Exercise Session 26-01-2022

→ Intelligent finite elements

↓

Why intelligent ?

⇓

Because some part of the FEM simulation is replaced by NN

Utilization of NNs at the gauss points

⇓

Embedding constitutive material behaviour at gauss points

E.g $\sigma = E\varepsilon$ (Hooke's law)

$\varepsilon \rightarrow NN \rightarrow \sigma$

Utilization of the NN at the element level

⇓

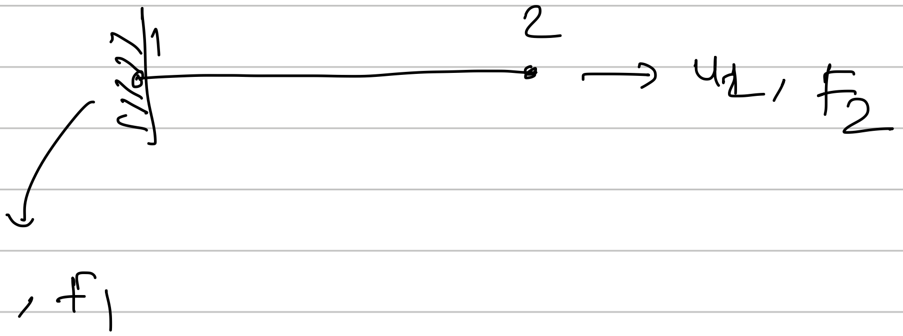
Utilizing function approximation capabilities of the NN at the element level.

E.g $F = kx$

$x \rightarrow NN \rightarrow F$

Our focus for today is on learning the behaviour of $F = kx$ [Element level]

The Element used for academic purposes is 1D frame elements.



Variation of strain within the element?

$\Longrightarrow$ Constant

Thus, number of gauss points used?

$\longrightarrow$ 1 //

Now, this helps us in simplifying the Task at hand

because at the element level if u derive then

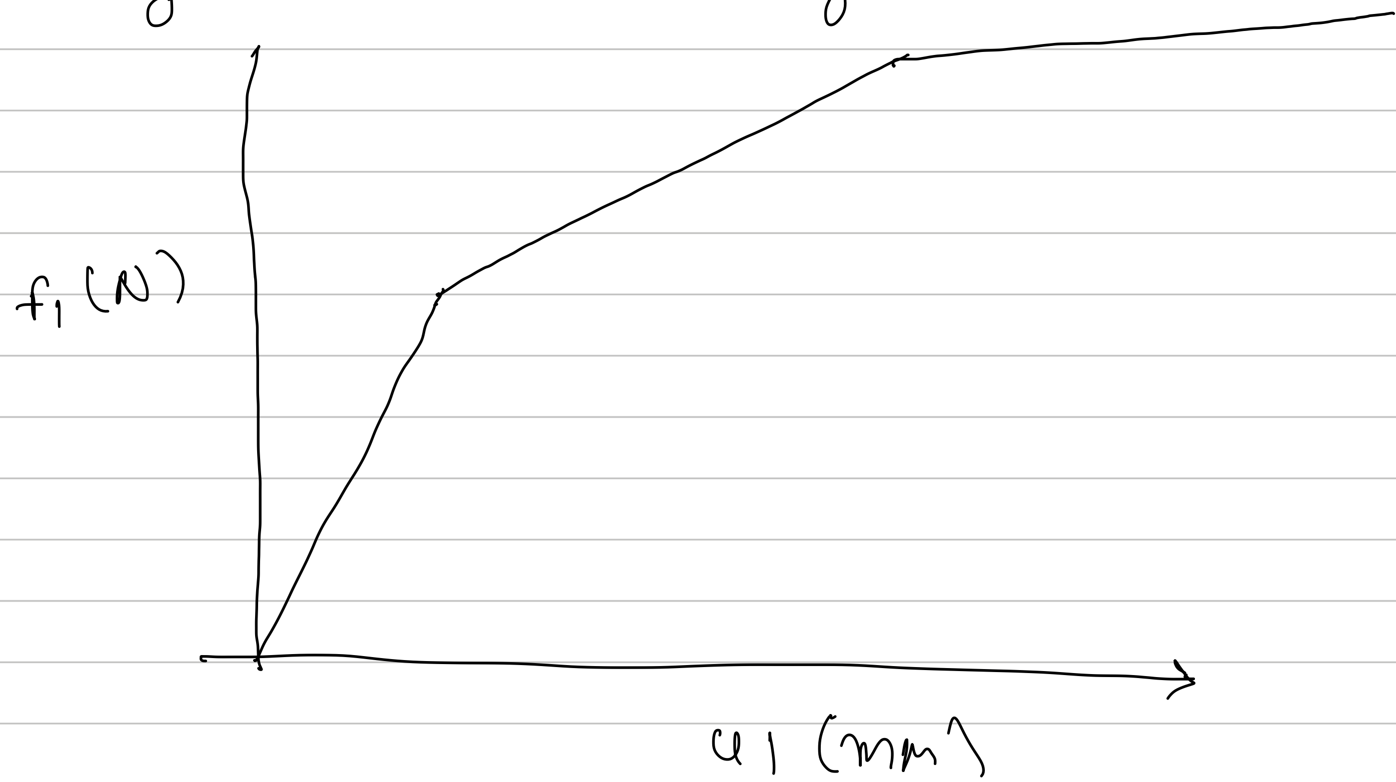
$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Where $k = + \dfrac{EA}{L}$

