NET v1.0:
A Framework to Simulate Permanent Damage in Elastomers under Quasi-static Deformations

Vahid Morovati, Roozbeh Dargazany
morovati@msu.edu, roozbeh@msu.edu

Michigan State University
Department of Civil and Environmental engineering

Copyright

Abstract

An analytical software for network evolution theory (NET) to describe the macroscopic behavior of filled rubber-like materials at different loading directions is presented. The software is based on a modular platform that is particularly designed to capture Mullins effect, permanent set, and deformation-induced anisotropy. Other inelastic features of elastomers can be modeled and added to the framework. Only the basic framework is covered in the NET v1.0, which is based on decomposition of the network to two parallel networks of pure rubber (CC) and a polymer-filler (PP).

Introduction

Here, the NET code is developed based on the network evolution theory presented by Dargazany and Itskov [1]. Employing the concept of network decomposition, they have decomposed the rubber into two parallel networks CC and PP, where CC was a hyper-elastic network and PP was responsible for Permanent damage and its features, such as the Mullins effect, permanent set and deformation-induced anisotropy. The main advantages of this model are simplicity, a small number of physically-motivated material parameters, fast fitting procedure, thermo-dynamical consistency and an easy finite element implementation procedure. The procedure of implementation into FE softwares were developed later [2], in which the network evolution model calculates the contribution of the inelastic features based on the free energy function. This code is organized as Table 1.

1. Problem Description of the Induced Anisotropy
2. Statistical mechanics of a single chain
3. Aggregate-polymer debonding
4. Network rearrangement
5. Numerical integration
6. Strain amplification
7. Three-dimensional generalization
8. Macroscale behavior
9. Plot an Example

Table 1 Essential steps in an analytical solution of network evolution model

- Here, ***NET v1.0*** is developed in Maple (R) to simulate the constitutive behavior of elastomers in terms of stress-strain curves.
- The code output is first Piola-Kirchhoff or nominal stress tensor, ***P***, while the input is mainly the deformation gradient tensor and its history in the past.
- This code developed based on uni-axial tensile loading in a specific direction, ω, after the material was subjected to uni-axial tensions in other directions.
- This set-up of the loading in different direction designated to show Mullins effect, permanent set and deformation-induced anisotropy.

Initializations

Load packages and define initial data by clicking on the icon below.

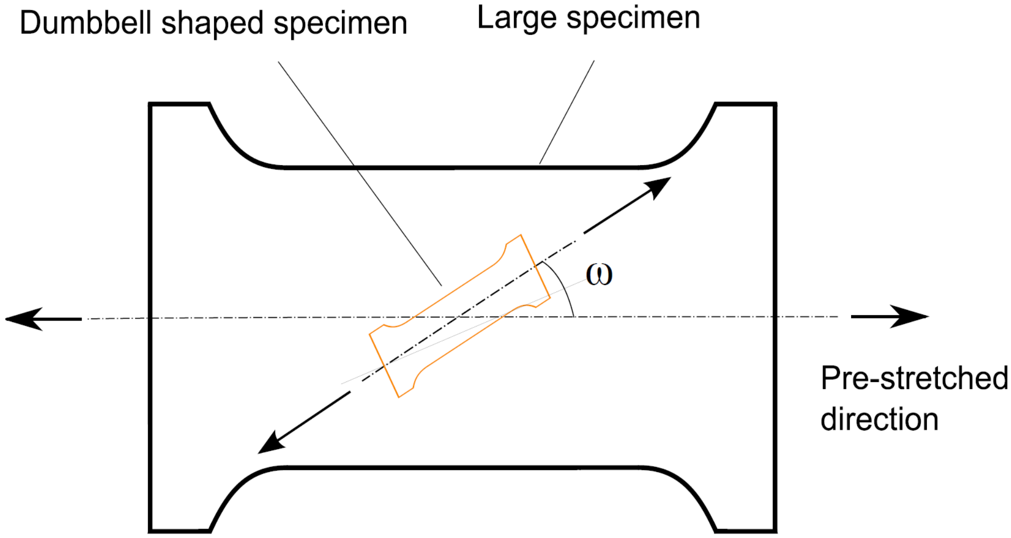
Problem Description of the Induced Anisotropy

Deformation-induced anisotropy is coded as the changes in the properties of the material in a specific direction, ω, after the material was subjected to uni-axial tensions in other directions. Multi-axial loading is considered through the following sequential loading pattern:

- i^{th} uni-axial tension is applied in direction ω_i with the maximum stretch of λ_max. The sample is unloaded to the stress-free state.

