1(i,j,n-2))*Idx3;
            v2d3 = -(4*v2(i,j,n-1)-v2(i,j,n-2))*Idx3;
            v3d3 = -(4*v3(i,j,n-1)-v3(i,j,n-2))*Idx3;

            Xv1 = v1d3 + v3d1; % equation (60)
            Xv2 = v2d3 + v3d2; % equation (61)
            Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)

            v1(i,j,n) = Xv1*Ibv1;
            v2(i,j,n) = Xv2*Ibv2;
            v3(i,j,n) = Xv3*Ibv3;

        end
    end


    for i = 2:l1

        for j = m1+1:m2

            v1d1 = (v1(i+1,j,n)-v1(i-1,j,n))*Idx1;
            v3d1 = (v3(i+1,j,n)-v3(i-1,j,n))*Idx1;

            v2d2 = (v2(i,j+1,n)-v2(i,j-1,n))*Idx2;
            v3d2 = (v3(i,j+1,n)-v3(i,j-1,n))*Idx2;

            v1d3 = -(4*v1(i,j,n-1)-v1(i,j,n-2))*Idx3;
            v2d3 = -(4*v2(i,j,n-1)-v2(i,j,n-2))*Idx3;
            v3d3 = -(4*v3(i,j,n-1)-v3(i,j,n-2))*Idx3;

            Xv1 = v1d3 + v3d1; % equation (60)
            Xv2 = v2d3 + v3d2; % equation (61)
            Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)

            v1(i,j,n) = Xv1*Ibv1;
            v2(i,j,n) = Xv2*Ibv2;
            v3(i,j,n) = Xv3*Ibv3;

        end
    end
```

```
for i = l3+1:l4-1

    for j = m1+1:m2

        v1d1 = (v1(i+1,j,n)-v1(i-1,j,n))*Idx1;
        v3d1 = (v3(i+1,j,n)-v3(i-1,j,n))*Idx1;

        v2d2 = (v2(i,j+1,n)-v2(i,j-1,n))*Idx2;
        v3d2 = (v3(i,j+1,n)-v3(i,j-1,n))*Idx2;

        v1d3 = -(4*v1(i,j,n-1)-v1(i,j,n-2))*Idx3;
        v2d3 = -(4*v2(i,j,n-1)-v2(i,j,n-2))*Idx3;
        v3d3 = -(4*v3(i,j,n-1)-v3(i,j,n-2))*Idx3;

        Xv1 = v1d3 + v3d1; % equation (60)
        Xv2 = v2d3 + v3d2; % equation (61)
        Xv3 = L2*(v1d1+v2d2) + (L2+2*M2)*v3d3; % equation (62)

        v1(i,j,n) = Xv1*Ibv1;
        v2(i,j,n) = Xv2*Ibv2;
        v3(i,j,n) = Xv3*Ibv3;

    end
end

%% Governing equations the foundation
for k = 2:n-1

    %% Governing equations at the periodic boundaries
    for i = 2:l4-1

        v3d1 = (v3(i+1,1,k)-v3(i-1,1,k))*Idx1;

        v2d2 = (v2(i,2,k)-v2(i,m4-1,k))*Idx2;
        v3d2 = (v3(i,2,k)-v3(i,m4-1,k))*Idx2;

        v1d3 = (v1(i,1,k+1)-v1(i,1,k-1))*Idx3;
        v2d3 = (v2(i,1,k+1)-v2(i,1,k-1))*Idx3;
        v3d3 = (v3(i,1,k+1)-v3(i,1,k-1))*Idx3;

        v1d11 = (v1(i+1,1,k)+v1(i-1,1,k))*Idx11;
        v2d11 = (v2(i+1,1,k)+v2(i-1,1,k))*Idx11;
        v3d11 = (v3(i+1,1,k)+v3(i-1,1,k))*Idx11;

        v1d22 = (v1(i,2,k)+v1(i,m4-1,k))*Idx22;
        v2d22 = (v2(i,2,k)+v2(i,m4-1,k))*Idx22;
        v3d22 = (v3(i,2,k)+v3(i,m4-1,k))*Idx22;

        v1d33 = (v1(i,1,k+1)+v1(i,1,k-1))*Idx33;
        v2d33 = (v2(i,1,k+1)+v2(i,1,k-1))*Idx33;
        v3d33 = (v3(i,1,k+1)+v3(i,1,k-1))*Idx33;
```

