Computational Solid Mechanics
Assignment 3
Report on Finite deformation nonlinear elasticity

- Anil Bettadahalli Channakeshava, Manisha Chetry

08/06/2016

1 Kirchhoff Saint-Venant material model :

1.1 Verification of Stress tensor:

From the given data, we know that stored elastic energy function is,

$$W(\varepsilon) = \frac{\lambda}{2}(tr\varepsilon)^2 + \mu tr(\varepsilon^2) = \frac{\lambda}{2}(\varepsilon_{ii})^2 + \mu\varepsilon_{jk}\varepsilon_{jk} \tag{1}$$

And now we need to verify that the stress tensor obtained from $\sigma = \partial W/\partial\varepsilon$ agrees with usual linear elasticity expression. Hence we are differentiating equation(1) with respect to ε and we obtain,

$$\sigma = \partial W/\partial\varepsilon$$

$$\sigma = \frac{\partial(\frac{\lambda}{2}\varepsilon_{ii}\varepsilon_{ii} + \mu\varepsilon_{jk}\varepsilon_{jk})}{\partial\varepsilon_{lm}} = \frac{\lambda}{2}\frac{\partial(\varepsilon_{ii}\varepsilon_{ii})}{\partial\varepsilon_{lm}} + \mu\frac{\partial(\varepsilon_{jk}\varepsilon_{jk})}{\partial\varepsilon_{lm}}$$

$$\sigma = \frac{\lambda}{2}\left[\frac{\varepsilon_{ii}\partial\varepsilon_{ii}}{\partial\varepsilon_{lm}} + \frac{\varepsilon_{ii}\partial\varepsilon_{ii}}{\partial\varepsilon_{lm}}\right] + \mu\left[\frac{\varepsilon_{jk}\partial\varepsilon_{jk}}{\partial\varepsilon_{lm}} + \frac{\varepsilon_{jk}\partial\varepsilon_{jk}}{\partial\varepsilon_{lm}}\right]$$

$$\implies \sigma = \frac{\lambda}{2}[\varepsilon_{ii}\delta_{li}\delta_{mi} + \varepsilon_{ii}\delta_{li}\delta_{mi}] + \mu[\varepsilon_{jk}\delta_{lj}\delta_{mk} + \varepsilon_{jk}\delta_{lj}\delta_{mk}]$$

$$\sigma_{\mathbf{lm}} = \lambda\varepsilon_{ii}\delta_{lm} + 2\mu\varepsilon_{lm}$$

Or we can also write as,

$$\sigma = \lambda tr(\varepsilon)\mathbf{I} + 2\mu\varepsilon \tag{2}$$

Hence as we can see that arriving at equation(2), we verified that the stress tensor obtained from $\sigma = \partial W/\partial\varepsilon$ agrees with usual linear elasticity expression.

1.2 Verification of Isotropicity:

From the lecture notes we know that relation between 2nd Piola-Kirchoff stress tensor and Cauchy stress tensor $\mathbf{C}$ is given as,

$$\mathbf{S} = 2\frac{\partial W}{\partial\mathbf{C}} = \frac{\partial W}{\partial\mathbf{E}} \tag{3}$$

We also know that material can be isotropic only if the strain energy function W depends only on the principal invariants of $\mathbf{C}$. And we also know that any strain energy function can be written in terms of the principal invariants of $\mathbf{C}$ as,

$$W(\mathbf{C}) = W(I_1(\mathbf{C}), I_2(\mathbf{C}), I_3(\mathbf{C}))$$

$$I_1(\mathbf{C}) = trace(\mathbf{C}) = C_{ii} = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$$

$$I_2(\mathbf{C}) = \frac{1}{2}\Big[(trace\mathbf{C})^2 - trace(\mathbf{C}^2)\Big] = \frac{1}{2}\Big[C_{ik}C_{ki} - C_{jj}^2\Big] = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2$$

$$I_3(\mathbf{C}) = det(\mathbf{C}) = \lambda_1^2\lambda_2^2\lambda_3^2$$

Lets assume the material as Isotropic and hence we write the energy equation purely in terms of function of principal invariants of $\mathbf{C}$ i.e differentiating W with respect to $\mathbf{C}$,

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

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

After grouping the terms,

$$\frac{\partial W}{\partial \mathbf{C}} = m_0\mathbf{I} + m_1\mathbf{C}^T + m_2\mathbf{C}^{-T} \tag{4}$$

We know from theorem given in lecture notes that $\mathbf{C}^{-1}$ is written as function of $\mathbf{C}^2$,$\mathbf{C}$ and $\mathbf{I}$. Imposing this in equation(4) we obtain,

$$\frac{\partial W}{\partial \mathbf{C}} = m_0\mathbf{I} + m_1\mathbf{C} + m_2\mathbf{C}^2 \tag{5}$$

But we know that formula for PK-2 stress tensor($\mathbf{S}$) can be yielded from above equation(5) as,

$$\mathbf{S} = 2\frac{\partial W}{\partial \mathbf{C}} = 2(m_0\mathbf{I} + m_1\mathbf{C} + m_2\mathbf{C}^2) \tag{6}$$

We know that the Cauchy stress tensor can also be written as,

$$\sigma = 2\rho(s_1\mathbf{I} + s_2\mathbf{B} + s_3\mathbf{B}^2) \tag{7}$$

Where $\mathbf{B}$ is a Right Cauchy-Green deformation tensor. And now from equations (6) and (7) we can observe that both are similar and are derived by assuming material is isotropic prior. Hence we can conclude that material is isotropic and our assumptions is correct.

