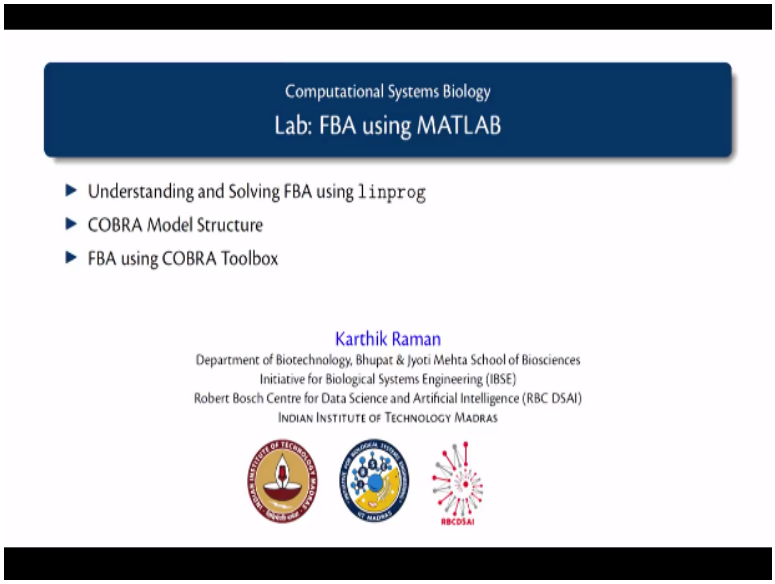
Computational Systems Biology
Karthik Raman
Department of Biotechnology
Indian Institute of Technology – Madras

Lecture - 62
Lab: FBA using MATLAB

(Refer Slide Time: 00:11)

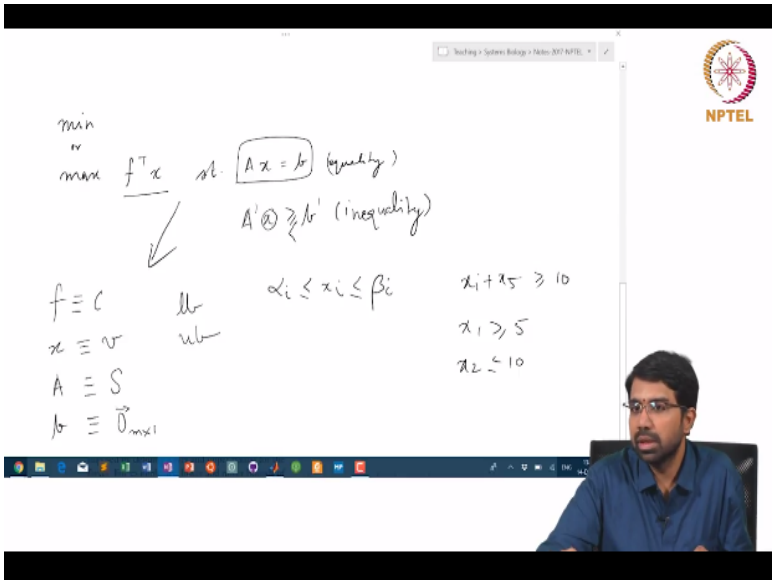


So welcome to this lab wherein we will study how we perform FBA using MATLAB. We will first look at linprog which is a standard MATLAB linear programming optimizer and see how we can assemble the arguments for linprog from the various components of the FBA that you already know which is basically maximize c transpose v such that sv=0 and some bounds and so on.

And we will also look at what is the model structure for any COBRA model. So COBRA stands for constraint-based reconstruction analysis. It is a very popular toolbox known as open COBRA toolbox and we will also see how we can perform FBA using the COBRA toolbox. So today we will have a lab session, we will look at how to simulate simple constraint-based models without using any toolboxes.

We will then move over to using the most popular COBRA toolbox for making FBA simulations and so on. So let us see how do you simulate such models just using MATLAB because it is just a simple linear program.

(Refer Slide Time: 01:10)



So the first thing to look at is how do we use linear programming or the function linprog in MATLAB. So to V cap what is the linear program minimize or maximize f transpose x such that Ax=b, this is what we would call an equality constraint. We can also have inequality constraints, f is the objective function so here f is the objective function, A is your matrix and this is basically your linear system of equalities right.

