Question 1

Question No. 01

Solution:-

We know that a function $f(x)$ has a fixed point at $x = a$ if,

$$f(a) = a.$$

(a) $g(x) = \frac{1}{2}\left(x + \frac{9}{x}\right)$, $r = 3.$

So,

$$g(r) = g(3) = \frac{1}{2}\left(3 + \frac{9}{3}\right)$$

$$= \frac{1}{2}\left(\frac{9+9}{3}\right)$$

$$= \frac{1}{2}(6)$$

$$g(r) = 3$$

$$\Rightarrow \boxed{g(r) = r = 3}$$

Thus, $g(x)$ has a fixed point at given $r = 3$

(b) $g(x) = x + 1 - \tan\left(\frac{x}{4}\right)$, $r = \pi$

So,

$$g(r) = g(\pi) = \pi + 1 - \tan\left(\frac{\pi}{4}\right)$$

$= \pi + 1 - 1$ $\quad\therefore \boxed{\tan\left(\frac{\pi}{4}\right) = 1}$

$= \pi$

$= r$

$\Rightarrow g(r) = r$

Thus, $g(x)$ has a fixed point at given $r = \pi$.

Question 2:

Question No.02

Solution:-

$f(x) = x^{-2} - \sin x = 0;$

By taking derivative of the above function get,

$df = @x \ \frac{-2}{x^3} - \cos(x):$

(a) Newton Ralphson (1.5)
Approximate inverse roots 1.06822.

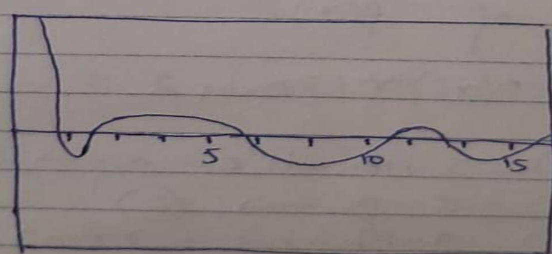
(b) Newton-Ralphson (2)
Approximate inverse roots 6.3083.

(c) Newton – Ralphson (4)

Approximate inverse root is 3.032645

(d) Newton-Ralphson (5)
Approximate inverse root is 9.41349280.

