Optimization Methods for Civil Engineering

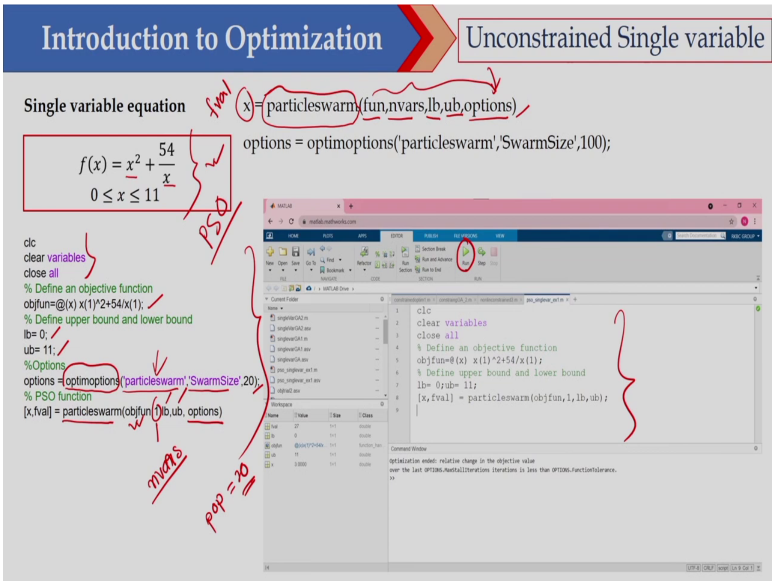
Prof. Rajib Kumar Bhattacharjya
Department of Civil Engineering
Indian Institute of Technology, Guwahati

Lecture - 35
Introduction to Optimization using MATLAB

Hello student, welcome back to the course on Optimization Methods for Civil Engineering. So, in the last class we have solved some optimization problems. So, mainly single variable optimization problem and multivariable optimization problem with constrain or without constrain using classical optimization methods in MATLAB.

So, in today's class we will solve the same problem whatever we have solved already using classical optimization methods. So, those problem we will solve using GA and particles from optimization. So, these two are two metaheuristic optimization methods. So, we will use initially particle swarm optimization PSO for solving those problems and then we will apply GA genetic algorithm for solving those problem in MATLAB.

(Refer Slide Time: 01:27)



Introduction to Optimization

Unconstrained Single variable

Single variable equation

x = particleswarm(fun,nvars,lb,ub,options)

options = optimoptions('particleswarm','SwarmSize',100);

$$f(x) = x^2 + \frac{54}{x}$$

$$0 \le x \le 11$$

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;
ub= 11;
%Options
options = optimoptions('particleswarm','SwarmSize',20);
% PSO function
[x,fval] = particleswarm(objfun,1,lb,ub, options)
```

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;ub= 11;
[x,fval] = particleswarm(objfun,1,lb,ub);
```

Name	Value	Size	Class
fval	27	1x1	double
lb	0	1x1	double
objfun	@(x)x(1)^2+54/x...	1x1	function_han...
ub	11	1x1	double
x	3.0000	1x1	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
>>
```

Let us see the first problem. So, it is a single variable problem and without constrain. So, there is no constrain here only we have that objective function and the problem is minimization type. The function here is f of x equal to x square plus 54 by x. So, I would like to solve this problem using PSO Particle Swarm Optimization. So, in MATLAB so, there is a function particle swarm so this function can be used for solving an optimization problem using particle swarm optimization method.

So, here this is the function and these are the argument of this particular function. So, what you have to do we have to define the objective function we have to define the number of variables and then we have to define lower bound and we have to define upper bound and there are some option we can define. So, these are the arguments, which is coming into this particular function.

And from this particular function, so we will get the optimal value of this problem or we will get the optimal solution. And we will also get the optimal function value ok. So, here I have shown you the MATLAB code for solving this problem using particle swarm.

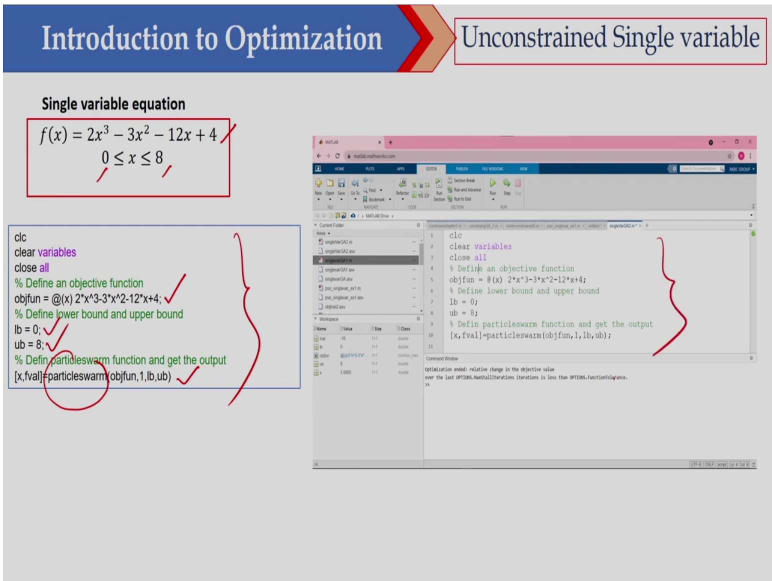
So, here these are the 3 lines so which will clear the environment and then I have define the objective function just like we have defined in case of classical optimization technique. And then you define what is lower bound what is upper bound. So, in this case lower bound is 0 upper bound is 11. And then I have defined the options here, so you can option you can define this optimoptions.

So, this is the function to define option. And here this is particle swarm and this is the actually function I am using particle swarm and SwarmSize is 20 that means the population here is 20, I have defined here so you can change it. So, you can also define the other parameters of this particular algorithms here. And after that I have use this one so, I have already defined objective function. And number of variable here is your this is nvars ok. So, number of variables.

So, that is one. And we have already defined lower bound, we have already defined upper bound and we have defined options. And then if I execute this particular m file then I will get the solution of this particular problem ok optimal solution of this particular problem. And here you can see so this is the MATLAB code and if I execute this one using the run button so then I will get the optimal solution of this problem.

So, I will show you in MATLAB, how you can, how you can means run this particular code. So, that I will show you. Before that let me explain the other problem that I will be solving in today’s class.

(Refer Slide Time: 04:47)



Introduction to Optimization

Unconstrained Single variable

Single variable equation

$$f(x) = 2x^3 - 3x^2 - 12x + 4$$
$$0 \le x \le 8$$

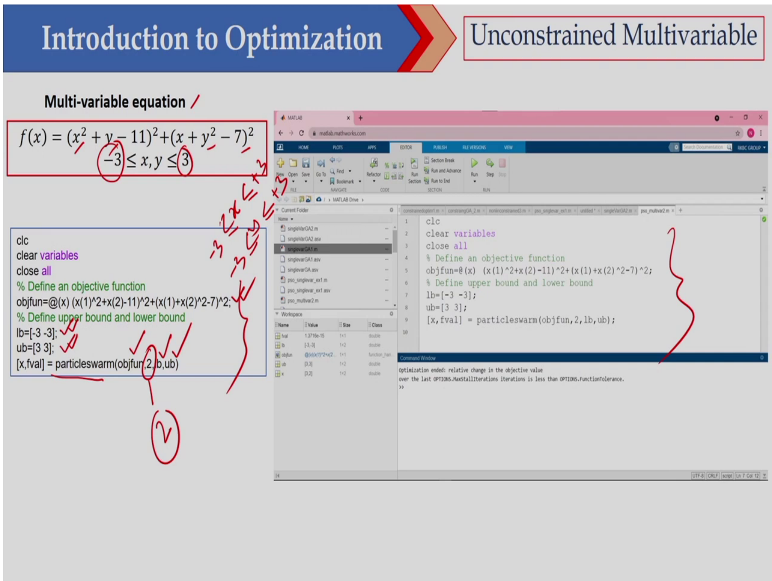
```
clc
clear variables
close all
% Define an objective function
objfun = @(x) 2*x^3-3*x^2-12*x+4;
% Define lower bound and upper bound
lb = 0;
ub = 8;
% Defin particleswarm function and get the output
[x,fval]=particleswarm(objfun,1,lb,ub)
```

```
clc
clear variables
close all
% Define an objective function
objfun = @(x) 2*x^3-3*x^2-12*x+4;
% Define lower bound and upper bound
lb = 0;
ub = 8;
% Defin particleswarm function and get the output
[x,fval]=particleswarm(objfun,1,lb,ub);
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
>>
```

The second problem is also a single variable optimization problem. So, we have only one variable that is x and the function is twice x cube minus thrice x square minus 12 x plus 4 and lower bound is 0 upper bound is 8. So, this is the optimization code. And here what you have to define, you have to define the objective function lower bound upper bound and then I am using particle swarm function to solve this problem using particle swarm optimization. So, this is the MATLAB code of this particular problem.

(Refer Slide Time: 05:26)



Introduction to Optimization

Unconstrained Multivariable

Multi-variable equation

$$f(x) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$
$$-3 \le x, y \le 3$$

$-3 \le x \le +3$

$-3 \le y \le +3$

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-3 -3];
ub=[3 3];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

2

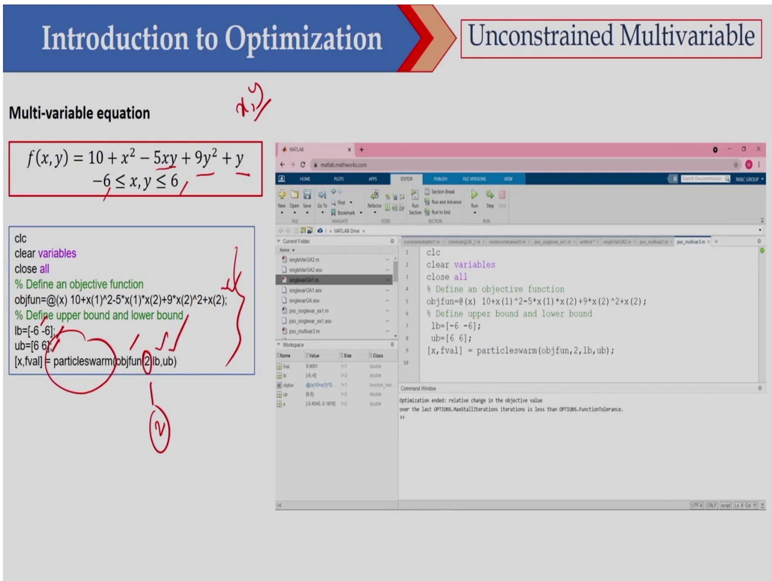
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-3 -3];
ub=[3 3];
[x,fval] = particleswarm(objfun,2,lb,ub);
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
>>
```

Now, let us see a multivariable problem. So, we do not have any constrain here so we have only the objective function. So, here we have two variables that is x and y. And the function is x square plus y minus 11 whole square plus x plus y square minus 7 whole square. And the range is between minus 3 and plus 3 ok. So, value of the x is between minus 3 and plus 3 ok and y is also between minus 3 and plus 3 ok.

And this is the, this is the MATLAB code. So, here I am defining the objective function and I am defining lower bound and upper bound so lower bound is minus 3 minus 3 upper bound is 3 3 and then you are using particle swarm and you are defining objective function. Now, my variable is 2 ok, so variable is 2 and then you define lower bound upper bound and if you solve it you will get the optimal value of this function.

(Refer Slide Time: 06:37)



Introduction to Optimization

Unconstrained Multivariable

Multi-variable equation

x, y

$$f(x,y) = 10 + x^2 - 5xy + 9y^2 + y$$
$$-6 \le x, y \le 6$$

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6 -6];
ub=[6 6];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

2

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
 lb=[-6 -6];
 ub=[6 6];
[x,fval] = particleswarm(objfun,2,lb,ub);
```

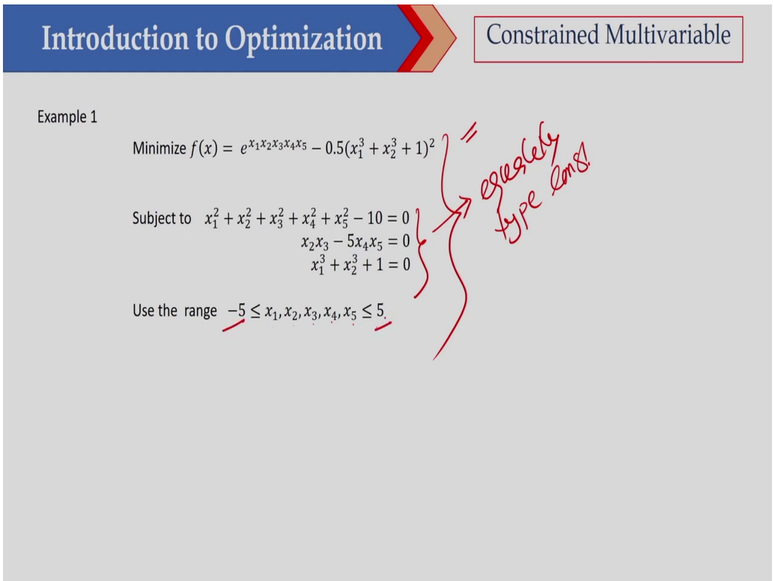
Command Window

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
>>
```

Name	Value	Size	Class
fval	9.9091	1x1	double
lb	[-6,-6]	1x2	double
objfun	@(x)10+x(1)^2-...	1x1	function_han...
ub	[6,6]	1x2	double
x	[-0.4545,-0.1818]	1x2	double

Now, let us see the next function and this is also a two variable problem; that means, we have x and y. And the function is 10 plus x square minus 5 x y plus 9 y square plus y, and range for x and y is minus 6 and plus 6. And this is the MATLAB code here I have defined the objective function and here the lower bound is minus 6 and minus 6 and upper bound is 6 6 and then you are using particle swarm ok. Objective function and here number of variable is again 2 lower bound and upper bound we have defined and I can get the solution of this particular problem.

(Refer Slide Time: 07:24)



Introduction to Optimization

Constrained Multivariable

Example 1

Minimize $f(x) = e^{x_1 x_2 x_3 x_4 x_5} - 0.5(x_1^3 + x_2^3 + 1)^2$

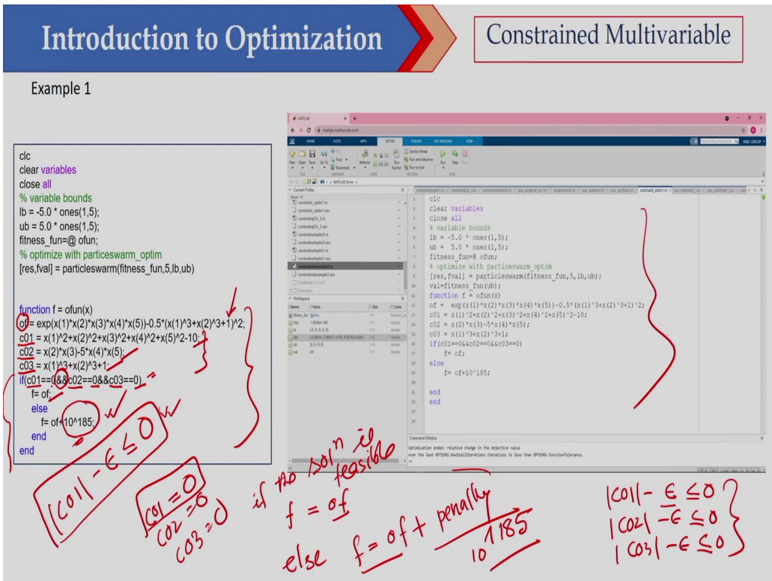
Subject to
$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$
$$x_2 x_3 - 5x_4 x_5 = 0$$
$$x_1^3 + x_2^3 + 1 = 0$$

Use the range $-5 \leq x_1, x_2, x_3, x_4, x_5 \leq 5$

= equality type cons.

Now, let us see a problem with constrain now. So, we have objective function and there are 3 equality type constrains so this is equality type constrain ok so type constrain ok. And the range for x 1 x 2 x 3 x 4 and x 5 is between minus 5 and plus 5, ok. So, this is a complicated problem so this problem we have solved using classical optimization method and here we will solve this problem using particle swarm optimization.

(Refer Slide Time: 08:01)



Introduction to Optimization

Constrained Multivariable

Example 1

```
clc
clear variables
close all
% variable bounds
lb = -5.0 * ones(1,5);
ub = 5.0 * ones(1,5);
fitness_fun=@ ofun;
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,5,lb,ub)


function f = ofun(x)
of= exp(x(1)*x(2)*x(3)*x(4)*x(5))-0.5*(x(1)^3+x(2)^3+1)^2;
c01 = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2-10;
c02 = x(2)*x(3)-5*x(4)*x(5);
c03 = x(1)^3+x(2)^3+1;
if(c01==0&&c02==0&&c03==0)
    f= of;
    else
      f= of+10^185;
    end
end
```

Optimization ended: relative change in the objective value over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

$|c01| - \epsilon \le 0$

$c01 = 0$
$c02 = 0$
$c03 = 0$

if the sol^n is feasible
$f = of$
else $f = of + \text{penalty}$
10^{185}

$$\left.\begin{array}{l} |c01| - \epsilon \le 0 \\ |c02| - \epsilon \le 0 \\ |c03| - \epsilon \le 0 \end{array}\right\}$$

So, here what you have to do we have constrain now ok so, we have constrain. Now, in the objective function I am using the penalty approach. So, what I am doing here. So, if the solution is feasible then the fitness value if I say here fitness value will be the objective function value.

If the solution is feasible, if the solution is feasible if the solution is feasible then f equal to objective function value, else f equal to objective function plus I am using some penalty ok. And that penalty I am putting it 10 to the power it is a very high number I am putting 10 to the power 185 ok so this number I am putting.

And so, because I would like to avoid this solution the infeasible solution. So, how I am doing that? That you have written the objective function. So, this is objective function. Then calculate the value of first constrain, second constrain, third constrain. So, this is I have written 3 lines so, the first one is first constrain, then second constrain, the third constrain.

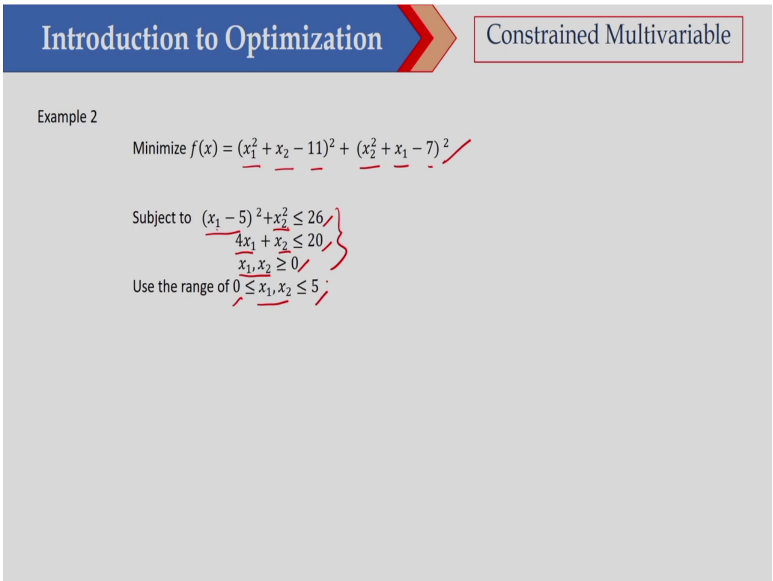
Now I have used the, if else statement here and if. So, the constrain should be equal to 0 because C1 01 should be equal to 0 C02 equal to 0 and C03 equal to 0. So, I have used if else statement so, if C01 equal to 0 and I have used n; that means, C02 equal to 0 and C03 equal to 0 ok.

So, in that case else f equal to objective function plus 10 to the power 185; that means, it is a very huge number I am putting. So, let us see whether I will get the solution or not. And if you are suppose sometime what may happen you may not get the solution. So, sometime it may be difficult to satisfy this equality constrain so in that case what I will do that C01 the absolute value of this one minus epsilon can be less than equal to 0.

So, I can convert this equality constrain to an inequality constrain. So, therefore, if the absolute value of C01 is less than epsilon so in that case it is fine. So, similarly I can I can convert the rr equation suppose, C01 minus epsilon is less than equal to 0 then C02 minus epsilon less than equal to 0 and C03 minus epsilon less than equal to 0. So, we can convert this equality constrain to inequality constrain so here what I what we can do we can actually allow some error ok.

So, we can allow some error and if the error is less than epsilon value so, in that case I can say that constrain has been satisfied, but in this case I have put that C1 equal to 0 C2 equal to 0 and C3 equal to 0. So, if the solution is feasible and if the solution is not feasible so in that case this will be equal to this will be equal to very large number. And this is the MATLAB code and once you are using this one. So, you should get the solution of this particular problem.

(Refer Slide Time: 11:51)



Introduction to Optimization

Constrained Multivariable

Example 2

Minimize $f(x) = (x_1^2 + x_2 - 11)^2 + (x_2^2 + x_1 - 7)^2$

Subject to $(x_1 - 5)^2 + x_2^2 \le 26$

$4x_1 + x_2 \le 20$

$x_1, x_2 \ge 0$

Use the range of $0 \le x_1, x_2 \le 5$

So, let us see another problem with constrain. So, in that case we have inequality constrain the earlier problem we have equality constrain, but here we have inequality constrain. So, let us see this problem. So, here this is the objective function that x 1 square plus x 2 minus 11 whole square, then x 2 square plus x 1 minus 7 whole square.

So, here there are two constrains the first one is that x 1 minus 5 whole square plus x 2 square less than equal to 26 and 4 x 1 plus x 2 less than equal to 20. And both x 1 and x 2 are positive and the range is that x 1 and x 2 are between 0 and 5, ok. So, this is the solution.

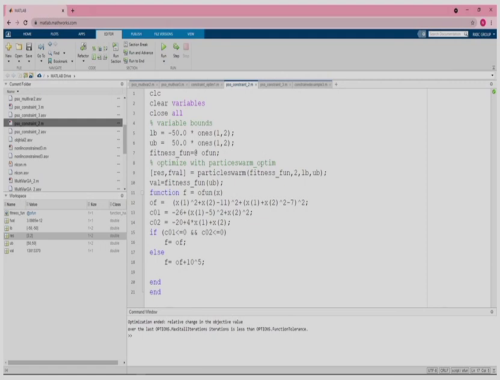
(Refer Slide Time: 12:45)

Introduction to Optimization

Constrained Multivariable

Example 2

```
clc
clear variables
close all
% variable bounds
lb = -50.0 * ones(1,2);
ub = 50.0 * ones(1,2);
fitness_fun=@ ofun;
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub)


function f = ofun(x)
of = (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
c01 = -26+(x(1)-5)^2+x(2)^2;
c02 = -20+4*x(1)+x(2);
if (c01<=0 && c02<=0)
   f= of;
   else
      f= of+10^5;
   end
end
```



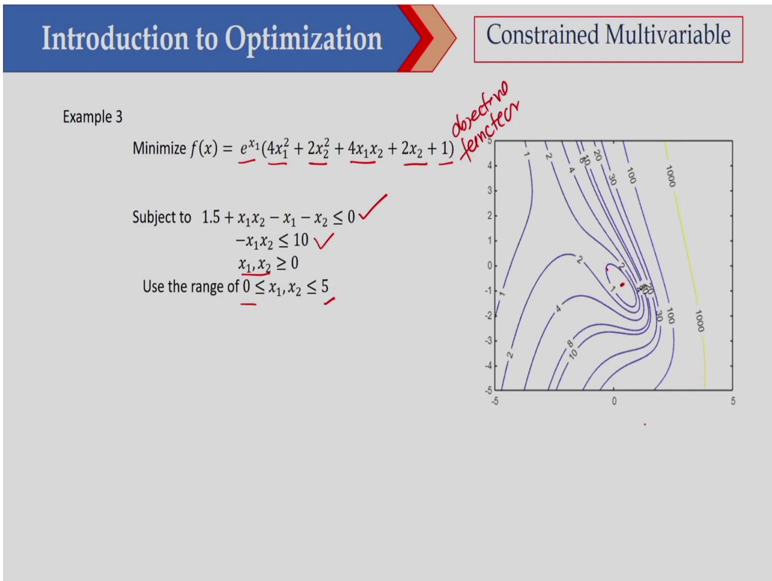
```
clc
clear variables
close all
% variable bounds
lb = -50.0 * ones(1,2);
ub =  50.0 * ones(1,2);
fitness_fun=@ ofun;
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub);
val=fitness_fun(ub);
function f = ofun(x)
of =  (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
c01 = -26+(x(1)-5)^2+x(2)^2;
c02 = -20+4*x(1)+x(2);
if (c01<=0 && c02<=0)
    f= of;
else
    f= of+10^5;

end
end
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
>>
```

And what I have done here? So, here also I have defined the objective function, this is the objective function and then this is constrain 1 and this is constrain 2. So, now in this case that this constrain should be less than equal to 0 and this is also less than equal to 0. If it is 0; that means, the solution is feasible then f equal to the objective function value, else f equal to objective function value plus a large number.

So, I have calculated the fitness function using this function ok and after that I have used particle swarm ok. And here this is the fitness function and then we have two variable and lower bound and upper bound. And I have already defined what is lower bound and what is upper bound and fitness function equal to objective function so that has been defined and if you are executing this code so you will get the solution of this problem.

(Refer Slide Time: 13:45)



Introduction to Optimization

Constrained Multivariable

Example 3

Minimize $f(x) = \underline{e^{x_1}}(\underline{4x_1^2} + \underline{2x_2^2} + \underline{4x_1x_2} + \underline{2x_2} + \underline{1})$ objective function

Subject to $1.5 + x_1x_2 - x_1 - x_2 \leq 0$ ✓

$-x_1x_2 \leq 10$ ✓

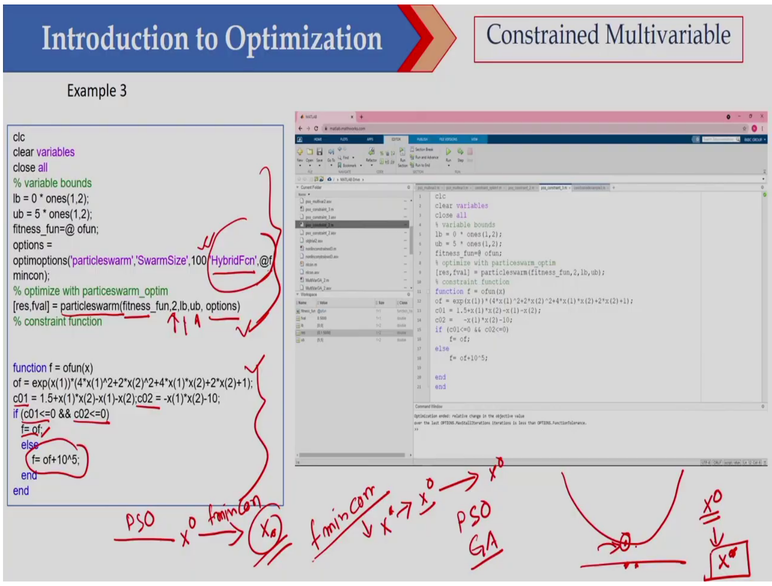
$\underline{x_1, x_2} \geq 0$

Use the range of $\underline{0} \leq x_1, x_2 \leq 5$

Now, let us look at another problem with constrain. So, the problem is that e to the power x 1 into 4 x 1 square plus twice x 2 square plus 4 x 1 x 2 plus twice x 2 plus 1 so, this is the objective function, this is the objective function. And we have two constrains so, this is the first one and this is the second one. And then x 1 and x 2 are positive and it is between 0 and 5 ok.

So, unconstrained optimization solution is somewhere here ok. So, this is the contour plot of this particular objective function. So, somewhere here and this is at and but the constrain solution will be somewhere here ok.

(Refer Slide Time: 14:43)



Introduction to Optimization

Constrained Multivariable

Example 3

```
clc
clear variables
close all
% variable bounds
lb = 0 * ones(1,2);
ub = 5 * ones(1,2);
fitness_fun=@ ofun;
options =
optimoptions('particleswarm','SwarmSize',100,'HybridFcn',@f
mincon);
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub, options)
% constraint function


function f = ofun(x)
of = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
c01 = 1.5+x(1)*x(2)-x(1)-x(2);c02 = -x(1)*x(2)-10;
if (c01<=0 && c02<=0)
  f= of;
  else
   f= of+10^5;
  end
end
```

So, let us see the code here again I have written this fitness function or objective function here, this is the objective function then these are two constrain that is c01 and c02. And in this case the c01 and c02 both are less than equal to 0 less than equal to 0.

