DSP LAB REPORT

By Rafi Rahman

MathWorks | Training Services

Rafi Rahman

has successfully completed

100%

of the

MATLAB Onramp

self-paced training course

24-Feb-2020

25/2/2020

Lab 1

Introduction

In this lab, we explore Fourier series' by trying to solve them by hand, as well as via MATLAB.

Question 1

Part A

1. a) $a_0 = \frac{1}{\pi/2} \int_0^{\pi/2} f(\theta)\, d\theta$

$$= \frac{2}{\pi}\left(\int_0^{\pi/4} 2\, d\theta + \int_{\pi/4}^{\pi/2} -2\, d\theta\right)$$

$$= 0$$

$$\omega = \frac{2\pi}{\pi/2} = 4$$

$$a_n = \frac{1}{4/\pi} \int_0^{\pi/2} f(\theta) \cos n\omega\theta\, d\theta$$

$$= \frac{4}{\pi}\left(\int_0^{\pi/4} 2\cos 4n\theta\, d\theta + \int_{\pi/4}^{\pi/2} -2\cos 4n\theta\, d\theta\right)$$

$$= 0$$

$$b_n = \frac{1}{4/\pi} \int_0^{\pi/2} f(\theta) \sin n\omega\theta\, d\theta$$

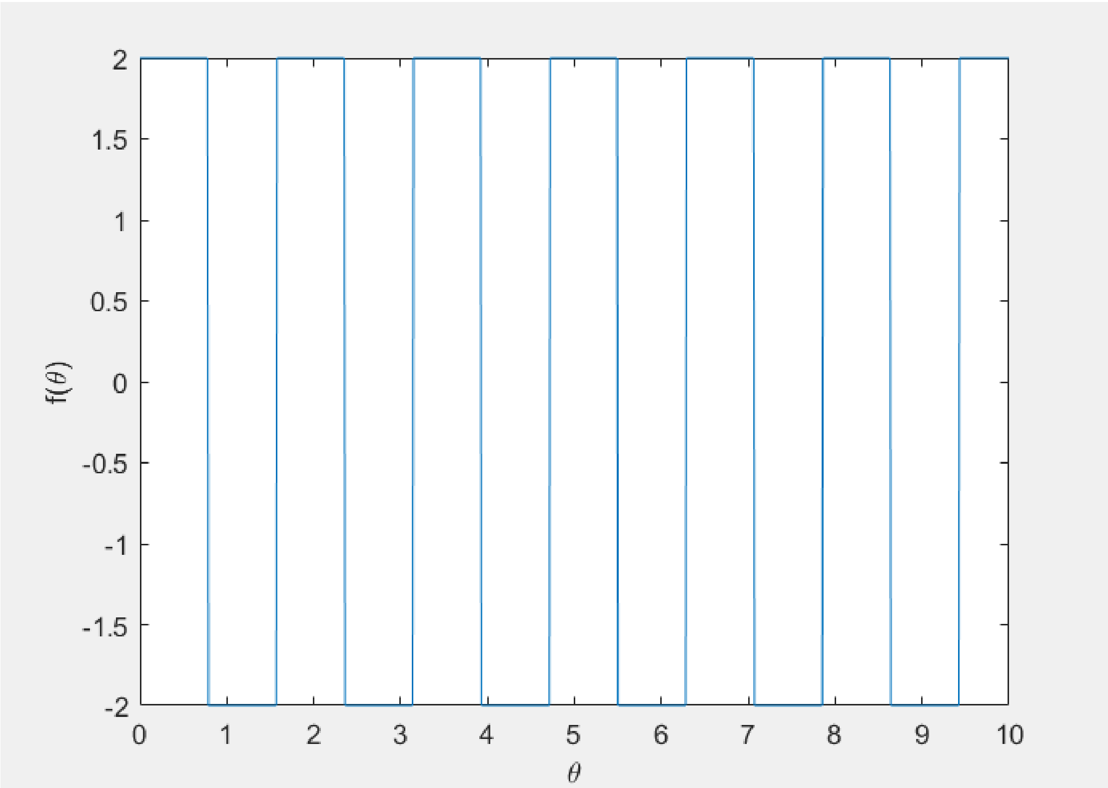
$$= \frac{4}{\pi}\left(\int_0^{\pi/4} \cancel{f(\theta)}\, 2\sin 4n\theta\, d\theta + \int_{\pi/4}^{\pi/2} -2\sin 4n\theta\, d\theta\right)$$

$$= \frac{8}{\pi}\left(\frac{-\cos n\pi + \cos\theta + \cos 2n\pi - \cos n\pi}{4n}\right)$$

$$= \begin{cases} \text{if } n \text{ even} \to 0 \\ \text{if } n \text{ odd} \to \frac{8}{n\pi} \\ \text{even } n \text{ in } \cos \to \text{Positive} \\ \text{odd } n \text{ in } \cos \to \text{Negative} \end{cases}$$

Part B

```
T = pi/2;
w = 2*pi/T;
t = 0:0.01:10;
F = 2*square(w*t);
plot(t,F)
```



Part C

```
syms t
T = pi/2;
n = 1:5;
a0 = (1/T)*int(2,t,0,pi/4) - (1/T)*int(2,t,pi/4,pi/2)
an = (2/T)*int(2*cos(n*t*2*pi/T),t,0,pi/4) -
(2/T)*int(2*cos(n*t*2*pi/T),t,pi/4,pi/2)
bn = (2/T)*int(2*sin(n*t*2*pi/T),t,0,pi/4) -
(2/T)*int(2*sin(n*t*2*pi/T),t,pi/4,pi/2)
F = a0;
for i = 1:length(an)
      F = F + an(1,i)*cos(i*t) + bn(1,i)*sin(i*t);
end
F
```

Result:

a0 = 0

an = [0,0,0,0,0]

bn = [5734161139222659/2251799813685248,0,1911387046407553/2251799813685248,0,5734161139222659/11258999068426240]

F = (1911387046407553*sin(3*t))/2251799813685248 + (5734161139222659*sin(5*t))/11258999068426240 + (5734161139222659*sin(t))/2251799813685248

Question 2

Part A

2. a) $f(t) = e^{-\frac{t}{2}}$, $T = \pi$, $\omega = \frac{2\pi}{\pi} = 2$

$\cancel{a_0 = \frac{1}{T}\int_0^T \frac{e^{-t/2}}{-\frac{1}{2}}}$ $\quad a_0 = \frac{1}{T}\int_0^T f(t)\,dT$

$$a_0 = \frac{\cancel{2}}{\pi}\left[\frac{e^{-t/2}}{-\frac{1}{2}}\right]_0^{\pi} = \frac{1}{\pi}\left[\frac{e^{-\frac{\pi}{2}}}{-\frac{1}{2}} - \frac{e^{0}}{-\frac{1}{2}}\right]$$

$$a_n = \frac{2}{\pi}\int_0^{\pi} f(t)\cos n\omega t\,dt$$

$$= \frac{2}{\pi}\int_0^{\pi} e^{-t/2}\cos 2nt\,dt$$

$$= \frac{2}{\pi}\left[\left(\frac{-2e^{-\frac{\pi}{2}}}{1+16n^2}\times 1 - \frac{2}{1+16n^2}\right)\right]$$

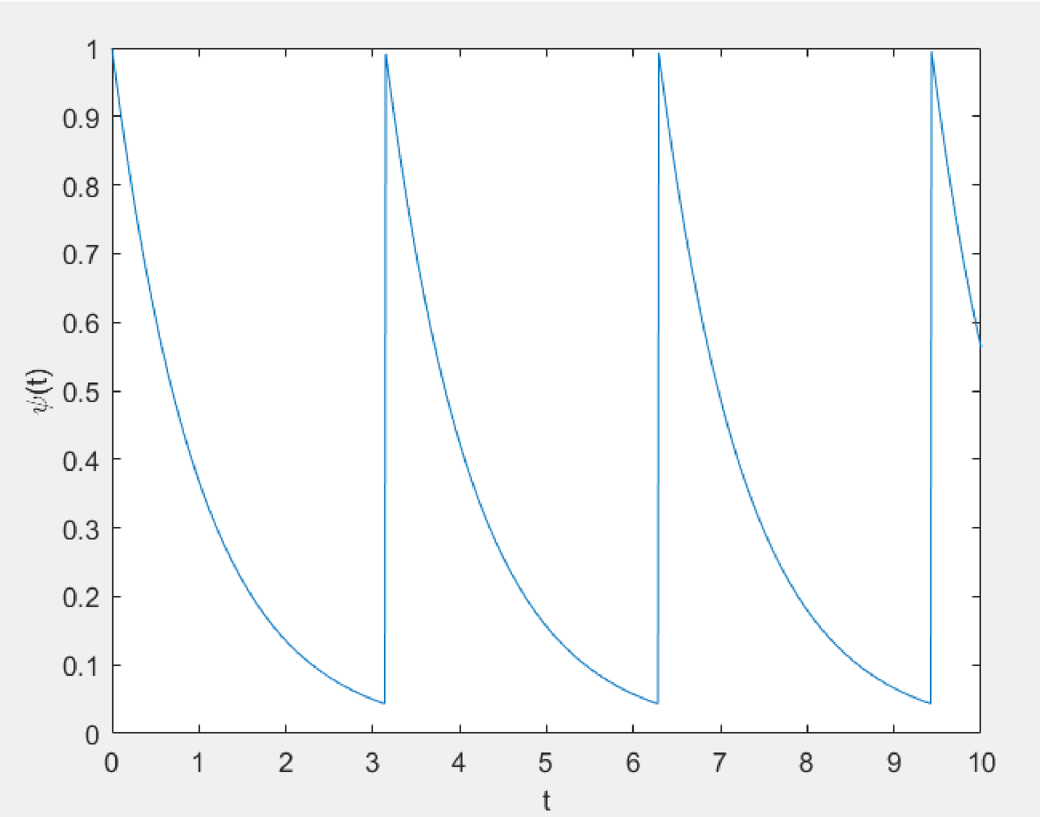
$$b_n = \frac{2}{\pi}\left[\frac{e^{-t/2}}{\frac{1}{4}+4n^2}\left(-\frac{1}{2}\sin 2nt - 2n\cos 2nt\right)\right]_0^{\pi}$$

$$= \frac{2}{\pi}\left[\frac{8n}{1+16n^2}\left(-e^{-\frac{\pi}{2}}+1\right)\right]$$

u3191010

Part B

```
T = pi;
w = 2*pi/T;
t = 0:0.01:10;
F = zeros(1001,1);
for n = 1:1001
      F(n,1) = exp(-rem((n - 1)/100,pi));
end
plot(t,F)
```



