

Universitat Politècnica de Catalunya



Master's degree in
Numerical Methods in Engineering

Computer Solid Mechanics

Assignment 3
Nonlinear elasticity

Gabriel Valdés Alonzo

II Semester A.Y. 2016-2017
Professor: Marino Arroyo

1. Kirchhoff Saint-Venant material model

Isotropic linear elasticity can be derived from balance of linear momentum, the linearized strain displacement relation $\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$, and the stored elastic energy function

$$W(\boldsymbol{\varepsilon}) = \frac{\lambda}{2}(\text{tr } \boldsymbol{\varepsilon})^2 + \mu \text{tr}(\boldsymbol{\varepsilon}^2) = \frac{\lambda}{2}(\varepsilon_{ii})^2 + \mu \varepsilon_{jk} \varepsilon_{jk}.$$

(1) Check that, the stress tensor obtained from $\boldsymbol{\sigma} = \partial W / \partial \boldsymbol{\varepsilon}$ agrees with the usual linear elasticity expression.

The matrix derivative of a trace can be seen in tensorial notation for the first term:

$$\begin{aligned} \frac{\partial \text{tr } \boldsymbol{\varepsilon}}{\partial \boldsymbol{\varepsilon}} &= \frac{\partial}{\partial \varepsilon_{ij}} (\varepsilon_{ii}) \\ &= \delta_{ij} \\ &= \mathbf{1} \end{aligned}$$

and for the second one:

$$\begin{aligned} \frac{\partial \text{tr}(\boldsymbol{\varepsilon}^2)}{\partial \boldsymbol{\varepsilon}} &= \frac{\partial}{\partial \varepsilon_{ij}} (\varepsilon_{ij}^2) \\ &= 2\varepsilon_{ij} \\ &= 2\boldsymbol{\varepsilon} \end{aligned}$$

Knowing this, we can compute the derivative of the function:

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\varepsilon}} W(\boldsymbol{\varepsilon}) &= \frac{2\lambda}{2} (\text{tr } \boldsymbol{\varepsilon}) \mathbf{1} + 2\mu \boldsymbol{\varepsilon} \\ &= \lambda (\text{tr } \boldsymbol{\varepsilon}) \mathbf{1} + 2\mu \boldsymbol{\varepsilon} \\ &= \lambda (\varepsilon_{ii}) \delta_{kl} + 2\mu \varepsilon_{kl} \end{aligned}$$

From theory it is known that the linear elasticity expression is $\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}$, which can be developed as:

$$\begin{aligned} \boldsymbol{\sigma} &= (\lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \mathbb{I}) : \boldsymbol{\varepsilon} \\ &= (\lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \mathbb{I})_{kl ij} \varepsilon_{ij} \\ &= \lambda \delta_{kl} \delta_{ji} \varepsilon_{ij} + 2\mu \delta_{kl ij} \varepsilon_{ij} \\ &= \lambda \delta_{kl} \varepsilon_{ii} + 2\mu \varepsilon_{kl} \end{aligned}$$

which is identical to the formulation obtained from W , so we can conclude that both expressions are the same.

Since the linearization of the Green-Lagrange strain tensor $\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I})$ is the small strain tensor $\boldsymbol{\varepsilon}$, it is natural to extend isotropic elasticity to nonlinear elasticity as

$$W(\mathbf{E}) = \frac{\lambda}{2}(\text{tr } \mathbf{E})^2 + \mu \text{tr}(\mathbf{E}^2) \quad (1)$$

This hyperelastic model is called Kirchhoff Saint-Venant material model.

(2) According to the definition we gave in class about isotropy in nonlinear elasticity, is this model isotropic?

The definition for isotropy in class was that any rotation Q is a material symmetry, also written as $W(F) = W(FQ)$. We can prove it taking a rotation matrix of the form

$$Q = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We choose an angle to work with (i.e, $\theta = \pi/2$), and applying it to the tensor:

$$QEQ^T = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} E_{22} & -E_{12} & -E_{23} \\ -E_{12} & E_{11} & E_{13} \\ -E_{23} & E_{13} & E_{33} \end{bmatrix}$$

Computing E^2 and $(QEQ^T)^2$ also:

$$E^2 = \begin{bmatrix} E_{11}^2 + E_{12}^2 + E_{13}^2 & E_{11}E_{12} + E_{12}E_{22} + E_{13}E_{23} & E_{11}E_{13} + E_{12}E_{23} + E_{13}E_{33} \\ E_{11}E_{12} + E_{12}E_{22} + E_{13}E_{23} & E_{12}^2 + E_{22}^2 + E_{23}^2 & E_{12}E_{13} + E_{22}E_{23} + E_{23}E_{33} \\ E_{11}E_{13} + E_{12}E_{23} + E_{13}E_{33} & E_{12}E_{13} + E_{22}E_{23} + E_{23}E_{33} & E_{13}^2 + E_{23}^2 + E_{33}^2 \end{bmatrix}$$

$$(QEQ^T)^2 = \begin{bmatrix} E_{12}^2 + E_{22}^2 + E_{23}^2 & -E_{11}E_{12} - E_{12}E_{22} - E_{13}E_{23} & -E_{12}E_{13} - E_{22}E_{23} - E_{23}E_{33} \\ -E_{11}E_{12} - E_{12}E_{22} - E_{13}E_{23} & E_{11}^2 + E_{12}^2 + E_{13}^2 & E_{11}E_{13} + E_{12}E_{23} + E_{13}E_{33} \\ -E_{12}E_{13} - E_{22}E_{23} - E_{23}E_{33} & E_{11}E_{13} + E_{12}E_{23} + E_{13}E_{33} & E_{13}^2 + E_{23}^2 + E_{33}^2 \end{bmatrix}$$

We know that for this model, the value of $W(E)$ is a linear combination of the traces of E and E^2 . As we see, with a rotation the values of the diagonal of both E and E^2 are not altered, just changes position. This means that the trace remains intact, meaning that the value of $W(E)$ will be equal to the value of $W(QEQ^T)$. This is a case where the rotation is complete in an axis. If we generalize to "semi-rotations" (for example $\theta = \pi/4$), we would have some multiplications for constant terms everywhere. In any case, at the moment of summing the diagonal, said terms should disappear, and give again the same trace. So in conclusion, the model is isotropic.

