Homework 1

BMEGEMMNWEP, Elasticity and Plasticity

Your data can be downloaded from the following link:

`http://www.mm.bme.hu/~kossa/NEPTUN-HW1.pdf`

Where the string `NEPTUN` has to be replaced with the Student's NEPTUN code. The link above is case-sensitive! Use uppercase letters for the file-name!

Problem description:

Two straight circular columns are mounted between rigid bodies as shown in the Figure. The lengths and the cross section areas of the columns are L_1, L_2, A_1 and A_2. The columns are made of linear elastic - plastic material, where the plastic behaviours (*perfectly plastic* or *linear isotropic hardening* or *linear kinematic hardening*) are specified by the particular HW code. The elastic material constants are the same for both columns: $E = 200$ GPa, $\nu = 0.3$. The initial configuration is a stress-free configuration with zero accumulated plastic strain. The initial yield stresses of the columns are σ_{Y01} and σ_{Y02}. The prescribed displacement of rigid part **A** is u. The loading of the assembly consists of two steps:

Step 1: Displacement $1.5u_{\mathrm{P}}$ is applied on the rigid part **A**.

Step 2: Part **A** is gradually released (force is reduced to 0 gradually).

Tasks:

a) Determine the particular value (u_{E}) of the displacement u, at which plastic deformation first initiates in the structure.

b) Obtain the minimum value (u_{P}) of the displacement u, at which, both colums are in elastic-plastic state.

c) Find the residual displacement (u_{RES}) of part **A** at the end of Step 2.

d) Determine the residual stresses in the columns at the end of Step 2.

e) Plot the stress-strain response $\sigma(\varepsilon)$ for both columns over the whole loading history. Use two separate coordinate systems ($\varepsilon_1 - \sigma_1$ and $\varepsilon_2 - \sigma_2$)!

f) Plot the $F(u)$ function over the whole loading path, where F denotes the applied external force on part **A**.

The picture of the assembly and the particular values of parameters L_1, L_2, A_1, A_2, σ_{Y01}, σ_{Y02}, H_{iso1}, H_{iso2}, $H_{\mathrm{kin1}}$and H_{kin2} are given in the data table below for a given HW code.

MŰEGYETEM 1782

BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
FACULTY OF MECHANICAL ENGINEERING

Elasticity and Plasticity
I. Homework

Gergő Kelle

April 16, 2024

Contents

a) Displacement when plastic deformation first initiate **5**

b) Displacement when both colums are in elastic-plastic state. **5**

c) Determination of residual displacement **6**

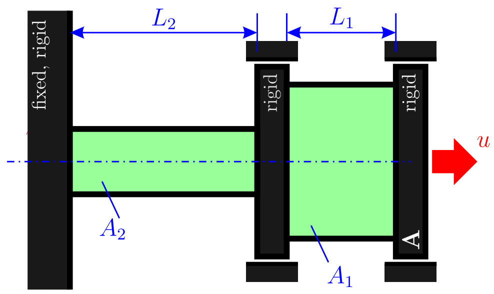
d) Determination of residual stress **8**

e) Plot the stress-strain response $\sigma(\varepsilon)$ for both column **9**

f) Plot the $F(u)$ function over the whole loading path **9**

Initial data

Sample problem



NEPTUN	GBBNUL
L_1 [mm]	900
L_2 [mm]	300
A_1 [mm^2]	500
A_2 [mm^2]	1500
σ_{Y01} [MPa]	250
σ_{Y02} [MPa]	150
H_{iso1} [GPa]	0
H_{kin1} [GPa]	25
H_{iso2} [GPa]	40
H_{kin2} [GPa]	0

Scale representation based on given data

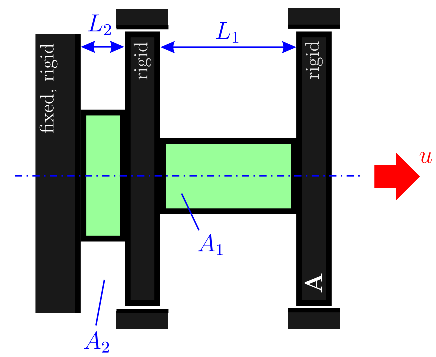


Figure 1: Data for NEPTUN code: GBBNUL and the scale representation of the problem.

Based on the scaled representation of the problem, *column 1* is expected to reach plastic deformation first regarding its size parameters.

Table 1: Given Data

Parameter	Value	Unit
L_1	900×10^{-3}	m
L_2	300×10^{-3}	m
A_1	500×10^{-6}	m^2
A_2	1500×10^{-6}	m^2
σ_{Y01}	250×10^{6}	Pa
σ_{Y02}	150×10^{6}	Pa
H_{iso1}	0	Pa
H_{kin1}	25×10^{9}	Pa
H_{iso2}	40×10^{9}	Pa
H_{kin2}	0	Pa
E	200×10^{9}	Pa
ν	0.3	–

Based on Table 1 we know that the *column 1* has linear kinematic hardening, while *column 2* has linear isotropic hardening characteristics.

a) Displacement when plastic deformation first initiate

Since the columns are connected in series the solution can be determined using a *while* cycle or by separatign it into 2 possible cases. In Case 1 *column 1* suffers plastic deformation faster while in Case 2 *column 2* does. The smaller the overall displacement is the faster it will occur thus it can be decided with a "min" function.

Case 1:

$$u_{01} = \frac{\sigma_{Y01}}{E} L_1 = 1.125 \text{ [mm]} \tag{1}$$

$$u_{E,01} = u_{01} \left(1 + \frac{A_1 \cdot L_2}{A_2 \cdot L_1}\right) = 1.25 \text{ [mm]} \tag{2}$$

Case 2:

$$u_{02} = \frac{\sigma_{Y02}}{E} L_2 = 0.225 \text{ [mm]} \tag{3}$$

$$u_{E,02} = u_{02} \left(1 + \frac{A_2 \cdot L_1}{A_1 \cdot L_2}\right) = 2.25 \text{ [mm]} \tag{4}$$

Decision: Since at this time both columns are deforming only in it's elastic domain we can say that ***column 1*** reach the border of plastic deformation faster. Thus I've selected u_E to be $u_{E,01}$.