So, if it is satisfied; that means, the solution is feasible then f equal to of or otherwise it is a large number in order to avoid that solution. And here I have defined the MATLAB code for solving this problem. So, here I have used optimization and swarm size I have used 100 and I have also use hybridization. So, that also I will discuss.

So, what will happen sometime what happen when you are using PSO algorithm or GA. So, you may not get the exact optimal solution suppose, I would like to solve this problem so, PSO and GA. So, this is the actual solution of this problem, but PSO is giving a near optimal solution so somewhere here.

So, what you can do I can take whatever solution you are getting after PSO run. So, that solution I can take as initial solution and I can refine using a classical method in order to get the exact optimal solution of this problem and this method is known as hybrid optimization method.

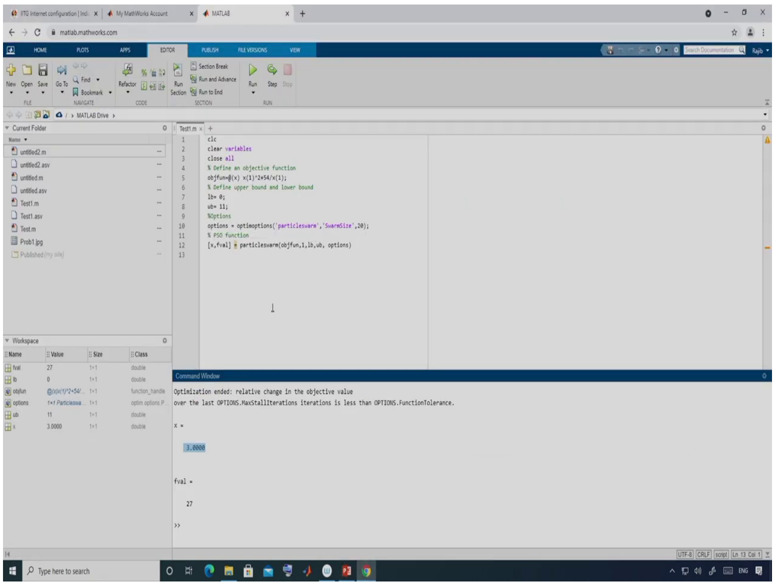
So, what we are doing initially we are using some metaheuristic optimization method and we are getting suppose a near optimal solution and that near optimal solution we are taking as an initial solution and then we apply the classical method. Because in the classical method we are using the gradient information and we can find the exact optimal solution of the problem. And this is known as hybridization.

So, here what you can do that if you are writing that hybrid function so; that means, this will go for yeah hybrid method and here I have written that hybrid function is fmincon so fmincon ok so this is a classical method. So, what you are doing so whatever solution you are getting using PSO that solution will be used as a initial optimal solution, then fmincon will be used to get the actual solution.

So, idea is that so you have you have used PSO ok, you got this particular solution or you can say and after that taking this initial solution you are using fmincon ok and you are getting the actual optimal solution ok. So, you are getting the actual optimal solution. So, in this case I have used the hybrid method. So, otherwise if you are not writing this one then hybrid method will not be used. So, you are only getting the solution obtained by the particle swarm optimization.

And then you are using particle swarm optimization so you are writing fitness the number of variable is 2, then lower bound then upper bound and then option ok. So, I will also show you how you can solve this problem using hybrid method. So, now, let us try and let us solve this problem using MATLAB ok. So, I will use the online version of MATLAB and let us see I will solve the first problem to the last problem we have discuss using particle swarm optimization.

(Refer Slide Time: 18:34)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;
ub= 11;
%Options
options = optimoptions('particleswarm','SwarmSize',20);
% PSO function
[x,fval] = particleswarm(objfun,1,lb,ub, options)
```

Workspace

Name	Value	Size	Class
fval	27	1×1	double
lb	0	1×1	double
objfun	@(x)x(1)^2+54/...	1×1	function_handle
options	1×1 Particleswa...	1×1	optim.options.P...
ub	11	1×1	double
x	3.0000	1×1	double

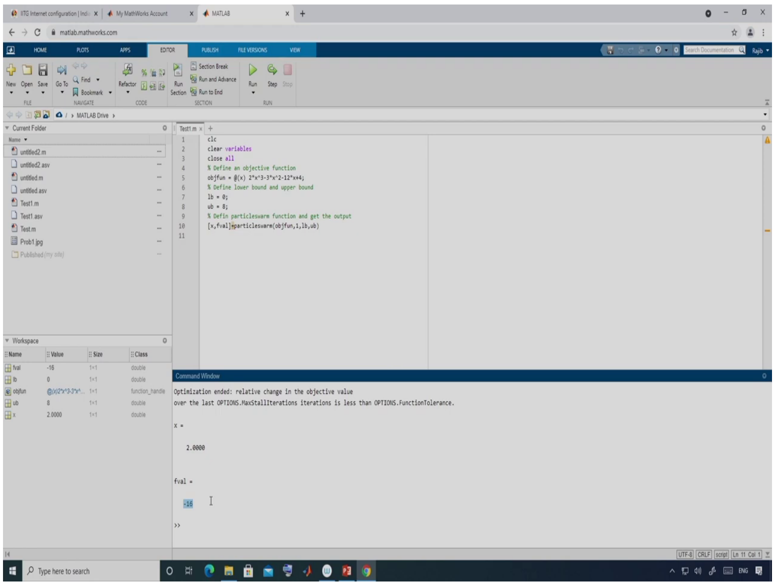
```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

    3.0000


fval =

    27

>>
```

So, let us take the first problem. So, this is the function so here you can see I have defined the objective function here and these three lines just to clean the environment. And then I have defined the objective function and then lower bound and upper bound and then option I have defined. So, swamp size I have defined 20. And then I have used this particular line to run the particles swarm optimization.

So, here what you have to define, you have to define the objective function. And then number of variable is 1 then lower bound, then upper bound and then option. So, if you are running this this few lines. So, hardly you just see there are only 7 to 8 lines and then if you execute this particular m file so, you will get the solution of this particular function. So, let us see.

So, I got the solution the solution is x equal to 3 and function value is 27. This problem also we have solved using the classical optimization methods and you got the same solution that is 3 and 27, ok. Now, let us see the second one, the second one is this particular function and let us solve this problem.

(Refer Slide Time: 19:58)



```
clc
clear variables
close all
% Define an objective function
objfun = @(x) 2*x^3-3*x^2-12*x+4;
% Define lower bound and upper bound
lb = 0;
ub = 8;
% Defin particleswarm function and get the output
[x,fval]=particleswarm(objfun,1,lb,ub)
```

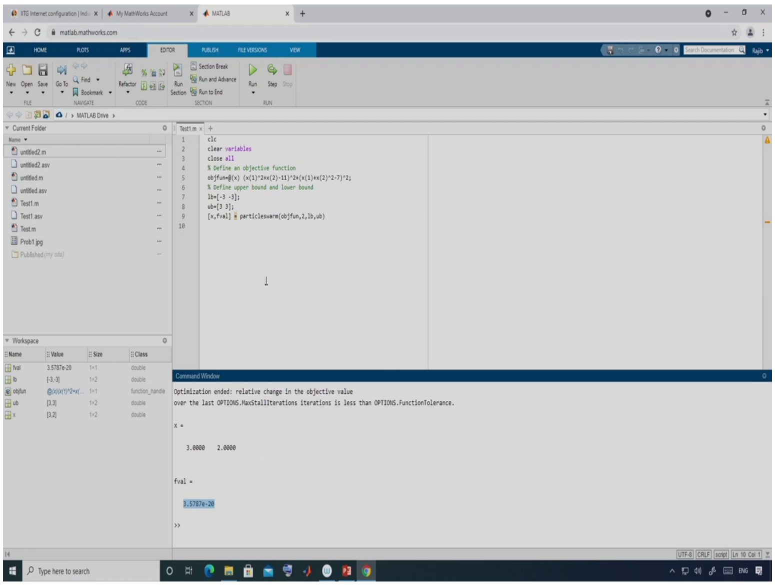
```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

    2.0000


fval =

   -16

>>
```

So, here this is the objective function I have defined and then lower bound and then upper bound and then I have used PSO. So, here objective function number of variable is one lower bound and upper bound I have defined. So, if I execute this particular MATLAB code, then let us see I will get the solution of this problem.

So, here the x star is 2 and function value is minus 16 ok. So, this is single variable optimization function. So, I got the solution that x equal to 2 and function value is minus 16.

Now, let us see the third problem. So, this is a multivariable function. So let us see this particular problem.

(Refer Slide Time: 20:52)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-3 -3];
ub=[3 3];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

Workspace

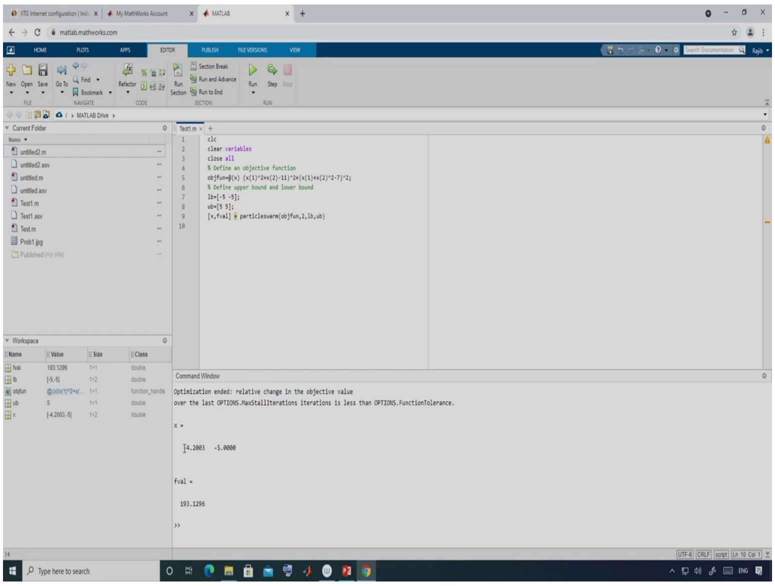
Name	Value	Size	Class
fval	3.5787e-20	1×1	double
lb	[-3,-3]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
ub	[3,3]	1×2	double
x	[3,2]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

    3.0000    2.0000


fval =

   3.5787e-20

>>
```

So, let me copy this lines so, I will paste it here, here you just see I have defined the objective function. So, now I have two variables that is x 1 and x 2 so, this x I have defined as a vector. So, here this is x 1 square plus x 2 minus 11 whole square, then x 1 square plus x 2 square minus 7 whole square.

And then lower bound is minus 3 minus 3 upper bound is plus 3 plus 3 and then I have used this particular function particle swarm. So, let me execute this I am getting this is the solution 3 2 and optimal function value is almost 0. So, I got the solution of this particular problem. So, if I change the range so, I can change it.

(Refer Slide Time: 21:44)

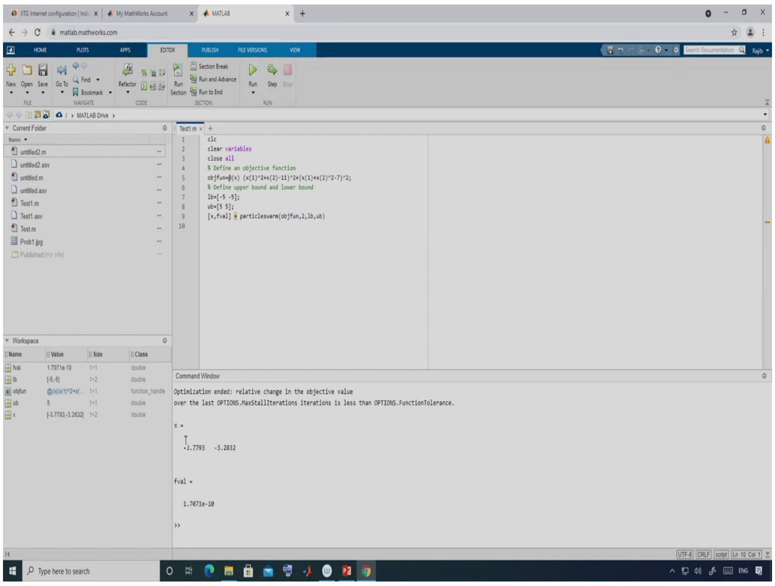


```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

   -4.2003   -5.0000


fval =

  193.1296

>>
```

Suppose, if I change it that minus 5 and minus 5 ok, then let us see what will happen. So, plus 5 and plus 5. So, if I run this particular function. So, I am getting a different solution, but this is not an optimal solution objective function value is 193.12, but it should be 0.

(Refer Slide Time: 22:11)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

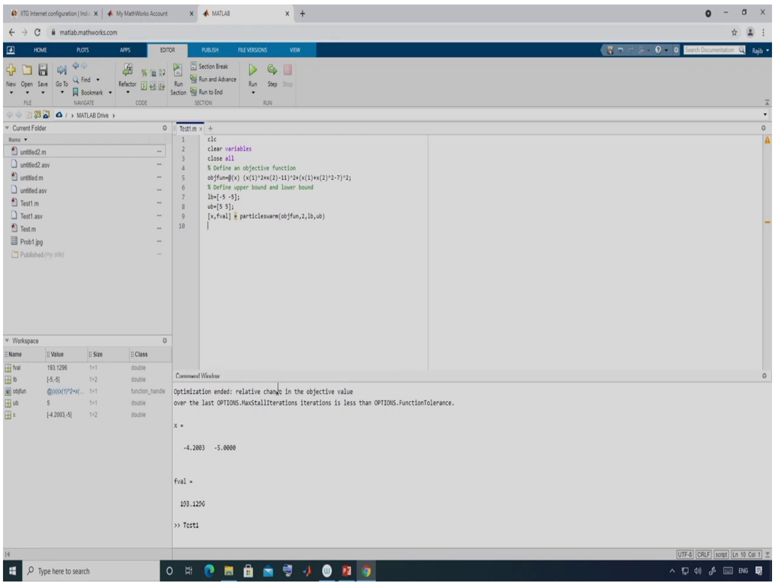
Workspace

Name	Value	Size	Class
fval	1.7071e-10	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
ub	5	1×1	double
x	[-3.7793,-3.2832]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


x =

   -3.7793   -3.2832



fval =

   1.7071e-10

>>
```

So, let me run this again, now I am getting a solution you just see this is one of the solution and objective function value is almost 0.

(Refer Slide Time: 22:21)



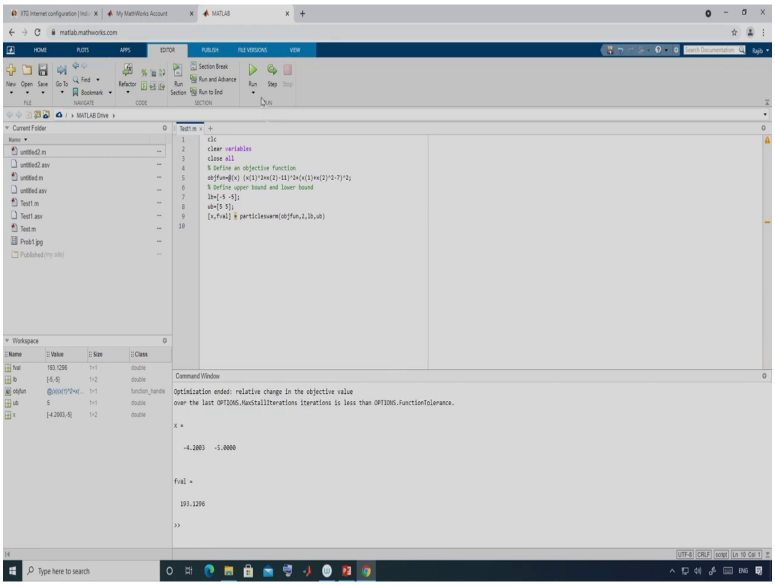
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

Name	Value	Size	Class
fval	193.1296	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
ub	5	1×1	double
x	[-4.2003,-5]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

   -4.2003   -5.0000


fval =

  193.1296

>> Test1
```

You can again try yeah, this time I am not getting ok.

(Refer Slide Time: 22:27)



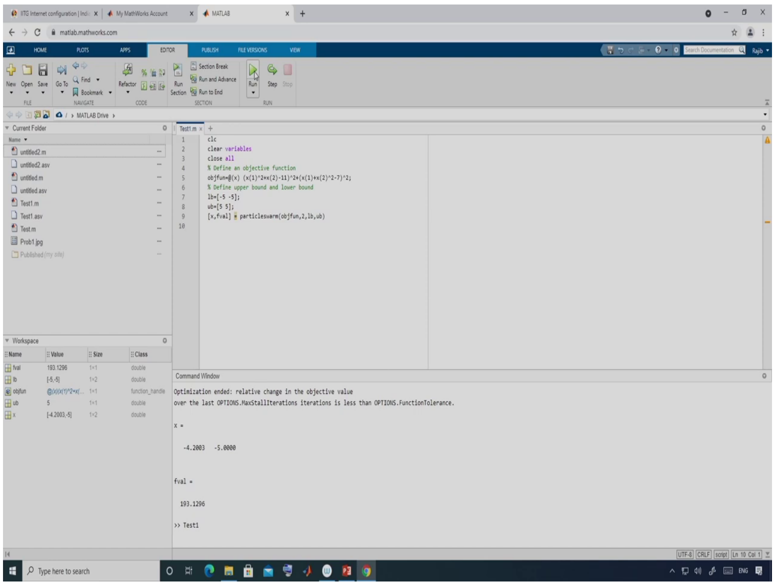
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

Name	Value	Size	Class
fval	193.1296	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
ub	5	1×1	double
x	[-4.2003,-5]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


x =

   -4.2003   -5.0000


fval =

  193.1296

>>
```

This is not 0, this is not 0, this is not 0.

(Refer Slide Time: 22:30)



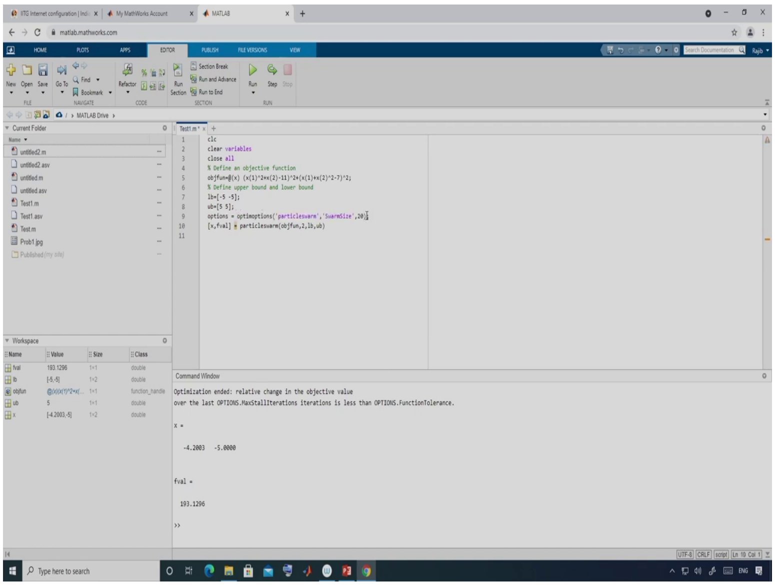
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

Name	Value	Size	Class
fval	193.1296	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
ub	5	1×1	double
x	[-4.2003,-5]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


x =

   -4.2003   -5.0000


fval =

  193.1296

>> Test1
```

So, I am not getting the solution. So, what I can do basically. So, I can put the option I can just see so here I am defining the option.

(Refer Slide Time: 22:47)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',20);
[x,fval] = particleswarm(objfun,2,lb,ub)
```

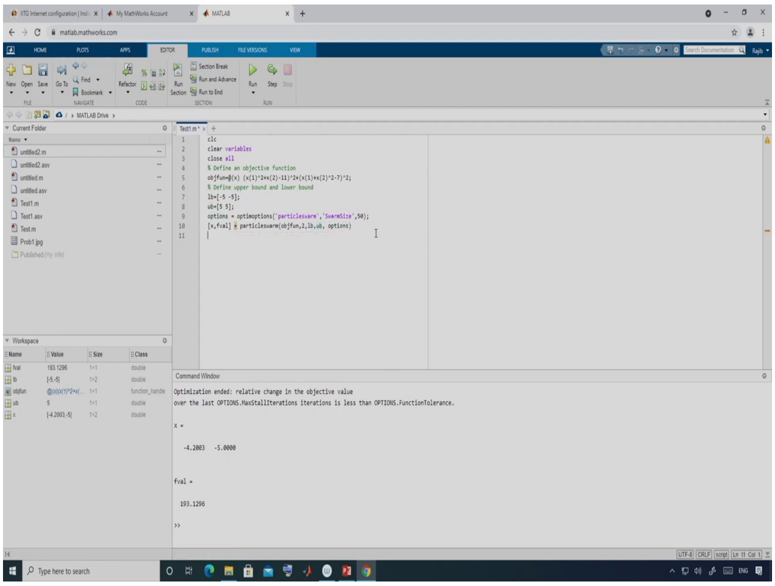
Workspace

Name	Value	Size	Class
fval	193.1296	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
ub	5	1×1	double
x	[-4.2003,-5]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


x =


   -4.2003   -5.0000



fval =


  193.1296

>>
```

So, what I will do? I will increase this swarm size.

(Refer Slide Time: 22:54)

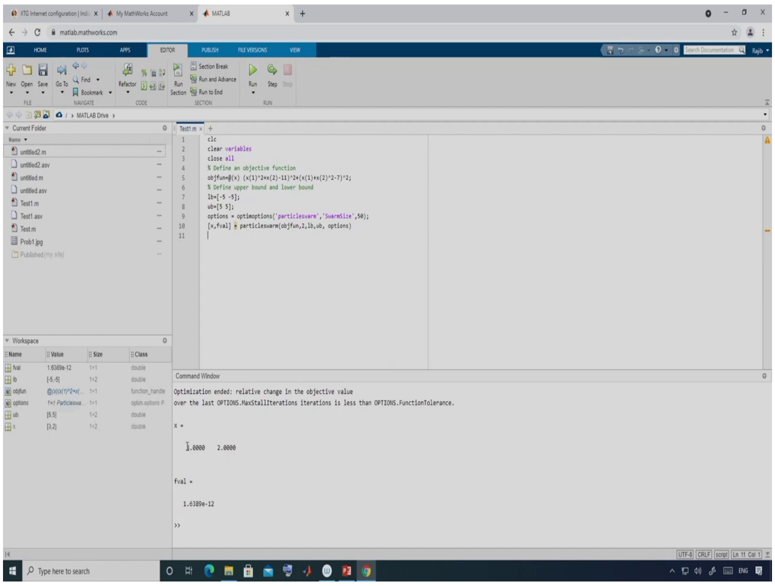


```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',50);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


x =


   -4.2003   -5.0000



fval =


  193.1296

>>
```

Suppose I am using 50, so what will happen just see if I am using 50 so, I have to write option here options ok. So, I am writing and I am using swarm size; that means, population size is 50.

(Refer Slide Time: 23:10)

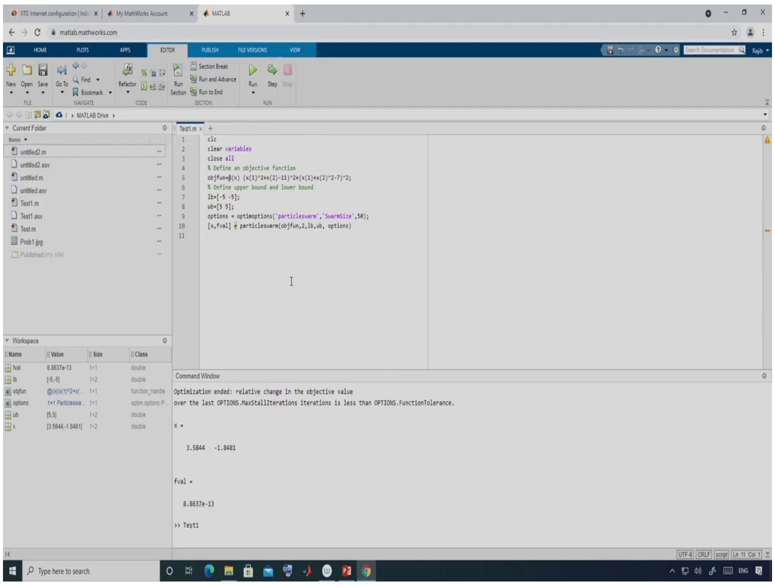


```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',50);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

    3.0000    2.0000


fval =

   1.6389e-12

>>
```

Now, I am getting a solution 3 2 you just see. So, by increasing population size so, if you are not getting solution, then you increase the population size or you can also change the other parameters. So, that I am not discussing here, but you can also change the other parameters, but here I am changing the population size ok, I got the solution.

(Refer Slide Time: 23:34)



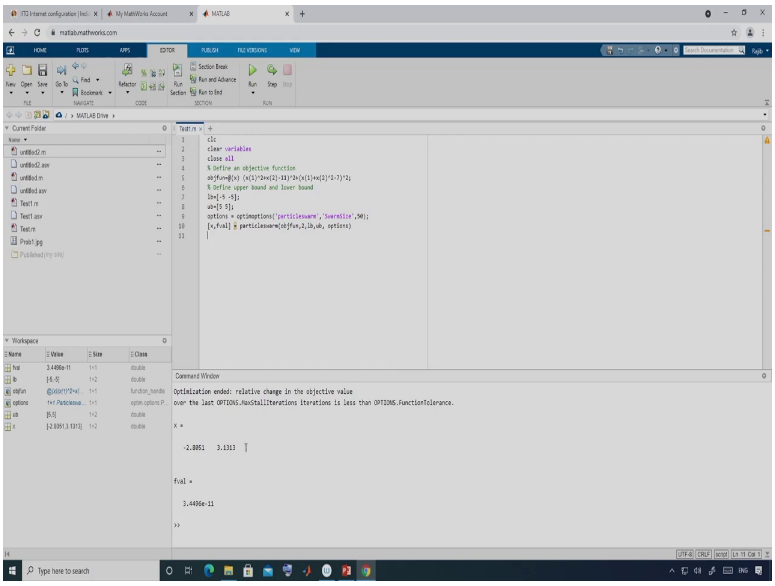
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',50);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

Name	Value	Size	Class
fval	8.8637e-13	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
options	1×1 Particleswa...	1×1	optim.options.P...
ub	[5,5]	1×2	double
x	[3.5844,-1.8481]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

   3.5844   -1.8481


fval =

  8.8637e-13

>> Test1
```

So, let me try again yeah I am getting another solution and this is also a solution because this value is 0 yeah.

(Refer Slide Time: 23:42)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',50);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

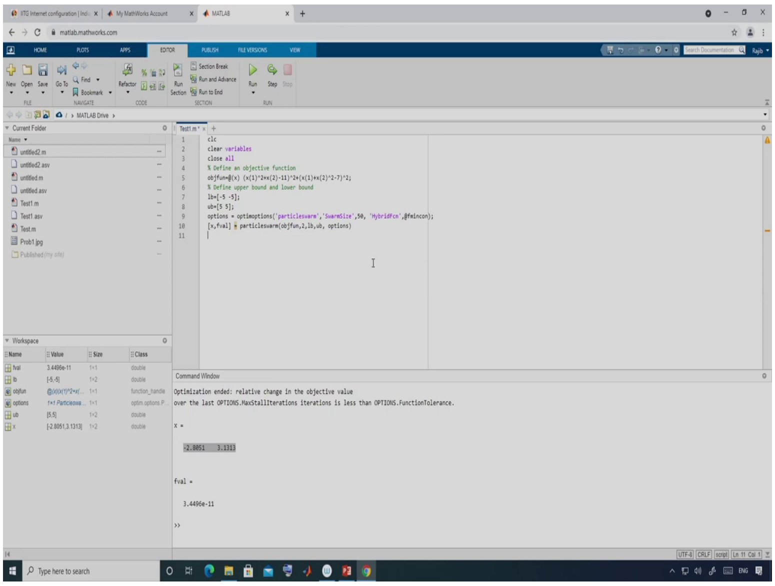
Name	Value	Size	Class
fval	3.4496e-11	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
options	1×1 Particleswa...	1×1	optim.options.P...
ub	[5,5]	1×2	double
x	[-2.8051,3.1313]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

   -2.8051    3.1313


fval =

   3.4496e-11

>>
```

So, you just see you are getting solution. So, by increasing the population size so you are getting one of the solutions. So, as you know for this particular problem. So, as you know for this particular problem we have 4 optimal solution and they are alternate optimal solutions, because objective function value is 0, but solution is different. So, we are getting one of that solution between minus 5 and plus 5.

