Plastic Hinge Formation: Simply-Supported Beam

By: **Nabil Fares**

Date: September 2021

Notebook Setup

Imports

In [2]:

```
import warnings
warnings.filterwarnings('ignore')


from IPython.display import IFrame
from IPython.display import YouTubeVideo

from IPython.display import Video

from IPython.display import HTML

import scipy
from scipy.interpolate import interp1d as interpolate
from scipy import integrate

import numpy as np
from sympy import linsolve, symbols
import sympy

import matplotlib
import matplotlib.pyplot as pyplot
import matplotlib.gridspec as gridspec
from matplotlib import rc as rcForMe
import matplotlib.patches as mpatches
import matplotlib.transforms as transforms
from matplotlib.ticker import (MultipleLocator, AutoMinorLocator, NullFormatt

from copy import copy

import math

from sympy import integrate as sintegrate

from IPython.display import display, Image

from sympy import init_printing


# rcForMe('text', usetex=True)

init_printing()

import physics.physics
from physics.physics import Q

import nfmodules.nfutilities as nfutil
```

Outline

- Formation of Plastic Hinge
- Simply-supported beam: Detailed calculations:
- Using ANSYS: Simply-Supported Beam

Formation of Plastic Hinge

A plastic hinge forms when the moment at a point reaches the plastic moment M_P. When that happens, the moment at the plastic hinge cannot exceed the plastic moment M_P but can maintain that value due to the assumed ductility of the material. If we think of an incremental increase in loading, the plastic hinge does not contribute to resisting the increment in load. Thus, after the formation of a plastic hinge, the analysis for incremental load can assume that the plastic hinge acts like an internal hinge.

So we think of a plastic hinge as equivalent to an internal hinge for any incremental load. If the structure modified by this equivalent internal hinge becomes statically unstable, plastic collapse occurs with a further increase in load. Of course, if we control the displacement instead of the load, then we can avoid this plastic collapse by avoiding an increase in the loading. We will use this idea when studying some structures using FEM software.

In manual calculations, we will assume the following:

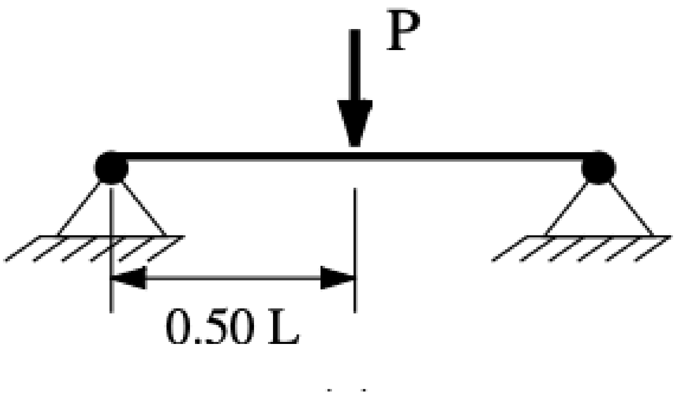
- For the purpose of calculating deflections and bending moment diagrams, points on the beam behave elastically (I.E. M = EI κ) until the moment at the cross-section reaches the plastic moment M_P (Note: the plastic moment M_P and not just the yield moment M_Y)
- When the moment at a point on the beam reaches the plastic moment M_P, a plastic hinge forms and the moment becomes equal to M_P at that point for further load increments. For the purpose of calculating **incremental** bending moment diagrams and **incremental** displacements, we place internal hinges at all plastic hinges and analyze the beam elastically until the formation of the next plastic hinge somewhere else in the beam.
- The beam or frame collapses when enough plastic hinges form so that if we replace internal hinges at the location of plastic hinges, we obtain a statically unstable structure.

The assumptions above are approximations because:

- At the location of the formation of a plastic hinge, a beam will behave elastically, then elasto-plastically with a non-linear moment-curvature relation and then finally as a plastic hinge. We have assumed a jump in behavior from elastic moment-curvature relation to plastic hinge behavior. Note that we have calcuated exact elasto-plastic moment curvature relations and it is possible to solve beam problems with those exact elastic-plastic moment-curvature relations.
- At locations **adjacent** to the plastic hinges, the state of the cross-sections are not elastic but rather elasto-plastic with the non-linear elasto-plastic moment-curvature relation mentioned above.

