Advanced Numerical Analysis
Prof. Sachin Patwardhan
Department of Chemical Engineering
Indian Institute of Technology – Bombay

Lecture - 48
Methods for Solving System of Differential Algebraic Equations

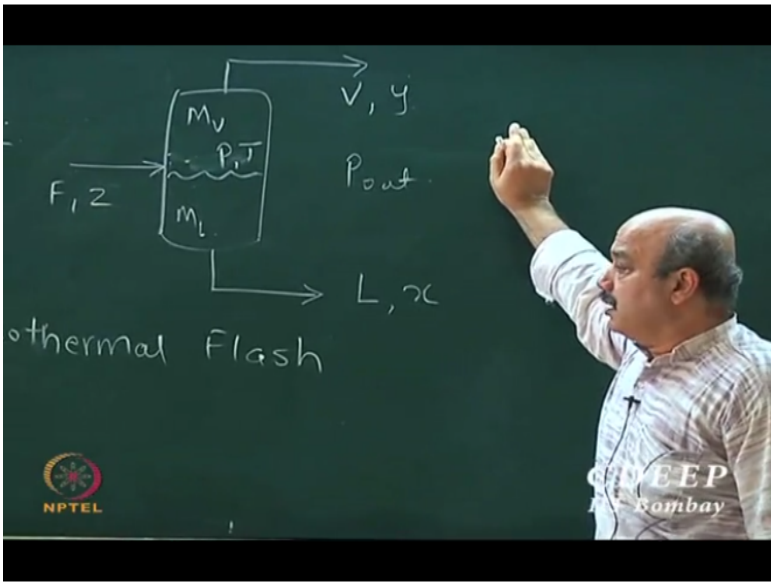
So, we have been having look at ordinary differential equations with initial value specified and we have looked at different methods of different numerical schemes of solving ordinary differential equations subject to initial conditions. Then we also had some introduction to how does one do analysis of convergence of error between the true solution and the approximate solution. And finally we saw one application of using an initial value problem solver for solving an ordinary differential equations boundary value problem.

So, this was shooting method and towards the end, I also mentioned something about stiff differential equations. So, stiff differential equations are the ones in which different variables have different time scales. So, some variables are acting on the fast time scale, some variables are acting on a slow time scale and that gives difficulties in choosing integration interval. So, convergence and choice of integration interval were tightly coupled and we had to look at Eigenvalues of the Jacobian matrix to get some clues into how to choose integration step size.

You also looked at a fix that is variable step size method by which when we do not know how to choose the integration step size, we could get over the difficulties. So, today I just want to briefly touch upon one very, very important topic. This is solving differential algebraic systems. These are popularly known as DAES. In fact, in the real chemical engineering applications, you are more likely to encounter differential algebraic systems than pure algebraic or pure differential systems.

Because in many situations, some phenomena have to be modeled as algebraic constraints okay. A classic example would be multi component flash.

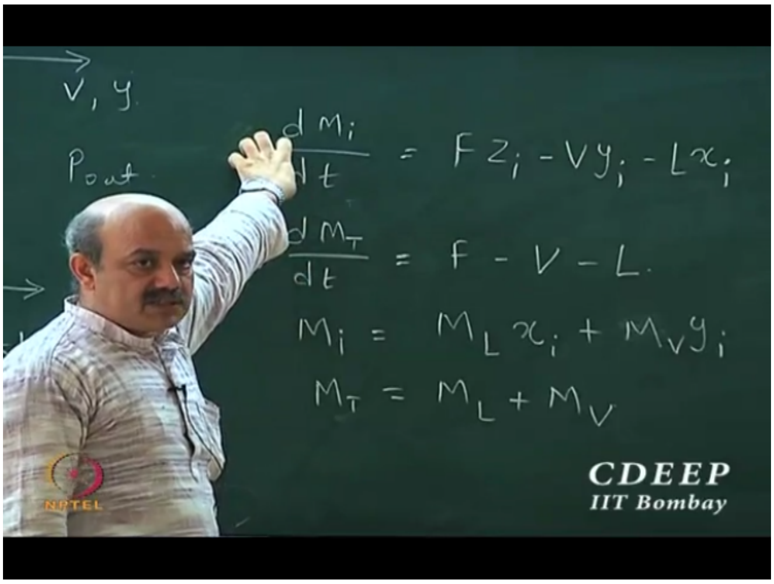
(Refer Slide Time: 03:34)



So, an isothermal multi component flash, I am just reducing the complexity by taking isothermal flash. When you are doing design of isothermal flash, you are typically concerned about the steady state operation. When you are looking only at a steady state operation, you only have to worry about algebraic equations. There are no derivatives coming into the picture.

