Proyecto Señales y Sistemas

Valentina Rozo Gonzalez
ID:20400745

1. Introducción

En este informe se encontrarán los coeficientes de la serie de Fourier correspondientes a un trozo de una señal no periódica y representar en tiempo la aproximación de esta señal mediante esta serie exponencial o trigonométrica

2. Señales

- Señal 1:

$$x(t): x^3, -1 \leq t \leq 1$$

a.

$$c_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)e^{\frac{-j2\pi kt}{T}}dt$$

$$c_k = \frac{1}{2}\int_{-\frac{2}{2}}^{\frac{2}{2}} x^3 e^{\frac{-j2\pi kt}{2}}dt$$

$$c_k = \frac{1}{2}\int_{-1}^{1} x^3 e^{-j\pi kt}dt$$

b. El siguiente código calcula los coeficientes de ambas formas de la Serie de Fourier y los imprime en la consola.

B. Forma de Serie Exponencial Compleja:

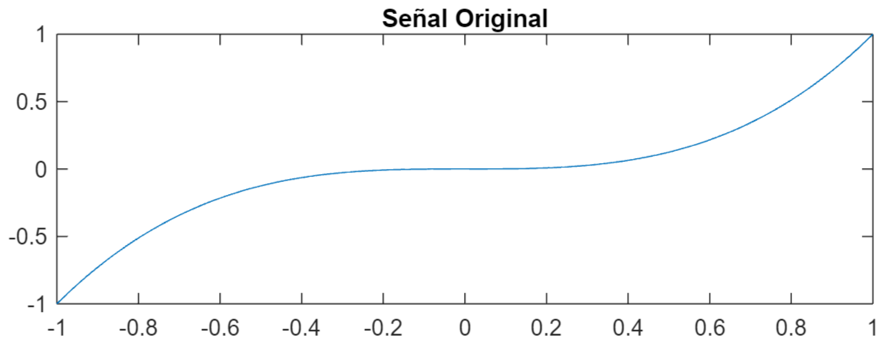
```
N = 20; % Número de coeficientes
T = 2; % Período

% Cálculo de coeficientes
ck = zeros(1, 2*N+1);

t = linspace(-1, 1, 1000); % Dominio de tiempo para la gráfica
x = @(t) t.^3; % Definición de la señal

for k = -N:N
    ck(k+N+1) = 1/T * integral(@(t) x(t) .* exp(-1i*2*pi*k*t/T), -T/2, T/2);
end

% Escribe los valores de los coeficientes en el informe
fprintf('Valores de los coeficientes de la Serie Exponencial Compleja:\n');
for k = -N:N
    fprintf('c_%d = %f\n', k, ck(k+N+1));
end
```

