EM 317 COMPUTATIONAL METHODS

Assignment 05

SAMARATHUNGE S.M.T.N. (E/19/346)

Question 01

Given the Function,

$$f(x) = \begin{cases} x + \pi & \text{if } -\pi \leq x \leq 0 \\ \pi & \text{if } 0 < x < \pi \end{cases}$$

Fourier Series Formula,

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos(nx) + b_n \sin(nx) \right)$$

Coefficients Formulas,

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\, dx$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx)\, dx$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx)\, dx$$

In [6]:

```
from sympy import symbols, pi, integrate, cos, sin, Piecewise

# Define the variable and function
x, n = symbols('x n')
f = Piecewise((x + pi, (x >= -pi) & (x <= 0)), (pi, (x > 0) & (x < pi)))

# Calculate a0, an, bn
a0 = (1/pi) * integrate(f, (x, -pi, pi))
an = (1/pi) * integrate(f * cos(n*x), (x, -pi, pi))
bn = (1/pi) * integrate(f * sin(n*x), (x, -pi, pi))

a0, an, bn
```

Out[6]:
```
(3*pi/2,
 Piecewise((pi*sin(pi*n)/n - cos(pi*n)/n**2 + n**(-2), (n > -oo) & (n < oo) & Ne(n,
0)), (3*pi**2/2, True))/pi,
 Piecewise((-pi*cos(pi*n)/n + sin(pi*n)/n**2, (n > -oo) & (n < oo) & Ne(n, 0)), (0, T
rue))/pi)
```

Calculated Coefficients,

$$a_0 = \frac{3\pi}{2}$$

$$a_n = \frac{1}{\pi}\left(\frac{\pi\sin(\pi n)}{n} - \frac{\cos(\pi n)}{n^2} + \frac{1}{n^2}\right)$$

$$b_n = \frac{1}{\pi}\left(-\frac{\pi\cos(\pi n)}{n} + \frac{\sin(\pi n)}{n^2}\right)$$

In [7]:
```python
from sympy import sin, cos, simplify

# Simplify an and bn for n even and odd
an_even = simplify(an.subs(n, 2*n))
an_odd = simplify(an.subs(n, 2*n + 1))
bn_even = simplify(bn.subs(n, 2*n))
bn_odd = simplify(bn.subs(n, 2*n + 1))

an_even, an_odd, bn_even, bn_odd
```

Out[7]:
```
(Piecewise(((2*pi*n*sin(2*pi*n) - cos(2*pi*n) + 1)/(4*pi*n**2), ((n > -oo) | (n > 0))
& ((n > -oo) | (n < oo)) & ((n > 0) | (n < 0)) & ((n < 0) | (n < oo))), (3*pi/2, Tru
e)),
 Piecewise(((-pi*(2*n + 1)*sin(2*pi*n) + cos(2*pi*n) + 1)/(pi*(2*n + 1)**2), ((n > -o
o) | (n > -1/2)) & ((n > -oo) | (n < oo)) & ((n > -1/2) | (n < -1/2)) & ((n < -1/2) |
(n < oo))), (3*pi/2, True)),
 Piecewise(((-2*cos(2*pi*n) + sin(2*pi*n)/(pi*n))/(4*n), ((n > -oo) | (n > 0)) & ((n
> -oo) | (n < oo)) & ((n > 0) | (n < 0)) & ((n < 0) | (n < oo))), (0, True)),
 Piecewise(((pi*(2*n + 1)*cos(2*pi*n) - sin(2*pi*n))/(pi*(2*n + 1)**2), ((n > -oo) |
(n > -1/2)) & ((n > -oo) | (n < oo)) & ((n > -1/2) | (n < -1/2)) & ((n < -1/2) | (n <
oo))), (0, True)))
```

Simplified Coefficients for Even and Odd n,

For even n,

$$a_n = \frac{2\pi n\sin(2\pi n) - \cos(2\pi n) + 1}{4\pi n^2}$$

$$b_n = \frac{-2\cos(2\pi n) + \frac{\sin(2\pi n)}{\pi n}}{4n}$$

For odd n,

$$a_n = \frac{-\pi(2n+1)\sin(2\pi n) + \cos(2\pi n) + 1}{\pi(2n+1)^2}$$

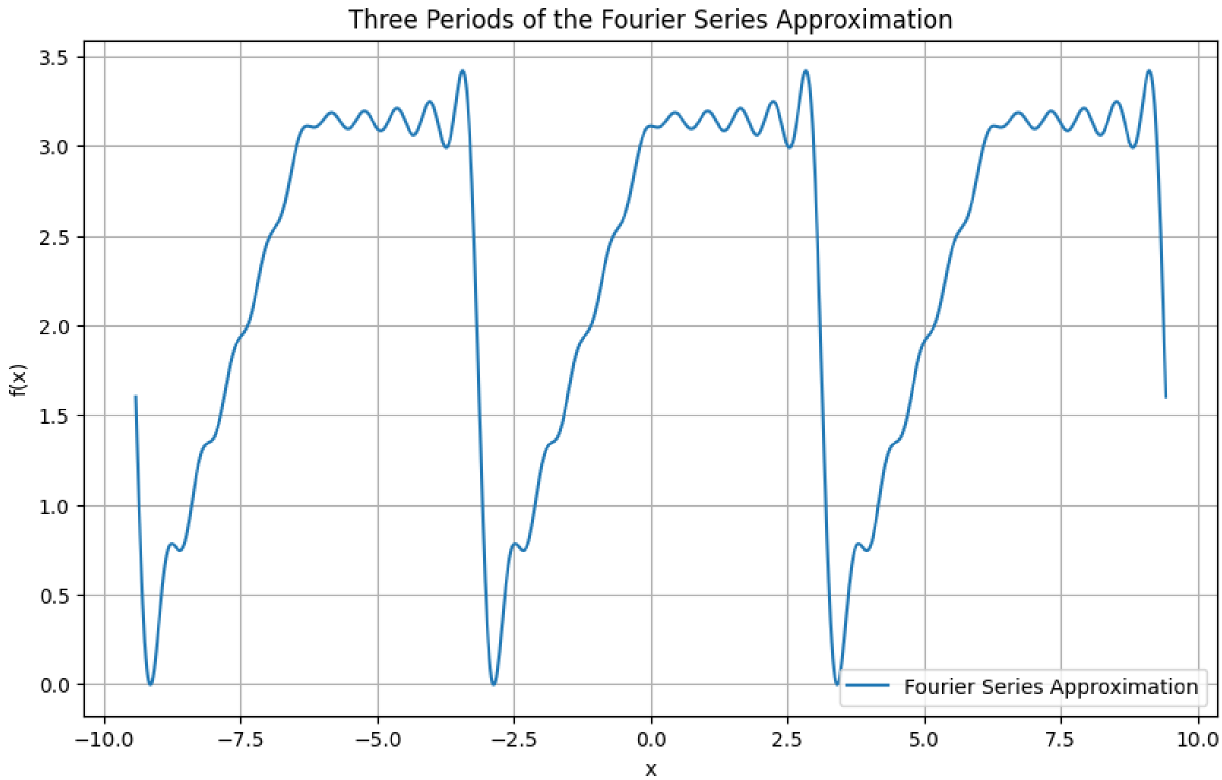
$$b_n = \frac{\pi(2n+1)\cos(2\pi n) - \sin(2\pi n)}{\pi(2n+1)^2}$$