Or otherwise what I can do? I can also use the hybrid function ok. So, there I can use. So, I can also use the hybrid function. So, let me copy this part.

(Refer Slide Time: 24:29)



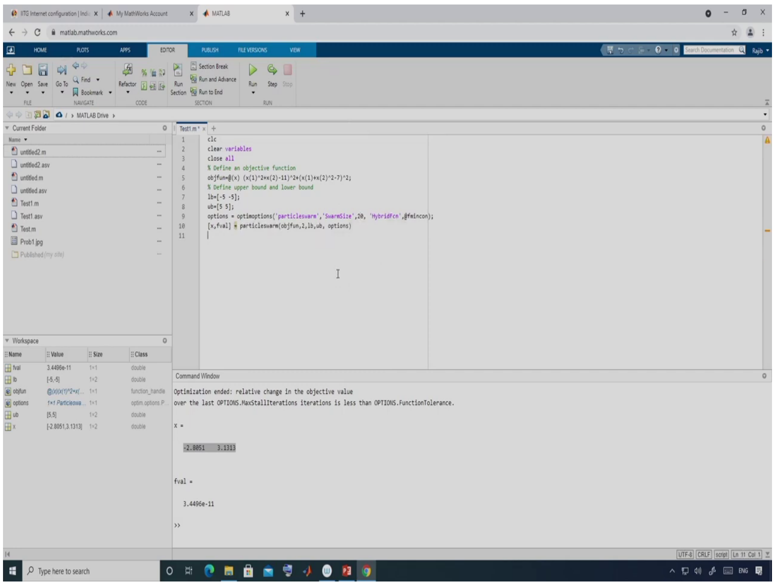
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',50, 'HybridFcn',@fmincon);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

Name	Value	Size	Class
fval	3.4496e-11	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
options	1×1 Particleswa...	1×1	optim.options.P...
ub	[5,5]	1×2	double
x	[-2.8051,3.1313]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

   -2.8051    3.1313


fval =

   3.4496e-11

>>
```

So, just see what will happen. So, here I am using the hybrid function.

(Refer Slide Time: 24:35)

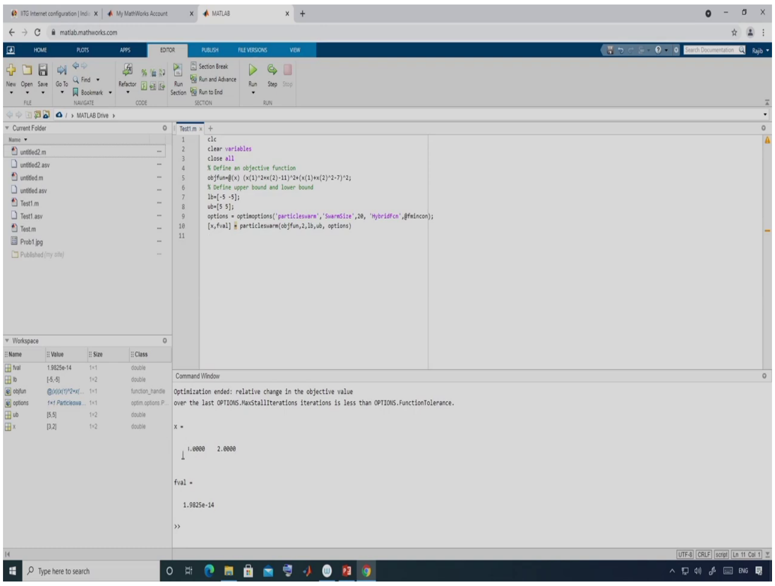


```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',20, 'HybridFcn',@fmincon);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

   -2.8051    3.1313


fval =

   3.4496e-11

>>
```

So, let me try with 20 solution now, let us see whether I am getting a solution; yeah, I am getting.

(Refer Slide Time: 24:43)



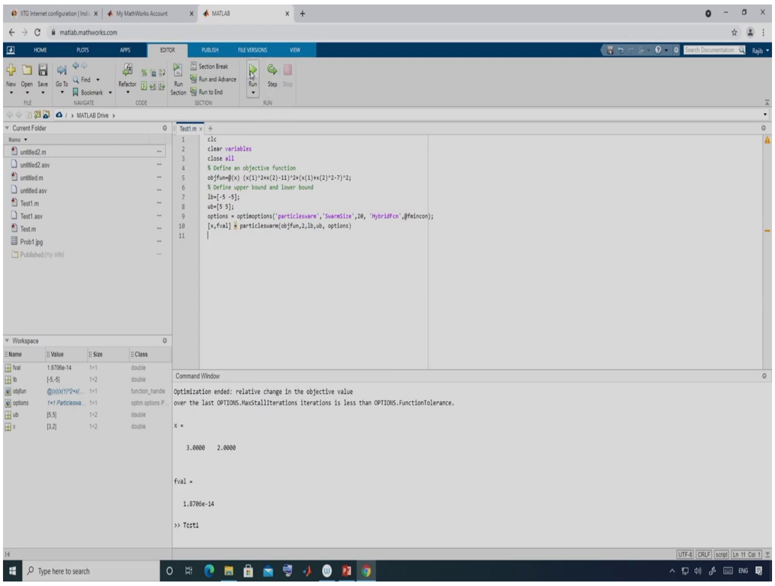
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',20, 'HybridFcn',@fmincon);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

Name	Value	Size	Class
fval	1.9825e-14	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
options	1×1 Particleswa...	1×1	optim.options.P...
ub	[5,5]	1×2	double
x	[3,2]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

    3.0000    2.0000


fval =

   1.9825e-14

>>
```

So, you just see this is 3 2 and exact solution I am getting an objective function value is almost 0, so with 20 population.

(Refer Slide Time: 24:55)



Test1.m

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',20, 'HybridFcn',@fmincon);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

Workspace

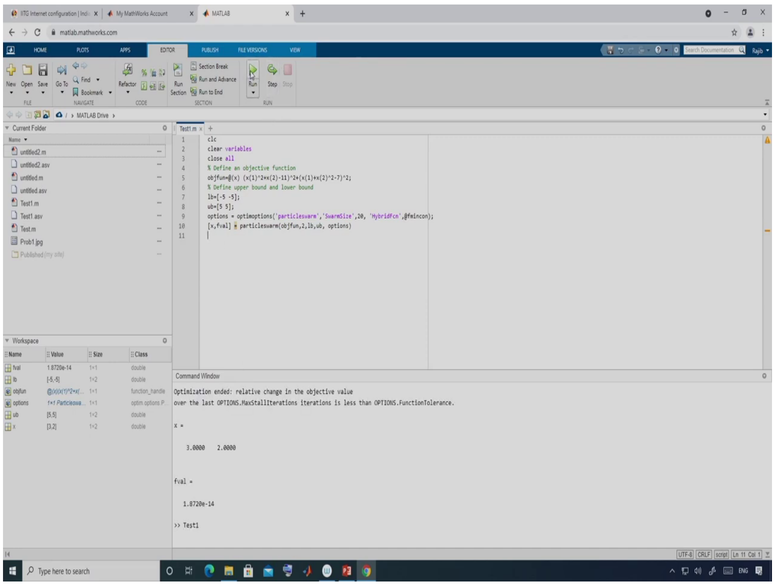
Name	Value	Size	Class
fval	1.8706e-14	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
options	1×1 Particleswa...	1×1	optim.options.P...
ub	[5,5]	1×2	double
x	[3,2]	1×2	double

Command Window

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


x =

    3.0000    2.0000


fval =

   1.8706e-14

>> Test1
```

Let me try again yeah, I am getting yeah.

(Refer Slide Time: 24:58)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',20, 'HybridFcn',@fmincon);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

Workspace

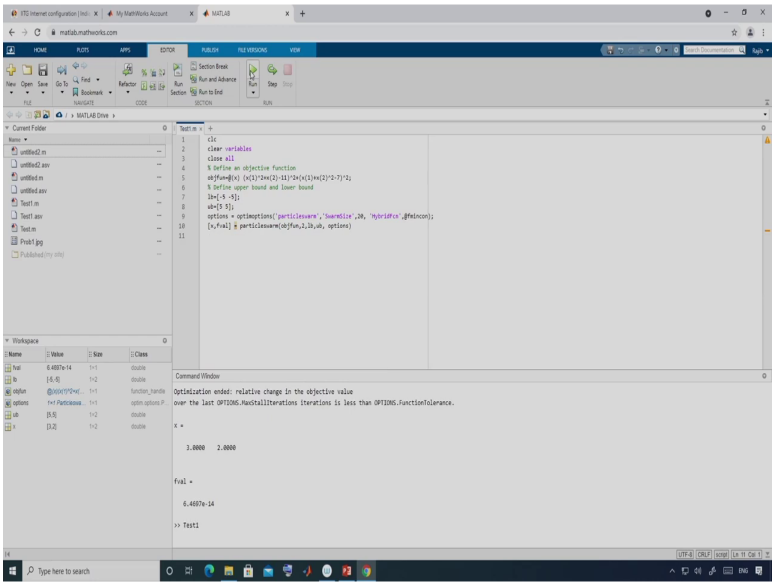
Name	Value	Size	Class
fval	1.8720e-14	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
options	1×1 Particleswa...	1×1	optim.options.P...
ub	[5,5]	1×2	double
x	[3,2]	1×2	double

Command Window

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

    3.0000    2.0000


fval =

   1.8720e-14

>> Test1
```

Let me try again. So, I am getting a solution.

(Refer Slide Time: 25:00)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',20, 'HybridFcn',@fmincon);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

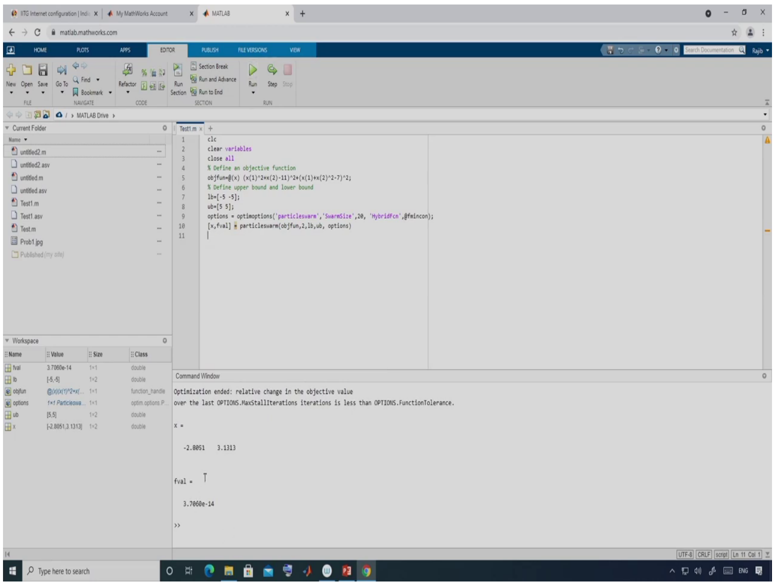
Workspace

Name	Value	Size	Class
fval	6.4697e-14	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
options	1×1 Particleswa...	1×1	optim.options.P...
ub	[5,5]	1×2	double
x	[3,2]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


x =


    3.0000    2.0000



fval =


   6.4697e-14

>> Test1
```

So, you just see I am getting a solution.

(Refer Slide Time: 25:02)



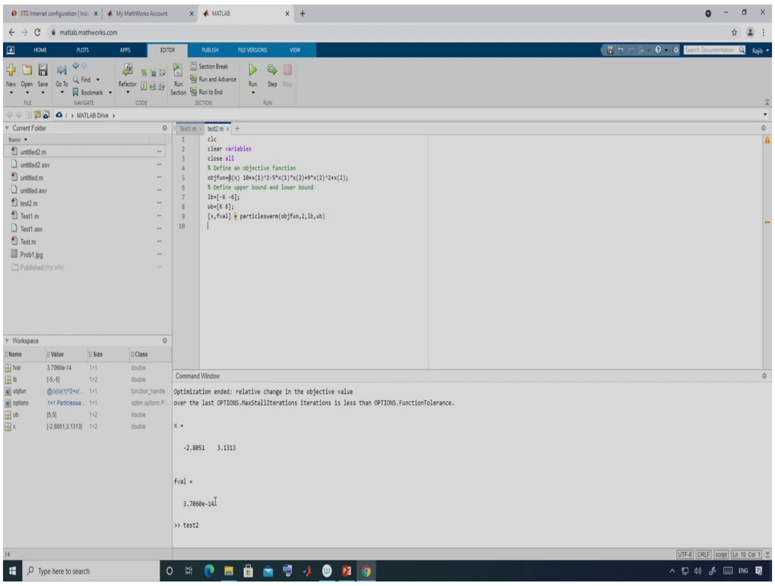
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',20, 'HybridFcn',@fmincon);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

   -2.8051    3.1313


fval =

   3.7060e-14

>>
```

So, in case of hybrid method so what is happening that whatever solution you are getting from PSO so that solution you are taking as an initial solution and then you are refining your solution or you are trying to get a better solution using the classical optimization method ok. So, this is your hybrid method we are using.

So, let me try another function so, let me try this particular function. So, maybe I can open a different file ok.

(Refer Slide Time: 25:37)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6 -6];
ub=[6 6];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

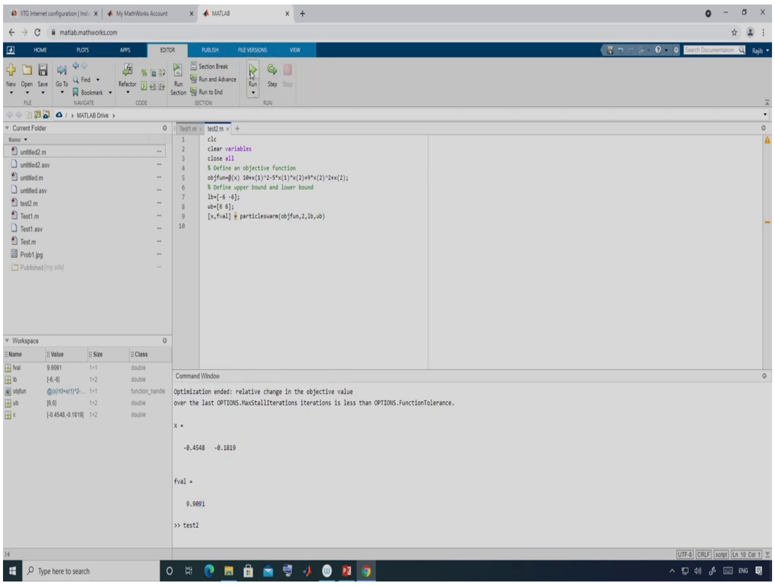
Workspace

Name	Value	Size	Class
fval	3.7060e-14	1×1	double
lb	[-5,-5]	1×2	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
options	1×1 Particleswa...	1×1	optim.options.P...
ub	[5,5]	1×2	double
x	[-2.8051,3.1313]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


x =

   -2.8051    3.1313



fval =

   3.7060e-14

>> test2
```

So, this is the problem I would like to solve using particle swarm optimization ok.

(Refer Slide Time: 25:46)



test2.m

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6 -6];
ub=[6 6];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

Workspace

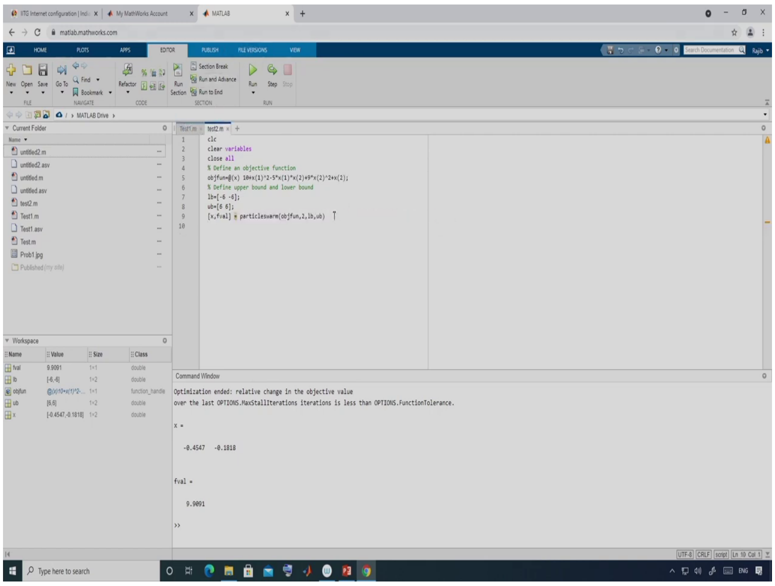
Name	Value	Size	Class
fval	9.9091	1×1	double
lb	[-6,-6]	1×2	double
objfun	@(x)10+x(1)^2-...	1×1	function_handle
ub	[6,6]	1×2	double
x	[-0.4548,-0.1819]	1×2	double

Command Window

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

   -0.4548   -0.1819


fval =

    9.9091

>> test2
```

So, I am getting a solution this is minus 0.4548 and minus 0.1819. So, I am not sure whether this is the solution. So, let me try again, let me try again, let me try again yeah.

(Refer Slide Time: 26:02)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6 -6];
ub=[6 6];
[x,fval] = particleswarm(objfun,2,lb,ub)
```

Workspace

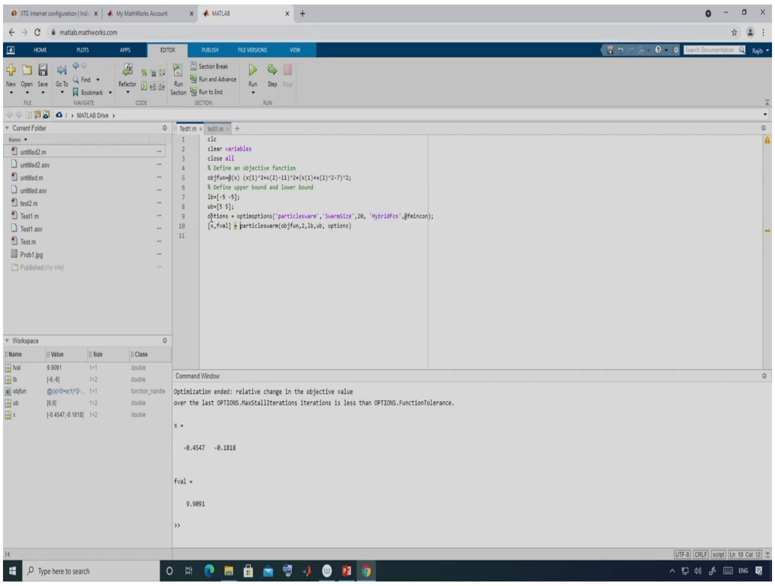
Name	Value	Size	Class
fval	9.9091	1×1	double
lb	[-6,-6]	1×2	double
objfun	@(x)10+x(1)^2-...	1×1	function_handle
ub	[6,6]	1×2	double
x	[-0.4547,-0.1818]	1×2	double

Command Window

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


x =

   -0.4547   -0.1818


fval =

    9.9091

>>
```

So, I am getting. So, I can also use hybrid method.

(Refer Slide Time: 26:08)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-5 -5];
ub=[5 5];
options = optimoptions('particleswarm','SwarmSize',20, 'HybridFcn',@fmincon);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

Workspace

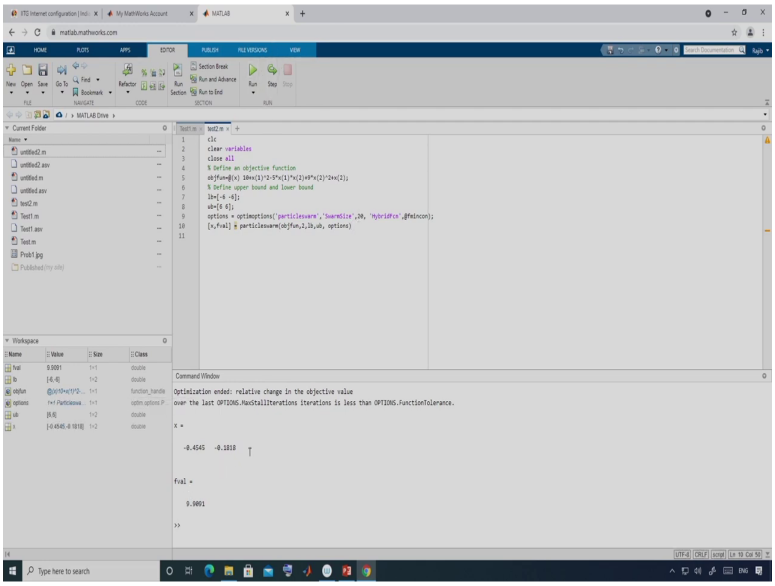
Name	Value	Size	Class
fval	9.9091	1×1	double
lb	[-6,-6]	1×2	double
objfun	@(x)10+x(1)^2-...	1×1	function_handle
ub	[6,6]	1×2	double
x	[-0.4547,-0.1818]	1×2	double

Command Window

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

  -0.4547   -0.1818


fval =

    9.9091

>>
```

Just see whether you are getting a better solution or not ok.

(Refer Slide Time: 26:19)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6 -6];
ub=[6 6];
options = optimoptions('particleswarm','SwarmSize',20, 'HybridFcn',@fmincon);
[x,fval] = particleswarm(objfun,2,lb,ub, options)
```

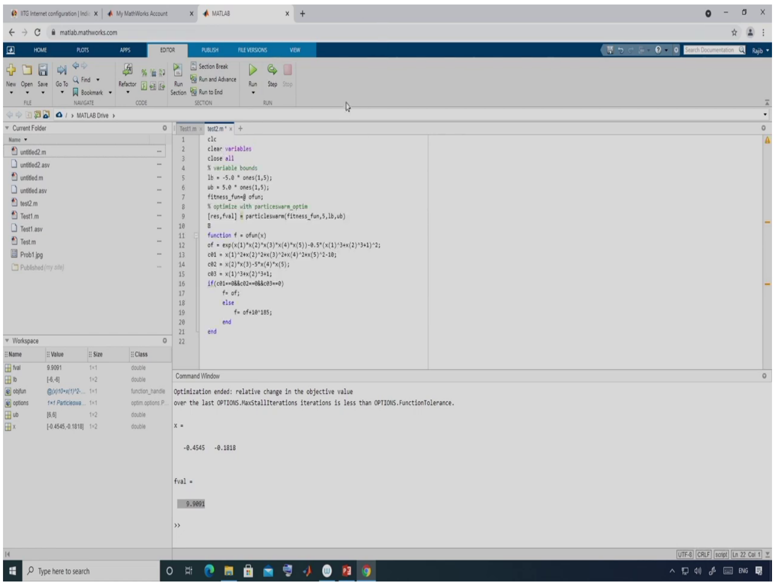
```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

  -0.4545   -0.1818


fval =

    9.9091

>>
```

So, what I will do I will put option here and I will write option options ok. So, I am writing options here yeah. So, I think yeah this is the solution of this particular problem. And objective function value is 9.9091 ok. So, I got the solution of this particular problem.

So, here what I have done? I have defined the objective function then lower bound upper bound I have defined then this is the option I have defined. So, here swarm size I have used 20 and I have used hybrid function that is f min con I have used and then I have used particle swarm optimization with this particular options.

So, I am getting the solution of this problem. So, let me try the next problem. So, next problem is an optimization problem with constrain. So, we have 3 equality constrains so, let me see. So, this is the MATLAB code let me copy this code and I will put it here ok.

(Refer Slide Time: 27:25)



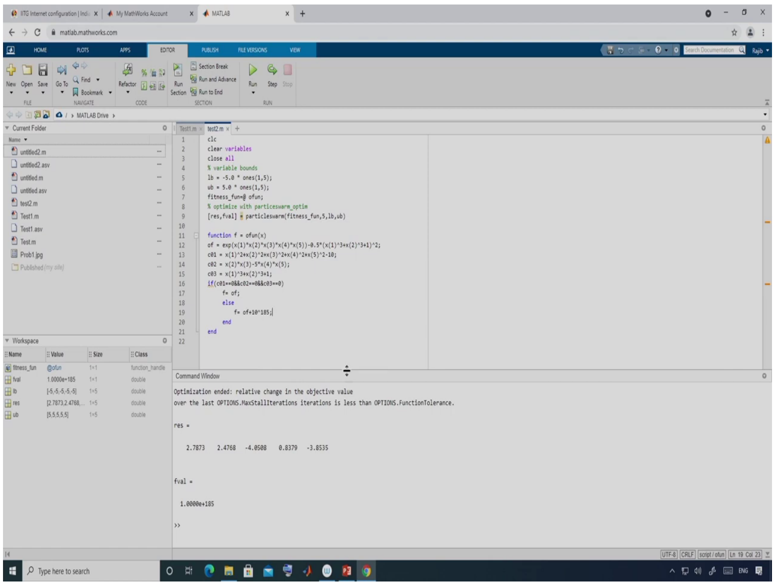
```
clc
clear variables
close all
% variable bounds
lb = -5.0 * ones(1,5);
ub = 5.0 * ones(1,5);
fitness_fun=@ ofun;
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,5,lb,ub)

function f = ofun(x)
of = exp(x(1)*x(2)*x(3)*x(4)*x(5))-0.5*(x(1)^3+x(2)^3+1)^2;
c01 = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2-10;
c02 = x(2)*x(3)-5*x(4)*x(5);
c03 = x(1)^3+x(2)^3+1;
if(c01==0&&c02==0&&c03==0)
     f= of;
     else
         f= of+10^185;
     end
end
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

x =

   -0.4545   -0.1818


fval =

    9.9091

>>
```

So, this is the code here. So, what I am doing so I am just writing this objective function so, this name of the function is ofun so, this is the objective function. And first line is objective function value and then this is constrain 1, constrain 2, constrain 3.

Now, what I have done? If constrain 1 equal to 0, constrain 2 equal to 0, and constrain 3 equal to 0; that means, that is a feasible solution. So, therefore, the fitness value equal to objective function value, else the fitness value is objective function value plus a large number. Because the problem is a minimization type problem so therefore, I am using a large number.

(Refer Slide Time: 28:18)



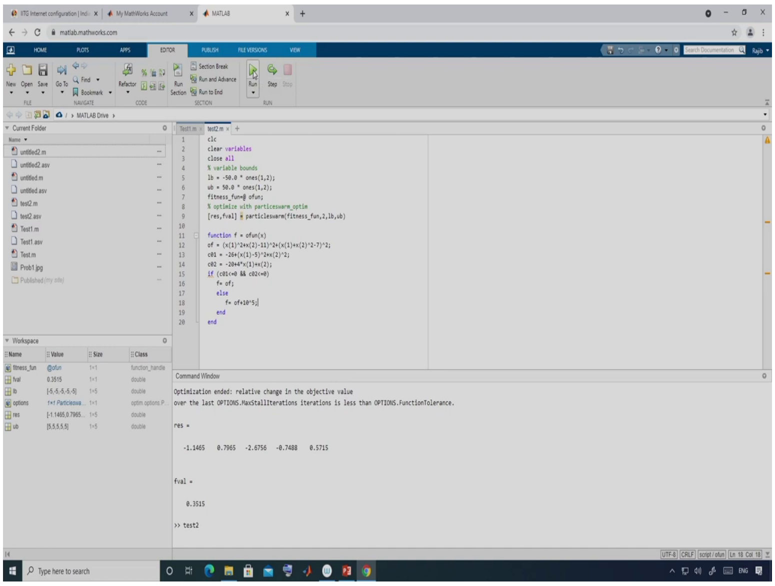
```
clc
clear variables
close all
% variable bounds
lb = -5.0 * ones(1,5);
ub = 5.0 * ones(1,5);
fitness_fun=@ ofun;
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,5,lb,ub)

function f = ofun(x)
of = exp(x(1)*x(2)*x(3)*x(4)*x(5))-0.5*(x(1)^3+x(2)^3+1)^2;
c01 = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2-10;
c02 = x(2)*x(3)-5*x(4)*x(5);
c03 = x(1)^3+x(2)^3+1;
if(c01==0&&c02==0&&c03==0)
    f= of;
    else
        f= of+10^185;
    end
end
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

res =

    2.7873    2.4768   -4.0508    0.8379   -3.8535


fval =

  1.0000e+185

>>
```

Name	Value	Size	Class
fitness_fun	@ofun	1×1	function_handle
fval	1.0000e+185	1×1	double
lb	[-5,-5,-5,-5,-5]	1×5	double
res	[2.7873,2.4768,...	1×5	double
ub	[5,5,5,5,5]	1×5	double

So, let me execute this. So, I am getting this particular solution let us go to the next problem. So, next problem is again this is a multivariable problem with two inequality constrains. So, let me copy this particular file.