```
v1d12 = (v1(i+1,2,k)-v1(i-1,2,k)...
    -v1(i+1,m4-1,k)+v1(i-1,m4-1,k))*Idx12;
v2d12 = (v2(i+1,2,k)-v2(i-1,2,k)...
    -v2(i+1,m4-1,k)+v2(i-1,m4-1,k))*Idx12;

v1d13 = (v1(i+1,1,k+1)-v1(i-1,1,k+1)...
    -v1(i+1,1,k-1)+v1(i-1,1,k-1))*Idx13;
v3d13 = (v3(i+1,1,k+1)-v3(i-1,1,k+1)...
    -v3(i+1,1,k-1)+v3(i-1,1,k-1))*Idx13;

v2d23 = (v2(i,2,k+1)-v2(i,m4-1,k+1)...
    -v2(i,2,k-1)+v2(i,m4-1,k-1))*Idx23;
v3d23 = (v3(i,2,k+1)-v3(i,m4-1,k+1)...
    -v3(i,2,k-1)+v3(i,m4-1,k-1))*Idx23;

Xv1 = a1d11(k)*v1d11 + a1d22(k)*v1d22 + a1d33(k)*v1d33 + a2d12(k)*v2d12 ...
    + a3d13(k)*v3d13 + a1d3(k)*v1d3 + a3d1(k)*v3d1; % equation (57)
Xv2 = b1d12(k)*v1d12 + b2d11(k)*v2d11 + b2d22(k)*v2d22 + b2d33(k)*v2d33 ...
    + b3d23(k)*v3d23 + b2d3(k)*v2d3 + b3d2(k)*v3d2; % equation (58)
Xv3 = c1d13(k)*v1d13 + c2d23(k)*v2d23 + c3d11(k)*v3d11 + c3d22(k)*v3d22 ...
    + c3d33(k)*v3d33 + c2d2(k)*v2d2 + c3d3(k)*v3d3; % equation (59)

v1(i,1,k) = Xv1*Iv1(k);
v2(i,1,k) = Xv2*Iv2(k);
v3(i,1,k) = Xv3*Iv3(k);

v3d1 = (v3(i+1,m4,k)-v3(i-1,m4,k))*Idx1;

v2d2 = (v2(i,1,k)-v2(i,m4-1,k))*Idx2;
v3d2 = (v3(i,1,k)-v3(i,m4-1,k))*Idx2;

v1d3 = (v1(i,m4,k+1)-v1(i,m4,k-1))*Idx3;
v2d3 = (v2(i,m4,k+1)-v2(i,m4,k-1))*Idx3;
v3d3 = (v3(i,m4,k+1)-v3(i,m4,k-1))*Idx3;

v1d11 = (v1(i+1,m4,k)+v1(i-1,m4,k))*Idx11;
v2d11 = (v2(i+1,m4,k)+v2(i-1,m4,k))*Idx11;
v3d11 = (v3(i+1,m4,k)+v3(i-1,m4,k))*Idx11;

v1d22 = (v1(i,1,k)+v1(i,m4-1,k))*Idx22;
v2d22 = (v2(i,1,k)+v2(i,m4-1,k))*Idx22;
v3d22 = (v3(i,1,k)+v3(i,m4-1,k))*Idx22;

v1d33 = (v1(i,m4,k+1)+v1(i,m4,k-1))*Idx33;
v2d33 = (v2(i,m4,k+1)+v2(i,m4,k-1))*Idx33;
v3d33 = (v3(i,m4,k+1)+v3(i,m4,k-1))*Idx33;

v1d12 = (v1(i+1,1,k)-v1(i-1,1,k)...
    -v1(i+1,m4-1,k)+v1(i-1,m4-1,k))*Idx12;
v2d12 = (v2(i+1,1,k)-v2(i-1,1,k)...
```

