Fatigue simulation based on Wang 2018.

Teo Yu Jie

Nanyang Technological University, Singapore
School of Mechanical and Aerospace Engineering
A200021@e.ntu.edu.sg

6th January 2023

Contents

- The code is first written in MATLAB.
- Half of the code is implemented in FORTRAN subroutine. The other half is retained for curve fitting.

The code to import the `.csv` file is given as follows:

```
        % Imports N = 1, eta = 1, lambda_max = 4.5, s-lambda loading curve.
dat1_imp    =   detectImportOptions('2d-s-lambda-eta-1-N-1-loading.csv');
dat1_dbl    =   readmatrix('2d-s-lambda-eta-1-N-1-loading.csv', dat1_imp);
load_x_lambda_Nr1     =   dat1_dbl(:,1);
load_y_lambda_Nr1     =   dat1_dbl(:,2);
```

This structure is recycled throught the script The indicative maximum $\lambda_{\max}$ or $N = 2$ will be used when importing data from different cycles.

We fit the first loading cycle to get the Ogden parameters (μ, and α)
We fit the decay curve for a constant $\lambda_{\max}$ to get κ and γ. Note that for our experiment, our gel is hardening due to dehydration, so γ is negative.

We fit the first **un**loading cycle to get the characteristic parameters r, c_0 β, nu_1, nu_3.
We fit the second **un**loading curve with what we have found from the previous parts to get c_u.
We fit the second **re**loading curve with what we have found from the previous parts to get c_r

We can fit the Ogden model Use Engauge digitiser to extract the data from the drawing of the line from the first loading cycle. We use the graph where $\lambda_{\max} = 4.5$ for consistency later, since the graph in the paper is the largest.
Use the graph of the first loading cycle. This is because $\eta = 1$ for the first loading cycle.
For the x-values, we use stretch λ given in the first cycle.
For the y-values we use the nominal stress s that depends on the stretch stretch λ given in the first cycle.
The fitting curve is hence of the form:

$$s = \mu \left(x^{\alpha-1} - x^{-\alpha-1} \right).$$

The independent variable is x, and the coefficients are the Ogden two parameters μ, and α.

To replicate the results from Wang [1], we force fit the Ogden parameters. By right, there should be no forcing of the upper limit and lower limit of the Ogden parameters.

```
fo_N1r        =   fitoptions(ft_N1r);
fo_N1r.Lower  =   [3.97, 2.158];
fo_N1r.Upper  =   [3.97, 2.158];

f_N1r         = fit(load_x_lambda_Nr1,load_y_lambda_Nr1,ft_N1r,
    fo_N1r);
co_N1r        = coeffvalues(f_N1r);
```

The code is to use x, and the fit its curve with the Ogden two parameters μ, and α given the nominal stress s are:

```
        ft_N1r    =   fittype(  'mu*(x^(alpha-1) - x^(-alpha-1))',...
                'independent', 'x', ...
                'coefficients', {'mu','alpha'});

fo_N1r         =   fitoptions(ft_N1r);
fo_N1r.Lower   =   [3.97, 2.158];
fo_N1r.Upper   =   [3.97, 2.158];

f_N1r          = fit(load_x_lambda_Nr1,load_y_lambda_Nr1,ft_N1r,fo_N1r);
co_N1r         = coeffvalues(f_N1r);

mu               = co_N1r(1,1);
alpha            = co_N1r(1,2);
```

We can fit the shakedown decay quickly. Use Engauge digitiser to extract the data from the drawing of the line relating the maximum stress and the number of cycles.

Use the graph when $\lambda_{\max} = 3.0$. This is because it is the most prominent of the three graphs.

For the x-values, we use the number of cycles N when $\lambda_{\max} = 3.0$.

For the y-values we use the nominal stress s that depends on the number of cycles when $\lambda_{\max} = 3.0$.

Note that $\eta = 1$ at the maximum stress. Derive the equation:

$$\frac{s_{\max}(N)}{s^0_{\max}} = \kappa N^{-\gamma}.$$

The independent variable is N, and the coefficients are the two shakedown decay parameters K, and γ. Note that $s^0_{\max}$ is constant.