(Refer Slide Time: 28:42)



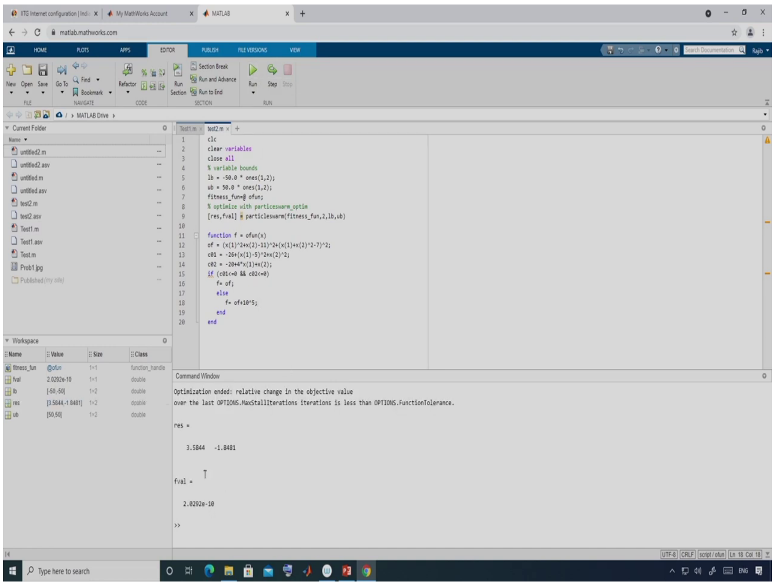
```
clc
clear variables
close all
% variable bounds
lb = -50.0 * ones(1,2);
ub = 50.0 * ones(1,2);
fitness_fun=@ ofun;
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub)

function f = ofun(x)
of = (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
c01 = -26+(x(1)-5)^2+x(2)^2;
c02 = -20+4*x(1)+x(2);
if (c01<=0 && c02<=0)
   f= of;
   else
      f= of+10^5;
   end
end
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


res =

  -1.1465    0.7965   -2.6756   -0.7488    0.5715



fval =

    0.3515

>> test2
```

So, here so, I have defined the lower bound and I have also defined the upper bound and then this particle swarm function is used and then this is the objective function I have calculated. So, objective function and there are two constrain. Now, in this case the, constrains should be less than equal to 0 ok.

So, c01 is less than equal to 0 and c02 is less than equal to 0. So, if there is no violation so in that case the objective function value is equal to fitness value or otherwise there will be a violation. So, that we have used and let me execute this particular m file yeah.

(Refer Slide Time: 29:33)



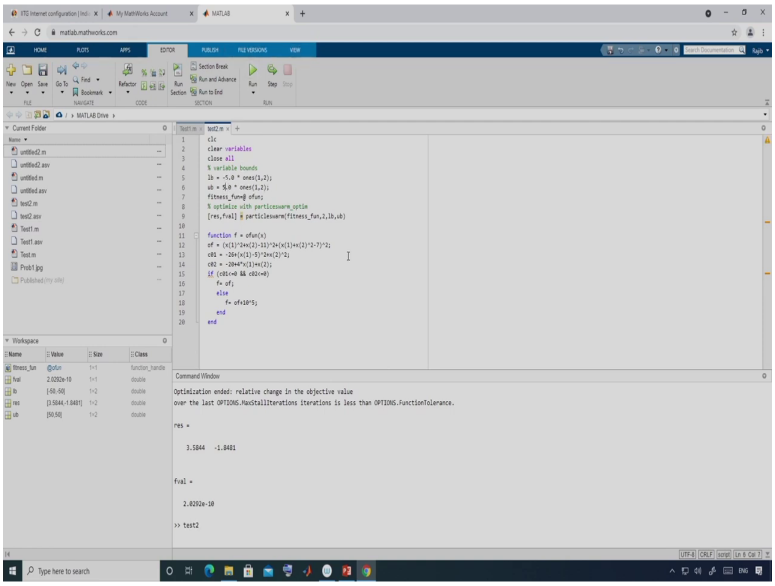
```
clc
clear variables
close all
% variable bounds
lb = -50.0 * ones(1,2);
ub = 50.0 * ones(1,2);
fitness_fun=@ ofun;
% optimize with particleswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub)

function f = ofun(x)
of = (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
c01 = -26+(x(1)-5)^2+x(2)^2;
c02 = -20+4*x(1)+x(2);
if (c01<=0 && c02<=0)
   f= of;
   else
      f= of+10^5;
   end
end
```

Name	Value	Size	Class
fitness_fun	@ofun	1×1	function_handle
fval	2.0292e-10	1×1	double
lb	[-50,-50]	1×2	double
res	[3.5844,-1.8481]	1×2	double
ub	[50,50]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

res =

    3.5844   -1.8481


fval =

   2.0292e-10

>>
```

So, I am getting a solution you just see this is, this is the solution and let me see so, I can change this lower bound and upper bound.

(Refer Slide Time: 29:41)

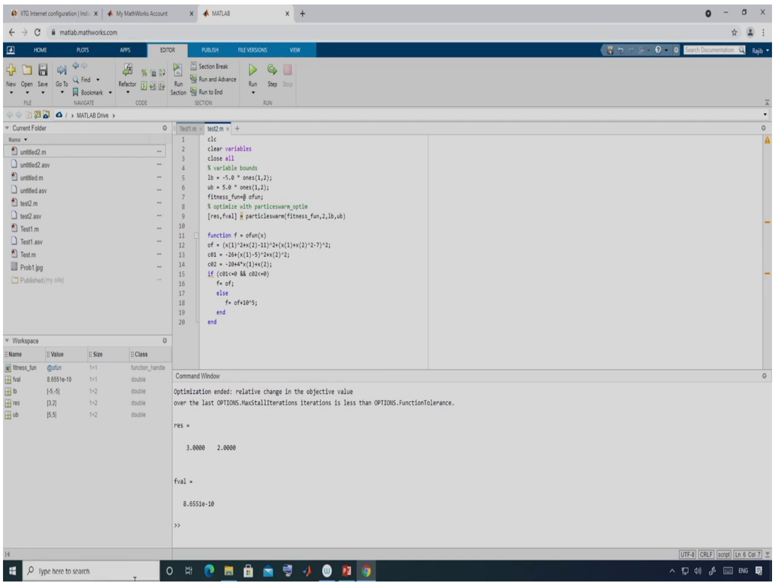


```
clc
clear variables
close all
% variable bounds
lb = -5.0 * ones(1,2);
ub = 5.0 * ones(1,2);
fitness_fun=@ ofun;
% optimize with particleswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub)

function f = ofun(x)
of = (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
c01 = -26+(x(1)-5)^2+x(2)^2;
c02 = -20+4*x(1)+x(2);
if (c01<=0 && c02<=0)
   f= of;
   else
      f= of+10^5;
   end
end
```

Name	Value	Size	Class
fitness_fun	@ofun	1×1	function_handle
fval	2.0292e-10	1×1	double
lb	[-50,-50]	1×2	double
res	[3.5844,-1.8481]	1×2	double
ub	[50,50]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

res =

    3.5844   -1.8481


fval =

   2.0292e-10

>> test2
```

(Refer Slide Time: 29:47)



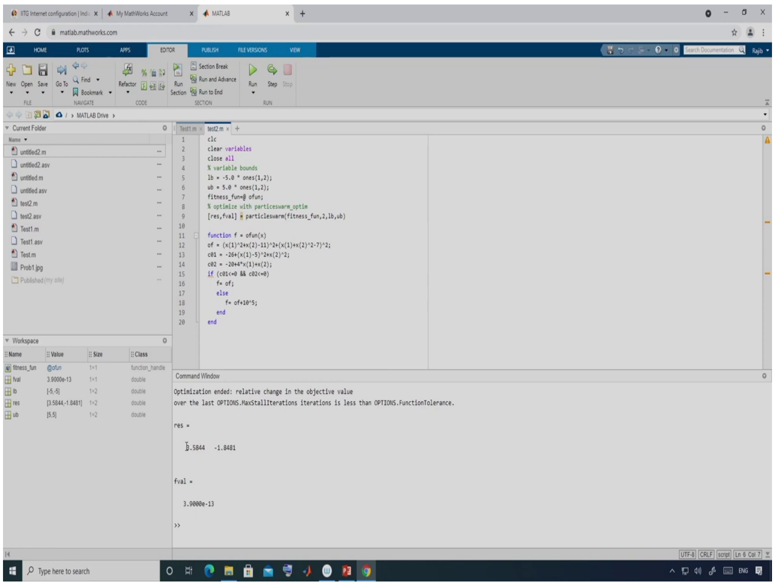
```
clc
clear variables
close all
% variable bounds
lb = -5.0 * ones(1,2);
ub = 5.0 * ones(1,2);
fitness_fun=@ ofun;
% optimize with particleswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub)

function f = ofun(x)
of = (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
c01 = -26+(x(1)-5)^2+x(2)^2;
c02 = -20+4*x(1)+x(2);
if (c01<=0 && c02<=0)
   f= of;
   else
      f= of+10^5;
   end
end
```

Name	Value	Size	Class
fitness_fun	@ofun	1×1	function_handle
fval	8.6551e-10	1×1	double
lb	[-5,-5]	1×2	double
res	[3,2]	1×2	double
ub	[5,5]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

res =

    3.0000    2.0000


fval =

   8.6551e-10

>>
```

So, between minus 5 and plus 5 yeah.

(Refer Slide Time: 29:55)

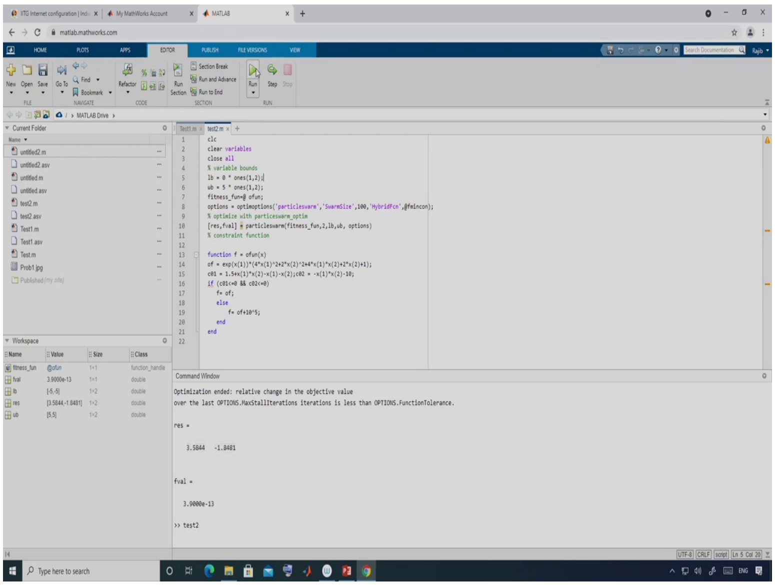


```
clc
clear variables
close all
% variable bounds
lb = -5.0 * ones(1,2);
ub = 5.0 * ones(1,2);
fitness_fun=@ ofun;
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub)

function f = ofun(x)
of = (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
c01 = -26+(x(1)-5)^2+x(2)^2;
c02 = -20+4*x(1)+x(2);
if (c01<=0 && c02<=0)
   f= of;
   else
      f= of+10^5;
   end
end
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

res =

   3.5844   -1.8481


fval =

  3.9000e-13

>>
```

I am getting 32. So, 32 is one of the solution so, I am getting that one and yeah. So, I am getting another solution. So, this is I am getting the constrained optimal solution of the problem. Now, let me go to the, another problem. So, this problem is there are again this is a constrained optimization problem with two inequality constrain ok.

(Refer Slide Time: 30:26)



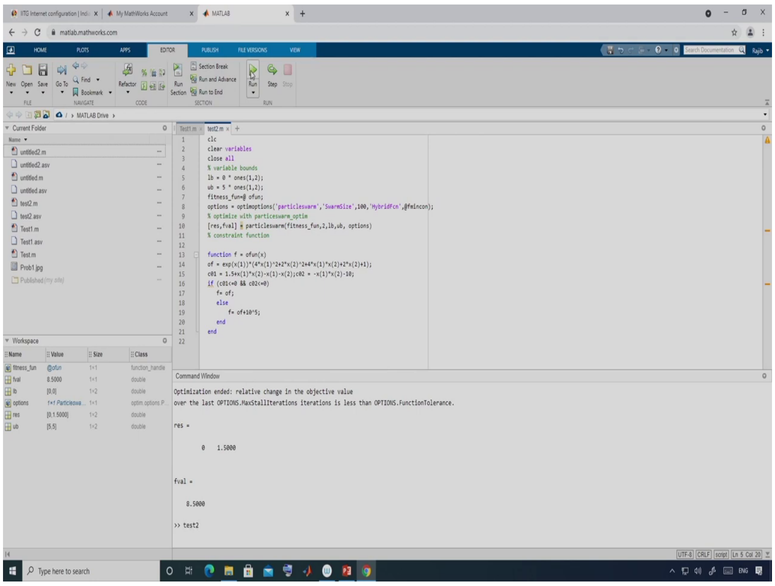
```
clc
clear variables
close all
% variable bounds
lb = 0 * ones(1,2);
ub = 5 * ones(1,2);
fitness_fun=@ ofun;
options = optimoptions('particleswarm','SwarmSize',100,'HybridFcn',@fmincon);
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub, options)
% constraint function

function f = ofun(x)
of = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
c01 = 1.5+x(1)*x(2)-x(1)-x(2);c02 = -x(1)*x(2)-10;
if (c01<=0 && c02<=0)
   f= of;
   else
       f= of+10^5;
   end
end
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.


res =

    3.5844   -1.8481



fval =

   3.9000e-13

>> test2
```

Name	Value	Size	Class
fitness_fun	@ofun	1×1	function_handle
fval	3.9000e-13	1×1	double
lb	[0,0]	1×2	double
res	[3.5844,-1.8481]	1×2	double
ub	[5,5]	1×2	double

So, this is the code and here you can see. So, here again I have defined the objective function here. So, there are two constrain that is c1 c01 and c02 and then we are calculating the value of f. So, if there is no violation then this value is objective function value or otherwise we are putting some big values ok so, in order to avoid that particular solution. So, let me execute this one yeah.

(Refer Slide Time: 31:01)



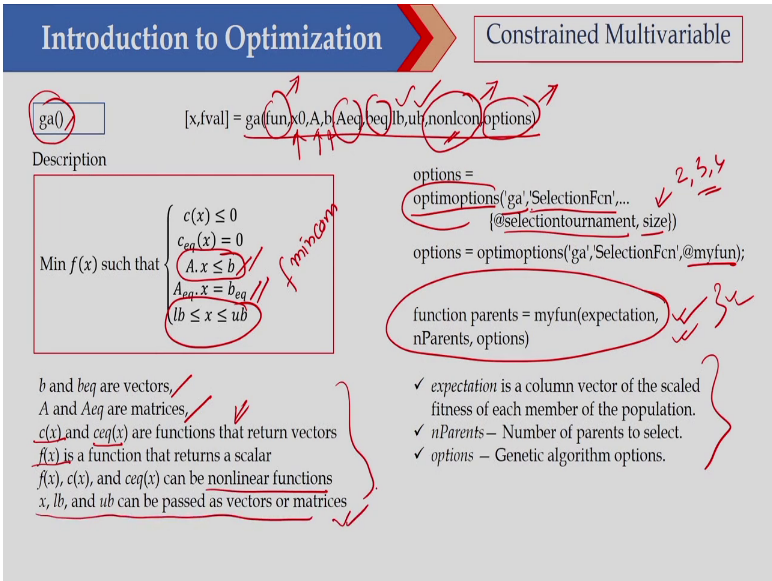
```
clc
clear variables
close all
% variable bounds
lb = 0 * ones(1,2);
ub = 5 * ones(1,2);
fitness_fun=@ ofun;
options = optimoptions('particleswarm','SwarmSize',100,'HybridFcn',@fmincon);
% optimize with particeswarm_optim
[res,fval] = particleswarm(fitness_fun,2,lb,ub, options)
% constraint function

function f = ofun(x)
of = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
c01 = 1.5+x(1)*x(2)-x(1)-x(2);c02 = -x(1)*x(2)-10;
if (c01<=0 && c02<=0)
   f= of;
   else
       f= of+10^5;
   end
end
```

Name	Value	Size	Class
fitness_fun	@ofun	1×1	function_handle
fval	8.5000	1×1	double
lb	[0,0]	1×2	double
options	1×1 Particleswa...	1×1	optim.options.P...
res	[0,1.5000]	1×2	double
ub	[5,5]	1×2	double

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

res =

       0    1.5000


fval =

    8.5000

>> test2
```

So, I am getting the solution of this particular problem. So, that is 0 and 1.500. So, in this case I have used the hybrid function. So, what is happening that after PSO solution so that solution has been used and to and then fmincon function is used and to get a better solution of this particular problem.

(Refer Slide Time: 31:32)



Introduction to Optimization

Constrained Multivariable

ga()

[x,fval] = ga(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)

Description

$$\text{Min } f(x) \text{ such that } \begin{cases} c(x) \le 0 \\ c_{eq}(x) = 0 \\ A.x \le b \\ A_{eq}.x = b_{eq} \\ lb \le x \le ub \end{cases}$$

fmincon

b and *beq* are vectors,
A and *Aeq* are matrices,
c(*x*) and *ceq*(*x*) are functions that return vectors
f(*x*) is a function that returns a scalar
f(*x*), *c*(*x*), and *ceq*(*x*) can be nonlinear functions
x, *lb*, and *ub* can be passed as vectors or matrices

```
options =
optimoptions('ga','SelectionFcn',...
        {@selectiontournament, size})
```

2,3,4

```
options = optimoptions('ga','SelectionFcn',@myfun);
```

```
function parents = myfun(expectation,
nParents, options)
```

3

- ✓ *expectation* is a column vector of the scaled fitness of each member of the population.
- ✓ *nParents* – Number of parents to select.
- ✓ *options* – Genetic algorithm options.

Now, let us see how we can solve this problem using genetic algorithm ok. So, here the genetic algorithm function is ga ok. So, I have to use this function and this is the syntax of this particular function. So, here I can handle lot of things. So, this is the value of the objective function ok. So, objective function then you are putting initial solution then A and b so, this is the inequality constrain ok. So, this is the inequality linear constrain.

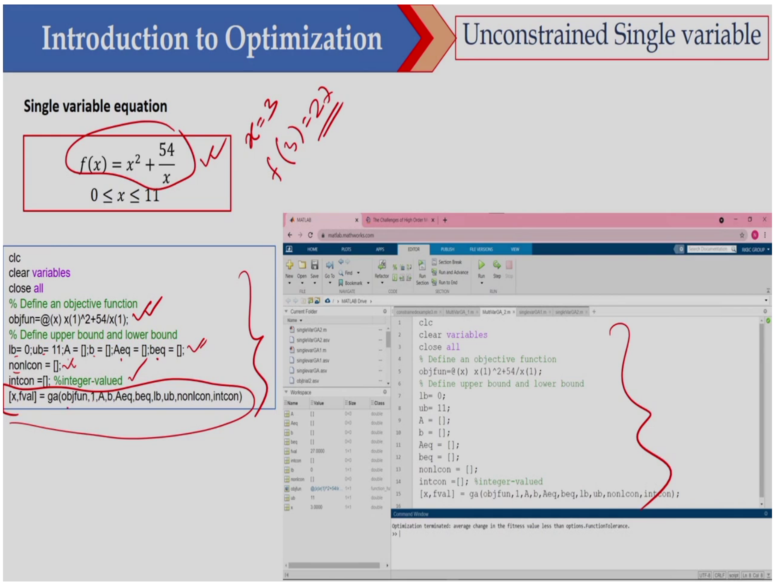
So, I can put just like fmincon; just like fmincon I can put linear inequality constrain and Aeq and beq that is the linear equality constrains. So, this is linear inequality constrain and this is equal equality constrain and I can also define the lower bound and upper bound lower bound and upper bound and here I can define the constrain if there is any non-linear constrains. So, I can define here and I can also give the option.

Now, in option if you are not giving the default options will be taken otherwise you can define your own option. So, one option may be so this is optimoption so, this keyword is same. So, here earlier I have used particles swarm, but in this case it is ga then you are using selection function. So, selection function you are using tournament selection and size of the tournament you can define that 2 3 4 whatever tournament size you can define.

So, you can also define your own function ok. So, own function that is my function so you can also define your own selection function suppose I am defining here. So, I am not discussing these things so this is little bit in this is an advanced topic. So, you can also define your own selection function here so, this way I can define. So, I have defined here what is b and beq that is their vector this is also A and Aeq is matrix and this c x and ceq x.

So, these are the function. So, basically non-linear constrain then f x is objective function and these are non-linear function and I can also define lower bound upper bound here ok.

(Refer Slide Time: 34:04)



Introduction to Optimization

Unconstrained Single variable

Single variable equation

$$f(x) = x^2 + \frac{54}{x}$$

$$0 \le x \le 11$$

$x=3$

$f(3)=27$

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;ub= 11;A = [];b = [];Aeq = [];beq = [];
nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,1,A,b,Aeq,beq,lb,ub,nonlcon,intcon)
```

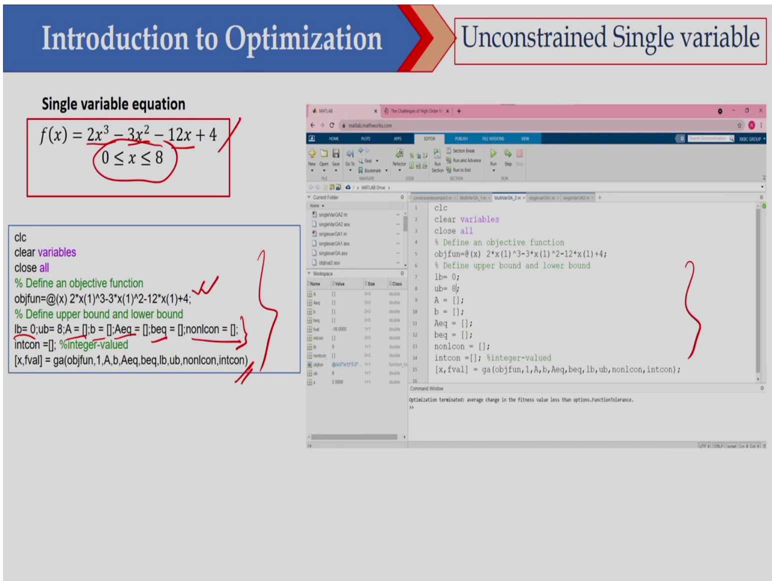
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;
ub= 11;
A = [];
b = [];
Aeq = [];
beq = [];
nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,1,A,b,Aeq,beq,lb,ub,nonlcon,intcon);
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
>>
```

So, let us see let us see how we can solve a simple problem using ga. So, let us start with a single variable optimization problem and that is x square plus 54 by x. So, I know the solution that is here x equal to 3 and f value f 3 equal to 27. So, I know the solution of this particular problem and here I have written this M-code so, objective function is defined then lower bound upper bound A b A equality b equality.

So, that has been defined and non-linear constrain we do not have there is no constrain here and. So, I did not define then and then I have used this ga function ok so, objective function then one ok. So, this is and I have written here and if you are running this particular code then you should get the solution of this problem that is 3 and 27ok, that is 3 and 27.

(Refer Slide Time: 35:07)



Introduction to Optimization

Unconstrained Single variable

Single variable equation

$$f(x) = 2x^3 - 3x^2 - 12x + 4$$
$$0 \leq x \leq 8$$

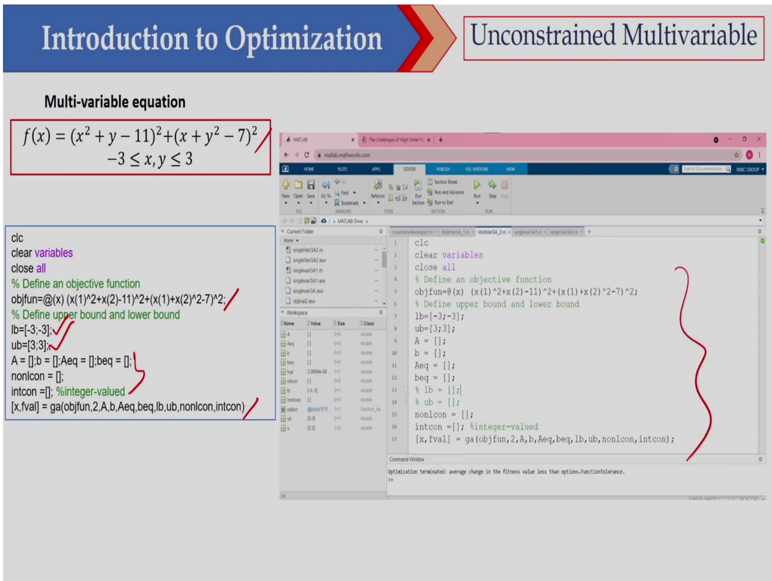
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 2*x(1)^3-3*x(1)^2-12*x(1)+4;
% Define upper bound and lower bound
lb= 0;ub= 8;A = [];b = [];Aeq = [];beq = [];nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,1,A,b,Aeq,beq,lb,ub,nonlcon,intcon)
```

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 2*x(1)^3-3*x(1)^2-12*x(1)+4;
% Define upper bound and lower bound
lb= 0;
ub= 8;
A = [];
b = [];
Aeq = [];
beq = [];
nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,1,A,b,Aeq,beq,lb,ub,nonlcon,intcon);
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
>>
```

Now, let us see another problem this problem is twice x cube plus thrice x square minus 12 x plus 4 and this is between 0 and 8 ok. And this is the code so, I am writing the objective function and this is I have defined. So, I do not have that I am I have defined lower bound and upper bound A b A equality b equality is not there non-linear constrain is not there. So, I am just defining this one. So, if I execute this one I should get the solution of this particular problem.

(Refer Slide Time: 35:42)



Introduction to Optimization

Unconstrained Multivariable

Multi-variable equation

$$f(x) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$
$$-3 \le x, y \le 3$$

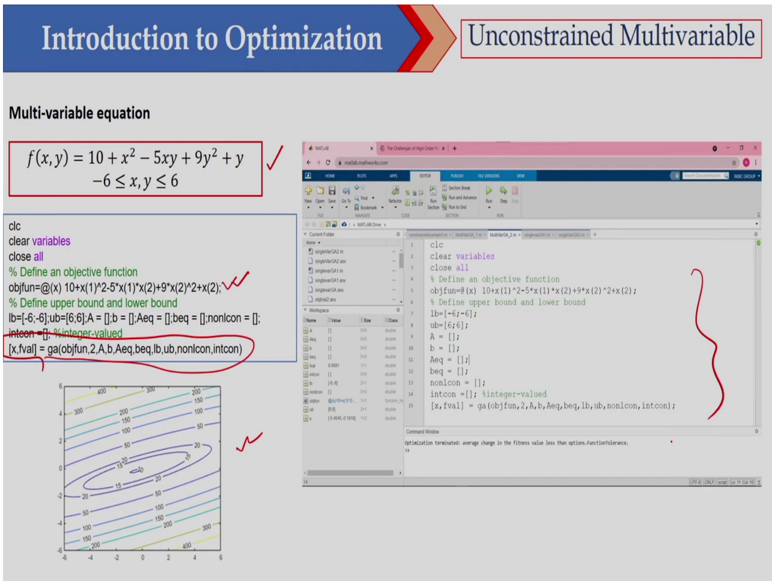
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-3;-3];
ub=[3;3];
A = [];b = [];Aeq = [];beq = [];
nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,2,A,b,Aeq,beq,lb,ub,nonlcon,intcon)
```

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-3;-3];
ub=[3;3];
A = [];
b = [];
Aeq = [];
beq = [];
% lb = [];
% ub = [];
nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,2,A,b,Aeq,beq,lb,ub,nonlcon,intcon);
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
>>
```

Now, this is a multivariable function so here you have defined the multivariable function and I have defined lower bound and upper bound and these are all not there ok. So, I am putting null and then I am using the ga function ok. So, this is the code and if you run it so, you should get the solution so; I will show you that one.