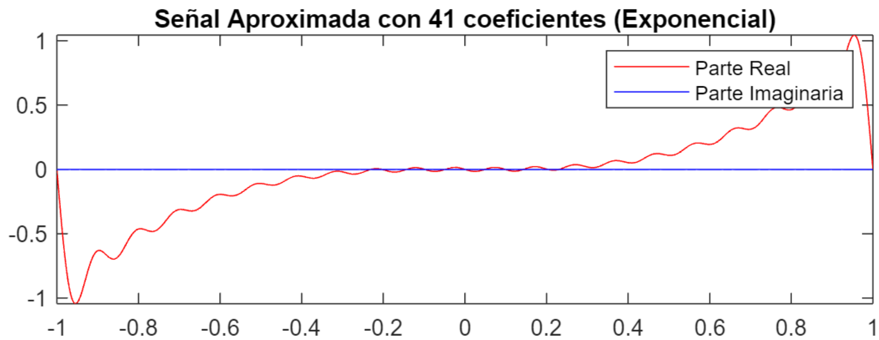
```
Valores de los coeficientes de la Serie Exponencial Compleja:
c_-20 = -0.000000
c_-19 = 0.000000
c_-18 = 0.000000
c_-17 = -0.000000
c_-16 = 0.000000
c_-15 = -0.000000
c_-14 = 0.000000
c_-13 = -0.000000
c_-12 = 0.000000
c_-11 = 0.000000
c_-10 = -0.000000
c_-9 = -0.000000
c_-8 = 0.000000
c_-7 = -0.000000
c_-6 = 0.000000
c_-5 = -0.000000
c_-4 = 0.000000
c_-3 = -0.000000
c_-2 = 0.000000
c_-1 = 0.000000
c_0 = -0.000000
c_1 = 0.000000
c_2 = 0.000000
c_3 = -0.000000
c_4 = 0.000000
c_5 = -0.000000
c_6 = 0.000000
c_7 = -0.000000
c_8 = 0.000000
c_9 = -0.000000
c_10 = -0.000000
c_11 = 0.000000
c_12 = 0.000000
c_13 = -0.000000
c_14 = 0.000000
c_15 = -0.000000
c_16 = 0.000000
c_17 = -0.000000
c_18 = 0.000000
c_19 = 0.000000
c_20 = -0.000000
```

```
Valores de los coeficientes de la Serie Trigonométrica:
a_0 = -0.000000     b_10 = -0.063275
b_0 = 0.000000      a_11 = 0.000000
a_1 = 0.000000      b_11 = 0.057584
b_1 = 0.249601      a_12 = 0.000000
a_2 = 0.000000      b_12 = -0.052828
b_2 = -0.269933     a_13 = -0.000000
a_3 = -0.000000     b_13 = 0.048795
b_3 = 0.197873      a_14 = 0.000000
a_4 = 0.000000      b_14 = -0.045332
b_4 = -0.153108     a_15 = -0.000000
a_5 = -0.000000     b_15 = 0.042327
b_5 = 0.124228      a_16 = -0.000000
a_6 = 0.000000      b_16 = -0.039694
b_6 = -0.104312     a_17 = -0.000000
a_7 = 0.000000      b_17 = 0.037369
b_7 = 0.089817      a_18 = -0.000000
a_8 = -0.000000     b_18 = -0.035301
b_8 = -0.078822     a_19 = -0.000000
a_9 = 0.000000      b_19 = 0.033450
b_9 = 0.070205      a_20 = 0.000000
a_10 = -0.000000    b_20 = -0.031783
```

c. El siguiente script solicitará al usuario el valor de N y el tipo de serie (Exponencial o Trigonométrica) que desea calcular. Luego, calculará la Serie de Fourier y mostrará la señal original y la señal aproximada en un gráfico. El usuario puede elegir cualquier valor de N sin limitaciones.

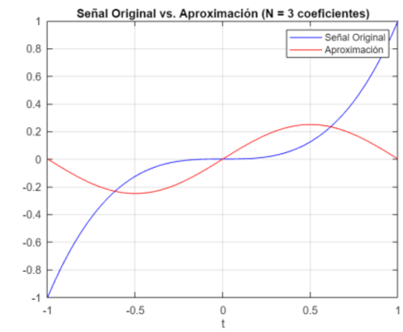
```matlab
% Solicitar al usuario el número de coeficientes N
N = input('Ingrese el número de coeficientes N: ');

% Verificar si el usuario desea Serie Exponencial Compleja o Serie Trigonométrica
serie_type = input('Elija el tipo de serie (Exponencial o Trigonométrica): ', 's');

T = 2; % Período
t = linspace(-1, 1, 1000); % Dominio de tiempo para la gráfica
x = @(t) t.^3; % Definición de la señal

if strcmpi(serie_type, 'Exponencial')
    % Cálculo de coeficientes Serie Exponencial Compleja
    ck = zeros(1, 2*N+1);
    for k = -N:N
        ck(k+N+1) = 1/T * integral(@(t) x(t) .* exp(-1i*2*pi*k*t/T), -T/2, T/2);
    end

    % Cálculo de la señal aproximada por Serie Exponencial Compleja
    x_approx = zeros(size(t));
    for k = -N:N
        x_approx = x_approx + ck(k+N+1) * exp(1i*2*pi*k*t/T);
    end

    % Gráficos
    figure;
    subplot(2, 1, 1);
    plot(t, x(t));
    title('Señal Original');
    subplot(2, 1, 2);
    plot(t, real(x_approx), 'r', t, imag(x_approx), 'b');
    title(['Señal Aproximada con ', num2str(2*N+1), ' coeficientes (Exponencial)']);
    legend('Parte Real', 'Parte Imaginaria');
else
    % Cálculo de coeficientes Serie Trigonométrica
    ak = zeros(1, N+1);
    bk = zeros(1, N+1);

    for k = 0:N
        ak(k+1) = (2/T) * integral(@(t) x(t) .* cos(2*pi*k*t/T), -T/2, T/2);
        bk(k+1) = (2/T) * integral(@(t) x(t) .* sin(2*pi*k*t/T), -T/2, T/2);
    end

    % Cálculo de la señal aproximada por Serie Trigonométrica
    x_approx = zeros(size(t));
    for k = 0:N
        x_approx = x_approx + ak(k+1) * cos(2*pi*k*t/T) + bk(k+1) * sin(2*pi*k*t/T);
    end
    % Gráficos
    figure;
    subplot(2, 1, 1);
    plot(t, x(t));
    title('Señal Original');
    subplot(2, 1, 2);
    plot(t, x_approx);
    title(['Señal Aproximada con ', num2str(N+1), ' coeficientes (Trigonométrica)']);
end
```

