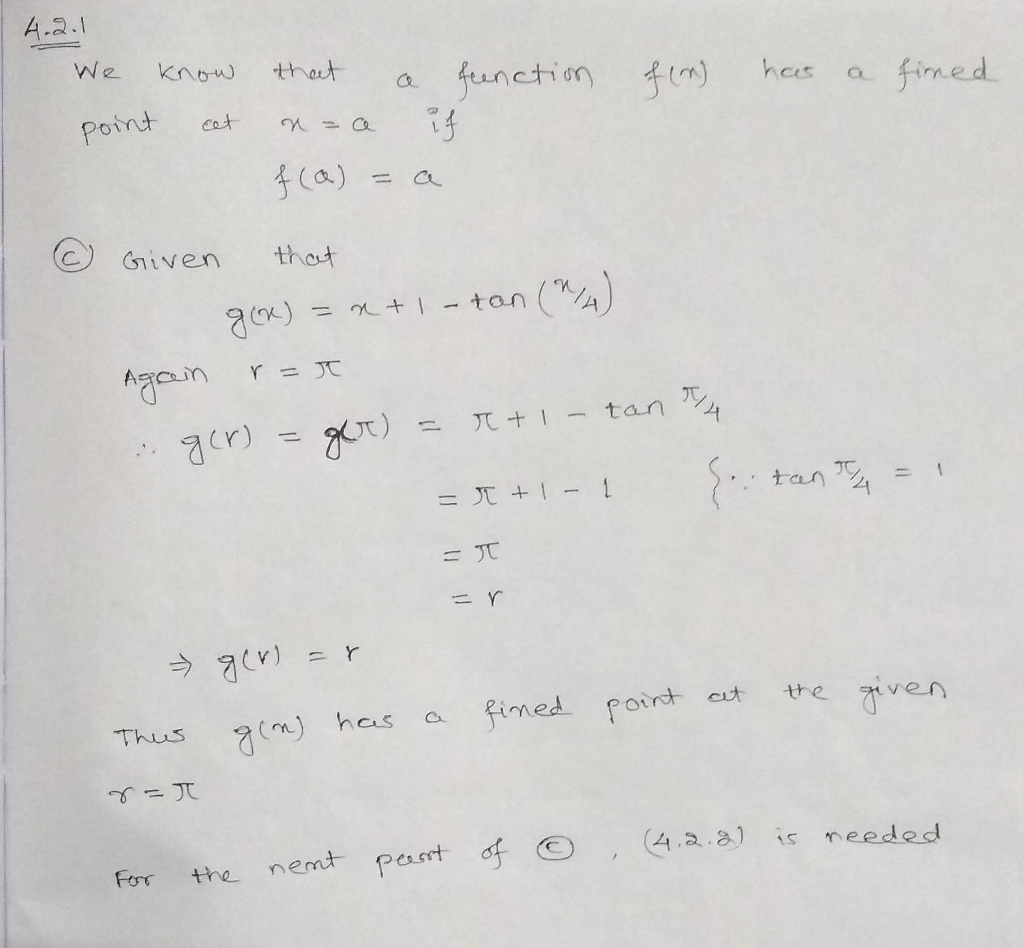
**Question 1**



A)

clc

clear

g = @(x) 1/2\*(x+9/x);

r = 3;

x = pi;

for k = 1:12

x(k + 1) = g(x(k));

end

err = abs(x - r);

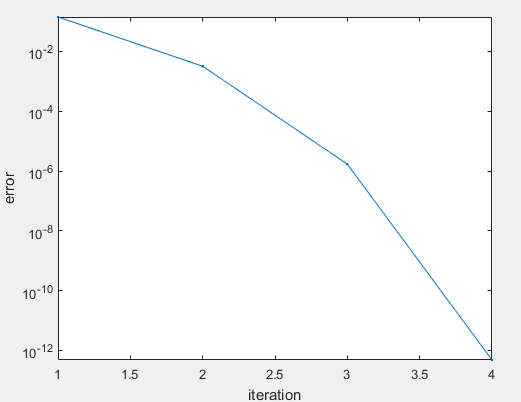
semilogy(err, '.-'), axis tight

xlabel('iteration'), ylabel('error')

p = polyfit(1:13, log(err), 1)

sigma = exp(p(1))

err(9:12) ./ err(8:11)



B)

clc

clear

g = @(x) x + 1 - tan(x/4);

r = pi;

x = 3;

for k = 1:12

x(k + 1) = g(x(k));

end

err = abs(x - r);

