Machine Design Homework 2

June 10, 2022

Gabe Morris

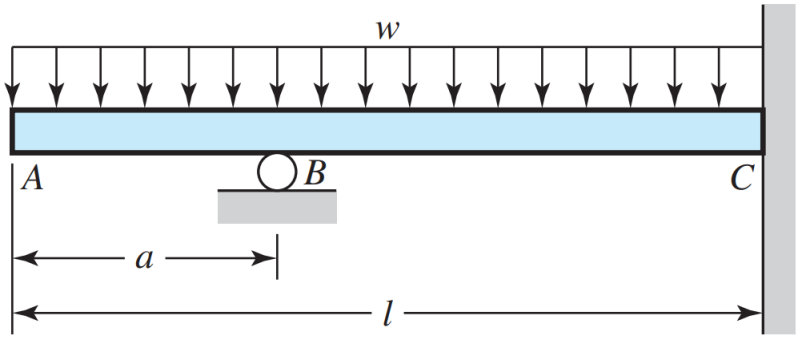
[1]:
```
import matplotlib.pyplot as plt
import sympy as sp
from IPython.display import display

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

Contents

ME 4403 Homework 2 Gabe Morris
gnm54

1 Problem 4-118

1.1 Given

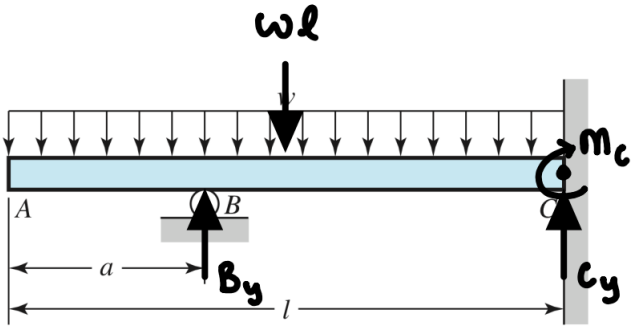


1.2 Find

Determine the support reactions using Castigliano's theory.

1.3 Solution

The free body diagram yields two equations with three unknowns,

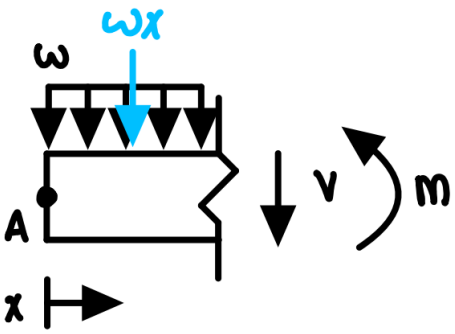


[2]:
```
By, w, l, Cy, Mc, a = sp.symbols('B_y w l C_y M_c a')
eq1 = sp.Eq(By + Cy, w*l)  # Forces in y direction
eq2 = sp.Eq(Mc + By*(l - a), w*l*(l/2))
display(eq1, eq2)
```

$B_y + C_y = lw$

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

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



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

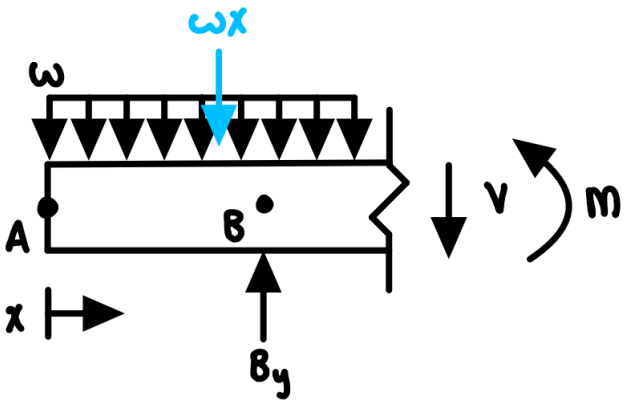
[3]:
```
# Shear equation
x = sp.Symbol('x')
V1 = -w*x
V1
```

[3]: $-wx$

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

[4]: $-0.5wx^2$

For $a \leq x \leq l$,



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

[5]: $B_y - wx$

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

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

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

[7]:
```
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]:
```
C, E, I, A, G, U_sym = sp.symbols('C E I A G U')

# With shear
# 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))

# Neglected 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]:
$$U = \int_{a}^{l} \frac{\left(B_y\left(-a + x\right) - 0.5wx^2\right)^2}{2EI} dx + \int_{0}^{a} \frac{0.125w^2x^4}{EI} dx$$