(3) Derive the second Piola-Kirchhoff stress \mathbf{S} .

The formula for the second Piola-Kirchhoff tensor comes directly from $W(\mathbf{E})$:

$$\begin{aligned} \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}} &= \frac{\partial}{\partial \mathbf{E}} \left(\frac{\lambda}{2}(\text{tr } \mathbf{E})^2 + \mu \text{tr}(\mathbf{E}^2) \right) \\ &= \frac{\lambda}{2} 2(\text{tr } \mathbf{E})\mathbf{1} + 2\mu\mathbf{E} \\ &= \lambda(\text{tr } \mathbf{E})\mathbf{1} + 2\mu\mathbf{E} \\ &= \lambda E_{ii}\delta_{jk} + 2\mu E_{jk} \end{aligned}$$

- (4) For a uniform deformation of a rod aligned with the X axis ($x = \Lambda X$, $y = Y$, $z = Z$, where $\Lambda > 0$ is the stretch ratio along the X direction) derive the relation between the nominal normal stress P (the xX component of the first Piola-Kirchhoff stress) and the stretch ratio Λ , $P(\Lambda)$, and plot it.

From the uniform deformation we have the relation

$$F = \begin{bmatrix} \Lambda & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$E = \frac{1}{2}(FF^T - I) = \begin{bmatrix} \frac{1}{2}(\Lambda^2 - 1) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Since we have a uniform deformation, the relation between stress and strain is given by Hooke's Law, namely $P_{11}(\Lambda) = S_{11}F_{11}$. The first component of the Piola-Kirchhoff tensor is

$$S_{11} = \lambda \text{tr}(E) + 2\mu E_{11} = \left(\frac{1}{2}\lambda + \mu\right) (\Lambda^2 - 1)$$

So the relation is

$$P_{11}(\Lambda) = \left(\frac{1}{2}\lambda + \mu\right) (\Lambda^3 - \Lambda)$$

So we can plot the normalized version in order to see the behavior (i.e., $f(x) = x^3 - x$):

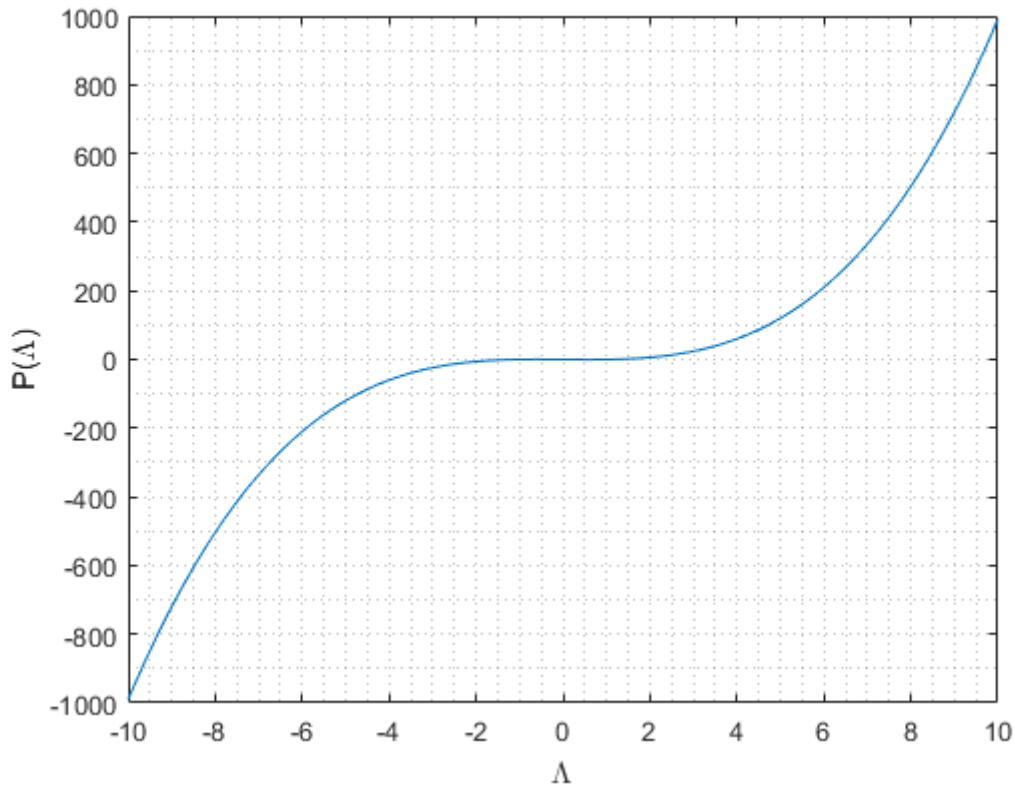


Figure 1: $P(\Lambda)$ vs. Λ .

- (5) Is the relation $P(\Lambda)$ monotonic? If not, derive the critical stretch Λ_{crit} at which the model fails with zero stiffness. Does this critical stretch depend on the elastic constants? Show that the material does not satisfy the growth conditions

$$W(\mathbf{E}) \longrightarrow +\infty \quad \text{when } J \longrightarrow 0^+.$$

Discuss your answers.

In order to see if the function is monotonic, we check the derivative:

$$\frac{dP(\Lambda)}{d\Lambda} = \left(\frac{1}{2}\lambda + \mu\right) (3\Lambda^2 - 1)$$

From here we see that from the range $[-\sqrt{1/3}, \sqrt{1.3}]$, the value of the derivative is negative, and elsewhere is positive, so the function is not monotonic since there is a change of direction. This points also serve as the critical points (i.e., Λ_{crit}) where the model fails with zero stiffness.