1.3 Derivation of Second Piola-Kirchhoff stress (S):

From the given data of assignment part(1) we have that after linearization, the hyperelastic model for Kirchoff Saint-Venant material model is given by,

$$W(\mathbf{E}) = \frac{\lambda}{2}(tr\mathbf{E})^2 + \mu tr(\mathbf{E}^2) \tag{8}$$

We know that $\mathbf{S}$ is defined as,

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

Therefore after differentiating the above equation, we obtain

$$\mathbf{S} = \frac{\partial W}{\partial \mathbf{E}} = \frac{\lambda}{\cancel{2}}\cancel{2}tr(\mathbf{E})\mathbf{1} + 2\mu\mathbf{E}$$

Hence we obtain the Second Piola-Kirchhoff stress tensor($\mathbf{S}$) as,

$$\boxed{\mathbf{S} = \lambda tr(\mathbf{E})\mathbf{1} + 2\mu\mathbf{E}} \tag{9}$$

1.4 Relation between nominal stress(P) and stretch ratio(Λ) :

We have from the given data for uniform deformation of rod aligned with X-axis, spatial description equations as $x = \Lambda X$, $y = Y$, $z = Z$ and $\Lambda > 0$ is the stretch ratio along X-direction. And we know that nominal normal stress P is given by,

$$\mathbf{P} = \mathbf{FS}$$

Where $\mathbf{F}$ is Deformation gradient ,

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

And $\mathbf{S}$ is second Piola-Kirchhoff stress tensor,

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

But the Green Lagrange strain tensor $\mathbf{E}$ is given as,

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) \tag{10}$$

Where $\mathbf{C}$=$\mathbf{F}^T\mathbf{F}$. And hence we obtain $\mathbf{C}$ as,

$$\mathbf{C} = \begin{bmatrix} \Lambda^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Therefore substituting $\mathbf{C}$ in equation(10), we obtain

$$\mathbf{E} = \frac{1}{2}\begin{bmatrix} \Lambda^2 - 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

$$\mathbf{S} = \frac{\lambda}{2}(\Lambda^2 - 1)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \mu\begin{bmatrix} \Lambda^2 - 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S} = \frac{1}{2}\begin{bmatrix} (\Lambda^2 - 1)(\lambda + \mu) & 0 & 0 \\ 0 & \lambda(\Lambda^2 - 1) & 0 \\ 0 & 0 & \lambda(\Lambda^2 - 1) \end{bmatrix}$$

Therefore $\mathbf{P}$ now becomes,

$$\mathbf{P} = \mathbf{FS} = \frac{1}{2}\begin{bmatrix} \Lambda & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} (\Lambda^2-1)(\lambda+\mu) & 0 & 0 \\ 0 & \lambda(\Lambda^2-1) & 0 \\ 0 & 0 & \lambda(\Lambda^2-1) \end{bmatrix}$$

$$\mathbf{P} = \frac{\Lambda(\Lambda^2-1)}{2}\begin{bmatrix} (\lambda+\mu) & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}.$$

The first term of Nominal stress is given by,

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

From the above equation(11), we can see the derived relation between the nominal stress $P(xX$ component of the first Piola-Kirchhoff stress) and the stretch ratio Λ. And the Plot in Figure(1) below refers to comparison made with nominal stress(P) versus stretch factor (Λ).