- To prepare the testing in the other direction, a new sample is cut from the current sample in a specific angle ω_(i+1)
- $(i+1)^{th}$ uni-axial tension is applied in direction ω_(i+1) until the maximum stretch of λ_max_(i+1).

Permanent damages will be transferred from previous loading steps to the next step, similar to if the sample (i+1) was cut out of the sample (i) as depicted in the Figure. In the following experiment, history of streches in different directions stored and used for calculation of constitutive model of current state.



Deformation gradiant for sample loaded in direction (ω),

$$\hat{F} := \left(\begin{bmatrix} \lambda & 0 & 0 \\ 0 & \frac{1}{\sqrt{\lambda}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{\lambda}} \end{bmatrix}\right):$$

$$RM := \left(Student[LinearAlgebra][RotationMatrix]\left(\mathrm{convert}\left(\omega * \mathrm{degrees, radians}\right), `<,>`(0, 0, 1)\right)\right):$$

$$FF := unapply\left(RM^{+}.\hat{F}.RM, \lambda, \mathrm{omega}\right):$$

where RM^{+} is the tranpose pf the rotation matrix RM.

As an example, loading is asumed to be applied in ***y directin.***

$$\omega_0 := 90 :$$
$$F := FF\left(\chi, \omega_0\right);$$
$$J := LinearAlgebra[Determinant](F) :$$

$$\begin{bmatrix} \frac{1}{\sqrt{\chi}} & 0 & 0 \\ 0 & \chi & 0 \\ 0 & 0 & \frac{1}{\sqrt{\chi}} \end{bmatrix} \tag{5.1}$$

The history of streches at different directions are stored in two vector ($\lambda\_history$, $\omega\_history$). In the case of induced anisotropy, number of loading in differnt directions in the past, n_history, the direction of loading, omega_histroy, and their correponding stretch histrory, lambda_histroy, are given as

$$n\_history := 1 :$$
$$\lambda\_history := [1.75] :$$
$$\omega\_history := [0] :$$

Note: The algorithm is capable to handle different loading scenarios . The problem definition can be changed to any specific sample geometry or test by replacement of the deformation gradient and the boundary conditions. Different loading scenarios can be modeled in this framework by changing $\hat{F}$ in this section. The boundary condition is generally considered to be $T_{33} = 0$.

- **Bi-axial tension**

$$\hat{F} := \left(\begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \frac{1}{\lambda^2} \end{bmatrix}\right)$$

- **Shear**

$$\hat{F} := \left(\begin{bmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

- **Compression**, the deformation gradient is the same as uni-axial tension and software can model it by adjusting plotting parameters only.

Statistical mechanics of a single chain

According to [1], the probability of existence of a polymer chain in which none of the segments between numbers 1 and n are attached to an aggregate surface, is given by:

```
A := -alpha*R̄^2/n − kappa*sqrt(alpha/Pi)*(2*sqrt(n)*exp(
    -alpha*R̄^2/n) + 2*R̄*sqrt(Pi*alpha)*(erf(R̄*sqrt(alpha
    /n)) - erf(R̄*sqrt(alpha))) - 2*exp(-alpha*R̄^2)) :
P := unapply(kappa*sqrt(alpha/Pi/n)*exp(A), n, R̄);
```

$$(n, \bar{R}) \to \kappa \sqrt{\frac{\alpha}{\pi n}}\, e^{-\frac{\alpha \bar{R}^2}{n} - \kappa \sqrt{\frac{\alpha}{\pi}} \left(2 \sqrt{n}\, e^{-\frac{\alpha \bar{R}^2}{n}} + 2 \bar{R} \sqrt{\pi \alpha} \left(\operatorname{erf}\left(\bar{R} \sqrt{\frac{\alpha}{n}}\right) - \operatorname{erf}\left(\bar{R} \sqrt{\alpha}\right)\right) - 2 e^{-\alpha \bar{R}^2}\right)} \qquad \textbf{(6.1)}$$

