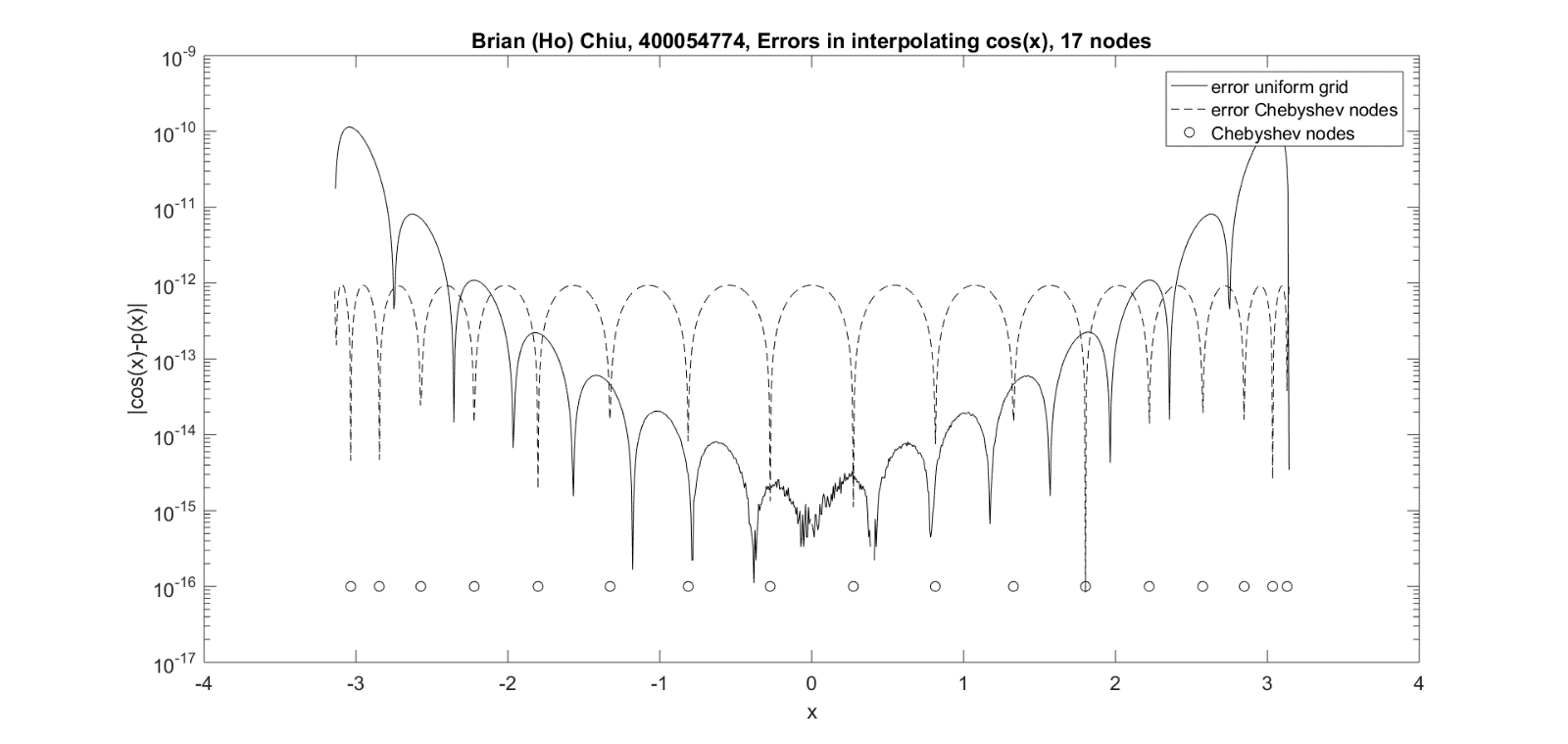
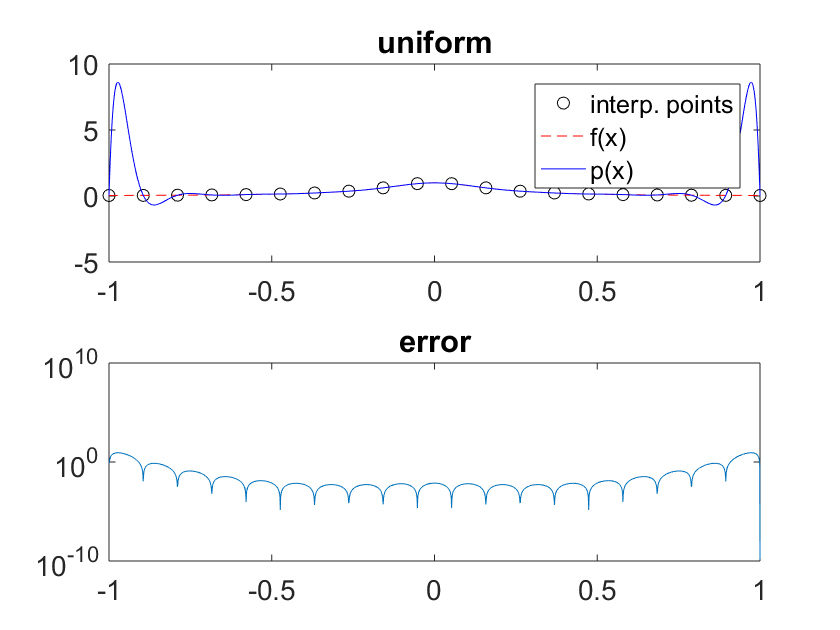