Where,

$$f(x) \sim \frac{3\pi}{4} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$$

In [8]:

```
import numpy as np
import matplotlib.pyplot as plt

# Define the Fourier series function
def fourier_series(x, N):
    a0 = 3 * np.pi / 2
    series = a0 / 2
    for n in range(1, N + 1):
        an = (np.pi * np.sin(np.pi * n) / n - np.cos(np.pi * n) / n**2 + 1 / n**2) / n
        bn = (-np.pi * np.cos(np.pi * n) / n + np.sin(np.pi * n) / n**2) / np.pi
        series += an * np.cos(n * x) + bn * np.sin(n * x)
    return series

# Plotting
x = np.linspace(-3 * np.pi, 3 * np.pi, 1000)
N = 10  # Number of terms in the Fourier series

plt.figure(figsize=(10, 6))
plt.plot(x, fourier_series(x, N), label="Fourier Series Approximation")
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Three Periods of the Fourier Series Approximation')
plt.grid(True)
plt.legend()
plt.show()
```



Question 02

(a) Given the function:

$$f(x) = \begin{cases} x & \text{if } 0 \leq x < 1 \\ 1 & \text{if } 1 \leq x < 2 \end{cases}$$

Since f is odd, \

$$a_0 = 0 \; and \; a_n = 0$$

Therefore, for an odd periodic function with period P, the Fourier sine series is given by:

$$f(x) \sim \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{L}\right)$$

where $L = P/2$ and

$$b_n = \frac{2}{L} \int_0^L f(x) \sin\left(\frac{n\pi x}{L}\right) dx$$

Here, $L = 4/2 = 2$.

$$b_n = \frac{2}{2} \int_0^2 f(x) \sin\left(\frac{n\pi x}{2}\right) dx$$

Since $f(x)$ is piecewise defined, the integral will be split into two parts:

$$b_n = \int_0^1 x \sin\left(\frac{n\pi x}{2}\right) dx + \int_1^2 1 \cdot \sin\left(\frac{n\pi x}{2}\right) dx$$

Calculating b_n:

In [10]:

```
from sympy import symbols, pi, integrate, sin

# Defining the symbols
x, n = symbols('x n')

# Defining the function for the two intervals
f1 = x * sin(n*pi*x/2)
f2 = sin(n*pi*x/2)

# Calculate bn for each n
bn = integrate(f1, (x, 0, 1)) + integrate(f2, (x, 1, 2))

bn.simplify()
```

Out[10]:

$$\begin{cases} \frac{2\left(-\pi n \cos\left(\pi n\right) + 2 \sin\left(\frac{\pi n}{2}\right)\right)}{\pi^2 n^2} & \text{for } (n > -\infty \vee n > 0) \wedge (n > -\infty \vee n < \infty) \wedge (n > 0 \vee n < 0) \wedge \\ 0 & \text{otherwise} \end{cases}$$

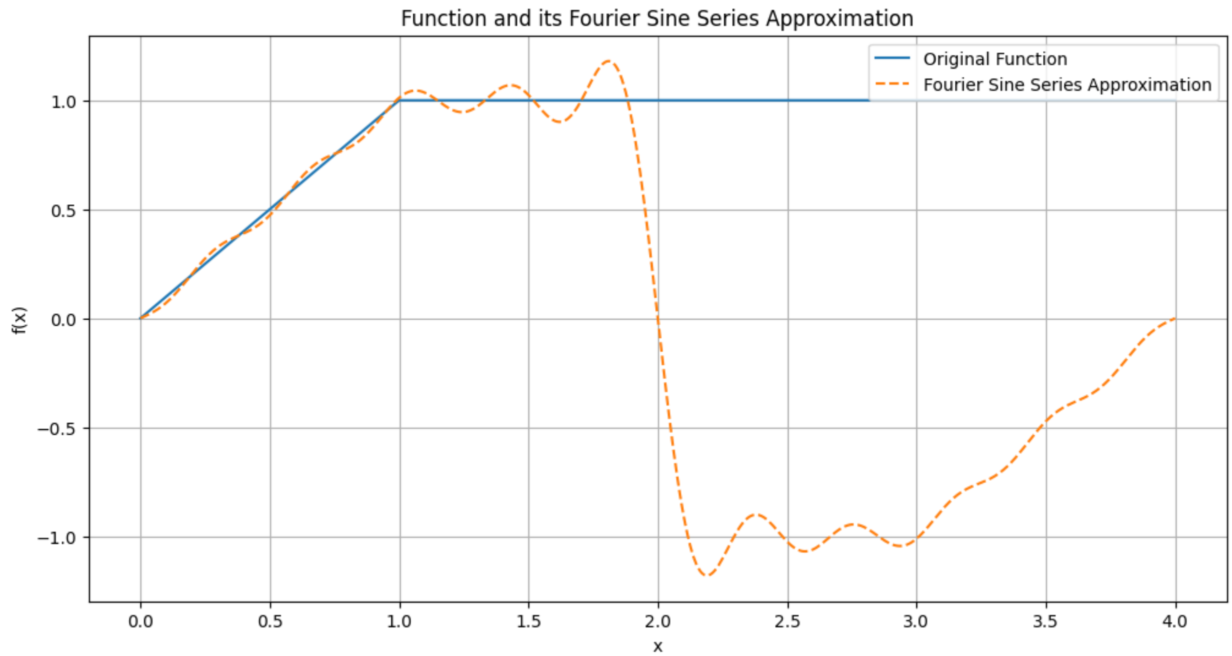
Simplified Coefficients,

$$b_n = \frac{2\left(-\pi n \cos(\pi n) + 2 \sin\left(\frac{\pi n}{2}\right)\right)}{\pi^2 n^2}$$