To show that the material does not satisfy the growth condition, we take the condition for J tending to 0 from the right. Since J is the determinant of F , this value is Λ . Looking at the condition

$$\begin{aligned} \lim_{J \rightarrow 0^+} W(E) &= \lim_{\Lambda \rightarrow 0^+} \frac{\lambda}{2} (\text{tr } E)^2 + \mu \text{tr}(E^2) \\ &= \lim_{\Lambda \rightarrow 0^+} \left(\frac{\lambda}{2} + \mu\right) \left(\frac{1}{2}(\Lambda^2 - 1)\right)^2 \\ &= \frac{1}{4} \left(\frac{\lambda}{2} + \mu\right) \end{aligned}$$

So the growth condition is not satisfied.

- (6) Consider now the modified Kirchhoff Saint-Venant material model:

$$W(\mathbf{E}) = \frac{\lambda}{2} (\ln J)^2 + \mu \text{tr}(\mathbf{E}^2)$$

Does this model circumvent the drawbacks of the previous model?

The easiest part to check is the growth condition. Using the values from the previous part:

$$\begin{aligned} \lim_{J \rightarrow 0^+} W(E) &= \lim_{\Lambda \rightarrow 0^+} \frac{\lambda}{2} (\ln J)^2 + \mu \text{tr}(E^2) \\ &= \lim_{\Lambda \rightarrow 0^+} \frac{\lambda}{2} (\ln \Lambda)^2 + \mu \left(\frac{1}{2}(\Lambda^2 - 1)\right)^2 \\ &= -\infty \end{aligned}$$

So the growth condition is now satisfied. To check the monotonic part, we obtain the Piola-Kirchhoff tensor:

$$S(E) = 2\mu E$$

so replacing values for 11:

$$S_{11} = \mu(\Lambda^2 - 1)$$

and the value for P is

$$P_{11}(\Lambda) = \mu(\Lambda^3 - \Lambda)$$

which still has the same critical points, so the problem is not really circumvented.

- (7) Implement the material model in Eq.(1) in the Matlab code. Perform the consistency test to check your implementation. Try to demonstrate the material instabilities of this model with a numerical example.

The model is implemented in the MATLAB program by modifying the NeoHookean functions. Since we already have the relation for $W(E)$ and $S(E)$, we need to obtain the one for \mathbb{C} , which is given by the relation

$$\mathbb{C} = \frac{\partial S(\mathbf{E})}{\partial \mathbf{E}}$$

and this gives

$$\mathbb{C} = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix}$$

plus the In the annex the code for `StVenant_3.m` is given in order to show the procedure. The results are shown for example 4:

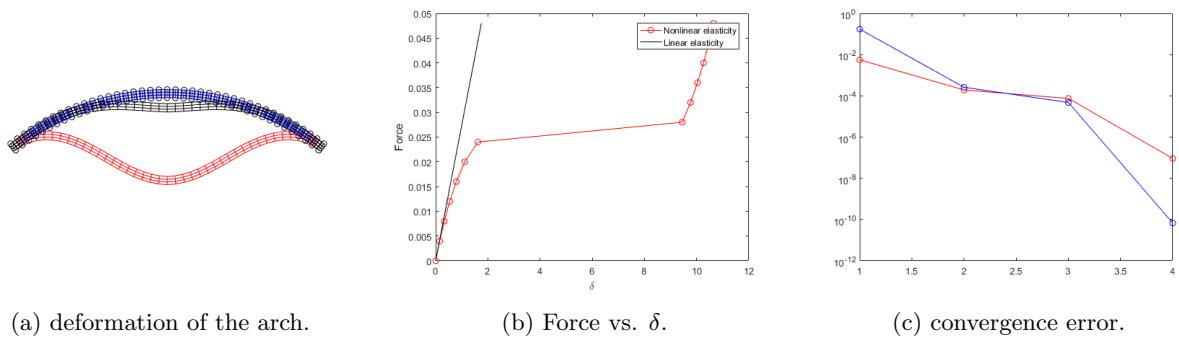


Figure 2: example 1.

for example 1:

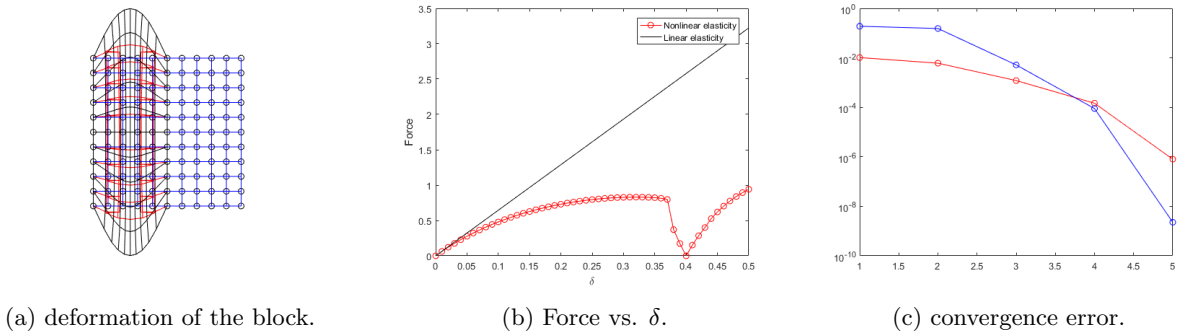
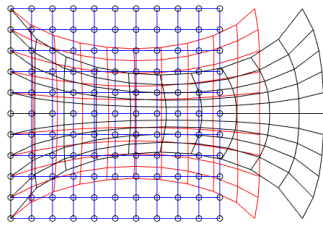
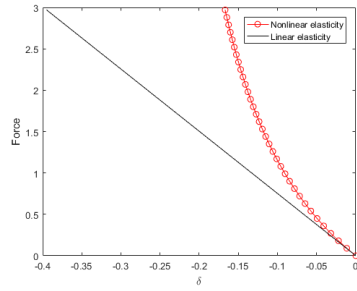


Figure 3: example 3, comparison.

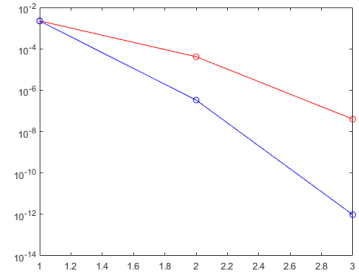
and also for example 0:



(a) deformation of the block.



