EEVAL PROJECT-1

PROJECT REPORT ON SOLVING ROSENBROCKS BANANA FUNCTION USING VARIOUS ALGORITHMS

SUBMITTED BY: KEERTHI SURESH

STUDENT ID: X-1994

BATCH: MSC Computer System and Networking - October 2020

SUBJECT: Evolutionary Algorithms

SUBMITTED TO: Dr Adam Raniszewski

DATE OF SUBMISSION: 22-11-2020

Table of Contents

OPTIMISATION PROBLEM TO SOLVE ROSENBROCK BANANA FUNCTION

OBJECTIVE

Find a minimum of the Rosenbrock’s (banana) function without constraints:

F (x)=[1− x +a]^2 +100 [y−b −(x− a)^2]^2

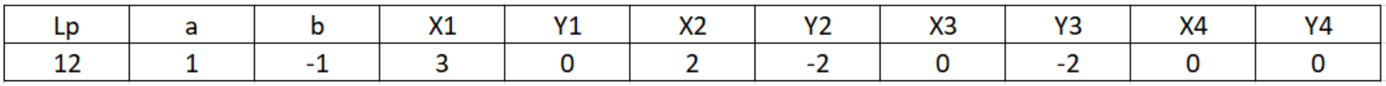
F(x) is called the banana function because of its curvature around the origin. It is notorious in optimization examples because of the slow convergence most methods exhibit when trying to solve this problem.

Here

```
rng(994)
randi(30)
```

```
ans = 12
```

Here I am using row number 12 from the given dataset.



Lp	a	b	X1	Y1	X2	Y2	X3	Y3	X4	Y4
12	1	-1	3	0	2	-2	0	-2	0	0

So, the f(x) gets modified to:

F (x)=[1− x +a]^2 +100 [y−b −(x− a)^2]^2

OPTIMISATION OF BANANA FUNCTION

Optimisation of Banana function can be done using fminunc function inside optimtool.

fminsearch finds the minimum of a scalar function of several variables, starting at an initial estimate. ... x = fminsearch (fun,x0) starts at the point x0 and finds a local minimum x of the function described in fun . x0 can be a scalar, vector, or matrix.

fminunc - Unconstrained nonlinear minimization -> Quasi Newton

fminunc - Unconstrained nonlinear minimization -> Trust Region

```
%defining X,Y values
X1 = 3;Y1 = 0;
X2 = 2;Y2 = -2;
X3 = 0;Y3 = -2;
X4 = 0;Y4 = 0;
save parameters.mat
```

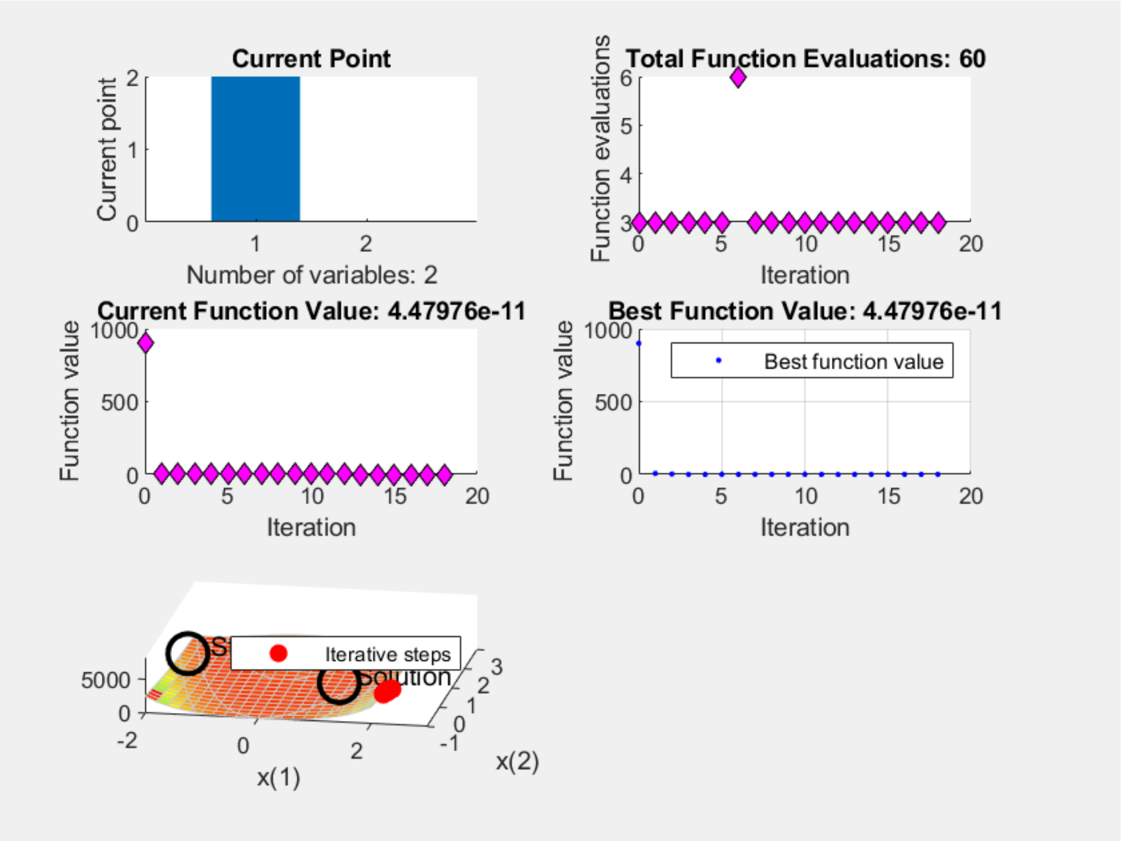
ALGORITHM:quassi-newton ; HESSIAN: bfgs

(X1,Y1):

```
close all; clear; clc;
% Setup optimization options
```

```
options = optimoptions(@fminunc,'Display','final','Algorithm','quasi-newton', ...
    'HessUpdate','bfgs', ...
    'PlotFcn',{@optimplotx,@optimplotfunccount,@optimplotfval, ...
    @optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [3,0];    %for x1,y1=(3,0)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options)
```



```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
xy_opt = 1×2
    2.0000   -0.0000
fval = 4.4798e-11
eflag = 1
output = struct with fields:
       iterations: 18
        funcCount: 60
         stepsize: 4.4966e-05
     lssteplength: 1
    firstorderopt: 7.2717e-06
        algorithm: 'quasi-newton'
          message: '↵Local minimum found.↵↵Optimization completed because the size of the gradient is less than↵the valu
```

```
x1y1a1i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 18
```

```
x1y1a1f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2 -1.3368e-05
```

```
x1y1a1o=fval;
disp(['Objective value = ' num2str(fval)])
```

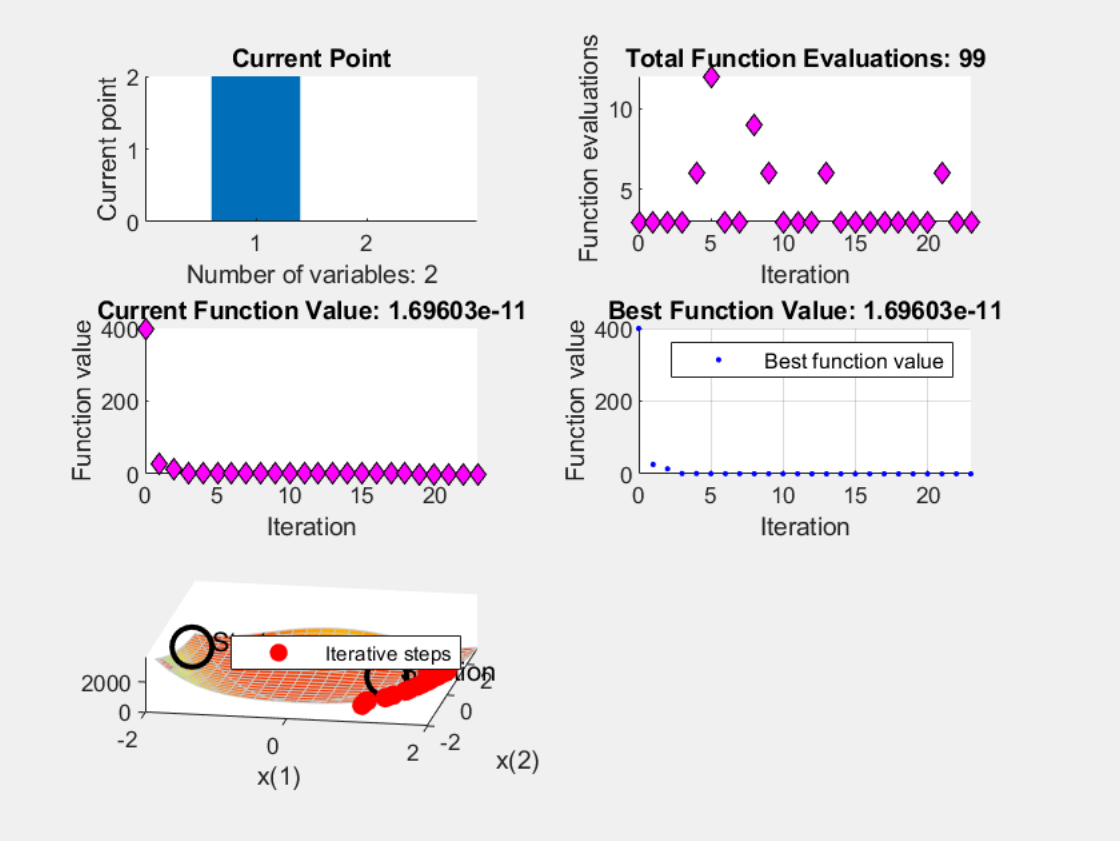
```
Objective value = 4.4798e-11
```

```
save ('parameters.mat','-append');
```