In [11]:

```
import numpy as np
import matplotlib.pyplot as plt

# Defining the original function
def original_function(x):
    return np.where(x < 1, x, 1)

# Defining the Fourier sine series function
def fourier_sine_series(x, N):
    L = 2
    series = 0
    for n in range(1, N + 1):
        bn = 2 * (-np.pi * n * np.cos(np.pi * n) + 2 * np.sin(np.pi * n / 2)) / (np.pi
        series += bn * np.sin(n * np.pi * x / L)
    return series

# Plotting
x_values = np.linspace(0, 4, 500)  # One period
N = 10  # Number of terms in the Fourier series

plt.figure(figsize=(12, 6))
plt.plot(x_values, original_function(x_values), label="Original Function")
plt.plot(x_values, fourier_sine_series(x_values, N), label="Fourier Sine Series Approx
plt.xlabel('x')
plt.ylabel('f(x)')
```

```
plt.title('Function and its Fourier Sine Series Approximation')
plt.grid(True)
plt.legend()
plt.show()
```



(b) Given the function,

$$f(x) = L - x, \quad 0 \leq x \leq L$$

Fourier Cosine Series Formula,

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi x}{L}\right)$$

Coefficient Formulas,

$$a_0 = \frac{1}{L} \int_0^L f(x)\, dx$$

$$a_0 = \frac{1}{L} \int_0^L (L - x)\, dx$$

$$a_n = \frac{2}{L} \int_0^L f(x) \cos\left(\frac{n\pi x}{L}\right) dx$$

$$a_n = \frac{2}{L}\int_0^L (L-x)\cos\left(\frac{n\pi x}{L}\right)dx$$

In [12]:
```
from sympy import symbols, pi, integrate, cos

# Defining the symbols
x, L, n = symbols('x L n')

# Defining the function
f = L - x

# Calculating a0 and an
a0 = (1/L) * integrate(f, (x, 0, L))
an = (2/L) * integrate(f * cos(n*pi*x/L), (x, 0, L))

a0.simplify(), an.simplify()
```

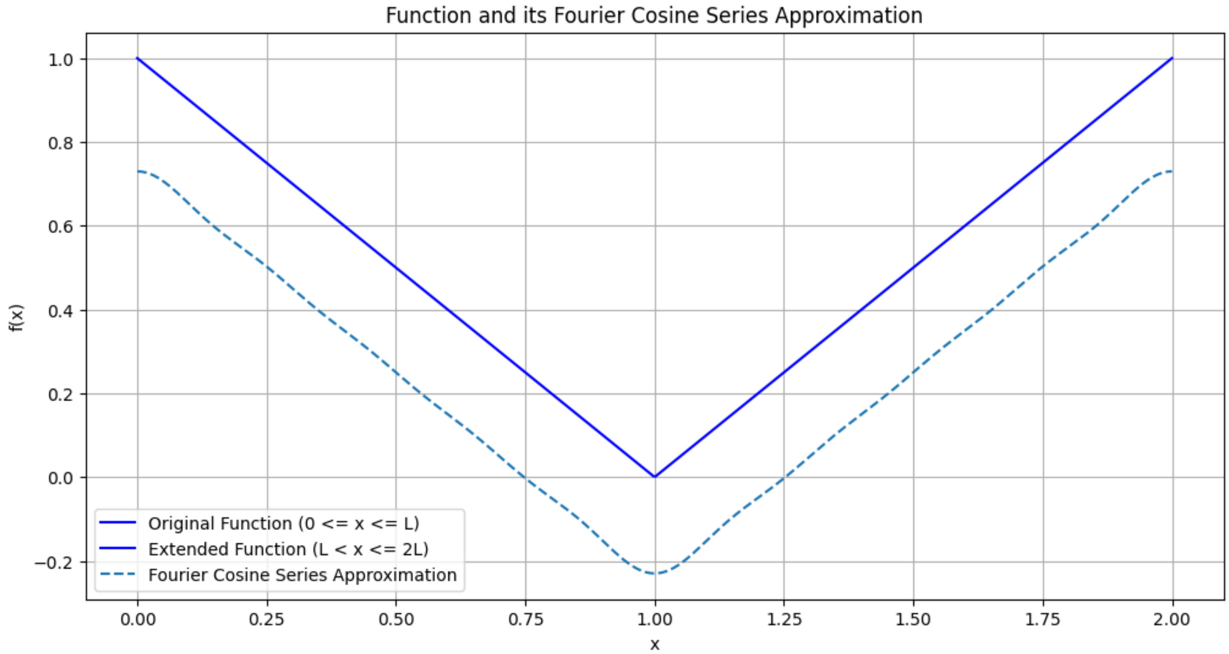
Out[12]:
```
(L/2,
 Piecewise((2*L*(1 - cos(pi*n))/(pi**2*n**2), ((n > -oo) | (n > 0)) & ((n > -oo) | (n
< oo)) & ((n > 0) | (n < 0)) & ((n < 0) | (n < oo))), (L, True)))
```