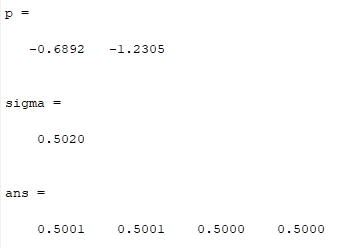
semilogy(err, '.-'), axis tight

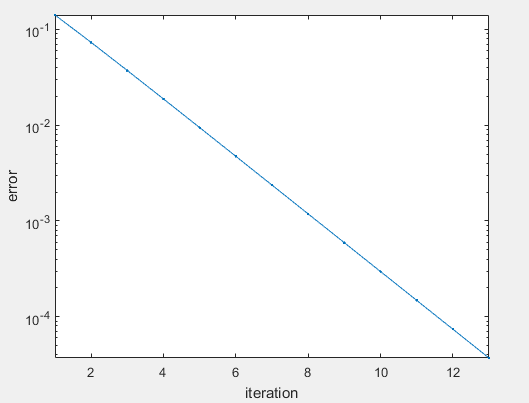
xlabel('iteration'), ylabel('error')

p = polyfit(1:13, log(err), 1)

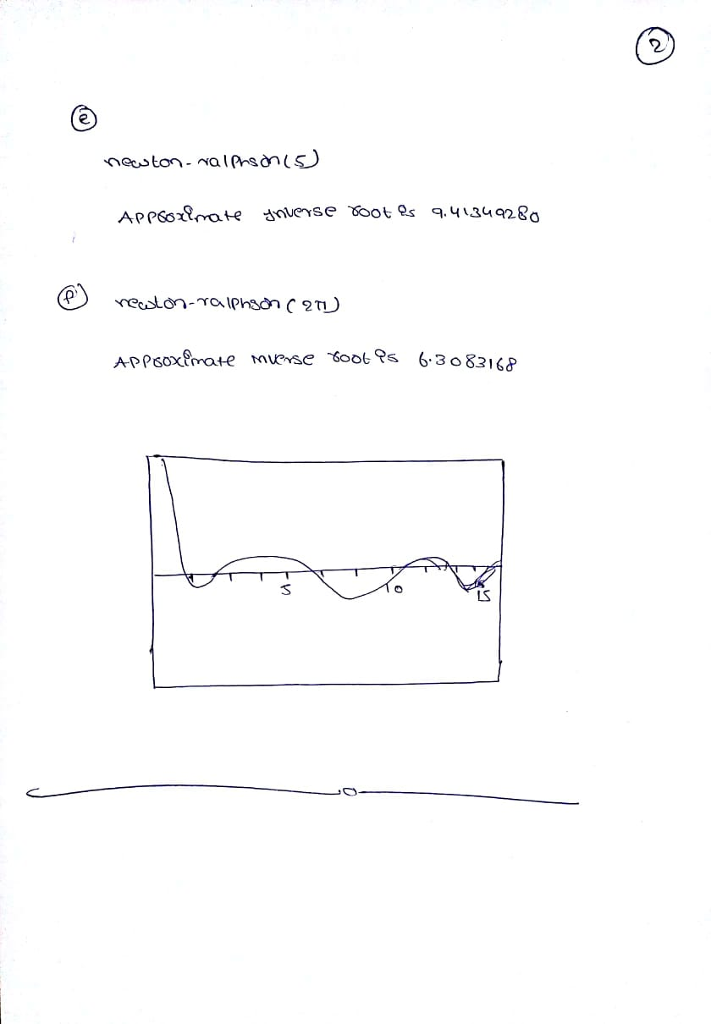
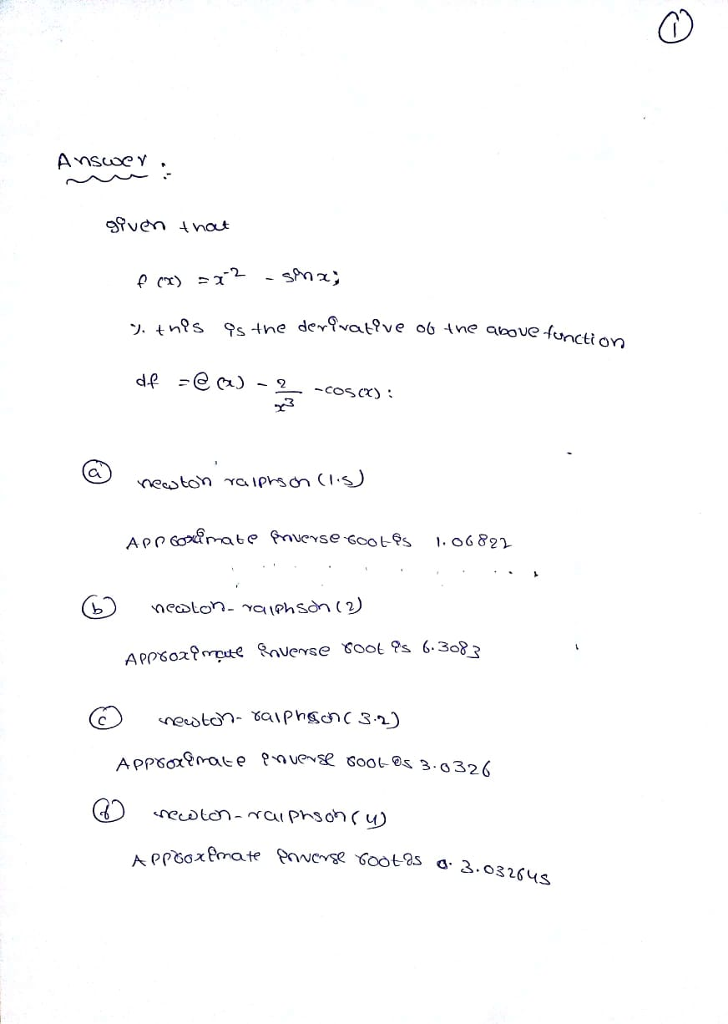
sigma = exp(p(1))

err(9:12) ./ err(8:11)





