Table of Contents


```
clear; % clear variables
clc; % clear command window
close all
```

f(x) = x

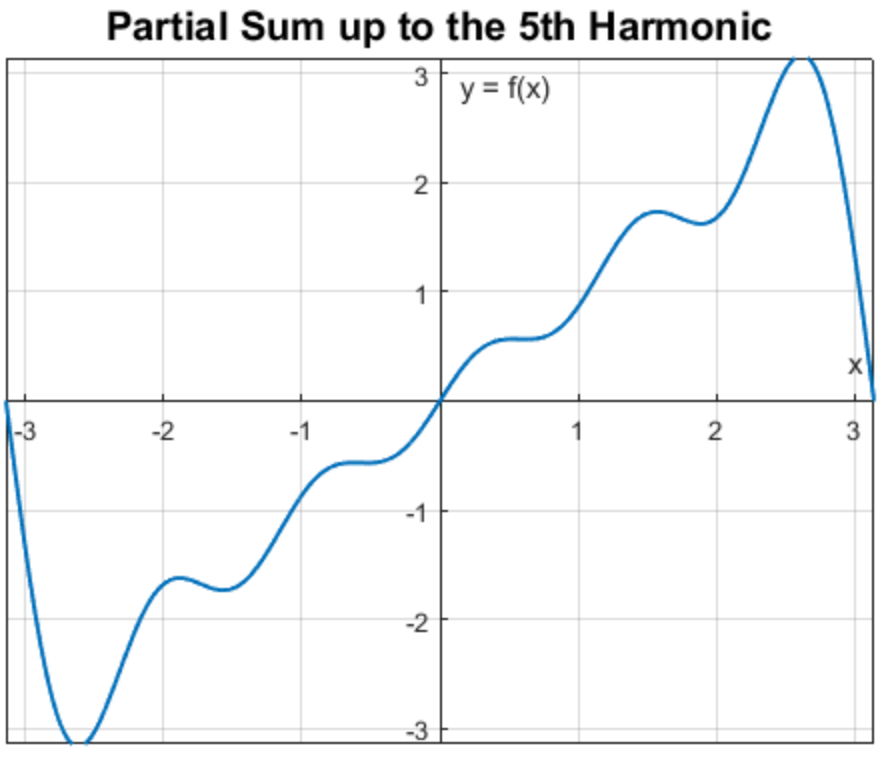
```
clear all; close all; clc;
%%%%%%%%%%%%%%%%%%%%%% variables to be changed by user %%%%%%%%%%%%%%%%%%%%%%
f = @(x) x; % define function f(x)
L = pi; % 1/2 of period, T = 2L
n_range = 0:5; % choose partial sum range
domain = [-pi,pi]; range = [-pi,pi]; % choose domain and range for plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% setup before plot %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define symbolic variables
syms x n
syms an(n) bn(n)
assume(n, {'integer','positive'}) % n is the group of natural numbers
% this simplifies integral computation: e.g. cos(n*pi) = (-1)^n
% find Fourier coefficients
% calculate a0
a0 = simplify( (1/(2*L)) * int( f(x) ,[-L,L]) );
% calculate an
an = simplify( (1/L) * int(f(x) * cos((n*pi)/L * x), x, [-L,L]) );
% calculate bn
bn = simplify( (1/L) * int(f(x) * sin((n*pi)/L * x), x, [-L,L]) );
% create the partial sum
f_fourier = @(k) a0 + symsum( an * cos((n*pi)/L * x) + bn * sin((n*pi)/L * x),
 n, 1,k);
%%%%%%%%%%%%%%%%%%% plot partial sums %%%%%%%%%%%%%%%%%%
% for loop to iterate plots
coefs = zeros(3,5);

for i = n_range
   % save coefficiants
   coefs(1,1) = double(subs(a0));
   if i > 0
       coefs(2,i) = double(subs(an, i));
       coefs(3,i) = double(subs(bn, i));
   end
   % plot partial sum
   fplot(f_fourier(i),domain,'LineWidth',1.5)
   % plot features
```

```
   ax = gca; % set axes to origin
   ax.XAxisLocation = 'origin';
   ax.YAxisLocation = 'origin';
   xlabel('x'); ylabel('y = f(x)'); % label axes
   xlim(domain); ylim(range) % domain and range
   title(sprintf('Partial Sum up to the %ith Harmonic',i),'FontSize',15) %
 title
   grid on
   pause(0.1) % delay between partial sum plots
end

fprintf('Coefficients for f(x) = x:\n');
fprintf("a0 = %.2f\n",coefs(1,1));
fprintf('an\t1\t2\t3\t4\t5\n\t');
fprintf("%.2f\t",coefs(2,:));
fprintf("\n")
fprintf('bn\t1\t2\t3\t4\t5\n\t');
fprintf("%.2f\t",coefs(3,:));
fprintf("\n");
```

```
Coefficients for f(x) = x:
a0 = 0.00
an 1 2 3 4 5
 0.00 0.00 0.00 0.00 0.00
bn 1 2 3 4 5
 2.00 -1.00 0.67 -0.50 0.40
```



