Technical University of Catalonia

COMPUTATIONAL SOLID MECHANICS

ASSIGNMENT 3

FINITE DEFORMATION NONLINEAR ELASTICITY

M.Sc. Computational Mechanics – CIMNE

Arzu Ahmadova

Mohammad Mohsen Zadehkamand

9 June 2017

Table of Contents

1. Kirchhoff Saint-Venant material model

1.1. Stress tensor

This Isotropic linear elastic can be derived from balance of linear momentum, the linearized strain displacement relation, and the stored elastic energy function

$$\varepsilon = \frac{1}{2}(\nabla u + \nabla u^T)$$

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

Using the linear elastic expression $\sigma = C:\varepsilon$ written as $\sigma = \lambda(tr\varepsilon)1 + 2\mu\varepsilon$ we can write the previous formula as:

$$\sigma_{ij} = \frac{\partial}{\partial\varepsilon_{ij}}\left(\frac{\lambda}{2}(\varepsilon_{kk})^2 + \mu\varepsilon_{ij}\varepsilon_{ij}\right) = \lambda\varepsilon_{kk}\delta_{ij} + 2\mu\varepsilon_{ij}$$

1.2. Isotropic verification

The definition of isotropic model is that any isotropic strain energy function can be written in terms of the principal invariants of C.

$$W(C) = W(I_1(C), I_2(C), I_3(C))$$

With

$$I_1(C) = traceC, \quad I_2(C) = \frac{1}{2}[(traceC)^2 - traceC^2], \quad I_3(C) = \det C = J^2$$

We here assume the material is isotropic and we could write the energy equation W in terms of the principal invariants of C.

$$\frac{\partial W}{\partial C} = \frac{\partial W}{\partial I_1}\frac{\partial I_1}{\partial C} + \frac{\partial W}{\partial I_2}\frac{\partial I_2}{\partial C} + \frac{\partial W}{\partial I_3}\frac{\partial I_3}{\partial C}$$

$$\frac{\partial W}{\partial C} = c_1 I + c_2(I_2(CI - C^T)) + c_3 I_3 C^{-T}$$

Grouping different terms

$$\frac{\partial W}{\partial C} = p_0 I + p_1 C^T + p_2 C^{-T}$$

With $C^{-1} = C^{-1}(I, C, C^2)$ continually, we group the last equation again,

$$\frac{\partial W}{\partial C} = p_0 I + p_1 C^T + p_2 C^2$$

As we know that the PK-2 stress tensor(S) can be yielded from above equation as

$$S=2\frac{\partial W}{\partial C}=2(p_0 I+p_1 C^T+p_2 C^2)$$

On the other hand, the isotropic Cauchy stress tensor has the form

$$\sigma=2\rho(s_1 I+s_2 B+s_3 B^2)$$

with B as a right Cauchy-Green deformation tensor. We can compare the last two equations sharing the same form, thereby concluding that the original model is isotropic. Actually, on the other hand, we could write

$$W(I_1,I_2)=$$

$$=\frac{1}{2}(\lambda+2\mu)[\frac{1}{2}(I_1-3)]^2-2\mu[\frac{1}{4}(-2I_1+I_2+3)]=\frac{1}{8}(\lambda+2\mu)(I_1-3)^2-\frac{\mu}{2}(-2I_1+I_2+3)$$

We again prove that the model is isotropic model!

1.3. Second Piola-Kirchhoff stress

The second Piola-Kirchhoff stress follows the next equation:

$$S=\frac{\partial W}{\partial E}$$

And after differentiating the energy equation we obtain

$$S=\lambda tr(E)1+2\mu E$$

1.4. Nominal stress P(Λ)

According to the known condition, we could compute F

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

With the definition of 2nd Piola-Kirchhoff stress: $S = F^{-1}P$. We can compute the nominal normal stress P=FS. Especially, the xX component of the first Piola-Kirchhoff stress and the stretch ratio Λ ,as is shown is Figure 1.