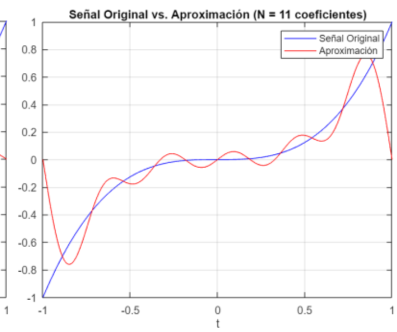


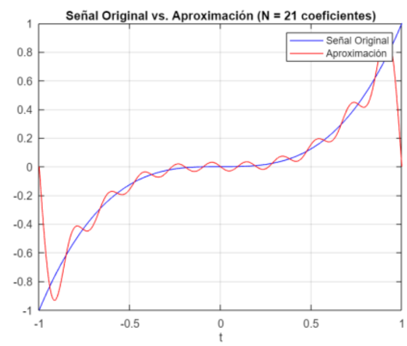


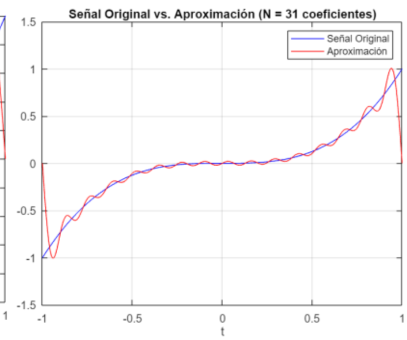
d. Este script generará gráficos superpuestos para diferentes valores de N (1, 5, 10, 15 y 20) de la Serie Exponencial Compleja. Puedes adaptar este script para la Serie Trigonométrica siguiendo un proceso similar. Después de ejecutar el script, tendrás imágenes que muestran la señal original y la aproximación de la Serie de Fourier para cada valor de N.

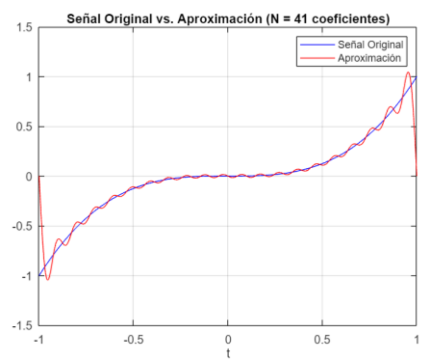
```
% Definir los valores de N que deseas analizar
N_values = [1, 5, 10, 15, 20];

T = 2; % Período
t = linspace(-1, 1, 1000); % Dominio de tiempo para la gráfica
x = @(t) t.^3; % Definición de la señal

for N = N_values
    % Cálculo de coeficientes Serie Exponencial Compleja
    ck = zeros(1, 2*N+1);
    for k = -N:N
        ck(k+N+1) = 1/T * integral(@(t) x(t) .* exp(-1i*2*pi*k*t/T), -T/2, T/2);
    end

    % Cálculo de la señal aproximada por Serie Exponencial Compleja
    x_approx = zeros(size(t));
    for k = -N:N
        x_approx = x_approx + ck(k+N+1) * exp(1i*2*pi*k*t/T);
    end

    % Gráficos
    figure;
    plot(t, x(t), 'b', t, real(x_approx), 'r');
    title(['Señal Original vs. Aproximación (N = ', num2str(2*N+1), ' coeficientes)']);
    xlabel('t');
    legend('Señal Original', 'Aproximación');
    grid on;
end
```











