deformsbook

Jake Grohs

2023-05-30

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

```
1 + 1
```

```
[1] 2
```

1 Strength of Materials Problem Workout

To scaffold your learning in this example, we have provided a free body diagram for you and a repeat of the problem statement.

A city planner is installing a new traffic light. Light A weighs 65 lb, while lights B and C weigh 50 lb each. The post at O has a hollow circular cross-section with an outer diameter of 5 inches and a wall thickness of 0.2 inches. It will be made from aluminum alloy with a tensile yield stress of 35 ksi and a compressive yield stress of 20 ksi. A factor of safety of 2 is required with respect to yield. You may ignore the weight of the post.

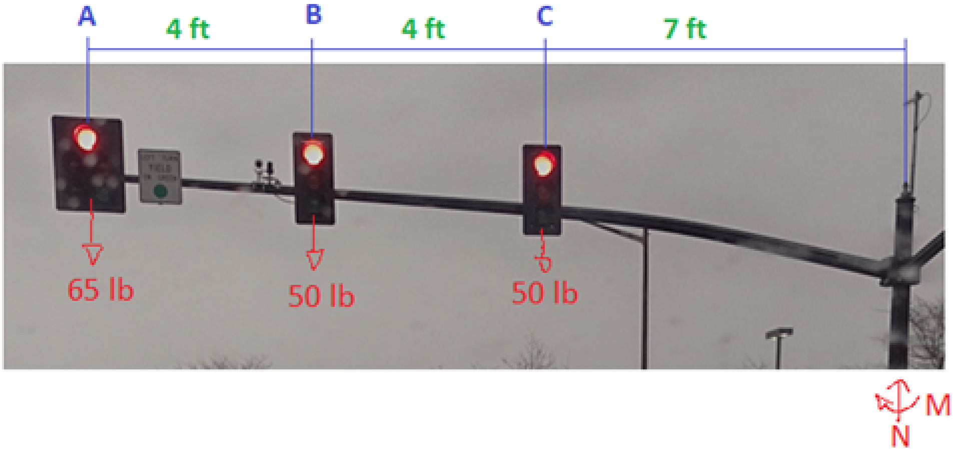

Figure 1.1: Figure 1: Three traffic light installation with loads

Please work through the problem step by step showing your math in the interactive interface here.

```
#| standalone: true
#| viewerHeight: 420
#| components: [viewer]
from shiny import App, render, ui, reactive
from sympy import solve, Eq
from sympy.abc import F,M
from sympy.parsing.sympy_parser import parse_expr
```

```
app_ui = ui.page_fluid(
   ui.head_content(
        ui.tags.script(
            src="https://mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMM
        ),
        ui.tags.script(
            "if (window.MathJax) MathJax.Hub.Queue(['Typeset', MathJax.Hub]);"
        ),
    ),

   # ui.layout_sidebar(
   #    ui.panel_main(
   #      ui.markdown("Use equilibrium equations to find the internal reaction forces:\$$\
   #      ui.input_text("forces","Please write the right side of the equation",placeholder
   #      ui.markdown("Use equilibrium eqeuations to find the internal reaction moment:\$$
   #      ui.input_text("moments","Please write the right side of the equation",placeholde
   #      ui.input_action_button("typesetMath", "Update (Typeset) Equations", class_="btn-
   #      ui.input_action_button("solveEquations", "Solve Equations", class_="btn-success"
   #      ),
   #    ui.panel_sidebar(ui.output_ui("dyn_ui_forces"),
   # ui.output_ui("dyn_ui_moments"),
   # ui.output_ui("ui_typeset"),
   # ui.output_ui("ui_solutions"),
   # ui.output_ui("ui_typesetSolutions")
   # ),
   # ),



   ui.markdown("Use equilibrium equations to find the internal reaction forces:\$$\Sigma F
   ui.output_ui("dyn_ui_forces"),
   ui.input_text("forces","Please write the right side of the equation",placeholder="Enter
   ui.markdown("Use equilibrium eqeuations to find the internal reaction moment:\$$\Sigma
   ui.output_ui("dyn_ui_moments"),
   ui.input_text("moments","Please write the right side of the equation",placeholder="Ente
   ui.output_ui("ui_typeset"),
   ui.input_action_button("typesetMath", "See Updated Equations", class_="btn-secondary"),
  ui.input_action_button("solveEquations", "Solve Equations", class_="btn-success"),
   ui.output_ui("ui_solutions"),
   ui.output_ui("ui_typesetSolutions")
```

```
)

def server(input, output, session):
    @output
    @render.ui
    def dyn_ui_forces():
      mystring_forces="$$ F_y="+input.forces()+"$$"
      mystring_forces=mystring_forces.replace("*", "\\times")
      return ui.markdown(mystring_forces)

    @output
    @render.ui
    def dyn_ui_moments():
        mystring_moments="$$ M_O="+input.moments()+"$$"
        mystring_moments=mystring_moments.replace("*", "\\times")
        return ui.markdown(mystring_moments)

    @output
    @render.ui
    @reactive.event(input.typesetMath, ignore_none=False)
    def ui_typeset():
        return ui.tags.script(
            "if (window.MathJax) MathJax.Hub.Queue(['Typeset', MathJax.Hub]);"
        )

    @output
    @render.ui
    @reactive.event(input.solveEquations, ignore_none=False)
    def ui_solutions():
      solvedForces=solve(Eq(F,parse_expr(input.forces())),F)
      solvedMoments=solve(Eq(M,parse_expr(input.moments())),M)
      mystring_solvedForces="$$ F_O="+str(solvedForces[0])+"$$"
      mystring_solvedMoments="$$ M_O="+str(solvedMoments[0])+"$$"
      return ui.markdown(mystring_solvedForces),ui.markdown(mystring_solvedMoments)


    @output
    @render.ui
    @reactive.event(input.solveEquations, ignore_none=False)
    def ui_typesetSolutions():
        return ui.tags.script(
            "if (window.MathJax) MathJax.Hub.Queue(['Typeset', MathJax.Hub]);"
        )

app = App(app_ui, server)
```