(b) Force vs. δ .



(c) convergence error.

Figure 4: example 3, comparison.

As can be seen, the model presents the expected results for the St. Venant model. The deformation of the arch, as well as the one from the block are in a non-linear fashion, but the shown deformations are according to the deflection of the geometry analyzed. Also, seeing the displacement vs. force plots it can be clearly seen the point in which buckling appears, especially in the arch figure, where the increase of deformation is too big after a little increment in force.

2. Implementation of line-search.

Implement a line-search algorithm to be used in combination with Newton's method. For this, I suggest you resort to Matlab's function `fminbd`, which performs 1D nonlinear minimization with bounds. You need to define a function `Ener_1D` that evaluates the energy along the line that passes through x in the direction of p (the descent search direction). The function `LineSearch` may include lines like the ones suggested next:

```
t=1;
opts=optimset('TolX',options.TolX,'MaxIter',options.n_iter_max_LS);
t = fminbd(@(t) Ener_1D(t,x_short,p),0,2,opts);
x_short=x_short+t*p;
```

Test the code with the examples where you expect buckling (the compression of the beams, or the deflection of the arch), and compare the results with and without line-search.

The code for `LineSearch` was already implemented in the provided code, so this part will be skipped. The testing of the function is done in order to see the effectiveness of the function.

For example 3 with Neo-Hookean model, the results are shown for both simple Newton-Raphson and the one with Line-Search:

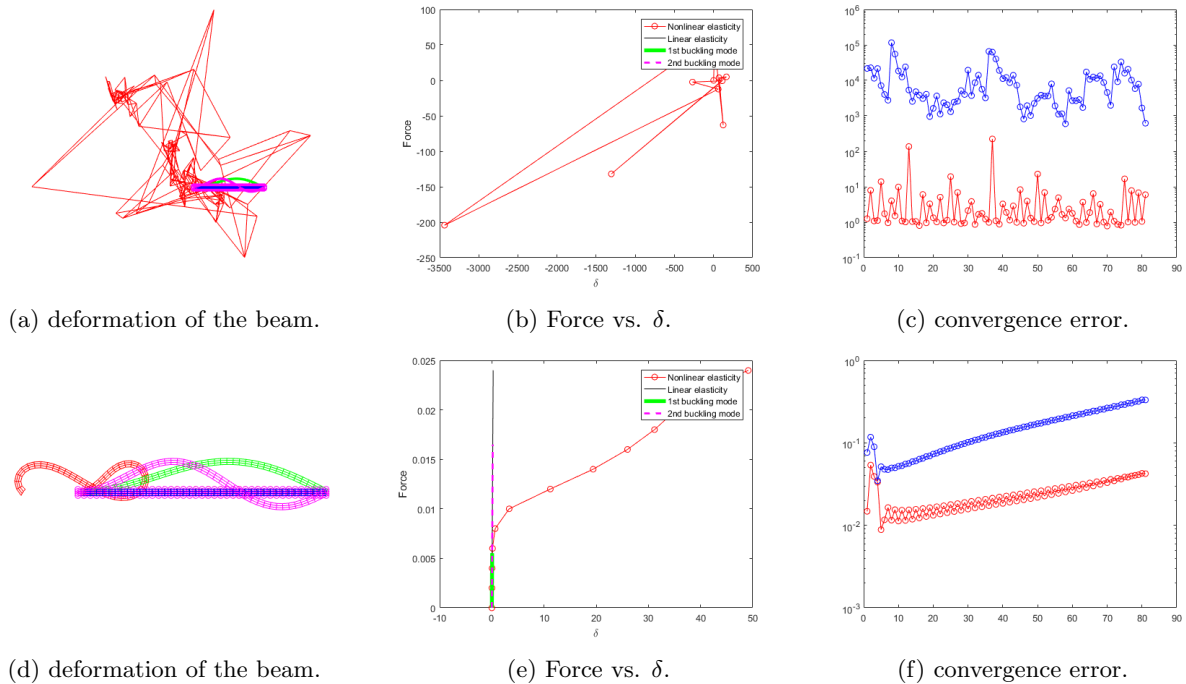


Figure 5: example 3, comparison.

For example 4, the results for both simple Newton-Raphson and the one with Line-Search:

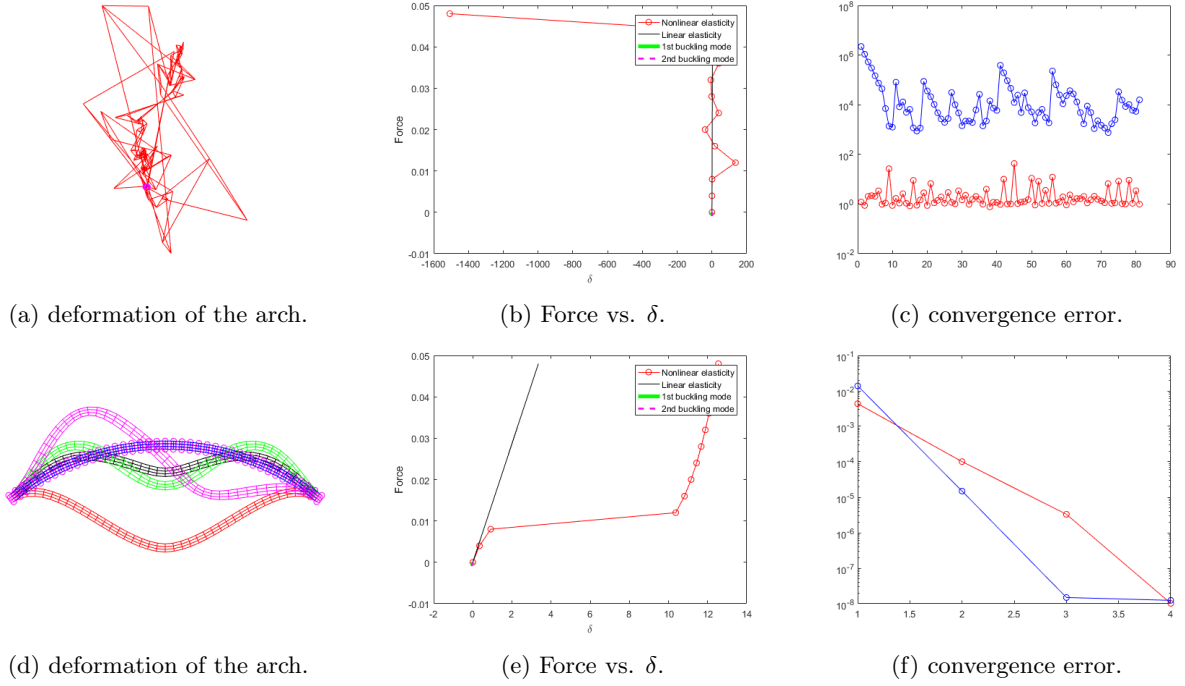


Figure 6: example 3, comparison.

And for example 5:

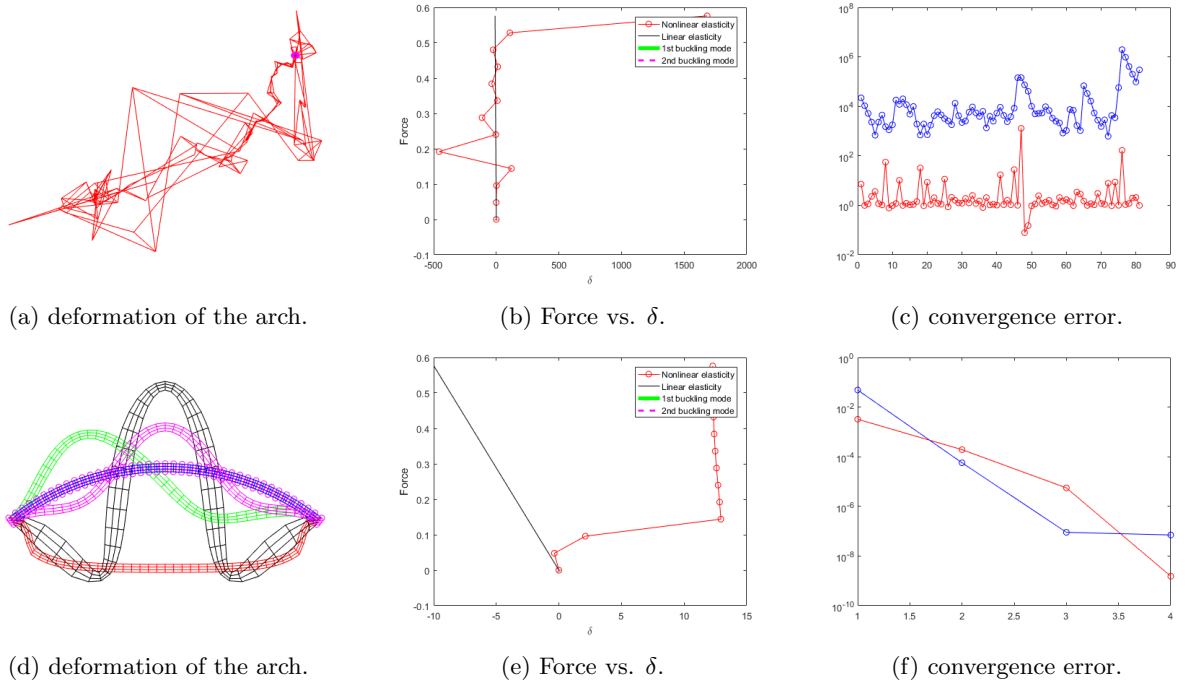


Figure 7: example 3, comparison.

The main conclusion to be extracted from all this examples is the fact that the system does not converge at all when the LineSearch algorithm is not implemented. This is mainly due to the fact that buckling is an unstable equilibrium, which at the slightest perturbation could create a movement big compared to the previous ones. This translates as an instability in the system due to singular matrices appearing at the moment of reaching the instability point, which sends the next steps into an inexistent solution.

LineSearch is an optimization tool that helps to identify the minimum energy of the system, which adds stability to the solutions, leading to the better solutions.

In the stabilized plots, it can be seen that in the deformed topography we obtain different curves, which correspond to the different modes of buckling. Each one of this deformed geometries correspond to a stable deformed shape of the beam or arch. Typically in structures buckling is considered a failure state because of the consequences that bring to the structure, but in material science this is not necessarily the case, since buckling is still an elastic case, that can be returned to the normal geometry by removing the load that is causing it. In every force vs. deformation graph shown for the stabilized case we can see perfectly the moment in which buckling is achieved, since for a very small increment of the force we get a big deformation.

Finally, it is worth noting that for the third example the convergence is not as perfect as the other cases, since we have an oscillating pattern of errors, and the error tend to increase in every step, so this is something that should be analyzed better in future applications.

3. Implementation of a material model.

The code you are given implements a plane-strain finite element method for finite deformation elasticity. A compressible Neo-Hookean material is already in place (modeling a slightly porous rubber for instance), whose strain energy density (or hyper-elastic potential) is

$$W(\mathbf{C}) = \frac{1}{2}\lambda_0(\ln J)^2 - \mu_0 \ln J + \frac{1}{2}\mu_0(\text{trace } \mathbf{C} - 3)$$

This constitutive model is isotropic. Note that, since we are considering plane strain, we can use a 2×2 reduced right Cauchy-Green deformation tensor and replace $\text{trace } \mathbf{C} - 3$ by $\text{trace } \mathbf{C} - 2$ in the above equation.