e. Este código calculará la potencia promedio de la señal original y la señal aproximada para varios valores de N y mostrará los resultados en la consola..

```
% Parámetros
T = 2; % Período
t = linspace(-1, 1, 1000); % Dominio de tiempo para la gráfica
x = @(t) t.^3; % Definición de la señal

% Valores de N a evaluar
N_values = [1, 5, 10, 15, 20];

for N = N_values
    % Cálculo de coeficientes Serie Trigonométrica
    ak = zeros(1, N+1);
    bk = zeros(1, N+1);

    for k = 0:N
        ak(k+1) = (2/T) * integral(@(t) x(t) .* cos(2*pi*k*t/T), -T/2, T/2);
        bk(k+1) = (2/T) * integral(@(t) x(t) .* sin(2*pi*k*t/T), -T/2, T/2);
    end

    % Cálculo de la señal aproximada por Serie Trigonométrica
    x_approx = zeros(size(t));
    for k = 0:N
        x_approx = x_approx + ak(k+1) * cos(2*pi*k*t/T) + bk(k+1) * sin(2*pi*k*t/T);
    end

    % Cálculo de la potencia promedio de la señal original
P_original = 1/T * integral(@(t) abs(x(t)).^2, -T/2, T/2);



    % Cálculo de la potencia promedio de la señal aproximada
    P_approx = 1/T * sum(ak.^2 + bk.^2);

    % Imprimir los resultados
    fprintf('N = %d:\n', N);
    fprintf('Potencia promedio de la señal original: %f\n', P_original);
    fprintf('Potencia promedio de la señal aproximada: %f\n', P_approx);
    fprintf('\n');
end
```

```
N = 1:
Potencia promedio de la señal original: 0.142857
Potencia promedio de la señal aproximada: 0.031150
N = 5:
Potencia promedio de la señal original: 0.142857
Potencia promedio de la señal aproximada: 0.106596
N = 10:
Potencia promedio de la señal original: 0.142857
Potencia promedio de la señal aproximada: 0.123643
N = 15:
Potencia promedio de la señal original: 0.142857
Potencia promedio de la señal aproximada: 0.129810
N = 20:
Potencia promedio de la señal original: 0.142857
Potencia promedio de la señal aproximada: 0.132984
```

- Señal 2:

$$x(t): x - x^3, -2 \leq t \leq 2$$

a.

$$c_k = \frac{1}{4}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{\frac{-j2\pi kt}{T}} dt$$

$$c_k = \frac{1}{4}\int_{-\frac{4}{2}}^{\frac{4}{2}} (x - x^3) e^{\frac{-j2\pi kt}{4}} dt$$

$$c_k = \frac{1}{4}\int_{-2}^{2} (x - x^3) e^{\frac{-j\pi kt}{2}} dt$$

b. El siguiente código calcula los coeficientes de ambas formas de la Serie de Fourier y los imprime en la consola.

Forma de Serie Exponencial Compleja:

```
N = 20; % Número de coeficientes
T = 4; % Período

% Cálculo de coeficientes
ck = zeros(1, 2*N+1);

t = linspace(-2, 2, 1000); % Dominio de tiempo para la gráfica
x = @(t) t - t.^3; % Definición de la señal

for k = -N:N
    ck(k+N+1) = 1/T * integral(@(t) (t - t.^3) .* exp(-1i*2*pi*k*t/T), -T/2, T/2);
end

% Escribe los valores de los coeficientes en el informe
fprintf('Valores de los coeficientes de la Serie Exponencial Compleja:\n');
for k = -N:N
    fprintf('c_%d = %f\n', k, ck(k+N+1));
end
```

```
Valores de los coeficientes de la Serie Exponencial Compleja:
c_-20 = 0.000000
c_-19 = -0.000000
c_-18 = -0.000000
c_-17 = 0.000000
c_-16 = -0.000000
c_-15 = 0.000000
c_-14 = -0.000000
c_-13 = 0.000000
c_-12 = -0.000000
c_-11 = -0.000000
c_-10 = 0.000000
c_-9 = 0.000000
c_-8 = -0.000000
c_-7 = 0.000000
c_-6 = -0.000000
c_-5 = 0.000000
c_-4 = -0.000000
c_-3 = 0.000000
c_-2 = -0.000000
c_-1 = 0.000000
c_0 = 0.000000
c_1 = 0.000000
c_2 = -0.000000
c_3 = 0.000000
c_4 = -0.000000
c_5 = 0.000000
c_6 = -0.000000
c_7 = 0.000000
c_8 = -0.000000
c_9 = 0.000000
c_10 = 0.000000
c_11 = -0.000000
c_12 = -0.000000
c_13 = 0.000000
c_14 = -0.000000
c_15 = 0.000000
c_16 = -0.000000
c_17 = 0.000000
c_18 = -0.000000
c_19 = -0.000000
c_20 = 0.000000
```

