ENGN 2020:

Homework #2

Brown University School of Engineering

Assigned: February 4, 2019, Due: February 11, 2019

Readings: Chapter 8, 20.

Turn in "paper" portions of this assignment as a single PDF upload to Canvas. A problem name such as K7-3-17 means Kreyszig (10th edition), problem #17 of Section 7.3.

Problem 1

Work through Problem K7-3-18. You should find the currents to be 2.45, 2.18, and .0.28 A; you do not need to turn your hand solution in. Write a function that solves this problem generically, for arbitrary values of R_1, R_2, R_3, ΔV_1, and ΔV_2 as labeled in Figure 1. Your code should use `numpy.linalg.solve`.

Submission

Label: hw2_1
Points: 2
Input variables: `R1`, `R2`, `R3`, `deltaV1`, `deltaV2` (all floating-point numbers)
Output: a numpy array of shape (3,1) containing I_1, I_2, I_3

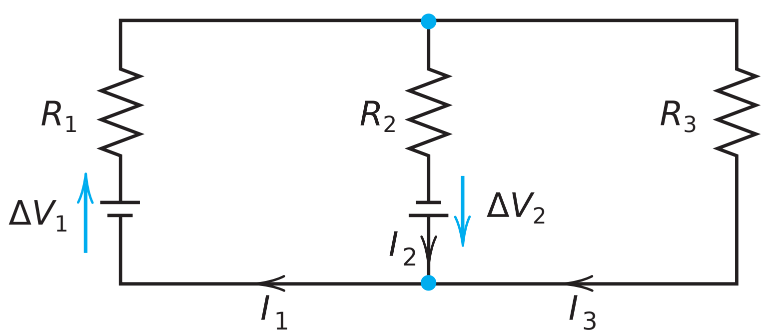

Figure 1: Labeled circuit.

Problem 2

Work through Problem K7-3-23 under the constraint that $x_1 = 1$; that is, the coefficient of the chemical being burned is one. Now, write a code to calculate the coefficients for O_2, CO_2, and H_2O for the combustion of any arbitary organic compound (that is, $C_aH_bO_c$, note this also contains oxygen). Your code should use `numpy.linalg.solve`.

> **Submission**
>
> **Label:** hw2_2
> **Points:** 1
> **Input variables:** `a`, `b`, `c` (all integers)
> **Output:** a numpy array of shape (3,1) containing x_2, x_3, x_4 (the coefficients of O_2, CO_2, and H_2O, respectively)

Problem 3

K7-3-24.

> **Paper Submission**
>
> (Turn in as part of your PDF solutions to Canvas.)
> **Points:** 2

Problem 4

K7-4-14 and K7-4-15.

> **Paper Submission**
>
> (Turn in as part of your PDF solutions to Canvas.)
> **Points:** 2

Problem 5

In this problem, we will look at the stretching of an elastic membrane, and develop a general computational solution. You should first familiarize yourself with Example 1 of Chapter 8.2.

Part a. Create a python script with a function that plots an undeformed unit circle ($x_1^2 + x_2^2 = 1$) as well as the oval that results when the unit circle has been transformed by a 2×2 matrix $\underline{\underline{\mathbf{A}}}$. That is, your plot should contain the undeformed circle and deformed oval of Figure 160 in the text. You do not need to turn in anything for part a, but will use your plot-making function in later parts of the problem.

Hint: You can create figures in python using `matplotlib.pyplot`; an example script to get you started is below. (Try googling "matplotlib thumbnail gallery" to get lots more examples.)
File attached here.

```
import numpy as np
from matplotlib import pyplot

def get_y(x, a, b, c):
    return a * x**2 + b * x + c

def plot_parabola(a, b, c, name):
    """Plots the parabola y = ax^2 + bx + c and saves it to name."""
    x_values = np.linspace(0., 15.)
    y_values = [get_y(x, a, b, c) for x in x_values]

    fig, ax = pyplot.subplots()
    ax.plot(x_values, y_values, '-g')
    ax.set_xlabel('x')
    ax.set_ylabel('y')
    fig.savefig(name)

plot_parabola(3, 2, -1, 'example-plot.pdf')
```

Part b. Write a function that takes in a 2×2 deformation matrix $\underline{\underline{\mathbf{A}}}$ and returns the two principle directions. Each eigenvector should be normalized and oriented such that the first element, corresponding to x_1, is positive.

Submission

Label: hw2_5b
Points: 1
Input variables: `A` (a 2×2 numpy array)
Output: a 2×2 numpy array containing an eigenvector in each column

Next, modify your function from part a to find and add these two principle directions to your figure, as in the textbook's Figure 160.

Part c. Use the function that you have created (that plots the deformed and undeformed shapes as well as the principle axes) to solve problems 1 through 6 of Section 8.2. Turn in one figure for each numbered problem. You do not need to turn in numerical values of eigenvectors if your figures are clear.

Paper Submission

(Turn in as part of your PDF solutions to Canvas.)
Points: 2