```
    -v2(i+1,m4-1,k)+v2(i-1,m4-1,k))*Idx12;

v1d13 = (v1(i+1,m4,k+1)-v1(i-1,m4,k+1)...
    -v1(i+1,m4,k-1)+v1(i-1,m4,k-1))*Idx13;
v3d13 = (v3(i+1,m4,k+1)-v3(i-1,m4,k+1)...
    -v3(i+1,m4,k-1)+v3(i-1,m4,k-1))*Idx13;

v2d23 = (v2(i,1,k+1)-v2(i,m4-1,k+1)...
    -v2(i,1,k-1)+v2(i,m4-1,k-1))*Idx23;
v3d23 = (v3(i,1,k+1)-v3(i,m4-1,k+1)...
    -v3(i,1,k-1)+v3(i,m4-1,k-1))*Idx23;

Xv1 = a1d11(k)*v1d11 + a1d22(k)*v1d22 + a1d33(k)*v1d33 + a2d12(k)*v2d12 ...
    + a3d13(k)*v3d13 + a1d3(k)*v1d3 + a3d1(k)*v3d1; % equation (57)
Xv2 = b1d12(k)*v1d12 + b2d11(k)*v2d11 + b2d22(k)*v2d22 + b2d33(k)*v2d33 ...
    + b3d23(k)*v3d23 + b2d3(k)*v2d3 + b3d2(k)*v3d2; % equation (58)
Xv3 = c1d13(k)*v1d13 + c2d23(k)*v2d23 + c3d11(k)*v3d11 + c3d22(k)*v3d22 ...
    + c3d33(k)*v3d33 + c2d2(k)*v2d2 + c3d3(k)*v3d3; % equation (59)

v1(i,m4,k) = Xv1*Iv1(k);
v2(i,m4,k) = Xv2*Iv2(k);
v3(i,m4,k) = Xv3*Iv3(k);

dv3d1 = (dv3(i+1,1,k)-dv3(i-1,1,k))*Idx1;

dv2d2 = (dv2(i,2,k)-dv2(i,m4-1,k))*Idx2;
dv3d2 = (dv3(i,2,k)-dv3(i,m4-1,k))*Idx2;

dv1d3 = (dv1(i,1,k+1)-dv1(i,1,k-1))*Idx3;
dv2d3 = (dv2(i,1,k+1)-dv2(i,1,k-1))*Idx3;
dv3d3 = (dv3(i,1,k+1)-dv3(i,1,k-1))*Idx3;

dv1d11 = (dv1(i+1,1,k)+dv1(i-1,1,k))*Idx11;
dv2d11 = (dv2(i+1,1,k)+dv2(i-1,1,k))*Idx11;
dv3d11 = (dv3(i+1,1,k)+dv3(i-1,1,k))*Idx11;

dv1d22 = (dv1(i,2,k)+dv1(i,m4-1,k))*Idx22;
dv2d22 = (dv2(i,2,k)+dv2(i,m4-1,k))*Idx22;
dv3d22 = (dv3(i,2,k)+dv3(i,m4-1,k))*Idx22;

dv1d33 = (dv1(i,1,k+1)+dv1(i,1,k-1))*Idx33;
dv2d33 = (dv2(i,1,k+1)+dv2(i,1,k-1))*Idx33;
dv3d33 = (dv3(i,1,k+1)+dv3(i,1,k-1))*Idx33;

dv1d12 = (dv1(i+1,2,k)-dv1(i-1,2,k)...
    -dv1(i+1,m4-1,k)+dv1(i-1,m4-1,k))*Idx12;
dv2d12 = (dv2(i+1,2,k)-dv2(i-1,2,k)...
    -dv2(i+1,m4-1,k)+dv2(i-1,m4-1,k))*Idx12;

dv1d13 = (dv1(i+1,1,k+1)-dv1(i-1,1,k+1)...
    -dv1(i+1,1,k-1)+dv1(i-1,1,k-1))*Idx13;
```