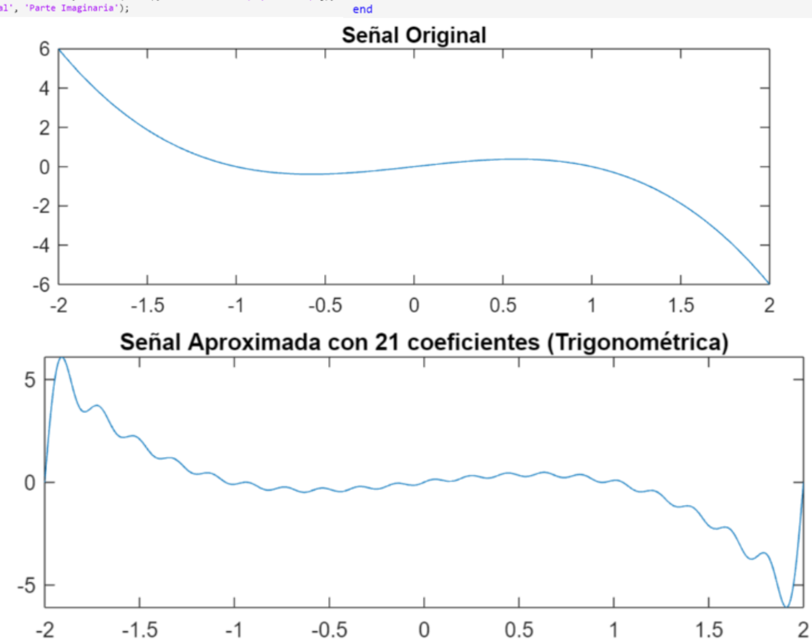
Forma de Serie Trigonométrica:

```
N = 20; % Número de coeficientes
T = 4; % Período
Z
% Cálculo de coeficientes
ak = zeros(1, N+1);
bk = zeros(1, N+1);

t = linspace(-2, 2, 1000); % Dominio de tiempo para la gráfica
x = @(t) t - t.^3; % Definición de la señal

for k = 0:N
    ak(k+1) = (2/T) * integral(@(t) (t - t.^3) .* cos(2*pi*k*t/T), -T/2, T/2);
    bk(k+1) = (2/T) * integral(@(t) (t - t.^3) .* sin(2*pi*k*t/T), -T/2, T/2);
end

% Escribe los valores de los coeficientes en el informe
fprintf('Valores de los coeficientes de la Serie Trigonométrica:\n');
for k = 0:N
    fprintf('a_%d = %f\n', k, ak(k+1));
    fprintf('b_%d = %f\n', k, bk(k+1));
end
```

```
Valores de los coeficientes de la Serie Trigonométrica:
a_0 = 0.000000
b_0 = 0.000000
a_1 = 0.000000
b_1 = -0.723571
a_2 = -0.000000
b_2 = 1.522841
a_3 = 0.000000
b_3 = -1.158567
a_4 = -0.000000
b_4 = 0.906552
a_5 = 0.000000
b_5 = -0.739175
a_6 = 0.000000
b_6 = 0.622286
a_7 = -0.000000
b_7 = -0.536647
a_8 = 0.000000
b_8 = 0.471418
a_9 = -0.000000
b_9 = -0.420166
a_10 = 0.000000
b_10 = 0.378876
a_11 = -0.000000
b_11 = -0.344921
a_12 = -0.000000
b_12 = 0.316518
a_13 = 0.000000
b_13 = -0.292415
a_14 = -0.000000
b_14 = 0.271709
a_15 = -0.000000
b_15 = -0.253731
a_16 = 0.000000
b_16 = 0.237977
a_17 = -0.000000
b_17 = -0.224059
a_18 = 0.000000
b_18 = 0.211676
a_19 = 0.000000
b_19 = -0.200586
a_20 = -0.000000
b_20 = 0.190599
```