The code is to use N, and the fit its curve with shakedown parameters κ, and γ given the maximum stress s at constant maximum stretch factor $\lambda_{\max}$ are:

```
ft_decay    =   fittype( 'kappa.*((N).^(-gamma))',...
    'independent', 'N', ...
    'coefficients', {'kappa','gamma'});

f_decay     = fit(decay_x,decay_y,ft_decay);
co_decay    = coeffvalues(f_decay);

kappa = co_decay(1,1);
gma = co_decay(1,2);
```

This anonymous function is **an approximate solution** for the nonlinear stress for the first cycle. You can derive this back substituting η and using the fact that $\lambda_2 = 1$, so $\lambda_1 = 1/\lambda_3$.

```
stress_anon = @(r, c0, beta, nu1, nu3, alpha, mu, lambda0, lambda)
    ...
(1 - (1./r) .* erf((((mu./alpha).*((lambda0.^alpha) + 1 + (lambda0.^-
    alpha) - 3) + (mu./2).*(nu1.*(lambda0.^2 - 1) + nu3.*((lambda0.^(-2)
     - 1)))) - ((mu./alpha).*((lambda.^alpha) + 1 + (lambda.^-alpha) -
    3)  + (mu./2).*(nu1.*(lambda.^2 - 1) + nu3.*((lambda.^(-2) - 1)))))
    ./(c0 + (beta .* ((mu./alpha).*((lambda0.^alpha) + 1 + (lambda0.^-
    alpha) - 3)  + (mu./2).*(nu1.*(lambda0.^2 - 1) + nu3.*((lambda0
    .^(-2) - 1)))))))) .* mu .* (lambda.^(alpha - 1) ...
- lambda .^(-alpha - 1)) + (1 - (1 - (1./r) .* erf((((mu./alpha).*((
    lambda0.^alpha) + 1 + (lambda0.^-alpha) - 3)  + (mu./2).*(nu1.*(
    lambda0.^2 - 1) + nu3.*((lambda0.^(-2) - 1)))) - ((mu./alpha).*((
    lambda.^alpha) + 1 + (lambda.^-alpha) - 3)  + (mu./2).*(nu1.*(lambda
    .^2 - 1) + nu3.*((lambda.^(-2) - 1))))).*(c0 + (beta .* ((mu./alpha)
    .*((lambda0.^alpha) + 1 + (lambda0.^-alpha) - 3)  + (mu./2).*(nu1.*(
    lambda0.^2 - 1) + nu3.*((lambda0.^(-2) - 1)))))))) .* mu .* ...
(nu1 .* lambda - nu3 .* (lambda .^ (-3)));
```

Please refer to the paper for the simplified equations. I only present the substitutions required to get the anonymous functions. We can derive the anonymous function as follows. You can derive this back substituting η and using the fact that $\lambda_2 = 1$, so $\lambda_1 = 1/\lambda_3$. This is based on our loading conditions and assuming the incompressibility of hydrogel where $\lambda_1\lambda_2\lambda_3 = 1$
We can substitute this into the Ogden damage function

$$\eta = \left\{1 - \frac{1}{r}\,\mathbf{erf}\left(\frac{W(\lambda_{\max,1}, 1) - W(\lambda_{\text{nom},1}, 1)}{c_0 + \beta W(\lambda_{\max,1}, 1)}\right)\right\} \tag{1}$$

We can then substitute this into the stress equation which is derived for our loading case:

$$\sigma = \eta\mu\left(\lambda^{\alpha-1} - \lambda^{-\alpha-1}\right) + (1-\eta)\,\mu\left(\nu_1\lambda - \nu_3\lambda^{-3}\right) \tag{2}$$

We can substitute this in.

$$\sigma = \left\{1 - \frac{1}{r}\,\mathbf{erf}\left(\frac{W(\lambda_{\max,1},1) - W(\lambda_{\text{nom},1},1)}{c_0 + \beta W(\lambda_{\max,1},1)}\right)\right\}\mu\left(\lambda^{\alpha-1} - \lambda^{-\alpha-1}\right)$$
$$+\left(1 - \left\{1 - \frac{1}{r}\,\mathbf{erf}\left(\frac{W(\lambda_{\max,1},1) - W(\lambda_{\text{nom},1},1)}{c_0 + \beta W(\lambda_{\max,1},1)}\right)\right\}\right)\mu\left(\nu_1\lambda - \nu_3\lambda^{-3}\right)$$

We know that the free energy function is the following when you use the fact the function for η and simplify W_1 and W_2. We assume that W is independent on η as a first approximation, Otherwise, iteration is required since η and W are codependent. This gives a good enough approximation to the hystersis curve i.e. $\eta \approx 0.5$.
More work will be needed to derive an anonymous function based on the first iteration of η, or to perform curve fitting.

$$W(\lambda_1, \lambda_2) \approx W_1(\lambda_1, \lambda_2) + W_2(\lambda_1, \lambda_2) = \left\{ \frac{\mu}{\alpha} \left(\lambda^{\alpha} + \lambda_{-\alpha} - 2 \right) + \frac{\mu}{2} \left(\nu_1 \left[\lambda^2 - 1 \right] + \nu_2 \left[\lambda^{-2} - 1 \right] \right) \right\}$$

Substituting this approximation in yields the anonymous function used in the program.