Part C

```
syms n t
T0 = pi;
w0 = 2*pi/10;
n = 1:5;
a0 = (1/T0)*int(exp(-t/2),t,0,pi);
an = (2/T0)*int(exp(-t/2)*cos(n*w0*t),t,0,pi);
bn = (2/T0)*int(exp(-t/2)*sin(n*w0*t),t,0,pi);
A = a0;
for i = 1:length(an)
      A = A + an(1,i)*cos(i*t) + bn(1,i)*sin(i*t)
end
```

u3191010

Result:

```
a0 = 5734161139222659/9007199254740992 - (5734161139222659*exp(-
pi/2))/9007199254740992

an = [5734161139222659/(18014398509481984*(pi^2/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(cos(pi^2/5)/2 -
(pi*sin(pi^2/5))/5))/(9007199254740992*(pi^2/25 + 1/4)),
5734161139222659/(18014398509481984*((4*pi^2)/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(cos((2*pi^2)/5)/2 -
(2*pi*sin((2*pi^2)/5))/5))/(9007199254740992*((4*pi^2)/25 + 1/4)),
5734161139222659/(18014398509481984*((9*pi^2)/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(cos((3*pi^2)/5)/2 -
(3*pi*sin((3*pi^2)/5))/5))/(9007199254740992*((9*pi^2)/25 + 1/4)),
5734161139222659/(18014398509481984*((16*pi^2)/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(cos((4*pi^2)/5)/2 -
(4*pi*sin((4*pi^2)/5))/5))/(9007199254740992*((16*pi^2)/25 + 1/4)),
5734161139222659/(18014398509481984*(pi^2 + 1/4)) -
(5734161139222659*exp(-pi/2)*(cos(pi^2)/2 -
pi*sin(pi^2)))/(9007199254740992*(pi^2 + 1/4))]

bn = [(5734161139222659*pi)/(45035996273704960*(pi^2/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(sin(pi^2/5)/2 +
(pi*cos(pi^2/5))/5))/(9007199254740992*(pi^2/25 + 1/4)),
(5734161139222659*pi)/(22517998136852480*((4*pi^2)/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(sin((2*pi^2)/5)/2 +
(2*pi*cos((2*pi^2)/5))/5))/(9007199254740992*((4*pi^2)/25 + 1/4)),
(17202483417667977*pi)/(45035996273704960*((9*pi^2)/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(sin((3*pi^2)/5)/2 +
(3*pi*cos((3*pi^2)/5))/5))/(9007199254740992*((9*pi^2)/25 + 1/4)),
(5734161139222659*pi)/(11258999068426240*((16*pi^2)/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(sin((4*pi^2)/5)/2 +
(4*pi*cos((4*pi^2)/5))/5))/(9007199254740992*((16*pi^2)/25 + 1/4)),
(5734161139222659*pi)/(9007199254740992*(pi^2 + 1/4)) -
(5734161139222659*exp(-pi/2)*(sin(pi^2)/2 +
pi*cos(pi^2)))/(9007199254740992*(pi^2 + 1/4))]

F = sin(2*t)*((5734161139222659*pi)/(22517998136852480*((4*pi^2)/25
+ 1/4)) - (5734161139222659*exp(-pi/2)*(sin((2*pi^2)/5)/2 +
(2*pi*cos((2*pi^2)/5))/5))/(9007199254740992*((4*pi^2)/25 + 1/4))) -
(5734161139222659*exp(-pi/2))/9007199254740992 +
sin(4*t)*((5734161139222659*pi)/(11258999068426240*((16*pi^2)/25 +
1/4)) - (5734161139222659*exp(-pi/2)*(sin((4*pi^2)/5)/2 +
(4*pi*cos((4*pi^2)/5))/5))/(9007199254740992*((16*pi^2)/25 + 1/4)))
+ cos(5*t)*(5734161139222659/(18014398509481984*(pi^2 + 1/4)) -
(5734161139222659*exp(-pi/2)*(cos(pi^2)/2 -
```

```
pi*sin(pi^2)))/(9007199254740992*(pi^2 + 1/4))) +
cos(t)*(5734161139222659/(18014398509481984*(pi^2/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(cos(pi^2/5)/2 -
(pi*sin(pi^2/5))/5))/(9007199254740992*(pi^2/25 + 1/4))) +
sin(5*t)*((5734161139222659*pi)/(9007199254740992*(pi^2 + 1/4)) -
(5734161139222659*exp(-pi/2)*(sin(pi^2)/2 +
pi*cos(pi^2)))/(9007199254740992*(pi^2 + 1/4))) +
cos(2*t)*(5734161139222659/(18014398509481984*((4*pi^2)/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(cos((2*pi^2)/5)/2 -
(2*pi*sin((2*pi^2)/5))/5))/(9007199254740992*((4*pi^2)/25 + 1/4))) +
cos(3*t)*(5734161139222659/(18014398509481984*((9*pi^2)/25 + 1/4)) -
(5734161139222659*exp(-pi/2)*(cos((3*pi^2)/5)/2 -
(3*pi*sin((3*pi^2)/5))/5))/(9007199254740992*((9*pi^2)/25 + 1/4))) +
cos(4*t)*(5734161139222659/(18014398509481984*((16*pi^2)/25 + 1/4))
- (5734161139222659*exp(-pi/2)*(cos((4*pi^2)/5)/2 -
(4*pi*sin((4*pi^2)/5))/5))/(9007199254740992*((16*pi^2)/25 + 1/4)))
+ sin(t)*((5734161139222659*pi)/(45035996273704960*(pi^2/25 + 1/4))
- (5734161139222659*exp(-pi/2)*(sin(pi^2/5)/2 +
(pi*cos(pi^2/5))/5))/(9007199254740992*(pi^2/25 + 1/4))) +
sin(3*t)*((17202483417667977*pi)/(45035996273704960*((9*pi^2)/25 +
1/4)) - (5734161139222659*exp(-pi/2)*(sin((3*pi^2)/5)/2 +
(3*pi*cos((3*pi^2)/5))/5))/(9007199254740992*((9*pi^2)/25 + 1/4))) +
5734161139222659/9007199254740992
```

Conclusion

From the activities in this lab, I got practice in solving Fourier series' and learned how to work with them in MATLAB.

3/3/2020

Lab 2

Introduction

In this lab, we continue to look at Fourier series', and we also look at DTFT.

Question 1

$$1.\ x_a(t) = e^{-1000|t|}$$

$$X(e^{j\omega}) = \int_{-\infty}^{\infty} x_a(t)\, e^{-j\omega t}\, dt$$

$$= \int_{-\infty}^{0} e^{1000t} \cdot e^{-j\omega t}\, dt + \int_{0}^{\infty} e^{1000t} \cdot e^{-j\omega t}\, dt$$

$$= \int_{-\infty}^{0} e^{(1000-j\omega)t}\, dt + \int_{0}^{\infty} e^{(1000-j\omega)t}\, dt$$

$$= \frac{1}{1000+j\omega} + \frac{1}{1000-j\omega}$$

$$= \frac{2000}{(1000)^2-(j\omega)^2}$$

Question 2

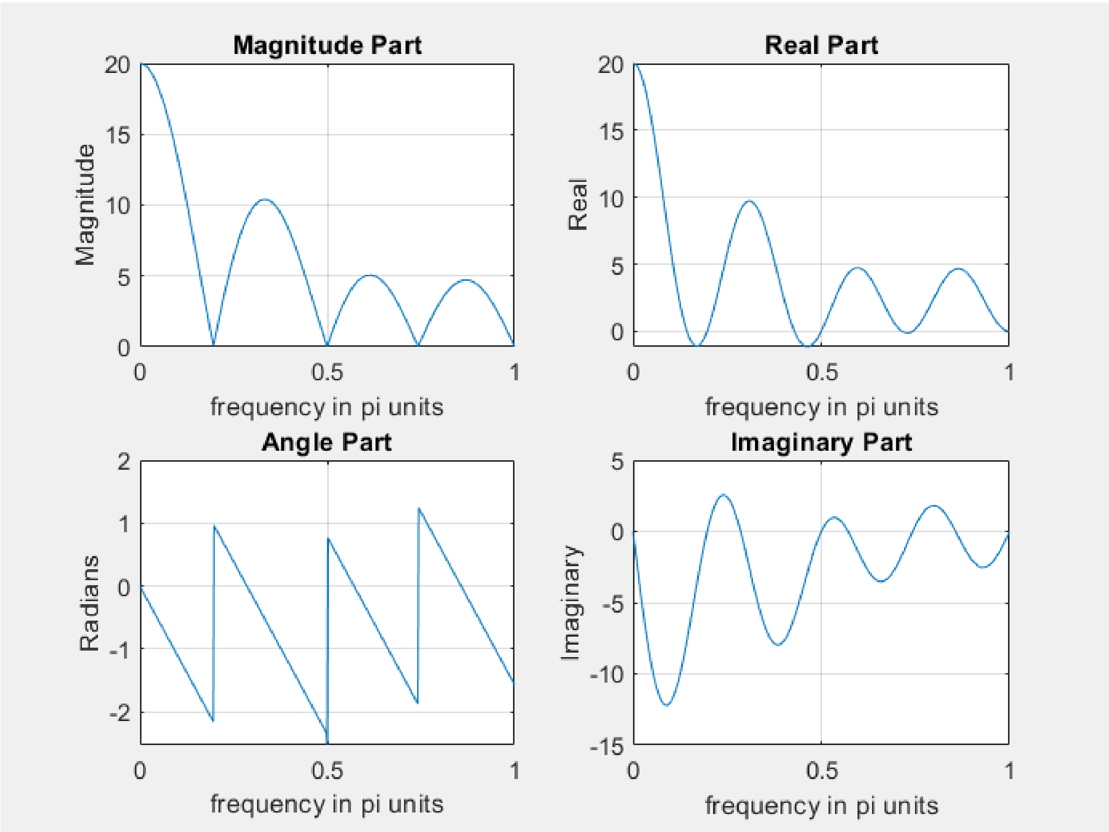
$$X(e^{j\omega}) = \sum_{-\infty}^{\infty} x(n)\, e^{-j\omega n}$$

$$= 4e^{-j\omega(0)} + 3e^{-j\omega(1)} + 2e^{-j\omega(2)} + e^{-j\omega(3)} + e^{-j\omega(4)} + 2e^{-j\omega(5)} + 3e^{-j\omega(6)} + 4e^{-j\omega(7)}$$

$$= 4 + 3e^{-j\omega} + 2e^{-2j\omega} + e^{-3j\omega} + e^{-4j\omega} + 2e^{-5j\omega} + 3e^{-6j\omega} + 4e^{-7j\omega}$$

```
w = [0:1:500]*pi/500;
X = 4 + 3*exp(-j*w) + 2*exp(-j*w*2) + exp(-j*w*3) + exp(-j*w*4) +
2*exp(-j*w*5) + 3*exp(-j*w*6) + 4*exp(-j*w*7);
magX= abs(X);
angX= angle(X);
realX= real(X);
imagX= imag(X);
subplot(2,2,1);
plot(w/pi,magX);
```

```
grid
xlabel('frequency in pi units'); title('Magnitude Part');
ylabel('Magnitude')
subplot(2,2,3);
plot(w/pi,angX);
grid
xlabel('frequency in pi units'); title('Angle Part');
ylabel('Radians')
subplot(2,2,2);
plot(w/pi,realX);
grid
xlabel('frequency in pi units'); title('Real Part');
ylabel('Real')
subplot(2,2,4);
plot(w/pi,imagX);
grid
xlabel('frequency in pi units'); title('Imaginary Part');
ylabel('Imaginary')
```



Question 3

3. $x(n) = \{4, 1, -1, 1\} \quad N = 4$

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}nk}$$

$$= \sum_{n=0}^{3} x(n)(-j)^{nk}$$

$$X(0) = \sum_{n=0}^{3} x(n)$$

$$= 4 + 1 - 1 + 1$$

$$= 5$$

$$X(1) = \sum_{n=0}^{3} x(n)(-j)^{n}$$

$$= 4(-j)^0 + 1(-j)^1 - 1(-j)^2 + 1(-j)^3$$

$$= 4 - j + 1 + j$$

$$= 5$$

$$X(2) = \sum_{n=0}^{3} x(n)(-j)^{2n}$$

$$= 4(-j)^0 + 1(-j)^{2(1)} - 1(-j)^{2(2)} + 1(-j)^{2(3)}$$

$$= 4 - 1 - 1 - 1$$

$$= 1$$

$$X(3) = \sum_{n=0}^{3} x(n)(-j)^{3n}$$

$$= 4(-j)^0 + 1(-j)^{3(1)} - 1(-j)^{3(2)} + 1(-j)^{3(3)}$$

$$= 4 + j + 1 - j$$

$$= 5$$

```
a = zeros(1,4);
x = [4,1,-1,1];
X = [];
for k = 0:3
for n = 0:3
a(1,n + 1) =x(1,n + 1)*(-j)^(n*k);
end
X = [X,sum(a)]
end
```

Result:

```
X = 5

X = 5 5

X = 5 5 1

X = 5 5 1 5
```

Conclusion

From this lab, I got further practice with Fourier series' and learned how to work with DTFT, both on paper and on MATLAB.

