hyperelasticity_Ogden

April 16, 2021

Dr. Kevin M. Moerman, *Lecturer in Biomedical Engineering*

National University of Ireland Galway.

This is an Octave (an open source alternative to MATLAB) Jupyter notebook

1 Hyperelastic materials

1.1 Introduction

- So-called **hyperelastic** formulations are non-linear constitutive (material behaviour) "laws" which are useful to describe nonlinear elastic materials undergoing large (finite strain) deformation.
- In hyperelasticity the constitutive or material law is defined by a so **strain energy density** function often denoted by a W or Ψ symbol.
- Ψ is a **scalar function** (so not a tensor or vector valued function).
- The strain energy density function has units of **energy per unit volume** such as J/m^3.
- However if one recalls that J can be written in terms of Nm, then we see that $J/m^3 = Nm/m^3 = N/m^2$, which means we may equivalently say that Ψ has **units of stress**.

1.2 Stress computation

- Derivatives of Ψ with a deformation metric provide a stress metric (there are different types of strains each with their own *work conjugate* stress type).
- For instance, the second *Piola-Kirchoff stress* $\mathbf{S}$ is obtained through the derivative with the *Green-Lagrange strain* $\mathbf{E}$

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

- We tend to focus on the *true stres* or **Cauchy stress** $\boldsymbol{\sigma}$, which is obtained through with the aid of the *deformation gradient tensor* $\mathbf{F}$:

$$\boldsymbol{\sigma} = J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^{\top}$$

- In some cases formulations are specified using the *principal stretches* λ_i. These may also be used to derived (principal) stresses e.g.:

$$\sigma_i = J^{-1}\lambda_i \frac{\partial \Psi}{\partial \lambda_i}$$

1.3 Three types of hyperelastic formulations

- Three types of hyperelastic formulation types are treated here with a focus on the Ogden formulation:

1. **Constrained** formulations (a.k.a "incompressible" formulations)

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \sum_{a=1}^{N} \frac{c_a}{m_a^2}\left(\lambda_1^{m_a} + \lambda_2^{m_a} + \lambda_3^{m_a} - 3\right)$$

2. **Unconstrained** or **coupled** formulations (a.k.a "compressible" formulations)

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \frac{\kappa'}{2}(J-1)^2 + \sum_{a=1}^{N} \frac{c_a}{m_a^2}\left(\lambda_1^{m_a} + \lambda_2^{m_a} + \lambda_3^{m_a} - 3 - m_a \ln(J)\right)$$

3. **Uncoupled** formulations (a.k.a "nearly incompressible" formulations)

$$\Psi(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3) = \frac{\kappa}{2}\ln{(J)}^2 + \sum_{a=1}^{N} \frac{c_a}{m_a^2}\big(\tilde{\lambda}_1^{m_a} + \tilde{\lambda}_2^{m_a} + \tilde{\lambda}_3^{m_a} - 3\big)$$

- This notebook discusses these and provides example implementations for uniaxial loading and first order $N = 1$ **Ogden hyperelastic** formulations.

For more background information see chapter 6 Hyperelasticity" in Holzapfel's book: *G. Holzapfel, Nonlinear solid mechanics: A continuum approach for engineering. John Wiley & Sons Ltd., 2000.*

1.3.1 Anatomy of the Ogden formulation

- Typically the Ogden formulation looks something like this (contrained form showed here, implementations vary depending on software):

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \sum_{a=1}^{N} \frac{c_a}{m_a^2}\big(\lambda_1^{m_a} + \lambda_2^{m_a} + \lambda_3^{m_a} - 3\big)$$

- If we "distribute" the -3 as a set of -1's, and work a factor $\frac{1}{m_a}$ into the summation we can see the above is equivalent to:

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \sum_{a=1}^{N} \frac{c_a}{m_a}\big(\frac{1}{m_a}(\lambda_1^{m_a} - 1) + \frac{1}{m_a}(\lambda_2^{m_a} - 1) + \frac{1}{m_a}(\lambda_3^{m_a} - 1)\big)$$

- Now one may recognize these as the Seth-Hill class of strains

$$E_i^{(m_a)} = \frac{1}{m_a}(\lambda_i^{m_a} - 1)$$

- For instance using $m_a = 2$ makes it use the Green-Lagrange strain $\mathbf{E}$

$$E_i = \frac{1}{2}(\lambda_i^2 - 1)$$

- Furthermore we may recognize that the sum of such parts for is actually the trace of such a strain tensor leading to:

$$\Psi = \sum_{a=1}^{N} \frac{c_a}{m_a}\mathrm{tr}(\mathbf{E}^{(m_a)})$$

- So the Ogden formulation is a powerful law where we define energies by scaling (multiplying) the trace of a chosen *"strain type"* (defined by m_a) by a stiffness parameter c_a. Summing lots of terms ($N > 1$) allows one to capture complex stiffnening behaviour.
- The Ogden formulation can also conveniently be used as the **"mother" of many other formulations**
- Using $N = 1$ and $m_1 = 2$ makes the Ogden formulation reduce to a **Neo-Hookean** formulation

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \frac{c_1}{4}\big(\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - 3\big) = \frac{c_1}{4}\big(I_1 - 3\big) = \frac{c_1}{2}\mathrm{tr}(\mathbf{E})$$

- The Neo-Hookean is one of the simples hyperelastic formulations and is named after the fact that it can be thought of as an extension of Hooke's law to non-linear solid mechanics (it reduces to Hooke's law for infinitesimal strains).
- Using $N = 2$ and $m_1 = -m_2 = 2$ makes the Ogden formulation reduce to a **Mooney-Rivlin** formulation (if $J = 1$)

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \frac{c_1}{4}\big(\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - 3\big) + \frac{c_2}{4}\big(\lambda_1^{-2} + \lambda_2^{-2} + \lambda_3^{-2} - 3\big) = \frac{c_1}{4}\big(I_1 - 3\big) + \frac{c_2}{4}\big(I_2 - 3\big)$$

In the above I_1 and I_2 are known as the first and second invariants of the right Cauchy green tensor $\mathbf{C}$*. These often appear in the literature.*

Defining shared variables used by the example numerical implementations

[1]:
```
%% Define parameters common to all examples

%Define material parameters
N=1; %The Ogden law order
c1=1; %The shear modulus like parameter
m1=12; %The non-linearity parameter
kp=1000; %Bulk modulus like parameter (used for constrained model)
k=kp; %Bulk modulus (used for uncoupled model)

%Derive applied stretch
appliedStretch=1.3; %Set applied stretch
nDataPoints=50; %Number of data points to use for evaluation and graph
lambda_3=linspace(1,appliedStretch,nDataPoints); %The 3 direction stretch
```

1.4 Constrained formulations

- The word "constrained" relates to the fact that incompressible behaviour (no volume change) is enforced in the formulation.
- These formulations are not really used in FEA and instead serve as means to easily derive analytical solutions for incompressible behaviour "by hand".