Calculated Coefficients,

$$a_0 = \frac{L}{2}$$

$$a_n = \frac{2L(1-\cos(\pi n))}{\pi^2 n^2}$$

In [13]:
```
# Define L and the Fourier cosine series function
L_value = 1
def fourier_cosine_series(x, N, L):
    a0 = L / 2
    series = a0 / 2
    for n in range(1, N + 1):
        an = 2 * L * (1 - np.cos(np.pi * n)) / (np.pi**2 * n**2)
        series += an * np.cos(n * np.pi * x / L)
    return series

# Plotting
x_values = np.linspace(0, 2 * L_value, 500)  # One period
N = 10  # Number of terms in the Fourier series

plt.figure(figsize=(12, 6))
plt.plot(x_values[:250], L_value - x_values[:250], label="Original Function (0 <= x <=
plt.plot(x_values[250:], x_values[250:] - L_value, label="Extended Function (L < x <=
plt.plot(x_values, fourier_cosine_series(x_values, N, L_value), label="Fourier Cosine
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Function and its Fourier Cosine Series Approximation')
plt.grid(True)
plt.legend()
plt.show()
```



Question 03

(a) Given the function,

$$f(x) = \begin{cases} x & \text{if } 0 \leq x < 0.5 \\ 0.5 & \text{if } 0.5 \leq x \leq 1 \end{cases}$$

$$O(x) = f(x) = \begin{cases} x & \text{if } 0 \leq x < 0.5 \\ 0.5 & \text{if } 0.5 \leq x \leq 1 \end{cases}$$

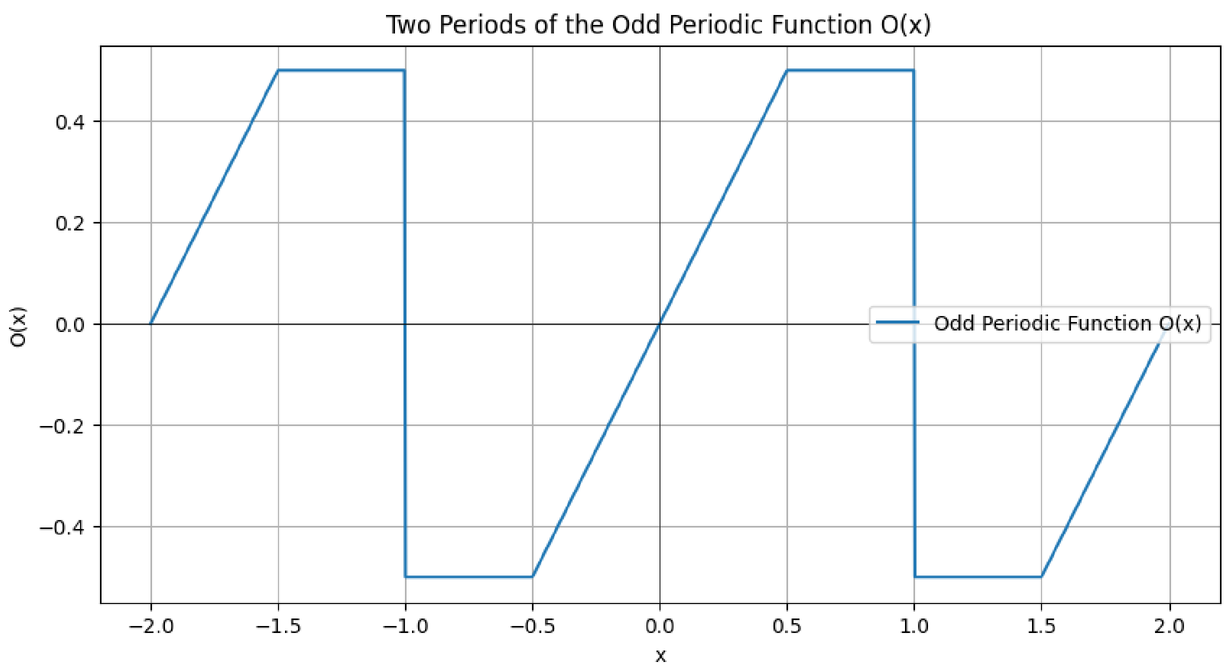
$$O(x) = -f(-x) = \begin{cases} -x & \text{if } -0.5 < x < 0 \\ -0.5 & \text{if } -1 \leq x \leq -0.5 \end{cases}$$

The function $O(x)$ is periodic with a period of 2.

In [14]:

```python
# Defining the odd periodic function O(x)
def odd_periodic_function(x):
    # Adjusting x for the periodicity
    x = x % 2
    if x > 1:
        x = x - 2

    # Defining O(x) based on the piecewise conditions
    if 0 <= x < 0.5:
        return x
    elif 0.5 <= x <= 1:
        return 0.5
    elif -0.5 < x < 0:
        return x
    elif -1 <= x <= -0.5:
        return -0.5
```

```python
    else:
        return 0

# x values for two periods
x_values = np.linspace(-2, 2, 1000)

# y values for O(x)
y_values = np.array([odd_periodic_function(x) for x in x_values])

# Plotting the function
plt.figure(figsize=(10, 5))
plt.plot(x_values, y_values, label="Odd Periodic Function O(x)")
plt.xlabel('x')
plt.ylabel('O(x)')
plt.title('Two Periods of the Odd Periodic Function O(x)')
plt.axhline(0, color='black',linewidth=0.5)
plt.axvline(0, color='black',linewidth=0.5)
plt.grid(True)
plt.legend()
plt.show()
```



(b) Finding the Fourier Series:

$$f(x) = \begin{cases} x & \text{if } 0 \leq x < 0.5 \\ 0.5 & \text{if } 0.5 \leq x \leq 1 \end{cases}$$

$$O(x) = \begin{cases} x & \text{if } 0 \leq x < 0.5 \\ 0.5 & \text{if } 0.5 \leq x \leq 1 \\ -x & \text{if } -0.5 < x < 0 \\ -0.5 & \text{if } -1 \leq x \leq -0.5 \end{cases}$$

Fourier Sine Series Formula for Odd Functions:

$$O(x) \sim \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{L}\right)$$

Coefficient Formula for b_n :

$$b_n = \frac{2}{L} \int_0^L O(x) \sin\left(\frac{n\pi x}{L}\right) dx$$

$$b_n = \frac{2}{L} \left(\int_0^{0.5} x \sin\left(\frac{n\pi x}{L}\right) dx + \int_{0.5}^{1} 0.5 \sin\left(\frac{n\pi x}{L}\right) dx \right)$$

