PS 5

2. We can follow the approach taken in Lecture 10 when constructing Legendre's Polynomials

$$H_0(x) = 1$$

$$H_1(x) = x - \frac{\overbrace{\langle x, 1\rangle}^{0}}{\langle 1, 1\rangle} 1 = x$$

$*$ Note: odd $f(x)\cdot g(x)$ func's will always have $\langle f, g\rangle = 0$

$$H_2(x) = x^2 - \frac{\langle x^2, 1\rangle}{\langle 1, 1\rangle} 1 - \frac{\overbrace{\langle x^2, x\rangle}^{0}}{\langle x, x\rangle} x = x^2 - \frac{\sqrt{2\pi}}{\sqrt{2\pi}} 1 = x^2 - 1$$

$$H_3(x) = x^3 - \frac{\overbrace{\langle x^3, 1\rangle}^{0}}{\langle 1, 1\rangle} 1 - \frac{\langle x^3, x\rangle}{\langle x, x\rangle} x - \frac{\overbrace{\langle x^3, x^2-1\rangle}^{0}}{\langle x^2, x^2-1\rangle}(x^2-1)$$

$$= x^3 - \frac{3\sqrt{2\pi}}{\sqrt{2\pi}} x = x^3 - 3x$$

$$H_4(x) = x^4 - \frac{\langle x^4, 1\rangle}{\langle 1, 1\rangle} 1 - \frac{\overbrace{\langle x^4, x\rangle}^{0}}{\langle x, x\rangle} x - \frac{\langle x^4, x^2-1\rangle}{\langle x^2-1, x^2-1\rangle}(x^2-1) - \frac{\overbrace{\langle x^4, x^3-x\rangle}^{0}}{\langle x^3-x, x^3-x\rangle}(x^3-x)$$

$$= x^4 - \frac{3\sqrt{2\pi}}{\sqrt{2\pi}} - \frac{12\sqrt{2\pi}}{2\sqrt{2\pi}}(x^2-1)$$

$$= x^4 - 3 - 6x^2 + 6 = x^4 - 6x^2 + 3$$

So, our first five monic Hermite polynomials are
$1,\ x,\ x^2-1,\ x^3-3x,\ x^4-6x^2+3$

3. For $W_1^{\perp}$, $\langle x, y \rangle_1 = x_1 y_1 + x_2 y_2 + x_3 y_3$

Then, we want to find u s.t. $\langle v_1, u \rangle_1 = \langle v_2, u \rangle_1 = 0$

So, we have: $u_1 + 2u_2 + 3u_3 = 0$
$2u_1 + 0u_2 + u_3 = 0$

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 0 \\ 2 & 0 & 1 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 3 & 0 \\ 0 & -4 & -5 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & 1/2 & 0 \\ 0 & 1 & 5/4 & 0 \end{array}\right]$$

$x_1 = -\frac{1}{2} x_3 \quad x_2 = -5/4\, x_3$

So, $W_1^{\perp} = \text{span}\left\{\begin{pmatrix} -1/2 \\ -5/4 \\ 1 \end{pmatrix}\right\}$

For $W_2^{\perp}$, $\langle x, y \rangle_2 = x_1 y_1 + 2x_2 y_2 + 3x_3 y_3$

$\langle v_1, u \rangle_2 = \langle v_2, u \rangle_2 = 0$

$$\left[\begin{array}{ccc|c} 1 & 4 & 9 & 0 \\ 2 & 0 & 3 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 4 & 9 & 0 \\ 0 & -8 & -15 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & 3/2 & 0 \\ 0 & 1 & 15/8 & 0 \end{array}\right]$$

$x_1 = -3/2\, x_3 \quad x_2 = -15/8\, x_3$

So, $W_2^{\perp} = \text{span}\left\{\begin{pmatrix} -3/2 \\ -15/8 \\ 1 \end{pmatrix}\right\}$

4. To check if A is complete, we check if the algebraic & geometric multiplicities match for every $\lambda \in A$.

$$A = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \det(A - \lambda I) = \det \begin{bmatrix} -\lambda & 0 & -1 \\ 0 & 1-\lambda & 0 \\ 1 & 0 & -\lambda \end{bmatrix}$$

$$= (-\lambda)(1-\lambda)(-\lambda) + (-1)(\lambda - 1)$$
$$= (1-\lambda)(\lambda^2 + 1)$$

all algebraic mult. = $\underline{1}$

$$(1-\lambda)(\lambda^2+1) = 0 \Rightarrow \lambda = 1, \lambda = i, \lambda = -i$$

$\underline{\lambda = 1}$

$$\left[\begin{array}{ccc|c} -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \Rightarrow \text{basis null: } \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$\underline{\lambda = i}$

$$\left[\begin{array}{ccc|c} -i & 0 & -1 & 0 \\ 0 & 1-i & 0 & 0 \\ 1 & 0 & -i & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & -i & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \Rightarrow \text{basis null: } \begin{pmatrix} i \\ 0 \\ 1 \end{pmatrix}$$

$\underline{\lambda = -i}$

$$\left[\begin{array}{ccc|c} i & 0 & -1 & 0 \\ 0 & 1+i & 0 & 0 \\ 1 & 0 & i & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 0 & i & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \Rightarrow \text{basis null: } \begin{pmatrix} -i \\ 0 \\ 1 \end{pmatrix}$$

Since each eigenvalue is complete ($\dim V_\lambda = m_\lambda$), the matrix is complete. Because all the eigenvectors are lin. independent, we also have an eigenbasis of $\mathbb{R}^3, \mathbb{C}^3$.

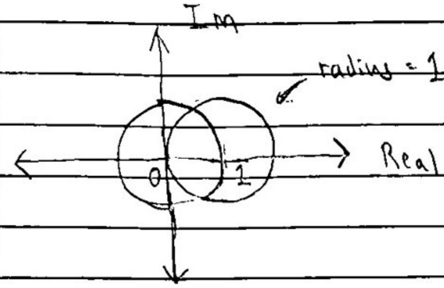
5. a) $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & -1 & 1 \end{bmatrix}$

$D_1 = \{ z \in \mathbb{C} : |z| \leq 1 \}$

$D_2 = \{ z \in \mathbb{C} : |z-1| \leq 1 \}$

$D_3 = \{ z \in \mathbb{C} : |z-1| \leq 1 \}$

So, we have two Gerschgorin disks with radius 1, one centered at 0 and one centered at 1.



b) We know from Gerschgorin Thm that $\text{spec}(A) \subset D_A$ & $\text{spec}(A^T) \subset D_{A^T}$. Thus to show for any matrix A, $\text{spec}(A) \subset D_{A^*} = D_A \cap D_{A^T}$, we will show that $\text{spec}(A) = \text{spec}(A^T)$. This is done by showing that they have the same roots to their characteristic eq.

$$\det(A - \lambda I) : \begin{cases} \det(A^T - \lambda I) \\ = \det((A - \lambda I)^T) \end{cases}$$ because $I^T = I$

We also know from determinant props. that the determinant of a matrix does not change when transposed. So,

$$\det(A - \lambda I) = \det((A - \lambda I)^T) = \det(A^T - \lambda I).$$

Therefore, the roots to the characteristic eq'n are the same & $\text{spec}(A) = \text{spec}(A^T)$. This necessarily means $\text{spec}(A) \subset D_A \cap D_{A^T} = D_{A^*}$.

$\square$

c) $A^T = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix}$

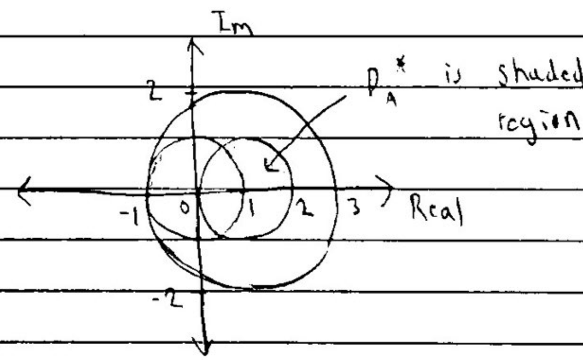
$D_1 = \{ z \in \mathbb{C} : |z| \leq 0 \}$

$D_2 = \{ z \in \mathbb{C} : |z-1| \leq 2 \}$

$D_3 = \{ z \in \mathbb{C} : |z-1| \leq 1 \}$

The only unique disk in D_{A^T} is D_2 which is a disk with radius 2 centered at 1.

Refined domain:



d) $\det(A - \lambda I) = \det \begin{bmatrix} -\lambda & 1 & 0 \\ 0 & 1-\lambda & 1 \\ 0 & -1 & 1-\lambda \end{bmatrix}$

$\Rightarrow -\lambda\left((1-\lambda)^2 + 1\right) = 0$

$\lambda = 0$ or $(1-\lambda)^2 = -1$

$1 - \lambda = \pm i$

$\lambda = 1 \pm i$

$\boxed{\lambda = 0,\ \lambda = 1+i,\ \&\ \lambda = 1-i \text{ are all in } D_{A^*}.}$

ACM/IDS 104 - Problem Set 5 - MATLAB Problems

Before writing your MATLAB code, it is always good practice to get rid of any leftover variables and figures from previous scripts.

```
clc; clear; close all;
```

Problem 1 (10 points) Application of Projections to Approximation

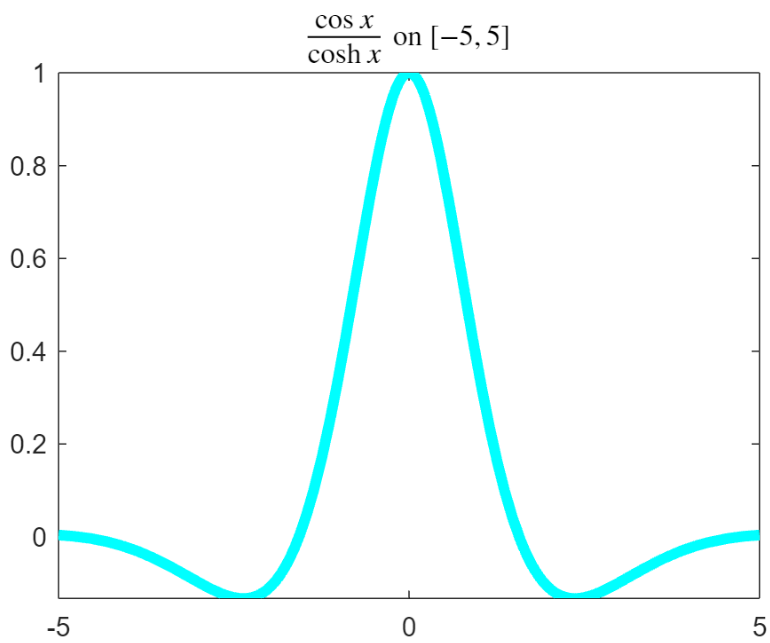
In Problem 4 of PS4, we saw that even higher degree interpolating polynomials may not be accurate approximations to complex functions. We have the function:

$$f(x) = \frac{\cos x}{\cosh x}, \quad \text{on } [-a, a], \quad a = 5$$

Let us recall how this function looks like and how its interpolating polynomials of degree $(n-1)$ for $n = 3, 5, 10, 15$ behave:

```
%{
Setup
%}
f = @(x) cos(x)./cosh(x); % our function
a = 5; % setting the value of a
n = [3 5 10 15]; % setting the number of points
sub = 1; % subplot index

%{
How f(x) looks like on [-5, 5]
%}
figure;
fplot(f, [-a, a], "-c", "lineWidth", 4);
title("$\frac{\cos{x}}{\cosh{x}}$ on $[-5, 5]$","Interpreter","latex");
```




```
%{
Read the discussion below and complete the code
%}
figure;
for ival = a
    for degree = n-1
        %{
        INTERPOLATING POLYNOMIALS -- no changes needed
        -> Select degree+1 points in the interval
        -> Evaluate f(x) on these points
        -> Find the polynomial coefficients
        %}
        pts = ones(degree+1, 2); % initializing the points
        pts(:, 1) = linspace(-ival, ival, degree+1); % setting the x-values
        for i = 1 : degree+1
            pts(i, 2) = f(pts(i, 1)); % evaluating cos(x) / cosh(x)
        end
        coeffs = polyfit(pts(:, 1), pts(:, 2), degree); % coefficients
        %{
        ORTHOGONAL PROJECTIONS -- TODO
        -> Get transformed Legendre polynomials
        -> Find alpha_k using L^2 inner product
        -> Evaluate alpha_k*Q_k
        %}
        % Getting 100 linearly spaced points from -5 to 5
        x_inputs = linspace(-ival, ival);
        y_inputs = zeros(100, 1);

        alpha_k = zeros(degree, 1);
```

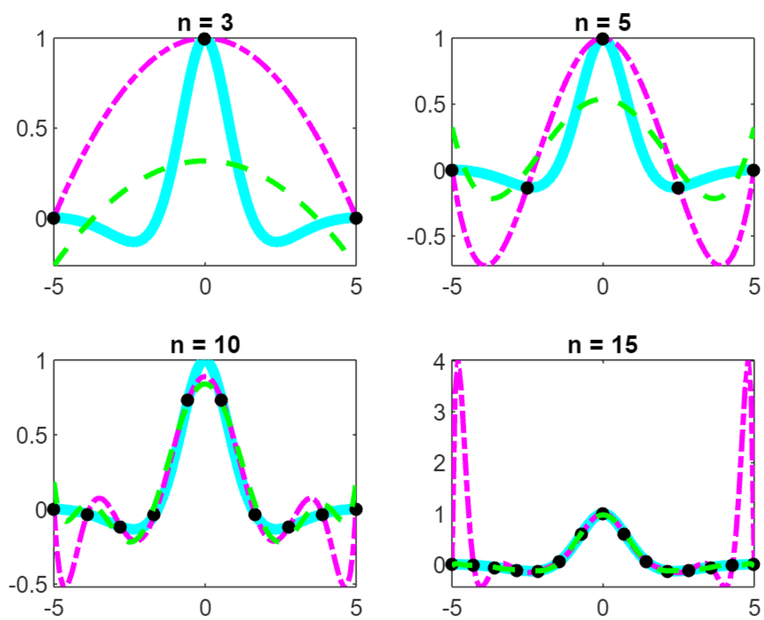
```
        for j = 0:degree
            % compute legendre on x/ival to normalize function bounds [-1, 1]
            g = @(x) (f(x) .* legendreP(j, x / ival));
            g_2 = @(x) (legendreP(j, x / ival) .* legendreP(j , x / ival));
            alpha_k(j + 1) = integral(g, -ival, ival) / integral(g_2, -ival, ival);
        end
        alpha_k

        for i = 1:100
            y_val = 0;
            for j = 0:degree
                y_val = y_val + legendreP(j, x_inputs(i) / ival) * alpha_k(j + 1);
            end
            y_inputs(i) = y_val;
        end
        % disp(y_inputs);

        %{
        PLOTTING
        Plot f(x), the sampled points, interpolating and approximating
        polynomials
        Please use different colors and linestyles
        %}
        subplot(2, 2, sub);
        fplot(f, [-ival, ival], "-c", "lineWidth", 4);
        hold on
        interpoints = linspace(-ival, ival);
        p = polyval(coeffs, interpoints); % evaluating coeffs in interval
        plot(interpoints, p, "-.m", "lineWidth", 2);
        plot(pts(:, 1), pts(:, 2), "ok", "MarkerSize", 2, "lineWidth", 3);
        plot(x_inputs, y_inputs, "--g", "MarkerSize", 2, "lineWidth", 2);
        title(strcat("n = ", int2str(degree+1)));
        sub = sub + 1; % increase subplot index
    end
end
```

```
alpha_k = 3×1
    0.1235
    0.0000
   -0.3888
alpha_k = 5×1
    0.1235
    0.0000
   -0.3888
    0.0000
    0.5881
alpha_k = 10×1
    0.1235
    0.0000
   -0.3888
    0.0000
    0.5881
   -0.0000
   -0.5815
```

```
   -0.0000
    0.4415
    0.0000
alpha_k = 15×1
    0.1235
    0.0000
   -0.3888
    0.0000
    0.5881
   -0.0000
   -0.5815
   -0.0000
    0.4415
    0.0000
      ⋮
```



Now, instead of interpolating polynomials, let us approximate $f(x)$ by its orthogonal projection onto the inner space $\mathscr{P}^{(n-1)}_{[-a,a]}$ of polynomials on $[-a, a]$, equipped with the L^2 inner product:

$$f(x) \approx p(x) = \mathrm{pr}_{\mathscr{P}^{(n-1)}_{[-a,a]}} f(x)$$

Recall (Lecture 10) that $p(x)$ is the closest (in the L^2 sense) polynomial to $f(x)$ in $\mathscr{P}^{(n-1)}_{[-a,a]}$, i.e.

$$p(x) = \arg \min_{q \in \mathscr{P}^{(n-1)}_{[-a,a]}} \| f(x) - q(x) \|$$

We know that the transformed Legendre polynomials $\widetilde{Q}_0(x), \cdots, \widetilde{Q}_{n-1}(x)$ form an orthogonal basis of $\mathscr{P}^{(n-1)}_{[-a,a]}$, and, therefore:

$$p(x) = \sum_{k=0}^{n-1} \alpha_k \widetilde{Q}_k(x)$$

where α_k are the coordinates of $p(x)$ in that basis.

Modify the above code to find the approximating polynomials as well. Plot each approximating polynomial on its corresponding subplot. Useful functions for this problem:

```
legendreP(), integral()
```