```
dv3d13 = (dv3(i+1,1,k+1)-dv3(i-1,1,k+1)...
    -dv3(i+1,1,k-1)+dv3(i-1,1,k-1))*Idx13;

dv2d23 = (dv2(i,2,k+1)-dv2(i,m4-1,k+1)...
    -dv2(i,2,k-1)+dv2(i,m4-1,k-1))*Idx23;
dv3d23 = (dv3(i,2,k+1)-dv3(i,m4-1,k+1)...
    -dv3(i,2,k-1)+dv3(i,m4-1,k-1))*Idx23;

Xdv1 = a1d11(k)*dv1d11 + a1d22(k)*dv1d22 + a1d33(k)*dv1d33 ...
    + a2d12(k)*dv2d12 + a3d13(k)*dv3d13 + a1d3(k)*dv1d3 + a3d1(k)*dv3d1;
Xdv2 = b1d12(k)*dv1d12 + b2d11(k)*dv2d11 + b2d22(k)*dv2d22 ...
    + b2d33(k)*dv2d33 + b3d23(k)*dv3d23 + b2d3(k)*dv2d3 + b3d2(k)*dv3d2;
Xdv3 = c1d13(k)*dv1d13 + c2d23(k)*dv2d23 + c3d11(k)*dv3d11 ...
    + c3d22(k)*dv3d22 + c3d33(k)*dv3d33 + c2d2(k)*dv2d2 + c3d3(k)*dv3d3;

dv1(i,1,k) = Xdv1*Iv1(k);
dv2(i,1,k) = Xdv2*Iv2(k);
dv3(i,1,k) = Xdv3*Iv3(k);

dv3d1 = (dv3(i+1,m4,k)-dv3(i-1,m4,k))*Idx1;

dv2d2 = (dv2(i,1,k)-dv2(i,m4-1,k))*Idx2;
dv3d2 = (dv3(i,1,k)-dv3(i,m4-1,k))*Idx2;

dv1d3 = (dv1(i,m4,k+1)-dv1(i,m4,k-1))*Idx3;
dv2d3 = (dv2(i,m4,k+1)-dv2(i,m4,k-1))*Idx3;
dv3d3 = (dv3(i,m4,k+1)-dv3(i,m4,k-1))*Idx3;

dv1d11 = (dv1(i+1,m4,k)+dv1(i-1,m4,k))*Idx11;
dv2d11 = (dv2(i+1,m4,k)+dv2(i-1,m4,k))*Idx11;
dv3d11 = (dv3(i+1,m4,k)+dv3(i-1,m4,k))*Idx11;

dv1d22 = (dv1(i,1,k)+dv1(i,m4-1,k))*Idx22;
dv2d22 = (dv2(i,1,k)+dv2(i,m4-1,k))*Idx22;
dv3d22 = (dv3(i,1,k)+dv3(i,m4-1,k))*Idx22;

dv1d33 = (dv1(i,m4,k+1)+dv1(i,m4,k-1))*Idx33;
dv2d33 = (dv2(i,m4,k+1)+dv2(i,m4,k-1))*Idx33;
dv3d33 = (dv3(i,m4,k+1)+dv3(i,m4,k-1))*Idx33;

dv1d12 = (dv1(i+1,1,k)-dv1(i-1,1,k)...
    -dv1(i+1,m4-1,k)+dv1(i-1,m4-1,k))*Idx12;
dv2d12 = (dv2(i+1,1,k)-dv2(i-1,1,k)...
    -dv2(i+1,m4-1,k)+dv2(i-1,m4-1,k))*Idx12;