**Question 2**

**Question 3**

**code.m**

close all  
clear  
clc

year = 1900:10:1990;  
t = (year-1900)/100;  
P = [76.0,92.0,105.7,122.8,131.7,150.7,179.0,205.0,226.5,248.7];  
P = P/100;

fun = @(c,t) (c(1)+c(2)\*exp(c(3)\*t));  
c0 = [1,1,1];  
c = lsqcurvefit(fun, c0, t, P);  
fprintf('c: [%f,%f,%f]\n', c);

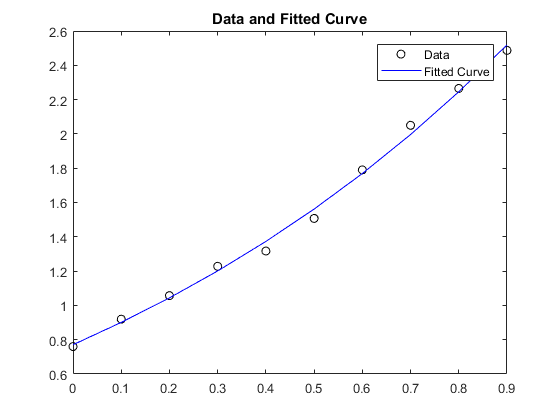
plot(t,P,'ko',t,fun(c,t),'b-');  
legend('Data','Fitted Curve');  
title('Data and Fitted Curve');

% Prediction for 2000  
P\_2000 = (c(1)+c(2)\*exp(c(3)\*(2000-1900)/100))\*100;  
fprintf('Prediction for 2000: %f\n', P\_2000);  
fprintf('Exact population for 2000 (internet): 282.2\n');

**output**

c: [-0.571814,1.342413,0.926683]  
Prediction for 2000: 281.927718  
Exact population for 2000 (internet): 282.2

**plot**



**Problem 4**

**MATLAB Script:**

clear

clc

close all

n = 6;

x = linspace(0, 2.5, n+1);

y = sin(x.^2);

t = x(:); y = y(:);

n = length(t) - 1;

h = diff(t);

Z = zeros(n);

I = eye(n); E = I(1:n-1,:);

J = I - diag(ones(n-1,1),1);

H = diag(h);

AL = [I Z Z Z];

vL = y(1:n);

AR = [I H H^2 H^3];

vR = y(2:n+1);

A1 = E\*[Z J 2\*H 3\*H^2];

v1 = zeros(n-1,1);

A2 = E\*[Z Z J 3\*H];

v2 = zeros(n-1,1);

nakL = [zeros(1,3\*n) [1 -1 zeros(1,n-2)]];

nakR = [zeros(1,3\*n) [zeros(1,n-2) 1 -1]];

A = [AL; AR; A1; A2; nakL; nakR];

v = [vL; vR; v1; v2; 0; 0];

z = A\v;

rows = 1:n;

a = z(rows);

b = z(n+rows);

c = z(2\*n+rows);

d = z(3\*n+rows);

S = zeros(size(x));

xx = 0:0.001:2.5;

for k = 1:n

index = (x>=t(k)) & (x<=t(k+1));

S(index) = polyval([d(k) c(k) b(k) a(k)], x(index)-t(k));

end

plot(xx,sin(xx.^2))

hold on

plot(x,y,'o')

hold on

% plot(xx,S(xx))

yy = spline(x,y,xx);