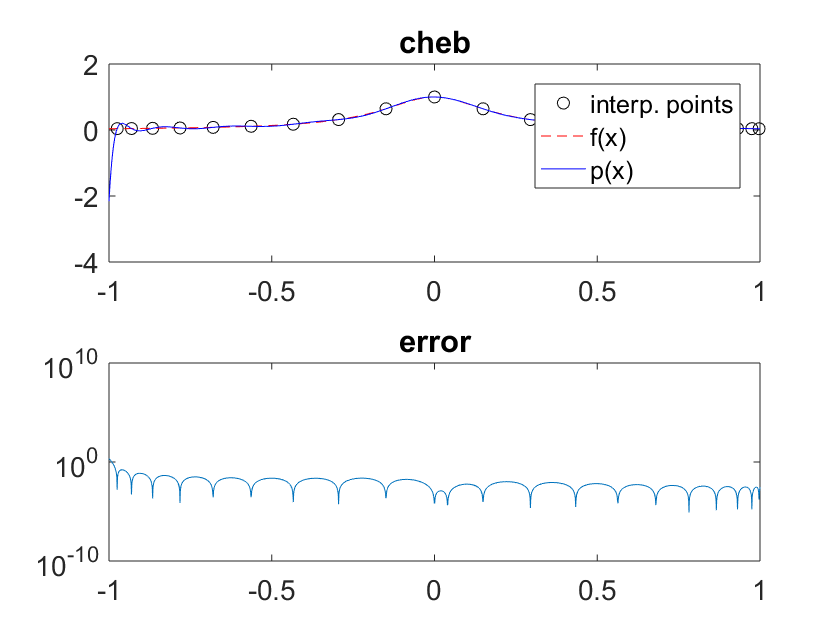
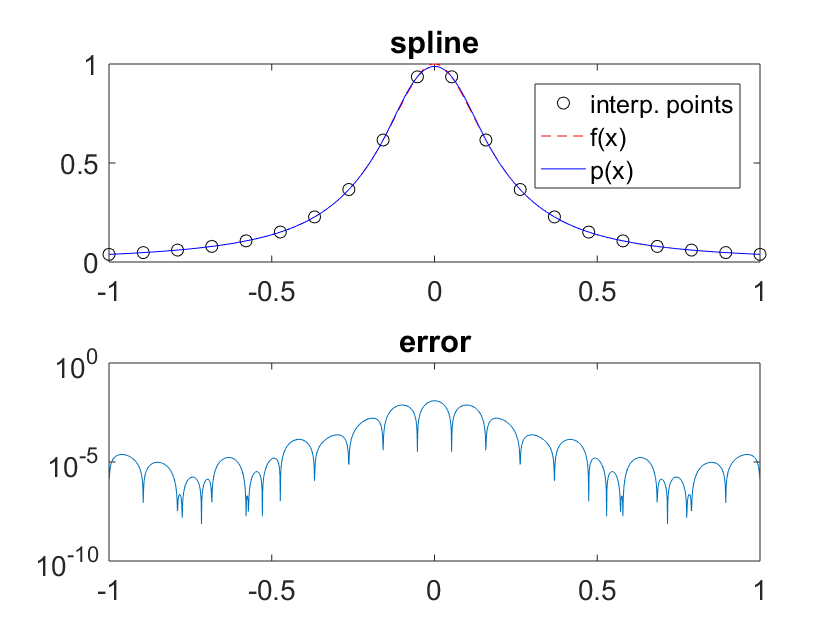
4X03 plots and code chiuh1