dv1d13 = (dv1(i+1,m4,k+1)-dv1(i-1,m4,k+1)...
    -dv1(i+1,m4,k-1)+dv1(i-1,m4,k-1))*Idx13;
dv3d13 = (dv3(i+1,m4,k+1)-dv3(i-1,m4,k+1)...
    -dv3(i+1,m4,k-1)+dv3(i-1,m4,k-1))*Idx13;

dv2d23 = (dv2(i,1,k+1)-dv2(i,m4-1,k+1)...
```

```
    -dv2(i,1,k-1)+dv2(i,m4-1,k-1))*Idx23;
dv3d23 = (dv3(i,1,k+1)-dv3(i,m4-1,k+1)...
    -dv3(i,1,k-1)+dv3(i,m4-1,k-1))*Idx23;

Xdv1 = a1d11(k)*dv1d11 + a1d22(k)*dv1d22 + a1d33(k)*dv1d33 ...
    + a2d12(k)*dv2d12 + a3d13(k)*dv3d13 + a1d3(k)*dv1d3 + a3d1(k)*dv3d1;
Xdv2 = b1d12(k)*dv1d12 + b2d11(k)*dv2d11 + b2d22(k)*dv2d22 ...
    + b2d33(k)*dv2d33 + b3d23(k)*dv3d23 + b2d3(k)*dv2d3 + b3d2(k)*dv3d2;
Xdv3 = c1d13(k)*dv1d13 + c2d23(k)*dv2d23 + c3d11(k)*dv3d11 ...
    + c3d22(k)*dv3d22 + c3d33(k)*dv3d33 + c2d2(k)*dv2d2 + c3d3(k)*dv3d3;

dv1(i,m4,k) = Xdv1*Iv1(k);
dv2(i,m4,k) = Xdv2*Iv2(k);
dv3(i,m4,k) = Xdv3*Iv3(k);

%% Governing equations of the foundation in the domain
for j = 2:m4-1

    v3d1 = (v3(i+1,j,k)-v3(i-1,j,k))*Idx1;

    v2d2 = (v2(i,j+1,k)-v2(i,j-1,k))*Idx2;
    v3d2 = (v3(i,j+1,k)-v3(i,j-1,k))*Idx2;

    v1d3 = (v1(i,j,k+1)-v1(i,j,k-1))*Idx3;
    v2d3 = (v2(i,j,k+1)-v2(i,j,k-1))*Idx3;
    v3d3 = (v3(i,j,k+1)-v3(i,j,k-1))*Idx3;

    v1d11 = (v1(i+1,j,k)+v1(i-1,j,k))*Idx11;
    v2d11 = (v2(i+1,j,k)+v2(i-1,j,k))*Idx11;
    v3d11 = (v3(i+1,j,k)+v3(i-1,j,k))*Idx11;

    v1d22 = (v1(i,j+1,k)+v1(i,j-1,k))*Idx22;
    v2d22 = (v2(i,j+1,k)+v2(i,j-1,k))*Idx22;
    v3d22 = (v3(i,j+1,k)+v3(i,j-1,k))*Idx22;

    v1d33 = (v1(i,j,k+1)+v1(i,j,k-1))*Idx33;
    v2d33 = (v2(i,j,k+1)+v2(i,j,k-1))*Idx33;
    v3d33 = (v3(i,j,k+1)+v3(i,j,k-1))*Idx33;

    v1d12 = (v1(i+1,j+1,k)-v1(i-1,j+1,k)...
        -v1(i+1,j-1,k)+v1(i-1,j-1,k))*Idx12;
    v2d12 = (v2(i+1,j+1,k)-v2(i-1,j+1,k)...
        -v2(i+1,j-1,k)+v2(i-1,j-1,k))*Idx12;

    v1d13 = (v1(i+1,j,k+1)-v1(i-1,j,k+1)...
        -v1(i+1,j,k-1)+v1(i-1,j,k-1))*Idx13;
    v3d13 = (v3(i+1,j,k+1)-v3(i-1,j,k+1)...
        -v3(i+1,j,k-1)+v3(i-1,j,k-1))*Idx13;

    v2d23 = (v2(i,j+1,k+1)-v2(i,j-1,k+1)...
        -v2(i,j+1,k-1)+v2(i,j-1,k-1))*Idx23;
```

