Home

Welcome to the course CTB3330 Structural Mechanics taught in the third year of Bachelor programme Civil Engineering at Delft University of Technology.

This TeachBook contains parts of the material for the course.

Go to /CTB3330 to view the most recent version of this book, or adapt the year in /CTB3330/2024 to the year when you took the course.

How to use this TeachBook 📘

Contents

This TeachBook combines interactive content on all topics and theory on Work and Energy methods.

Interactive features

This TeachBook includes interactive coding features! Click 🚀 –> **Live Code** on the top right corner of interactive page to start up a python-kernel in your browser! Any interactions you do here are not stored. You can also download those pages as a notebook to apply the content on your own computer.

Spot a mistake?

If you spot any mistakes, you can click on the GitHub icon –> 💡, login with a GitHub account and report your issue. It'll be solved soon!

Personalised book

If you'd like to make this TeachBook more personal by adding (private or public) annotations I can recommend the Hypothesis extension. This is only for your own use, I won't monitor public post on this platform.

Version

This is the `2024-2025`-version of the TeachBook. Updates during this course are communicated on the relevant pages and in the changelog. After each of the workshops, updates will follow containing the solutions to the practice exercises.

Contact information 💬

Contents

This course is taught by Tom van Woudenberg and Hans Welleman. Please contact us if you've any questions, feedback or when you've personal circumstances which we should know.

Tom van Woudenberg

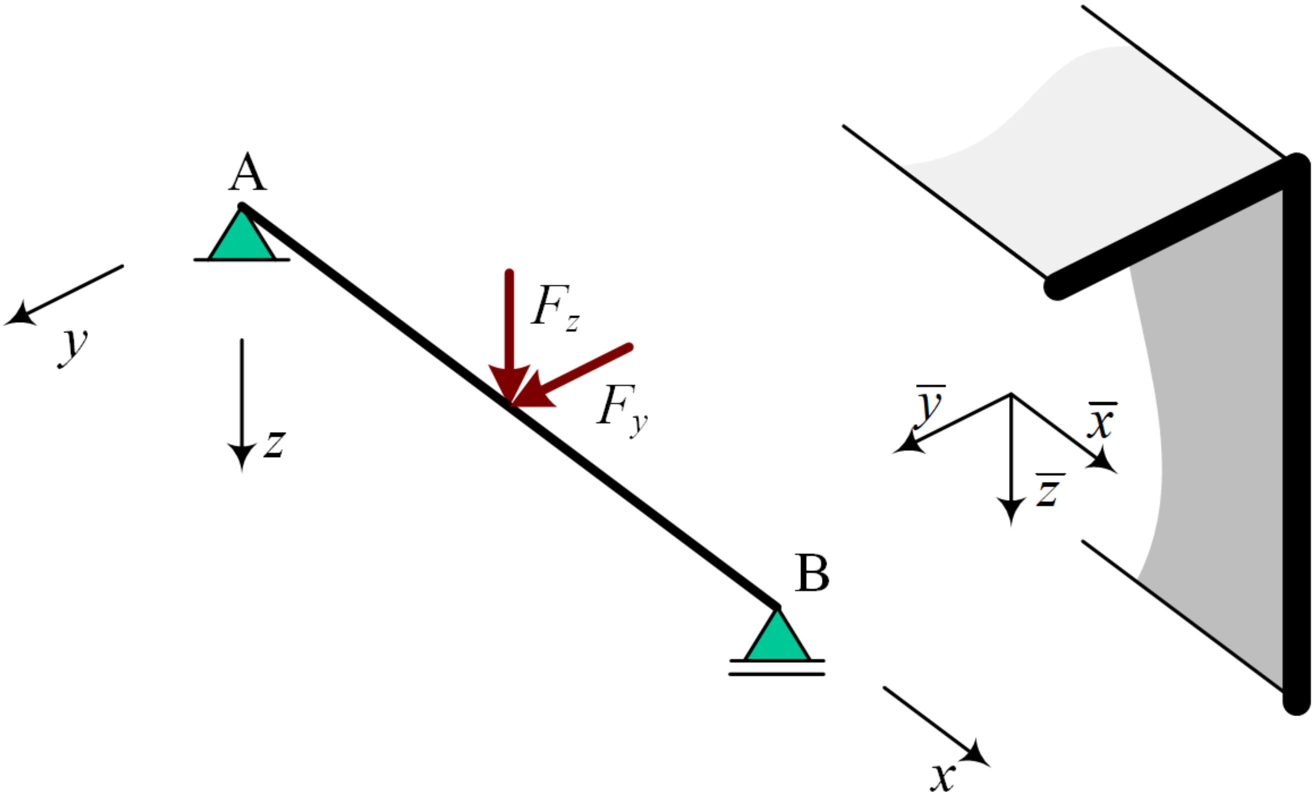
- Room 6.45
- 015-2789739
- T.R.vanWoudenberg@tudelft.nl

Hans Welleman

- Room 6.55
- 015-2784856
- J.W.Welleman@tudelft.nl

Unsymmetrical bending

The bending behaviour of the following beam is calculated:



If you download this file you can play around with the actual widget

```
import sympy as sym
sym.init_printing()
```

```
x = sym.symbols('x')
F_z, F_y = sym.symbols('F_z F_y')
C1, C2, C3, C4, C5, C6, C7, C8 = sym.symbols('C1 C2 C3 C4 C5 C6 C7 C8')

qz = sym.SingularityFunction(x,750,-1)*1000 * F_z
qy = sym.SingularityFunction(x,750,-1)*1000 * F_y
```

```
Vz = -sym.integrate(qz,x) + C1
Mz = sym.integrate(Vz,x) + C2

Vy = -sym.integrate(qy,x) + C3
My  =sym.integrate(Vy,x) + C4
```

```
eq1 = sym.Eq(Mz.subs(x,0),0)
eq2 = sym.Eq(Mz.subs(x,1500),0)
eq3 = sym.Eq(My.subs(x,0),0)
eq4 = sym.Eq(My.subs(x,1500),0)
sol = sym.solve([eq1,eq2,eq3,eq4],(C1,C2,C3,C4))
Mz = Mz.subs(sol)
My = My.subs(sol)
```

```
b = 75
h = 150
t = 10
A = b*t+h*t - t*t
NC_z = ( (b-t) * t * t/2 + h * t * h / 2 ) / A
NC_y = ( (h-t)*t*t/2 + b * t * b / 2 ) / A
Izz = (1/12 * (b-t) * t **3 + 1/12 * t * h**3 + b * t * (NC_z-t/2)**2 + h * t * (NC_z - h
Iyy = (1/12 * t * b **3 + 1/12 * (h - t) * t**3 + h * t * (t/2-NC_y)**2 + b * t * (NC_y -
Iyz = (b * t * (b/2 - NC_y) * (- NC_z+t/2) + h * t * (- NC_y+t/2) * (h / 2 - NC_z) - t *

E = 210000
kappa_y = 1e-4/E*1/(Izz*Iyy-(-Iyz)**2)*( Izz*My - Iyz*Mz)
kappa_z = 1e-4/E*1/(Izz*Iyy-(-Iyz)**2)*( -Iyz*My + Iyy*Mz)
```

```
phi_y = sym.integrate(kappa_z,x) + C5
phi_z = -sym.integrate(kappa_y,x) + C6
w_y = sym.integrate(phi_z,x) + C7
w_z = -sym.integrate(phi_y,x) + C8
```

```
eq5 = sym.Eq(w_y.subs(x,0),0)
eq6 = sym.Eq(w_y.subs(x,1500),0)
eq7 = sym.Eq(w_z.subs(x,0),0)
eq8 = sym.Eq(w_z.subs(x,1500),0)
sol = sym.solve([eq5,eq6,eq7,eq8],(C5,C6,C7,C8))
y, z = sym.symbols('y z')
```

```
u = - y * phi_z + z * phi_y

sigma = E * (kappa_y * y + kappa_z * z)

sigma.subs(x,750).subs(y,75-15.83).subs(z,-51.67).subs(F_z,27).subs(F_y,9)
neutral_axis = sym.solve(sigma,z)[0]
```

ODE with adapted load

This page solves a cantilerver beam

```
import sympy as sym
sym.init_printing()
```

```
E, a = sym.symbols('E, a')

EI_yy = EI_zz = E * (a * 2) * (a * 2) **3 / 36
EI_yz = E * (a * 2 * (a * 2) **3 ) / (36 * 2 * a / a)
```

```
x, q_z, L= sym.symbols('x,  q_z, L')
u_y, u_z = sym.Function('u_y'), sym.Function('u_z')
ode_y = sym.Eq( EI_yy * sym.diff(u_y(x), x, 4), ( - EI_yz * EI_yy * q_z) / (EI_yy * EI_zz
ode_z = sym.Eq( EI_zz * sym.diff(u_z(x), x, 4), ( + EI_yy * EI_zz * q_z) / (EI_yy * EI_zz

C1, C2, C3, C4, C5, C6, C7, C8 = sym.symbols('C1, C2, C3, C4, C5, C6, C7, C8')
sol = sym.dsolve([ode_y,ode_z], [u_y(x),u_z(x)])
u_y = sol[0].rhs
u_z = sol[1].rhs

phi_y = -sym.diff(u_z,x)
phi_z = sym.diff(u_y,x)
kappa_y = -sym.diff(phi_z,x)
kappa_z = sym.diff(phi_y,x)
M_y = EI_yy * kappa_y + EI_yz * kappa_z
M_z = EI_zz * kappa_z + EI_yz * kappa_y

eq1 = sym.Eq(u_y.subs(x,0),0)
eq2 = sym.Eq(u_z.subs(x,0),0)
eq3 = sym.Eq(phi_y.subs(x,0),0)
eq4 = sym.Eq(phi_z.subs(x,0),0)
eq5 = sym.Eq(u_y.subs(x,L),0)
eq6 = sym.Eq(u_z.subs(x,L),0)
eq7 = sym.Eq(M_y.subs(x,L),0)
eq8 = sym.Eq(M_z.subs(x,L),0)

sol = sym.solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8], [C1, C2, C3, C4, C5, C6, C7, C8])
display(sol)
```

$$\left\{ C_1 : 0,\ C_2 : 0,\ C_3 : -\frac{3L^2q_z}{16Ea^4},\ C_4 : \frac{15Lq_z}{16Ea^4},\ C_5 : 0,\ C_6 : 0,\ C_7 : \frac{3L^2q_z}{8Ea^4},\ C_8 : -\frac{15Lq_z}{8Ea^4} \right\}$$

```
display(u_y.subs(sol))
```

$$-\frac{3L^2q_zx^2}{32Ea^4} + \frac{5Lq_zx^3}{32Ea^4} - \frac{q_zx^4}{16Ea^4}$$

```python
display(u_z.subs(sol))
```

$$\frac{3L^2 q_z x^2}{16Ea^4} - \frac{5Lq_z x^3}{16Ea^4} + \frac{q_z x^4}{8Ea^4}$$

```python
display(sym.simplify(M_y.subs(sol)))
```

$$0$$

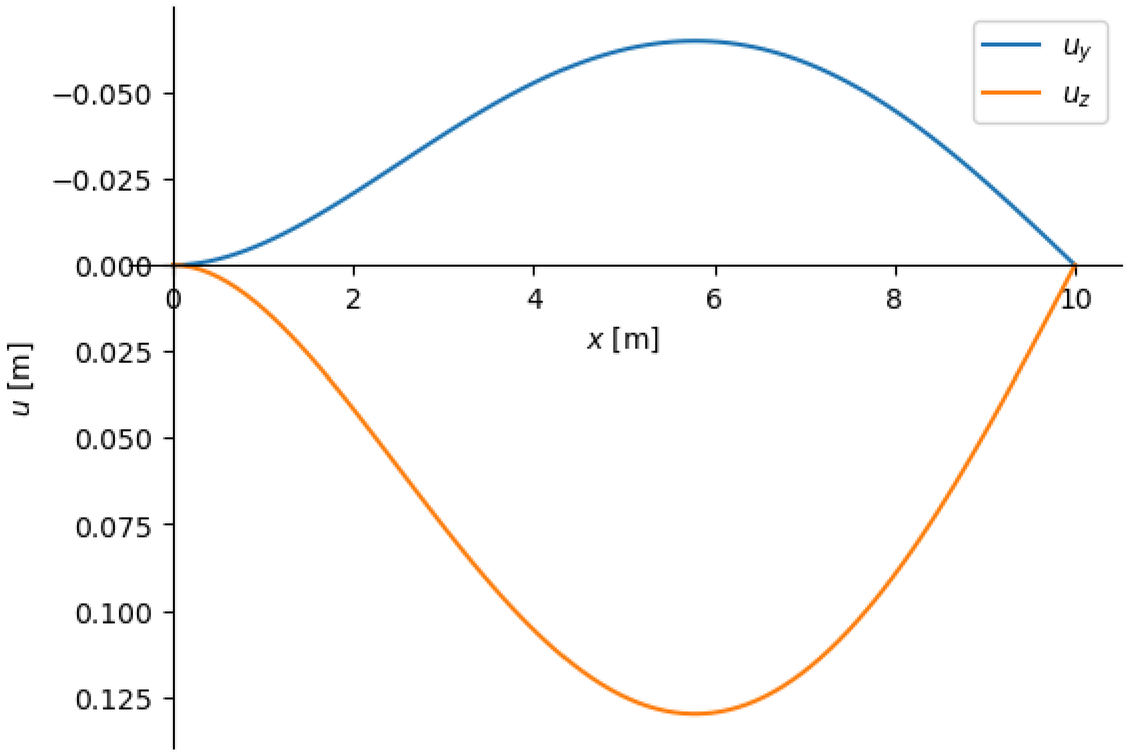
```python
display(sym.simplify(M_z.subs(sol)))
```

$$\frac{q_z\left(-L^2 + 5Lx - 4x^2\right)}{8}$$

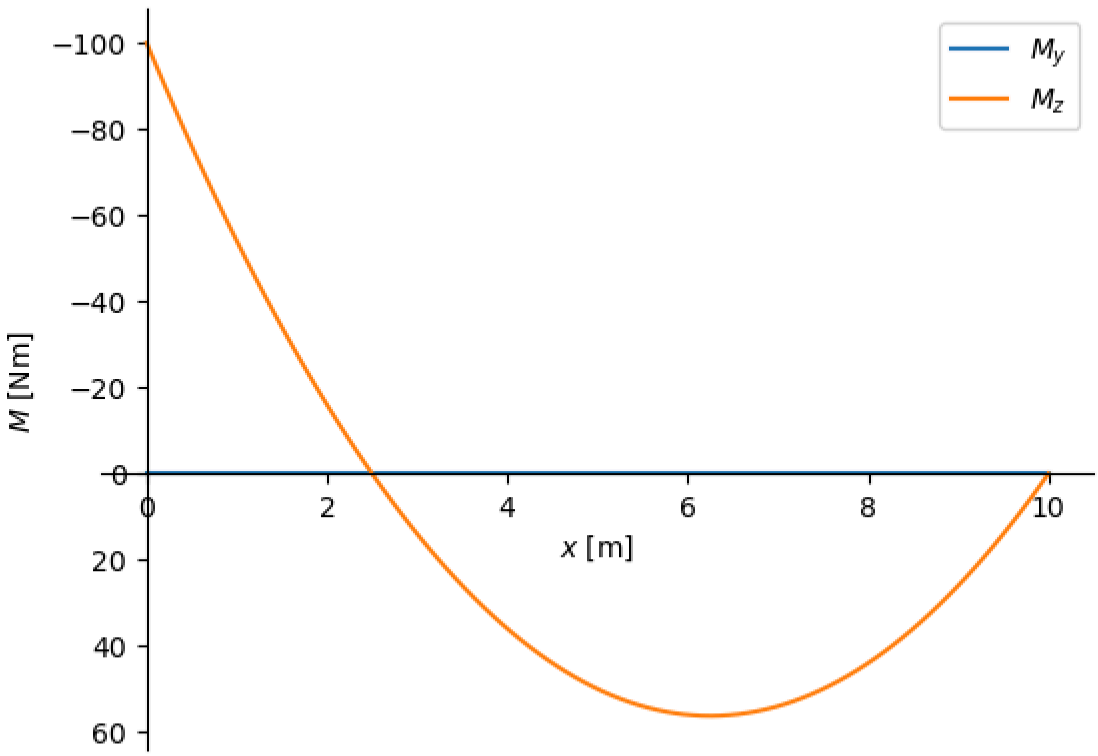
```python
u_y_func = sym.lambdify([x, q_z, L, E, a], u_y.subs(sol))
u_z_func = sym.lambdify([x, q_z, L, E, a], u_z.subs(sol))
M_y_func = sym.lambdify([x, q_z, L, E, a], M_y.subs(sol))
M_z_func = sym.lambdify([x, q_z, L, E, a], M_z.subs(sol))
```

```python
import numpy as np
import matplotlib.pyplot as plt
```

```python
x_plot = np.linspace(0,10,100)
plt.plot(x_plot,u_y_func(x_plot,8,10,100e6,0.1),label='$u_y$')
plt.plot(x_plot,u_z_func(x_plot,8,10,100e6,0.1),label='$u_z$')
plt.legend(loc='best')
plt.ylabel('$u$ [m]')
plt.xlabel('$x$ [m]')
plt.gca().invert_yaxis()
plt.gca().spines['left'].set_position('zero')
plt.gca().spines['bottom'].set_position('zero')
plt.gca().spines['right'].set_color('none')
plt.gca().spines['top'].set_color('none')
```




```
plt.figure()
plt.plot(x_plot,M_y_func(x_plot,8,10,100e6,0.1),label='$M_y$')
plt.plot(x_plot,M_z_func(x_plot,8,10,100e6,0.1),label='$M_z$')
plt.legend(loc='best')
plt.ylabel('$M$ [Nm]')
plt.xlabel('$x$ [m]')
plt.gca().invert_yaxis()
plt.gca().spines['left'].set_position('zero')
plt.gca().spines['bottom'].set_position('zero')
plt.gca().spines['right'].set_color('none')
plt.gca().spines['top'].set_color('none')
```



Evaluate strain energy using SymPy

```
import sympy as sym
```

```
L, x, q, GA_C, EI = sym.symbols('L, x, q, GA_C, EI')
```

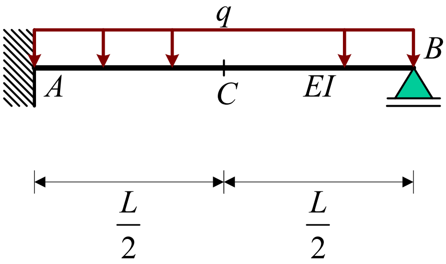
```
 M = -q * (L-x)**2 / 2
E_c_M = sym.integrate(M**2/EI/2,(x,0,L))
display(E_c_M)
```

$$\frac{L^5 q^2}{40EI}$$

```
V = (L-x)*q
E_c_V = sym.integrate(V**2/GA_C/2,(x,0,L))
display(E_c_V)
```

$$\frac{L^3 q^2}{6GA_C}$$

Example deflection using Castigliano




```
import sympy as sym
```

```
q, L, x, F_c, B_v, EI = sym.symbols('q, L, x, F_c, B_v, EI')
```

```
M_1 = -q * (L - x)**2 / 2 - F_c * (L/2 - x) + B_v * (L-x)
M_2 = -q * (L - x)**2 / 2 + B_v * (L-x)
display(M_1)
display(M_2)
```

$$B_v(L - x) - F_c\left(\frac{L}{2} - x\right) - \frac{q(L - x)^2}{2}$$

$$B_v(L - x) - \frac{q(L - x)^2}{2}$$

```
display(M_1.diff(B_v))
display(M_2.diff(B_v))
```

$$L - x$$

$$L - x$$

```

```

```
w = sym.integrate(M_1 * M_1.diff(B_v) / EI, (x,0,L/2)) + sym.integrate(M_2 * M_2.diff(B_v
B_v_sol = sym.solve(w,B_v)[0]
display(B_v_sol)
```

$$\frac{5F_c}{16} + \frac{3Lq}{8}$$

```
M_1 = M_1.subs(B_v,B_v_sol)
M_2 = M_2.subs(B_v,B_v_sol)
display(M_1)
display(M_2)
```

$$-F_c\left(\frac{L}{2} - x\right) - \frac{q(L-x)^2}{2} + \left(\frac{5F_c}{16} + \frac{3Lq}{8}\right)(L-x)$$

$$-\frac{q(L-x)^2}{2} + \left(\frac{5F_c}{16} + \frac{3Lq}{8}\right)(L-x)$$

```
display(M_1.diff(F_c))
display(M_2.diff(F_c))
```

$$-\frac{3L}{16} + \frac{11x}{16}$$

$$\frac{5L}{16} - \frac{5x}{16}$$

```
w_C = sym.integrate(M_1 * M_1.diff(F_c) / EI, (x,0,L/2)) + sym.integrate(M_2 * M_2.diff(F
display(w_C.subs(F_c,0))
```

$$\frac{L^4q}{192EI}$$

Potential energy conceptual

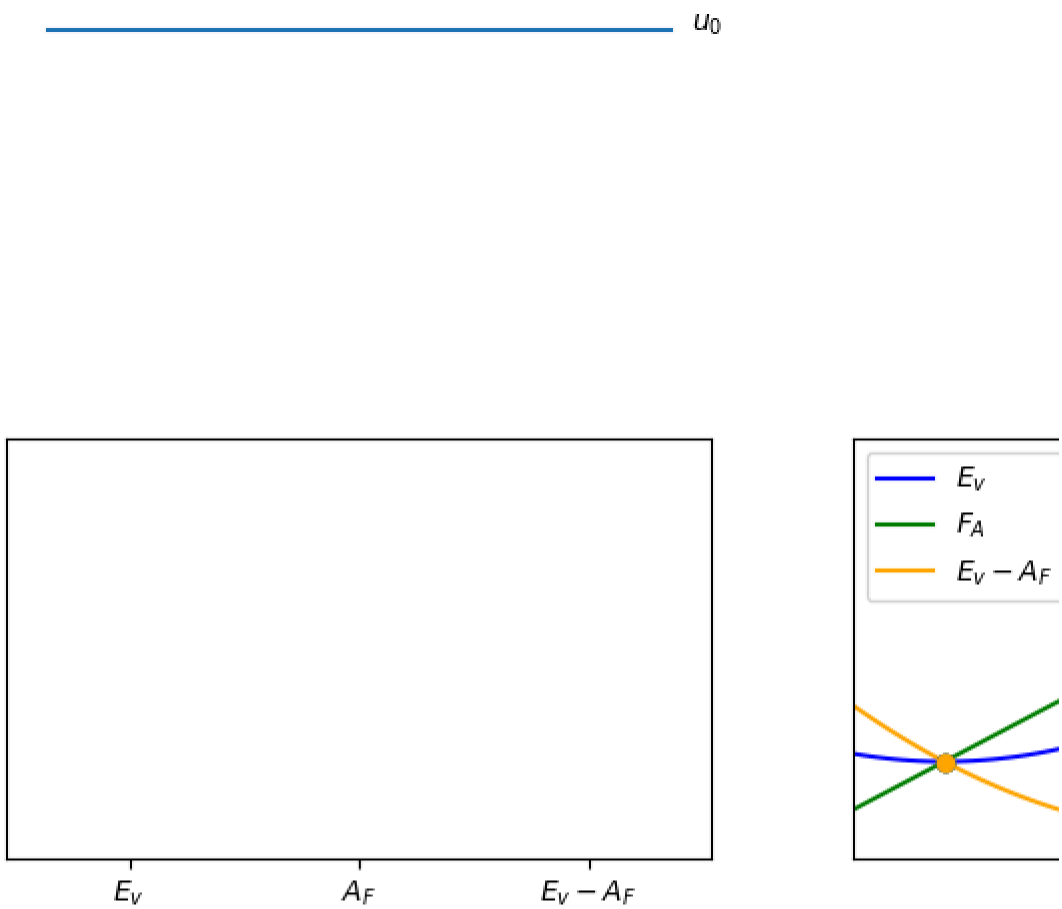
Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some potential energy

Displacement u_0 0.0000

trial_function

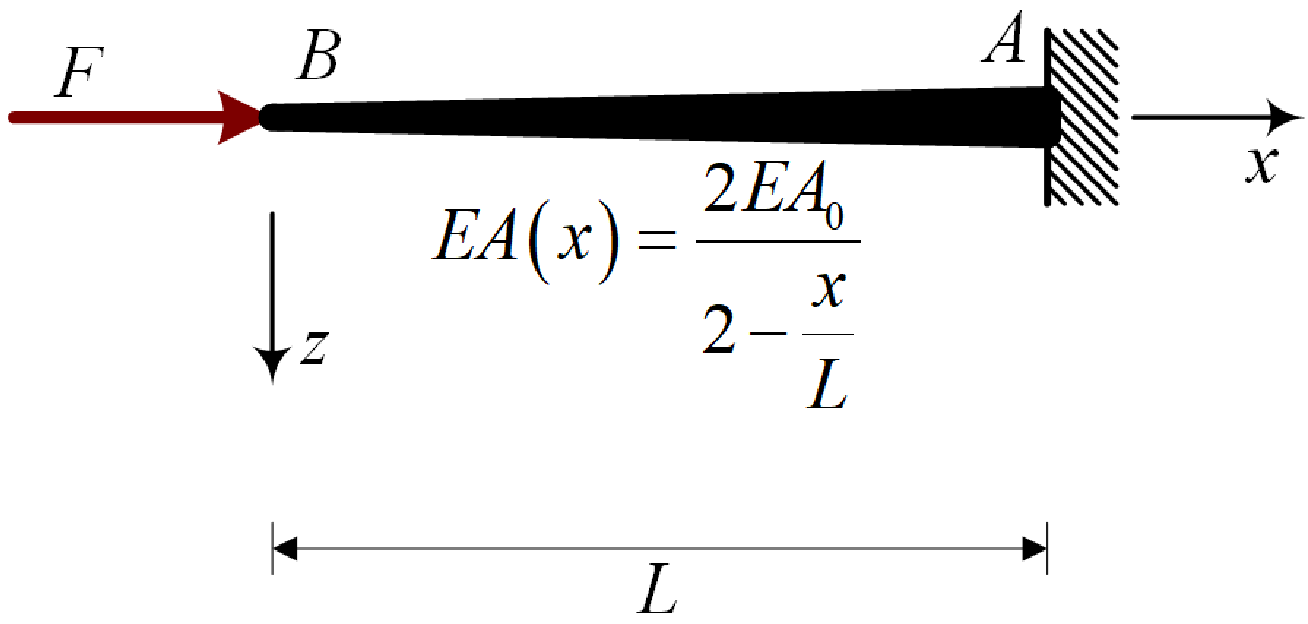
Polynomial cos cosh

Figure 1



Potential energy vs ODE

Contents



Solving use trial function

With trial function:

```
import sympy as sym
EA_0, a, x, L, F = sym.symbols('EA_0, a, x, L, F')
EA = 2 * EA_0 / (2 - x / L)
u_trial = a * (1 - x / L)
display(u_trial)
```

$$a\left(1 - \frac{x}{L}\right)$$

And strain energy:

```
eps_trial = u_trial.diff(x)
Ev = sym.simplify(sym.integrate( EA * eps_trial**2 / 2,(x,0,L)))
display(Ev)
```

$$\frac{EA_0 a^2 \log{(2)}}{L}$$

And total potential energy:

```
V = Ev  - F * u_trial.subs(x,0)
display(V)
eq1 = sym.Eq(sym.diff(V,a),0)
```

$$\frac{EA_0 a^2 \log{(2)}}{L} - Fa$$

Solving for a gives:

```
a_sol = sym.solve(eq1,a)[0]
u_trial_subs = u_trial.subs(a,a_sol)
display(u_trial_subs)
```

$$\frac{FL\left(1 - \frac{x}{L}\right)}{2EA_0 \log{(2)}}$$

Solving with ODE

```
u = sym.Function('u')
EA = 2 * EA_0 / (2 - x / L)
```

The ODE is defined as:

```
DV = sym.Eq((EA*u(x).diff(x)).diff(x),0)
display(DV)
```

$$\frac{2EA_0 \frac{d^2}{dx^2} u(x)}{2 - \frac{x}{L}} + \frac{2EA_0 \frac{d}{dx} u(x)}{L\left(2 - \frac{x}{L}\right)^2} = 0$$

The general solution of this ODE is:

```
C1, C2 = sym.symbols('C1, C2')
u_sol = sym.dsolve(DV,u(x)).rhs
display(u_sol)
```

$$C_1 + 2C_2Lx - \frac{C_2x^2}{2}$$

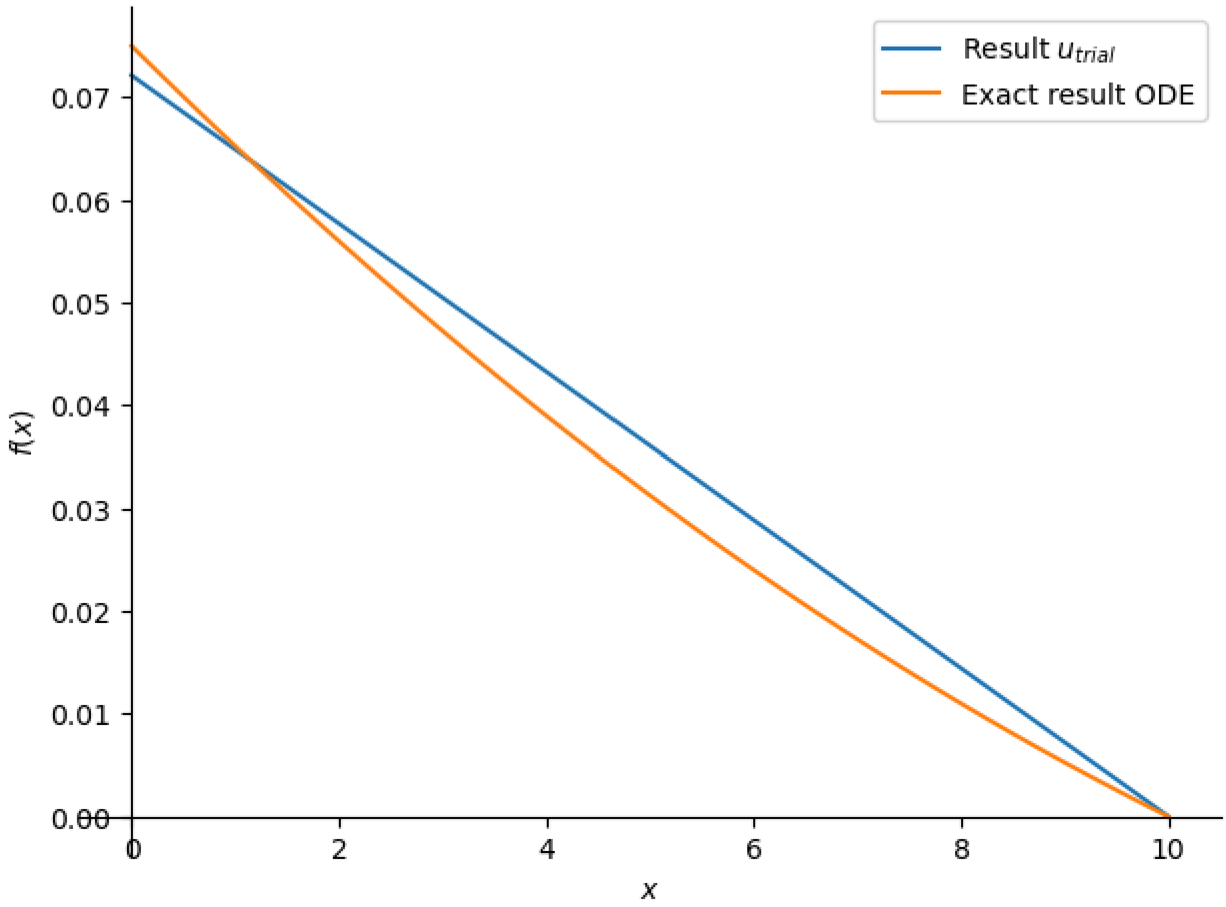
Solving for the boundary conditions gives:

```
eps = u_sol.diff(x)
N = EA * eps
eq1 = sym.Eq(N.subs(x,0),-F)
eq2 = sym.Eq(u_sol.subs(x,L),0)
sol = sym.solve((eq1,eq2),(C1,C2))
u_subs = u_sol.subs(sol)
display(u_subs)
```

$$\frac{3FL}{4EA_0} - \frac{Fx}{EA_0} + \frac{Fx^2}{4EA_0L}$$

Comparison

```
p0 = sym.plot(u_trial_subs.subs([(F,10),(EA_0,1000),(L,10)]), (x , 0 , 10),label='Result
p1 = sym.plot(u_subs.subs([(F,10),(EA_0,1000),(L,10)]), (x , 0 , 10),label='Exact result
p0.append(p1[0])
p0.show()
```



Result u_{trial}
Exact result ODE
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
$f(x)$
0
2
4
6
8
10
x

Exercise potential energy statically indeterminate structure

Contents

Find the displacement in C using potential energy and compare it to the result from solving the differential equation.

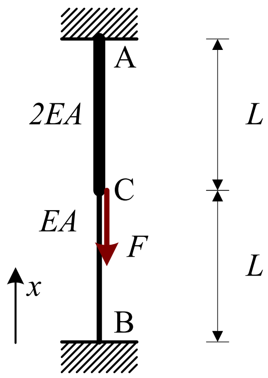


Fig. 1 Structure

Solve using potential energy

Let's start by defining the trial function. Let's assume a parabolic displacement field with the shape $u\left(x\right) = a + bx + cx^2$.

Exercise (Boundary conditions kinematical admissible trial function)

The displacement field must be kinematically admissible. Which boundary conditions must be enforced for that?

Solution to Exercise (Boundary conditions kinematical admissible trial function)

- $u(0) = 0$
- $u(2L) = 0$

Exercise (Kinematical admissibility trial function)

Now, let's use SymPy to rewrite the trial function so that it is kinematically admissible.

```
import sympy as sym
a, b, c, x, L, EA, F = sym.symbols('a, b, c, x, L, EA, F')
```

```
u_trial_general = a + b * x + c * x **2

eq1 = sym.Eq(u_trial_general.subs(x, #value to substitute for x), #right hand si
eq2 = #...
sol = sym.solve((eq1,eq2),(a,b))
display(sol)
u_trial = u_trial_general.subs(sol)
display(u_trial)
```

Let's check your solution graphically for a value of 1 for c and 5 for L:

```
import matplotlib.pyplot as plt
import numpy as np
%config InlineBackend.figure_formats = ['svg']
```

```
%matplotlib inline
u_trial_func = sym.lambdify(x, u_trial.subs(sol).subs(c,1).subs(L,5))
x_plot = np.linspace(0, 10,100)
fig, ax = plt.subplots()
ax.plot(u_trial_func(x_plot),x_plot)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('u(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```

Solution to Exercise (Kinematical admissibility trial function)

The answer should be $-2\,L\,c\,x + c\,x^2$ and the graph should match your boundary conditions of exercise on boundary conditions kinematical admissible trial function

```
import sympy as sym
a, b, c, x, L, EA, F = sym.symbols('a, b, c, x, L, EA, F')
```

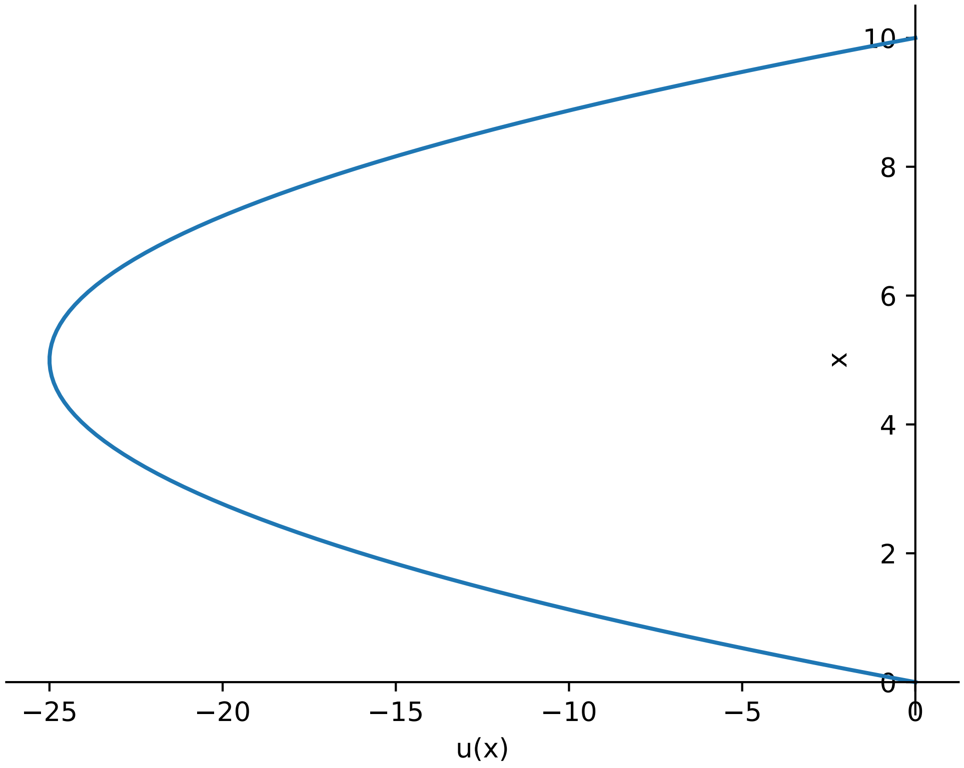
```
u_trial_general = a+b*x+c*x**2

eq1 = sym.Eq(u_trial_general.subs(x,0), 0)
eq2 = sym.Eq(u_trial_general.subs(x,2*L), 0)
sol = sym.solve((eq1,eq2),(a,b))
display(sol)
u_trial = u_trial_general.subs(sol)
display(u_trial)
```

```
{a: 0, b: -2*L*c}
```

$$-2Lcx + cx^2$$

```
import matplotlib.pyplot as plt
import numpy as np
%config InlineBackend.figure_formats = ['svg']
```

```
%matplotlib inline
u_trial_func = sym.lambdify(x, u_trial.subs(sol).subs(c,1).subs(L,5))
x_plot = np.linspace(0, 10,100)
fig, ax = plt.subplots()
ax.plot(u_trial_func(x_plot),x_plot)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('u(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```



Exercise (epsilon)

For the strain energy we needed the strain distribution instead of the displacement. The strain is the derivative of the displacement, which can be found with SymPy:

For this exercise you just have to run the cell below.

```
epsilon_trial = sym.diff(u_trial,x)
display(epsilon_trial)
```

Solution to Exercise (epsilon)

You should have found: $\epsilon\left(x\right) = -2\,L\,c + 2\,c\,x$

```
epsilon_trial = sym.diff(u_trial,x)
display(epsilon_trial)
```

$$-2Lc + 2cx$$

Exercise (strain energy)

Now let's evaluate the strain energy:

```
E_v_trial = sym.integrate(#term to integrate,(x,#integrate from, #integrate to))
display(E_v_trial)
```

Solution to Exercise (strain energy)

The answer should be $2\ EA\ L^3c^2$

```
E_v_trial = sym.integrate(EA*epsilon_trial**2/2,(x,0,L))+sym.integrate(EA*2*epsi
display(E_v_trial)
```

$$2EAL^3c^2$$

Exercise (Work by force)

What's the work done by the force F?

Solution to Exercise (Work by force)

$$-F \cdot u(L)$$

Exercise (Potential energy)

Find the relation for the potential energy.

```
V = #expression for V
display(V)
```

Solution to [Exercise (Potential energy)]

The answer should be $2EA\,L^3c^2 - F\,L^2c$

```
V = E_v_trial - ( -F * u_trial.subs(x,L))
display(V)
```

$$2EAL^3c^2 - FL^2c$$

Exercise (Stationary strain energy)

Now, let's impose the stationary condition for the strain energy in an equation which we can use to solve for the unknown c.

```
eq3 = sym.Eq(#left hand side of equation, #right hand side of equation)
display(eq3)
c_sol = sym.solve(eq3,c)[0]
display(c_sol)
u_trial_sol = u_trial.subs(c,c_sol)
display(u_trial_sol)
```

Let's check your solution graphically for a value of 5 m for L, 2000 kN for EA and 30 kN for F:

```
%matplotlib inline
L_test = 5
EA_test = 2000
F_test = 30
u_trial_func = sym.lambdify(x, u_trial_sol.subs(L,L_test).subs(EA,EA_test).subs(
x_plot = np.linspace(0, L_test*2,100)
fig, ax = plt.subplots()
ax.plot(u_trial_func(x_plot),x_plot)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('u(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```

Solution to Exercise (Stationary strain energy)

The answer should be $-\frac{F\,x}{2\,EA}+\frac{F\,x^2}{4\,EA\,L}$

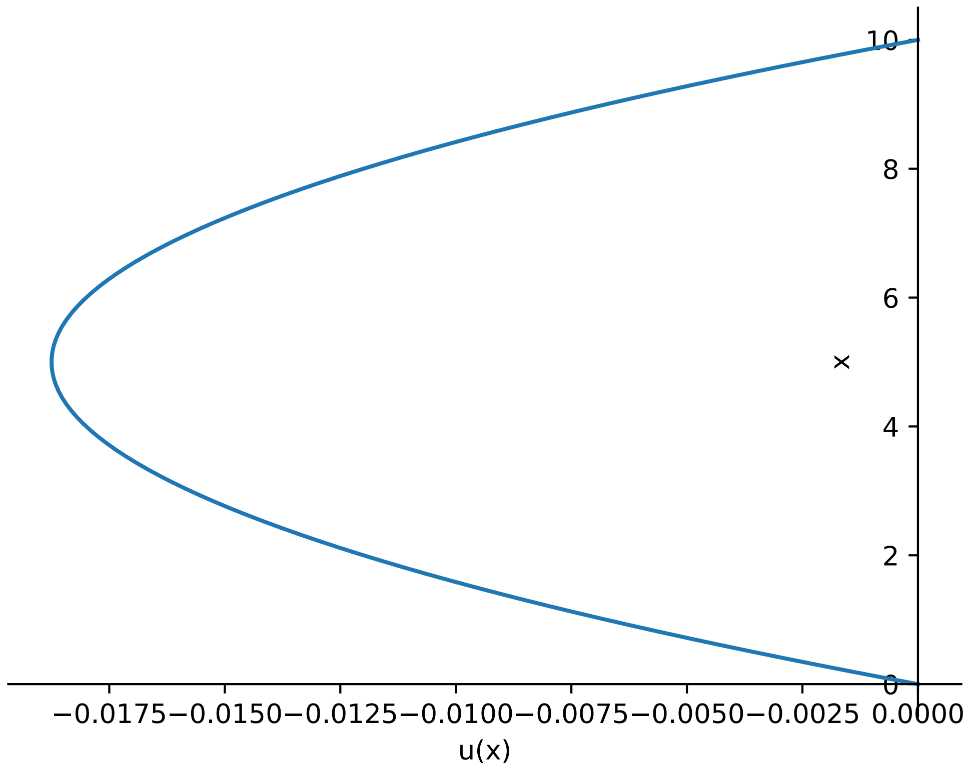
```
eq3 = sym.Eq(V.diff(c),0)
display(eq3)
c_sol = sym.solve(eq3,c)[0]
display(c_sol)
u_trial_sol = u_trial.subs(c,c_sol)
display(u_trial_sol)
```

$$4EAL^3c - FL^2 = 0$$

$$\frac{F}{4EAL}$$

$$-\frac{Fx}{2EA}+\frac{Fx^2}{4EAL}$$

```
%matplotlib inline
L_test = 5
EA_test = 2000
F_test = 30
u_trial_func = sym.lambdify(x, u_trial_sol.subs(L,L_test).subs(EA,EA_test).subs(
x_plot = np.linspace(0, L_test*2,100)
fig, ax = plt.subplots()
ax.plot(u_trial_func(x_plot),x_plot)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('u(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```


10
8
6
4
2
0
x
−0.0175
−0.0150
−0.0125
−0.0100
−0.0075
−0.0050
−0.0025
0.0000
u(x)

Exercise (Normal force)

What is the normal force distribution?

```
epsilon = #...

N_trial_sol_AC = #
N_trial_sol_BC = #

display(N_trial_sol_AC.simplify())
display(N_trial_sol_BC.simplify())
```

Let's check your solution graphically for a value of 5 m for L, 2000 kN for EA and 30 kN for F:

```
%matplotlib inline
L_test = 5
EA_test = 2000
F_test = 30
fig, ax = plt.subplots()

N_trial_func_AC = sym.lambdify(x, N_trial_sol_AC.subs(sol).subs(L,L_test).subs(E
N_trial_func_BC = sym.lambdify(x, N_trial_sol_BC.subs(sol).subs(L,L_test).subs(E
x_plot_AC = np.linspace(0, L_test,2)
x_plot_BC = np.linspace(L_test, L_test*2,2)
ax.plot(N_trial_func_AC(x_plot_AC),x_plot_AC)
ax.plot(N_trial_func_BC(x_plot_BC),x_plot_BC)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('N(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```

Solution to Exercise (Normal force)

You should have found:

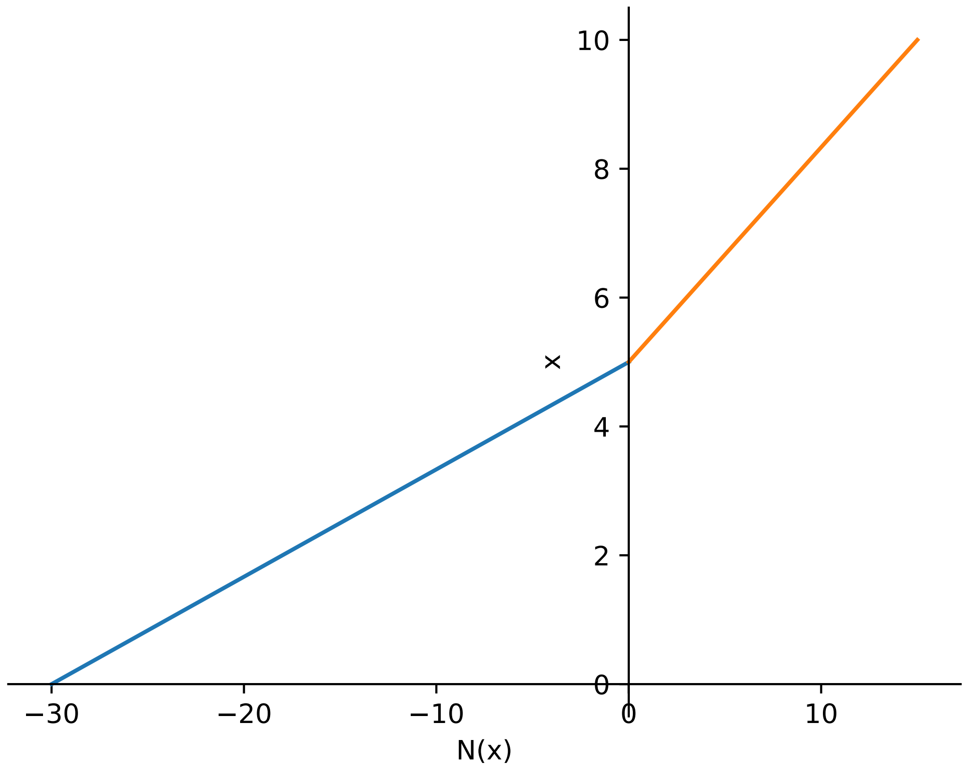
- $N_{\text{AC,trial}}(x) = \frac{F\,(-L+x)}{L}$
- $N_{\text{BC,trial}}(x) = \frac{F\,(-L+x)}{2\,L}$

```
epsilon = sym.diff(u_trial_sol,x)

N_trial_sol_AC = 2 * EA * epsilon
N_trial_sol_BC = EA * epsilon

display(N_trial_sol_AC.simplify())
display(N_trial_sol_BC.simplify())
```

$$\frac{F\,(-L+x)}{L}$$

$$\frac{F\,(-L+x)}{2L}$$

```
%matplotlib inline
L_test = 5
EA_test = 2000
F_test = 30
fig, ax = plt.subplots()

N_trial_func_AC = sym.lambdify(x, N_trial_sol_AC.subs(sol).subs(L,L_test).subs(E
N_trial_func_BC = sym.lambdify(x, N_trial_sol_BC.subs(sol).subs(L,L_test).subs(E
x_plot_AC = np.linspace(0, L_test,2)
x_plot_BC = np.linspace(L_test, L_test*2,2)
ax.plot(N_trial_func_AC(x_plot_AC),x_plot_AC)
ax.plot(N_trial_func_BC(x_plot_BC),x_plot_BC)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('N(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```



Solve with differential equation

Now, let’s solve the same structure using the differential equation.

Exercise (Generalised relations)

Given a load $q = 0$, integrate twice to find the normal force, strain and displacement relations with integration constants:

```
C_1, C_2, C_3, C_4 = sym.symbols('C_1, C_2, C_3, C_4')

q_AC = 0
q_BC = 0

N_AC = C_1 + sym.integrate(#..., x)
N_BC = C_3 + #...

display(N_AC)
display(N_BC)

epsilon_AC = #...
epsilon_BC = #...

display(epsilon_AC)
display(epsilon_BC)

u_AC = C_2 + #...
u_BC = C_3 + #...

display(u_AC)
display(u_BC)
```

Solution to Exercise (Generalised relations)

You should have found:

- $N_{\text{AC}} \;\; = C_1$
- $N_{\text{BC}} \;\; = C_3$
- $\epsilon_{\text{AC}} \;\; = \dfrac{C_1}{2\,EA}$
- $\epsilon_{\text{BC}} \;\; = \dfrac{C_3}{EA}$
- $u_{\text{AC}} \;\; = \dfrac{C_1}{2\,EA}x + C_2$
- $u_{\text{BC}} \;\; = \dfrac{C_3}{EA}x + C_4$

```python
C_1, C_2, C_3, C_4 = sym.symbols('C_1, C_2, C_3, C_4')

q_AC = 0
q_BC = 0

N_AC = C_1 + sym.integrate(q_AC, x)
N_BC = C_3 + sym.integrate(q_BC, x)

display(N_AC)
display(N_BC)

epsilon_AC = N_AC / EA / 2
epsilon_BC = N_BC / EA

display(epsilon_AC)
display(epsilon_BC)

u_AC = C_2 + sym.integrate(epsilon_AC, x)
u_BC = C_4 + sym.integrate(epsilon_BC, x)

display(u_AC)
display(u_BC)
```

$$C_1$$

$$C_3$$

$$\frac{C_1}{2EA}$$

$$\frac{C_3}{EA}$$

$$\frac{C_1 x}{2EA} + C_2$$

$$\frac{C_3 x}{EA} + C_4$$

Exercise (Boundary conditions)

What are the boundary conditions?