Answer for Task a)

$$u_E = \min\left(u_{E,01}, u_{E,02}\right) = u_{E,01} = 1.25 \text{ [mm]} \tag{5}$$

b) Displacement when both colums are in elastic-plastic state.

Since the forces acting on the columns must be equal, the equilibrium equation can be written in terms of displacement as:

$$u_{01} \frac{E \cdot A_1}{L_1} + u_{01x} \frac{A_1}{L_1} \frac{E \cdot H_{\text{kin1}}}{E + H_{\text{kin1}}} = u_{P,02} \frac{A_2 \cdot E}{L_2} \tag{6}$$

Where $u_{P,02} = u_{02}$ and the additional displacement u_{01x} in *column 1* is due to elastic-plastic state transition. The equation can be solved for u_{01x} as:

$$u_{01x} = \frac{L_1}{A_1} \left(\frac{E + H_{\text{kin1}}}{E \cdot H_{\text{kin1}}}\right) \left(\frac{u_{P,02} \cdot A_2 \cdot E}{L_2} - \frac{u_{01} \cdot A_1 \cdot E}{L_1}\right) \tag{7}$$

The total displacement $u_{P,01}$ for *column 1* considering both elastic and elastic-plastic deformation is:

$$u_{P,01} = u_{01} + u_{01x} = 9.225 \text{ [mm]} \tag{8}$$

The minimum value of the displacement u_P at which both columns are in elastic-plastic state is:

$$u_P = u_{P,01} + u_{P,02} = 9.45 \text{ [mm]} \tag{9}$$

To make sure my calculation was right, let's check force equilibrium:

$$F_{P,01} = u_{01x}\,\frac{A_1}{L_1}\,\frac{E\cdot H_{\text{kin1}}}{E+H_{\text{kin1}}} + u_{E,01}\,\frac{E\cdot A_1}{L_1} = 225000.0 \text{ [N]} \tag{10}$$

$$F_{P,02} = u_{02}\,\frac{E\cdot A_2}{L_2} = 225000.0 \text{ [N]} \tag{11}$$

$$F_{P,01} = F_{P,02} \quad \checkmark \tag{12}$$

Answer for Task b)

$$u_P = 9.45 \text{ [mm]} \tag{13}$$

c) Determination of residual displacement

Since $u_{\text{S1}} = 1.5u_P$ we can state that the additional displacement that take place in this step is $\Delta u_P = 0.5u_P$. The additional displacement affect the columns differently based on their elastic-plastic deformation characteristics. The additional displacement is distributed between *column 1* and *column 2* in the following way:

$$\Delta u_{01}\,\frac{A_1}{L_1}\,\frac{E\cdot H_{\text{kin1}}}{E+H_{\text{kin1}}} = \Delta u_{02}\,\frac{A_2}{L_2}\,\frac{E\cdot H_{\text{iso2}}}{E+H_{\text{iso2}}}, \tag{14}$$

$$\frac{\Delta u_{01}}{\Delta u_{02}} = \frac{L_1\cdot A_2}{A_1\cdot L_2}\,\frac{E\cdot H_{\text{iso2}}\,(E+H_{\text{kin1}})}{E\cdot H_{\text{kin1}}\,(E+H_{\text{iso2}})} = 13.5. \tag{15}$$

Since both columns has linear hardening across the elastic-plastic domain, it allows me to determine that the displacement distribution between *column 1* and *column 2* is in a ratio of 13.5:1.

$$u_{\text{S1,01}} = \frac{\Delta u_P}{\frac{\Delta u_{01}}{\Delta u_{02}}+1}\cdot\frac{\Delta u_{01}}{\Delta u_{02}} + u_{01} = 13.6241 \text{ [mm]} \tag{16}$$

$$u_{\text{S1,02}} = \frac{\Delta u_P}{\frac{\Delta u_{01}}{\Delta u_{02}}+1} + u_{02} = 0.5508 \text{ [mm]} \tag{17}$$

$$u_{\text{S1}} = u_{\text{S1,01}} + u_{\text{S1,02}} = 14.175 \text{ [mm]} \qquad (= 1.5\ u_P) \tag{18}$$

The resulting force can be determined as:

$$F_{\text{S1}} = (u_{S1,01} - u_{01})\ \frac{A_1}{L_1}\,\frac{E\ H_{\text{kin1}}}{E+H_{\text{kin1}}} + u_{01}\ E\,\frac{A_1}{L_1} = 279.31\ \text{ [kN]} \tag{19}$$

$$\varepsilon^{\text{e}}_{\text{step1,01}} = \frac{F_{\text{S1}}}{A_1}\,\frac{1}{E} = 0.002793 \tag{20}$$

$$\varepsilon_{\text{step1,01}} = \frac{u_{\text{step1,01}}}{L_1} = 0.015138 \tag{21}$$

$$\varepsilon^{\text{p}}_{\text{step1,01}} = \varepsilon_{\text{step1,01}} - \varepsilon^{\text{e}}_{\text{step1,01}} = 0.012345 \tag{22}$$

$$\varepsilon^{\text{e}}_{\text{step1,02}} = \frac{F_{\text{S1}}}{A_2}\,\frac{1}{E} = 0.000931 \tag{23}$$