c. El siguiente script solicitará al usuario el valor de N y el tipo de serie (Exponencial o Trigonométrica) que desea calcular. Luego, calculará la Serie de Fourier y mostrará la señal original y la señal aproximada en un gráfico. El usuario puede elegir cualquier valor de N sin limitaciones.

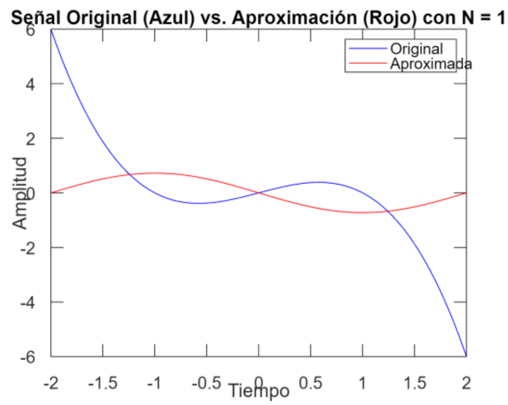
```
% Solicitar al usuario el número de coeficientes N
N = input('Ingrese el número de coeficientes N: ');

% Verificar si el usuario desea Serie Exponencial Compleja o Serie Trigonométrica
serie_type = input('Elija el tipo de serie (Exponencial o Trigonométrica): ', 's');

T = 4; % Período
t = linspace(-2, 2, 1000); % Dominio de tiempo para la gráfica
x = @(t) t - t.^3; % Definición de la señal

if strcmpi(serie_type, 'Exponencial')
    % Cálculo de coeficientes Serie Exponencial Compleja
    ck = zeros(1, 2*N+1);
    for k = -N:N
        ck(k+N+1) = 1/T * integral(@(t) (t - t.^3) .* exp(-1i*2*pi*k*t/T), -T/2, T/2);
    end

    % Cálculo de la señal aproximada por Serie Exponencial Compleja
    x_approx = zeros(size(t));
    for k = -N:N
        x_approx = x_approx + ck(k+N+1) * exp(1i*2*pi*k*t/T);
    end

    % Gráficos
    figure;
    subplot(2, 1, 1);
    plot(t, x(t));
    title('Señal Original');
    subplot(2, 1, 2);
    plot(t, real(x_approx), 'r', t, imag(x_approx), 'b');
    title(['Señal Aproximada con ', num2str(2*N+1), ' coeficientes (Exponencial)']);
    legend('Parte Real', 'Parte Imaginaria');
else
    % Cálculo de coeficientes Serie Trigonométrica
    ak = zeros(1, N+1);
    bk = zeros(1, N+1);

    for k = 0:N
        ak(k+1) = (2/T) * integral(@(t) (t - t.^3) .* cos(2*pi*k*t/T), -T/2, T/2);
        bk(k+1) = (2/T) * integral(@(t) (t - t.^3) .* sin(2*pi*k*t/T), -T/2, T/2);
    end

    % Cálculo de la señal aproximada por Serie Trigonométrica
    x_approx = zeros(size(t));
    for k = 0:N
        x_approx = x_approx + ak(k+1) * cos(2*pi*k*t/T) + bk(k+1) * sin(2*pi*k*t/T);
    end

    % Gráficos
    figure;
    subplot(2, 1, 1);
    plot(t, x(t));
    title('Señal Original');
    subplot(2, 1, 2);
    plot(t, x_approx);
    title(['Señal Aproximada con ', num2str(N+1), ' coeficientes (Trigonométrica)']);
end
```

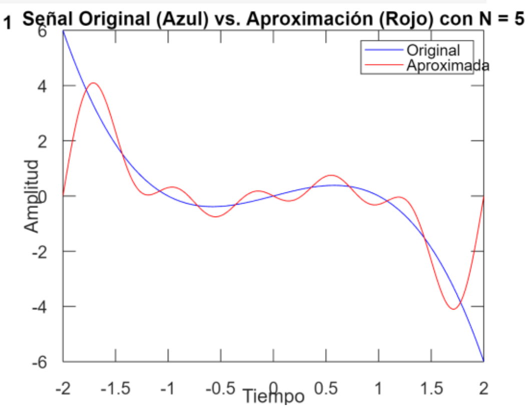


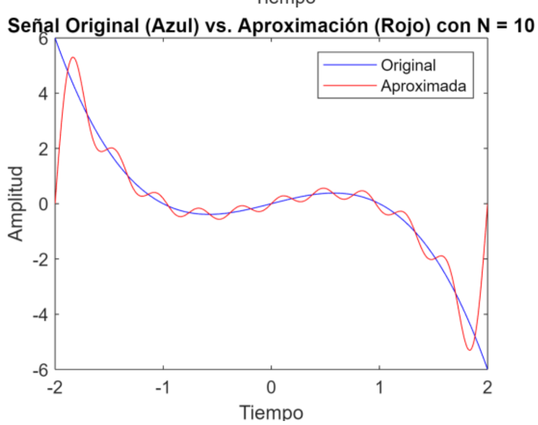
d. Este script generará gráficos superpuestos para diferentes valores de N (1, 5, 10, 15 y 20) de la Serie Exponencial Compleja. Puedes adaptar este script para la Serie Trigonométrica siguiendo un proceso similar. Después de ejecutar el script, tendrás imágenes que muestran la señal original y la aproximación de la Serie de Fourier para cada valor de N.

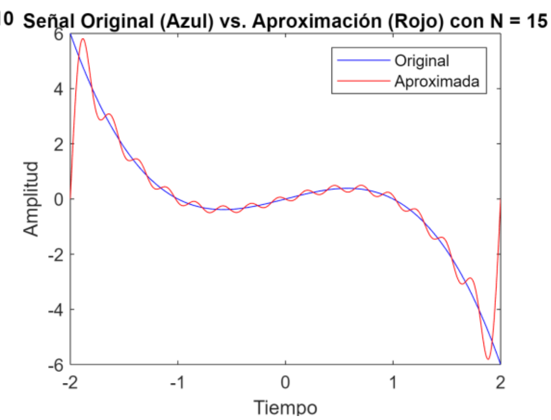
```
T = 4; % Período
t = linspace(-2, 2, 1000); % Dominio de tiempo para la gráfica
x = @(t) t - t.^3; % Definición de la señal

N_values = [1, 5, 10, 15, 20];

for N = N_values
    % Cálculo de coeficientes Serie Trigonométrica
    ak = zeros(1, N+1);
    bk = zeros(1, N+1);

    for k = 0:N
        ak(k+1) = (2/T) * integral(@(t) (t - t.^3) .* cos(2*pi*k*t/T), -T/2, T/2);
        bk(k+1) = (2/T) * integral(@(t) (t - t.^3) .* sin(2*pi*k*t/T), -T/2, T/2);
    end
```

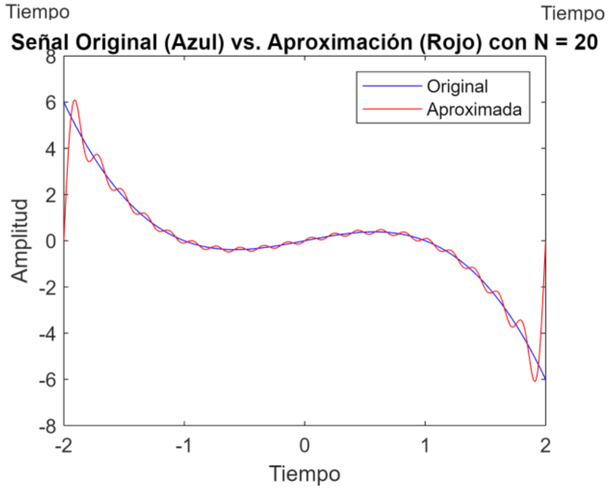
```
    % Cálculo de la señal aproximada por Serie Trigonométrica
    x_approx = zeros(size(t));
    for k = 0:N
        x_approx = x_approx + ak(k+1) * cos(2*pi*k*t/T) + bk(k+1) * sin(2*pi*k*t/T);
    end

    % Gráficos
    figure;
    plot(t, x(t), 'b', t, x_approx, 'r');
    title(['Señal Original (Azul) vs. Aproximación (Rojo) con N = ' num2str(N)]);
    xlabel('Tiempo');
    ylabel('Amplitud');
    legend('Original', 'Aproximada');
end
```











