

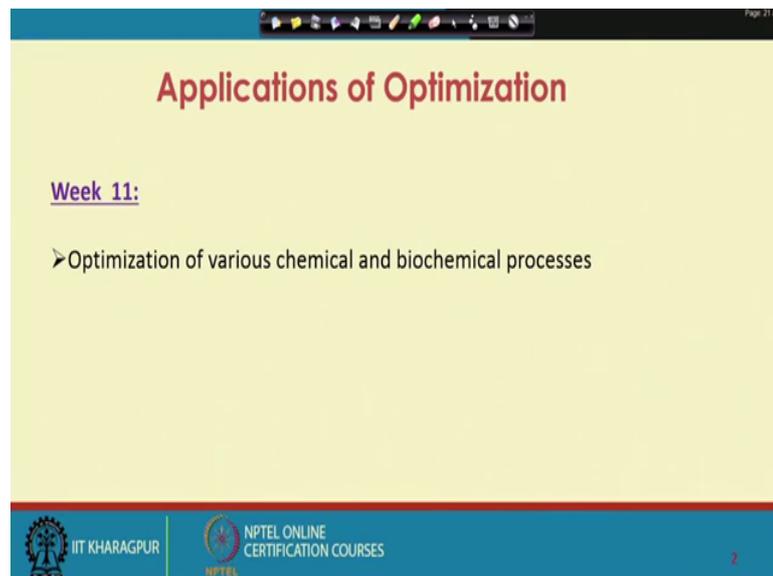
Optimization in Chemical Engineering
Prof. Debasis Sarkar
Department of Chemical Engineering
Indian Institute of Technology, Kharagpur

Lecture –51
Application of Optimization

Welcome to 11, this is lecture 51, the first lecture of week 11. So, as of now we have talked about theories of optimization, we have talked about techniques of optimization, various techniques, unconstrained optimization, constraint optimization, linear programming, non-linear programming with constraint constraints etcetera. So, now, in this week 11 and in the last week, week 12 we will talk about various Applications of Optimization as well as use of software tools for solving optimization problems.

So, today we will start our discussion with applications of optimizations. So, now, in the beginning of today's lecture we will first talk about two MATLAB functions known as fmincon and fsolve. And then we will take an example to demonstrate the use of fmincon and fsolve to solve an optimization problem which is taken from chemical engineering area.

(Refer Slide Time: 01:35)



So, let us start with applications of optimization; mostly we will take examples from chemical and biochemical processes.

(Refer Slide Time: 01:50)

The slide is titled "MATLAB Function fmincon" and describes the problem it solves. The text on the slide is as follows:

MATLAB function fmincon solves a General Nonlinear Programming Problem as follows:

Minimize $f(x)$

Such that: $c(x) \leq 0$

$c_{eq}(x) = 0$

$A \cdot x \leq b$

$A_{eq} \cdot x = b_{eq}$

$LB \leq x \leq UB$

Handwritten annotations in purple ink include:

- "May be objective function" pointing to $f(x)$.
- "Nonlinear inequality constraint" pointing to $c(x) \leq 0$.
- "equality" pointing to $c_{eq}(x) = 0$.
- "Linear inequality" pointing to $A \cdot x \leq b$.
- "equality" pointing to $A_{eq} \cdot x = b_{eq}$.
- "Bounds on x" pointing to $LB \leq x \leq UB$.
- "x is a vector of decision variables" written at the bottom right.

The slide also features logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, and a small video inset of a speaker in the bottom right corner.

So, first let us start with use of fmincon solver; fmincon is basically a solver for non-linear programming problem and it is a part of MATLAB's optimization toolboxes. Fmincon can solve a general non-linear programming problem with constraints and the problem that it can solve is shown here. So, we solve a minimization problem, so if you have a maximization problem we have to convert it to minimization problem by taking minus sign.

Note that all the constraints are of less or equal to type or equal to type. So, the problem is minimize $f(x)$ such that $c(x) \leq 0$, $c_{eq}(x) = 0$, $Ax \leq b$, $A_{eq}x = b_{eq}$ and x is bounded between lower bound and upper bound.

So, $c(x) \leq 0$ is basically non-linear inequality constraint, $c_{eq}(x) = 0$ is non-linear equality constraint. $Ax \leq b$ is linear inequality constraint, $A_{eq}x = b_{eq}$ is linear equality constraint.

And x is bounded between lower bound and upper bound. So, you can write bounds on decision variable x . So, x is a vector of decision variables, $f(x)$ may be non-linear objective function. So, this is the kind of problem that MATLAB's fmincon will solve. So, if you can cast your problem in this format, MATLAB can solve this optimization problem for you. So, we will see how it does.

(Refer Slide Time: 05:26)

The slide displays the syntax for the MATLAB function `fmincon`. The title is "MATLAB Function fmincon : Syntax". The syntax is listed as follows:

```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

Handwritten annotations in purple include:

- Arrows pointing from `A` and `b` in the first syntax line to $Ax \leq b$.
- Arrows pointing from `Aeq` and `beq` in the second syntax line to $Aeqx = beq$.

Below the syntax, the following output formats are shown:

```
[x,fval] = fmincon(___)
[x,fval,exitflag,output] = fmincon(___)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(___)

