IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

SECURE OUTSOURCING OF NON-LINEAR OPTIMIZATION IN CLOUD COMPUTING

[1] **Dr.M.RAJARAJESWARI,** [2] **S.MONIKA**
[1]Associate Professor, [2] Student
Department of Mathematics
Hindusthan College of Arts and Science (Autonomous), Coimbatore, Tamilnadu, India

ABSTRACT
Non-linear programming is generally very difficult to find an approximate solution NLPproblem. Security is the primary barrier, mainly for customers for their data are exhaust andyield during computing. This researches secure outsourcing of non-linear programmingcomputing. By validating the result explore the NLP computing and derive the sufficient conditions that result must satisfy. MATLAB Programming to Non-linearprogramming problem by Golden Section Search Method.

Key Words: NLP problem**s,** Cloud computing, MATLAB, Golden Section Method.

1.1 INTRODUCTION TO CLOUD COMPUTING:

Cloud computing often refer to as the "cloud" is a server. Cloud computing refers to on- demand internet access to computing resources such as program, service, data storage, development tools, networking capabilities, etc. that are placed at a remote data center. Cloud computing means storing and accessing data, files and programs over the internet. The cloud environment provides an easily accessible online portal that makes handy for the user to manage the compute, storage, network, and application resources.

1.2 NON-LINEAR PROGRAMMING:

A Non-linear programming is a process of solving an optimization problem where the objective function or the constraints are non-linear. Non-linear programming is the general form of linear programming problem, except the objective functions but the constraints may be linear or non-linear terms. Non-linear programming is a mathematical technique for findingan optimal solution. Non-linear programming are non-linear equations because the decision variables in the objective function having the degree more than one. Several methods have developed for solving non-linear programming problems. Non-linear programming is to calculate the maxima, minima or the stationary points of an objective function of unknown decision variables and conditional to satisfactions of a system of equality and inequality constraints. Non-linear programming is also known as the non-linear optimization.

1.3 GENERAL FORM OF NON-LINEAR PROGRAMMING:

Let z be a real valued objective function of n variables defined by:Optimize (max or min) $z = f(x_1, x_2, \quad x_n)$
Let $(b_1, b_2, \ldots\ldots\ldots\ldots\ldots\ldots b_m)$ be a set of constraints, such that:

$g_1(x_1, x_2, \ldots\ldots\ldots x_n)$ $[\leq$ or $\geq$ or $=]$ b_1

$g_2(x_1, x_2, \ldots\ldots\ldots x_n)$ $[\leq$ or $\geq$ or $=]$ b_2

$g_3(x_1, x_2, \ldots\ldots\ldots x_n)$ $[\leq$ or $\geq$ or $=]$ b_3

$g(x_1, x_2, \ldots\ldots\ldots x_n)$ $[\leq$ or $\geq$ or $=]$ b_m

Where , $g_1(x_1, x_2, \ldots\ldots\ldots x_n)$ $[\leq$ or $\geq$ or $=]$ b_1 ,……. , $g_m(x_1, x_2, \ldots\ldots\ldots x_n)$ $[\leq$ or $\geq$ or $=]$ b_m are structural constraints. Let $_j \geq 0$ where j = 1, 2, ...n be a non-negative constraint.

2. GOLDEN SECTION SEARCH METHOD

2.1 GOLDEN SECTION SEARCH METHOD:

Golden section search method is a technique to find out the extremum (maximum or minimum) for a strictly unimodal function constrict the range of values. The golden section search method is efficient way to progressively reduce the interval locating the minimum/ maximum. This method is for minimising a unimodal function over interval [a, b].
Consider a function f(x)

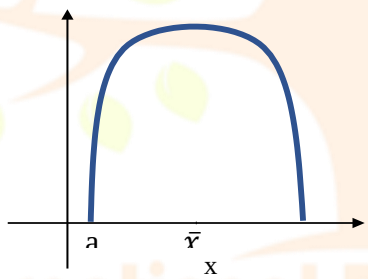
Max f(x)