[9]:
```
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]:
```
# 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]:
$$-\frac{0.333333 B_y a^3}{EI} + \frac{1.0 B_y a^2 l}{EI} - \frac{1.0 B_y a l^2}{EI} + \frac{0.333333 B_y l^3}{EI} - \frac{0.0416667 a^4 w}{EI} + \frac{0.166667 a l^3 w}{EI} - \frac{0.125 l^4 w}{EI} = 0$$

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

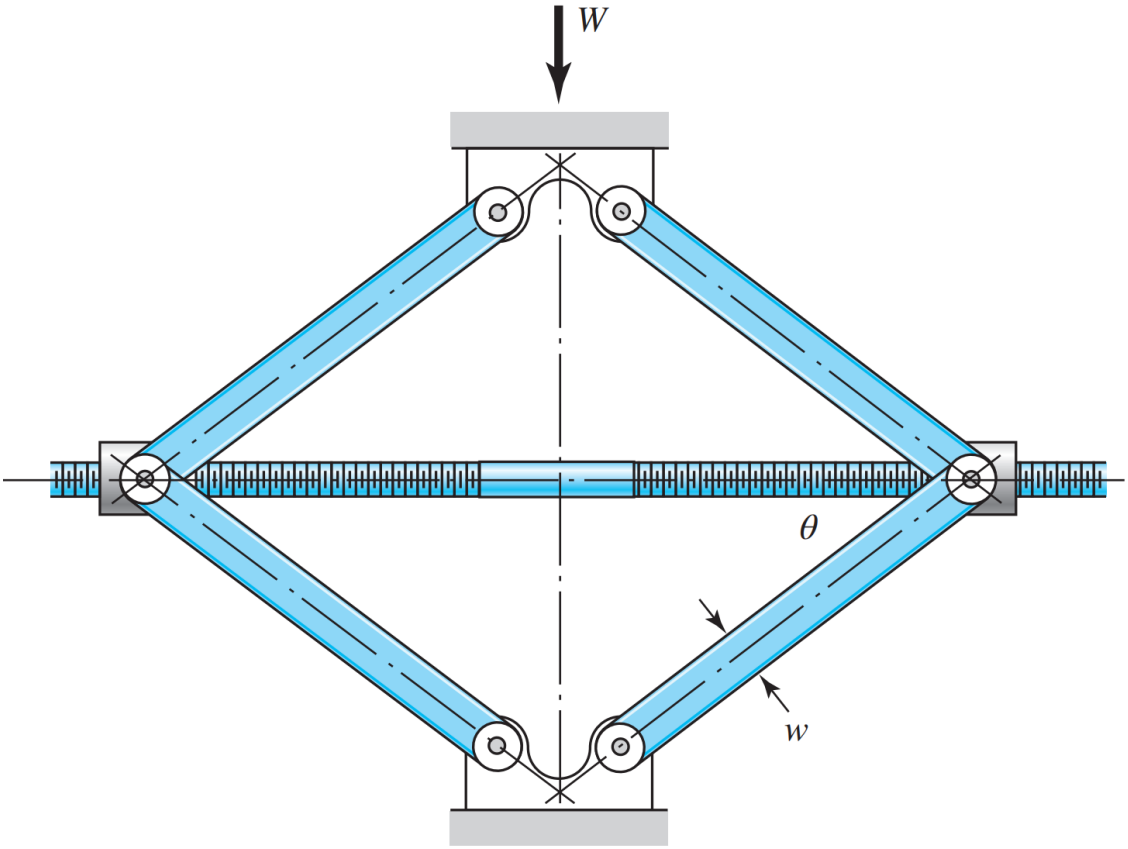
$$B_y = \frac{0.125 w \left(-a^2 - 2.0 a l - 3.0 l^2\right)}{a - l}$$

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

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

2 Problem 4-132

2.1 Given



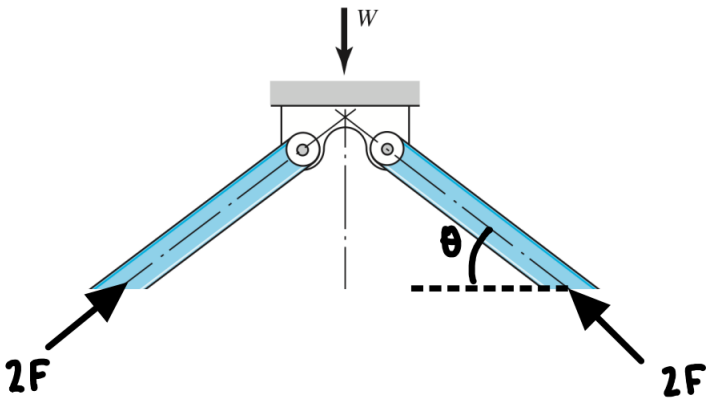
The figure above shows a schematic drawing of a vehicular jack that is to be designed to support a maximum mass of 300 kg based on the use of a design factor $n_d = 3.50$. The opposite-handed threads on the two ends of the screw are cut to allow the link angle θ to vary from 15 to 70°. The links are to be machined from AISI 1010 hot-rolled steel bars. Each of the four links is to consist of two bars, one on each side of the central bearings. The bars are to be 350 mm long and have a bar width of $w = 30\ mm$. The pinned ends are to be designed to secure an end-condition constant of at least C = 1.4 for out-of-plane buckling.

2.2 Find

Find a suitable preferred thickness and the resulting factor of safety for this thickness.

2.3 Solution

Start by finding the compressive force in the members,



After acquiring the force, the critical force P_cr may be obtained using the relationship with the desired factor of safety.

[12]:
```
w = 300*sp.S('9.81')
