



Aalto University  
School of Engineering

# COE-C2004 - Materials Science and Engineering

## Exercise 3

*Prof. Junhe Lian*

*Wenqi Liu, Jiaojiao Wu (Teaching assistant)*

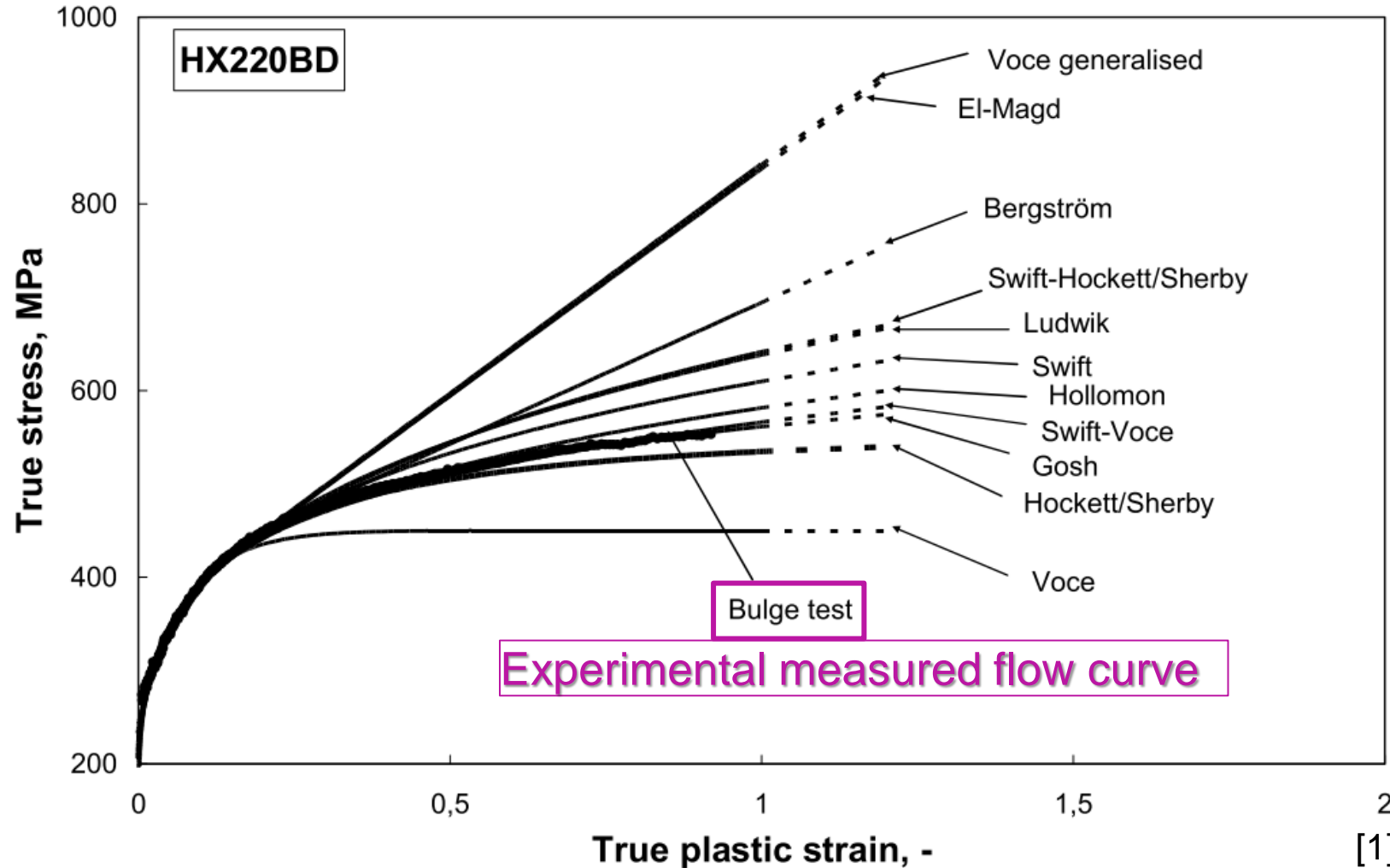
# Outline

- Software introduction
  - Materials hardening law
  - ABAQUS

# Materials hardening law - Flow curve derivation



# Materials hardening law



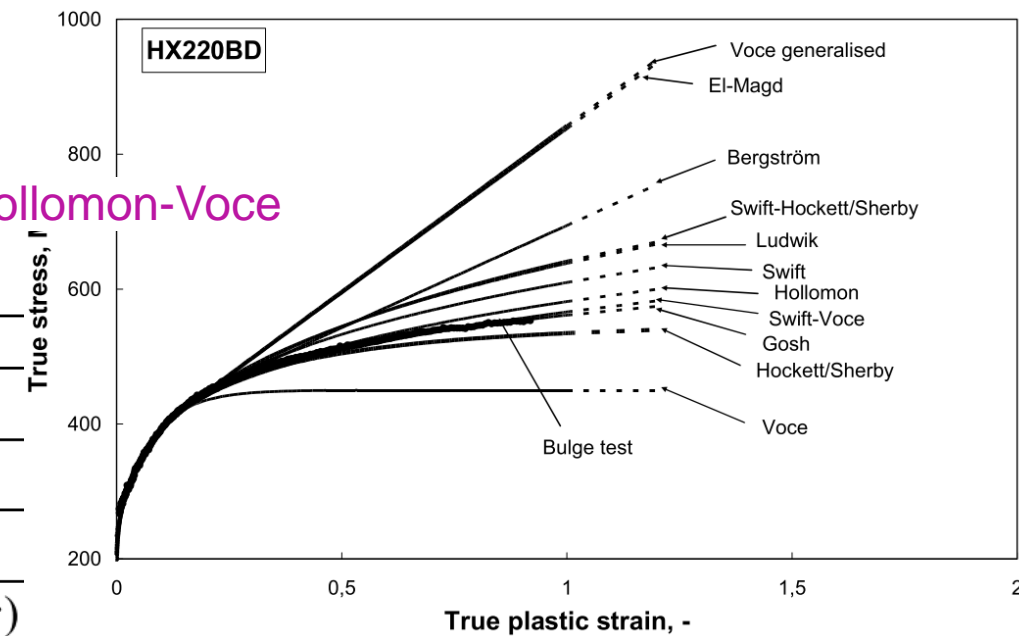
- In numerical simulation, a smooth and extended flow curve is necessary.
- waved exp. data
- in Abaqus, the strain value shall be always in increasing order
- measured true strain limitation  $\sim <1$
- local true strain could  $>2\sim3$  during deformation

➡ Flow curve derivation based on mathematic functions, i.e. hardening laws.

# Materials hardening law

- Individual or combined format, e.g. combined Swift-Voce or Hollomon-Voce

Model	equation
Hollomon [25]	$\sigma(\varepsilon) = C_1 \cdot \varepsilon^{C_2}$
Ludwik [26]	$\sigma(\varepsilon) = C_1 + C_2 \cdot \varepsilon^{C_3}$
Swift [27]	$\sigma(\varepsilon) = C_1 \cdot (C_2 + \varepsilon)^{C_3}$