In [16]:
```
# The function O(x) for the integration
Ox = Piecewise((x, (x >= 0) & (x < 0.5)), (0.5, (x >= 0.5) & (x <= 1)))

# Calculate bn
bn = (2/L) * integrate(Ox * sin(n*pi*x/L), (x, 0, L))
```

In [17]:
```
# Simplify bn with L = 1
L_value = 1
bn_simplified = bn.subs(L, L_value).simplify()

bn_simplified
```

Out[17]:

$$\begin{cases} \frac{-1.0\pi n \cos\left(\pi n\right)+2.0 \sin\left(\frac{\pi n}{2}\right)}{\pi^2 n^2} & \text{for}\ n > 0 \vee n < 0 \\ 0 & \text{otherwise} \end{cases}$$

Simplified Coefficient b_n:

$$b_n = \frac{-\pi n \cos(\pi n) + 2 \sin\left(\frac{\pi n}{2}\right)}{\pi^2 n^2}$$

Thus, the Fourier sine series for $O(x)$ is given by:

$$O(x) \sim \sum_{n=1}^{\infty} \frac{-\pi n \cos(\pi n) + 2 \sin\left(\frac{\pi n}{2}\right)}{\pi^2 n^2} \sin\left(\frac{n\pi x}{L}\right)$$

(c) Fourier Sine Series Formula for Odd Functions:

$$O(x) \sim \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{L}\right)$$

Coefficient Formula for b_n:

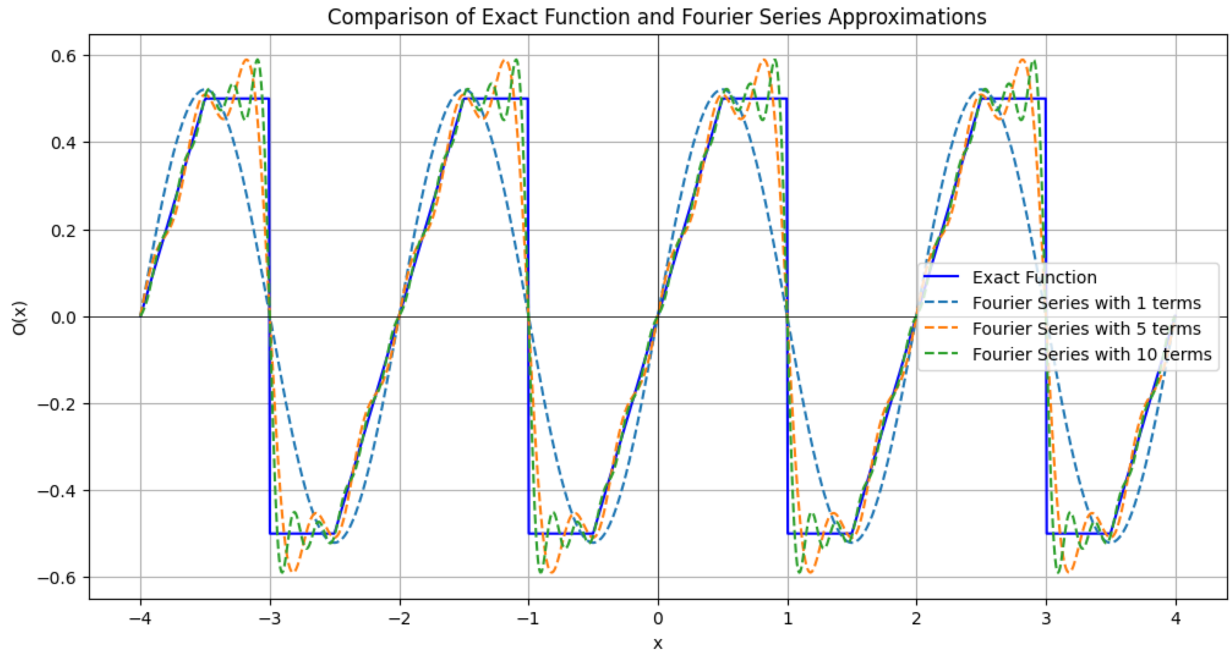
$$b_n = \frac{-\pi n \cos(\pi n) + 2\sin\left(\frac{\pi n}{2}\right)}{\pi^2 n^2}$$

for $n \neq 0$ and $L = 1$.

In [18]:

```
import numpy as np
import matplotlib.pyplot as plt

# Function to calculate the nth coefficient bn
def fourier_coefficient(n):
    return (-np.pi * n * np.cos(np.pi * n) + 2 * np.sin(np.pi * n / 2)) / (np.pi**2 *

# Function to calculate the Fourier series approximation
def fourier_series(x, N, L=1):
    series = 0
    for n in range(1, N + 1):
        bn = fourier_coefficient(n)
        series += bn * np.sin(n * np.pi * x / L)
    return series

# Define the exact function O(x)
def exact_function(x):
    x_mod = x % 2
    if x_mod > 1:
        x_mod = x_mod - 2

    if 0 <= x_mod < 0.5:
        return x_mod
    elif 0.5 <= x_mod <= 1:
        return 0.5
    elif -0.5 < x_mod < 0:
        return x_mod
    elif -1 <= x_mod <= -0.5:
        return -0.5
    return 0

# x values for the plot
x_values = np.linspace(-4, 4, 1000)

# Plotting
plt.figure(figsize=(12, 6))

# Plotting the exact function
plt.plot(x_values, [exact_function(x) for x in x_values], label="Exact Function", colo

# Plotting Fourier series approximations with different numbers of terms
terms = [1, 5, 10]
for N in terms:
    plt.plot(x_values, fourier_series(x_values, N), label=f"Fourier Series with {N} te

plt.xlabel('x')
plt.ylabel('O(x)')
plt.title('Comparison of Exact Function and Fourier Series Approximations')
plt.axhline(0, color='black',linewidth=0.5)
plt.axvline(0, color='black',linewidth=0.5)
plt.grid(True)
```

```
plt.legend()
plt.show()
```