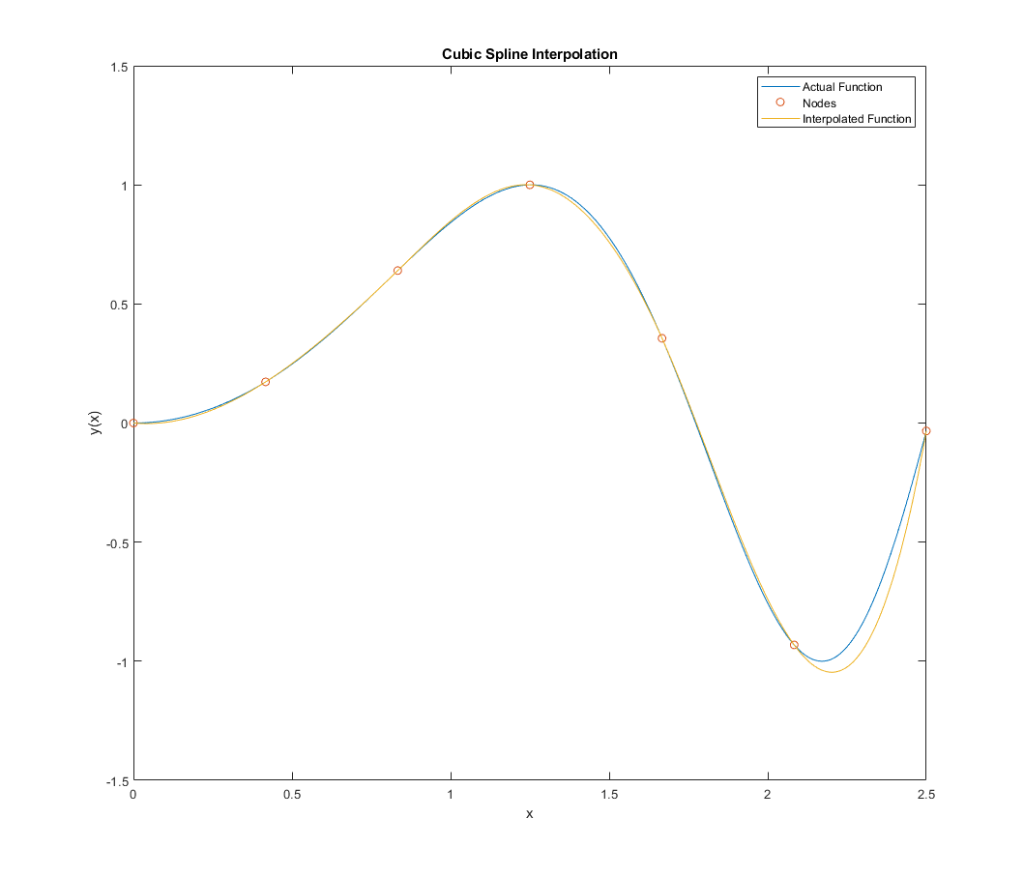
plot(xx,yy)

xlabel('x'), ylabel('y(x)')