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

$$P=\Lambda\ \lambda(trE)+2\Lambda\mu E$$

And the first nominal stress is given by:

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

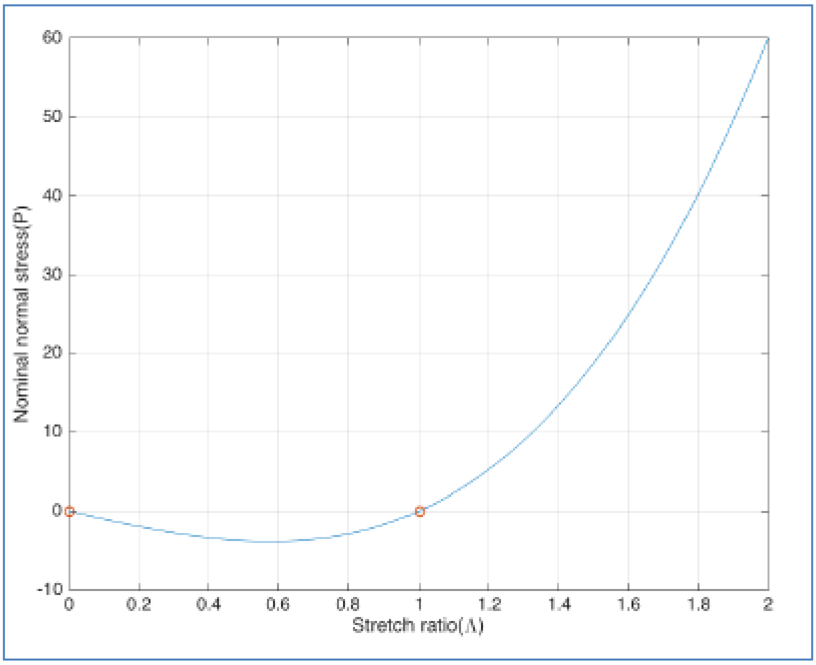

Figure 1: The nominal normal stress P with respect to the stretch ratio

1.5. *Monotonic verification*

The relation P(Λ) is not monotonic as is shown in Figure1. When $\Lambda \leq$ 0.5, P(Λ) shows a decreasing trends while later it increase to zero and then goes up monotonically. With the last equation, we could compute the derivative of P with respect to Λ

$$\frac{dP}{d\Lambda} = \frac{3\Lambda^2 - 1}{2}(\lambda + \mu) = 0$$

Hence, the point fails with zeros stiffness which does not dependent on the elastic constant,

$$\Lambda_{crit} = \frac{1}{\sqrt{3}}$$

We know that J=detF=Λ. When J$\rightarrow 0^+$,meaning that $\Lambda \rightarrow 0^+$, we could obtain the expression W in terms of Λ

$$W = \frac{1}{2}(trE)^2 + \mu trE^2 = \frac{1}{4}(\frac{\lambda}{2} + \mu)$$

When $\Lambda \rightarrow 0^+$, W$\rightarrow \frac{1}{4}(\frac{\lambda}{2} + \mu)$

1.6. Modifed Kirchhoff Saint-Venant material model

With the modified Kirchhoff Saint-Venant material model:

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

and noting that J= $\sqrt{\det E}$, especially in 1D case , J= $\sqrt{\det C}=\det(2E+1)$,

$$tr(E^2)=E^2 \quad \text{with} \quad E=\frac{1}{2}(\Lambda^2-1)$$

It endowed as $\rightarrow$ $$W(E)=\frac{\lambda}{8}(\ln(2E+1))^2+\mu E^2$$

The second Piola-Kirchhoff stress is:

$$\frac{\partial W}{\partial E}=\frac{\lambda}{8}\frac{\ln(2E+1)}{2E+1}+2\mu E=\frac{\lambda}{8}\frac{\ln\Lambda}{\Lambda}+\mu(\Lambda^2-1)$$

Thus the nominal normal stress is

$$P(\Lambda)=FS=\frac{\lambda}{8}\ln\Lambda+\mu\Lambda(\Lambda^2-1)$$

can also be written in term of Λ

$$W=\frac{1}{2}(\ln\Lambda)^2+\frac{\mu}{4}(\Lambda^2-1)^2$$

P(Λ) is monotonic. However, when $\Lambda=1, P(\Lambda)=0,$ which follows the actual fact that there is no stress for undeformed configuration. On the other hand, when J $\rightarrow 0^+$ the meaning that $\Lambda\rightarrow 0^+$, we have W $\rightarrow +\infty$.

1.7. Implementation of Kirchhoff Saint-Venant material model

The Matlab function called KSV (Kirchhoff Saint-Venant model) has been implemented in the given code. The function computes the following tasks:

- ✓ Green-Lagrange deformation tensor (E)
- ✓ Stored elastic energy (W)
- ✓ Piola-Kirchhoff stress tensor (S)
- ✓ Tangent elastic constitutive tensor (C)

Once the function has been implemented, the consistency test has been performed with the **Check derivatives.m** function. In order to check the problem a new material was created and linked with the KSV function. The new material has the following properties:

μ = 0.9, λ = 150, *potential* = 2

The example run was *upsetting of a block, dead load and* the consistency test gave us an relative error of 0.0098 %. **Figure 2a** shows the deformation shape for a linear and non-linear case. It can be noticed how the non-linear response gets unstable. **Figure 2b** shows the relative error (logaritmic scale) in term of number of iterations and it can be point out how the non-linear model converges. From **Figure 2c** the non-linear behaviour is shown: it can be appreciated how the force is linear for a small deformation and then, after almost a constant response, it increases drastically.