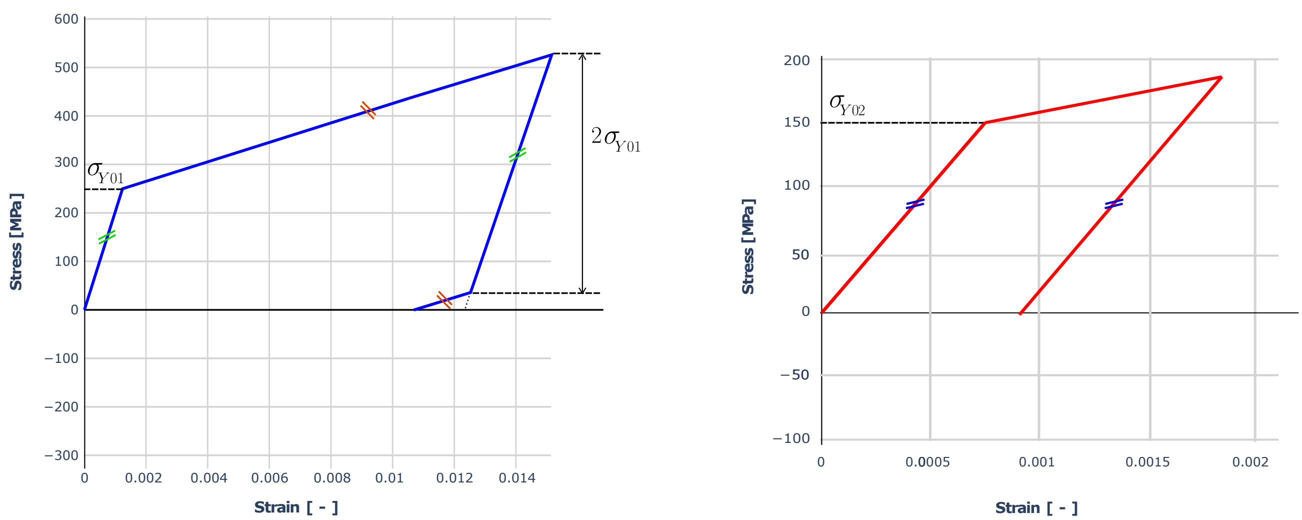
$$\varepsilon_{\text{step1,02}} = \frac{u_{\text{step1,02}}}{L_2} = 0.001836 \tag{24}$$

$$\varepsilon^{\text{p}}_{\text{step1,02}} = \varepsilon_{\text{step1,02}} - \varepsilon^{\text{e}}_{\text{step1,02}} = 0.000905 \tag{25}$$

$$\sigma_{\text{step1,01}} = \sigma_{Y01} + \frac{E \cdot H_{\text{kin1}}}{E + H_{\text{kin1}}}\ \varepsilon^{\text{p}}_{\text{step1,01}} = 524.329\ [\text{MPa}] \tag{26}$$

$$\sigma_{\text{step1,02}} = \sigma_{Y02} + \frac{E \cdot H_{\text{iso2}}}{E + H_{\text{iso2}}}\ \varepsilon^{\text{p}}_{\text{step1,02}} = 180.172\ [\text{MPa}] \tag{27}$$

After step 1, the unloading (step 2) is not that trivial. The residual displacement (u_{RES}) can be determined based on the stress-strain curves if we consider the consequences of the linear hardening during the unloading process.



(a) *Column 1* (b) *Column 2*

Figure 2: Stress-strain response

Since *column 1* is made out of kinematic hardening material and it reaches $\sigma > 2\sigma_{Y01}$ at the end of step 1, during unloading the stress-strain characteristic changes at $\sigma_{\text{step1,01}} - 2\sigma_{Y01} = 24.329$ [MPa] as Figure (2a) shows. We are interested in the exact strain value when it reaches $\sigma = 0$ during unloading which is:

$$\varepsilon_{\text{step2,01}} = 0.011406 \tag{28}$$

$$\varepsilon_{\text{step2,02}} = 0.000935 \tag{29}$$

From the determined strain values the residual displacement can be easily calculated as:

$$u_{\text{RES,01}} = \varepsilon_{\text{step2,01}} \cdot L_1 = 10.265\ \ [\text{mm}] \tag{30}$$

$$u_{\text{RES,02}} = \varepsilon_{\text{step2,02}} \cdot L_2 = 0.281\ \ [\text{mm}] \tag{31}$$

$$u_{\text{RES}} = u_{\text{RES,01}} + u_{\text{RES,02}} = 10.546\ \ [\text{mm}] \tag{32}$$

Answer for Task c)

My initial setup will be $u_{\text{RES}} = 10.546$ [mm] longer after the 2 step loading process. The residiual displacement is distreibuted between the columns in the following way:

$$u_{\text{RES},01} = 10.265 \;\; [\text{mm}] \,,$$

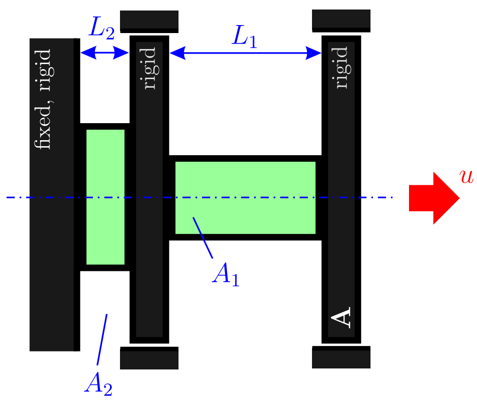
$$u_{\text{RES},02} = 0.281 \;\; [\text{mm}] \,.$$

d) Determination of residual stress

Answer for Task d)

Considering that there's a rigid but movable wall on the right side in my scenario, it's evident that by the end of step 2, both columns have zero residual stress.

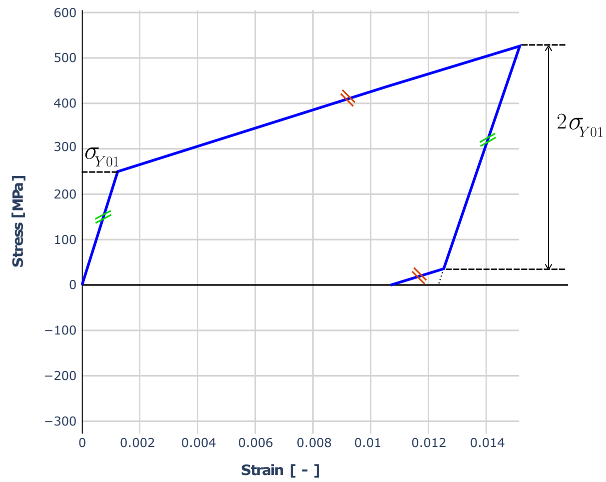
$$\sigma_{RES} = 0 \tag{33}$$



e) Plot the stress-strain response $\sigma(\varepsilon)$ for both column

The plots were already used during Task **c)**.