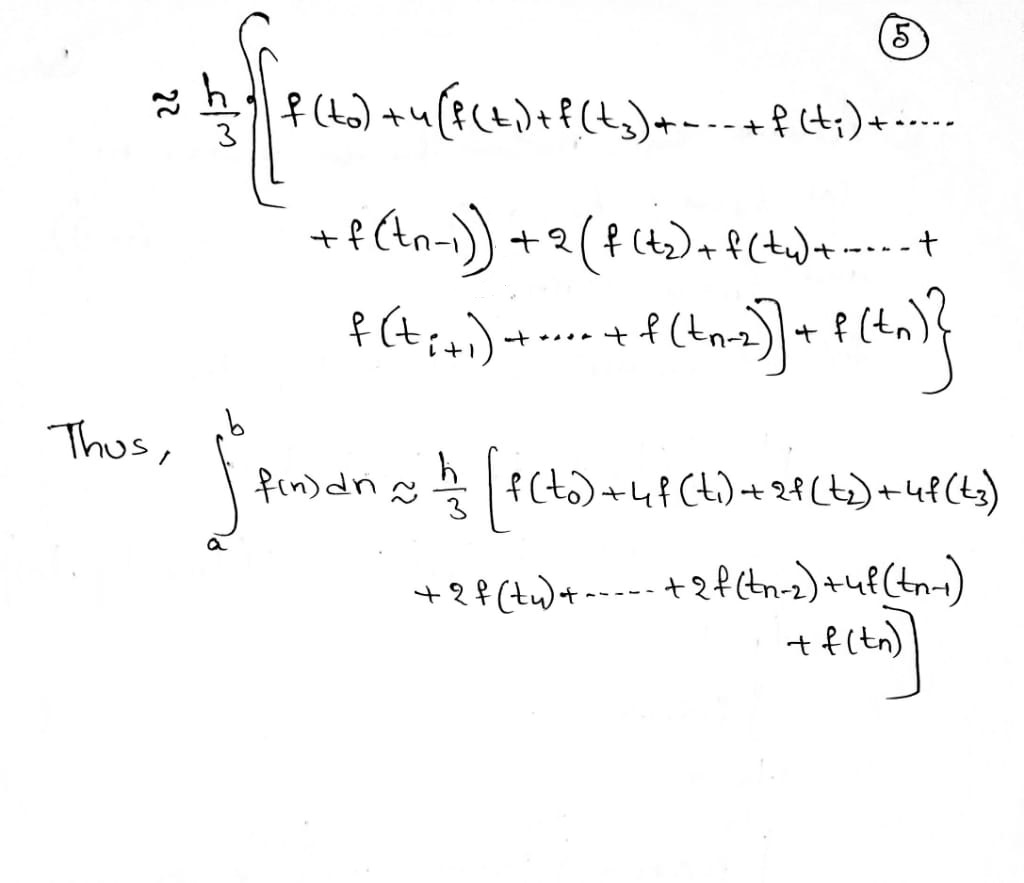
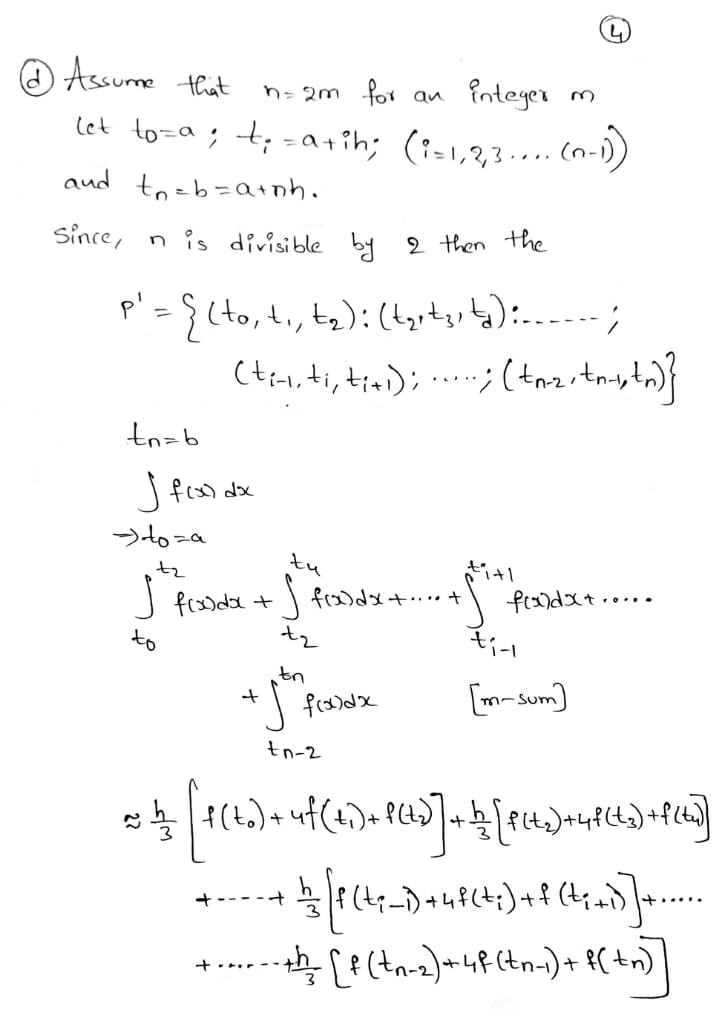
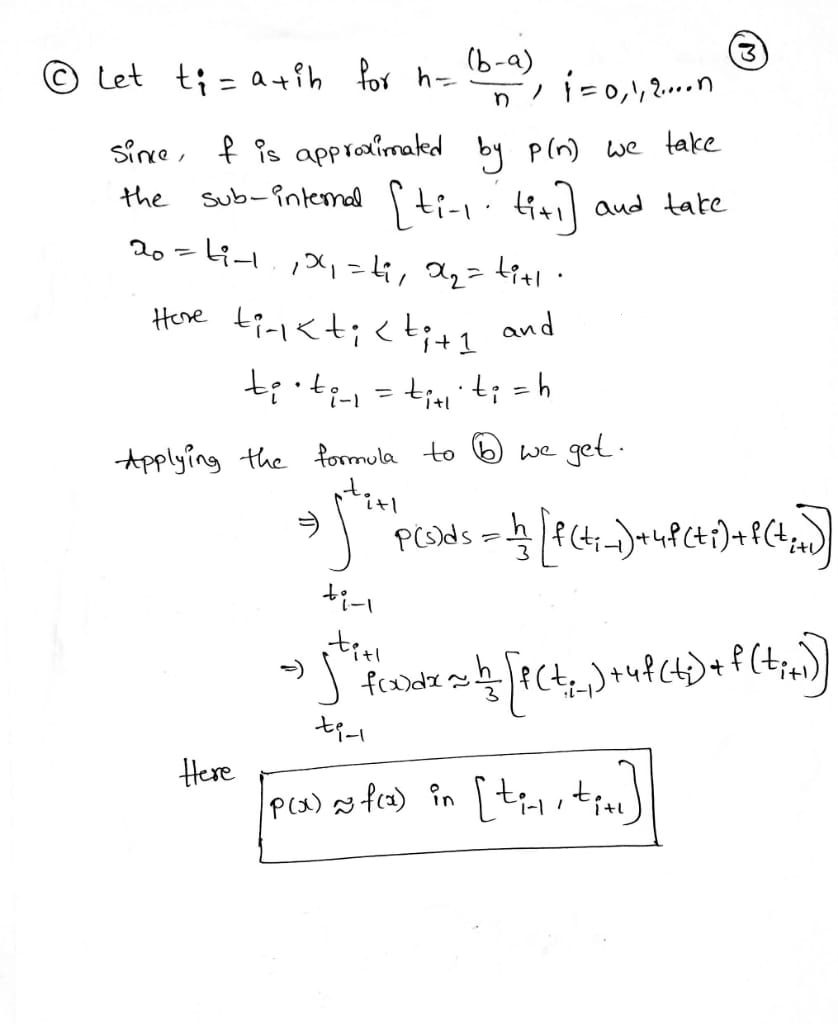