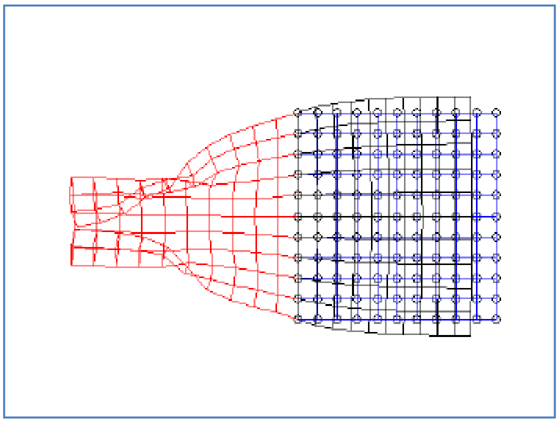

Figure2a. Deformation mesh in KSV model

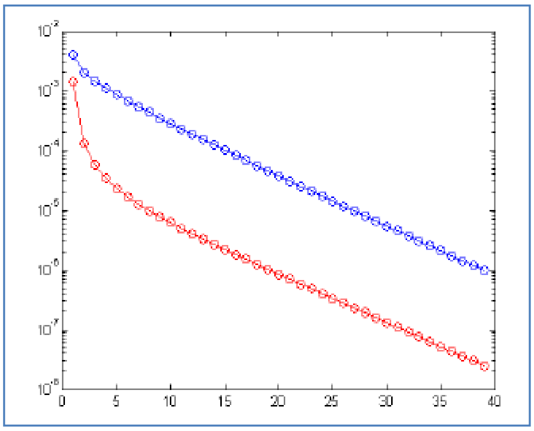

Figure2b. Log(relative error) - iterations KSV model

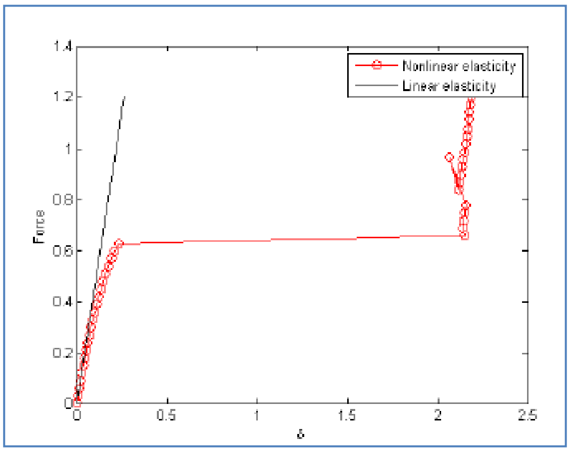

Figure2c . Force vs deformation diagram KSV model

2. Implementation of line-search

The goal of this part is to study the effect of implementing a line-search algorithm to be used in combination with Newton's method. This algorithm is already implemented in the code. For this, we resort to Matlab's function **fminbd**, which performs 1D nonlinear minimization with bounds. Besides, function **Ener_1D** is defined that evaluates the energy along the line that passes through x in the directions (the descent search direction). Firstly we summarize the algorithm of Newton's method with line search proceeds as:

- Solve $J(x^k)\Delta x^k = -r(x^k)$
- Consider an energy-descent search direction, $s^k = \Delta x^k$ if $x^k \le 0$
  and $s^k = -\Delta x^k$ if $r(x^k)^T > 0$
- Solve the 1D minimization problem $\min_\alpha \Pi(x^k + \alpha s^k)$
- Update $x^{k+1} = x^k + \alpha s^k$

The code with line-search method is verified with example 4 (arch, dead load at center of the arch). The results using normal Newton's method and normal Newton's method with line search are shown in the **Figure3**. It is obvious that we are unable to obtain the deformed arch with normal Newton's method. It is thus advisable to use line-search for help. For the same tolerance, with line-search, the solution is converged at 4 steps while that for Newton's method is 80 steps. Moreover, if the deformed shape, **Figure 3b** is studied, it can be seen how buckling phenomena appears yielding a stable equilibrium situation in which all the eigenvalues are positive. Furthermore, it is also interesting to stress how the force-displacement curve changes due to buckling, **Figure 4**.