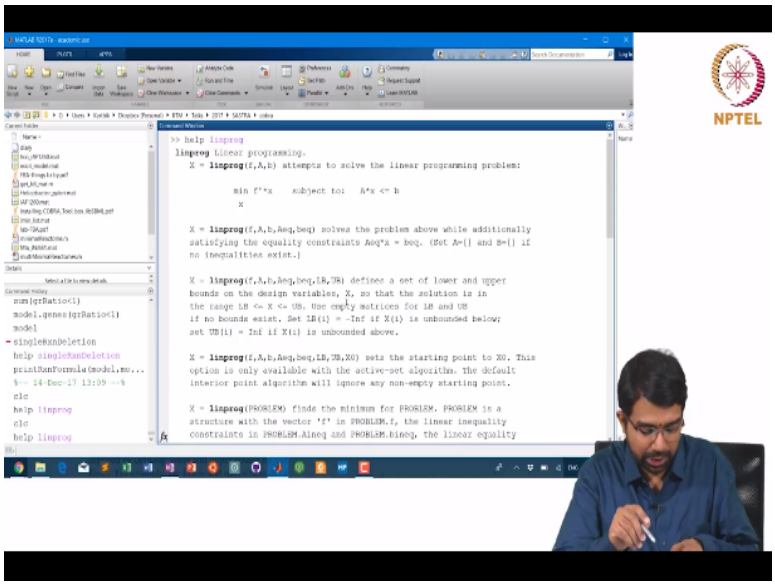
You can also have actually have some inequality constraints whereas in FBA we only have a single objective function and equality constraints. Do we have inequality constraints? **"Professor - student conversation starts."** You know the lower bound and upper bounds so those can actually be specified as inequality constraints but these are very simple constraints. These are just upper and lower bounds.

So they usually specified separately, so most solvers will have a specification for upper and lower bounds. We have variable x that you want to find and you can basically give alpha i<=xi this is possible. I did not catch that, it is a same thing know, it is the same thing, it just says this is the constraint is, this is the linear constraint that is given to the solver in terms of upper and lower bounds.

See if you had a constraint x1+x5>=10 then you need to specify differently. If all your saying is x1>=5, x2<=10 you can directly specify them as upper and lower bounds. It does not affect how the solver works. So now let us just this is the as what I called yesterday the canonical FBA form. What are all the variables in our case? What is f? c. What is x? x is v. What is A? Stoichiometric matrix.

What is b? 0 n cross 1 and lb, ub. Is it clear? These are all are variables that we need to worry about. **“Professor - student conversation ends.”**

(Refer Slide Time: 04:55)



Now let us go back to linprog. So linprog has multiple ways of calling it. So the basic thing first thing you need to notice it is a minimize f transpose x not maximize. So you will have to instruct it to maximize differently or you just say –f right, minimize of –f is same as maximize of f, subject to Ax<=b. So you can call it in several ways, the first one is linprog f, A, b right.

We do not have this kind of a inequality constraint, so let us ignore that. Then, we have linprog f, A, b, A equality, b equality that sounds better, even better for us is x=linprog of f, A, b, A equality, b equality, lb, ub. So this defines a set of lower and upper bounds on the design variables x so the solution is in the range lb<=x<=ub perfect. This is in perfect tune with the constraints we saw here yesterday.

(Refer Slide Time: 05:55)

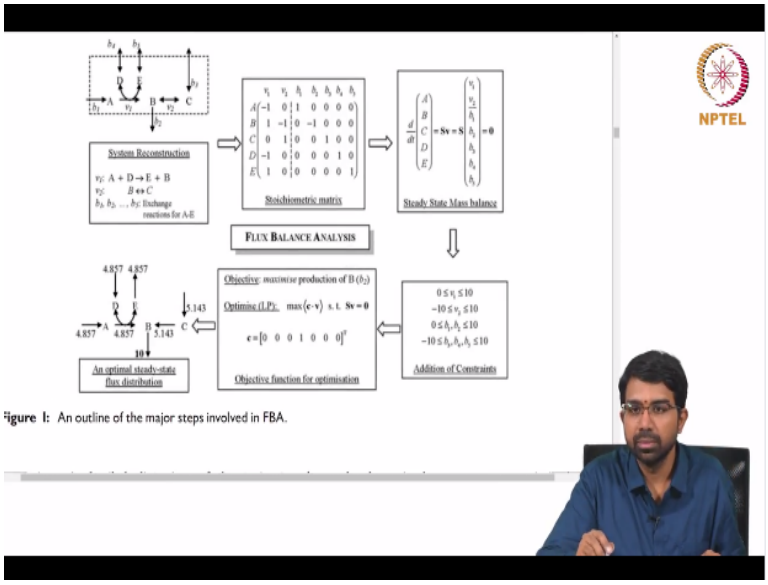
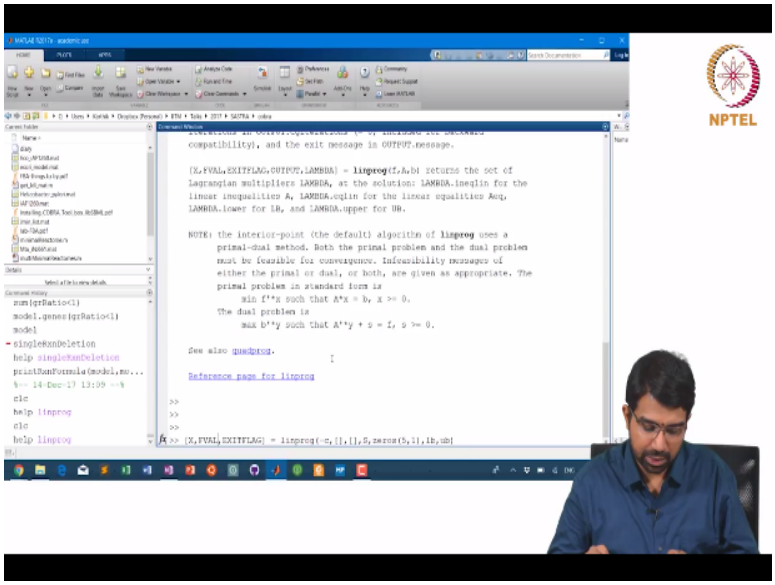