(e) Newton – Ralphson (2π)
Approximate inverse root is 6.3083168



Question 3

code.m

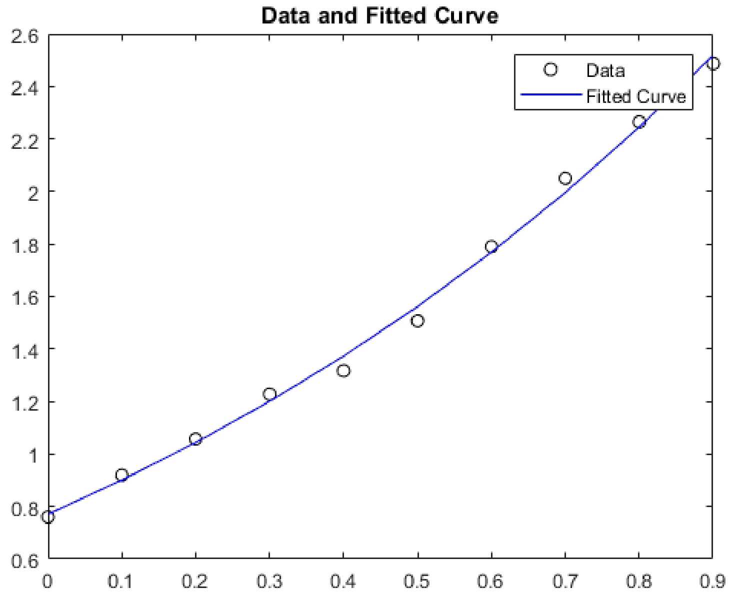
```
close all
clear
clc

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



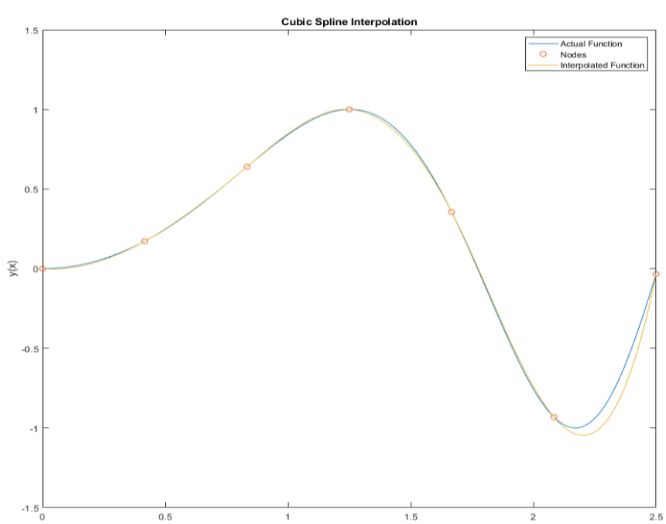
Problem 4

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 No. 05

Solution:-

(a) Given that,

$$P(x) = \beta + \frac{r-\alpha}{2h} x + \frac{\alpha - 2\beta}{2h^2} x^2$$

given points are $(-h, \alpha)$, $(0, \beta)$ and (h, r).

let,

$x_0 = h$; $x_1 = 0$; $x_2 = h$ then

$f(x_0) = \alpha$; $F(x_1) = \beta$; $F(x_2) = \gamma$

let, $p(x) = a + bx + cx^2$ be the approximate polynomial of $f(x)$.

They $f(x_i) = P(x_i) \; \forall \; i = 0, 1, 2$

$$\therefore p(-h) = a - hb^2 + h^2c = \alpha \longrightarrow (1)$$

$$p(0) = a = \beta \longrightarrow (2)$$

$$p(h) = a + hb + h^2c = \gamma \longrightarrow (3)$$

By solving eq (1), (2) & (3) we get,

$$a = \beta \; ; \; b = \frac{\gamma - \alpha}{2h} \; ; \; c = \frac{\alpha - 2\beta + \gamma}{2h^2}$$

Hence,

$$\boxed{P(x) = \beta + \frac{\gamma - \alpha}{2h} x + \frac{\alpha - 2\beta + \gamma}{2h^2} x^2}$$

(2) Given that $\int_{-h}^{h} p(s)\,ds$

$$= \int_{-h}^{h} \left\{ \beta + \frac{\gamma-\alpha}{2h} s + \frac{\alpha-2\beta+\gamma}{2h^2} s^2 \right\} ds$$

$$= \beta\Big[s\Big]_{-h}^{h} + \frac{\gamma-\alpha}{2h}\cdot\frac{1}{2}\Big[s^2\Big]_{-h}^{h} + \frac{\alpha-2\beta+\gamma}{2h^2}\cdot\frac{1}{3}\Big[s^3\Big]_{-h}^{h}$$

$$= 2h\beta + 0 + \frac{\alpha-2\beta+\gamma}{h^2}\cdot\frac{1}{3}\cdot h^3$$

$$= \frac{h}{3}\left[6\beta + \alpha - 2\beta + \gamma\right]$$

$$= \frac{h}{3}\left[\alpha + 4\beta + \gamma\right]$$

So,

$$\boxed{\int_{-h}^{h} p(s)\,ds = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right]}$$