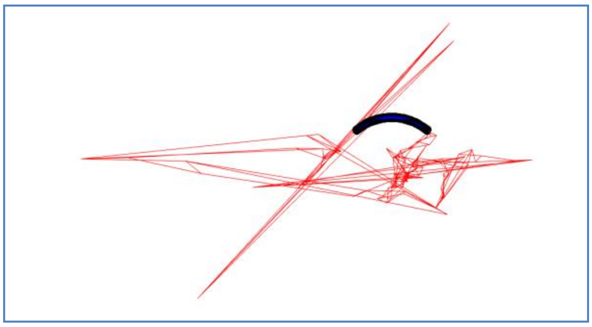

Figure3a. Deformation mesh without line search

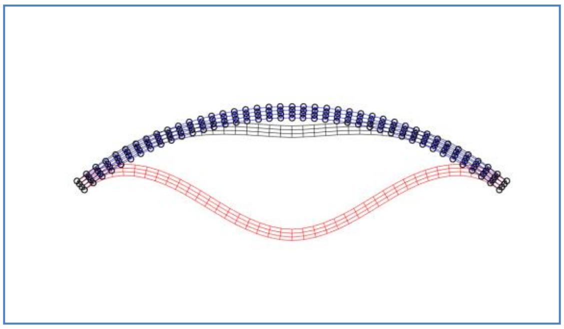

Figure3b. Deformation mesh with line search

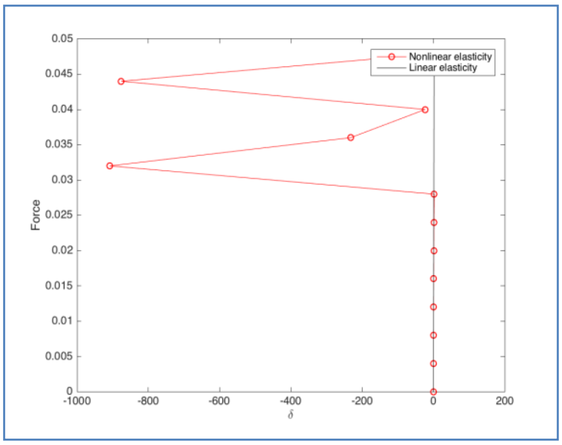

Figure4a. Force vs displacement without line search

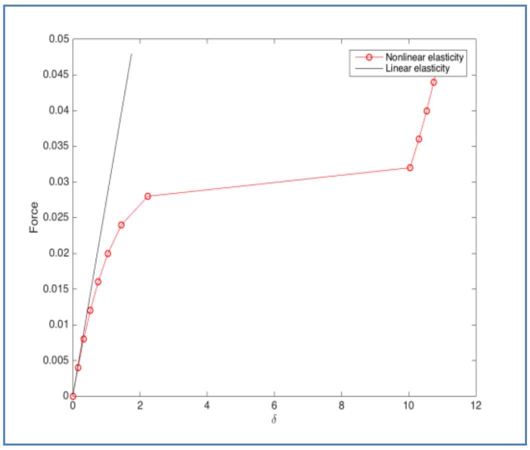

Figure4b. Force vs displacement with line search

3. Implementation of an anisotropic material model

3.1. Formulation

The original code implements a plane-strain finite element method for finite deformation elasticity using a compressible **isotropic Neo-Hookean material** (namely, for modeling a slightly porous rubber), whose strain energy density or hyper-elastic potential is:

$$W(C) = \frac{1}{2}\lambda(lnJ)^2 - \mu\, lnJ + \frac{1}{2}\mu(tr\, C - 2)$$

The goal of this part is to consider an anisotropic material, more specifically, a **transversely isotropic material**. For this case consider a material constitutive law for a rubber reinforced by fibers, all aligned in the same direction. Such a model depends on the principal invariants of C, and additionally by the fourth invariant:

$$W(C) = \frac{1}{2}\mu(tr\, C - 2) - \mu\, lnJ + k\, G(J) + c_0(e^{c_1(\sqrt{I4}-1)^4} - 1)$$

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. In this formula G(J) provides the volumetric response of the material and I4(C) the fourth invariant.

$$G(J) = \frac{1}{4}(J^2 - 1 - 2\, lnJ) \qquad and \qquad I4(C) = N^{fib}.C.N^{fib}$$

The orientation of the fibers is given in the reference configuration by a unit vector

$$N^{fib} = \begin{bmatrix} cos\,\theta \\ sin\theta \end{bmatrix}$$

So, using the main theory provided in class we may find the 1st derivative of strain energy density as the 2nd Piola-Kirchhoff stress tensor

$$S = 2\frac{\partial W}{\partial C} = \mu(Id - C^{-1}) + \frac{k}{2}C^{-1}(J^2 - 1) + 4c_0c_1e^{c_1(\sqrt{I4}-1)^4}\frac{(\sqrt{I4}-1)^3}{\sqrt{I4}}NTN$$