More information at:
https://in.mathworks.com/help/optim/ug/fmincon.html
```

The slide footer includes the IIT KHARAGPUR logo and the NPTEL ONLINE CERTIFICATION COURSES logo. A small video inset of a presenter is visible in the bottom right corner.

So, what is the Syntax? So, the syntax are shown here. So, you first have to define a function where you will write down the objective function. So, the function fun will written the value of the objective function at the decision variables to the fmincon solver, so you have to supply this function. So, you have to write an m file where the objective function will be evaluated; x 0 is the starting point, A and b comes from linear inequality constraint.

So, A the coefficient matrix, x is the vector, and b is the right and side vector. So, if you do not have any other constraint your syntax is this. If you have linear equality constraints, then you have to add these two; that means this is for. So, A underscore eq is basically the coefficient matrix for equality constraint, so this is equality. So, you do not want to supply bounds on x, you syntax up to this.

Next, if you want to bounds on decision variables use this. Now, your optimization problem mean bond non-linear constrains. In that case, similar to objective function file you also have to write another function file where you will mention the non-linear constraints. So this is for non-linear constrains. So, I had given name nonlcon, you can give whatever name you want. So, idea is that you have to supply a function where you will write down the non-linear constraints associated with the optimization problem.

Note that this function needs to be supplied separately, it has to written by you. The linear inequality constraint and the linear equality constraint can be put in a standardized

form. So, that is why it is possible to care of linear equality constraint and inequality constraint by supplying the coefficient matrix and the right hand side vectors, namely, A , b , A equality, b equality. But the general non-linear constraints can take any different form. So, you have to supply a function where you will define the non-linear constraints.

Then we have also seen the use of options before. So, you can also give options as input where you will tell the `fmincon` solver that you want to say display the results of each iterations, you want to change the tolerance, you want to choose a particular algorithm so that `fmincon` will use that particular algorithm only so on and so forth. Now, here we have seen the we have written the output of the `fmincon` is only x that is the decision variable.

There is the solution of the optimization problem, but you can also have the function value at the decision variable, you can have exit flag which will tell you about, how the whether the solution you obtained and the convergence was achieved for the solutions or not. So, it will tell you about the how the algorithm solve the problem that you have supplied. Similarly output, gives you various information about the algorithm such as number of functional evaluations, it tells you about what algorithm it used so on and so forth.

Lambda it gives you the Lagrange multipliers, the gradients at the decision variables solutions, also the hessian matrix of the function evaluated at the decision variable solution x . So, the complete syntax will be this for the left hand side and this for the right hand side. So, this part and this part, note that this last part. You can obtain more information about the use of `fmincon` from the MATLAB website.

(Refer Slide Time: 12:33)

The slide displays MATLAB code for minimizing a function subject to a constraint. The objective function is $f(x,y) = x + y$ and the constraint is $x^2 + y^2 = 1$. The code defines two functions: `func` for the objective and `nonlincon` for the constraint. Handwritten annotations in pink include underlines for the objective and constraint, and labels A, b and Aeq, beq pointing to the constraint definition.

```
Minimize  $f(x,y) = x + y$ 
Subject to:  $x^2 + y^2 = 1$ 

function [c,ceq] = nonlincon(X)
% function for the nonlinear constraints.
% We have to define two outputs for fmincon: the
% nonlinear inequality constraints, and the equality
% constraints.
% C(X) <= 0
% Ceq(X) = 0
x = X(1);
y = X(2);
c = [];
ceq = x^2 + y^2 - 1;
end

function f = func(X)
x = X(1);
y = X(2);
f = x + y;
end
```

Now, let us take a quick example on the use of `fmincon`. You want to minimize the function x plus y subject to x square plus y square equal to 1. So, you have a simple linear objective function with one non-linear equality constraint so that is the problem. So, first thing will be to write a function file for the objective function. Objective function is minimization of x plus y ; so this is the function file that I write.

So, have to write the function file in this format; function `f` equal to I have given the function name as `func`, you can give any name you want, say my function etcetera, x is the decision variables. So, they are the trail vectors for the solutions. So, this is the vector which contains this x and this y . So, x equal to capital X of 1, y equal to capital X of 2; so `f`, the objective function is nothing but x plus y .

So, while `fmincon` will solve these optimization problem, it will send the trial vectors x which contains small x and small y to these function file `func`; `func`. And the value of the objective function `f` will be evaluated and returned to the `fmincon` solver ok. So, I have only linear equality constraints, sorry non-linear equality constraints. I do not have linear inequality constraints I do not have linear equality constraint. I only have non-linear equality constraint, I do not have non-linear inequality constraint I have only non-linear equality constraints. So, you have so those A, b, Aeq, beq are not relevant for me; so they are not present. But I have a non-linear equality constraint. So, I have to define a function for that.

(Refer Slide Time: 15:52)

MATLAB Function fmincon : Example

Minimize $f(x,y) = x + y$
Subject to: $x^2 + y^2 = 1$

```
function [c,ceq] = nonlincon(X)
% function for the nonlinear constraints.
% We have to define two outputs for fmincon: the
% nonlinear inequality constraints, and the equality
% constraints.
C(X) <= 0
Ceq(X) = 0
end
```

function f = func(X)
x = X(1);
y = X(2);
f = x + y;
end

Handwritten annotations:
- A box around the function definition: $Ceq = 0$
- A box around the equality constraint: $Ceq = 0$
- A box around the inequality constraint: $C(X) <= 0$
- A box around the equality constraint: $x^2 + y^2 - 1 = 0$
- A box around the equality constraint: $Ceq(X) = 0$
- A box around the equality constraint: $Ceq(X) = 0$
- A box around the equality constraint: $Ceq(X) = 0$

So, I am defining a function nonlincon for this non-linear equality constraint. Note that you have to supply a single function for both non-linear equality and non-linear inequality constraint. So, look at the format of the function, how this is how you define the function. Output of this function is c and ceq; so c for non-linear inequality constraint and ceq is non-linear equality constraint.

So, I have only non-linear equality constrain. So, c is null matrix for me, c is not required for me. So, I have given this as the name of the function that defines non-linear constraints. We have to define two outputs for fmincon, the non-linear inequality constraint which is c and the equality which is ceq. So these are those two constraints. For me this I do not have in the problem this I have.

So, again this x contains small x and small y for the problem. So those things are define and the non-linear equality constraint $x^2 + y^2 = 1$ has to be written as $x^2 + y^2 - 1 = 0$. Because if you remember the first slides it was written as $ceq = 0$. So, you have express the non-linear equality constraint as $ceq = 0$. So, ceq will be $x^2 + y^2 - 1$.

(Refer Slide Time: 18:29)

MATLAB Function fmincon : Example