(Refer Slide Time: 36:06)



Introduction to Optimization

Unconstrained Multivariable

Multi-variable equation

$$f(x,y) = 10 + x^2 - 5xy + 9y^2 + y$$
$$-6 \le x, y \le 6$$

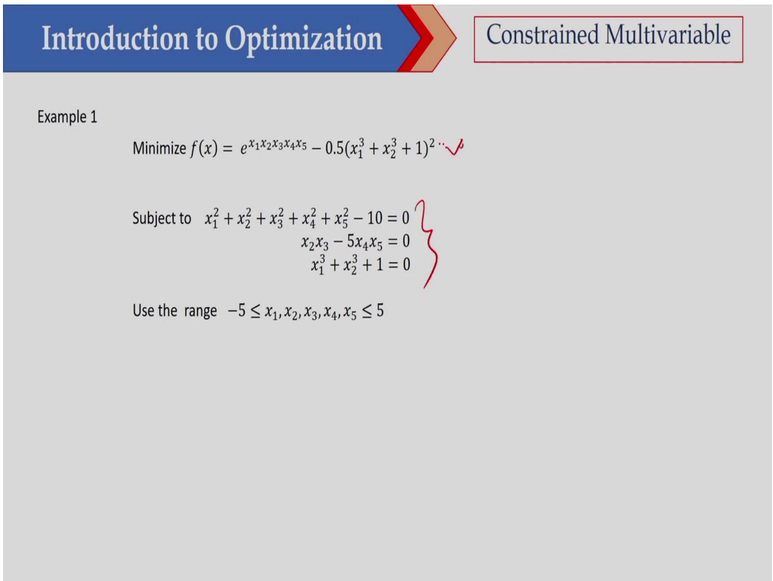
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6;-6];ub=[6;6];A = [];b = [];Aeq = [];beq = [];nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,2,A,b,Aeq,beq,lb,ub,nonlcon,intcon)
```

```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6;-6];
ub=[6;6];
A = [];
b = [];
Aeq = [];
beq = [];
nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,2,A,b,Aeq,beq,lb,ub,nonlcon,intcon);
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
>>
```

So, this is another problem. So, this is again multivariable problem without constrain so here I have defined the function and then you apply this ga function to get the solution of this particular problem ok. So, this is the MATLAB code.

(Refer Slide Time: 36:27)



Introduction to Optimization

Constrained Multivariable

Example 1

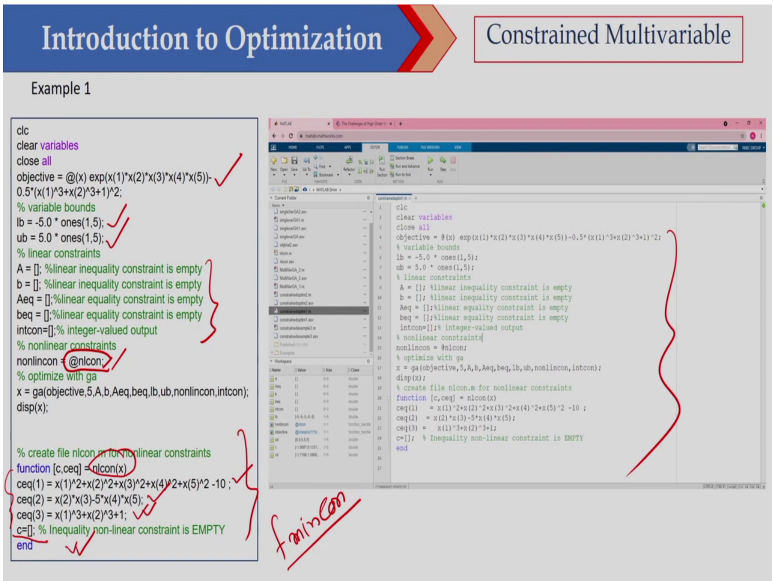
Minimize $f(x) = e^{x_1x_2x_3x_4x_5} - 0.5(x_1^3 + x_2^3 + 1)^2$ ✓

$$\text{Subject to} \quad \left.\begin{aligned} x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 &= 0 \\ x_2x_3 - 5x_4x_5 &= 0 \\ x_1^3 + x_2^3 + 1 &= 0 \end{aligned}\right\}$$

Use the range $-5 \le x_1, x_2, x_3, x_4, x_5 \le 5$

And this is the constrain problem. So, we have three inequality constrains so, the problem has already been defined.

(Refer Slide Time: 36:36)



Introduction to Optimization

Constrained Multivariable

Example 1

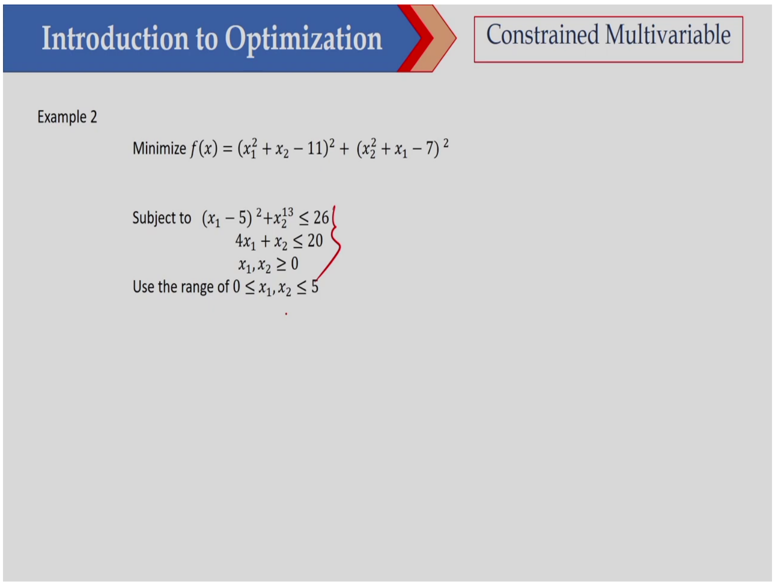
```
clc
clear variables
close all
objective = @(x) exp(x(1)*x(2)*x(3)*x(4)*x(5))-
0.5*(x(1)^3+x(2)^3+1)^2;
% variable bounds
lb = -5.0 * ones(1,5);
ub = 5.0 * ones(1,5);
% linear constraints
A = []; %linear inequality constraint is empty
b = []; %linear inequality constraint is empty
Aeq = [];%linear equality constraint is empty
beq = [];%linear equality constraint is empty
intcon=[];% integer-valued output
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
x = ga(objective,5,A,b,Aeq,beq,lb,ub,nonlincon,intcon);
disp(x);


% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq(1) = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2 -10 ;
ceq(2) = x(2)*x(3)-5*x(4)*x(5);
ceq(3) = x(1)^3+x(2)^3+1;
c=[]; % Inequality non-linear constraint is EMPTY
end
```

So, let me see how we can explain this problem in ga. So, in case of PSO I cannot define the non-linear constrain. So, therefore, I have written my own function for defining the non-linear constrain, but here you can actually define the non-linear constrain separately just like your fmincon.

So, what I have done here. So, this is the objective function then this is the lower bound and upper bound and these are all empty and non-linear constrain is your nlcon so, that is the non-linear function. So, here I have define the non-linear function. So, this is the first constrain, this is the second constrain and this is the third constrain and all of them are equality constrain and there is no inequality constrain so, I am writing it null ok.

So, this is similar to fmincon, fmincon ok function already I have defined these things in the last class ok. So, last class I have defined how you can use fmincon so, there also we have defined this constraint like this. So, I have written this MATLAB code here. So, I will also explain the MATLAB code, when I will go to the MATLAB online so, there I will show you and if I execute this line so, I should get the solution of this problem.

(Refer Slide Time: 37:59)



Introduction to Optimization

Constrained Multivariable

Example 2

Minimize $f(x) = (x_1^2 + x_2 - 11)^2 + (x_2^2 + x_1 - 7)^2$

Subject to $(x_1 - 5)^2 + x_2^{13} \leq 26$

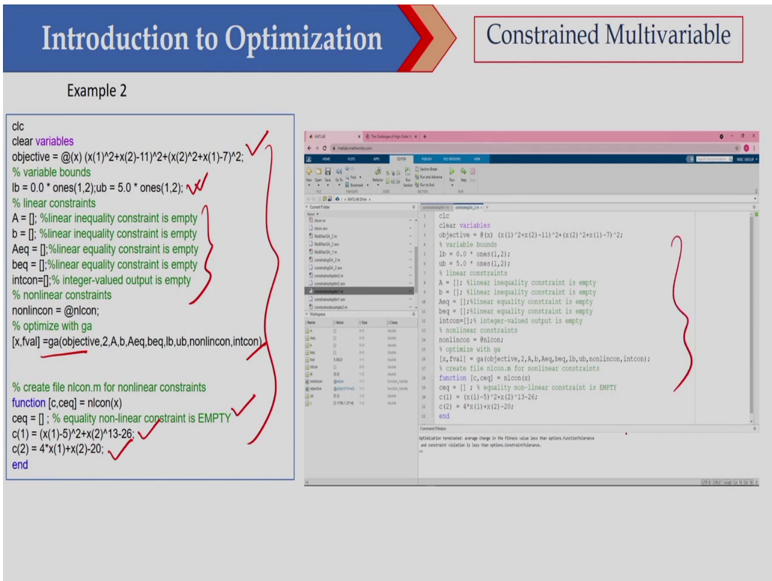
$4x_1 + x_2 \leq 20$

$x_1, x_2 \geq 0$

Use the range of $0 \leq x_1, x_2 \leq 5$

Now, this is another problem with two inequality constraints. So, this problem has already been defined.

(Refer Slide Time: 38:05)



Introduction to Optimization

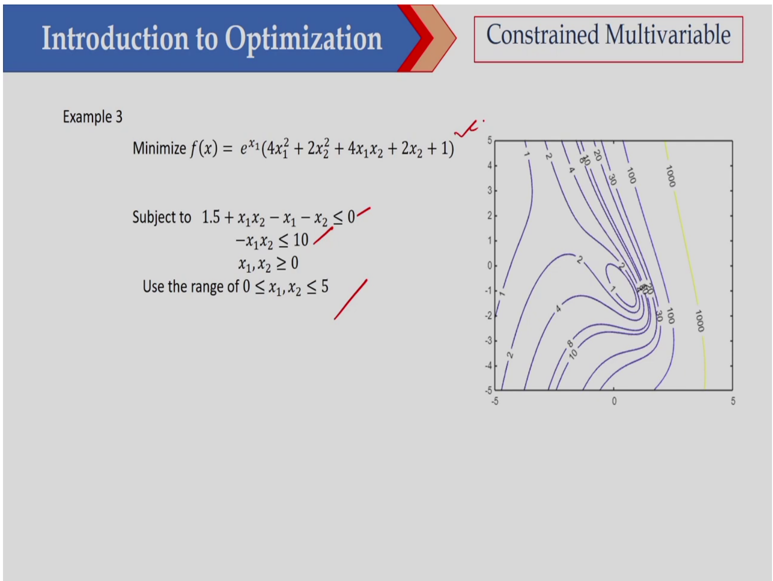
Constrained Multivariable

Example 2

```
clc
clear variables
objective = @(x) (x(1)^2+x(2)-11)^2+(x(2)^2+x(1)-7)^2;
% variable bounds
lb = 0.0 * ones(1,2);ub = 5.0 * ones(1,2);
% linear constraints
A = []; %linear inequality constraint is empty
b = []; %linear inequality constraint is empty
Aeq = [];%linear equality constraint is empty
beq = [];%linear equality constraint is empty
intcon=[];% integer-valued output is empty
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
[x,fval] =ga(objective,2,A,b,Aeq,beq,lb,ub,nonlincon,intcon)


% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq = [] ; % equality non-linear constraint is EMPTY
c(1) = (x(1)-5)^2+x(2)^13-26;
c(2) = 4*x(1)+x(2)-20;
end
```

And this is the MATLAB code here so, this is the objective function then you are defining lower bound and these are all empty, then if you are using the ga function ok. So, here you have defined these two constrain so, there is no equality constrain so, that is empty ok, null you are putting and there are two inequality constrains c 1 and c 2 ok. So, I have written the code here.

(Refer Slide Time: 38:32)



Introduction to Optimization

Constrained Multivariable

Example 3

Minimize $f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$

Subject to $1.5 + x_1x_2 - x_1 - x_2 \leq 0$

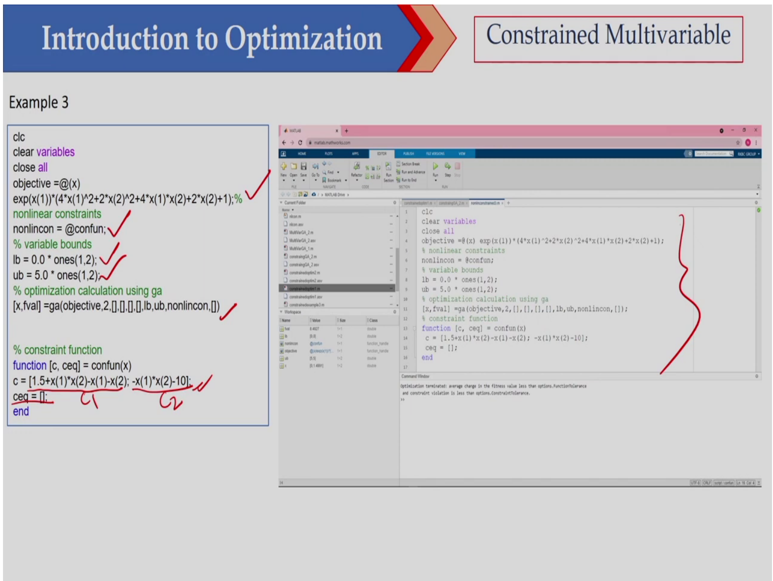
$-x_1x_2 \leq 10$

$x_1, x_2 \geq 0$

Use the range of $0 \leq x_1, x_2 \leq 5$

And this is the last problem. So, here we have two inequality constrain with non-linear objective function.

(Refer Slide Time: 38:40)



Introduction to Optimization

Constrained Multivariable

Example 3

```
clc
clear variables
close all
objective =@(x)
exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);%
nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon,[])


% constraint function
function [c, ceq] = confun(x)
c = [1.5+x(1)*x(2)-x(1)-x(2); -x(1)*x(2)-10];
ceq = [];
end
```

C1 C2

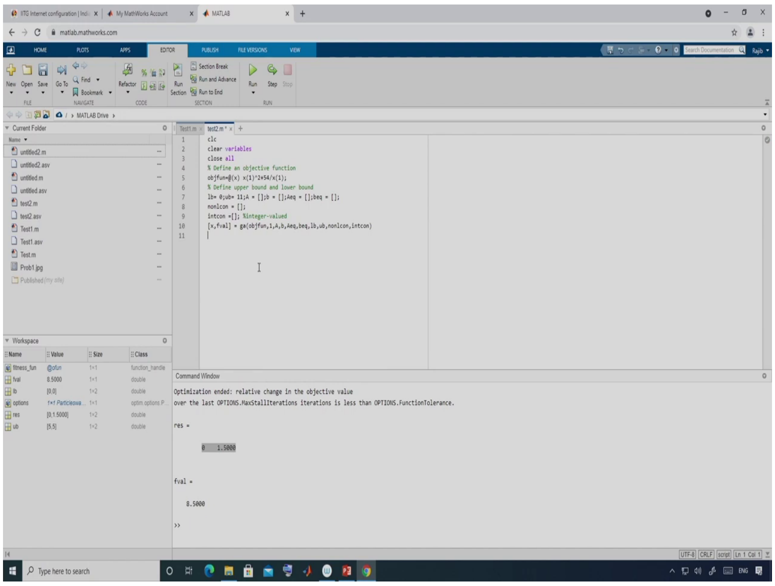
```
clc
clear variables
close all
objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon,[]);
% constraint function
function [c, ceq] = confun(x)
 c = [1.5+x(1)*x(2)-x(1)-x(2); -x(1)*x(2)-10];
 ceq = [];
end
```

Command Window

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.
>>
```

And this is the non-linear objective function I have defined and then this is the con function lower bound upper bound and this is the ga. So, here we have two inequality constrains so, this is the first constrain this is C 1 and this is your second constrain this is C 2 and that is the second non-linear constrain and we do not have any inequality type constrain and if you are running this one, then you should get the solution of this particular problem. So, let us go to MATLAB and just see.

(Refer Slide Time: 39:22)



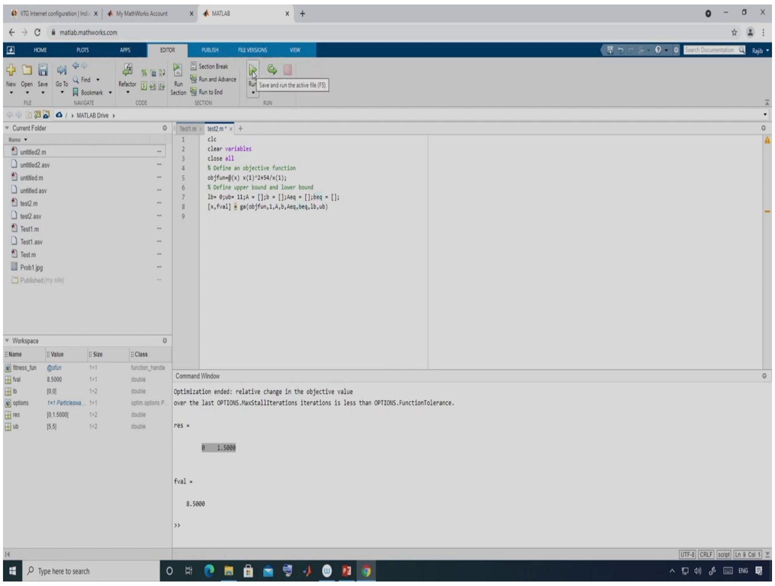
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;ub= 11;A = [];b = [];Aeq = [];beq = [];
nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,1,A,b,Aeq,beq,lb,ub,nonlcon,intcon)
```

```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

res =

        0    1.5000


fval =

    8.5000

>>
```

Name	Value	Size	Class
fitness_fun	@ofun	1×1	function_handle
fval	8.5000	1×1	double
lb	[0,0]	1×2	double
options	1×1 Particleswa...	1×1	optim.options.P...
res	[0,1.5000]	1×2	double
ub	[5,5]	1×2	double

So, let me explain from here. So, let me solve the first problem using ga.

(Refer Slide Time: 39:35)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;ub= 11;A = [];b = [];Aeq = [];beq = [];
[x,fval] = ga(objfun,1,A,b,Aeq,beq,lb,ub)
```

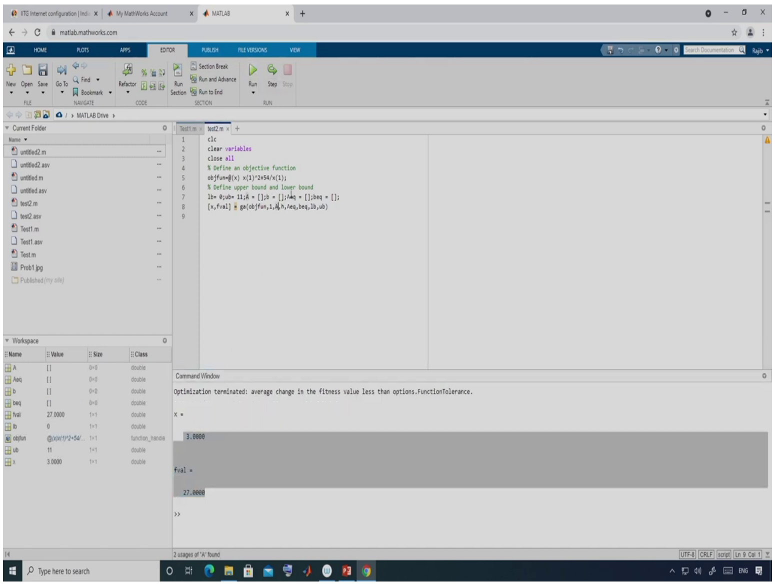
```
Optimization ended: relative change in the objective value
over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.

res =

     0    1.5000


fval =

    8.5000

>>
```

So, this is the first problem and I will put it here and here we have only objective function and lower bound and upper bound ok so; that means, this part is not there so, I can remove also so, because this is not there. And I have lower bound and upper bound.

So, up to here and then I have defined lower bound upper bound then A b A equality b equality I have defined and non-linear constrain is not there. So, I can also remove this one so, this is not there ok. So, these few lines are sufficient so, I have defined objective function and then I am using this particular ga function.

So, here I have defined the objective function number of variable is 1, then A b A equality b equality is null then lower bound and upper bound I have defined ok. So, let me run this particular AMP file

(Refer Slide Time: 40:17)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;ub= 11;A = [];b = [];Aeq = [];beq = [];
[x,fval] = ga(objfun,1,A,b,Aeq,beq,lb,ub)
```

Command Window

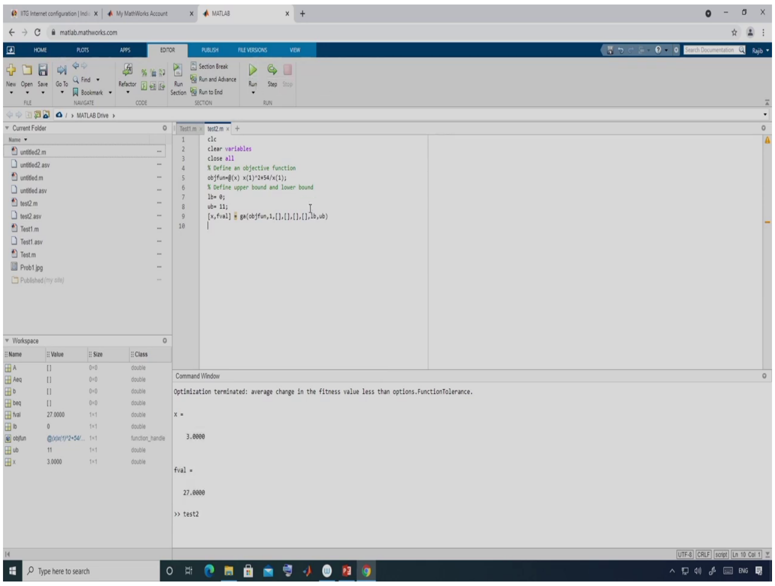
```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    3.0000


fval =

   27.0000

>>
```

Workspace

Name	Value	Size	Class
A	[]	0×0	double
Aeq	[]	0×0	double
b	[]	0×0	double
beq	[]	0×0	double
fval	27.0000	1×1	double
lb	0	1×1	double
objfun	@(x)x(1)^2+54/...	1×1	function_handle
ub	11	1×1	double
x	3.0000	1×1	double

So, I am getting the solution that is 3 2.

(Refer Slide Time: 40:22)



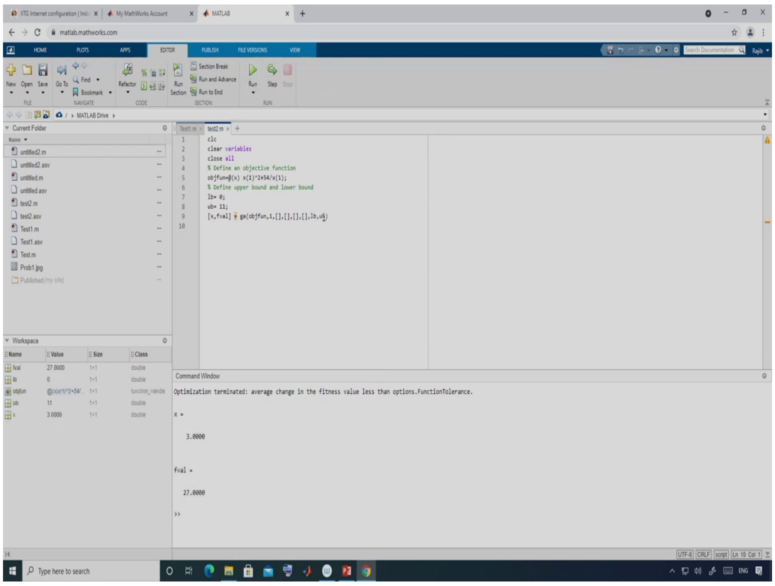
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;
ub= 11;
[x,fval] = ga(objfun,1,[],[],[],[],lb,ub)
```

Name	Value	Size	Class
A	[]	0×0	double
Aeq	[]	0×0	double
b	[]	0×0	double
beq	[]	0×0	double
fval	27.0000	1×1	double
lb	0	1×1	double
objfun	@(x)x(1)^2+54/...	1×1	function_handle
ub	11	1×1	double
x	3.0000	1×1	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    3.0000


fval =

   27.0000

>> test2
```

So, what I can do I can also put the null here. So, this is the empty matrix. So, I can empty matrix of vector I can put it and lower bound and upper bound I have define ok. So; that means, these particular things I can remove ok. So, I have defined lower bound and upper bound and then I am just executing this particular line. So, let me see whether I am getting the solution or not.

(Refer Slide Time: 41:01)



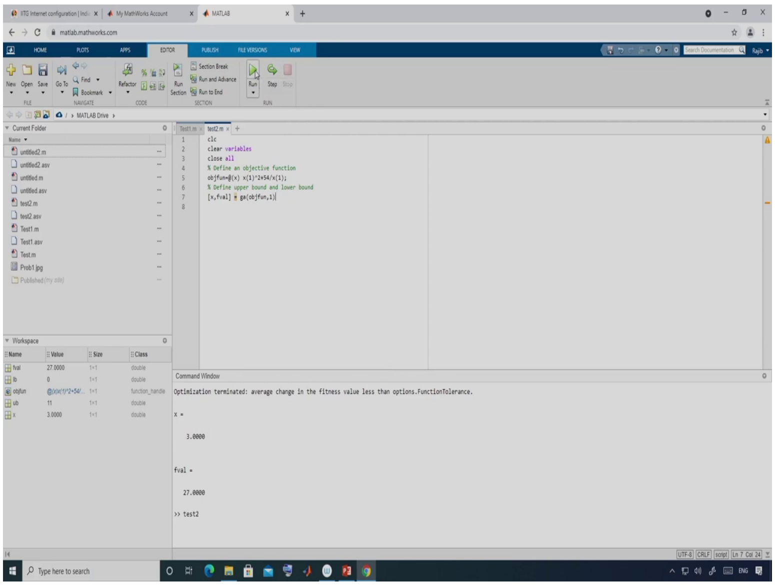
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
lb= 0;
ub= 11;
[x,fval] = ga(objfun,1,[],[],[],[],lb,ub)
```

Name	Value	Size	Class
fval	27.0000	1×1	double
lb	0	1×1	double
objfun	@(x)x(1)^2+54/...	1×1	function_handle
ub	11	1×1	double
x	3.0000	1×1	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    3.0000


fval =

   27.0000

>>
```

Yes, I am getting the solution.

(Refer Slide Time: 41:06)



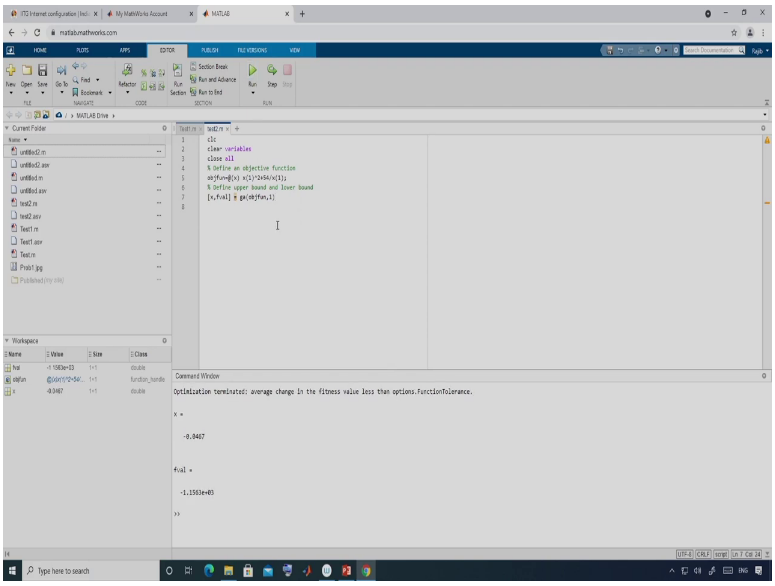
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
[x,fval] = ga(objfun,1)
```