(3) let $x_i = a + ih$ for $h = \frac{(b-a)}{n}$, $i = 0, 1, 2, \dots, n$.

Since, f is approximated by $P(n)$ we take sub-interval $[x_{i-1}, x_{i+1}]$ and take

$$x_0 = x_{i-1} \; ; \; x_1 = x_i \; ; \; x_2 = x_{i+1}$$

Hence,

$$x_{i-1} < x_i < x_{i+1} \text{ and}$$
$$x_i - x_{i-1} = x_{i+1} - x_i = h.$$

Applying the formula to 2 we get.

$$\Rightarrow \int_{x_{i-1}}^{x_{i+1}} p(s)\,d(s) = \frac{h}{3}\left[f(x_{i-1}) + 4f(x_i) + f(x_{i+1})\right]$$

$$\Rightarrow \int_{x_{i-1}}^{x_{i+1}} f(x)\,dx \simeq \frac{h}{3}\left[f(x_{i-1}) + 4f(x_i) + f(x_{i+1})\right]$$

Here,

$$\boxed{p(x) \simeq f(x) \text{ in } [x_{i-1}, x_{i+1}]}$$

(4) Assume that $n = 2m$ for an integer m

Let,

$$x_0 = a;\ x_i = a + ih;\ (i = 1, 2, 3, \ldots, (n-1))$$

and $x_n = b = a + nh$.

Since n is divided by 2 then the

$$p' = \{(x_0, x_1, x_2); (x_2, x_3, x_4); \ldots; (x_{i-1}, x_i, x_{i+1}); \ldots; (x_{n-2}, x_{n-1}, x_n)\}$$

$$\to \int_{x_0 = a}^{x_n = b} f(x)\,dx$$

$$\int_{x_0}^{x_2} f(x)\,d(x) + \int_{x_2}^{x_4} f(x)\,dx + \cdots + \int_{x_{i-1}}^{x_{i+1}} f(x)\,dx + \cdots$$

$$+ \int_{x_{n-2}}^{x_n} f(x)\,dx \quad [m\text{-sum}]$$

$$\approx \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right] + \frac{h}{3}\left[f(x_2) + 4f(x_3) + f(x_4)\right]$$

$$+ \cdots + \frac{h}{3}\left[f(x_{i-1}) + 4f(x_i) + f(x_{i+1})\right] + \cdots +$$

$$+\frac{h}{3}\left[f(x_{n-2})+4(f(x_{n-1})+f(x_n)\right]$$

$$\approx\frac{h}{3}\Big\{\Big[f(x_0)+4f(x_1)+f(x_3)+\cdots+f(x_i)+\cdots+ f(x_{n-1})\Big)+2\Big(f(x_2)+f(x_4)+\cdots+ f(x_{i+1})+\cdots f(x_{n-2})\Big]+f(x_n)\Big\}$$

Thus,

$$\boxed{\int_a^b f(n)\,dn\approx\frac{h}{3}\Big[f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+2f(x_4)+\cdots+2f(x_{n-2})+4f(x_{n-1})+f(x_n)\Big]}$$

Question 6

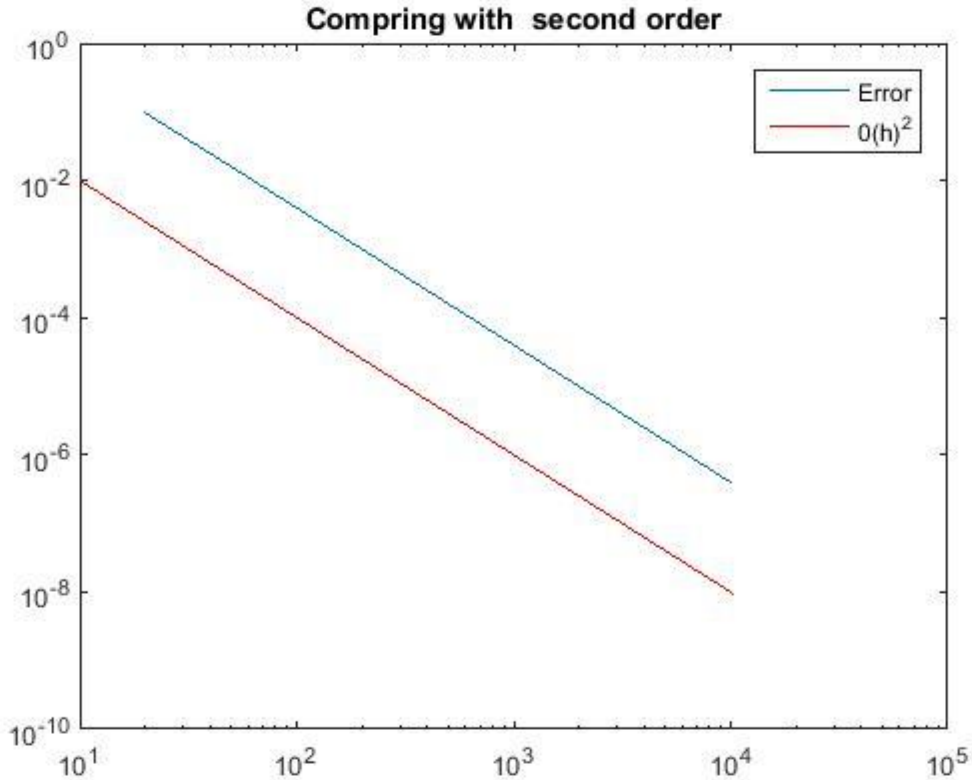
a)