```
A = [];  
B = []; % the linear inequality constraints: A*X <= B  
Aeq = [];  
Beq = []; % the linear equality constraints: Aeq*X = B  
LB = [];  
UB = []; % LB <= X <= UB  
X0 = [1, 1]; % initial guess
```

$[X, FVAL, EXITFLAG, OUTPUT, LAMBDA] = \text{fmincon}(@\text{func}, X0, A, B, Aeq, Beq, LB, UB, @\text{nonlincon})$

Handwritten annotations: A circle around the code with arrows pointing to 'A' and 'B' in the constraint definitions, and another circle containing the handwritten text 'A → A' and 'B → B'.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, once we have done this we can now call fmincon solver to solve this problem. So, I did not have A, so A is null, similarly B, this capital A for A, capital B for small b according to our previous notation.

(Refer Slide Time: 19:08)

MATLAB Function fmincon : Example

```
A = [];  
B = []; % the linear inequality constraints: A*X <= B  
Aeq = [];  
Beq = []; % the linear equality constraints: Aeq*X = B  
LB = [];  
UB = []; % LB <= X <= UB  
X0 = [1, 1]; % initial guess
```

$[X, FVAL, EXITFLAG, OUTPUT, LAMBDA] = \text{fmincon}(@\text{func}, X0, A, B, Aeq, Beq, LB, UB, @\text{nonlincon})$

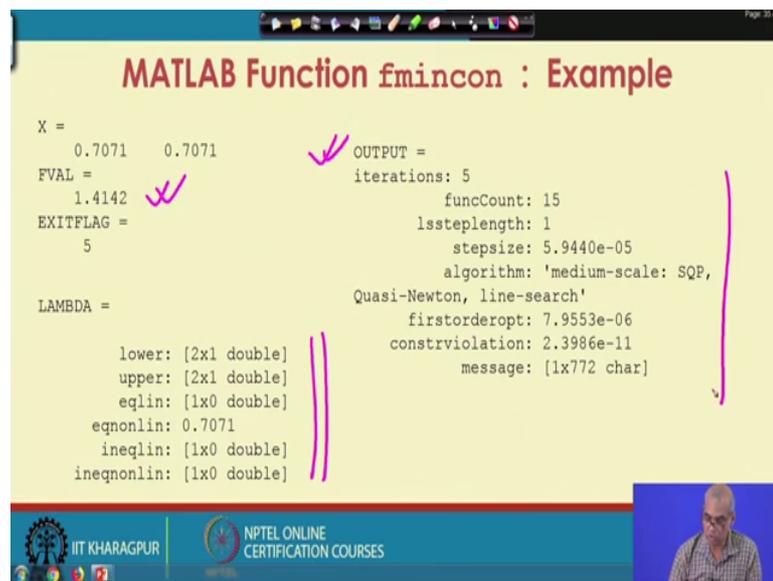
Handwritten annotations: Checkmarks next to the constraint definitions, and a handwritten equation: $X_0 = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, I do not have linear equality constraints, so they are also null. Lower bound upper bound I am not specifying, as initial guess is small x equal to 1, small y equal to 1. So, x_0 is basically small x small y which is 1, 1; this is supplier initial guess. Now, you call the fmincon solver.

Note that this sequence is very important; you must supply all the arguments strictly in this sequence. So, you do not have A, B even then you have to define like this. So, now, you write this statement on the MATLAB command mode and press the enter or you can create a script file, m file, there you can write the statement and then run that m file.

(Refer Slide Time: 20:33)



```
MATLAB Function fmincon : Example

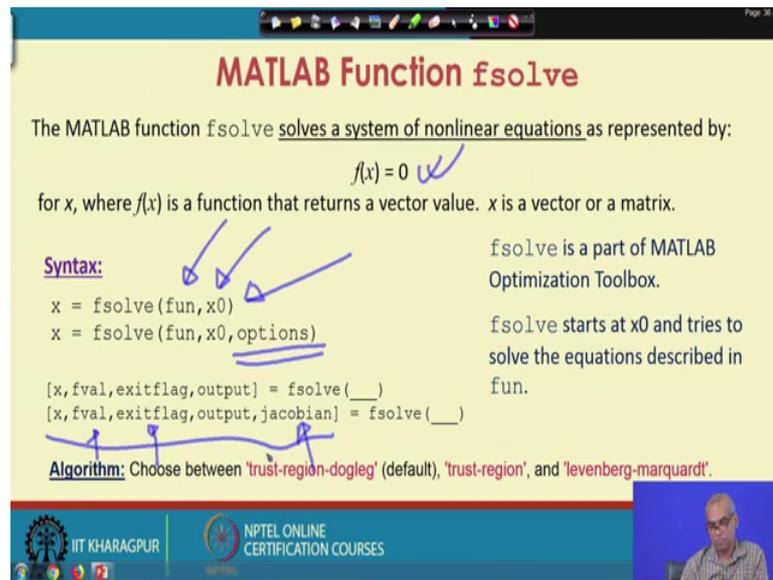
X =
    0.7071    0.7071
FVAL =
    1.4142
EXITFLAG =
     5
LAMBDA =
    lower: [2x1 double]
    upper: [2x1 double]
    eqlin: [1x0 double]
    eqnonlin: 0.7071
    ineqlin: [1x0 double]
    ineqnonlin: [1x0 double]