We want to consider now an anisotropic material, more specifically, a transversely isotropic material. We consider a material constitutive law for a rubber reinforced by fibers, all aligned in the same direction in such a way that perpendicular to the fibers, the material remains isotropic. The orientation of the fibers is given in the reference configuration by a unit vector \mathbf{N}^{fib} . Such a model depends on the principal invariants of \mathbf{C} , and additionally by the fourth invariant

$$I_4(\mathbf{C}) = \mathbf{N}^{fib} \cdot \mathbf{C} \cdot \mathbf{N}^{fib} = C_{IJ} N_I^{fib} N_J^{fib}.$$

More specifically,

$$W(\mathbf{C}) = \frac{1}{2}\mu_0(\text{trace } \mathbf{C} - 3) - \mu_0 \ln J + \kappa \mathcal{G}(J) + c_0 \left\{ \exp \left[c_1 (\sqrt{I_4(\mathbf{C})} - 1)^4 \right] - 1 \right\},$$

where μ_0 , κ , c_0 and c_1 are material parameters, and $\mathcal{G}(J)$ provides the volumetric response of the material. We consider

$$\mathcal{G}(J) = \frac{1}{4}(J^2 - 1 - 2 \ln J).$$

The last term in the strain energy function specifies the contribution to the deformation energy of the fibers, and as typical in biological fibers, with this model these become stiffer the more deformed they are.

- (a) Implement this material model into the code. The code is prepared for this model (`material=2`), including the definition of the material parameters in `preprocessing.m`.

The model was implemented in a similar fashion the the code in problem 1.7. We have the form for $W(\mathbf{C})$, so we obtain the Piola tensor by using the formula

$$\mathbf{S} = 2 \frac{\partial W}{\partial \mathbf{C}}$$

and the constitutive tensor by

$$\mathbb{C} = 2 \frac{\partial \mathbf{S}}{\partial \mathbf{C}}$$

From class we have some derivatives for basic matrices and invariants. Using this, we reach to the form for the Piola tensor:

$$\mathbf{S} = \mu_0(\mathbb{I} - \mathbf{C}^{-1}) + \frac{1}{2}\kappa(J^2 - 1)\mathbf{C}^{-1} + 4c_0c_1 \frac{1}{\sqrt{I_4(\mathbf{C})}} (\sqrt{I_4} - 1)^3 \exp \left[c_1 (\sqrt{I_4} - 1)^4 \right] \mathbf{N}_{fib} \otimes \mathbf{N}_{fib}^T$$

and deriving again for \mathbb{C}

$$\begin{aligned} \mathbb{C} = & \mu_0(C_{IK}^{-1}C_{JL}^{-1} + C_{IL}^{-1}C_{JK}^{-1}) + \kappa(J^2C_{IJ}^{-1}C_{KL}^{-1} - \frac{1}{2}(J^2 - 1)(C_{IK}^{-1}C_{JL}^{-1} + C_{IL}^{-1}C_{JK}^{-1})) + \dots \\ & \dots + 4c_0c_1 \exp \left[c_1 (\sqrt{I_4} - 1)^4 \right] (\sqrt{I_4} - 1)^2 N_I N_J \left[\frac{3}{I_4} - \frac{1}{\sqrt{I_4^3}}(\sqrt{I_4} - 1) + \frac{4c_1}{\sqrt{I_4}}(\sqrt{I_4} - 1)^4 \right] N_K N_L \end{aligned}$$

In the annex the code for this model is shown for reference under the name `transv_isotr_3.m`.

- (b) Check the correctness (consistency test) of your implementation by running the script `Check_Derivatives.m` with `material=2`. This script checks the gradient of the energy (out-of-balance forces) and the Hessian of the energy (tangent stiffness matrix) by numerical differentiation. Check also that when solving a mechanical problem with this mode, Newton's method converges quadratically.

Running the code through `Check_Derivatives.m` we obtain a convergence of the gradient and Hessian for every step, so we can safely assume that the model is working correctly. Furthermore, we run example number 3 in order to see the convergence of the error:

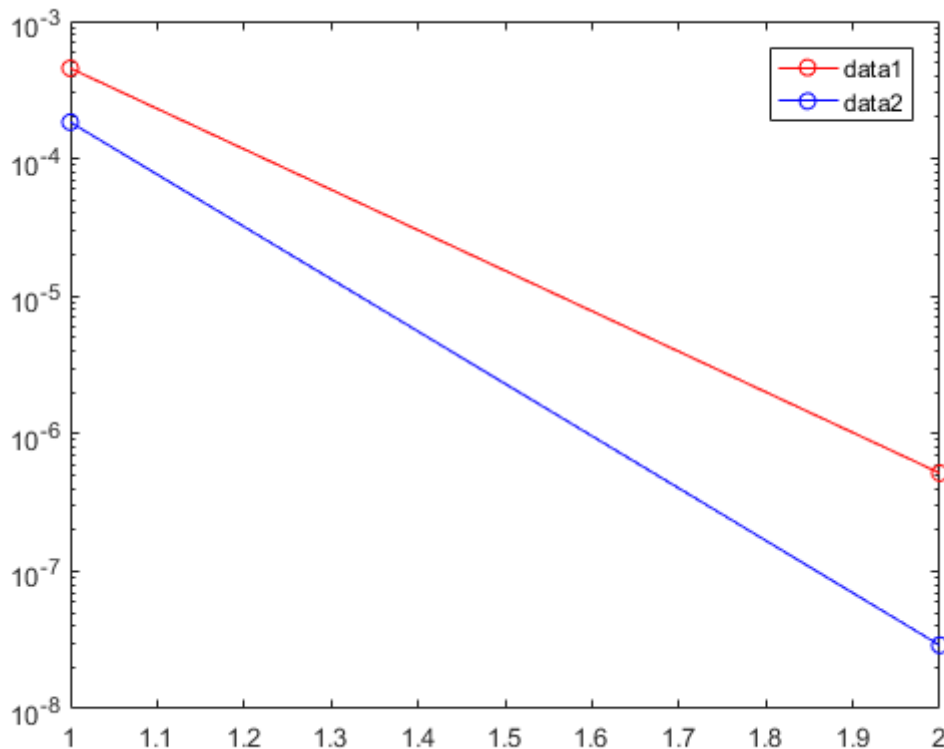


Figure 8: convergence of error for example 3 for $\theta = 0$.

where we see that we have a linear dependence of the error, whose slope should be of value 2.

