Muhideen Ogunlowo and Brett Tulibacki Group Homework 1

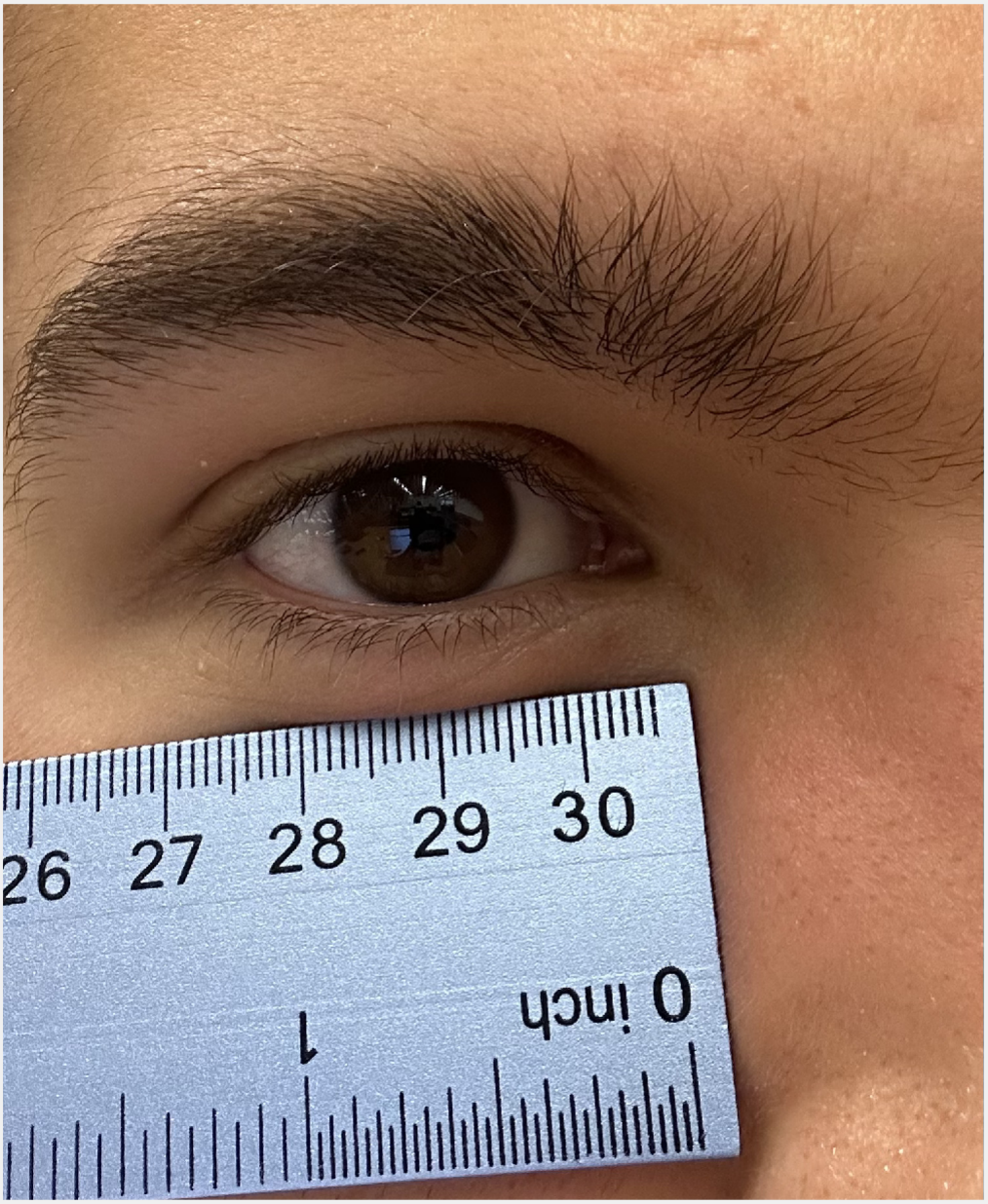
```
% Script: EyePicture.m
% Clear workspace and close all figures
clear all;
close all;

%Importing image from file
A = imread('file:///Users/aogunlowo19/Downloads/EyePic.jpg');
eyepic = A;

% Display the image and prompt for point selection
imshow(eyepic)
disp('Click on 12 input points on the upper eyelid curve')
```

```
Click on 12 input points on the upper eyelid curve
```

```
%data set for upper eyelid curve
[x0_U, y0_U] = ginput(1);
[x1_U, y1_U] = ginput(2); %collection of data points for conversion to cm
lcm_U = sqrt((x1_U(2) - x1_U(1))^2 + ((y1_U(2) - y1_U(1))^2)); %conversion
to cm
[x2_U, y2_U] = ginput(12); %input 12 points on the upper eyelid
xU = (x2_U - x0_U) / lcm_U; %conversion of data sets to cm
yU = -(y2_U - y0_U) / lcm_U;%conversion of data sets to cm

disp('Click on 12 input points on the lower eyelid curve')
```

