ENSC 474

Assignment #2

Spring 2019

301306459

Cyrus WaChong

cwachong@sfu.ca

1) Question 1:

For a function to be linear it must be homogeneous and additive. For a function to be homogeneous, it must meet the property:

$$f(ax) = af(x), \qquad a \in R$$

For a function to be additive, it must follow the following property:

$$f(a+b) = f(a) + f(b)$$

Writing two equations, one with a, and other with b:

$$f(a) \to 2{y_1}' + 3y_1 = 4a$$

$$f(b) \to 2{y_2}' + 3y_2 = 4b$$

$$a + b \to f \to 2y' + 3y = 4(a+b)$$

$$f(a) + f(b) = 4a + 4b = 4(a+b) = f(a+b)$$

Adding the first and second equation:

$$2(y_1' + y_2') + 3(y_1 + y_2) = 4(a+b)$$

If we plug in a value, such as y_3 for y_1 and y_2, then we can see that the equation is additive.

For homogeneous:

$$f(ax) \to 2ay' + 3ay = 4ax$$

$$af(x) \to a(2y' + 3y) = 4ax$$

By inspecting the two equations, one can conclude that they are equal. Therefore, the system is linear, since it is both homogeneous and additive.

Using MATLAB, I solved the equation, it can be found below

```
#######################################################################
Question 1
solution to differential equation is y(t)=(4*exp(t*w*1i))/(w*2i + 3) + C1*exp(-(3*t)/2)
#######################################################################
```

Figure 1: MATLAB solution for differential equation.

2) Question 2:

Once making c1 and c2 column vectors, and combining the two vectors together into a matrix (MNx2) A, we use the equation "Ax=b" to solve.

With that equation, we can solve for x by calculating "x=A\b", or in the case of this code, the function mldivide was used.

This returned a vector of 2 values, one of value α and the other of value β, which in this case were 0.6 and 0 respectively.

```
y = mldivide(A,b);
```

Figure 2: function used to find two constants

This meant that the closest value using the c1 vector and α (since β is 0) is the vector [0.6, 0.6, 0.6, 0.6, 0.6]. This is the reconstructed vector.

The distance was calculated using the norm function, and inputting the difference of the result of the b vector.

```
########################################################################
Question 2
value of alpha is 0.600000
value of beta is 0.000000
result = [0.6 0.6 0.6 0.6 0.6]
distance between actual b value and reconstructed is 1.095445
########################################################################
```

Figure 3: Output for question 2

The value of ~1.1 is almost double the value of any of the 5 dimensions. This seems quite far, since it is much larger that any of the other values, but not excessively far.

3) Question 3:

I now added a vector c3 = [4, 4, 7, 4, 15], and simply added this column to my A matrix. Using the same mldivide statement, we got the 3 coefficients, including the new gamma.

```
result = y(1)*c1 + y(2)*c2 + y(3)*c3; %reconstruct vector
```

Figure 4: Equation to find the result with the found coefficients.

```
######################################################################
Question 3
value of alpha is 0.573585
value of beta is -0.290566
value of gamma is 0.132075
result = [0.81 0.52 0.63 -0.06 1.1]
distance between actual b value and reconstructed is 0.678511
######################################################################
```

Figure 5: Output for Question 3

The distance when using three basis vectors gave a much smaller value than question 2. The distance found in question 3 was 0.679, which is almost half of the previous question.

This may be since with three vectors, you are no longer limited to representing every point with 2 dimensions (assuming they are all linearly independent), so in all cases, you would have a more "accurate" reconstructed value.

This means that the distance between the reconstructed and original point should be less that when representing with a lower number of vectors.

4) Question 4:

Using the same mugshot as last assignment, just cropped to show the face, and an image of a dog found on google, which was then cropped to the same size; I proceeded onto the next step.

To unwrap the images into (MNx1) column vectors, I used the following lines of code:

```
img1 = img1(:);
img2 = img2(:);
img3 = img3(:);
```

Figure 6: lines of code to unwrap image into column vectors

When my face was projected onto the non human face (in this case, the face of a dog), the resulting image looked almost unchanged from the original dog grayscale image:

Figure 7: the resulting image with my face projected onto a dogs face

Orthogonal in this case means the dot product of the matrices is zero, or each value is 90 degrees separated. These two images were <u>not orthogonal,</u> and for two images to be orthogonal seems to be quite unlikely, because each value on the original image must be orthogonal to the value on the orthogonal image.

For 4.4, using the same mldivide function, we can find the two values of alpha and beta, because we have now converted the images to column vectors, allowing us to treat it the same as before.

The values for part 4.4 of alpha and beta were found to be 0.596 and 0.246 respectively. This may be seen below in the diagram.

These values are not ideal, since the first two images are not taken at the same time. If they were much more similar, we would have seen much more of the human image used, and not as much of the dog image. This means that the result is not the same, but could be closer.

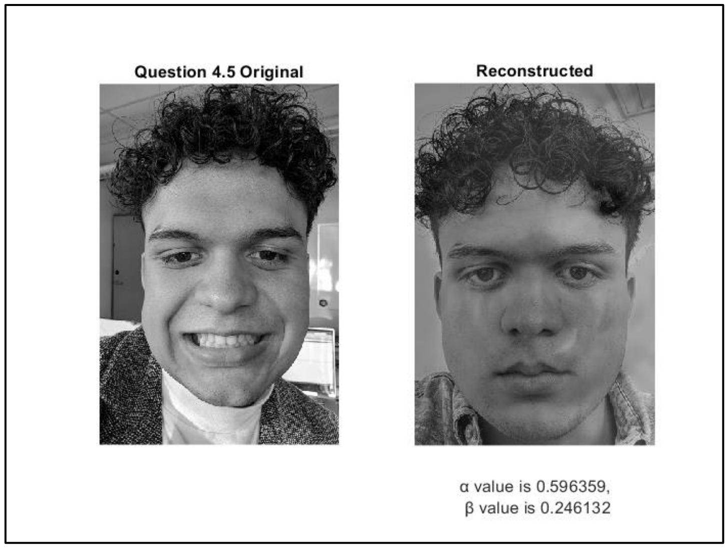

Figure 8: Using two basis images to represent a third image.

This is like trying to build the vector [1, 1, 1] with vectors [1, 0, 0] and [0, 1, 0]. You can get close, but you will never reach the exact value.

Figure 9: Three outputs for part 4.6, using three different images for I

As can be seen above, three new images were used for I, and the program tried to rebuild them using a combination of the original mugshot and the picture of the dog.

The best values for both alpha and beta in each image were:

Image	Alpha value	Beta Value
4.6_1 (Headphones)	0.396313	0.425951
4.6_2 (Water bottle)	0.641926	0.431138
4.6_3 (Smiling Selfie)	0.716869	0.307816

As can be seen, the image with the highest alpha value is the selfie, which is expected, since one of the basis images in another picture of myself. The other two returned images that looked nothing like the original image, but that was expected when only 2 basis images are used.

The image with more black (the headphones) used the image of the dog more heavily, which makes sense because the dog image was a darker image.

All in all, the images were expected to be quite inaccurate, since we are tying to represent an image with two other random images, you cannot expect an accurate recreation.