In this equation,

κ= the average area of active adsorption sites available for one bond

$\bar{R}$=normalized end-to-end distance with respect to the segment length (R/l)

and

for the valence angle θ = 70.5°.

```
alpha := ((1-cos(θ))/(1 + cos(θ)))^2;
θ := convert(70.5*degrees, radians) :   alpha := 0.27 :
```

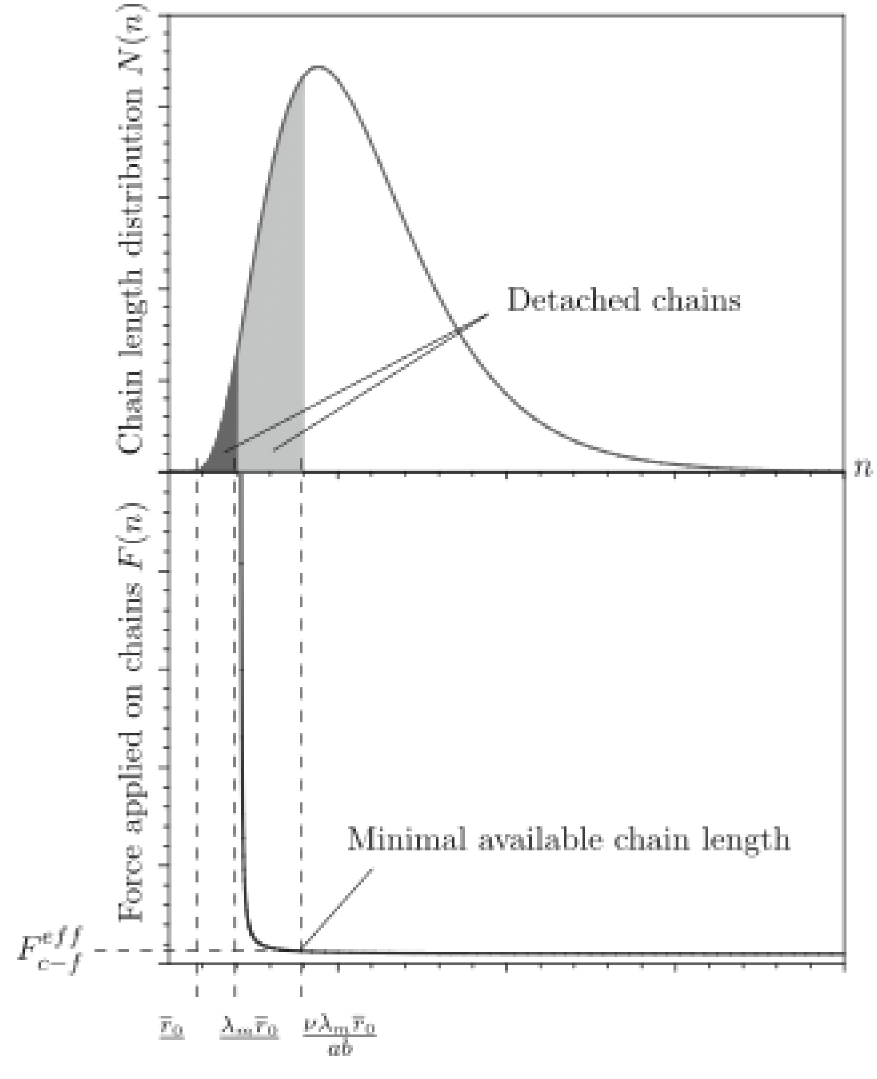
$$\frac{(1 - \cos(\theta))^2}{(1 + \cos(\theta))^2} \qquad \textbf{(6.2)}$$

PP Netowrk

It host all damages, The damages are considered to be resulted from two simoltanous mechanisms

1.Aggregate-polymer debonding

In the course of deformation, polymer chains begin to slide on or debond from the aggregates. This debonding starts with the shortest chain and gradually involves longer and longer chains.