Solution to Exercise (Boundary conditions)

- $u_{\text{AC}}\ (0) = 0$
- $u_{\text{AC}}\ (L) = u_{\text{BC}}\ (L)$
- $u_{\text{BC}}\ (2\ L) = 0$
- $N_{\text{AC}}\ (L) + F = N_{\text{BC}}\ (L)$

Exercise (Solve boundary conditions)

Now, let's use SymPy to solve for the boundary conditions using `sym.solve` similarly as you did in the exercise for the kinematical admissibility trial function and find the solution for u_AC , u_BC ,N_\text{AC}andN_\text{BC}

```
eq4 = #...
#...

C_sol = sym.solve((eq4,#...),[#])
display(sol)

u_AC_sol = u_AC.subs(C_sol)
u_BC_sol = u_BC.subs(C_sol)
display(u_AC_sol)
display(u_BC_sol)
N_AC_sol = #
N_BC_sol = #
display(N_AC_sol)
display(N_BC_sol)
```

Let's check your solution graphically for a value of 5 m for L, 2000 kN for EA and 30 kN for F:

```
%matplotlib inline
u_AC_func = sym.lambdify(x, u_AC_sol.subs(sol).subs(L,L_test).subs(EA,EA_test).s
u_BC_func = sym.lambdify(x, u_BC_sol.subs(sol).subs(L,L_test).subs(EA,EA_test).s
x_plot_AC = np.linspace(0, L_test,100)
x_plot_BC = np.linspace(L_test, L_test*2,100)
fig, ax = plt.subplots()
ax.plot(u_AC_func(x_plot_AC),x_plot_AC)
ax.plot(u_BC_func(x_plot_BC),x_plot_BC)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('u(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```

```
N_AC_func = sym.lambdify(x, N_AC_sol.subs(L,L_test).subs(EA,EA_test).subs(F,F_te
N_BC_func = sym.lambdify(x, N_BC_sol.subs(L,L_test).subs(EA,EA_test).subs(F,F_te
x_plot_AC = np.linspace(0, L_test,2)
x_plot_BC = np.linspace(L_test, L_test*2,2)
fig, ax = plt.subplots()
ax.plot(N_AC_func(x_plot_AC)*np.ones(2),x_plot_AC)
ax.plot(N_BC_func(x_plot_BC)*np.ones(2),x_plot_BC)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('u(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```

Solution to Exercise (Generalised relations)

You should find:

- $u_{\text{AC}} = -\frac{F\,x}{3\,EA}$
- $u_{\text{BC}} = -\frac{2\,F\,L}{3\,EA} + \frac{F\,x}{3\,EA}$
- $N_{\text{AC}} = -\frac{2\,F}{3}$
- $N_{\text{BC}} = -\frac{F}{3}$

```
eq4 = sym.Eq(u_AC.subs(x,0), 0)
eq5 = sym.Eq(u_AC.subs(x,L), u_BC.subs(x,L))
eq6 = sym.Eq(u_BC.subs(x,2*L), 0)
eq7 = sym.Eq(N_AC.subs(x,L)+F, N_BC.subs(x,L))

display(eq4)
display(eq5)
display(eq6)
display(eq7)

C_sol = sym.solve((eq4,eq5,eq6,eq7),(C_1,C_2,C_3,C_4))
display(C_sol)

u_AC_sol = u_AC.subs(C_sol)
u_BC_sol = u_BC.subs(C_sol)
display(u_AC_sol)
display(u_BC_sol)
N_AC_sol = N_AC.subs(C_sol)
N_BC_sol = N_BC.subs(C_sol)
display(N_AC_sol)
display(N_BC_sol)
```

$$C_2 = 0$$

$$\frac{C_1 L}{2EA} + C_2 = \frac{C_3 L}{EA} + C_4$$

$$\frac{2C_3 L}{EA} + C_4 = 0$$

$$C_1 + F = C_3$$

```
{C_1: -2*F/3, C_2: 0, C_3: F/3, C_4: -2*F*L/(3*EA)}
```

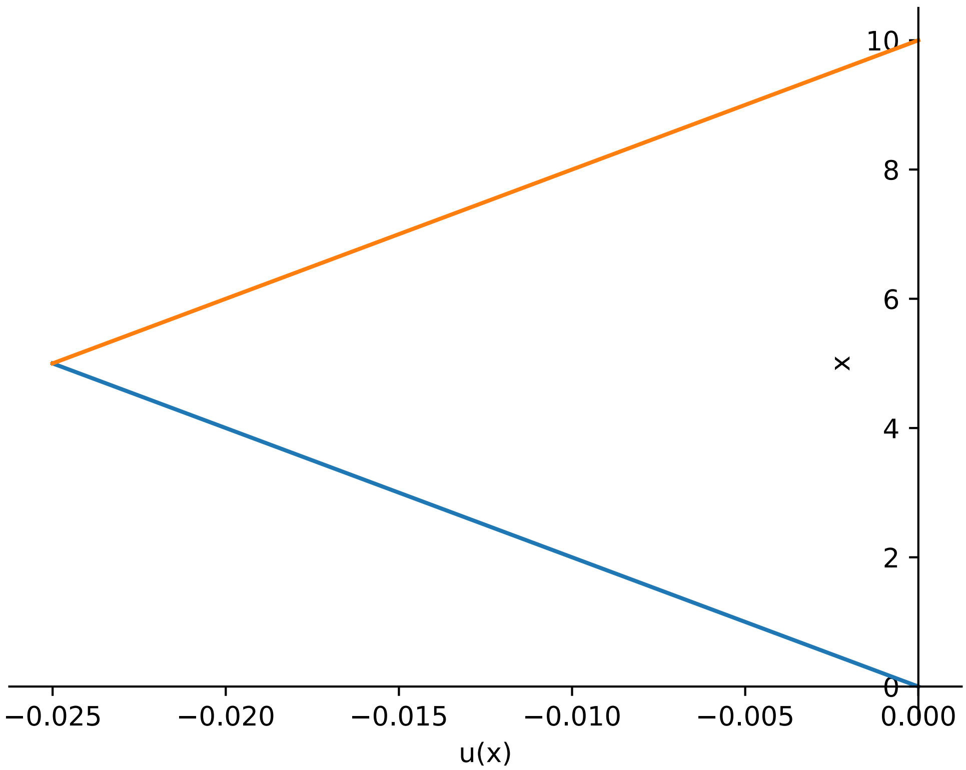
$$-\frac{Fx}{3EA}$$

$$-\frac{2FL}{3EA} + \frac{Fx}{3EA}$$

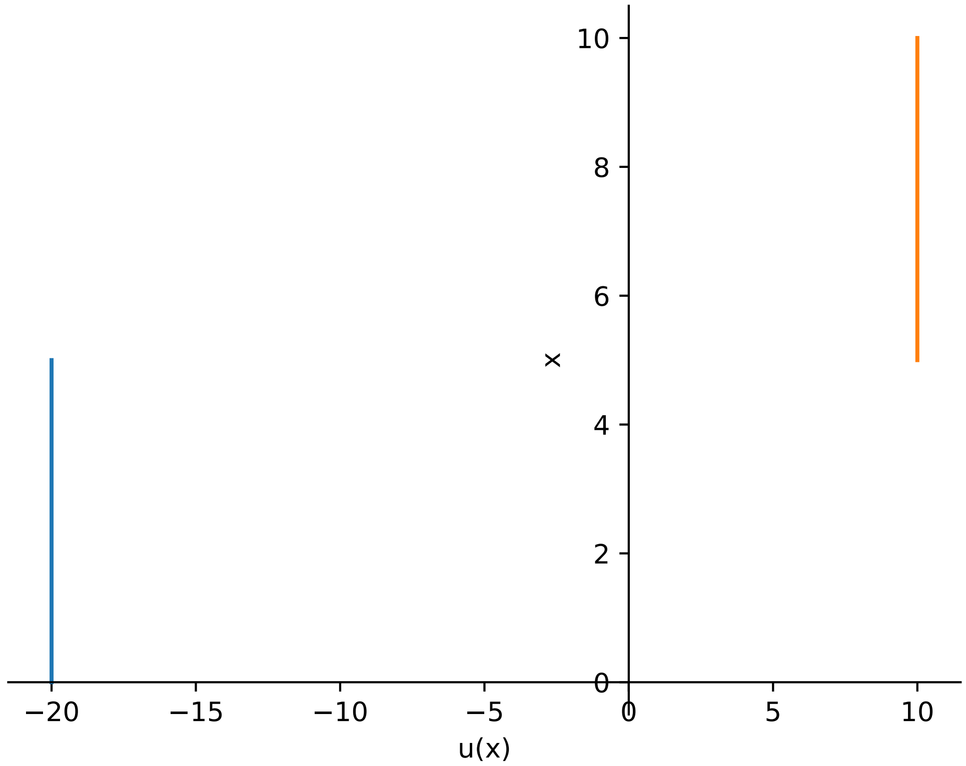
$$-\frac{2F}{3}$$

$$\frac{F}{3}$$

```
%matplotlib inline
u_AC_func = sym.lambdify(x, u_AC_sol.subs(sol).subs(L,L_test).subs(EA,EA_test).s
u_BC_func = sym.lambdify(x, u_BC_sol.subs(sol).subs(L,L_test).subs(EA,EA_test).s
x_plot_AC = np.linspace(0, L_test,100)
x_plot_BC = np.linspace(L_test, L_test*2,100)
fig, ax = plt.subplots()
ax.plot(u_AC_func(x_plot_AC),x_plot_AC)
ax.plot(u_BC_func(x_plot_BC),x_plot_BC)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('u(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```



```
N_AC_func = sym.lambdify(x, N_AC_sol.subs(L,L_test).subs(EA,EA_test).subs(F,F_te
N_BC_func = sym.lambdify(x, N_BC_sol.subs(L,L_test).subs(EA,EA_test).subs(F,F_te
x_plot_AC = np.linspace(0, L_test,2)
x_plot_BC = np.linspace(L_test, L_test*2,2)
fig, ax = plt.subplots()
ax.plot(N_AC_func(x_plot_AC)*np.ones(2),x_plot_AC)
ax.plot(N_BC_func(x_plot_BC)*np.ones(2),x_plot_BC)
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.set_xlabel('u(x)')
ax.set_ylabel('x')
fig.patch.set_alpha(0.0)
ax.patch.set_alpha(0.0)
```



Now, let’s compare the plots of both solution strategies, you can try different values:

```
from ipywidgets import widgets, interact
```

```
%matplotlib widget

fig, ax = plt.subplots()
fig.canvas.toolbar_visible = False
def func(L_test, EA_test, F_test):
    ax.clear()
    u_trial_func = sym.lambdify(x, u_trial_sol.subs(sol).subs(L,L_test).subs(EA,EA_test).
    u_AC_func = sym.lambdify(x, u_AC_sol.subs(sol).subs(L,L_test).subs(EA,EA_test).subs(F
    u_BC_func = sym.lambdify(x, u_BC_sol.subs(sol).subs(L,L_test).subs(EA,EA_test).subs(F
    x_plot = np.linspace(0, L_test*2,100)
    x_plot_AC = np.linspace(0, L_test,2)
    x_plot_BC = np.linspace(L_test, L_test*2,2)
    ax.plot(u_AC_func(x_plot_AC),x_plot_AC,label='u_AC ODE')
    ax.plot(u_BC_func(x_plot_BC),x_plot_BC, label='u_BC ODE')
    ax.plot(u_trial_func(x_plot),x_plot, label='u trial')
    ax.spines['right'].set_color('none')
    ax.spines['top'].set_color('none')
    ax.spines['bottom'].set_position('zero')
    ax.spines['left'].set_position('zero')
    ax.set_xlabel('u(x)')
    ax.set_ylabel('x')
    ax.legend(loc='best')
    fig.patch.set_alpha(0.0)
    ax.patch.set_alpha(0.0)

interact(func, L_test = widgets.FloatSlider(min=0.1, max=5, value=2, step=0.1, descriptio
         EA_test = widgets.FloatSlider(min=1000, max=10000, value=2000, step=100, descrip
         F_test = widgets.FloatSlider(min=10, max=100, value=30, step=1, description="For
```

```
%matplotlib widget
fig, ax = plt.subplots()
fig.canvas.toolbar_visible = False

def func(L_test, EA_test, F_test):
    ax.clear()
    N_trial_func_AC = sym.lambdify(x, N_trial_sol_AC.subs(L,L_test).subs(EA,EA_test).subs
    N_trial_func_BC = sym.lambdify(x, N_trial_sol_BC.subs(L,L_test).subs(EA,EA_test).subs
    N_AC_func = sym.lambdify(x, N_AC_sol.subs(L,L_test).subs(EA,EA_test).subs(F,F_test))
    N_BC_func = sym.lambdify(x, N_BC_sol.subs(L,L_test).subs(EA,EA_test).subs(F,F_test))
    x_plot = np.linspace(0, L_test*2,2)
    x_plot_AC = np.linspace(0, L_test,2)
    x_plot_BC = np.linspace(L_test, L_test*2,2)
    ax.plot(N_AC_func(x_plot_AC)*np.ones(2),x_plot_AC,label='u_AC ODE')
    ax.plot(N_BC_func(x_plot_BC)*np.ones(2),x_plot_BC, label='u_BC ODE')
    ax.plot(N_trial_func_AC(x_plot_AC),x_plot_AC, label='u_AC trial')
    ax.plot(N_trial_func_BC(x_plot_BC),x_plot_BC, label='u_BC trial')
    ax.spines['right'].set_color('none')
    ax.spines['top'].set_color('none')
    ax.spines['bottom'].set_position('zero')
    ax.spines['left'].set_position('zero')
    ax.set_xlabel('N(x)')
    ax.set_ylabel('x')
    ax.legend(loc='best')
    fig.patch.set_alpha(0.0)
    ax.patch.set_alpha(0.0)