(d) Given the function,

$$f(x) = \begin{cases} x & \text{if } 0 \leq x < 0.5 \\ 0.5 & \text{if } 0.5 \leq x \leq 1 \end{cases}$$

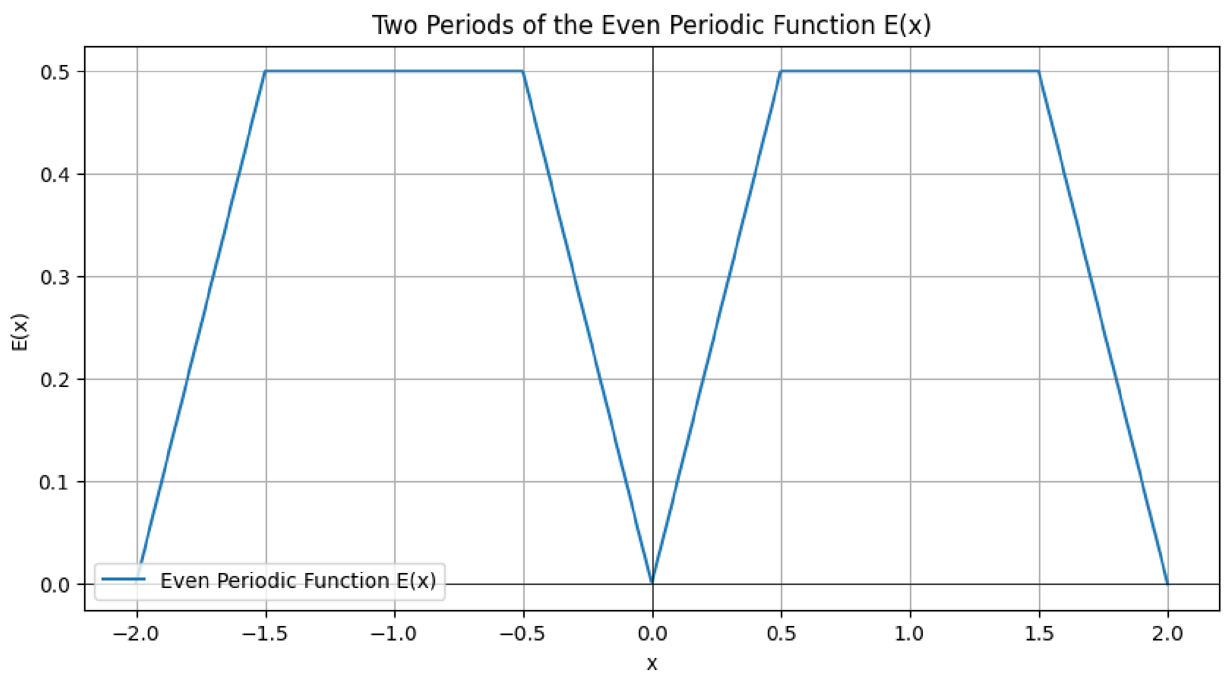
$$E(x) = f(x) = \begin{cases} x & \text{if } 0 \leq x < 0.5 \\ 0.5 & \text{if } 0.5 \leq x \leq 1 \end{cases}$$

$$E(x) = f(-x) = \begin{cases} -x & \text{if } -0.5 \leq x < 0 \\ 0.5 & \text{if } -1 \leq x \leq -0.5 \end{cases}$$

The function $E(x)$ is periodic with a period of 2.

In [19]:
```
# The even periodic function E(x)
def even_periodic_function(x):
    # Adjusting x for the periodicity
    x = x % 2
    if x > 1:
        x = x - 2

    # Define E(x) based on the piecewise conditions
    if 0 <= x < 0.5:
        return x
    elif 0.5 <= x <= 1:
        return 0.5
    elif -0.5 <= x < 0:
        return -x
    elif -1 <= x <= -0.5:
        return 0.5
```

```python
    return 0

# x values for two periods
x_values = np.linspace(-2, 2, 1000)

# y values for E(x)
y_values = np.array([even_periodic_function(x) for x in x_values])

# Plotting the function
plt.figure(figsize=(10, 5))
plt.plot(x_values, y_values, label="Even Periodic Function E(x)")
plt.xlabel('x')
plt.ylabel('E(x)')
plt.title('Two Periods of the Even Periodic Function E(x)')
plt.axhline(0, color='black',linewidth=0.5)
plt.axvline(0, color='black',linewidth=0.5)
plt.grid(True)
plt.legend()
plt.show()
```



(e) Fourier Cosine Series Formula for Even Functions:

$$E(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi x}{L}\right)$$

$$a_0 = \frac{2}{L} \int_0^L E(x)\, dx$$

$$a_0 = \frac{2}{L}\left(\int_0^{0.5} x\,dx + \int_{0.5}^{1} 0.5\,dx\right)$$

$$a_n = \frac{2}{L}\int_0^L E(x)\cos\left(\frac{n\pi x}{L}\right)dx$$

$$a_n = \frac{2}{L}\left(\int_0^{0.5} x\cos\left(\frac{n\pi x}{L}\right)dx + \int_{0.5}^{1} 0.5\cos\left(\frac{n\pi x}{L}\right)dx\right)$$

In [21]:
```
# The function E(x) for the integration
Ex = Piecewise((x, (x >= 0) & (x < 0.5)), (0.5, (x >= 0.5) & (x <= 1)))

# Calculate a0 and an
a0 = (2/L) * integrate(Ex, (x, 0, L))
an = (2/L) * integrate(Ex * cos(n*pi*x/L), (x, 0, L))
```

In [22]:
```
# Simplify a0 and an with L = 1
L_value = 1
a0_simplified = a0.subs(L, L_value).simplify()
an_simplified = an.subs(L, L_value).simplify()

a0_simplified, an_simplified
```

Out[22]:
```
(0.750000000000000,
 Piecewise(((pi*n*sin(pi*n) + 2*cos(pi*n/2) - 2)/(pi**2*n**2), (n > 0) | (n < 0)),
(0.75, True)))
```