The length of the shortest available chain in the deformed subnetwork is then obtained by

$$\boxed{n_{cf} := \frac{\nu \cdot \overset{d}{\lambda}_{\max} \cdot R0}{a\, b} :}$$

In aforementioned equation, ν > 1 denotes a sliding ratio and is a material parameter and $\overset{d}{\lambda}_{\max}$

denotes the maximal micro-stretch reached in direction ***d***. upper bond of the relative length is $n_{\max}$ (material parameter).

Accordingly, the set of available relative lengths of chains bounded to aggregates in the direction d can be expressed by

$$DD\left(\overset{d}{\lambda}_{\max}\right) = \left\{ n \,\middle|\, \nu \frac{\overset{d}{\lambda}_{\max} R0}{a\, b} \le n \le n_{\max} \right\}$$

$$DD := unapply\left(n_{cf}..n\_max, \overset{d}{\lambda}_{\max}\right);$$

$$a := \frac{\alpha^2}{\cos\left(\frac{\theta}{2}\right)} : a := 2.44 :$$

$$b := \frac{\cos\left(\frac{\theta}{2}\right)^2}{\alpha^2} : b := 0.33 :$$

$$y1 \rightarrow \frac{\nu\, y1\, R0}{a\, b} ..n\_max \tag{7.1.1}$$

2. Network rearrangement

The concept of chain rearrangement in a rubber network suggests that the detachment of chains from the aggregate surface does not necessarily result in the complete loss of their role in the network entropic energy, but it may also lead to the activation of some new segments. Thus, all in all, one can assume that the total number of active segments remains constant. This assumption yields

$$\hat{N}\left(n, \bar{r}\right) = N_0\, \Phi\left(\overset{d}{\lambda}_m\right) \; P\left(n, \bar{r}\right) = N_0\, \Phi\left(\overset{d}{\lambda}_m\right) \hat{P}(n)$$

where $\Phi(\overset{d}{\lambda}_{\max})$,

```
Phi := unapply(1 / int(N * P(N, R0), N=DD(x)), x) :
```

$$\hat{N} := unapply\left(N0 \cdot \mathrm{Phi}\left(\overset{d}{\lambda}_{\max}\right), \overset{d}{\lambda}_{\max}\right);$$

$$y1 \rightarrow N0 \Bigg/ \left(\int_{1.241927472\, \nu\, y1\, R0}^{n\_max} \left(0.2931615071\, N \kappa \sqrt{\frac{1}{N}}\, \mathrm{e}^{-\frac{0.27\, R0^2}{N}} - 0.2931615071\, \kappa \left(2 \sqrt{N}\, \mathrm{e}^{-\frac{0.27\, R0^2}{N}} + 1.841988074\, R0 \left(\mathrm{erf}\left(0.5196152423\, R0 \sqrt{\frac{1}{N}} \right) - \mathrm{erf}(0.5196152423\, R0) \right) - 2\, \mathrm{e}^{-0.27\, R0^2} \right) \mathrm{d}N \right) \right) \quad \textbf{(7.2.1)}$$

In this equation, We consider R0 and N0 as material constants.
So, we can calculate the energy of a subnetwork of PP network in direction ***d*** as:

$$\overset{d}{\Psi} = N_0\, \Phi\left(\overset{d}{\lambda}_m\right) \int_{DD\left(\overset{d}{\lambda}_{\max}\right)} \hat{P}\,(n)\, \psi_c\left(n, \bar{R}\, \overset{d}{\lambda}\right) \mathrm{d}n$$

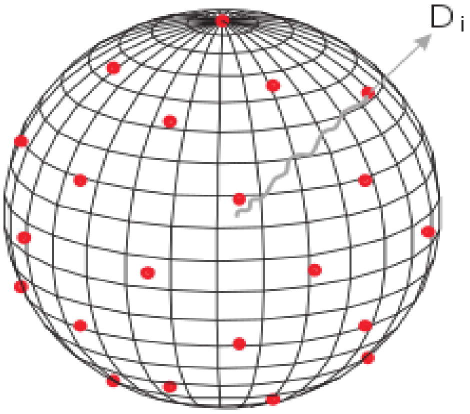
Numerical integration

For 3D generalization, we need to sum microscopic strain energies of all active chains available within the network which can be calculated by integration over the unit sphere.
Integeration over the unit sphere can be carried out numerically by