interact(func, L_test = widgets.FloatSlider(min=0.1, max=5, value=2, step=0.1, descriptio
         EA_test = widgets.FloatSlider(min=1000, max=10000, value=2000, step=100, descrip
         F_test = widgets.FloatSlider(min=10, max=100, value=30, step=1, description="For
```

Exercise (Comparison)

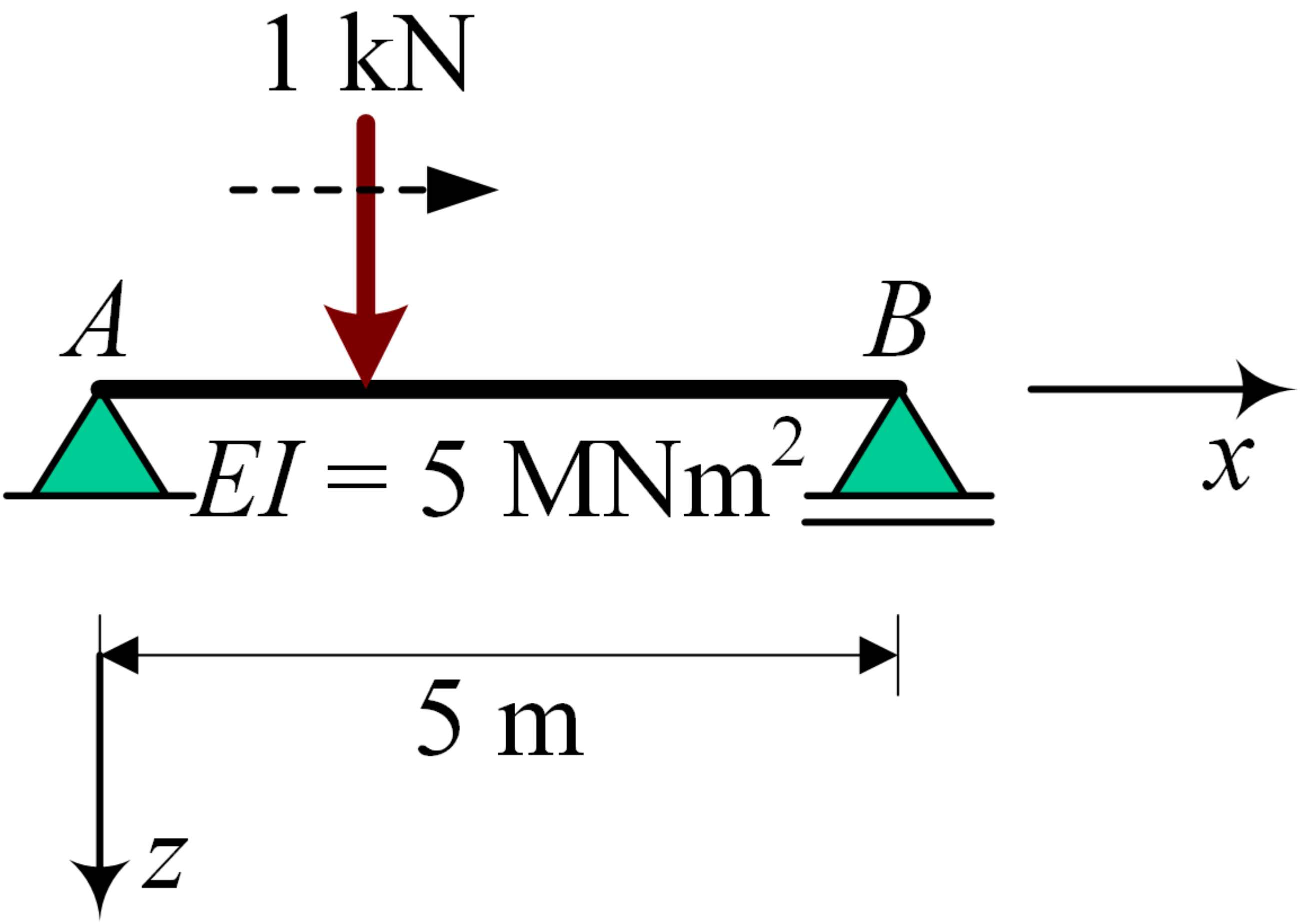
Would you consider the trial function solution for proper estimates for displacements? How about normal forces?

Solution to Exercise (Comparison)

The estimation for displacement seems okayish, but the difference with the ‘real’ solution for normal force distribution is much bigger so it might not be advised to use that.

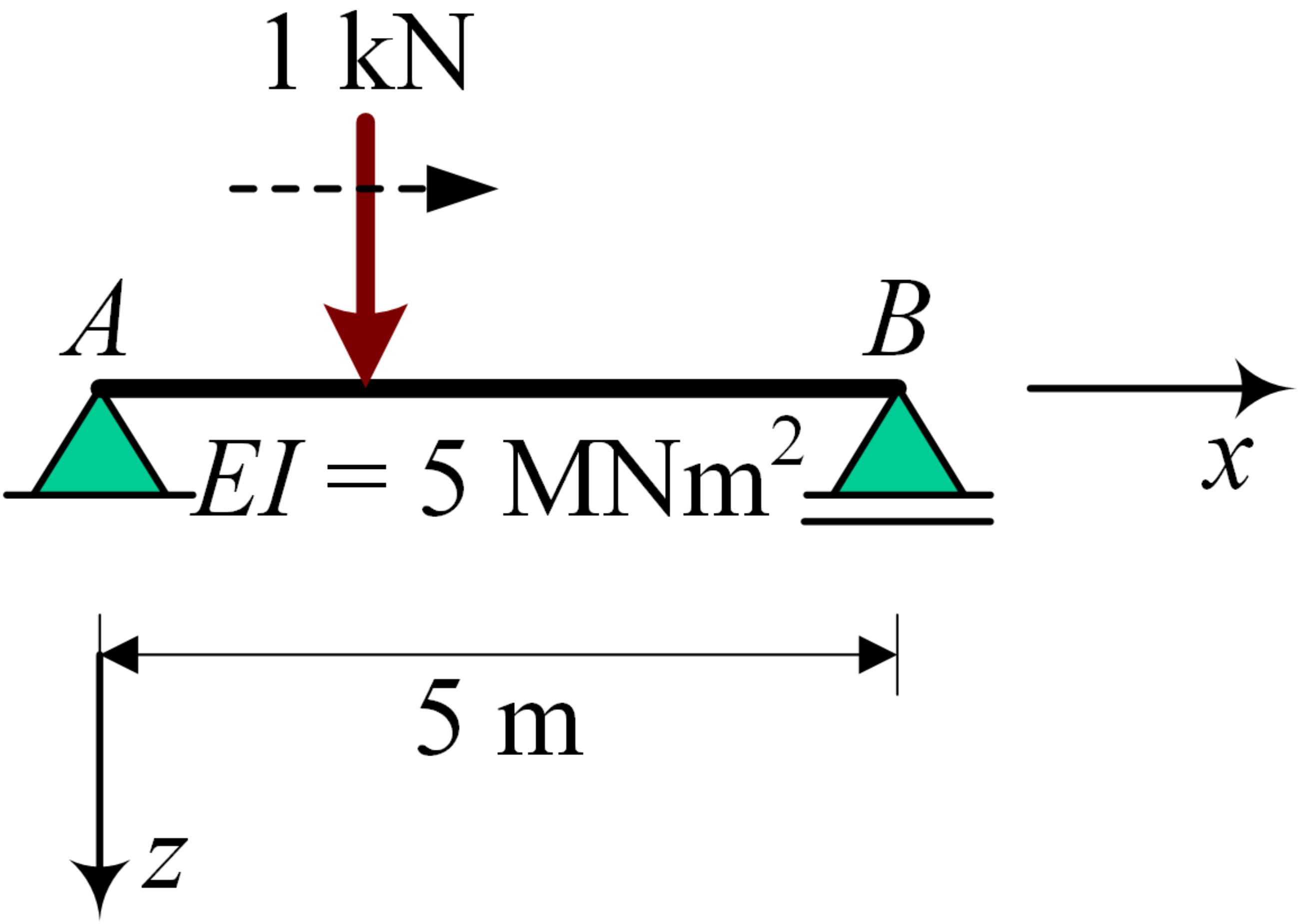
Simply supported beam

The influence lines of the following structure will be investigated



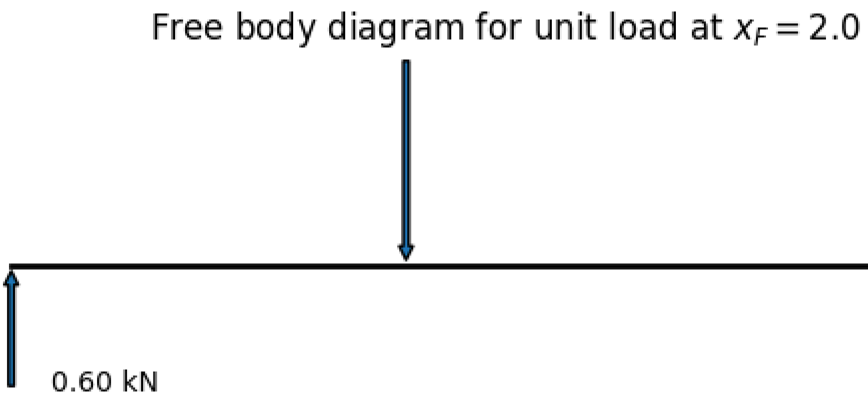
Support reaction A

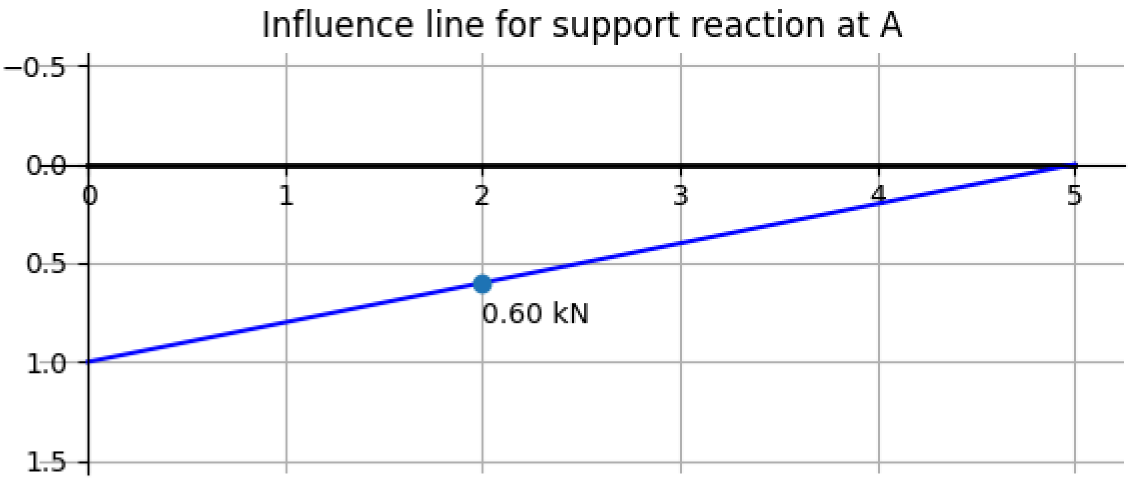
Click 🚀 –> Live Code on the top right corner of this screen to investigate some influence lines!



Location unit load x_F = ... (m) 2.0

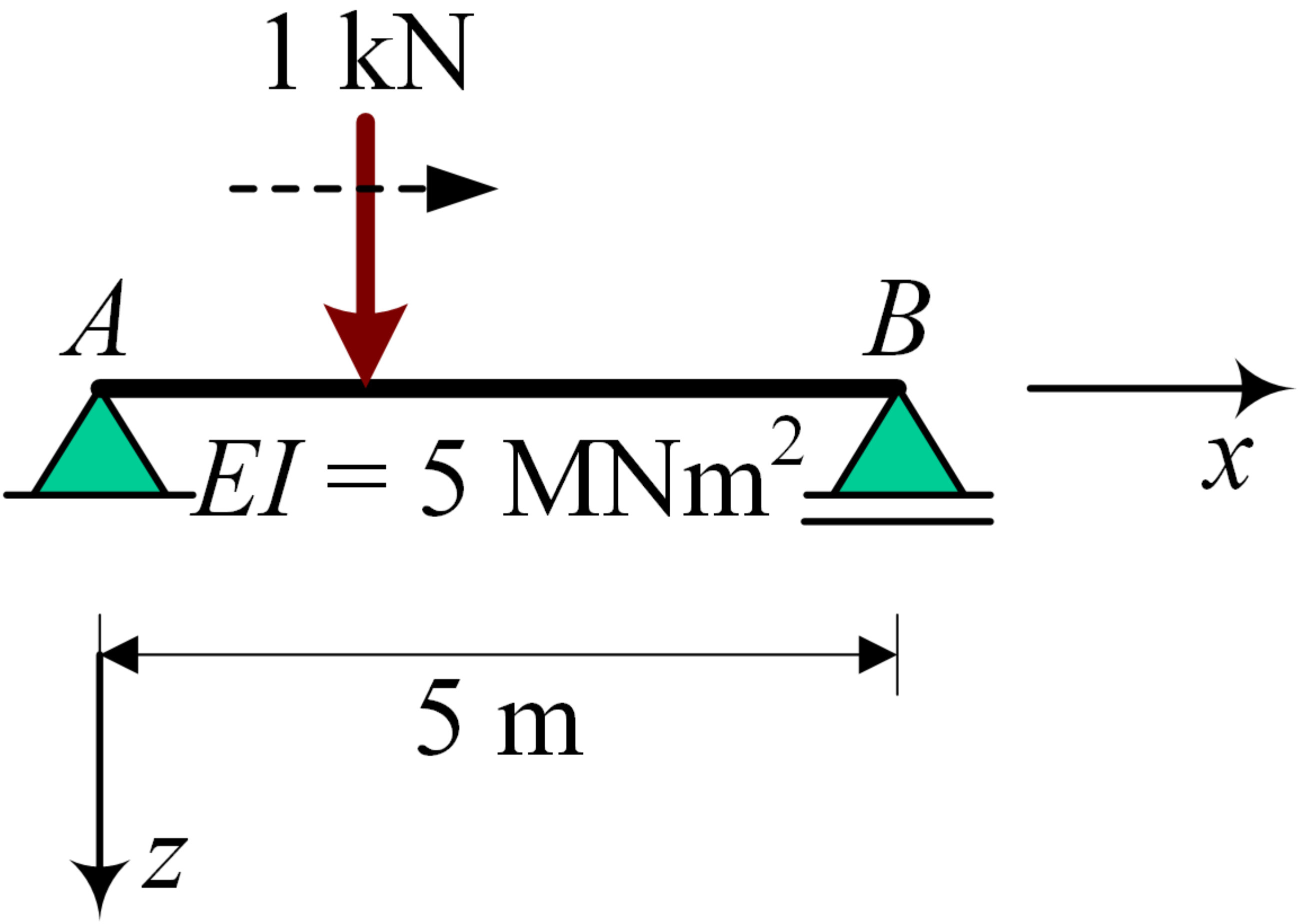
Figure 1





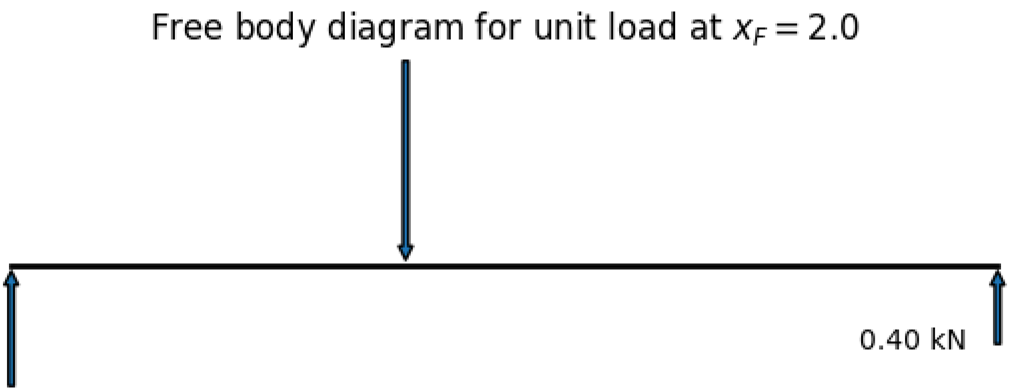
Support reaction B

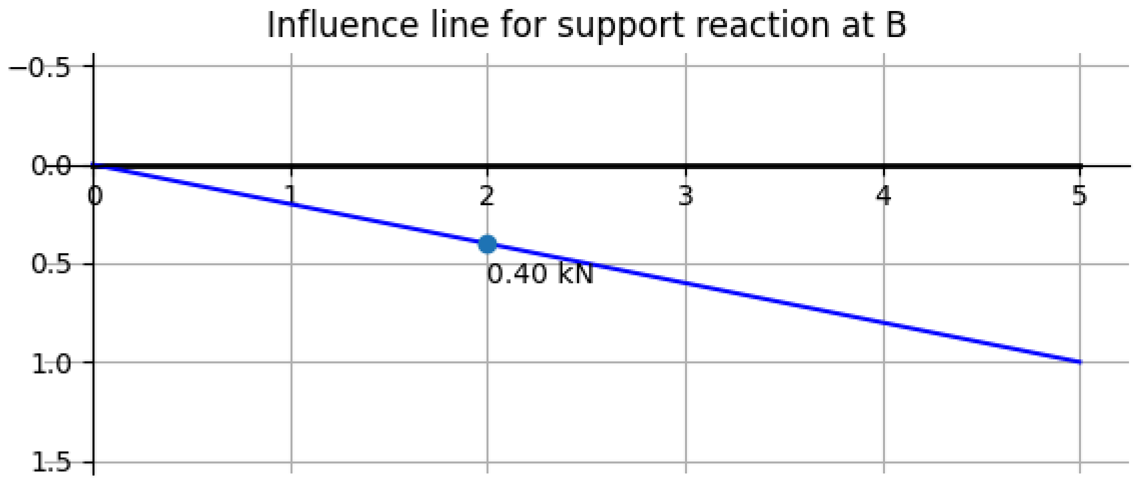
Click 🚀 –> Live Code on the top right corner of this screen to investigate some influence lines!



Location unit load x_F = ... (m) 2.0

Figure 1

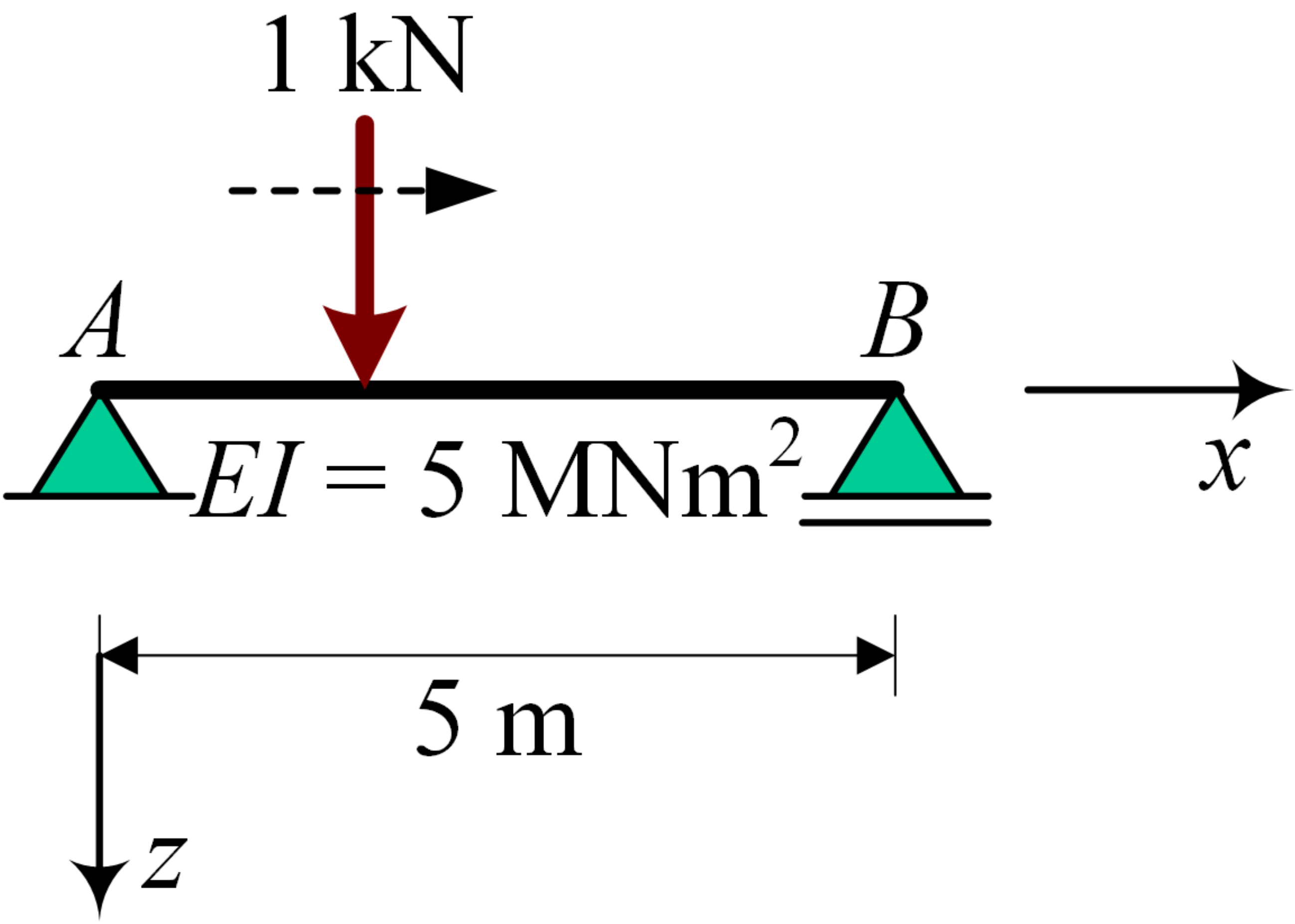




Moments

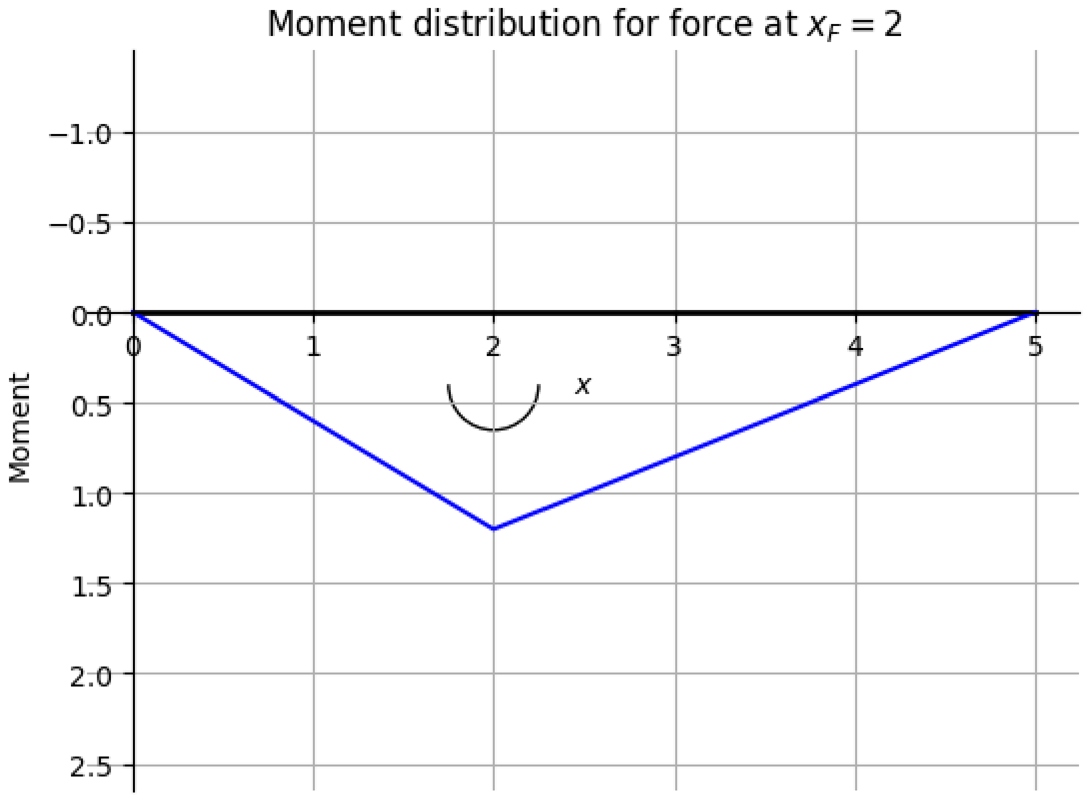
Contents

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some influence lines!

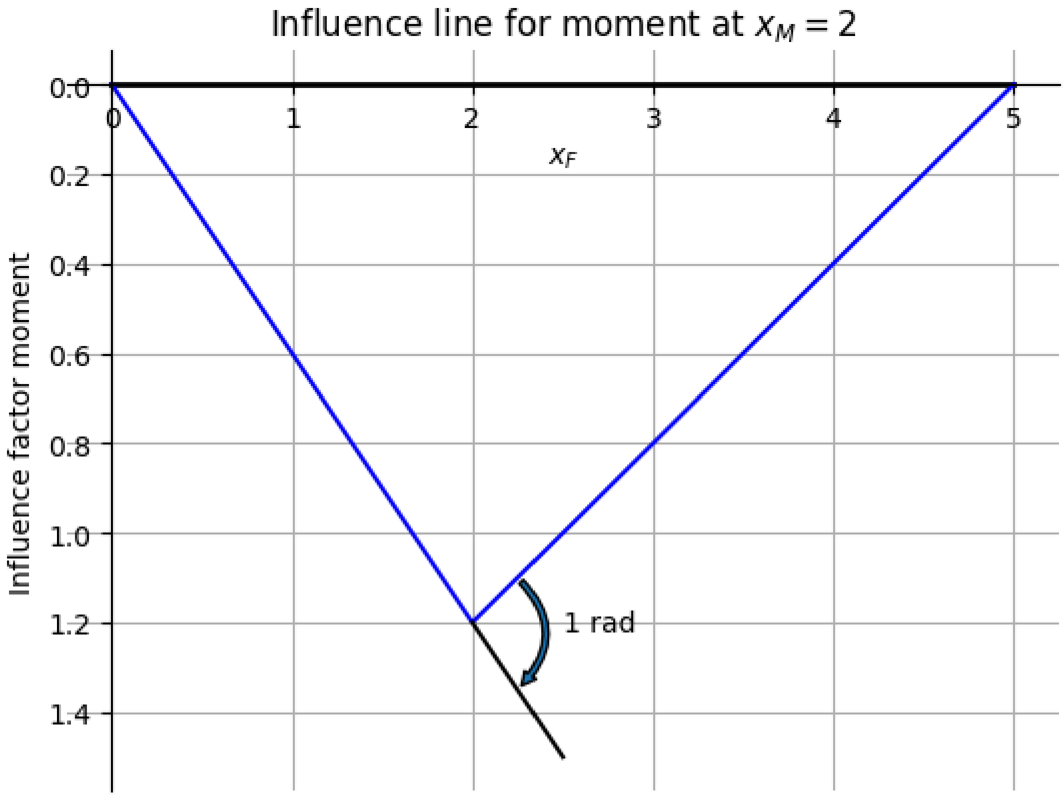


Find M-line

$$\begin{cases} 0 & \text{for } x > 5 \\ 2 - \frac{2x}{5} & \text{for } x > 2 \\ \frac{3x}{5} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Finding influence line for M

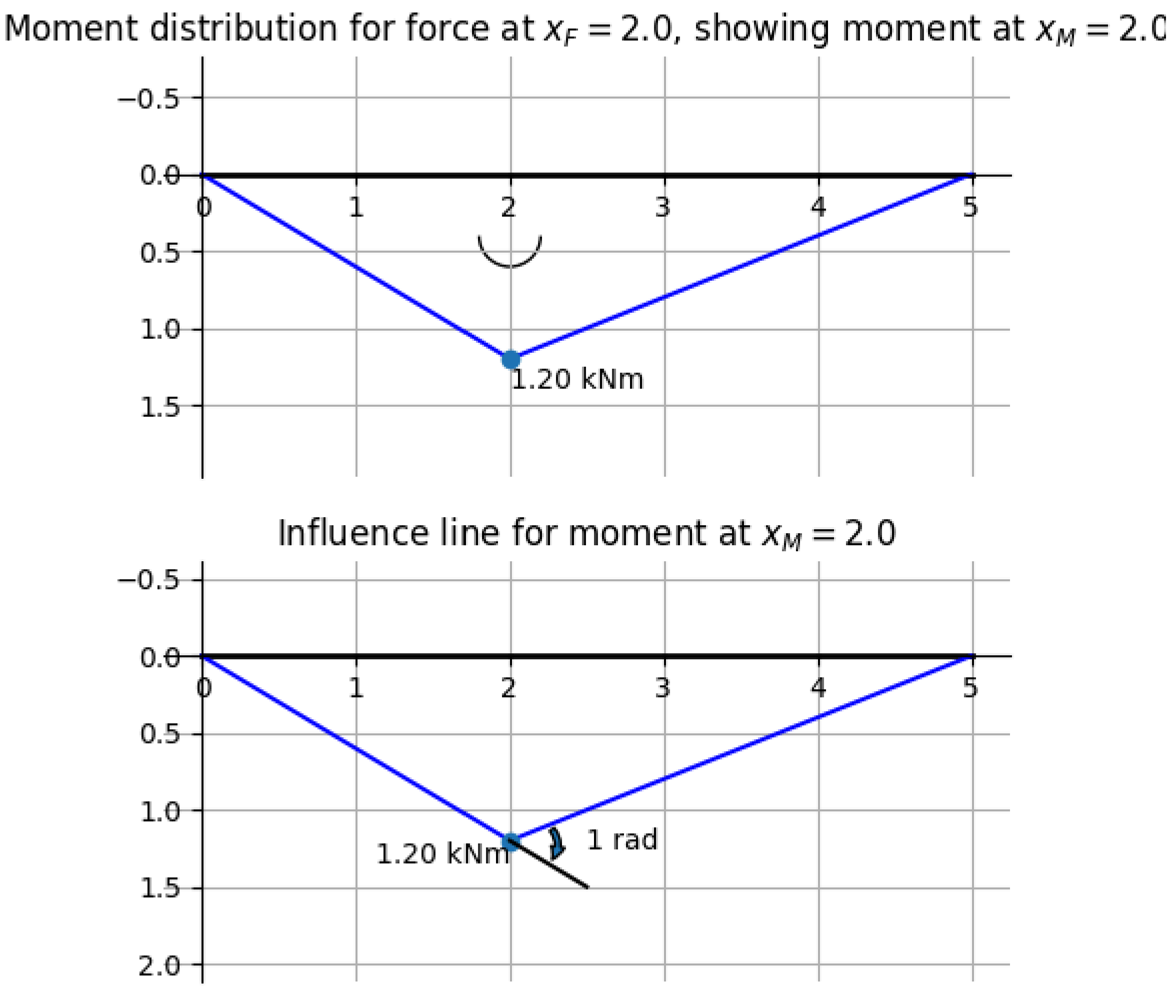


Comparison M-line and influence line for

moment at $x = \ldots$

Location moment x_M = ... (m)	○	2.0
Location unit load x_F = ... (m)	○	2.0

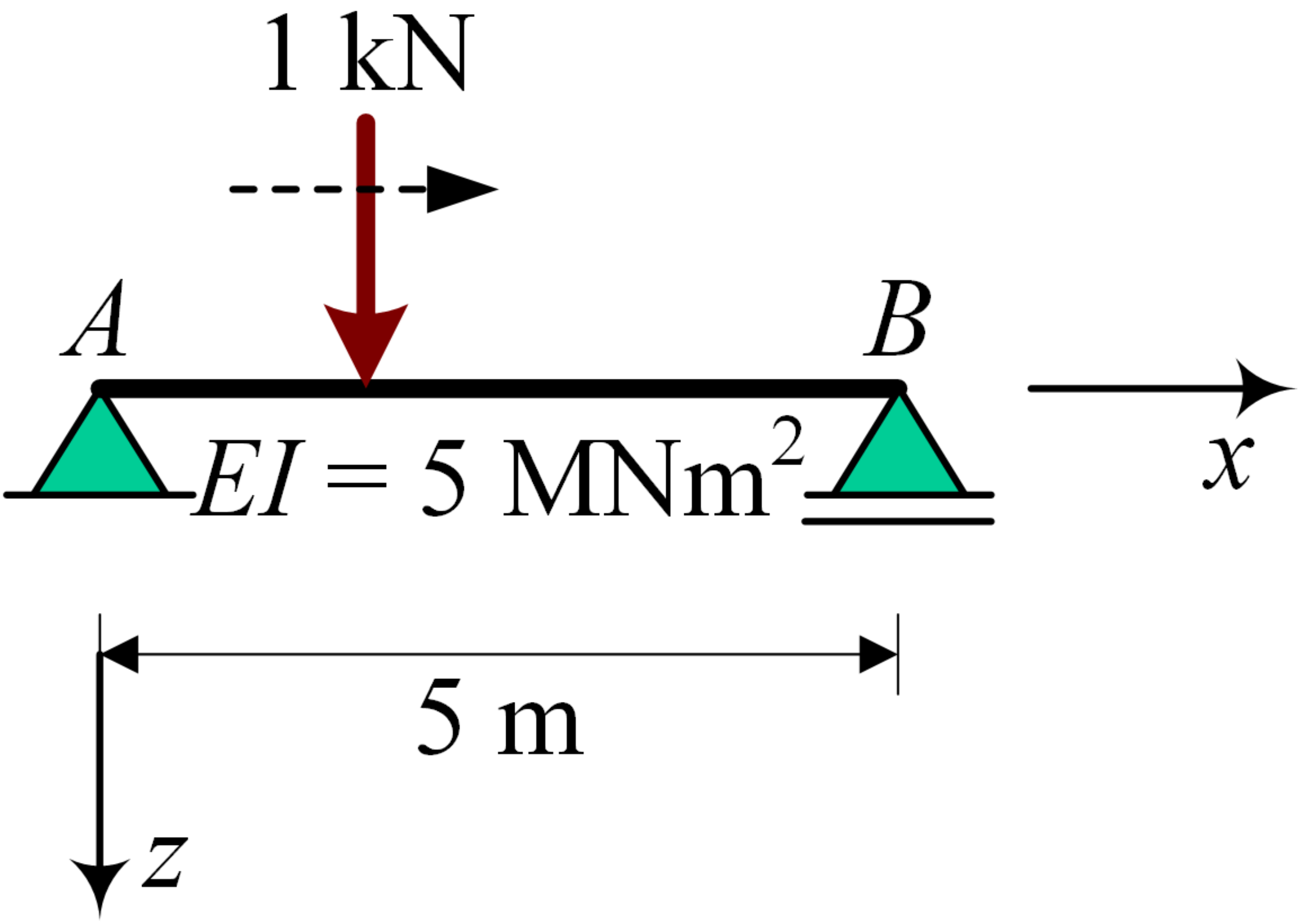
Figure 1



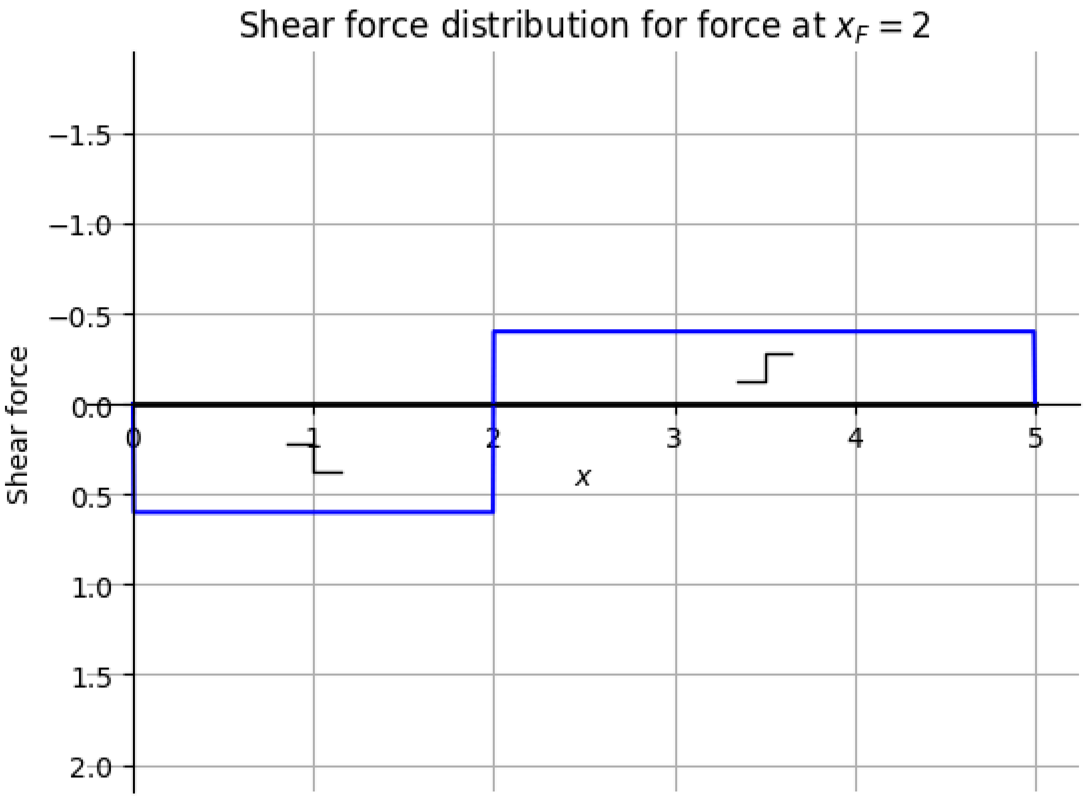
Shear force

Contents

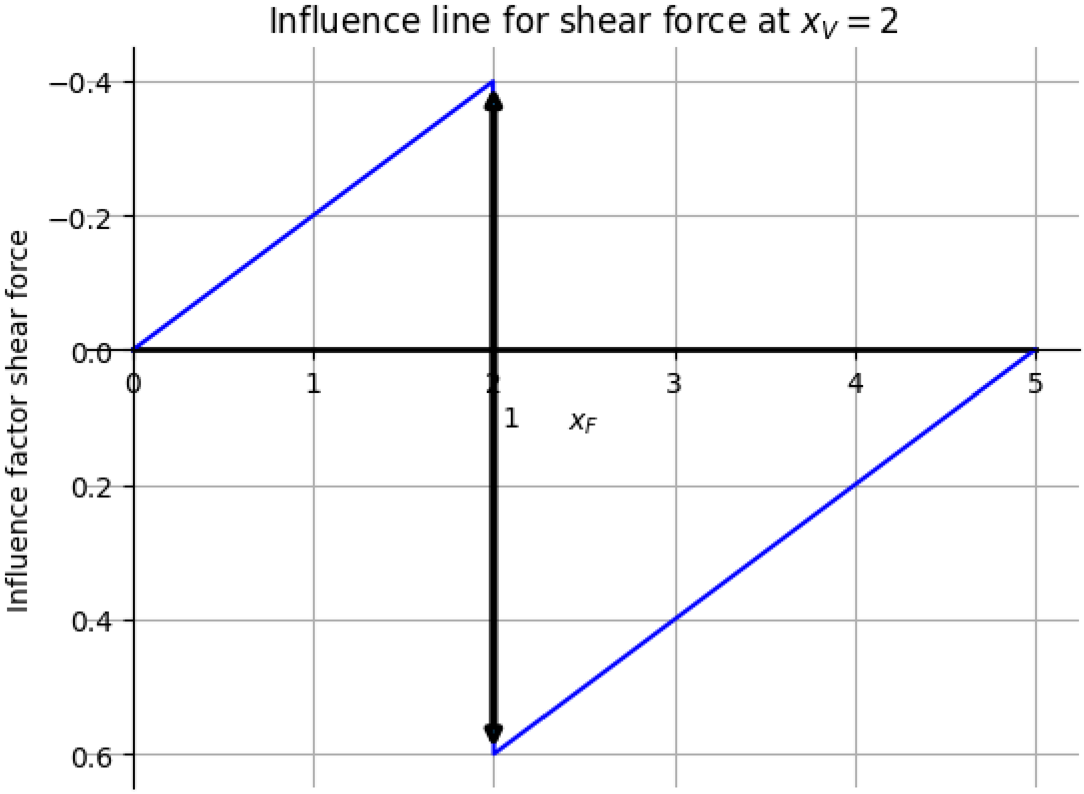
Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some influence lines!



$$\begin{cases} 0 & \text{for } x > 5 \\ -\frac{2}{5} & \text{for } x > 2 \\ \frac{3}{5} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Find influence line V

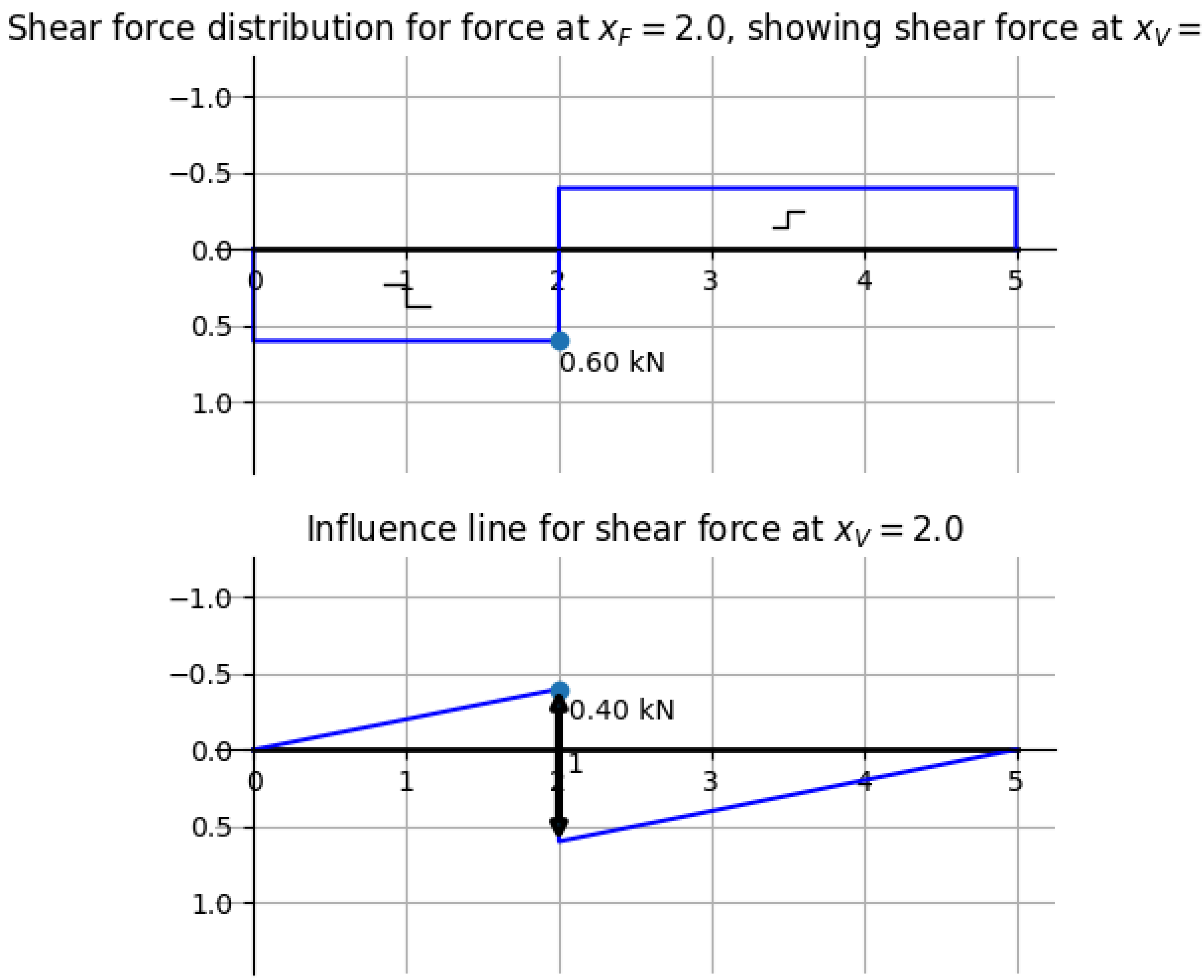


Comparison V-line and influence line for

shear force at $x = ...$

Location shear force x_V = ... (m)	○	2.0
Location force x_F = ... (m)	○	2.0

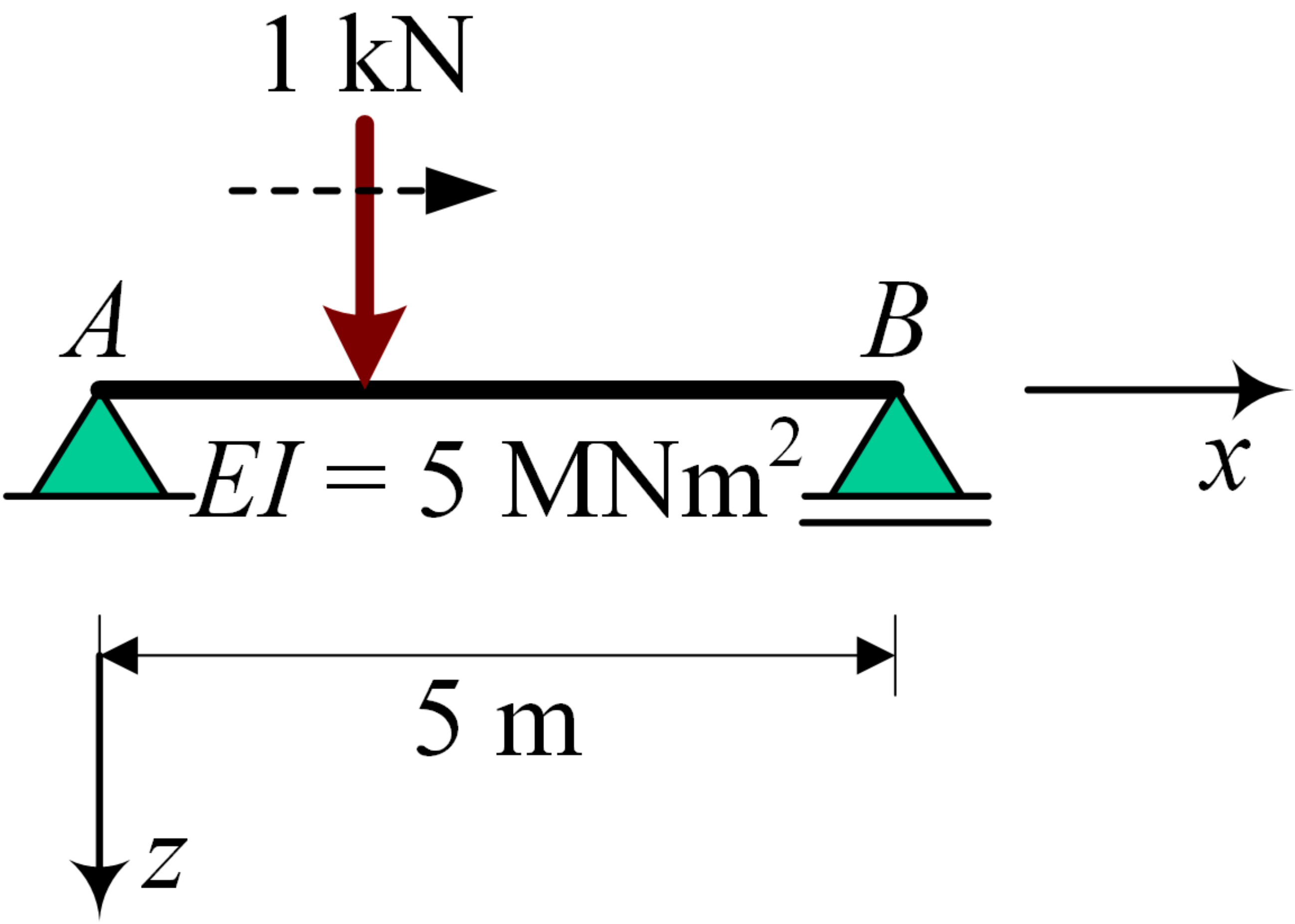
Figure 1



Displacement

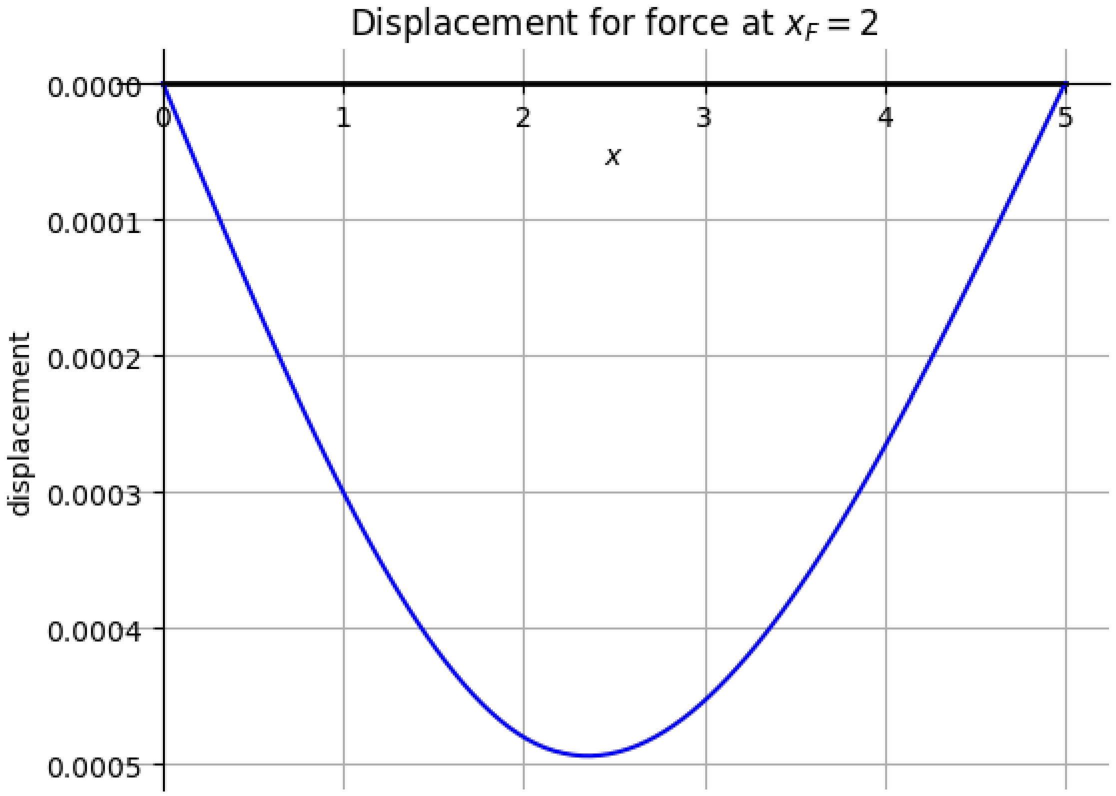
Contents

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some influence lines!



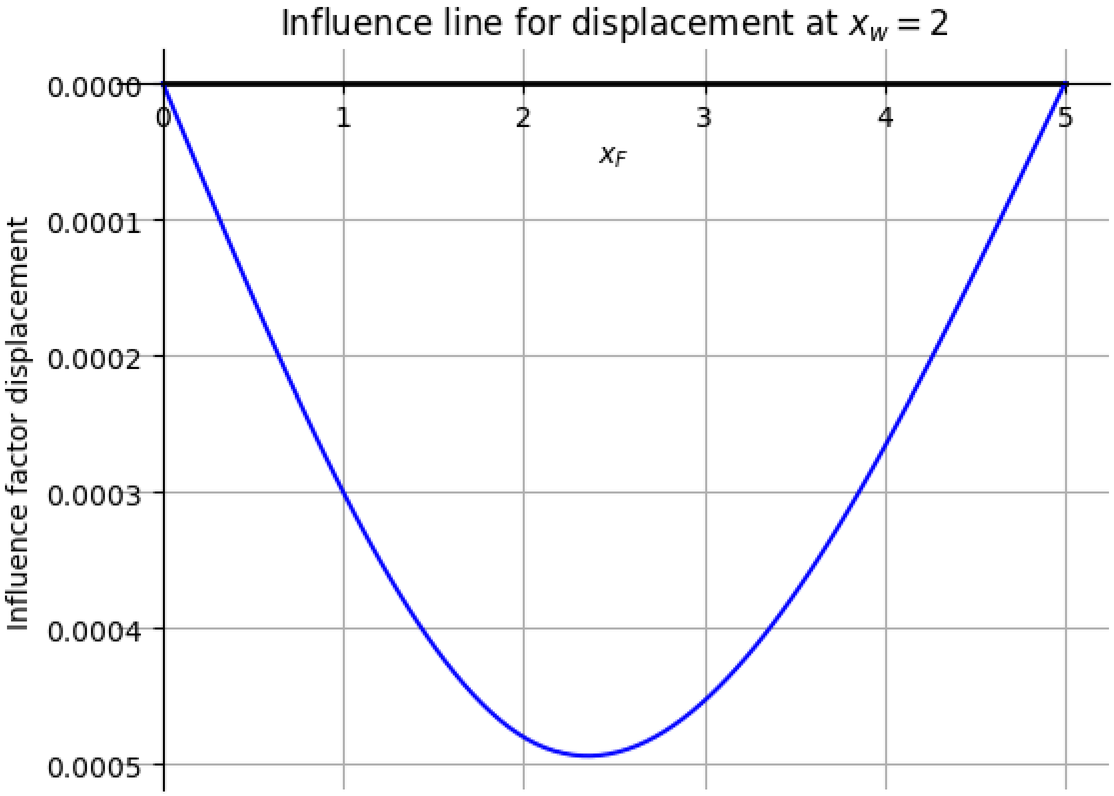
Find displacement w

$$\begin{cases} \frac{7 \cdot (5 - x)}{25000} & \text{for}\ x > 5 \\ \frac{x^{3} - 15 x^{2} + 54 x - 20}{75000} & \text{for}\ x > 2 \\ \frac{x \left(16 - x^{2}\right)}{50000} & \text{for}\ x > 0 \\ \frac{x}{3125} & \text{otherwise} \end{cases}$$



Find influence line w

$$\begin{cases} \frac{7\cdot(5-x)}{25000} & \text{for } x > 5 \\ \frac{x^3-15x^2+54x-20}{75000} & \text{for } x > 2 \\ \frac{x\left(16-x^2\right)}{50000} & \text{for } x > 0 \\ \frac{x}{3125} & \text{otherwise} \end{cases}$$



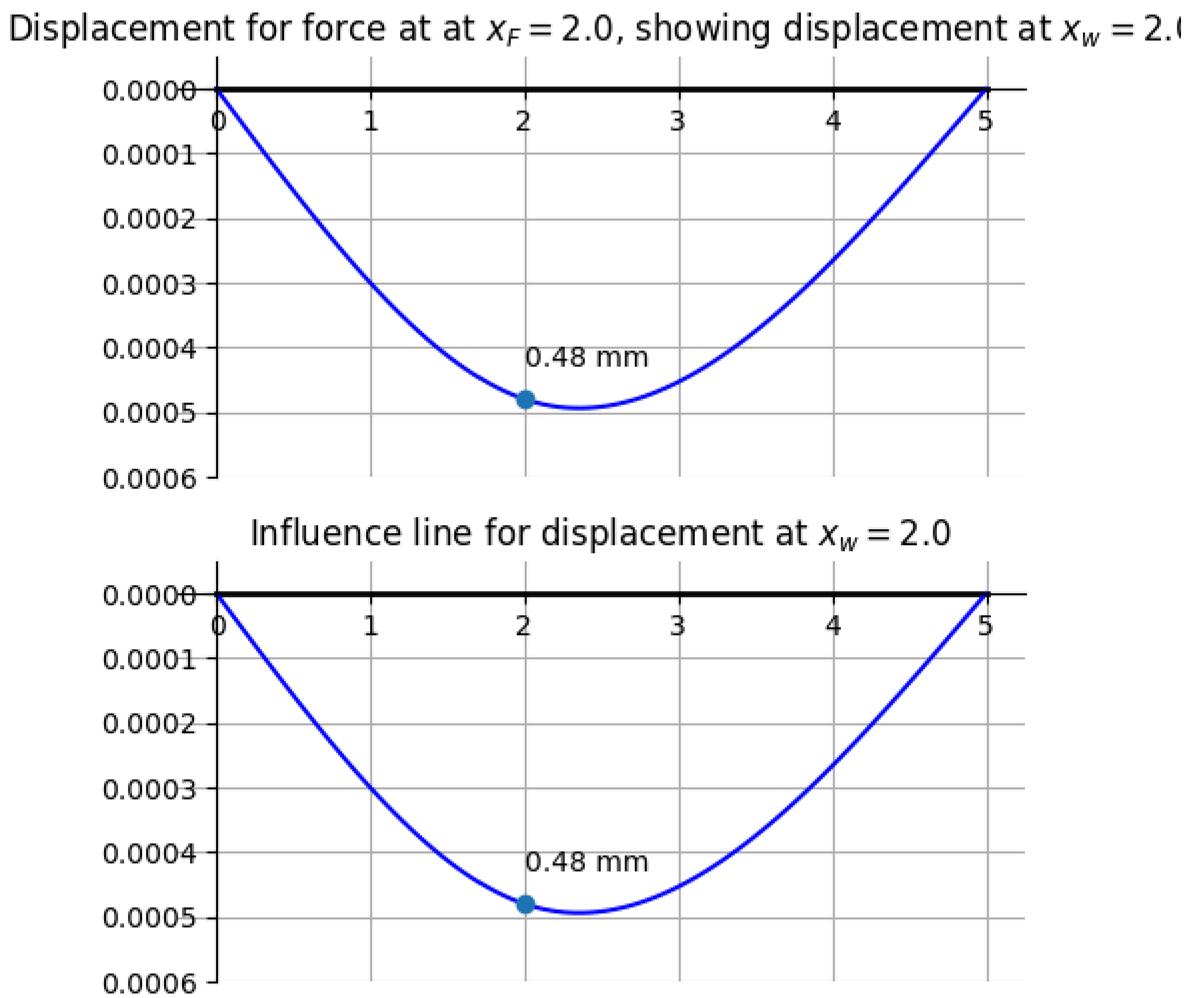
Comparison displacement and influence

line for displacement at $x =$...

Location displacement x_w = ... (m) 2.0

Location force x_F = ... (m) 2.0

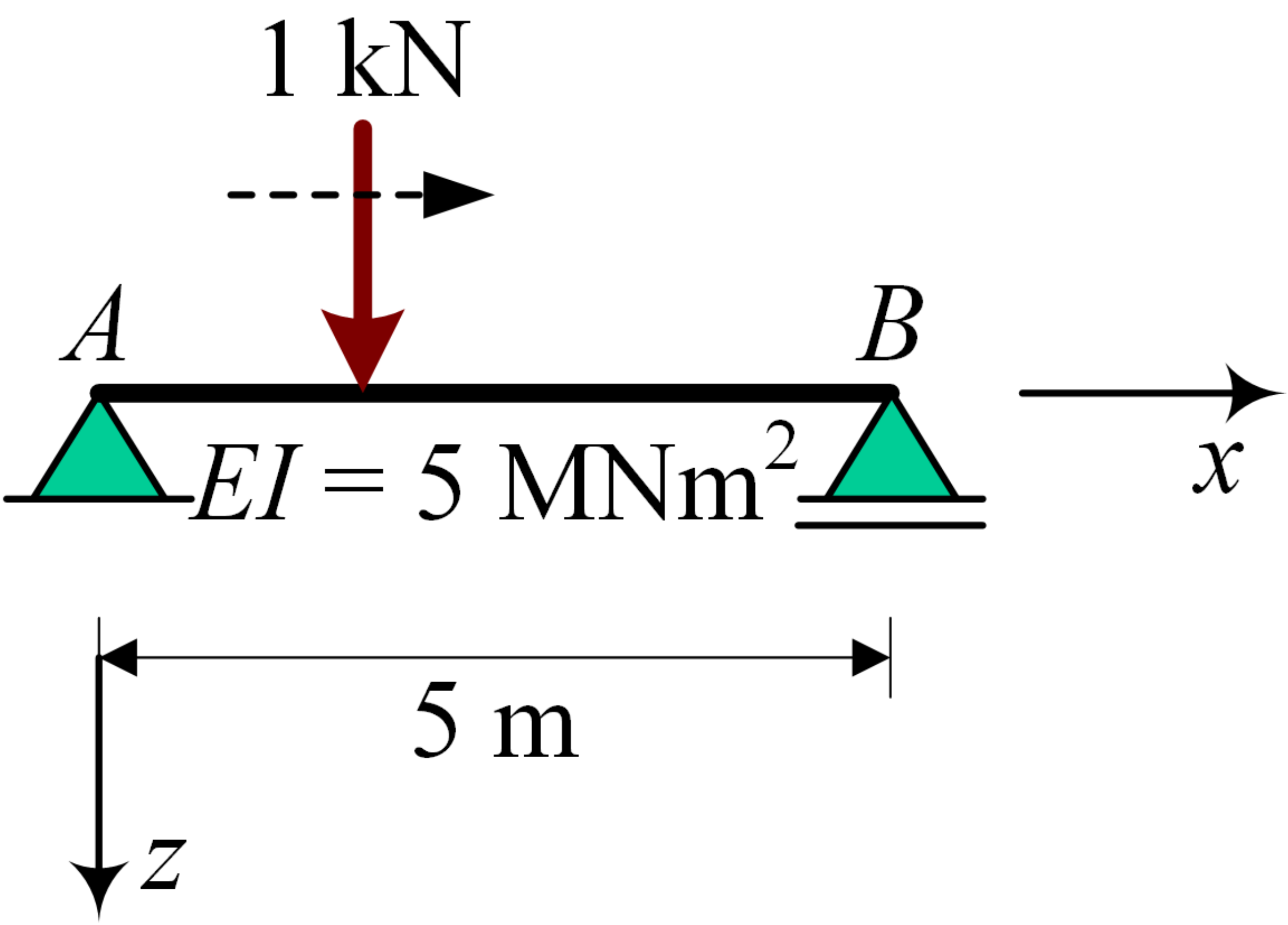
Figure 1



Rotation

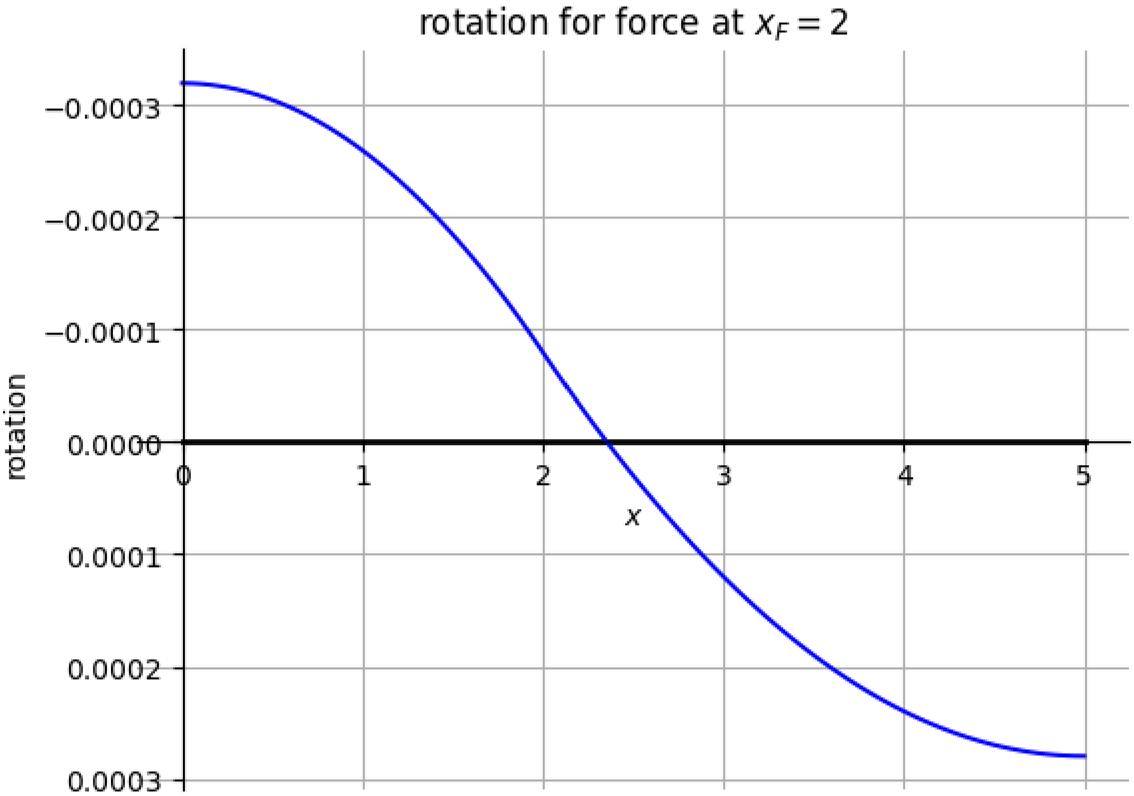
Contents

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some influence lines!



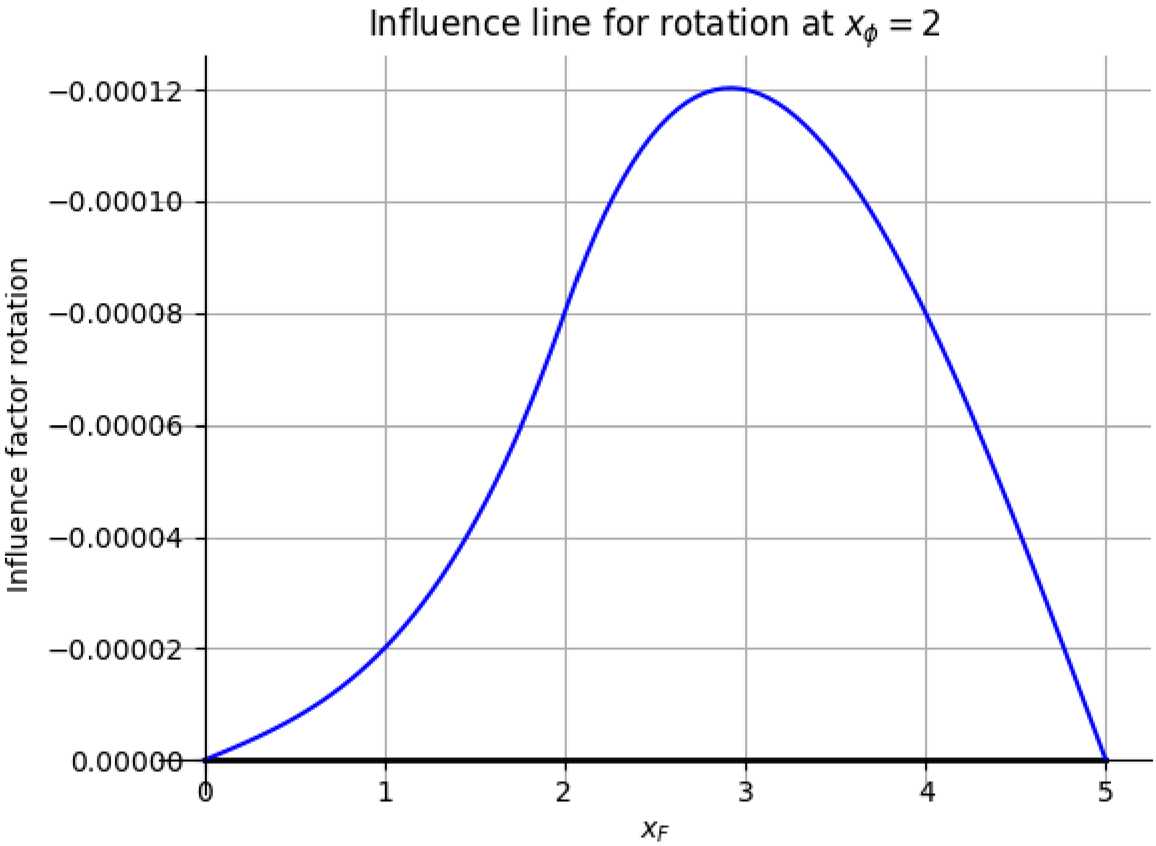
Find rotation line ϕ

$$\begin{cases} \frac{7}{25000} & \text{for}\ x > 5 \\ \frac{-x^2+10x-18}{25000} & \text{for}\ x > 2 \\ \frac{3x^2-16}{50000} & \text{for}\ x > 0 \\ -\frac{1}{3125} & \text{otherwise} \end{cases}$$



Find influence line ϕ

$$\begin{cases} \frac{13(x-5)}{150000} & \text{for } x > 5 \\ \frac{-x^3-2x+15(x-2)^2}{150000} & \text{for } x > 2 \\ -\frac{x\left(x^2+2\right)}{150000} & \text{for } x > 0 \\ -\frac{x}{75000} & \text{otherwise} \end{cases}$$



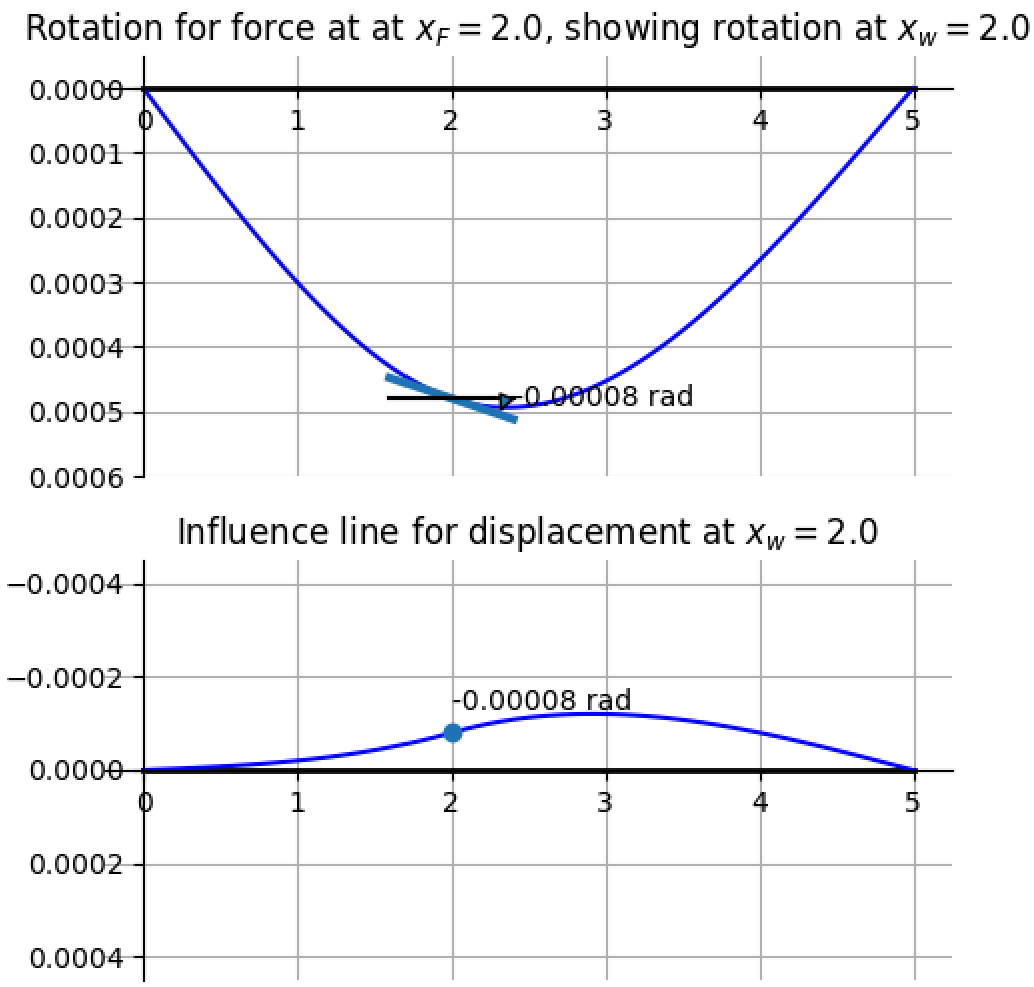
Comparison displacement with ϕ and

influence line for ϕ at $x = \ldots$

Location rotation x_phi = ... (m) 2.0

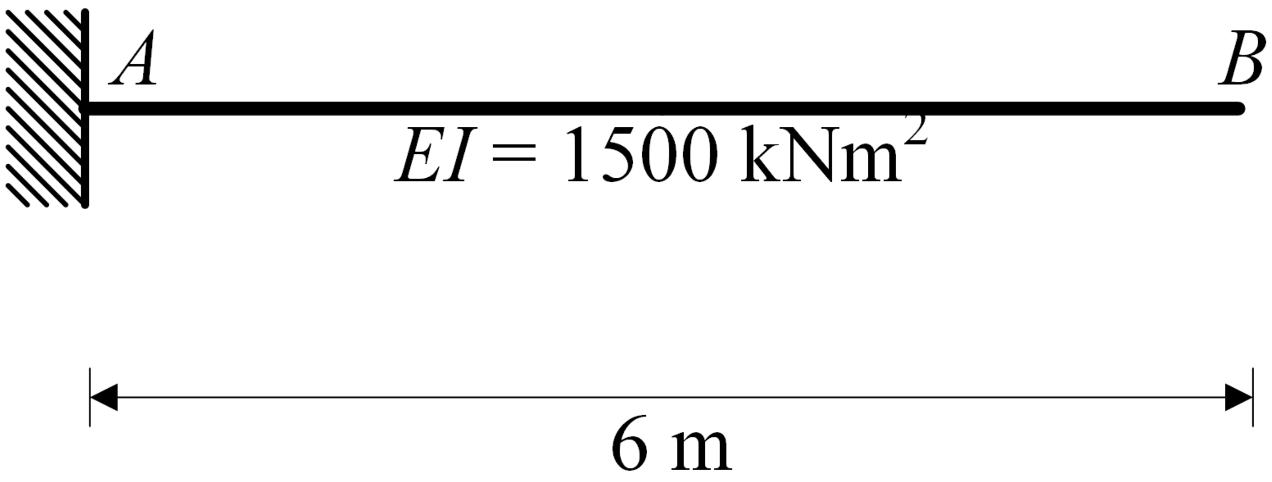
Location force x_F = ... (m) 2.0

Figure 1



Cantilever beam

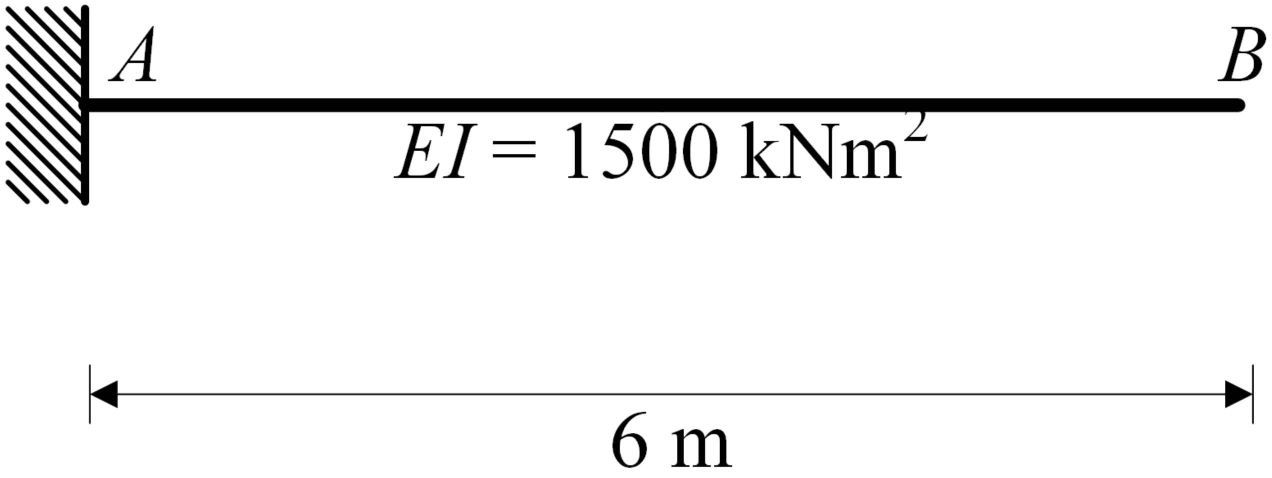
The influence lines of the following structure will be investigated



Moments

Contents

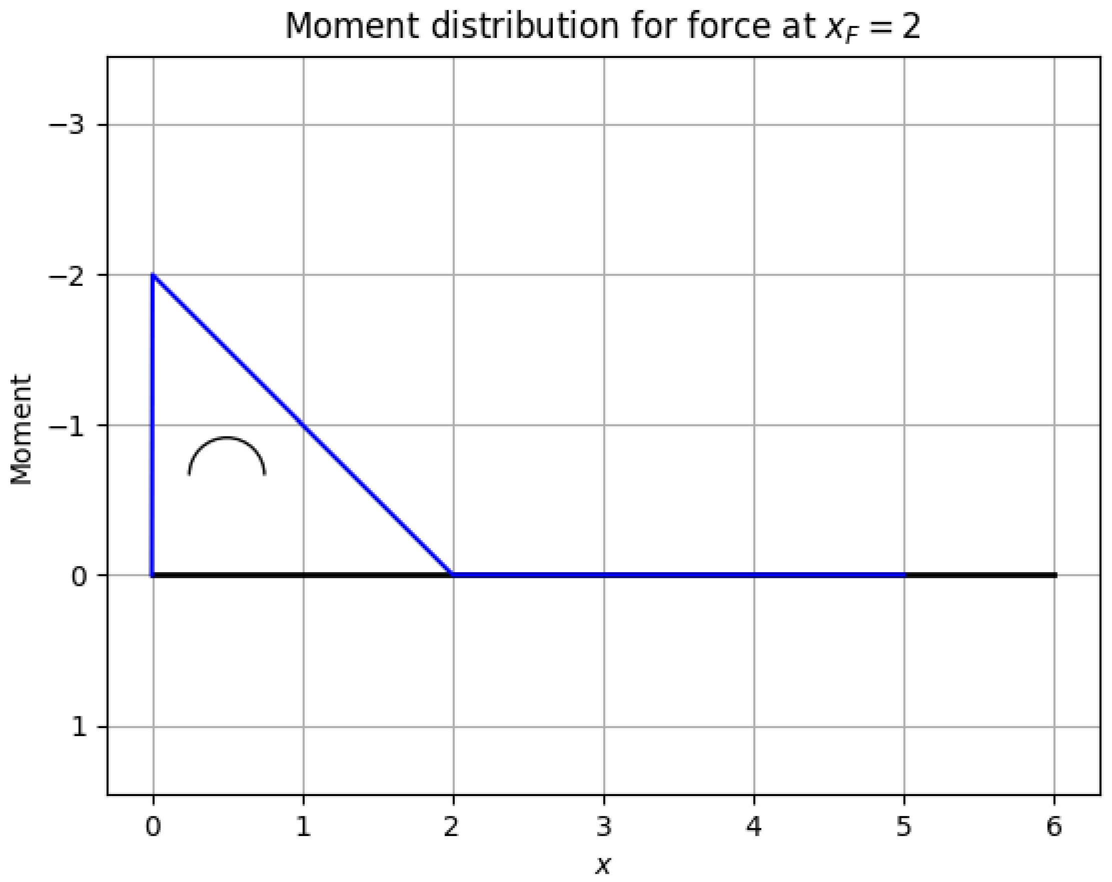
Click 🚀 –> Live Code on the top right corner of this screen to investigate some influence lines!



Find M-line

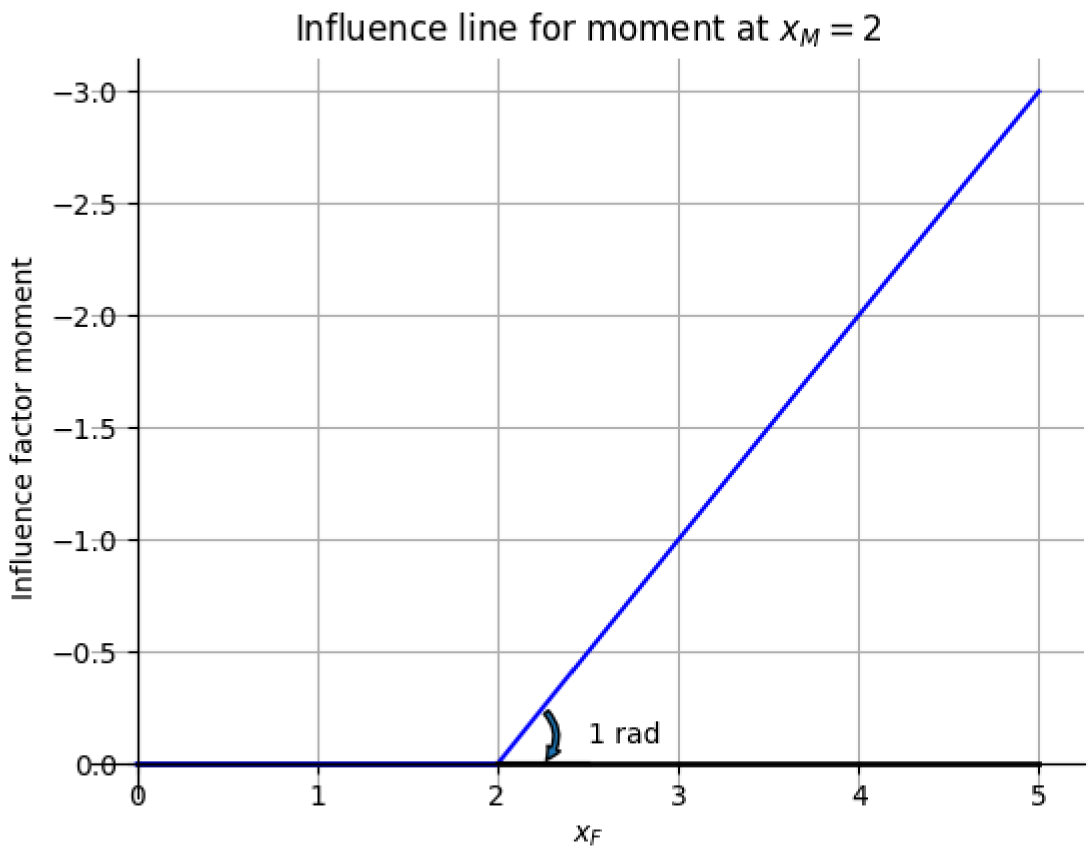
$$\begin{cases} 0 & \text{for } x > 2 \\ x - 2 & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Figure 1



Finding influence line for M

Figure 2



x=3.84 y=−2.913

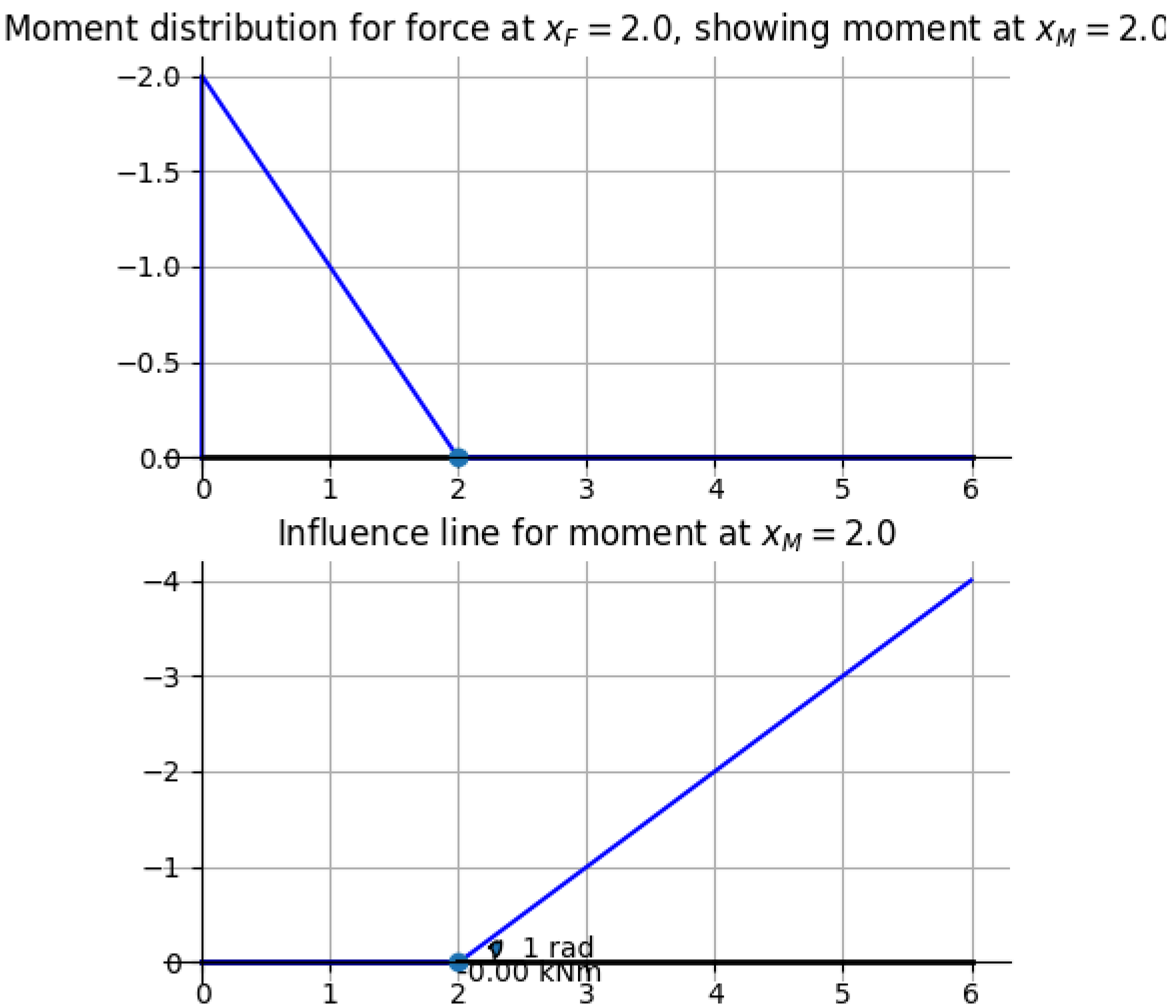
Comparison M-line and influence line for

moment at $x = \ldots$

Location moment x_M = ... (m) 2.0

Location unit load x_F = ... (m) 2.0

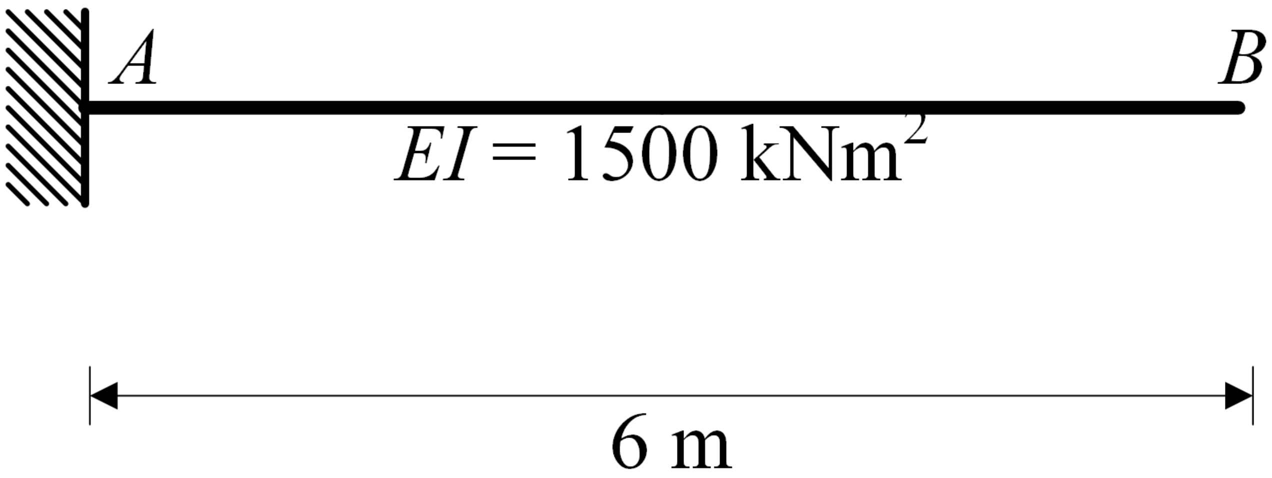
Figure 3



Shear force

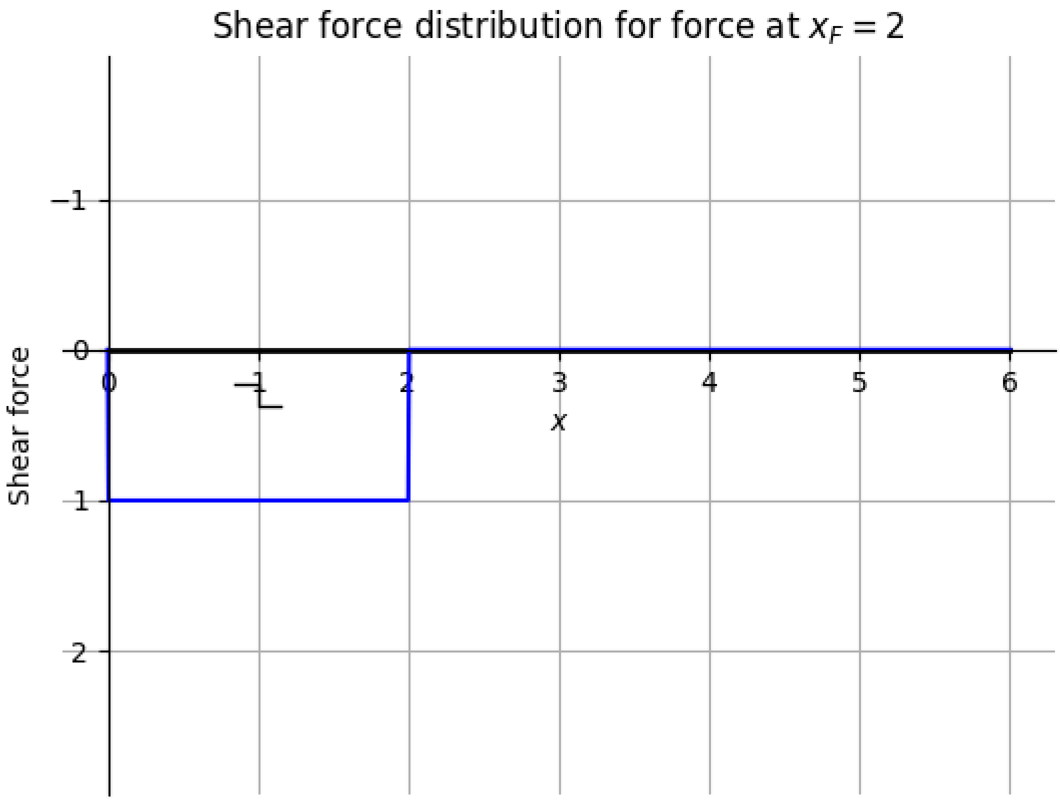
Contents

Click 🚀 –> Live Code on the top right corner of this screen to investigate some influence lines!



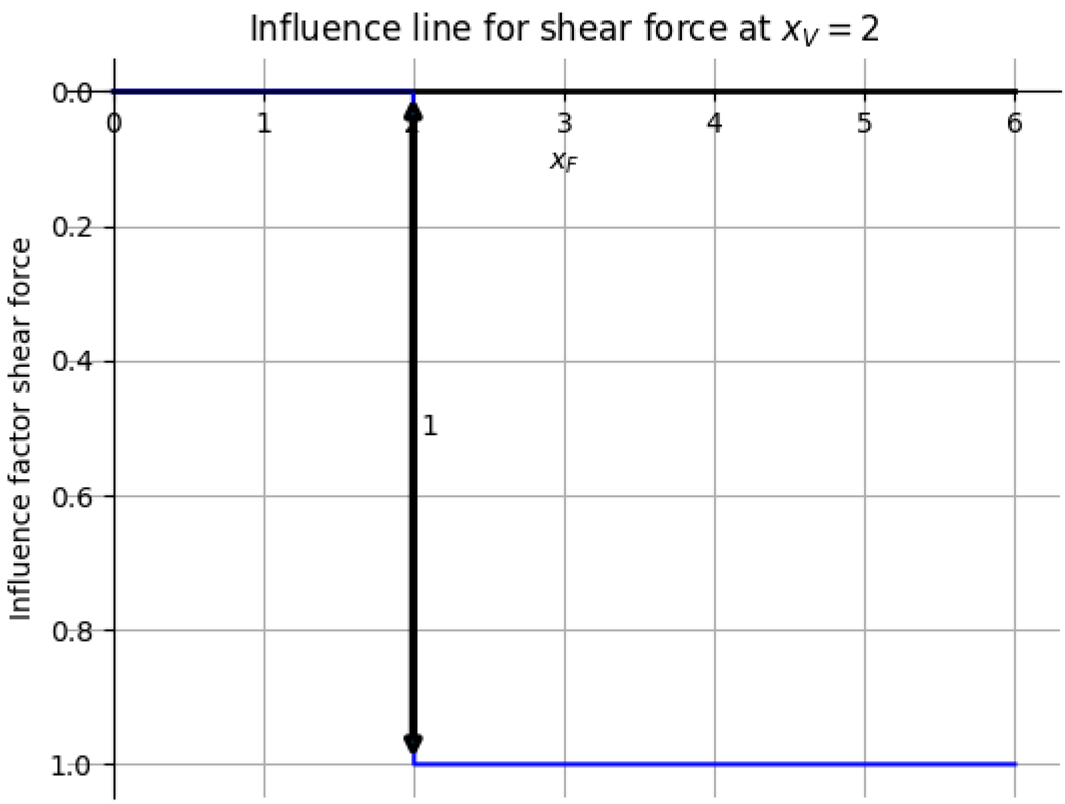
$$\begin{cases} 0 & \text{for}\: x > 2 \\ -\infty & \text{for}\: x = 0 \\ 1 & \text{for}\: x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Figure 1



Find influence line V

Figure 2

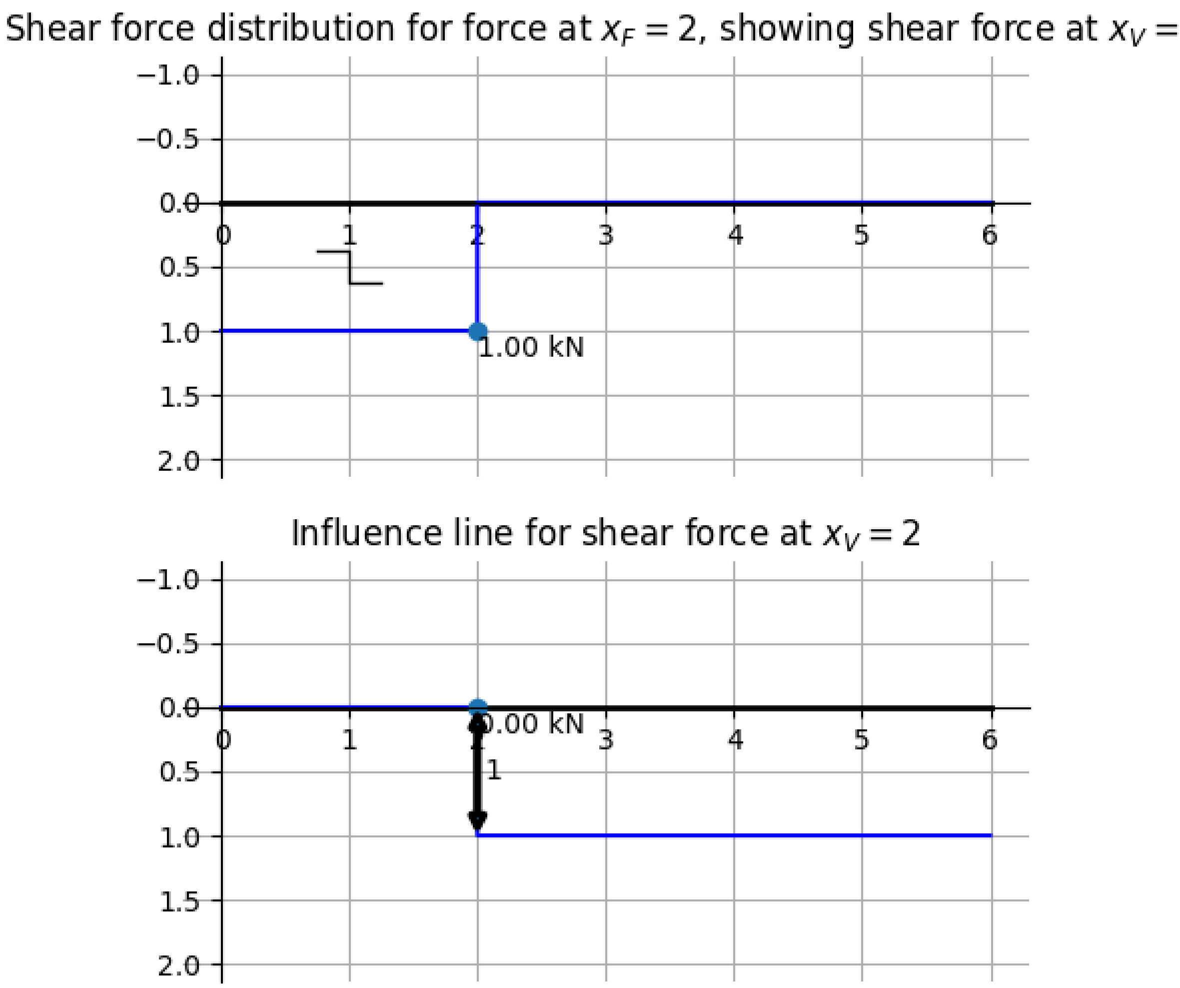


Comparison V-line and influence line for

shear force at $x = \ldots$

Location shear force x_V = ... (m)	○	2.0
Location force x_F = ... (m)	○	2.0

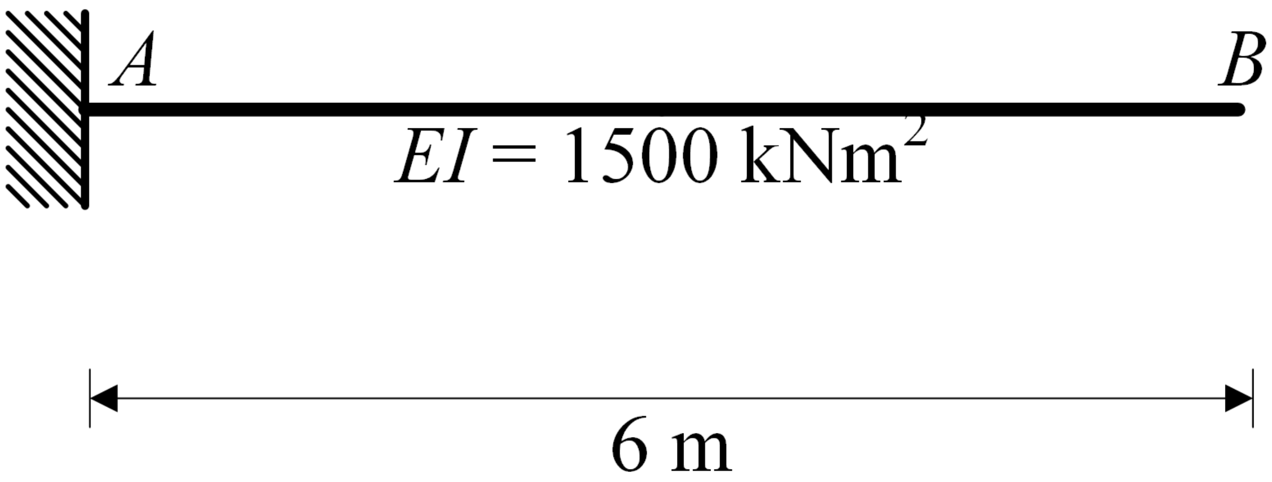
Figure 3



Displacement

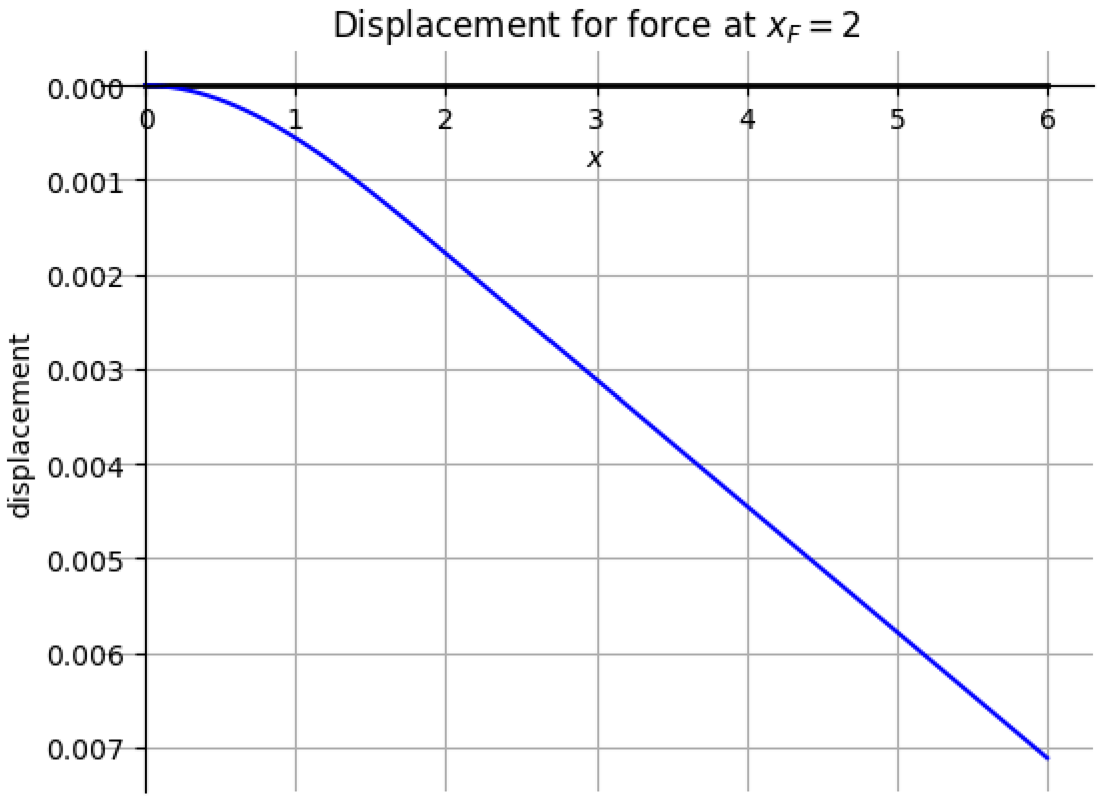
Contents

