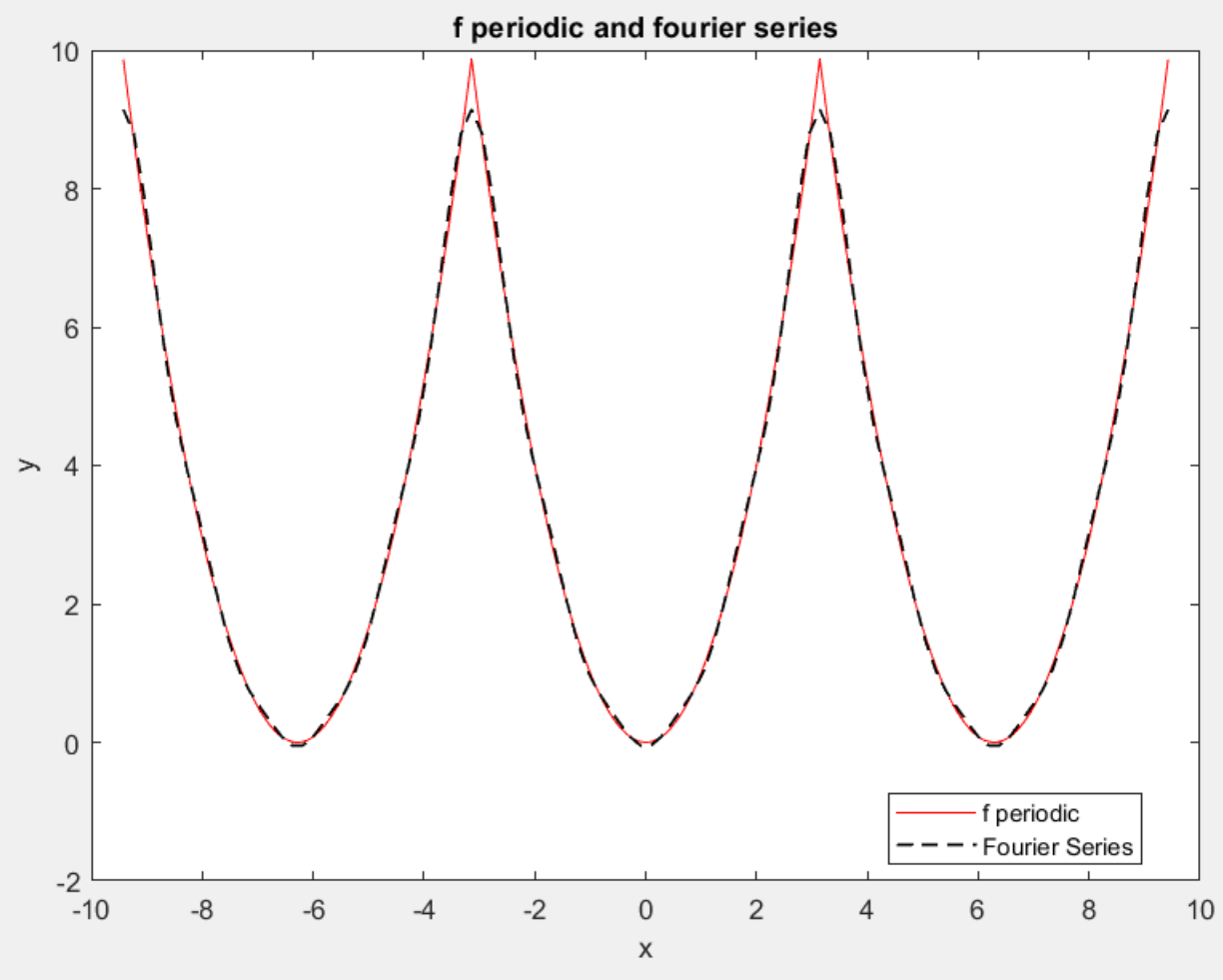
# Fourier Calculation Code

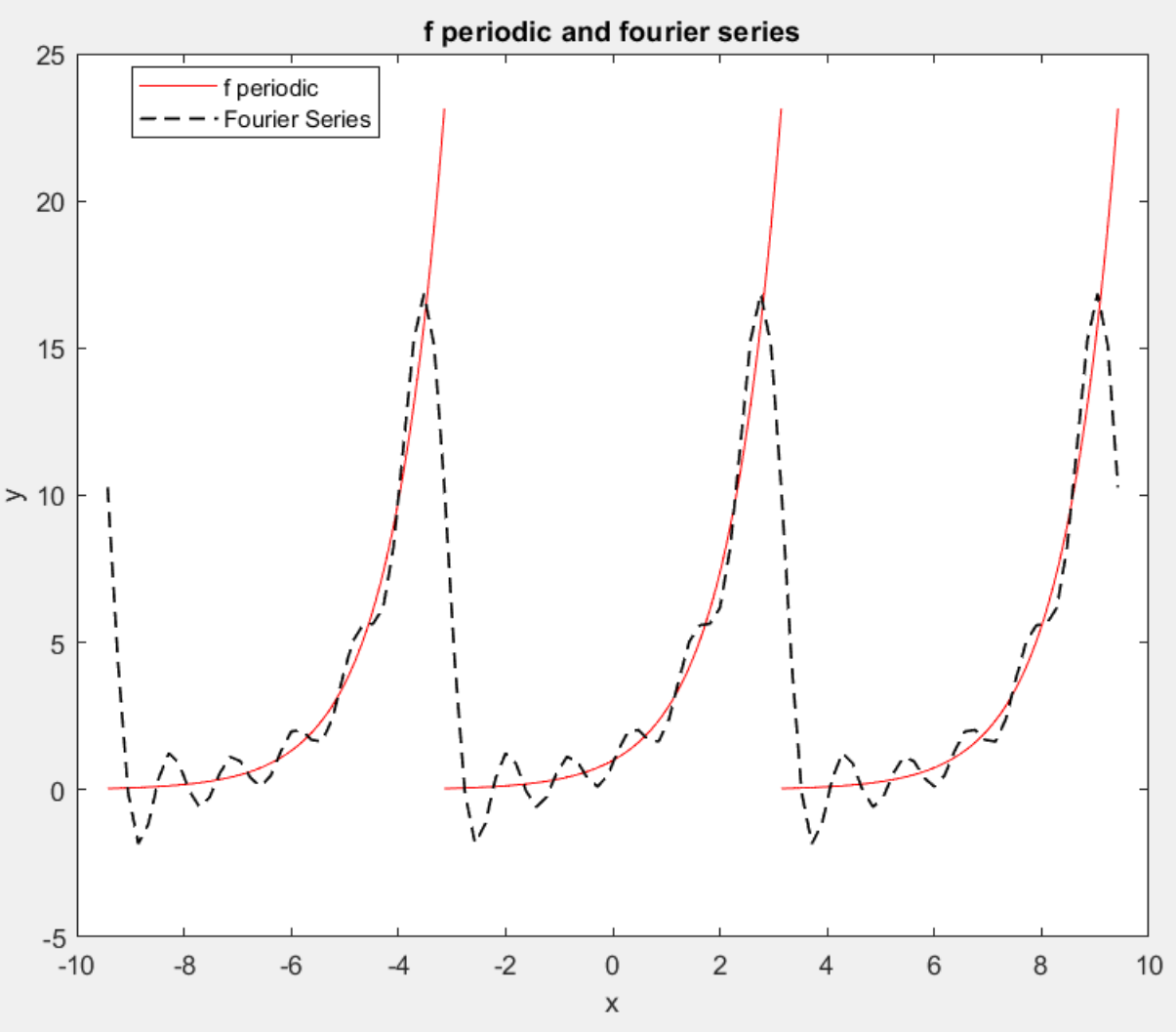
```
function[p, fs] = fourier_calculation(f, n) %inputs: f function_handle, n natural
number;
                                %outputs: p plot of f_periodic
                                %along with fourier series on [-3pi, 3pi]

    a_o = 1 / pi * integral(f, -pi, pi); %calculate a_o
    a_o_by_two = a_o / 2;
    a_k = cell(1, n); %initialize coeffs
    b_k = cell(1, n);
    sin_kx = cell(1, n); % sin_kx =(will be) b_k * sin(k * x)
    cos_kx = cell(1, n); % cos_kx =(will be) a_k * cos(k * x)
    syms y %intialize symbolic y
    for k = 1:n %for as many terms as the user wishes calculate the coeffs
        % a_k, b_k as well as the product a_k, b_k with cos(k*x), sin(k*x)
        a_k{k} = 1 / pi * integral(@(x) f(x) .* cos(k * x), -pi, pi);

        cos_kx{k}(y) = a_k{k} .* cos(k * y);
        b_k{k} = 1 / pi * integral(@(x) f(x) .* sin(k * x), -pi, pi);

        sin_kx{k}(y) = b_k{k} .* sin(k * y);

    end
fsum = sum([cos_kx{1, :}] + [sin_kx{1, :}]); %add summation terms
x_l = linspace(-3*pi, -pi); %intervals
x_m = linspace(-pi, pi);
x_r = linspace(pi, 3*pi);
x = linspace(-3*pi, 3*pi);
g = @(x) x + 2 * pi; %for shifting f(x)
h = @(x) x - 2 * pi; %to form f_periodic
f_l = @(x) f(g(x)); %compose with f
f_r = @(x) f(h(x));%to do this
fs = a_o_by_two + fsum; %finally form the nth partial fourier sum

figure %plot
 p = plot(x_l, f_l(x_l), 'red', ...
     x_m, f(x_m), 'red',...
     x_r, f_r(x_r), 'red', ...
     x, fs(x), 'black');
 p(4).LineStyle = '--';
 p(4).LineWidth = 1;
 legend('f periodic', '', '', 'Fourier Series')
 legend('Location', 'best')
 title('f periodic and fourier series')
end
```