Figure 1: An outline of the major steps involved in FBA.

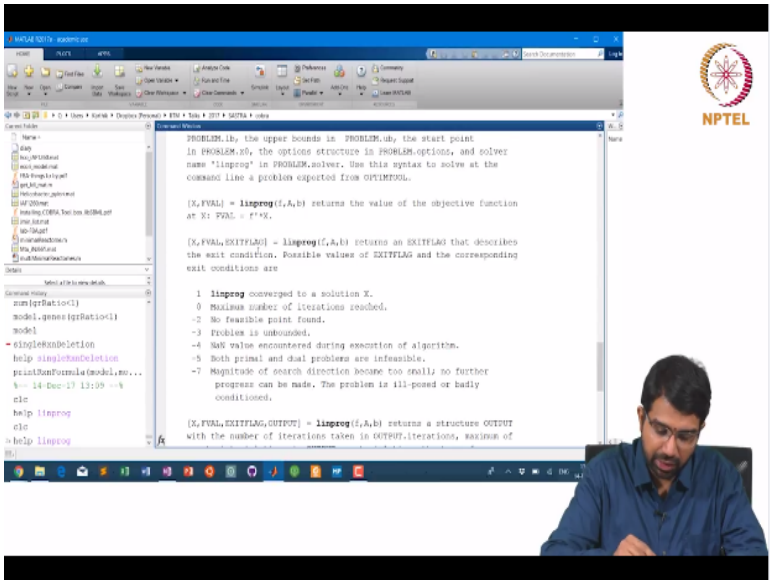
"Professor - student conversation starts." It is dimension of x. you know for single variables that is what. It is a matrix know, A is a matrix, so it is all inequality constraints are clubbed into the A matrix, all equality constraints are clubbed into A equality matrix. So this looks good. So you know what is f, what is f? c, A? no inequality constraints, b empty, A eq=s, B eq=0s, lb ub you know, we do not even need to give a x0.

(Refer Slide Time: 07:30)



So can you set up this problem now, empty, empty so A and b do not exist, so we will just put empty in that space. So I would essentially call it as X, FVAL, EXITFLAG is linprog of –c because I want to maximize, empty, empty s, 0s, lb, ub right, function value. So let us once again look at them. What are the output variables?

(Refer Slide Time: 07:56)

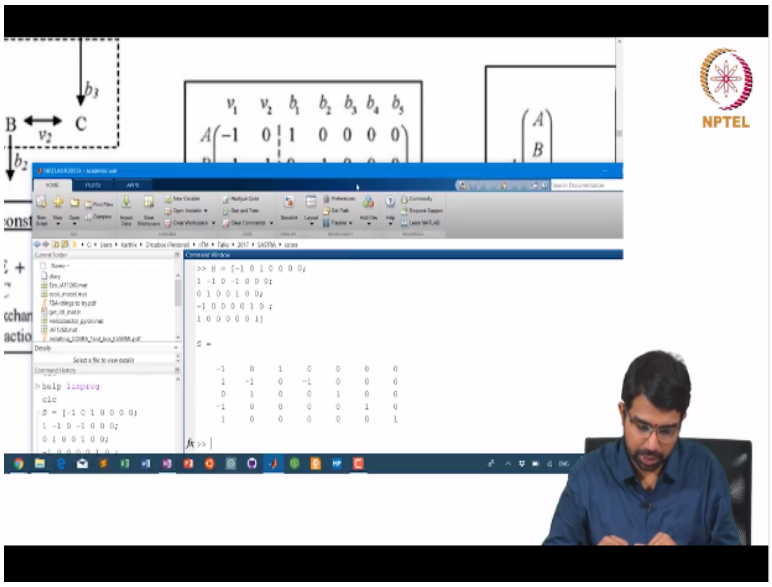


The output variables are X, FVAL and EXIT FLAG very important thing you need to note is the EXITFLAG. You need to check the value of EXITFLAG before you make any decisions on what to use the optimized value or not right because an optimization is essentially a search. What is the guarantee that this search successfully terminated right? There is a possibility so if you try to optimize this term x+y=5 and x+y<0 as a constraint, it is not going to work right. **"Professor - student conversation ends."**

So it would say that the no feasible point is found right or occasionally it can reach the maximum number of iterations and so on and so forth right. So you need to check if your status is 1, so let us assemble the problem now and FVAL is the value of the function. So as a reminder which of these is unique? FVAL is unique. You may get different X's using different algorithms right, different solvers right.

