# DSP Lab 12
# Name: Basil khowaja bk08432

## Task 1)

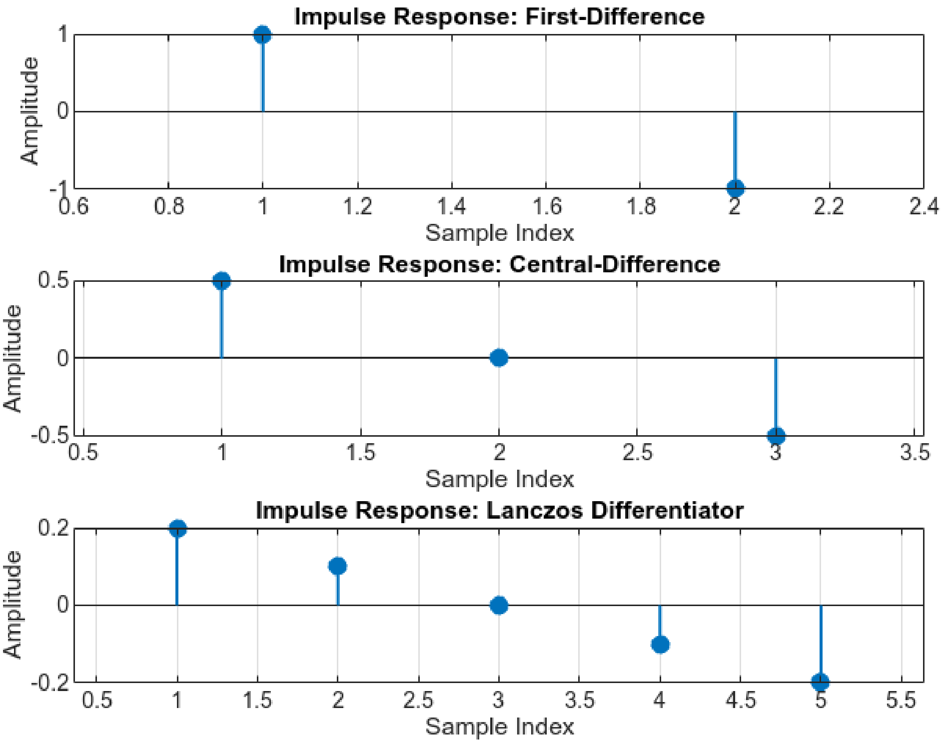
**code:**

```
%part A try 2:
clear; close all; clc;

b_fd = [1, -1];          %first-Difference coefficients
b_cd = [0.5, 0, -0.5];     %central-Difference coefficients
b_lanczos = [0.2, 0.1, 0, -0.1, -0.2];  %lanczos coefficients

%%impulse Responses
figure('Name','Impulse Responses','NumberTitle','off');

subplot(3,1,1);
stem(b_fd, 'filled', 'LineWidth', 1.2);
grid on;
title('Impulse Response: First-Difference');
xlabel('Sample Index'); ylabel('Amplitude');

subplot(3,1,2);
stem(b_cd, 'filled', 'LineWidth', 1.2);
grid on;
title('Impulse Response: Central-Difference');
xlabel('Sample Index'); ylabel('Amplitude');

subplot(3,1,3);
stem(b_lanczos, 'filled', 'LineWidth', 1.2);
grid on;
title('Impulse Response: Lanczos Differentiator');
xlabel('Sample Index'); ylabel('Amplitude');

%%frequency Response (Magnitude)
N = 1024;  %fft length

B_fd = fftshift(fft(b_fd, N));
B_cd = fftshift(fft(b_cd, N));
B_lanczos = fftshift(fft(b_lanczos, N));

w = linspace(-pi, pi, N);

Mag_fd = abs(B_fd);
Mag_cd = abs(B_cd);
Mag_lanczos = abs(B_lanczos);

Mag_ideal = abs(w);

figure('Name','Frequency Responses','NumberTitle','off');
plot(w, Mag_ideal, 'r', 'LineWidth', 2); hold on;
plot(w, Mag_fd, 'b', 'LineWidth', 2);
plot(w, Mag_cd, 'g', 'LineWidth', 2);
plot(w, Mag_lanczos, 'm', 'LineWidth', 2);
grid on;
xlabel('\omega (rad/sample)'); ylabel('Magnitude');
title('Frequency Responses (-\pi to \pi)');
```