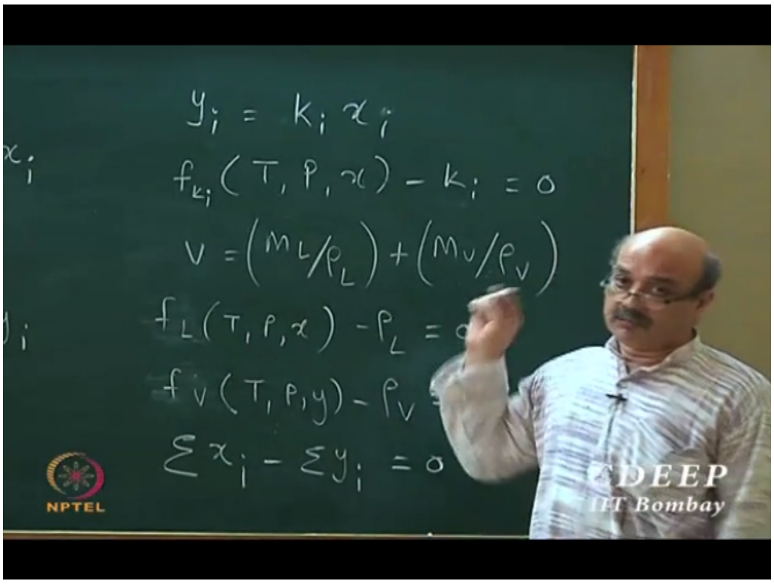
But, if I want to simulate dynamics associated with this flash okay, more complex problem of course will be distillation column, flash is only one tray like a model of the flash might resemble with just one tray. But, just to give you an idea here of how differential algebraic systems will arise. I am just going to write down differential equations and algebraic equations under the assumption that temperature is specified and constant.

(Refer Slide Time: 05:25)



So, this is the component balance for the ith component okay. These are overall mass balance component balance okay. What is overall mass for ith component than total mass. Liquid mass plus vapor mass.

(Refer Slide Time: 06:42)



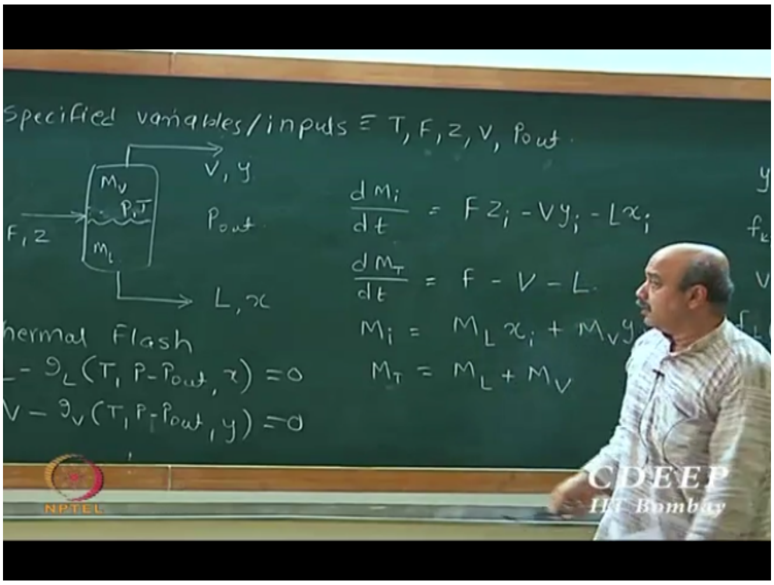
And you have this yi = kixi thermo dynamical relationships. I am just writing this in a abstract way. There is some thermo dynamic model for computing ki values okay. And then the relationship between yi and xi is given by this. This is the thermo dynamic model representation, each ki will be found by some thermo dynamic model okay. Well, then you have total volume in the flash. You will have models for the density in the liquid phase, density in the vapor phase.

Density in the liquid phase will be function of composition, pressure, temperature okay. And so is the case for the vapor phase density. And then you will have 2 basic composition equations, sigma xi=1, sigma yi=1 or you can combine them and say that sigma xi-sigma yi=0. Remember I just want to do dynamic simulation of the isothermal flash okay. So, I cannot ignore the 2 dynamic terms or 2 differential equations that I get.

Not 2 differential equations I am sorry, if I have n components, there will be n+1 differential equations. The n components will be n+1 differential equations, one for each component balance and so there are n components so n differential equations for each component +1 overall mass balance okay. So, there are n+1 differential equations and many more algebraic equations. Somehow the algebraic equations, you might be able to eliminate and substitute but not all okay.