Answer for Task e)



f) Plot the $F(u)$ function over the whole loading path

In order to have proper documentation I've summarized the calculated forces along the loading process. It is important to note that since the force equlibrium must be satisfied every given step we can freely calculate with *column 1* or *column 2*, because if the calculation is right, it will give us the same result.

$$F_E = u_{\mathrm{E},01}\ E\ \frac{A_1}{L_1} \tag{34}$$

$$F_P = u_{\mathrm{P},02}\ E\ \frac{A_2}{L_2} \tag{35}$$

$$F_{\mathrm{S1}} = \left(u_{S1,01} - u_{\mathrm{E},01}\right)\ \frac{A_1}{L_1}\ \frac{E\ H_{\mathrm{kin1}}}{E + H_{\mathrm{kin1}}} + u_{E,01}\ E\ \frac{A_1}{L_1} \tag{36}$$

$$F_{\mathrm{K}} = \left(\sigma_{\mathrm{step1},01} - 2\sigma_{\mathrm{Y01}}\right) A_1 \tag{37}$$

Table 2: Force and Displacement Values for Specific Points

Point	Force (kN)	Displacement (mm)
E	125	1.25
P	225	9.45
S1	279.31	14.175
K	12.16	11.691
S2	0	10.546

The named points mentioned in Table 2 are presented on Figure 3.

Answer for Task f)

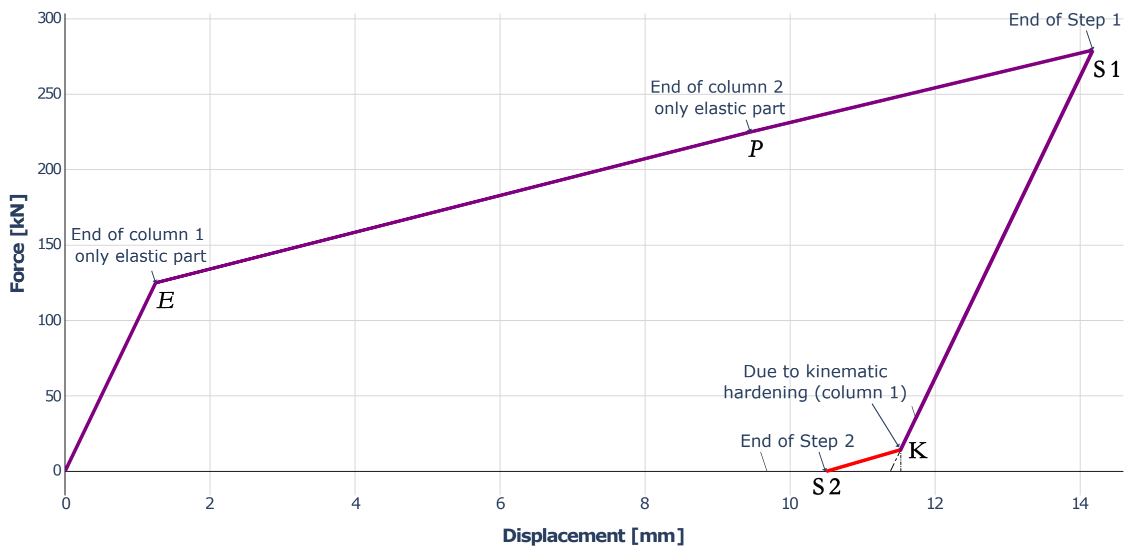


Figure 3: External force-displacement function

It is noteworthy that the gradient from point **E** to **P** differs from the gradient from point **P** to **S1**, primarily because initially only *column 1* experiences elastic-plastic deformation. However, after point **P**, both columns enter the elastic-plastic deformation domain. Another notable observation is that due to *column 1* large deformation and due to it's kinematic hardening, during the unloading process it's stess-strain function suffers another change in characteristics. After that phenomena occurs the force-displacement plot's color is switched to red to indicate the change.

In [1]:
```python
import numpy as np
from IPython.display import display, Markdown, Math
import plotly.graph_objects as go
```

In [2]:
```python
L1 = 900e-3  # m
L2 = 300e-3  # m
A1 = 500e-6  # m^2
A2 = 1500e-6  # m^2
sig_Y01 = 250e6  # Pa
sig_Y02 = 150e6  # Pa
H_iso1 = 0e9  # Pa | isotropic hardening
H_kin1 = 25e9  # Pa | kinematic hardening
H_iso2 = 40e9  # Pa
H_kin2 = 0e9  # Pa
E = 200e9  # Pa
nu = 0.3
```

a)

Determine the particular value (u_E) of the displacement u, at which plastic deformation first initiates in the structure.

In [3]:
```python
u_01 = sig_Y01 / E * L1
u_02 = sig_Y02 / E * L2
print("u_01 =", u_01*1000, " mm")
print("u_02 =", u_02*1000, " mm")
```

```
u_01 = 1.1250000000000002  mm
u_02 = 0.225  mm
```

In [4]:
```python
#Case 1
u_E_1 = u_01 *(1+(A1*L2)/(A2*L1))
display(Markdown(fr'Case 1 total displacement: {u_E_1*1e3:.3f} [mm]'))

#Case 2
u_E_2 = u_02 *(1+(A2*L1)/(A1*L2))
display(Markdown(fr'Case 2 total displacement: {u_E_2*1e3:.3f} [mm]'))

if u_E_1 < u_E_2:
    u_E = u_E_1
    u_P_pre = u_02
    print("Column 1 reach the border of only plastic deformation faster.")
else:
    u_E = u_E_2
    u_P_pre = u_01
    print("Column 2 reach the border of only plastic deformation faster.")

display(Markdown(fr'$u_E$ = {u_E*1e3:.3f} [mm]'))
```

Case 1 total displacement: 1.250 [mm]

Case 2 total displacement: 2.250 [mm]

```
Column 1 reach the border of only plastic deformation faster.
```

u_E = 1.250 [mm]

b)

Obtain the minimum value (u_P) of the displacement u, at which, both colums are in elastic-plastic state.