Click 🚀 –> Live Code on the top right corner of this screen to investigate some influence lines!



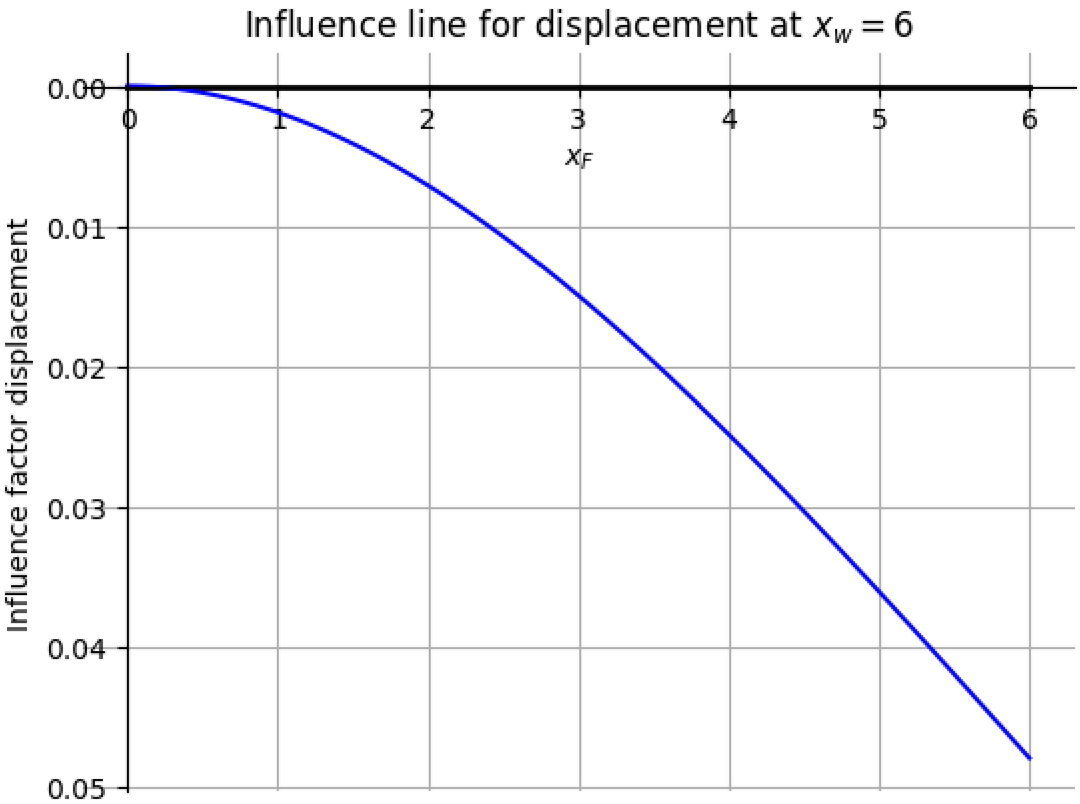
Find displacement w

$$\begin{cases} \frac{x}{750} - \frac{1}{1125} & \text{for } x > 2 \\ \frac{x^2 \cdot (6-x)}{9000} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Find influence line w

$$\begin{cases} \frac{x}{750} - \frac{1}{1125} & \text{for } x > 2 \\ \frac{x^2 \cdot (6-x)}{9000} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



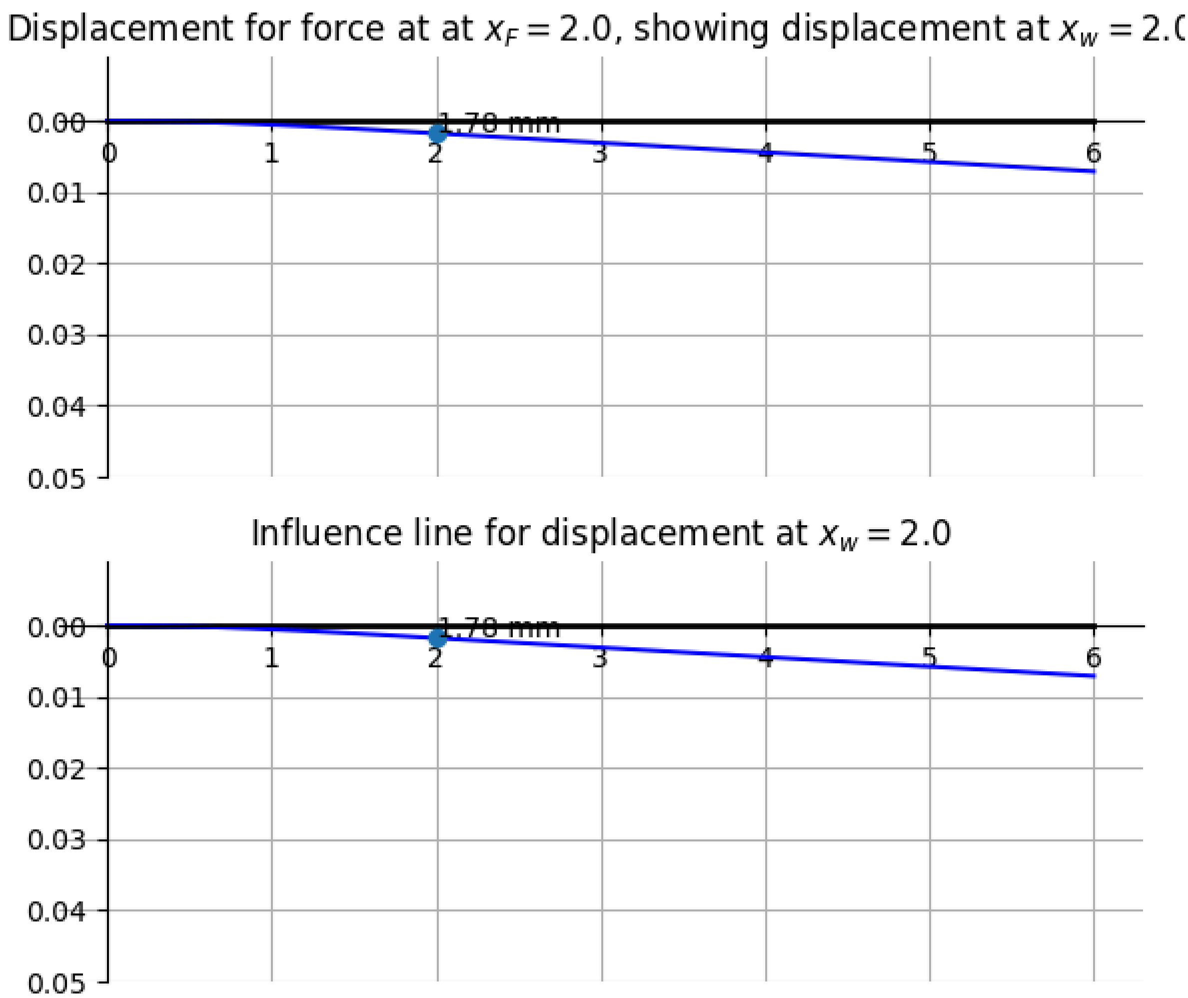
Comparison displacement and influence

line for displacement at $x = ...$

Location displacement x_w = ... (m) 2.0

Location force x_F = ... (m) 2.0

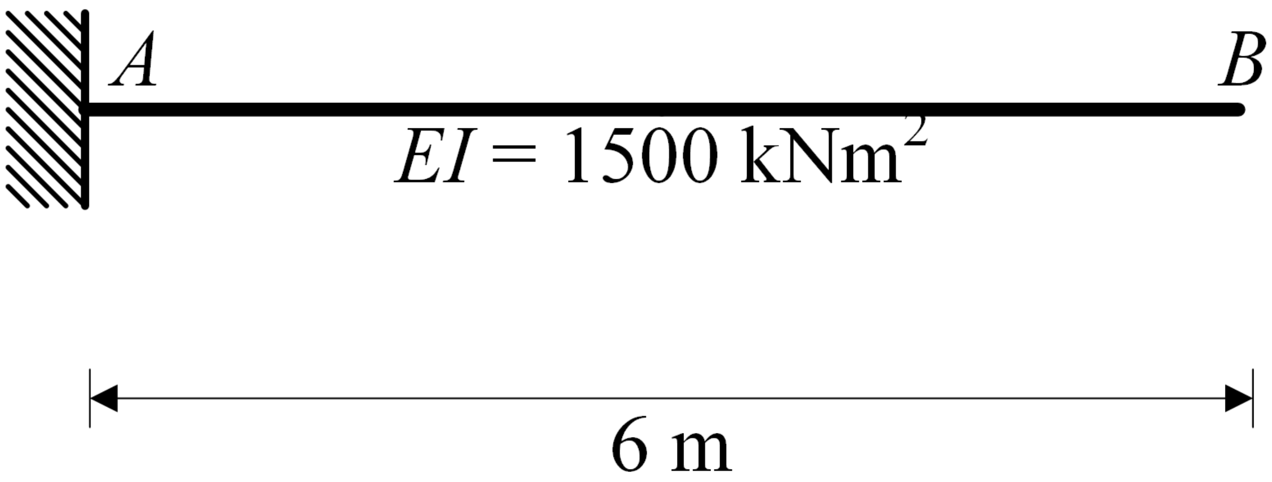
Figure 1



Rotation

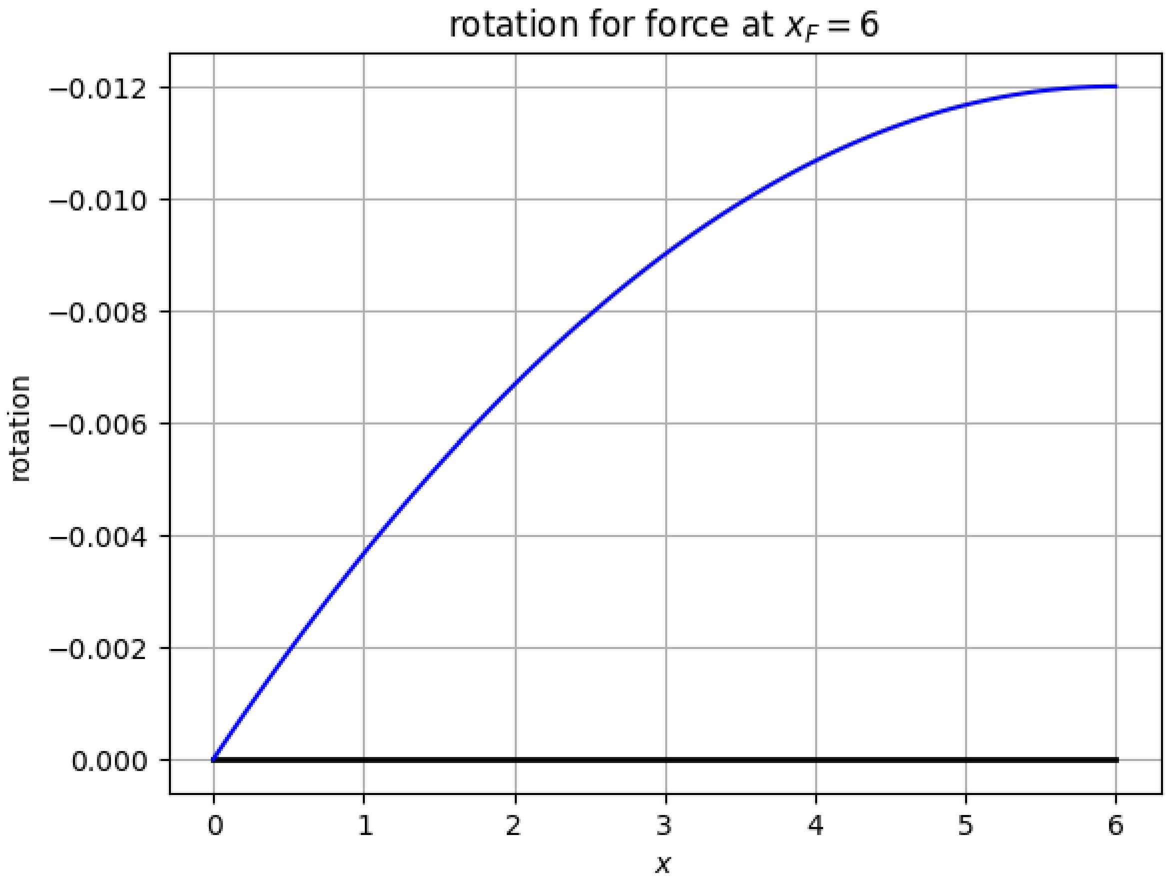
Contents

Click 🚀 –> Live Code on the top right corner of this screen to investigate some influence lines!



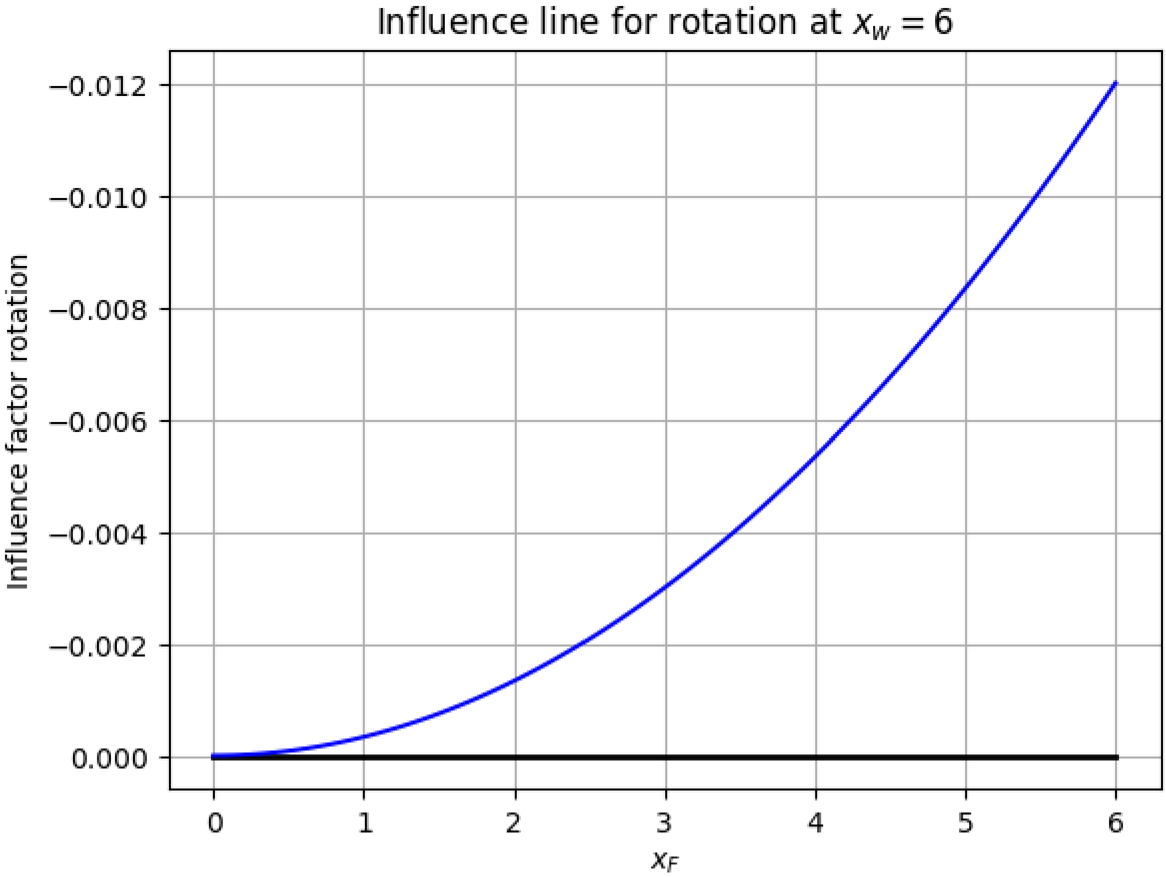
Find rotation line ϕ

$$\begin{cases} -\frac{1}{750} & \text{for } x > 2 \\ \frac{x(x-4)}{3000} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Find influence line ϕ

$$\begin{cases} \frac{3-x}{250} & \text{for}\ x > 6 \\ -\frac{x^2}{3000} & \text{for}\ x > 0 \\ 0 & \text{otherwise} \end{cases}$$



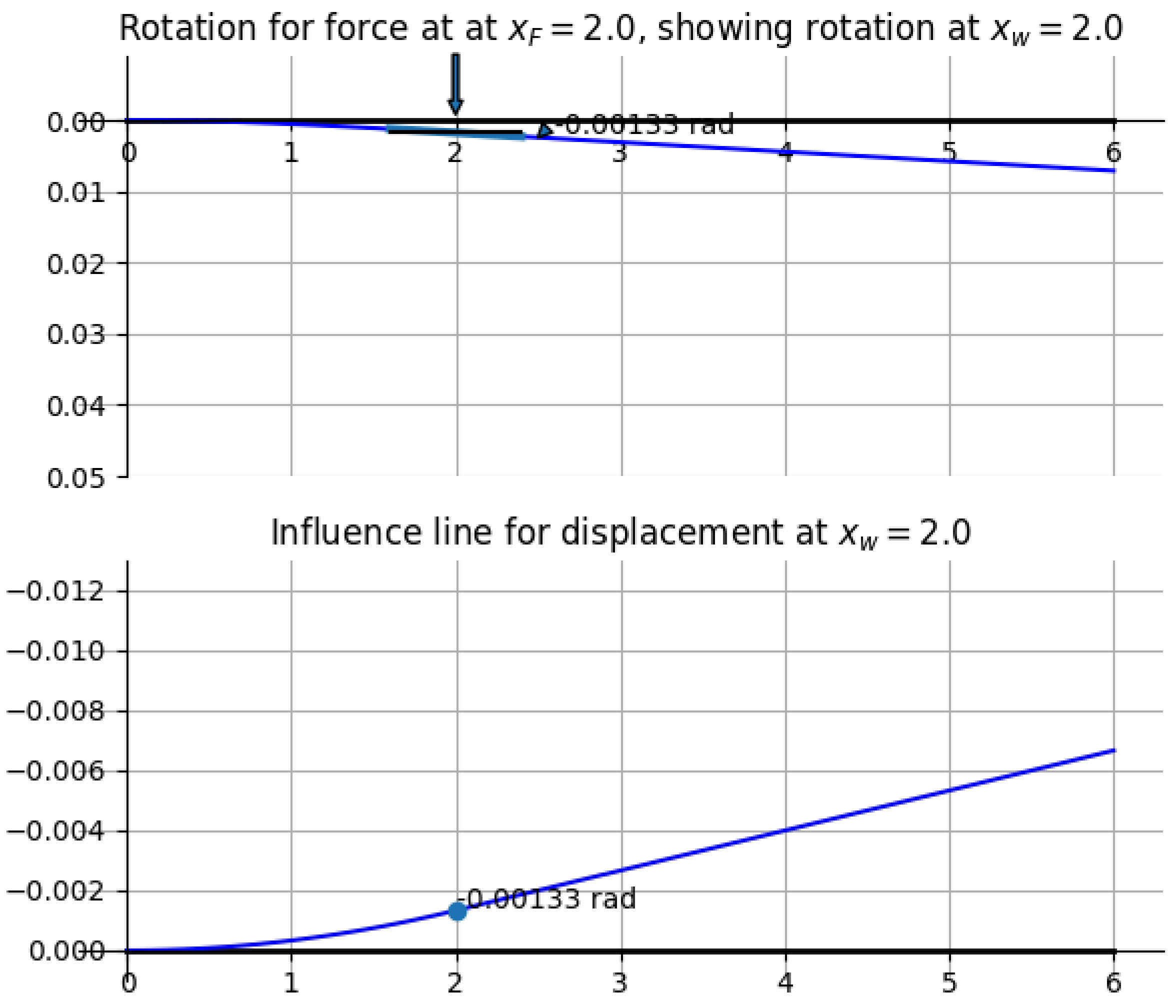
Comparison displacement with ϕ and

influence line for ϕ at $x =...$

Location rotation x_phi = ... (m) 2.0

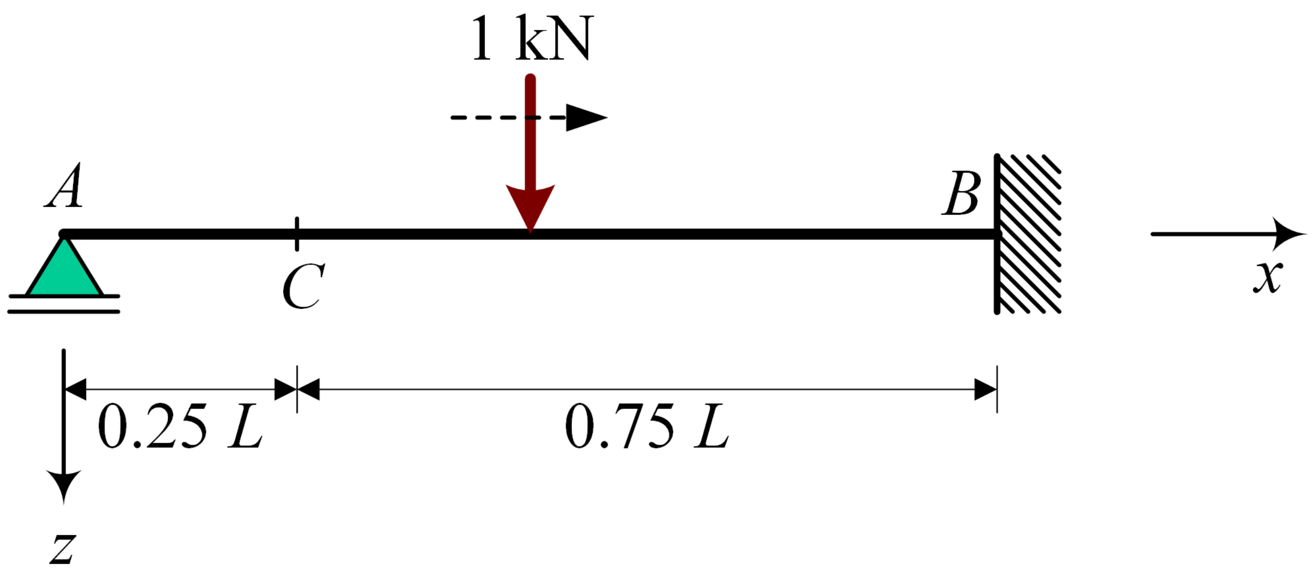
Location force x_F = ... (m) 2.0

Figure 1



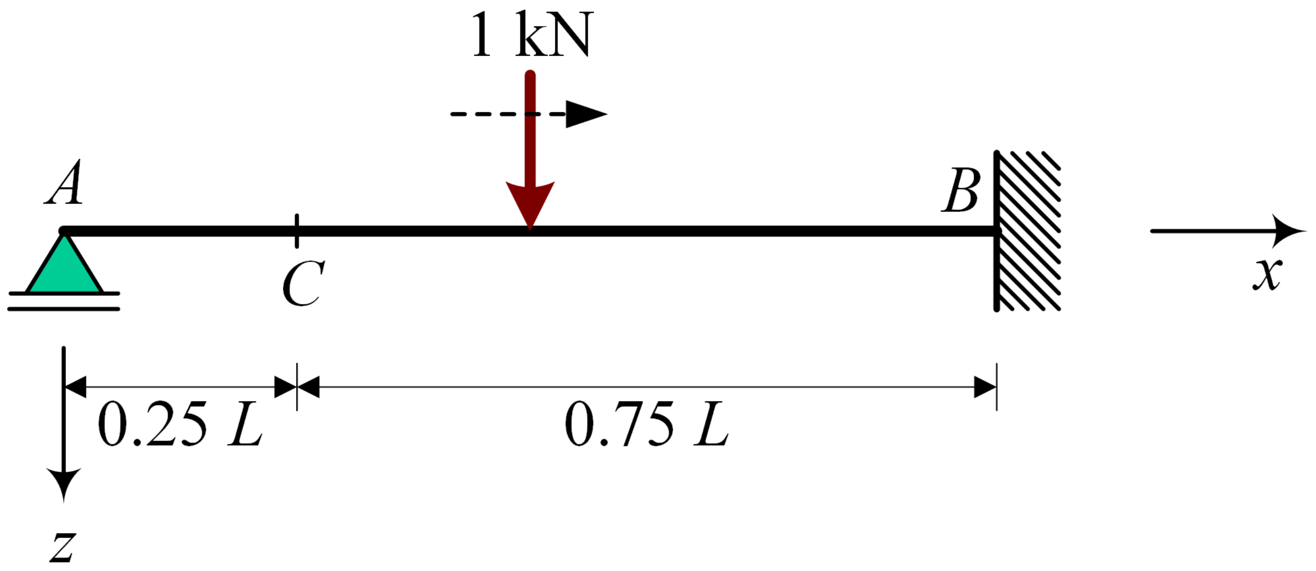
Statically indeterminate beam 1

The influence lines of the following structure will be investigated



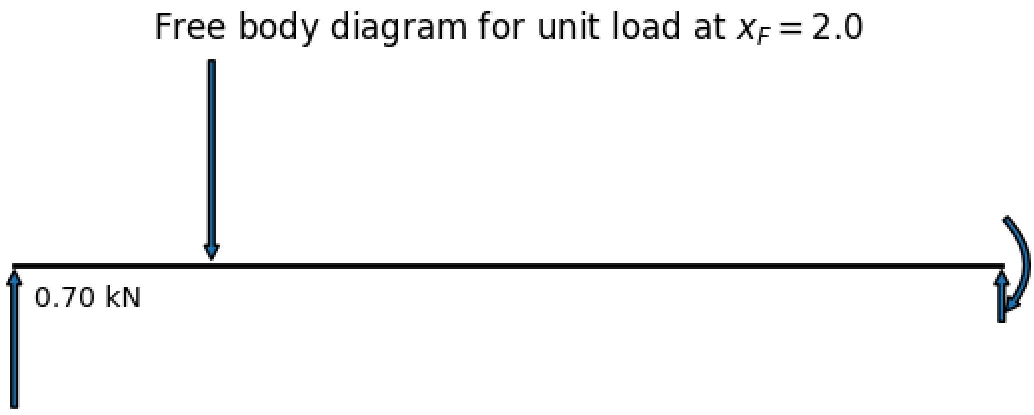
Support reaction A

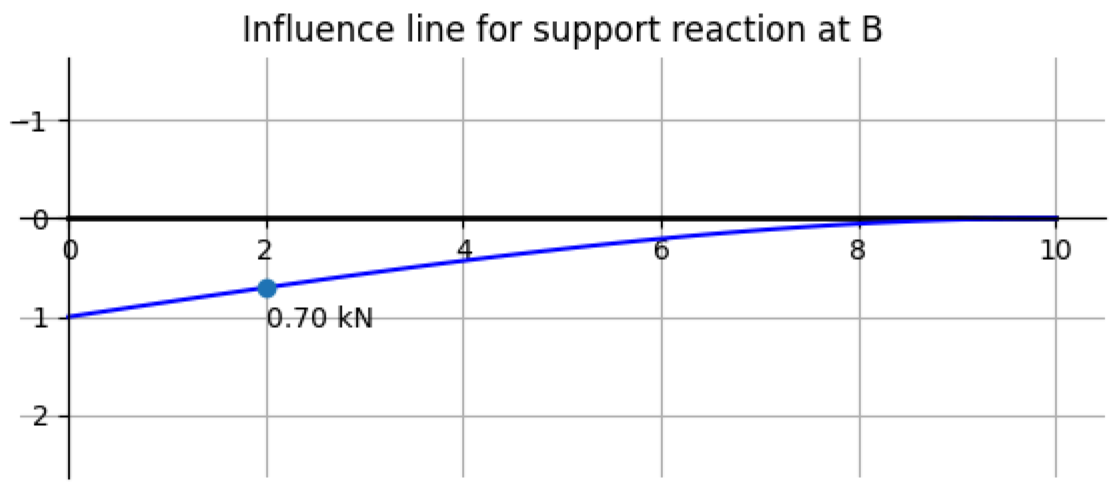
Click 🚀 –> Live Code on the top right corner of this screen to investigate some influence lines!



Location unit load x_F = ... (m)   2.0

Figure 1



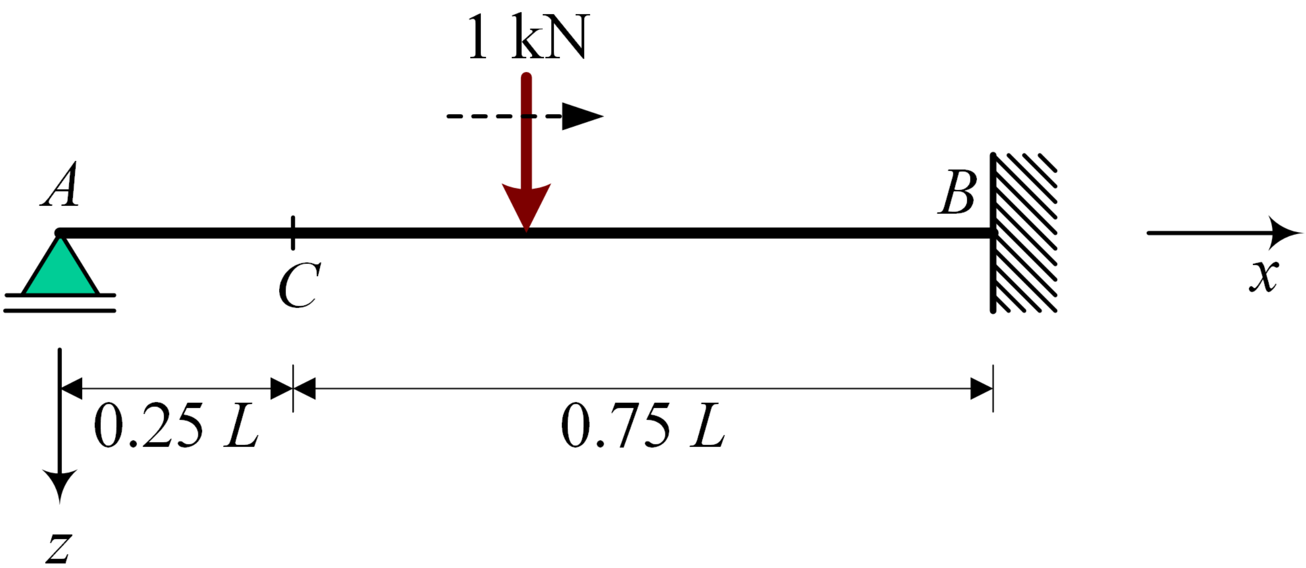


x=2.04 y=−0.003

Moments

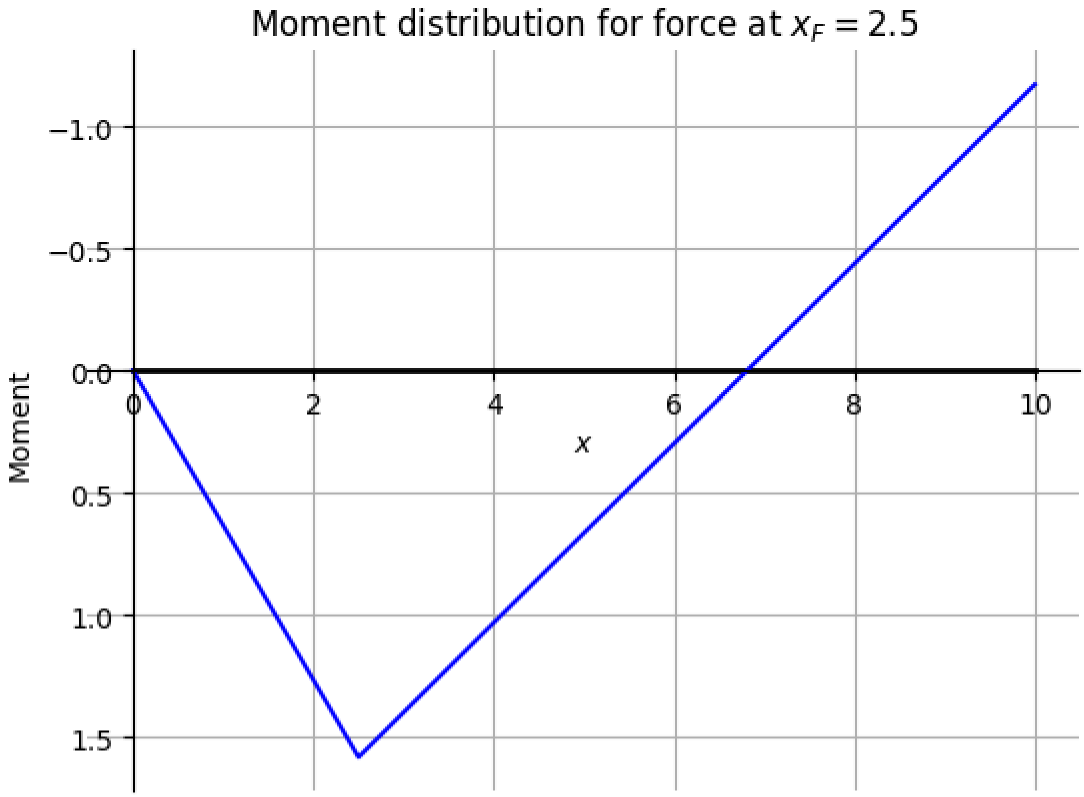
Contents

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some influence lines!



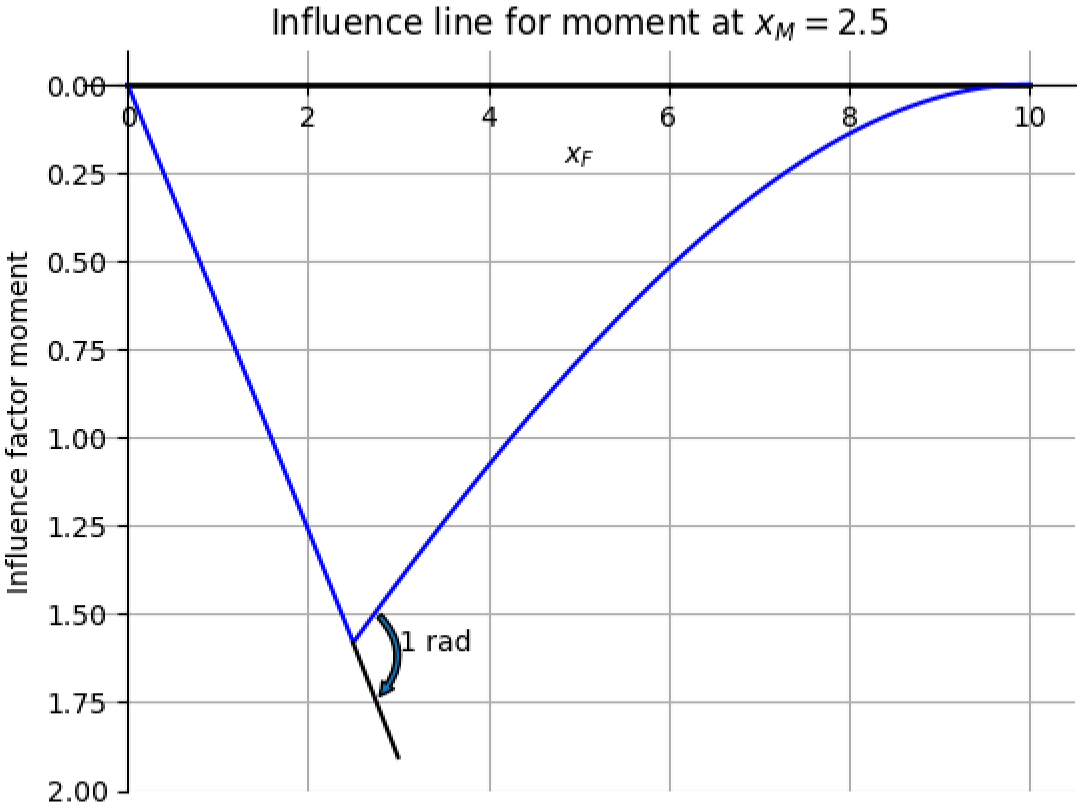
Find M-line

$$\begin{cases} 0 & \text{for } x > 10 \\ 2.5 - 0.3671875x & \text{for } x > 2.5 \\ 0.6328125x & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Finding influence line for M

$$\begin{cases} \frac{x^3 - 300x + 2000}{800} & \text{for } x > \frac{5}{2} \\ \frac{x\left(x^2 + 500\right)}{800} & \text{otherwise} \end{cases}$$



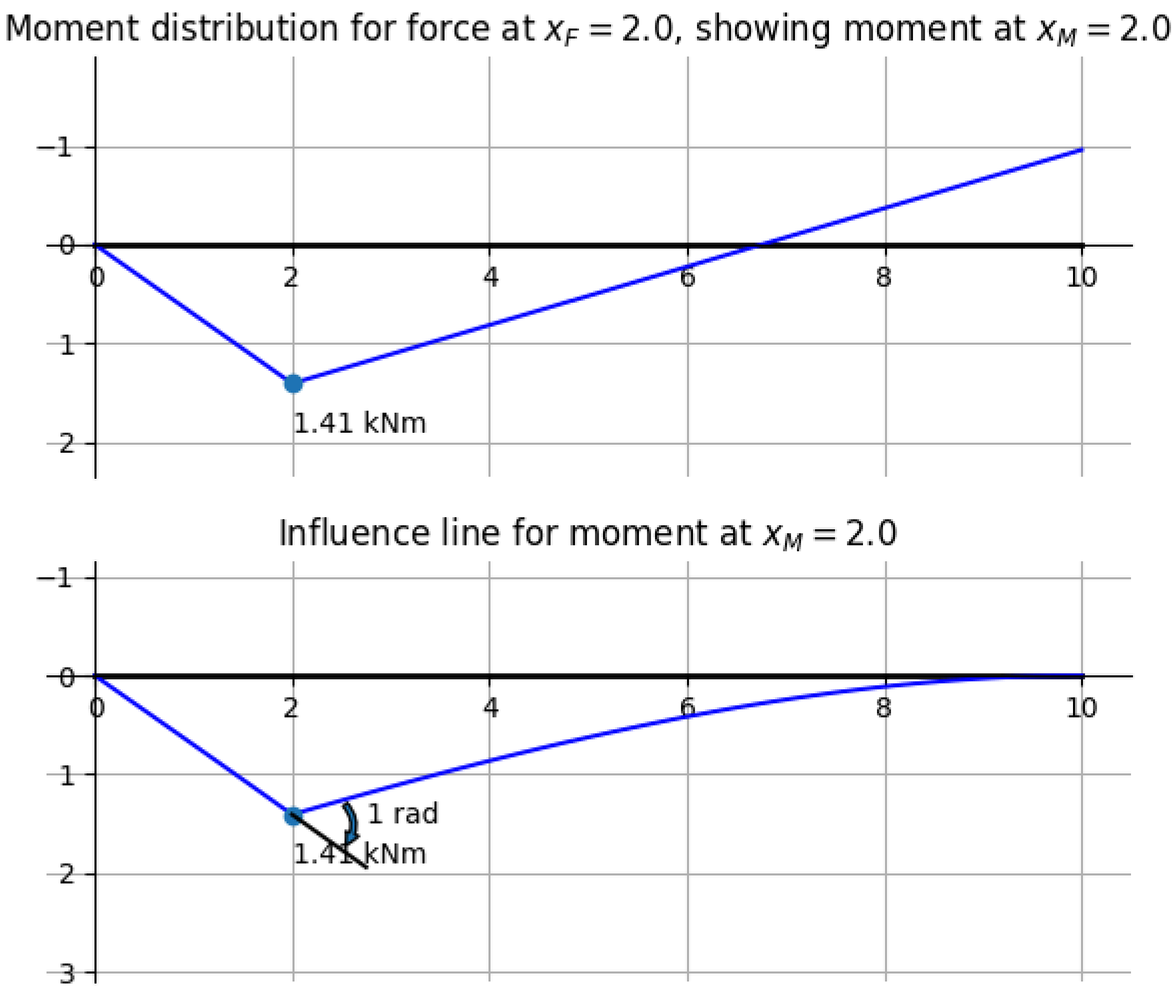
Comparison M-line and influence line for

moment at $x = \ldots$

Location moment x_M = ... (m) 2.0

Location unit load x_F = ... (m) 2.0

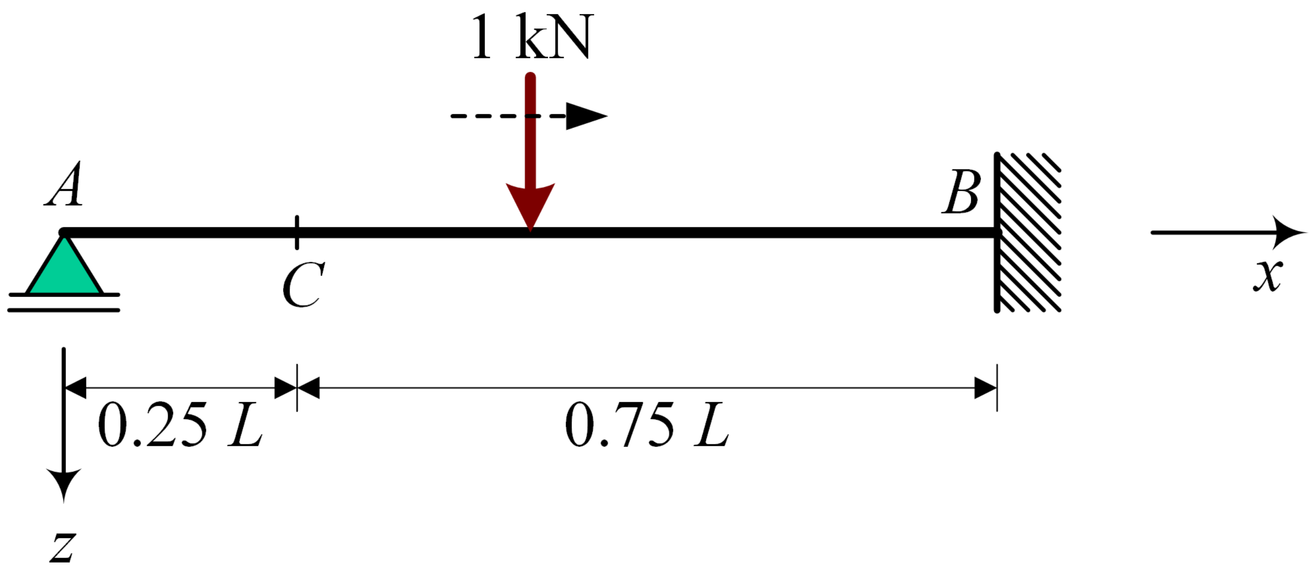
Figure 1



Shear forces

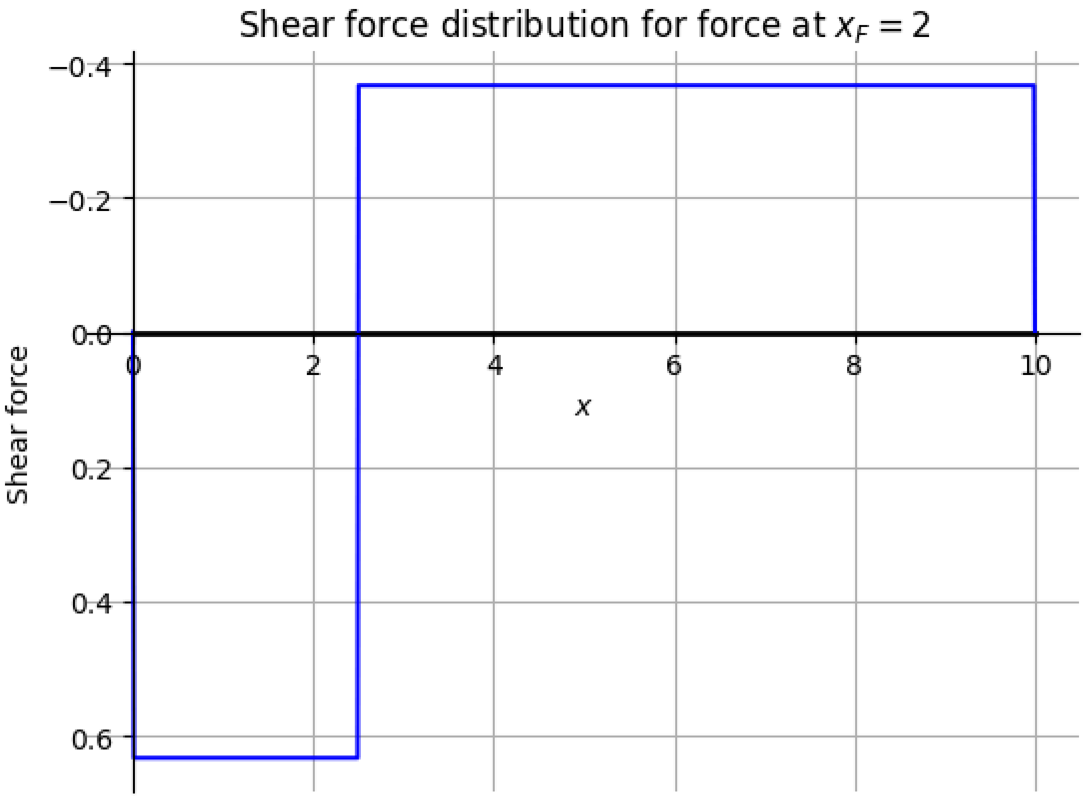
Contents

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some influence lines!



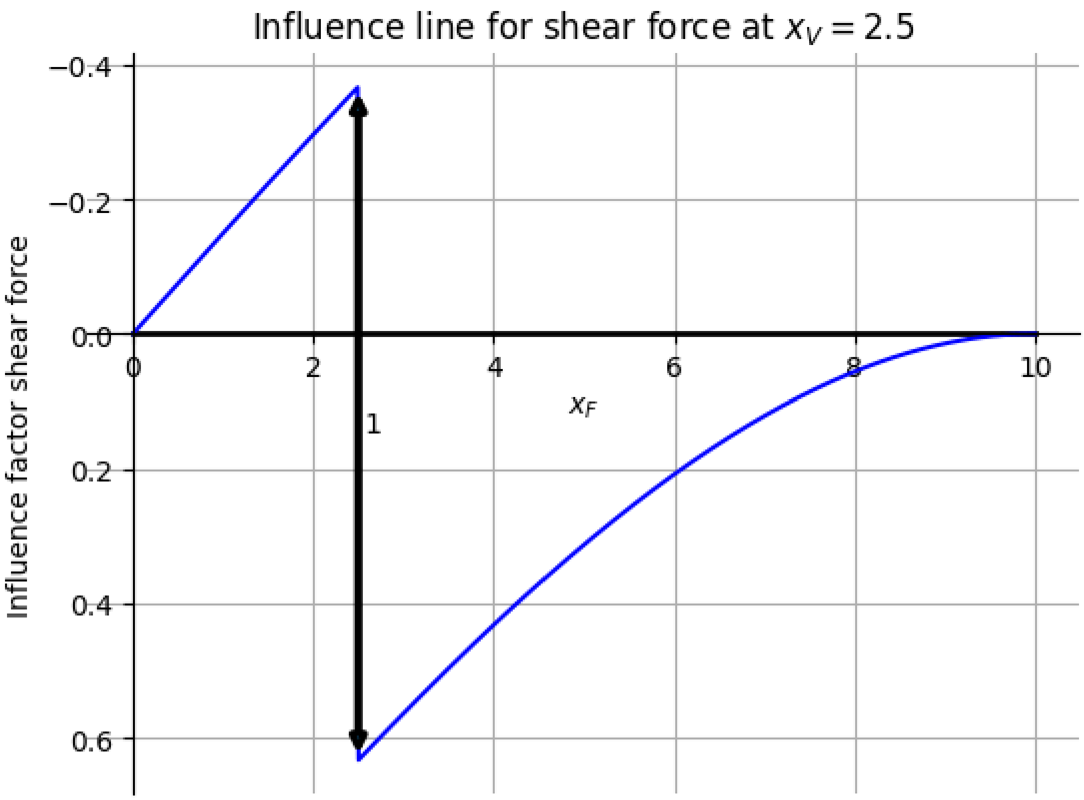
Find V-line

$$\begin{cases} 0 & \text{for}\ x > 10 \\ \infty & \text{for}\ x = 10 \\ -0.3671875 & \text{for}\ x > 2.5 \\ 0.6328125 & \text{for}\ x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Finding influence line for V

$$\begin{cases} \frac{x^{3}}{2000} - \frac{3x}{20} + 1 & \text{for } x > 2 \\ \frac{x\left(x^{2} - 300\right)}{2000} & \text{otherwise} \end{cases}$$



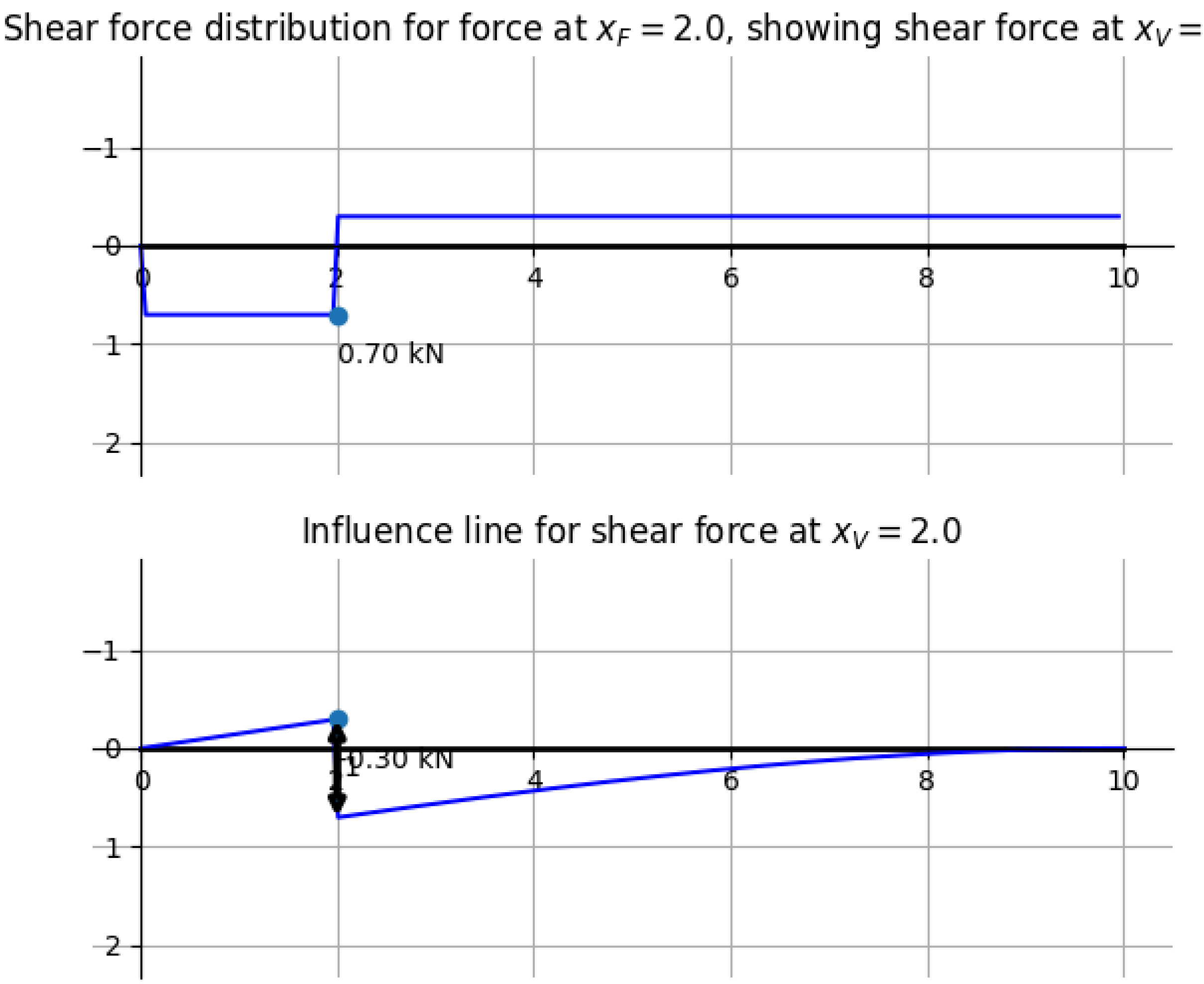
Comparison V-line and influence line for

shear force at $x = \ldots$

Location shear force x_V = ... 2.0

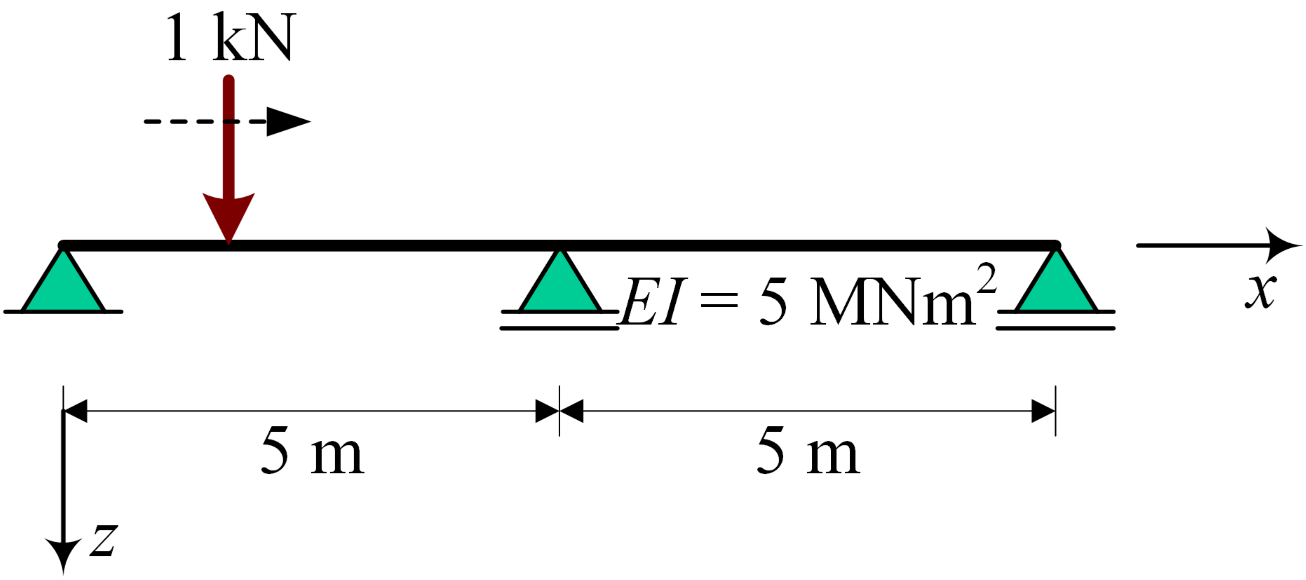
Location unit load x_F = ... (m) 2.0

Figure 1



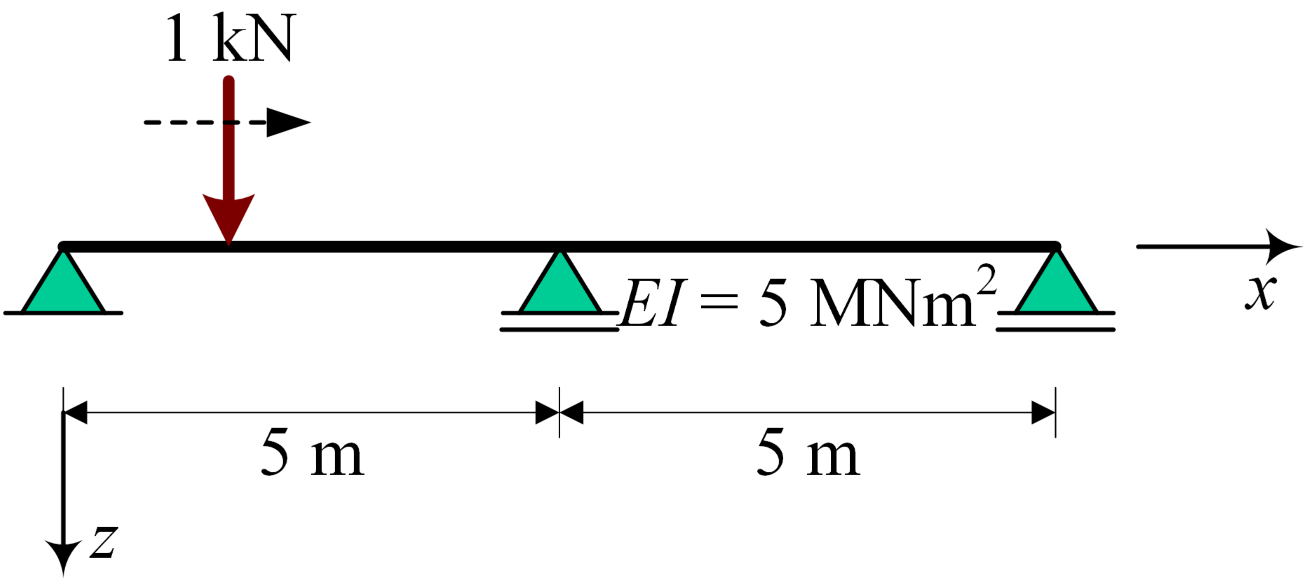
Statically indeterminate beam 2

The influence lines of the following structure will be investigated



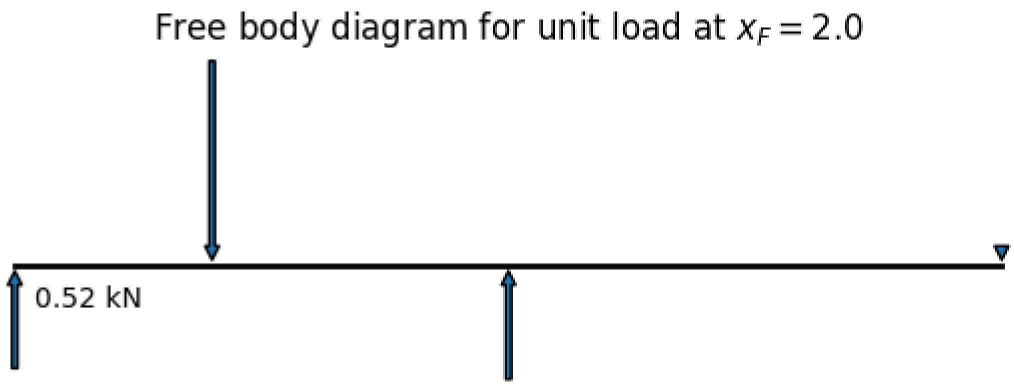
Support reaction A

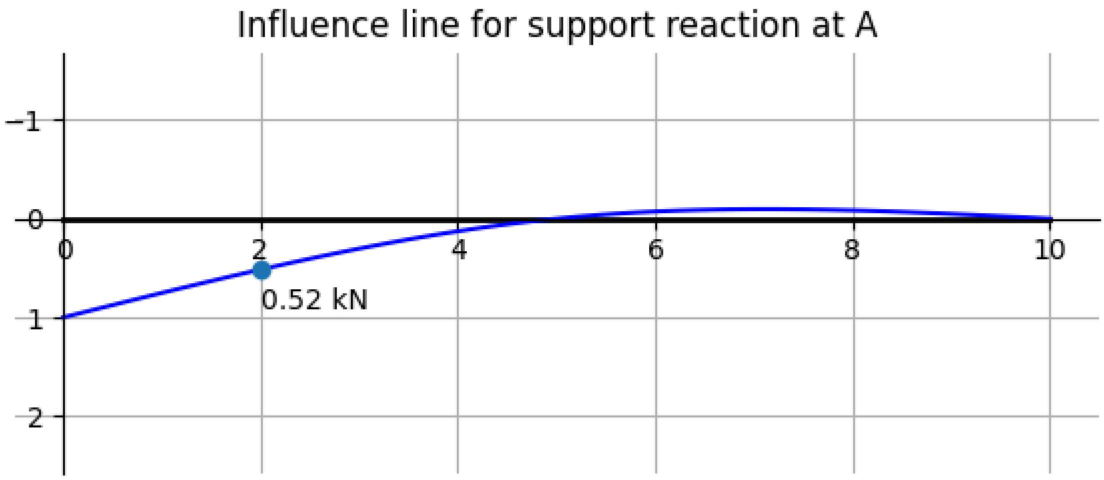
Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some influence lines!



Location unit load x_F = ... (m) 2.0

Figure 1

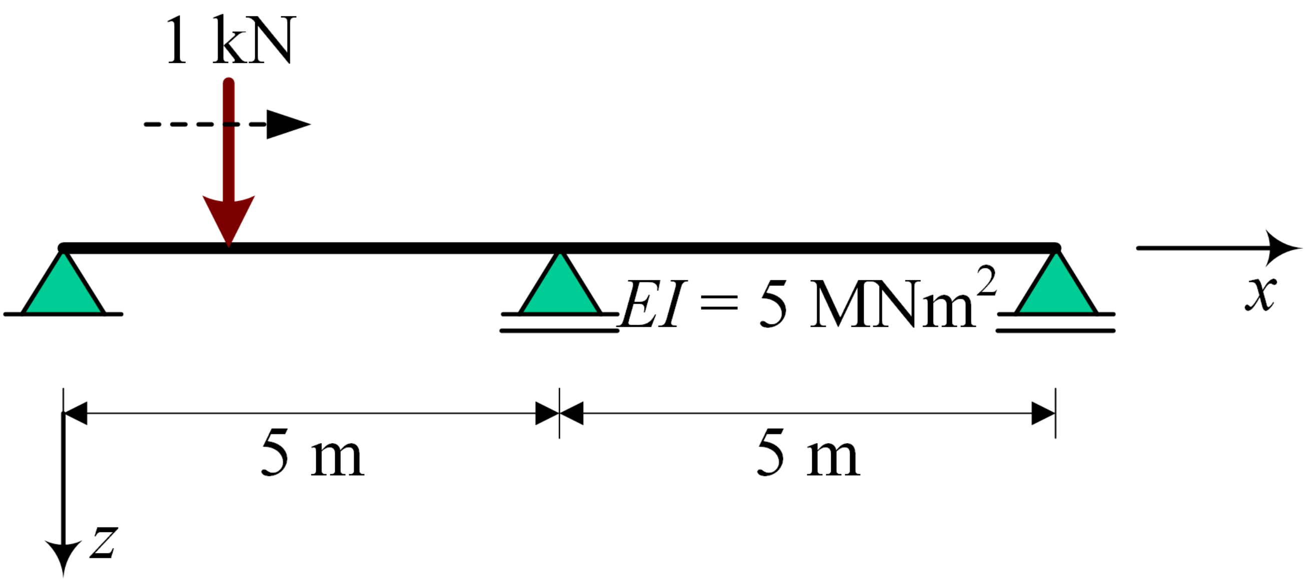




Moments

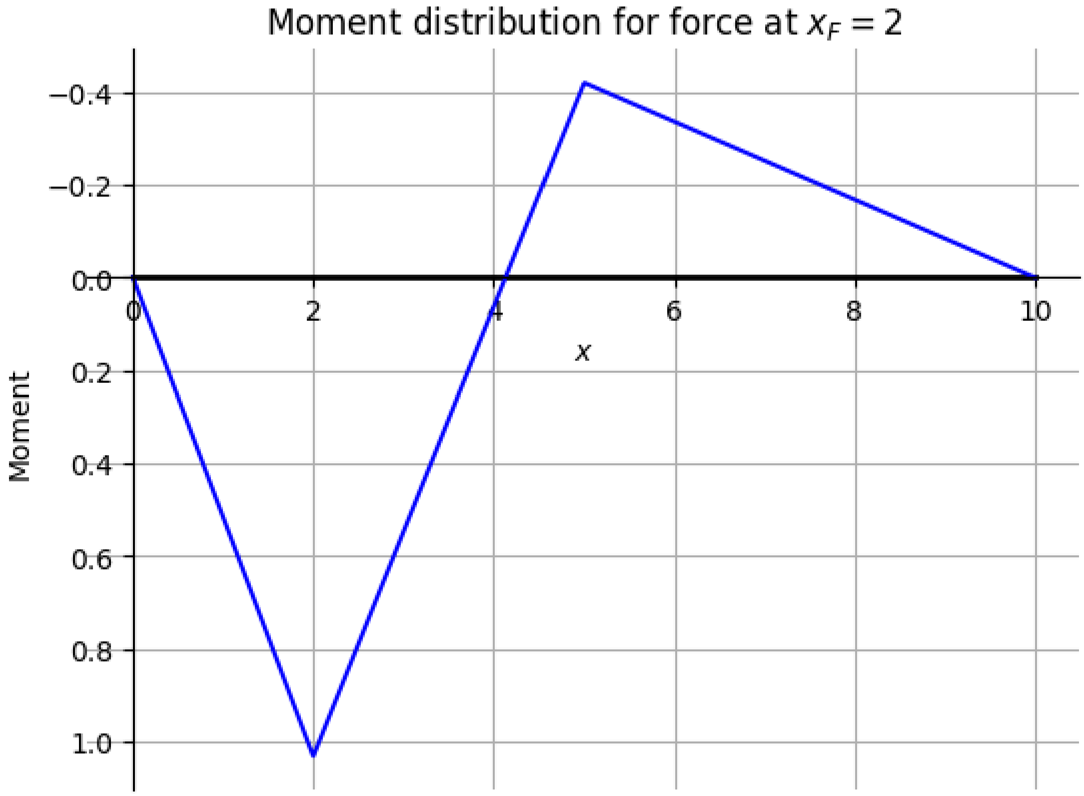
Contents

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some influence lines!



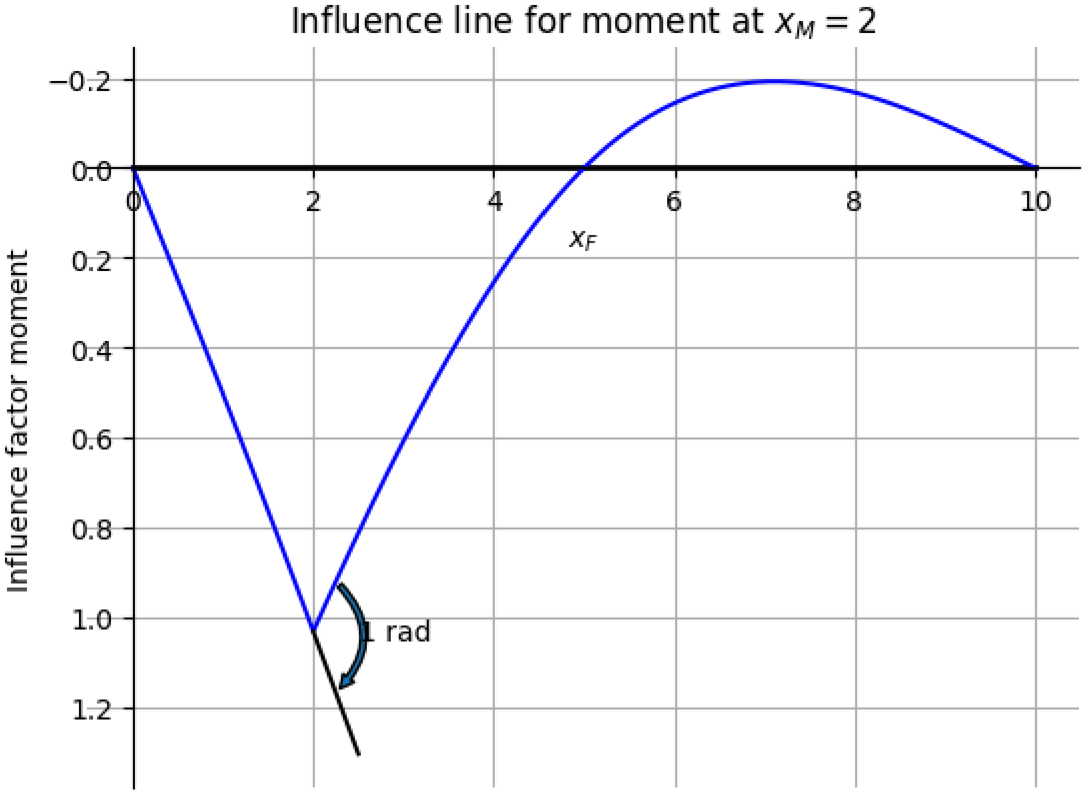
Find M-line

$$\begin{cases} 0 & \text{for}\ x > 10 \\ \frac{21(x-10)}{250} & \text{for}\ x > 5 \\ 2 - \frac{121x}{250} & \text{for}\ x > 2 \\ \frac{129x}{250} & \text{for}\ x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Finding influence line for M

$$\begin{cases} \frac{x}{10} - 1 & \text{for } x > 10 \\ -\frac{x^3}{250} + \frac{3x^2}{25} - \frac{11x}{10} + 3 & \text{for } x > 5 \\ \frac{x^3}{250} - \frac{x}{2} + 2 & \text{for } x > 2 \\ \frac{x\left(x^2+125\right)}{250} & \text{for } x > 0 \\ \frac{x}{2} & \text{otherwise} \end{cases}$$



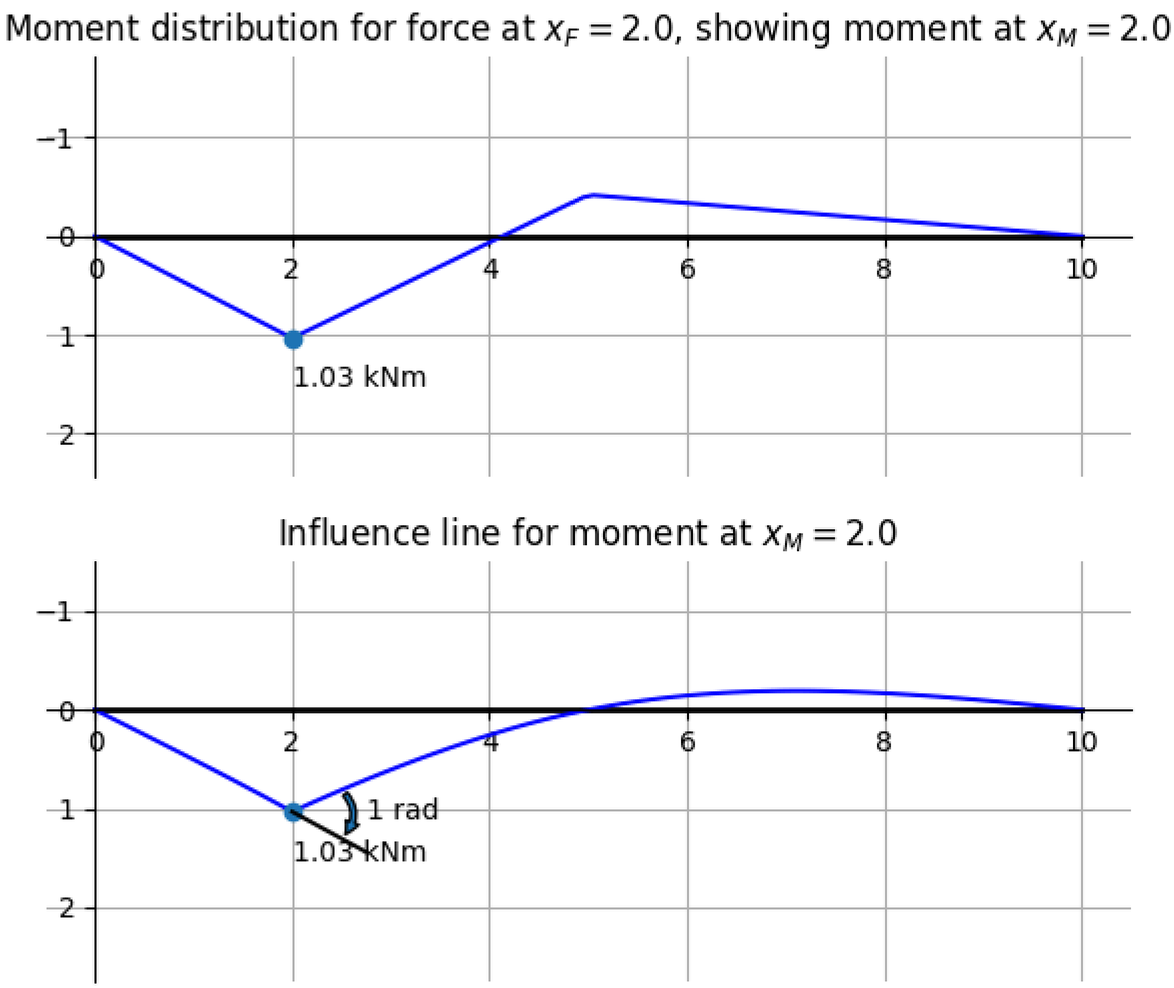
Comparison M-line and influence line for

moment at $x =\ldots$

Location moment x_M = ... (m) 2.0

Location unit load x_F = ... (m) 2.0

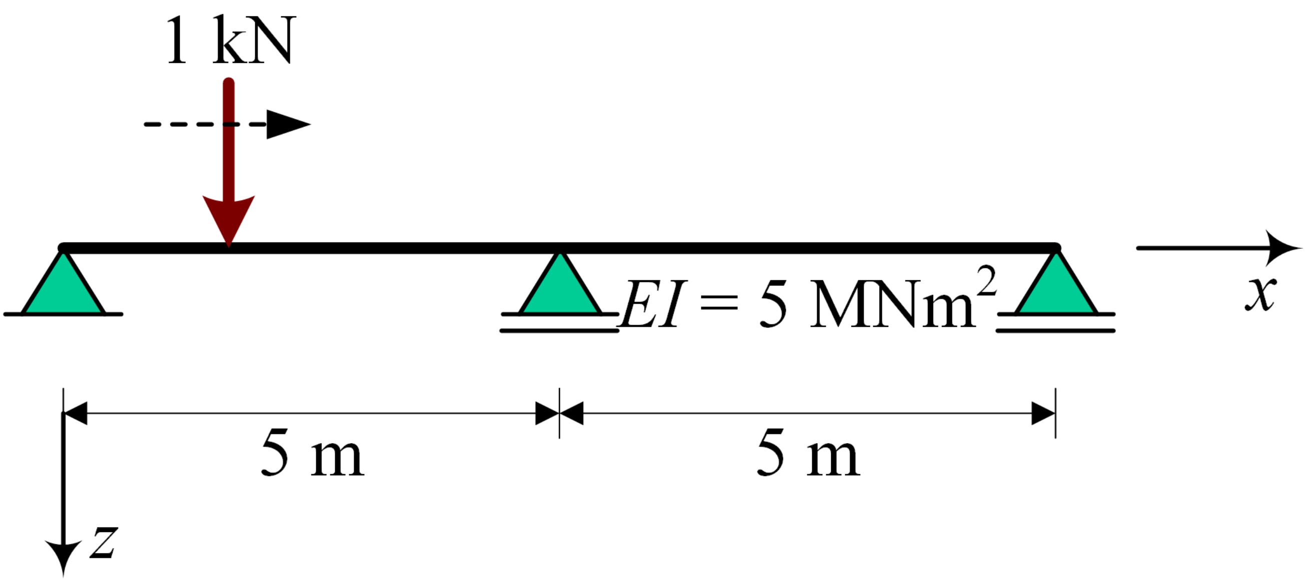
Figure 1



Shear forces

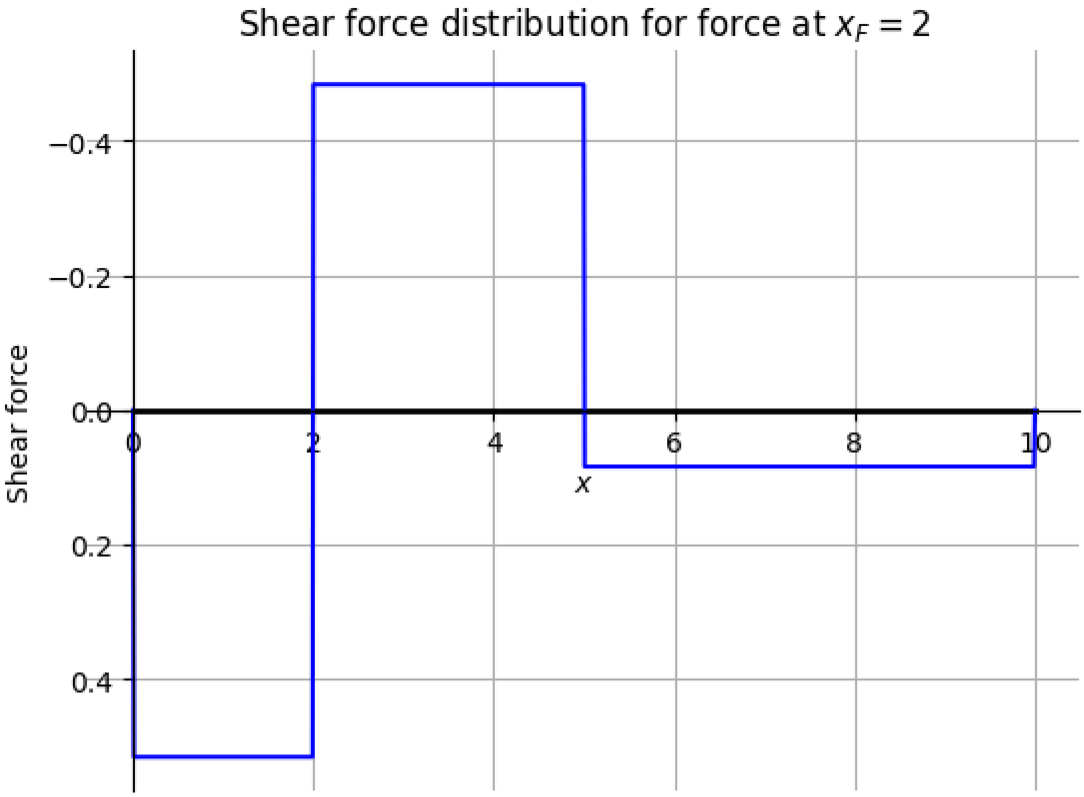
Contents

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some influence lines!



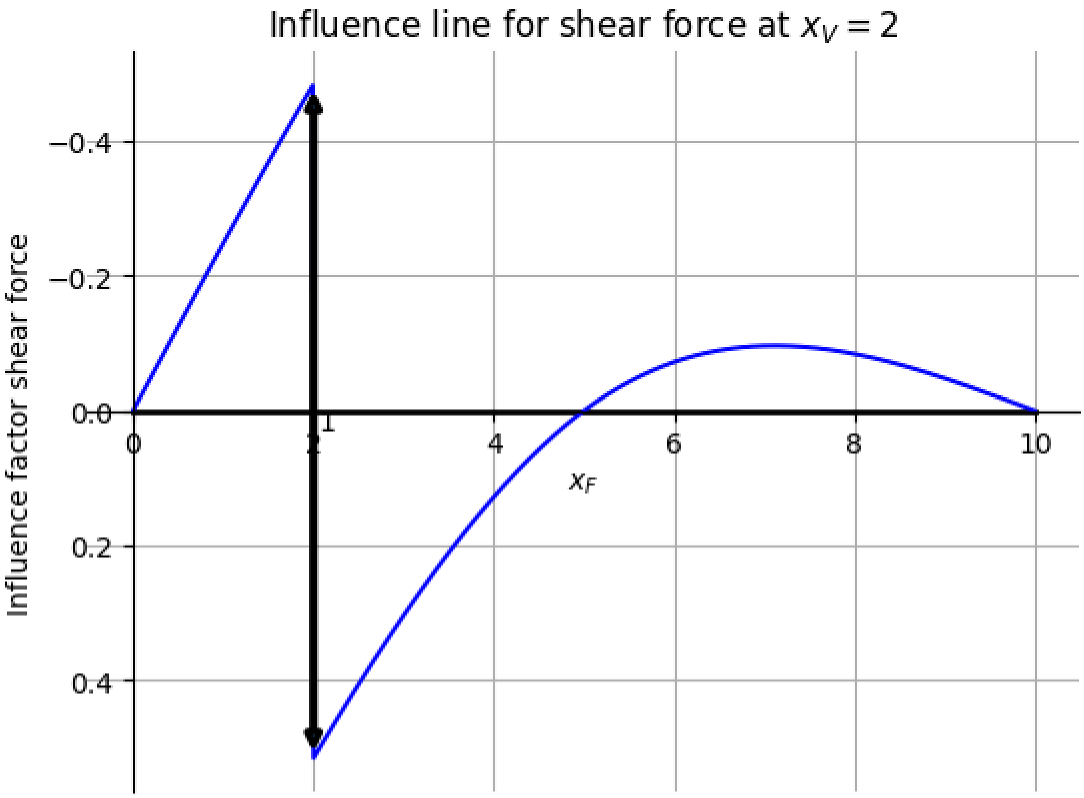
Find V-line

$$\begin{cases} 0 & \text{for}\ x > 10 \\ \frac{21}{250} & \text{for}\ x > 5 \\ - \frac{121}{250} & \text{for}\ x > 2 \\ \frac{129}{250} & \text{for}\ x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Finding influence line for V

$$\begin{cases} \frac{4993(x-10)}{100000} & \text{for } x > 10 \\ \frac{4993(-x^3+30x^2-275x+750)}{2500000} & \text{for } x > 5 \\ \frac{14879x^3+1500x^2-1879875x+7502000}{7500000} & \text{for } x > 2 \\ \frac{x(5043x^2-625625)}{2500000} & \text{for } x > 0 \\ -\frac{1001x}{4000} & \text{otherwise} \end{cases}$$



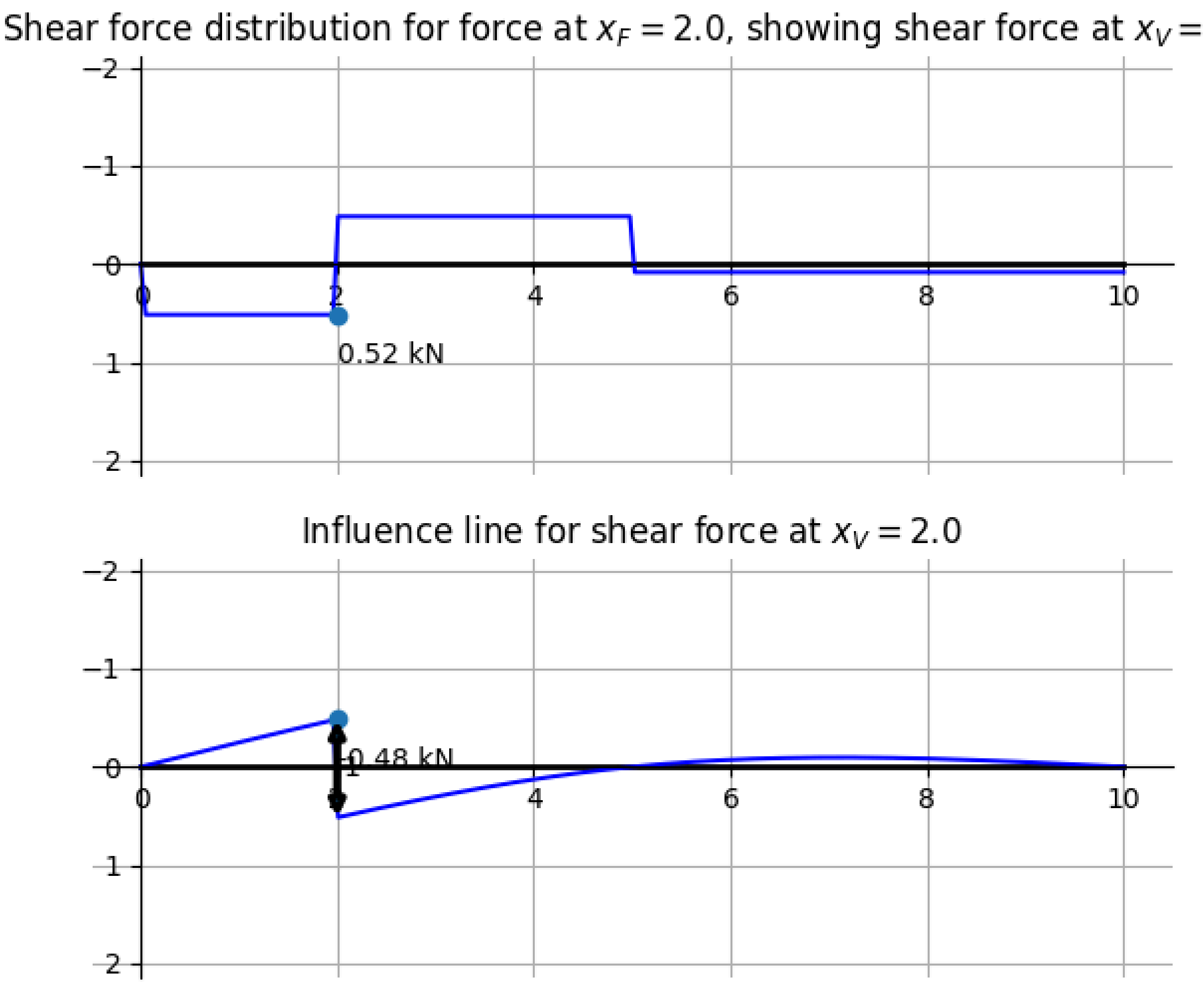
Comparison V-line and influence line for

shear force at $x = \ldots$

Location shear force x_V = ...   2.0

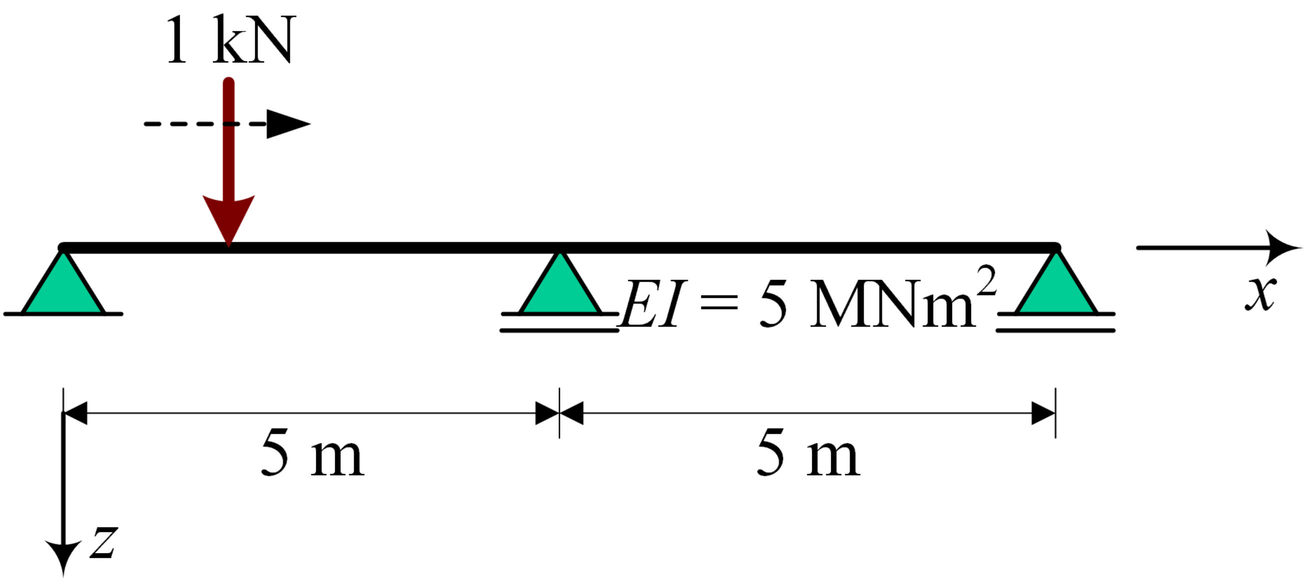
Location unit load x_F = ... (m)   2.0

Figure 1



ODE

Contents




