Matlab Data Analysis for Single Fibers

Institute of Bioproducts and Paper Technology
Graz University of Technology

by
Michael Thoman

Contents

1 Data preparation **2**

2 Excel sheet **3**

3 Matlab script **5**

4 Detailed calculation of Young's modulus **10**
4.1 Initialization of parameters 10
4.2 Computing local slopes using a moving window 10
4.3 Identifying the most stable Young's modulus value 11
4.4 Smoothing the result by averaging over a stable range 11
4.5 Final computed Young's modulus 11
4.6 Advantages of this Method 11

5 Results **12**

1 Data preparation

The following Matlab script is used to evaluate the maximum values of stress and strain, as well as to calculate the Young's modulus and the total displacement. These data are then used to create stress-strain curves and a boxplot of the calculated Young's modulus. For the evaluation, the cross-sectional areas and the length of the respective fibers must be determined in advance and the raw data of force and displacement must be available in a txt. file. The raw data is exported from the trios files of the DMA measurement as shown in the following figure and saved in a text file.

Force	Total Displacement
N	mm
1.11293e-3	-0.0237229
3.52859e-4	-0.0235385
5.33716e-5	-0.0234603
-7.72545e-5	-0.0234342
1.91619e-4	-0.0233280
1.32989e-4	-0.0232144
1.36923e-4	-0.0231883
1.58744e-4	-0.0230710
1.73506e-4	-0.0229573
2.63774e-4	-0.0229369
1.73702e-4	-0.0228381
1.92484e-4	-0.0226910
3.85178e-4	-0.0226817

Figure 1: txt. File

The matlab script requires that the text files with the raw data and the Excel spreadsheet used, into which the calculated data is written after running the program, are in the same folder in the same directory.

Sup_13.txt	20.08.2024 09:38	Textdokument	414 KB
Sup_14.txt	20.08.2024 09:39	Textdokument	216 KB
Sup_15.txt	20.08.2024 09:39	Textdokument	232 KB
Sup_16.txt	20.08.2024 09:40	Textdokument	460 KB
Sup_17.txt	20.08.2024 09:41	Textdokument	164 KB
Sup_18.txt	20.08.2024 09:42	Textdokument	333 KB
Sup_19.txt	20.08.2024 09:43	Textdokument	319 KB
Sup_20.txt	20.08.2024 10:08	Textdokument	413 KB
Sup_21.txt	20.08.2024 10:09	Textdokument	412 KB
Sup_22.txt	20.08.2024 10:10	Textdokument	375 KB
Table1.xlsx	26.02.2025 10:01	Microsoft Excel-Ar...	7 KB
Tee_01.txt	17.07.2024 13:38	Textdokument	512 KB
Tee_02.txt	17.07.2024 13:38	Textdokument	688 KB
Tee_03.txt	17.07.2024 13:44	Textdokument	602 KB
Tee_04.txt	17.07.2024 13:49	Textdokument	442 KB

Figure 2: Excel-Sheet and txt. Files in the Folder

2 Excel sheet

The required excel sheet is filled in according to the schema shown in Figure 3. For this purpose, the text files of the same name to be calculated are entered in the “Name” column, which were created in advance. In addition, the individually calculated cross-sectional areas and lengths (span length of the DMA) of the fibers are entered in the “Area” and “Length” columns. The name of the fiber is entered in the “Type” column. this name will later also be the title of the stress-strain curve and is used to generate a separate plot for each fiber type.

	A	B	C	D
1	Name	Area [µm^2]	Length [mm]	Type
2	Sup_01.txt	337,15	15	Sup. Merino
3	Sup_02.txt	337,15	15	Sup. Merino
4	Sup_03.txt	337,15	15	Sup. Merino
5	Sup_04.txt	337,15	15	Sup. Merino
6	Ult_01.txt	233,15	15	Ult. Merino
7	Ult_02.txt	233,15	15	Ult. Merino
8	Ult_03.txt	233,15	15	Ult. Merino
9	Ult_04.txt	233,15	15	Ult. Merino
10	Pol_01.txt	398,96	15	Polworth
11	Pol_02.txt	398,96	15	Polworth
12	Pol_03.txt	398,96	15	Polworth
13	Pol_04.txt	398,96	15	Polworth

Figure 3: Excel-Sheet before calculation

After the script is executed, the calculated data of the maximum values, as well as total displacement and Young's Modulus are automatically added to the table.