In [5]:
```python
## Determine displacements
u_01x = L1/A1 * (E + H_kin1)/(E*H_kin1)*(u_02*A2*E/L2 - u_01*A1*E/L1)
u_P_01 = u_01 + u_01x
u_P_02 = u_02
u_P = u_P_01 + u_02
print("u_P = ", u_P*1e3, " mm")
print("u_P_01 = ", u_P_01*1e3, " mm")
print("u_P_02 = ", u_P_02*1e3, "mm")
```

```
u_P =  9.449999999999996  mm
u_P_01 =  9.224999999999998  mm
u_P_02 =  0.225 mm
```

In [6]:
```python
## Force equlibrium
F1 = u_01x * A1 / L1 * (E*H_kin1)/(E+H_kin1) + u_01 * E*A1/L1
F2 = u_P_02 * E*A2/L2
print("F1 =", F1, " N")
print("F2 =", F2, " N")
```

```
F1 = 225000.0  N
F2 = 225000.0  N
```

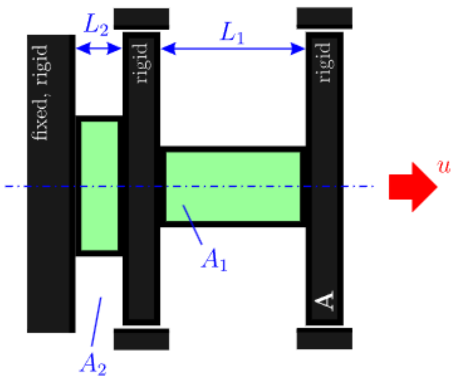
In [7]:
```python
## Determine strain values of point P

# Elastic
eps_P_c1_e = F1 / A1 * 1/E
eps_P_c2_e = F2 / A2 * 1/E

# Total
eps_P_c1 = u_P_01/L1
```

```
eps_P_c2 = u_P_02/L2

# Elastic-Plastic
eps_P_c1_p = eps_P_c1-eps_P_c1_e
eps_P_c2_p = eps_P_c2-eps_P_c2_e # = 0
```

c)

Find the residual displacement (u_{RES}) of part A at the end of Step 2.

Step 1: Displacement 1.5 u_P is applied on the rigid part A.
Step 2: Part A is gradually released (force is reduced to 0 gradually).



STEP 1

In [8]:
```
## Step 1 displacement

# Distribution of 1.5*u_P between the columns
Delta_u_u= L1*A2/(A1*L2)* (E*H_iso2*(E+H_kin1))/(E*H_kin1*(E+H_iso2))
Delta_u_P=0.5*u_P
Delta_u_P_devided= Delta_u_P/(Delta_u_u+1)
Delta_u_01 = Delta_u_P_devided*Delta_u_u
Delta_u_02 = Delta_u_P_devided

print(Delta_u_01+ Delta_u_02, " = ", Delta_u_P)
u_S1_01 = u_P_01 + Delta_u_01 # Végleges érték c1-re step 1 végén
u_S1_02 = u_P_02 + Delta_u_02 # Végleges érték c2-re step 1 végén
print("u_S1_01 = ",u_S1_01*1e3, " mm")
print("u_S1_02 = ",u_S1_02*1e3, " mm")

# It must be 0
print("u_P*1.5 - (u_P_01 + u_P_02) ~ must be 0: ", u_P*1.5 - (u_P_01+ u_P_02))
```

```
0.004724999999999998  =  0.004724999999999998
u_S1_01 =  13.624137931034479  mm
u_S1_02 =  0.5508620689655169  mm
u_P*1.5 - (u_P_01 + u_P_02) ~ must be 0:  0.004724999999999998
```

In [9]:
```
## Force calculation
## 1.5*u_P = u_S1
u_S1 = u_S1_01+u_S1_02
F_step1_01 = (u_S1_01-u_01) * A1 / L1 * (E*H_kin1)/(E+H_kin1) + u_01 * E*A1/L1

print("F_step1_01 =", F_step1_01, " N")
```

```
F_step1_01 = 279310.3448275862  N
```

In [10]:
```
## Determine strain values of point S1

# Elastic
eps_stp1_c1_e = F_step1_01/A1 * 1/E
eps_stp1_c2_e = F_step1_01/A2 * 1/E

# Total
eps_stp1_c1 = u_S1_01/L1
eps_stp1_c2 = u_S1_02/L2

# Elastic-Plastic
eps_stp1_c1_p = eps_stp1_c1 - eps_stp1_c1_e
eps_stp1_c2_p = eps_stp1_c2 - eps_stp1_c2_e
```

In [11]:
```
if H_kin1 != 0:
    sig_stp1_c1 = sig_Y01 + (E*H_kin1)/(E+H_kin1)*eps_stp1_c1_p
    print("Column 1: Linear kinematic hardening characteristic")
elif H_iso1 != 0:
    sig_stp1_c1 = sig_Y01 + (E*H_iso1)/(E+H_iso1)*eps_stp1_c1_p
    print("Column 1: Linear isotropic hardening characteristic")
else:
    sig_stp1_c1 = sig_Y01
    print("Column 1: Perfectly plastic characteristic")
```

```python
if H_kin2 != 0:
    sig_stp1_c2 = sig_Y02 + (E*H_kin2)/(E+H_kin2)*eps_stp1_c2_p
    print("Column 2: Linear kinematic hardening characteristic")
elif H_iso2 != 0:
    sig_stp1_c2 = sig_Y02 + (E*H_iso2)/(E+H_iso2)*eps_stp1_c2_p
    print("Column 2: Linear isotropic hardening characteristic")
else:
    sig_stp1_c2 = sig_Y02
    print("Column 2: Perfectly plastic characteristic")

print("Sigma_step1_c1 = ", sig_stp1_c1/1e6, " MPa")
print("Sigma_step1_c2 = ", sig_stp1_c2/1e6, " MPa")
```

```
Column 1: Linear kinematic hardening characteristic
Column 2: Linear isotropic hardening characteristic
Sigma_step1_c1 =  524.3295019157086  MPa
Sigma_step1_c2 =  180.17241379310343  MPa
```

STEP 2

In [12]:

```python
eps_RES_1 = eps_stp1_c1 - sig_stp1_c1/E
u_RES_1 = eps_RES_1*L1
eps_RES_2 = eps_stp1_c2 - sig_stp1_c2/E
u_RES_2 = eps_RES_2*L2
u_RES = u_RES_1 + u_RES_2
u_RES
```