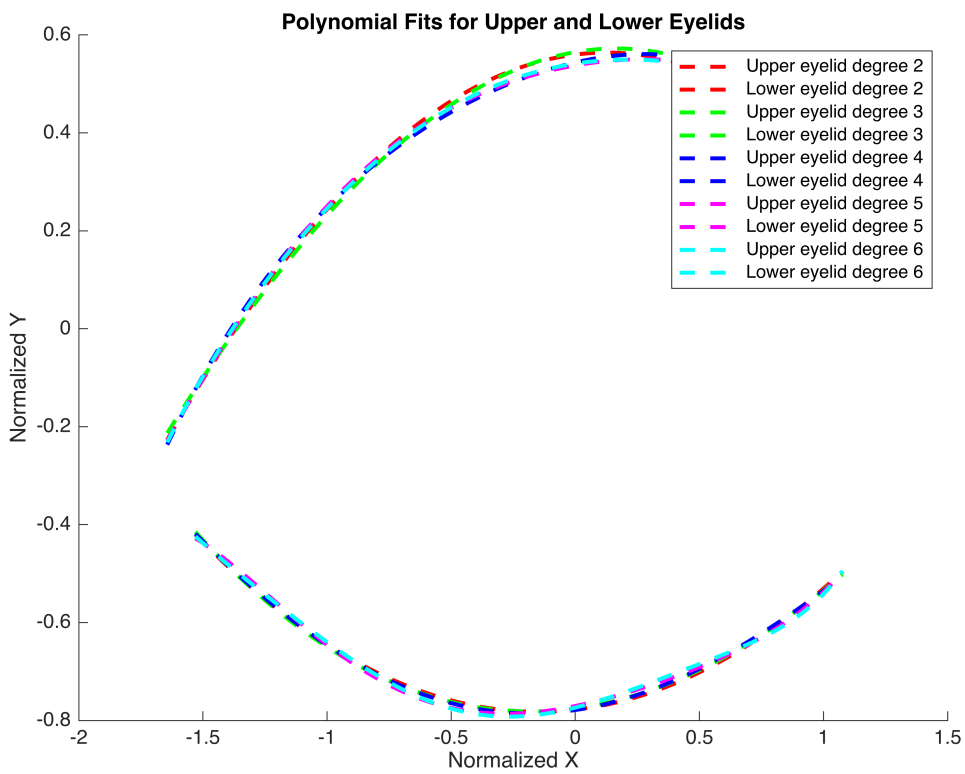
```
Click on 12 input points on the lower eyelid curve
```

```
%data set for lower eyelid curve
[x0_L, y0_L] = ginput(1); %data set for lower eyelid curve
[x1_L, y1_L] = ginput(2); %collection of data points for conversion to cm
lcm_L = sqrt((x1_L(2) - x1_L(1))^2 + ((y1_L(2) - y1_L(1))^2));%conversion
to cm
[x2_L, y2_L] = ginput(12);%input 12 points on the upper eyelid
```



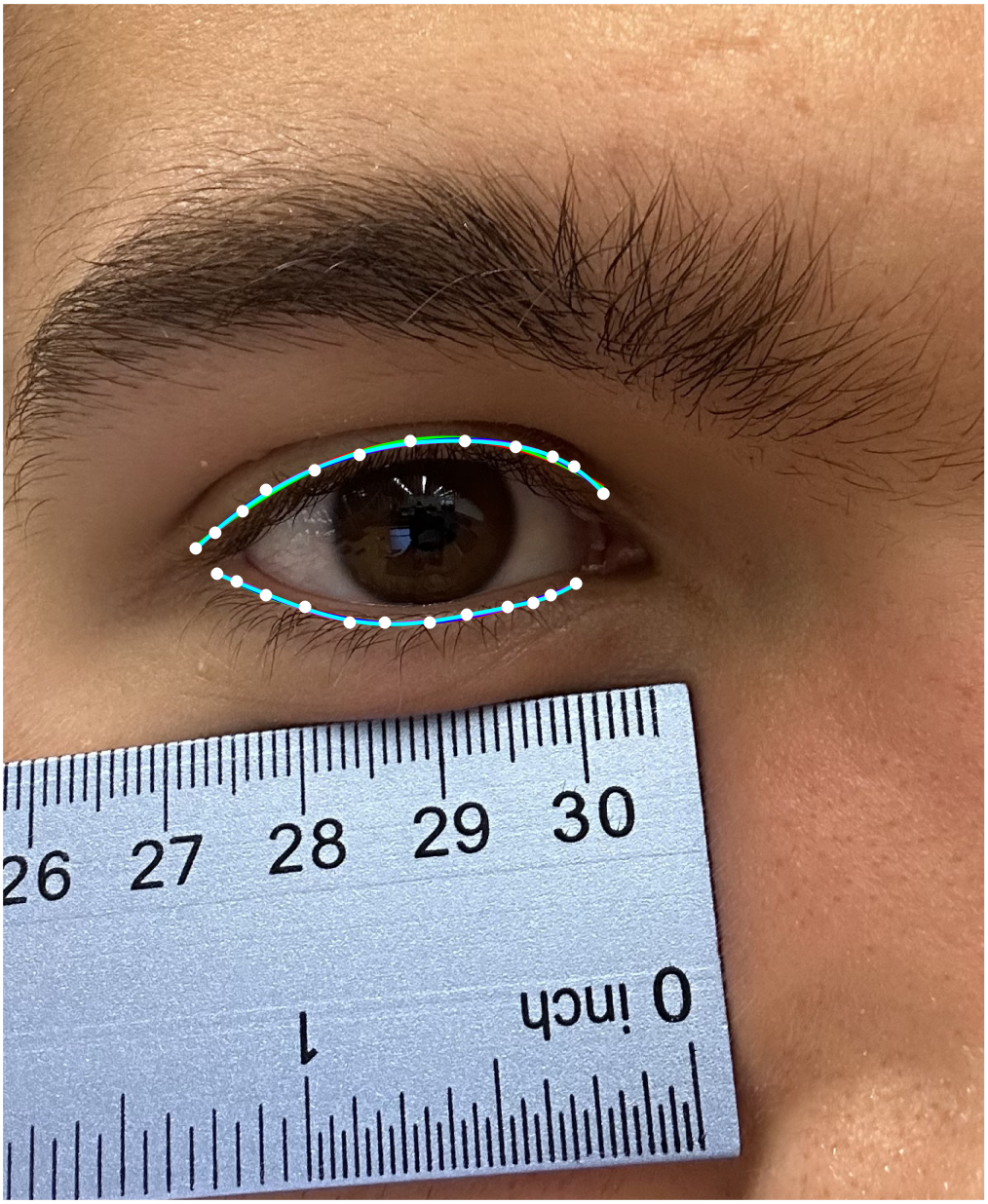

```
xL = (x2_L - x0_L) / lcm_L;%conversion of data sets to cm
yL = -(y2_L - y0_L) / lcm_L;%conversion of data sets to cm

% Defininition of colors for the curves for visual distinction
colors = ['r', 'g', 'b', 'm', 'c'];

% Initialization of the figure for polynomial graph
figure('Name', 'Polynomial Fits');
hold on;

% Loop over degrees for polynomial fitting
for d = 2:6
    % Fit polynomial for upper eyelid
```

```
    c_U = polyfit(xU, yU, d);
    xvals_U = linspace(min(xU), max(xU), 100);
    yvals_U = polyval(c_U, xvals_U);
    % Plot polynomial for upper eyelid
    plot(xvals_U, yvals_U, 'Color', colors(d-1), 'Linestyle',
'--','Linewidth', 2, 'DisplayName', ['Upper eyelid degree ' num2str(d)]);

    % Fit polynomial for lower eyelid
    c_L = polyfit(xL, yL, d);
    xvals_L = linspace(min(xL), max(xL), 100);
    yvals_L = polyval(c_L, xvals_L);
    % Plot polynomial for lower eyelid
    plot(xvals_L, yvals_L, 'Color', colors(d-1), 'Linestyle',
'--','Linewidth', 2, 'DisplayName', ['Lower eyelid degree ' num2str(d)]);
end

% Add legend and labels to the graph
legend('show');
xlabel('Normalized X');
ylabel('Normalized Y');
title('Polynomial Fits for Upper and Lower Eyelids');
hold off;
```




```
% Overlay of the curves on the image in a new figure
figure('Name', 'Eyelid Curves on Image');
imshow(eyepic);
```

```
hold on;

% Overlay of the polynomial fits on the image
for d = 2:6
    % Fit and plot for upper eyelid
    c_U = polyfit(xU, yU, d);
    xvals_U = linspace(min(xU), max(xU), 100);
    yvals_U = polyval(c_U, xvals_U);
    xpic_U = xvals_U .* lcm_U + x0_U;
    ypic_U = -yvals_U .* lcm_U + y0_U;
    plot(xpic_U, ypic_U, 'Color', colors(d-1), 'LineWidth', 2);

    % Fit and plot for lower eyelid
    c_L = polyfit(xL, yL, d);
    xvals_L = linspace(min(xL), max(xL), 100);
    yvals_L = polyval(c_L, xvals_L);
    xpic_L = xvals_L .* lcm_L + x0_L;
    ypic_L = -yvals_L .* lcm_L + y0_L;
    plot(xpic_L, ypic_L, 'Color', colors(d-1), 'LineWidth', 2);
end

% Plot of the original points for upper and lower eyelids
plot(x2_U, y2_U, 'wo', 'MarkerFaceColor', 'w');
plot(x2_L, y2_L, 'wo', 'MarkerFaceColor', 'w');

% Final image display
hold off;
```




```
% Table of L2 norms
L2_norms_U = zeros(1, 5);
L2_norms_L = zeros(1, 5);
for d = 2:6
    % Polynomial fit for upper eyelid
    c_U = polyfit(xU, yU, d);
    residuals_U = yU - polyval(c_U, xU);
    L2_norms_U(d-1) = norm(residuals_U, 2);

    % Polynomial fit for lower eyelid
    c_L = polyfit(xL, yL, d);
    residuals_L = yL - polyval(c_L, xL);
```

```
    L2_norms_L(d-1) = norm(residuals_L, 2);
end

degrees = (2:6)';
L2_table = table(degrees, L2_norms_U', L2_norms_L', 'VariableNames',
{'Degree', 'Upper_Eyelid_L2', 'Lower_Eyelid_L2'});
disp(L2_table);
```

Degree	Upper_Eyelid_L2	Lower_Eyelid_L2
2	0.0863547567882257	0.0466809644732314
3	0.0800083944280782	0.0448155439152001
4	0.0517750735485335	0.0434840298393381
5	0.0457636588591309	0.0327656836197586
6	0.0443447220234628	0.0289202575024328

The best fit is degree 6 because the l2 norm of the residuals was the closest to the line of fit.

Question 7

```
d = 6; % Degree of the polynomial

% Upper eyelid pixel values
V_pixel_U = vander(x2_U); % Vandermonde matrix with pixel values
V_pixel_U = V_pixel_U(:, end-d:end); % Select the last d+1 columns for
degree d
cond_pixel_U = cond(V_pixel_U); % Condition number for pixel values

% Upper eyelid scaled values
V_scaled_U = vander(xU); % Vandermonde matrix with scaled values
V_scaled_U = V_scaled_U(:, end-d:end); % Select the last d+1 columns for
degree d
cond_scaled_U = cond(V_scaled_U); % Condition number for scaled values

% Display of the condition numbers
disp(['Condition number using pixel values (upper eyelid): ',
num2str(cond_pixel_U)]);
```

```
Condition number using pixel values (upper eyelid): 4.030125241747444e+20
```

```
disp(['Condition number using scaled values (upper eyelid): ',
num2str(cond_scaled_U)]);
```

```
Condition number using scaled values (upper eyelid): 179.2594
```

```
%We used OpenAI, to get extract the vandermode matrix from the loop
```

The condition number of the upper eyelid using pixel values is badly conditioned $(4.03 * 10^{20})$ compared to the Condition number using scaled values (179.2594)