```
import sympy as sym
```

```
x = sym.symbols('x')
```

```
EI = 5000
```

Find displacement line

```python
q_BC = 0
q_AB = sym.SingularityFunction(x,3,-1)
display(q_AB)
```

$$\langle x - 3\rangle^{-1}$$

```python
C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8 = sym.symbols('C_1, C_2, C_3, C_4, C_5, C_6, C_7,
V_AB = - sym.integrate(q_AB,x) + C_1
M_AB = sym.integrate(V_AB,x) + C_2
kappa_AB = M_AB / EI
phi_AB = sym.integrate(kappa_AB,x)+C_3
w_AB = - sym.integrate(phi_AB,x)+C_4
display(w_AB)
```

$$-\frac{C_1 x^3}{30000} - \frac{C_2 x^2}{10000} - C_3 x + C_4 + \frac{\langle x - 3\rangle^3}{30000}$$

```python
V_BC = - sym.integrate(q_BC,x) + C_5
M_BC = sym.integrate(V_BC,x) + C_6
kappa_BC = M_BC / EI
phi_BC = sym.integrate(kappa_BC,x)+C_7
w_BC = - sym.integrate(phi_BC,x)+C_8
display(w_BC)
```

$$-\frac{C_5 x^3}{30000} - \frac{C_6 x^2}{10000} - C_7 x + C_8$$

```python
eq1 = sym.Eq(w_AB.subs(x,0),0)
eq2 = sym.Eq(M_AB.subs(x,0),0)
eq3 = sym.Eq(w_AB.subs(x,5),0)
eq4 = sym.Eq(w_BC.subs(x,5),0)
eq5 = sym.Eq(M_AB.subs(x,5),M_BC.subs(x,5))
eq6 = sym.Eq(phi_AB.subs(x,5),phi_BC.subs(x,5))
eq7 = sym.Eq(w_BC.subs(x,10),0)
eq8 = sym.Eq(M_BC.subs(x,10),0)
```