OUTPUT =
    iterations: 5
    funcCount: 15
    lssteplength: 1
    stepsize: 5.9440e-05
    algorithm: 'medium-scale: SQP,
    Quasi-Newton, line-search'
    firstorderopt: 7.9553e-06
    constrviolation: 2.3986e-11
    message: [1x772 char]
```

The slide displays the output of the MATLAB function `fmincon`. The output is divided into two main sections: the primary solution variables and the optimization process details. The primary variables include the optimal point `X` (0.7071, 0.7071), the function value `FVAL` (1.4142), the exit flag `EXITFLAG` (5), and the Lagrange multiplier structure `LAMBDA`. The `LAMBDA` structure contains fields for lower and upper bounds, equality and inequality linear constraints, and equality and inequality nonlinear constraints. The optimization details section, labeled `OUTPUT`, provides information such as the number of iterations (5), function evaluations (15), step size (5.9440e-05), the algorithm used ('medium-scale: SQP, Quasi-Newton, line-search'), first-order optimality (7.9553e-06), constraint violation (2.3986e-11), and a message field.

So, if you do this you will get this as output. So, the MATLAB will return the solution of the optimization problem as X equal to 0.7, X equal to 0.7071 and Y equal to 0.7071; so X equal to Y. This is the function value, these are the Lagrange multiplier. This structure will contain the Lagrange multiplier information. This is the output tells you about step size, algorithm, function count, etcetera.

(Refer Slide Time: 21:37)



The slide is titled "MATLAB Function fsolve" in red. It explains that the MATLAB function `fsolve` solves a system of nonlinear equations $f(x) = 0$ for x , where $f(x)$ is a function that returns a vector value. x is a vector or a matrix. The slide lists the syntax: `x = fsolve(fun, x0)` and `x = fsolve(fun, x0, options)`. It also shows the full output: `[x, fval, exitflag, output] = fsolve(__)` and `[x, fval, exitflag, output, jacobian] = fsolve(__)`. The algorithm options are listed as 'trust-region-dogleg' (default), 'trust-region', and 'levenberg-marquardt'. The slide is part of an NPTEL online certification course from IIT Kharagpur. There is a small video inset of a man in the bottom right corner.

MATLAB Function fsolve

The MATLAB function `fsolve` solves a system of nonlinear equations as represented by:

$$f(x) = 0$$

for x , where $f(x)$ is a function that returns a vector value. x is a vector or a matrix.

Syntax:

```
x = fsolve(fun, x0)
x = fsolve(fun, x0, options)
```

```
[x, fval, exitflag, output] = fsolve(__)
[x, fval, exitflag, output, jacobian] = fsolve(__)
```

Algorithm: Choose between 'trust-region-dogleg' (default), 'trust-region', and 'levenberg-marquardt'.

`fsolve` is a part of MATLAB Optimization Toolbox.

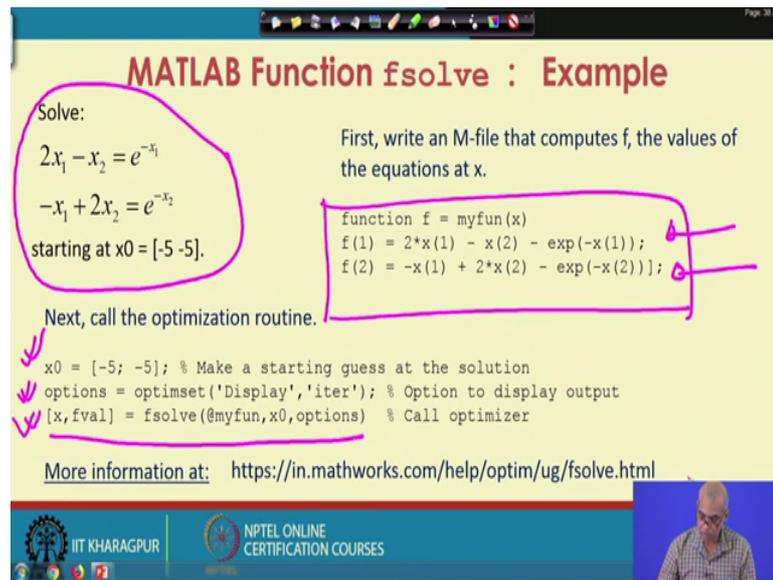
`fsolve` starts at x_0 and tries to solve the equations described in `fun`.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, next let us take another example; now this time the use of `fsolve`. The MATLAB function `fsolve` solves a system of non-linear equations as represented by $f(x) = 0$. So, basically `fsolve` solves a system of non-linear equations. So, it will find out x vector for which $f(x) = 0$. So, `fsolve` is a part of MATLAB optimization toolbox, the syntax are very simple, $x = \text{fsolve}$.

Then you have to supply the function where you will write the equation that you are going to solve and many have to supply the starting initial guess vector. You can also use options to supply specific instruction to the solver `fsolve`. You can also have these as outputs apart from the solution vector you can have the function value at the solution vector as `fval`, `exitflag` will tell you about the termination criteria whether it was terminated successfully, `output` will give you informations about the optimization algorithm, function number of function evaluation etcetera. And it will also give you the value of the Jacobin, so all things are possible. `fsolve` uses Trust-region-dogleg as default algorithm, it can also use trust region and Levenberg-Marquardt algorithm. If you want use Levenberg-Marquardt algorithm or Trust-region algorithm, you have to make use of options to tell this information to the solver.

(Refer Slide Time: 24:07)



The slide is titled "MATLAB Function fsolve : Example". It is divided into several sections:

- Solve:** A purple circle highlights the equations $2x_1 - x_2 = e^{-x_1}$ and $-x_1 + 2x_2 = e^{-x_2}$, along with the starting point $x_0 = [-5 -5]$.
- Text:** "First, write an M-file that computes f, the values of the equations at x."
- Code Block:** A purple box contains the MATLAB function definition:

```
function f = myfun(x)
f(1) = 2*x(1) - x(2) - exp(-x(1));
f(2) = -x(1) + 2*x(2) - exp(-x(2));
```
- Text:** "Next, call the optimization routine."
- Code Block:** A purple box contains the MATLAB code to solve the system:

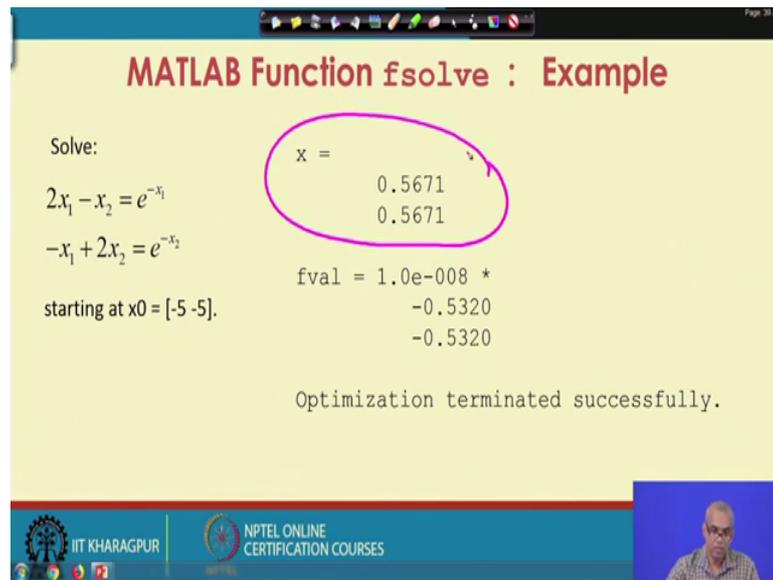
```
x0 = [-5; -5]; % Make a starting guess at the solution
options = optimset('Display','iter'); % Option to display output
[x,fval] = fsolve(@myfun,x0,options) % Call optimizer
```
- Text:** "More information at: <https://in.mathworks.com/help/optim/ug/fsolve.html>"

The slide footer includes the IIT KHARAGPUR logo and the NPTEL ONLINE CERTIFICATION COURSES logo. A small video inset of a speaker is visible in the bottom right corner.

Now, let us take a quick example; let say I have these two equations which needs to be solved simultaneously two non-linear equation starting from minus 5 minus 5. So, first thing we write an M file that will compute the value of the equations at x, the trial solution. So, this is how I write that I define the function, look at the defined function definition function f equal to myfun of x. So, x contains x 1 and x 2; so you simply write the first equation as well as the second equation.

Now, you call the optimization routine fsolve, first you supply the initial guess x 0 equal to minus 5 minus 5. You can supply options through optimset display iter; that means, every iterations will be displayed or you may not want it. And then you call fsolve as x fval equal to fsolve at myfun x 0 options. So, this is the syntax that has to be used and this is called optimization routine, more information can be found at MATLAB website.

(Refer Slide Time: 25:54)



MATLAB Function fsolve : Example

Solve:

$$2x_1 - x_2 = e^{-x_1}$$
$$-x_1 + 2x_2 = e^{-x_2}$$

starting at $x_0 = [-5 \ -5]$.

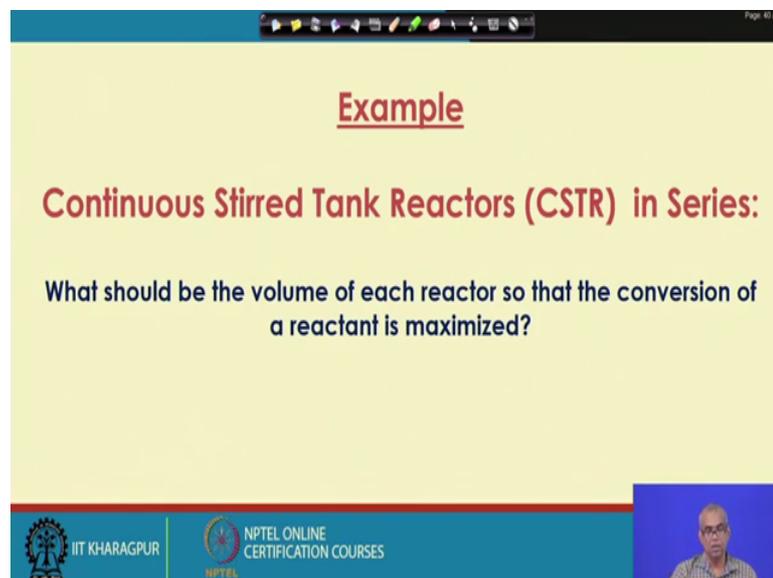
```
x =  
    0.5671  
    0.5671  
  