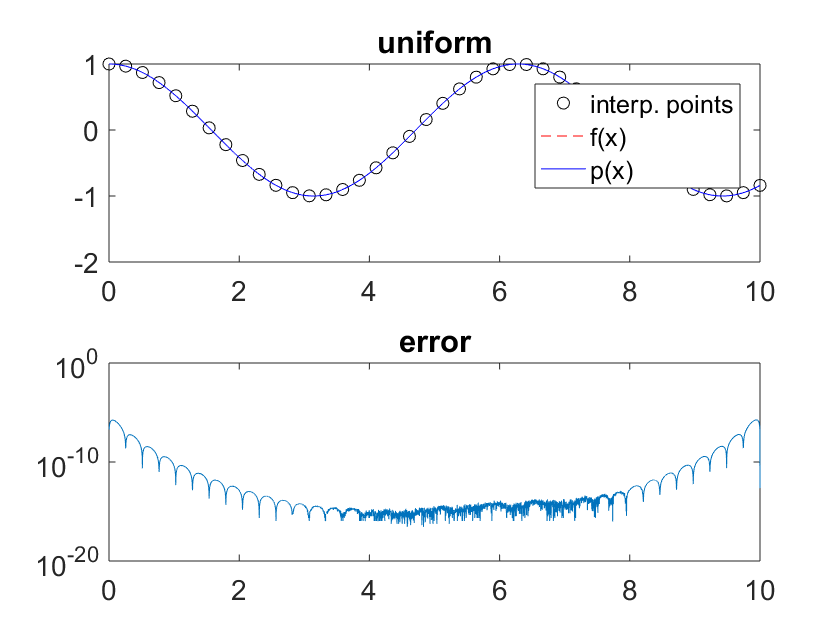
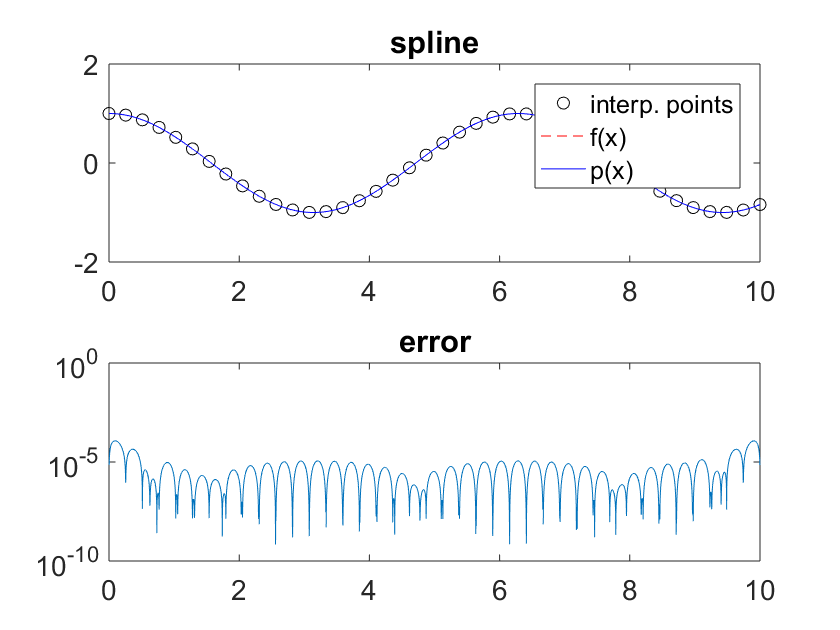
400054774