	A	B	C	D	E	F	G	H	I
1	Name	Area [µm^2]	Length [mm]	Type	MaxForce [N]	MaxStress [GPa]	TotalDispl [mm]	MaxStrain [-]	Emodul [GPa]
2	Sup_01.txt	337,15	15	Sup. Merino	0,066	0,197	0,594	0,624	4,349
3	Sup_02.txt	337,15	15	Sup. Merino	0,049	0,145	0,421	0,434	3,063
4	Sup_03.txt	337,15	15	Sup. Merino	0,082	0,245	0,603	0,602	4,206
5	Sup_04.txt	337,15	15	Sup. Merino	0,052	0,156	0,500	0,518	3,192
6	Ult_01.txt	233,15	15	Ult. Merino	0,043	0,185	0,525	0,525	4,655
7	Ult_02.txt	233,15	15	Ult. Merino	0,025	0,107	0,419	0,505	2,642
8	Ult_03.txt	233,15	15	Ult. Merino	0,069	0,296	0,296	0,319	8,424
9	Ult_04.txt	233,15	15	Ult. Merino	0,035	0,151	0,435	0,452	3,519
10	Pol_01.txt	398,96	15	Polworth	0,141	0,353	0,556	0,615	7,061
11	Pol_02.txt	398,96	15	Polworth	0,149	0,373	0,357	0,324	8,608
12	Pol_03.txt	398,96	15	Polworth	0,083	0,208	0,336	0,358	4,284
13	Pol_04.txt	398,96	15	Polworth	0,160	0,401	0,493	0,603	6,489

Figure 4: Excel-Sheet after calculation

3 Matlab script

Here is a detailed explanation of each section of the MATLAB program. At the beginning of the program, the Excel file "Table.xlsx" is read (Name of the Excel file in the script must be the same as in the work path). Four columns from the table are extracted and stored in MATLAB arrays:

- **file** contains the names of the data files that will be processed later
- **area** stores the cross-sectional area of the tested specimens
- **einsp** represents the clamping length of the specimens
- **ticker** serves as a grouping variable for the test series

```
filename='Table.xlsx';
A=readtable(filename,'sheet','Tabelle1');
file=table2array(A(:,1));
area=table2array(A(:,2));
einsp=table2array(A(:,3));
ticker=table2array(A(:,4));
```

Several arrays are initialized with zeros to store computed values later:

- **maxforce** stores the maximum measured force for each specimen
- **maxstress** stores the maximum stress
- **maxstrain** are used to store the maximum strain
- **Emodul** stores the calculated Young's modulus

```
maxforce=zeros(length(file),1);
maxstress=zeros(length(file),1);
maxstrain_displ=zeros(length(file),1);
maxstrain=zeros(length(file),1);
totaldispl=zeros(length(file),1);
Emodul=zeros(length(file),1);
```

The program iterates through all the files in the file list using a for-loop, processing each file sequentially. Since the data may belong to different groups, the program checks if ticker(i) differs from the previous value. If a new group is detected, a new plot window is opened, and the current dataset is completed. This ensures that related datasets are displayed in separate plots.

The current file from the text file list is read as a table. The first column, which contains force values, is smoothed using a Gaussian smoothing function over 50 data points to reduce unwanted fluctuations or noise. Simultaneously, displacement values from the second column are extracted.

```
C = readtable(string(file(i)));
force = table2array(smoothdata(C(:,2), 'gaussian', 50));
displacement = table2array(C(:,3));
```

The next step is to determine the stress and strain from the raw data using the following formula. Then the highest values for stress and force are identified and stored in the maxstress and maxforce arrays.

```
% Calculate stress and strain
stress = force / area(i);
strain = displacement / einsp(i);

% Determine maximum stress and force
maxstress(i) = max(stress);
maxforce(i) = max(force);
```

Since material failure often occurs after reaching maximum stress, the curve is truncated at this point. This ensures that only the values relevant for calculating material properties are retained:

```
for a=1:length(stress)
    if stress(a)==maxstress(i)
        break
    end
end

if length(stress)<a
    stressCut=stress;
    strainCut=strain;
else
    stressCut=stress(1:a);
    strainCut=strain(1:a);
end
```

To eliminate initial noise, all values below a very small threshold (0.00001) are removed. The curve is then adjusted so that the first value starts at zero. This ensures that all plots correctly begin at the origin.

```
threshold = 0.00001;
startIndex = find(stressCut >= threshold, 1);

if ~isempty(startIndex)
    stressCut = stressCut(startIndex:end);
    strainCut = strainCut(startIndex:end);
end

if ~isempty(stressCut) && ~isempty(strainCut)
    stressCut = stressCut - stressCut(1);
    strainCut = strainCut - strainCut(1);
end
```

After applying corrections, the total remaining strain is calculated. This is obtained by computing the difference between the first and last remaining points of the truncated strain curve.

```
if ~isempty(strainCut)
    totaldispl(i) = strainCut(end) - strainCut(1);
end
```