Simplified Coefficients:

$$a_0 = 0.75$$

$$a_n = \frac{\pi n\sin(\pi n) + 2\cos\left(\frac{\pi n}{2}\right) - 2}{\pi^2 n^2}$$

Thus, the Fourier cosine series for (E(x)) is given by:

$$E(x) \sim \frac{0.75}{2} + \sum_{n=1}^{\infty}\frac{\pi n\sin(\pi n) + 2\cos\left(\frac{\pi n}{2}\right) - 2}{\pi^2 n^2}\cos\left(\frac{n\pi x}{L}\right)$$

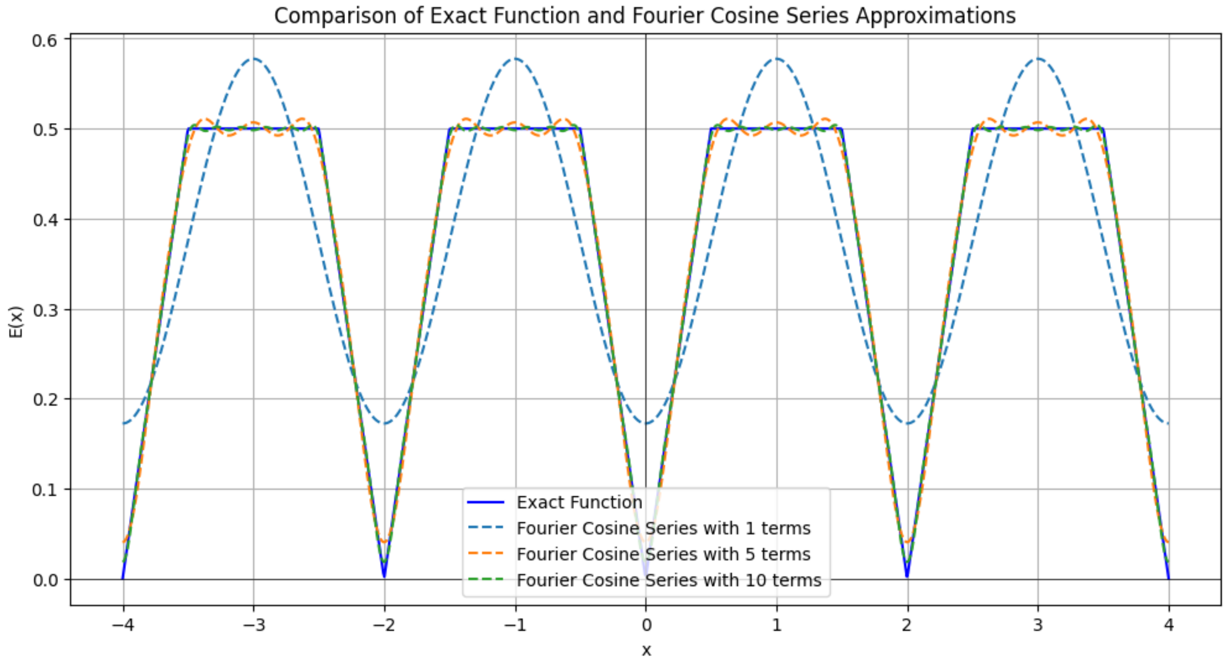
(f) The Fourier series for $E(x)$ is given by:

$$E(x) \sim \frac{0.75}{2} + \sum_{n=1}^{\infty}\frac{\pi n\sin(\pi n) + 2\cos\left(\frac{\pi n}{2}\right) - 2}{\pi^2 n^2}\cos\left(\frac{n\pi x}{L}\right)$$

where $L = 1$.

In [23]:

```
import numpy as np
import matplotlib.pyplot as plt

# Function to calculate the nth cosine coefficient an
def fourier_cosine_coefficient(n):
    if n == 0:
        return 0.75 / 2
    else:
        return (np.pi * n * np.sin(np.pi * n) + 2 * np.cos(np.pi * n / 2) - 2) / (np.p

# Function to calculate the Fourier cosine series approximation
def fourier_cosine_series(x, N, L=1):
    series = fourier_cosine_coefficient(0)
    for n in range(1, N + 1):
        an = fourier_cosine_coefficient(n)
        series += an * np.cos(n * np.pi * x / L)
    return series

# The exact function E(x)
def exact_function_e(x):
    x_mod = x % 2
    if x_mod > 1:
        x_mod = x_mod - 2

    if 0 <= x_mod < 0.5:
        return x_mod
    elif 0.5 <= x_mod <= 1:
        return 0.5
    elif -0.5 <= x_mod < 0:
        return -x_mod
    elif -1 <= x_mod <= -0.5:
        return 0.5

    return 0

# x values for the plot
x_values = np.linspace(-4, 4, 1000)  # Two periods

# Plotting
plt.figure(figsize=(12, 6))

# Plotting the exact function
plt.plot(x_values, [exact_function_e(x) for x in x_values], label="Exact Function", co

# Plotting Fourier series approximations with different numbers of terms
terms = [1, 5, 10]
for N in terms:
    plt.plot(x_values, fourier_cosine_series(x_values, N), label=f"Fourier Cosine Seri

plt.xlabel('x')
plt.ylabel('E(x)')
plt.title('Comparison of Exact Function and Fourier Cosine Series Approximations')
plt.axhline(0, color='black',linewidth=0.5)
plt.axvline(0, color='black',linewidth=0.5)
plt.grid(True)
plt.legend()
plt.show()
```



(g) The comparison between the exact function $E(x)$ and its Fourier cosine series approximations:

1. **Convergence to the Exact Function**:
   - As the number of terms in the Fourier series increases, the approximation becomes increasingly closer to the exact function.
   - Particularly near points of discontinuity (like at $x = 0.5$ and $x = 1$), the Fourier series requires more terms to accurately approximate the function.
2. **Smoothness and Continuity**:
   - The Fourier series approximations are smooth and continuous functions, even when the original function has discontinuities. This is a general property of Fourier series: they inherently produce smooth approximations.
3. **Periodicity**:
   - Both the original function $E(x)$ and its Fourier series are periodic with the same period.