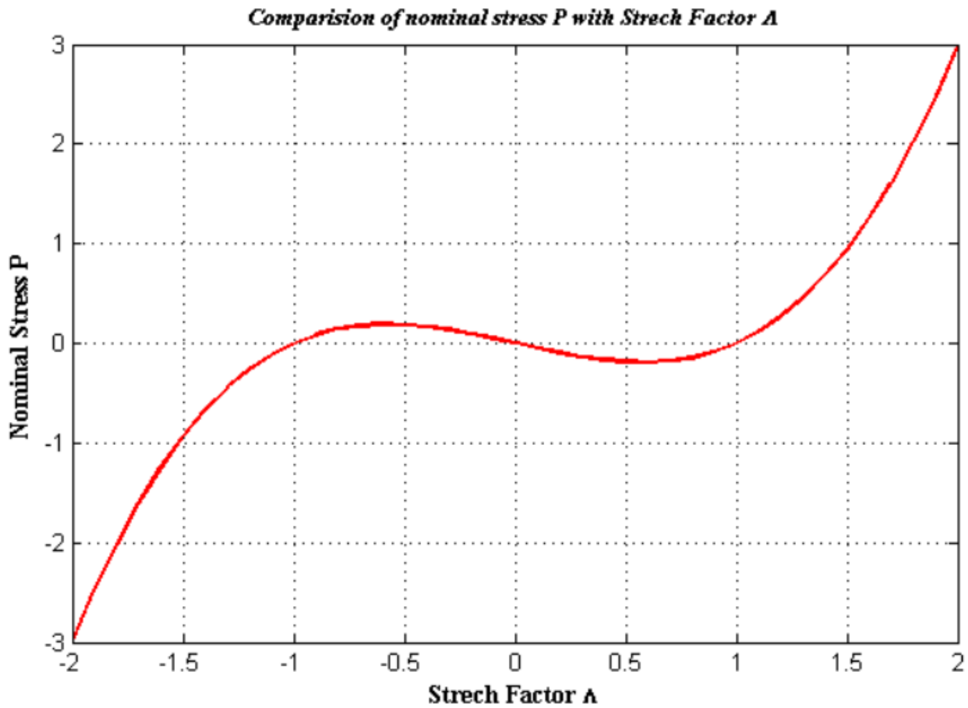

Figure 1: Plot of nominal stress (P) with stretch factor (Λ)

1.5 Verifying whether relation $P(\Lambda)$ is monotonic or not :

We can say that $P(\Lambda)$ is not monotonic as it fails at $\Lambda = \pm 1$. And we can derive the critical stretch value by differentiating the equation(11) w.r.t Λ i.e.,

$$\frac{\partial \mathbf{P}}{\partial \Lambda} = \frac{\partial}{\partial \Lambda}\left(\frac{\Lambda(\Lambda^2-1)}{2}(\lambda+\mu)\right)$$

$$\implies \frac{\partial \mathbf{P}}{\partial \Lambda} = \frac{3\Lambda^2-1}{2}(\lambda+\mu) \tag{12}$$

Hence after equating the above equation(12) to 0, we obtain critical stretch(Λ_{crit}) at which model fails with zero stiffness as,

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

Hence we can conclude that critical stiffness does not depend on any elastic constants. And now to show that the material does not satisfy the growth conditions i.e., $W(\mathbf{E}) \to +\infty$ when $J \to 0^+$, first lets substitute value of trace($\mathbf{E}$) in equation(8) which yields,

$$W(\mathbf{E}) = \frac{\lambda}{2}\Big(\frac{\Lambda^2 - 1}{2}\Big)^2 + \mu\frac{(\Lambda^2 - 1)^2}{4} \tag{13}$$

Hence from the above equation(13), we can observe that by using Jacobian (J equals to Λ) when tends to 0^+, W does not tend to ∞. Hence the material does not satisfy the growth conditions.

1.6 Modified Kirchhoff Saint-Venant material model :

From the given data, the modified Kirchhoff Saint-Venant material model given as,

$$W(\mathbf{E}) = \frac{\lambda}{2}(lnJ)^2 + \mu tr(\mathbf{E}^2) \tag{14}$$

After substituting values of J and $tr(\mathbf{E^2})$ in the above equation we can write it as,

$$W(\mathbf{E}) = \frac{\lambda}{2}(ln\Lambda)^2 + \mu\Big(\frac{\Lambda^2 - 1}{2}\Big) \tag{15}$$

Here we can observe from the above equation(15), when the Jacobian (J equals to Λ) tends to 0^+ ($ln\Lambda^2$) increases to larger value, and hence $W(\mathbf{E}) \to +\infty$. Therefore we can conclude that this modified model overcomes the drawback of previous model by satisfying the growth conditions.

1.7 Implementation of new material model :

The implementation of this material model(Kirchhoff Saint-Venant material model) in the MATLAB subroutine is shown in the *Appendix 1.* The consistency test was performed by using the MATLAB Subroutine `'Check_Derivatives.m'`. The code was working fine as expected and we performed consistency test on two examples : *'Upsetting of the block, dead load' and 'Compression of a slender beam, dead load'*. Both the examples passed the consistency test giving very less relative errors of 'gradient' and 'hessian' with the Kirchhoff Saint-Venant material model, thus verifying that our implementation is correct. Below Figures (2) and (3) shows the plots of two examples which was taken for verifying the consistency test for the new material model.

In order to explain the material instabilities that was caused by this model we took the example of *'Upsetting of a block, imposed displacements'* as numerical example. To show the material instabilities, first we have fixed all the dof's along 'y' to 0, so that no geometrical instabilities will occur. And also it is to be noted that since no perturbations are given, so buckling does not happen and hence geometric instabilities does not occur . We also verified that the relation $P(\Lambda)$ is not monotonic (*see section 1.5)* and once the critical stretch is reached, the stress begins to decrease and this exhibits the softening behavior. This produces an unstable state because when loading beyond the critical stretch, a small increment in stress will produce a non-linear change in the stretch increment. We can observe from below Figure (4) that beyond a particular point there is a large increase in deformation while the force remains constant.