fval = 1.0e-008 *  
    -0.5320  
    -0.5320
```

Optimization terminated successfully.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, if you do this you will get the solution as 0.5671, 0.5671 for x_1 and x_2 . So, this is how you can use `fmincon` and `fsolve`.

(Refer Slide Time: 26:19)



Example

Continuous Stirred Tank Reactors (CSTR) in Series:

What should be the volume of each reactor so that the conversion of a reactant is maximized?

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, we will take an example from Continuous Stirred Tank Reactor in series. We are asking what should be the volume of each reactor so that conversion of reactant is maximized.

(Refer Slide Time: 26:34)

Reactors in Series

Three Continuous Stirred Tank Reactors (CSTRs) are connected in series. In these reactors, a reaction that consumes component A takes place: $A \rightarrow \text{Product}$

The volumetric flow rate to each CSTR is same (F_0).

Reaction constant $k = 1.0 \text{ m}^{-1.5}/(\text{hr kmol}^{0.5})$
Volumetric flow rate $F_0 = 2.0 \text{ m}^3/\text{hr}$
Feed concentration of reactant A, $C_{A0} = 1 \text{ kmol}/\text{m}^3$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, you have reactors in series, so you have three CSTR in series. In these reactors, a reaction that consumes component A takes place. So, a reaction A to Products takes place in each of the CSTR's. The volumetric flow rate to each CSTR is same F_0 , but note that the concentration are all different. Inlet concentration to reactor 1 is C_{A0} and then here the reaction takes place A to Products. So, the concentration of the stream that goes from CSTR 1 to CSTR 2 is C_{A1} , here also A to Product reaction takes place. So, the concentration of the stream that goes from the CSTR 2 to CSTR 3 is C_{A2} and here also A to Product takes place.

So, the concentration that comes out of the stream coming from the CSTR 3 is C_{A3} . So, reaction constraint k are given, volumetric flow rate is given as 2 meter cube per hour that is same for each reactor. Feed concentration of reactant A is 1 kilo mole per meter cube. So, what is known is the reaction constraint, the flow rate of the streams to each reactor is known, inlet concentration to CSTR 1 is known. What are unknowns is the volume of each CSTR V_1 , V_2 , V_3 and also the concentration exit concentration of CSTR 1, CSTR 2 and CSTR 3; that means, C_{A1} , C_{A2} and C_{A3} are all unknown.

(Refer Slide Time: 28:56)

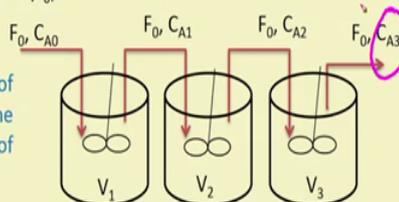
Reactors in Series

Three Continuous Stirred Tank Reactors (CSTRs) are connected in series. In these reactors, a reaction that consumes component A takes place: $A \rightarrow \text{Product}$

The volumetric flow rate to each CSTR is same (F_0).

The objective is to determine the volumes of Reactors 1, 2, and 3 that minimize the concentration of reactant (A) at the outlet of the third reactor (C_{A3}).

There is a constraint that the total volume (i.e., the sum of the volumes of all three reactors) must be less than or equal to 7 m^3 .



Reaction constant $k = 1.0 \text{ m}^{-1.5}/(\text{hr kmol}^{0.5})$

Volumetric flow rate $F_0 = 2.0 \text{ m}^3/\text{hr}$

Feed concentration of reactant A, $C_{A0} = 1 \text{ kmol/m}^3$

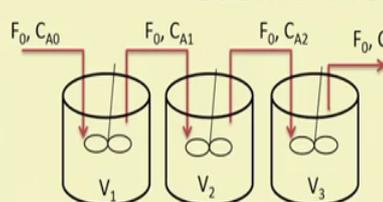
Page 41/41

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

The objective is to determine the volumes of the reactors; 1, 2 and 3 that minimize the concentration of reactant A at the outlet of the third reactor. So, we want to minimize the concentration A here; that means, you have one the maximum amount of A to be converted to product B. There is a constraint that the volume that is the sum of the volumes of all three reactors must be less or equal to 7 meter cube. So, V_1 plus V_2 plus V_3 must be less or equal to 7 meter cube. So, the optimization problem is that find out the volume V_1 , V_2 , V_3 such that C_{A3} is minimized, at the same time V_1 plus V_2 plus V_3 less or equal to 7 meter cube is satisfied.

(Refer Slide Time: 29:53)

Reactors in Series: Modeling



$F_0 C_{A0} - F_0 C_{A1} - \gamma_{A1} V_1$

$\gamma_{A1} = k C_{A1}^{1.5}$

$A \rightarrow \text{Product}$

Mass Balances:
 $F_0 (C_{A(i-1)} - C_{Ai}) = r_{Ai} V_i, \quad i \in \{1, 2, 3\}$

Reaction Rates:
 $r_{Ai} = k C_{Ai}^{1.5}, \quad i \in \{1, 2, 3\}$

Reaction constant $k = 1.0 \text{ m}^{-1.5}/(\text{hr kmol}^{0.5})$

Volumetric flow rate $F_0 = 2.0 \text{ m}^3/\text{hr}$

Feed concentration of reactant A, $C_{A0} = 1 \text{ kmol/m}^3$

Page 41/41

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, first thing is to do the modeling of this CSTR's in series. So, you write down the mass balances equations. So, mass balance equation, this is steady state operation. So, amount of A that comes in to a reactor minus the amount of A that goes out of this reactor minus the amount of the A that is consumed will be equal to 0. So, we can write this as mass balance equation for each CSTR. $F_0 C_{A0}$ say for CSTR 1, you can write as $F_0 C_{A0}$ is the amount that goes in to CSTR 1 with the inlet reactor string.

Then what goes out from CSTR 1 is $F_0 C_{A1}$. And what gets converted to product is $r_{A1} V_1$. Similarly, you can write this for CSTR 2, CSTR 3 as well and in general they can be written as shown. So, reaction rate constraints, reaction rates are given as r_{Ai} equal to $k C_{Ai}$ to the power 1.5. So, k is given as 1, F_0 is given as 2, C_{A0} given as 1 in appropriate units as shown.

(Refer Slide Time: 32:19)

Reactors in Series: Modeling

F_0, C_{A0} → F_0, C_{A1} → F_0, C_{A2} → F_0, C_{A3}

V_1 V_2 V_3

A → Product

Mass Balances:

$$F_0(C_{A(i-1)} - C_{Ai}) = r_{Ai}V_i, \quad i \in \{1, 2, 3\}$$

Reaction Rates:

$$r_{Ai} = kC_{Ai}^{1.5}, \quad i \in \{1, 2, 3\}$$

Mass Balances:
 $2(C_{A0} - C_{A1}) = r_{A1}V_1$
 $2(C_{A1} - C_{A2}) = r_{A2}V_2$
 $2(C_{A2} - C_{A3}) = r_{A3}V_3$

Reaction Rates:
 $r_{A1} = C_{A1}^{1.5}$
 $r_{A2} = C_{A2}^{1.5}$
 $r_{A3} = C_{A3}^{1.5}$

Reaction constant $k = 1.0 \text{ m}^{1.5}/(\text{hr kmol}^{0.5})$
 Volumetric flow rate $F_0 = 2.0 \text{ m}^3/\text{hr}$
 Feed concentration of reactant A, $C_{A0} = 1 \text{ kmol}/\text{m}^3$

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES

So, you can write down this three mass balance equation for CSTR 1, 2 and 3. And these are the reaction rates. So this completes my modeling.

(Refer Slide Time: 32:30)

Reactors in Series: Problem Formulation

A → Product

Mass Balance
 $F_0(C_{A(i-1)} - C_{Ai}) = r_{Ai}V_i, \quad i \in \{1, 2, 3\}$

Reaction Rate
 $r_{Ai} = kC_{Ai}^{1.5}, \quad i \in \{1, 2, 3\}$

Find CSTR volumes to minimize C_{A3} .

minimize C_{A3}

Subject to

$$\begin{cases} 2(C_{A0} - C_{A1}) - r_{A1}V_1 = 0 \\ 2(C_{A1} - C_{A2}) - r_{A2}V_2 = 0 \\ 2(C_{A2} - C_{A3}) - r_{A3}V_3 = 0 \\ r_{A1} - C_{A1}^{1.5} = 0 \\ r_{A2} - C_{A2}^{1.5} = 0 \\ r_{A3} - C_{A3}^{1.5} = 0 \end{cases}$$

Model equation

$$V_1 + V_2 + V_3 \leq 7$$

Linear inequality constraint

$V = [V_1, V_2, V_3]^T$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, now the problem formulation then will be minimize C_{A3} , find out V vector which involves V_1 , V_2 and V_3 . Subject to these model equations along with the rate equations and these linear inequality constraint. So, I have a minimization problem, where I am minimizing C_{A3} subject to the model equations which are all equality types and one linear inequality constraint $V_1 + V_2 + V_3 \leq 7$. So, you can solve it using `fmincon`.

(Refer Slide Time: 33:28)

Reactors in Series: MATLAB Program

Write the model equation of the system in a function file