C.K

10/3/2020

Lab 3

Introduction

Here, we dive into Z-transforms and learn to work with them on MATLAB.

Question 1

Part A

1. a) $y(n) = 0.9y(n-1) + x(n)$

$$\cancel{y(z) = 0.9} \quad Y(z) = 0.9Y(z) \cdot z^{-1} + X(z)$$

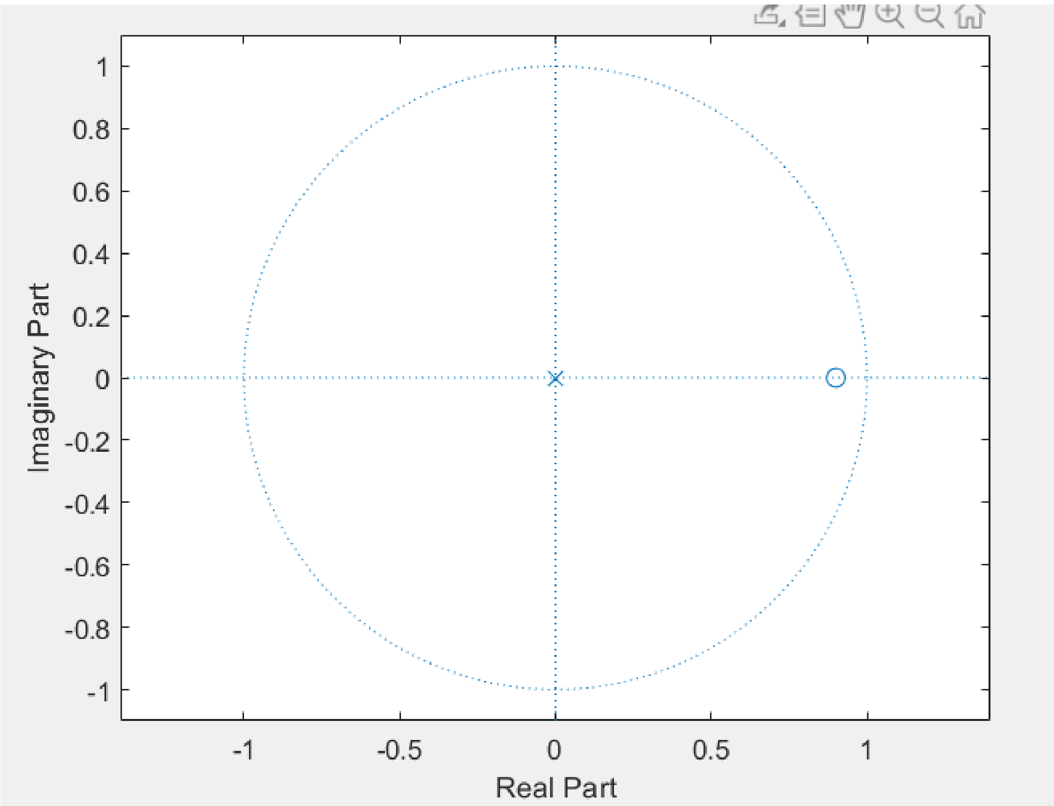
$$H(z) = \frac{Y(z)}{X(z)}$$

$$= \frac{1}{1 - 0.9z^{-1}}$$

$$= \frac{z}{z - 0.9}$$

Part B

```
a = [1 -0.9];
b = [1];
zplane(a,b)
```



Part C

c) Poles: $z - 0.9 = 0$

$z = 0.9$

Zeroes: $z = 0$

stable, because poles are inside the unit disc.

Question 2

2. $H(z) = \dfrac{z+1}{z^2 - 0.9z + 0.81}$

$$\frac{Y(z)}{X(z)} = \frac{z+1}{z^2 - 0.9z + 0.81}$$

$$Y(z)\left[z^2 - 0.9z + 0.81\right] = X(z)\left[z+1\right]$$

$$Y(z)z^2 - Y(z)0.9z + Y(z)0.81$$
$$= X(z)z + X(z)$$

$$y(n+2) - 0.9y(n+1) + 0.81y(n) = x(n+1)$$
$$+ x(n)$$

Question 3

3-a) $x(n) = \{3, 2, 1, -2, -3\}$
$\quad\quad [-2 \quad -1 \quad 0 \quad 1 \quad 2]$

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

$$= 3z^2 + 2z + 1 - 2z^{-1} - 3z^{-2}$$

ROC: $|z| < \infty$, $z \neq 0$

$$= \frac{3}{z^2} + \frac{2}{z} + 1 - 2z - 3z^2$$

$$= \frac{3 + 2z + z^2 - 2z^2 - 3z^4}{z^2}$$

Conclusion

From this lab, I got practice with Z-transforms and aspects of it such as pole-zero plots and determining system stability. I learned how to find the difference equation knowing *H(z)*. Last but not least, I also learned how to find the region of convergence for a sequence.

17/3/2020

Lab 4

Introduction

In this lab, we explore Z-transforms even further, especially within the context of MATLAB.

Question 1

Part A

1. a) $H(z) = \frac{Y(z)}{X(z)} = \cancel{0.02554z^2 + 0.07528z}$

$$= \frac{0.02554z^2 + 0.07528z + 0.01412}{z^3 - 1.849z^2 + 1.27z - 0.3058}$$

$$Y(z)(z^3 - 1.849z^2 + 1.27z - 0.3058)$$
$$= X(z)(0.02554z^2 + 0.07528z + 0.01412)$$

$$y(n+3) - 1.849y(n+2) + 1.27y(n+1) - 0.3058y(n) = 0.02554x(n+2) + 0.07528x(n+1) + 0.01412x(n)$$

$$\Downarrow$$

$$y(n) - 1.849y(n-1) + 1.27y(n-2) - 0.3058y(n-3) = 0.02554x(n-1) + 0.07528x(n-2) + 0.01412x(n-3)$$

Part B

```
Poles = roots([1 -1.849 1.27 -0.3059])
Zeros = roots([0.02554 0.07528 0.01412])
```

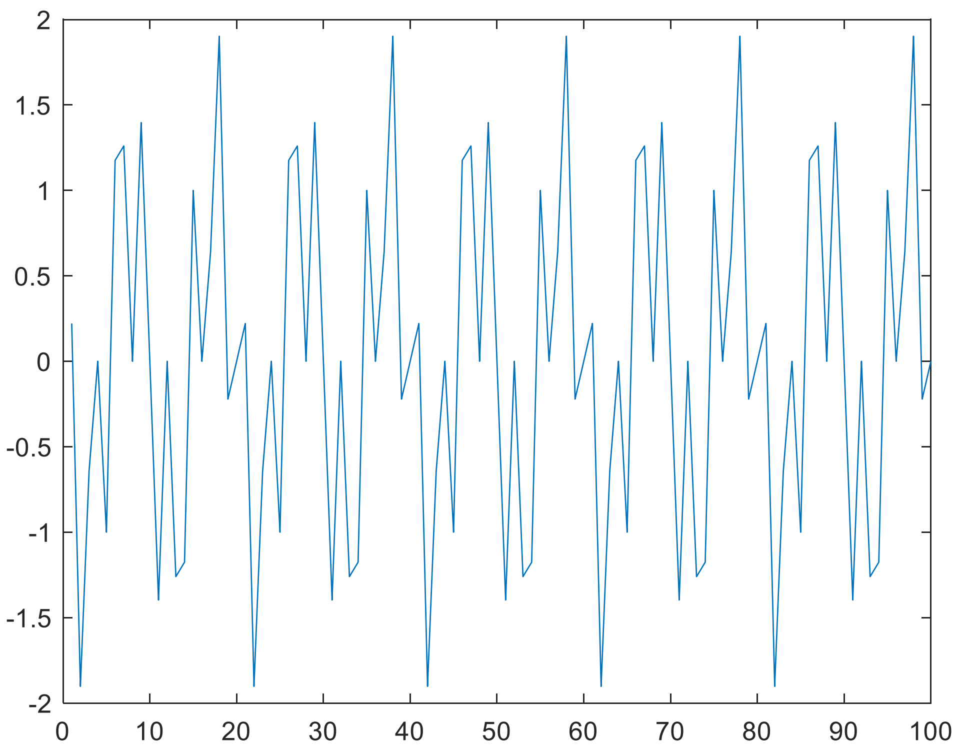
Result:

```
poles =
   0.6483 + 0.3653i
   0.6483 - 0.3653i
   0.5524 + 0.0000i

zeros =
   -2.7462
   -0.2013
```

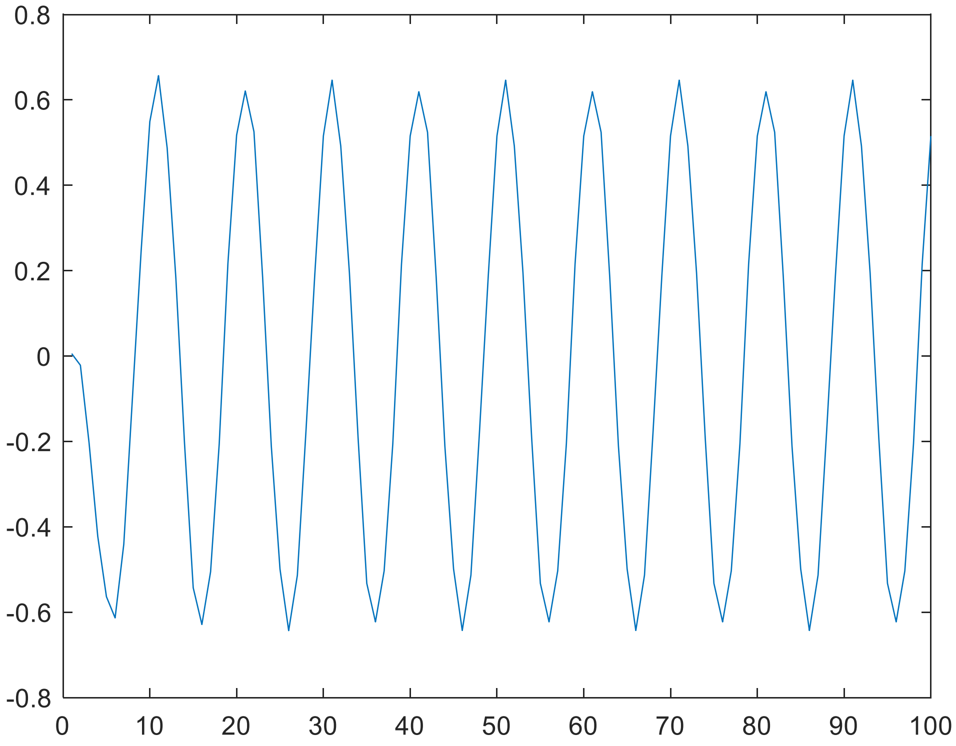
Part C

```
n = 1:100;
x(n) = sin(3.8*pi*n) + sin(2.7*pi*n);
plot(n,x)
```