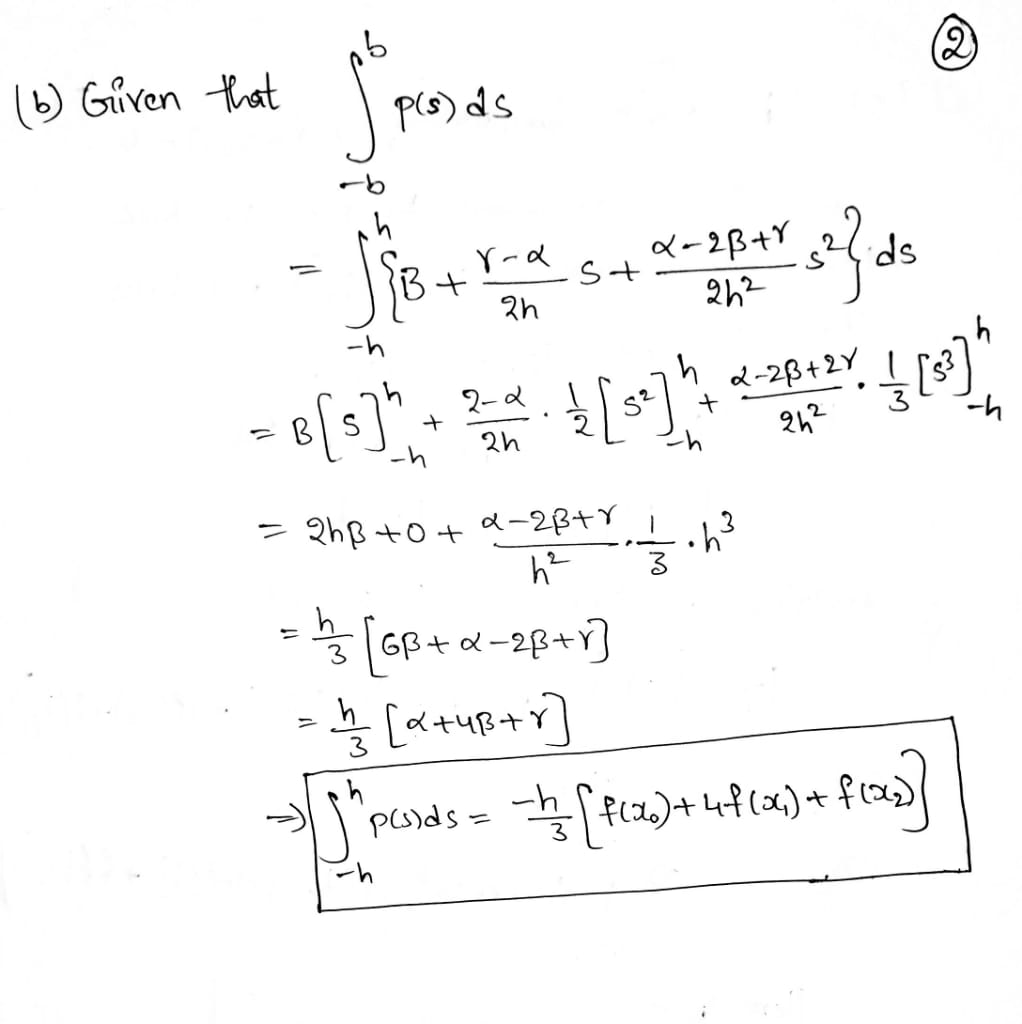
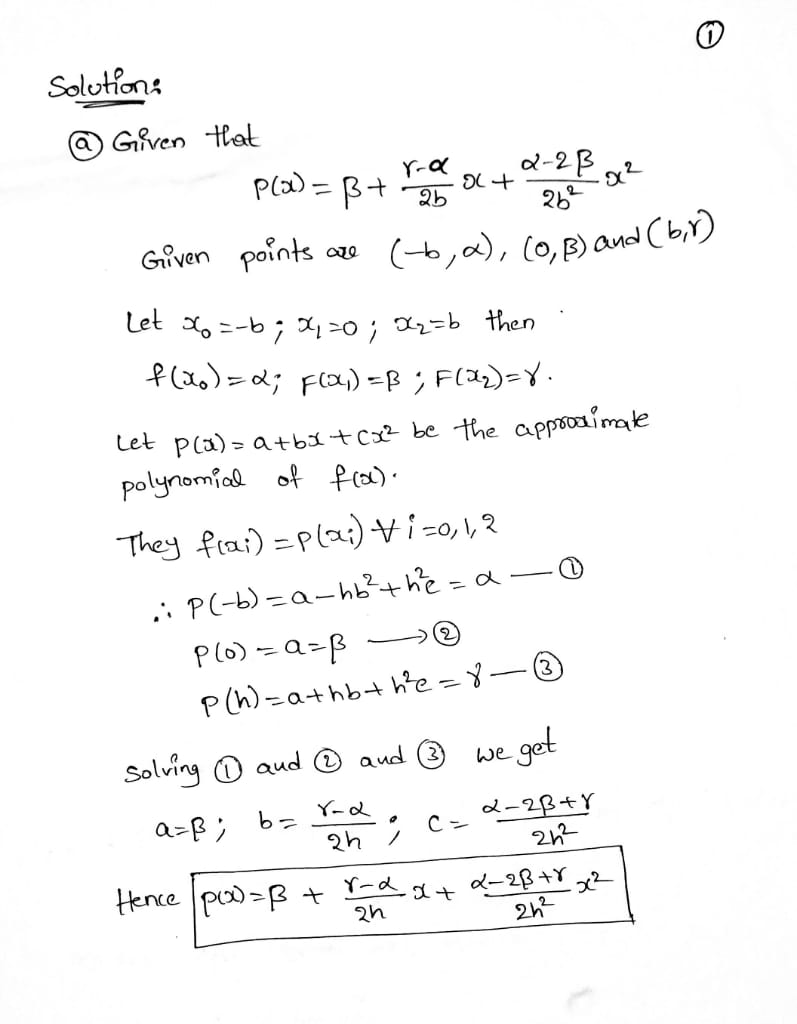
title('Cubic Spline Interpolation')

