

# Machine Design Final Exam

August 1, 2022

Gabe Morris

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

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

def sub_return(expression, substitution, sym=None):
    """
    Displays a substituted expression and returns the value.

    :param expression: Symbolic expression
    :param substitution: A list of tuples to be passed into the .sub method
    :param sym: An optional symbol to display prior to operations
    :return: A simplified value of what is substituted.
    """
    with sp.evaluate(False):
        expr_sub = expression.subs(substitution)
        simp = expr_sub.doit().n()
        if sym is None:
            lev1 = sp.Eq(expression, expr_sub)
            lev2 = sp.Eq(lev1, simp, evaluate=False)
            display(lev2)
        else:
            lev1 = sp.Eq(sym, expression)
            lev2 = sp.Eq(lev1, expr_sub, evaluate=False)
            lev3 = sp.Eq(lev2, simp, evaluate=False)
            display(lev3)
    return simp

def get_principal(s_x, s_y, tau_xy):
    """
    Shows the principal stress calculations for a 2D element.

    :param s_x: normal stress along the x-axis
    :param s_y: normal stress along the y-axis
    :param tau_xy: shear stress on face x in the direction of y
```

```

:~return: principal stresses
"""
s_x_, s_y_, t_xy_ = sp.symbols(r'\sigma_x \sigma_y \tau_{xy}')
symbols_ = sp.symbols(r'\sigma_1 \sigma_2 \tau_1 \tau_2')
expr1_ = (s_x_ + s_y_)/2 + sp.sqrt(((s_x_ - s_y_)/2)**2 + t_xy_**2)
expr2_ = (s_x_ + s_y_)/2 - sp.sqrt(((s_x_ - s_y_)/2)**2 + t_xy_**2)
expr3_ = sp.sqrt(((s_x_ - s_y_)/2)**2 + t_xy_**2)
expr4_ = -sp.sqrt(((s_x_ - s_y_)/2)**2 + t_xy_**2)
sub_list = [(s_x_, s_x), (s_y_, s_y), (t_xy_, tau_xy)]
ans = []
for sym_, expr_ in zip(symbols_, [expr1_, expr2_, expr3_, expr4_]):
    ans.append(sub_return(expr_, sub_list, sym=sym_))
return ans

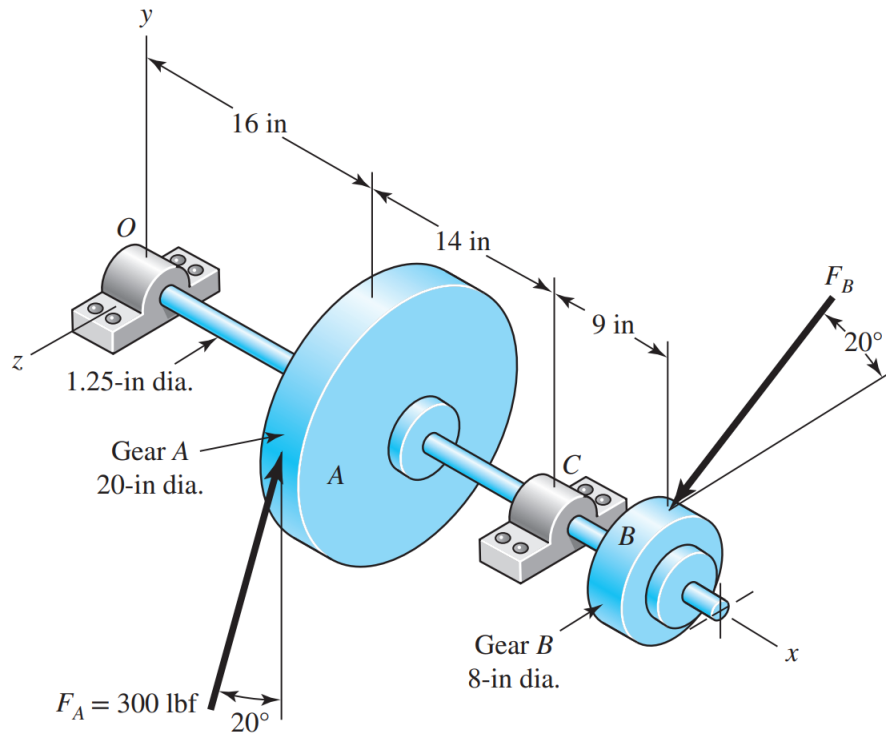
```

## Contents

<b>1</b>	<b>Problem 3-83</b>	<b>4</b>
1.1	Given . . . . .	4
1.2	Find . . . . .	4
1.3	Solution . . . . .	5
1.3.1	Part A and Part B . . . . .	5
1.3.2	Part C . . . . .	6
1.3.3	Part D . . . . .	6
1.3.4	Part E . . . . .	7

## 1 Problem 3-83

### 1.1 Given



A gear reduction unit uses the countershaft shown in the figure above. Gear A receives power from another gear with a transmitted force  $F_A$  applied at the  $20^\circ$  pressure angle as shown. The power is transmitted through the shaft and delivered through gear B through a transmitted force  $F_B$  at the pressure angle shown.

### 1.2 Find

- Determine the force  $F_B$ , assuming the shaft is running at a constant speed.
- Find the bearing reaction forces, assuming the bearings act as simple supports.
- Draw a shear-force and bending moment diagrams for the shaft. If needed, make one set for the horizontal plane and another set for the vertical plane.
- At the point of maximum bending moment, determine the bending stress and the torsional shear stress.
- At the point of maximum bending moment, determine the principal stresses and the maximum shear stress.

## 1.3 Solution

### 1.3.1 Part A and Part B

$F_B$  and the reaction forces may be found by summing the forces and taking the moments about the point  $O$ .