```python
sol = sym.solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8],[C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8
```

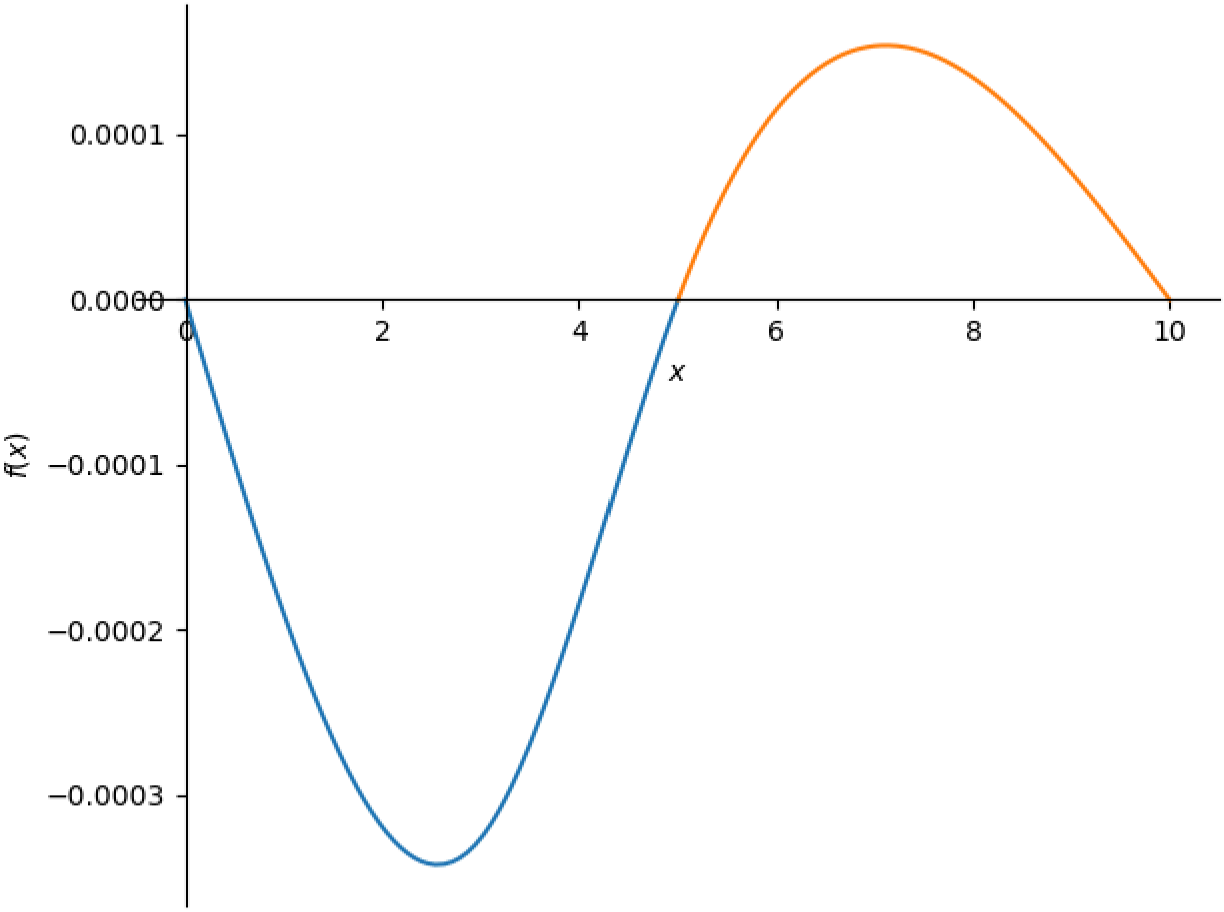
```python
w_sol_AB = w_AB.subs(sol)
display(w_sol_AB)
```

$$-\frac{19x^3}{1875000} + \frac{x}{5000} + \frac{\langle x-3 \rangle^3}{30000}$$

```
w_sol_BC = w_BC.subs(sol)
display(w_sol_BC)
```

$$-\frac{x^3}{312500} + \frac{3x^2}{31250} - \frac{11x}{12500} + \frac{3}{1250}$$

```
sym.plot((-w_sol_AB,(x,0,5)),(-w_sol_BC,(x,5,10)));
```



Find influence lines A_v with Müller-Breslau

```
q_BC = 0
q_AB = 0
display(q_AB)
```

```
0
```

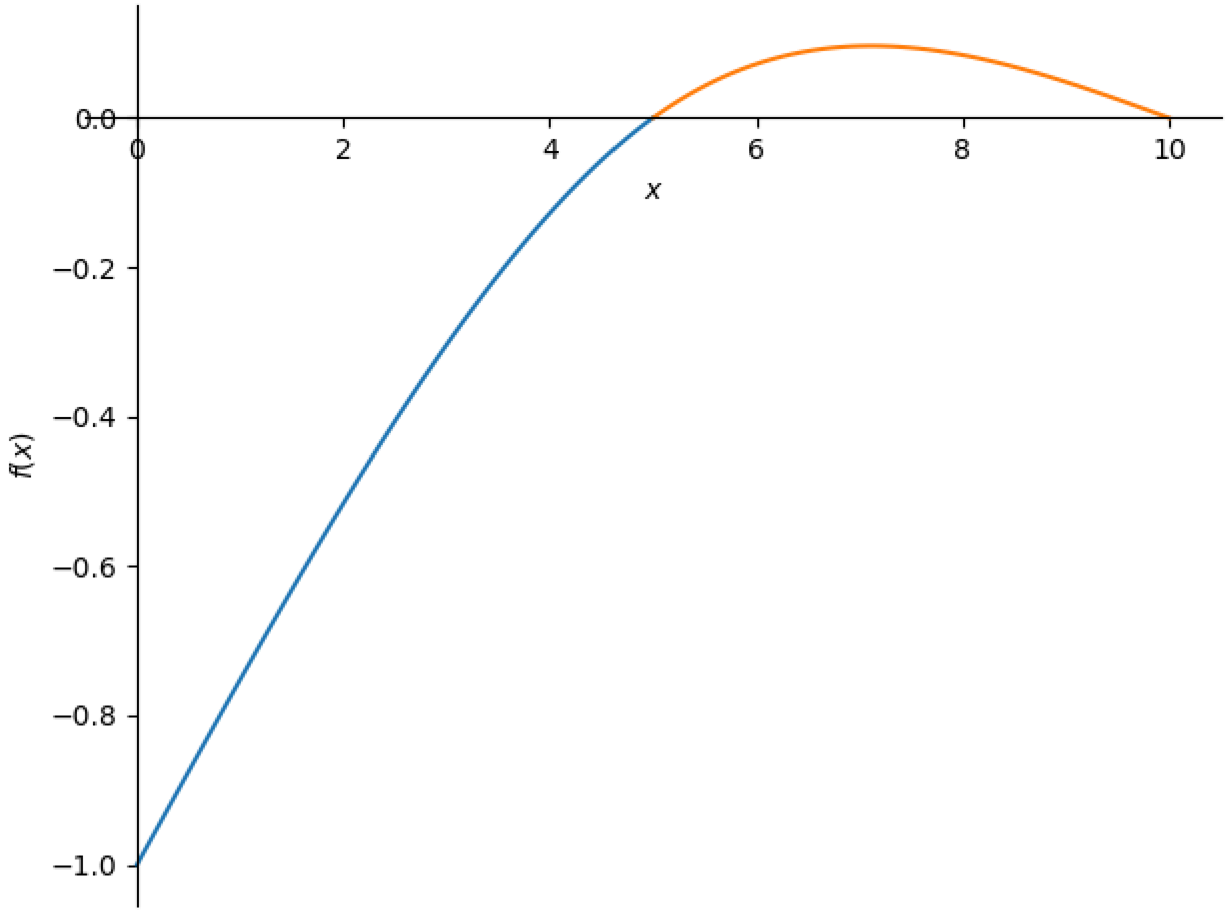
```
V_AB = - sym.integrate(q_AB,x) + C_1
M_AB = sym.integrate(V_AB,x) + C_2
kappa_AB = M_AB / EI
phi_AB = sym.integrate(kappa_AB,x)+C_3
w_AB = - sym.integrate(phi_AB,x)+C_4
```

```
eq1 = sym.Eq(w_AB.subs(x,0),1)
```

```
sol = sym.solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8],[C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8
```

```
w_sol_AB = w_AB.subs(sol)
w_sol_BC = w_BC.subs(sol)
```

```
sym.plot((-w_sol_AB,(x,0,5)),(-w_sol_BC,(x,5,10)));
```



Find influence line V for $x = 5^-$ with Müller-Breslau

```
w_AB = - sym.integrate(phi_AB,x)+C_4 + sym.SingularityFunction(x,4.9999,0)
display(w_AB)
```

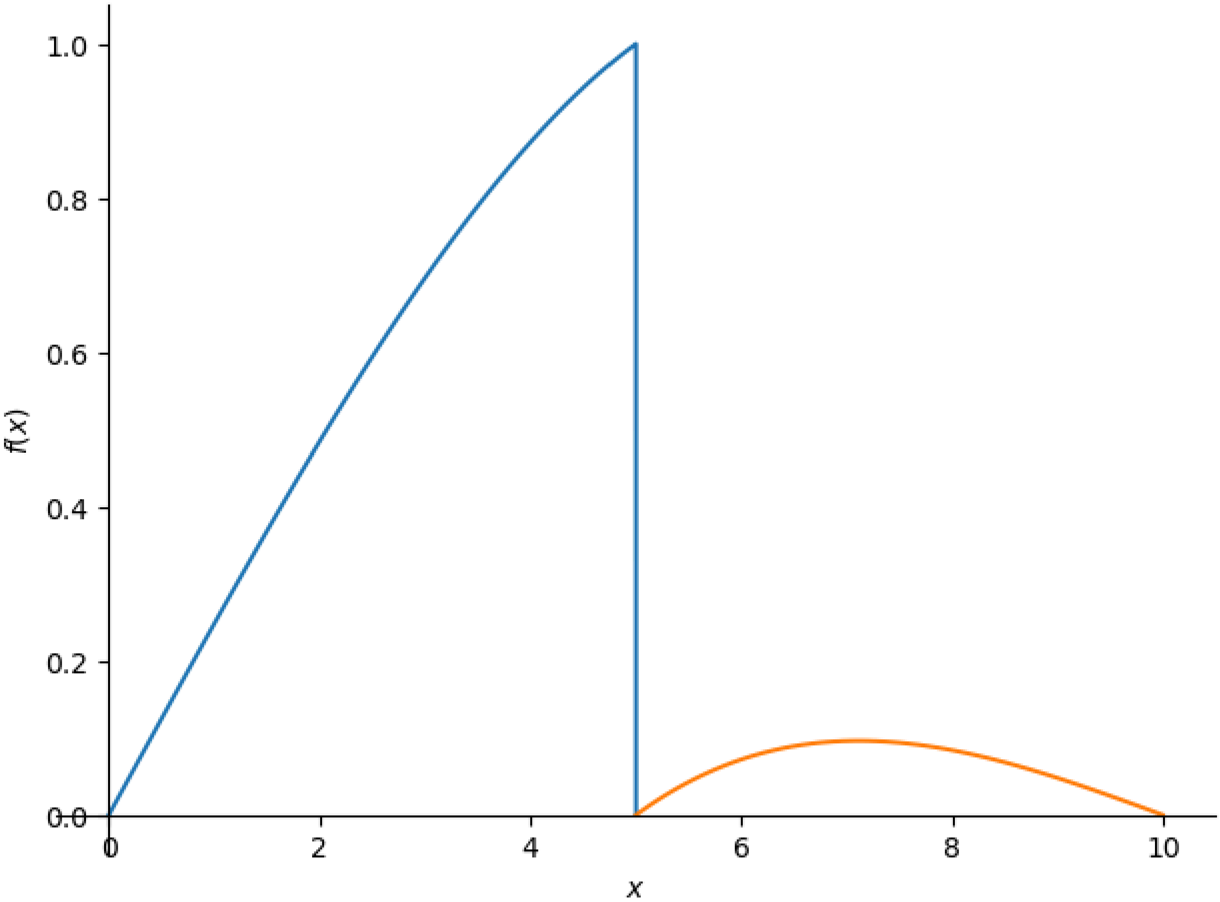
$$-\frac{C_1 x^3}{30000} - \frac{C_2 x^2}{10000} - C_3 x + C_4 + \langle x - 4.9999 \rangle^0$$

```
eq1 = sym.Eq(w_AB.subs(x,0),0)
eq2 = sym.Eq(M_AB.subs(x,0),0)
eq3 = sym.Eq(w_AB.subs(x,5),0)
eq4 = sym.Eq(w_BC.subs(x,5),0)
eq5 = sym.Eq(M_AB.subs(x,5),M_BC.subs(x,5))
eq6 = sym.Eq(phi_AB.subs(x,5),phi_BC.subs(x,5))
eq7 = sym.Eq(w_BC.subs(x,10),0)
eq8 = sym.Eq(M_BC.subs(x,10),0)
```

```
sol = sym.solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8],[C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8
```

```
w_sol_AB = w_AB.subs(sol)
w_sol_BC = w_BC.subs(sol)
```

```
sym.plot((-w_sol_AB,(x,0,5)),(-w_sol_BC,(x,5,10)));
```



Find influence line M for $x = 3$ with Müller-Breslau

```
q_AB = 0
q_BC = 0
V_AB = -sym.integrate(q_AB,x)+C_1
M_AB = sym.integrate(V_AB,x)+C_2
kappa_AB = M_AB/ EI
phi_AB = sym.integrate(kappa_AB,x)+C_3 + sym.SingularityFunction(x,3,0)
display(phi_AB)
w_AB = sym.integrate(phi_AB,x)+C_4
```

$$\frac{C_1 x^2}{10000} + \frac{C_2 x}{5000} + C_3 + \langle x - 3 \rangle^0$$

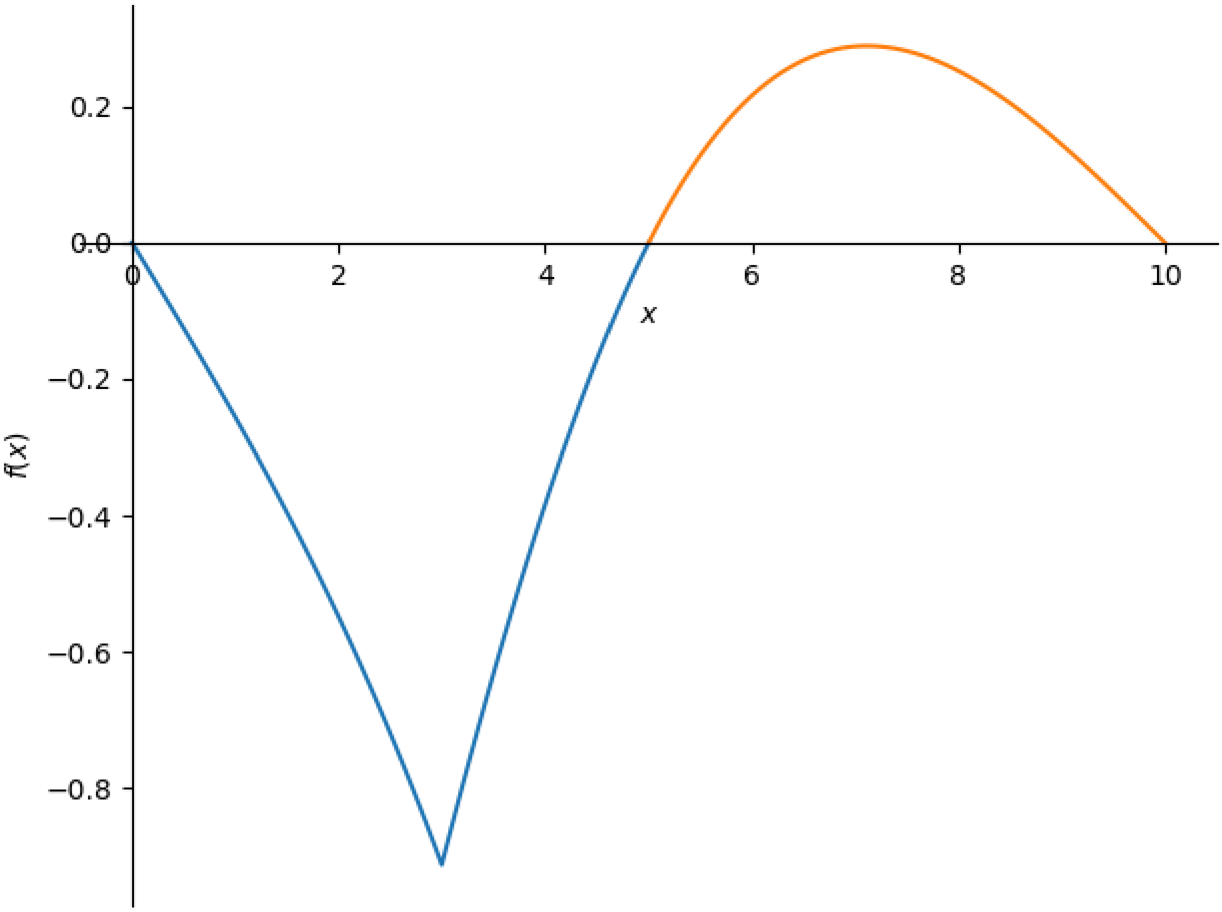
```
V_BC = -sym.integrate(q_BC,x)+C_5
M_BC = sym.integrate(V_BC,x)+C_6
kappa_BC = M_BC/EI
phi_BC = sym.integrate(kappa_BC,x) + C_7
w_BC = sym.integrate(phi_BC,x) + C_8
```

```
eq1 = sym.Eq(w_AB.subs(x,0),0)
eq2 = sym.Eq(M_AB.subs(x,0),0)
eq3 = sym.Eq(w_AB.subs(x,5),0)
eq4 = sym.Eq(w_BC.subs(x,5),0)
eq5 = sym.Eq(M_AB.subs(x,5),M_BC.subs(x,5))
eq6 = sym.Eq(phi_AB.subs(x,5),phi_BC.subs(x,5))
eq7 = sym.Eq(w_BC.subs(x,10),0)
eq8 = sym.Eq(M_BC.subs(x,10),0)
```

```
sol = sym.solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7,eq8],[C_1,C_2,C_3,C_4,C_5,C_6,C_7,C_8])
```

```
w_AB_sol = w_AB.subs(sol)
w_BC_sol = w_BC.subs(sol)
```

```
sym.plot((w_AB_sol,(x,0,5)),((w_BC_sol),(x,5,10)));
```



Find influence line with singularity load

```python
x_F = sym.symbols('x_F',positive=True)
A_v, B_v, C_v = sym.symbols('A_v, B_v, C_v')

q = -A_v *sym.SingularityFunction(x,0,-1) - B_v * sym.SingularityFunction(x,5,-1) - C_v *
```

```python
C_1, C_2, C_3, C_4,= sym.symbols('C_1, C_2, C_3, C_4')
V = - sym.integrate(q,x) + C_1
M = sym.integrate(V,x) + C_2
kappa = M / EI
phi = sym.integrate(kappa,x)+C_3
w = - sym.integrate(phi,x)+C_4
display(w)
```

$$-\frac{A_v\langle x\rangle^3}{30000}-\frac{B_v\langle x-5\rangle^3}{30000}-\frac{C_1x^3}{30000}-\frac{C_2x^2}{10000}-C_3x+C_4-\frac{C_v\langle x-10\rangle^3}{30000}+\frac{\langle x-x}{300}$$

```python
eq1 = sym.Eq(w.subs(x,0),0)
eq2 = sym.Eq(M.subs(x,0),0)
eq3 = sym.Eq(V.subs(x,-sym.nsimplify(1/1000)),0)
eq4 = sym.Eq(w.subs(x,5),0)
eq5 = sym.Eq(w.subs(x,10),0)
eq6 = sym.Eq(M.subs(x,10),0)
eq7 = sym.Eq(V.subs(x,10+sym.nsimplify(1/1000)),0)
```

```python
sol = sym.solve([eq1,eq2,eq3,eq4,eq5,eq6,eq7],[C_1, C_2, C_3, C_4, C_5, A_v, B_v, C_v])
for key in sol:
    display(sym.Eq(key, sol[key]))
```

$$A_v=-\frac{\langle 5-x_F\rangle^3}{250}-\frac{\langle 10-x_F\rangle^1}{20}+\frac{\langle 10-x_F\rangle^3}{500}$$

$$B_v=\frac{\langle 5-x_F\rangle^3}{125}+\frac{3\langle 10-x_F\rangle^1}{10}-\frac{\langle 10-x_F\rangle^3}{250}$$

$$C_1=0$$

$$C_2=0$$

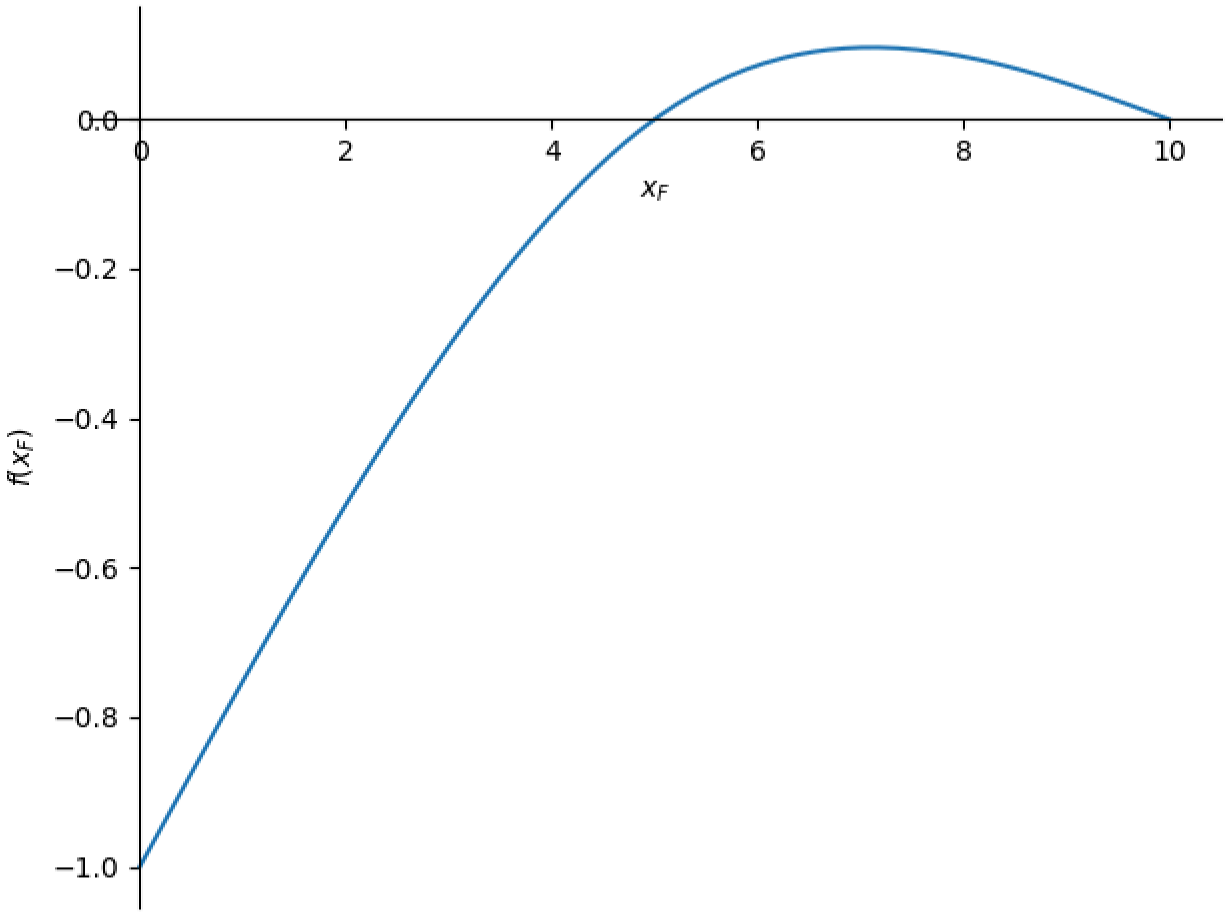
$$C_3=\frac{\langle 5-x_F\rangle^3}{100000}+\frac{\langle 10-x_F\rangle^1}{24000}-\frac{\langle 10-x_F\rangle^3}{600000}$$

$$C_4 = 0$$

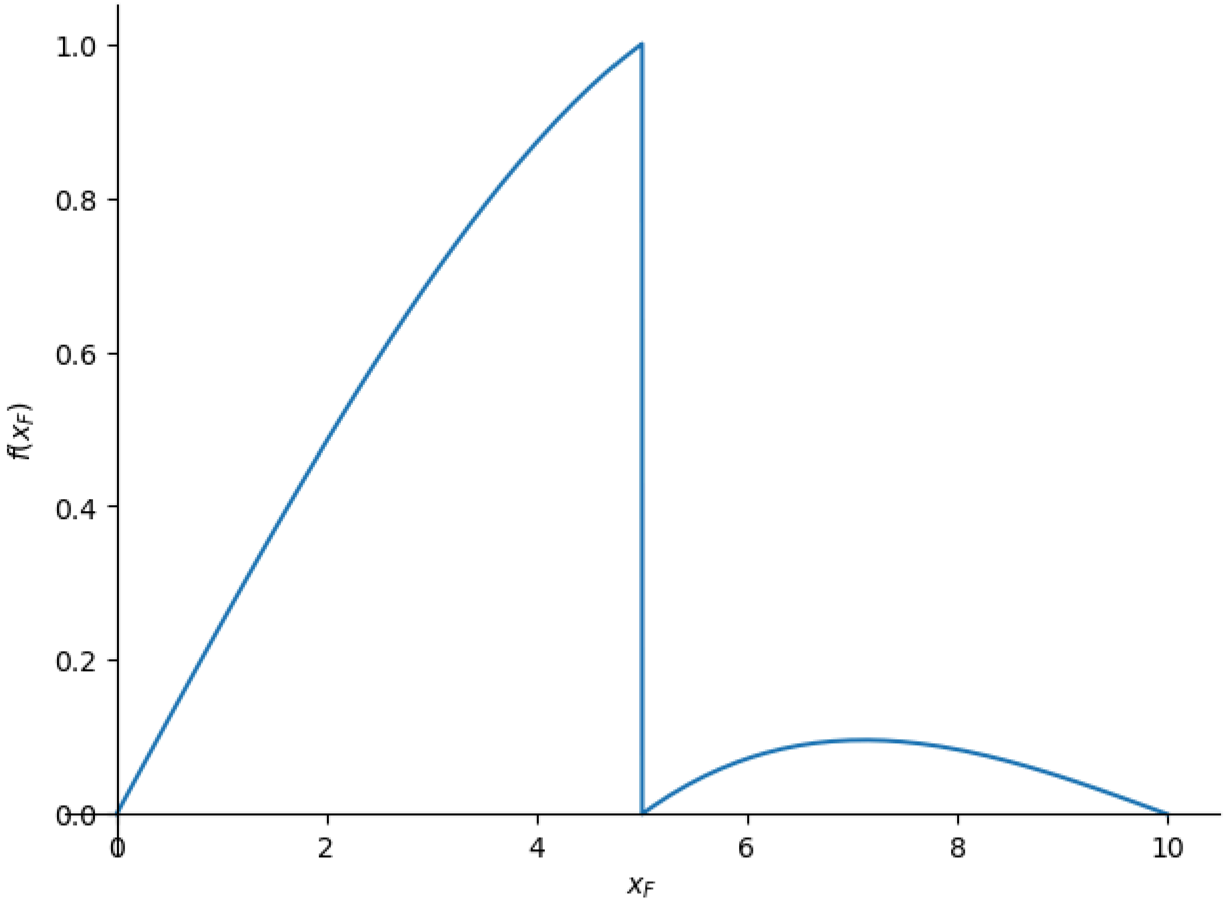
$$C_v = -\frac{\langle 5 - x_F \rangle^3}{250} - \frac{\langle 10 - x_F \rangle^1}{4} + \frac{\langle 10 - x_F \rangle^3}{500} + \left\langle \frac{10001}{1000} - x_F \right\rangle^0$$

```
display(sol[A_v].rewrite(sym.Piecewise).simplify())
Av_inf = sol[A_v].rewrite(sym.Piecewise).simplify()
sym.plot(-Av_inf,(x_F,0,10));
```

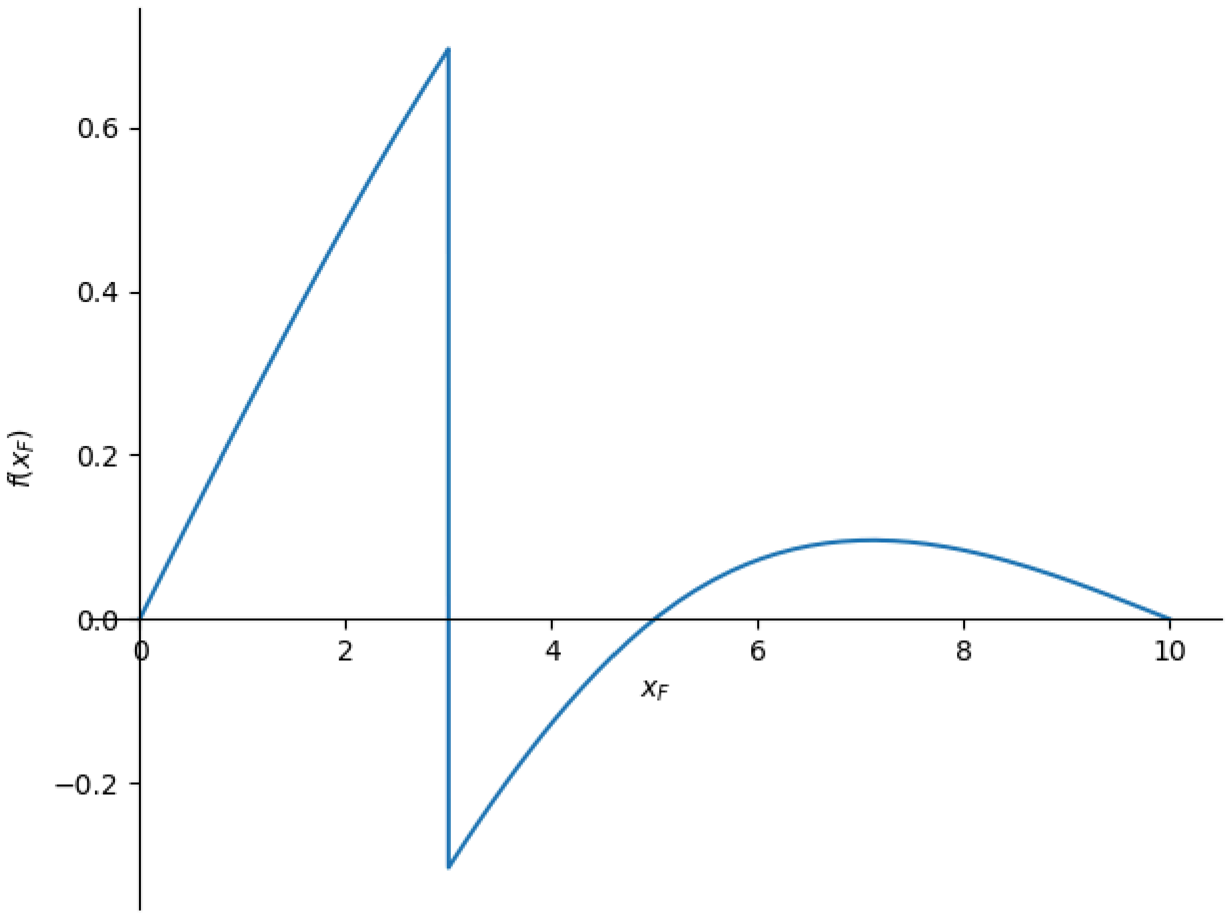
$$\begin{cases} \frac{x_F^3}{500} - \frac{x_F}{4} + 1 & \text{for } x_F < 5 \\ \frac{25x_F - (x_F - 10)^3 - 250}{500} & \text{for } x_F < 10 \\ 0 & \text{otherwise} \end{cases}$$




```
V_inf = V.subs(sol).subs(x,5-0.0001).rewrite(sym.Piecewise).simplify()
sym.plot(-V_inf,(x_F,0,10));
```

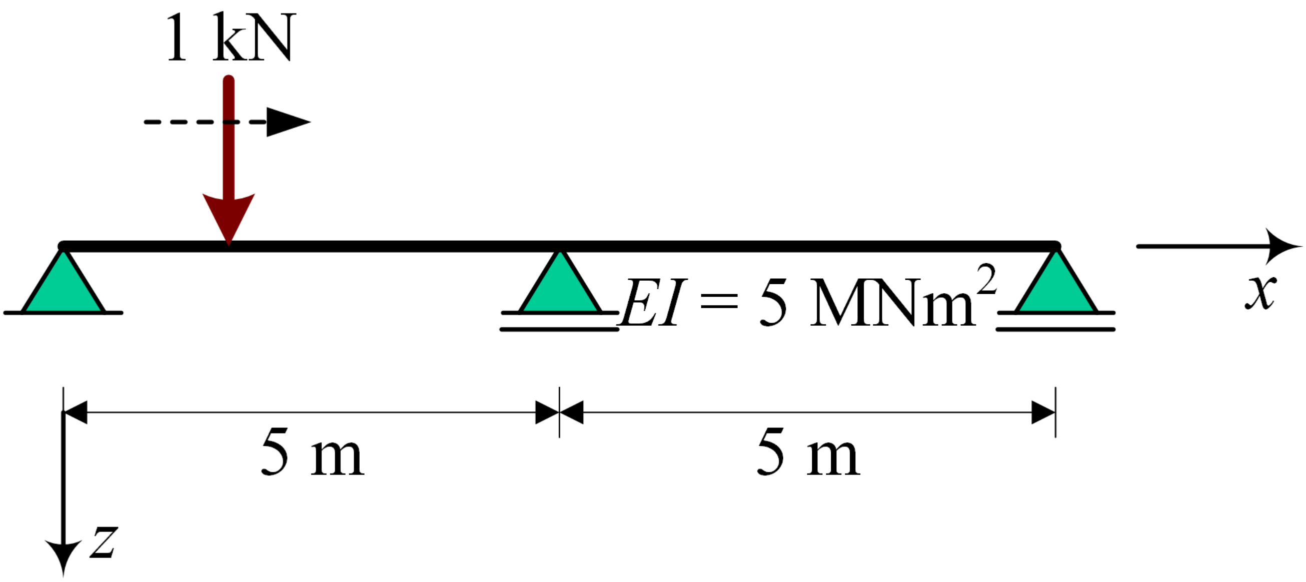