Name	Value	Size	Class
fval	27.0000	1×1	double
lb	0	1×1	double
objfun	@(x)x(1)^2+54/...	1×1	function_handle
ub	11	1×1	double
x	3.0000	1×1	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    3.0000


fval =

   27.0000

>> test2
```

Suppose, if I am not putting anything so, what will happen. So, only I am defining objective function and number of variable that is one ok. So, I do not need this lines also let me see, whether I am getting the solution of this particular problem.

(Refer Slide Time: 41:27)

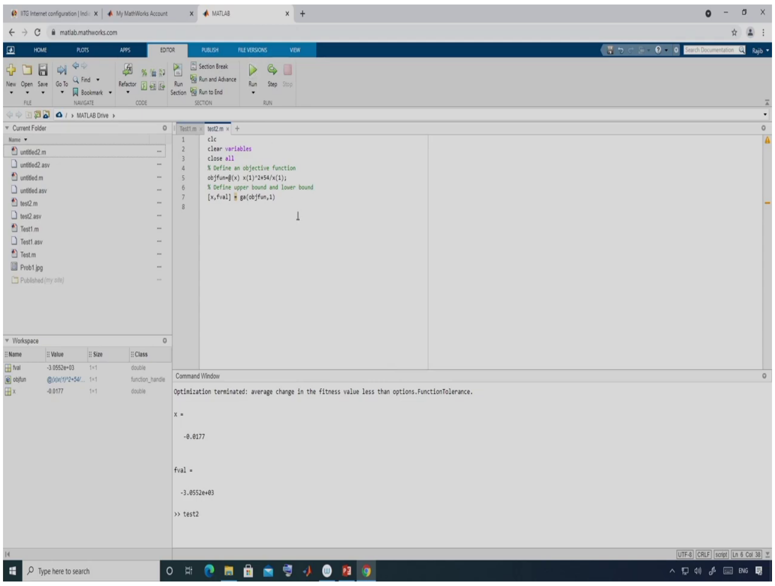


```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
[x,fval] = ga(objfun,1)
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

   -0.0467


fval =

  -1.1563e+03

>>
```

So, I am not getting so; that means, j is starting from your either 0 0 or somewhere. So, that is why I am not getting this one. So, I have to define the lower bound and upper bound ok so, then only I will get the solution, but if you increase your population. So, you may get the solution also you can try that one, but otherwise you put the lower bound and upper bound to get the solution ok.

(Refer Slide Time: 42:03)

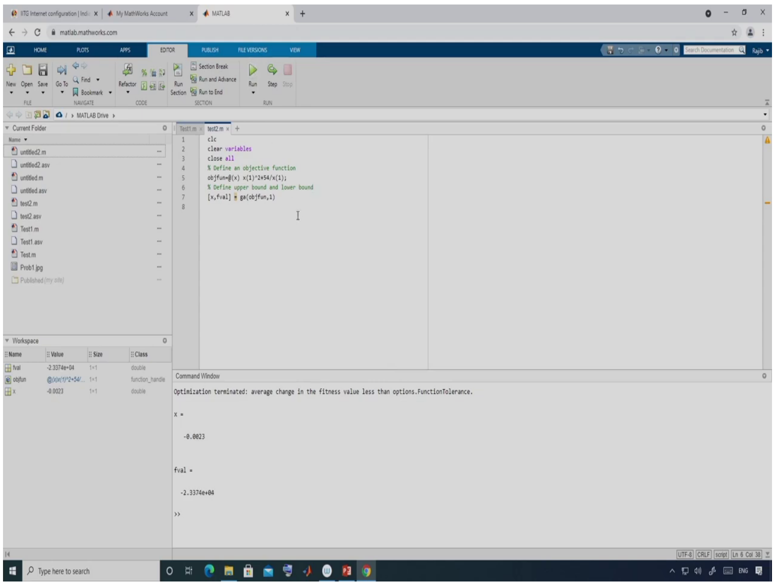


```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
[x,fval] = ga(objfun,1)
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

   -0.0177


fval =

  -3.0552e+03

>> test2
```

Name	Value	Size	Class
fval	-3.0552e+03	1×1	double
objfun	@(x)x(1)^2+54/...	1×1	function_handle
x	-0.0177	1×1	double

(Refer Slide Time: 42:05)



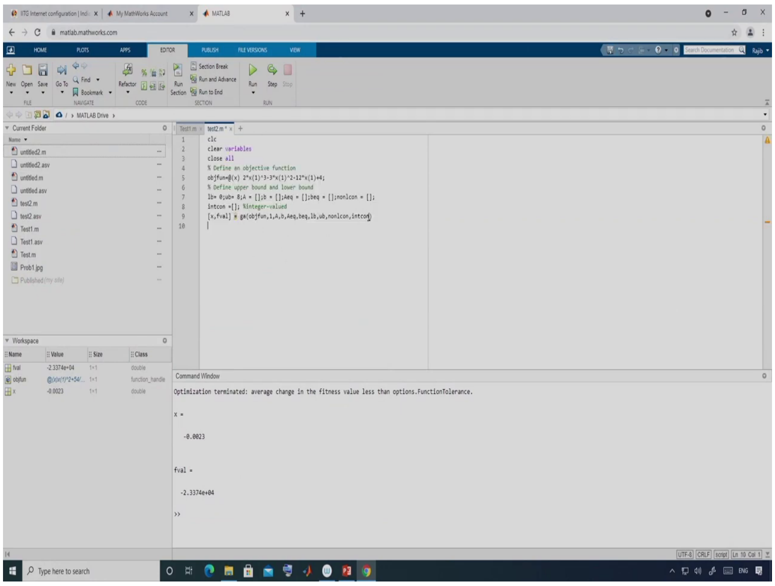
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) x(1)^2+54/x(1);
% Define upper bound and lower bound
[x,fval] = ga(objfun,1)
```

Name	Value	Size	Class
fval	-2.3374e+04	1×1	double
objfun	@(x)x(1)^2+54/...	1×1	function_handle
x	-0.0023	1×1	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

  -0.0023


fval =

  -2.3374e+04

>>
```

Or otherwise you can increase your population size or you can also increase the number of iteration so, you may get the solution. So, let me try the second function. So, this function I have defined here so, let me copy and yeah.

(Refer Slide Time: 42:21)



```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 2*x(1)^3-3*x(1)^2-12*x(1)+4;
% Define upper bound and lower bound
lb= 0;ub= 8;A = [];b = [];Aeq = [];beq = [];nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,1,A,b,Aeq,beq,lb,ub,nonlcon,intcon)
```

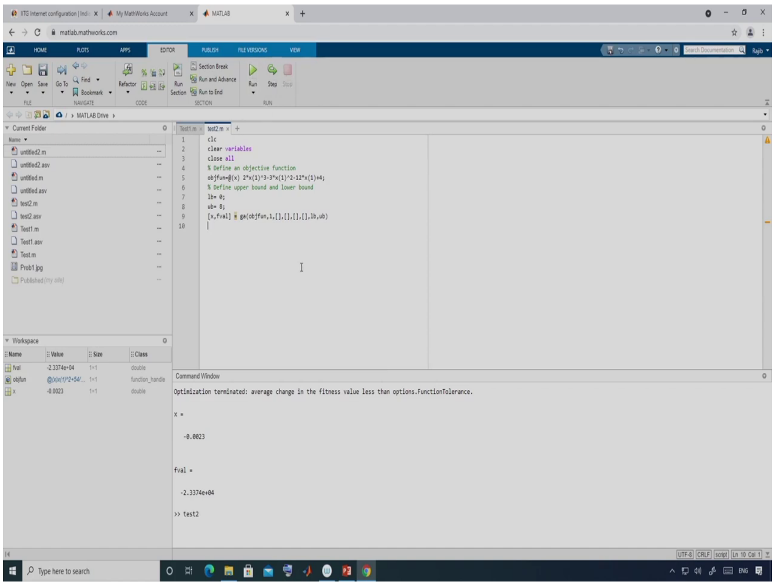
Workspace

Name	Value	Size	Class
fval	-2.3374e+04	1×1	double
objfun	@(x)x(1)^2+54/...	1×1	function_handle
x	-0.0023	1×1	double

Command Window

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

   -0.0023


fval =

  -2.3374e+04

>>
```

(Refer Slide Time: 42:26)

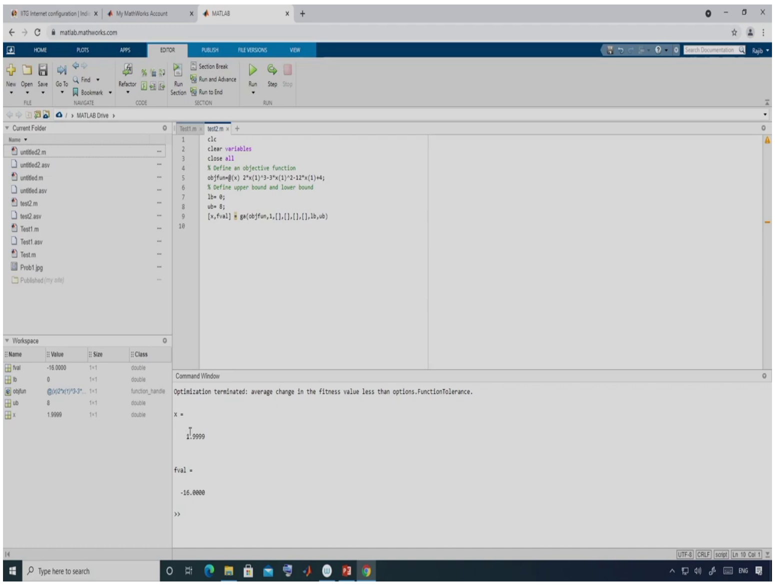


```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 2*x(1)^3-3*x(1)^2-12*x(1)+4;
% Define upper bound and lower bound
lb= 0;
ub= 8;
[x,fval] = ga(objfun,1,[],[],[],[],lb,ub)
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

  -0.0023


fval =

 -2.3374e+04

>> test2
```

So, here I do not have any constrain so, let me delete this part ok. So, I do not have this one so, I am just deleting this part and I am defining A b A equality b equality and lower bound and upper bound I have define ok so, this is not required. So, I have defined lower bound and upper bound, lower bound is 0 upper bound is 8, ok. So, let me run this particular m file yeah.

(Refer Slide Time: 43:14)



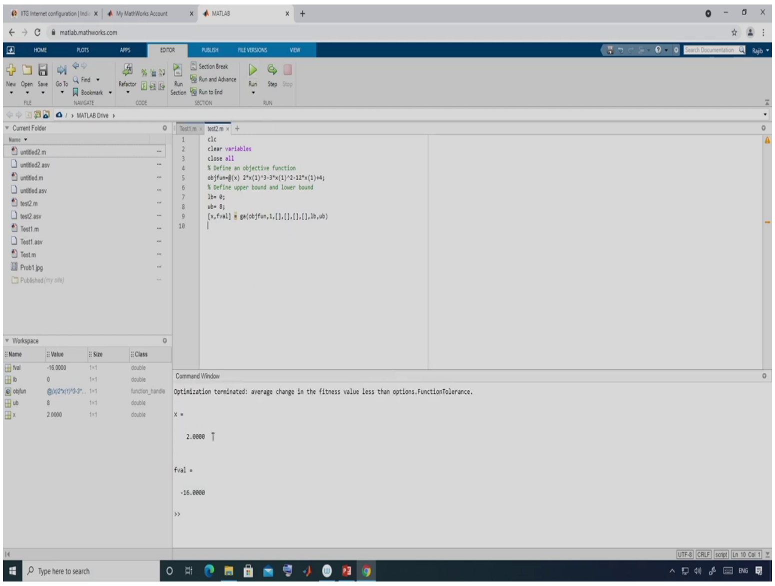
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 2*x(1)^3-3*x(1)^2-12*x(1)+4;
% Define upper bound and lower bound
lb= 0;
ub= 8;
[x,fval] = ga(objfun,1,[],[],[],[],lb,ub)
```

Name	Value	Size	Class
fval	-16.0000	1×1	double
lb	0	1×1	double
objfun	@(x)2*x(1)^3-3*...	1×1	function_handle
ub	8	1×1	double
x	1.9999	1×1	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    1.9999


fval =

  -16.0000

>>
```

So, I am getting actual solution is 2 and 16. So, I am getting 1.999.

(Refer Slide Time: 43:23)



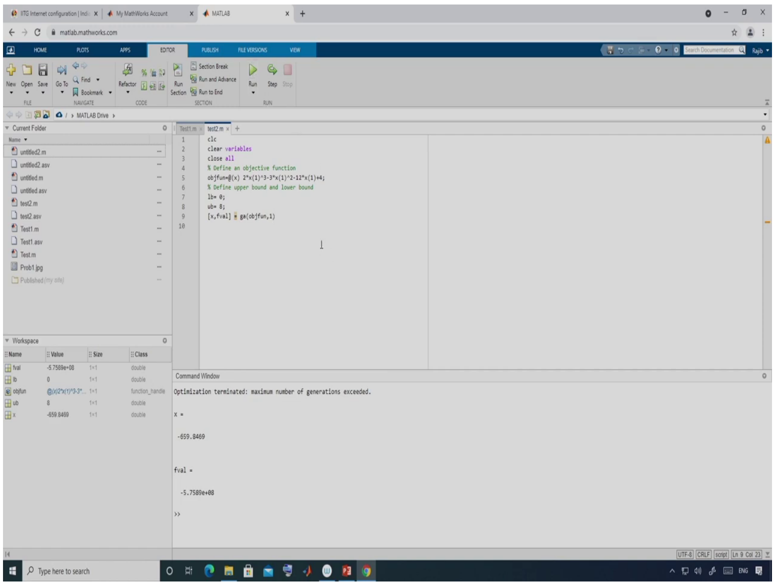
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 2*x(1)^3-3*x(1)^2-12*x(1)+4;
% Define upper bound and lower bound
lb= 0;
ub= 8;
[x,fval] = ga(objfun,1,[],[],[],[],lb,ub)
```

Name	Value	Size	Class
fval	-16.0000	1×1	double
lb	0	1×1	double
objfun	@(x)2*x(1)^3-3*...	1×1	function_handle
ub	8	1×1	double
x	2.0000	1×1	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.


x =

    2.0000


fval =

  -16.0000

>>
```

But anyway, let us see. So, I am getting 2 and 16 ok so, second time I am getting 2 and 16. So, this is the solution of this particular problem.

(Refer Slide Time: 43:46)

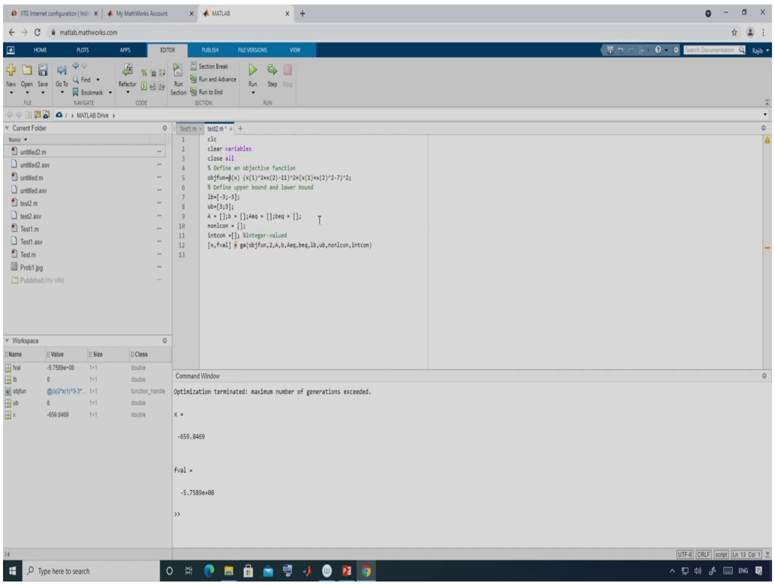


```
clc
clear variables
close all
% Define an objective function
objfun=@(x) 2*x(1)^3-3*x(1)^2-12*x(1)+4;
% Define upper bound and lower bound
lb= 0;
ub= 8;
[x,fval] = ga(objfun,1)
```

```
Optimization terminated: maximum number of generations exceeded.

x =

 -659.8469


fval =

  -5.7589e+08

>>
```

So, this problem you can try without defining the lower bound upper bound just see. So, I am not getting that solution so, I am getting a different solution here. So, let me try this next problem so, this is a multi-variable problem without constrain.

(Refer Slide Time: 43:58)



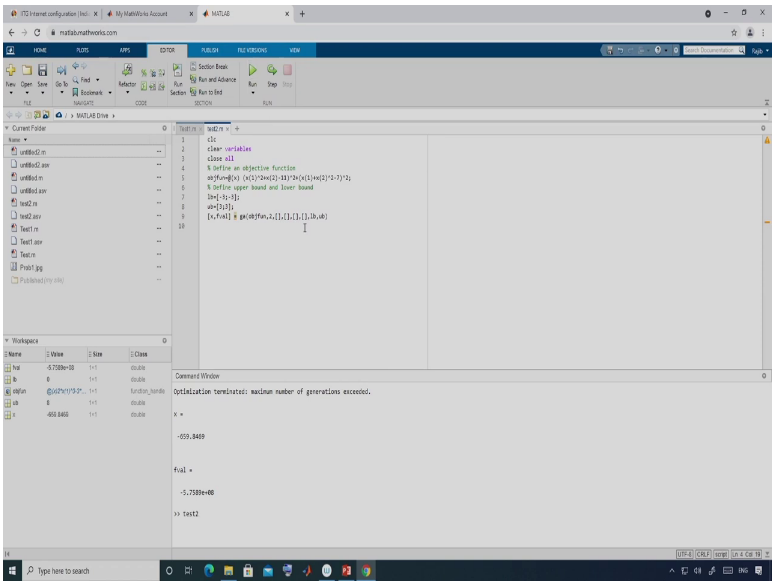
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-3;-3];
ub=[3;3];
A = [];b = [];Aeq = [];beq = [];
nonlcon = [];
intcon =[]; %integer-valued
[x,fval] = ga(objfun,2,A,b,Aeq,beq,lb,ub,nonlcon,intcon)
```

Name	Value	Size	Class
fval	-5.7589e+08	1×1	double
lb	0	1×1	double
objfun	@(x)2*x(1)^3-3*...	1×1	function_handle
ub	8	1×1	double
x	-659.8469	1×1	double

```
Optimization terminated: maximum number of generations exceeded.

x =

 -659.8469


fval =

  -5.7589e+08

>>
```

So, this is a multivariable problem without constrain. So, I do not have constrain here.

(Refer Slide Time: 44:06)



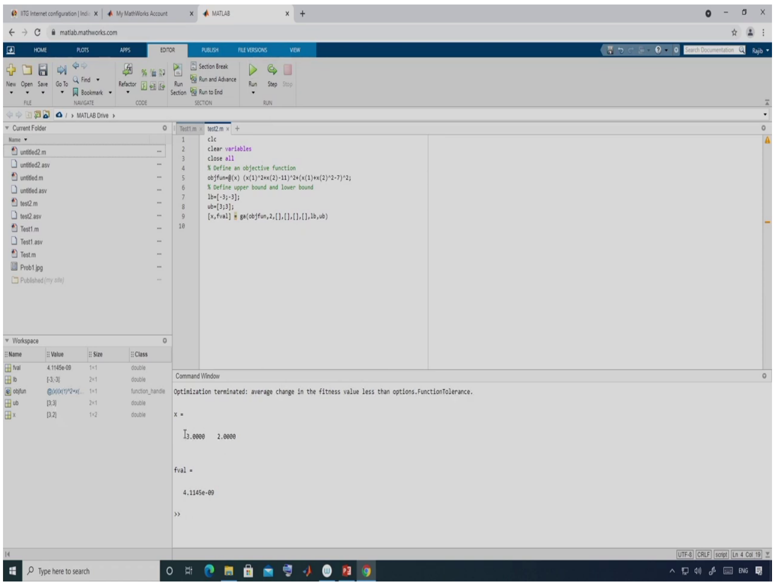
```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-3;-3];
ub=[3;3];
[x,fval] = ga(objfun,2,[],[],[],[],lb,ub)
```

```
Optimization terminated: maximum number of generations exceeded.

x =

 -659.8469


fval =

  -5.7589e+08

>> test2
```

So, I can remove this part. So, this is not required and I am defining the A b A equality b equality and lower bound and upper bound. So, this is I do not need this one ok so, I do not need this one. So, I have defined the lower bound and upper bound so, this is minus 3 minus 3 and plus 3 plus 3 and this is the objective function.

So, you need the objective function and you need and you need the lower bound and upper bound ok. So, n number of variable is 2 in this case.

(Refer Slide Time: 44:55)

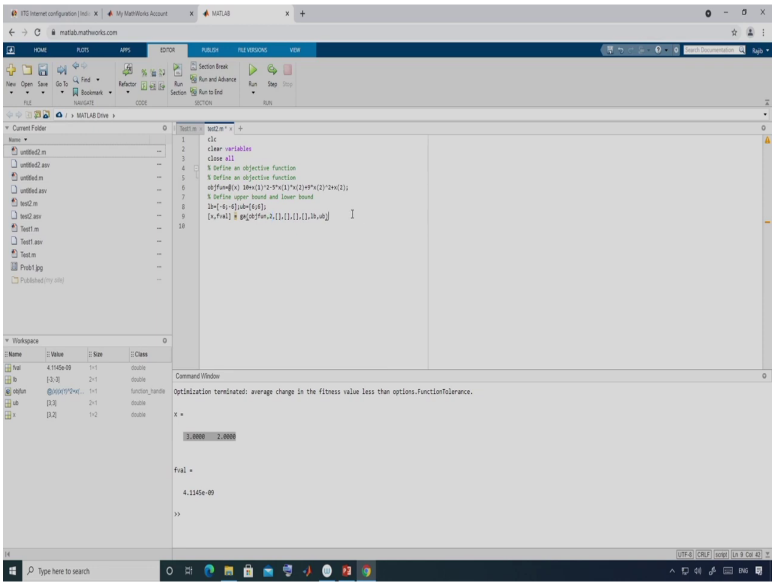


```
clc
clear variables
close all
% Define an objective function
objfun=@(x) (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;
% Define upper bound and lower bound
lb=[-3;-3];
ub=[3;3];
[x,fval] = ga(objfun,2,[],[],[],[],lb,ub)
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    3.0000    2.0000


fval =

   4.1145e-09

>>
```

So, let me see yes, I am getting one solution ok. So, this is between minus 3 and plus 3, ok then let me go to the next problem. So, this problem is again a multivariable problem. So, what I can do yeah, I can copy this part.

(Refer Slide Time: 45:15)



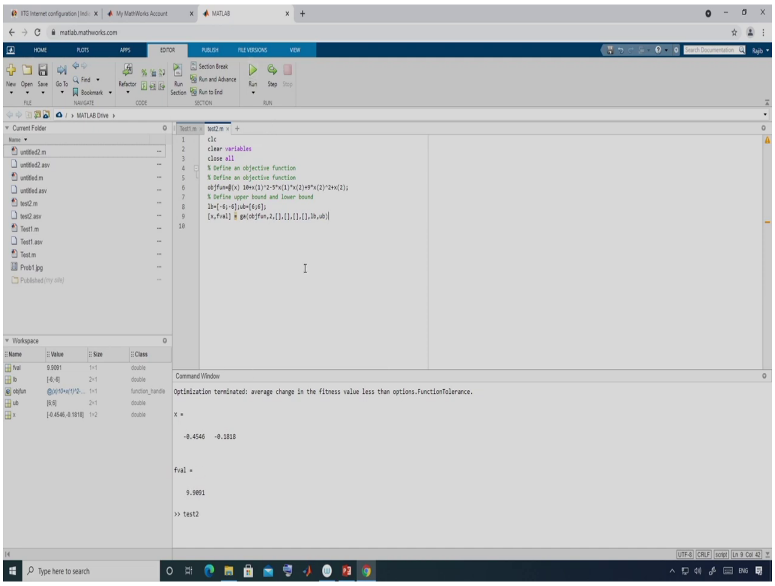
```
clc
clear variables
close all
% Define an objective function
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6;-6];ub=[6;6];
[x,fval] = ga(objfun,2,[],[],[],[],lb,ub)
```

Name	Value	Size	Class
fval	4.1145e-09	1×1	double
lb	[-3;-3]	2×1	double
objfun	@(x)(x(1)^2+x(...	1×1	function_handle
ub	[3;3]	2×1	double
x	[3,2]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

   3.0000    2.0000


fval =

   4.1145e-09

>>
```

So, I have defined the objective function and then lower bound and upper bound that is n number of variables r 2 ok.