Where

$$NTN = \frac{\partial I4(C)}{\partial C} = \left[{N_1^{fib}}^2 \quad , \quad {N_2^{fib}}^2 \quad , \quad N_1^{fib}N_2^{fib}\right] in\, Voigt\, notation$$

And for the 4th order tangent elasticity tensor we will obtain 2nd derivative of strain energy density

$$CC = 4\frac{\partial W}{\partial C} = \left[\mu - \frac{k}{2}(J^2 - 1)\right]\left(-2\frac{\partial C^{-1}}{\partial C}\right) + kJ^2C^{-1}C^{-1} +$$

$$+8c_0c_1NTN.NTNe^{c_1(\sqrt{I4}-1)^4}[2C_1\frac{(\sqrt{I4}-1)^6}{I4} + 1 - \frac{3}{2\sqrt{I4}} + \frac{1}{2\sqrt{I4}^3}]$$

Where

$$\left(\frac{\partial C^{-1}}{\partial C}\right)_{IJKL} = -\frac{1}{2}\left(C_{IK}^{-1}C_{JL}^{-1} + C_{IL}^{-1}C_{JK}^{-1}\right)$$

So, by using the prepared code (***material=2***), we should complete the code by providing 3 functions for the anisotropic model, just like the already implemented isotropic model (**transv_isotr_1.m**, **transv_isotr_2.m**, **transv_isotr_3**). It is worthwhile mentioning that For the main file we are using the **main_incremental_iterative.m**. Moreover, all modifications which are added to model are provided in the ***Appendix***.

3.2. Checking derivatives

In this part using the script **Check_Derivatives.m** with **material=2** and **example=0** we have checked the correctness (consistency test) of our implementation. This script checks the gradient of the energy (out-of-balance forces) and the Hessian of the energy (tangent stiffness matrix) by numerical differentiation. If any error or mistake is in the definition of the gradient or the hessian of energy function, this code would alert lots of warnings.

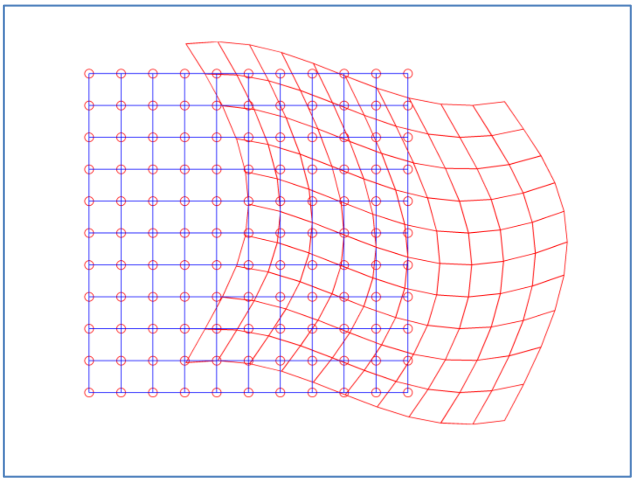

Figure 5. non symmetric and strange deformed shape for checking gradient and hessian

3.3. Numerical simulation

As a numerical simulation we have solved **example=0**, which is the modeling of dead load applied on an elastic block in tension. Different representative orientations of the fibbers are considered to be able to capture the effect and whole behavior of the model.

- $\Theta = 0$ fibers aligned with the loading direction
- $\Theta = \pi/6$
- $\Theta = \pi/4$
- $\Theta = \pi/2$ fibers aligned perpendicular to the loading direction

Figure6 is related to the numerical results of this test. The mechanical interpretation of the numerical results is provided as the following.

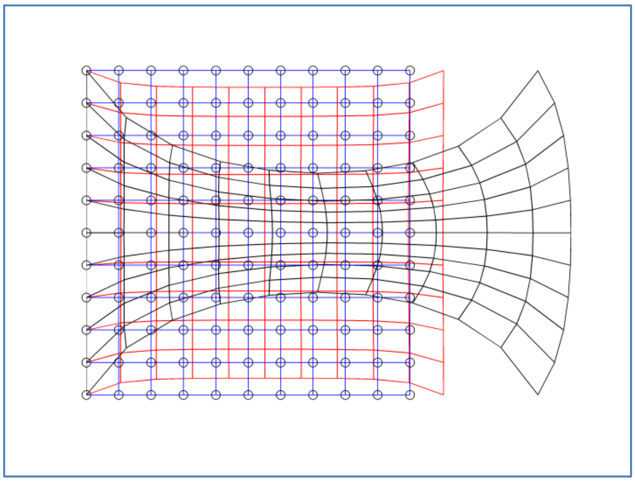

Figure 6a. deformed shape (θ = 0)