```
v3d23 = (v3(i,j+1,k+1)-v3(i,j-1,k+1)...
    -v3(i,j+1,k-1)+v3(i,j-1,k-1))*Idx23;

Xv1 = a1d11(k)*v1d11 + a1d22(k)*v1d22 + a1d33(k)*v1d33 + a2d12(k)*v2d12 ...
    + a3d13(k)*v3d13 + a1d3(k)*v1d3 + a3d1(k)*v3d1; % equation (57)
Xv2 = b1d12(k)*v1d12 + b2d11(k)*v2d11 + b2d22(k)*v2d22 + b2d33(k)*v2d33 ...
    + b3d23(k)*v3d23 + b2d3(k)*v2d3 + b3d2(k)*v3d2; % equation (58)
Xv3 = c1d13(k)*v1d13 + c2d23(k)*v2d23 + c3d11(k)*v3d11 + c3d22(k)*v3d22 ...
    + c3d33(k)*v3d33 + c2d2(k)*v2d2 + c3d3(k)*v3d3; % equation (59)

v1(i,j,k) = Xv1*Iv1(k);
v2(i,j,k) = Xv2*Iv2(k);
v3(i,j,k) = Xv3*Iv3(k);

dv3d1 = (dv3(i+1,j,k)-dv3(i-1,j,k))*Idx1;

dv2d2 = (dv2(i,j+1,k)-dv2(i,j-1,k))*Idx2;
dv3d2 = (dv3(i,j+1,k)-dv3(i,j-1,k))*Idx2;

dv1d3 = (dv1(i,j,k+1)-dv1(i,j,k-1))*Idx3;
dv2d3 = (dv2(i,j,k+1)-dv2(i,j,k-1))*Idx3;
dv3d3 = (dv3(i,j,k+1)-dv3(i,j,k-1))*Idx3;

dv1d11 = (dv1(i+1,j,k)+dv1(i-1,j,k))*Idx11;
dv2d11 = (dv2(i+1,j,k)+dv2(i-1,j,k))*Idx11;
dv3d11 = (dv3(i+1,j,k)+dv3(i-1,j,k))*Idx11;

dv1d22 = (dv1(i,j+1,k)+dv1(i,j-1,k))*Idx22;
dv2d22 = (dv2(i,j+1,k)+dv2(i,j-1,k))*Idx22;
dv3d22 = (dv3(i,j+1,k)+dv3(i,j-1,k))*Idx22;

dv1d33 = (dv1(i,j,k+1)+dv1(i,j,k-1))*Idx33;
dv2d33 = (dv2(i,j,k+1)+dv2(i,j,k-1))*Idx33;
dv3d33 = (dv3(i,j,k+1)+dv3(i,j,k-1))*Idx33;

dv1d12 = (dv1(i+1,j+1,k)-dv1(i-1,j+1,k)...
    -dv1(i+1,j-1,k)+dv1(i-1,j-1,k))*Idx12;
dv2d12 = (dv2(i+1,j+1,k)-dv2(i-1,j+1,k)...
    -dv2(i+1,j-1,k)+dv2(i-1,j-1,k))*Idx12;

dv1d13 = (dv1(i+1,j,k+1)-dv1(i-1,j,k+1)...
    -dv1(i+1,j,k-1)+dv1(i-1,j,k-1))*Idx13;
dv3d13 = (dv3(i+1,j,k+1)-dv3(i-1,j,k+1)...
    -dv3(i+1,j,k-1)+dv3(i-1,j,k-1))*Idx13;

dv2d23 = (dv2(i,j+1,k+1)-dv2(i,j-1,k+1)...
    -dv2(i,j+1,k-1)+dv2(i,j-1,k-1))*Idx23;
dv3d23 = (dv3(i,j+1,k+1)-dv3(i,j-1,k+1)...
    -dv3(i,j+1,k-1)+dv3(i,j-1,k-1))*Idx23;