$$\Psi_0 = \frac{1}{A_s}\int_s W_0^d \, \mathrm{d}u^d = \sum_{i=1}^{k} W_0^{d_i} w_i$$

where ω_i : are the weight factors corresponding to the collocation directions d_i : (i =1.. k). A set of k = 21 integration points on the half-sphere is chosen

$$\begin{aligned}
&\mathit{Int\_data} := \mathit{Matrix}(\mathit{readdata}(\text{"intpoints.dat"}, 4)) :\\
&x_c, y_c, z_c, w_c := \mathit{Column}(\mathit{Int\_data}, [1..4]) :\\
&\mathit{Num}_c := \mathit{Dimension}(w_c) :
\end{aligned}$$



Strain amplification

In a filler-reinforced rubber network, the stretch of the polymer chains between aggregates (micro-stretch) generally exceeds the stretch applied to the rubber matrix (macrostretch).
This strain amplification concept is based on the fact that filler aggregates are considerably stiffer than the polymer chains connecting them together.

$$\begin{aligned}
&\Lambda := \mathit{unapply}\left(\frac{\chi - Cc^{Pp}}{1 - Cc^{Pp}}, \chi\right);\\
&Pp := \frac{1}{3} :\\
&Cc := 0.2 :
\end{aligned}$$

$$\chi \rightarrow \frac{\chi - Cc^{Pp}}{1 - Cc^{Pp}} \qquad \textbf{(9.1)}$$

Three-dimensional generalization

- Calculation of parametrs in direction *d*

In order impliment model, we should calculate stretch 'χ_i' and maximum stretch 'χ_i_m' in the integration directions.

```
d := Vector([x, y, z]) :
DOD := unapply(OuterProductMatrix(d, d), x, y, z) :
χ_d := unapply(sqrt(Transpose(d).F.F.d), χ, x, y, z) :
χ_i := seq(χ_d(χ, x_c[j], y_c[j], z_c[j])  ,j = 1 ..Num_c) :
```

```
χ_di_max := unapply(sqrt(Transpose(d).FF(χm, Ω).FF(χm, Ω)
     .d), χm, Ω, x, y, z) :
if n_history ≥ 1 then
        χ_i_max := seq(max(1, seq(evalf(χ_di_max(λ_history[k],
     ω_history[k], x_c[j], y_c[j], z_c[j])),  k = 1 ..n_history)), j = 1
     ..Num_c) :
 else
        χ_i_max := seq(1, j = 1 ..Num_c) :
 end if:
χ_di_m := seq(χ_di_max(χm, ω_0, x_c[j], y_c[j], z_c[j])  ,j = 1 ..Num_c) :
χ_i_m := seq( (max(χ_i_max[j], χ_i[j], χ_di_m[j])) ,j = 1
     ..Num_c) :
```

Macroscale behavior

Based on [2], The constitutive equation for the first Piola–Kirchhoff stress tensor P can be written as.

$$P = \sum_{i=1}^{k} \left(P_{cc}(\boldsymbol{D_i}) + P_{pp}(\boldsymbol{D_i})\right) \frac{w_i}{\substack{d_i \\ \chi}} J^{-\frac{1}{3}} \mathbf{F} : \left(\boldsymbol{D_i} \times \boldsymbol{D_i}\right)$$

where

$$P_{cc}(x) = N_c \sqrt{b n_c}\, L^{-1}\left(\frac{\frac{x}{\chi}}{\sqrt{b n_c}} \right),$$

$$P_{pp}(x) = \frac{\bar{R}}{a}\,\frac{N_0}{1-C^p}\,\Phi\left(\overset{x}{\lambda}_m\right)\int_{DD\left(\overset{x}{\lambda}_{\max}\right)} \hat{P}(n)\,L^{-1}\left(\frac{\bar{R}\,\overset{x}{\lambda}}{abn}\right)\mathrm{d}n$$

