Question 1.

The Vandermonde form:

Question 2.

a)

i. The newton interpolation is given below

function [y\_new,ai] = newton\_interpolation(x,y,x2)

ai = zeros(length(y),1);

n = length(y);

%Calculate the a\_i's

for k = 1:n %since matlab does not use 0 indexing

ai(k) = y(k);

for i = k+1:n

y(i) = (y(i)-y(k))/(x(i) - x(k));

end

end

ai(n) = y(n);

%evaluate y using the indexes

y\_new(1:length(x2)) = ai(n);

for j = 1:length(x2)

for m = n-1:-1:1

y\_new(j) = y\_new(j)\*(x2(j)-x(m))+ai(m);

end

end

end

ii. The spline interpolation is given below:

%Cubic spline method

function [S,z] = spline\_interpolation(t,y, x2)

n = length(t);

len\_k = length(x2);

% All starting indexes are shifted by 1 since there

% is no zero indexing in matlab

for i= 1:n-1

h(i) = t(i+1) - t(i);

b(i) = (y(i+1) - y(i))/h(i);

end

%Forward elimination

u(2) = 2\*(h(2) + h(1));

v(2) = 6\*(b(2) - b(1));

for i = 3:n-1

u(i) = 2\*(h(i-1) + h(i)) - ((h(i-1))^2) / u(i-1);

v(i) = 6\*(b(i) - b(i-1)) - (h(i-1)\*v(i-1))/u(i-1);

end

%Back substitution

z(n) = 0;

for i = n-1:-1:2

z(i) = (v(i) - h(i)\*z(i+1))/u(i);

end

z(1) = 0;

for k = 1:length(x2)

for i = 1:n-1

if((x2(k) - t(i+1)) <= 0 )

break;

end

end

h2 = t(i+1) - t(i);

B = -h2\*z(i+1)/6 - h2\*z(i)/3 + (y(i+1) - y(i))/h2;

D = (1/(6\*h2)) \* (z(i+1) - z(i));

S(k) = y(i) + (x2(k) - t(i))\*(B+ (x2(k)-t(i))\*(z(i)/2 + (x2(k) - t(i))\*D));

end

end

iii. The least squares method is given below

function [c,y2] = leastsquares(x,y,x2)

n = length(x);

A = ones(n,3);

%Allocate A

A(1:n,2) = x.^2;

A(1:n,3) = x.^4;

%Find the c's

c = A\y;

%Calculate the y interpolation

y2 = c(1) + c(2) \*x2.^2 + c(3)\*x2.^4;

end

The functions are evaluated using A5.m given below

%This calls and plots the newton\_interpolation,spline\_interpolation and

%leastsquare.m functions

% Array of knots

knots = -1:(2/6):1;

knots = knots';

y = f(knots);

x2 = -1:0.01:1;

y2 = f(x2);

figure(1);

hold on;

plot(x2,y2);

[y\_newt,ai] = newton\_interpolation(knots,y,x2);

plot(x2,y\_newt,'-g');

[S,z] = spline\_interpolation(knots,y,x2);

plot(x2,S,'-r');

[c,gx] = leastsquares(knots,y,x2);

plot(x2,gx,'-c');

xlabel('x');

ylabel('y');

[h,~]=legend('f(x)','p(x)','S(x)','g(x)');

%Print the results

fprintf('Coefficients of p(x) \n');

fprintf(' %10.5f \n',ai);

fprintf('Coefficients of S(x) \n');

fprintf(' %10.5f \n',z);

fprintf('Coefficients of g(x) \n');

fprintf('%10.5f \n',c);

%New nodes

knots2 = -1:(2/12):1;

knots2 = knots2';

y2\_p = f(knots2);

fprintf('Num.\t x \t f(x) \t f(x) - p(x) f(x) - S(x) f(x) - g(x) \n');

for i = 1:13

fx = f(knots2(i));

[px,a2] = newton\_interpolation(knots,y,knots2(i));

px = fx - px;