legend('Actual Function', 'Nodes', 'Interpolated Function')

**Plot:**



**Question 5**



**Question 6**

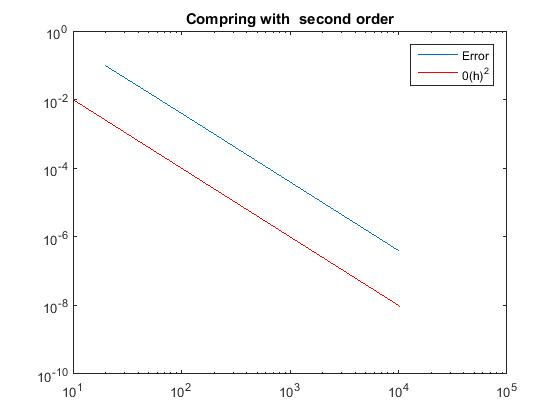
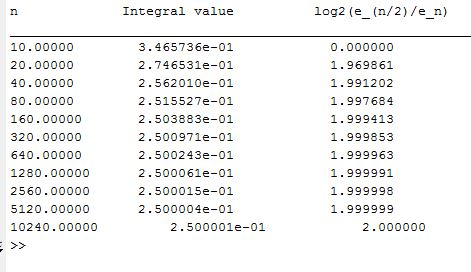
a)

clc;  
clear all;  
clear all;  
f=@(x) x\*log(x+1); % function

a=0; % lower limit  
b=1; % upper limit

n=1;  
k=1;  
Err(1)=0;  
while(n<=1024)  
N(k)=10\*n;  
h=(b-a)/n;% step length  
x=a:h:b;  
for i=1:length(x)  
y(i)=f(x(i));  
end  
% Trapezoid formula  
l=length(x);  
Th=((h/2)\*((y(1)+y(l))+2\*(sum(y)-y(1)-y(l))));  
Error=abs(Th-1/4); % error  
n=n\*2;  
Trpa(k)=Th;  
Err(k+1)=Error;  
Log\_err(k)=log2( Err(k)/ Err(k+1));  
k=k+1;  
end  
disp('n Integral value log2(e\_(n/2)/e\_n) ')  
disp('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')  
Log\_err(1)=0;  
for i=1:k-1  
fprintf('%0.5f \t %15e \t %15f \n',N(i),Trpa(i), Log\_err(i))  
end  
loglog(N,Err(1:k-1))  
hold on  
loglog(N,1./N.^2,'r')  
legend('Error','0(h)^2')  
title('Comparing with second order')

%%%%% Solution



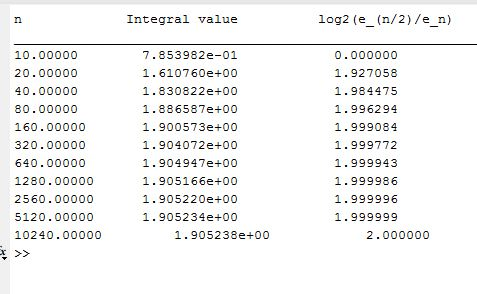
**b)**

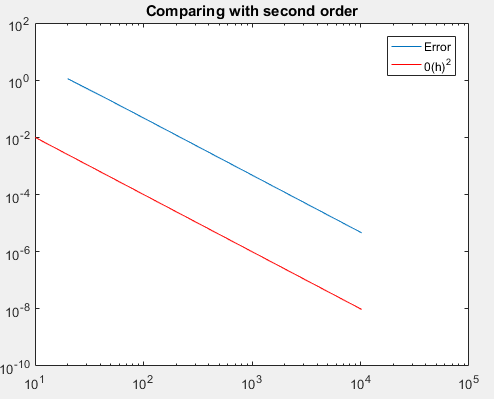
clc;  
clear all;  
clear all;  
f=@(x) exp(x)\*cos(x); % function