In such cases when you cannot eliminate all the variables, you will get a differential algebraic system okay. And then I am just writing 2 more equations, you have to write equation for the liquid flow out. Liquid flow out will be function of the mass here, if it is gravity kind of flow and vapor flow out will be function of pressure in - pressure out. Here also it will be function of pressure in - pressure out. So, I am just writing the general equation here.

(Refer Slide Time: 10:03)



These are the 2 flow equations okay. Now, the variables that are specified in this particular case, these correspond to T, F, Z, V and P out okay. These are the variables which have been specified. Isothermal flash so, T is specified okay, V is specified because you have designed the flash for a certain V or you want to keep the V constant okay. Then, feed conditions are specified and the pressure out is specified.

So, these things are specified and then you want to actually solve this coupled differential algebraic equations together okay. A typical problem in chemical engineering, where if you have reaction occurring there with 2 feeds coming in becomes more complex okay. You have conversion from one to another and then you have to maintain a quite likely that your products are gas phase under reactants are liquid phase. So, a very common scenario and when you are drawing the product.

So, differential equations and algebraic equations are coupled and this is what we have to solve together okay. So, just algebraic equations, just differential equations these situations arise only in certain cases but, if you are doing dynamic simulations more often than not, you get into DAE systems okay. Other example I had given previously of DAE systems was orthogonal collocation. It is not necessary that we also get DAE systems. You get DAE systems, when you are solving only dynamics.

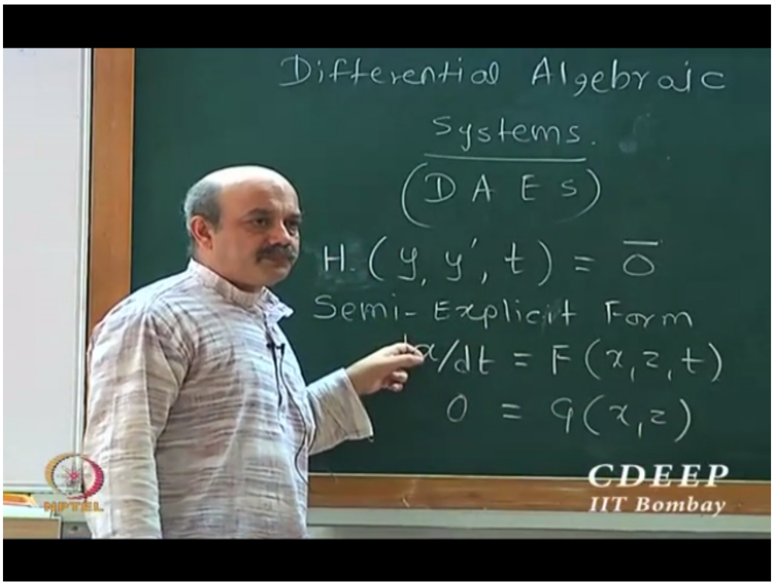
If you remember tubular reactor with axial mixing, TRAM problem okay. When we discretize this problem in space okay using orthogonal collocations. The boundary conditions or algebraic conditions okay. Of course in that particular problem was dynamics of TRAM but, if you had to do solve a Laplace equation by method of lines okay. So, you have differential equations okay in what direction, in the other direction, you have discretized okay. And you might get algebraic conditions.

So DAEs need not arise strictly when you have time. It can arise, we have only 2 spatial coordinates and you are discretizing in one spatial coordinate, you are solving as a ODE in the other coordinate. So, it can arise in some other context, not just time simulations. But in time simulations, quiet often you get differential algebraic systems. So, solving differential algebraic systems is much more complex than solving just the differential systems or algebraic systems okay.

And you should know at least something about that. So, the idea of this one lecture is to give you a very, very brief introduction to some of the issues that are involved in solving a DAEs okay. What I am going to do is as I said just touch the tip of the iceberg but, you should know that there exist separate methods for solving DAEs and also toolbox in mat lab or some other thing like si lab would have solvers, which can handle DAEs, mat lab has a solver 1 pi ODE, 1 pi S, S stands for stiff.

So, ODE 1 pi S solves differential algebraic equations, solves equations simultaneously.

(Refer Slide Time: 15:14)



So, now let me abstract this, let me well, the most general form of differential algebraic equations, I can write as F y, y prime t=0 okay, where y is a vector and you have some coupled equations of derivatives and algebraic. So, some of these could be algebraic state some of these could be differential states. What I mean by algebraic states and differential states, we often write what is called as semi explicit form.