[Sx,z2] = spline\_interpolation(knots,y,knots2(i));

Sx = fx - Sx;

[c2,gx] = leastsquares(knots,y,knots2(i));

gx = fx - gx;

fprintf('%d\t%10.5f\t %10.5f\t %10.5f\t %10.5f\t %10.5f\t \n',i,knots2(i),fx,px,Sx,gx);

end

The results are below

Coefficients of p(x)

0.03846

0.13232

0.62113

1.86807

-8.23119

13.13489

-13.13489

Coefficients of S(x)

0.00000

-1.81814

14.72616

-27.21602

14.72616

-1.81814

0.00000

Coefficients of g(x)

0.71921

-2.38618

1.71948

Num. x f(x) f(x) - p(x) f(x) - S(x) f(x) - g(x)

1 -1.00000 0.03846 0.00000 0.00000 -0.01404

2 -0.83333 0.05446 -0.55342 -0.01868 0.16310

3 -0.66667 0.08257 -0.00000 -0.00000 0.08424

4 -0.50000 0.13793 0.22935 0.05393 -0.09220

5 -0.33333 0.26471 0.00000 0.00000 -0.21060

6 -0.16667 0.59016 -0.18172 -0.12892 -0.06409

7 0.00000 1.00000 0.00000 0.00000 0.28079

8 0.16667 0.59016 -0.18172 -0.12892 -0.06409

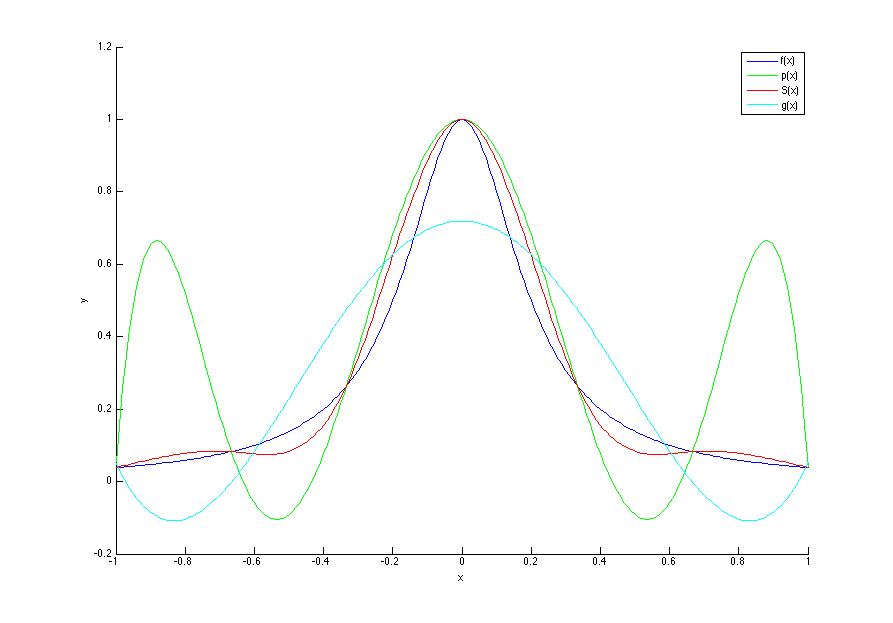
9 0.33333 0.26471 -0.00000 -0.00000 -0.21060

10 0.50000 0.13793 0.22935 0.05393 -0.09220

11 0.66667 0.08257 -0.00000 -0.00000 0.08424

12 0.83333 0.05446 -0.55342 -0.01868 0.16310

13 1.00000 0.03846 0.00000 0.00000 -0.01404



Part b)

%This calls and plots the newton\_interpolation,spline\_interpolation and

%leastsquare.m functions

% with chebysev knots

% Array of knots

knots=zeros(7,1);

for i = 1:7

knots(i) = cos(pi\*(2\*(i-1) + 1)/(14));

end

%Convert to ascending order of knots

knots=flipud(knots);

y = f(knots);

x2 = -1:0.01:1;

y2 = f(x2);