Young's modulus is calculated using a local linear regression method:

- A sliding window technique is used to compute slopes over consecutive points (In this case 20 points)
- A linear regression is performed for each window
- The highest and most stable slope is taken as Young's modulus

Since Young's modulus describes the slope of elastic deformation, an average over the most stable region of the highest slope is taken.

```
window_size = 20;
num_points = length(strainCut);
slopes = zeros(num_points - window_size, 1);

for j = 1:(num_points - window_size)
    strain_window = strainCut(j:j+window_size);
    stress_window = stressCut(j:j+window_size);

    p = polyfit(strain_window, stress_window, 1);
    slopes(j) = p(1);
end

[max_slope, max_idx] = max(slopes);
stable_range = max(max_idx-5,1):min(max_idx+5,length(slopes));
Emodul(i) = mean(slopes(stable_range));
```

For each specimen, a stress-strain curve is plotted. The stress values are converted to GPa for appropriate scaling. The plotted curves are stored so they can later be displayed in a dropdown menu.

```
h = plot(strainCut, stressCut*1000, 'DisplayName', file{i},'LineWidth', 1);
plotHandles = [plotHandles, h];
plotLabels{end+1} = file{i};
```

Once all calculations are completed, the computed values are organized in a table and saved back to the Excel file. The calculated values are written into columns E to I of the Excel file as previously shown in Figure 4.

```
T = table(maxforce, maxstress*1000, totaldispl, maxstrain,Emodul*1000, ...
'VariableNames', {'MaxForce [N]','MaxStress [GPa]','TotalDispl [mm]',...
'MaxStrain [-]','Emodul [GPa]'});

writetable(T,filename,'sheet','Tabelle1','Range','E:I');
```

To visually represent the calculated Young's modulus values, a boxplot is generated. The data is divided into different groups based on the ticker grouping variable.

- The median, first and third quartiles, and whiskers (data within 1.5 times the interquartile range) are computed
- The boxplot is manually drawn, with each group represented as a separate box
- A color gradient is used to visually distinguish between the groups
- Additionally, individual data points are plotted using a scatter plot as black dots

4 Detailed calculation of Young's modulus

Young's modulus, or E-Modulus, is a fundamental material property that quantifies the stiffness of a material. It is defined as the slope of the initial linear (elastic) portion of the stress-strain curve.

4.1 Initialization of parameters

- A window size of data points is used for local regression
- The total number of strain data points is counted
- An array slopes is initialized to store the computed local slopes

```
window_size = 20;  % Define the number of points used for local regression
num_points = length(strainCut);  % Total number of available strain points
slopes = zeros(num_points - window_size, 1); % Initialize an array to store local slopes
```

4.2 Computing local slopes using a moving window

The for-loop iterates over all data points, but it stops window-size points before the end (to prevent out-of-bounds errors). In each iteration:

- A subset (window) of strain and stress values is selected
- A linear regression (polyfit) is performed on this subset
- The slope of the regression line (first coefficient of p) is stored in slopes(j)

```
for j = 1:(num_points - window_size)
    % Select a small window of strain and stress values
    strain_window = strainCut(j:j+window_size);
    stress_window = stressCut(j:j+window_size);

    % Perform linear regression to fit a straight line
    p = polyfit(strain_window, stress_window, 1);

    % Store the computed slope (Young's modulus candidate)
    slopes(j) = p(1);
end
```

Since Young's modulus corresponds to the linear elastic region, the window-based approach helps identify the most stable slope.

4.3 Identifying the most stable Young's modulus value

The maximum value in the slopes array is determined and the corresponding index is stored in max_idx.

```
[max_slope, max_idx] = max(slopes);  % Find the maximum slope and its index
```

4.4 Smoothing the result by averaging over a stable range

Instead of using only the maximum slope, we take an average over a small range (±5 points around max_idx) to smooth out fluctuations. This ensures the computed Young's modulus is stable and not affected by small local variations.

```
stable_range = max(max_idx-5,1):min(max_idx+5,length(slopes)); % Define stable range
Emodul(i) = mean(slopes(stable_range)); % Compute the mean over stable range
```

4.5 Final computed Young's modulus

The final E-Modulus value is stored in:

```
Emodul(i) = mean(slopes(stable_range));
```

This value represents the stiffness of the material based on the linear elastic portion of the stress-strain curve.

4.6 Advantages of this Method

- Robust against noise → Instead of using a single data point, multiple points are considered
- Automatic selection of the best elastic region → Finds the most stable, maximum slope
- Smoothing prevents errors → Small fluctuations are removed via averaging

This approach ensures an accurate and reliable computation of Young's modulus from the experimental stress-strain data.