1.4.1 The constrained Ogden formulation

- The constrained Ogden formulation is often presented as:

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \sum_{a=1}^{N} \frac{c_a}{m_a^2} \left(\lambda_1^{m_a} + \lambda_2^{m_a} + \lambda_3^{m_a} - 3\right)$$

- However, something is missing in the above, namely the treatment of the hydrostatic presssure p and its contribution.

$$\Psi(\lambda_1, \lambda_2, \lambda_3, p) = U(p) + \sum_{a=1}^{N} \frac{c_a}{m_a^2} \left(\lambda_1^{m_a} + \lambda_2^{m_a} + \lambda_3^{m_a} - 3\right)$$

- For these constrained forms the contribution $U(p)$ is not derived from the constitutive equation but is instead determined using the boundary conditions.
- Below an example for uniaxial loading is presented.
- The uniaxial load (e.g. a tensile or compressive stretch) is here specified in the 3rd (or Z) direction, which means $\lambda_3 \neq 1$.
- Using the "incompressiblity" and uniaxial loading assumption we can formulate some useful relations to help solve for the stress.
- First of all, the uniaxial conditions mean the other "lateral" stretches are equivalent:

$$\lambda_1 = \lambda_2$$

- Secondly, if the material is truely incompressible we have $J = \lambda_1\lambda_2\lambda_3 = 1$, and since $\lambda_1 = \lambda_2$ we can derive:

$$J = \lambda_1\lambda_1\lambda_3 = \lambda_1^2\lambda_3 = 1 \rightarrow \lambda_1 = \lambda_2 = \sqrt{\frac{1}{\lambda_3}} = \lambda_3^{-\frac{1}{2}}$$

- Thirdly for uniaxial conditions there is only one none-zero stress, the applied stress $\sigma_3 = \sigma_{33}$, therefore:

$$\sigma_1 = \sigma_2 = \sigma_{11} = \sigma_{22} = 0$$

- So now with an assummed $J = 1$, the ability to express all stretches in terms of λ_3 (the known applied stretch), and the fact that $\sigma_1 = \sigma_2 = 0$, we are ready to start tackling the full stress evaluation.

- First the Cauchy stress tensor $\boldsymbol{\sigma}$ is defined as:

$$\boldsymbol{\sigma} = \bar{\boldsymbol{\sigma}} - \bar{p}\mathbf{I}$$

- The contribution $\bar{\boldsymbol{\sigma}}$ is derived from the constitutive equation:

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \sum_{a=1}^{N} \frac{c_a}{m_a^2}\left(\lambda_1^{m_a} + \lambda_2^{m_a} + \lambda_3^{m_a} - 3\right)$$

  and is obtained from:

$$\bar{\sigma}_i = \lambda_i \frac{\partial \Psi}{\partial \lambda_i}$$

- Leading to:

$$\bar{\sigma}_i = \sum_{a=1}^{N} \frac{c_a}{m_a} \lambda_i^{m_a}$$

- The next step is to determine $\bar{p}$ in this relation:

$$\boldsymbol{\sigma} = \bar{\boldsymbol{\sigma}} - \bar{p}\mathbf{I}$$

- First lets rewrite the above in terms of the principal components σ_i

$$\sigma_i = -\bar{p} + \sum_{a=1}^{N} \frac{c_a}{m_a} \lambda_i^{m_a}$$

- Next we use $\sigma_1 = \sigma_2 = 0$ to derive an expression for $\bar{p}$:

$$\sigma_1 = \sigma_2 = -\bar{p} + \sum_{a=1}^{N} \frac{c_a}{m_a} \lambda_1^{m_a} = 0$$

$$\rightarrow \bar{p} = \sum_{a=1}^{N} \frac{c_a}{m_a} \lambda_1^{m_a}$$

- Finaly, implementing $\lambda_1 = \lambda_2 = \lambda_3^{-\frac{1}{2}}$ leads to:

$$\rightarrow \bar{p} = \sum_{a=1}^{N} \frac{c_a}{m_a} (\lambda_3^{-\frac{1}{2}})^{m_a} = \sum_{a=1}^{N} \frac{c_a}{m_a} \lambda_3^{-\frac{m_a}{2}}$$

- Which therefore allows for the formulation of an expression for $\bar{\sigma}_3$:

$$\sigma_3 = -\bar{p} + \sum_{a=1}^{N} \frac{c_a}{m_a} \lambda_3^{m_a} = -\sum_{a=1}^{N} \frac{c_a}{m_a} \lambda_3^{-\frac{m_a}{2}} + \sum_{a=1}^{N} \frac{c_a}{m_a} \lambda_3^{m_a}$$

- Which can be simplified to:

$$\sigma_3 = \sum_{a=1}^{N} \frac{c_a}{m_a} \left(\lambda_3^{m_a} - \lambda_3^{-\frac{m_a}{2}}\right)$$

- The full Cauchy stress tensor can then be written as:

$$\boldsymbol{\sigma} = \sigma_3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \sum_{a=1}^{N} \frac{c_a}{m_a} \left(\lambda_3^{m_a} - \lambda_3^{-\frac{m_a}{2}} \right) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Note that although $\bar{p}$ is a type of pressure contribution, it should not be confused with the full hydrostatic pressure p which is derived from:

$$p = -\frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma}) = -\frac{\sigma_3}{3}$$