# Test 1



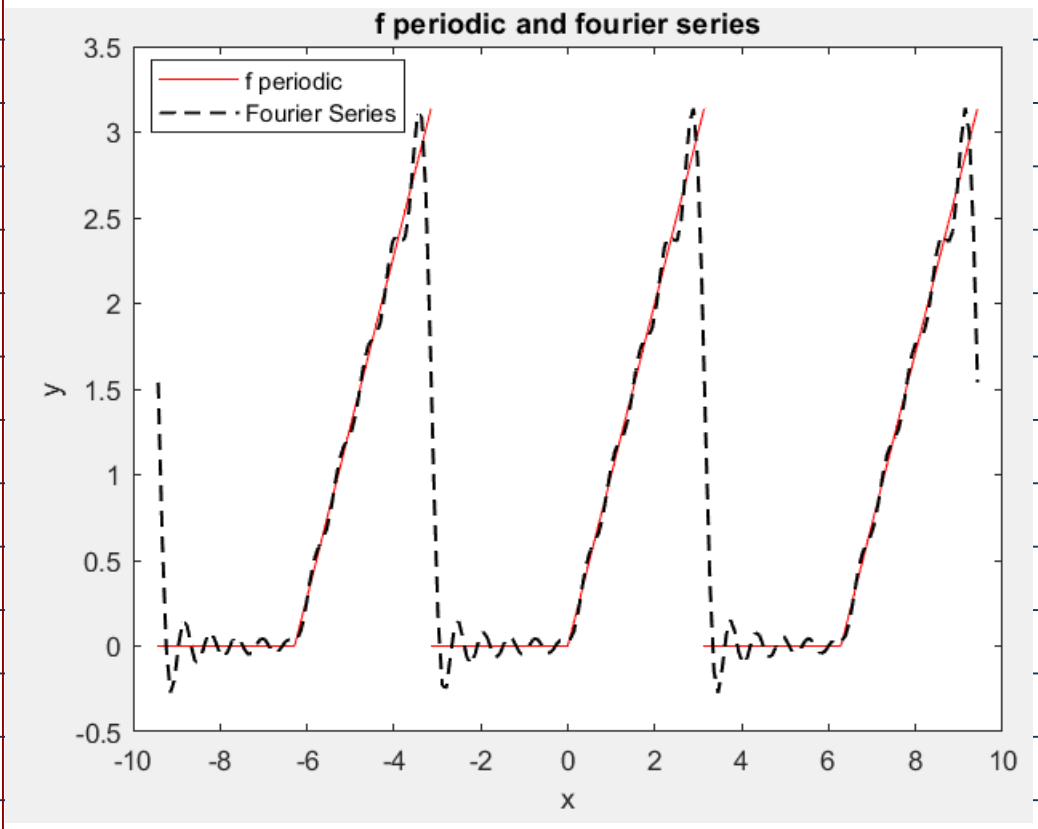

```
fourier_calculation(@(x) x.^2, 5)
```

# Test 2



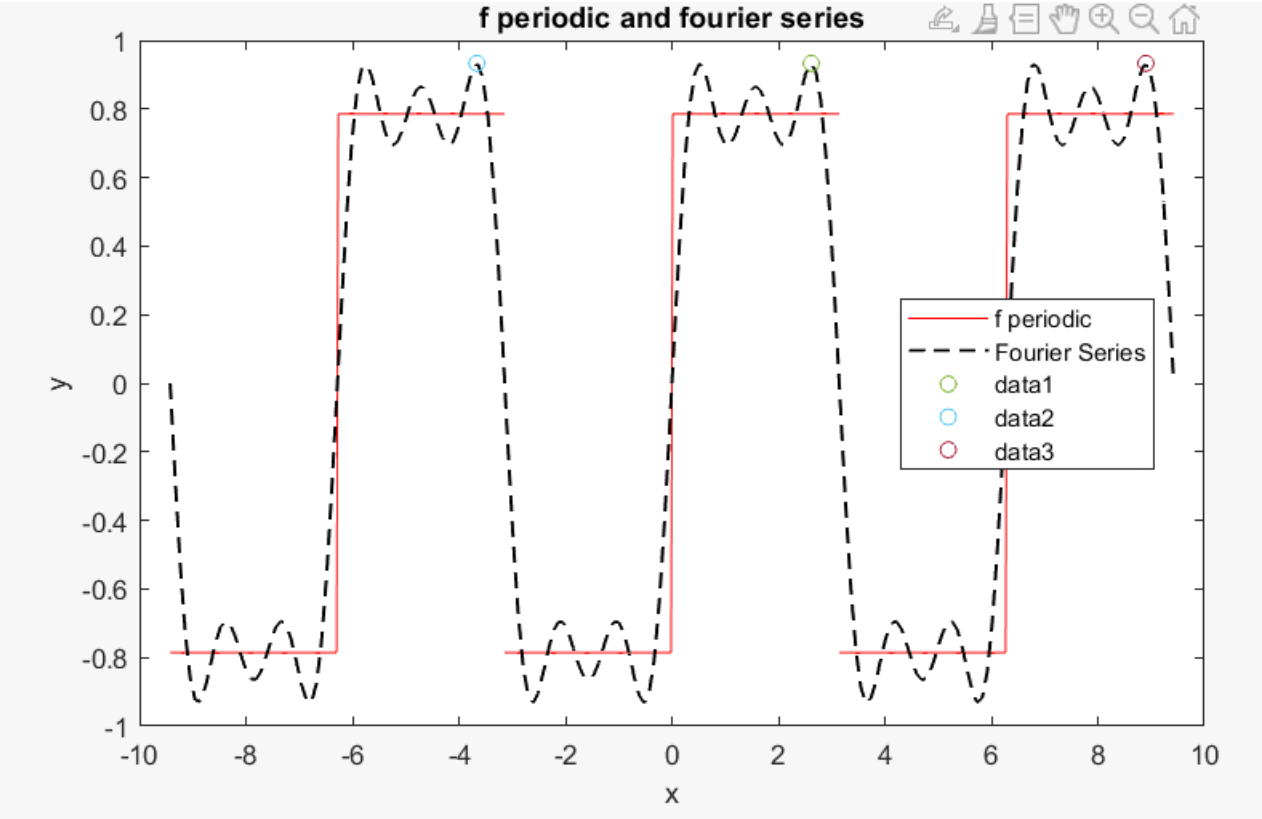

fourier_calculation(@(x) exp(x), 5)

# Test 3




```
f=@(x)x.*(x>0)

fourier_calculation(f, 10)
```

# Gibb's Phenomenon




```
function[t] = GibbsExploration(n)
f = @(x)pi/4.*(x>0)-pi/4.*(x<0); %square wave
h = @(x) x - 2 * pi; %shift 2*pi right
f_r = @(x) f(h(x)); %f is shifted
f_jump = abs(f(pi) - f_r(pi)); %mag of jump
if f_jump == 0 %there is not jump
    error("continuous function do not exhibit Gibb's phenomenon") %Gibbs doesnt
happen
end
jump_bd = 0.09 * f_jump; %jump times nine percent
x = linspace(pi / 2, 3 * pi /2, 2000); %intialize domain to find max on
[p, fs] = fourier_calculation(f, n); %calculate sequence of partial fourier sums(plot
output too)
[~, where] = max(fs(x)); %store max location
syms y;
mux = eval(subs(fs, y, x(where))) ;
overshoot = abs(mux - f(pi));  %magitude of overshoot,
   if overshoot >= jump_bd
      t = 1 %stores 1 in t when overshoot is greater than about 9% of 1/2 the jump
       p
       hold on
       plot(x(where), fs(x(where)), 'o', ...
           x(where) - 2*pi, fs(x(where)), 'o', ...
           x(where) + 2*pi, fs(x(where)), 'o') %plot max of FS
       hold off
      disp("Gibb's holds") %should always happen
   else
       disp("Gibb's fails-contradiction") %should never happen
       t = 0
   end
end
```

GibbsExploration(5)

```
Gibb's holds

ans =

     1
```