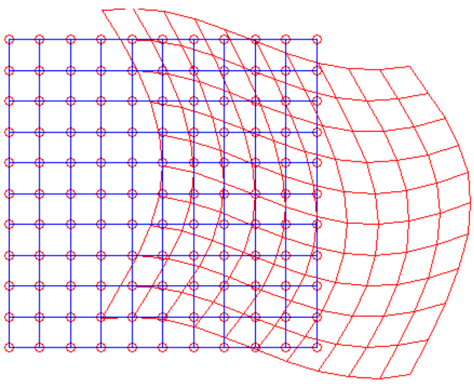

Figure 2: Upsetting of the block, dead load after passing consistency test .

Figure 3: Compression of slender beam, dead load after passing consistency test

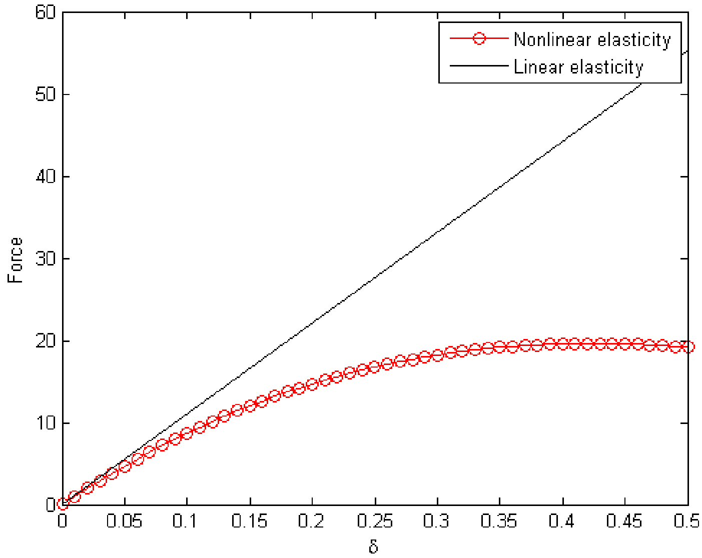

Figure 4: Force v/s displacement curve showing the Material instability.

2 Implementation of Line-search. :

The line search method first finds a descent direction along which the objective function 'f' will be reduced and then computes a step size that determines how far 'x' should move along that direction. The descent direction can be computed by various methods, such as gradient descent, Newton's method and Quasi-Newton method. The step size can be determined either exactly or inexactly.

2.1 Testing of Line search

The test was carried out for example 4('arch, dead load at center of the arch'). The results obtained using normal Newton-Raphson's method were diverging when the buckling starts. It can be seen in the Figures (5) and (6) below. This problem can be optimized by implementing Newton-Raphson with Linesearch and the results obtained are stable and converging which can be seen below in Figures (7) and (8), but iterations at the time of buckling was very large. The implementation of this method in MATLAB subroutine `'Equilibrate.m'` is shown in the *Appendix 2*.

i) Implementation of Newton-Raphson for 'example.4' without Line-search :

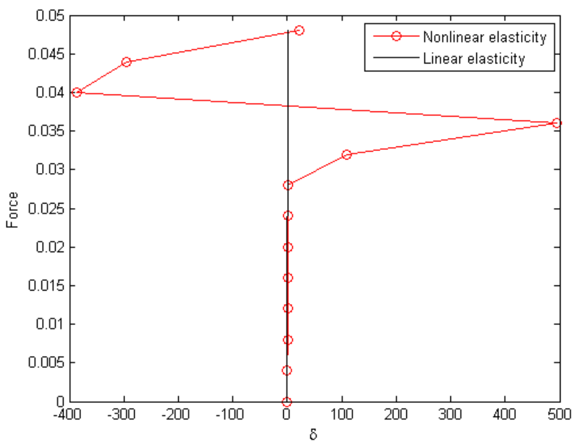

Figure 5: Force v/s Displacement

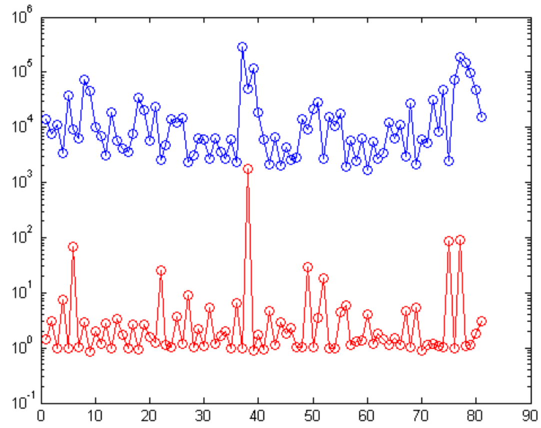

Figure 6: Error Plot

ii) Implementation of Newton-Raphson for 'example.4' with Line-search :