Inverse Langevin function, a proper approximation approach for the inverse Langevin function can be chosen depending on the elongation range of polymer chains, Padé approximants show better agreement with the exact values when υ<1.04. Generally in filled elastomers the initial guess is ν~1.01

if $\nu < 1.04$ **then**

$$L_{inv} := (x) \rightarrow \frac{3 \cdot x}{1 - x^3} :$$

else

$$L_{inv} := (x) \rightarrow 3 \cdot x + \frac{9}{5} \cdot x^3 + \frac{297}{175} \cdot x^5 + \frac{1539}{875} \cdot x^7 + \frac{126117}{67375} \cdot x^9 + \frac{43733439}{21896875} \cdot x^{11} + \frac{231321177}{109484375} \cdot x^{13} + \frac{20495009043}{9306171875} \cdot x^{15} + \frac{1073585186448381}{476522530859375} \cdot x^{17} + \frac{4387445039583}{1944989921875} \cdot x^{19} :$$

end if:

```
L__inv := proc (x) options operator, arrow; 3*x/
(1-x^3) end proc
```

$$x \rightarrow \frac{3\,x}{1 - x^3} \tag{11.1}$$

$$P_c := seq\left(Nc * \sqrt{b * nc} * L_{inv}\left(\frac{\chi\_i[j]}{\sqrt{b * nc}}\right), j = 1\,..Num_c\right) :$$

$$P_p := seq\left(\frac{R0}{a} \cdot \frac{N0}{1 - Cc^{Pp}} \cdot \mathrm{Phi}(\Lambda(\chi\_i\_m[j])) \cdot int\left(L_{inv}\left(\frac{R0 * \Lambda(\chi\_i[j])}{a \cdot b \cdot n}\right) * P(n, R0), n = DD(\Lambda(\chi\_i\_m[j]))\right), j = 1\,..Num_c\right) :$$

Due to the incompressibility of elastomers, one can use the following lagrange multiplier, p, to satisfy the icompressibility condition

$$T = \frac{\partial \Psi_m}{\partial F} - p\,F^{-T} = \frac{\partial \Psi_{pp}}{\partial F} + \frac{\partial \Psi_{cc}}{\partial F} - p\,F^{-T}$$

For uniaxial test sample in ω direction, we have $T_{33} = 0$,

$$DOD\_p := F.DOD(x, y, z) :$$

$$p1 := unapply(DOD\_p[3, 3] \cdot F[3, 3], x, y, z) :$$

$$DD\_n := unapply\Big(eval\Big((RM^{+}.(F.DOD(x, y, z) - p1(x, y, z) \cdot MatrixInverse(F^{+})).RM) [1, 1], \omega = -\omega_0\Big), x, y, z\Big) :$$

$$DOD\_i := seq\big(DD\_n\big(x_c[j], y_c[j], z_c[j]\big)\ \ , j = 1 ..Num_c\big) :$$

where DOD_p represents the tensor $\mathbf{F} : \left(\boldsymbol{D}_i \times \boldsymbol{D}_i\right)$, pl the lagrange ultiplier p, DD_n is the 1,1 component of the tensor $\mathbf{F} : \left(\boldsymbol{D}_i \times \boldsymbol{D}_i\right)$ at the current loading direction in the global coordinates. Here, DOD_i represents the stretch field of the current sample which is stretched in direction D.

Summation of caculated stress in different direction

$$P\_n := seq\left(\left(\left(P_c[k] + P_p[k]\right) \cdot \frac{w_c[k]}{\chi\_i[k]} \cdot J^{-\frac{1}{3}} \cdot DOD\_i[k]\right), k = 1 ..Num_c\right) :$$

$$Pt := \sum_{i=1}^{Num_c} P\_n[i] :$$

$$First\_piola := unapply(Pt, \chi, \chi m) :$$

where P_n[i] reresents the stress component in each integration direction.