Out[12]:

```
0.011545258620689652
```

Summarize forces

In [13]:

```python
# E
F_E = u_01*E*A1/L1
u_E_02 = F_E / (E*A2/L2)
print("F_E =", F_E, " N")

# P
F_P_1 = u_01x * A1 / L1 * (E*H_kin1)/(E+H_kin1) + u_01 * E*A1/L1
F_P_2 = u_02 * E*A2/L2
F_P = F_P_1
print("F_P =", F_P, " N")

# S1
F_S1 = F_step1_01
print("F_S1 =", F_S1, " N")

# S2
F_S2 = 0
print("F_S2 =", F_S2, " N")
```

```
F_E = 125000.00000000001  N
F_P = 225000.0  N
F_S1 = 279310.3448275862  N
F_S2 = 0  N
```

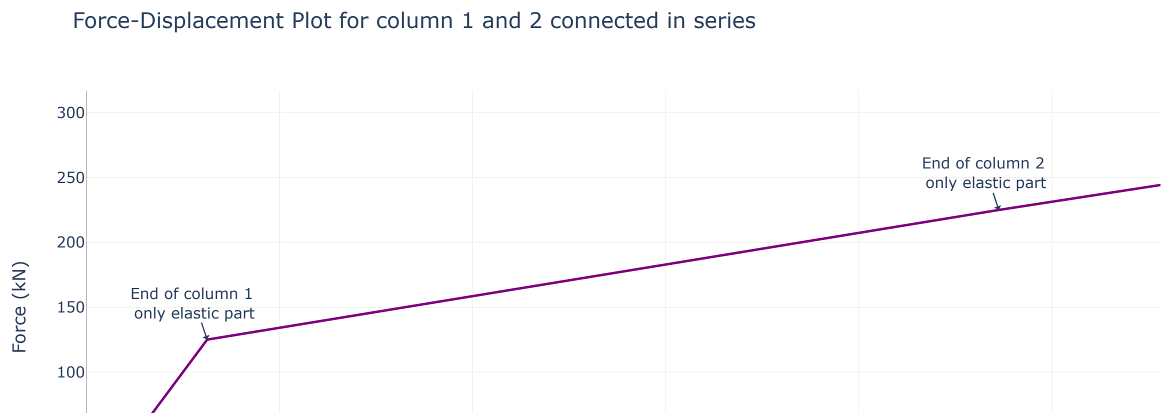
In [14]:

```python
## Summary of forces and displacements


print("-------- 0 to E --------")
print("F_E_NEW =", F_E, " N")
print("u_E =", u_E*1e3, " mm")
print("u_E_01 =", u_01*1e3, " mm")
print("u_E_02 =", u_E_02*1e3, " mm")
print(" ")
print("-------- E to P --------")
print("F_P_NEW =", F_P, " N")
print("u_P =", u_P*1e3, " mm")
print("u_P_01 =", u_P_01*1e3, " mm")
print("u_P_02 =", u_P_02*1e3, " mm")
print(" ")
print("------ P to Step1 ------")
print("F_step1_01 =", F_S1, " N")
print("u_step1 =", u_S1*1e3, " mm")
print("u_step1_01 =", u_S1_01*1e3, " mm")
print("u_step1_02 =", u_S1_02*1e3, " mm")
print(" ")
print("---- Step1 to Step2 ----")
print("F_step2 =", F_S2, " N")
print("u_RES =", u_RES*1e3, " mm")
print("u_RES_1 =", u_RES_1*1e3, " mm")
print("u_RES_2 =", u_RES_2*1e3, " mm")
```

```
-------- 0 to E --------
F_E_NEW = 125000.00000000001  N
u_E = 1.2500000000000002  mm
u_E_01 = 1.1250000000000002  mm
u_E_02 = 0.125  mm

-------- E to P --------
F_P_NEW = 225000.0  N
u_P = 9.449999999999996  mm
u_P_01 = 9.224999999999998  mm
u_P_02 = 0.225  mm

------ P to Step1 ------
F_step1_01 = 279310.3448275862  N
u_step1 = 14.174999999999995  mm
u_step1_01 = 13.624137931034479  mm
u_step1_02 = 0.5508620689655169  mm

---- Step1 to Step2 ----
F_step2 = 0  N
u_RES = 11.545258620689651  mm
u_RES_1 = 11.26465517241379  mm
u_RES_2 = 0.2806034482758618  mm
```

In [15]:

```python
F_E_plot = F_E / 1e3
F_P_plot = F_P / 1e3
F_stp1_plot = F_S1  / 1e3
u_E_plot = u_E * 1e3
u_P_plot = u_P * 1e3
u_stp1_plot = 1.5* u_P * 1e3
u_RES_plot = u_RES * 1e3

# Create figure
fig = go.Figure()

# Add lines for column 1
fig.add_trace(go.Scatter(x=[0, u_E_plot], y=[0, F_E_plot],
                         mode='lines',
                         name='Elastic',
                         line=dict(color='purple')))
fig.add_trace(go.Scatter(x=[u_E_plot, u_P_plot], y=[F_E_plot, F_P_plot],
                         mode='lines',
                         name='Elastic & Plastic',
                         line=dict(color='purple')))
fig.add_trace(go.Scatter(x=[u_P_plot, u_stp1_plot], y=[F_P_plot, F_stp1_plot],
                         mode='lines',
                         name='Step 1',
                         line=dict(color='purple')))
fig.add_trace(go.Scatter(x=[u_stp1_plot, u_RES_plot], y=[F_stp1_plot, F_S2],
                         mode='lines',
                         name='Step 2',
                         line=dict(color='purple')))

fig.add_annotation(x=u_E_plot, y=F_E_plot,
                   text="End of column 1 <br>only elastic part",
                   showarrow=True,
                   arrowhead=1)
fig.add_annotation(x=u_P_plot, y=F_P_plot,
                   text="End of column 2 <br>only elastic part",
                   showarrow=True,
                   arrowhead=1)
fig.add_annotation(x=u_stp1_plot, y=F_stp1_plot,
                   text="End of Step 1",
                   showarrow=True,
                   arrowhead=1)
fig.add_annotation(x=u_RES_plot, y=F_S2,
                   text="End of Step 2",
                   showarrow=True,
                   arrowhead=1)

# Update layout
fig.update_layout(title='Force-Displacement Plot for column 1 and 2 connected in series',
                  xaxis_title='Displacement (mm)',
                  yaxis_title='Force (kN)',
                  xaxis=dict(zeroline=True, zerolinecolor='black', zerolinewidth=1),
                  yaxis=dict(zeroline=True, zerolinecolor='black', zerolinewidth=1),
                  plot_bgcolor='white',
                  xaxis_showgrid=True,
                  yaxis_showgrid=True,
                  xaxis_gridcolor='lightgrey',
                  yaxis_gridcolor='lightgrey')

fig.show()
```