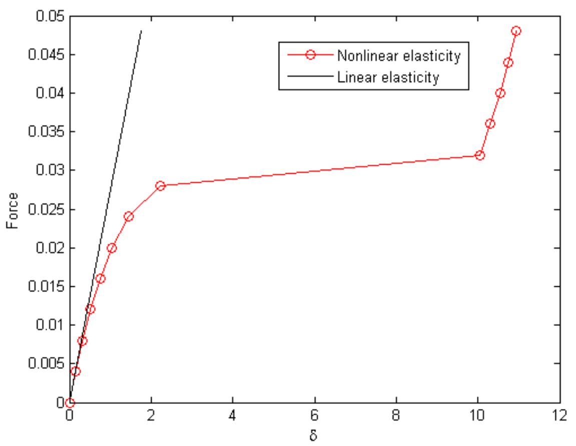

Figure 7: Force v/s Displacement

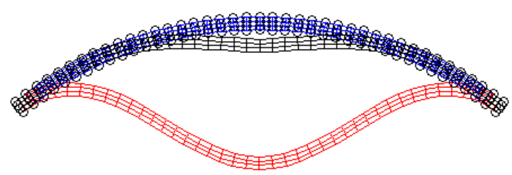

Figure 8: Solution plot

3 Implementation of new material model(Transversely isotropic material model) :

3.1 Implementation of new material model :

Given model whose strain energy density (or hyper-elastic potential) is:

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

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

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

Such a model depends on the principal invariants of $\mathbf{C}$ and additionally by the fourth invariant which is given by,

$$I_4(\mathbf{C}) = \mathbf{N}^{fib} \cdot \mathbf{C} \cdot \mathbf{N}^{fib} = C_{ij} N_i^{fib} N_j^{fib}$$

Where $\mathbf{N}^{fib}$ is a unit vector which gives the orientation of fibers in the reference configuration.

As per the given question we have implemented the 'Transversely isotropic model' with given model parameters in the MATLAB and obtained results. The implementation of this model in MATLAB can be seen in *Appendix 3*.

3.2 Consistency Test :

The consistency test was performed by using the MATLAB Subroutine `'Check_Derivatives.m'`. The code was working fine as expected and we performed consistency test on two examples : *'Upsetting of a block, imposed displacements' and 'Compression of a slender beam, imposed displacements'*. Both the examples passed the consistency test giving very less relative errors of 'gradient' and 'hessian' with the 'Transversely isotropic material model', thus verifying that our implementation is correct. Below Figures (9) and (10) shows the plots of two examples which was taken for verifying the consistency test for the new material model.

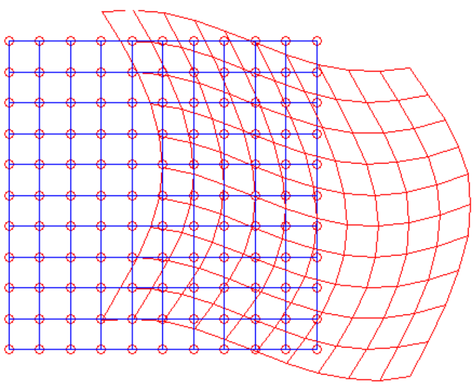

Figure 9: Upsetting of a block, imposed displacements after passing consistency test .

Figure 10: Compression of a slender beam, imposed displacements after passing consistency test

Quadratic Convergence:

To show that Newton's Method(without line-search) converges quadratically, we have considered example 0(upsetting of a block, dead load in tension) using material model 'Transversely isotropic material model' with model parameters: $\theta = \pi/6$, $\mu = 1$, $c_0 = 80$, $c_1 = 5$ and $\kappa = 100$ and the obtained error plot is shown below in Figure (11).

As we can see from the error plot below, that the relative error converges quadratically in three iterations. The relative error is of the order 10^{-12} and 10^{-8} in third iteration for quadratic convergence which would be of order 10^{-8} and 10^{-6} respectively for linear convergence and also we can observe that there is faster convergence(due to quadratic convergence) soon after 2nd iteration . This shows that when the problem is solved with transversely istropoic model, Newton's method converges quadratically.

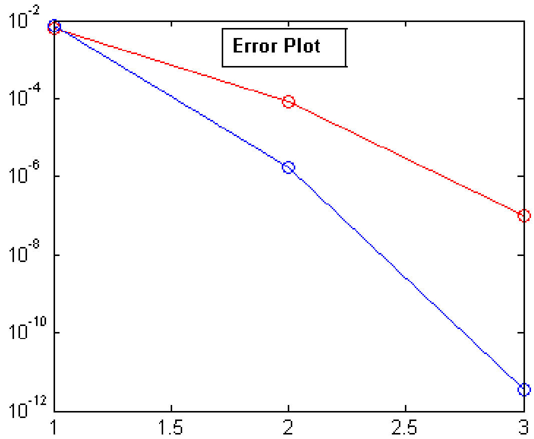

Figure 11: Error Plot showing Quadratic Convergence

3.3 Testing for different cases :

We have solved the example 0(A dead load applied on an elastic block in tension) with new material model considering different θ values : $\theta = 0$, $\theta = \pi/6$, $\theta = \pi/4$ and $\theta = \pi/2$ with unit vector, $\mathbf{N}^{fib} = [cos\theta, sin\theta]^T$ and with model parameters: $\mu = 1$, $c_0 = 80$, $c_1 = 5$ and $\kappa = 100$. The obtained plots for different θ values are shown below.