Subject to $a \le x \le b$

2.1 UNIMODAL FUNCTION:

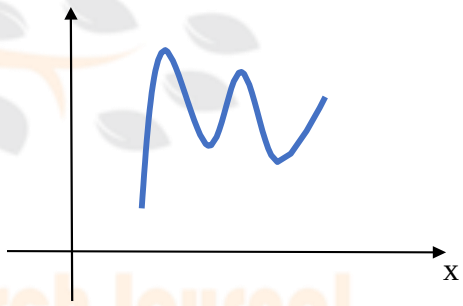
The function which has either only a minimum or maximum in a particular[a, b]. a function f(x) is a unimodal on [a, b] if for some $\bar{x}$ on [a, b], f(x) is strictly increasing on [a, $\bar{x}$] and strictly decreasing on [$\bar{x}$, b].
Let $\bar{x}$ denotes the optimal solution.

f(x)



A unimodal function on [a, b]
$\bar{x}$ = local maximum

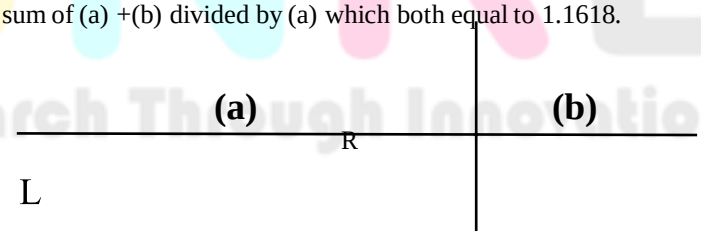
f(x)



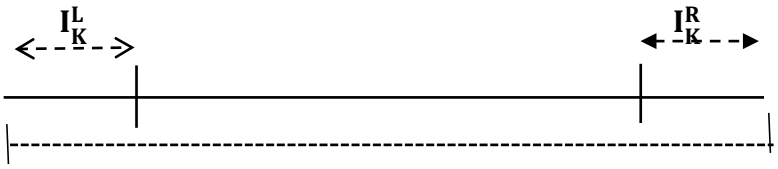
Not a unimodal on [a, b]

2.3 GOLDEN RATIO:

The golden ratio is also known as golden section, golden mean, divine proportion or Greekletter phi (ϕ) exists when line is divided into two parts and the longer part (a) divided by
the smaller part (b) is equal to sum of (a) +(b) divided by (a) which both equal to 1.1618.



NOTATION:



IK

- $I_K = R - L$ Length of the search interval of kth iteration.
- $I^L = x_{1,k} - L$ Length of the left part of the section interval I_K.
- $I^R = R - x_{2,K}$ Length of the right part of the search interval I_K.

2.4 DERIVATION OF GOLDEN RATIO:

How to choose x_1 and x_2

Choose two points x_1 and x_2 in between the L and R according to the following rule:

1) $I_K^L = I_K^R$, for all k

2) Only point either $x1$ or $x2$ is computed For instance

$x_{1,k+1} \leftarrow x_{2,K}$ and $x_{2,k+1}$ is computed for I_{K+1}^R

$x_{2,k+1} \leftarrow x_{1,k}$ and $x_{1,k+1}$ is computed for I_{K+1}^L

such that $I_K = I_{K+1} + I_{K+2}$, for all k

.3) $\frac{I_K}{I_{K+1}} = \frac{I_{K+1}}{I_{K+2}} = c$ (constant) for all k.

Since $I_{K+1} = I_{K+2}$

This is called Golden section criterion.

2.5 GOLDEN SECTION RULE ALGORITHM:

Let initial interval [a, b] be given and the number of iterations be n or tolerance error €

1) Set l = a, R = b; k = 1 (iteration number)

2) Compute x_1 or x_2 as

x_2 = L + 0.618 (R – L) and

x_1 = L + R - x_2

Repeat step 2 until requirement met.

3. PROPOSED SYSTEM:

If f($x1$) < f($x2$) then

$L_{K+1} = L_K$;

$R_K \leftarrow x_2$; $x_2 \leftarrow x_1$ i.e.,

preserve L and update R with $x2$.