```
[2]: # Using matrices takes less thinking and more letting python do the work.
Oy, Oz, Cy, Cz, F_B = sp.symbols('O_y O_z C_y C_z F_B')
zero = sp.ZeroMatrix(3, 1)
F_A_mat = sp.Matrix([0, 300*sp.cos(sp.rad(20)), -300*sp.sin(sp.rad(20))])
F_B_mat = sp.Matrix([0, -F_B*sp.sin(sp.rad(20)), F_B*sp.cos(sp.rad(20))])
O = sp.Matrix([0, Oy, Oz])
C = sp.Matrix([0, Cy, Cz])

# Sum forces
sys1 = sp.Eq(O + F_A_mat + F_B_mat + C, zero)

# Sum moments about O
rA = sp.Matrix([16, 0, 10])
rC = sp.Matrix([16 + 14, 0, 0])
rB = sp.Matrix([16 + 14 + 9, 4, 0])
sys2 = sp.Eq(rA.cross(F_A_mat) + rC.cross(C) + rB.cross(F_B_mat), zero)

sol = sp.solve([sys1, sys2], dict=True)[0]
for key, value in sol.items():
    display(sp.Eq(key, value.n())) # lbf
```

$$C_y = 183.118820416782$$

$$C_z = -861.477082334154$$

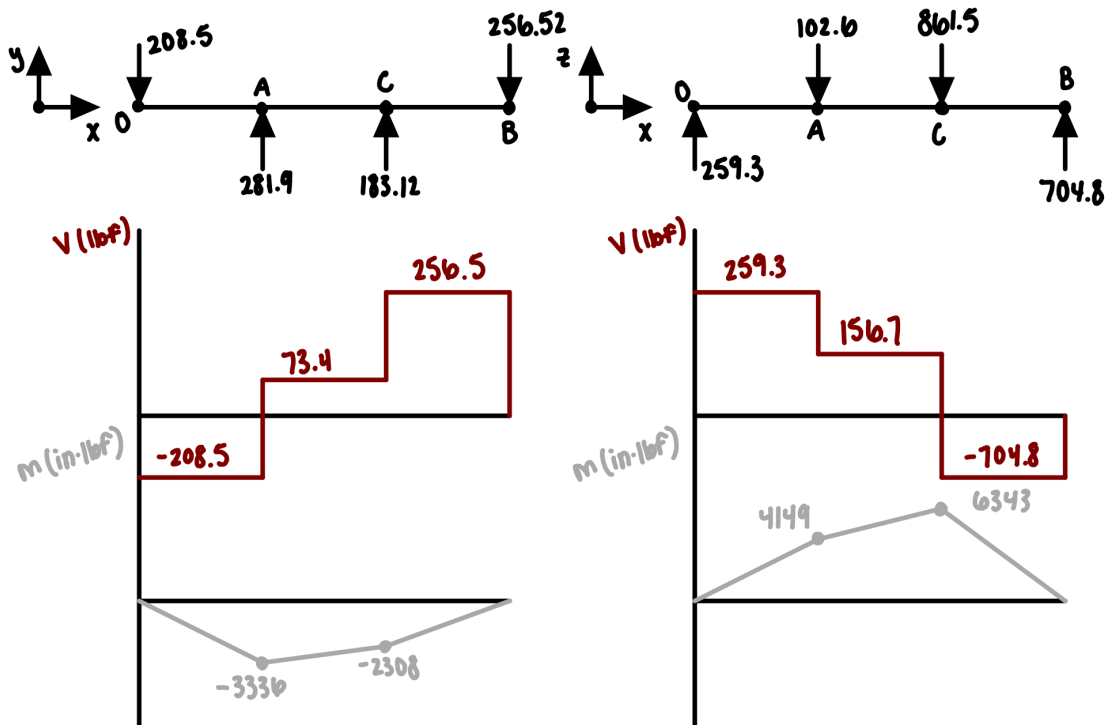
$$F_B = 750.0$$

$$O_y = -208.511499158303$$

$$O_z = 259.313659742423$$

The reaction forces above assume all positive directions.

### 1.3.2 Part C



### 1.3.3 Part D

The magnitudes of the moment at section A and section C must be compared.

```
[3]: M_A = sp.sqrt(sp.S('3336.18399')**2 + sp.S('4149.01856')**2)
M_C = sp.sqrt(sp.S('6342.92519')**2 + sp.S('2308.63597')**2)
display(M_A, M_C)
```

5323.95328927262

6750.00000058625