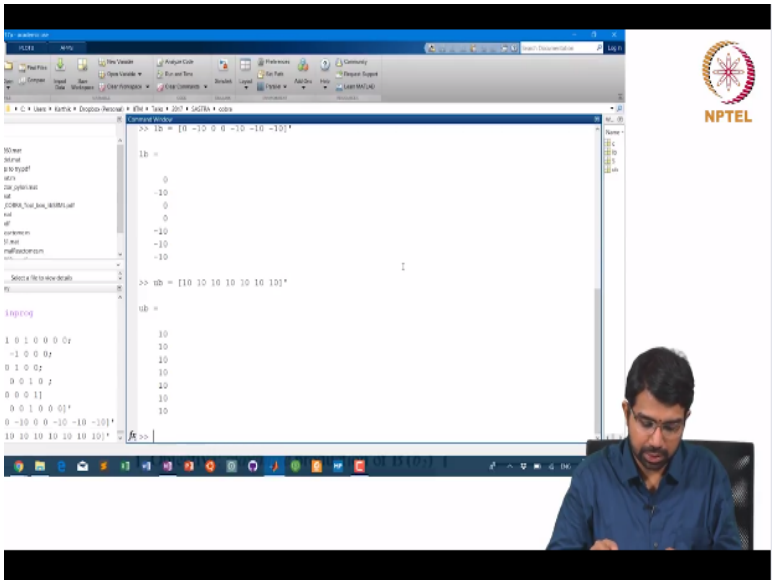
"Professor - student conversation starts." Invariable because the most of the algorithm used for LP are deterministic. **"Professor - student conversation ends."** What we discuss in the previous class was that there are several equivalent optima that are possible equivalent in the sense c transpose x is the same, x will be different so what is s?

(Refer Slide Time: 09:25)



You can just type this out. So that looks correct.

(Refer Slide Time: 10:44)

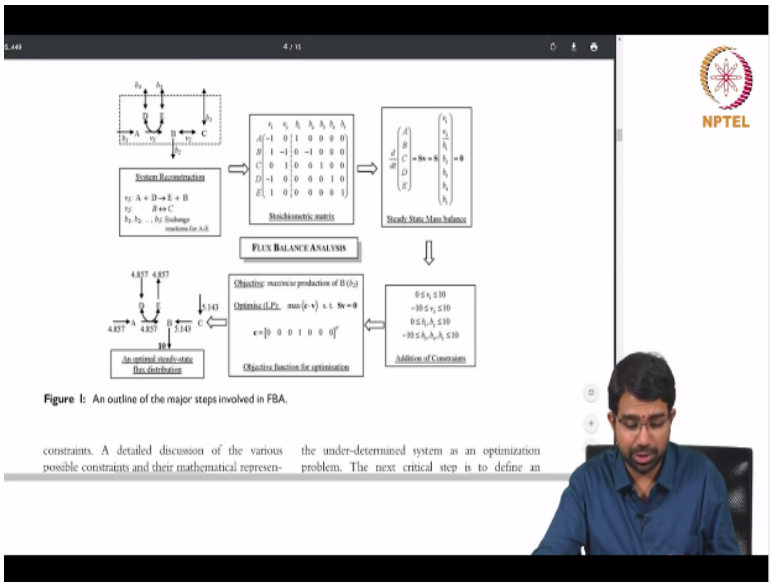


We now have the s matrix in the memory. What is the 0 0 0 1 0 0 0 lb=this, ub=this. Now I can say X, FVAL, EXITFLAG=linprog –c, s, 0s, 5, 1 5 0s, 5 metabolites, lb, ub. So importantly it says optimal solution has been found. There are other solvers, so there are commercial solvers thankfully many of them are free for academics, so there are solvers like Gurobi and son on. We will be using them when we look at harder models.

Larger models I think linprog might be too slow although linprog seems to have improved a lot since even the previous version of MATLAB I think. So what is the value of X you expect? So what is the value of FVAL you expect? FVAL is the value of b2 right so what is

the value of b2 you expect? 10 but you might have a very sparse solution where all these are 0, this is 0, this is 0, this is 0, this is 0 and this is 10, 10 and 10.