We often deal with what is called as semi explicit form. The semi explicit form is dx/dt=well, let us use a little bit of a different notation here, let us make this capital G and because we have been using F for derivative. X, z, t and one minute not G, we will use here H, H is some nonlinear function of y, y prime and t. and then I can write a semi explicit form, which is 0=G of x, z okay. The flash equations, you can rearrange into this semi explicit form dx/dt=F x, z, t okay.

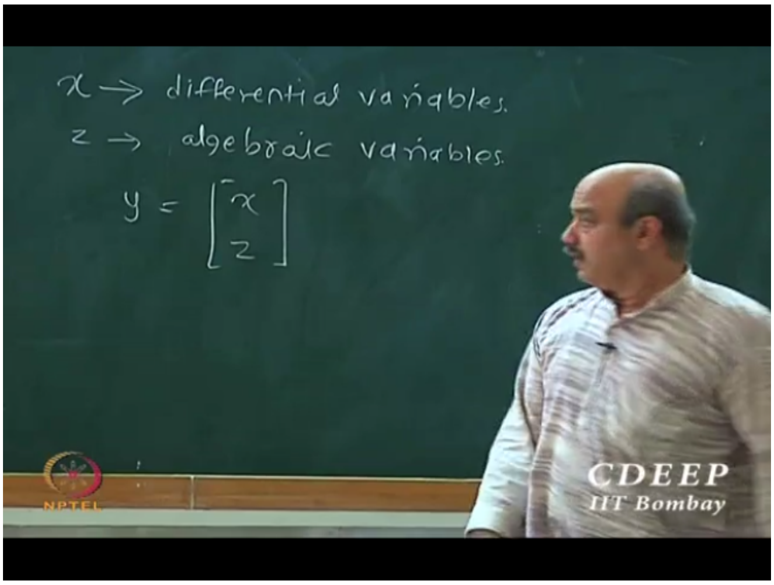
And 0=G of x,z, where x we call as differential variables and z we call as algebraic variables. Differential variables as in you get dx/dt terms okay. Algebraic variables because in the raw form, you do not derivatives of z okay. You may have to find out derivatives of z for some reason we will come to that okay. But, right now, in this lecture we are going to look at this simplified form, most of the cases that we encounter can be written into this form okay.

And then, we want to solve them simultaneously okay. Now, you can appreciate here that these are relatively difficult because, initial condition for ODE, initial value problem okay, you can specify an arbitrary initial condition. Here you cannot specify the arbitrary initial condition. The initial conditions must satisfy this constraint. This is a non-linear algebraic.

Linear or non-linear, generally in chemical engineering problems, these constraints should be non-linear.

There are non-linear algebraic constraints, this must be met by the initial condition on x. so, you have to give a initial condition of x on z, which satisfies this constraint. That is very, very important okay. And as you march in time okay, at all times this constraint has to be satisfied okay. So, solving DAEs is a scale more difficult okay.

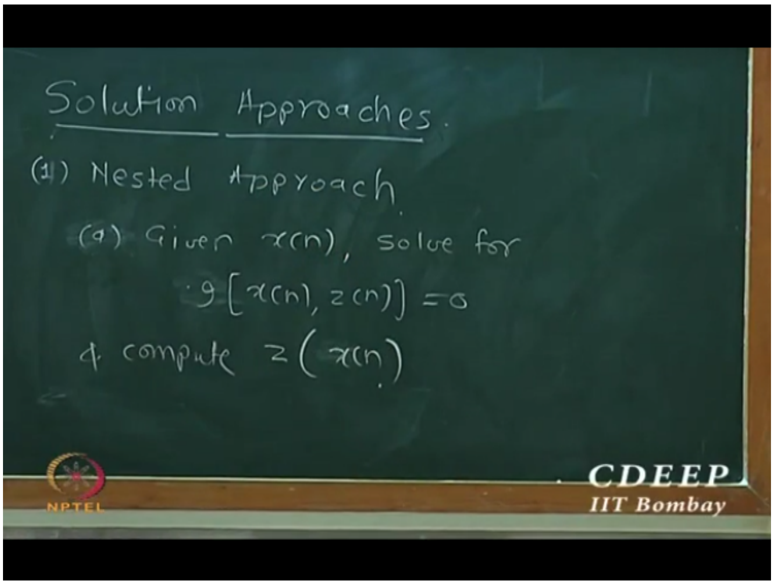
(Refer Slide Time: 18:42)



In this classification, so this particular form is called as fully implicit whereas this is called as semi explicit form. And in this x we call as differential variables and z are algebraic variables. And of course y is nothing but x z put together okay. Y is nothing but x z put together. Actually, the fully implicit form, if you take this definition okay, then you can convert semi implicit form into so, that is not really a.