```
function F = CSTR(CA)
global V
F0 = 2; % Flow rate
k = 1; % Reaction constant
CA0 = 1; % Initial concentration of reactant A
V1 = V(1); % Volume of reactor 1
V2 = V(2); % Volume of reactor 2
V3 = V(3); % Volume of reactor 3

% Reaction rates
R1 = k*CA(1)^1.5;
R2 = k*CA(2)^1.5;
R3 = k*CA(3)^1.5;

% Model equations
F(1) = F0*(CA0 - CA(1)) - V1*R1;
F(2) = F0*(CA(1) - CA(2)) - V2*R2;
F(3) = F0*(CA(2) - CA(3)) - V3*R3;
end
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, first we write the model equation of the system in a function file. So, this is the function file where I am writing down the model equations. Note that I have defined this decision variable V as global variables. So, that it is visible to other functions that I am going to write for the same problem. So, if the variable is declared as global this will be accessible or visible to all other associated programs for this optimization problem.

So, you define the parameters and the variables. So, these are the equations for the three reaction rates and these are the three equations for the three mass balances in CSTR 1, 2 and 3.

(Refer Slide Time: 34:39)

Reactors in Series: MATLAB Program

Objective: Find CSTR volumes to minimize C_{A3} . We have to solve CSTR models to find C_{A3} .

Solve the model equations and define the objective function in another function file

```
function f = objfun(Vol)
global V
V = Vol; % Volume of all three reactors
CA_guess = [1 1 1]; % Initial guess value of concentration
CA = fsolve(@CSTR,CA_guess); % Solve model equations
f = CA(3); % Objective function
end
```

The slide features logos for IIT Kharagpur and NPTEL Online Certification Courses. A small video inset in the bottom right corner shows a man speaking.

So, now our objective is to find the CSTR volume that will minimize the concentration of A in the exit stream of CSTR 3 that is CA_3 . So, now, we have solve this CSTR model equations to find the concentrations C_{A1} , C_{A2} and C_{A3} . So, my unknowns are V_1 , V_2 , V_3 , C_{A1} , C_{A2} , C_{A3} . So, the `fmincon` solver will try to find out the minimum of minimum value C_{A3} by trying various values of V_1 , V_2 and V_3 while satisfying all the constraints and satisfying the model equations as well.

So, we need to solve the model equations for given V_1 , V_2 , V_3 that the `fmincon` solver will supply. So, we need to solve the model equations and define the objective function in another function file; so this is that function file. So, function `f` equal to `objfun` volume; this volume contains basically V_1 , V_2 V_3 . So, then you have define the variable V as global you see, so it is visible in that CSTR function as well as this `objfun` function.