For the current problem $L = 1000\,mm$

$A = 625\,mm^2$, $E = 50000\,MPa$.

We will be looking at elastoplastic material behaviour. → The operating range looks something like the following.



(Piecewise linear variation) ⟹ further simplifies our problem.

Small question, which activation function is piecewise non-linear?

Now, what?

We need to decide on input and output variables

Since strain is constant

$$\varepsilon = \frac{u_2 - u_1}{L} \quad \text{or} \quad \varepsilon = \frac{\Delta u}{L}$$

What will be the displacements at two nodes ?

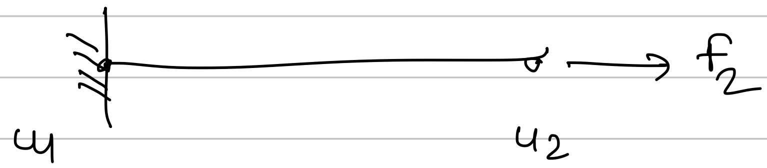
Say if $\varepsilon = 1e^{-3}$, then

$$1e^{-3} = \frac{\Delta u}{L} \quad \therefore \quad \Delta u = 1mm$$

or

$$\Delta u = -1mm$$

Suppose



$$\therefore \quad \varepsilon = \frac{u_2 - u_1}{L} \quad \therefore \quad 1 = u_2$$

Say $f_2 = 1000 N$, then $f_1 = -1000 N$.

So finding displacement at only one node is enough so that means

we choose u_2 as input

and the output as f_2

$$u_2 \longrightarrow \text{NN} \longrightarrow f_2$$

Topology of your choice

FFNN, RBFN or CNNs.

For the exercise we will use FFNN, try with others.

In FEM, we have two important terms

→ Internal force vector → Is the output of the NN

→ Stiffness matrix → How to compute with NN ?

So there are different ways of implementing it:

(I) Numeric Differentiation.
(II) Automatic Differentiation.

(I) Numeric Differentiation.

Q] → What are we diff^{r} → ?

We are differentiating the NN

let $\mathcal{M}_{NN} \Rightarrow$ Our trained model then.

$$\boxed{\frac{\partial f_i^n}{\partial u_m^n} = \frac{\mathcal{M}_{NN} : (u_m^n + \varepsilon_m) - \mathcal{M}_{NN} : (u_m^n)}{\varepsilon_m}}$$

This is how we compute the components of the stiffness matrix.

$$\frac{\partial f_1}{\partial u_1} = \frac{\mathcal{M}_{NN} : (u_1^n + \varepsilon_1) - \mathcal{M}_{NN} : (u_1^n)}{\varepsilon_1}$$

Since $K_{el} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & -\frac{\partial f_1}{\partial u_2} \\ -\frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \end{bmatrix}$

So if we compute for only one element, that is enough.

(II) Automatic Differentiation

→ Back propagate to compute the stiffness

$$\frac{\partial f_i^n}{\partial u_m^n} = \frac{\partial f_s^n}{\partial A_s^n} \cdot \frac{\partial A_s^n}{\partial z_s^n} \sum_{i=1}^{L'} \left(\frac{\partial z_s^n}{\partial A_i^k} \frac{\partial A_i^k}{\partial z_i^k} \cdots \cdots \right.$$

$$\left. \sum_{p=1}^{L} \frac{\partial A_p^1}{\partial z_p^1} \cdot \frac{\partial z_p^1}{\partial u_m^n} \right)$$