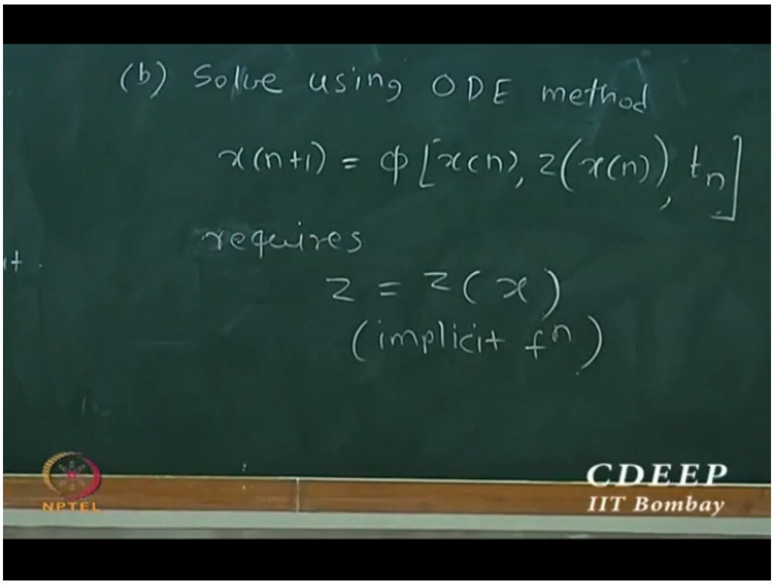
So, we will only look at semi implicit form to simplify our life. Now, how do you solve this? Okay.

(Refer Slide Time: 20:14)



So, solution approaches, one is called as nested approach okay and the other one is called as simultaneous approach. So, the nested approach, first thing we do here is okay. So, basically we get z of okay. The first approach is that given xn, you solve for this and the next step would be using ODE method.

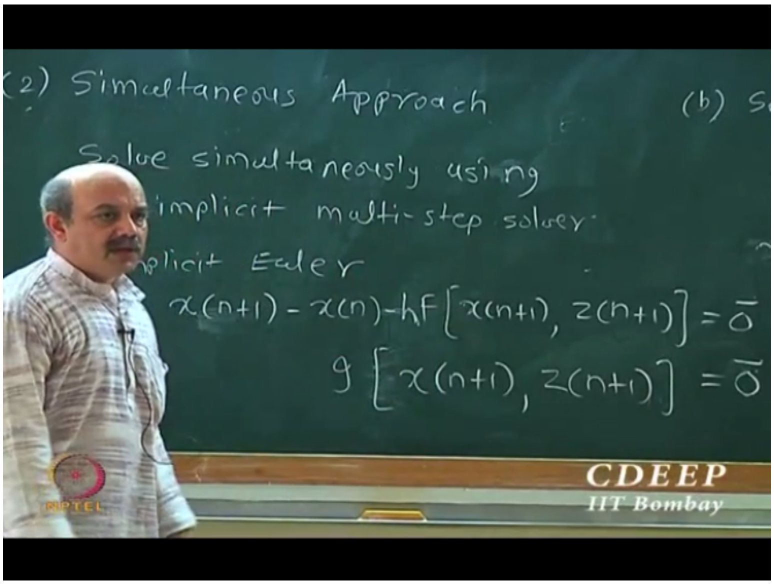
(Refer Slide Time: 22:14)



Now, what it means is that, when you are solving this differential equation, you are going from n to n+1. Internally when you are computing your function derivative okay. Inside, you may have to solve every time inside you may have to call Runge-Kutta method or Newton's method okay. Within your function evaluations, suppose you are doing Runge-Kutta, you have to do evaluations or the function at intermediate points okay.

At intermediate point, when you do calculations okay, we will have to solve for zxn okay. So, we will have to solve for you need an explicit way of or you need an implicit function. So, this approach requires z to be obtained as a function of x internally. Every time, let us say Runge-Kutta, if you find an intermediate x, for that x you have to compute corresponding z and then proceed. You cannot do. So, internally every time you have to compute or you have to solve the algebraic equations every time okay. So, that is one approach okay.