(Refer Slide Time: 45:28)



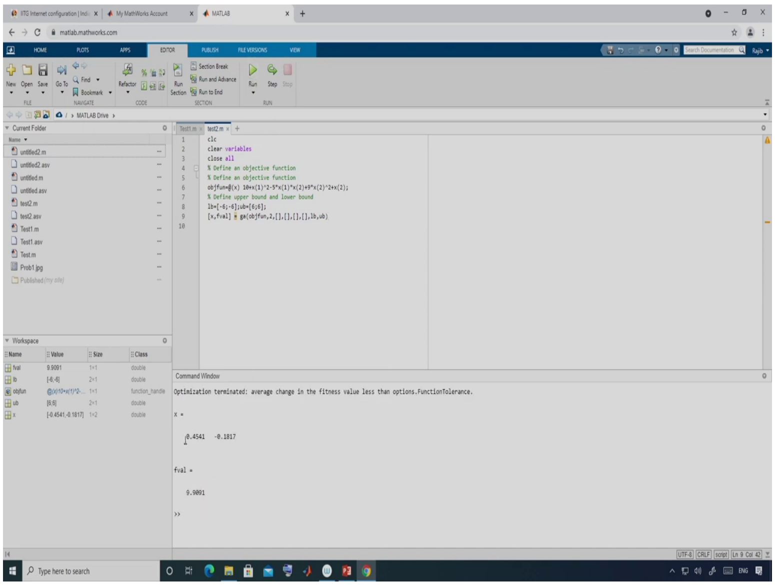
```
clc
clear variables
close all
% Define an objective function
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6;-6];ub=[6;6];
[x,fval] = ga(objfun,2,[],[],[],[],lb,ub)
```

Name	Value	Size	Class
fval	9.9091	1×1	double
lb	[-6;-6]	2×1	double
objfun	@(x)10+x(1)^2-...	1×1	function_handle
ub	[6;6]	2×1	double
x	[-0.4546,-0.1818]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

   -0.4546   -0.1818


fval =

    9.9091

>> test2
```

So, let me execute this one yeah I am getting a solution ok.

(Refer Slide Time: 45:33)

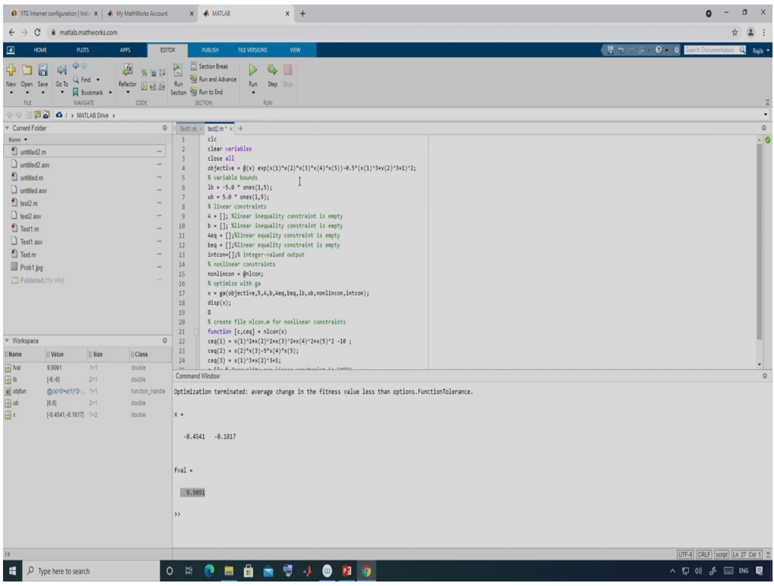


```
clc
clear variables
close all
% Define an objective function
% Define an objective function
objfun=@(x) 10+x(1)^2-5*x(1)*x(2)+9*x(2)^2+x(2);
% Define upper bound and lower bound
lb=[-6;-6];ub=[6;6];
[x,fval] = ga(objfun,2,[],[],[],[],lb,ub)
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    0.4541   -0.1817


fval =

    9.9091

>>
```

So, this is the solution of this particular problem so, second time also I am getting the same solution. So, now, let me try a constrained problem. So, this is the constrained problem and this is the code of this particular, this is the code of, this is the code to solve this particular problem ok.

(Refer Slide Time: 45:53)



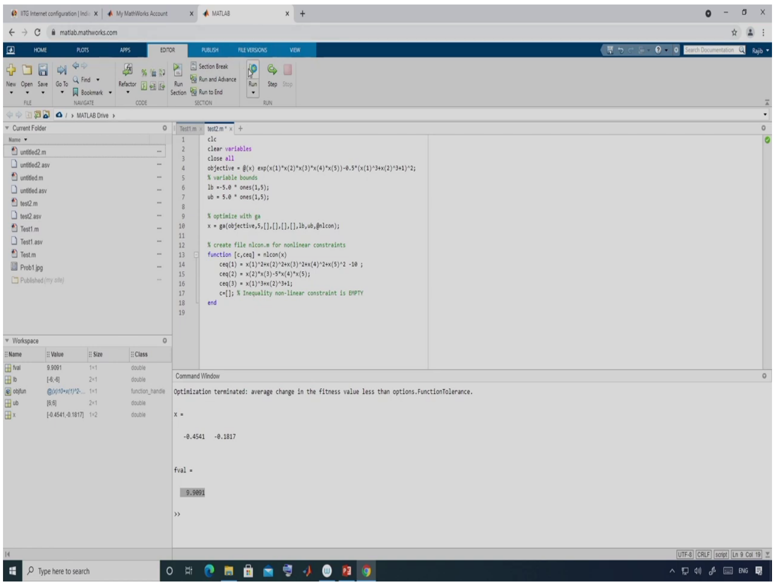
```
clc
clear variables
close all
objective = @(x) exp(x(1)*x(2)*x(3)*x(4)*x(5))-0.5*(x(1)^3+x(2)^3+1)^2;
% variable bounds
lb = -5.0 * ones(1,5);
ub = 5.0 * ones(1,5);
% linear constraints
A = []; %linear inequality constraint is empty
b = []; %linear inequality constraint is empty
Aeq = [];%linear equality constraint is empty
beq = [];%linear equality constraint is empty
intcon=[];% integer-valued output
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
x = ga(objective,5,A,b,Aeq,beq,lb,ub,nonlincon,intcon);
disp(x);

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq(1) = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2 -10 ;
ceq(2) = x(2)*x(3)-5*x(4)*x(5);
ceq(3) = x(1)^3+x(2)^3+1;
```

Name	Value	Size	Class
fval	9.9091	1×1	double
lb	[-6;-6]	2×1	double
objfun	@(x)10+x(1)^2-...	1×1	function_handle
ub	[6;6]	2×1	double
x	[-0.4541,-0.1817]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

   -0.4541   -0.1817


fval =

    9.9091

>>
```

So, here you just see.

(Refer Slide Time: 46:02)



```
clc
clear variables
close all
objective = @(x) exp(x(1)*x(2)*x(3)*x(4)*x(5))-0.5*(x(1)^3+x(2)^3+1)^2;
% variable bounds
lb =-5.0 * ones(1,5);
ub = 5.0 * ones(1,5);

% optimize with ga
x = ga(objective,5,[],[],[],[],lb,ub,@nlcon);

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
    ceq(1) = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2 -10 ;
    ceq(2) = x(2)*x(3)-5*x(4)*x(5);
    ceq(3) = x(1)^3+x(2)^3+1;
    c=[]; % Inequality non-linear constraint is EMPTY
end
```

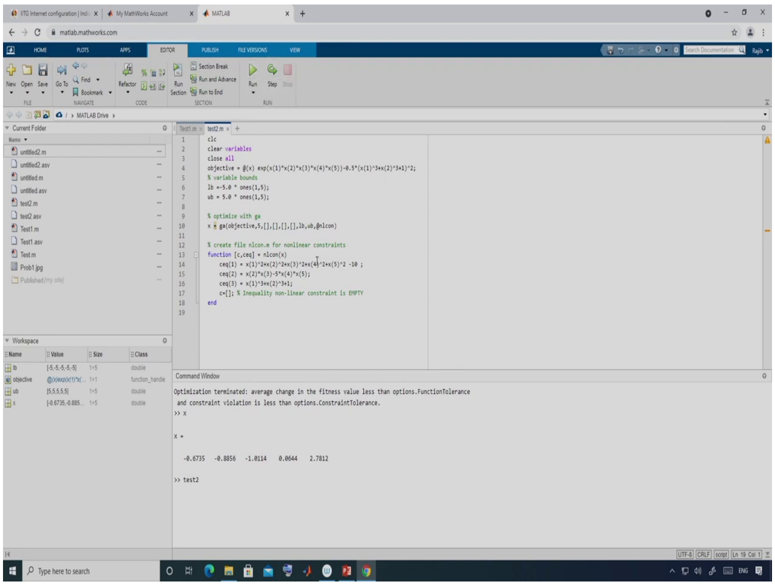
```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

   -0.4541   -0.1817


fval =

    9.9091

>>
```

Name	Value	Size	Class
fval	9.9091	1×1	double
lb	[-6;-6]	2×1	double
objfun	@(x)10+x(1)^2-...	1×1	function_handle
ub	[6;6]	2×1	double
x	[-0.4541,-0.1817]	1×2	double

Anyway, so, I would like to delete this part in order to reduce the number of lines ok. So, I am defining here, I am defining here this is null, this is null, this is null, this is also empty ok and this is integer value is not there and I can define that n con somewhere here itself ok. So, this is not required ok and this is not ok and.

Then I have defined the constrain function here. So, we have three constrain so, this is the first one ok, this is the second one, this is the third one, and that c inequality constrains are not there so, therefore, this is empty and I hope, I have defined the problem here and if you execute this particular m file so, you should get the solution of this problem.

(Refer Slide Time: 47:29)

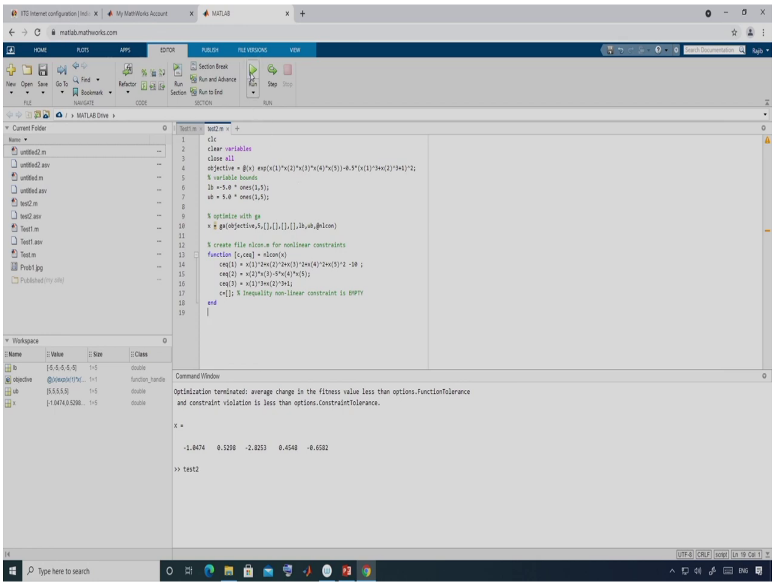


```
clc
clear variables
close all
objective = @(x) exp(x(1)*x(2)*x(3)*x(4)*x(5))-0.5*(x(1)^3+x(2)^3+1)^2;
% variable bounds
lb =-5.0 * ones(1,5);
ub = 5.0 * ones(1,5);

% optimize with ga
x = ga(objective,5,[],[],[],[],lb,ub,@nlcon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
    ceq(1) = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2 -10 ;
    ceq(2) = x(2)*x(3)-5*x(4)*x(5);
    ceq(3) = x(1)^3+x(2)^3+1;
    c=[]; % Inequality non-linear constraint is EMPTY
end
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.
>> x

x =

  -0.6735   -0.8856   -1.0114    0.0644    2.7812

>> test2
```

So, let me see so, I did not so, I can check what is the solution is yeah. So, this is the solution of this particular problem you can check we have already solved this problem using classical method.

(Refer Slide Time: 47:46)

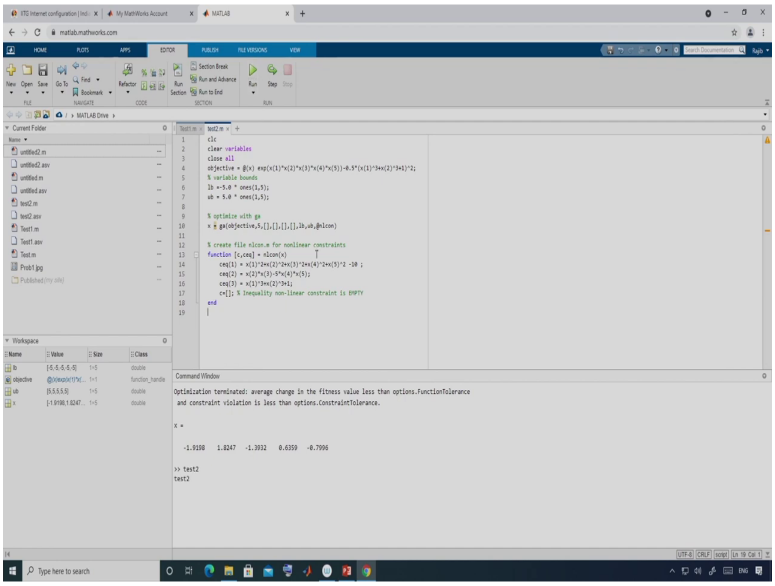


```
clc
clear variables
close all
objective = @(x) exp(x(1)*x(2)*x(3)*x(4)*x(5))-0.5*(x(1)^3+x(2)^3+1)^2;
% variable bounds
lb =-5.0 * ones(1,5);
ub = 5.0 * ones(1,5);

% optimize with ga
x = ga(objective,5,[],[],[],[],lb,ub,@nlcon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
    ceq(1) = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2 -10 ;
    ceq(2) = x(2)*x(3)-5*x(4)*x(5);
    ceq(3) = x(1)^3+x(2)^3+1;
    c=[]; % Inequality non-linear constraint is EMPTY
end
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =


   -1.0474    0.5298   -2.8253    0.4548   -0.6582

>> test2
```

So, you can also see actually what is the solution you got using classical method.

(Refer Slide Time: 47:49)



```
clc
clear variables
close all
objective = @(x) exp(x(1)*x(2)*x(3)*x(4)*x(5))-0.5*(x(1)^3+x(2)^3+1)^2;
% variable bounds
lb =-5.0 * ones(1,5);
ub = 5.0 * ones(1,5);

% optimize with ga
x = ga(objective,5,[],[],[],[],lb,ub,@nlcon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
    ceq(1) = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2 -10 ;
    ceq(2) = x(2)*x(3)-5*x(4)*x(5);
    ceq(3) = x(1)^3+x(2)^3+1;
    c=[]; % Inequality non-linear constraint is EMPTY
end
```

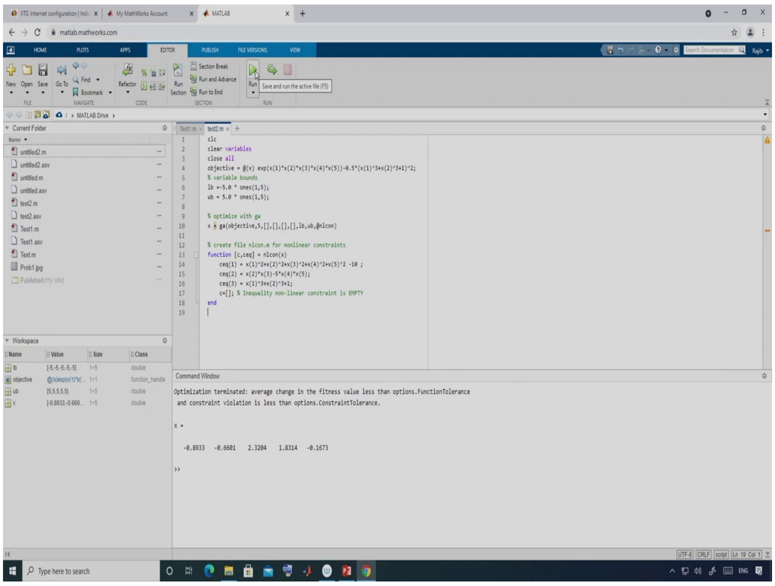
Workspace

Name	Value	Size	Class
lb	[-5,-5,-5,-5,-5]	1×5	double
objective	@(x)exp(x(1)*x(...	1×1	function_handle
ub	[5,5,5,5,5]	1×5	double
x	[-1.9198,1.8247...	1×5	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =


   -1.9198    1.8247   -1.3932    0.6359   -0.7996


>> test2
test2
```

So, this is yeah. So, you are getting some different solution ok.

(Refer Slide Time: 47:54)



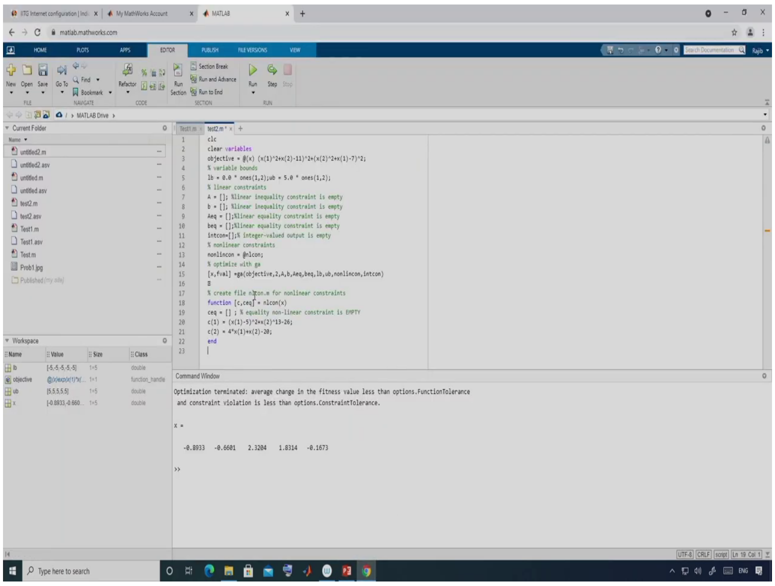
```
clc
clear variables
close all
objective = @(x) exp(x(1)*x(2)*x(3)*x(4)*x(5))-0.5*(x(1)^3+x(2)^3+1)^2;
% variable bounds
lb =-5.0 * ones(1,5);
ub = 5.0 * ones(1,5);

% optimize with ga
x = ga(objective,5,[],[],[],[],lb,ub,@nlcon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
    ceq(1) = x(1)^2+x(2)^2+x(3)^2+x(4)^2+x(5)^2 -10 ;
    ceq(2) = x(2)*x(3)-5*x(4)*x(5);
    ceq(3) = x(1)^3+x(2)^3+1;
    c=[]; % Inequality non-linear constraint is EMPTY
end
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.

x =

   -0.8933   -0.6601    2.3204    1.8314   -0.1673

>>
```

So, you are not getting the same solution so, then we have to see which one is the actual solution of this particular problem. So, if you are not getting the actual solution then what you can, then what you can do? You can use the hybrid method ok to get the exact optimal solution of the problem. So, in this case you may not get the exact optimal solution of the problem because we are not using gradient information here.

So, in that case you can use the hybrid method ok. Then let me go to the next problem.

(Refer Slide Time: 48:36)

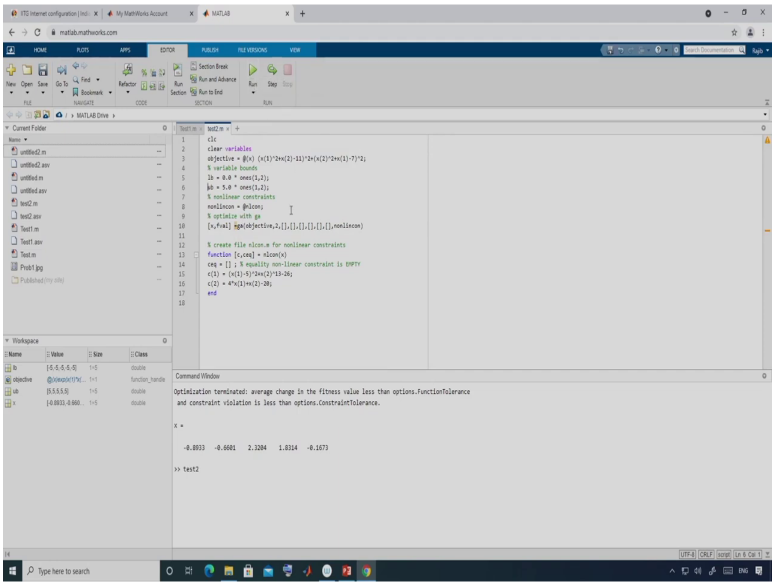


```
clc
clear variables
objective = @(x) (x(1)^2+x(2)-11)^2+(x(2)^2+x(1)-7)^2;
% variable bounds
lb = 0.0 * ones(1,2);ub = 5.0 * ones(1,2);
% linear constraints
A = []; %linear inequality constraint is empty
b = []; %linear inequality constraint is empty
Aeq = [];%linear equality constraint is empty
beq = [];%linear equality constraint is empty
intcon=[];% integer-valued output is empty
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
[x,fval] =ga(objective,2,A,b,Aeq,beq,lb,ub,nonlincon,intcon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq = [] ; % equality non-linear constraint is EMPTY
c(1) = (x(1)-5)^2+x(2)^13-26;
c(2) = 4*x(1)+x(2)-20;
end
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =


   -0.8933   -0.6601    2.3204    1.8314   -0.1673


>>
```

So, next problem we have two inequality constrain. So, let me see that one.

(Refer Slide Time: 48:41)



```
clc
clear variables
objective = @(x) (x(1)^2+x(2)-11)^2+(x(2)^2+x(1)-7)^2;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
[x,fval] =ga(objective,2,[],[],[],[],[],[],nonlincon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq = [] ; % equality non-linear constraint is EMPTY
c(1) = (x(1)-5)^2+x(2)^13-26;
c(2) = 4*x(1)+x(2)-20;
end
```

Name	Value	Size	Class
lb	[-5,-5,-5,-5,-5]	1×5	double
objective	@(x)exp(x(1)*x(...	1×1	function_handle
ub	[5,5,5,5,5]	1×5	double
x	[-0.8933,-0.660...	1×5	double

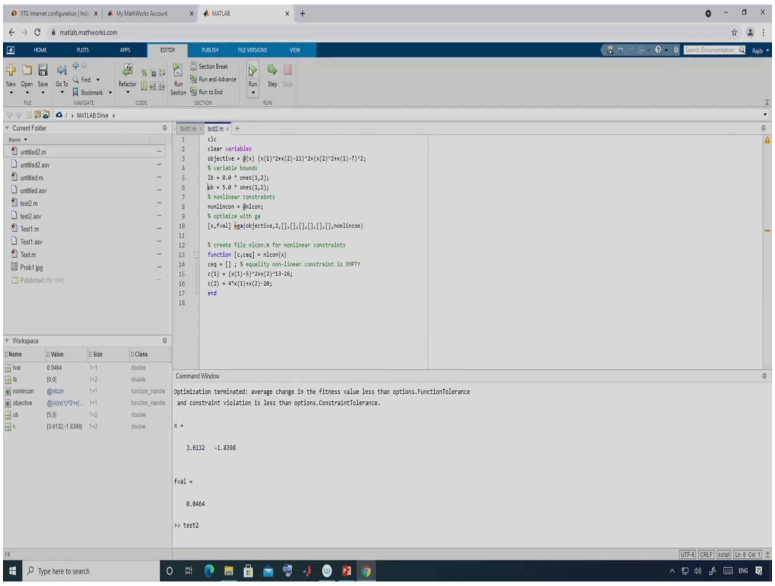
```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.

x =

  -0.8933   -0.6601    2.3204    1.8314   -0.1673

>> test2
```

So, here I am defining here itself. So, we have non-linear constrain, but this is not there so, I am removing it.

So, here you just see we have we do not have any equality constrain. So, this is empty and we have two inequality constrain so that I am putting c 1 and c 2. So, after that I am running this particular line in order to get the solution and then solution your x and x value you will get that the objective function value you will get at optimal solution that you will get and I have defined this problem.

So, lower bound I have defined so, lower bound is 0 and upper bound is 5 here. So, lower bound and upper bound is defined here and so, if I run this particular file so, I should get the solution of the problem.

(Refer Slide Time: 49:55)



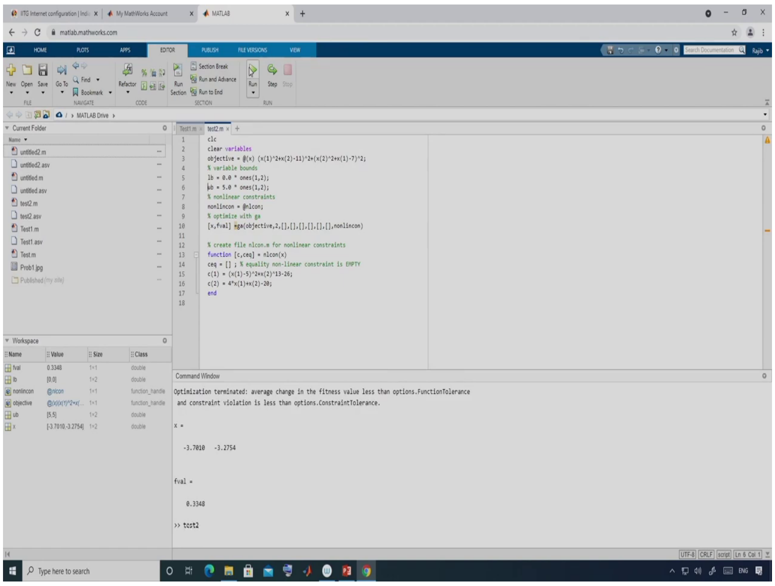
```
clc
clear variables
objective = @(x) (x(1)^2+x(2)-11)^2+(x(2)^2+x(1)-7)^2;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
[x,fval] =ga(objective,2,[],[],[],[],[],[],nonlincon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq = [] ; % equality non-linear constraint is EMPTY
c(1) = (x(1)-5)^2+x(2)^13-26;
c(2) = 4*x(1)+x(2)-20;
end
```

Name	Value	Size	Class
fval	0.0464	1×1	double
lb	[0,0]	1×2	double
nonlincon	@nlcon	1×1	function_handle
objective	@(x)(x(1)^2+x(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[3.6132,-1.8398]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =

    3.6132   -1.8398


fval =

    0.0464

>> test2
```

So, I am getting a constrain solution, just see this is one solution.

(Refer Slide Time: 50:00)



```
clc
clear variables
objective = @(x) (x(1)^2+x(2)-11)^2+(x(2)^2+x(1)-7)^2;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
[x,fval] =ga(objective,2,[],[],[],[],[],[],nonlincon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq = [] ; % equality non-linear constraint is EMPTY
c(1) = (x(1)-5)^2+x(2)^13-26;
c(2) = 4*x(1)+x(2)-20;
end
```

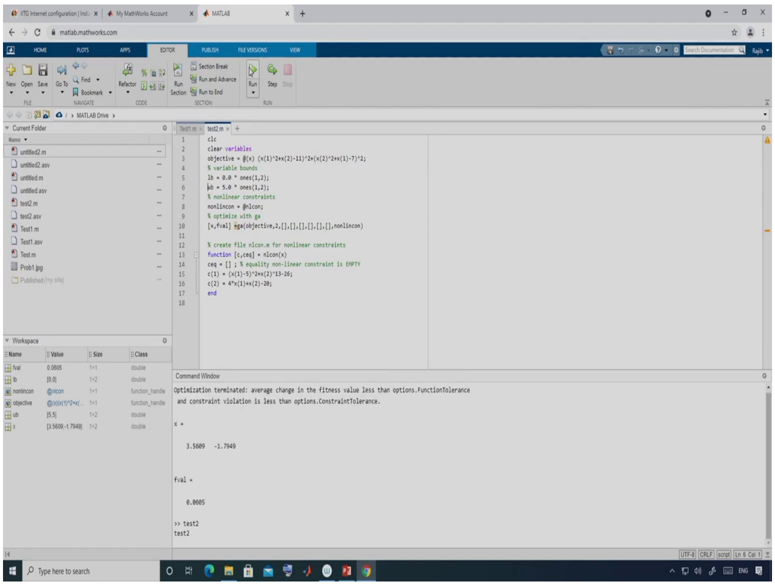
Workspace

Name	Value	Size	Class
fval	0.3348	1×1	double
lb	[0,0]	1×2	double
nonlincon	@nlcon	1×1	function_handle
objective	@(x)(x(1)^2+x(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[-3.7010,-3.2754]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.

x =

   -3.7010   -3.2754


fval =

    0.3348

>> test2
```

This is another solution yeah.

(Refer Slide Time: 50:02)

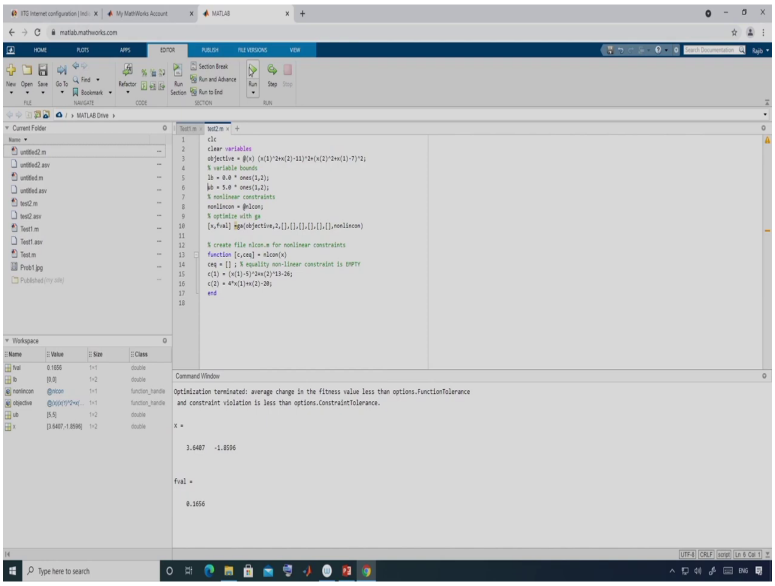


```
clc
clear variables
objective = @(x) (x(1)^2+x(2)-11)^2+(x(2)^2+x(1)-7)^2;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
[x,fval] =ga(objective,2,[],[],[],[],[],[],nonlincon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq = [] ; % equality non-linear constraint is EMPTY
c(1) = (x(1)-5)^2+x(2)^13-26;
c(2) = 4*x(1)+x(2)-20;
end
```

Name	Value	Size	Class
fval	0.0605	1×1	double
lb	[0,0]	1×2	double
nonlincon	@nlcon	1×1	function_handle
objective	@(x)(x(1)^2+x(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[3.5609,-1.7949]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =


    3.5609   -1.7949



fval =


    0.0605

>> test2
test2
```

(Refer Slide Time: 50:05)