However, the above assumptions are widely used and give very good results. If we need more accurate results, we use FEM software.

Simply-Supported Beam: Detailed calculations



A simply-supported beam is statically determinate so that the moment diagram does not depend on the moment-curvature relation as long as there is no collapse. The maximum moment is always under the load and has a value of $PL/4$.

Two load levels are distinct:

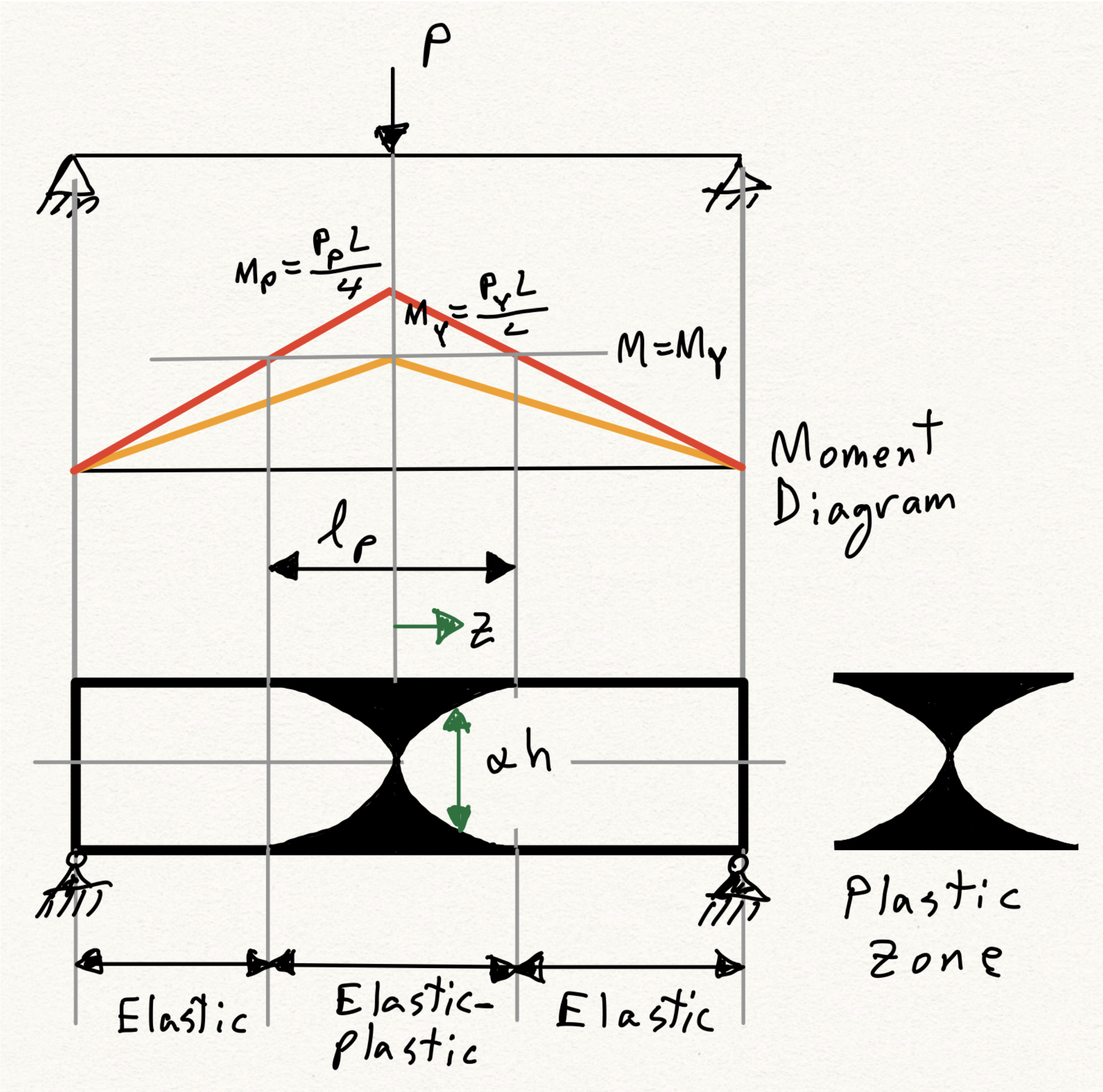
- When the maximum moment equals M_Y. We call that load level $P_Y = 4M_Y/L$.
- When the maximum moment equals M_P. We call that load level $P_P = 4M_P/L$. Beyond that point we have collapse.