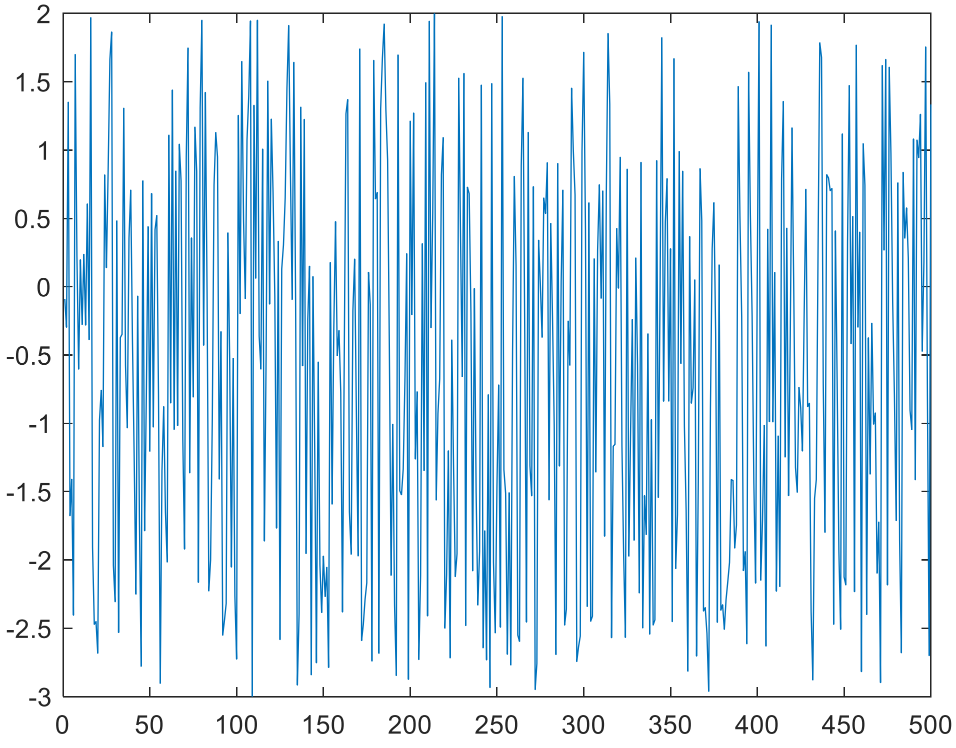
Part D

```
a = [1 -1.849 1.27 -0.3058];
b = [0.02554 0.07528 0.01412];
y = filter(b,a,x);
plot(n,y)
```



Question 2

```
X = rand(1,500)*5 - 3;
plot(X)
```



Question 3

Part A

$$h(n) = \left[\frac{1}{8}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}\right]$$
$$\quad -1 \quad 0 \quad 1 \quad 2 \quad 3$$

$$H(z) = \sum_{-\infty}^{\infty} h(n) \cdot z^{-n}$$

$$= \frac{1}{8} z + \frac{1}{4} + \frac{1}{4} z^{-1} + \frac{1}{4} z^{-2} + \frac{1}{8} z^{-3}$$

$$= \frac{z + 2 + 2z^{-1} + 2z^{-2} + z^{-3}}{8}$$

$$= \frac{z^4 + 2z^3 + 2z^2 + 2z + 1}{8z^3}$$

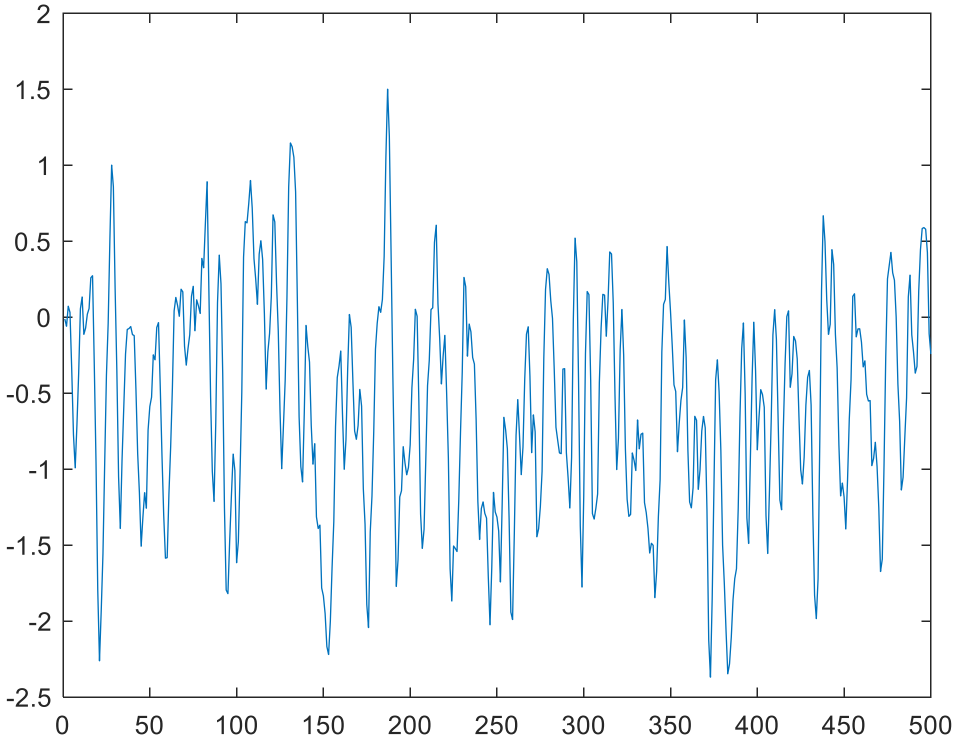
$$Y(z) \cdot 8z^3 = X(z) \cdot z^4 + X(z) \cdot 2z^3 + X(z) \cdot 2z^2 + X(z) \cdot 2z + X(z)$$

$$8y(n+3) = x(n+4) + 2x(n+3) + 2x(n+2) + 2x(n+1) + x(n)$$

$$8y(n-1) = x(n) + 2x(n-1) + 2x(n-2) + 2x(n-3) + x(n-4)$$

Part B

```
a = [8];
b = [1 2 2 2 1];
y = filter(b,a,X)
plot(y)
```



The filtering effect attempts to get rid of the noise for a cleaner and more readable signal.

Conclusion

From this lab, I got further practice with Z-transform, this time doing the inverse version of it. I also learned to work with these problems in MATLAB and about filtering noisy signals by using the filter function.

Unfortunately, I didn’t manage to get this lab signed off as this took place in week 6, and right after, the COVID mess went into full gear and uni shut down. I didn’t realise until I was putting this report together.

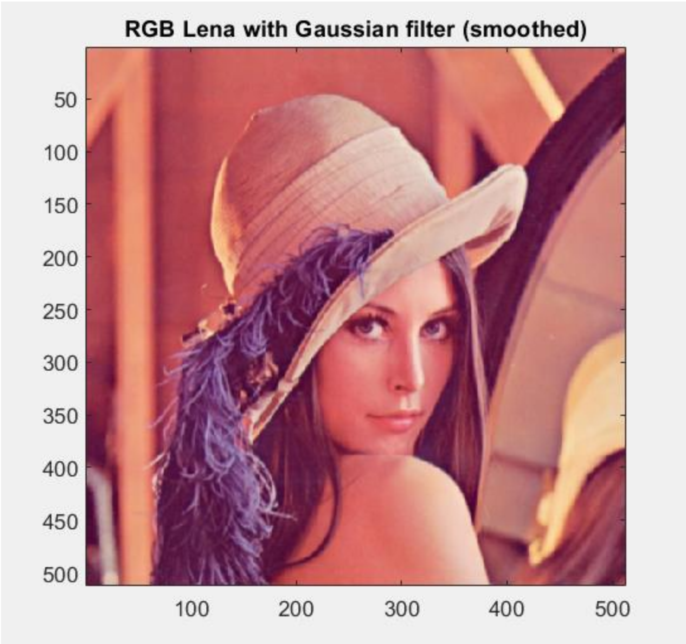
Lab 5

Introduction

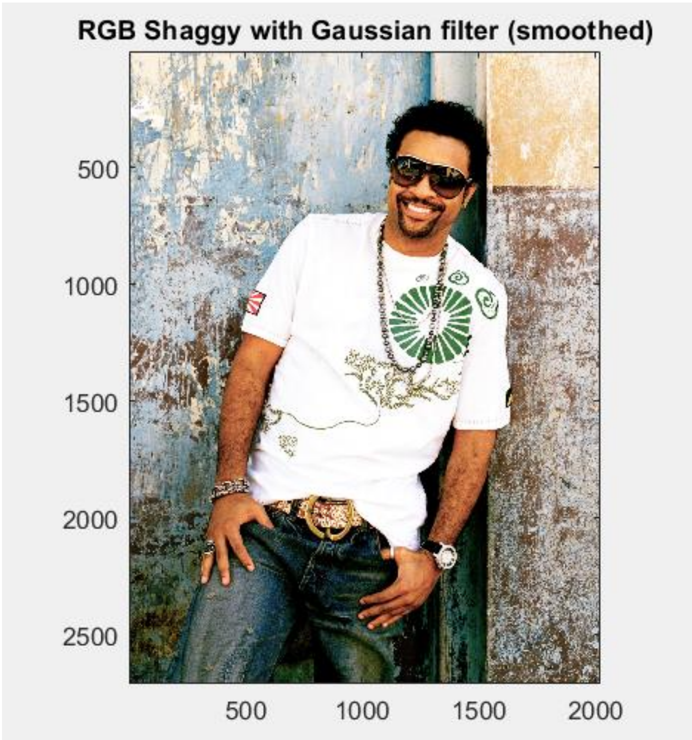
In this lab, we manipulate Lena.bmp using MATLAB’s Image Processing Toolbox.