For $\theta = 0$:

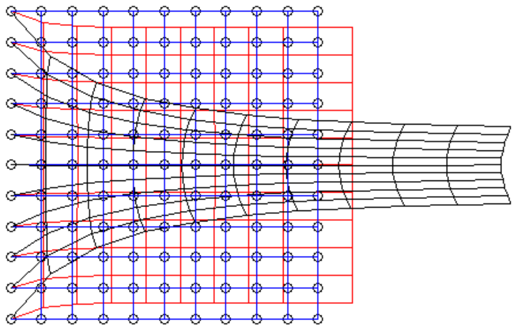

Figure 12: A dead load applied on an elastic block in tension(3e0 N).

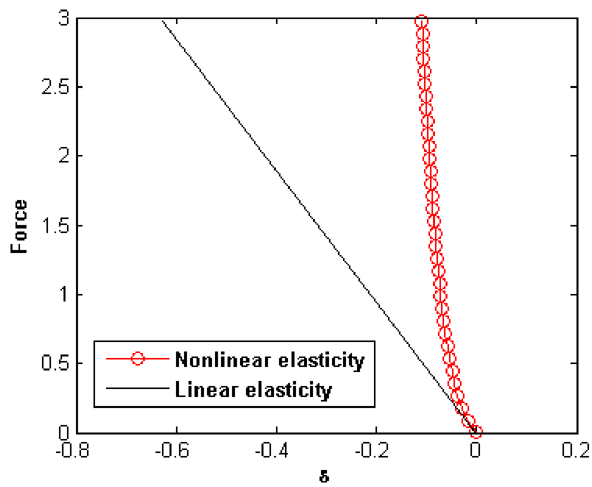

Figure 13: Force v/s Displacement

Above Figures (12) and (13) shows the deformation of 'dead load applied on an elastic block' submitted to tensile load of 3e0 N and Force v/s displacement plot for $\theta = 0$. We know that the unit vector,$\mathbf{N}^{fib} = [cos\theta, sin\theta]^T$ describes the orientation of the fibers in the reference configuration. Here since the value of θ=0, we can observe that the fibers are aligned with the loading direction, material behaves like isotropic and deformation is showing linear even for non-linear algorithm. We can also observe this behavior even from Force v/s displacement plot, where nonlinear elasticity showing almost linear behavior after some time.

For $\theta = \pi/6$:

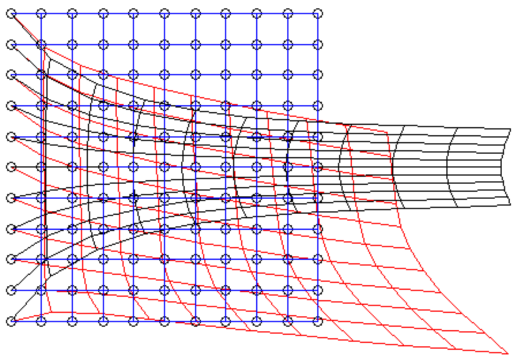

Figure 14: A dead load applied on an elastic block in tension(3e0 N).

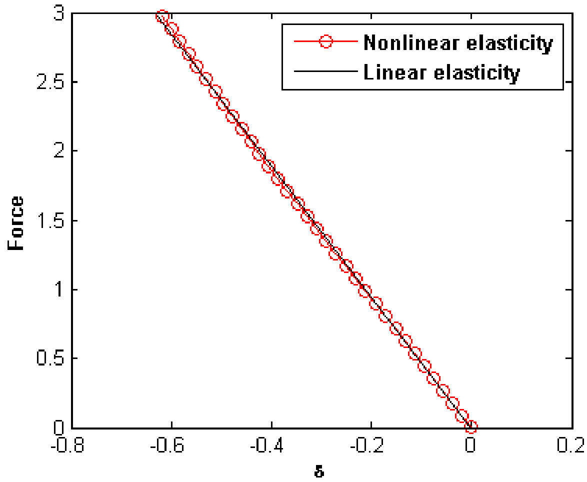

Figure 15: Force v/s Displacement

Above Figures (14) and (15) shows the deformation of 'dead load applied on an elastic block' submitted to tensile load of 3e0 N and Force v/s displacement plot for $\theta = \pi/6$. Since value of $\theta = \pi/6$, fibers are not aligned with the loading direction, which leads to form non-linear deformation for the given tensile loading along the θ value..

For $\theta = \pi/4$:

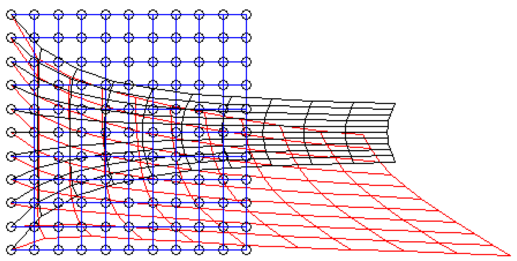

Figure 16: A dead load applied on an elastic block in tension(3e0 N).