```
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')
```

%%%%% Soltion

```
n              Integral value         log2(e_(n/2)/e_n)
_________________________________________________________
10.00000         3.465736e-01            0.000000
20.00000         2.746531e-01            1.969861
40.00000         2.562010e-01            1.991202
80.00000         2.515527e-01            1.997684
160.00000        2.503883e-01            1.999413
320.00000        2.500971e-01            1.999853
640.00000        2.500243e-01            1.999963
1280.00000       2.500061e-01            1.999991
2560.00000       2.500015e-01            1.999998
5120.00000       2.500004e-01            1.999999
10240.00000          2.500001e-01            2.000000
>>
```



b)

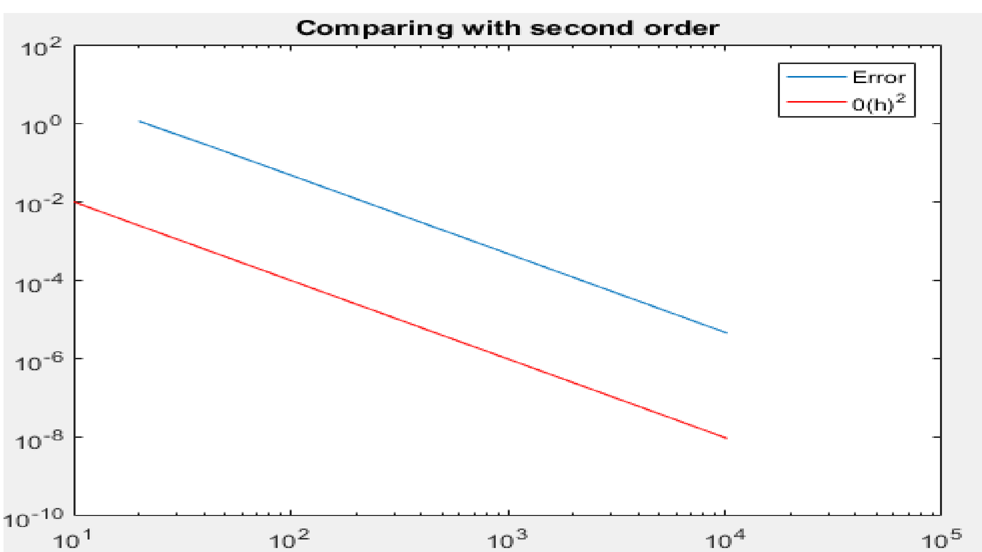
```
clc;
clear all;
clear all;
f=@(x) exp(x)*cos(x); % function