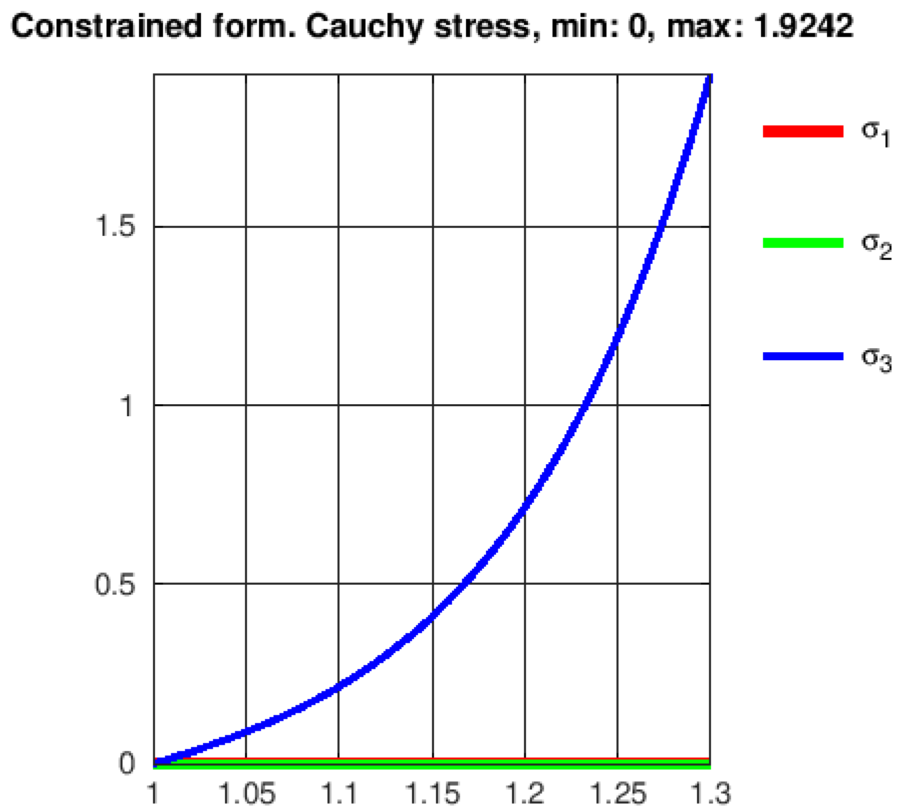
1.4.2 Numerical implementation

Compute stresses

[2]:
```
%% The constrained formulation

%Direct stress computation
S3=(c1/m1).*(lambda_3.^m1-lambda_3.^(-m1/2));
S1=zeros(size(S3));
S2=zeros(size(S3));

%Compute Jacobian for plotting
lambda_1=sqrt(1./lambda_3);
lambda_2=lambda_1;
J=lambda_1.*lambda_2.*lambda_3;
```

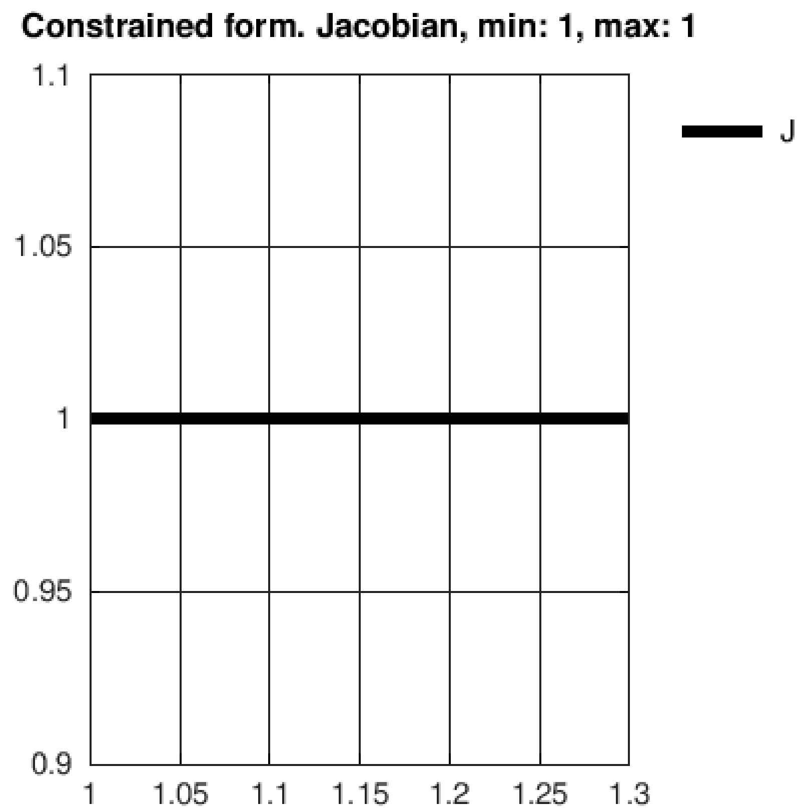
Visualize stresses

[3]:
```
%Visualize stress graphs
figure; hold on;
title(['Constrained form. Cauchy stress, min: ',num2str(min(S3(:))),...
', max: ',num2str(max(S3(:)))]); %Add title
h1=plot(lambda_3,S1,'r-','LineWidth',20); %The 1 direction principal stress
h2=plot(lambda_3,S2,'g-','LineWidth',15); %The 2 direction principal stress
h3=plot(lambda_3,S3,'b-','LineWidth',10); %The 3 direction principal stress
hl=legend([h1 h2 h3],{'\sigma_1','\sigma_2','\sigma_3'}); %Add legend
set(hl,'FontSize',15,'Location','NorthEastOutside','Box','off'); %Adjust legend
axis tight; axis square; grid on; box on;
set(gca,'FontSize',15);
```



Visualize Jacobian

[4]:
```
%Visualize Jacobian
figure; hold on;
title(['Constrained form. Jacobian, min: ',num2str(min(J(:))),...
', max: ',num2str(max(J(:)))]); %Add title
h1=plot(lambda_3,J,'k-','LineWidth',20); %The 1 direction principal stress
hl=legend([h1],{'J'}); %Add legend
set(hl,'FontSize',15,'Location','NorthEastOutside','Box','off'); %Adjust legend
axis tight; axis square; grid on; box on;
set(gca,'FontSize',15);
```


Constrained form. Jacobian, min: 1, max: 1
1.1
1.05
1
0.95
0.9
1
1.05
1.1
1.15
1.2
1.25
1.3
J

1.5 Unconstrained formulations

These formulations are also known as coupled formulations (some literature refers to these formulations as "compressible").

- The unconstrained Ogden formulation is given by

$$\Psi(\lambda_1, \lambda_2, \lambda_3) = \frac{\kappa'}{2}(J-1)^2 + \sum_{a=1}^{N} \frac{c_a}{m_a^2}\big(\lambda_1^{m_a} + \lambda_2^{m_a} + \lambda_3^{m_a} - 3 - m_a \ln(J)\big)$$

- The principal Cauchy stresses σ_i can be computed from:

$$\sigma_i = J^{-1}\lambda_i \frac{\partial \Psi}{\partial \lambda_i}$$

- Leading to:

$$\sigma_i = \kappa'\big(J-1\big) + J^{-1}\sum_{a=1}^{N} \frac{c_a}{m_a}\big(\lambda_i^{m_a} - 1\big)$$

1.5.1 Step-by-step derivation:

1. First compute
$$J^{-1}\lambda_i \frac{\partial}{\partial \lambda_i}\big(\frac{\kappa'}{2}(J-1)^2\big)$$
2. Take derivative with respect to λ_3 and use symmetry
$$= J^{-1}\lambda_3 \frac{\partial}{\partial \lambda_3}\big(\frac{\kappa'}{2}(\lambda_1\lambda_2\lambda_3-1)^2\big)$$
3. Expand square
$$= J^{-1}\lambda_3 \frac{\partial}{\partial \lambda_3}\big(\frac{\kappa'}{2}(\lambda_1^2\lambda_2^2\lambda_3^2 - 2\lambda_1\lambda_2\lambda_3 + 1)\big)$$
4. Evaluate derivative
$$= J^{-1}\lambda_3 \frac{\partial}{\partial \lambda_3}\big(\frac{\kappa'}{2}(\lambda_1^2\lambda_2^2\lambda_3^2 - 2\lambda_1\lambda_2\lambda_3 + 1)\big) = J^{-1}\lambda_3\big(\frac{\kappa'}{2}(2\lambda_1^2\lambda_2^2\lambda_3 - 2\lambda_1\lambda_2)\big)$$
5. Remove factor of 2
$$= J^{-1}\lambda_3\kappa'\big(\lambda_1^2\lambda_2^2\lambda_3 - \lambda_1\lambda_2\big)$$
6. Work in factor λ_3
$$= J^{-1}\kappa'\big(\lambda_1^2\lambda_2^2\lambda_3^2 - \lambda_1\lambda_2\lambda_3\big)$$
7. Recognize J and J^2
$$= J^{-1}\kappa'\big(J^2 - J\big)$$
8. Process division by J (multiply by J^{-1}). This result holds for any λ_i
$$= \kappa'\big(J-1\big)$$