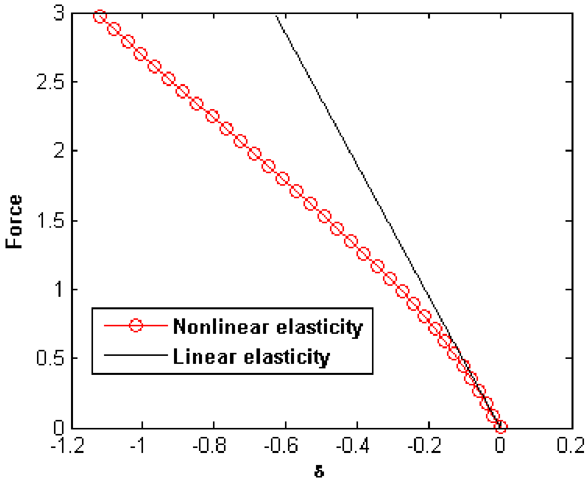

Figure 17: Force v/s Displacement

Above Figures (16) and (17) shows the deformation of 'dead load applied on an elastic block' submitted to tensile load of 3e0 N and Force v/s displacement plot for $\theta = \pi/4$. Since value of $\theta = \pi/4$,similar to previous case here also fibers are not aligned with the loading direction, which leads to form non-linear deformation for the given tensile loading along the θ value.

For $\theta = \pi/2$:

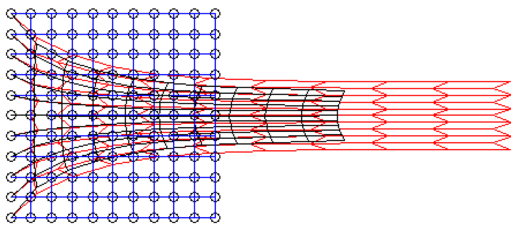

Figure 18: A dead load applied on an elastic block in tension(3e0 N).

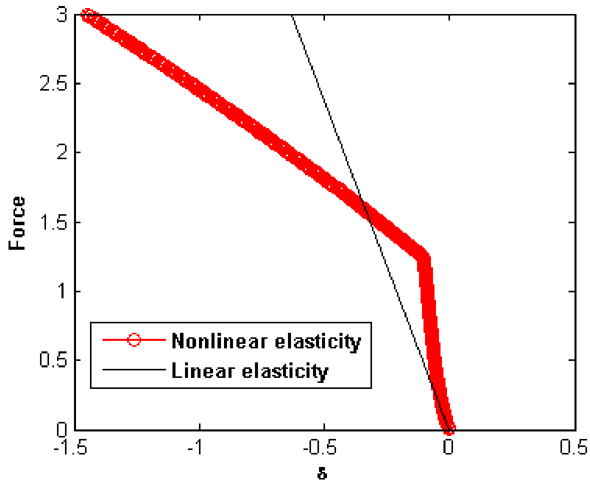

Figure 19: Force v/s Displacement

Above Figures (18) and (19) shows the deformation of 'dead load applied on an elastic block' submitted to tensile load of 3e0 N and Force v/s displacement plot for $\theta = \pi/2$. Here since the value of $\theta = \pi/2$, we can observe that the fibers are perpendicular with the loading direction and material behaves like isotropic.

Appendices

Appendix 1: MATLAB code subroutines for 'Kirchhoff Saint-Venant material model'

In '*Kir_stVen_1.m*'

```
function [W]=Kir_stVen_1(C,lambda,mu,icode)
E=0.5*([C(1) C(3) 0;C(3) C(2) 0;0 0 0]-[1 0 0;0 1 0;0 0 0]);
W=0.5*lambda*(trace(E))^2+mu*trace(E^2);
end
```

In '*Kir_stVen_2.m*'

```
function [W,S]=Kir_stVen_2(C,lambda,mu,icode)
E=0.5*([C(1) C(3) 0;C(3) C(2) 0;0 0 0]-[1 0 0;0 1 0;0 0 0]);
W=0.5*lambda*(trace(E))^2+mu*trace(E^2);
X = [];
X=lambda*trace(E)*[1 0 0;0 1 0;0 0 0]+2*mu*E;
S=zeros(1,3);
S(1)=X(1,1);
S(2)=X(2,2);
S(3)=X(1,2);
end
```

In '*Kir_stVen_2.m*'

```
function [W,S,CC]=Kir_stVen_3(C,lambda,mu,icode)
E=0.5*([C(1) C(3) 0;C(3) C(2) 0;0 0 0]-[1 0 0;0 1 0;0 0 0]);
W=0.5*lambda*(trace(E))^2+mu*trace(E^2);
X = [];
CC=zeros(3);
X=lambda*trace(E)*[1 0 0;0 1 0;0 0 0]+2*mu*E;
S=zeros(1,3);
S(1)=X(1,1);
S(2)=X(2,2);
S(3)=X(1,2);

CC(1,1)=lambda+2*mu;
CC(1,2)=lambda;
CC(1,3)=0;
CC(2,2)=lambda+2*mu;
CC(2,3)=0;
CC(3,3)=2*mu;
CC(2,1)=CC(1,2);
CC(3,1)=CC(1,3);
CC(3,2)=CC(2,3);

end
```