(X2,Y2):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm','quasi-newton', ...
    'HessUpdate','bfgs', ...
    'PlotFcn',{@optimplotx,@optimplotfunccount,@optimplotfval, ...
    @optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [2,-2];    %for x2,y2=(2,-2)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Local minimum found.
```

```
Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

```matlab
x2y2a1i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 23
```

```matlab
x2y2a1f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2  8.1952e-06
```

```matlab
x2y2a1o=fval;
disp(['Objective value = ' num2str(fval)])
```

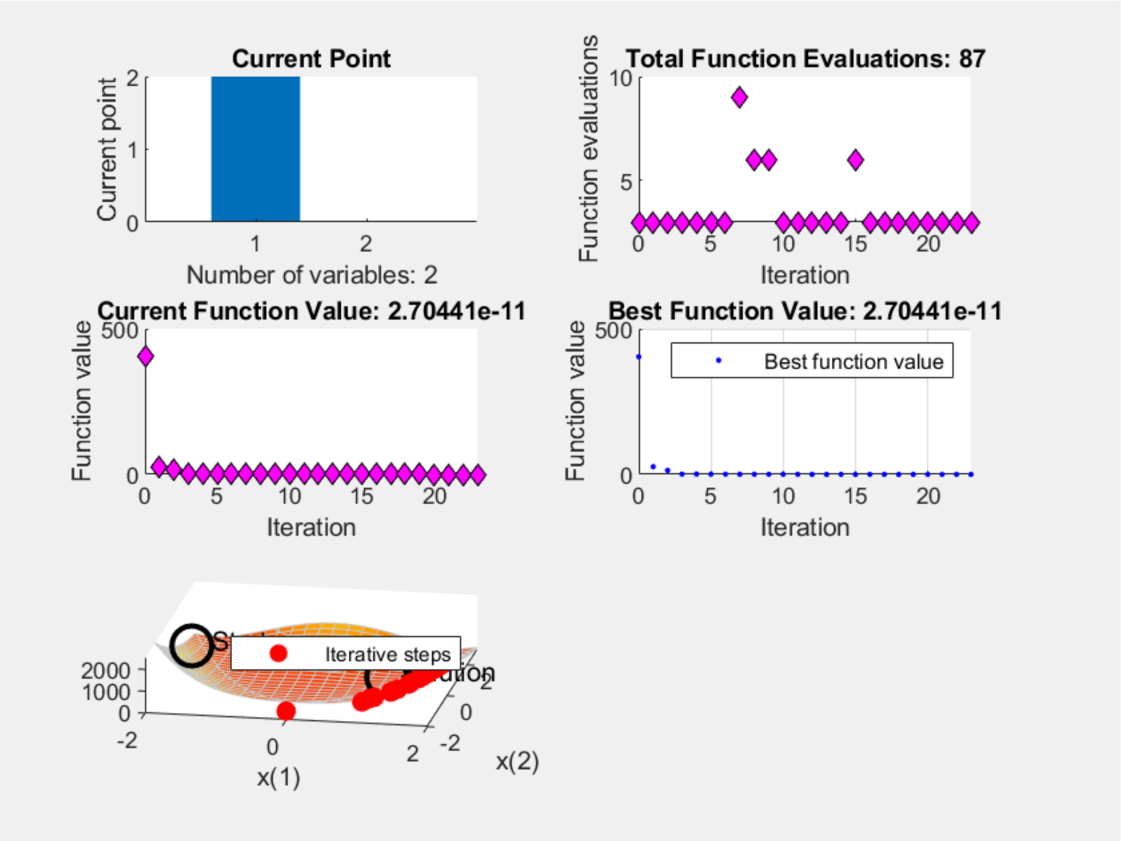
```
Objective value = 1.696e-11
```

```matlab
save ('parameters.mat','-append');
```

(X3,Y3):

```matlab
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm', ...
    'quasi-newton','HessUpdate','bfgs', ...
    'PlotFcn',{@optimplotx,@optimplotfunccount,@optimplotfval, ...
    @optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [0,-2];    %for x3,y3=(0,-2)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

```
x3y3a1i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 23
```

```
x3y3a1f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2 -9.2745e-06
```

```
x3y3a1o=fval;
disp(['Objective value = ' num2str(fval)])
```

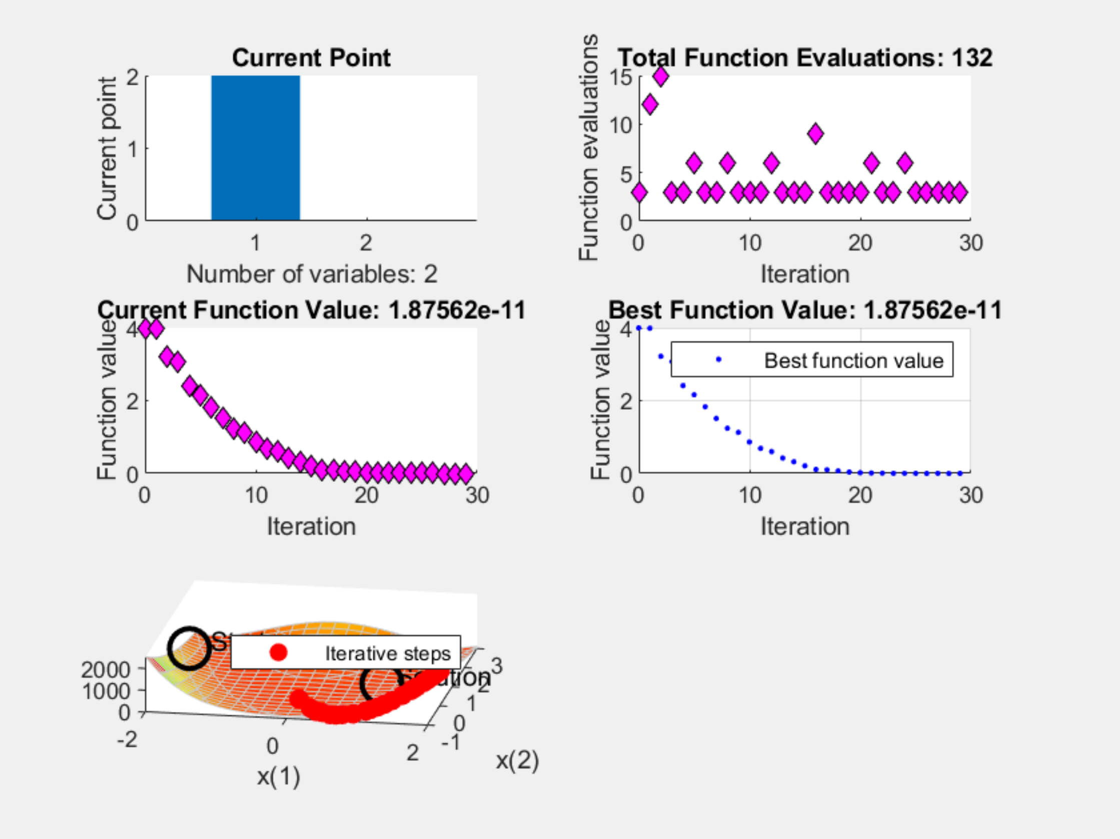
```
Objective value = 2.7044e-11
```