Voce [28]	$\sigma(\varepsilon) = C_1 + (C_2 - C_1) \cdot \exp(-C_3 \cdot \varepsilon)$
Hockett/Sherby [29]	$\sigma(\varepsilon) = C_2 - (C_2 - C_1) \cdot \exp(-C_3 \cdot \varepsilon^{C_4})$
Gosh [30]	$\sigma(\varepsilon) = C_1 + C_2 \cdot (C_3 + \varepsilon)^{C_4}$
Swift-Voce [31]	$\sigma(\varepsilon) = C_1 \cdot \sigma_{Swift}(\varepsilon) + (1 - C_2) \cdot \sigma_{Voce}(\varepsilon)$
Swift-Hockett/Sherby [32]	$\sigma(\varepsilon) = C_1 \cdot (C_2 + \varepsilon)^{C_3} + C_4 \cdot \exp(-C_5 \cdot \varepsilon^{C_6})$
El-Magd [33]	$\sigma(\varepsilon) = C_1 + C_2 \cdot \varepsilon + C_3 \cdot [1 - \exp(-C_4 \cdot \varepsilon)]$
Voce generalised [34]	$\sigma(\varepsilon) = C_1 + (C_2 + C_3 \cdot \varepsilon) \cdot [1 - \exp(-C_4 \cdot \varepsilon)]$
Bergström [35], [36]	$\sigma(\varepsilon) = C_1 + C_2 \cdot \left( C_3 \cdot (C_4 + \varepsilon) + \{1 - \exp[-C_5 \cdot (C_4 + \varepsilon)]\}^{C_6} \right)$



[1]

# Flow curve derivation - example by Matlab

The image shows the MATLAB R2020b interface with the following components:

- Top Bar:** HOME, PLOTS, APPS, VARIABLE, VIEW.
- APPs Tab:** Design App, Get More Apps, Install App, Package App, Curve Fitting (highlighted with a red box).
- Workspace:** Lists variables: flowcurve (1790x2 double), strain (1790x1 double), stress (1790x1 double).
- Current Folder:** Shows files like MSE1\_CrystalOri.m, importEBSDexample.m, ebsd2.ctf, E3flowcurve.xlsx, A3Task4Data.xlsx, A3Task4Data.SFIT, A1EBSDdata.ctf, ~SE3flowcurve.xlsx, and temp.
- Variables - flowcurve:** A table showing the data for the flowcurve variable.
- Command Window:** Contains the code: 

```
>> stress=flowcurve(:,2);  
>> strain=flowcurve(:,1);  
>>
```

**1. Import your experimental flow curve**

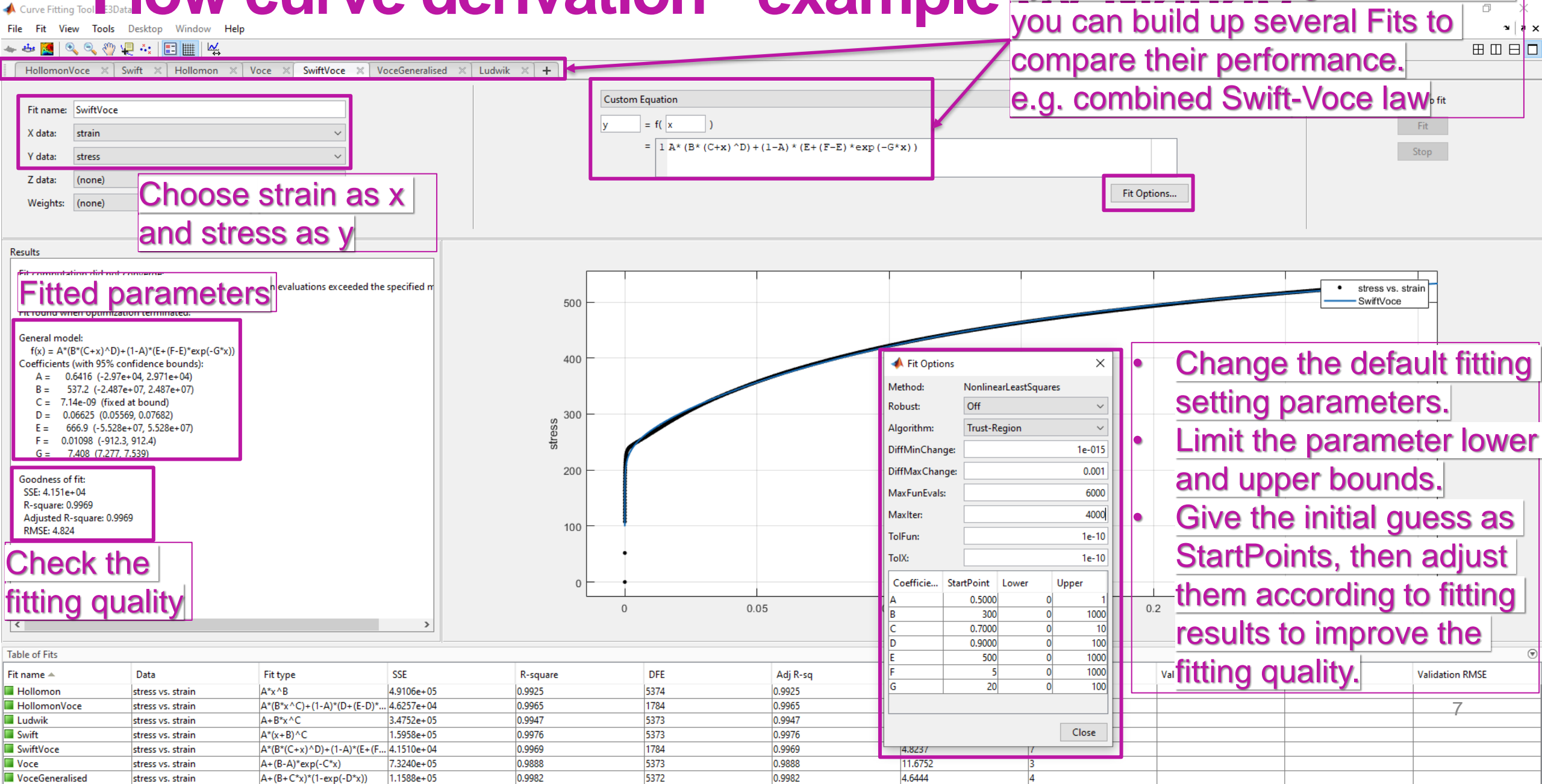
**2. Assign to stress and strain variables**