a=0; % lower limit  
b=pi/2; % upper limit

n=1;  
k=1;  
Err(1)=0;  
while(n<=1024)  
N(k)=10\*n;  
h=(b-a)/n;% step length  
x=a:h:b;  
for i=1:length(x)  
y(i)=f(x(i));  
end  
% Trapezoid formula  
l=length(x);  
Th=((h/2)\*((y(1)+y(l))+2\*(sum(y)-y(1)-y(l))));  
Error=abs(Th-(exp(pi/2)-1)/2); % error  
n=n\*2;  
Trpa(k)=Th;  
Err(k+1)=Error;  
Log\_err(k)=log2( Err(k)/ Err(k+1));  
k=k+1;  
end  
disp('n Integral value log2(e\_(n/2)/e\_n) ')  
disp('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')  
Log\_err(1)=0;  
for i=1:k-1  
fprintf('%0.5f \t %15e \t %15f \n',N(i),Trpa(i), Log\_err(i))  
end  
loglog(N,Err(1:k-1))  
hold on  
loglog(N,1./N.^2,'r')  
legend('Error','0(h)^2')  
title('Comparing with second order')

%%%%%%% Solution





**C)**

clc;

clear all;

close all;

f=@(x) sqrt(x)\*log(x); % function

a=0; % lower limit

b=1; % upper limit

n=1;

k=1;

Err(1)=0;

while(n<=1024)

N(k)=10\*n;

h=(b-a)/n;% step length

x=a:h:b;

for i=1:length(x)

y(i)=f(x(i));

end

% Trapezoid formula

l=length(x);

Th=((h/2)\*((y(1)+y(l))+2\*(sum(y)-y(1)-y(l))));

Error=abs(Th-4/9); % error

n=n\*2;

Trpa(k)=Th;

Err(k+1)=Error;

Log\_err(k)=log2( Err(k)/ Err(k+1));

k=k+1;

end

disp('n Integral value log2(e\_(n/2)/e\_n) ')

disp('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')

Log\_err(1)=0;

for i=1:k-1

fprintf('%0.5f \t %15e \t %15f \n',N(i),Trpa(i), Log\_err(i))

end

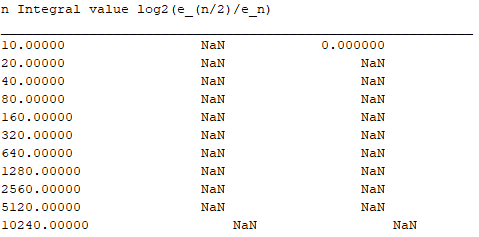
loglog(N,Err(1:k-1))

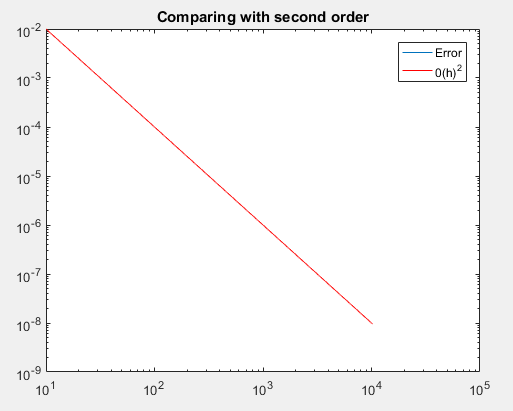
hold on

loglog(N,1./N.^2,'r')

legend('Error','0(h)^2')

title('Comparing with second order')





# Question 07

function [C,X,Y] = cheby(f,n,a,b)

if nargin==2

a=-1;

b=1;

end

d = pi/(2\*n+2);

C = zeros(1,n+1);

for k=1:n+1,

X(k) = cos((2\*k-1)\*d);

end

X = (b-a)\*X/2+(a+b)/2;

x = X;

Y = eval(f);

for k = 1:n+1,

z = (2\*k-1)\*d;

for j = 1:n+1,

C(j) = C(j) + Y(k)\*cos((j-1)\*z);

end

end

C = 2\*C/(n+1);

C(1) = C(1)/2;

plot(X,Y)

title(['chebyshev at node:',num2str(n)])

hold on

plot(X,C,'rO')

end

clc

clear

close

x=0:pi/100:2\*pi;

f=cosh(sin(x));

plot(x,f)

hold on

syms x

f=cosh(sin(x));

n=40;

a=-1;

b=1;

[C,X,Y] = cheby(f,n,a,b);

legend('original plot','chebyshev','coefficient list')

xlim([0 2\*pi])

grid on