```
save ('parameters.mat','-append');
```

(X4,Y4):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm', ...
    'quasi-newton','HessUpdate','bfgs', ...
    'PlotFcn',{@optimplotx,@optimplotfunccount,@optimplotfval, ...
    @optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [0,0];     %for x4,y4=(0,0)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

```
x4y4a1i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 29
```

```
x4y4a1f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2 -8.6524e-06
```

```
x4y4a1o=fval;
disp(['Objective value = ' num2str(fval)])
```

```
Objective value = 1.8756e-11
```

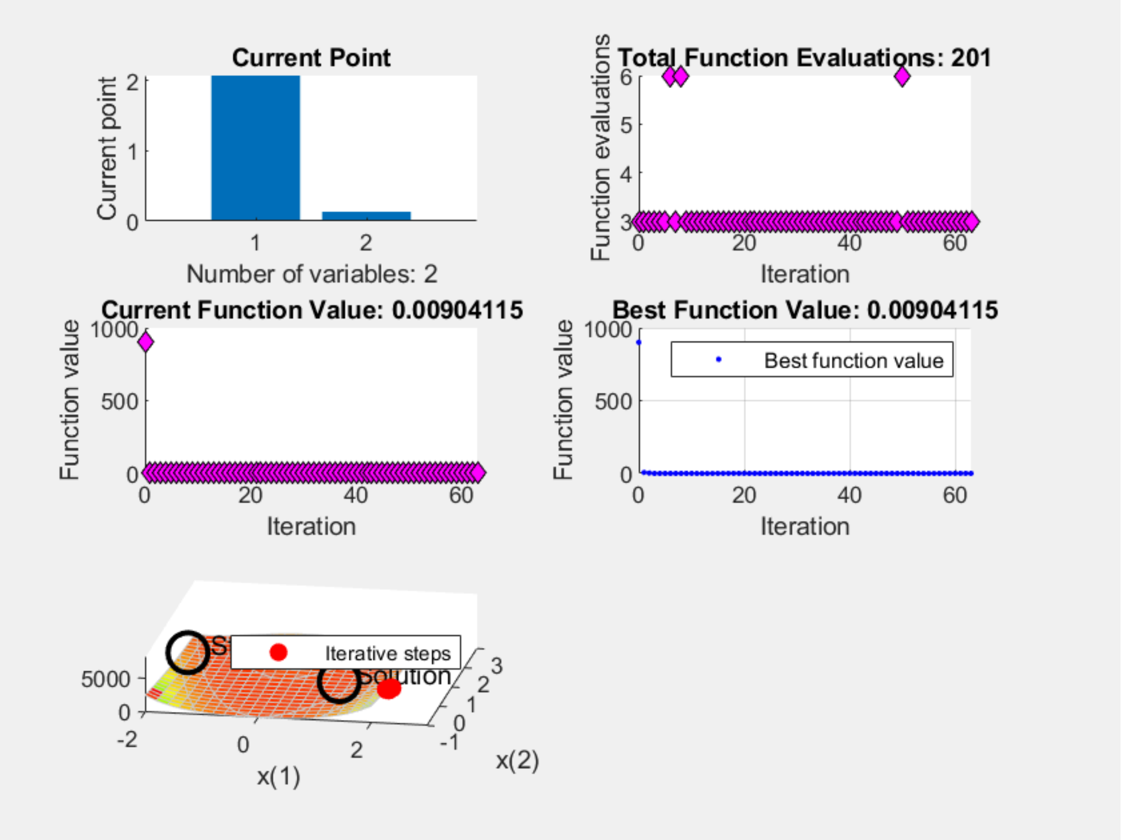
```
save ('parameters.mat','-append');
```

ALGORITHM:quassi-newton ; HESSIAN: dfp(Inverse Hessian)

(X1,Y1):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm', ...
    'quasi-newton','HessUpdate','dfp', ...
    'PlotFcn',{@optimplotx,@optimplotfunccount,@optimplotfval, ...
    @optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [3,0];    %for x1,y1=(3,0)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Solver stopped prematurely.

fminunc stopped because it exceeded the function evaluation limit,
options.MaxFunctionEvaluations = 2.000000e+02.
```

```
x1y1a2i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 63
```

```
x1y1a2f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2.0676      0.13313
```

```
x1y1a2o=fval;
disp(['Objective value = ' num2str(fval)])
```

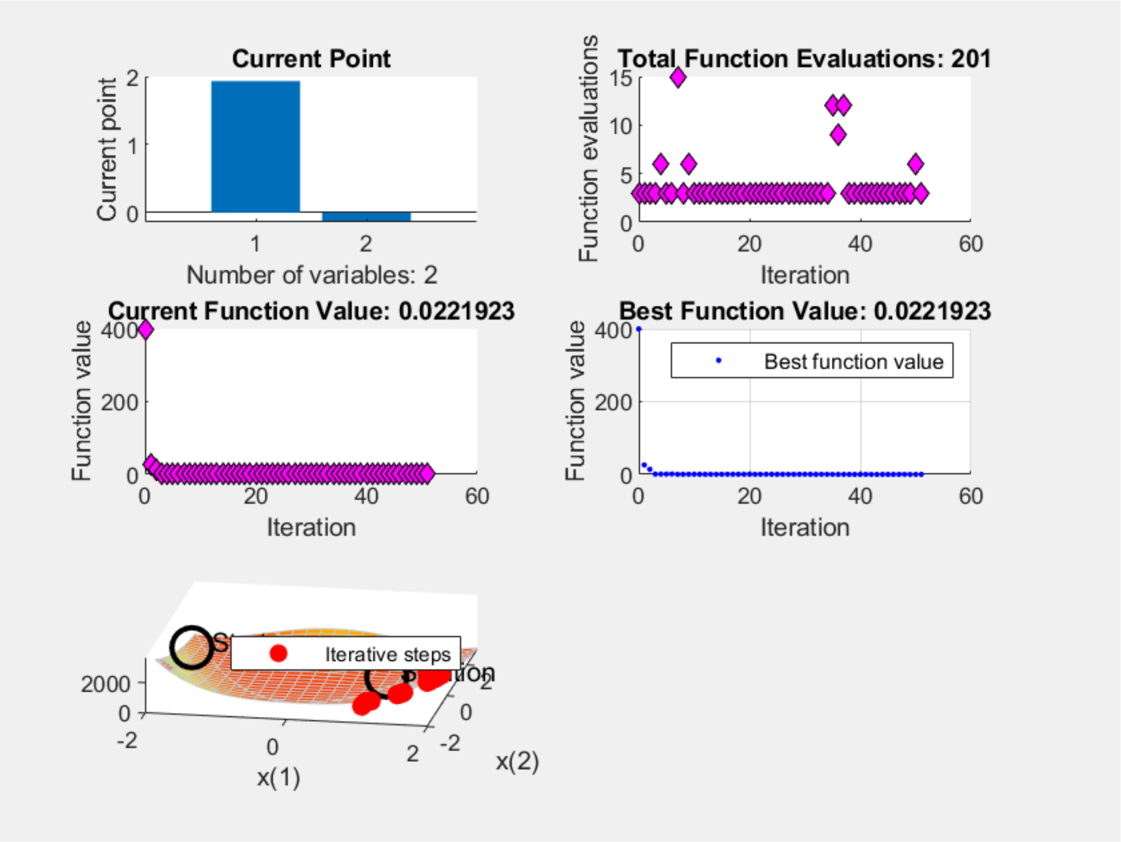
```
Objective value = 0.0090412
```

```
save ('parameters.mat','-append');
```

(X2,Y2):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm', ...
    'quasi-newton','HessUpdate','dfp', ...
    'PlotFcn',{@optimplotx,@optimplotfunccount,@optimplotfval, ...
    @optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [2,-2];    %for x2,y2=(2,-2)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Solver stopped prematurely.

fminunc stopped because it exceeded the function evaluation limit,
options.MaxFunctionEvaluations = 2.000000e+02.
```

```
x2y2a2i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 51
```

```
x2y2a2f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 1.9346    -0.13997
```