9. Now compute the next part:

$$J^{-1}\lambda_i\frac{\partial}{\partial\lambda_i}\left(\sum_{a=1}^{N}\frac{c_a}{m_a^2}\big(\lambda_1^{m_a}+\lambda_2^{m_a}+\lambda_3^{m_a}-3-m_a\ln\left(J\right)\big)\right)$$

10. First notice that summation can be moved:

$$=\sum_{a=1}^{N}J^{-1}\lambda_i\frac{\partial}{\partial\lambda_i}\left(\frac{c_a}{m_a^2}\big(\lambda_1^{m_a}+\lambda_2^{m_a}+\lambda_3^{m_a}-3-m_a\ln\left(J\right)\big)\right)$$

11. Next take derivative with respect to λ_3 and aim to use symmetry with respect to any stretch

$$=\sum_{a=1}^{N}J^{-1}\lambda_3\frac{\partial}{\partial\lambda_3}\left(\frac{c_a}{m_a^2}\big(\lambda_1^{m_a}+\lambda_2^{m_a}+\lambda_3^{m_a}-3-m_a\ln\left(J\right)\big)\right)$$

12. Use $\frac{\partial}{\partial\lambda_i}(\lambda_i^{m_a})=m_a\lambda_i^{m_a-1}$ and $\ln\left(J\right)=\ln\left(\lambda_1\lambda_2\lambda_3\right)=\ln\left(\lambda_1\right)+\ln\left(\lambda_2\right)+\ln\left(\lambda_3\right)$

$$=\sum_{a=1}^{N}J^{-1}\lambda_3\frac{c_a}{m_a^2}\big(m_a\lambda_3^{m_a-1}-\frac{m_a}{\lambda_3}\big)$$

13. Multiply by λ_3 and move J^{-1}

$$=J^{-1}\sum_{a=1}^{N}\frac{c_a}{m_a^2}\big(m_a\lambda_3^{m_a}-m_a\big)$$

14. Simplify by removing m_a factor

$$=J^{-1}\sum_{a=1}^{N}\frac{c_a}{m_a}\big(\lambda_3^{m_a}-1\big)$$

15. Generalise for any λ_i:

$$=J^{-1}\sum_{a=1}^{N}\frac{c_a}{m_a}\big(\lambda_i^{m_a}-1\big)$$

16. Combine step 8 and 15 to produce overall result:

$$\sigma_i=\kappa'\big(J-1\big)+J^{-1}\sum_{a=1}^{N}\frac{c_a}{m_a}\big(\lambda_i^{m_a}-1\big)$$

1.5.2 How to compute stresses?

- The stress equations have the unknown J as well as λ_1 and λ_2:

$$\sigma_i=\kappa'\big(J-1\big)+J^{-1}\sum_{a=1}^{N}\frac{c_a}{m_a}\big(\lambda_i^{m_a}-1\big)$$

- The uniaxial loading conditions and boundary conditions help simplify this to a single unknown
- First of all uniaxial loading in the 3rd or Z-direction means

$$\lambda_1 = \lambda_2$$

- Next we can use the definition of the Jacobian to come to expressions for λ_1 and λ_2
- Since we have $J = \lambda_1\lambda_2\lambda_3$, and $\lambda_1 = \lambda_2$ we can derive:

$$J = \lambda_1\lambda_2\lambda_3 = \lambda_1\lambda_1\lambda_3 = \lambda_1^2\lambda_3$$

$$\rightarrow \lambda_1 = \lambda_2 = \sqrt{\frac{J}{\lambda_3}}$$

- The above shows that although λ_3 is known, knowledge of J is required in order to determine λ_1 and λ_2. Or conversely λ_1 (or λ_2) needs to be determined allowing for the computation of J. Eitherway one unknown remains.
- To solve for the unkown J we may use the fact that $\sigma_1 = \sigma_2 = 0$

$$\sigma_1 = \kappa'\big(J-1\big) + J^{-1}\sum_{a=1}^{N}\frac{c_a}{m_a}\big(\lambda_1^{m_a}-1\big) = 0$$

- If we solve for J we can use $\lambda_1 = \sqrt{\frac{J}{\lambda_3}}$ and write:

$$\sigma_1 = \kappa'\big(J-1\big) + J^{-1}\sum_{a=1}^{N}\frac{c_a}{m_a}\left(\left(\frac{J}{\lambda_3}\right)^{\frac{m_a}{2}}-1\right)$$

- Or if instead we solve for λ_1 we can use $J = \lambda_1^2\lambda_3$ and write:

$$\sigma_1 = \kappa'\big((\lambda_1^2\lambda_3)-1\big) + \frac{1}{\lambda_1^2\lambda_3}\sum_{a=1}^{N}\frac{c_a}{m_a}\big(\lambda_1^{m_a}-1\big) = 0$$