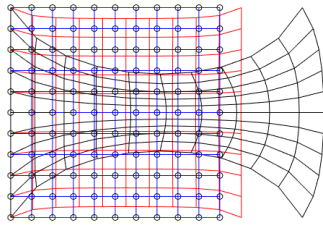
- (c) Solve `example=0`, a dead load applied on an elastic block in tension, with a few representative orientations of the fibers. Consider $\theta = 0$ (fibers aligned with the loading direction), $\theta = \pi/6$, $\theta = \pi/4$ and $\theta = \pi/2$ (fibers perpendicular to the loading direction), where

$$\mathbf{N}^{fib} = [\cos \theta, \sin \theta]^T.$$

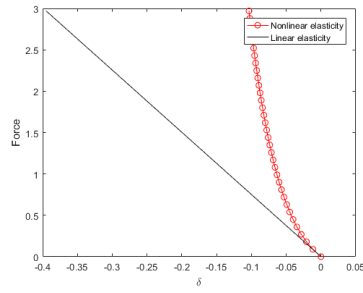
Explain the results from a mechanical viewpoint.

The plots for every angle are shown below.

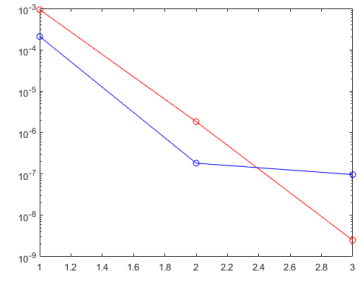
For $\theta = 0$:



(a) deformation of the block.



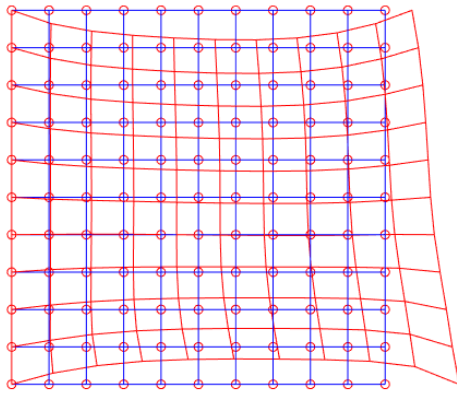
(b) Force vs. δ .



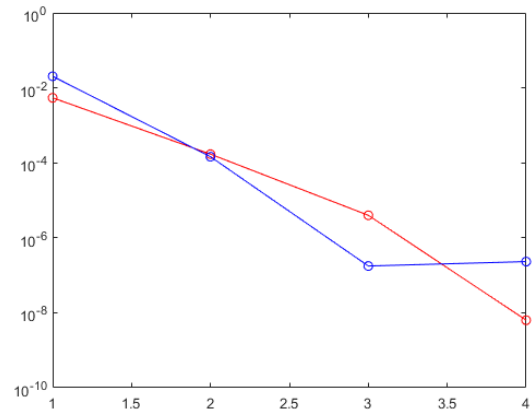
(c) convergence error.

Figure 9: example 0, $\theta = 0$.

For $\theta = \pi/6$:



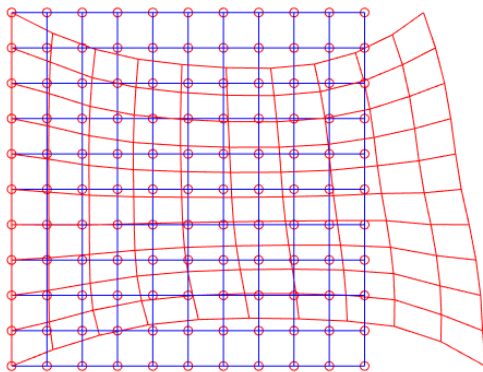
(a) deformation of the block.



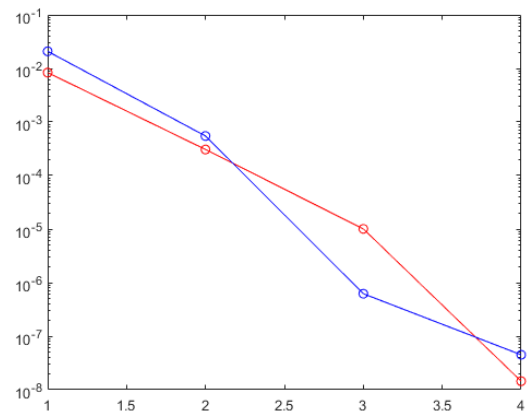
(b) convergence error.

Figure 10: example 0, $\theta = \pi/6$.

For $\theta = \pi/4$:



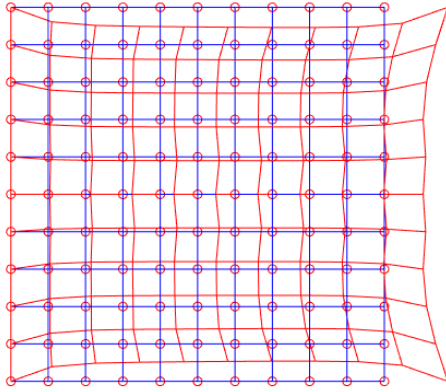
(a) deformation of the block.



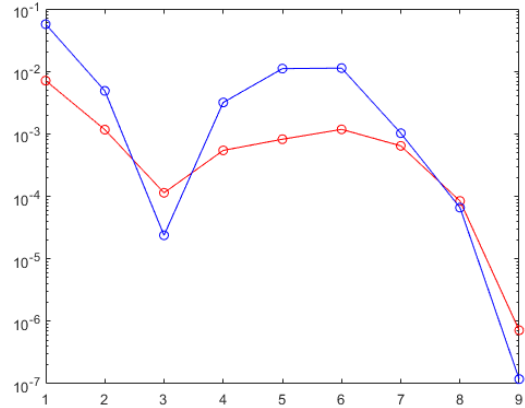
(b) convergence error.

Figure 11: example 0, $\theta = \pi/4$.

And for $\theta = \pi/2$:



(a) deformation of the block.