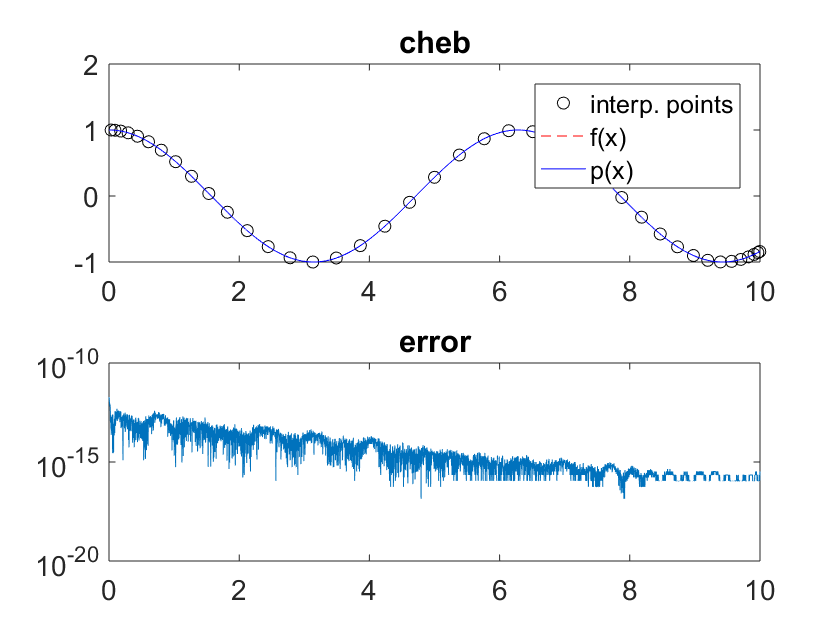
Question 6

Question 7

Plots with given function



Plots with test values, a = 0, b= 10, n = 40, f = cos(x)



**newtonint.m code:**

function c = newtonint(x,y)

n = length(x);

c = zeros(1,n);

c(1) = y(1); %%initial c

for i = 2:n

terms = 0;

p\_prev = 1;

for j = 1:i-1

terms = terms + c(j)\*p\_prev; %%subtracting value

p\_prev = p\_prev\*(x(i) - x(j)); %%deviding value

end

c(i) = (y(i) - terms)/p\_prev; %%isolate

end

end

**horner.m code:**

%%%%%% refrence

%%%%%% https://ece.uwaterloo.ca/~dwharder/NumericalAnalysis/05Interpolation/horner/

function p = horner(c,x,X)

k = length(X); %%k different values to test

p = zeros(1,k); %%init p

n = length(x);

for vals = 1:k

val = c(end);

for i = n-1:-1:1

val = val\*(X(vals) - x(i)) + c(i); %%nested

end

p(vals) = val;

end

end

**cosi.m code:**

function [x,c] = cosi(a,b,tol)

n = 1; %initial number of points

err = inf;

while abs(err) > tol

n = n+1;

h = (b-a)/(n);

err = (1/(4\*(n+1)))\*h^(n+1);

end

x = linspace(a,b,n+1);

y = cos(x);

c = newtonint(x,y);

end

**cosi2.m code:**

function [x, c] = cosi2(a,b,n)

x = zeros(1,n); %init x

for i = 0:n-1

%%chebychev spacing

x(i+1) = 0.5\*(a+b) + 0.5\*(b-a)\*cos((2\*i+1)\*pi/(2\*n+2));

end

y = cos(x);

c = newtonint(x,y);

end

**interp3.m code:**

function [x,fx,X,fX,p] = interp3 (a,b,n,f,method)

X = linspace(a,b,100\*n);

fX = f(X);

switch method

case 'uniform'

x = linspace(a,b,n);

fx = f(x);

c = newtonint(x,fx);

p = horner(c,x,X);

case 'spline'

x = linspace(a,b,n);

fx = f(x);

p = spline(x,fx,X);

case 'cheb'

x = zeros(1,n); %init x

for i = 0:n-1

%%chebychev spacing

x(i+1) = 0.5\*(a+b) + 0.5\*(b-a)\*cos((2\*i+1)\*pi/(2\*n+2));

end

fx = f(x);

c = newtonint(x,fx);

p = horner(c,x,X);

end

end