- Solving these is not trivial but numerical solutions are derived below for J

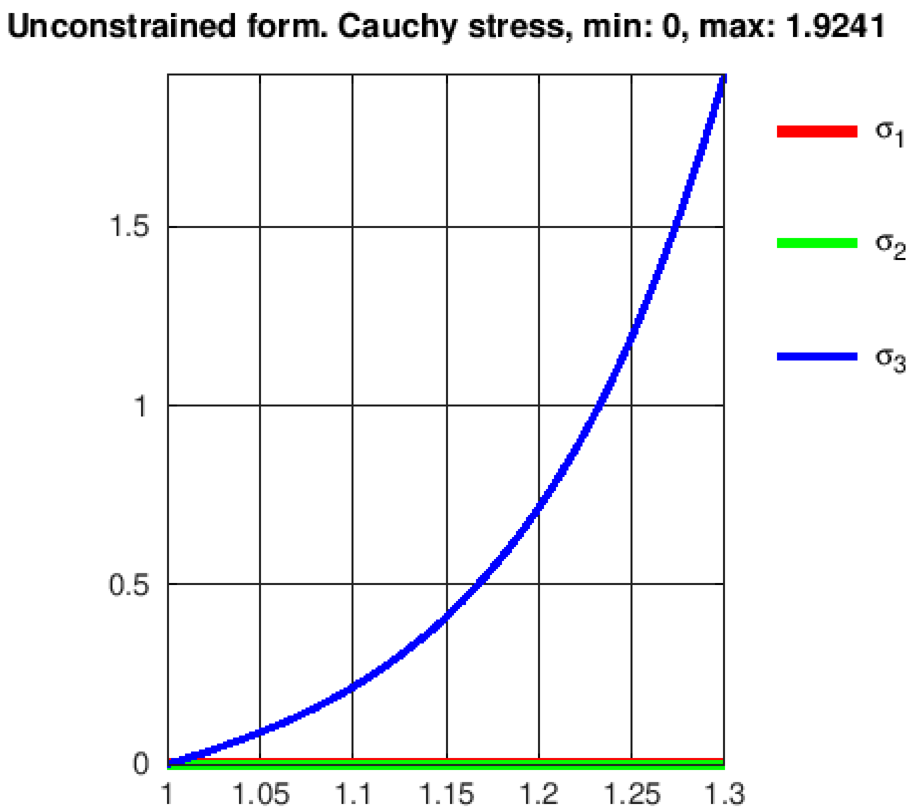
1.5.3 Numerical implementation

Compute stresses

[5]:
```
%% The unconstrained or coupled formulation

% One approach is to define a function for S1 and to find the J for which it is␣
↪zero.
% For this application the fzero function is useful to find J for S1(J)=0.

%Compute Jacobian given boundary conditions S1=S2=0
J=zeros(size(lambda_3)); %Initialize an arry of J values which are all zeros
for q=1:1:nDataPoints %Loop over all data points
    %Create stress function with current lambda
    S1_fun=@(J) kp*(J-1)+(1/J)*(c1/m1)*((sqrt(J/lambda_3(q)).^m1)-1);

    %Find Jacobian for zero stress, use J=1 as initial
    J(q)=fzero(S1_fun,1); %Find root of nonlinear function
end

%Compute transverse stretches using J values
lambda_1=sqrt(J./lambda_3);
lambda_2=lambda_1; %Due to uniaxial loading

%Compute principal stresses (note, these are not ordered)
S1=kp*(J-1)+(1./J).*(c1/m1).*((lambda_1.^m1)-1);
S2=kp*(J-1)+(1./J).*(c1/m1).*((lambda_2.^m1)-1);
S3=kp*(J-1)+(1./J).*(c1/m1).*((lambda_3.^m1)-1);
```

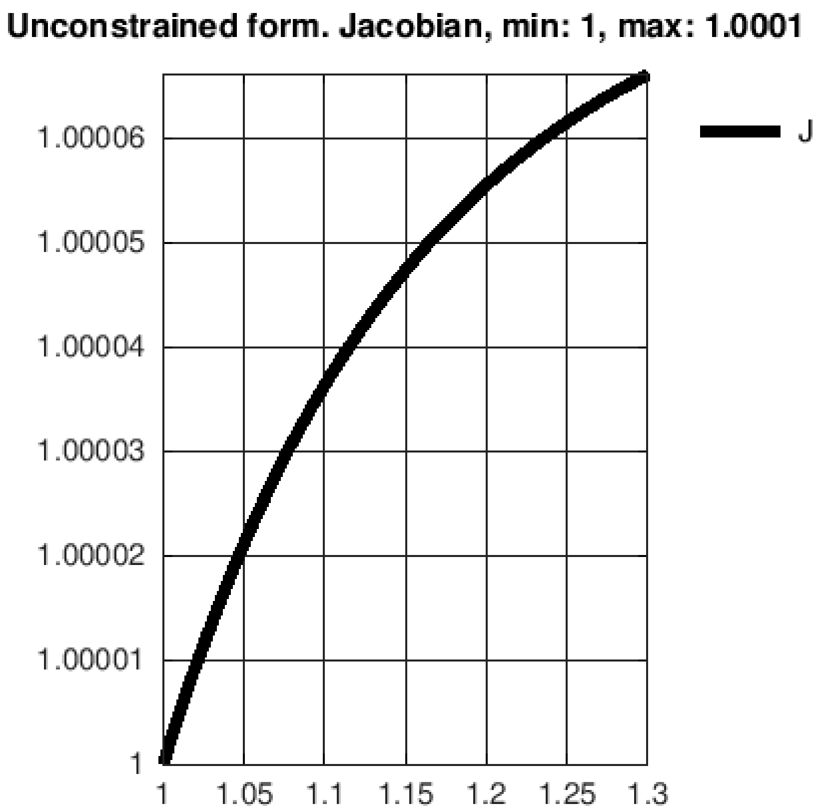
Visualize stresses

[6]:
```
%Visualize stress graphs
figure; hold on;
title(['Unconstrained form. Cauchy stress, min: ',num2str(min(S3(:))),...
', max: ',num2str(max(S3(:)))]); %Add title
h1=plot(lambda_3,S1,'r-','LineWidth',20); %The 1 direction principal stress
h2=plot(lambda_3,S2,'g-','LineWidth',15); %The 2 direction principal stress
h3=plot(lambda_3,S3,'b-','LineWidth',10); %The 3 direction principal stress
hl=legend([h1 h2 h3],{'\sigma_1','\sigma_2','\sigma_3'}); %Add legend
set(hl,'FontSize',15,'Location','NorthEastOutside','Box','off'); %Adjust legend
axis tight; axis square; grid on; box on;
set(gca,'FontSize',15);
```



Visualize Jacobian

[7]:
```
%Visualize Jacobian
figure; hold on;
title(['Unconstrained form. Jacobian, min: ',num2str(min(J(:))),...
', max: ',num2str(max(J(:)))]); %Add title
h1=plot(lambda_3,J,'k-','LineWidth',20); %The 1 direction principal stress
hl=legend([h1],{'J'}); %Add legend
set(hl,'FontSize',15,'Location','NorthEastOutside','Box','off'); %Adjust legend
axis tight; axis square; grid on; box on;
set(gca,'FontSize',15);
```