Using the unloading curve in the first hystersis cycle, the standard procedure for fitting is as follows:
We specify the damage parameters associated with the Ogden damage error function. The anonymous function is the damage function but with the parameters of the curve.

```
ft_N1u       =   fittype( stress_anon,...
        'coefficients', {'r','c0','beta','nu1','nu3'}, ...
        'independent', 'lambda', ...
        'problem', {'alpha', 'mu', 'lambda0'} ...
        );
```

The upper and lower limits of the parameters will need to be specified to ensure that the anisotropic Poisson's ratio and energy variables are nonnegative.

```
fo_N1u        =   fitoptions(ft_N1u);
fo_N1u.Lower  =   [1, 5, 0, 0.1, 0.1];
fo_N1u.Upper  =   [Inf, Inf, Inf, Inf, Inf];
```

We now fit the unloading curve. The problem statements are the parameters found from the Ogden two parameter model μ and α, as well as the maximum stretch of the data $\lambda_{\max}$.
Do note that μ is actually the shear modulus.

```
f_N1u          =    fit(load_x_lambda_Nu1, ...
                load_y_sigma_Nu1, ...
                ft_N1u, ...
                fo_N1u, ...
                'problem', {alpha, mu, lambdamax});
```

The coefficients found in the curve fit of the Ogden damage function will be used for the fitting of the second hysteresis cycle.
Therefore, we extract them accordingly here.

```
r           = co_N1u(1,1);
c0          = co_N1u(1,2);
beta        = co_N1u(1,3);
nu1         = co_N1u(1,4);
nu3         = co_N1u(1,5);
```

This anonymous function solves the nonlinear stress for the second cycle. Same derivation as before. **Notice the decay parameter in the anonymous function $\kappa N^{-\gamma}$.**

```
    stress_N_u_anon = @(cu, r, beta, nu1, nu3, alpha, mu, lambda0, kappa
        , gma, N, lambda) ...
kappa .* (N .^ (- gma)) .* ...
(1 - (1./r) .* erf((((mu./alpha).*((lambda0.^alpha) + 1 + (lambda0.^-
    alpha) - 3) + (mu./2).*(nu1.*(lambda0.^2 - 1) + nu3.*((lambda0.^(-2)
     - 1)))) - ((mu./alpha).*((lambda.^alpha) + 1 + (lambda.^-alpha) -
    3)  + (mu./2).*(nu1.*(lambda.^2 - 1) + nu3.*((lambda.^(-2) - 1)))))
    ./(cu + (beta .* ((mu./alpha).*((lambda0.^alpha) + 1 + (lambda0.^-
    alpha) - 3)  + (mu./2).*(nu1.*(lambda0.^2 - 1) + nu3.*((lambda0
    .^(-2) - 1))))))))) .* mu .* (lambda.^(alpha - 1) ...
- lambda .^(-alpha - 1)) + (1 - (1 - (1./r) .* erf((((mu./alpha).*((
    lambda0.^alpha) + 1 + (lambda0.^-alpha) - 3)  + (mu./2).*(nu1.*(
    lambda0.^2 - 1) + nu3.*((lambda0.^(-2) - 1)))) - ((mu./alpha).*((
    lambda.^alpha) + 1 + (lambda.^-alpha) - 3)  + (mu./2).*(nu1.*(lambda
    .^2 - 1) + nu3.*((lambda.^(-2) - 1))))))./(cu + (beta .* ((mu./alpha)
    .*((lambda0.^alpha) + 1 + (lambda0.^-alpha) - 3)  + (mu./2).*(nu1.*(
    lambda0.^2 - 1) + nu3.*((lambda0.^(-2) - 1))))))))) .* mu .* ...
(nu1 + lambda - nu3 + (lambda .^(-3)));
```