**3. Use 'Curve Fitting' or based on your own codes.**

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0										
2	5.1403e-07	51.9671										
3	6.3316e-06	108.0024										
4	7.5465e-06	113.4588										
5	4.3010e-06	118.9831										
6	8.9208e-06	124.4956										
7	2.5615e-06	130.0939										
8	1.3242e-06	135.6718										
9	1.6147e-06	141.3367										
10	0	146.9385										
11	0	152.5192										
12	3.1754e-07	157.9701										
13	5.2208e-06	163.5534										
14	1.6987e-05	169.0421										
15	2.9577e-05	174.4601										
16	3.9478e-05	179.7724										
17	5.0436e-05	185.0092										
18	7.1721e-05	190.1524										
19	8.2830e-05	195.1331										
20	1.1071e-04	199.9825										
21	1.4532e-04	204.5226										
22	1.8707e-04	208.9155										
23	2.2749e-04	212.9927										
24	2.7164e-04	216.7834										
25	3.2247e-04	220.0772										
26	3.8099e-04	223.0254										
27	4.4305e-04	225.6321										
28	5.1067e-04	227.8175										
29	5.8048e-04	229.6098										
30	6.5128e-04	231.2059										

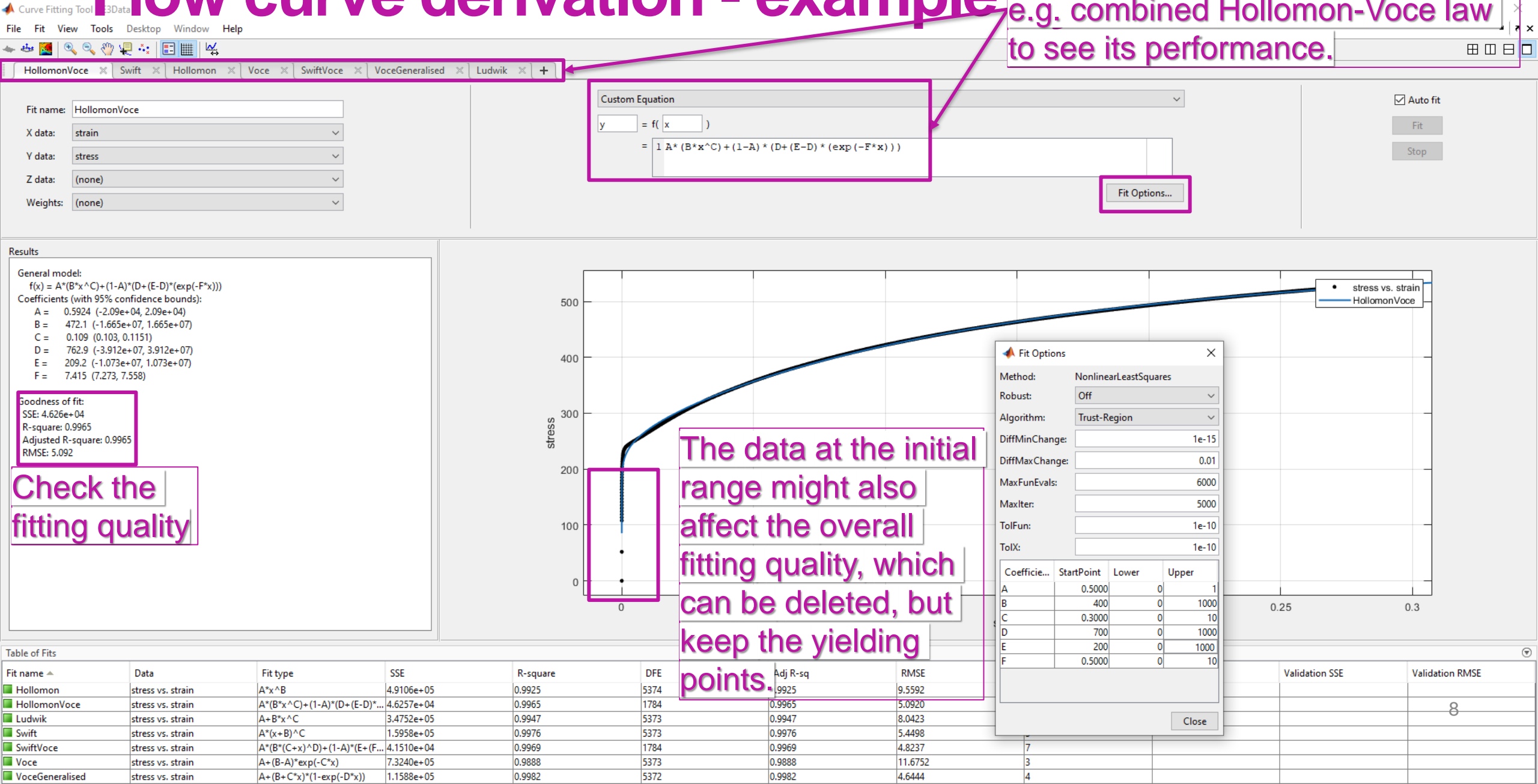
# Flow curve derivation - example

Type in the hardening law function, you can build up several Fits to compare their performance, e.g. combined Swift-Voce law