Alternative solving method featuring interpolation

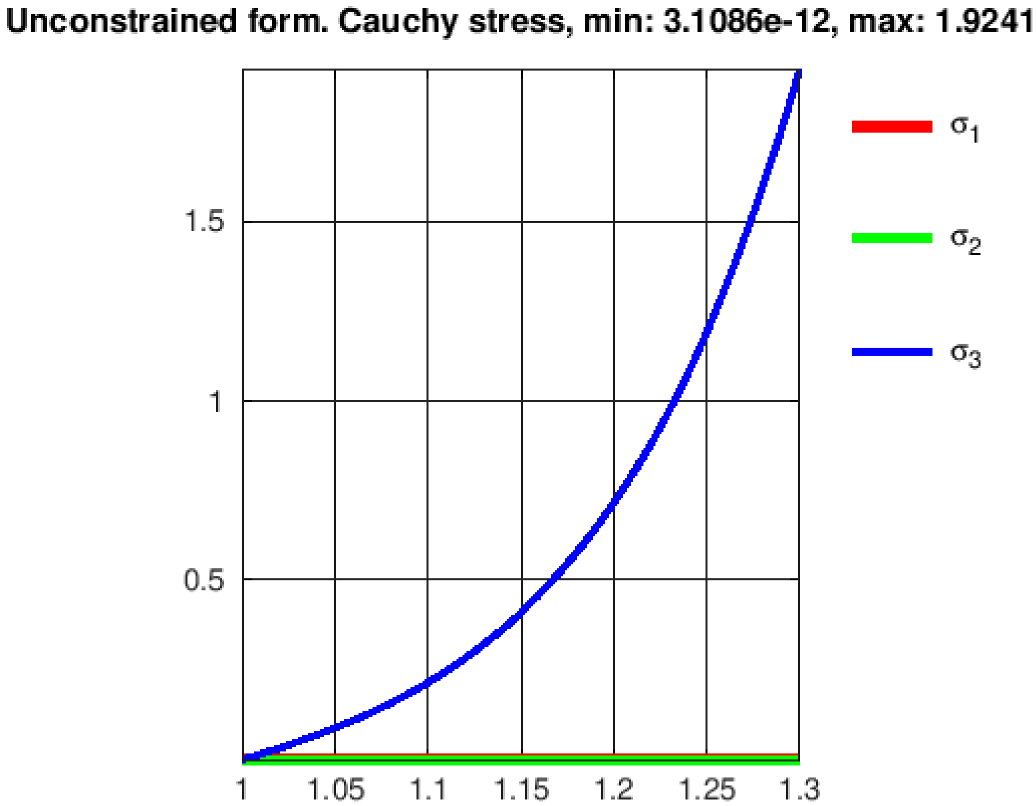
[8]:
```
%%
% For this approach the function for S1 is evaluate for a range of J which␣
↪should cover the required J.
% Where this graph crosses the x-axis S1(J)=0, and this point is approximated␣
↪using interpolation.

%Compute Jacobian given boundary conditions S1=S2=0
nTestPoints=100; %Set up a number of test values (more=better but slower)
J_test=linspace(0.9,1.1,nTestPoints); %The test J values
J=zeros(size(lambda_3)); %Initialize an arry of J values which are all zeros
for q=1:1:nDataPoints %Loop over all data points
    %Compute test stresses
    S1_test=kp*(J_test-1)+(1./J_test).*(c1/m1).*((sqrt(J_test./lambda_3(q)).
↪^m1)-1);

    %Find Jacobian for S1(J)=0 using interpolation
    % J(q)=interp1(S1_test,J_test,0,'linear'); %linear interpolation
    J(q)=interp1(S1_test,J_test,0,'pchip'); %piece-wise cubic hermite␣
↪interpolation
```

```
end

%Compute transverse stretches using J values
lambda_1=sqrt(J./lambda_3);
lambda_2=lambda_1; %Due to uniaxial loading

%Compute principal stresses (note, these are not ordered)
S1=kp*(J-1)+(1./J).*(c1/m1).*((lambda_1.^m1)-1);
S2=kp*(J-1)+(1./J).*(c1/m1).*((lambda_2.^m1)-1);
S3=kp*(J-1)+(1./J).*(c1/m1).*((lambda_3.^m1)-1);
```

Visualize stresses

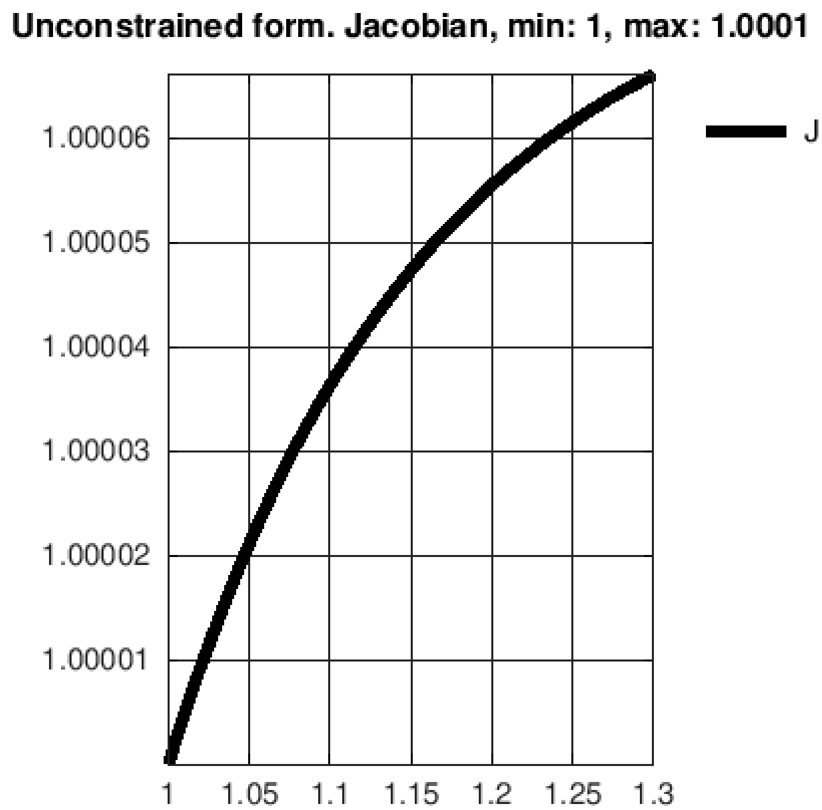
[9]:

```
%Visualize stress graphs
figure; hold on;
title(['Unconstrained form. Cauchy stress, min: ',num2str(min(S3(:))),...
', max: ',num2str(max(S3(:)))]); %Add title
h1=plot(lambda_3,S1,'r-','LineWidth',20); %The 1 direction principal stress
h2=plot(lambda_3,S2,'g-','LineWidth',15); %The 2 direction principal stress
h3=plot(lambda_3,S3,'b-','LineWidth',10); %The 3 direction principal stress
hl=legend([h1 h2 h3],{'\sigma_1','\sigma_2','\sigma_3'}); %Add legend
set(hl,'FontSize',15,'Location','NorthEastOutside','Box','off'); %Adjust legend
axis tight; axis square; grid on; box on;
set(gca,'FontSize',15);
```



Visualize Jacobian

[10]:
```
%Visualize Jacobian
figure; hold on;
title(['Unconstrained form. Jacobian, min: ',num2str(min(J(:))),...
', max: ',num2str(max(J(:)))]); %Add title
h1=plot(lambda_3,J,'k-','LineWidth',20); %The 1 direction principal stress
hl=legend([h1],{'J'}); %Add legend
set(hl,'FontSize',15,'Location','NorthEastOutside','Box','off'); %Adjust legend
axis tight; axis square; grid on; box on;
set(gca,'FontSize',15);
```


Unconstrained form. Jacobian, min: 1, max: 1.0001
1.00006
1.00005
1.00004
1.00003
1.00002
1.00001
1
1.05
1.1
1.15
1.2
1.25
1.3
J

1.6 Uncoupled formulations

- Given the numerical difficulties in handling truely incompressible behaviour (theoretically requiring $\kappa = \infty$) a special class of constitutive formulations has been developed referred to as *uncoupled* formulations.
- These uncoupled formulations are useful to model nearly-incompressible behaviour
- The term *uncoupled* relates to the fact that strain energy density Ψ is split into two additively seperated parts, namely:

1. A purely *deviatoric* (or isochoric = no volume change) part relating to shape change only Ψ_{dev}
2. A purely *volumetric* part relating to volume change only Ψ_{vol}

$$\Psi = \Psi_{dev} + \Psi_{vol}$$

1.6.1 Uncoupling the deformation

- To accomodate the split special shape and volume changing deformation metrics are required.
- The Jacobian or volume ratio J is already suitable to describe volume change ($J = 0.9$ means 10% volume loss, $J = 1.1$ means 10% volume gain).
- From the definition $J = \lambda_1\lambda_2\lambda_3$ one could imagine a single "spherical" average stretch λ which is the same in all directions such that:

$$J = \lambda_1\lambda_2\lambda_3 = \lambda\lambda\lambda = \lambda^3 \rightarrow \lambda = J^{\frac{1}{3}}$$

- To "take away" the effect of this spherical volume changing stretch λ from each of the stretches we can multiply them by $\frac{1}{\lambda} = J^{-\frac{1}{3}}$:

$$\tilde{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$$

- This introduces the *deviatoric stretches* denoted $\tilde{\lambda}_i$
- We can check if these deviatoric stretches really only change the shape by computing $\tilde{J}$ which should be 1 in magnitude for all stretches:

$$\tilde{J} = \tilde{\lambda}_1\tilde{\lambda}_2\tilde{\lambda}_3 = J^{-\frac{1}{3}}\lambda_1 J^{-\frac{1}{3}}\lambda_2 J^{-\frac{1}{3}}\lambda_3 = J^{-\frac{1}{3}}J^{-\frac{1}{3}}J^{-\frac{1}{3}}\lambda_1\lambda_2\lambda_3 = \frac{1}{J}J = 1$$

1.6.2 The uncoupled Ogden formulation

- The uncoupled Ogden formulation is given as:

$$\Psi(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3) = \frac{\kappa}{2}\ln(J)^2 + \sum_{a=1}^{N}\frac{c_a}{m_a^2}\left(\tilde{\lambda}_1^{m_a} + \tilde{\lambda}_2^{m_a} + \tilde{\lambda}_3^{m_a} - 3\right)$$

- Where

$$\Psi_{vol} = \frac{\kappa}{2}\ln(J)^2$$

and

$$\Psi_{dev} = \sum_{a=1}^{N}\frac{c_a}{m_a^2}\left(\tilde{\lambda}_1^{m_a} + \tilde{\lambda}_2^{m_a} + \tilde{\lambda}_3^{m_a} - 3\right)$$

- The principal Cauchy stresses σ_i can be computed from:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_{vol} + \boldsymbol{\sigma}_{dev}$$

- The volumetric stress $\boldsymbol{\sigma}_{vol}$ is derived from:

$$\boldsymbol{\sigma}_{vol} = p\mathbf{I}$$

  where the hydrostatic pressure is now derived directly from the constitutive equation:

$$p = \frac{\partial \Psi_{vol}}{\partial J}$$

  resulting in:

$$\boldsymbol{\sigma}_{vol} = \kappa \frac{\ln(J)}{J}\mathbf{I}$$

- The deviatoric stress $\boldsymbol{\sigma}_{dev}$ is derived from:

$$\sigma_{dev_i} = J^{-1}\lambda_i \frac{\partial \Psi_{dev}}{\partial \lambda_i} = J^{-1}\left(\tilde{\lambda}_i \frac{\partial \Psi_{dev}}{\partial \tilde{\lambda}_i} - \frac{1}{3}\sum_{j=1}^{3} \tilde{\lambda}_j \frac{\partial \Psi_{dev}}{\partial \tilde{\lambda}_j}\right)$$

- Since $J = \lambda_1\lambda_2\lambda_3$, and $\lambda_1 = \lambda_2$ (due to uniaxial loading in the 3rd direction) we can derive:

$$J = \lambda_1\lambda_2\lambda_3 = \lambda_1\lambda_1\lambda_3 = \lambda_1^2\lambda_3$$
$$\rightarrow \lambda_1 = \lambda_2 = \sqrt{\frac{J}{\lambda_3}}$$

- Using

$$\lambda_i \frac{\partial \Psi_{dev}}{\partial \tilde{\lambda}_i} = \sum_{a=1}^{N} \frac{c_a}{m_a} \tilde{\lambda}_i^{m_a}$$

  we can formulate

$$\sigma_{dev_i} = J^{-1} \sum_{a=1}^{N} \frac{c_a}{m_a} \left(\tilde{\lambda}_i^{m_a} - \frac{1}{3}\left(\tilde{\lambda}_1^{m_a} + \tilde{\lambda}_2^{m_a} + \tilde{\lambda}_3^{m_a}\right)\right)$$

$$\sigma_{dev_i} = J^{-1} \sum_{a=1}^{N} \frac{c_a}{m_a} \left(\tilde{\lambda}_i^{m_a} - \frac{1}{3}\left(\tilde{\lambda}_1^{m_a} + \tilde{\lambda}_2^{m_a} + \tilde{\lambda}_3^{m_a}\right)\right)$$

- And using $\lambda_1 = \lambda_2 = \sqrt{\frac{J}{\lambda_3}}$

$$\sigma_{dev_i} = J^{-1} \sum_{a=1}^{N} \frac{c_a}{m_a} \left(\tilde{\lambda}_i^{m_a} - \frac{1}{3}\left(2\left(\frac{J}{\lambda_3}\right)^{\frac{m_a}{2}} + \tilde{\lambda}_3^{m_a}\right)\right)$$

- Leading to:

$$\sigma_i = \kappa \frac{\ln(J)}{J} + J^{-1} \sum_{a=1}^{N} \frac{c_a}{m_a} \left(\tilde{\lambda}_i^{m_a} - \frac{1}{3} \left(2 \left(\frac{J}{\lambda_3} \right)^{\frac{m_a}{2}} + \tilde{\lambda}_3^{m_a} \right) \right)$$

- Numerical metods are now needed to solve for J such that $\sigma_1 = \sigma_2 = 0$
- **Note/tip**: To achieve nearly incompressible behaviour ($J \approx 1$), the bulk modulus κ is often set several orders of magnitude higher than the effective shear modulus (e.g. c_1 here). The codes here use $\kappa = 1000c_1$.