If f($x1$) < f($x2$) then

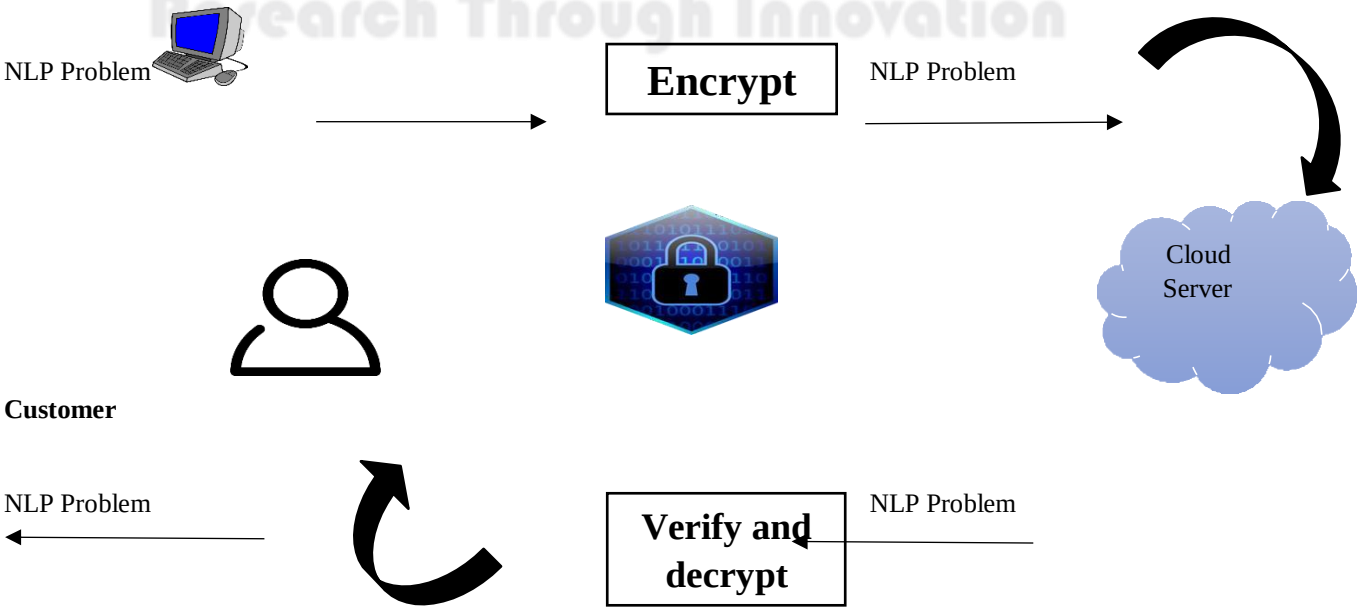
$R_{K+1} = R_K$;

$L_{K+1} \leftarrow x_1$; $x_1 \leftarrow x_2$

i.e., preserve R and

update L with x_1.

A Non-Linear Programming outsourcing scheme which provides a complete outsourcing solution for the privacy protection of problem as well as both input and output. Itstarts from an overview of secure NLP outsourcing in cloud private data owned by the customer.



3.SOLVING PROBLEM BY GOLDEN SECTION SEARCH METHOD

PROBLEM:

Perform 6 iterations of the Golden section search method to minimize $f(x) = \left\{x^2, \frac{1-x}{2}\right\}$ over [-1, 1] and hence identify the final interval I_6

SOLUTION:

Minimize $f(x) = \begin{cases} \frac{1-x}{2}, & x < 0 \\ x^2, & x \geq 0 \end{cases}$ over [-1,1]

ITERATION 1:

$X_2 = L + 0.618\,(R - L)$ Here L = -1, R = 1

$X_2 = -1 + 0.618\,(1-(-1))$

$= -1 + 0.618\,(2)$

$= -1 + 1.236$

$X_2 = 0.236$

$X_1 = L + R - X_2$

$= -1 + 1 - 0.236$

$X_1 = -0.236$

$f(x_1) = \frac{1-x}{2}$

$= \frac{1+0.236}{2}$

$= \frac{1.236}{2}$

$= 0.618$

$f(x_2) = \frac{1-x}{2}$

$= \frac{1-0.236}{2}$

$= \frac{0.764}{2}$

$= 0.382$

ITERATION 2:

$X_2 = L + 0.618\,(R - L)$ Here L = -0.236, R = 1

$X_2 = -0.236 + 0.618\,(1 - (-0.236))$

$= -0.236 + 0.618\,(1+0.236)$

$= -0.236 + 0.618\,(1.236)$

$= -0.236 + 0.7638$

$X_2 = 0.528$

$X_1 = L + R - X_2$

$= -0.236 + 1 - 0.528$

$= 1 - 0.764$

$X_1 = 0.236$

$f(x_1) = \frac{1-x}{2}$

$= \frac{1-0.236}{2}$

$= \frac{0.764}{2}$

$= 0.382$

$f(x_2) = x^2$

$= (0.528)^2$

$= 0.27878$

ITERATION 3:

$X_2 = L + 0.618 (R – L)$ Here $L = 0.236$, $R = 1$

$X_2 = 0.236 + 0.618 (1 – 0.236)$

$= 0.236 + 0.618 (0.764)$

$= 0.236 + 0.4721$

$X_2 = 0.708$

$X_1 = L + R - X_2$

$= 0.236 + 1 – 0.708$

$= 1.236 – 0.708$

$X_1 = 0.528$

$f(x_1) = x^2$

$= (0.528)^2$

$= 0.27878$

$f(x_2) = x^2$

$= (0.708)^2$

$= 0.50126$

ITERATION 4:

$X_2 = L + 0.618 (R – L)$ Here $L = 0.236$, $R = 0.708$

$X_2 = 0.236 + 0.618 (0.708 – 0.236)$

$= 0.236 + 0.618 (0.472)$

$= 0.236 + 0.2916$

$X_2 = 0.528$

$X_1 = L + R - X_2$

$= 0.236 + 0.708 - 0.528$

$= 0.944 – 0.528$

$X_1 = 0.416$

$f(x_1) = \frac{1-x}{2}$

$= \frac{1-0.416}{2}$

$= \frac{0.584}{2}$

$= 0.292$

$f(x_2) = x^2$

$= (0.528)^2$

$= 0.27878$

ITERATION 5:

$X_2 = L + 0.618 (R – L)$ Here $L = 0.416$, $R = 0.708$

$X_2 = 0.416 + 0.618 (0.708 – 0.416)$

$= 0.416 + 0.618 (0.292)$

$= 0.416 + 0.1804$

$X_2 = 0.596$

$X_1 = L + R - X_2$

$= 0.416 + 0.708 – 0.596$

$= 1.124 – 0.596$

$X_1 = 0.528$

$f(x_1) = x^2$

$= (0.528)^2$

$= 0.27878$

$f(x_2) = x^2$

$= (0.596)^2$

$= 0.35522$

ITERATION 6:

$X_2 = L + 0.618 (R – L)$ Here $L = 0.416$, $R = 0.596$

$X_2 = 0.416 + 0.618 (0.596 – 0.416)$

$= 0.416 + 0.618 (0.18)$

$= 0.416 + 0.1112$

$X_2 = 0.528$

$X_1 = L + R - X_2$

$X_1 = 0.416 + 0.596 – 0.528$

$= 1.012 – 0.528$

$X_1 = 0.484$

$f(x_1) = \frac{1-x}{2}$

$= \frac{1-0.484}{2}$

$= \frac{0.516}{2}$

$= 0.258$

$f(x_2) = x^2$

$= (0.528)^2$

$= 0.27878$

K	**L**	**R**	x_1	x_2	$f(x_1)$	$f(x_2)$	**L/R**
1	-1	1	-0.236	0.236	0.618	0.382	R
2	-0.236	1	0.236	0.528	0.382	0.27878	R
3	0.236	1	0.528	0.708	0.27878	0.50126	L
4	0.236	0.708	0.416	0.528	0.292	0.27878	R
5	0.416	0.708	0.528	0.596	0.27878	0.35522	L
6	0.416	0.596	0.484	0.528	0.258	0.27878	