Question 1

```
B = imgaussfilt(A,0.7);
figure;
image(B);
axis image;
title("RGB Lena with Gaussian filter (smoothed)");
```

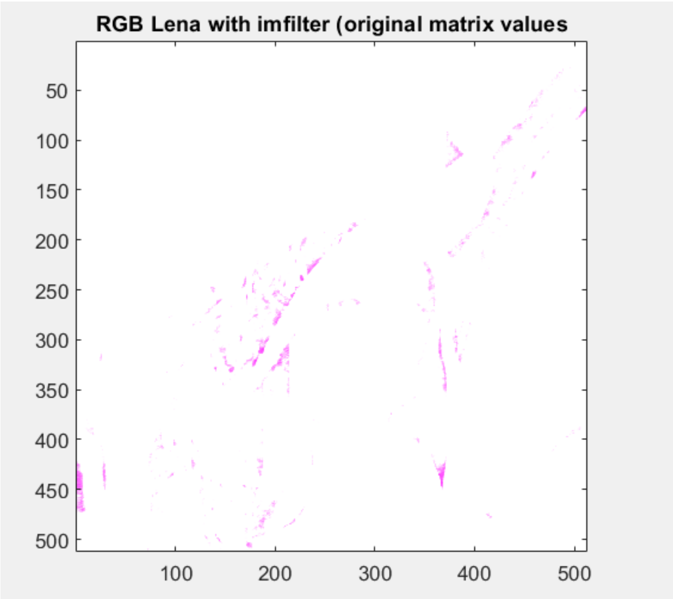


Question 2



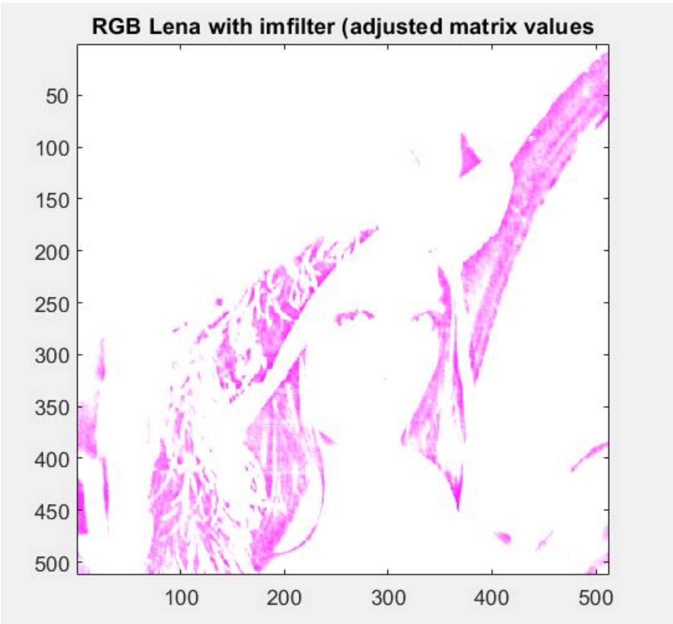
Question 3

```
h = [1 2 4;4 1 0;0 1 2];
C = imfilter(A,h);
figure;
image(C);
axis image;
title("RGB Lena with imfilter (original matrix values)");
```



Question 4

```
[1 2 4;4 1 0;0 1 2];
```



Question 5

```
Q = rgb2gray(A);
figure;
imshow(Q);
axis image;
title("RGB Lena with rgb2gray");
```

Question 6

```
F = rgb2gray(A);
figure;
imshow(F);
axis image;
edge(F);
title("Edges of RGB Lena");
```

Question 7

```
B = imgaussfilt(A,0.7);
X = A - B;
alpha = 4.5;
C = A + alpha*X;
figure;
subplot(1,2,1),image(A);
title("RGB Lena (Original)");
axis image;
subplot(1,2,2),image(C);
title("RGB Lena (Sharpened)");
axis image;
```

Conclusion

This was a very interesting lab and it felt nice to use MATLAB in a kind of real-world application such as this. I learnt how to use the functions in MATLAB's Image Processing Toolbox, as well as how to sharpen, make grayscale, and find the edges of an image through the use of different filters.

25/4/2020

Lab 6

Introduction

In this lab, we use a pre-trained CNN called SqueezeNet, an ImageNet model with a very small architecture, and feed an image to it to see how it works.

Step 1

```
net = squeezenet;
im = imread("buttercup.jpg");
imshow(im);
```

u3191010

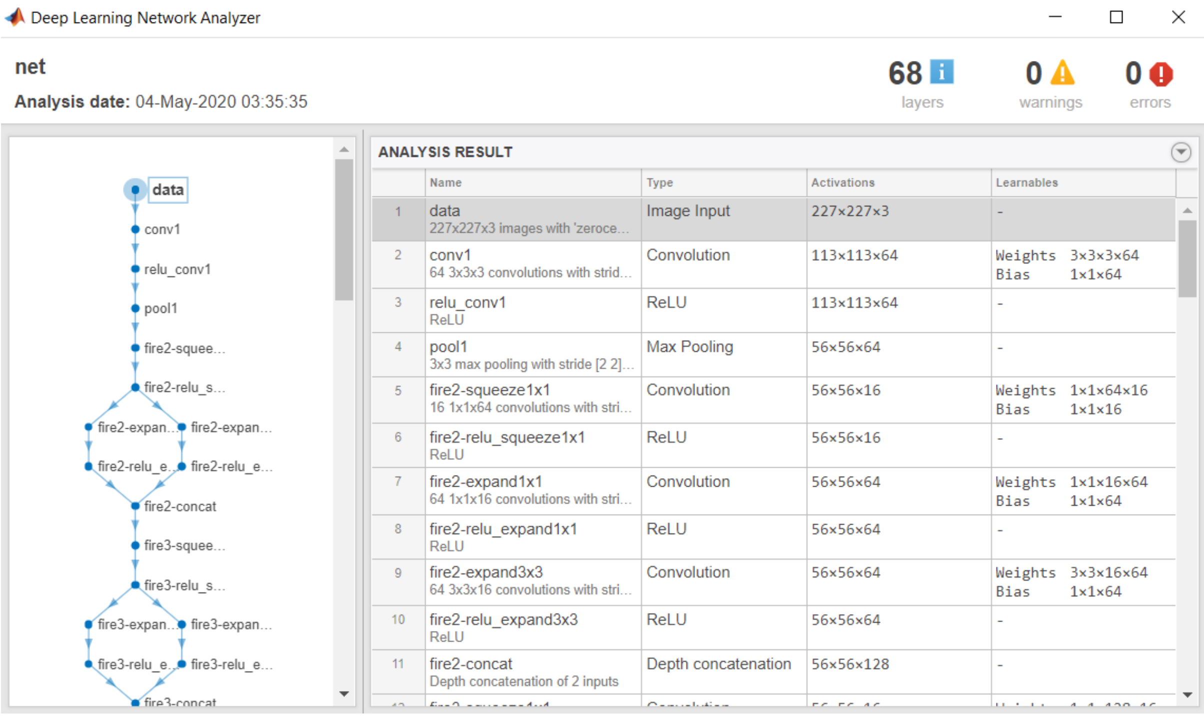
```
imgSize = size(im);
imgSize = imgSize(1:2)
```

```
imgSize =

   227   227
```

Step 2

```
analyzeNetwork(net);
```

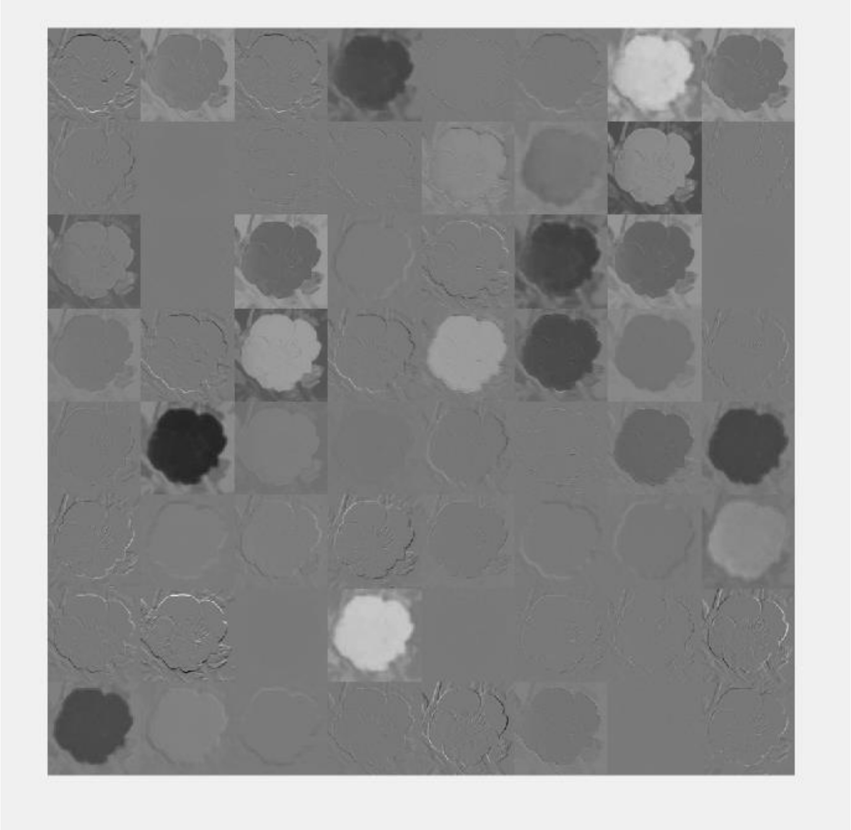


	Name	Type	Activations	Learnables
1	data 227x227x3 images with 'zeroce...	Image Input	227×227×3	-
2	conv1 64 3x3x3 convolutions with strid...	Convolution	113×113×64	Weights 3×3×3×64 Bias 1×1×64
3	relu_conv1 ReLU	ReLU	113×113×64	-
4	pool1 3x3 max pooling with stride [2 2]...	Max Pooling	56×56×64	-
5	fire2-squeeze1x1 16 1x1x64 convolutions with stri...	Convolution	56×56×16	Weights 1×1×64×16 Bias 1×1×16
6	fire2-relu_squeeze1x1 ReLU	ReLU	56×56×16	-
7	fire2-expand1x1 64 1x1x16 convolutions with stri...	Convolution	56×56×64	Weights 1×1×16×64 Bias 1×1×64
8	fire2-relu_expand1x1 ReLU	ReLU	56×56×64	-
9	fire2-expand3x3 64 3x3x16 convolutions with stri...	Convolution	56×56×64	Weights 3×3×16×64 Bias 1×1×64
10	fire2-relu_expand3x3 ReLU	ReLU	56×56×64	-
11	fire2-concat Depth concatenation of 2 inputs	Depth concatenation	56×56×128	-

Step 3

```
act1 = activations(net,im,"conv1");
```

```
sz = size(act1);
act1 = reshape(act1,[sz(1) sz(2) 1 sz(3)]);
I = imtile(mat2gray(act1), "GridSize", [8 8]);
imshow(I);
```