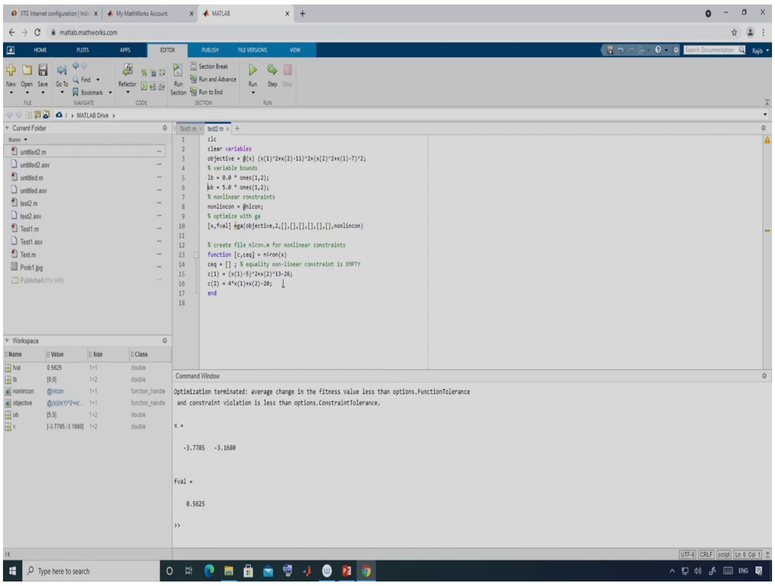


```
clc
clear variables
objective = @(x) (x(1)^2+x(2)-11)^2+(x(2)^2+x(1)-7)^2;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
[x,fval] =ga(objective,2,[],[],[],[],[],[],nonlincon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq = [] ; % equality non-linear constraint is EMPTY
c(1) = (x(1)-5)^2+x(2)^13-26;
c(2) = 4*x(1)+x(2)-20;
end
```

Name	Value	Size	Class
fval	0.1656	1×1	double
lb	[0,0]	1×2	double
nonlincon	@nlcon	1×1	function_handle
objective	@(x)(x(1)^2+x(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[3.6407,-1.8596]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.

x =

    3.6407   -1.8596


fval =

    0.1656
```

(Refer Slide Time: 50:06)



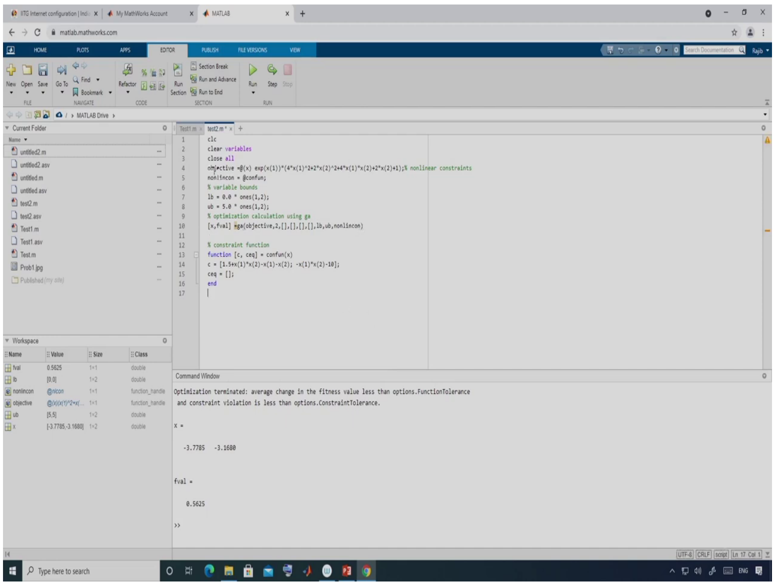
```
clc
clear variables
objective = @(x) (x(1)^2+x(2)-11)^2+(x(2)^2+x(1)-7)^2;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% nonlinear constraints
nonlincon = @nlcon;
% optimize with ga
[x,fval] =ga(objective,2,[],[],[],[],[],[],nonlincon)

% create file nlcon.m for nonlinear constraints
function [c,ceq] = nlcon(x)
ceq = [] ; % equality non-linear constraint is EMPTY
c(1) = (x(1)-5)^2+x(2)^13-26;
c(2) = 4*x(1)+x(2)-20;
end
```

Name	Value	Size	Class
fval	0.5625	1×1	double
lb	[0,0]	1×2	double
nonlincon	@nlcon	1×1	function_handle
objective	@(x)(x(1)^2+x(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[-3.7785,-3.1680]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.

x =

   -3.7785   -3.1680


fval =

    0.5625

>>
```

So, you are getting a, the constrain solution of this particular problem. Let me execute this particular function so, this is the third problem so, third problem is so, we have two inequality constrain.

(Refer Slide Time: 50:24)



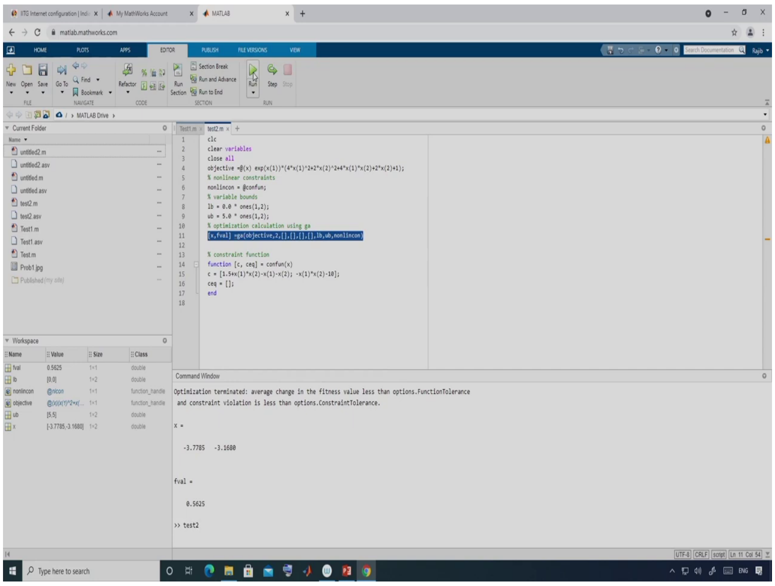
```
clc
clear variables
close all
objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
c = [1.5+x(1)*x(2)-x(1)-x(2); -x(1)*x(2)-10];
ceq = [];
end
```

Name	Value	Size	Class
fval	0.5625	1×1	double
lb	[0,0]	1×2	double
nonlincon	@nlcon	1×1	function_handle
objective	@(x)(x(1)^2+x(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[-3.7785,-3.1680]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.

x =

   -3.7785   -3.1680


fval =

    0.5625

>>
```

So, here I have defined the objective function.

(Refer Slide Time: 50:33)



```
clc
clear variables
close all
objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
c = [1.5+x(1)*x(2)-x(1)-x(2); -x(1)*x(2)-10];
ceq = [];
end
```

Workspace

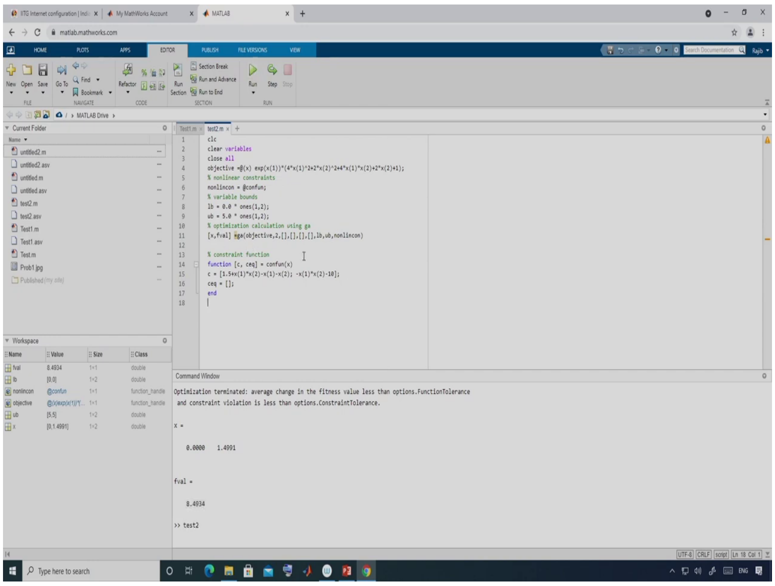
Name	Value	Size	Class
fval	0.5625	1×1	double
lb	[0,0]	1×2	double
nonlincon	@nlcon	1×1	function_handle
objective	@(x)(x(1)^2+x(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[-3.7785,-3.1680]	1×2	double

Command Window

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =

   -3.7785   -3.1680


fval =

    0.5625

>> test2
```

Then this is the non-linear constrain so, non-linear constrains defined here. So, we have two constrains here so, this is the first one and this is the second one ok. And there is no equality type constrain. And lower bound and upper bound has been defined then after that if you are executing this particular line so, you should get the solution of this particular problem yeah.

(Refer Slide Time: 51:04)



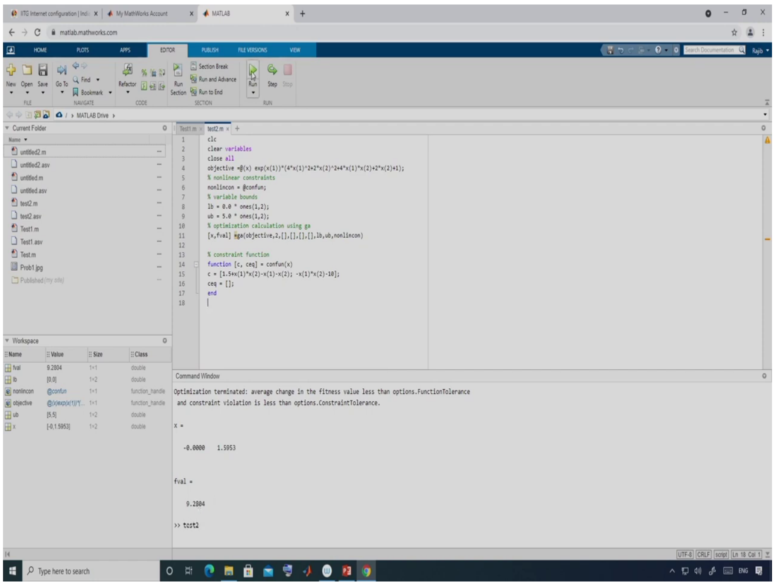
```
clc
clear variables
close all
objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
c = [1.5+x(1)*x(2)-x(1)-x(2); -x(1)*x(2)-10];
ceq = [];
end
```

Name	Value	Size	Class
fval	8.4934	1×1	double
lb	[0,0]	1×2	double
nonlincon	@confun	1×1	function_handle
objective	@(x)exp(x(1))*(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[0,1.4991]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =


    0.0000    1.4991



fval =


    8.4934

>> test2
```

So, I am getting solution is 0 and 1.5. So, 0 and 1.5 is the solution.

(Refer Slide Time: 51:15)



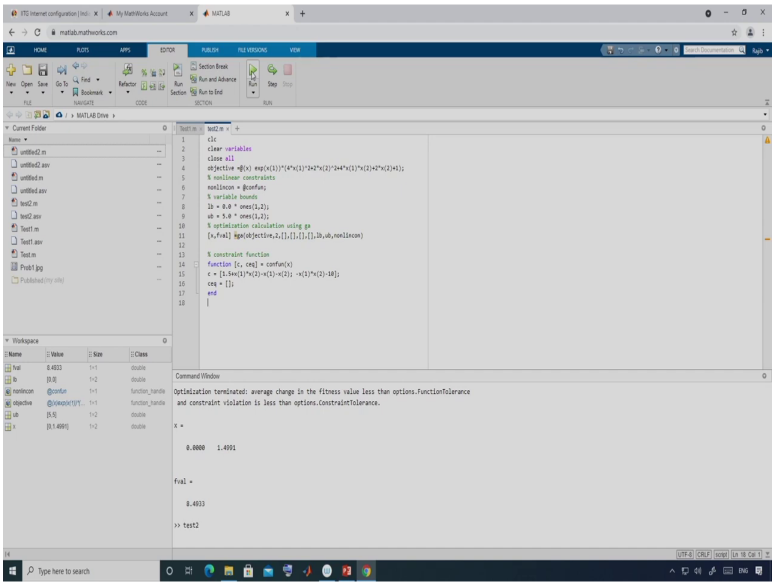
```
clc
clear variables
close all
objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
c = [1.5+x(1)*x(2)-x(1)-x(2); -x(1)*x(2)-10];
ceq = [];
end
```

Name	Value	Size	Class
fval	9.2804	1×1	double
lb	[0,0]	1×2	double
nonlincon	@confun	1×1	function_handle
objective	@(x)exp(x(1))*(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[-0,1.5953]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.

x =

  -0.0000    1.5953


fval =

    9.2804

>> test2
```

And you can see so, I am getting the solution ok 0 and 1.59 I am getting.

(Refer Slide Time: 51:20)



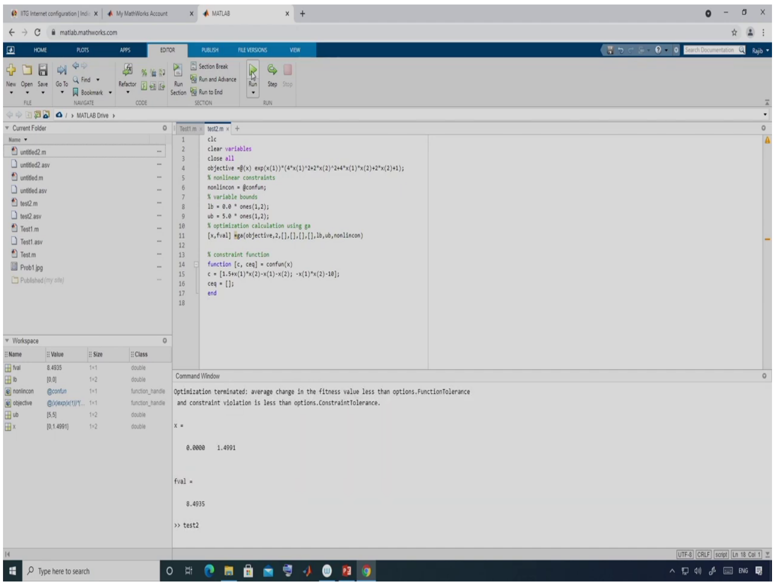
```
clc
clear variables
close all
objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
c = [1.5+x(1)*x(2)-x(1)-x(2); -x(1)*x(2)-10];
ceq = [];
end
```

Name	Value	Size	Class
fval	8.4933	1×1	double
lb	[0,0]	1×2	double
nonlincon	@confun	1×1	function_handle
objective	@(x)exp(x(1))*(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[0,1.4991]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =

    0.0000    1.4991


fval =

    8.4933

>> test2
```

But this solution is 0 and 1.5 ok.

(Refer Slide Time: 51:23)

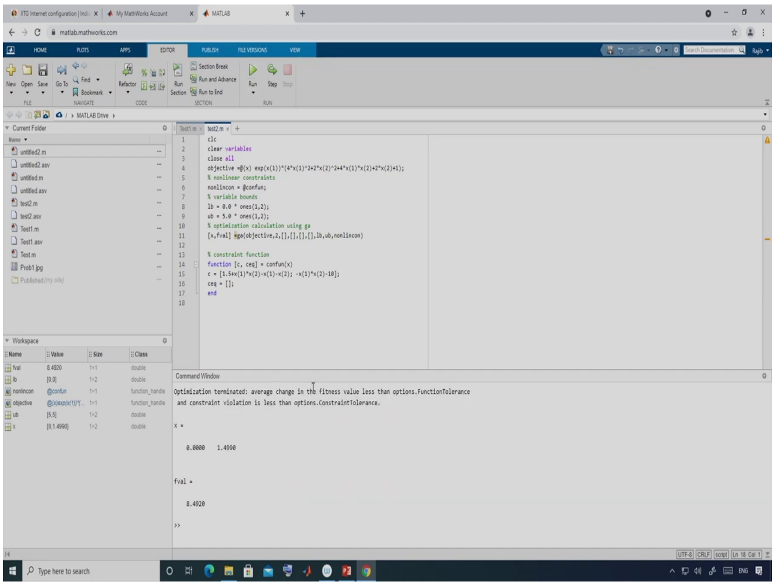


```matlab
clc
clear variables
close all
objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
c = [1.5+x(1)*x(2)-x(1)-x(2); -x(1)*x(2)-10];
ceq = [];
end
```

Name	Value	Size	Class
fval	8.4935	1×1	double
lb	[0,0]	1×2	double
nonlincon	@confun	1×1	function_handle
objective	@(x)exp(x(1))*(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[0,1.4991]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =

    0.0000    1.4991


fval =

    8.4935

>> test2
```

(Refer Slide Time: 51:26)



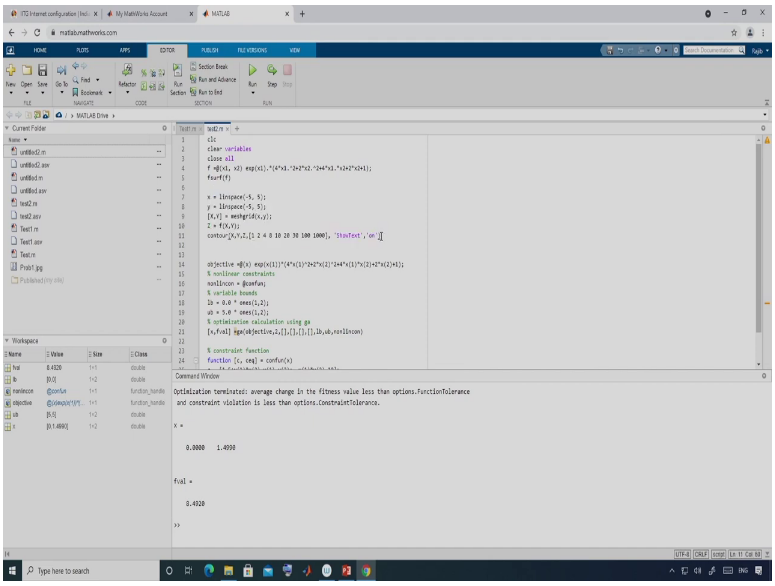
```
clc
clear variables
close all
objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
c = [1.5+x(1)*x(2)-x(1)-x(2); -x(1)*x(2)-10];
ceq = [];
end
```

Name	Value	Size	Class
fval	8.4920	1×1	double
lb	[0,0]	1×2	double
nonlincon	@confun	1×1	function_handle
objective	@(x)exp(x(1))*(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[0,1.4990]	1×2	double

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =

    0.0000    1.4990


fval =

    8.4920

>>
```

Now, if you want to see how you can plot this function so, that I have already shown you, but let me show that one. So, if I yeah so, this is the lines.

(Refer Slide Time: 51:45)



```
clc
clear variables
close all
f =@(x1, x2) exp(x1).*(4*x1.^2+2*x2.^2+4*x1.*x2+2*x2+1);
fsurf(f)

x = linspace(-5, 5);
y = linspace(-5, 5);
[X,Y] = meshgrid(x,y);
Z = f(X,Y);
contour(X,Y,Z,[1 2 4 8 10 20 30 100 1000], 'ShowText','on')


objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
```

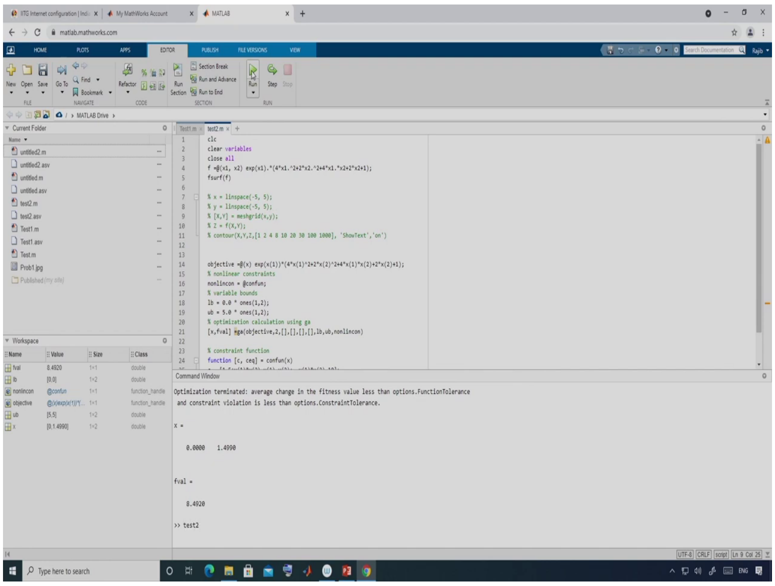
```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.

x =

   0.0000    1.4990


fval =

   8.4920

>>
```

Workspace

Name	Value	Size	Class
fval	8.4920	1×1	double
lb	[0,0]	1×2	double
nonlincon	@confun	1×1	function_handle
objective	@(x)exp(x(1))*(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[0,1.4990]	1×2	double

So, if you are using this line so, you can plot the function. So, you can plot the function using this lines ok so, this lines actually. So, here you have to define the function and then f surf you are using so, fsurf means you will get the surface. So, let me see the first surface, let me see and then I will show you the contour ok.

(Refer Slide Time: 52:13)



```
clc
clear variables
close all
f =@(x1, x2) exp(x1).*(4*x1.^2+2*x2.^2+4*x1.*x2+2*x2+1);
fsurf(f)

% x = linspace(-5, 5);
% y = linspace(-5, 5);
% [X,Y] = meshgrid(x,y);
% Z = f(X,Y);
% contour(X,Y,Z,[1 2 4 8 10 20 30 100 1000], 'ShowText','on')


objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
```

Workspace

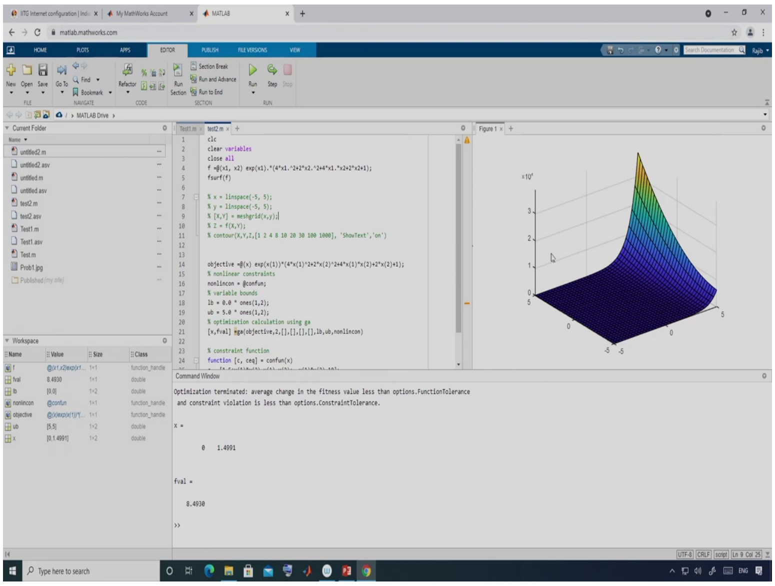
Name	Value	Size	Class
fval	8.4920	1×1	double
lb	[0,0]	1×2	double
nonlincon	@confun	1×1	function_handle
objective	@(x)exp(x(1))*(...	1×1	function_handle
ub	[5,5]	1×2	double
x	[0,1.4990]	1×2	double

Command Window

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =

    0.0000    1.4990



fval =

    8.4920

>> test2
```

So, let me show you the surface.

(Refer Slide Time: 52:22)

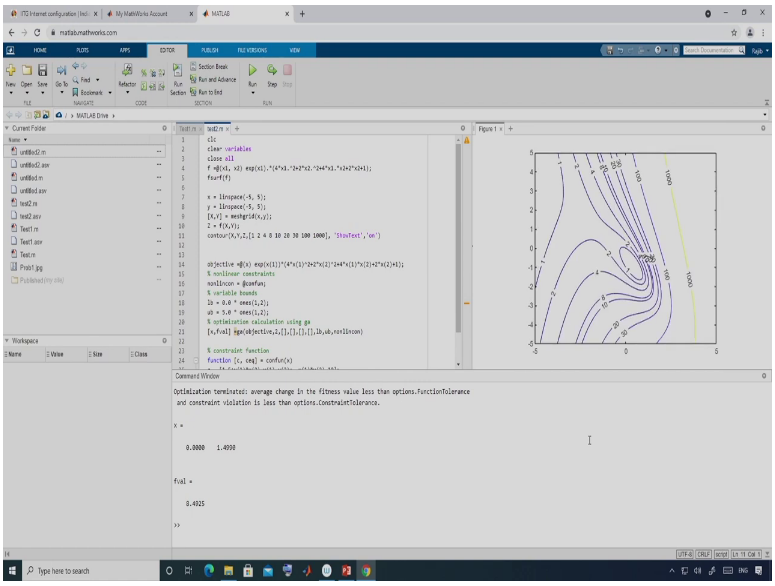


```
clc
clear variables
close all
f =@(x1, x2) exp(x1).*(4*x1.^2+2*x2.^2+4*x1.*x2+2*x2+1);
fsurf(f)

% x = linspace(-5, 5);
% y = linspace(-5, 5);
% [X,Y] = meshgrid(x,y);
% Z = f(X,Y);
% contour(X,Y,Z,[1 2 4 8 10 20 30 100 1000], 'ShowText','on')


objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.

x =

         0    1.4991


fval =

    8.4930

>>
```

Yeah, I got the solution and you can see the surface here yeah. So, this is the surface. So, you cannot see actually where is the optimal solution this is quite flat so, therefore, let me see the control lines ok.

(Refer Slide Time: 52:41)



```
clc
clear variables
close all
f =@(x1, x2) exp(x1).*(4*x1.^2+2*x2.^2+4*x1.*x2+2*x2+1);
fsurf(f)

x = linspace(-5, 5);
y = linspace(-5, 5);
[X,Y] = meshgrid(x,y);
Z = f(X,Y);
contour(X,Y,Z,[1 2 4 8 10 20 30 100 1000], 'ShowText','on')


objective =@(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
% nonlinear constraints
nonlincon = @confun;
% variable bounds
lb = 0.0 * ones(1,2);
ub = 5.0 * ones(1,2);
% optimization calculation using ga
[x,fval] =ga(objective,2,[],[],[],[],lb,ub,nonlincon)

% constraint function
function [c, ceq] = confun(x)
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
 and constraint violation is less than options.ConstraintTolerance.


x =

   0.0000    1.4990


fval =

   8.4925

>>
```

So, I can plot the contour and I can see where is the solution ok. So, let me run it again so, this time I will get the contours yeah. So, I am getting the solution is unconstrained solution is somewhere here, but constrained solution is 0 and 1.5. So, x is 0 and 1.5 is somewhere here ok. So, this is 1.5 somewhere here constrained solution of this particular problem.

So, you can also plot so, MATLAB is quite good in graphics. So, you can actually plot the surface, you can plot the contours ok and you can see actually where the optimum is there ok optimum solution is there ok.

So, thank you very much. So, in this class we have solved few problems using genetic algorithm and particle swarm optimization method. So, initially we have solved the problem using particle swarm optimization. So, this is particle swarm in MATLAB and then we have solved the same problem using genetic algorithm.

So, what you can do in the last class we have solved this problem using classical method and you can try to solve this problem using classical method as well as genetic algorithm or particle swarm optimization. Then you can feel the difference that some problem is quite easy to solve using classical method, but some problem, which is basically a non-linear non-convex problem.

So, if you have more than one optimal solution and one of them is the global optimal solution. So, in that case you may not get the global optimal solution using classical methods. So, in that case you can get the global optimal solution of the problem using genetic algorithm.

So, I have also mentioned one important point here that when you are using particle swarm optimization or genetic algorithm so, you may not get the exact optimal solution of the problem. So, because you are not using gradient information. So, you may not get the global optimal solution of the problem, exact global optimal solution of the problem, but you may get a solution, which is very near to the global optimal solution of the problem.

So, this is not the actual optimal solution, but it is a near global optimal solution. So, then what you can do? You can use the hybrid method. So, what you will do? So, whatever solution you have got using genetic algorithm or PSO. So, that solution you can take as an initial solution and you can apply your classical method ok. So, you will get the actual optimal solution of the problem.

So, this method is known as hybrid optimization method and this facility is also available in MATLAB. So, you can directly use the hybrid method to get the exact optimal solution of the problem. So, I want that you please solve this problem yourself and just see the difference when you are solving this problem using classical method as well as genetic algorithm.

Thank you.