And now, if you note the model equations these model equations are nothing but equality constraints. These models equations are nothing but a set of non-linear equations. So, system of non-linear equations; so system of non-linear can be solved by fsolve. So, that is what in these function value in this function file where I am going determine the objective function volume I am calling fsolve with an argument for that function to solve for C A1, C A2 and C A3. And I am getting C A3 as the objective function volume. So, the initial guess for fsolve is 1 1 1; that means, V1 equal to 1, V2 equal to 1 V3 equal to 1.

So, this function file which basically supplies the value of the objective function, it solves the model equations which are nothing but a system of three non-linear equations by using fsolve. And then you take, so, fsolve will give you will return this CA which has C A1, C A2 and C A3 all the concentrations. So, I take C A3 as C A3 is the exit concentration of CSTR 3. So, concentration of A in the exit stream of CSTR 3 that is we want to minimize, so that is my objective function volume. So, once we do this we have define the all the necessary functions.

(Refer Slide Time: 38:26)

Reactors in Series

Now write this code in MATLAB command window (or write a script)

```

V0 = [2 3 2]; % Initial guess value of volume [V1 V2 V3]
A = [1 1 1]; % Linear inequality constraint
b = 7;
Aeq = [];
beq = [];
lb = [];
ub = [];
Options = optimset('Algorithm','interior-point','MaxIter',1000,'MaxFunEvals',10000,'Display','iter');
[V, fval] = fmincon(@objfun,V0,A,b,Aeq,beq,lb,ub,[],options);
% display results
fprintf('V1 = %f \n', V(1))
fprintf('V2 = %f \n', V(2))
fprintf('V3 = %f \n', V(3))
fprintf('The optimal value of CA3 = %f \n', fval)

```

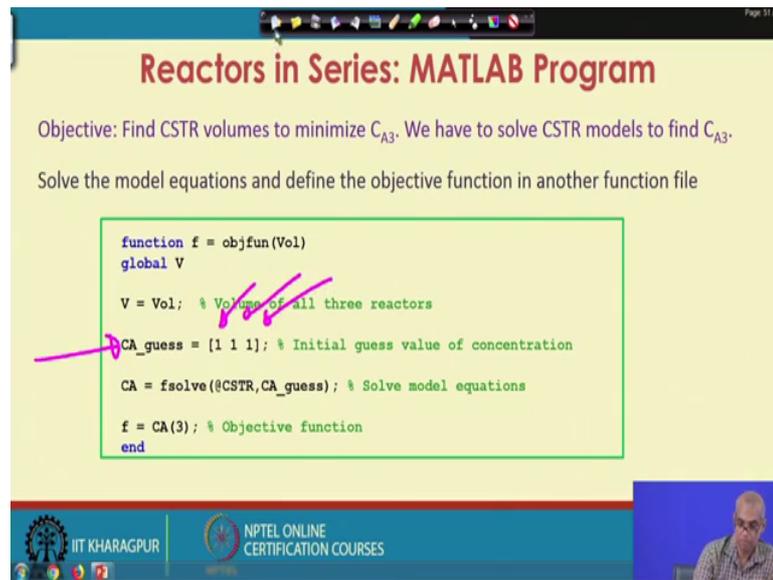
Solve the optimization problem using MATLAB function fmincon

$Ax \leq b$
 $V_1 + V_2 + V_3 \leq 7$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, I can call fmincon using the usual syntax. So, this is my say initial guess for volume V1 V2 V3.

(Refer Slide Time: 38:43)



The slide is titled "Reactors in Series: MATLAB Program". It contains the following text and code:

Objective: Find CSTR volumes to minimize C_{A3} . We have to solve CSTR models to find C_{A3} .

Solve the model equations and define the objective function in another function file

```
function f = objfun(Vol)
global V
V = Vol; % Volume of all three reactors
CA_guess = [1 1 1]; % Initial guess value of concentration
CA = fsolve(@CSTR,CA_guess); % Solve model equations
f = CA(3); % Objective function
end
```

The slide also features the IIT Kharagpur and NPTEL Online Certification Courses logos at the bottom, and a small video inset of a speaker in the bottom right corner.

So, this is this initial guess was for fsolve; this is C_{A1} , C_{A2} and C_{A3} . Now, this initial guess is for fmincon these are the initial guess for volumes, V_1 , V_2 , V_3 . So, this is A and B thus $Ax \leq b$ which says that V_1 plus V_2 plus V_3 has to be less or equal to 7. We do not have linear equality constraints; we do not have bounds. I am sending an options, so this is how I now call fmincon.

Note that I do not have any non-linear equality constraints. If at the model equations are non-linear equality constraints, but I am taking care of by solving using fsolve. So, then I just display the results.

(Refer Slide Time: 40:04)

Page 53/53

Reactors in Series

Result

$V_1 = 2.077897$
 $V_2 = 2.331545$
 $V_3 = 2.590543$
The optimal value of $C_{A3} = 0.210472$

$V_1 = 2.0779 \text{ m}^3 \quad V_2 = 2.3315 \text{ m}^3 \quad V_3 = 2.5905 \text{ m}^3$
 $C_{A3} = 0.2105 \text{ kmol/m}^3$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

If you do this we will get the solution as V_1 equal to 2.07, V_2 equal to 2.33 and V_3 equal to 2.59. And optimal point is C_{A3} is 0.210472. So, this is the optimal solutions. So, with this you can make use of `fmincon` and `fsolve` to solve an optimization problem related to chemical reaction engineering; reactors in series. So, with this we stop lecture 51 here.