```
x2y2a2o=fval;
disp(['Objective value = ' num2str(fval)])
```

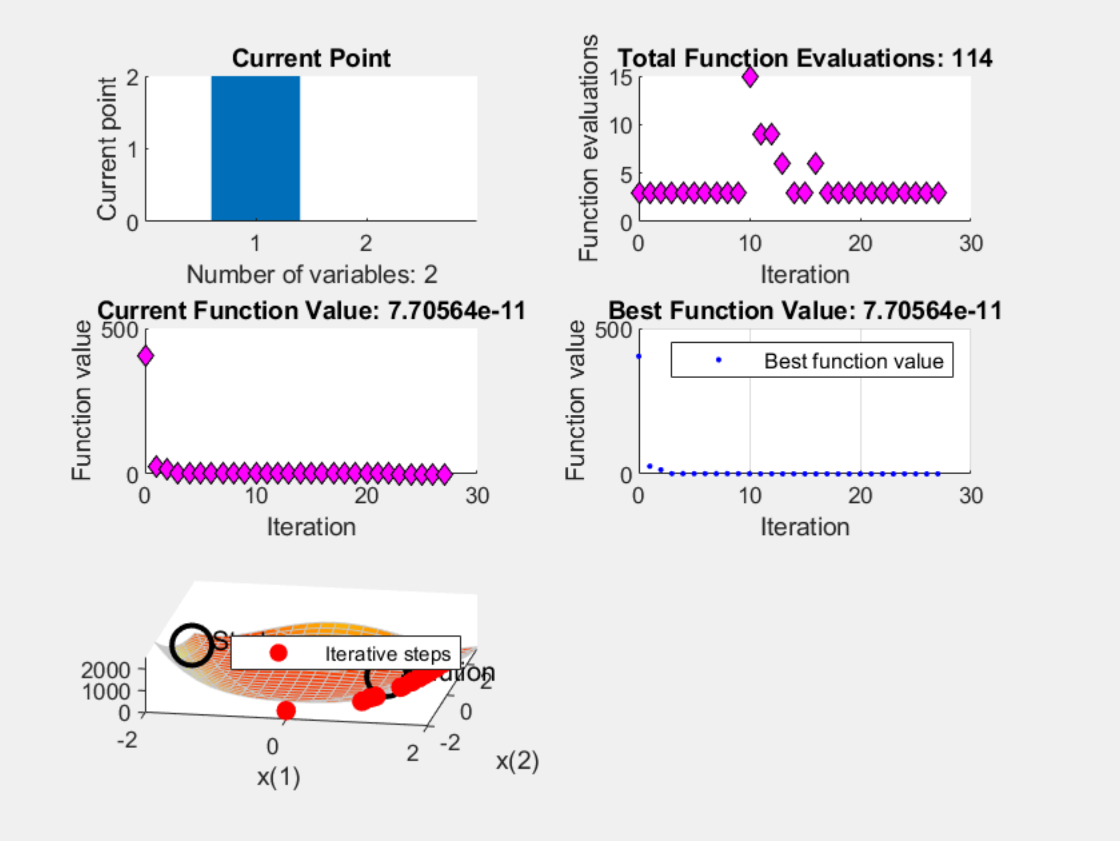
```
Objective value = 0.022192
```

```
save ('parameters.mat','-append');
```

(X3,Y3):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm', ...
    'quasi-newton','HessUpdate','dfp', ...
    'PlotFcn',{@optimplotx,@optimplotfunccount,@optimplotfval, ...
    @optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [0,-2];    %for x3,y3=(0,-2)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Local minimum found.
```

```
Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

```
x3y3a2i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 27
```

```
x3y3a2f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2 -1.6478e-05
```

```
x3y3a2o=fval;
disp(['Objective value = ' num2str(fval)])
```

```
Objective value = 7.7056e-11
```

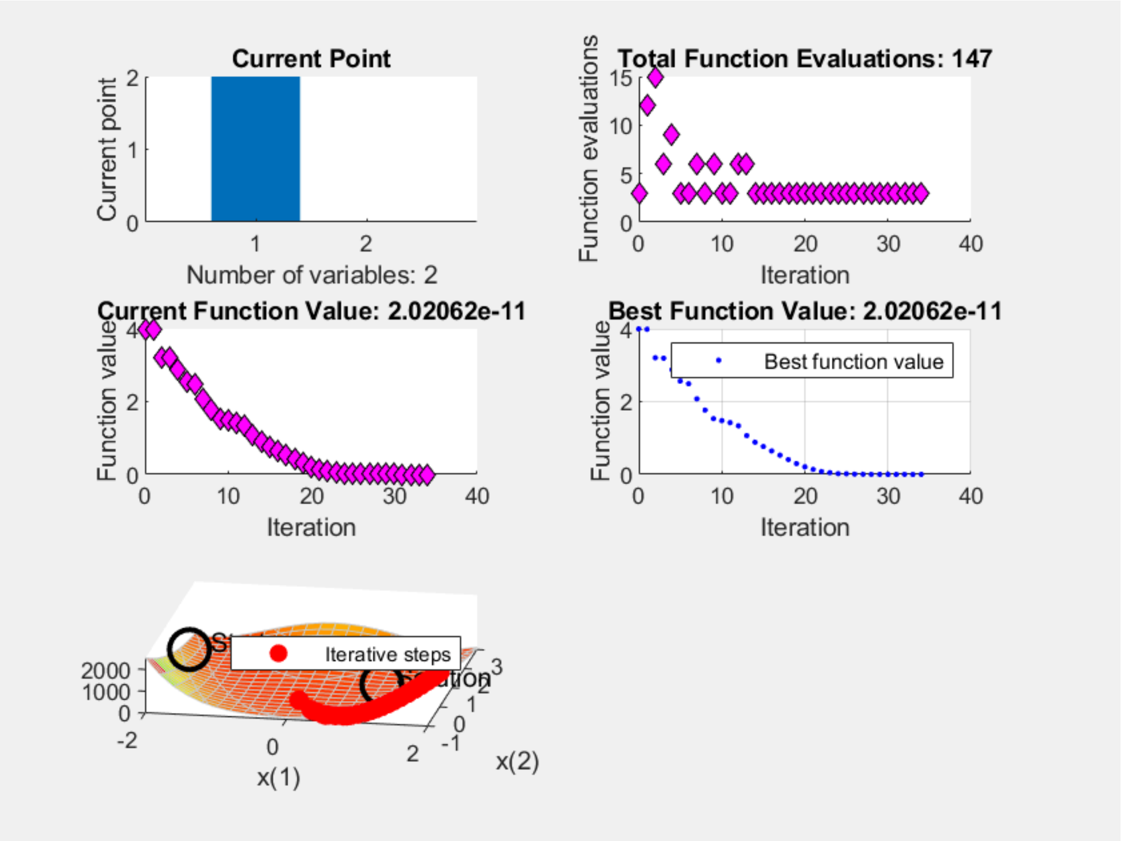
```
save ('parameters.mat','-append');
```

(X4,Y4):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm', ...
    'quasi-newton','HessUpdate','dfp','PlotFcn',{@optimplotx,@optimplotfunccount, ...
    @optimplotfval,@optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [0,0];    %for x4,y4=(0,0)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```

(X1,Y1):



```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

```
x4y4a2i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 34
```

```
x4y4a2f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2 -8.9798e-06
```

```
x4y4a2o=fval;
disp(['Objective value = ' num2str(fval)])
```

```
Objective value = 2.0206e-11
```

```
save ('parameters.mat','-append');
```