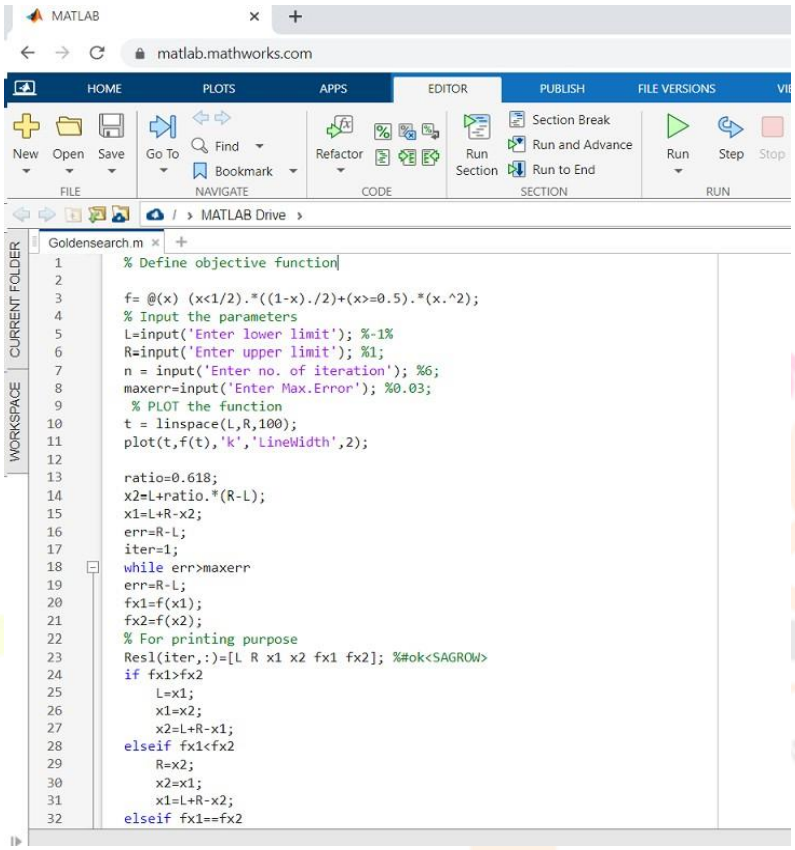
Thus, $x_{min} \in [0.416, 0.596]$

Hence, $x^* = \frac{0.416+0.596}{2}$

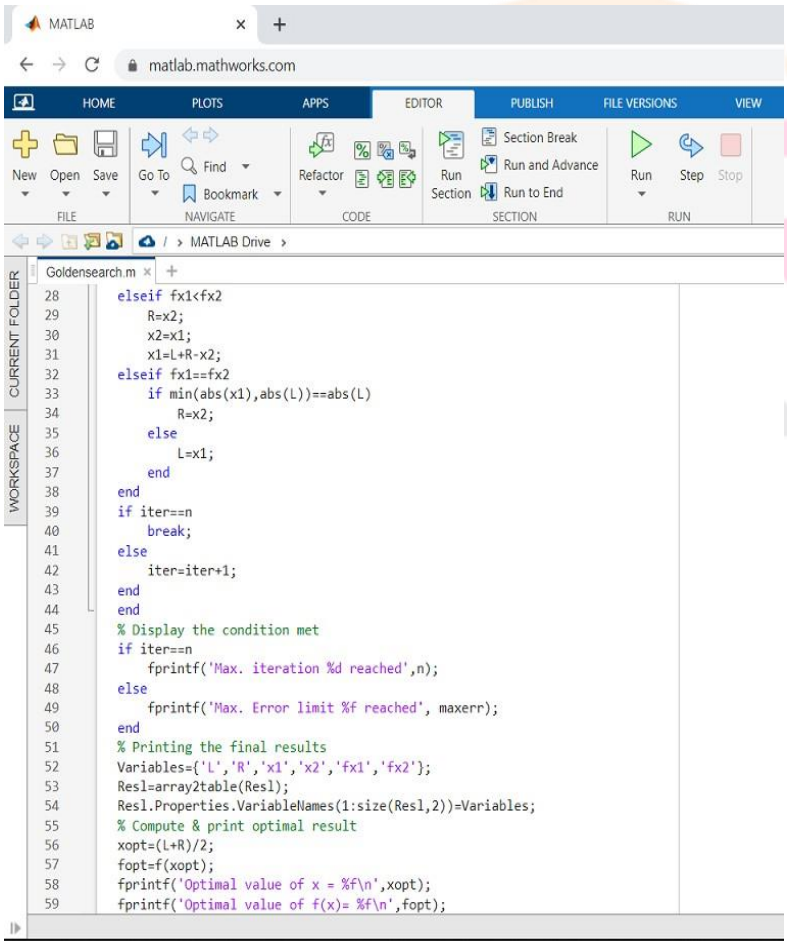
Therefore, Min f(x') = 0.2640

MATLAB CODING FOR GOLDEN SECTION SEARCH METHOD:

```
% Define objective function

f= @(x) (x<1/2).*((1-x)./2)+(x>=0.5).*(x.^2);
% Input the parameters L=input('Enter lower limit'); %-1%R=input('Enter upper limit'); %1;
n = input('Enter no. of iteration'); %6; maxerr=input('Enter Max.Error'); %0.03;
% PLOT the functiont = linspace(L,R,100);
plot(t,f(t),'k','LineWidth',2);

ratio=0.618; x2=L+ratio.*(R-L);x1=L+R-x2;
err=R-L;iter=1;
while err>maxerrerr=R-L; fx1=f(x1); fx2=f(x2);
% For printing purpose
Resl(iter,:)=[L R x1 x2 fx1 fx2]; %#ok<SAGROW>if fx1>fx2
L=x1;
x1=x2; x2=L+R-x1;
elseif fx1<fx2R=x2;
x2=x1; x1=L+R-x2;
elseif fx1==fx2
if min(abs(x1),abs(L))==abs(L)R=x2;
else
L=x1;
endend
if iter==nbreak;
else
iter=iter+1;end
end
% Display the condition metif iter==n
fprintf('Max. iteration %d reached',n);else
fprintf('Max. Error limit %f reached', maxerr);end
% Printing the final results Variables={'L','R','x1','x2','fx1','fx2'}; Resl=array2table(Resl);
Resl.Properties.VariableNames(1:size(Resl,2))=Variables;
% Compute & print optimal resultxopt=(L+R)/2;
fopt=f(xopt);
fprintf('Optimal value of x = %f\n',xopt); fprintf('Optimal value of f(x)= %f\n',fopt)
```