```
M_inf = V.subs(sol).subs(x,3).rewrite(sym.Piecewise).simplify()
sym.plot(-M_inf,(x_F,0,10));
```



Envelope of moments and influence lines

Contents



Find influence surface M for all load and

moment locations

Figure 1

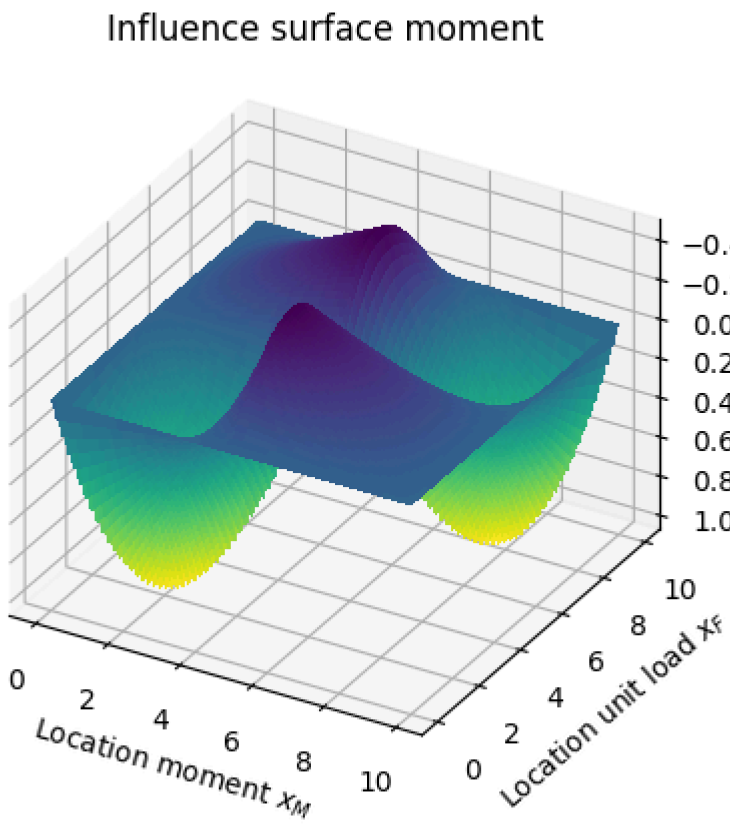
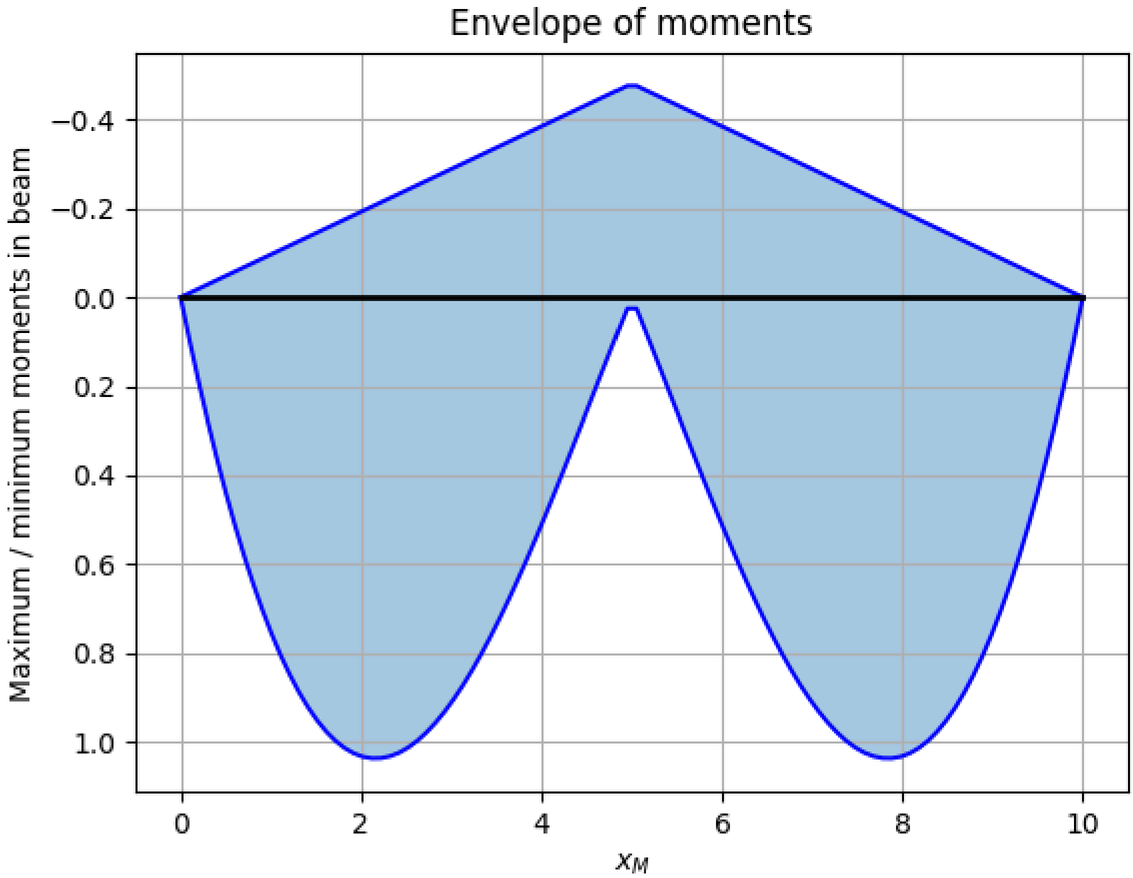
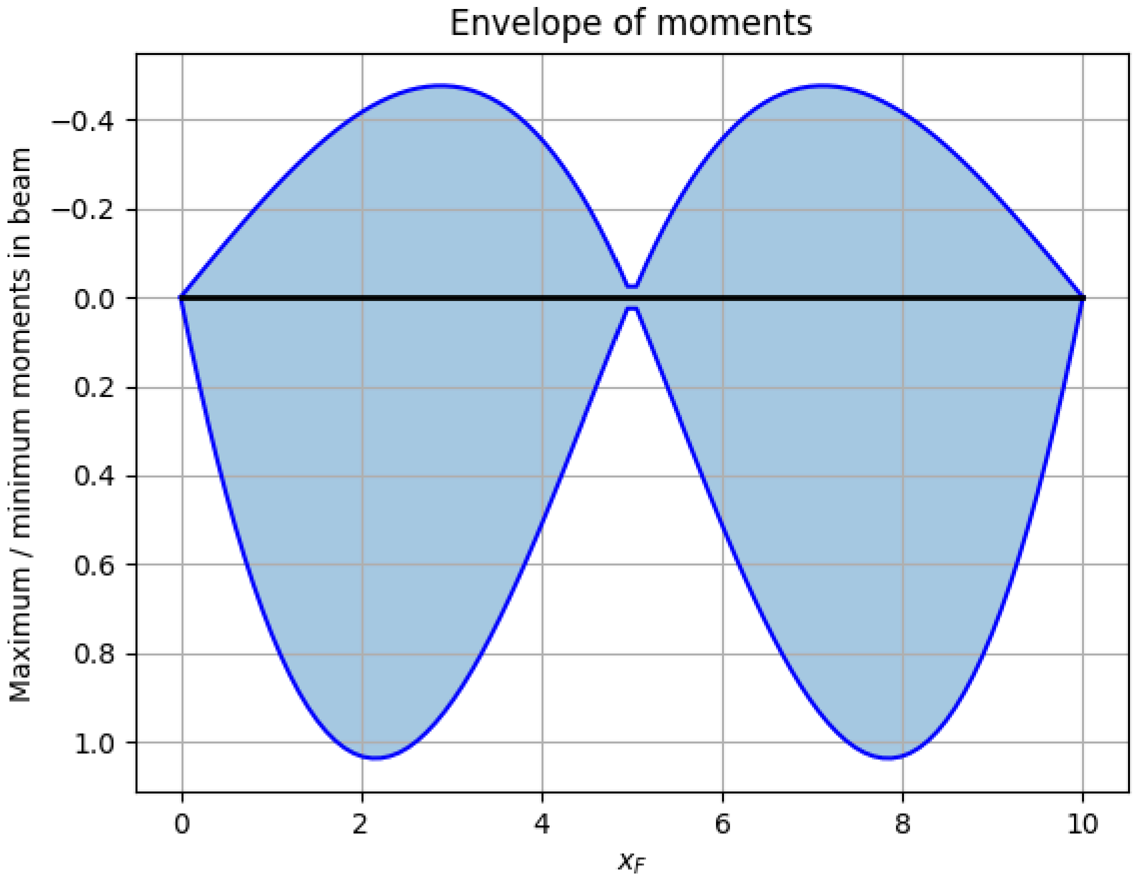


Figure 2



Envelope of moments
Maximum / minimum moments in beam
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
x_M

Figure 3



Envelope of moments
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Maximum / minimum moments in beam
0
2
4
6
8
10
x_F

Exercise statically indeterminate structures and displacements

Given is the following structure:

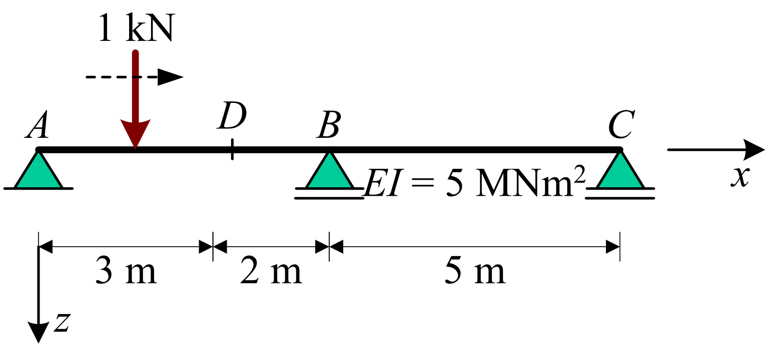


Fig. 2 Statically indeterminate structure with 3 supports

Exercise (Influence line support reaction at A)

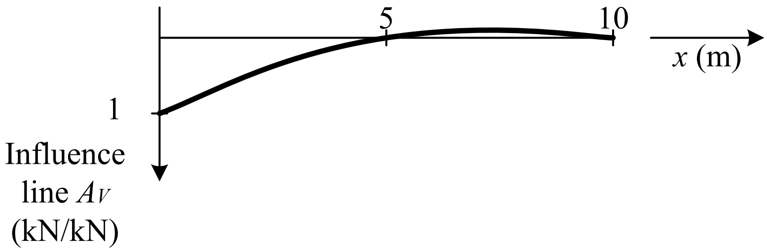
Draw the influence line for the support reaction at A, assuming a positive support reaction to point upwards. Only indicate the values for which no calculations are required.

Which parts of the influence line are curved with a positive curvature ◡?

- [ ] $0 < x < 5$
- [ ] $5 < x < 10$
- [ ] All the parts are not curved or have a negative curvature.

Check

Solution to Exercise (Influence line support reaction at A)



See an interactive plot of this influence line on this page (slow on phones)

Exercise (Influence line shear force just left of B)

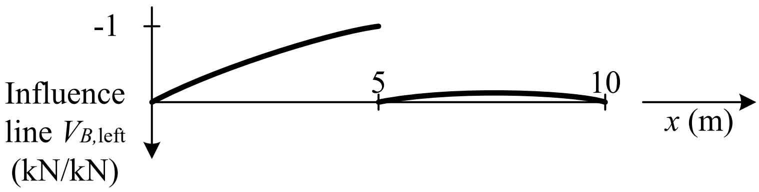
Draw the influence line for the shear force just left of B. Only indicate the values for which no calculations are required.

Which parts of the influence line are curved with a positive curvature ◡?

- [ ] $0 < x < 5$
- [ ] $5 < x < 10$
- [ ] All the parts are not curved or have a negative curvature.

Check

Solution to Exercise (Influence line shear force just left of B)



See an interactive plot of this influence line on this page (slow on phones)

Exercise (Influence line moment in B)

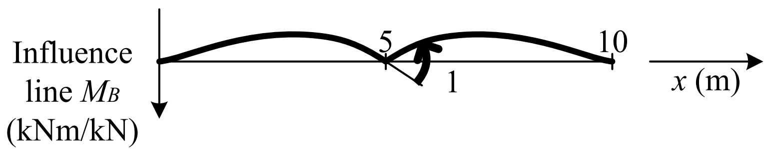
Draw the influence line for the moment in B. Only indicate the values for which no calculations are required.

Which parts of the influence line are curved with a positive curvature ◡?

- [ ] \(0 < x < 5\)
- [ ] \(5 < x < 10\)
- [ ] All the parts are not curved or have a negative curvature.

Check

Solution to Exercise (Influence line moment in B)



See an interactive plot of this influence line on this page (slow on phones)

Exercise (Influence line shear force in D)

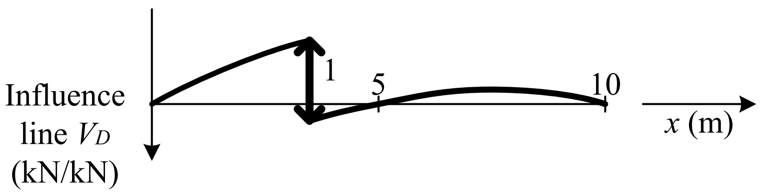
Draw the influence line for the shear force in D. Only indicate the values for which no calculations are required.

Which parts of the influence line are curved with a positive curvature ◡?

- [ ] \(0 < x < 5\)
- [ ] \(5 < x < 10\)
- [ ] All the parts are not curved or have a negative curvature.

Check

Solution to Exercise (Influence line shear force in D)



See an interactive plot of this influence line on this page (slow on phones)

Exercise (Influence line moment in D)

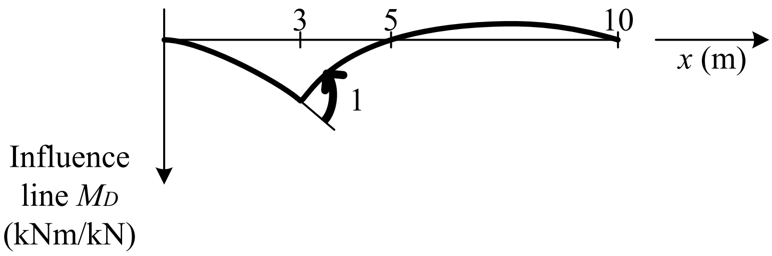
Draw the influence line for the moment in D. Only indicate the values for which no calculations are required.

Which parts of the influence line are curved with a positive curvature ◡?

- ☐ \(0 < x < 5\)
- ☐ \(5 < x < 10\)
- ☐ All the parts are not curved or have a negative curvature.

Check

Solution to Exercise (Influence line moment in D)



See an interactive plot of this influence line on this page (slow on phones)

Exercise (Influence line shear force just right of B)

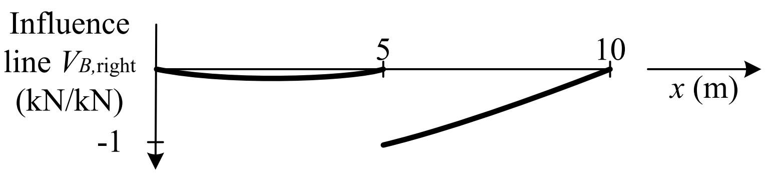
Draw the influence line for the shear force just right of B. Only indicate the values for which no calculations are required.

Which parts of the influence line are curved with a positive curvature ◡?

- [ ] $0 < x < 5$
- [ ] $5 < x < 10$
- [ ] All the parts are not curved or have a negative curvature.

Check

Solution to Exercise (Influence line shear force just right of B)



See an interactive plot of this influence line on this page (slow on phones)

Given is the following structure:

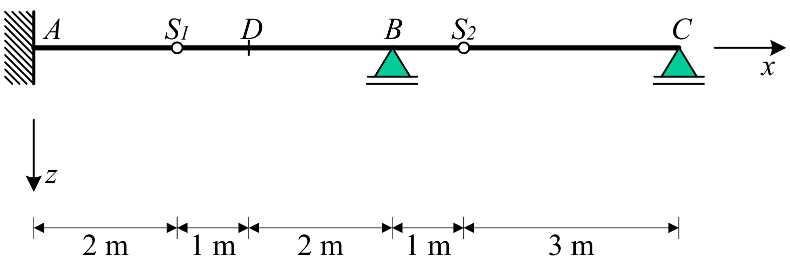


Fig. 3 Statically indeterminate structure with 3 supports

Exercise (Influence line displacement of D)

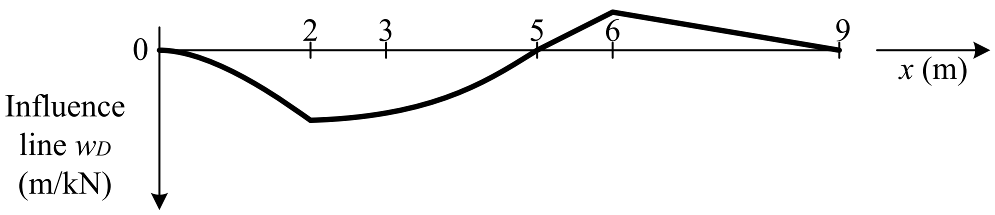
Draw the influence line for the displacement of D. Only indicate the values for which no calculations are required.

Which parts of the influence line are curved with a positive curvature ◡?

- [ ] $0 < x < 2$
- [ ] $2 < x < 3$
- [ ] $3 < x < 5$
- [ ] $5 < x < 6$
- [ ] $6 < x < 9$
- [ ] All the parts are not curved or have a negative curvature.

Check

Solution to Exercise (Influence line displacement of D)



Exercise (Influence line rotation of B)

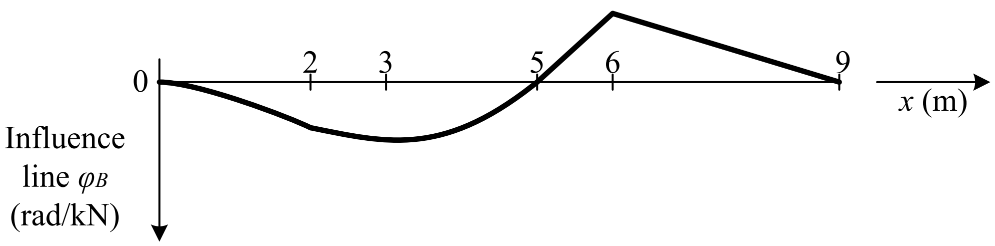
Draw the influence line for the rotation of B. Only indicate the values for which no calculations are required.

Which parts of the influence line are curved with a positive curvature ◡?

- [ ] \(0 < x < 2\)
- [ ] \(2 < x < 3\)
- [ ] \(3 < x < 5\)
- [ ] \(5 < x < 6\)
- [ ] \(6 < x < 9\)
- [ ] All the parts are not curved or have a negative curvature.

Check

Solution to Exercise (Influence line rotation of B)



Exercise (Influence line displacement of S2)

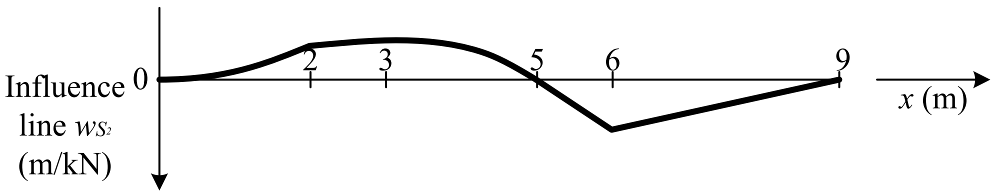
Draw the influence line for the displacement of S2. Only indicate the values for which no calculations are required.

Which parts of the influence line are curved with a positive curvature ◡?

- [ ] $0 < x < 2$
- [ ] $2 < x < 3$
- [ ] $3 < x < 5$
- [ ] $5 < x < 6$
- [ ] $6 < x < 9$
- [ ] All the parts are not curved or have a negative curvature.

Check

Solution to Exercise (Influence line displacement of S2)



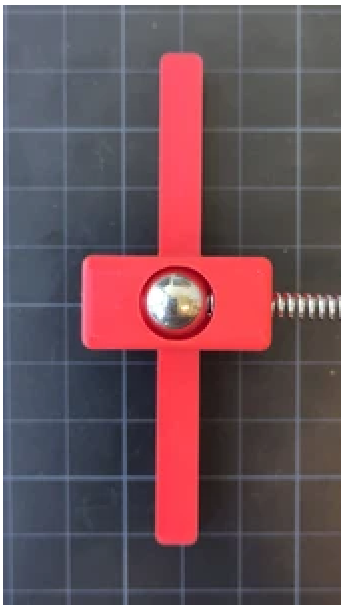
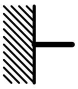
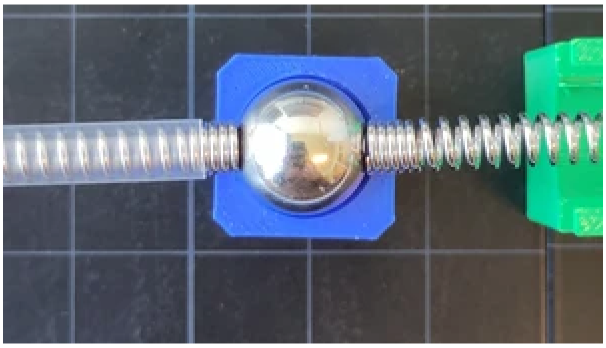
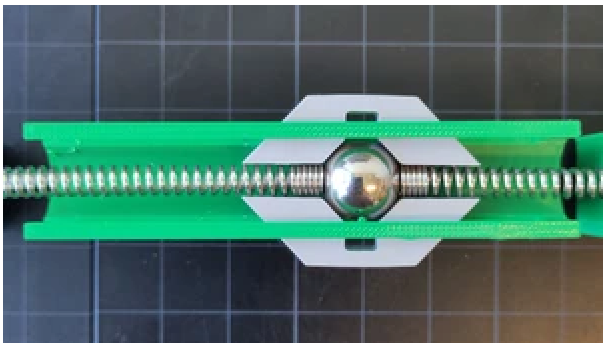
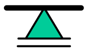
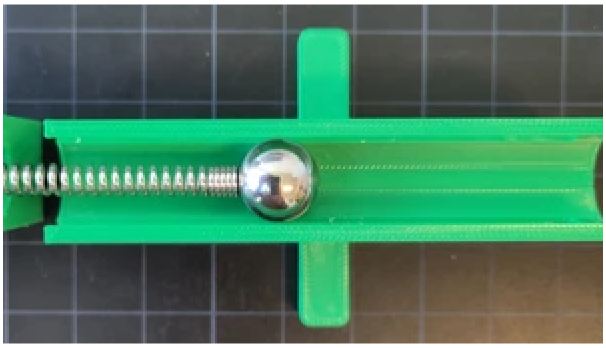
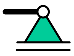
MOLA exercise

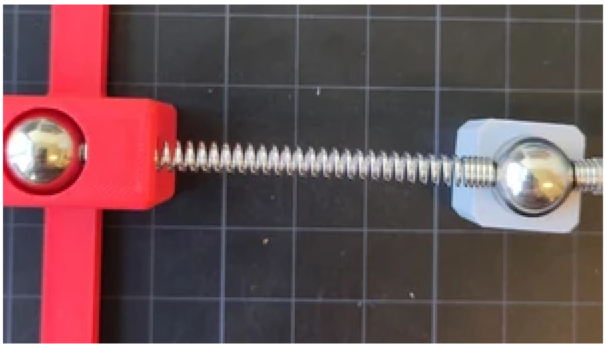
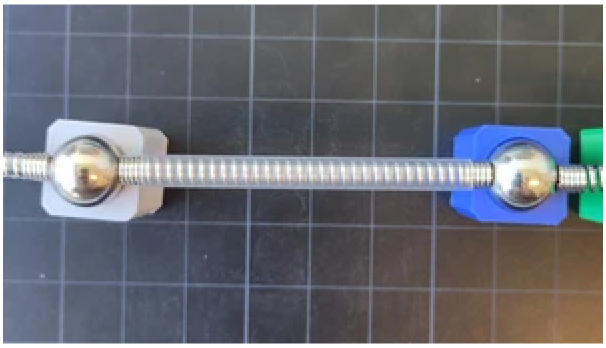
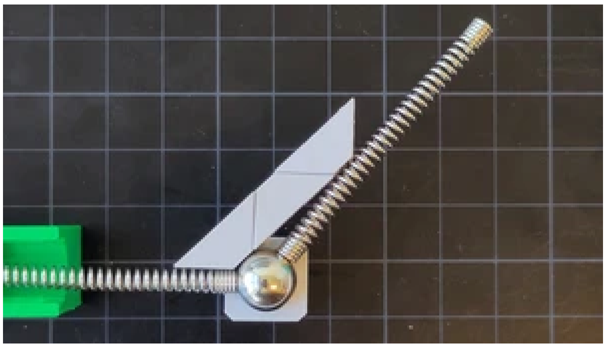
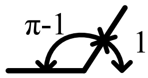
Contents

In this exercise, you're going to find influence lines using MOLA. It's meant to give you insight in when these influence lines have straight and/or curved parts.

Components

We'll use the following components:

MOLA	Model
	
	
	
	

MOLA	Model
	
	EI = ∞
	

Simply supported beam

Let’s start with the most basic model, a simply supported beam:

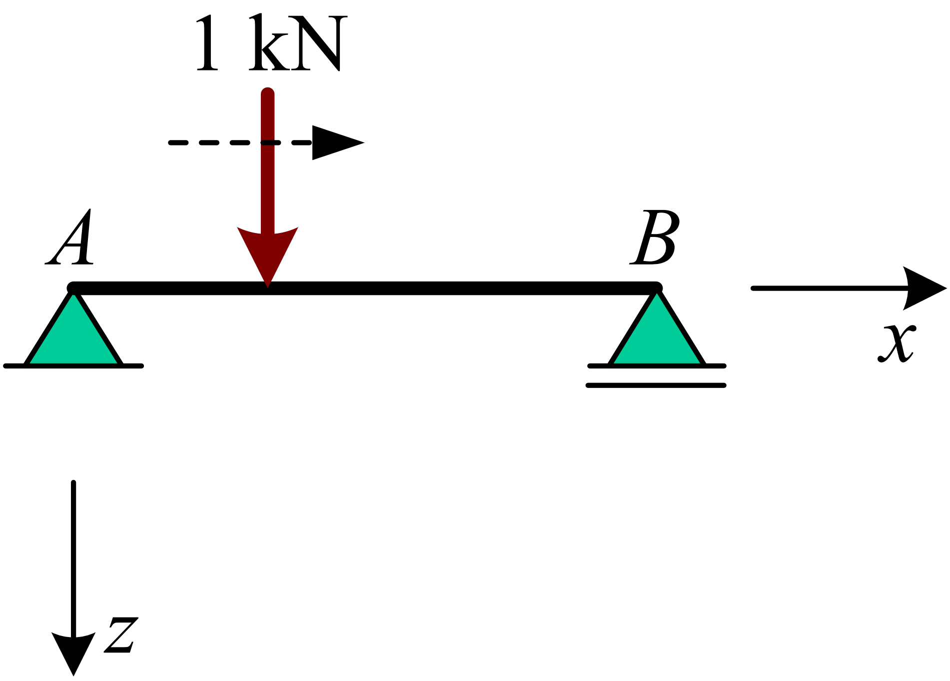
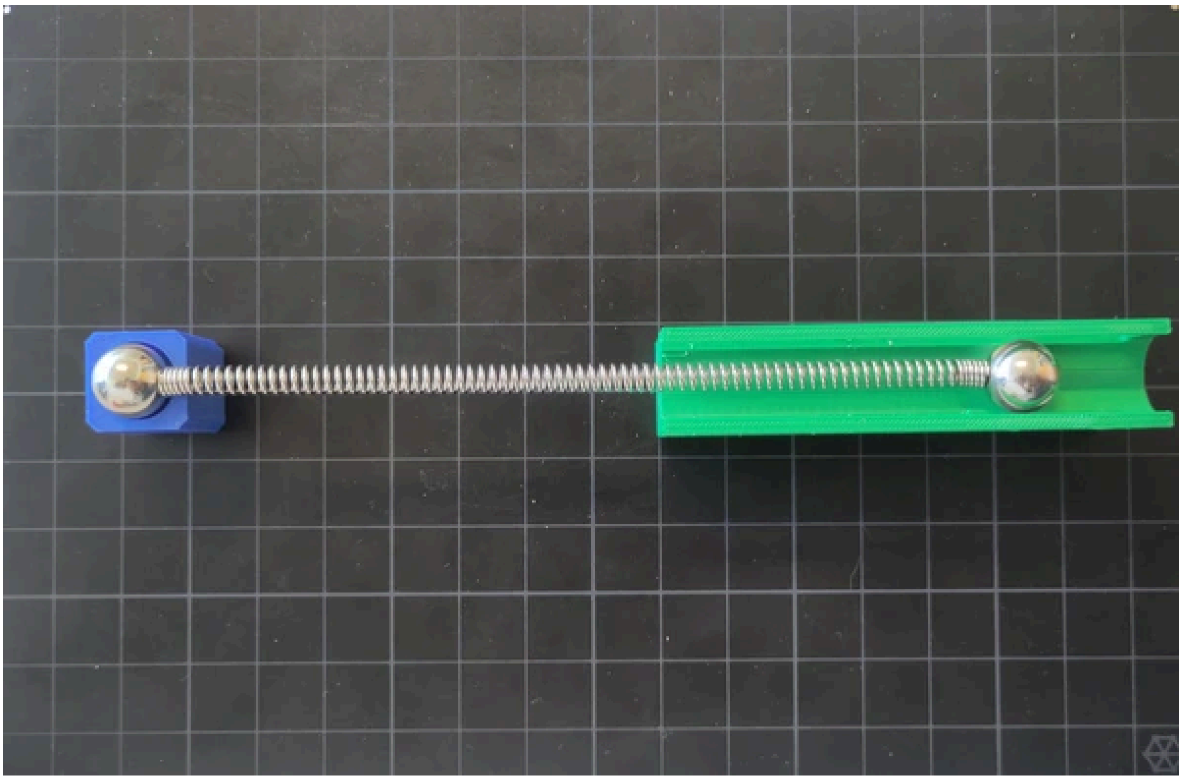


Fig. 4 Simply supported beam

Exercise (Simply supported beam)

Make the simply supported beam with MOLA

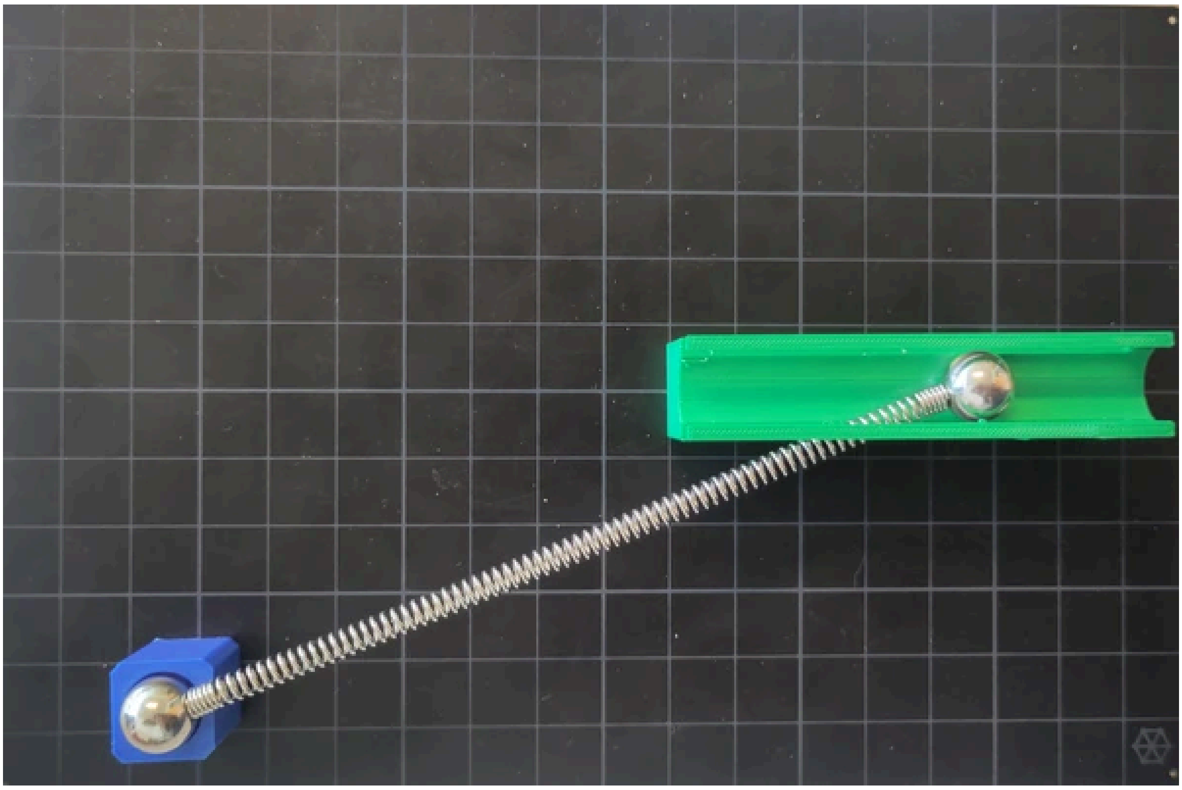
Solution to Exercise (Simply supported beam)



Exercise (Influence line vertical support reaction at A for simply supported beam)

Show the influence line of the vertical support reaction at A

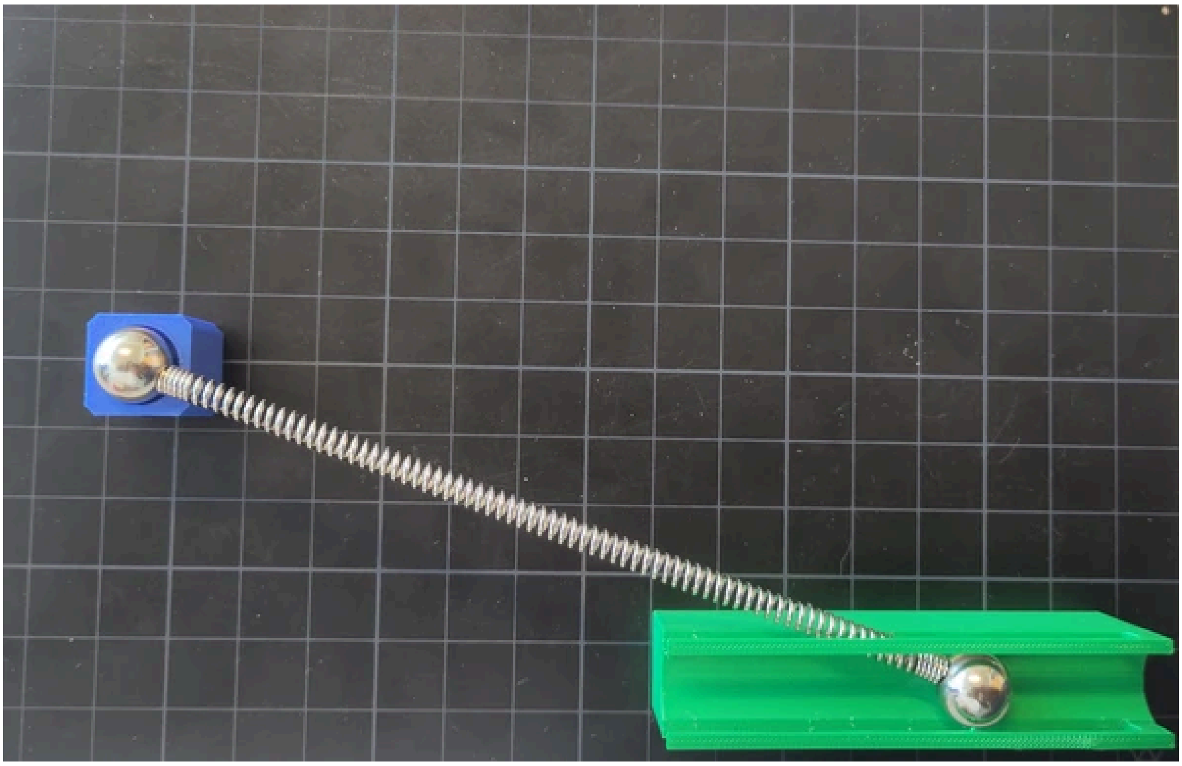
Solution to [Exercise (Influence line vertical support reaction at A for simply supported beam)]



Exercise (Influence line vertical support reaction at B for simply supported beam)

Show the influence line of the vertical support reaction at B

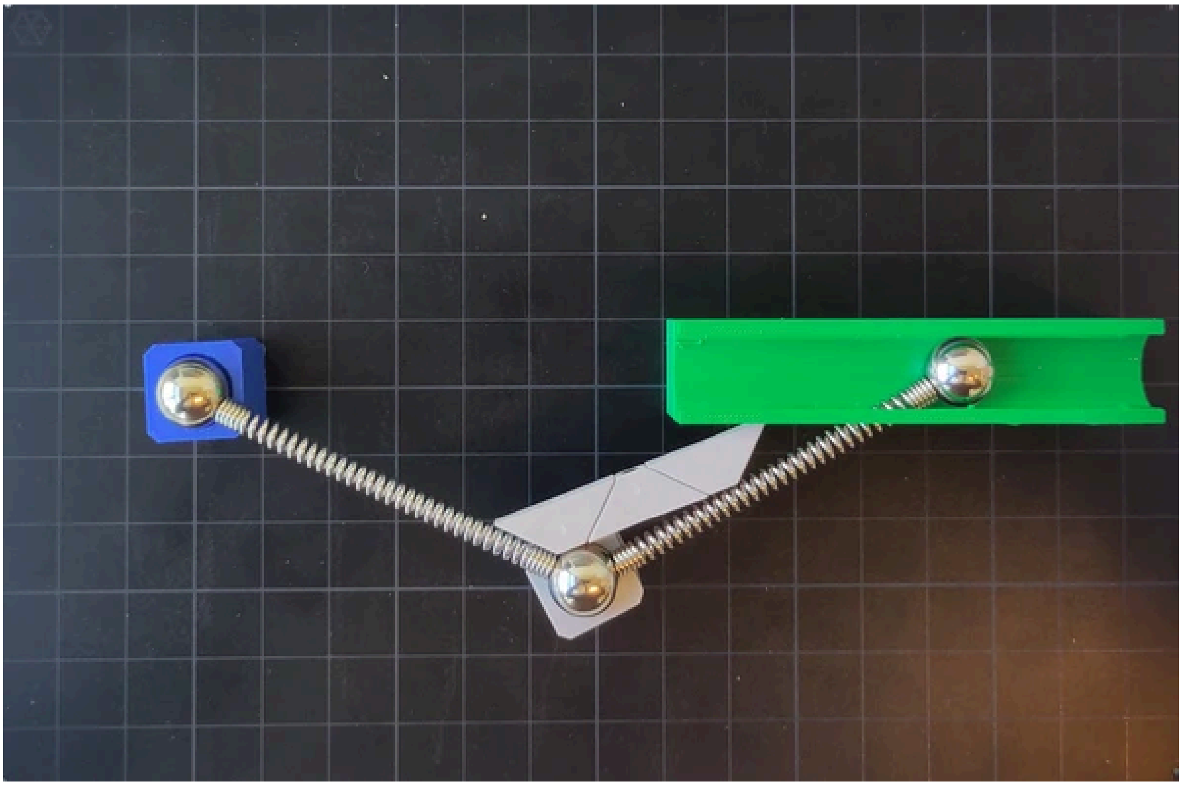
Solution to [Exercise (Influence line vertical support reaction at B for simply supported beam)]



Exercise (Influence line bending moment for simply supported beam)

Show the influence line for the bending moment halfway the beam

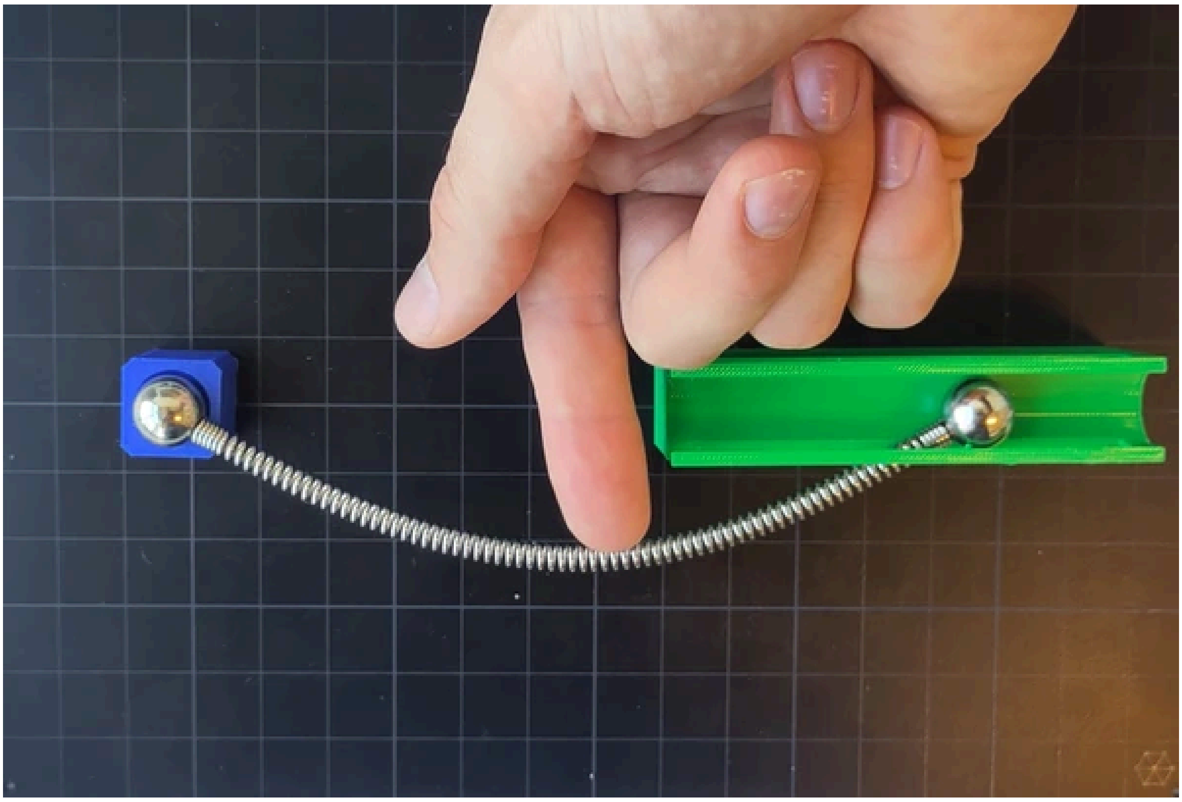
Solution to [Exercise (Influence line bending moment for simply supported beam)]



Exercise (Influence line displacement for simply supported beam)

Show the influence line for the displacement halfway the beam

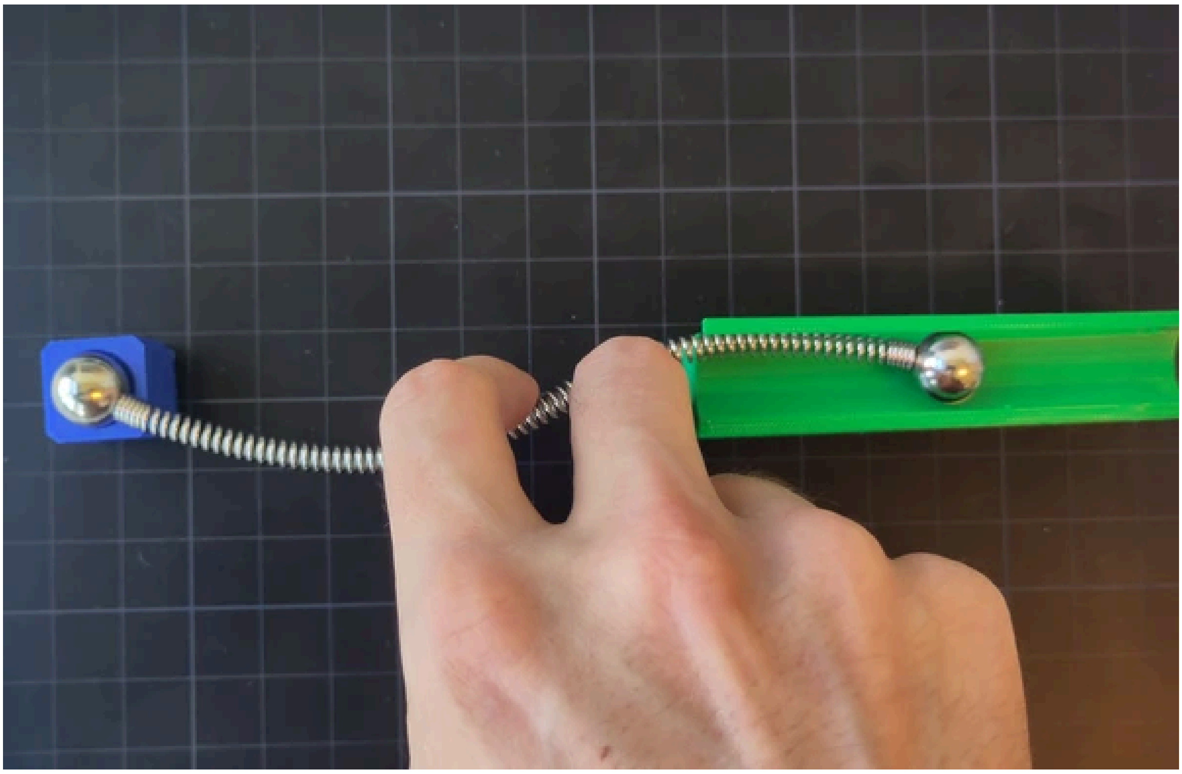
Solution to Exercise (Influence line displacement for simply supported beam)



Exercise (Influence line rotation for simply supported beam)

Show the influence line for the rotation halfway the beam

Solution to [Exercise (Influence line rotation for simply supported beam)]



Statically determinate hinged beam

Let’s increase complexity a bit, by looking at a statically determinate hinged beam:

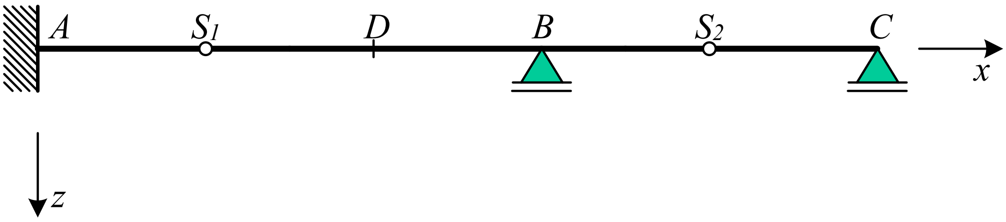
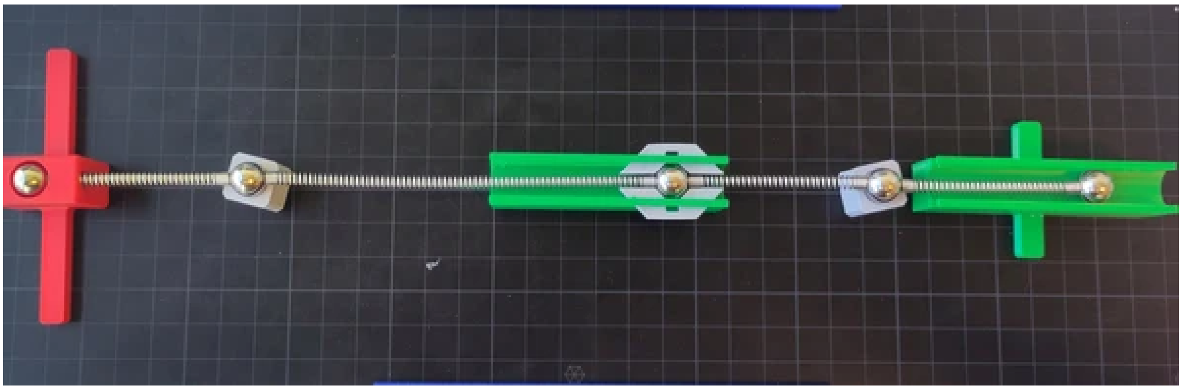


Fig. 5 Statically determinate hinged beam

Exercise (Statically determinate hinged beam)

Make the statically determinate hinged beam with MOLA

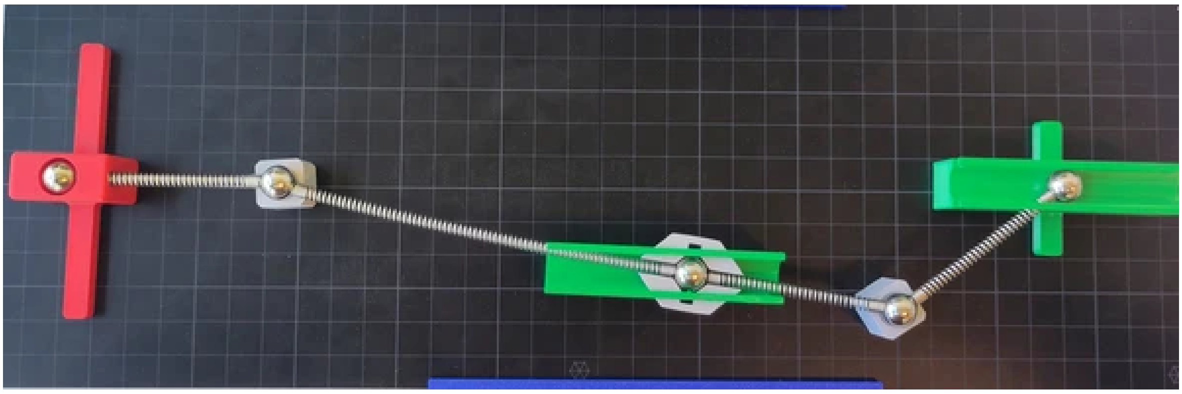
Solution to Exercise (Statically determinate hinged beam)



Exercise (Influence line vertical support reaction at B for the statically determinate hinged beam)

Show the influence line of the vertical support reaction at B

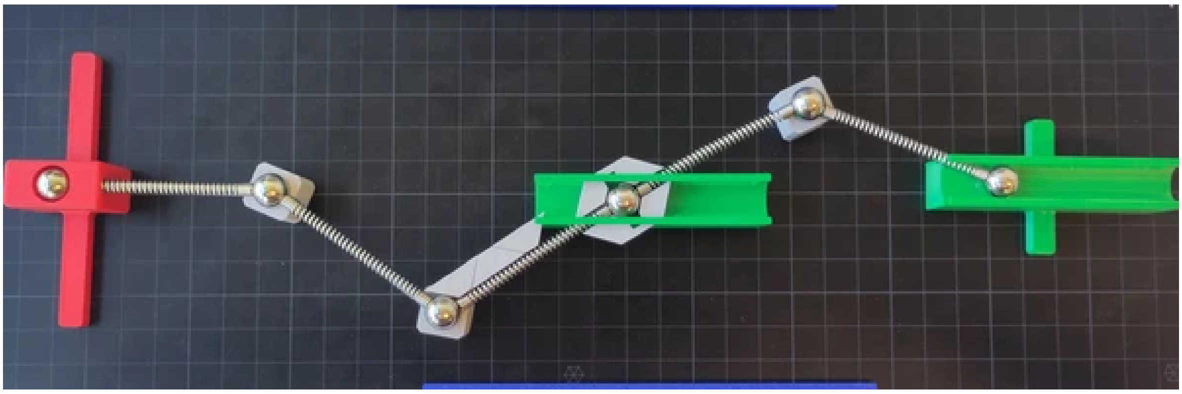
Solution to Exercise (Influence line vertical support reaction at B for the statically determinate hinged beam)



Exercise (Influence line bending moment at D for the statically determinate hinged beam)

Show the influence line for the bending moment at D for the statically determinate hinged beam

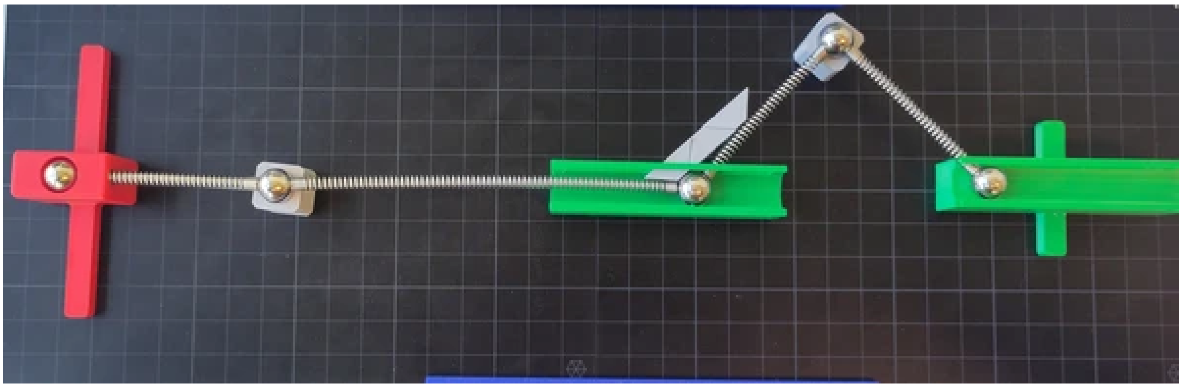
Solution to [Exercise (Influence line bending moment at D for the statically determinate hinged beam)]



Exercise (Influence line bending moment at B for the statically determinate hinged beam)

Show the influence line for the bending moment at B for the statically determinate hinged beam

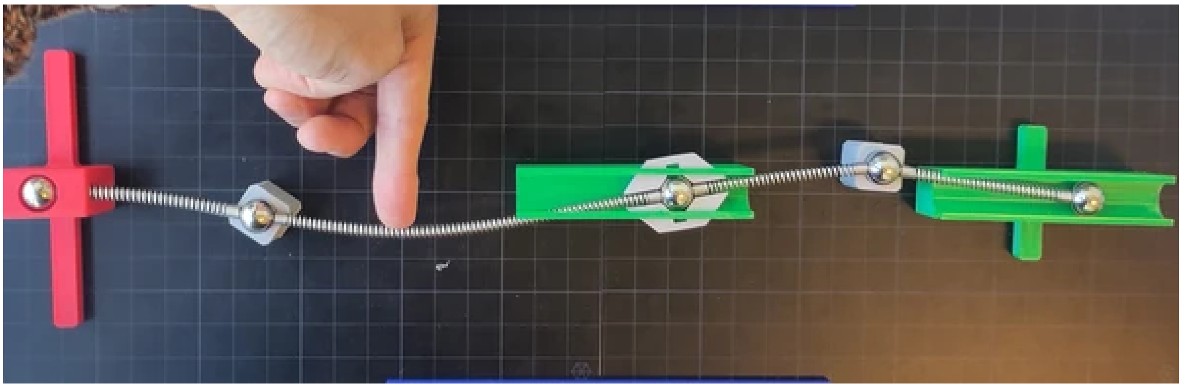
Solution to [Exercise (Influence line bending moment at B for the statically determinate hinged beam)]



Exercise (Influence line displacement in D for the statically determinate hinged beam)

Show the influence line for the displacement in D for the statically determinate hinged beam

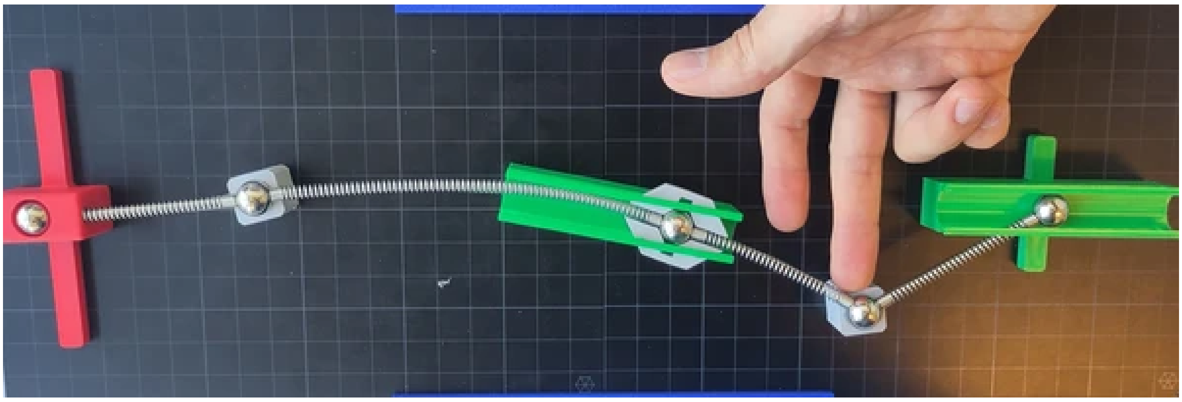
Solution to [Exercise (Influence line displacement in D for the statically determinate hinged beam)]



Exercise (Influence line displacement in S2 for the statically determinate hinged beam)

Show the influence line for the displacement in S2 for the statically determinate hinged beam

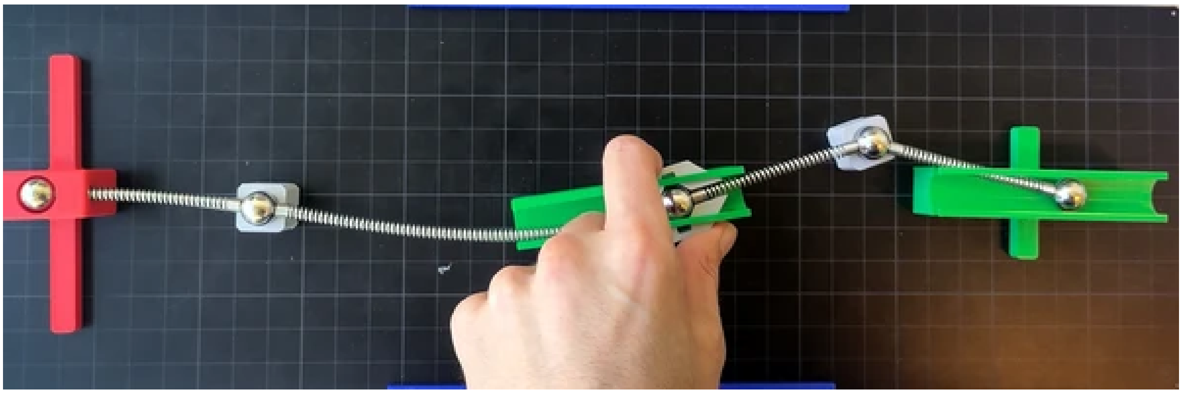
Solution to [Exercise (Influence line displacement in S2 for the statically determinate hinged beam)]



Exercise (Influence line rotation in B for the statically determinate hinged beam)

Show the influence line for the rotation in B for the statically determinate hinged beam

Solution to [Exercise (Influence line rotation in B for the statically determinate hinged beam)]



Statically indeterminate hinged beam with stiff part

Let’s consider a statically indeterminate beam with an infinitive stiff part:

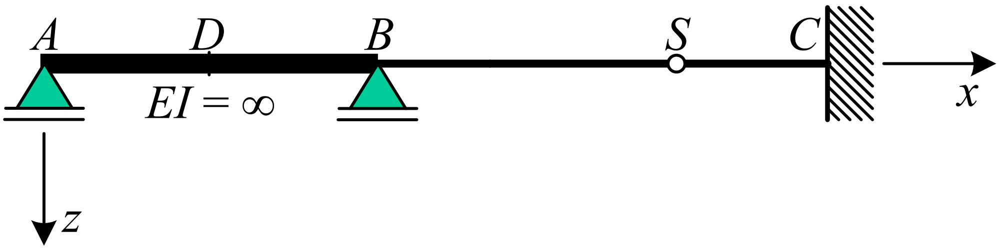
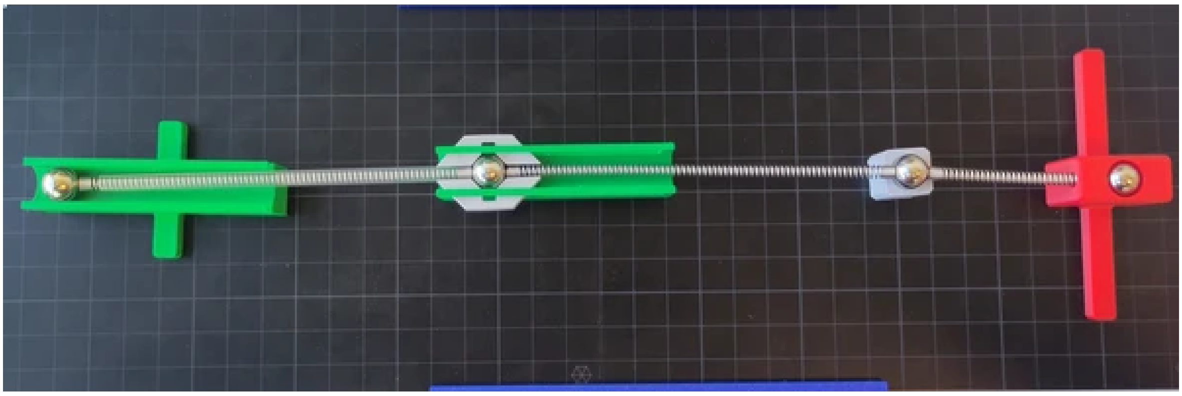


Fig. 6 Statically indeterminate hinged beam with stiff part

Exercise (Statically indeterminate beam with stiff part)

Make the statically indeterminate beam with stiff part with MOLA

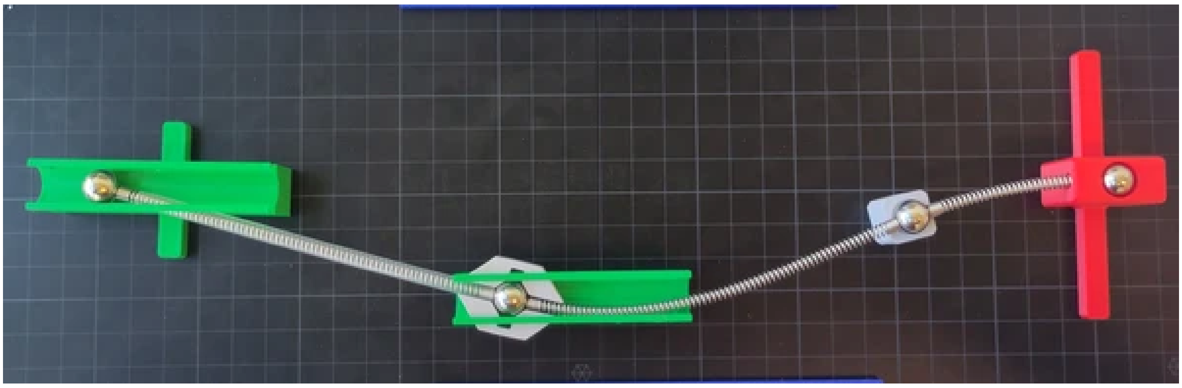
Solution to [Exercise (Statically indeterminate beam with stiff part)]



Exercise (Influence line vertical support reaction at B for the statically indeterminate beam with stiff part)

Show the influence line of the vertical support reaction at B for the statically indeterminate beam with stiff part

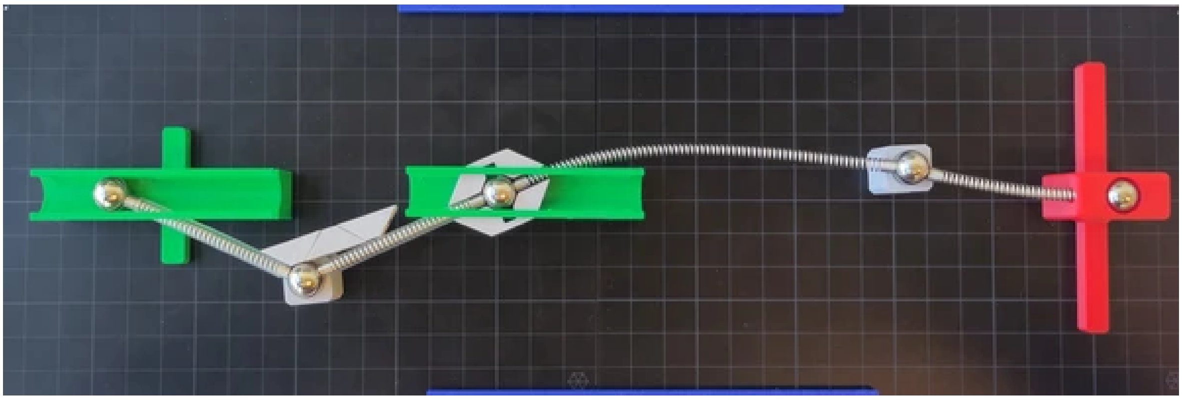
Solution to [Exercise (Influence line vertical support reaction at B for the statically indeterminate beam with stiff part)]



Exercise (Influence line bending moment halfway AB for the statically indeterminate hinged beam with stiff part)

Show the influence line for the bending moment halfway AB for the statically indeterminate hinged beam with stiff part

Solution to [Exercise (Influence line bending moment halfway AB for the statically indeterminate hinged beam with stiff part)]



Statically indeterminate hinged beam

Finally, let's consider another statically indeterminate hinged beam.

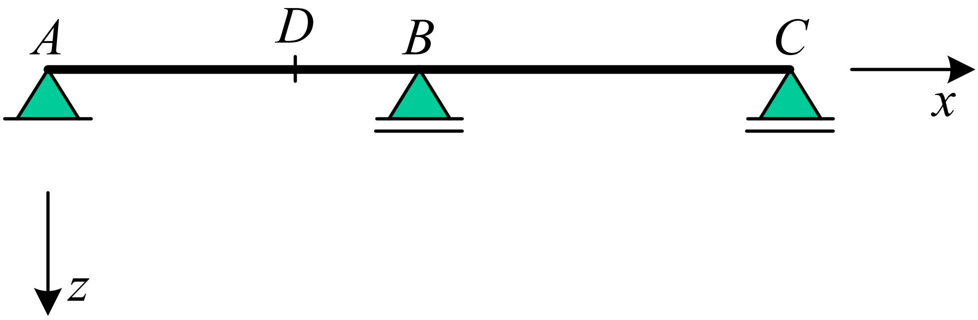
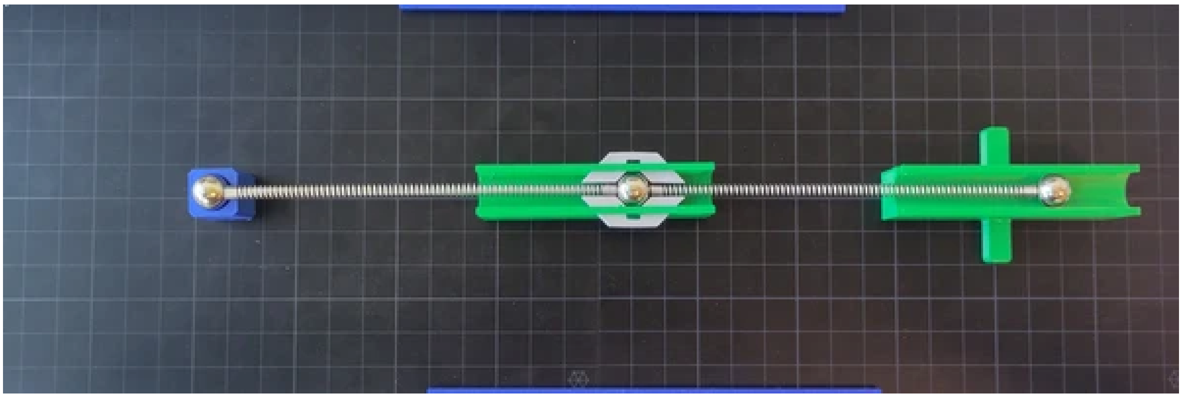


Fig. 7 Statically indeterminate hinged beam

Exercise (Statically indeterminate beam)

Make the statically indeterminate beam with MOLA

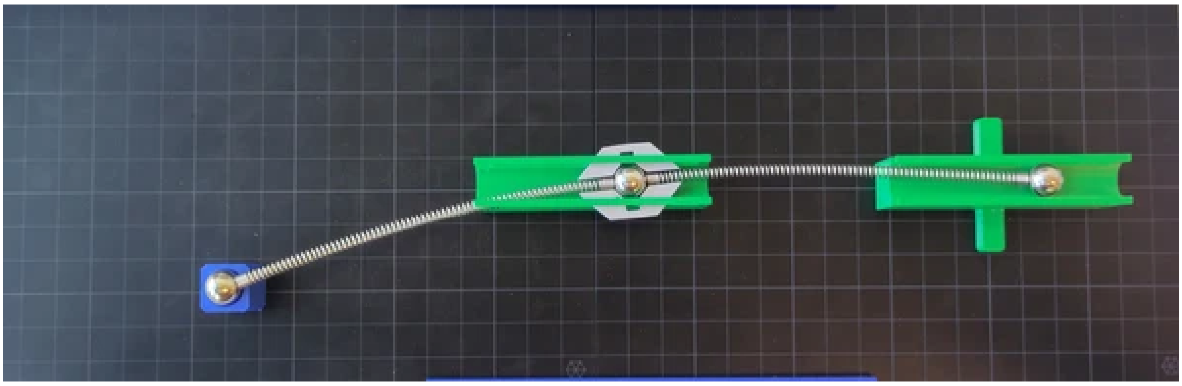
Solution to Exercise (Statically indeterminate beam)



Exercise (Influence line vertical support reaction at A for the statically indeterminate beam)

Show the influence line of the vertical support reaction at A for the statically indeterminate beam

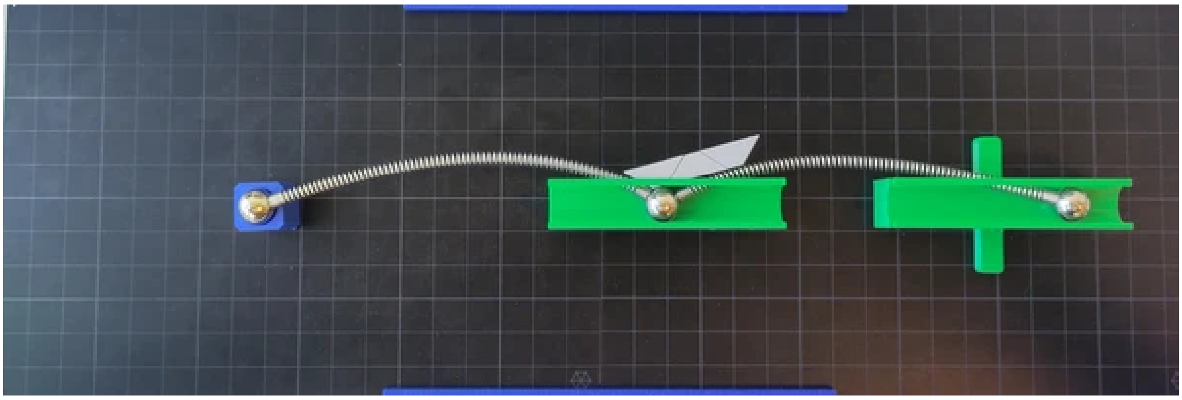
Solution to Exercise (Influence line vertical support reaction at A for the statically indeterminate beam)



Exercise (Influence line bending moment at B for the statically indeterminate hinged beam)

Show the influence line for the bending moment at B for the statically indeterminate hinged beam

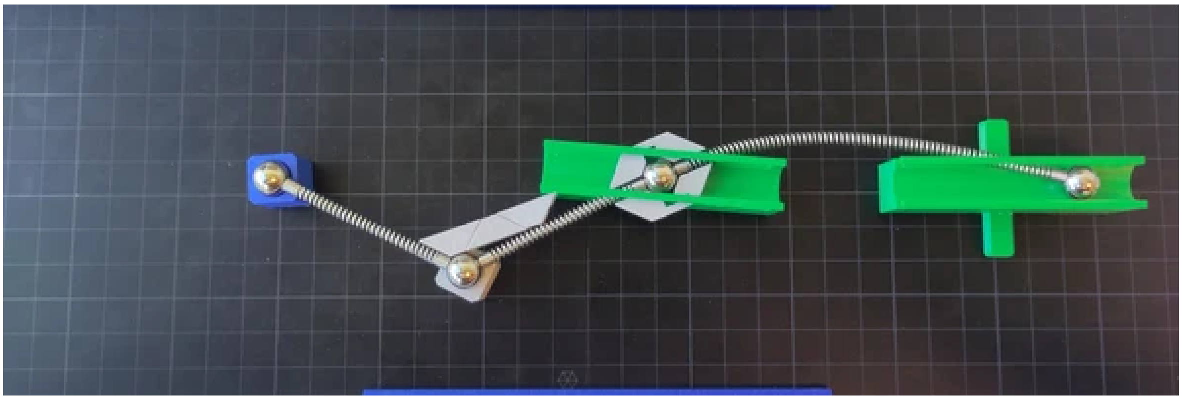
Solution to Exercise (Influence line bending moment at B for the statically indeterminate hinged beam)



Exercise (Influence line bending moment at D for the statically indeterminate hinged beam)

Show the influence line for the bending moment at D for the statically indeterminate hinged beam

Solution to Exercise (Influence line bending moment at D for the statically indeterminate hinged beam)

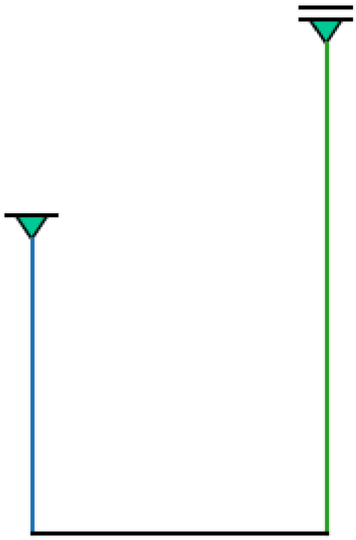


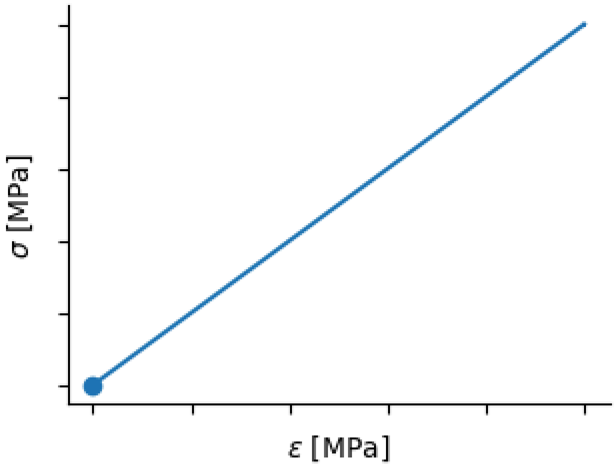
Extension elements statically determinate 1

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some plasticity!

w: 0.00

Figure 1





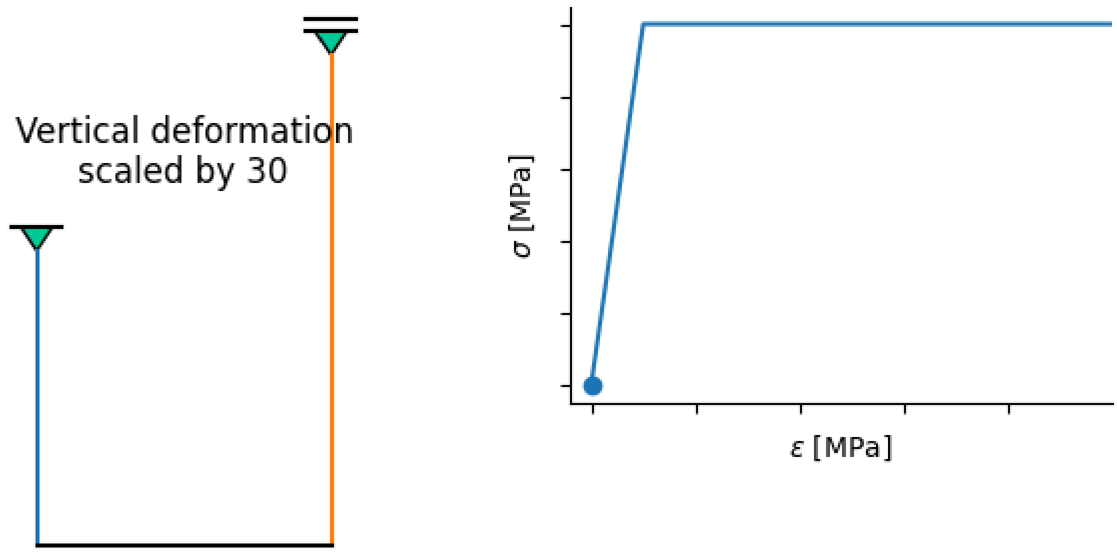
x=2.87 y=8.92

Extension elements statically determinate 2

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some plasticity!

w: 0.00

Figure 1

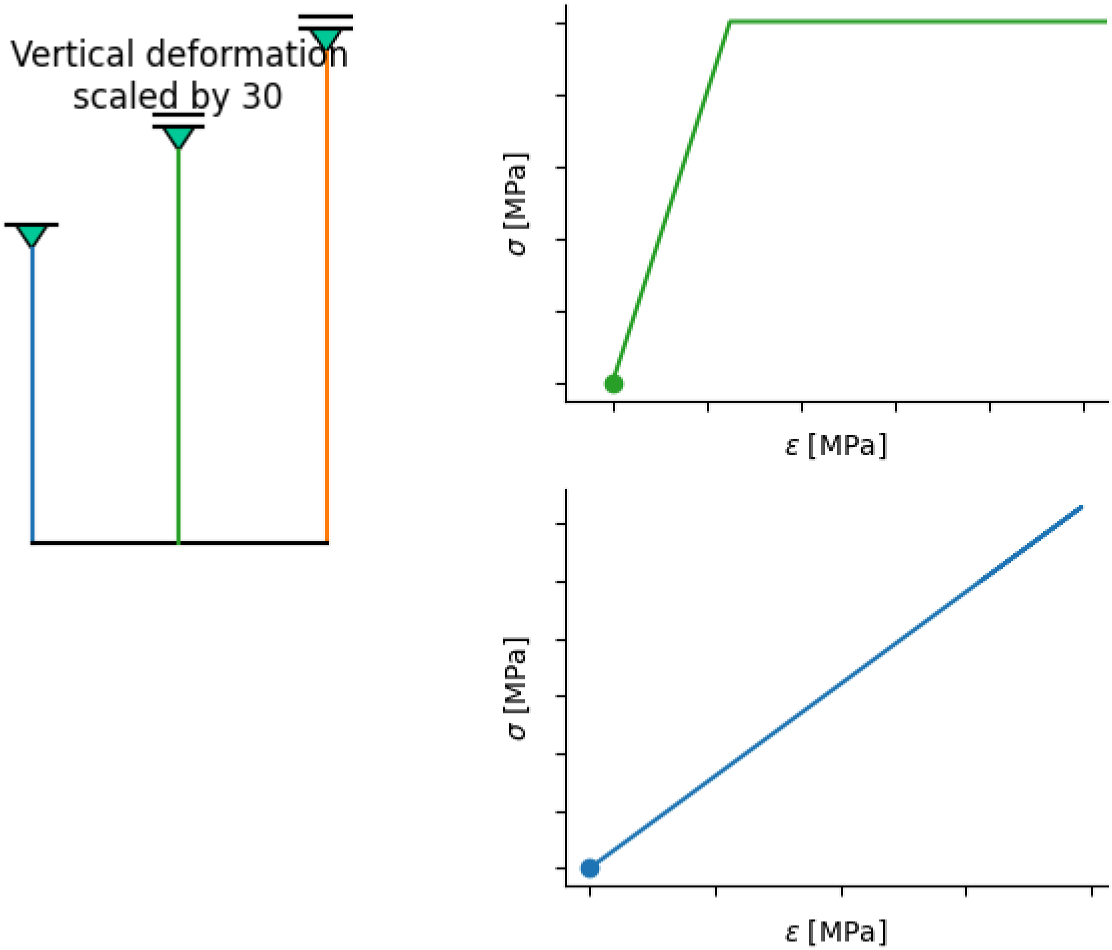


Extension elements statically indeterminate

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some plasticity!

w: 0.00

Figure 1

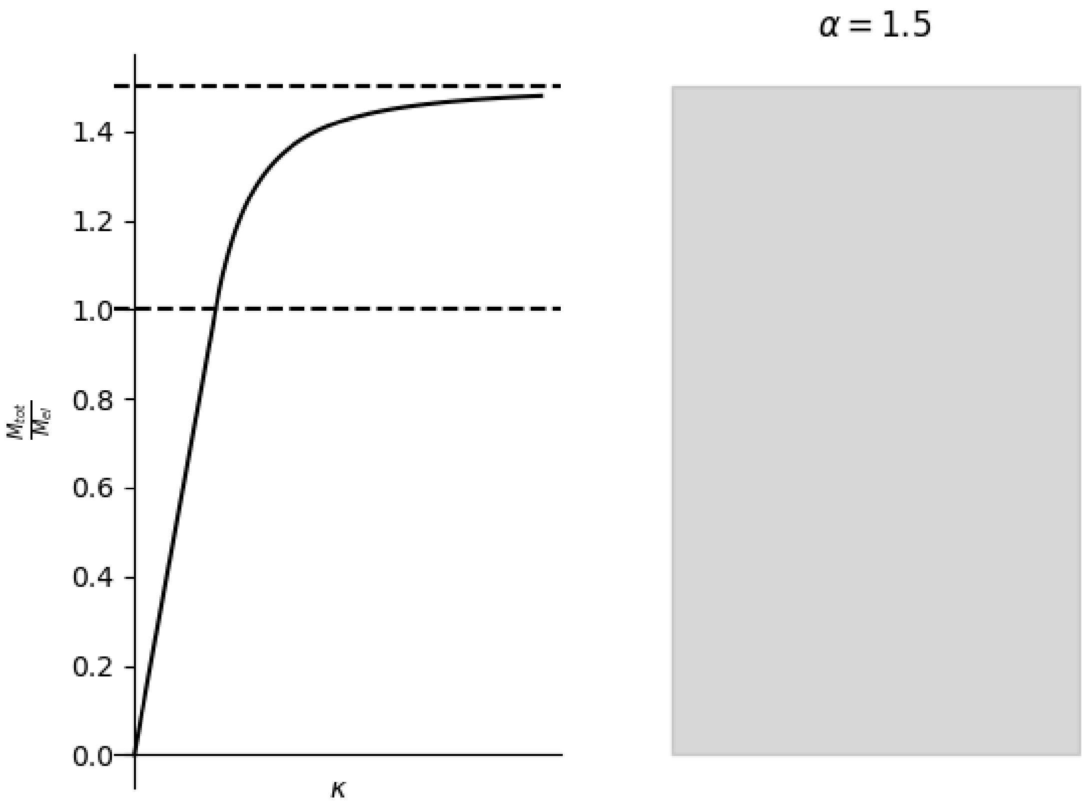


Shape factor and M-\kappa curve

Click 🚀 –> **Live Code** on the top right corner of this screen to investigate some plasticity!

Gap in cros… 0.00

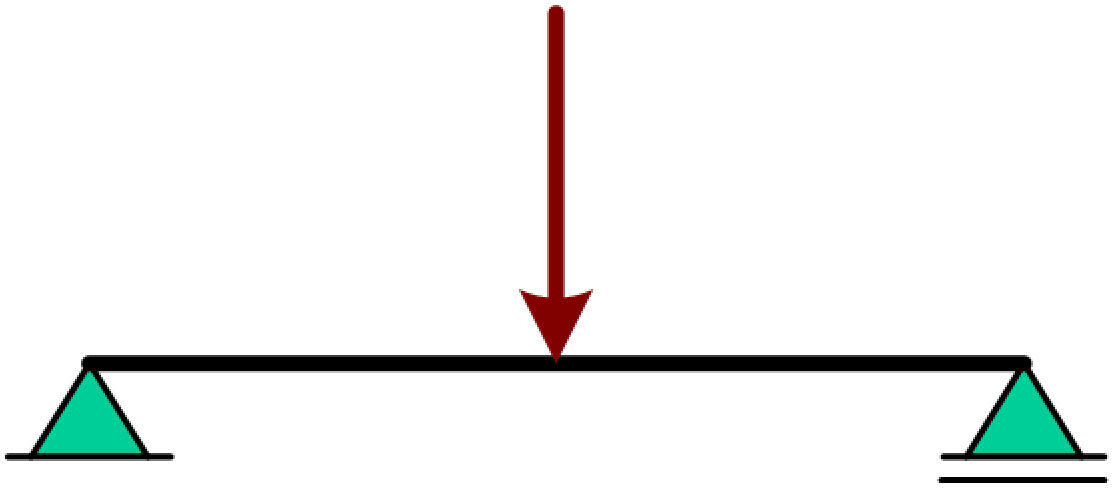
Figure 1



Plastic hinges

Click 🚀 –> Live Code on the top right corner of this screen to investigate some plasticity!

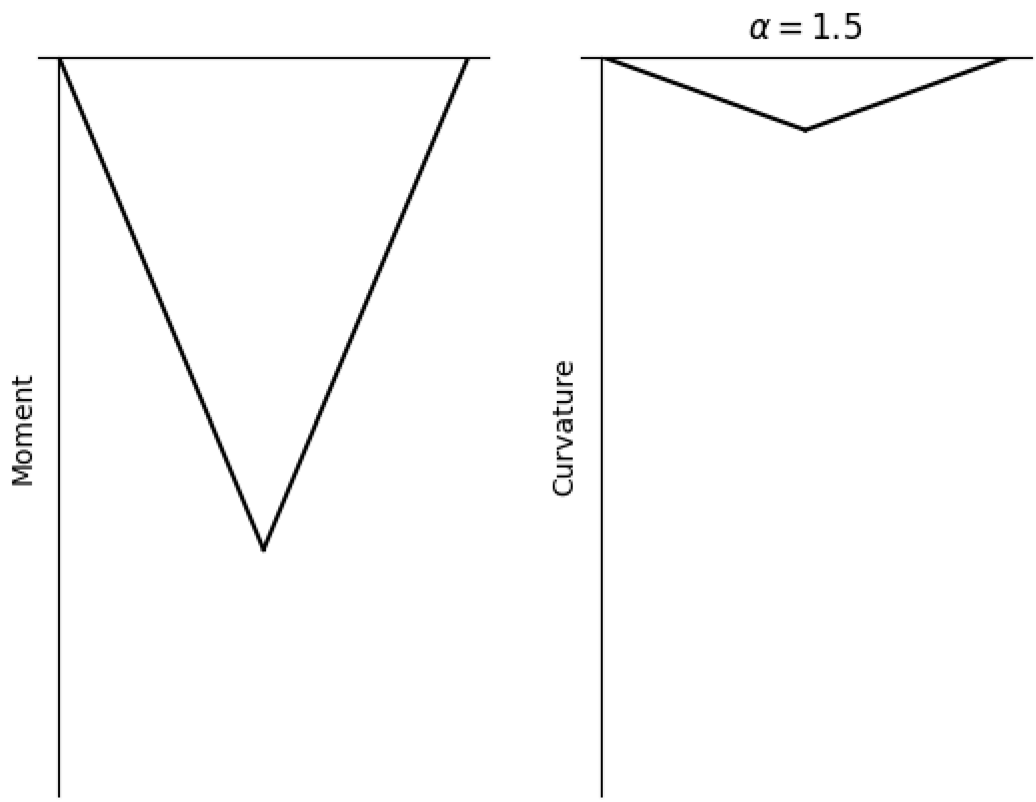
We're investigating the plastic behaviour of the following structure:



Load 0.00

Gap (m) in … 0.00

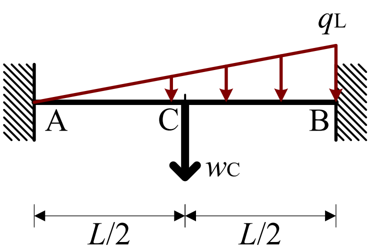
Figure 1



Incremental method

Contents

The following structure is investigated