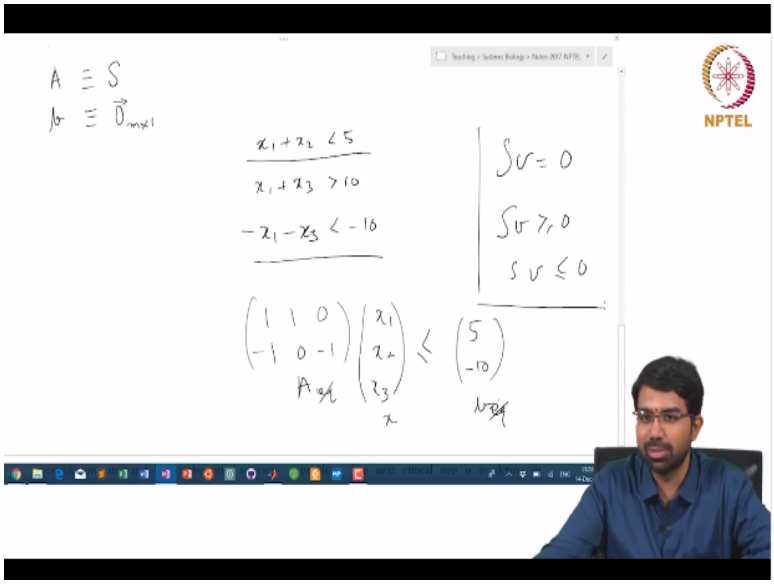
(Refer Slide Time: 12:13)



Let us see what the solution looks like, so FVAL is -10 because we negated the c function. EXITFLAG is 1 because anyways its optimal solution was found and X is this you see it is quite sparse, it is 0 -10 0 10 10 0 0 so it is 0 for v1, -10 for v2 meaning it goes in this direction C to B and 0 for b1, 10 for b2, 10 for b3 and 0 0 for the others. So it is basically saying 0 0 0 0 10 10 10 **"Professor - student conversation starts."** So it made a very sparse solution.

This is similar to what we were talking in the previous class right. If you make the solution that you have as sparse as possible right. Suppose you wanted x+y=10 and at the same time minimizing the number of nonzero elements in that vector, so you would get an answer of 0 10 or 10 0. Which constraints? These are the lb's and ub's, I did include them. So that goes into A equality, put them into a matrix right. Why is that a little confusing?

(Refer Slide Time: 14:24)

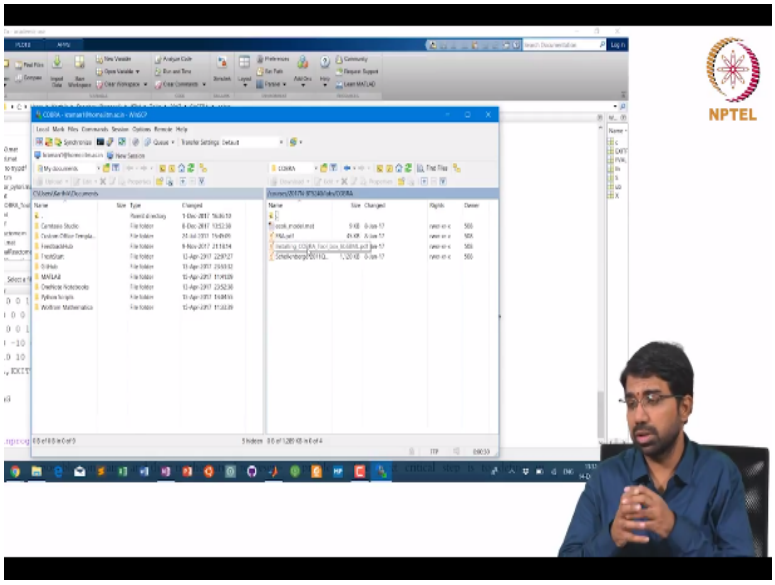


So tell me your equations right, so first thing is let us look at the MATLAB Hill, the constraints are in one direction only Ax<=B so the greater than constraint you change it, you make it –x1, -x3<-10 right. Now these two you can stick them into a matrix right, you can say 1 1 0 -1 0 -1*x1 x2 x3 <=5 -10 Ax b0. So in fact can you tell me how do you convert this into an inequality constraint?

It is an equality constraint how do you convert it into an inequality constraint? So you say Sv>=0 and <=0. You cannot right because I am saying it is also <=0 but you are unnecessarily amplifying the problem. It is good to know these because occasionally you might need to do this to debug something. So have you done this, have you solved the linear program?