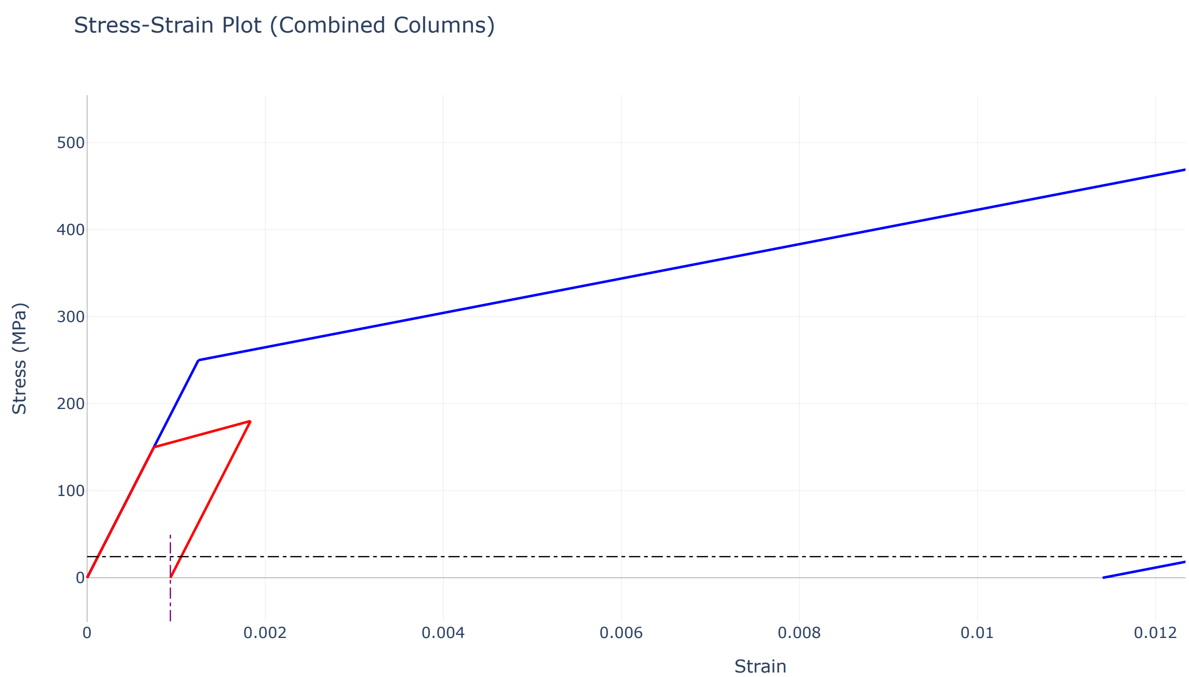


In [16]:
```
## Error: The stress-strain characteristic of column 1 changes during unloading thus the plot is incorrect!
## Find the critical point
```

In [17]:
```
eps_E_c1 = sig_Y01/E
eps_E_c2 = sig_Y02/E # As for column 2 this is present at point P, not at E
```

In [18]:
```
sig_Y01_plot = sig_Y01 / 1e6
sig_stp1_c1_plot = sig_stp1_c1 / 1e6

sig_Y02_plot = sig_Y02 / 1e6
sig_stp1_c2_plot = sig_stp1_c2 / 1e6

eps_unload_y_c1 = eps_stp1_c1 - (sig_Y01/E * 1/ sig_Y01_plot) * 2 * (sig_Y01_plot)
sig_unload_y_c1 = sig_stp1_c1_plot - (2 * sig_Y01 / 1e6)

eps_unload_y_c2 = eps_stp1_c2 - (sig_Y02/E* 1/ sig_Y02_plot) * 2 * (sig_Y02_plot)
sig_unload_y_c2 = sig_stp1_c2_plot - (2 * sig_Y02_plot)

eps_res_c1_v5 = eps_unload_y_c1 - sig_unload_y_c1/((sig_stp1_c1_plot-sig_Y01_plot)/(eps_stp1_c1-eps_E_c1))

# Create figure
fig_combined = go.Figure()

# Add lines for column 1
fig_combined.add_trace(go.Scatter(x=[0, eps_E_c1], y=[0, sig_Y01_plot],
                                  mode='lines',
                                  name='Elastic Column 1',
                                  line=dict(color='blue')))
fig_combined.add_trace(go.Scatter(x=[eps_E_c1, eps_stp1_c1], y=[sig_Y01_plot, sig_stp1_c1_plot],
                                  mode='lines',
                                  name='Step 1 Column 1',
                                  line=dict(color='blue')))
fig_combined.add_trace(go.Scatter(x=[eps_stp1_c1, eps_unload_y_c1], y=[sig_stp1_c1_plot, sig_unload_y_c1],
                                  mode='lines',
                                  name='Step 2 Column 1',
                                  line=dict(color='blue')))
fig_combined.add_trace(go.Scatter(x=[eps_unload_y_c1, eps_res_c1_v5], y=[sig_unload_y_c1, 0],#eps_unload_y_c1 * -1*(sig_stp1_c1_plot-
                                  mode='lines',
                                  name='Step 2 after yield Column 1',
                                  line=dict(color='blue')))

# Add lines for column 2
fig_combined.add_trace(go.Scatter(x=[0, eps_E_c2], y=[0, sig_Y02_plot],
                                  mode='lines',
                                  name='Elastic Column 2',
                                  line=dict(color='red')))
fig_combined.add_trace(go.Scatter(x=[eps_E_c2, eps_stp1_c2], y=[sig_Y02_plot, sig_stp1_c2_plot],
                                  mode='lines',
                                  name='Step 1 Column 2',
                                  line=dict(color='red')))
fig_combined.add_trace(go.Scatter(x=[eps_stp1_c2, eps_RES_2], y=[sig_stp1_c2_plot, 0],
                                  mode='lines',
                                  name='Step 2 Column 2',
                                  line=dict(color='red')))

# Add vertical lines
fig_combined.add_shape(type="line",
                       x0=eps_RES_1, y0=-50,
                       x1=eps_RES_1, y1=50,
                       line=dict(color="green", width=1, dash="dashdot"))
fig_combined.add_shape(type="line",
                       x0=eps_RES_2, y0=-50,
```

```
                        x1=eps_RES_2, y1=50,
                        line=dict(color="purple", width=1, dash="dashdot"))

fig_combined.add_shape(type="line",
                        x0=0, y0=sig_unload_y_c1,
                        x1=0.014, y1=sig_unload_y_c1,
                        line=dict(color="black", width=1, dash="dashdot"))

# Update layout
fig_combined.update_layout(title='Stress-Strain Plot (Combined Columns)',
                            xaxis_title='Strain',
                            yaxis_title='Stress (MPa)',
                            xaxis=dict(zeroline=True, zerolinecolor='black', zerolinewidth=1),
                            yaxis=dict(zeroline=True, zerolinecolor='black', zerolinewidth=1),
                            plot_bgcolor='white',
                            xaxis_showgrid=True,
                            yaxis_showgrid=True,
                            xaxis_gridcolor='lightgrey',
                            yaxis_gridcolor='lightgrey',
                            width=1400,  # Set width to 1600 pixels for 16:9 aspect ratio
                            height=600)  # Set height to 900 pixels for 16:9 aspect ratio)

fig_combined.show()
fig_combined.write_image("stress-strain_plot_EAP1_v2.pdf")
```