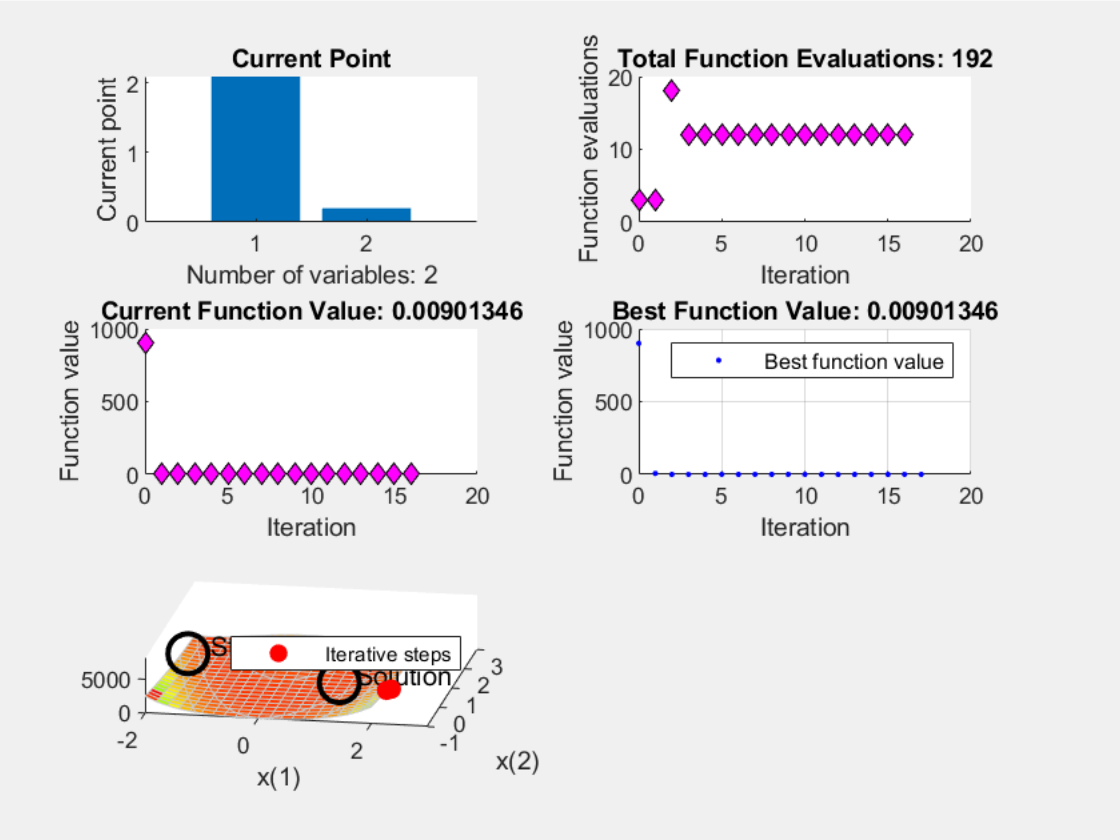
ALGORITHM:quassi-newton ; HESSIAN: steepest descent

(X1,Y1):

```
close all; clear; clc;
```

```
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm', ...
    'quasi-newton','HessUpdate','steepdesc','PlotFcn',{@optimplotx, ...
    @optimplotfunccount,@optimplotfval,@optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [3,0];    %for x1,y1=(3,0)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Solver stopped prematurely.

fminunc stopped because it exceeded the function evaluation limit,
options.MaxFunctionEvaluations = 2.000000e+02.
```

```
x1y1a3i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 17
```

```
x1y1a3f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2.0949      0.19892
```

```
x1y1a3o=fval;
disp(['Objective value = ' num2str(fval)])
```

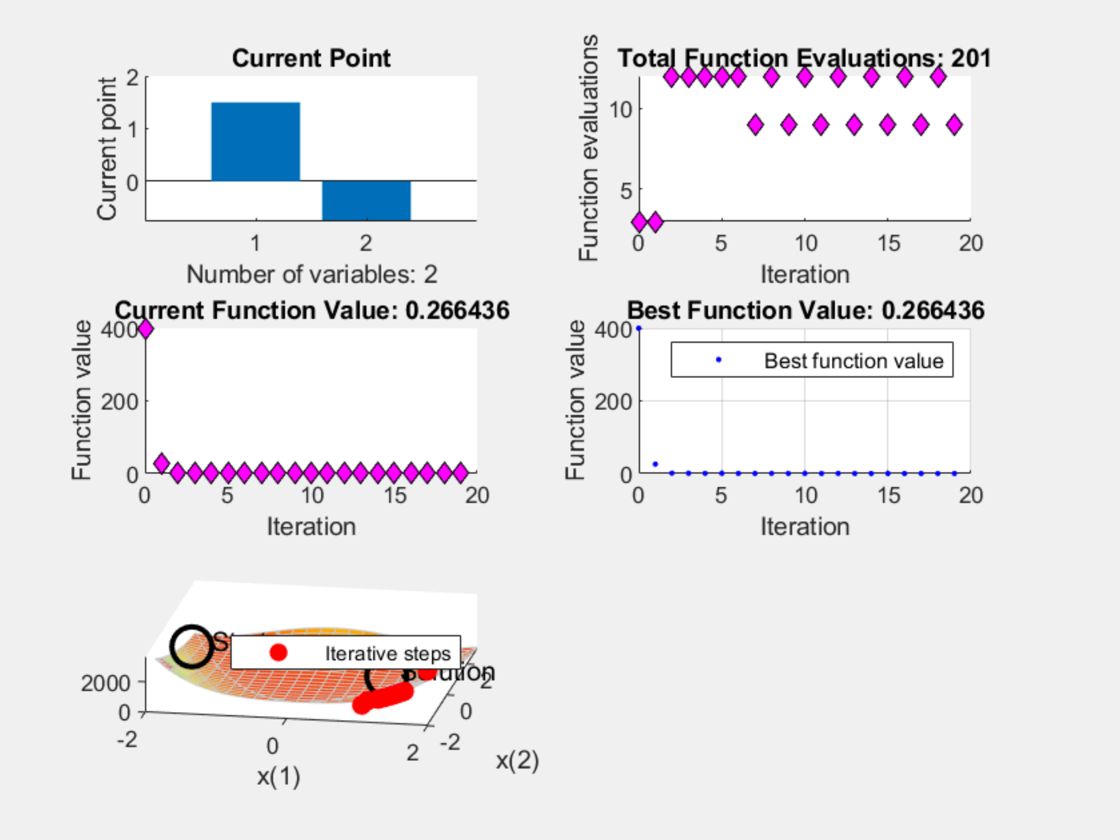
```
Objective value = 0.0090135
```

```
save ('parameters.mat','-append');
```

(X2,Y2):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm','quasi-newton', ...
    'HessUpdate','steepdesc','PlotFcn',{@optimplotx,@optimplotfunccount, ...
    @optimplotfval,@optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [2,-2];    %for x2,y2=(2,-2)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Solver stopped prematurely.

fminunc stopped because it exceeded the function evaluation limit,
options.MaxFunctionEvaluations = 2.000000e+02.
```

```
x2y2a3i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 19
```

```
x2y2a3f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 1.4986    -0.76364
```

```
x2y2a3o=fval;
```

```
disp(['Objective value = ' num2str(fval)])
```

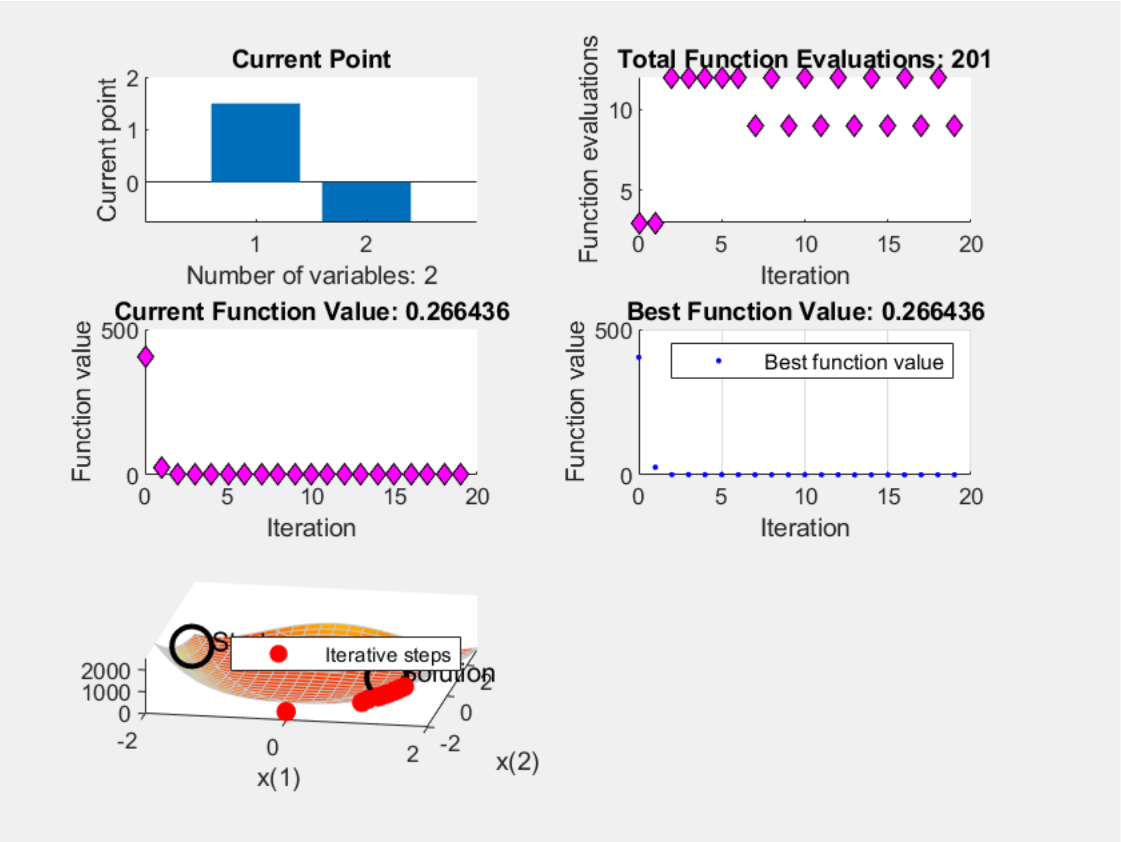
```
Objective value = 0.26644
```