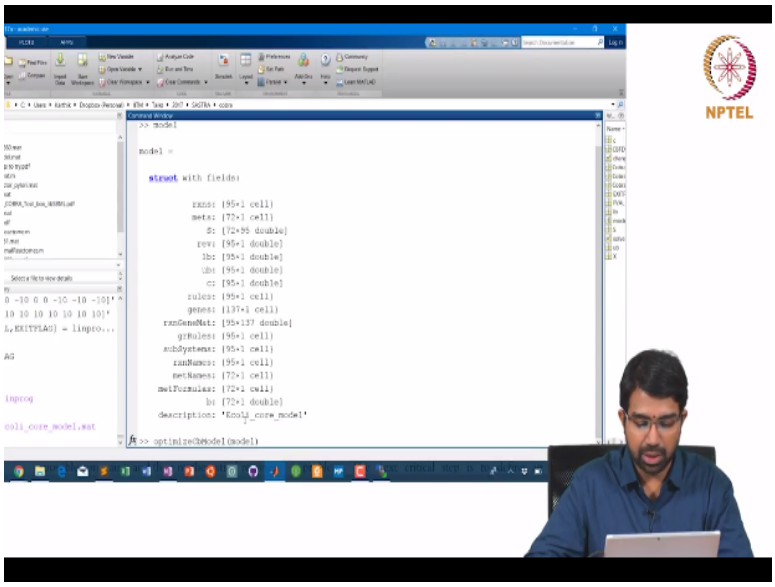
No, no you have to download them separately. It supports many solvers; it does not have any solver. No toolbox will come in with a solver but it supports many solvers, it has interfaces for several solvers.

(Refer Slide Time: 16:55)



Can you download this file? Labscobraecoli_model.mat and also this installing COBRA toolbox. You can follow these instructions and install the COBRA toolbox, so you will have a formal model to the COBRA toolbox in the next lab. For now let us look at how a COBRA based model looks like. **"Professor - student conversation ends."**

(Refer Slide Time: 17:21)



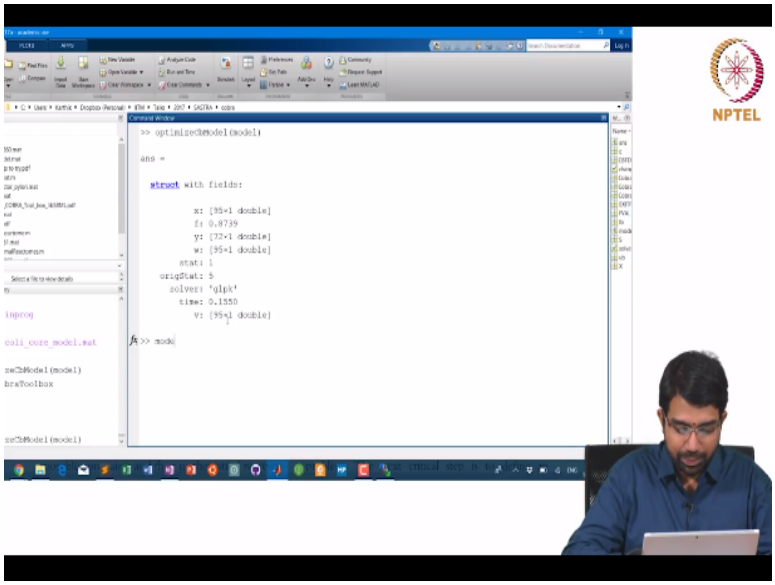
COBRA stands for constraint-based reconstruction analysis toolbox, COBRA toolbox, so how does the model look like? So the model is a MATLAB structure with several fields, we can make sense of most of them I would say. So rxns stands for reactions id's, mets stands for metabolite names or id's I mean metabolite id's essentially or the internal naming of the metabolite, s is your stoichiometric matrix.

So obviously that is going to be m cross r so it is 72 cross 95, rev as you might guess is a vector that talks to you about whether your reaction is reversible or not. It is a Boolean vector, lb and ub are lower bound and upper bounds, c is your objective function. Rules tell you something about what genes are involved with what reactions and so on. Then, genes is a list of genes.

Reaction gene matrix tells you what genes are incident on what reactions. Let us skip some of these, they are not so important. Reaction names tell you what are the names of the reactions, something like exchange of glucose and so on, met names are the metabolite names, you know that therefore the glc is glucose and things like that, met formulas are chemical formulae of the metabolites, b is nothing but your 0 vector right m cross 1 0 vector and this is the description.

This is a simplified E. coli core model, so several reactions are lumped together, the actual E. coli model has something like 1260 genes and so on. This is a very simplified reaction model with just 137 genes involving 95 reactions. So I will use the COBRA toolbox now without actually telling you how to use it and run it and let us see if I get a solution.