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

n	Integral value	log2(e_(n/2)/e_n)
10.00000	7.853982e-01	0.000000
20.00000	1.610760e+00	1.927058
40.00000	1.830822e+00	1.984475
80.00000	1.886587e+00	1.996294
160.00000	1.900573e+00	1.999084
320.00000	1.904072e+00	1.999772
640.00000	1.904947e+00	1.999943
1280.00000	1.905166e+00	1.999986
2560.00000	1.905220e+00	1.999996
5120.00000	1.905234e+00	1.999999
10240.00000	1.905238e+00	2.000000

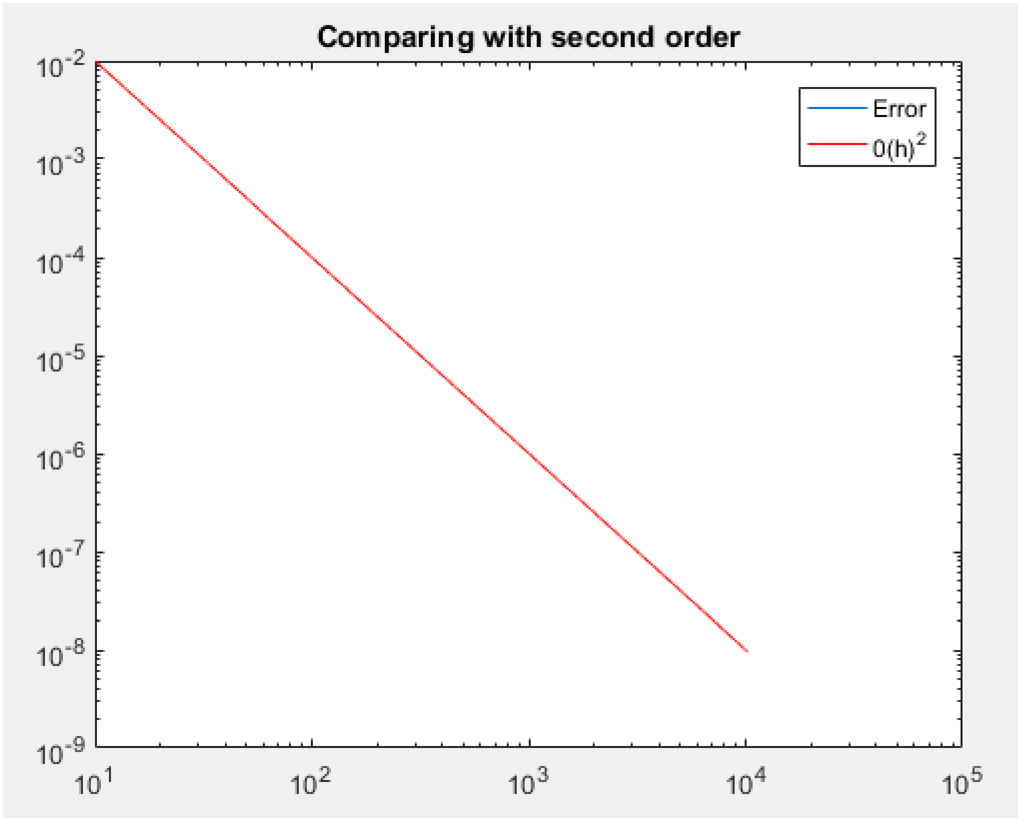
```
>>
```



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','O(h)^2')
title('Comparing with second order')
```

```
n Integral value log2(e_(n/2)/e_n)
_____________________________________________________________
10.00000                 NaN              0.000000
20.00000                 NaN                   NaN
40.00000                 NaN                   NaN
80.00000                 NaN                   NaN
160.00000                NaN                   NaN
320.00000                NaN                   NaN
640.00000                NaN                   NaN
1280.00000               NaN                   NaN
2560.00000               NaN                   NaN
5120.00000               NaN                   NaN
10240.00000                  NaN                   NaN
```



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
```