In [19]:
```
u_RES_1_v5 = eps_res_c1_v5*L1
u_RES_2_v5 = eps_RES_2*L2
u_RES_v5 = u_RES_1_v5 + u_RES_2_v5

print("---- Compensated Step 2 ----")
print("F_step2 =", F_S2, " N")
print("u_RES =", u_RES_v5*1e3, " mm")
print("u_RES_1 =", u_RES_1_v5*1e3, " mm")
print("u_RES_2 =", u_RES_2_v5*1e3, " mm")
```

```
---- Compensated Step 2 ----
F_step2 = 0  N
u_RES = 10.546228448275865  mm
u_RES_1 = 10.265625000000004  mm
u_RES_2 = 0.2806034482758618  mm
```

In [20]:
```
u_bp_c1 = eps_unload_y_c1*L1
u_bp_c1*1e3

eps_bp_c2 = eps_stp1_c2 - (sig_stp1_c2-sig_unload_y_c1*1e6)/E
u_bp_c2 = eps_bp_c2*L2
u_bp_c2*1e3
```

Out[20]:
```
0.31709770114942476
```

In [21]:
```
F_E_plot = F_E / 1e3
F_P_plot = F_P / 1e3
F_stp1_plot = F_S1  / 1e3
u_E_plot = u_E * 1e3
u_P_plot = u_P * 1e3
u_stp1_plot = 1.5* u_P * 1e3
u_RES_plot = u_RES_v5  * 1e3
```

```
F_bp = sig_unload_y_c1*A1*1e6
F_bp_plot = F_bp / 1e3
u_RES_v5_plot = u_RES_v5 * 1e3

fig = go.Figure()

fig.add_trace(go.Scatter(x=[0, u_E_plot], y=[0, F_E_plot],
                         mode='lines',
                         name='Elastic',
                         line=dict(color='purple')))
fig.add_trace(go.Scatter(x=[u_E_plot, u_P_plot], y=[F_E_plot, F_P_plot],
                         mode='lines',
                         name='Elastic & Plastic',
                         line=dict(color='purple')))
fig.add_trace(go.Scatter(x=[u_P_plot, u_stp1_plot], y=[F_P_plot, F_stp1_plot],
                         mode='lines',
                         name='Step 1',
                         line=dict(color='purple')))
fig.add_trace(go.Scatter(x=[u_stp1_plot, (u_bp_c2+u_bp_c1)*1e3], y=[F_stp1_plot, F_bp_plot],
                         mode='lines',
                         name='Step 2',
                         line=dict(color='purple')))
fig.add_trace(go.Scatter(x=[(u_bp_c2+u_bp_c1)*1e3, u_RES_v5_plot], y=[F_bp_plot, F_S2],
                         mode='lines',
                         name='Step 2',
                         line=dict(color='red')))

fig.add_annotation(x=u_E_plot, y=F_E_plot,
                   text="End of column 1 <br>only elastic part",
                   showarrow=True,
                   arrowhead=1)
fig.add_annotation(x=u_P_plot, y=F_P_plot,
                   text="End of column 2 <br>only elastic part",
                   showarrow=True,
                   arrowhead=1)
fig.add_annotation(x=u_stp1_plot, y=F_stp1_plot,
                   text="End of Step 1",
                   showarrow=True,
                   arrowhead=1)
fig.add_annotation(x=(u_bp_c2+u_bp_c1)*1e3, y=F_bp_plot,
                   text="Kinematic hardening <br> effect of column 1",
                   showarrow=True,
                   arrowhead=1)
fig.add_annotation(x=u_RES_v5_plot, y=F_S2,
                   text="End of Step 2",
                   showarrow=True,
                   arrowhead=1)

fig.add_shape(type="line",
                       x0=u_bp_c1*1e3+u_bp_c2*1e3, y0=0,
                       x1=u_bp_c1*1e3+u_bp_c2*1e3, y1=12,
                       line=dict(color="black", width=1, dash="dashdot"))

# Update layout
fig.update_layout(xaxis_title='Displacement (mm)',
                  yaxis_title='Force (kN)',
                  xaxis=dict(zeroline=True, zerolinecolor='black', zerolinewidth=1),
                  yaxis=dict(zeroline=True, zerolinecolor='black', zerolinewidth=1),
                  plot_bgcolor='white',
                  xaxis_showgrid=True,
                  yaxis_showgrid=True,
                  xaxis_gridcolor='lightgrey',
                  yaxis_gridcolor='lightgrey')

fig.show()
fig.write_image("force_disp_new.pdf")
```