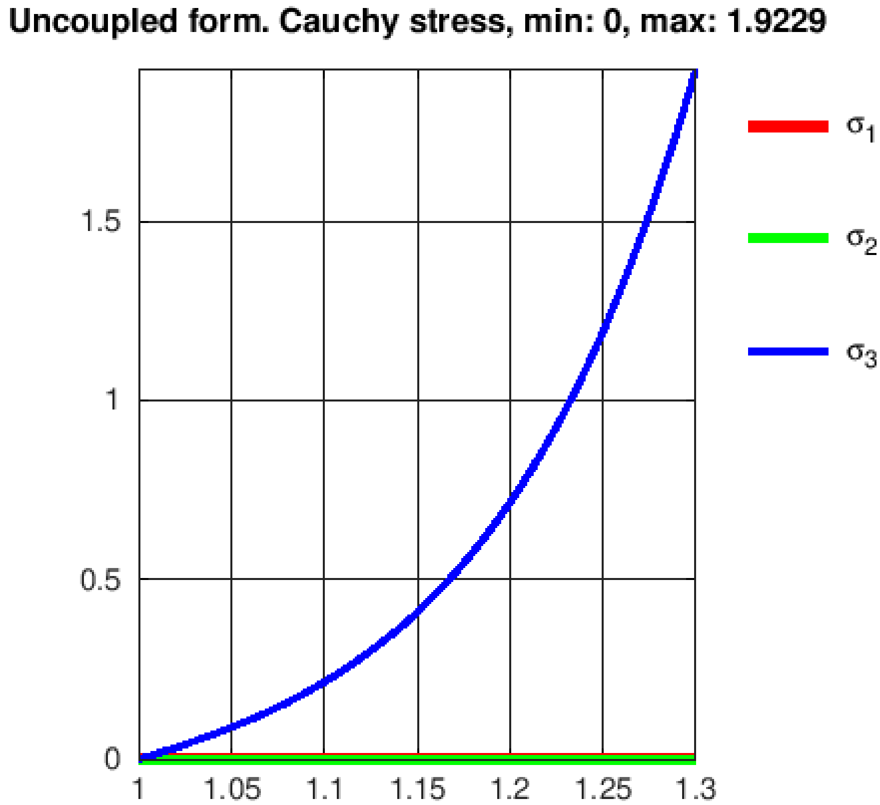
1.6.3 Numerical implementation

Compute stresses

[11]:
```
%% The uncoupled formulation
% One approach is to define a function for S1 and to find the J for which it is␣
↪zero.
% For this application the fzero function is useful to find J for S1(J)=0.

%Compute Jacobian given boundary conditions S1=S2=0
J=zeros(size(lambda_3)); %Initialize an arry of J values which are all zeros
for q=1:1:nDataPoints %Loop over all data points
    %Create stress function with current lambda
    S1_fun=@(J) k*(log(J)/J)+(1/J)*(c1/m1)*(sqrt(J/lambda_3(q))^m1...
    -((1/3)*(2*(J/lambda_3(q))^(m1/2)+lambda_3(q)^m1)));

    %Find Jacobian for zero stress, use J=1 as initial
    J(q)=fzero(S1_fun,1); %Find root of nonlinear function
end

%Compute transverse stretches using J values
lambda_1=sqrt(J./lambda_3);
lambda_2=lambda_1; %Due to uniaxial loading

%Compute principal stresses (note, these are not ordered)
S1=k*(log(J)./J)+(1./J).*(c1/m1).*(lambda_1.^m1-((1/3)*(lambda_1.^m1+lambda_2.
↪^m1+lambda_3.^m1)));
S2=k*(log(J)./J)+(1./J).*(c1/m1).*(lambda_2.^m1-((1/3)*(lambda_1.^m1+lambda_2.
↪^m1+lambda_3.^m1)));
S3=k*(log(J)./J)+(1./J).*(c1/m1).*(lambda_3.^m1-((1/3)*(lambda_1.^m1+lambda_2.
↪^m1+lambda_3.^m1)));
```

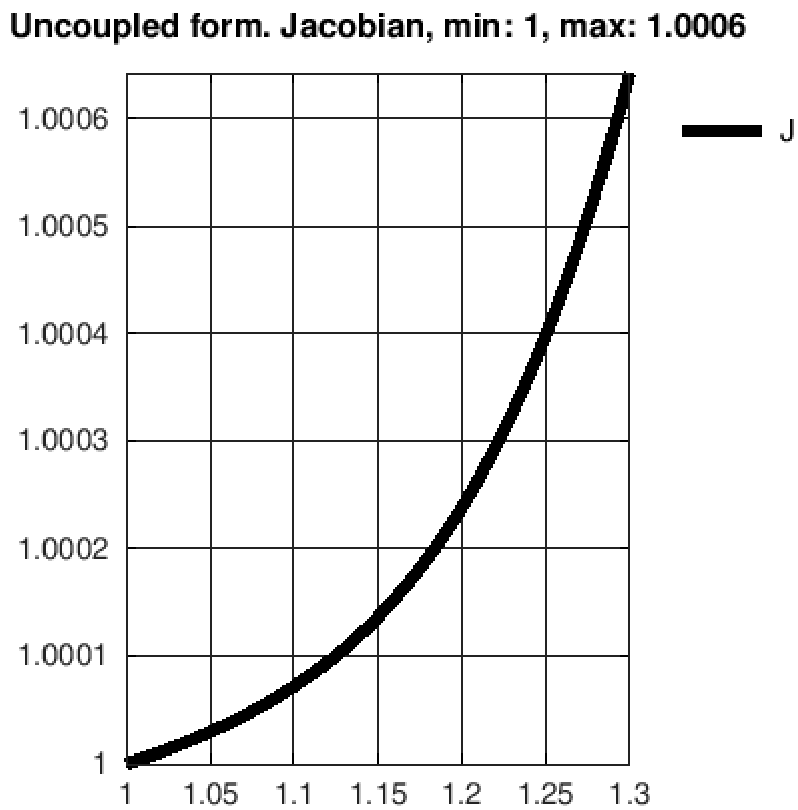
Visualize stresses

[12]:
```
%Visualize stress graphs
figure; hold on;
title(['Uncoupled form. Cauchy stress, min: ',num2str(min(S3(:))),...
', max: ',num2str(max(S3(:)))]); %Add title
h1=plot(lambda_3,S1,'r-','LineWidth',20); %The 1 direction principal stress
h2=plot(lambda_3,S2,'g-','LineWidth',15); %The 2 direction principal stress
h3=plot(lambda_3,S3,'b-','LineWidth',10); %The 3 direction principal stress
hl=legend([h1 h2 h3],{'\sigma_1','\sigma_2','\sigma_3'}); %Add legend
set(hl,'FontSize',15,'Location','NorthEastOutside','Box','off'); %Adjust legend
axis tight; axis square; grid on; box on;
set(gca,'FontSize',15);
```



Visualize Jacobian

[13]:
```
%Visualize Jacobian
figure; hold on;
title(['Uncoupled form. Jacobian, min: ',num2str(min(J(:))),...
', max: ',num2str(max(J(:)))]); %Add title
h1=plot(lambda_3,J,'k-','LineWidth',20); %The 1 direction principal stress
hl=legend([h1],{'J'}); %Add legend
set(hl,'FontSize',15,'Location','NorthEastOutside','Box','off'); %Adjust legend
axis tight; axis square; grid on; box on;
set(gca,'FontSize',15);
```


Uncoupled form. Jacobian, min: 1, max: 1.0006
1.0006
1.0005
1.0004
1.0003
1.0002
1.0001
1
1
1.05
1.1
1.15
1.2
1.25
1.3
J