Using the unloading curve in the second hystersis cycle, the standard procedure for fitting is as follows:
We specify the damage parameters associated with the Ogden damage error function.
The only parameter we need to fit is c_u.

```
ft_Nu2    =   fittype( stress_N_u_anon,...
        'coefficients', {'cu'}, ...
        'independent', 'lambda', ...
        'problem', {'r','beta','nu1','nu3','alpha', 'mu', 'lambda0',
            'kappa', 'gma', 'N'} ...
        );
```

We decided to force the value of c_u in order to better replicate the results.

```
fo_Nu2          =   fitoptions(ft_Nu2);
fo_Nu2.Lower    =   25;
fo_Nu2.Upper    =   25;
```

We now fit the unloading curve. The problem statements are the parameters found from all of the previous parts.
Note that you must include the decay parameters κ, N, and γ. For the second cycle $N = 2$.

```
f_Nu2           =   fit(load_x_lambda_Nu2, ...
load_y_sigma_Nu2, ...
ft_Nu2, ...
fo_Nu2, ...
'problem', {r, beta, nu1, nu3, alpha, mu, lambdamax, kappa, gma, N})
    ;
```

Finally, assign the parameter c_u out.

```
co_Nu2          =   coeffvalues(f_Nu2);
cu              =   co_Nu2;
```

This anonymous function solves the nonlinear stress for the second cycle. Same derivation as before. **Notice the decay parameter in the anonymous function $\kappa N^{-\gamma}$.**

```
    stress_N_r_anon = @(cr, r, beta, nu1, nu3, alpha, mu, lambda0, kappa
        , gma, N, lambda) ...
kappa .* (N .^ (- gma)) .* ...
(1 - (1./r) .* erf((((mu./alpha).*((lambda0.^alpha) + 1 + (lambda0.^-
    alpha) - 3) + (mu./2).*(nu1.*(lambda0.^2 - 1) + nu3.*((lambda0.^(-2)
     - 1)))) - ((mu./alpha).*((lambda.^alpha) + 1 + (lambda.^-alpha) -
    3)  + (mu./2).*(nu1.*(lambda.^2 - 1) + nu3.*((lambda.^(-2) - 1)))))
    ./(cr + (beta .* ((mu./alpha).*((lambda0.^alpha) + 1 + (lambda0.^-
    alpha) - 3)  + (mu./2).*(nu1.*(lambda0.^2 - 1) + nu3.*((lambda0
    .^(-2) - 1))))))))) .* mu .* (lambda.^(alpha - 1) ...
- lambda .^(-alpha - 1)) + (1 - (1 - (1./r) .* erf((((mu./alpha).*((
    lambda0.^alpha) + 1 + (lambda0.^-alpha) - 3)  + (mu./2).*(nu1.*(
    lambda0.^2 - 1) + nu3.*((lambda0.^(-2) - 1)))) - ((mu./alpha).*((
    lambda.^alpha) + 1 + (lambda.^-alpha) - 3)  + (mu./2).*(nu1.*(lambda
    .^2 - 1) + nu3.*((lambda.^(-2) - 1))))).*(cr + (beta .* ((mu./alpha)
    .*((lambda0.^alpha) + 1 + (lambda0.^-alpha) - 3)  + (mu./2).*(nu1.*(
    lambda0.^2 - 1) + nu3.*((lambda0.^(-2) - 1)))))))) .* mu .* ...
(nu1 + lambda - nu3 + (lambda.^(-3)));
```

Using the unloading curve in the second hystersis cycle, the standard procedure for fitting is as follows:
We specify the damage parameters associated with the Ogden damage error function.
The only parameter we need to fit is c_r.

```
ft_Nr2          =   fittype( stress_N_r_anon,...
        'coefficients', {'cr'}, ...
        'independent', 'lambda', ...
        'problem', {'r','beta','nu1','nu3','alpha', 'mu', 'lambda0',
            'kappa', 'gma', 'N'} ...
        );
```

We decided to force the value of c_r in order to better replicate the results.

```
fo_Nr2          =   fitoptions(ft_Nr2);
fo_Nr2.Lower    =   45;
fo_Nr2.Upper    =   45;
```

We now fit the unloading curve. The problem statements are the parameters found from all of the previous parts.
Note that you must include the decay parameters κ, N, and γ. For the second cycle $N = 2$.

```
f_Nr2           =   fit(load_x_lambda_Nr2, ...
                load_y_sigma_Nr2, ...
                ft_Nr2, ...
                fo_Nr2, ...
                'problem', {r, beta, nu1, nu3, alpha, mu, lambdamax,
                    kappa, gma, N});
```