(b) convergence error.

Figure 12: example 0, $\theta = \pi/2$.

As can be seen, the model converges perfectly when $\theta = 0$, but there are some convergence problems for any other value of the angle. This could be explained by the fact that we are considering an anisotropic model for the material, which has different properties according to the rotation of the material. Since the force we are applying is rotating, this causes instabilities on the code for the values in which the displacement is not considered, leading to the results. In the deformation plots can be seen the effect of the force oriented in other angles. For $\theta = 0$, we see that the material is stretched perfectly uniaxially, but for the other cases we have a kind of rotation, that is given by the shear stresses applied to the material.

Even though the algorithm has problems with convergence, we still see that we have a decreasing error, expect for the $\pi/2$ case, were there are some oscillation to it.

Annex

Kirchhoff-St.Venant model

```
function [W,S,CC]=StVenant_3(C,lambda,mu,icode)

E    = 0.5*([C(1) C(3); C(3) C(2)]-eye(2));
trE  = trace(E);
trE2 = trace(E*E);
W    = 0.5*lambda*trE^2 + mu*trE2;

S = lambda*trE*eye(2) + 2*mu*E;
S = [S(1,1) S(2,2) S(1,2)];

CC = zeros(3);
CC(1,1) = lambda+2*mu;
CC(2,2) = lambda+2*mu;
CC(1,2) = lambda;
CC(2,1) = lambda;
CC(3,3) = mu;
```

Anisotropic model

```
function [W,S,CC]=transv_isotr_3(C,c0,c1,kappa,mu,N_fib)

Cm = [C(1) C(3);C(3) C(2)];
J2 = C(1)*C(2)-C(3)*C(3);
J = sqrt(J2);
logJ = log(J);
trC = C(1)+C(2);
gJ = 0.25*(J2-1-2*logJ);
I4 = N_fib'*Cm*N_fib;

W = 1/2*mu*(trC-2) - mu*logJ + kappa*gJ + c0*(exp(c1*(sqrt(I4)-1)^4)-1);

C_inv = [C(2) C(1) -C(3)]/J2;
C_invm = [C_inv(1) C_inv(3);C_inv(3) C_inv(2)];

NN = N_fib*N_fib';

S = mu*(eye(2)-C_invm)+0.5*kappa*(J2*C_invm-C_invm)+4*c0*c1*(sqrt(I4)-1)^3*
(sqrt(I4))^( -1)*exp(c1*(sqrt(I4)-1)^4)*NN;
S = [S(1,1) S(2,2) S(1,2)];

CC(1,1) = mu*(C_inv(1)*C_inv(1)+C_inv(1)*C_inv(1))+kappa*(J2*C_inv(1)*C_inv(1)-0.5*(J2-1)*
(C_inv(1)*C_inv(1)+C_inv(1)*C_inv(1)))+4*c0*c1/I4*(exp(c1*(sqrt(I4)-1)^4))*(sqrt(I4)-1)^2*(3-
I4^(-1/2))*(sqrt(I4)-1)+4*c1*(sqrt(I4)-1)^4)*NN(1,1)*NN(1,1);
CC(1,2) = mu*(C_inv(3)*C_inv(3)+C_inv(3)*C_inv(3))+kappa*(J2*C_inv(1)*C_inv(2)-0.5*(J2-1)*
(C_inv(3)*C_inv(3)+C_inv(3)*C_inv(3)))+4*c0*c1/I4*(exp(c1*(sqrt(I4)-1)^4))*(sqrt(I4)-1)^2*(3-
I4^(-1/2))*(sqrt(I4)-1)+4*c1*(sqrt(I4)-1)^4)*NN(1,2)*NN(1,2);
CC(1,3) = mu*(C_inv(1)*C_inv(3)+C_inv(1)*C_inv(3))+kappa*(J2*C_inv(1)*C_inv(3)-0.5*(J2-1)*
(C_inv(1)*C_inv(3)+C_inv(1)*C_inv(3)))+4*c0*c1/I4*(exp(c1*(sqrt(I4)-1)^4))*(sqrt(I4)-1)^2*(3-
I4^(-1/2))*(sqrt(I4)-1)+4*c1*(sqrt(I4)-1)^4)*NN(1,1)*NN(1,2);
CC(2,2) = mu*(C_inv(2)*C_inv(2)+C_inv(2)*C_inv(2))+kappa*(J2*C_inv(2)*C_inv(2)-0.5*(J2-1)*
(C_inv(2)*C_inv(2)+C_inv(2)*C_inv(2)))+4*c0*c1/I4*(exp(c1*(sqrt(I4)-1)^4))*(sqrt(I4)-1)^2*(3-
I4^(-1/2))*(sqrt(I4)-1)+4*c1*(sqrt(I4)-1)^4)*NN(2,2)*NN(2,2);
CC(2,3) = mu*(C_inv(2)*C_inv(3)+C_inv(2)*C_inv(3))+kappa*(J2*C_inv(2)*C_inv(3)-0.5*(J2-1)*
(C_inv(2)*C_inv(3)+C_inv(2)*C_inv(3)))+4*c0*c1/I4*(exp(c1*(sqrt(I4)-1)^4))*(sqrt(I4)-1)^2*(3-
I4^(-1/2))*(sqrt(I4)-1)+4*c1*(sqrt(I4)-1)^4)*NN(2,2)*NN(1,2);
CC(3,3) = mu*(C_inv(1)*C_inv(2)+C_inv(3)*C_inv(3))+kappa*(J2*C_inv(3)*C_inv(3)-0.5*(J2-1)*
(C_inv(1)*C_inv(2)+C_inv(1)*C_inv(2)))+4*c0*c1/I4*(exp(c1*(sqrt(I4)-1)^4))*(sqrt(I4)-1)^2*(3-
I4^(-1/2))*(sqrt(I4)-1)+4*c1*(sqrt(I4)-1)^4)*NN(1,2)*NN(1,2);
CC(2,1) = CC(1,2);
CC(3,1) = CC(1,3);
CC(3,2) = CC(2,3);
```