Xdv1 = a1d11(k)*dv1d11 + a1d22(k)*dv1d22 + a1d33(k)*dv1d33 ...
```

```
                        + a2d12(k)*dv2d12 + a3d13(k)*dv3d13 + a1d3(k)*dv1d3 + a3d1(k)*dv3d1;
                    Xdv2 = b1d12(k)*dv1d12 + b2d11(k)*dv2d11 + b2d22(k)*dv2d22 ...
                        + b2d33(k)*dv2d33 + b3d23(k)*dv3d23 + b2d3(k)*dv2d3 + b3d2(k)*dv3d2;
                    Xdv3 = c1d13(k)*dv1d13 + c2d23(k)*dv2d23 + c3d11(k)*dv3d11 ...
                        + c3d22(k)*dv3d22 + c3d33(k)*dv3d33 + c2d2(k)*dv2d2 + c3d3(k)*dv3d3;

                    dv1(i,j,k) = Xdv1*Iv1(k);
                    dv2(i,j,k) = Xdv2*Iv2(k);
                    dv3(i,j,k) = Xdv3*Iv3(k);

                end
            end
        end

        %% Newton's method
        v1(:,:,:) = dv1(:,:,:) + v1(:,:,:);
        v2(:,:,:) = dv2(:,:,:) + v2(:,:,:);
        v3(:,:,:) = dv3(:,:,:) + v3(:,:,:);

    end

    %% Terminating condition
    for k = 2:n
        PX(k) = norm(v1(:,:,k),2) + norm(v2(:,:,k),2) + norm(v3(:,:,k),2) ...
            + norm(dv1(:,:,k),2) + norm(dv2(:,:,k),2) + norm(dv3(:,:,k),2);
    end

    ppp = norm(PX,2);
    p = abs(1-ppp/pp);

    pp = ppp;

end

%% Shear
for i = 2:l4-1
    for j = 2:m4-1

        v3d2 = (v3(i,j+1,n)-v3(i,j-1,n))*Idx2;
        v2d3 = (3*v2(i,j,n)-4*v2(i,j,n-1)+v2(i,j,n-2))*Idx3;
        tx2(i,j) = M2*(v2d3+v3d2);

    end
end

normV = norm(v2(:,:,n),2); % Total boundary displacement
normT = norm(tx2,2); % Total shear

%% Save data file
filename = 'ShellMembrane.mat';
save(filename)
```

References

[1] M. Abramowitz, I. A. Stegun, et al. *Handbook of mathematical functions*, volume 1046. Dover New York, 1965.

[2] A. A. L. Baldelli and B. Bourdin. On the asymptotic derivation of winkler-type energies from 3d elasticity. *Journal of Elasticity*, pages 1–27, 2015.

[3] R. Burden, J. Faires, and A. Burden. *Numerical analysis*. Nelson Education, 2015.

[4] K. Jayawardana. *Mathematical Theory of Shells on Elastic Foundations: An Analysis of Boundary Forms, Constraints, and Applications to Friction and Skin Abrasion*. PhD thesis, University College London, 2016.

[5] K. Jayawardana, C. Mordacq, C. Ortner, and H. S. Park. An analysis of the boundary layer in the 1d surface cauchy–born model. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(01):109–123, 2013.

[6] N. Kikuchi and J.T. Oden. *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, 1988.

[7] B. E. Meserve and R. E. Pingry. Some notes on the prismoidal formula. *The Mathematics Teacher*, 45(4):257–263, 1952.

[8] Unknown. Matlab. https://uk.mathworks.com/help/matlab/ref/norm.html, 2016. 2-Norm of Matrix.