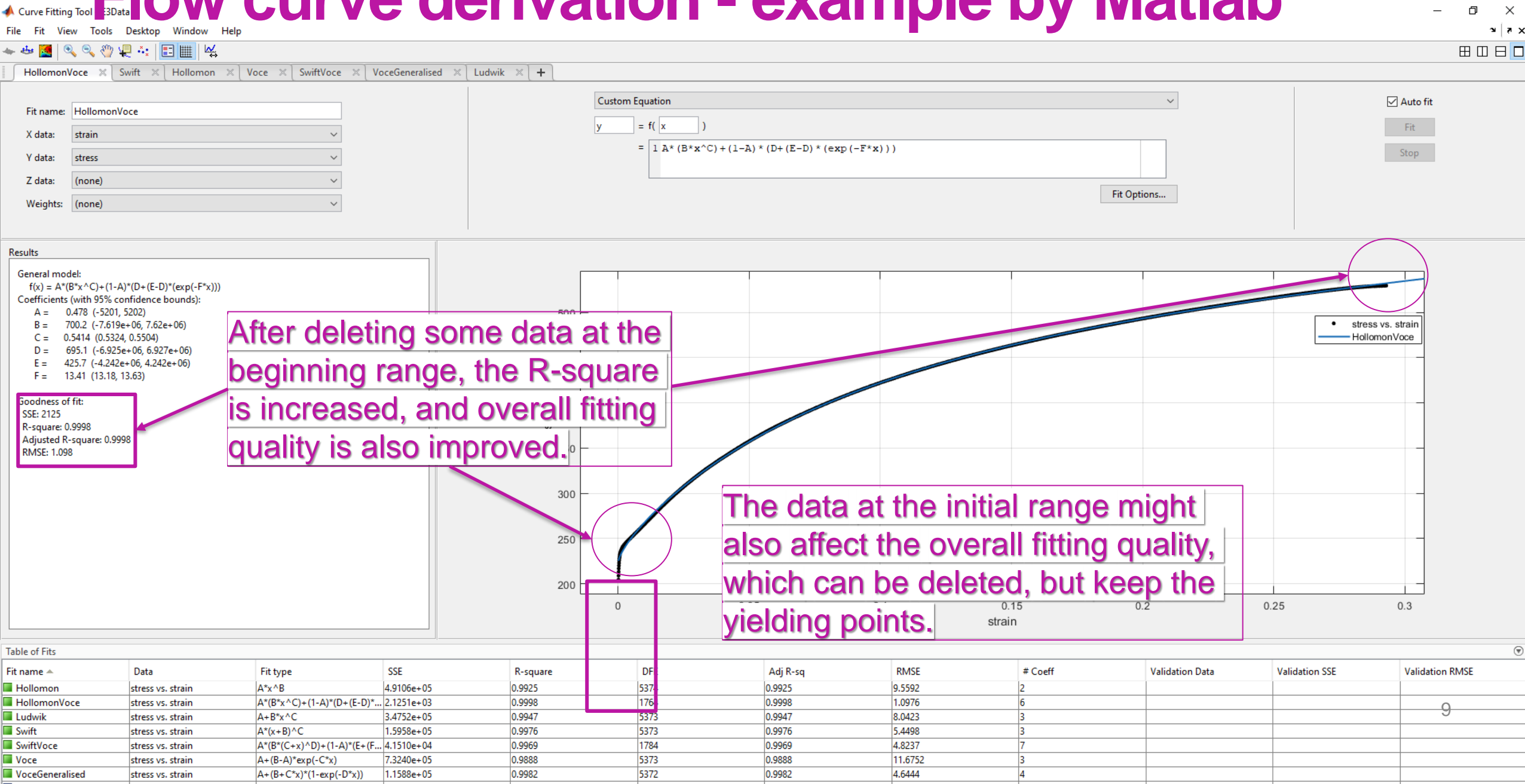
# Flow curve derivation - example

Change to another hardening law, e.g. combined Hollomon-Voce law to see its performance.





# Flow curve derivation - example by Matlab



# Flow curve derivation - example by Matlab

MATLAB R2020b - academic use

HOME PLOTS APPS VARIABLE VIEW

Workspace

Name	Value
ans	0.5000
FitFlowCurve	300x2 double
flowcurve	1790x2 double
strain	1770x1 double
stress	1770x1 double
x	1x300 double
y	1x300 double

Current Folder

- MSEe3.m
- MSE1\_CrystalOri.m
- importEBSDexample.m
- ebsd2.ctf
- E3flowcurve.xlsx
- E3flowcurve.txt
- E3Data.SFIT
- A3Task4Data.xlsx
- A3Task4Data.SFIT
- A1EBSDdata.ctf
- ~E3flowcurve.xlsx
- temp

Variables - FitFlowCurve

	1	2	3	4	5	6	7	8	9	10	11	12
1	222.2154	0										
2	224.6897	1.0000e-04										
3	225.9189	2.0000e-04										
4	226.9235	3.0000e-04										
5	227.8095	4.0000e-04										
6	228.6187	5.0000e-04										
7	229.3727	6.0000e-04										
8	230.0845	7.0000e-04										
9	230.7626	8.0000e-04										
10	231.4131	9.0000e-04										
11	232.0401	1.0000e-03										
12	237.5094	0.0020										
13	242.1738	0.0030										
14	246.4023	0.0040										
15	250.3405	0.0050										
16	254.0639	0.0060										
17	257.6181	0.0070										
18	261.0330	0.0080										
19	264.3297	0.0090										
20	267.5236	0.0100										
21	270.6264	0.0110										
22	273.6476	0.0120										
23	276.5944	0.0130										
24	279.4731	0.0140										

Based on your chosen hardening law and fitted parameters, generate the fitted flow curve for further simulation.

