Strength of Materials Problem Exercises

Curated by James Lord, Jake Grohs, …

2024-01-05

Table of contents

Welcome to Demo Site

Welcome to this demonstration site of the Strength of Materials Open Problem Exercises companion to the Strength of Materials Open Textbook.

At this time, we are simply using this site as a demonstration and shell for our ongoing work. The intent is to demonstrate a more traditional static style problem exercise pack along with dynamic versions which allow students to quickly check answers and receive basic feedback, and/or to input their math in an interactive interface which will provide them with targeted feedback based on their atempted solution.

This work is still very much in progress and you may find bugs. We would welcome any input or feedback you have about this. Thanks!

Part I

Original Demo Problems

Static Problem Statement

This is a static rendering of the problem with fixed variables that correspond to the variables used in the problem solution provided.

Can i change this?

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. Please calculate the magnitude of the maximum combined stress in the post. You may ignore the weight of the post.

Figure 0.1: Figure 1: Three traffic light installation with loads

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 0.2: 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: $\sigma_T = -54.7 - 6460 = -6520\ psi$

Dynamic Problem Statement

This is a dynamic rendering of the problem with dynamic variables based on the username entered. Please note at this time that the figure displays incorrect values. This will be corrected when drawn by the graphic artist.

Problem Image

Figure 0.3: Figure 1: Three traffic light installation with loads

```
#| standalone: true
#| viewerHeight: 600
#| components: [viewer]

from shiny import App, render, ui, reactive
import random
import asyncio
import io
import math
from datetime import datetime
from pathlib import Path

problem_ID="1"
```

```
light_a=reactive.Value("__")
lights_bc=reactive.Value("__")
attempts=["Timestamp,Attempt,Answer,Feedback\n"]

app_ui = ui.page_fluid(
    ui.markdown("**Please enter your ID number from your instructor and click to generate you
    ui.input_text("ID","", placeholder="Enter ID Number Here"),
    ui.input_action_button("generate_problem", "Generate Problem", class_="btn-primary"),
    ui.markdown("**Problem Statement**"),
    ui.output_ui("ui_problem_statement"),
    ui.input_text("answer","Your Answer in units of psi", placeholder="Please enter your answ
    ui.input_action_button("submit", "Submit Answer", class_="btn-primary"),
    ui.download_button("download", "Download File to Submit", class_="btn-success"),
)


def server(input, output, session):
    @output
    @render.ui
    def ui_problem_statement():
        return[ui.markdown(f"A city planner is installing a new traffic light. Light A weighs

    @reactive.Effect
    @reactive.event(input.generate_problem)
    def randomize_vars():
        random.seed(input.ID())
        light_a.set(round(65+65*(.5-random.random())*.2))
        lights_bc.set(round(50+50*(.5-random.random())*.2))

    @reactive.Effect
    @reactive.event(input.submit)
    def _():
        instr= (light_a()+2*lights_bc()/math.pi*(2.5**2 - 2.3**2))+ ((-1*lights_bc()*7- light
        #check=math.isclose(float(input.answer()),instr,rel_tol=0.001)
        if math.isclose(float(input.answer()),instr,rel_tol=0.001):
           check="*Correct*"
        else:
           check="*Not Correct.*"

        if check=="*Not Correct.*" and math.isclose(abs(float(input.answer())),abs(instr),rel
           extra_check="An extra check says you may have a sign error."
        else:
```

```
            extra_check=""
        #extra_check = "An extra check says you may have a sign error." if math.isclose(abs(i
        feedback=ui.markdown(f"Your answer of {input.answer()} is {check} {extra_check} For r
        attempts.append(f"{datetime.now()}, {input.submit()},{input.answer()},{check}\n")
        m=ui.modal(
          feedback,
          title="Feedback",
          easy_close=True
        )
        ui.modal_show(m)

    @session.download(
        filename=lambda: f"Problem_Log-{problem_ID}-{input.ID()}.csv"
    )
    async def download():
        # This version uses a function to generate the filename. It also yields data
        # multiple times.
        await asyncio.sleep(0.25)
        yield f"{problem_ID}_{input.submit()}_{input.ID()}\n"
        yield ''.join(attempts)


app = App(app_ui, server)
```