```
[4]: # M_C is the greater value
M_C = sp.S(6750) # in*lb
d = sp.S('1.25')
sig = M_C*32/(sp.pi*d**3) # psi
T = 300*sp.cos(sp.rad(20))*10 # static torque
tau = T*16/(sp.pi*d**3) # psi
display(sig.n(), tau.n())
```

35202.5269328378

7351.01217595663

### 1.3.4 Part E

$$\sigma_1 = \frac{\sigma_x}{2} + \frac{\sigma_y}{2} + \sqrt{\tau_{xy}^2 + \left(\frac{\sigma_x}{2} - \frac{\sigma_y}{2}\right)^2} = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 35203.0 + \sqrt{7351.0^2 + \left(\left(-\frac{1}{2}\right) 0 + \frac{1}{2} \cdot 35203.0\right)^2} = 36675.875$$

$$\sigma_2 = \frac{\sigma_x}{2} + \frac{\sigma_y}{2} - \sqrt{\tau_{xy}^2 + \left(\frac{\sigma_x}{2} - \frac{\sigma_y}{2}\right)^2} = -\sqrt{7351.0^2 + \left(\left(-\frac{1}{2}\right) 0 + \frac{1}{2} \cdot 35203.0\right)^2} + \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 35203.0 = -1473.375$$

$$\tau_1 = \sqrt{\tau_{xy}^2 + \left(\frac{\sigma_x}{2} - \frac{\sigma_y}{2}\right)^2} = \sqrt{7351.0^2 + \left(\left(-\frac{1}{2}\right) 0 + \frac{1}{2} \cdot 35203.0\right)^2} = 19074.625$$

$$\tau_2 = -\sqrt{\tau_{xy}^2 + \left(\frac{\sigma_x}{2} - \frac{\sigma_y}{2}\right)^2} = -\sqrt{7351.0^2 + \left(\left(-\frac{1}{2}\right) 0 + \frac{1}{2} \cdot 35203.0\right)^2} = -19074.625$$