```
save ('parameters.mat','-append');
```

(X3,Y3):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm','quasi-newton', ...
    'HessUpdate','steepdesc','PlotFcn',{@optimplotx,@optimplotfunccount, ...
    @optimplotfval,@optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [0,-2];    %for x3,y3=(0,-2)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Solver stopped prematurely.

fminunc stopped because it exceeded the function evaluation limit,
options.MaxFunctionEvaluations = 2.000000e+02.
```

```
x3y3a3i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 19
```

```
x3y3a3f=xy_opt;
```

```
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 1.4986     -0.76364
```

```
x3y3a3o=fval;
disp(['Objective value = ' num2str(fval)])
```

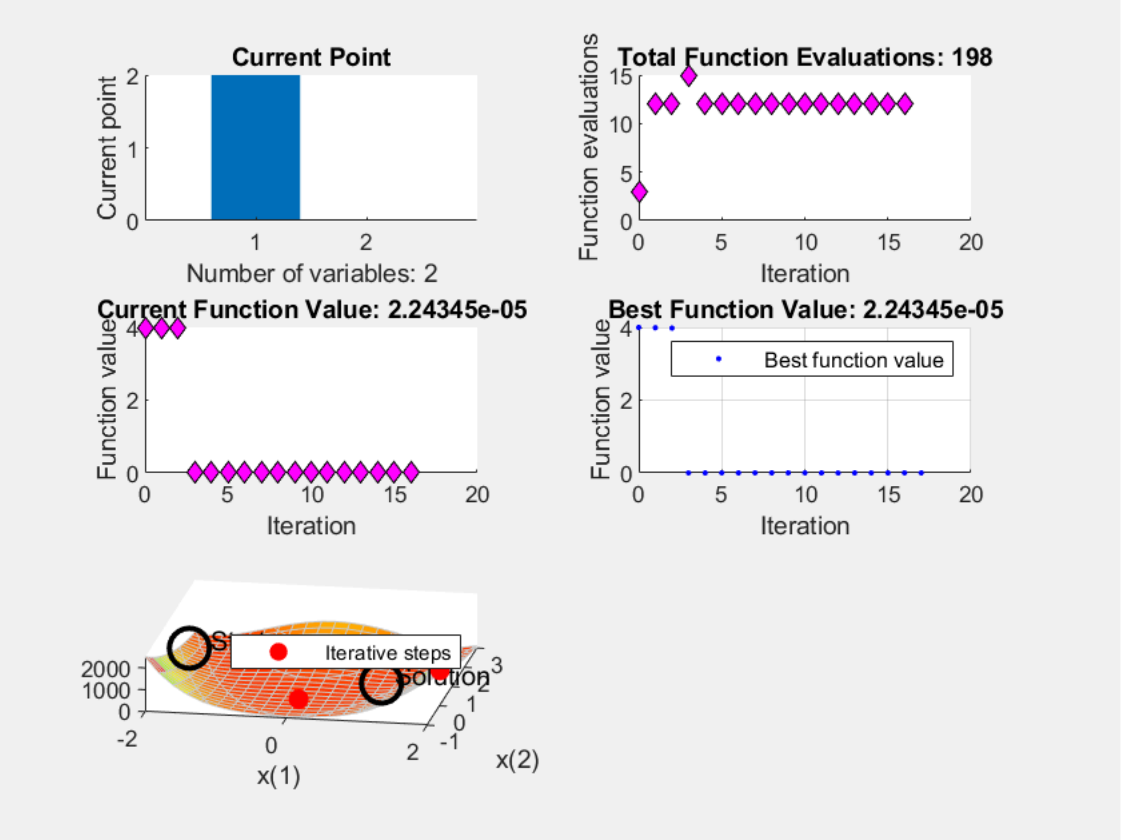
```
Objective value = 0.26644
```

```
save ('parameters.mat','-append');
```

(X4,Y4):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm','quasi-newton', ...
    'HessUpdate','steepdesc','PlotFcn',{@optimplotx,@optimplotfunccount, ...
    @optimplotfval,@optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [0,0];    %for x4,y4=(0,0)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrock_func,xy_val,options);
```



```
Solver stopped prematurely.

fminunc stopped because it exceeded the function evaluation limit,
options.MaxFunctionEvaluations = 2.000000e+02.
```

```
x4y4a3i=output.iterations;
```

```
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 17
```

```
x4y4a3f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2.0047    0.0095039
```

```
x4y4a3o=fval;
disp(['Objective value = ' num2str(fval)])
```

```
Objective value = 2.2434e-05
```

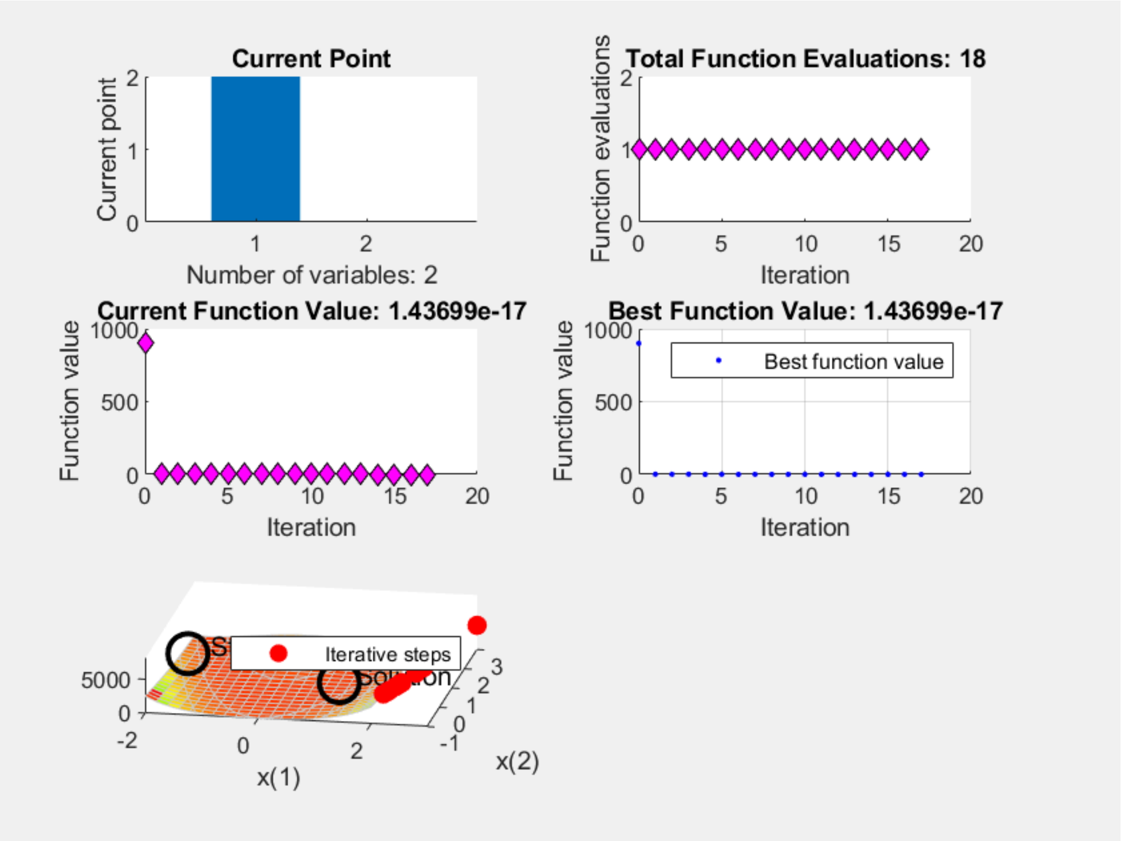
```
save ('parameters.mat','-append');
```

ALGORITHM:trust-region ;SPECIFY OBJECTIVE GRADIENT : true

(X1,Y1):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm','trust-region', ...
    'SpecifyObjectiveGradient',true,'PlotFcn',{@optimplotx,@optimplotfunccount, ...
    @optimplotfval,@optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [3,0];    %for x1,y1=(3,0)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrockwithgrad,xy_val,options);
```



```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