Note that the ratio of the load to first form a plastic hinge divided by the load to cause first yield is:

$\frac{P_P}{P_Y} = \frac{4M_P/L}{4M_Y/L} = \frac{M_P}{M_Y} = f$ (f was called the shape factor in a previous lecture)

This result that the ratio of plastic to yield forces equals the section's shape factor is general because the moments are linearly related to the load until the formation of the first plastic hinge.

Those moment diagrams are shown in the figure below along with a sketch of the shape of the plastically deforming zone when the load reaches P_P.



Shape of the plastic zone

To determine the shape of the plastic zone, we need to find the relation between '$\alpha\ h$', the extent of the elastic zone at cross-sections where there is combined elastic and plastic deformations, and z, the location relative to the plastic hinge location. Those are depicted in the figure above.

The plan is as follows:

- Find the relation between z and the bending moment M.
- Find the relation between the bending moment M and '$\alpha\ h$'.
- Cancel M from the above two relations to get a relation between z and '$\alpha\ h$'

From the bending moment diagram, we have:

$$M = M_P(1 - \frac{2z}{L})$$

From our study of moment-curvature relations, we had:

$$\frac{M}{M_Y} = 4\ \frac{1}{\alpha_{shape\ for\ I}}\frac{bh}{A}\left(\frac{\kappa}{\kappa_Y}\int_{-\alpha/2}^{\alpha/2} w\,s^2 ds + \int_{\alpha/2}^{1/2} w\,s\ ds\right) \qquad \frac{\kappa}{\kappa_Y} > 1$$

and

$$\alpha = \frac{\kappa_Y}{\kappa}$$

$$\alpha_{shape\ for\ I} = 4\ \frac{bh}{A} \int_{-1/2}^{1/2} w\ s^2\ ds$$

$$\alpha_{moha} = 4 \frac{bh}{A} \int_{0}^{1/2} w\ s\ ds$$

$$f = \frac{M_P}{M_Y} = \frac{\alpha_{moha}}{\alpha_{shape\ for\ I}}$$

Therefore:

$$\frac{M}{M_P} = \frac{\int_{-\alpha/2}^{\alpha/2} w\, s^2 ds + \alpha\ \int_{\alpha/2}^{1/2} w\ s\ ds}{\alpha\ \int_{0}^{1/2} w\ s\ ds}$$

Now cancelling M/M_P from the relation involving z and that involving α, we get:

$$\frac{z}{L} = \frac{1}{2} - \frac{1}{2} \frac{\int_{-\alpha/2}^{\alpha/2} w\, s^2 ds + \alpha\ \int_{\alpha/2}^{1/2} w\ s\ ds}{\alpha\ \int_{0}^{1/2} w\ s\ ds}$$

Rectangular Cross-Section

For a rectangular cross section, $w = 1$ and we get:

$$\frac{z}{L} = \frac{1}{2} - \frac{1}{2} \frac{\alpha^3/12 + \alpha(1/8 - \alpha^2/8)}{\alpha/8} \quad \Rightarrow$$

$$\frac{z}{L} = \frac{1}{6}\alpha^2$$

Note that this is the equation of a parabola. From this result, we can also determine ℓ_p by setting $\alpha = 1$ to get the extent of z on one side. We then multiply by 2 to get:

$$\ell_p = 2 \times \frac{1}{6} \times 1^2 \times L = \frac{1}{3} \times L$$

Other Cross-Sections

We will only calculate ℓ_p which corresponds to $2 \times z$ when $\alpha = 1$. The result is the table below:

Name	Shape	ℓ_p
Ideal I-beam		0.0635
Hourglass parabola sharp		0.2
Hourglass Triangles		0.25
Hourglass parabola fat		0.2857
Rectangle		0.3333
Ellipse		0.411
Parabolas diamond fat		0.4667
Triangle diamond		0.5
Parabolas diamond sharp		0.6

We note that the plastic zone size increases as the relative amount of material increases near the centroid.

Using ANSYS: Simply-Supported Beam

Using ANSYS: Simply-Supported Beam - point force

In [ ]:

```
Video("./videos/simply-supported-beam-point-force.mp4", width=1100, embed=Tru
```

Using ANSYS: Simply-Supported Beam - uniform load

In [ ]:

```
Video("./videos/simply-supported-beam-uniform-load.mp4", width=1100, embed=Tr
```

Procedures, Calculations and Work-in-progress Notes

Rectangular cross-section

In [6]:

```
yy = symbols('y', real=True)
ss = symbols('s', real=True)
alpha = symbols('alpha', real=True)
```

In [7]:
```
i1 = sintegrate(ss**2, (ss, -alpha/2, alpha/2))
i2 = alpha * sintegrate(ss, (ss, alpha/2, 1/2))
i3 = alpha * sintegrate(ss, (ss, 0, 1/2))
display(i1)
display(i2)
display(i3)
r = sympy.simplify((1/2) - (1/2)*(i1 + i2)/i3)
display(r)
```

$$\frac{\alpha^3}{12}$$

$$\alpha\left(0.125 - \frac{\alpha^2}{8}\right)$$

$$0.125\alpha$$

$$0.166666666666667\alpha^2$$

General section ℓ_p

$$\frac{z}{L} = \frac{1}{2} - \frac{1}{2}\frac{\int_{-\alpha/2}^{\alpha/2} w\, s^2 ds + \alpha\ \int_{\alpha/2}^{1/2} w\ s\ ds}{\alpha\ \int_0^{1/2} w\ s\ ds}$$

In [8]:
```
def find_lp(what):
    i1 = integrate.quad(lambda s: what(s) * s**2, -1/2, 1/2)[0]
    i2 = integrate.quad(lambda s: what(s) * s, 1/2, 1/2)[0]
    i3 = 1 * integrate.quad(lambda s: what(s) * s, 0, 1/2)[0]
    return round(1 - (i1 + i2)/i3, 4)
```

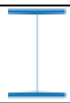
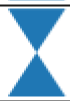
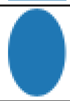
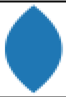
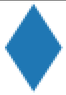
In [9]:
```
find_lp(nfutil.width_ellipse)
```