Step 4

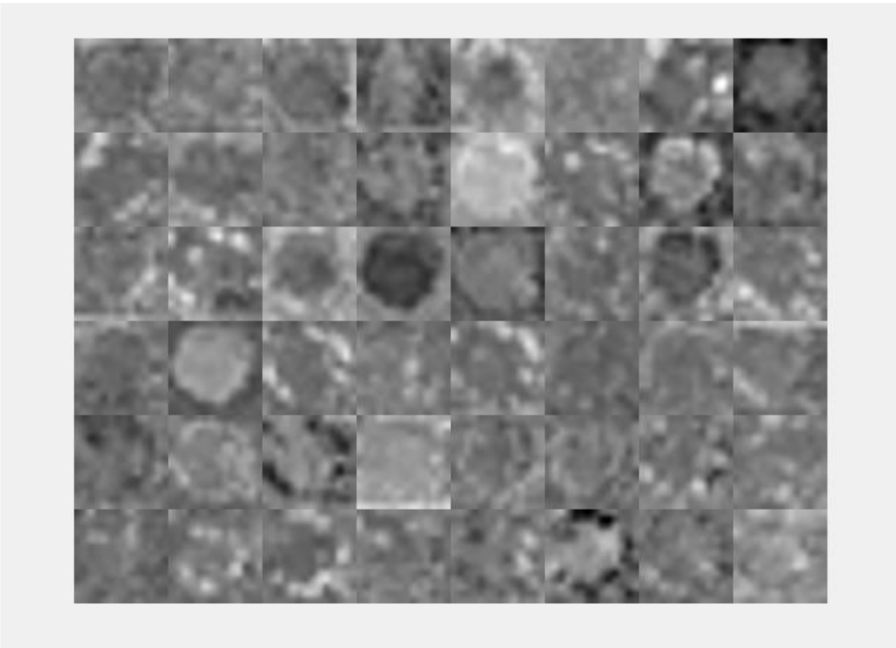
```
act1ch42 = act1(:,:,:,42);
act1ch42 = mat2gray(act1ch42);
act1ch42 = imresize(act1ch42,imgSize);
I = imtile({im,act1ch42});
imshow(I);
```

Step 5

```
[maxValue,maxValueIndex] = max(max(max(act1)));
act1chMax = act1(:,:,:,maxValueIndex);
act1chMax = mat2gray(act1chMax);
act1chMax = imresize(act1chMax,imgSize);
I = imtile({im,act1chMax});
imshow(I);
```

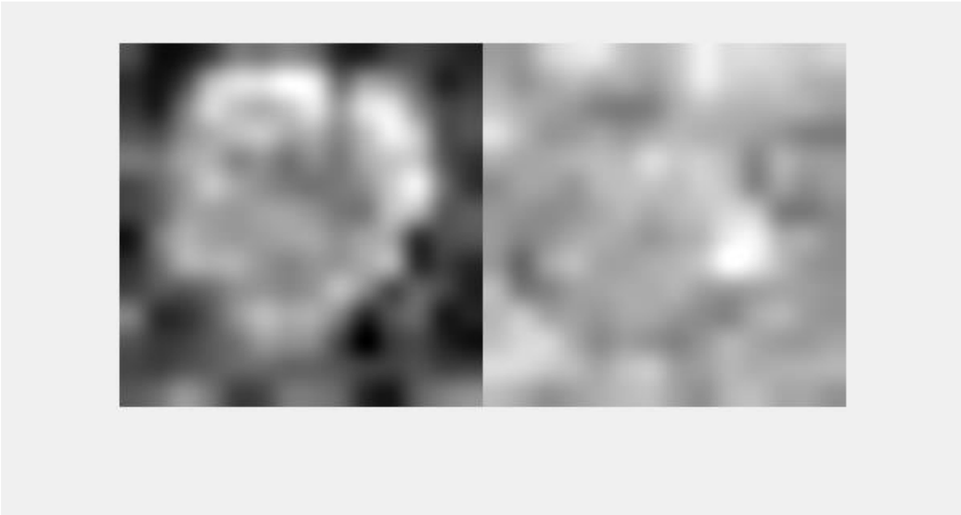
Step 6

```
act6 = activations(net, im, "fire6-squeeze1x1");
sz = size(act6);
act6 = reshape(act6, [sz(1) sz(2) 1 sz(3)]);
I = imtile(imresize(mat2gray(act6), [64 64]), "GridSize", [6 8]);
imshow(I)
```



Step 7

```
[maxValue6,maxValueIndex6] = max(max(max(act6)));
act6chMax = act6(:,:,:,maxValueIndex6);
act6chMax = mat2gray(act6chMax);
act6chMax = imresize(act6chMax,imgSize);
I = imtile({im,act6chMax});
imshow(I);
I = imtile(imresize(mat2gray(act6(:,:,:,[15 48])),imgSize));
imshow(I);
```

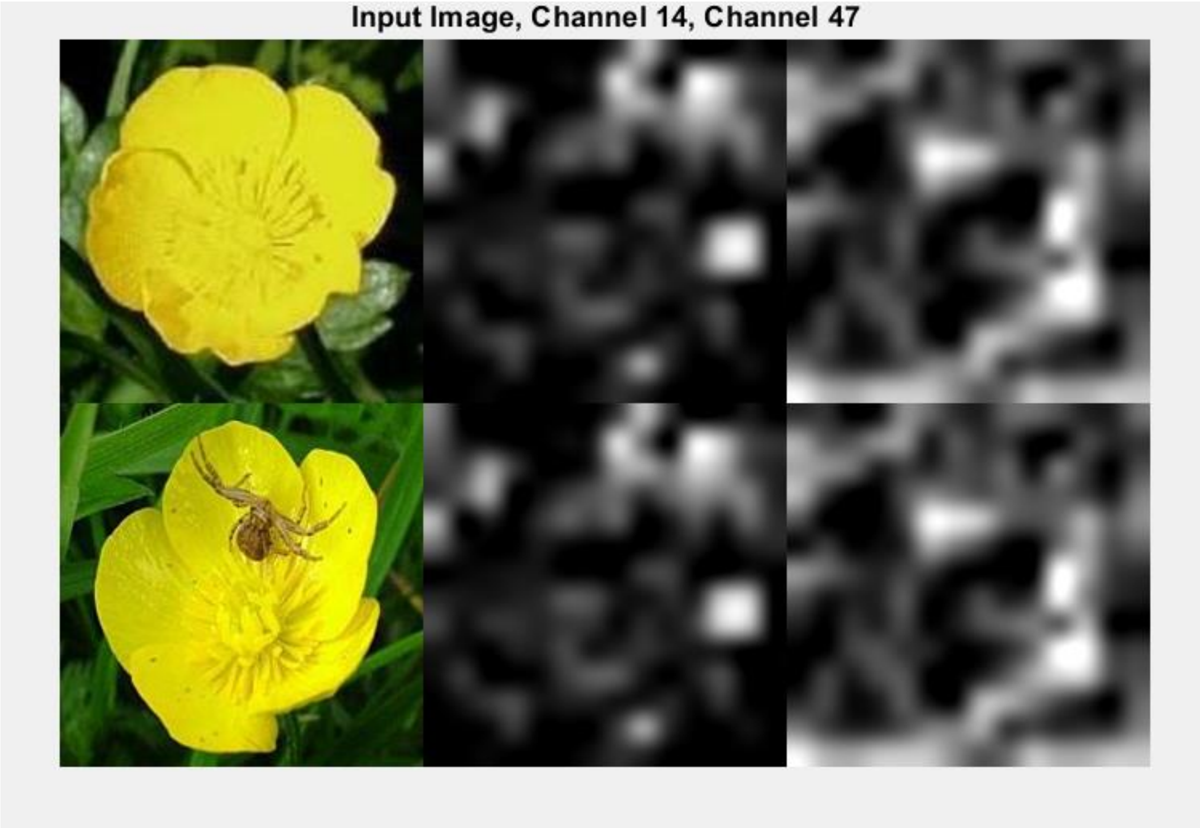


Step 8

```
imClosed = imread("buttercupSpider.jpg");
imshow(imClosed);
act6Closed = activations(net,imClosed,"fire6-relu_squeeze1x1");
sz = size(act6Closed);
act6Closed = reshape(act6Closed,[sz(1),sz(2),1,sz(3)]);
```

Step 9

```
channelsClosed = repmat(imresize(mat2gray(act6Closed(:,:,:,[14
17])),imgSize),[1 1 3]);
channelsOpen = repmat(imresize(mat2gray(act6Closed(:,:,:,[14
17])),imgSize),[1 1 3]);
I = imtile(cat(4,im,channelsOpen*255,imClosed,channelsClosed*255));
imshow(I)
title("Input Image, Channel 14, Channel 47");
```



Conclusion

From this lab, I learned how to further use MATLAB's Image Processing Toolbox, and I also learned how to use the Deep Learning Toolbox. I also got a taste of how to use a CNN, how it works, and what it is capable of.

Lab 7

Introduction

In this lab, we look at different data clustering methods and use them in MATLAB.

Question 1

Part A

```
x = rand(100,2)*100
```

Result:

```
x =
   81.4724   16.2182
   90.5792   79.4285
   12.6987   31.1215
   91.3376   52.8533
   63.2359   16.5649
   ...       ...
```

Part B

```
k = 3;
idx = kmeans(x,k)
```

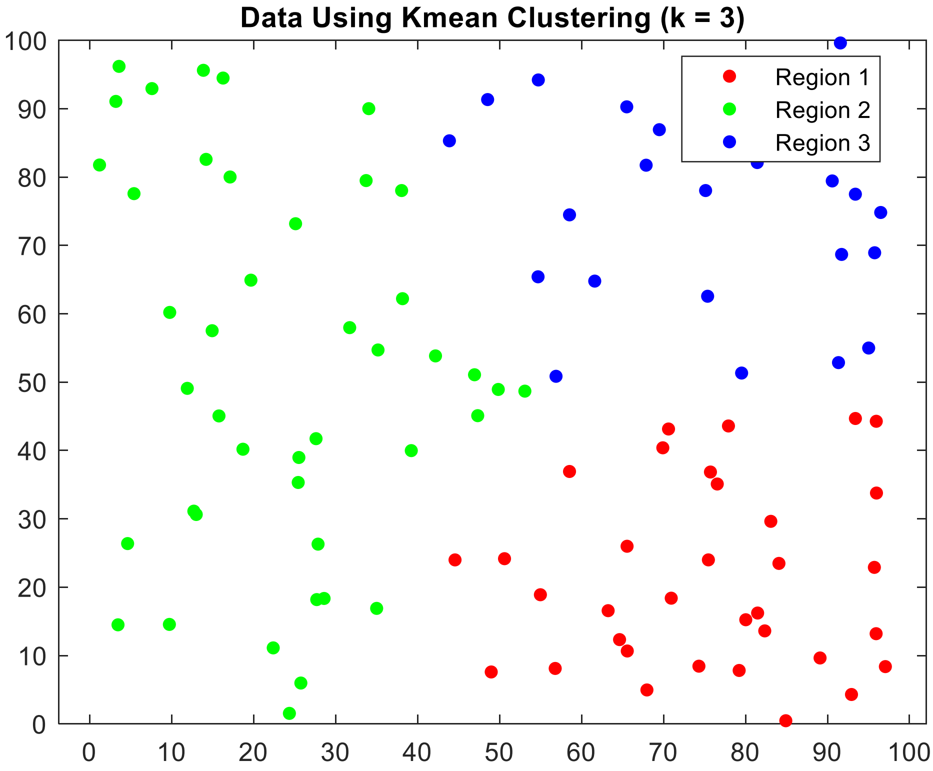
Result:

```
idx =

     1
     3
     2
     3
     1
     ...
```