(Refer Slide Time: 24:35)



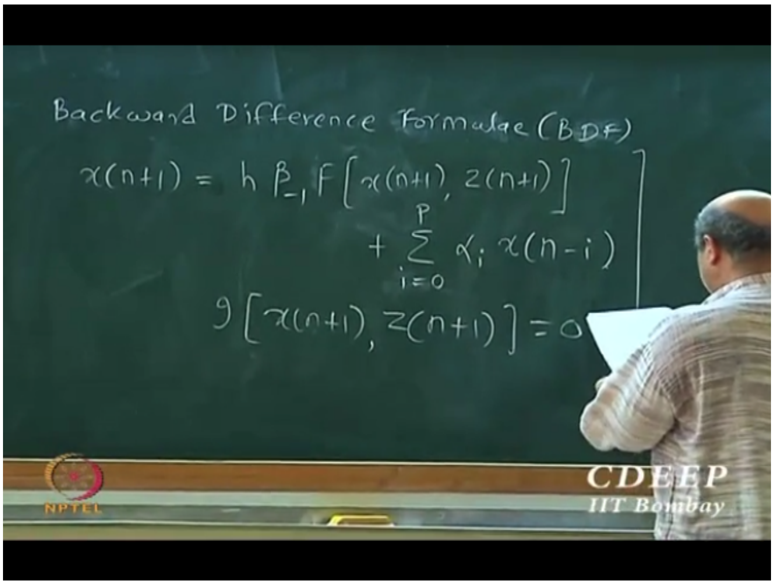
The second approach, is simultaneous approach. In the simultaneous approach, you use an implicit method okay to solve the differential algebraic equations together. So, I would say solve using an implicit multi-step solver okay. For example, simplest implicit multi-step solver would be implicit Euler okay. In implicit Euler, what I can do is I can solve for xn+1-xn-Fxnzn=0 and sorry this is implicit method so, xn+1, zn+1=0 and gx.

The simplest implicit solver would be implicit Euler okay h times right, h is my integration step size, h times this okay. So, this is my simplest implicit method. See, what is the advantage here? Advantage here is that both algebraic equations and differential equations get solved simultaneously and you get when you start from xn and zn, which are consistent, you get xn+1, zn+1 okay. Automatically xn, when you are going to the new point, xn+1, zn+1 they are consistent.

Because, you are making sure this happens. When you march again, again xn+2, zn+2 are consistent with the algebraic equations okay. So, implicit Euler and how do you solve this if this F is a non-linear function, you have to solve this using Newton's method or some non-linear algebraic equation solver. So, this you have to treat them as a simultaneous, these 2 equations have to be solved simultaneously.

Xn is known to you, xn+1 is not known, zn+1 is not known. You have to solve them simultaneously and then okay. Every time you solve, you march in time, you get consistent solutions okay. The other general class of multi step methods, which are used here, they are called as Backward difference solver BDS solver.

(Refer Slide Time: 28:27)

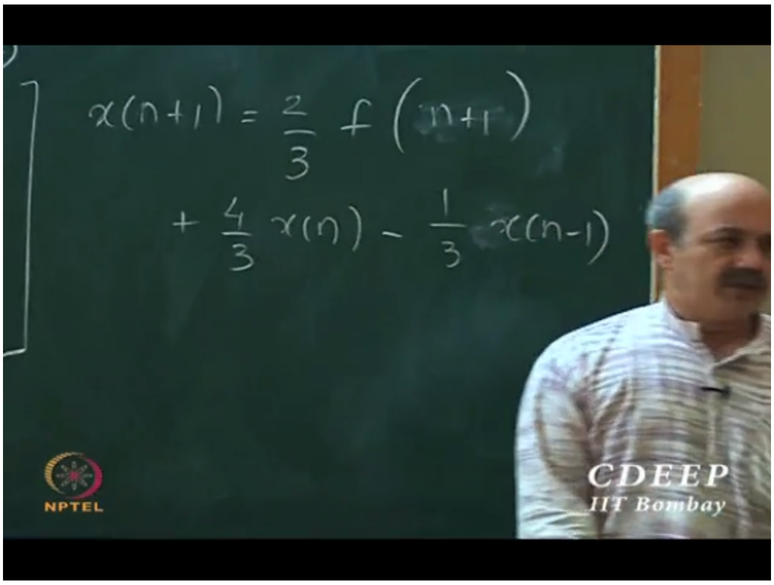


More popularly known as BDF okay. So, these are in our multi step parlance, these are methods of xn+1=h beta-1 F. see, we had earlier seen Adams Moulton gears algorithm, Predictor-Corrector so, this is general multi step algorithm. Here we do not use, we only use one derivative that is Fn+1 okay and all past x values okay and then here, you solve this together with g of xn+1. So, we solve this equation and this equation simultaneously okay.

