Machine Design Homework 2

June 10, 2022

Gabe Morris

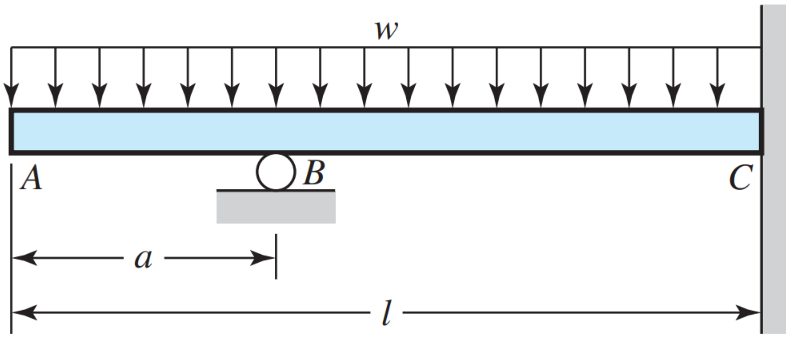
[1]:
```python
import matplotlib.pyplot as plt
import sympy as sp
from IPython.display import display

plt.style.use('maroon_ipynb.mplstyle')
```

Contents

1 Problem 4-118 **3**
1.1 Given . . . 3
1.2 Find . . . 3
1.3 Solution . . . 3

1 Problem 4-118

1.1 Given

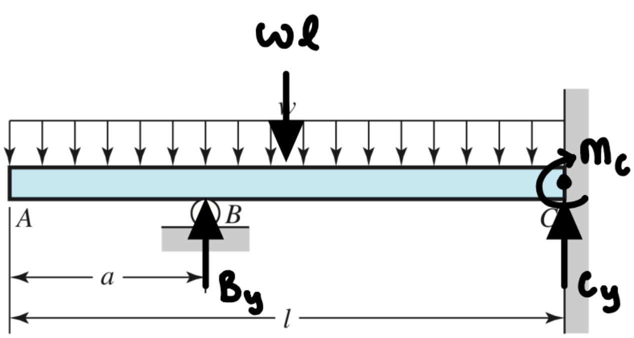


1.2 Find

Determine the support reactions using Castigliano's theory.

1.3 Solution

The free body diagram yields two equations with three unknowns,

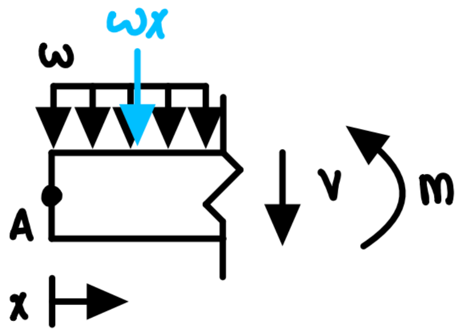


[2]:
```
By, w, l, Cy, Mc, a = sp.symbols('B_y w l C_y M_c a')
eq1 = sp.Eq(By + Cy, w*l)  # Forces in y direction
eq2 = sp.Eq(Mc + By*(l - a), w*l*(l/2))
display(eq1, eq2)
```

$B_y + C_y = lw$

$B_y(-a + l) + M_c = \frac{l^2w}{2}$

The bending and shear diagram equations as a function of x may be extracted like so,



The above figure is for $0 \le x \le a$.

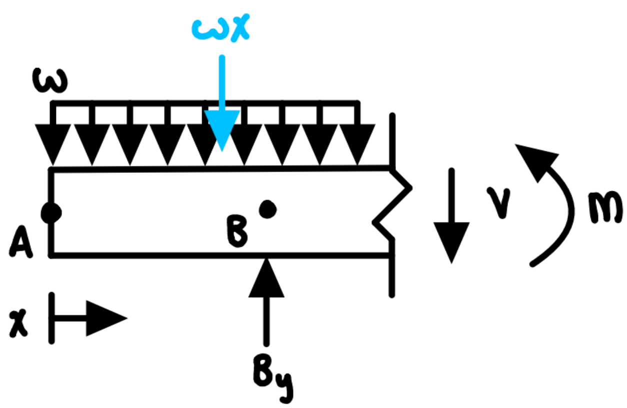
[3]:
```
# Shear equation
x = sp.Symbol('x')
V1 = -w*x
V1
```

[3]: $-wx$

[4]:
```
# Moment equation
M1 = -sp.S('0.5')*w*x**2
M1
```

[4]: $-0.5wx^2$

For $a \le x \le l$,



[5]:
```
V2 = By - w*x
V2
```

[5]: $B_y - wx$

[6]:
```
M2 = By*(x - a) - sp.S('0.5')*w*x**2
M2
```

[6]: $B_y\left(-a + x\right) - 0.5wx^2$

All together, the moment and shear equation may be represented as the piecewise functions below.