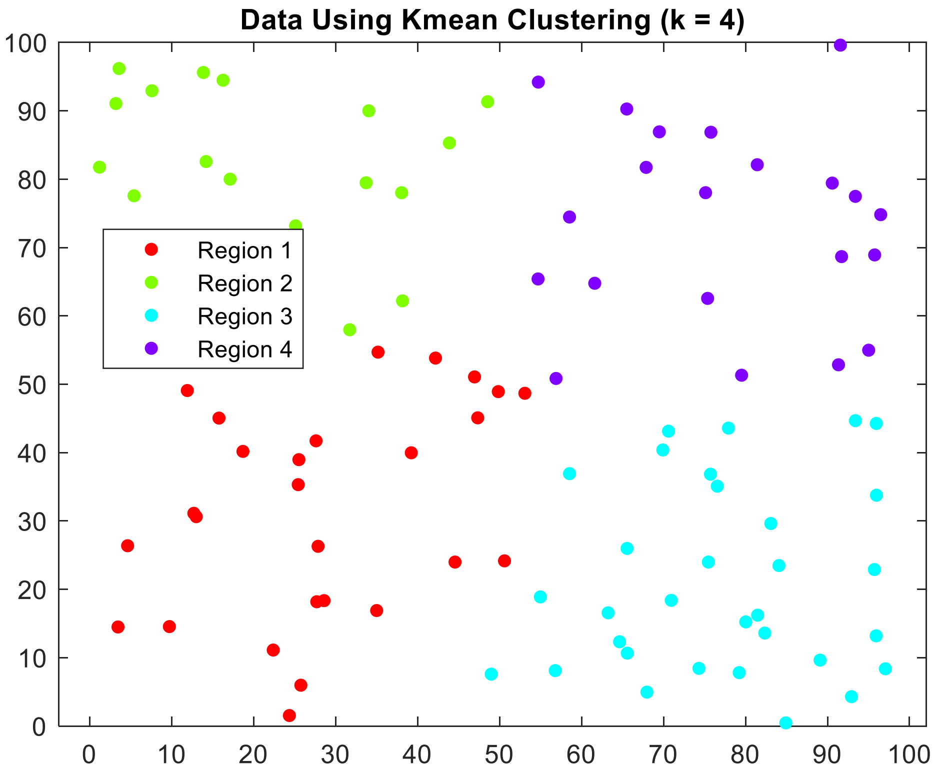
Part C

```
gscatter(x(:,1),x(:,2),idx);
legend("Region 1","Region 2","Region 3");
title("Data Using Kmean Clustering (k = 3)");
```

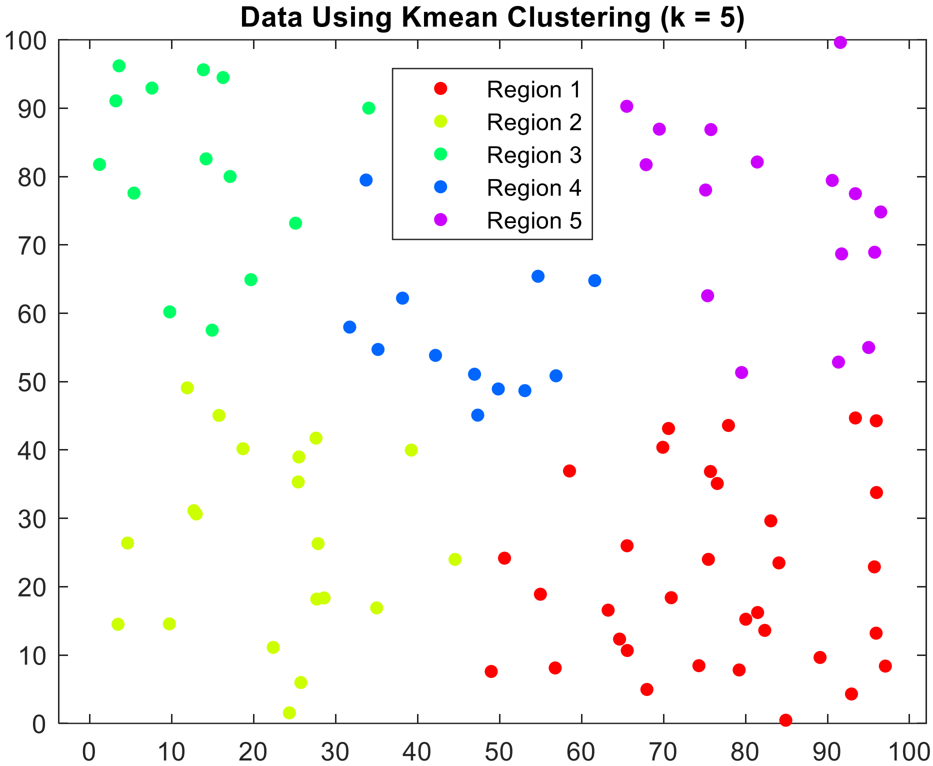

Data Using Kmean Clustering (k = 3)
Region 1
Region 2
Region 3
0
10
20
30
40
50
60
70
80
90
100

Part D

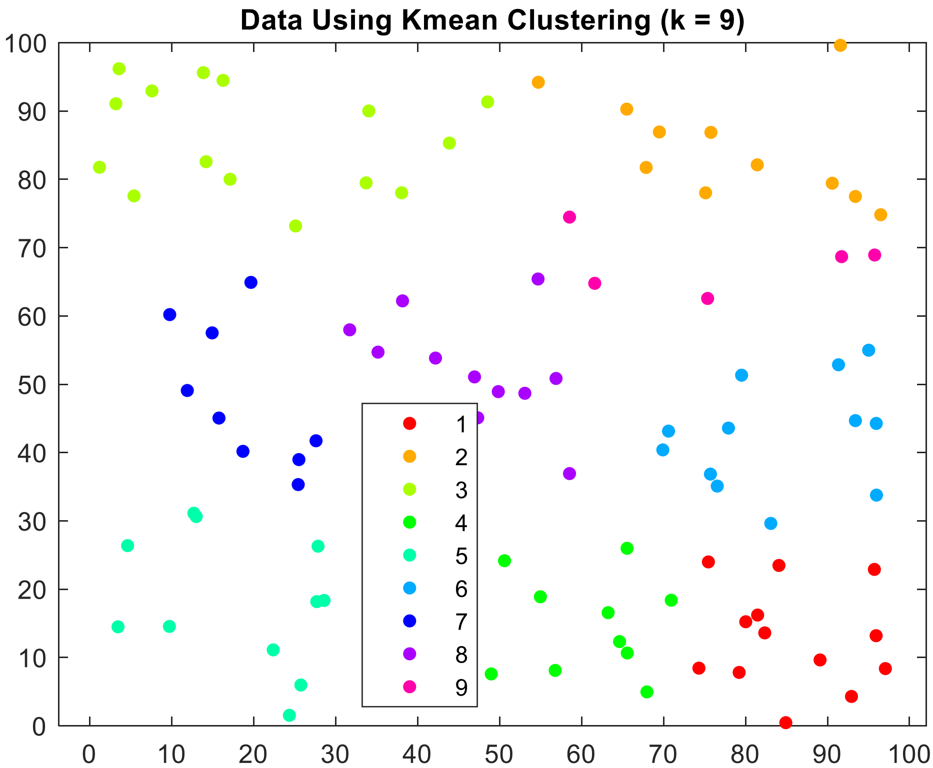
k = 4


Data Using Kmean Clustering (k = 4)
Region 1
Region 2
Region 3
Region 4
0
10
20
30
40
50
60
70
80
90
100

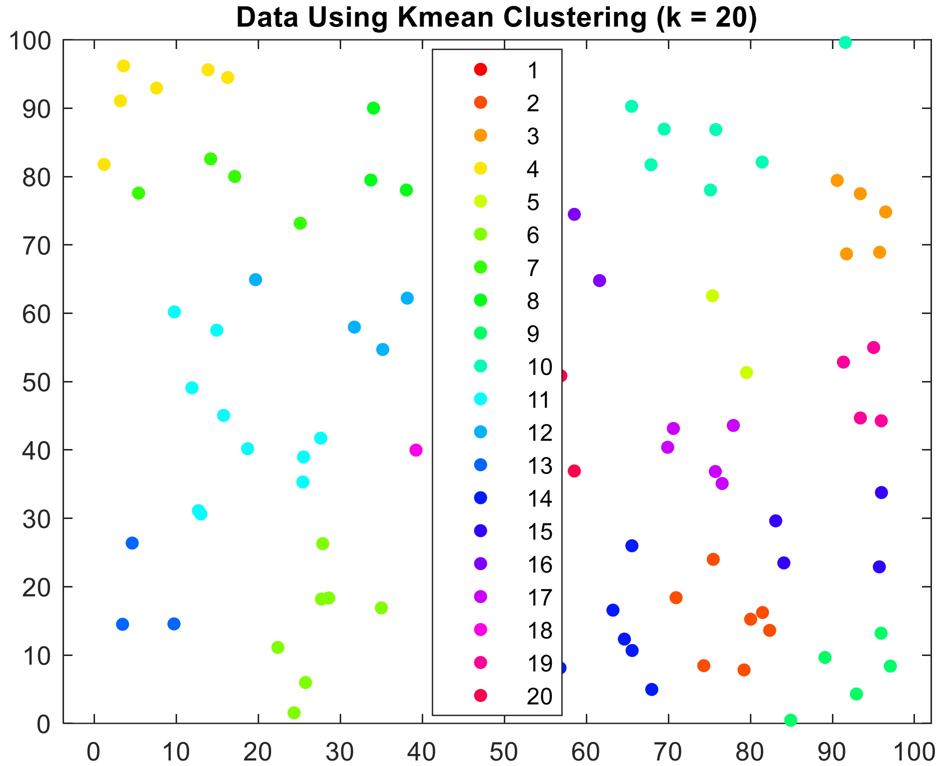
k = 5


Data Using Kmean Clustering (k = 5)
Region 1
Region 2
Region 3
Region 4
Region 5
0
10
20
30
40
50
60
70
80
90
100

k = 9


Data Using Kmean Clustering (k = 9)
1
2
3
4
5
6
7
8
9
0
10
20
30
40
50
60
70
80
90
100

k = 20



Part E

As the number of clusters increases, the individual clusters become smaller and take up less values.