In '*preprocessing.m*'

```
case 2
      mod1.potential = 2; % 2 is Kirchkoff Saint Venant
      mod1.mu=1;
      mod1.lambda=100;
```

```
    otherwise
        error('Material not implemented')
 end
```

Appendix 2: MATLAB code modified subroutines for 'Newton-Raphson with LineSearch'

In '*Equilibrate.m*'

```
case 1, %Newton-Raphson Linesearch
      iter=0;
      err_x=100;
      err_f=100;
      [Ener,grad_E,Hess_E] = Ener_short(x_short,3);
      while (iter<=options.n_iter_max)& ...
        ( (err_x>options.tol_x) | ...
        (err_f>options.tol_f))
        iter=iter+1;
        dx = -Hess_E\grad_E;

        if grad_E'*dx < 0 %%%% LINE SEARCH
            p=dx;
        else
            p=-dx;
        end
        t = 1;
        opts=optimset('TolX',options.TolX,'MaxIter',options.n_iter_max_LS);
        t = fminbnd(@(t) Ener_1D(t,x_short,p),0,2,opts);
        x_short=x_short+t*p;
        [Ener,grad_E,Hess_E] = Ener_short(x_short,3);
        err_x=norm(dx)/norm(x_short);
        err_f=norm(grad_E);
        err_plot=[err_plot err_x];
        err_plot1=[err_plot1 err_f];
      end
```

Appendix 3: MATLAB code subroutines for 'Transversely isotropic material model'

Here we have implemented material model in single function *transv_isotr.m* instead of using 3 functions *transv_isotr_1.m*,*transv_isotr_2.m* and *transv_isotr_3.m*

In '*transv_isotr.m*'

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


J  = sqrt(C(1)*C(2)-C(3)^2) ;
I4 = N_fib'*[C(1) C(3); C(3) C(2)]*N_fib ;
C_inv=[C(2) C(1) -C(3)]/J^2;

W1 = 1/2*mu*(C(1)+C(2)-2) ;
W2 = -mu*log(J) ;
W3 = kappa*1/4*(J^2-1-2*log(J)) ;
W4 = c0*(exp((c1*sqrt(I4)-1)^4)-1);
W = W1+W2+W3+W4 ;

Nf_Nft = [N_fib(1)^2 ; N_fib(2)^2 ; N_fib(1)*N_fib(2)] ;

S1 = mu*[1 1 0] ;
```

```
S2 = -mu*C_inv ;
S3 = kappa/2*(J^2-1)*C_inv  ;
S4 = 4*c0*exp(c1*(sqrt(I4)-1)^4)*c1*(sqrt(I4)-1)^3 /sqrt(I4)*Nf_Nft' ;
S = S1+S2+S3+S4 ;


Cinv = [C(2) -C(3); -C(3) C(1)]/J^2 ;
for i=1:2
    for j=1:2
        DCinv(i,j) = 1/2*(Cinv(i,j)*Cinv(i,j) + Cinv(i,j)*Cinv(i,j)) ;
    end
end
DCinv(1,3) = 1/2*(Cinv(1,1)*Cinv(1,2) + Cinv(1,2)*Cinv(1,1)) ;
DCinv(2,3) = 1/2*(Cinv(2,1)*Cinv(2,2) + Cinv(2,2)*Cinv(2,1)) ;
DCinv(3,3) = 1/2*(Cinv(1,1)*Cinv(2,2) + Cinv(1,2)*Cinv(2,1)) ;
DCinv(3,1) = DCinv(1,3) ;
DCinv(3,2) = DCinv(2,3) ;

DCinv = -DCinv ;

CC2 = -2*mu*DCinv ;
CC3 = 2*kappa/2*(J^2-1)*DCinv + 2*kappa/2*J^2*(C_inv'*C_inv) ;
m1 = 8*c0*exp(c1*(sqrt(I4)-1)^4)*c1^2*(sqrt(I4)-1)^6 /I4 ;  %diff of exponential
m2 = 6*c0*exp(c1*(sqrt(I4)-1)^4)*c1*(sqrt(I4)-1)^2 /I4  ;   % diff of cubic term
m3 = 2*c0*exp(c1*(sqrt(I4)-1)^4)*c1*(sqrt(I4)-1)^3 /(I4^1.5) ;   % diff of sq.root
CC4 = 2*(m1+m2-m3)*(Nf_Nft*Nf_Nft') ;
CC = CC2+CC3+CC4 ;

end
```

In '*preprocessing.m*'

```
case 2
      mod1.potential = 2; % 2 is transversely isotropic model
      mod1.mu=1.;
      mod1.c0=80;
      mod1.c1=5;
      mod1.kappa=100;
      theta=pi/6;
      mod1.N_fib=[cos(theta); sin(theta)];

    otherwise
        error('Material not implemented')
end
```