So, the simplest example of this type is implicit Euler. A BDF solver, backward difference formulae, these are called as BDF formulae okay. A nice thing is that, you have xn+1, zn+1 appearing in derivative and here okay. And then only use past x okay. We are only using past x values. By the way, some of you had this question about, if you have multi step methods, how do you initialize. Because, in multi step methods, you need suppose it is the p step method, you need first p values right.

So, one approach suggested by Petzold and Ascher, there is a very good book on ordinary differential equations by Petzold and Ascher. So, they say that you start, for a p step method, you know you start one with Euler method, then 2 step method, then 3 step method, 4 step method. So, for first p steps, you use 1 to p-1 step methods. Generate the initial points. After that you continue using p step method okay. So, that is one way of initializing okay.

(Refer Slide Time: 32:16)



So, these backward difference formulae so, just to give you a example of this would be xn+1 = 2 third fn+1. So, this is one of the backward difference formulae and then you can write more. You can get this from the text book or you can derive. You know how to derive this, BDF formulae you will be able to derive. So, we are only going to use the derivative fn+1 and all x values from xn to xn-p okay. You can derive these coefficients.

So, likewise there are many methods okay. The other way of solving this differential algebraic equations, I am not going to do it on the board, because we have done orthogonal collocations quite a bit on detail. But, other way of solving is orthogonal collocations. What is the advantage of orthogonal collocations? It converts a differential equation into algebraic equations. Then you can club the algebraic equation coming from orthogonal collocation discretization and algebraic equation, which you have here and solve it together okay using Newton's method.

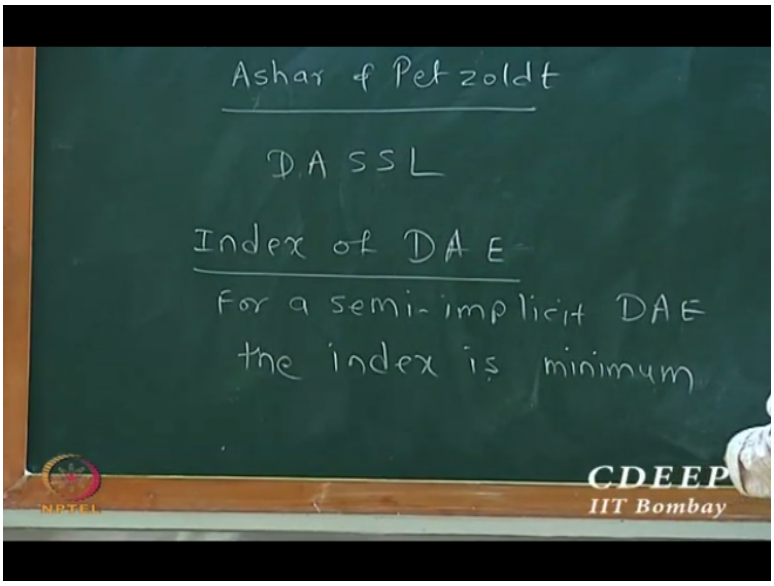
So, advantage of orthogonal collocation is to conversion to algebraic equations and then everything can be solved as one set of algebraic problem okay. So, that I am just leaving it to you to work out yourself may be in the exam or otherwise when it comes to. So, these equations can be solved using these BDF approaches or using something like Runge-Kutta, if you have implicit function. Z as a implicit function of x, then you can solve it using, you can call internally inside your Runge-Kutta, you can call Newton's method and then solve it.

So, these are the ways of solving. But, solving these becomes much more complex than just solving the ODEs because, in the function evaluation itself, we have to call algebraic equations solver, which could be non-linear. So, iterations inside iterations and so on. So, it can become quite complex okay. Now, unfortunately these equations, it is not just unfortunately just using these methods for solving this. Because there are many other issues that come up when you have differential and algebraic systems together.

One of the issue that you must know is index of a differential algebraic system okay. And what is difficult to solve is high index problem okay. We will define what is index of a differential algebraic system. Index one systems are easy to solve, high index systems are difficult to solve and you have to do some tricks. The main thing is that, when you have a differential algebraic system, you have to have the initial condition consistent okay.

That is x0 and z0 should satisfy the algebraic constraint. That is very, very important okay.

