10.2

✅ 5. Investigate the Gibbs Phenomenon

Observation:

From the generated plots, as more harmonics (higher N) are added, you can see ripples near sharp transitions or discontinuities.

Explanation:

This is the **Gibbs Phenomenon** — a characteristic overshoot (~9%) near discontinuities that does **not disappear**, even as $N \to \infty$. The overshoot becomes narrower but remains in height.

In Your Code:

- The overshoot is clearly visible when $N = 10$, becomes sharper at $N = 20$, and more localized at $N = 65$.

Done ↓

✅ 6. What is the Effect of Increasing N on the Plots?

Summary:

- The approximation of $x(t)$ improves.
- Sharp features and edges become more accurate.
- More high-frequency content is captured.
- Gibbs phenomenon is still present near discontinuities.
- The signal converges closer to the original with increasing N.

✅ 4. Generate Plots for Different N Values

Example script to visualize:

matlab

Copy Edit

```matlab
t = -2:0.0001:2;
N_values = [5, 10, 20, 65];

for i = 1:4
    N = N_values(i);
    x = fourier_series_xt(t, N);
    subplot(2,2,i);
    plot(t, x);
    grid on;
    xlabel(['x', num2str(i), '(t)']);
    ylabel('Time(t)');
    title(['Signal with terms up to ', num2str(N)]);
end
```

This generates the exact subplot figure like you attached — four approximations of $x(t)$ using increasing numbers of harmonics.

Done ↓

✅ 3. MATLAB Function for $x(t)$

You can create a function as:

matlab

```matlab
function x = fourier_series_xt(t, N)
    x = zeros(size(t));
    for n = 1:2:N  % Only odd harmonics
        bn = (8/(pi^2)) * (1/n^2) * sin(n*pi/2);
        x = x + bn * sin(n * pi * t);
    end
end
```

✅ 2. Find the Unknowns for Writing the Function

⬛ Range of Time:

The time vector in the code is:

```matlab
t = -2:0.0001:2;
```

This represents 2 full periods since the fundamental period $T_0 = \frac{2\pi}{\omega_0} = \frac{2\pi}{\pi} = 2$.

⬛ Input Parameters of the Function:

- Time vector `t`
- Number of harmonics `N` (odd values only)
- Fourier coefficient formula for b_n

✅ 1. Synthesize x(t) using Fourier Series Coefficients

You are given:

- $a_0 = 0$, $a_n = 0$
- $b_n = \frac{8}{n^2\pi^2} \sin\left(\frac{n\pi}{2}\right)$
- Fundamental frequency: $\omega_0 = \pi$

So, the signal $x(t)$ is synthesized as:

$$x(t) = \sum_{n=1,3,5,\dots}^{N} \frac{8}{n^2\pi^2} \sin\left(\frac{n\pi}{2}\right) \cdot \sin(n\pi t)$$

10.1

✅ 5. Plotting for Various N

Your plotting code is correct. Just make sure the function is saved in a file named `xt_original.m` or defined in the same script. Here's the whole plotting section for clarity:

matlab Copy Edit

```matlab
t = -1:0.001:1;
N_vals = [5, 10, 15, 60];

figure;
for i = 1:length(N_vals)
    xt = xt_original(t, N_vals(i));

    subplot(2,2,i);
    plot(t, xt, 'LineWidth', 1.5);
    title(['x(t) with N = ', num2str(N_vals(i))]);
    xlabel('Time (t)');
    ylabel('x(t)');
    grid on;
end
```

Done ↓

✅ 4. MATLAB Function for $x(t)$

You already wrote the function nearly correctly! Here's the **clean version**:

matlab Copy Edit

```matlab
function xt = xt_original(t, N)
    xt = 0.5 * ones(size(t));  % DC term a0
    for n = 1:2:N  % Only odd harmonics
        bn = 1/pi;
        xt = xt + bn * sin(2*pi*n*t);
    end
end
```

Done ↓

✅ 3. Unknowns for Writing the Function

Time Range:

matlab

```matlab
t = -1:0.001:1;  % Two full periods of the signal since T = 1
```

Input Parameters of the Function:

- Time vector `t`
- Number of harmonics `N` (odd terms only)
- $b_n = \frac{1}{\pi}$ for odd n

4. MATLAB Funct

You already wrote the function ne

$0,\quad \text{if } n \text{ is even}$

✅ 2. Synthesize $x(t)$ Using Fourier Coefficients

$$x(t) = \frac{1}{2} + \sum_{n=1,3,5,\ldots}^{N} \frac{1}{\pi} \cdot \sin(n \cdot 2\pi t)$$

Only **odd harmonics** contribute due to the nature of b_n.

✅ 3. Unknowns for Writing the Function

Done ↓

✅ 1. Given Trigonometric Fourier Coefficients

We are told:

- $\omega_0 = 2\pi \Rightarrow T = 1$
- $a_0 = \frac{1}{2}$
- $a_n = 0$
- $b_n = \frac{1}{2\pi}[1 - \cos(n\pi)]$

> Since $\cos(n\pi) = (-1)^n$, then:

$$b_n = \frac{1}{2\pi}\left[1 - (-1)^n\right] \Rightarrow b_n = \begin{cases} \frac{1}{\pi}, & \text{if } n \text{ is odd} \\ 0, & \text{if } n \text{ is even} \end{cases}$$

✅ 6. Gibbs Phenomenon Explanation

■ Observation from Plots:

- Near discontinuities (e.g. edges or step changes), you'll see ripples/overshoots.
- These become sharper and more localized as N increases.

■ What is Gibbs Phenomenon?

It's the **persistent overshoot (~9%)** near discontinuities when approximating a non-smooth periodic signal with a Fourier series.

Even when $N \to \infty$, the **height of the overshoot doesn't go away**, only the width narrows.

This is seen clearly in your higher-N plots like $N = 60$.

✅ 7. What is the Effect of Increasing N on the Plots?

As N increases...	Effect
Accuracy	Improves — the waveform resembles the actual signal more closely
Resolution	More details and sharp transitions are captured
Overshoot	Gibbs phenomenon becomes narrower but remains in height
Smoothness	The approximation follows the true function more tightly
High-frequency components	More are included, improving fidelity