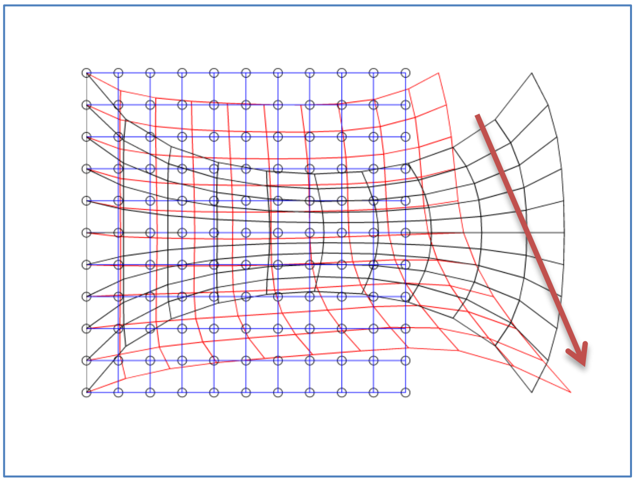

Figure 6b. deformed shape (θ = 30 deg)

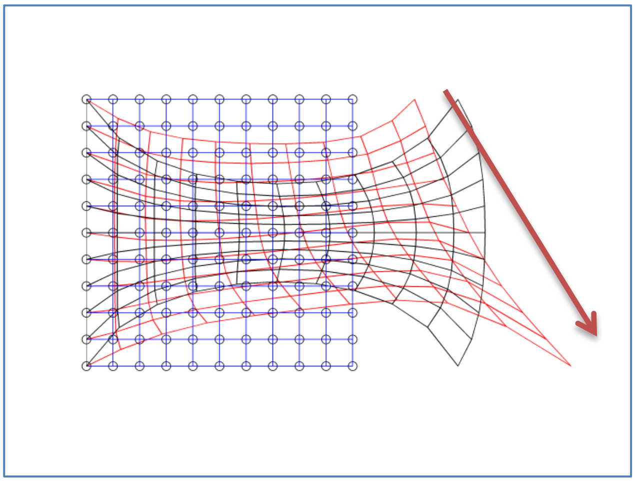

Figure 6c. deformed shape (θ = 45 deg)

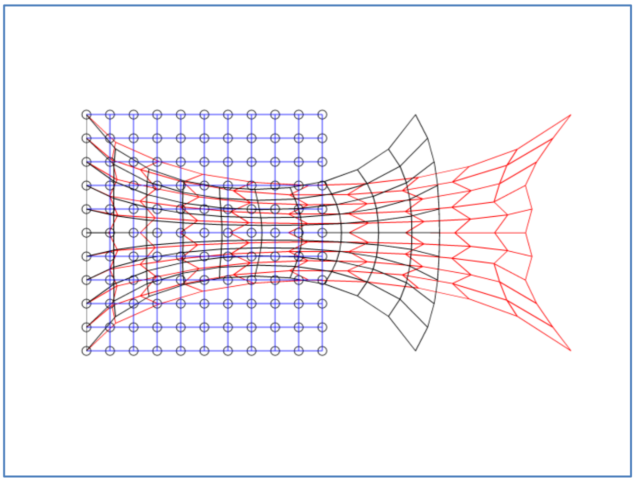

Figure 6d. deformed shape (θ = 90 deg)

As it was supposed the more fibers are aligned in the direction of loading, the stiffer the behavior would be. This fact is completely manifested in **Figure6**. As we the lowest deformation is regarded to the first case in which the fibers are aligned with the loading direction. By increasing a bit the deviation of fiber orientation from the loading directions (namely in 2^{nd} or 3^{rd} case) we may see two effects:

- Non-symmetric deformed shape: due to the non-symmetric nature of loading and fiber orientation in this anisotropic model (the lower corner is deformed more that the upper corner which is completely rational).
- The higher overall deformation: due to decreased stiffness because of non-aligned fibers in load direction.

It is also interesting to note that the final case in which the fibers are aligned perpendicular to the loading direction, acts completely symmetric (as it was supposed to) and the deformed shape is the most exaggerated one comparing to previous cases.

