

Please answer the following questions in complete sentences in submit the solution on Blackboard Sept. 2nd, 2016.

## Homework 1

### Problem 1: G & C, Chapter 1, Problem 2 (10 points)

Please complete parts (a) and (b) of G & C, Chapter 1, Problem 2

### Problem 2: A simple economy (10 points)

(From Matrices with Applications (Hugh G. Campbell))

Suppose we construct a simple model economy in which there are three industries – the crude oil industry, the refining industry that produces gasoline, and the utility industry which supplies electricity. Then suppose there are six types of consumers – the general public, the government, the export firms, and the three industries. Both the industries and the consumers exercise certain consumptive demands on each of the industries. For instance, suppose that the crude oil industry needs to use 4 units of gasoline and 2 units of electricity. The demand vector, then for the crude oil industry, is expressed as

$$\alpha_c = (0, 4, 2).$$

Likewise, we specify the other demand vectors. In each case we list the goods demanded in the form:

$$\alpha = (\text{crude oil, gasoline, electricity}).$$

The demand vectors are:

crude oil industry	$\alpha_c = (0, 4, 2)$
refining industry	$\alpha_r = (8, 0, 6)$
utility industry	$\alpha_u = (1, 6, 0)$
general public	$\alpha_1 = (1, 9, 5)$
government	$\alpha_2 = (8, 8, 8)$
export firms	$\alpha_3 = (7, 2, 0)$

1. Write a mathematical expression for the total demands on all three industries in terms of the demand vectors  $\alpha$  and evaluate the total demands.
2. Now suppose that the price of crude oil is 4-dollars per unit, the price of gasoline is 3-dollars per unit, and the price of electricity is 2-dollars per unit. We can express this with the vector  $\beta = (4, 3, 2)$ . Assuming that the industries produce exactly what is demanded of them determine the profit of the crude oil industry.

### Problem 3: Adding acceleration to the raptor problem (15 points)

In our in-class model for the xkcd raptor problem, we ignored the impact of acceleration. Wired did a long post about acceleration of sprinters. See the url <http://www.wired.com/wiredscience/2012/08/maximum-acceleration-in-the-100-m-dash/>

1. (10 points) Using information from this post, construct a mathematical model of the position of sprinter on a line. Document and justify your assumptions
2. Determine difference in the distance traveled by the “idealized” sprinter that immediately begins running at 12 meters-per-second to the “real” sprinter that must accelerate to 12 meters-per-second after six second.

### Problem 4: Search engines (15 points)

1. Write down the matrix for G&C, Chapter 1, Problem 5. You do not have to evaluate the similarity
2. Write down the adjacency matrix for the graph in Figure 1.12
3. Evaluate the probability in G&C, Chapter 1, Problem 6a.

### Problem 5: Intro to Julia (15 points)

For this problem, we’ll use Julia instead of Matlab.

1. Please complete all parts of G & C, Chapter 2, Problem 2
2. Please complete all parts of G & C, Chapter 2, Problem 3
3. Please complete all parts of G & C, Chapter 2, Problem 7 The hint for Julia would be

**Updates based on plotly() errors** (And also one correction! Note that the coordinates typed from the book were wrong. The new coordinates are included in the update.)

**Original code** (This no longer works for me :( )

```
using Plots
plotly(size=(500,500)) # use plotly for zooming and 500-by-500 for equal size
theta = linspace(0, 2*pi, 10000)
r = sqrt(2)
x = 2. + r*cos(theta)
y = 2. + r*sin(theta)
plot(x,y)
```

**PlotlyJS code** (This one works for me, but others have had issues with it.)

```
Pkg.add("PlotlyJS")
using Plots
plotlyjs(size=(500,500))
theta = linspace(0, 2*pi, 10000)
r = sqrt(2)
x = 2. + r*cos(theta)
y = 1. + r*sin(theta)
```

```
plot(x,y)
```

**PlotlyJS code** (This one works for me, but others have had issues with it.)

```
using Plots
pyplot(size=(500,500)) # this won't give you interactivity unless you are running from the command
theta = linspace(0, 2*pi, 10000)
r = sqrt(2)
x = 2. + r*cos(theta)
y = 1. + r*sin(theta)
plot(x,y)
```

## Problem 6: Drawing the Julia set (25 points)

In this set of steps, you will write a Julia program to evaluate and draw the mandelbrot set.

1. The Julia set is a fractal. [https://en.wikipedia.org/wiki/Julia\\_set#/media/File:Julia\\_set\\_\(ice\).png](https://en.wikipedia.org/wiki/Julia_set#/media/File:Julia_set_(ice).png) Read about the Julia set, either in the textbook or on Wikipedia, or in another reference and answer the following question. The best description I found was here

<http://paulbourke.net/fractals/juliaset/>

from Wikipedia. Describe how the  $(x, y)$  coordinate of this pixel relates to the function  $\phi(z) = z^2 + c$ .

2. A key step in building the julia set fractal is checking when the sequence

$$z_0 = 0, z_{n+1} = z_n^2 + c$$

remains bounded. Implement the following Julia function

```
"""
`julia_check` is a subroutine useful for working with the Julia
set fract. The function
returns the number of steps that we can iterate the sequence
    z = zinput (from input)
    znext = z*z + c
until the maginitude of the value z becomes larger than two or
until we hit 256 steps.
"""
function julia_check(z,c)
# fill this part in
end

and report the values of

julia_check(-1, 0.5+0.5im)
julia_check(0, 0.5+0.5im)
julia_check(-1, 0.5+0.25im)
julia_check(0, 0.5+0.25im)
```

3. We are now going to look at images of the complex plane spanned by  $Re(z) = [-1.75, 1.75]$  and  $Im(z) = [-1.75, 1.75]$ . (This is the region where the julia fractal is usually drawn.) Suppose we have pixels representing 20 equally spaced points on the real axis and pixels representing 20 equally spaced points on the imaginary axis, fill in the following code

```

nxpts =
nypts =
Z = zeros(nypts, nxpts)
for jj=1:nypts
    for ii=1:nxpts
        Z(jj,ii) =
    end
end
end

```

such that  $Z(jj,ii)$  = the magnitude of the point represented by  $(j, i)$  in the complex plane. Show the result using the command

```
grayim(Z/maximum(Z))
```

**Hint** To include the output in your homework, note that you can save an image using

```

using FileIO
using ImageMagick
save("juliaset.png", Z/maximum(Z))

```

and then looking for the file juliaset in the juliabox file browser.

4. Now update your code and let the value of  $Z(jj,ii)$  be the number of iterations taken by the mandelbrot iteration to escape when that point is input to your `julia_check` function with  $c = 0.5 + 0.5i$
5. Improve the resolution of your mandelbrot picture as much as you can. (For full credit, you'll need to use at least 1,000,000 pixels (1000-by-1000).)
6. Change the value of  $c$  to 0.3 and show your new picture.
7. Which one do you like better?

## Problem 7: Fun with Julia (5 points)

Describe what you see or hear after the following commands.

1. Type the command `1:100 |> sum`, what does this do?
2. Type the command `?hi`
3. Type the command `"2 plus 2 is $(2+2)."`