Question 2

Part A

```
k = 3;
clst = fitgmdist(x,k)
```

Result:

```
clst =
Gaussian mixture distribution with 3 components in 2 dimensions

Component 1:
Mixing proportion: 0.207857
Mean:    73.5398    13.3242

Component 2:
Mixing proportion: 0.502565
```

```
Mean:    65.1992    59.0025

Component 3:
Mixing proportion: 0.289578
Mean:    16.3923    49.4626
```

Part B

```
fit = cluster(clst,x)
```

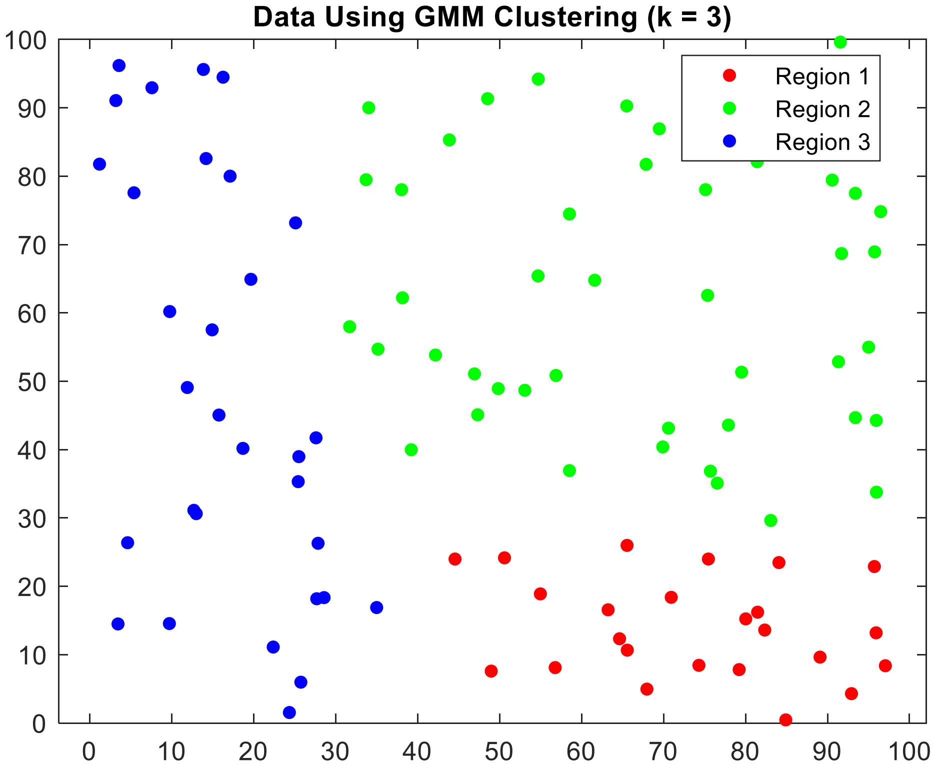
Result:

```
fit =

     1
     2
     3
     2
     1
     ...
```

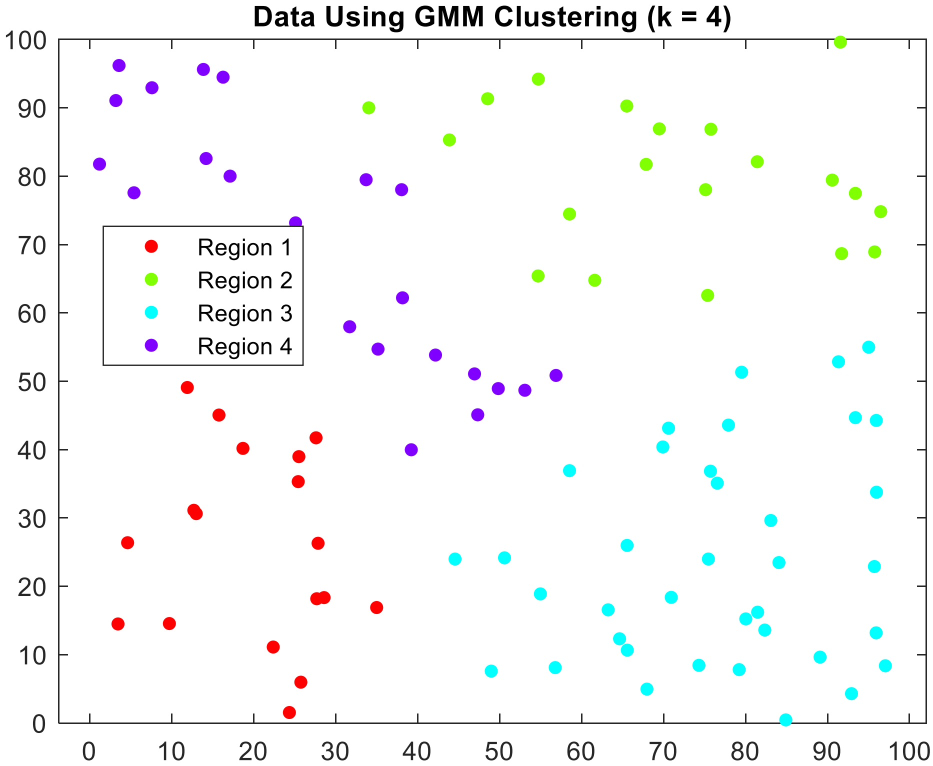
Part C

```
gscatter(x(:,1),x(:,2),fit);
legend("Region 1","Region 2","Region 3");
title("Data Using GMM Clustering (k = 3)");
```

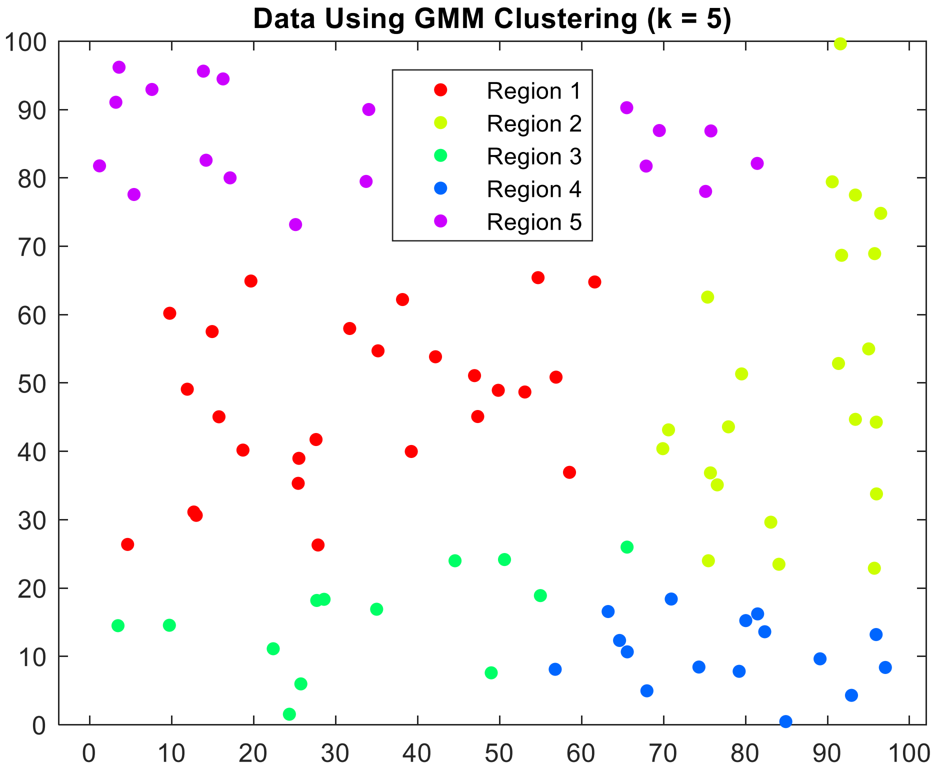


Part D

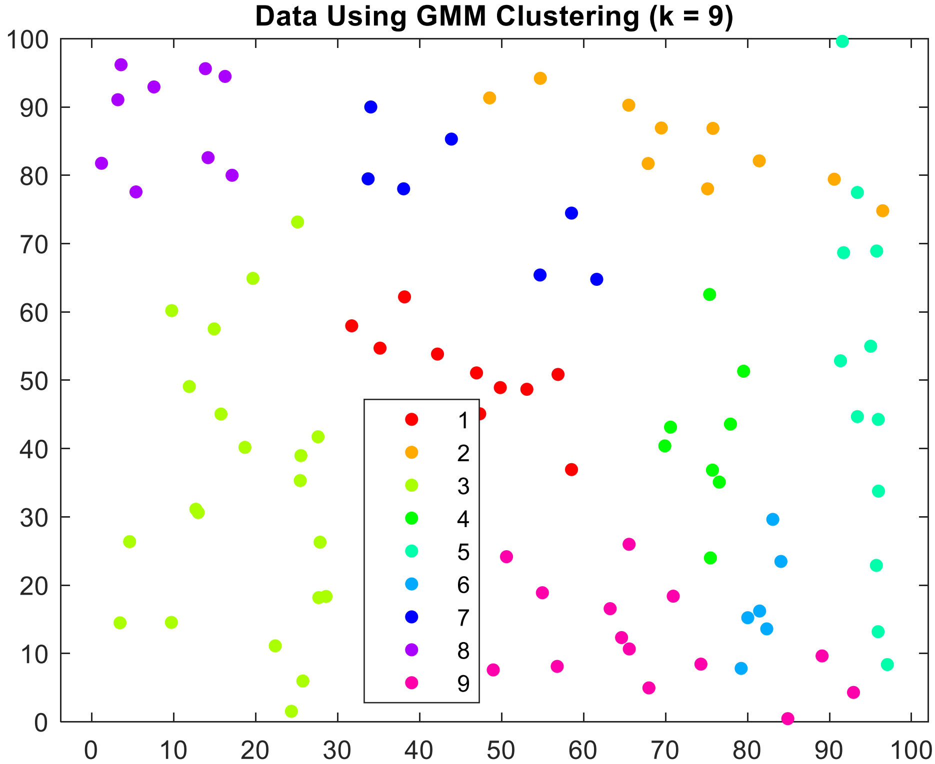
k = 4


Data Using GMM Clustering (k = 4)
Region 1
Region 2
Region 3
Region 4
0
10
20
30
40
50
60
70
80
90
100

k = 5


Data Using GMM Clustering (k = 5)
Region 1
Region 2
Region 3
Region 4
Region 5
0
10
20
30
40
50
60
70
80
90
100

k = 9



k = 20 gives us an error.

Part E

As the number of clusters increases, the clusters become smaller and take up less values. Likelihood of data overlap increases as well, leading to errors, and when the number gets too high, the error becomes unavoidable.

Part F

GMM clusters are more accurate, as they do not assume shapes for clusters; k-mean clusters assume a circular shape but are less complex. K-mean clusters also do not have to deal with errors at higher numbers, whereas GMM clusters do.

Question 3

Mean-shift Clustering: This is a sliding-window-based algorithm that attempts to find dense areas of data points. It is a centroid-based algorithm, meaning that the goal is to locate the centre points of each group, which works by updating candidates for centre points to be the mean of the points within the sliding-window. These candidate windows are then filtered in a post-processing stage to eliminate near-duplicates, forming the final set of centre points and their corresponding groups. Unlike K-mean clustering, there is no need to select the number of clusters, as mean-shift is automatically able to discover this, which is a very significant advantage. Also, the cluster centres converge towards the points of maximum density, which is quite useful, as it is more intuitive to understand and perfect for data analysis. However, its drawback is that the selection of the window size/radius can be non-trivial, leading to outliers being included in clusters.

DBSCAN: This is a density-based clustered algorithm similar to mean-shift, but possessing a couple of notable advantages. DBSCAN begins with an arbitrary starting data point that has not been visited; the neighbourhood points are determined and extracted using the distance epsilon (all points which are within this distance are neighbourhood points). If there are a sufficient number of neighbourhood points, judged by the minPoints variable, the clustering process starts and the current data point becomes the first point in the new cluster; otherwise, the point is labelled as noise, but might later become part of the cluster. The point is then marked as visited. This process is repeated for all the neighbourhood points and they are added to the cluster. Once the current cluster is completed, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise. Eventually all the points are marked as visited. DBSCAN has a lot of great advantages. One advantage is that it does not require a pre-set number of clusters, much like mean-shift clustering. However, unlike mean-shift clustering, it is able to identify outliers as noise and does not throw them into a cluster. Additionally, it is able to find arbitrarily sized and shaped clusters quite well. Its main drawback is that it doesn’t perform as well as other clustering methods when dealing with clusters of varying density, due to the use of the distance epsilon and minPoints, making it difficult to work with when dealing with high-dimensional data.

Conclusion

Thanks to this lab, I became familiar with data clustering and its different techniques, as well as their advantages and disadvantages. I also learned to use these techniques in the context of MATLAB. Ultimately, very interesting, with lots of applications.

12/5/2020