Plot an Example

Here we used the data from tabel 3 reference [2] to illustrate capability of this algorithm.
In this part we assumed material parameters as bellow an ploted stress and strain graph.

$$\kappa := 14.824 :$$
$$\nu := 1.0065 :$$
$$n\_max := 100 :$$
$$R0 := 6.406 :$$
$$Nc := 1.8141 :$$
$$nc := 100 :$$
$$N0 := 2.835 :$$

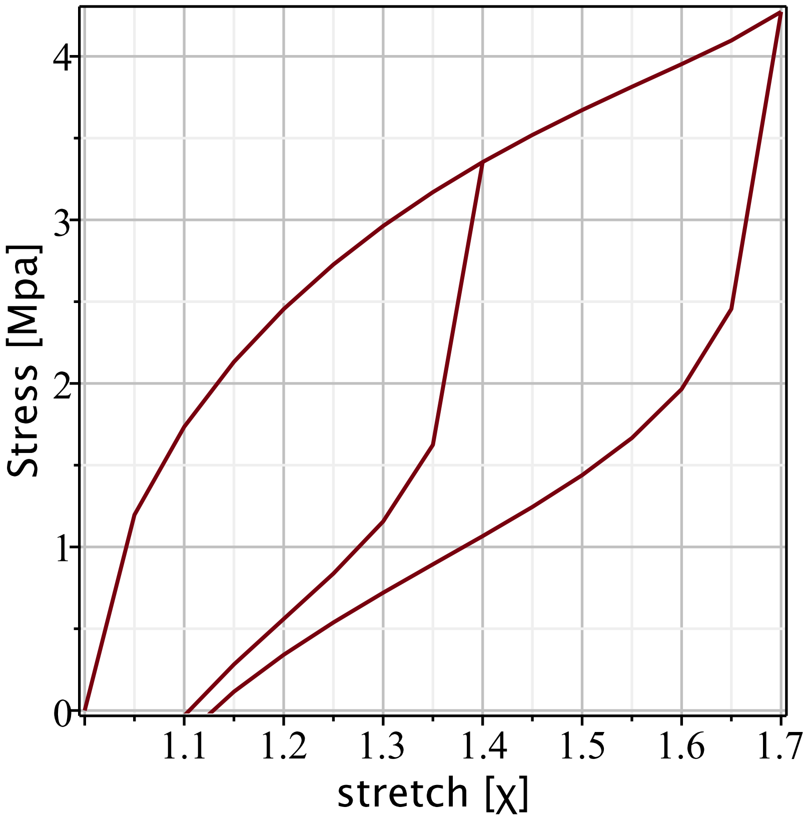
```
plot_interval := 0.05 :
Final_stretch := 1.70 :
sigma_max := First_piola(Final_stretch , Final_stretch) :
unloading_lim_1 := Final_stretch :
unloading_lim_2 := 1.4 :
```

```
F1[1] := plot( {seq( [1 + x * plot_interval, evalf (First_piola(1 + x
      * plot_interval, 1 + x * plot_interval) ) ], x = 0 .. (Final_stretch - 1)
      / plot_interval) }, x = 1 ..Final_stretch, y = 0 ..sigma_max) :

F1[2] := plot( {seq( [1 + x * plot_interval, evalf (First_piola(1 + x
      * plot_interval, unloading_lim_1) ) ], x = 0 .. (unloading_lim_1 - 1)
      / plot_interval) }, x = 1 ..Final_stretch, y = 0 ..sigma_max) :

F1[3] := plot( {seq( [1 + x * plot_interval, evalf (First_piola(1 + x
      * plot_interval, unloading_lim_2) ) ], x = 0 .. (unloading_lim_2 - 1)
      / plot_interval) }, x = 1 ..Final_stretch, y = 0 ..sigma_max) :
```

```
display(F1[kk] $ kk = 1 ..3, axes = boxed, labels = ["stretch [χ]",
      "Stress [Mpa]"], labelfont = ["calibri", "bold", 15], labeldirections
      = ["horizontal", "vertical"], labelfont = ["HELVETICA", 12],
      axesfont = ["calibri", "ROMAN", 12], axis = [gridlines = [colour
      = gray, majorlines = 2]]) ;
```



Refrences

[1] Dargazany, Roozbeh, and Mikhail Itskov. "A network evolution model for the anisotropic Mullins effect in carbon black filled rubbers." International Journal of Solids and Structures 46, no. 16 (2009): 2967-2977.

[2] Dargazany, Roozbeh, Vu Ngoc Khiêm, Uwe Navrath, and Mikhail Itskov. "Network evolution model of anisotropic stress softening in filled rubber-like materials: Parameter identification and finite element implementation." Journal of Mechanics of Materials and Structures 7, no. 8 (2013): 861-885.