Question 04

(a) The complex Fourier series for a function $f(t)$ with period T is given by:

$$f(t) \sim \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi n}{T}t}$$

where c_n are the Fourier coefficients given by:

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)e^{-i\frac{2\pi n}{T}t}\,dt$$

For $f(t) = |t|$ with period $T = 4$, the Fourier coefficients are:

$$c_n = \frac{1}{4}\int_{-2}^{2} |t|e^{-i\frac{\pi n}{2}t}\,dt$$

Since $f(t)$ is even, the integral of $|t|$ times the sine part of the exponential (which is odd) will be zero. Therefore, the integral simplifies to the integral of $|t|$ times the cosine part:

$$c_n = \frac{1}{4}\int_{-2}^{2} |t|\cos\left(\frac{\pi n}{2}t\right)dt$$

Calculating the coefficients c_n.

In [24]:
```
from sympy import symbols, integrate, cos, Abs

# Defining the symbols
t, n = symbols('t n')

# Defining the function
f = Abs(t)

# Period of the function
T = 4

# Calculating the Fourier coefficient cn
cn = (1/T) * integrate(f * cos(n * pi * t / 2), (t, -T/2, T/2))

cn.simplify()
```

Out[24]:

$$\begin{cases} \frac{2.0(\pi n\sin(\pi n)+\cos(\pi n)-1)}{\pi^2 n^2} & \text{for } (n > -\infty \vee n > 0) \wedge (n > -\infty \vee n < \infty) \wedge (n > 0 \vee n < 0) \wedge \\ 1.0 & \text{otherwise} \end{cases}$$

The Fourier coefficients c_n for the function $f(t) = |t|$ with a period of 4 are:

- For $n \neq 0$:

$$c_n = \frac{2\,(\pi n\sin(\pi n) + \cos(\pi n) - 1)}{\pi^2 n^2}$$

- For $n = 0$:

$$c_0 = 1$$

Therefore, the Fourier series for $f(t) = |t|$ in complex exponential form is given by:

$$f(t) \sim 1 + \sum_{n=-\infty, n\neq 0}^{\infty} \frac{2\,(\pi n\sin(\pi n) + \cos(\pi n) - 1)}{\pi^2 n^2} e^{i\frac{\pi n}{2}t}$$

(b) The complex Fourier series for a function $g(t)$ with period T is given by:

$$g(t) \sim \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi n}{T}t}$$

where c_n are the Fourier coefficients given by:

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} g(t)e^{-i\frac{2\pi n}{T}t}\,dt$$

For $g(t) = t$ with period $T = 4$, the Fourier coefficients are:

$$c_n = \frac{1}{4}\int_{-2}^{2} te^{-i\frac{\pi n}{2}t}\,dt$$

Since $g(t)$ is an odd function, the integral of t times the cosine part of the exponential (which is even) will be zero. Therefore, the integral simplifies to the integral of t times the sine part:

$$c_n = \frac{1}{4}\int_{-2}^{2} t\sin\left(\frac{\pi n}{2}t\right)\,dt$$

In [25]:
```
from sympy import sin

# Calculating the Fourier coefficient cn for g(t)
cn_g = (1/T) * integrate(t * sin(n * pi * t / 2), (t, -T/2, T/2))

cn_g.simplify()
```

Out[25]:

$$\begin{cases} \frac{2.0(-\pi n\cos(\pi n)+\sin(\pi n))}{\pi^2 n^2} & \text{for } (n > -\infty \vee n > 0) \wedge (n > -\infty \vee n < \infty) \wedge (n > 0 \vee n < 0) \wedge \\ 0 & \text{otherwise} \end{cases}$$

Calculating the coefficients c_n.

The Fourier coefficients c_n for the function $g(t) = t$ with a period of 4 are:

- For $n \neq 0$:

$$c_n = \frac{2\left(-\pi n\cos(\pi n) + \sin(\pi n)\right)}{\pi^2 n^2}$$

- For $n = 0$:

$$c_0 = 0$$

Therefore, the Fourier series for $g(t) = t$ in complex exponential form is given by:

$$g(t) \sim \sum_{n=-\infty, n\neq 0}^{\infty} \frac{2\left(-\pi n\cos(\pi n) + \sin(\pi n)\right)}{\pi^2 n^2} e^{i\frac{\pi n}{2}t}$$

(c) The complex Fourier series for a function $h(t)$ with period T is given by:

$$h(t) \sim \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi n}{T}t}$$

where c_n are the Fourier coefficients given by:

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} h(t)e^{-i\frac{2\pi n}{T}t}\,dt$$

For $h(t) = t$ with period $T = 2$, and given that $h(t)$ is defined on $0 \leq t \leq 2$, we need to adjust the function to reflect its periodicity. \ \ Calculating the Fourier coefficients c_n.

In [26]:
```
# h(t)
ht = Piecewise((t, (t >= 0) & (t <= 1)), (0, True))

T_h = 2
```

In [27]:
```
from sympy import exp, I

# Calculating the Fourier coefficient cn for h(t)
cn_h = (1/T_h) * integrate(ht * exp(-I * n * pi * t / 1), (t, -T_h/2, T_h/2))

cn_h.simplify()
```

Out[27]:

$$\begin{cases} \frac{0.5(i\pi n - e^{i\pi n} + 1)e^{-i\pi n}}{\pi^2 n^2} & \text{for } (n > -\infty \vee n > 0) \wedge (n > -\infty \vee n < \infty) \wedge (n > 0 \vee n < 0) \wedge (n \\ 0.25 & \text{otherwise} \end{cases}$$

The Fourier coefficients c_n for the function $h(t) = t$ with a period of 2 are:

- For $n \neq 0$:

$$c_n = \frac{0.5\left(i\pi n - e^{i\pi n} + 1\right)e^{-i\pi n}}{\pi^2 n^2}$$

- For $n = 0$:

$$c_0 = 0.25$$

Therefore, the Fourier series for $h(t) = t$ in complex exponential form is given by:

$$h(t) \sim 0.25 + \sum_{n=-\infty, n\neq 0}^{\infty} \frac{0.5\left(i\pi n - e^{i\pi n} + 1\right)e^{-i\pi n}}{\pi^2 n^2} e^{i\pi n t}$$