Assign the strain values with the proper intervals and extend to true strain of 2, e.g.

True strain range	Strain interval
0-0.001	0.0001
0.002-0.1	0.001
0.11-2	0.01

Command Window

```
>> % Assign to stress and strain variables
strain=flowcurve(:,1);
stress=flowcurve(:,2);
>> % If necessary, delete some data at the beginning to
>> strain(1:20,:)=[];
stress(1:20,:)=[];
>> %% After fitting
% Define the strain values to 2 based on the proper intervals.
>> x=[0:0.0001:0.001,0.002:0.001:0.1,0.11:0.01:2];
>> y=0.478*(700.2.*x.^0.5414)+(1-0.478)*(695.1+(425.7-695.1)*exp(-13.41.*x));
>> FitFlowCurve=[y',x'];
```

# Flow curve derivation - example by Matlab

- Summary

```
%% Import the exp. data
% Assign to stress and strain variables
strain=flowcurve(:,1);
stress=flowcurve(:,2);
```

```
%% Fit the exp. flow curve by 'Curve Fitting' or your own codes.
```

```
%% To improve the fitting quality, you can delete some data at the beginning range.
```

```
strain(1:20,:)=[];
```

```
stress(1:20,:)=[];
```

```
%% After fitting
```

```
% Define the strain values to 2 based on the proper intervals.
```

```
x=[0:0.0001:0.001,0.002:0.001:0.1,0.11:0.01:2];
```

```
% Calculate the stress according to the fitted hardening law.
```

```
% e.g. the combined Hollomon-Voce law.
```

```
y=0.478*(700.2.*x.^0.5414)+(1-0.478)*(695.1+(425.7-695.1)*exp(-13.41.*x));
```

```
% Obtain your fitted flow curve for Abaqus simulation.
```

```
FitFlowCurve=[y',x'];
```

Tip: The flow curve fitting and extension can also be achieved by Origin, Excel, Python, etc.