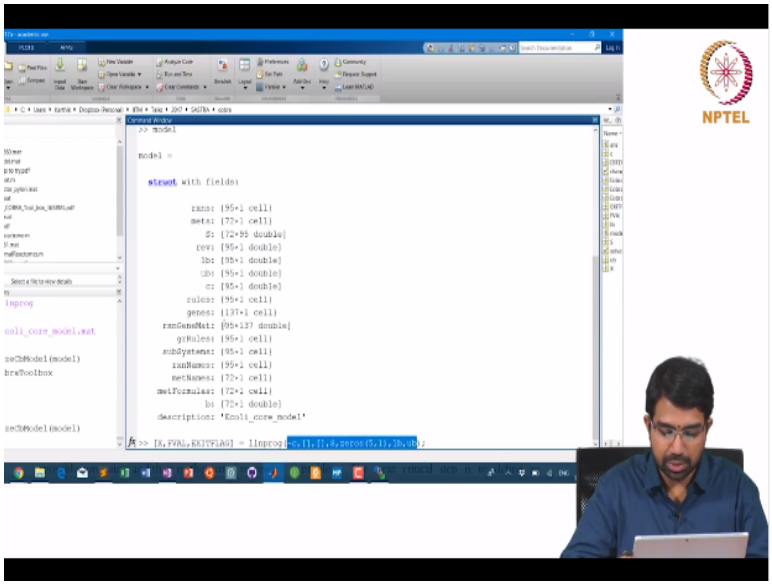
(Refer Slide Time: 19:26)



I need to initialize the COBRA toolbox. So I now get a solution, optimizeCbModel means optimize constraint-based model. We will go through all these functions in a later lab but here what you have is the output, you have an x vector which the understand can be varying right, you can have multiple optima; you have an f value which should be the same no matter what solver you use.

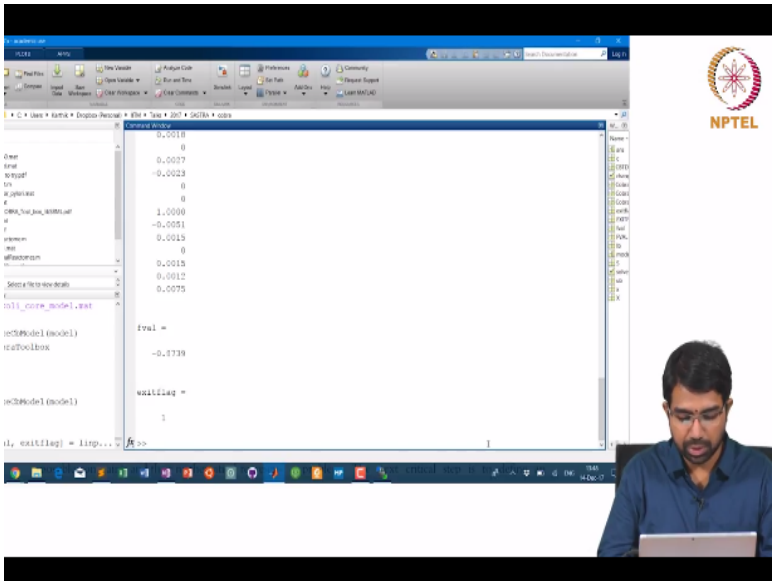
Then, these are other factors. Let us not worry about them at this point. Then, there is a status of the solver, the solver, the time it took and so on. So can you get the same solution using linprog now?

(Refer Slide Time: 20:18)



You have model, you can use a combination of things from model as input to linprog, you can just basically say X, FVAL, EXITFLAG=something, something, something. So the solution is simple.

(Refer Slide Time: 20:38)

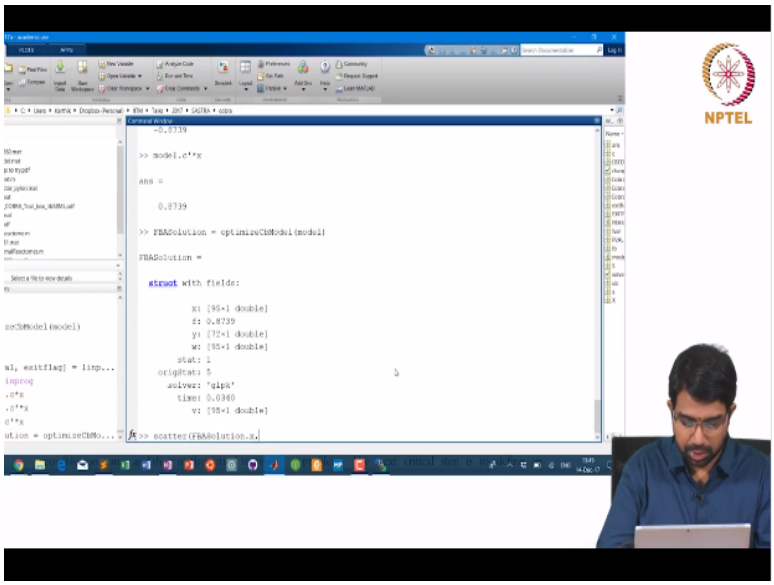


What was the formula we used previously? This was the formula we used previously. So the same idea we use again. We say X, FVAL, EXITFLAG=linprog of –model.c empty empty empty I think that is fine. It really does not make a difference so empty empty then model.s

model.b model.lb model.ub. We get the same FVAL of -0.8739. So here it says clearly set A=empty and B=empty if no inequalities exist.