2 Workout Example Solution

2.1 Worked Out Solution

This demonstrates a worked out solution to the problem. The best way to begin is by drawing a free body diagram.

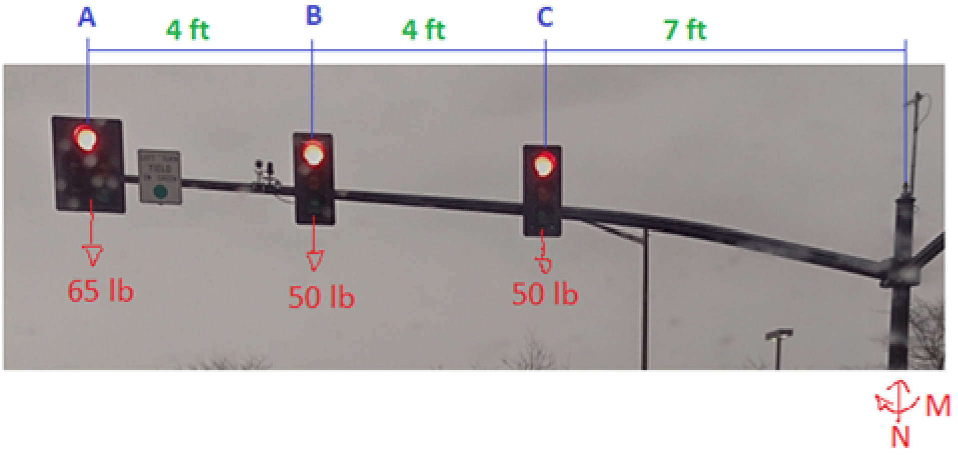

Figure 2.1: Figure 3: Three traffic light installation with loads

Use equilibrium equations to find the internal loads:

$$\Sigma F_y = 0 : N - 65 - 50 - 50 = 0$$
$$N = 165\ lbs$$

$$\Sigma M_O = 0 : -M + (50 \times 7) + (50 \times 11) + (65 \times 15) = 0$$
$$M = 1875\ lb \cdot ft = 22500\ lb \cdot in$$

Now, determine the cross-sectional properties:

$$A = \pi(r_0^2 - r_i^2) = \pi(2.5^2 - 2.3^2) = 3.02\ in^2 I = \frac{\pi}{4}(r_0^4 - r_i^4) = \frac{\pi}{4}(2.5^4 - 2.3^4) = 8.70\ in^4$$

Calculate stress due to normal force:

$$\sigma_n = \frac{F}{A} = \frac{-165\ lbs}{3.02\ in^2} = -54.7\ psi$$

Calculate maximum stress due to bending moment (will have same magnitude in both tension and compression):

$$\sigma_m = \pm\frac{M_c}{I} = \pm\frac{22500 \times 2.5}{8.70} = \pm 6460\ psi$$

Determine combined tensile stress: $\sigma_T = -54.7 + 6460 = 6410\ psi$

Determine combined compressive stress: $\sigma_T = -54.7 - 6460 = -6520\ psi$

3 Summary

In summary, this book has no content whatsoever.

```r
1 + 1
```

```
[1] 2
```

References