```
#import packages
import sympy as sym
import matplotlib.pyplot as plt
sym.init_printing()
```

```
#Defines symbols and functions
EI, q_L, x, L, M_p = sym.symbols('EI, q_L, x, L, M_p')
w = sym.Function('w')
```

```
# Define the ODE for the bending of the beam
ODE_bending = sym.Eq(w(x).diff(x, 4) *EI, q_L/L*x)
display(ODE_bending)
```

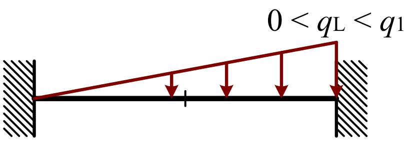
$$EI\frac{d^4}{dx^4}w(x) = \frac{q_L x}{L}$$

```
# Solve the ODE
w = sym.dsolve(ODE_bending, w(x)).rhs
display(w)
```

$$C_1 + C_2x + C_3x^2 + C_4x^3 + \frac{q_Lx^5}{120EIL}$$

```
# Define the continuum fields of phi, kappa, M, and V
phi = -w.diff(x)
kappa = phi.diff(x)
M = EI * kappa
V = M.diff(x)
```

Linear behaviour: fixed beam at both ends



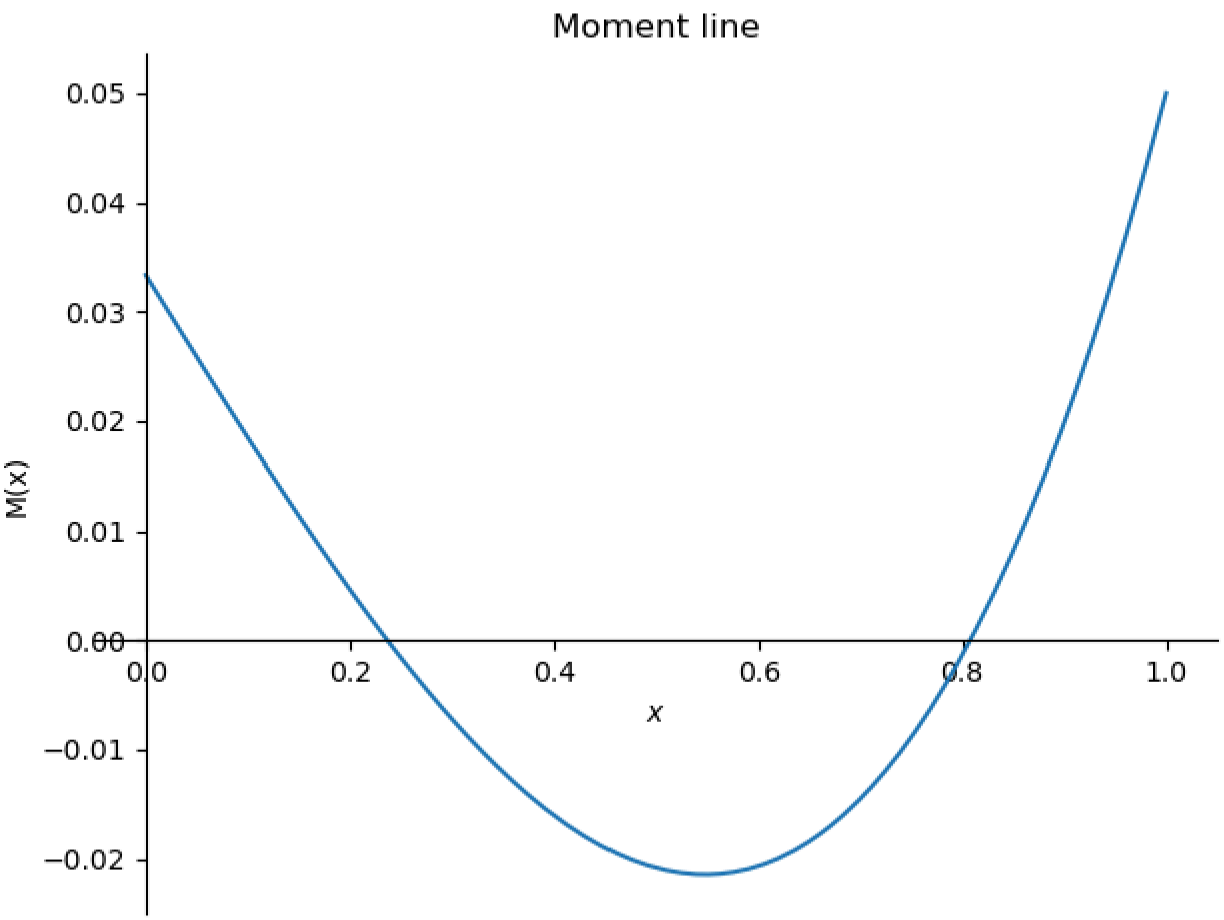

```
# Define the boundary conditions
eq1 = sym.Eq(w.subs(x,0),0)
eq2 = sym.Eq(w.subs(x,L),0)
eq3 = sym.Eq(phi.subs(x,0),0)
eq4 = sym.Eq(phi.subs(x,L),0)
```

```
# Solve the integration constants
sol1 = sym.solve([eq1, eq2, eq3, eq4 ], sym.symbols('C1, C2, C3, C4'))
display(sol1)
```

$$\left\{C_1 : 0,\ C_2 : 0,\ C_3 : \frac{L^2q_L}{60EI},\ C_4 : -\frac{Lq_L}{40EI}\right\}$$

```
# Define some dummy values to plot the results
plotvalues = {EI:1, q_L:1, L:1, M_p:0.01}
```

```
# Plot the moment line to identify the maximum moment
sym.plot(-M.subs(sol1).subs(plotvalues),(x,0,1),ylabel='M(x)',title='Moment line');
```



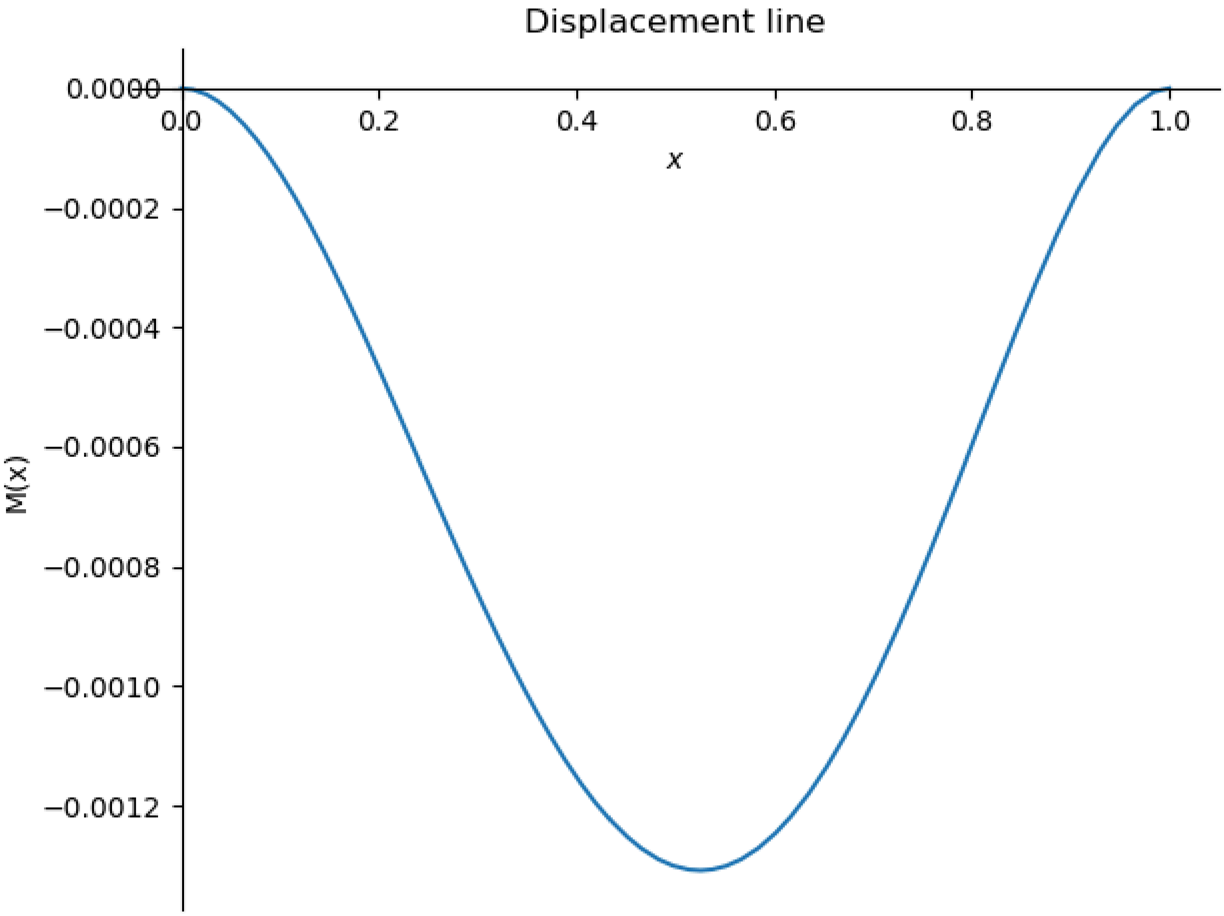

```
# Obtain the maximum moment at x = L
display(M.subs(sol1).subs(x,L))
```

$$-\frac{L^2 q_L}{20}$$

```
# Obtain the load corresponding to the first yield moment
q1 = sym.solve(sym.Eq(-M.subs(sol1).subs(x,L),M_p), q_L)[0]
display(q1)
```

$$\frac{20 M_p}{L^2}$$

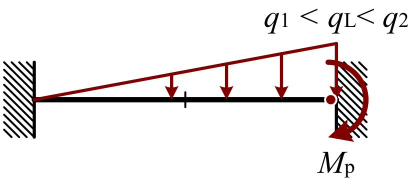
```
# Plot the displacement line
sym.plot(-w.subs(sol1).subs(plotvalues),(x,0,1),ylabel='M(x)',title='Displacement line');
```




```
# Obtain the displacement at x = L/2
w1 = w.subs(sol1).subs(q_L, q1).subs(x,L/2)
display(w1)
```

$$\frac{5L^2 M_p}{192EI}$$

Plastic behaviour: Beam simply supported on right with M_p working at it

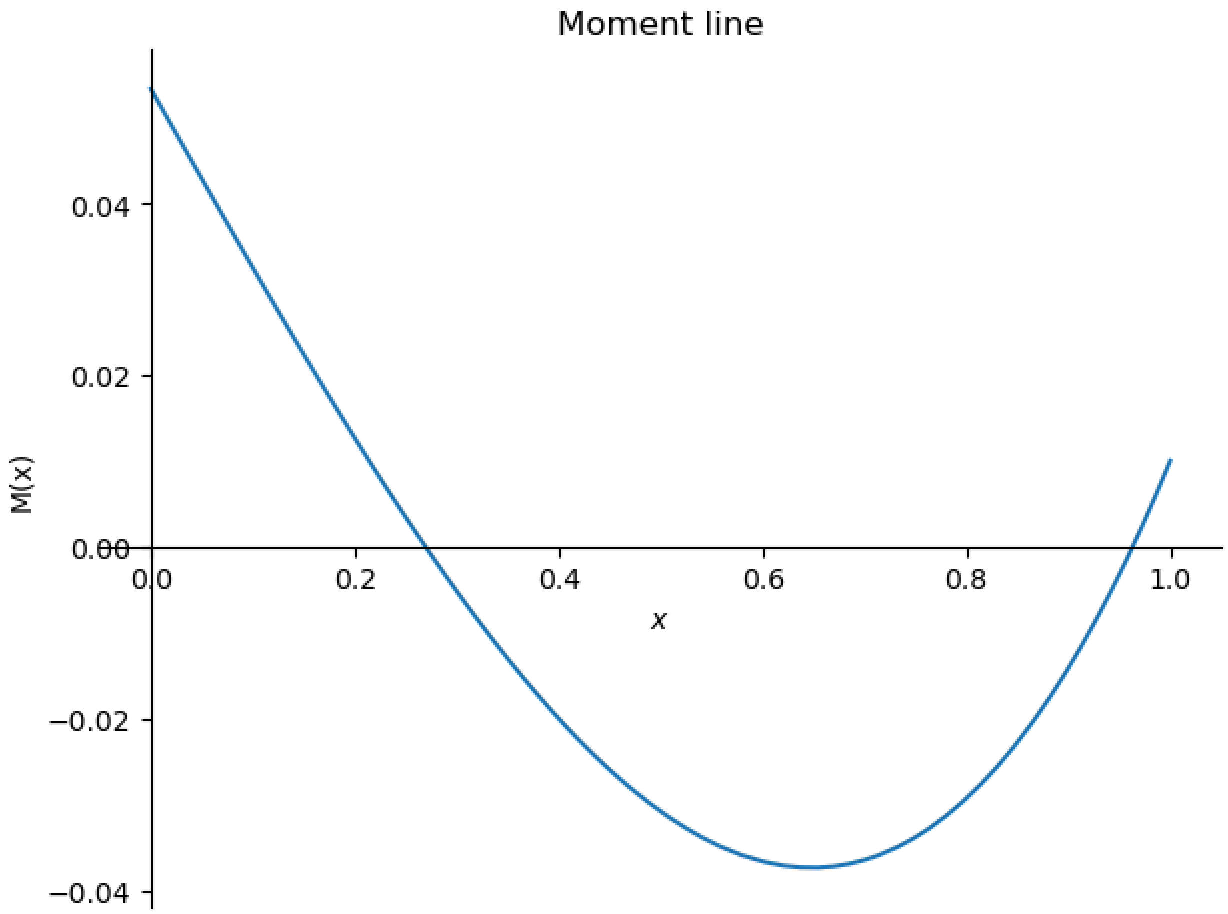



```
# Define the boundary conditions
eq1 = sym.Eq(w.subs(x,0),0)
eq2 = sym.Eq(w.subs(x,L),0)
eq3 = sym.Eq(phi.subs(x,0),0)
eq4 = sym.Eq(M.subs(x,L),-M_p)
```

```python
# Solve the integration constants
sol2 = sym.solve([eq1, eq2, eq3, eq4 ], sym.symbols('C1, C2, C3, C4'))
display(sol2)
```

$$\left\{C_1 : 0,\ C_2 : 0,\ C_3 : \frac{7L^2 q_L - 60M_p}{240EI},\ C_4 : \frac{-3L^2 q_L + 20M_p}{80EIL}\right\}$$

```python
# Plot the moment line to identify the maximum moment
sym.plot(-M.subs(sol2).subs(plotvalues),(x,0,1),ylabel='M(x)',title='Moment line');
```



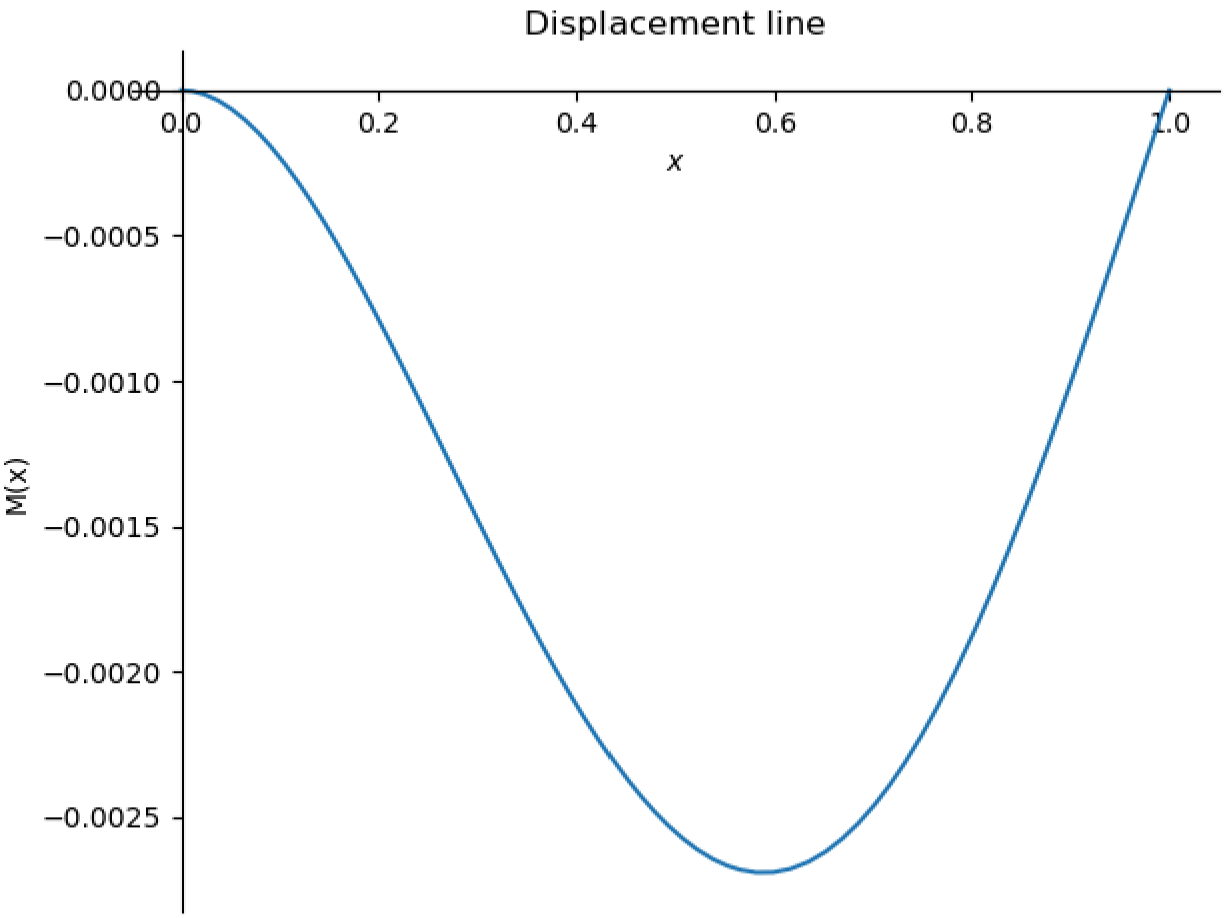

```python
# Obtain the maximum moment at x = 0
display(M.subs(sol2).subs(x,0))
```

$$-\frac{7L^2 q_L}{120} + \frac{M_p}{2}$$

```python
# Obtain the load corresponding to the second yield moment
q2 = sym.solve(sym.Eq(-M.subs(sol2).subs(x,0),M_p), q_L)[0]
display(q2)
```

$$\frac{180M_p}{7L^2}$$

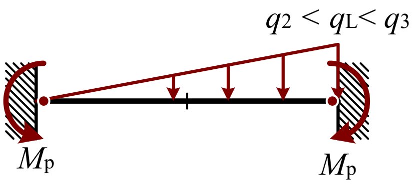
```
# Plot the displacement line
sym.plot(-w.subs(sol2).subs(plotvalues),(x,0,1),ylabel='M(x)',title='Displacement line');
```




```
# Obtain the displacement at x = L/2
w2 = w.subs(sol2).subs(q_L, q2).subs(x,L/2)
display(w2)
```

$$\frac{19L^2M_p}{448EI}$$

Plastic behaviour: Beam simply supported on both ends with M_p working at it

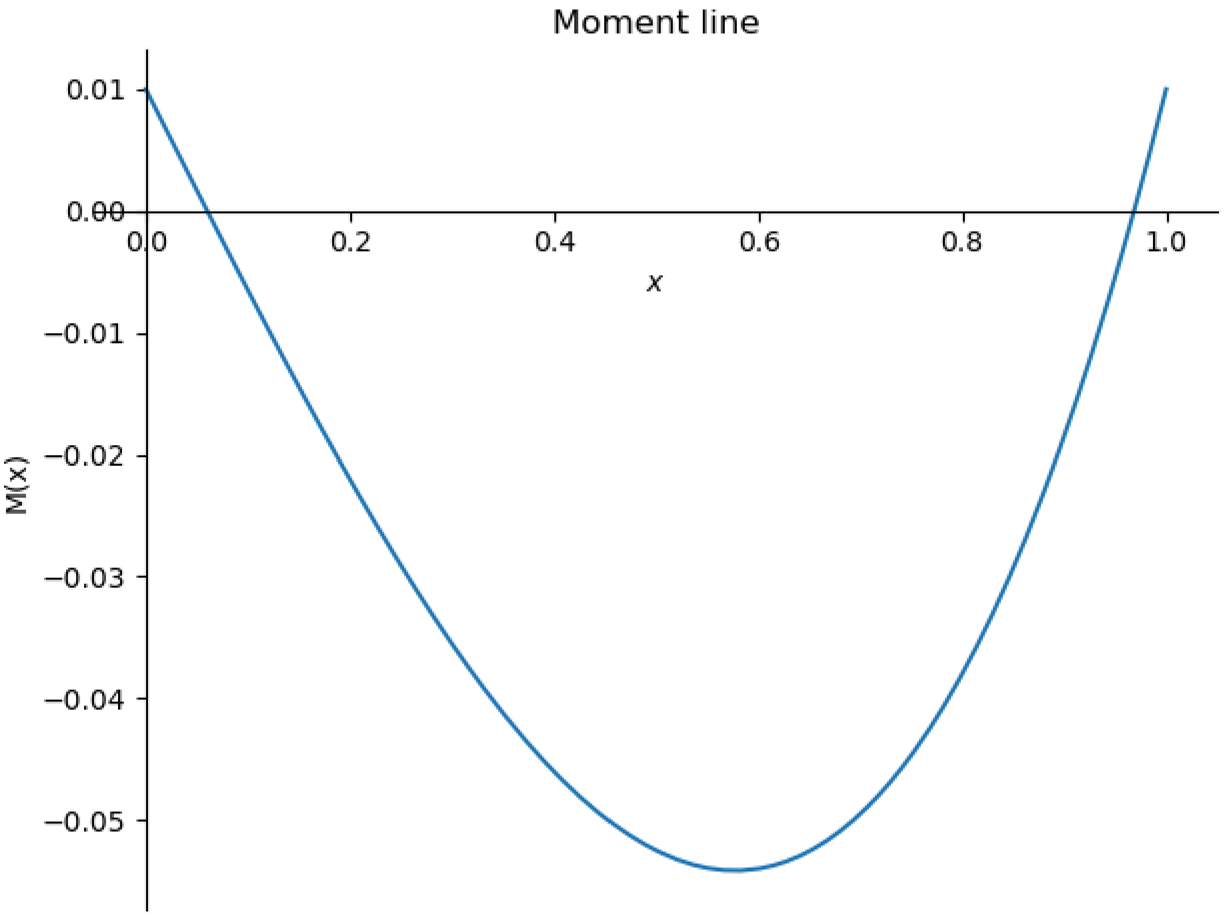


```python
# Define the boundary conditions
eq1 = sym.Eq(w.subs(x,0),0)
eq2 = sym.Eq(w.subs(x,L),0)
eq3 = sym.Eq(M.subs(x,0),-M_p)
eq4 = sym.Eq(M.subs(x,L),-M_p)
```

```python
# Solve the integration constants
sol3 = sym.solve([eq1, eq2, eq3, eq4 ], sym.symbols('C1, C2, C3, C4'))
display(sol3)
```

$$\left\{C_1 : 0,\ C_2 : \frac{7L^3 q_L - 180 L M_p}{360EI},\ C_3 : \frac{M_p}{2EI},\ C_4 : -\frac{L q_L}{36EI}\right\}$$

```python
# Plot the moment line to identify the maximum moment
sym.plot(-M.subs(sol3).subs(plotvalues),(x,0,1),ylabel='M(x)',title='Moment line');
```




```python
# Find the location of the maximum moment
x_M_max = sym.solve(sym.Eq(V.subs(sol3),0),x)[1]
display(x_M_max)
```

$$\frac{\sqrt{3}L}{3}$$

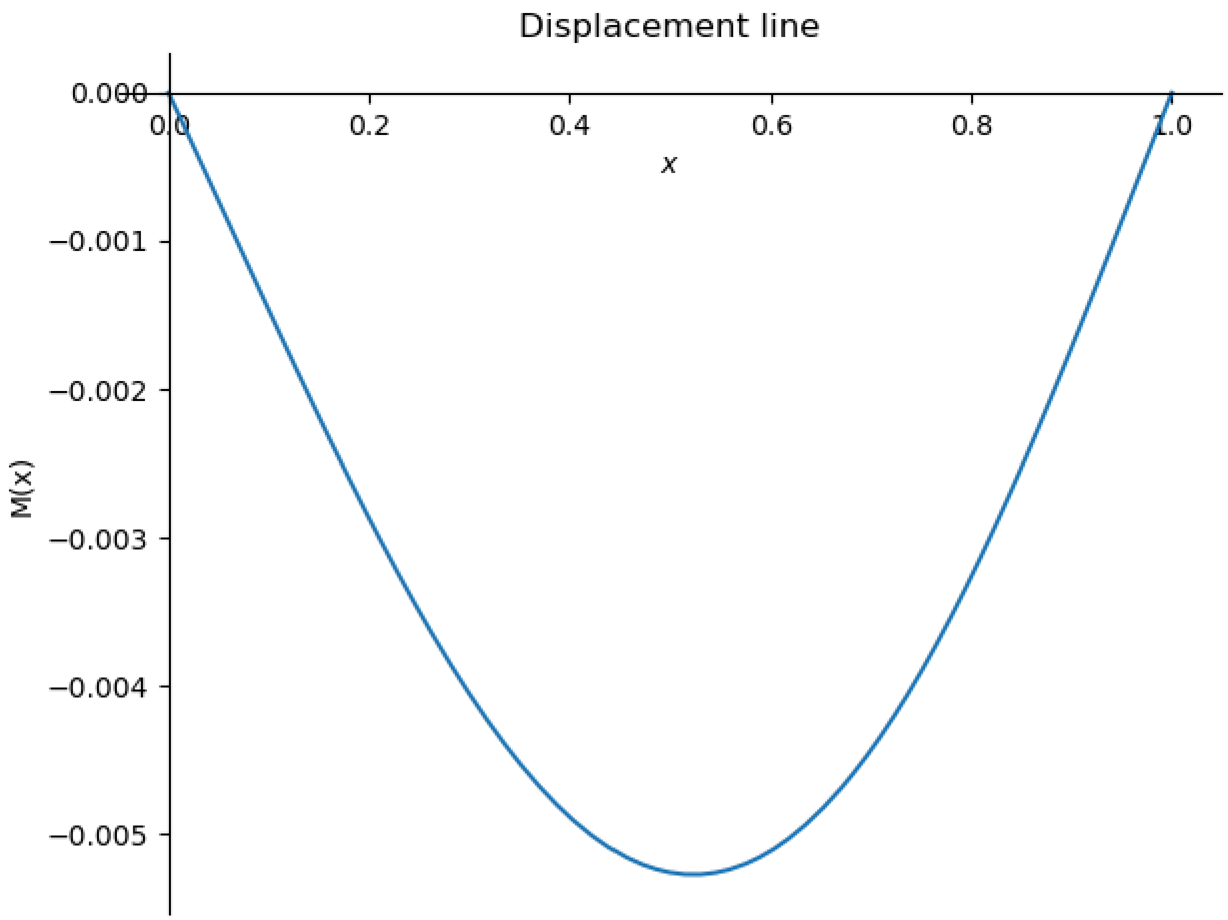
```python
# Obtain the maximum moment at x = x_M_max
display(M.subs(sol3).subs(x,x_M_max))
```

$$EI\left(\frac{\sqrt{3}L^2 q_L}{27EI} - \frac{M_p}{EI}\right)$$

```python
# Obtain the load corresponding to the third yield moment
q3 = sym.solve(sym.Eq(M.subs(sol3).subs(x,x_M_max).subs(x,0),M_p), q_L)[0]
display(q3)
```

$$\frac{18\sqrt{3}M_p}{L^2}$$

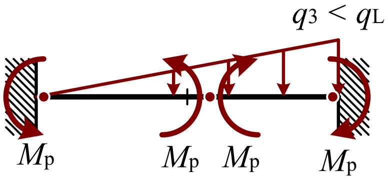
```python
# Plot the displacement line
sym.plot(-w.subs(sol3).subs(plotvalues),(x,0,1),ylabel='M(x)',title='Displacement line');
```




```python
# Obtain the displacement at x = L/2
w3 = w.subs(sol3).subs(q_L, q3).subs(x,L/2).simplify()
display(sym.simplify(w3))
```

$$\frac{L^2 M_p\left(-16 + 15\sqrt{3}\right)}{128EI}$$

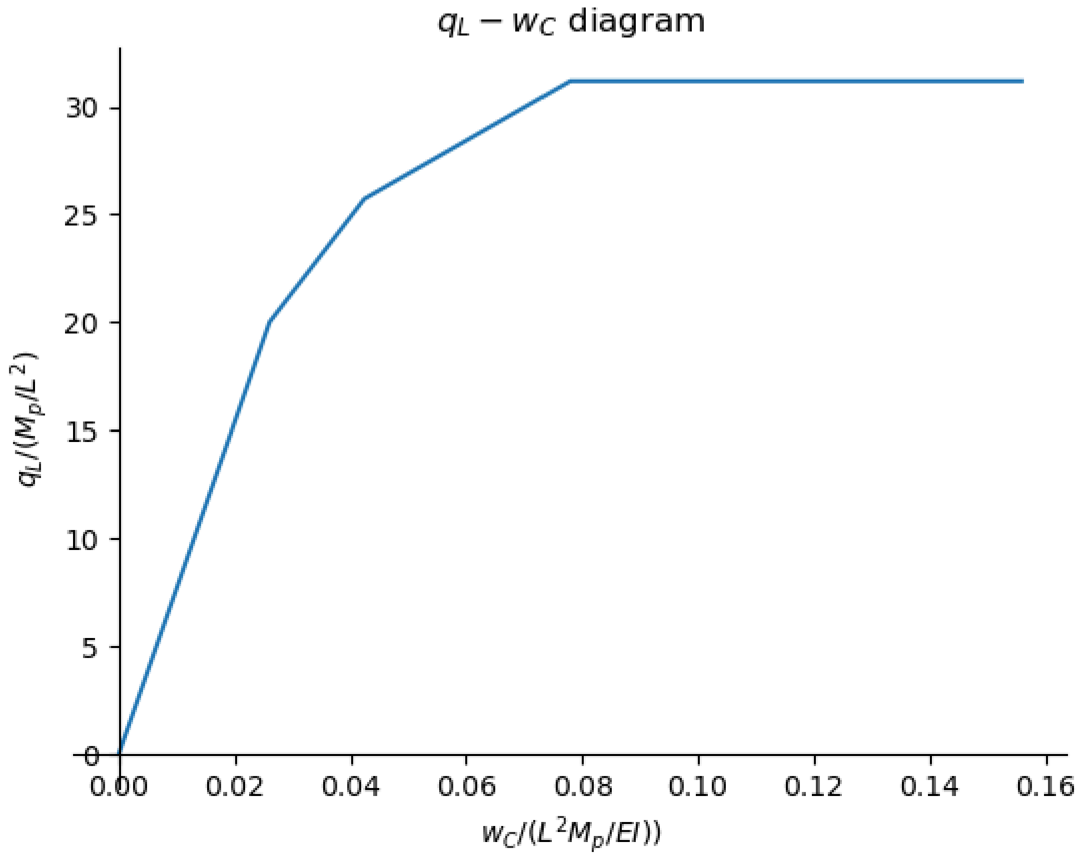
Mechanism: Beam simply supported on both ends with M_p working at it and a plastic hinge at $\frac{\sqrt{3}L}{3}$



$q_L - w_C$ diagram

```
# Obtain the displacements and q values as coefficients of a constant term.
w_list = [0,w1.coeff(L**2*M_p/EI),w2.coeff(L**2*M_p/EI),w3.coeff(L**2*M_p/EI),w3.coeff(L*
q_list = [0,q1.coeff(M_p/L**2),q2.coeff(M_p/L**2),q3.coeff(M_p/L**2),q3.coeff(M_p/L**2)]
```

```
# Plot the q_L-w diagram
plt.plot(w_list,q_list)
plt.xlabel('$w_C / (L^2 M_p / EI))$')
plt.ylabel('$q_L / (M_p / L^2)$')
plt.title('$q_L-w_C$ diagram')
plt.gca().spines['right'].set_color('none')
plt.gca().spines['top'].set_color('none')
plt.gca().spines['bottom'].set_position('zero')
plt.gca().spines['left'].set_position('zero')
```



$q_L - w_C$ diagram
$q_L/(M_p/L^2)$
$w_C/(L^2 M_p/EI))$
0
5
10
15
20
25
30
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16

Credits and License

Contents

You can refer to this book as:

> Tom van Woudenberg from Delft University of Technology (2025) *CTB3330 Strutural Mechanics 4*. https://oit.tudelft.nl/CTB3330/2024. Source files at TUDelft-books/CTB3330

You can refer to individual chapters or pages within this book as:

> `<Title of Chapter or Page>`. Tom van Woudenberg from Delft University of Technology (2025) *CTB3330 Strutural Mechanics 4*. https://oit.tudelft.nl/CTB3330/2024/. Source files at . Source files at TUDelft-books/CTB3330: `./book/<path to file(s)>` chapter, accessed `date`.

We anticipate that the content of this book will change significantly. Therefore, we recommend using the source code directly with the citation above that refers to the GitHub repository and lists the date and name of the file. Although content will be added over time, chapter titles and URL's in this book are expected to remain relatively static. However, we make no guarantee, so if it is important for you to reference a specific location/commit within the book.

How the book is made

This book is created using open source tools: it is a JupyterBook that is written using Markdown and Jupyter notebooks. Additional tooling is used from the TeachBooks initiative to enhance the editing and reading experience. The files are stored on a public GitHub repository. The website can be viewed at TUDelft-books/CTB3330. View the repository README file or contact the authors for additional information.

Changelog

Contents

v1.0.0 Start course

Nothing has changed in this book since the start of this course