[7]:
```python
V = sp.Piecewise((V1, (x >= 0) & (x <= a)), (V2, (x >= a) & (x <= l)))
M = sp.Piecewise((M1, (x >= 0) & (x <= a)), (M2, (x >= a) & (x <= l)))
display(V, M)
```

$$\begin{cases} -wx & \text{for } a \geq x \wedge x \geq 0 \\ B_y - wx & \text{for } l \geq x \wedge a \leq x \end{cases}$$

$$\begin{cases} -0.5wx^2 & \text{for } a \geq x \wedge x \geq 0 \\ B_y\left(-a + x\right) - 0.5wx^2 & \text{for } l \geq x \wedge a \leq x \end{cases}$$

Castigliano's theory involves computing the total energy, which is

$$U = \int \frac{M^2}{2EI}dx + \int \frac{CV^2}{2AG}dx$$

We can integrate across each section. Watch as `sympy` impressively solves this huge integral for us, symbolically. If we neglect the shear stress, we will arrive at the same answer for the superposition approach. This assumption is usually valid, since beams are typically much longer than their diameters/cross-sectional distance.

[8]:
```python
C, E, I, A, G, U_sym = sp.symbols('C E I A G U')
# U1 = sp.Integral(M1**2/(2*E*I) + C*V1**2/(2*A*G), (x, 0, a))
# U2 = sp.Integral(M2**2/(2*E*I) + C*V2**2/(2*A*G), (x, a, l))

# Neglect shear
U1 = sp.Integral(M1**2/(2*E*I), (x, 0, a))
U2 = sp.Integral(M2**2/(2*E*I), (x, a, l))
U = U1 + U2
sp.Eq(U_sym, U)
```

[8]:

$$U = \int\limits_{a}^{l} \frac{\left(B_y\left(-a + x\right) - 0.5wx^2\right)^2}{2EI}\,dx + \int\limits_{0}^{a} \frac{0.125w^2x^4}{EI}\,dx$$

[9]:
```python
U_doit = U.doit().expand().n(6)
sp.Eq(U_sym, U_doit)
```

[9]:

$$U = -\frac{0.166667B_y^2a^3}{EI} + \frac{0.5B_y^2a^2l}{EI} - \frac{0.5B_y^2al^2}{EI} + \frac{0.166667B_y^2l^3}{EI} - \frac{0.0416667B_ya^4w}{EI} + \frac{0.166667B_yal^3w}{EI} - \frac{0.125B_yl^4w}{EI} + \frac{0.025l^5w^2}{EI}$$

Castigliano's Theory is,

$$\delta_i = \frac{\partial U}{\partial F_i}$$

δ_i is the displacement at the point where the force F_i occurs. We know the deflection at point B is 0.

$$\frac{\partial U}{\partial B_y} = 0$$

[10]:
```python
# Take the derivative of the expression above and set equal to 0
eq3 = sp.Eq(U_doit.diff(By).expand().n(6), 0)
eq3
```

[10]:
$$-\frac{0.333333B_ya^3}{EI} + \frac{1.0B_ya^2l}{EI} - \frac{1.0B_yal^2}{EI} + \frac{0.333333B_yl^3}{EI} - \frac{0.0416667a^4w}{EI} + \frac{0.166667al^3w}{EI} - \frac{0.125l^4w}{EI} = 0$$

[11]:
```python
sol = sp.solve([eq1, eq2, eq3], (By, Cy, Mc), dict=True)[0]
for key, value in sol.items():
    display(sp.Eq(key, value.simplify()))
```

$$B_y = \frac{0.125w\left(-a^2 - 2.0al - 3.0l^2\right)}{a - l}$$

$$C_y = \frac{0.125w\left(a^2 + 10.0al - 5.0l^2\right)}{a - l}$$

$$M_c = w\left(-0.125a^2 - 0.25al + 0.125l^2\right)$$