```
x1y1a4i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 17
```

```
x1y1a4f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2  7.0089e-09
```

```
x1y1a4o=fval;
disp(['Objective value = ' num2str(fval)])
```

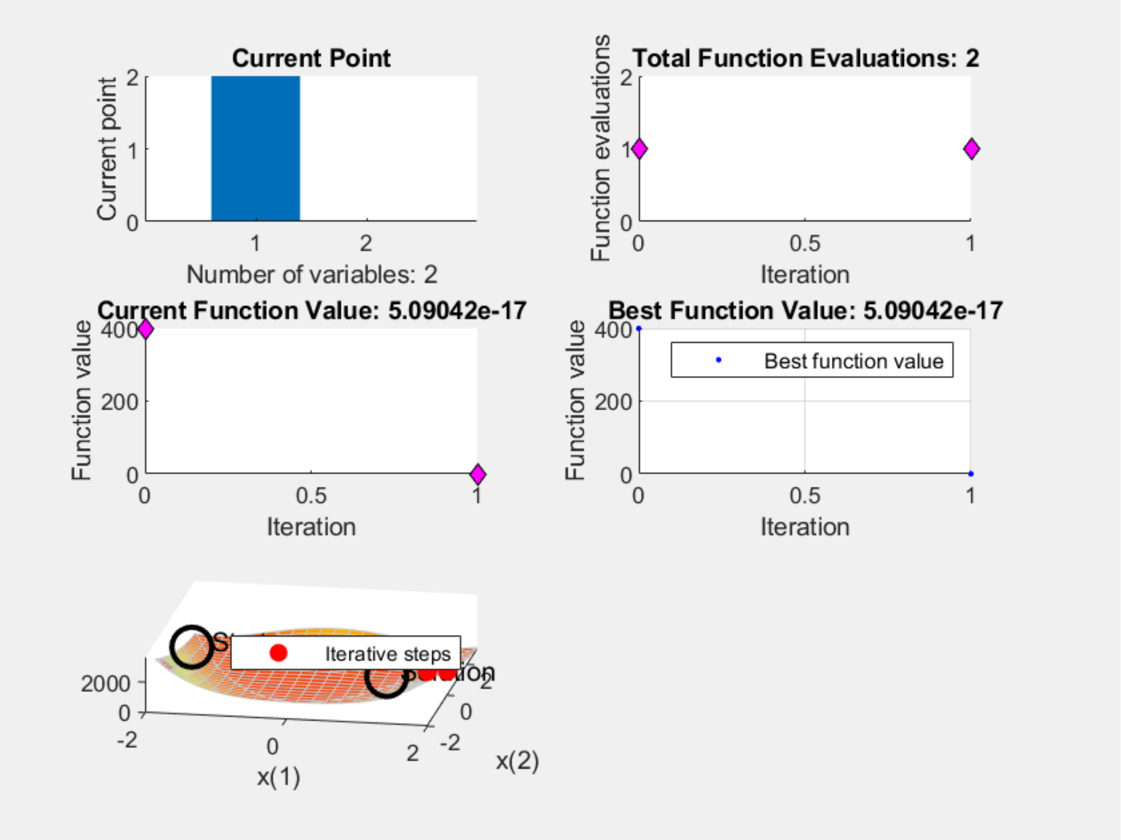
```
Objective value = 1.437e-17
```

```
save ('parameters.mat','-append');
```

(X2,Y2):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm','trust-region', ...
    'SpecifyObjectiveGradient',true,'PlotFcn',{@optimplotx,@optimplotfunccount, ...
    @optimplotfval,@optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [2,-2];    %for x2,y2=(2,-2)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrockwithgrad,xy_val,options);
```



```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

```
x2y2a4i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 1
```

```
x2y2a4f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2  1.4269e-08
```

```
x2y2a4o=fval;
disp(['Objective value = ' num2str(fval)])
```

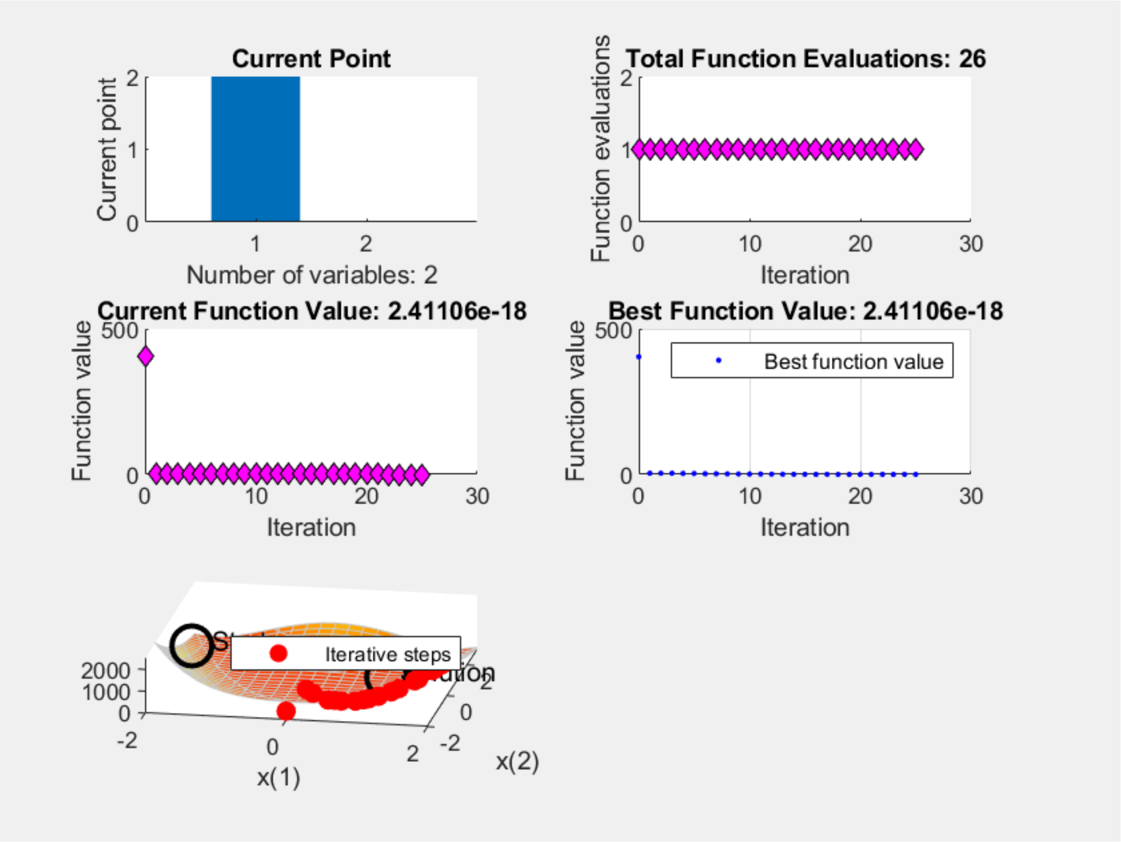
```
Objective value = 5.0904e-17
```

```
save ('parameters.mat','-append');
```

(X3,Y3):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm','trust-region', ...
    'SpecifyObjectiveGradient',true,'PlotFcn',{@optimplotx,@optimplotfunccount, ...
    @optimplotfval,@optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [0,-2];    %for x3,y3=(0,-2)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrockwithgrad,xy_val,options);
```



```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

```
x3y3a4i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 25
```

```
x3y3a4f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2 -2.9807e-09
```