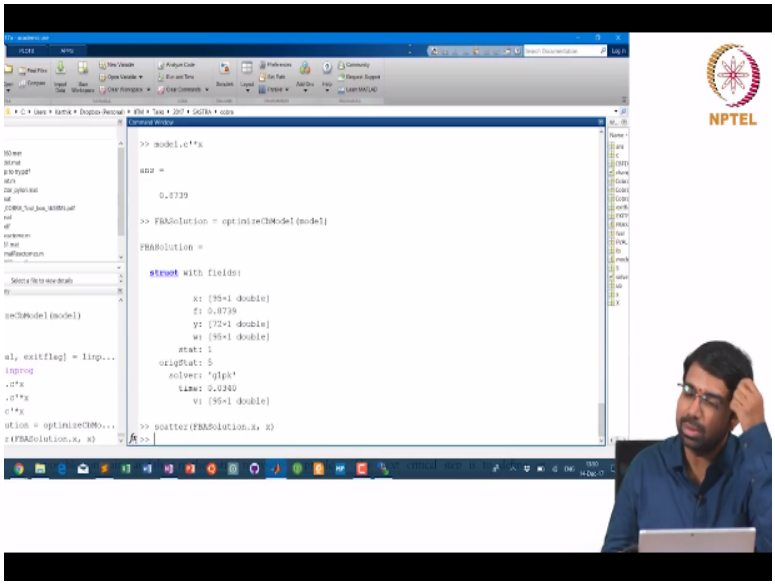
I thought there was a way to ask to maximize but no big deal right you can always say –c **"Professor - student conversation starts."**

(Refer Slide Time: 22:18)



So you will just do –c*x so –model.c*x transpose okay. You do not need to do the – any number, you already did it right. So the x value is actually proper. **"Professor - student conversation ends."**

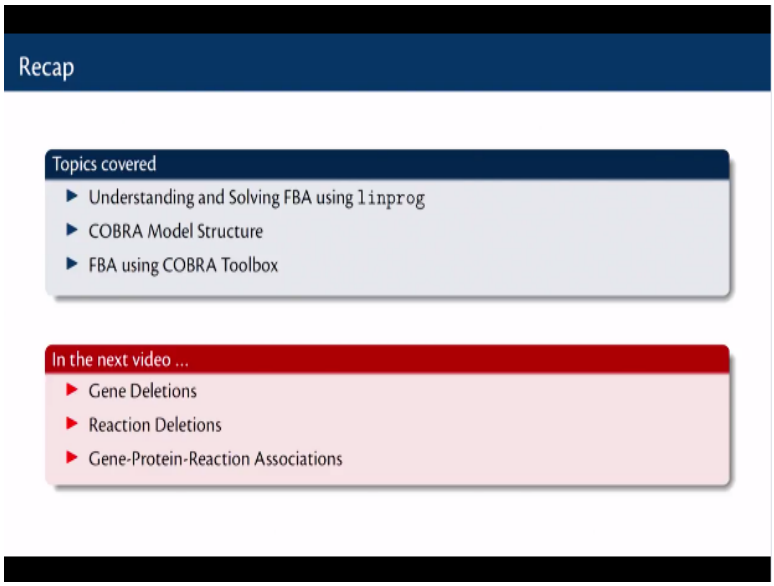
(Refer Slide Time: 22:43)



So let us again try so let us compare scatter FBA solution.x versus our x, so you have two values that are really off but because I think COBRA will try to make the solutions sparse and

things like that anyway okay so got this. I think this is important. So okay we will stop here for now.

(Refer Slide Time: 23:01)



So I hope you got a good overview of how we perform FBA using MATLAB. For simple models, we can use things like linprog and for more complex models, genome scale models one tries to use the COBRA toolbox and some commercial optimizers which are most often free for academics such as Gurobi and so on. We also viewed how the COBRA model structure looks like and we also performed an FBA using the COBRA toolbox itself.

In the next video, we will look at perturbations to these metabolic networks. We have already studied how you optimize or perform flux balance analysis on a given metabolic network in a particular condition. Now what happens when you start perturbing this metabolic network by either you know overexpressing a gene or deleting genes or deleting multiple genes and so on. So in the next video, we will start looking at gene deletions, reaction deletions and the concept of gene-protein-reaction associations.