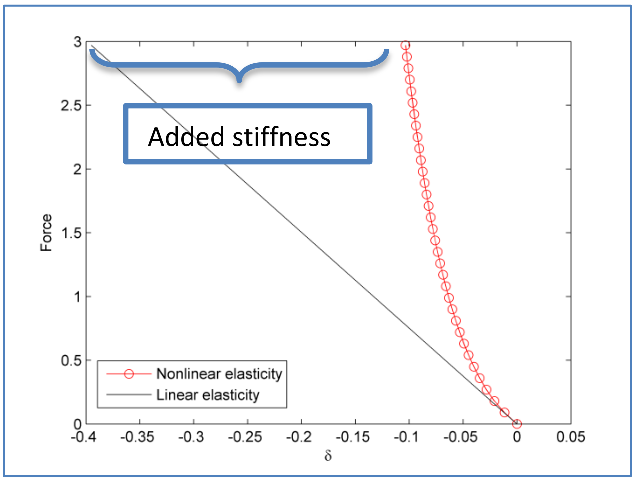

Figure 7a. force-deflection (θ = 0)

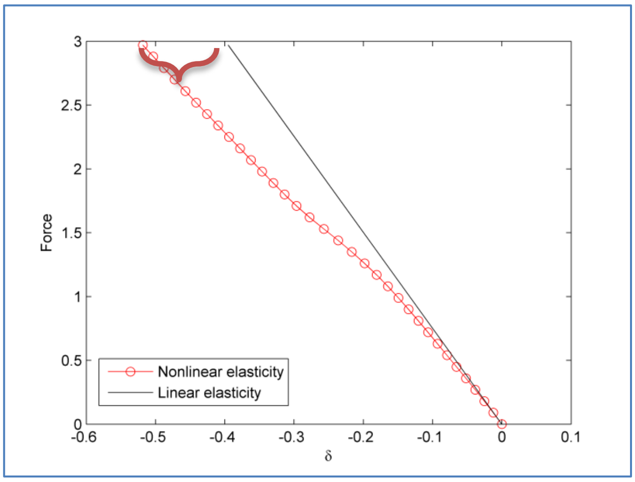

Figure 7b. force-deflection (θ = 30 deg)

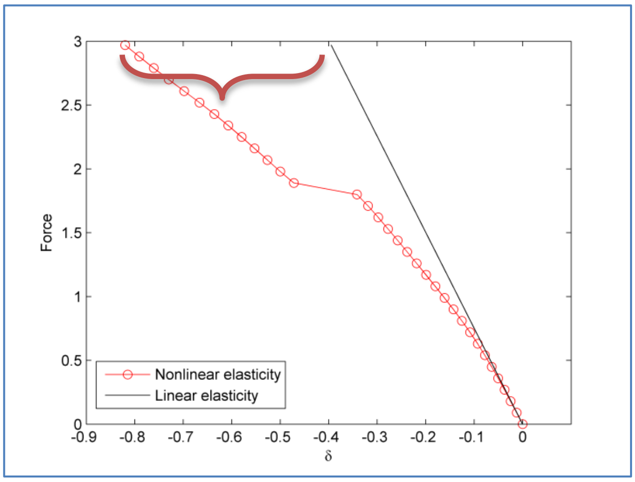

Figure 7c. force-deflection (θ = 45 deg)

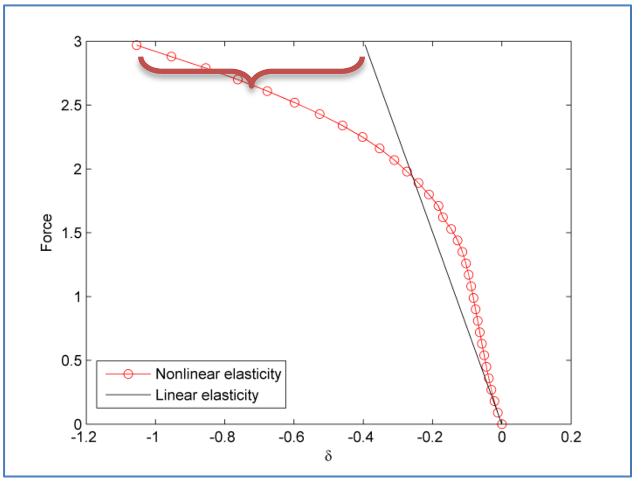

Figure 7d. force-deflection (θ = 90 deg)

As it was mentioned in previous figures, one would expect that by increasing the degree between the direction of fiber orientation and the loading vector, the overall deformation would be higher. This is because when the fibers are aligned exactly in the direction of load, they would increase significantly the stiffness of the model. This fact is completely manifested in **Figure7**. As we the marks on the figures show, the black line is the expected elastic behavior without any fiber. In all cases it is fixed to a value of around 0.4. but as we may see only in the first case which (θ=0), the stiffness is improved (blue mark). But in the rest of cases (red marks) the stiffness is less than the elastic model and as higher the deviation ratio goes the difference of the elastic and hyper elastic results increases also.