e. Este código calculará la potencia promedio de la señal original y la señal aproximada para varios valores de N y mostrará los resultados en la consola.

```
T = 4; % Período
t = linspace(-2, 2, 1000); % Dominio de tiempo para la gráfica
x = @(t) t - t.^3; % Definición de la señal

N_values = [1, 5, 10, 15, 20];
P_original = 1/T * integral(@(t) (x(t).^2), -T/2, T/2); % Potencia promedio de la señal original

for N = N_values
    % Cálculo de coeficientes Serie Trigonométrica
    ak = zeros(1, N+1);
    bk = zeros(1, N+1);

    for k = 0:N
        ak(k+1) = (2/T) * integral(@(t) (t - t.^3) .* cos(2*pi*k*t/T), -T/2, T/2);
        bk(k+1) = (2/T) * integral(@(t) (t - t.^3) .* sin(2*pi*k*t/T), -T/2, T/2);
    end

    % Cálculo de la potencia promedio de la señal aproximada
    P_approx = 0.5 * sum(ak.^2 + bk.^2); % Teorema de Parseval

    % Imprimir los resultados
    fprintf('N = %d:\n', N);
    fprintf('Potencia promedio de la señal original: %f\n', P_original);
    fprintf('Potencia promedio de la señal aproximada: %f\n', P_approx);
    fprintf('\n');
end
```

```
N = 1:

Potencia promedio de la señal original: 4.076190

Potencia promedio de la señal aproximada: 0.261778

N = 5:

Potencia promedio de la señal original: 4.076190

Potencia promedio de la señal aproximada: 2.776547

N = 10:

Potencia promedio de la señal original: 4.076190

Potencia promedio de la señal aproximada: 3.385323

N = 15:

Potencia promedio de la señal original: 4.076190

Potencia promedio de la señal aproximada: 3.606756

N = 20:

Potencia promedio de la señal original: 4.076190

Potencia promedio de la señal aproximada: 3.720858
```

3. Conclusiones

Serie de Fourier para $x(t)$: x^3:

1. La Serie de Fourier para esta señal se puede expresar en términos de coeficientes complejos utilizando la Serie Exponencial Compleja o en términos de coeficientes reales utilizando la Serie Trigonométrica.
2. El número de coeficientes (N) determina la calidad de la aproximación. A medida que N aumenta, la aproximación es más precisa.
3. El cálculo de los coeficientes y la generación de las gráficas permiten visualizar cómo la Serie de Fourier puede aproximar una señal no periódica mediante la suma de componentes sinusoidales.

Serie de Fourier para $x(t)$: $x - x^3$,:

1. Al igual que en el caso anterior, esta señal se puede expresar en términos de coeficientes complejos o reales, según se prefiera la Serie Exponencial Compleja o la Serie Trigonométrica.
2. El proceso es similar al del primer caso, con la diferencia de que se aplica a una señal diferente. Los coeficientes se calculan de la misma manera, pero los resultados serán diferentes debido a la diferencia en la forma de la señal original.
3. Al variar el valor de N en la aproximación, se puede observar cómo la Serie de Fourier se ajusta a la forma de la señal $x(t)$: $x - x^3$,, lo que permite comprender cómo diferentes frecuencias y componentes armónicos contribuyen a la aproximación.

En ambos casos, es importante destacar que la Serie de Fourier es una herramienta valiosa para descomponer señales periódicas y no periódicas en sus componentes fundamentales, lo que facilita el análisis y la representación de señales en el dominio de la frecuencia. Además, la elección entre la Serie Exponencial Compleja y la Serie Trigonométrica depende de la comodidad y preferencia del usuario, ya que ambos enfoques son equivalentes en términos de representación de la señal.