%Plot the results

figure(1);

hold on;

plot(x2,y2);

[y\_newt,ai] = newton\_interpolation(knots,y,x2);

plot(x2,y\_newt,'-g');

[S,z] = spline\_interpolation(knots,y,x2);

plot(x2,S,'-r');

[c,gx] = leastsquares(knots,y,x2);

plot(x2,gx,'-c');

xlabel('x');

ylabel('y');

[h,~]=legend('f(x)','p(x)','S(x)','g(x)');

%Print the results

fprintf('Coefficients of p(x) \n');

fprintf(' %10.5f \n',ai);

fprintf('Coefficients of S(x) \n');

fprintf(' %10.5f \n',z);

fprintf('Coefficients of g(x) \n');

fprintf('%10.5f \n',c);

%New nodes

knots2 = -1:(2/12):1;

knots2 = knots2';

y2\_p = f(knots2);

fprintf('Num.\t x \t f(x) \t f(x) - p(x) f(x) - S(x) f(x) - g(x) \n');

for i = 1:13

fx = f(knots2(i));

[px,a2] = newton\_interpolation(knots,y,knots2(i));

px = fx - px;

[Sx,z2] = spline\_interpolation(knots,y,knots2(i));

Sx = fx - Sx;

[c2,gx] = leastsquares(knots,y,knots2(i));

gx = fx - gx;

fprintf('%d\t%10.5f\t %10.5f\t %10.5f\t %10.5f\t %10.5f\t \n',i,knots2(i),fx,px,Sx,gx);

end

Coefficients of p(x)

0.04038

0.10894

0.40328

1.65101

-4.90515

6.62140

-6.79168

Coefficients of S(x)

0.00000

-2.62433

11.92394

-19.10516

11.92394

-2.62433

0.00000

Coefficients of g(x)

0.78776

-2.60868

1.96959

Num. x f(x) f(x) - p(x) f(x) - S(x) f(x) - g(x)

1 -1.00000 0.03846 0.10204 0.00289 -0.11020

2 -0.83333 0.05446 -0.06923 -0.00687 0.12845

3 -0.66667 0.08257 0.13475 0.03482 0.06517

4 -0.50000 0.13793 0.09148 0.04164 -0.12076

5 -0.33333 0.26471 -0.16173 -0.11535 -0.25752

6 -0.16667 0.59016 -0.24049 -0.19967 -0.12665

7 0.00000 1.00000 0.00000 -0.00000 0.21224

8 0.16667 0.59016 -0.24049 -0.19967 -0.12665

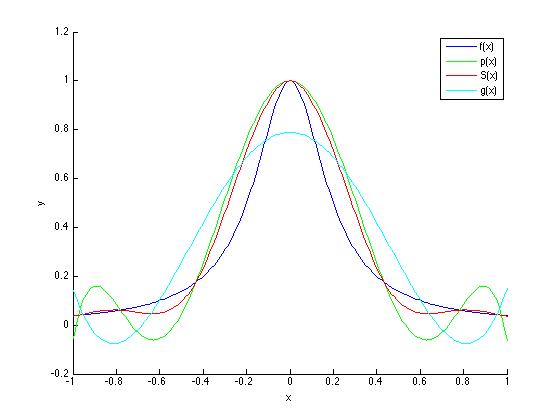
9 0.33333 0.26471 -0.16173 -0.11535 -0.25752

10 0.50000 0.13793 0.09148 0.04164 -0.12076

11 0.66667 0.08257 0.13475 0.03482 0.06517

12 0.83333 0.05446 -0.06923 -0.00687 0.12845

13 1.00000 0.03846 0.10204 0.00289 -0.11020



(c ) From the plot, the Chebyshev nodes converge closer to the f(x) and also have less swings in their data points starting out. The p(x) plot has much larger deviations from f(x) starting out using the chebyshev knots than using the regularly spaced ones in particular.