Out[9]: 0.411

In [10]:
```
overAllSizeInInches = (6, 8)  # width, height
bodyFontSize = 14
def makeSmallFigure(widthFunction, fig, gridPart):
    y1 = np.arange(0,0.51,0.01)
    x1 = list(map(lambda y: 0.5 * widthFunction(y), y1))
    y2 = np.flip(y1)
    x2 = list(map(lambda y: -0.5 * widthFunction(y), y2))
    y3 = -np.flip(y2)
    x3 = list(map(lambda y: -0.5 * widthFunction(y), y3))
    y4 = np.flip(-y1)
    x4 = list(map(lambda y: 0.5 * widthFunction(y), y4))
    innerGrid = gridPart.subgridspec(1, 1, wspace=0, hspace=0)
    ax = fig.add_subplot(innerGrid[0,0])
    ax.set_aspect(1)
    ax.fill(np.concatenate((x1,x2, x3, x4)), np.concatenate((y1,y2, y3, y4)))
    ax.grid(False)
    ax.set_frame_on(False)
    ax.set_axis_off()
    return ax

def makeText(text,fig, gridPart):
    innerGrid = gridPart.subgridspec(1, 1, wspace=0, hspace=0)
    ax = fig.add_subplot(innerGrid[0,0])
    ax.text(0.1,0.45,text, fontsize=bodyFontSize)
```

```
    ax.grid(False)
    ax.set_frame_on(False)
    ax.set_axis_off()
    return ax

figMain = pyplot.figure(dpi=80)
figMain.set_size_inches(overAllSizeInInches)
widths = [5, 1.5, 1.5]
heights = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
gridspec = figMain.add_gridspec(ncols=3, nrows=10, hspace=0, wspace=0, width_
axs = gridspec.subplots()

row = 0
makeText("Name", figMain, gridspec[0,0]);
makeText("Shape", figMain, gridspec[0,1]);
makeText("$\ell_p$", figMain, gridspec[0,2]);

row += 1
makeText("Ideal I-beam", figMain, gridspec[row,0]);
makeSmallFigure(lambda y: 0.65 * nfutil.width_ibeam(y), figMain, gridspec[row
makeText(find_lp(nfutil.width_ibeam), figMain, gridspec[row,2]);

row += 1
makeText("Hourglass parabola sharp", figMain, gridspec[row,0]);
makeSmallFigure(lambda y: 0.65 * nfutil.width_hourglass_parabolas_sharp(y), f
makeText(find_lp(nfutil.width_hourglass_parabolas_sharp), figMain, gridspec[r

row += 1
makeText("Hourglass Triangles", figMain, gridspec[row,0]);
makeSmallFigure(lambda y: 0.65 * nfutil.width_hourglass_triangles(y), figMain
makeText(find_lp(nfutil.width_hourglass_triangles), figMain, gridspec[row,2])

row += 1
makeText("Hourglass parabola fat", figMain, gridspec[row,0]);
makeSmallFigure(lambda y: 0.65 * nfutil.width_hourglass_parabolas_fat(y), fig
makeText(find_lp(nfutil.width_hourglass_parabolas_fat), figMain, gridspec[row

row += 1
makeText("Rectangle", figMain, gridspec[row,0]);
makeSmallFigure(lambda y: 0.65 * nfutil.width_rectangle(y), figMain, gridspec
makeText(find_lp(nfutil.width_rectangle), figMain, gridspec[row,2]);

row += 1
makeText("Ellipse", figMain, gridspec[row,0]);
makeSmallFigure(lambda y: 0.65 * nfutil.width_ellipse(y), figMain, gridspec[r
makeText(find_lp(nfutil.width_ellipse), figMain, gridspec[row,2]);

row += 1
makeText("Parabolas diamond fat", figMain, gridspec[row,0]);
makeSmallFigure(lambda y: 0.65 * nfutil.width_diamond_parabolas_fat(y), figMa
makeText(find_lp(nfutil.width_diamond_parabolas_fat), figMain, gridspec[row,2

row += 1
makeText("Triangle diamond", figMain, gridspec[row,0]);
makeSmallFigure(lambda y: 0.65 * nfutil.width_diamond_triangles(y), figMain,
makeText(find_lp(nfutil.width_diamond_triangles), figMain, gridspec[row,2]);

row += 1
makeText("Parabolas diamond sharp", figMain, gridspec[row,0]);
makeSmallFigure(lambda y: 0.65 * nfutil.width_diamond_parabolas_sharp(y), fig
makeText(find_lp(nfutil.width_diamond_parabolas_sharp), figMain, gridspec[row

for ax in axs.flat:
    ax.set_xticks([])
    ax.set_yticks([])
```

```
pyplot.savefig('plastic-zone-length.png', facecolor='white')
```

Name	Shape	ℓ_p
Ideal I-beam		0.0635
Hourglass parabola sharp		0.2
Hourglass Triangles		0.25
Hourglass parabola fat		0.2857
Rectangle		0.3333
Ellipse		0.411
Parabolas diamond fat		0.4667
Triangle diamond		0.5
Parabolas diamond sharp		0.6

In [ ]: