Jack Weissenberger Code for 9/15/17

Task 1:

d = linspace(-2\*pi, 2\*pi,10);

f = sin(d);

a = vander(d);

c = inv(a)\*f';

x = linspace(-2\*pi, 2\*pi,100);

s = sin(x);

p = polyval(c,x);

plot(x, s, 'r')

hold on

plot(x, p, 'b--')

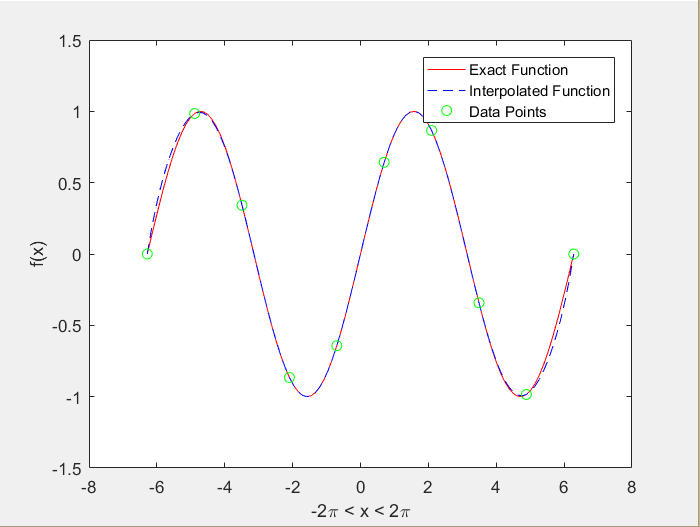
plot(d, f, 'go')

xlabel('-2\pi < x < 2\pi')

ylabel('f(x)')

legend('Exact Function','Interpolated Function', 'Data Points')

hold off



Task 2:

d = linspace(-4\*pi, 4\*pi,10);

f = sin(d);

a = vander(d);

c = inv(a)\*f';

x = linspace(-4\*pi, 4\*pi,100);

s = sin(x);

p = polyval(c,x);

plot(x, s, 'r')

hold on

plot(x, p, 'b--')

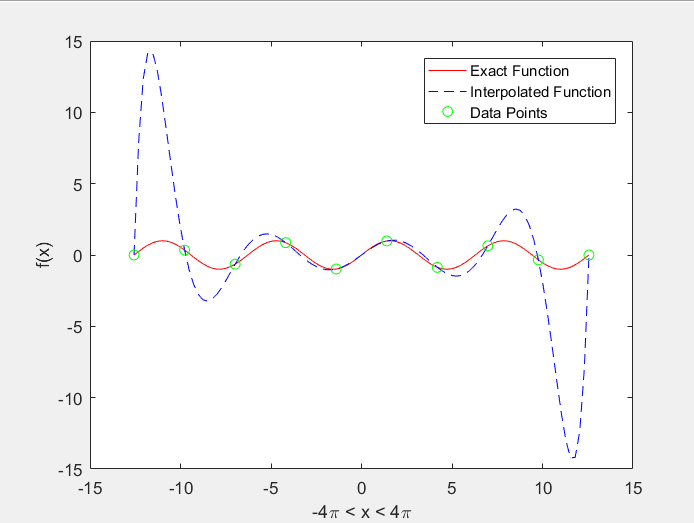
plot(d, f, 'go')

xlabel('-4\pi < x < 4\pi')

ylabel('f(x)')

legend('Exact Function','Interpolated Function', 'Data Points')

hold off



d = linspace(-8\*pi, 8\*pi,10);

f = sin(d);

a = vander(d);

c = inv(a)\*f';

x = linspace(-8\*pi, 8\*pi,100);

s = sin(x);

p = polyval(c,x);

plot(x, s, 'r')

hold on

plot(x, p, 'b--')

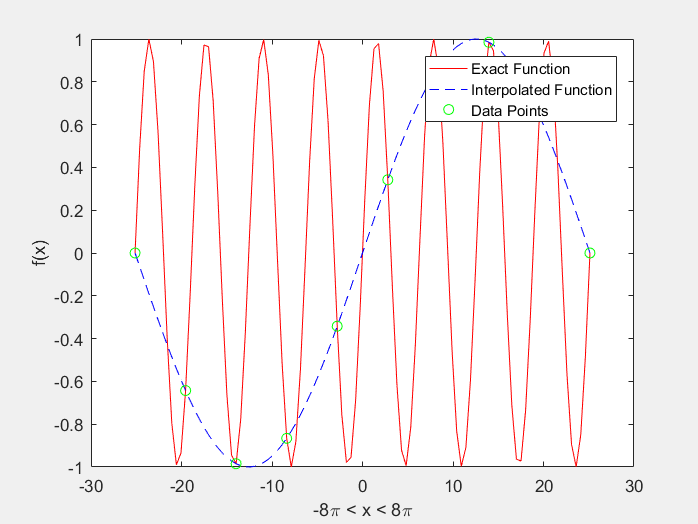
plot(d, f, 'go')

xlabel('-8\pi < x < 8\pi')

ylabel('f(x)')

legend('Exact Function','Interpolated Function', 'Data Points')

hold off



Task 3:

Here’s the function for cosine

function [ q ] = task3( x )

%Task3 Plots a function and an interpolating polynomial

% Input: x: a vector of twenty values

% Gets 20 values of x on 0-20 then calculates the values of the function

% for those x values and plots the interpolating polynomial through those

% points on the interval [?10, 10]. The points do

% not need to be uniformly spaced.

func = cos(x);

a = vander(x);

c = a\func';

l = linspace(-10, 10,100);

s = cos(l);

p = polyval(c,l);

plot(l, s, 'r')

hold on

plot(l, p, 'b--')

plot(x, func, 'go')

xlabel('-10 < x < 10')

ylabel('f(x)')

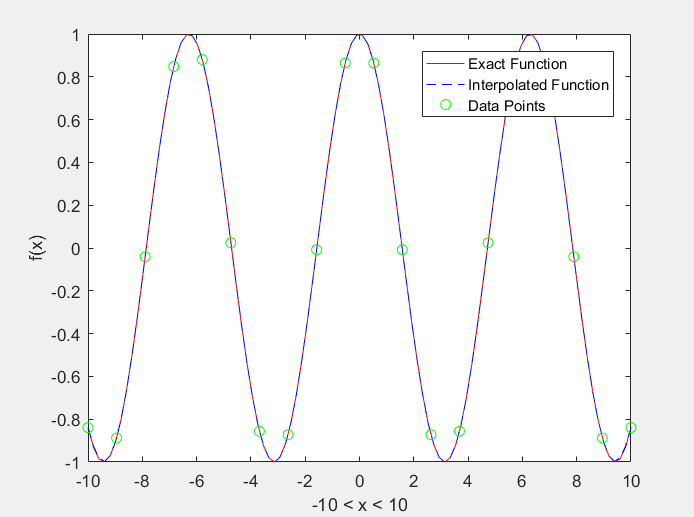
legend('Exact Function','Interpolated Function', 'Data Points')

hold off

q= 'The function has been plotted';

end

Input to the command line: task3(linspace(-10, 10, 20))



Task 3 with the tangent function

function [ q ] = task3( x )

%Task3 Plots a function and an interpolating polynomial

% Input: x: a vector of twenty values

% Gets 20 values of x on 0-20 then calculates the values of the function

% for those x values and plots the interpolating polynomial through those

% points on the interval [?10, 10]. The points do

% not need to be uniformly spaced.

func = tan(x);

a = vander(x);

c = a\func';

l = linspace(-10, 10,100);

s = tan(l);

p = polyval(c,l);

plot(l, s, 'r')

hold on

plot(l, p, 'b--')

plot(x, func, 'go')

xlabel('-10 < x < 10')

ylabel('f(x)')

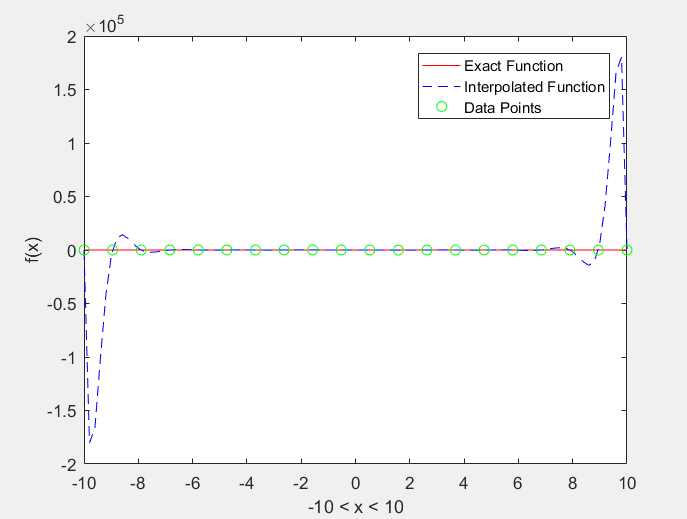
legend('Exact Function','Interpolated Function', 'Data Points')

hold off

q= 'The function has been plotted';

end

command line input: task3(linspace(-10,10,20))



Task 3 with x^6

function [ q ] = task3( x )

%Task3 Plots a function and an interpolating polynomial

% Input: x: a vector of twenty values

% Gets 20 values of x on 0-20 then calculates the values of the function

% for those x values and plots the interpolating polynomial through those

% points on the interval [?10, 10]. The points do

% not need to be uniformly spaced.

func = x.^6;

a = vander(x);

c = a\func';

l = linspace(-10, 10,100);

s = l.^6;

p = polyval(c,l);

plot(l, s, 'r')

hold on

plot(l, p, 'b--')

plot(x, func, 'go')

xlabel('-10 < x < 10')

ylabel('f(x)')

legend('Exact Function','Interpolated Function', 'Data Points')

hold off

q= 'The function has been plotted';

end

input to the command line task3(linspace(-10,10,20))