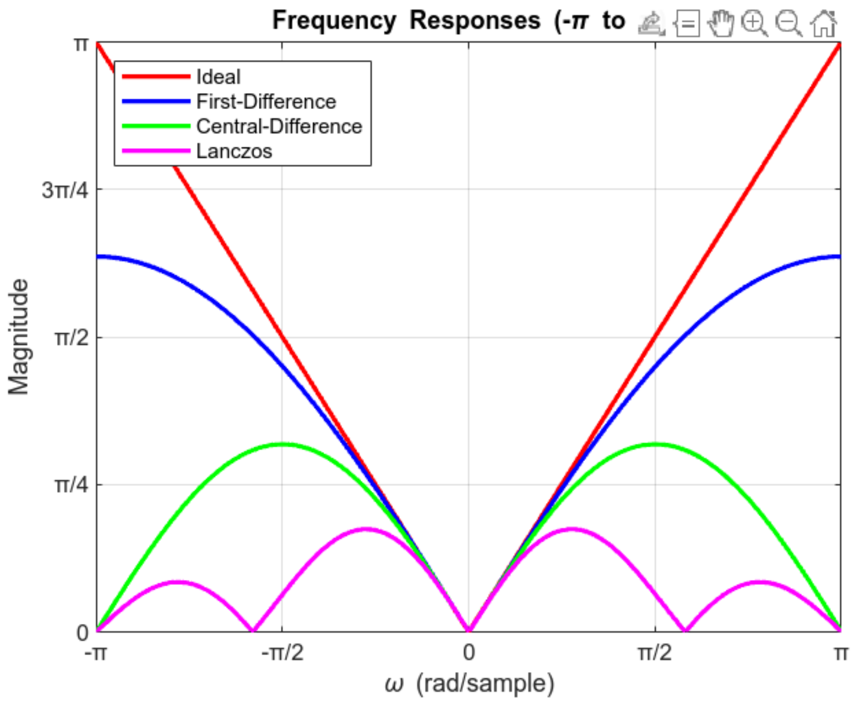
```matlab
xlim([-pi, pi]);
ylim([0, pi]);

xticks([-pi -pi/2 0 pi/2 pi]);
xticklabels({'-π','-π/2','0','π/2','π'});
yticks([0 pi/4 pi/2 3*pi/4 pi]);
yticklabels({'0','π/4','π/2','3π/4','π'});

legend('Ideal', 'First-Difference', 'Central-Difference', 'Lanczos', 'Location', 'NorthWest');

%%phase Responses
figure('Name','Phase Responses','NumberTitle','off');
subplot(3,1,1);
plot(w, angle(B_fd), 'b', 'LineWidth', 2);
grid on;
title('Phase Response: First-Difference');
xlabel('\omega (rad/sample)'); ylabel('Phase (rad)');

subplot(3,1,2);
plot(w, angle(B_cd), 'g', 'LineWidth', 2);
grid on;
title('Phase Response: Central-Difference');
xlabel('\omega (rad/sample)'); ylabel('Phase (rad)');

subplot(3,1,3);
plot(w, angle(B_lanczos), 'm', 'LineWidth', 2);
grid on;
title('Phase Response: Lanczos Differentiator');
xlabel('\omega (rad/sample)'); ylabel('Phase (rad)');

%%clean Sinusoid Output through Differentiators
n = 0:50;
x_clean = sin(0.2*pi*n);  %clean sinusoidal signal

y_fd_clean = firstDifference(x_clean);
y_cd_clean = centralDifference(x_clean);
y_lanczos_clean = lanczosDifferentiator(x_clean);

figure('Name','Clean Sinusoid Through Differentiators','NumberTitle','off');
subplot(4,1,1);
plot(n, x_clean, 'k', 'LineWidth', 1.5);
grid on;
title('Input: sin(0.2πn)');
xlabel('n'); ylabel('x(n)');

subplot(4,1,2);
plot(n, y_fd_clean, 'b', 'LineWidth', 1.5);
grid on;
title('Output: First-Difference');
xlabel('n'); ylabel('y_{FD}(n)');

subplot(4,1,3);
plot(n, y_cd_clean, 'g', 'LineWidth', 1.5);
grid on;
title('Output: Central-Difference');
xlabel('n'); ylabel('y_{CD}(n)');

subplot(4,1,4);
plot(n, y_lanczos_clean, 'm', 'LineWidth', 1.5);
```

```
grid on;
title('Output: Lanczos Differentiator');
xlabel('n'); ylabel('y_{L}(n)');

%%noisy Sinusoid Outputs
x_noisy = sin(0.2*pi*n) + 0.1*randn(size(n));

y_fd_noisy = firstDifference(x_noisy);
y_cd_noisy = centralDifference(x_noisy);
y_lanczos_noisy = lanczosDifferentiator(x_noisy);

figure('Name','Noisy Sinusoid Through Differentiators','NumberTitle','off');
subplot(4,1,1);
plot(n, x_noisy, 'k', 'LineWidth', 1.2);
grid on;
title('Noisy Sinusoid: x(n)');
xlabel('n'); ylabel('Amplitude');

subplot(4,1,2);
plot(n, y_fd_noisy, 'b', 'LineWidth', 1.2);
grid on;
title('Output: First-Difference');
xlabel('n'); ylabel('Amplitude');

subplot(4,1,3);
plot(n, y_cd_noisy, 'g', 'LineWidth', 1.2);
grid on;
title('Output: Central-Difference');
xlabel('n'); ylabel('Amplitude');

subplot(4,1,4);
plot(n, y_lanczos_noisy, 'm', 'LineWidth', 1.2);
grid on;
title('Output: Lanczos Differentiator');
xlabel('n'); ylabel('Amplitude');

%%function definitions
function y = firstDifference(x)
    % y(n) = x(n) - x(n-1)
    b = [1, -1];
    y = filter(b, 1, x);
end

function y = centralDifference(x)
    % y(n) = [x(n) - x(n-2)] / 2
    b = [0.5, 0, -0.5];
    y = filter(b, 1, x);
end

function y = lanczosDifferentiator(x)
    b = [0.2, 0.1, 0, -0.1, -0.2];
    y = filter(b, 1, x);
end
```

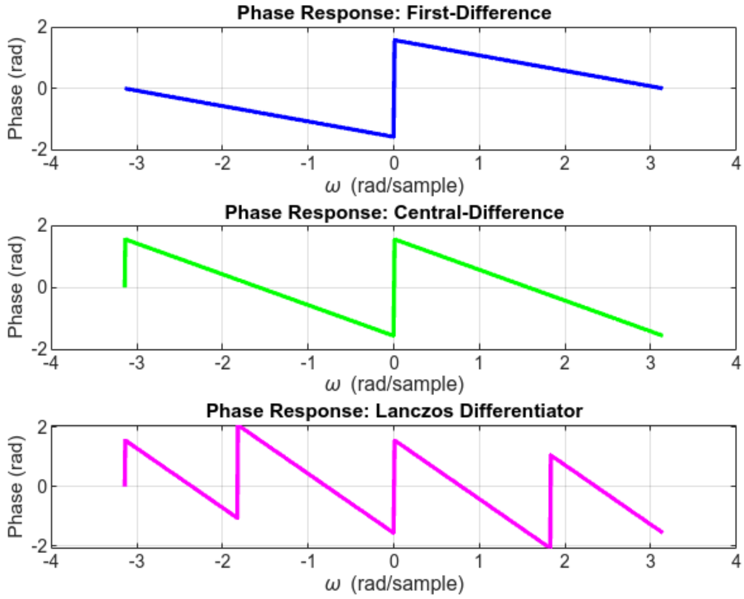
**outputs:**



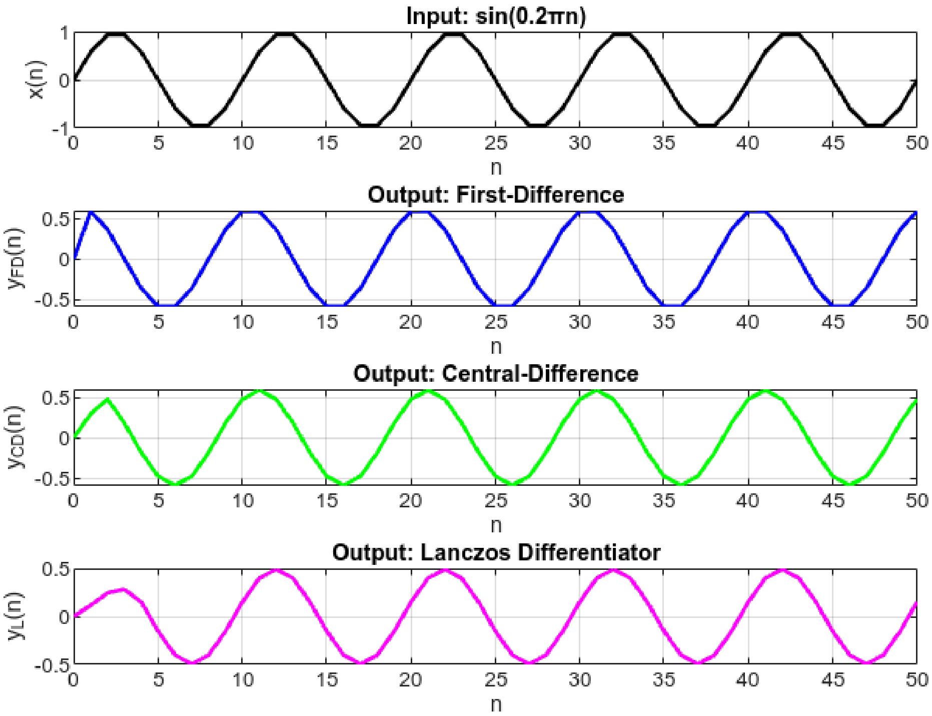
Impulse Response: First-Difference
Amplitude
Sample Index
Impulse Response: Central-Difference
Amplitude
Sample Index
Impulse Response: Lanczos Differentiator
Amplitude
Sample Index

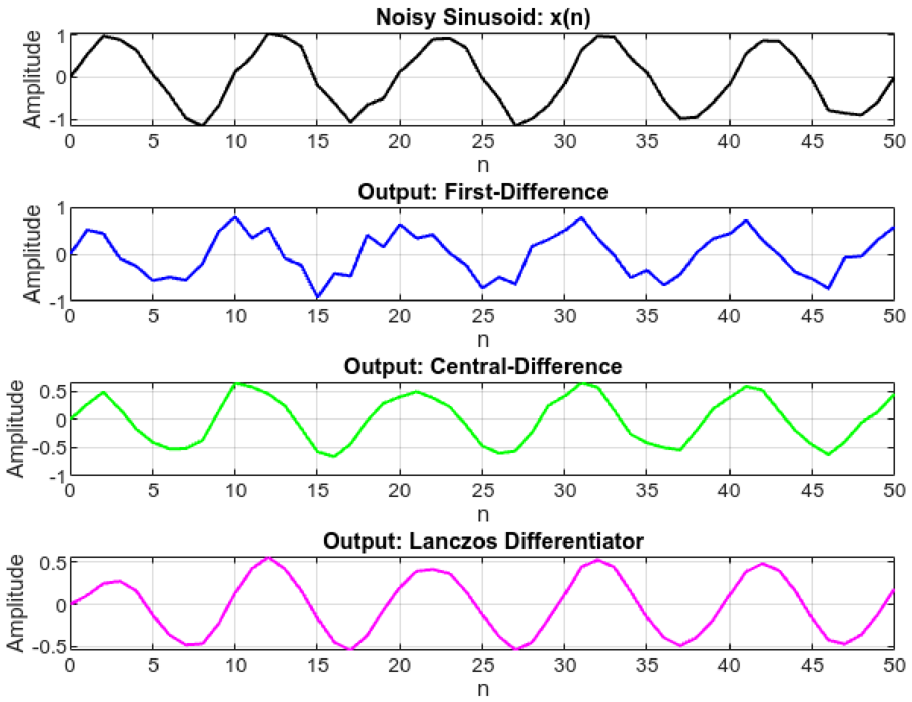
Frequency Responses (-π to
Ideal
First-Difference
Central-Difference
Lanczos
Magnitude
ω (rad/sample)



Phase Response: First-Difference
Phase (rad)
ω (rad/sample)
Phase Response: Central-Difference
Phase (rad)
ω (rad/sample)
Phase Response: Lanczos Differentiator
Phase (rad)
ω (rad/sample)

Input: sin(0.2πn)
x(n)
n
Output: First-Difference
y_FD(n)
n
Output: Central-Difference
y_CD(n)
n
Output: Lanczos Differentiator
y_L(n)
n




## Answers:

### 1. Impulse Responses:

- First-Difference Differentiator has coefficients `[1, -1]`. It contains two coefficients, and thus it is inherently short and simple. Its impulse response is asymmetrical because the coefficients are not symmetric around a central point. As a result, this filter does not have linear phase characteristics.
- Central-Difference Differentiator has coefficients `[0.5, 0, -0.5]`. It has three coefficients arranged symmetrically about the center (middle coefficient is at index 2, with equal magnitude coefficients at equal distances on both sides). Because the coefficients are symmetric (except for a sign reversal, making it an odd symmetric filter), this symmetry leads to linear phase characteristics, meaning it will delay all frequencies equally, preserving signal integrity and shape.
- Lanczos Differentiator has five coefficients `[0.2, 0.1, 0, -0.1, -0.2]`. Similar to the central-difference differentiator, these coefficients are symmetric around a central zero. This symmetry (odd symmetry in this case, equal magnitude and opposite signs) ensures linear phase behavior. With more taps (coefficients), the impulse response is longer and smoother, improving the accuracy of differentiation.

**Reasoning:** Symmetry in the impulse response means the filter introduces equal delays for all frequency components (linear phase). Asymmetric coefficients mean different frequencies are delayed differently (non-linear phase), causing signal distortion in practical situations.

**2. Frequency Responses:**

In the frequency response plots, we compare each differentiator's magnitude response against the ideal differentiator (a straight line, since ideal differentiation multiplies by frequency ω):

- **First-Difference Differentiator** closely matches the ideal line at low frequencies but diverges significantly as frequency increases, showing a clear deviation at higher frequencies. This deviation occurs because it only uses two coefficients and thus has limited ability to approximate a linear response over a broad frequency range.
- **Central-Difference Differentiator** follows the ideal line better than the First-Difference, particularly at mid-frequencies. At very low and very high frequencies, however, it still deviates noticeably. The extra coefficient allows it to approximate the ideal response more accurately, but the limited number of taps (only three) still restricts its effectiveness at higher frequencies.
- **Lanczos Differentiator** has a smoother and more accurate approximation at low frequencies compared to the other two differentiators due to its longer impulse response (five taps). The presence of more taps allows finer control over frequency response, making it the closest approximation to ideal differentiation among the three. However, at higher frequencies, it also shows significant deviation due to the inherent practical limits of any finite-length differentiator.

**Reasoning:**
More taps (coefficients) in a differentiator generally allow better control over the frequency response, improving accuracy at different frequencies. However, limited taps will always introduce ripples and deviations from the ideal response at higher frequencies.

**3. Clean Sinusoid Outputs:**

When a clean sinusoid (`sin(0.2πn)`) passes through each differentiator, the output is the derivative of the sine wave, which is a cosine wave.

- **First-Difference Differentiator** output has the largest amplitude because its magnitude response at 0.2π is highest among the three differentiators. Thus, the output is an amplified cosine wave, clearly reflecting its frequency response characteristics.
- **Central-Difference Differentiator** provides a cosine-shaped output with slightly smaller amplitude, due to its reduced magnitude response at 0.2π. This happens because its frequency response is closer to ideal, avoiding unnecessary amplification of mid-frequencies.
- **Lanczos Differentiator** output has the smallest amplitude, again reflecting its lower magnitude response at this frequency. This demonstrates its smoother behavior, providing stable differentiation at lower frequencies without unwanted amplification.

**Reasoning:**
The amplitude of the output at any given frequency directly reflects the magnitude response of the differentiator at that frequency.

**4. Noisy Sinusoid Outputs (Noise Behavior):**

In the presence of noise (`sin(0.2πn) + noise`), each differentiator's output differs greatly in noise amplification:

- **First-Difference Differentiator** heavily amplifies the noise, showing the most erratic and noisy output. This happens because its magnitude response increases significantly with frequency, causing it to amplify unwanted high-frequency noise.
- **Central-Difference Differentiator** suppresses noise better than the first-difference, producing a smoother output. This improved performance comes from having symmetric coefficients and slightly reduced gain at high frequencies, effectively attenuating the noise.
- **Lanczos Differentiator** offers the best noise suppression. Its longer impulse response with five taps smooths the response, providing a more controlled frequency response that attenuates the high-frequency noise effectively, resulting in the cleanest output.

**Reasoning:**
More taps and smoother frequency response generally lead to better noise suppression, since noise tends to have high-frequency components that sophisticated differentiators effectively attenuate.

**Question 1: Which differentiator provides better noise suppression?**
The Lanczos Differentiator clearly provides the best noise suppression, evident from the smoother and cleaner output due to its more sophisticated frequency response and increased number of taps.

**Question 2: How does group delay affect the alignment of signals in real-world applications?**
Group delay represents how signals of different frequencies are delayed differently by a filter. Nonlinear group delay causes distortion because different frequencies arrive at different times, misaligning signal components. Linear-phase filters, like the Central-Difference and Lanczos differentiators (due to symmetric coefficients), have constant group delay. Thus, all frequency components arrive simultaneously, preserving the shape and alignment of signals, which is crucial in applications like audio processing, communication systems, and radar systems.

**Question 3: What is the impact of increasing the number of taps in the specialized differentiator on the frequency response?**
Increasing the number of taps improves the differentiator's ability to closely approximate the ideal frequency response. It reduces ripples and deviations, especially at mid-to-high frequencies, leading to better noise suppression and accurate differentiation across a wider frequency range. The trade-off is increased computational complexity and latency, which can be an issue in real-time systems.

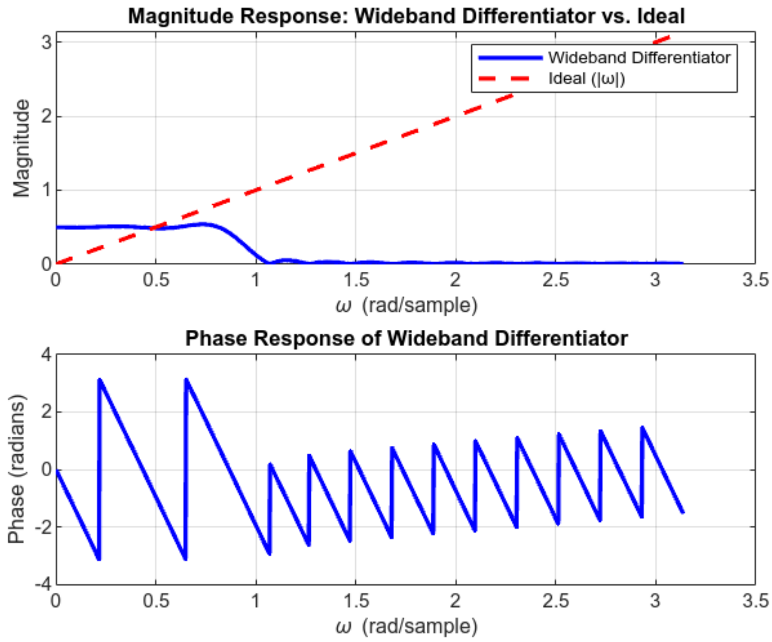
Part B)

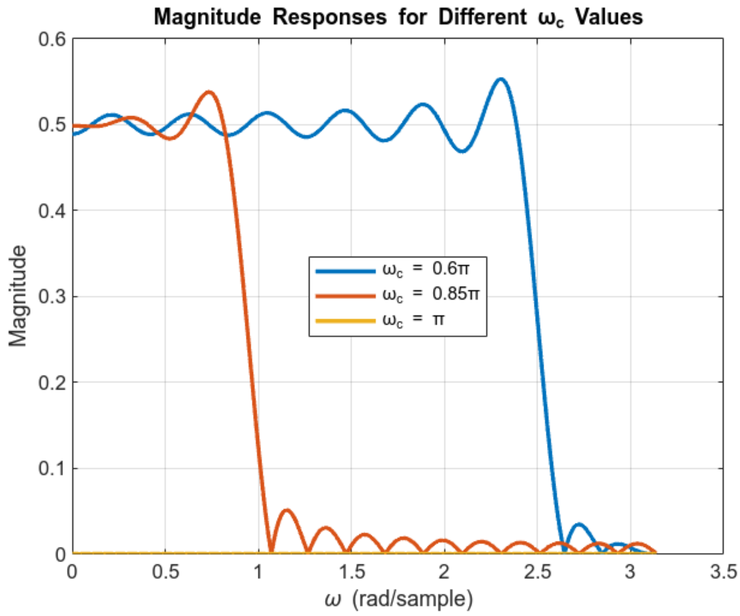
code:

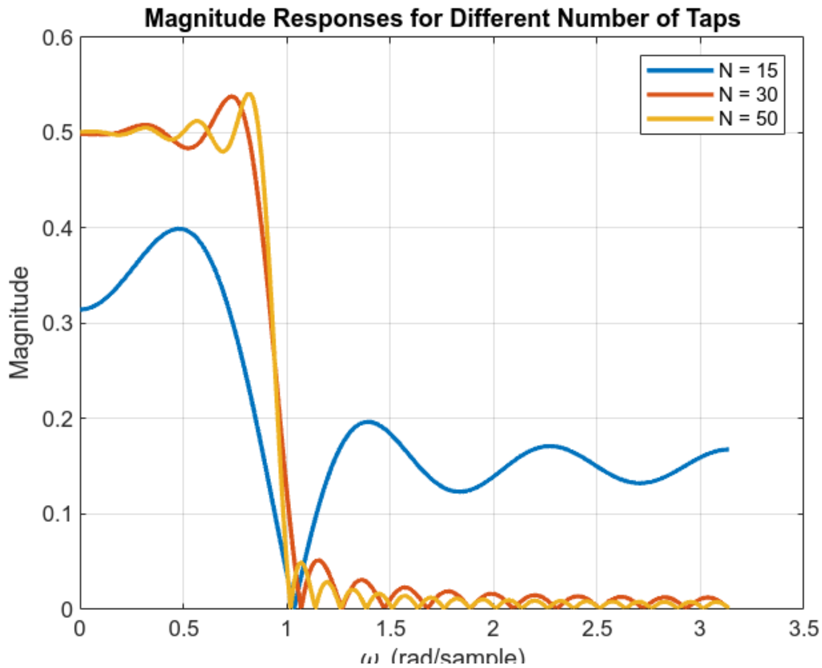
```
%part b
N = 30;
omega_c = 0.85 * pi;
hgen = widebandDifferentiator(N, omega_c);
figure('Name','Impulse Response of Wideband Differentiator','NumberTitle','off');
stem(0:(N-1), hgen, 'filled', 'LineWidth',1.5);
grid on;
title('Impulse Response: Wideband Differentiator (N = 30, ω_c = 0.85π)');
xlabel('Index k'); ylabel('h_{gen}(k)');

[H_gen, w] = freqz(hgen, 1, 1024);
Mag_ideal = abs(w);

figure('Name','Frequency Response: Wideband Differentiator','NumberTitle','off');
subplot(2,1,1);
plot(w, abs(H_gen), 'b', 'LineWidth',2); hold on;
plot(w, Mag_ideal, 'r--', 'LineWidth',2);
xlabel('\omega (rad/sample)');
ylabel('Magnitude');
title('Magnitude Response: Wideband Differentiator vs. Ideal');
legend('Wideband Differentiator','Ideal (|ω|)');
grid on;

subplot(2,1,2);
plot(w, angle(H_gen), 'b', 'LineWidth',2);
xlabel('\omega (rad/sample)');
ylabel('Phase (radians)');
title('Phase Response of Wideband Differentiator');
grid on;

omega_c_vals = [0.6*pi, 0.85*pi, pi];
figure('Name','Variation of ω_c','NumberTitle','off');
for i = 1:length(omega_c_vals)
    h_temp = widebandDifferentiator(N, omega_c_vals(i));
    [H_temp, w_temp] = freqz(h_temp, 1, 1024);
    plot(w_temp, abs(H_temp), 'LineWidth', 2); hold on;
end
xlabel('\omega (rad/sample)'); ylabel('Magnitude');
title('Magnitude Responses for Different ω_c Values');
legend('ω_c = 0.6π', 'ω_c = 0.85π', 'ω_c = π','Location','Best');
grid on;

N_vals = [15, 30, 50];
figure('Name','Impact of Number of Taps','NumberTitle','off');
for i = 1:length(N_vals)
    h_temp = widebandDifferentiator(N_vals(i), 0.85*pi);
    [H_temp, w_temp] = freqz(h_temp, 1, 1024);
    plot(w_temp, abs(H_temp), 'LineWidth', 2); hold on;
end
xlabel('\omega (rad/sample)'); ylabel('Magnitude');
title('Magnitude Responses for Different Number of Taps');
legend('N = 15', 'N = 30', 'N = 50','Location','Best');
grid on;
```

```matlab
hgen_windowed = hgen .* hamming(N)';
[H_gen_win, w_win] = freqz(hgen_windowed, 1, 1024);

figure('Name','Windowing Effect','NumberTitle','off');
plot(w, abs(H_gen), 'b', 'LineWidth', 2); hold on;
plot(w_win, abs(H_gen_win), 'k', 'LineWidth', 2);
xlabel('\omega (rad/sample)');
ylabel('Magnitude');
title('Wideband Differentiator: Non-windowed vs. Hamming Windowed');
legend('Non-windowed', 'Hamming Windowed','Location','Best');
grid on;

n = 0:100;
x_high = sin(0.8*pi*n);
x_low  = sin(0.2*pi*n);
y_high = filter(hgen, 1, x_high);
y_low  = filter(hgen, 1, x_low);

figure('Name','Practical Application','NumberTitle','off');
subplot(2,1,1);
plot(n, x_high, 'k', 'LineWidth', 1.5); hold on;
plot(n, y_high, 'b', 'LineWidth', 1.5);
title('High-Frequency Input (sin(0.8πn)) and Differentiator Output');
xlabel('n'); ylabel('Amplitude');
legend('Input','Output');
grid on;

subplot(2,1,2);
plot(n, x_low, 'k', 'LineWidth', 1.5); hold on;
plot(n, y_low, 'r', 'LineWidth', 1.5);
title('Low-Frequency Input (sin(0.2πn)) and Differentiator Output');
xlabel('n'); ylabel('Amplitude');
legend('Input','Output');
grid on;


function h = widebandDifferentiator(N, omega_c)
    M = (N - 1) / 2;
    h = zeros(1, N);
    for k = 0:(N - 1)
        d = k - M;
        if abs(d) < 1e-12
            h(k + 1) = 0;
        else
            h(k + 1) = (cos(omega_c * d) * sin(omega_c * d)) / (pi * d);
        end
    end
end
```

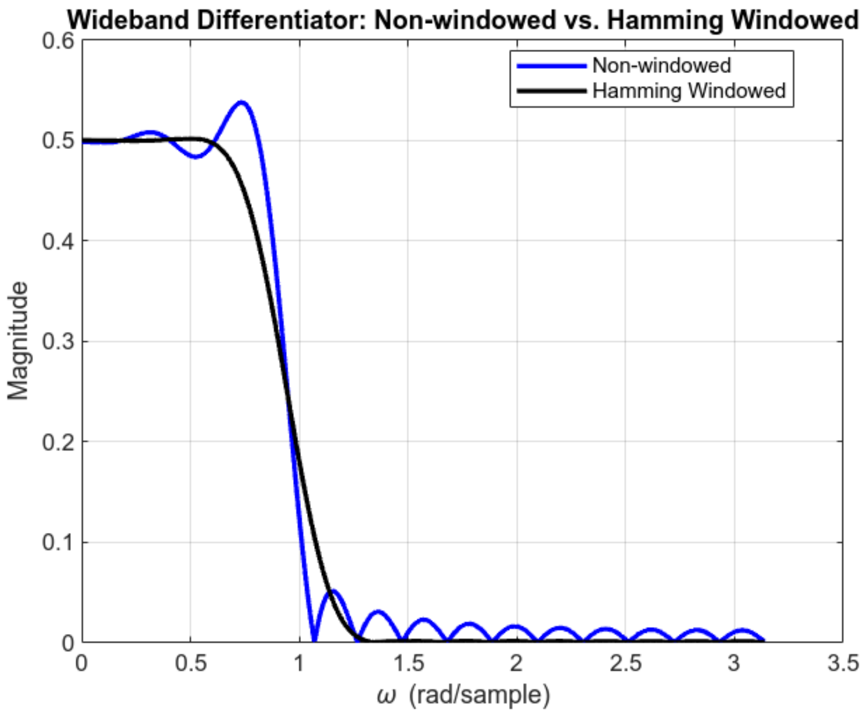
output:


Magnitude Response: Wideband Differentiator vs. Ideal
Wideband Differentiator
Ideal (|ω|)
Magnitude
ω (rad/sample)
Phase Response of Wideband Differentiator
Phase (radians)
ω (rad/sample)

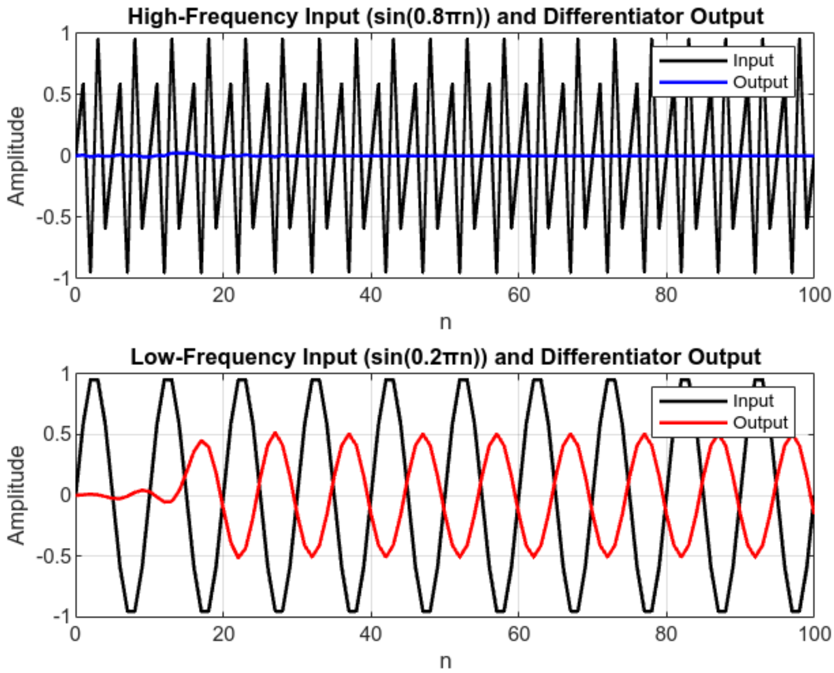

Magnitude Responses for Different ωc Values
ωc = 0.6π
ωc = 0.85π
ωc = π
Magnitude
ω (rad/sample)



Magnitude Responses for Different Number of Taps
N = 15
N = 30
N = 50
Magnitude

Wideband Differentiator: Non-windowed vs. Hamming Windowed
Non-windowed
Hamming Windowed
Magnitude
ω (rad/sample)




Part B
Discussion Answers:

- **What is the effect of increasing the cutoff frequency ωc on the differentiator's ability to approximate an ideal differentiator?**
  Increasing ωc expands the differentiator's frequency range, allowing differentiation of wider-bandwidth signals. However, it reduces accuracy at high frequencies due to pronounced ripples and deviations from the ideal response.

- **How does truncating the coefficients to a finite length affect the frequency response?**
  Truncating coefficients introduces ripples in the frequency response (Gibbs phenomenon). These ripples cause deviations from the ideal differentiator, particularly near the cutoff frequency.

- **What is the benefit of using windowing on differentiator coefficients?**
  Applying a window (e.g., Hamming window) reduces ripple effects caused by truncation, resulting in a smoother, more stable frequency response that better approximates the ideal differentiator.

- **Why is the choice of taps N crucial in designing wideband differentiators?**
  Choosing N impacts accuracy, computational complexity, and latency. A small N results in poor frequency approximation, while a large N improves performance but demands greater computational resources, highlighting the need for balance based on application requirements.

Observation of the outputs:

**Magnitude and Phase Response (Wideband vs. Ideal):**
The wideband differentiator magnitude response closely matches the ideal differentiator at lower frequencies but diverges significantly beyond the cutoff frequency ($\omega c = 0.85\pi$), indicating practical bandwidth limitation. The phase response shows linear segments separated by discontinuities caused by zeros in the frequency response, reflecting changes in the differentiator's sign, typical for finite-impulse response filters.

**Magnitude Responses for Different ωc Values:**
When varying ωc ($0.6\pi$, $0.85\pi$, $\pi$), a wider ωc increases bandwidth but introduces greater ripple and deviation from ideal differentiation. A narrower ωc ($0.6\pi$) produces better accuracy but restricts differentiation to lower frequencies, highlighting a trade-off between bandwidth and accuracy.

**Magnitude Responses for Different Number of Taps (N):**
Increasing the number of taps (15, 30, 50) improves accuracy by reducing ripple and extending linear approximation to higher frequencies. Shorter filters ($N=15$) exhibit larger ripples, while longer filters ($N=50$) are smoother, better approximating the ideal differentiator but requiring greater computational resources.

**Windowing Effect (Non-windowed vs. Hamming Windowed):**
Applying a Hamming window reduces ripple significantly, smoothing the frequency response and providing a closer match to the ideal differentiator. Non-windowed coefficients have higher ripples, negatively impacting differentiation accuracy, especially near cutoff frequencies.