F, theta = sp.symbols(r'F \theta')
eq1 = sp.Eq(w, 4*F*sp.sin(theta))
F_sol = sp.solve(eq1, F)[0]
sp.Eq(F, F_sol)
```

[12]:
$$F = \frac{735.75}{\sin\left(\theta\right)}$$

The value of $\sin\theta$ should be smallest in order to get the greatest force. We will use the 15° limit position to get the greater force value.

[13]:
```
# Solving for F
F_ = (F_sol.subs(theta, 15*sp.pi/180)).n()
F_  # in N
```

[13]:
2842.71970676873

We can use the relationship for long columns with central loading,

$$\frac{P_{cr}}{A} = \frac{C\pi^2 E}{\left(\frac{l}{k}\right)}$$

[14]:
```
# Calculate P critical
P_cr = sp.S('3.5')*F_
P_cr  # in N
```

[14]:
9949.51897369055

[15]:
```
# Solving for the thickness
t = sp.Symbol('t')  # thickness symbol
w, E, l, C = sp.S('0.03'), sp.S('207e9'), sp.S('0.35'), sp.S('1.4')
Sy = sp.S('180e6')
```

```
k = t/sp.sqrt(12)
A = t*w
eq1 = sp.Eq(P_cr, A*C*sp.pi**2*E/(l/k)**2)
eq1.n()
```

[15]:

$$9949.51897369055 = 58371660315.0142t^3$$

[16]:
```
t_sol = sp.solve(eq1)[0]
display(t_sol)  # solution in m
```

0.0055445547890295

The thickness should be no less than $6\ mm$ because 6 mm is the next biggest standard size. We can figure out the P_{cr} by plugging this value back into Euler's equation. First, we need to check and see if this is a valid solution.

[17]:
```
l_k1 = sp.sqrt(2*sp.pi**2*C*E/Sy)
(l/k.subs(t, sp.S('0.006'))) > l_k1
```

[17]:

True

Because $\left(\frac{l}{k}\right) > \left(\frac{l}{k}\right)_1$, the solution is valid. Now we can calculate the factor of safety.

[18]:
```
P_cr_ = eq1.rhs.subs(t, sp.S('0.006'))
P_cr_.n()
```

[18]:

12608.2786280431

[19]:
```
# Getting new factor of safety
(P_cr_/F_).n()
```

[19]:

4.43528730533011