Finally, assign the parameter c_r out.

```
co_Nr2          =   coeffvalues(f_Nr2);
cr              =   co_Nr2;
```

This set of code makes the plots full screen.

```
res = get(0,'screensize');
fig = figure;
set(fig, 'position', res);
```

The first line specifys the plot for the first hystersis cycle.
Holding the plot is important.

```
subplot(1,2,1);

hold on;
```

The limit of the plot axes are specified with the following code.

```
xlim([1 4.5]);
ylim([-0.5 23]);
```

The plots of the experimental data are given. *u* means unloading. *r* means reloading/loading.
The text behind the plots are colors and tick labels.

```
N1u_expt    =   plot(load_x_lambda_Nu1,load_y_sigma_Nu1,'Ro');
N1u_fit     =   plot(f_N1u,'B-');
N1r_expt    =   plot(load_x_lambda_Nr1,load_y_lambda_Nr1,'Ro');
N1r_fit     =   plot(f_N1r,'B-');
```

The tick legends are specified as follows. Use the LaTeXinterpreter when you specify the location and labels.

```
lot1_legend = legend([N1u_expt N1u_fit N1r_expt, N1r_fit], ...
'Unloading (experimental)', ...
'Unloading (curve fit)', ...
'Loading (experimental)', ...
'Loading (curve fit)', ...
'Location','southeast','interpreter','latex');
```

This code gives the titles and axes labels.

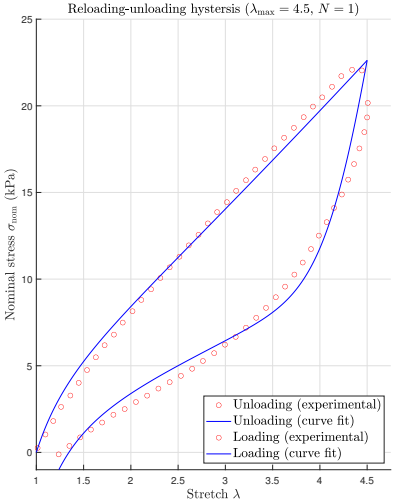
```
title1 = title('Reloading-unloading␣hystersis␣($\lambda_\mathrm{max}
    ␣=␣4.5$,␣$N␣=␣1$)','interpreter','latex');
xlabel1 = xlabel('Stretch␣$\lambda$','interpreter','latex');
ylabel1 = ylabel('Nominal␣stress␣$\sigma_{\mathrm{nom}}$␣(kPa)','
    interpreter','latex');
```

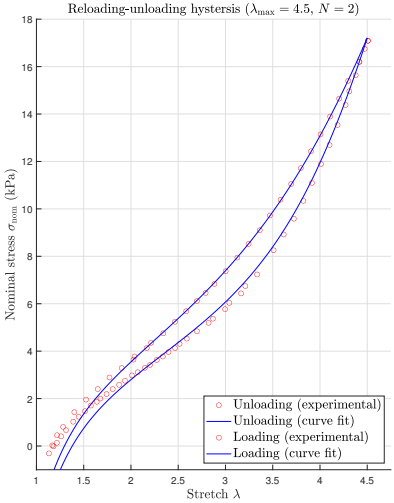
This code sets the fontsize for the axes and text used in the plot.

```
    ax = gca;
ax.FontSize = 12;

set(plot1_legend,'FontSize',16);
set(title1,'FontSize',16);
set(xlabel1,'FontSize',16);
set(ylabel1,'FontSize',16);
```

This code sets the overall limits of the entire graph. The previous one sets the limit of the plots.
A grid is imposed on the figure.
Then, use `hold off` to release the figures, and do the next plot for $N = 2$, which has very similar code since the program is standardised.

```
xlim([1 4.75]);
ylim([-1 25]);
grid on

hold off;

subplot(1,2,2);
```


Reloading-unloading hysteresis ($\lambda_{max} = 4.5$, $N = 1$)
Nominal stress σ_{nom} (kPa)
Stretch λ
Unloading (experimental)
Unloading (curve fit)
Loading (experimental)
Loading (curve fit)


Reloading-unloading hysteresis ($\lambda_{max} = 4.5$, $N = 2$)
Nominal stress σ_{nom} (kPa)
Stretch λ
Unloading (experimental)
Unloading (curve fit)
Loading (experimental)
Loading (curve fit)