❖ *Appendix: Code*

1. transv_isotr_1

```
function [W]=transv_isotr_1(C,c0,c1,kappa,mu,N_fib)

J2=C(1)*C(2)-C(3)*C(3);
J=sqrt(J2);
logJ=log(J);

Cmat=[C(1) C(3) ; C(3) C(2)];

GJ   = 1/4* (J^2 - 1 - 2*logJ);
I4C  = N_fib' * Cmat * N_fib;
expp = exp(c1*(sqrt(I4C)-1)^4);

W = 1/2*mu*(C(1)+C(2)-2) - mu * logJ + kappa * GJ + c0 * (expp-1);
```

strain energy

2. transv_isotr_2

```
function [W,S]=transv_isotr_2(C,c0,c1,kappa,mu,N_fib)

J2=C(1)*C(2)-C(3)*C(3);
J=sqrt(J2);
logJ=log(J);

Cmat=[C(1) C(3) ; C(3) C(2)];

GJ   = 1/4* (J^2 - 1 - 2*logJ);
I4C  = N_fib' * Cmat * N_fib;
expp = exp(c1*(sqrt(I4C)-1)^4);

W = 1/2*mu*(C(1)+C(2)-2) - mu * logJ + kappa * GJ + c0 * (expp-1);

S =[];

C_inv=[C(2) C(1) -C(3)]/J2;
NTN = [N_fib(1)*N_fib(1) , N_fib(2)*N_fib(2) , N_fib(1)*N_fib(2)];

S = mu*([1 1 0]-C_inv) + kappa/2*(J^2-1)*C_inv + ...
    + 4 * NTN * c0 * c1 * expp * (sqrt(I4C)-1)^3/sqrt(I4C);
```

strain energy

1st derivative of strain energy

3. transv_isotr_3

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

J2=C(1)*C(2)-C(3)*C(3);
J=sqrt(J2);
logJ=log(J);

Cmat=[C(1) C(3) ; C(3) C(2)];

GJ   = 1/4* (J^2 - 1 - 2*logJ);
I4C  = N_fib' * Cmat * N_fib;
expp = exp(c1*(sqrt(I4C)-1)^4);

W = 1/2*mu*(C(1)+C(2)-2) - mu * logJ + kappa * GJ + c0 * (expp-1);
```

strain energy

```
S =[];
CC=zeros(3);

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

1st derivative of strain energy

```
NTN = [N_fib(1)*N_fib(1) , N_fib(2)*N_fib(2) , N_fib(1)*N_fib(2)];

S = mu*([1 1 0]-C_inv) + kappa/2*(J^2-1)*C_inv + ...
    + 4 * NTN * c0 * c1 * expp * (sqrt(I4C)-1)^3/sqrt(I4C);

II=sqrt(I4C);

for i=1:3
    for j=1:3
CC(i,j)= kappa * J^2 * C_inv(i) * C_inv(j) + ...
   ( mu - kappa/2*(J^2-1) )* (C_inv(i)*C_inv(j)+C_inv(i)*C_inv(j)) + ...
   NTN(i)*NTN(j)* 8*c0*c1*expp* ( 2*c1/I4C*(II-1)^6 + 1 - 3/(2*II) + 1/(2*II^3) );
    end
end
CC(1,2)= kappa * J^2 * C_inv(1) * C_inv(2) + ...
  ( mu - kappa/2*(J^2-1) )* (C_inv(3)*C_inv(3)+C_inv(3)*C_inv(3)) + ...
  NTN(1)*NTN(2)* 8*c0*c1*expp * ( 2*c1/I4C*(II-1)^6 + 1 - 3/(2*II) + 1/(2*II^3) );

CC(3,3)= kappa * J^2 * C_inv(3) * C_inv(3) + ...
  ( mu - kappa/2*(J^2-1) )* (C_inv(1)*C_inv(2)+C_inv(3)*C_inv(3)) + ...
  NTN(3)*NTN(3)* 8*c0*c1*expp * ( 2*c1/I4C*(II-1)^6 + 1 - 3/(2*II) + 1/(2*II^3) );

CC(2,1)=CC(1,2);
CC(3,1)=CC(1,3);
CC(3,2)=CC(2,3);
```

2nd derivative of strain energy