5 Results

After running the entire program, several results are calculated, stored, and visualized. These results include:

- Computed Values Saved to the Excel File
  - These values provide insights into the mechanical properties of the tested material
- Stress-Strain Curves
  - Each file is processed individually and displayed in a separate graph
  - If multiple groups exist (based on ticker), they are plotted in different figures
  - A dropdown menu allows enabling or disabling specific curves
- Boxplot of Young's Modulus
  - a boxplot of Young's modulus is generated for the different groups (ticker)
  - Each group is displayed in a unique color
  - The boxplot helps analyze variations in material stiffness across different samples
  - It enables the comparison of different material specimens or testing conditions

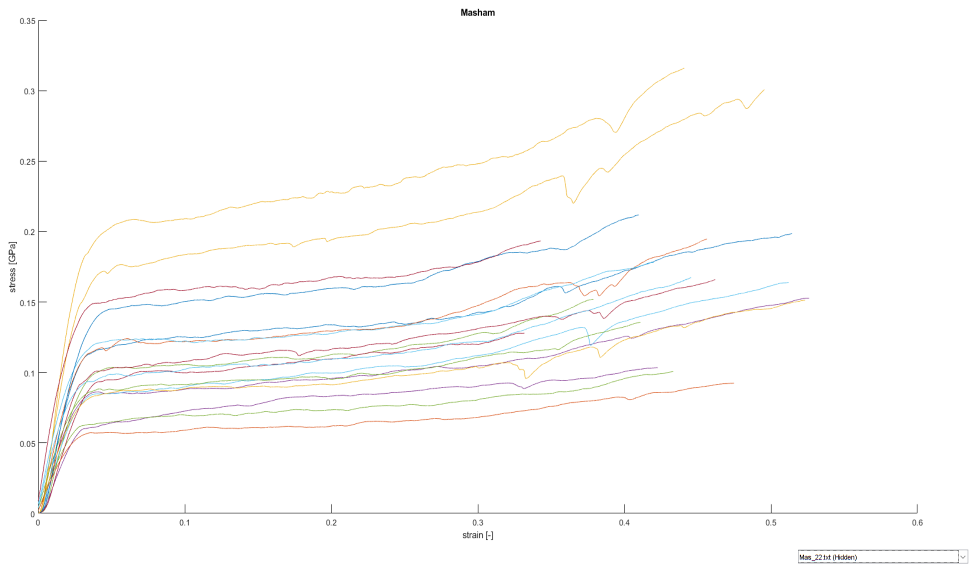

Figure 5: Stress-Strain Curve

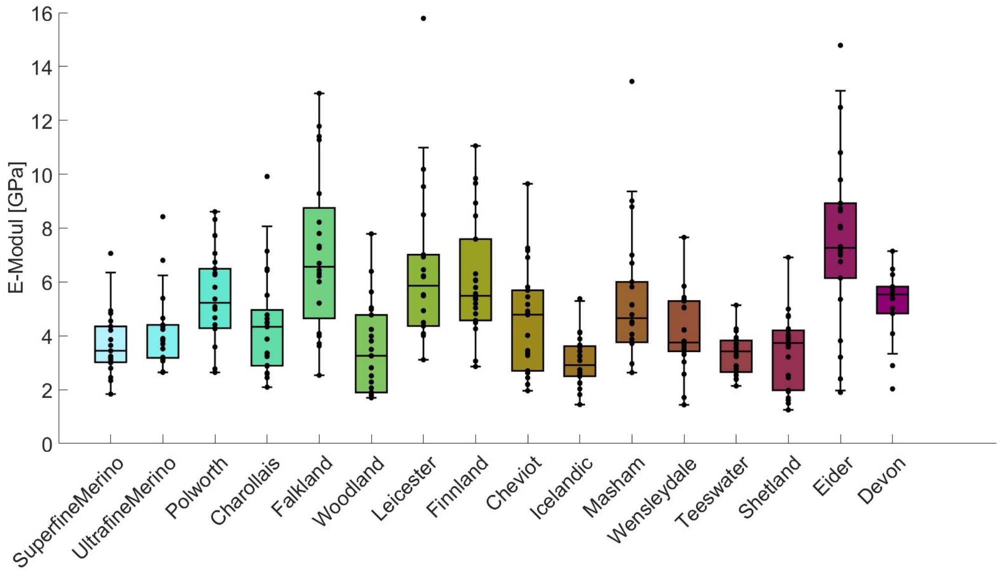

Figure 6: Boxplot of Young's modulus generated for different types