f(x) = x + 1

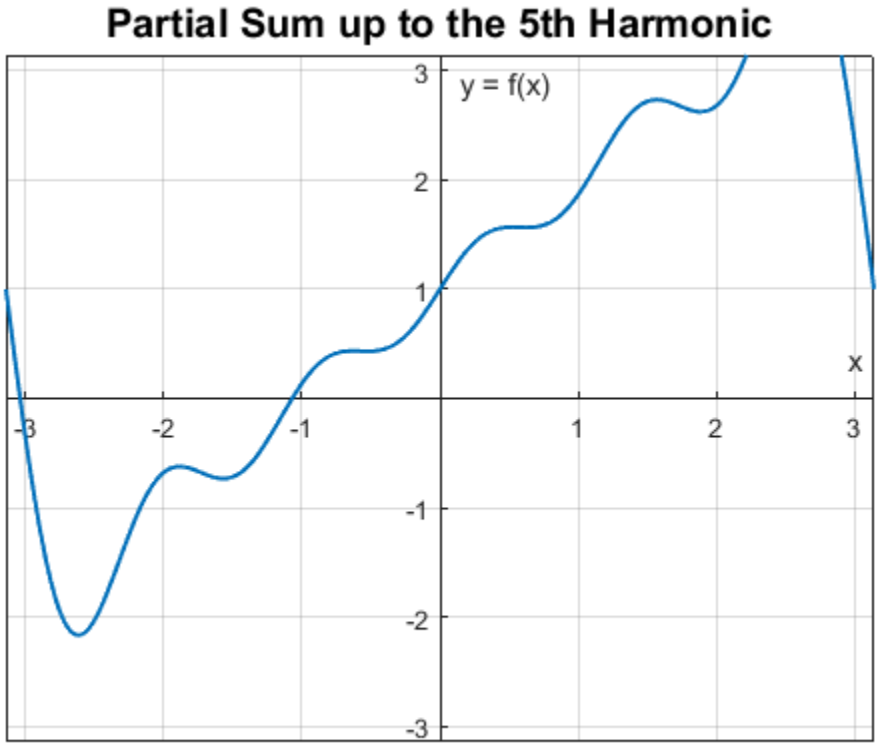
```
clear all; close all; clc;
%%%%%%%%%%%%%%%%%%%%%% variables to be changed by user %%%%%%%%%%%%%%%%%%%%%%
f = @(x) x + 1; % define function f(x)
L = pi; % 1/2 of period, T = 2L
n_range = 0:5; % choose partial sum range
domain = [-pi,pi]; range = [-pi,pi]; % choose domain and range for plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% setup before plot %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define symbolic variables
syms x n
syms an(n) bn(n)
assume(n, {'integer','positive'}) % n is the group of natural numbers
% this simplifies integral computation: e.g. cos(n*pi) = (-1)^n
% find Fourier coefficients
% calculate a0
a0 = simplify( (1/(2*L)) * int( f(x) ,[-L,L]) );
% calculate an
an = simplify( (1/L) * int(f(x) * cos((n*pi)/L * x), x, [-L,L]) );
% calculate bn
bn = simplify( (1/L) * int(f(x) * sin((n*pi)/L * x), x, [-L,L]) );
% create the partial sum
f_fourier = @(k) a0 + symsum( an * cos((n*pi)/L * x) + bn * sin((n*pi)/L * x),
 n, 1,k);
%%%%%%%%%%%%%%%%%%% plot partial sums %%%%%%%%%%%%%%%%%
% for loop to iterate plots
coefs = zeros(3,5);

for i = n_range
   % save coefficiants
   coefs(1,1) = double(subs(a0));
   if i > 0
       coefs(2,i) = double(subs(an, i));
       coefs(3,i) = double(subs(bn, i));
   end
   % plot partial sum
   fplot(f_fourier(i),domain,'LineWidth',1.5)
   % plot features
   ax = gca; % set axes to origin
   ax.XAxisLocation = 'origin';
   ax.YAxisLocation = 'origin';
   xlabel('x'); ylabel('y = f(x)'); % label axes
   xlim(domain); ylim(range) % domain and range
   title(sprintf('Partial Sum up to the %ith Harmonic',i),'FontSize',15) %
 title
   grid on
   pause(0.1) % delay between partial sum plots
end

fprintf('Coefficients for f(x) = x + 1:\n');
fprintf("a0 = %.2f\n",coefs(1,1));
fprintf('an\t1\t2\t3\t4\t5\n\t');
fprintf("%.2f\t",coefs(2,:));
```

```
fprintf("\n")
fprintf('bn\t1\t2\t3\t4\t5\n\t');
fprintf("%.2f\t",coefs(3,:));
fprintf("\n");
```

```
Coefficients for f(x) = x + 1:
a0 = 1.00
an 1 2 3 4 5
 0.00 0.00 0.00 0.00 0.00
bn 1 2 3 4 5
 2.00 -1.00 0.67 -0.50 0.40
```



f(x) = -x + 1

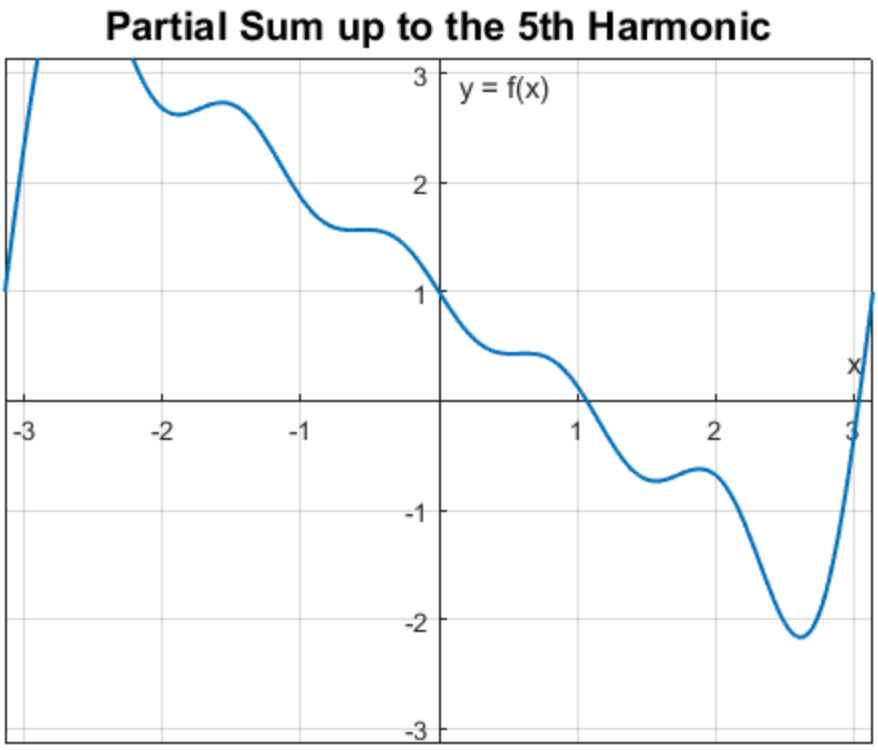
```
clear all; close all; clc;
%%%%%%%%%%%%%%%%%%%%%% variables to be changed by user %%%%%%%%%%%%%%%%%%%%%%
f = @(x) - x + 1; % define function f(x)
L = pi; % 1/2 of period, T = 2L
n_range = 0:5; % choose partial sum range
domain = [-pi,pi]; range = [-pi,pi]; % choose domain and range for plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% setup before plot %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define symbolic variables
syms x n
syms an(n) bn(n)
assume(n, {'integer','positive'}) % n is the group of natural numbers
% this simplifies integral computation: e.g. cos(n*pi) = (-1)^n
```

```
% find Fourier coefficients
% calculate a0
a0 = simplify( (1/(2*L)) * int( f(x) ,[-L,L]) );
% calculate an
an = simplify( (1/L) * int(f(x) * cos((n*pi)/L * x), x, [-L,L]) );
% calculate bn
bn = simplify( (1/L) * int(f(x) * sin((n*pi)/L * x), x, [-L,L]) );
% create the partial sum
f_fourier = @(k) a0 + symsum( an * cos((n*pi)/L * x) + bn * sin((n*pi)/L * x),
 n, 1,k);
%%%%%%%%%%%%%%%%%% plot partial sums %%%%%%%%%%%%%%%%%
% for loop to iterate plots
coefs = zeros(3,5);

for i = n_range
   % save coefficiants
   coefs(1,1) = double(subs(a0));
   if i > 0
       coefs(2,i) = double(subs(an, i));
       coefs(3,i) = double(subs(bn, i));
   end
   % plot partial sum
   fplot(f_fourier(i),domain,'LineWidth',1.5)
   % plot features
   ax = gca; % set axes to origin
   ax.XAxisLocation = 'origin';
   ax.YAxisLocation = 'origin';
   xlabel('x'); ylabel('y = f(x)'); % label axes
   xlim(domain); ylim(range) % domain and range
   title(sprintf('Partial Sum up to the %ith Harmonic',i),'FontSize',15) %
 title
   grid on
   pause(0.1) % delay between partial sum plots
end

fprintf('Coefficients for f(x) = -x + 1:\n');
fprintf("a0 = %.2f\n",coefs(1,1));
fprintf('an\t1\t2\t3\t4\t5\n\t');
fprintf("%.2f\t",coefs(2,:));
fprintf("\n")
fprintf('bn\t1\t2\t3\t4\t5\n\t');
fprintf("%.2f\t",coefs(3,:));
fprintf("\n");
```

```
Coefficients for f(x) = -x + 1:
a0 = 1.00
an 1 2 3 4 5
 0.00 0.00 0.00 0.00 0.00
bn 1 2 3 4 5
 -2.00 1.00 -0.67 0.50 -0.40
```



Published with MATLAB® R2023a