```
x3y3a4o=fval;
```

```
disp(['Objective value = ' num2str(fval)])
```

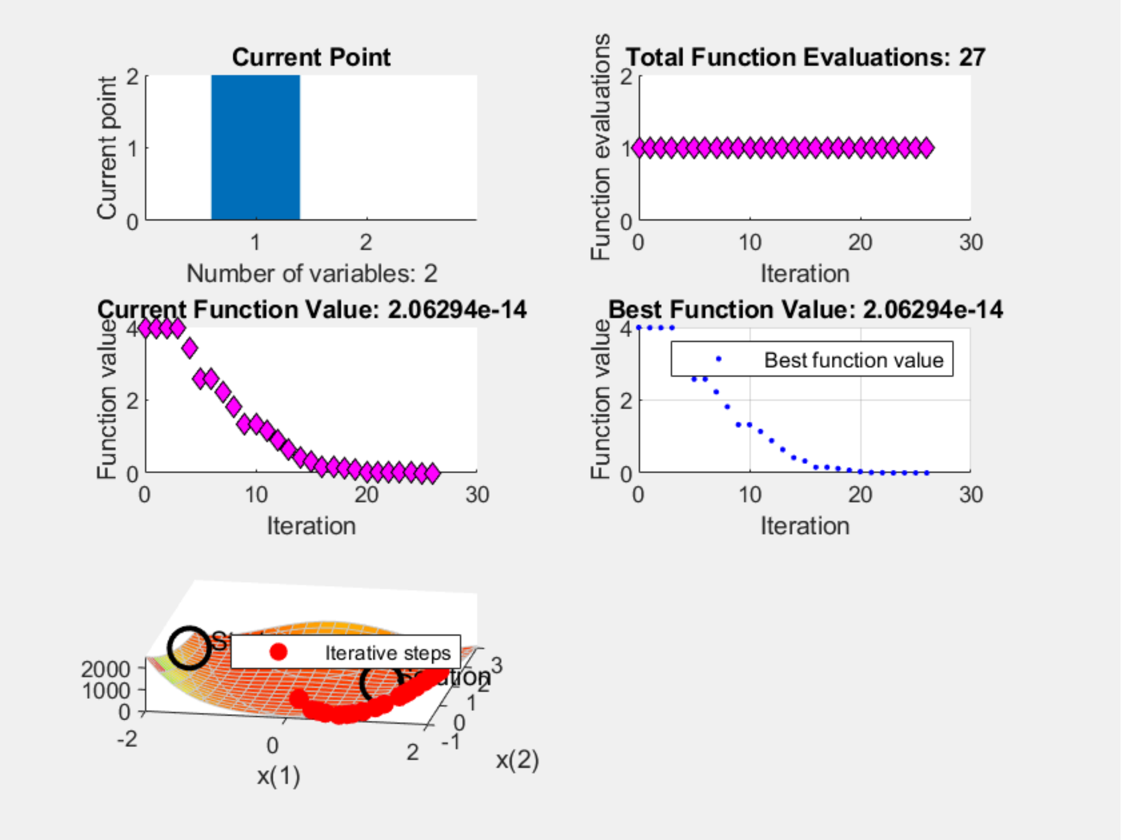
```
Objective value = 2.4111e-18
```

```
save ('parameters.mat','-append');
```

(X4,Y4):

```
close all; clear; clc;
% Setup optimization options
options = optimoptions(@fminunc,'Display','Final','Algorithm','trust-region', ...
    'SpecifyObjectiveGradient',true,'PlotFcn',{@optimplotx,@optimplotfunccount, ...
    @optimplotfval,@optimplotfvalconstr,@bananaout});

% Define xy values
xy_val = [0,0];    %for x4,y4=(0,0)
% Call optimization algorithm
[xy_opt,fval,eflag,output] = fminunc(@rosenbrockwithgrad,xy_val,options);
```



```
Local minimum possible.

fminunc stopped because the final change in function value relative to
its initial value is less than the value of the function tolerance.

<stopping criteria details>
```

```
x4y4a4i=output.iterations;
disp(['iterations = ' num2str(output.iterations)])
```

```
iterations = 26
```

```
x4y4a4f=xy_opt;
disp(['Fpoint = ' num2str(xy_opt)])
```

```
Fpoint = 2 -1.9943e-07
```

```
x4y4a4o=fval;
disp(['Objective value = ' num2str(fval)])
```

```
Objective value = 2.0629e-14
```

```
save ('parameters.mat','-append');
```

TABLES:

Number of iterations,final points and objective value for four initial points for algorithm 1(quassi-newton with hessian update= bfgs)

```
load parameters.mat;
table([X1;X2;X3;X4],[Y1;Y2;Y3;Y4],[x1y1a1i;x2y2a1i;x3y3a1i;x4y4a1i], ...
```

```
ans = 4×5 table
```

	X	Y	ITERATIONS:	FINAL POINT		OBJECTIVE VA...
1	3	0	18	2.0000	-1.3368e-05	4.4798e-11
2	2	-2	23	2.0000	8.1952e-06	1.6960e-11
3	0	-2	23	2.0000	-9.2745e-06	2.7044e-11
4	0	0	29	2.0000	-8.6524e-06	1.8756e-11

```
    [x1y1a1f;x2y2a1f;x3y3a1f;x4y4a1f],[x1y1a1o;x2y2a1o;x3y3a1o;x4y4a1o],'VariableNames', ...
    {'X','Y','ITERATIONS:','FINAL POINT','OBJECTIVE VALUE'})
```

Number of iterations,final points and objective value for four initial points for algorithm 2(quassi-newton with hessian update=dfp)

```
table([X1;X2;X3;X4],[Y1;Y2;Y3;Y4],[x1y1a2i;x2y2a2i;x3y3a2i;x4y4a2i], ...
```

```
ans = 4×5 table
```

	X	Y	ITERATIONS:	FINAL POINT		OBJECTIVE VA...
1	3	0	63	2.0676	0.1331	0.0090
2	2	-2	51	1.9346	-0.1400	0.0222
3	0	-2	27	2.0000	-0.0000	0.0000
4	0	0	34	2.0000	-0.0000	0.0000

```
    [x1y1a2f;x2y2a2f;x3y3a2f;x4y4a2f],[x1y1a2o;x2y2a2o;x3y3a2o;x4y4a2o],'VariableNames', ...
    {'X','Y','ITERATIONS:','FINAL POINT','OBJECTIVE VALUE'})
```

Number of iterations,final points and objective value for four initial points for algorithm 3(quassi-newton with hessian update=steepest descent)

```
table([X1;X2;X3;X4],[Y1;Y2;Y3;Y4],[x1y1a3i;x2y2a3i;x3y3a3i;x4y4a3i], ...
```

ans = 4×5 table

	X	Y	ITERATIONS:	FINAL POINT		OBJECTIVE VA...
1	3	0	17	2.0949	0.1989	0.0090
2	2	-2	19	1.4986	-0.7636	0.2664
3	0	-2	19	1.4986	-0.7636	0.2664
4	0	0	17	2.0047	0.0095	0.0000

```
    [x1y1a3f;x2y2a3f;x3y3a3f;x4y4a3f],[x1y1a3o;x2y2a3o;x3y3a3o;x4y4a3o],'VariableNames', ...
    {'X','Y','ITERATIONS:','FINAL POINT','OBJECTIVE VALUE'})
```

Number of iterations,final points and objective value for four initial points for algorithm 4(trust-region with specify objective gradient=true)

```
table([X1;X2;X3;X4],[Y1;Y2;Y3;Y4],[x1y1a4i;x2y2a4i;x3y3a4i;x4y4a4i], ...
```

ans = 4×5 table

	X	Y	ITERATIONS:	FINAL POINT		OBJECTIVE VA...
1	3	0	17	2.0000	7.0089e-09	1.4370e-17
2	2	-2	1	2.0000	1.4269e-08	5.0904e-17
3	0	-2	25	2.0000	-2.9807e-09	2.4111e-18
4	0	0	26	2.0000	-1.9943e-07	2.0629e-14

```
    [x1y1a4f;x2y2a4f;x3y3a4f;x4y4a4f],[x1y1a4o;x2y2a4o;x3y3a4o;x4y4a4o],'VariableNames', ...
    {'X','Y','ITERATIONS:','FINAL POINT','OBJECTIVE VALUE'})
```

FUNCTIONS USED:

Rosenbrock function

```
function f = rosenbrock_func(in)
    % Unpack inputs
    x = in(1);
    y = in(2);

    % The Rosenbrock function in 2D
    f=(2-x)^2+100*(y+1-(x-1)^2)^2;

end
```

Rosenbrock function with gradient

```
function [f,g] = rosenbrockwithgrad(in)
   % Unpack inputs
    x = in(1);
    y = in(2);

    % The Rosenbrock function in 2D
    f=(2-x)^2+100*(y+1-(x-1)^2)^2;

if nargout > 1 % gradient required
   % g = [-400*(y-x^2)*x-2*(1-x);200*(y-x^2)];
    g=[-2*(2-x)-400*(y+1-(x-1)^2)*(x-1);200*(y+1-(x-1)^2)];
end
end
```

CONCLUSION

For the given constraints a=1,b=-1, the minimum value was observed for all the points, when we were using **Trust-Region algorithm** with the **specify objective gradient='True'** with the most **minimal iteration of 1** which is less compared to the others inorder to find the minimum of the rosenbrock function . The least value was observed for(x2,y2) and it was **5.0904e-17.**

The maximum number of iterations were observed for Quasi-Newton Algorithm having Inverse Hessian Update and it was maximum of upto **63 iterations.**