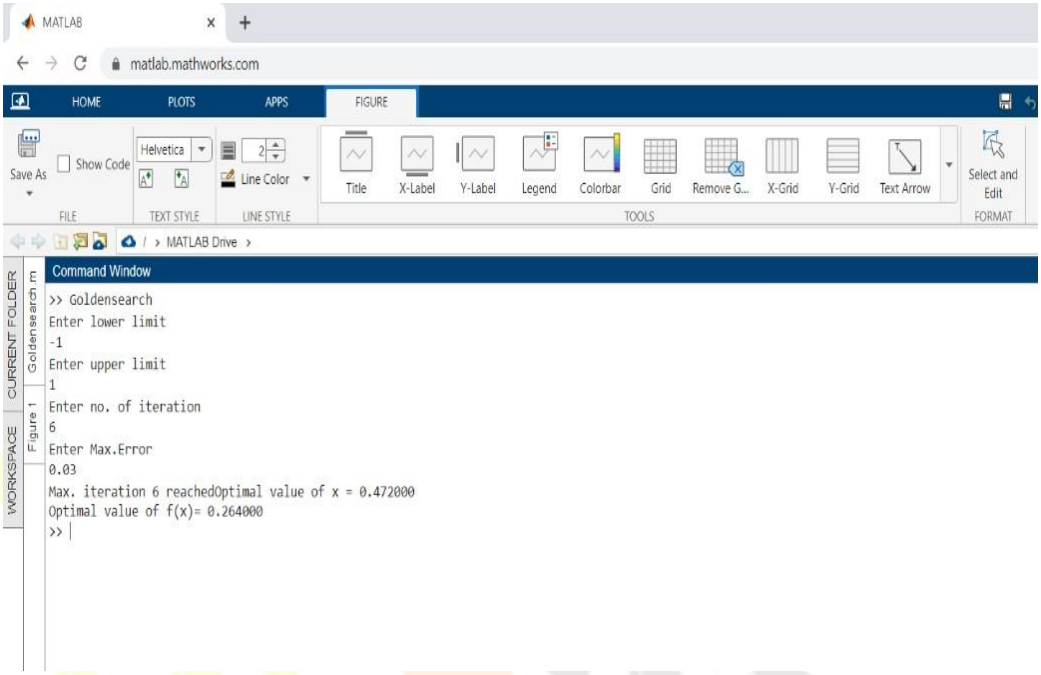
```
% Define objective function

f= @(x) (x<1/2).*((1-x)./2)+(x>=0.5).*(x.^2);
% Input the parameters
L=input('Enter lower limit'); %-1%
R=input('Enter upper limit'); %1;
n = input('Enter no. of iteration'); %6;
maxerr=input('Enter Max.Error'); %0.03;
 % PLOT the function
t = linspace(L,R,100);
plot(t,f(t),'k','LineWidth',2);

ratio=0.618;
x2=L+ratio.*(R-L);
x1=L+R-x2;
err=R-L;
iter=1;
while err>maxerr
err=R-L;
fx1=f(x1);
fx2=f(x2);
% For printing purpose
Resl(iter,:)=[L R x1 x2 fx1 fx2]; %#ok<SAGROW>
if fx1>fx2
    L=x1;
    x1=x2;
    x2=L+R-x1;
elseif fx1<fx2
    R=x2;
    x2=x1;
    x1=L+R-x2;
elseif fx1==fx2
```



```
elseif fx1<fx2
    R=x2;
    x2=x1;
    x1=L+R-x2;
elseif fx1==fx2
    if min(abs(x1),abs(L))==abs(L)
        R=x2;
    else
        L=x1;
    end
end
if iter==n
    break;
else
    iter=iter+1;
end
end
% Display the condition met
if iter==n
    fprintf('Max. iteration %d reached',n);
else
    fprintf('Max. Error limit %f reached', maxerr);
end
% Printing the final results
Variables={'L','R','x1','x2','fx1','fx2'};
Resl=array2table(Resl);
Resl.Properties.VariableNames(1:size(Resl,2))=Variables;
% Compute & print optimal result
xopt=(L+R)/2;
fopt=f(xopt);
fprintf('Optimal value of x = %f\n',xopt);
fprintf('Optimal value of f(x)= %f\n',fopt);
```

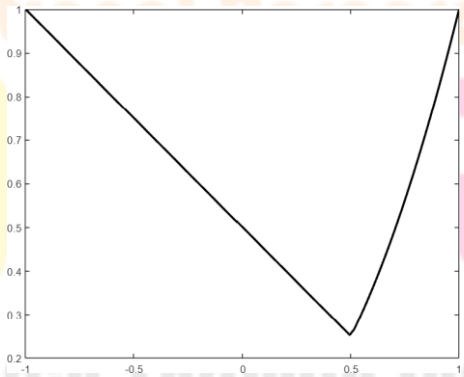
GOLDEN SECTION SEARCH MATLAB CODING

OUTPUT:



COMMAND WINDOW OF GOLDEN SECTION SEARCH CODING

GRAPH OF GOLDEN SECTION SEARCH:



GOLDEN SECTION SEARCH GRAPH

CONCLUSION

Formalized the issue of safely reevaluating NLP calculations in distributed computing, andprovided such a proposed model which satisfies info and result security. Such outcome confirmation instrument is incredibly effective and causes near zero extra expense on both cloud server and clients. Broad security investigation and analysis results show the prompt practicability of our plan.

REFERENCES

1.Simmons, D.M., Non-linear Programming for Operations Research, Prentice-Hall, Englewood Cliffs, N.J., 1975

2.Hadley, G., Non-Linear and Dynamic Programming, Addison-Wesley, Reading Mass,1964

3.Cong Wang, Secure Optimization Computation Outsourcing in Cloud Computing: A case study of Linear Programming, IEEE transactions on computers, vol, 65, no. 1, January 2016.

4.Wayne L. Winston, Operations Research Applications And Algorithms, Fourth Edition

5.J. Kowalik and M. R. Osborne (1968), Methods for unconstrained optimization problems, American Elsevier Publishing Company, Inc..