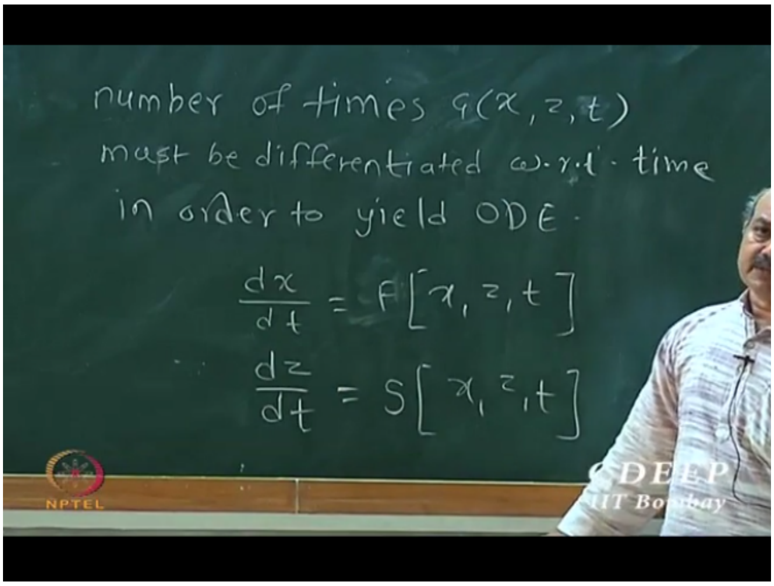
(Refer Slide Time: 35:59)



There is a text book by Ascher and Petzold on ordinary differential equations, a very good text book, which gives a very good mix of algorithms and theory. Talks in detail about differential algebraic solvers, there is a separate chapter dedicated to these systems and then Petzold has developed a software, which is very, very commonly used. I think it is a freeware and is called DASSL okay.

Differential algebraic equation solver called DASSL, which is I think this is a freeware, it is a Fortran program. You can download and then convert into a mat lab code and call it through mat lab and you can use. At least some small dimensional systems can be handled. Now, what is the index.

(Refer Slide Time: 37:50)



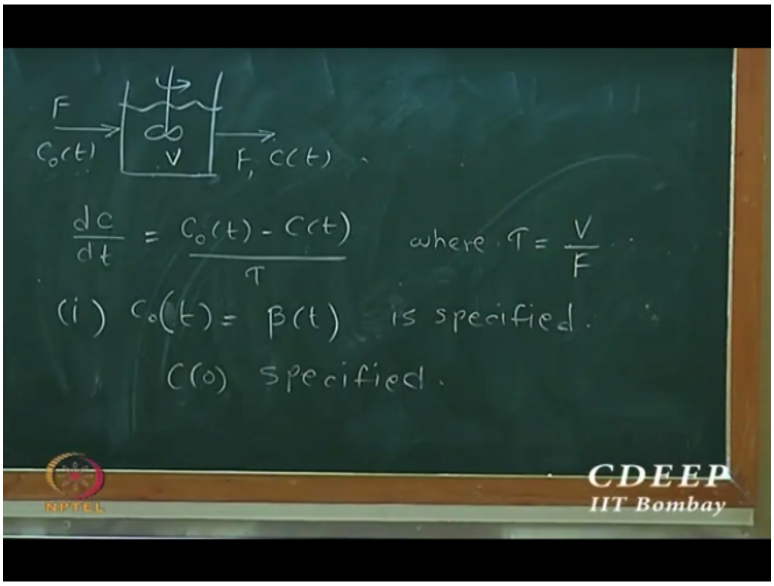
The index is number of times you should differentiate the algebraic equations okay to get a set of ordinary differential equations ODEs okay. Now, you might say well looks like you can convert this system into set of ODEs and then you can solve it using ODE solver. The problem is when you have algebraic equations okay, which means some derivatives are 0 all the time okay. This resulting system will be highly stiff system. It is not at all easy to solve.

It should be a stiff system though theoretically you can do this, it will be stiff system because, typically when you arrive at differential algebraic system or equations, you have chosen to neglect certain time derivatives. Because those time, the time scale on which the derivative operator very, very small, you neglect the time and you just look at the algebraic constraints. So even though on paper you can do this by repeatedly differentiating the algebraic constraints, this coupled set of equations will be difficult to solve.

Anyway, so the number of times you have to do differentiation do arrive at a set of differential equations is called as index. And in general, index one problems are easy to solve okay. If you differentiate the algebraic constraints only once and if you can convert them into set of differential equations great. If you cannot, index 2 systems or index 3 systems or high index systems are very, very difficult to solve.

And then you have to convert them typically the approach taken is to convert a high index system into a low index or index one system and then solve it as a index one system okay. Let us look at an example where we get high index problem. I will take a simple system, just a mixing tank okay.

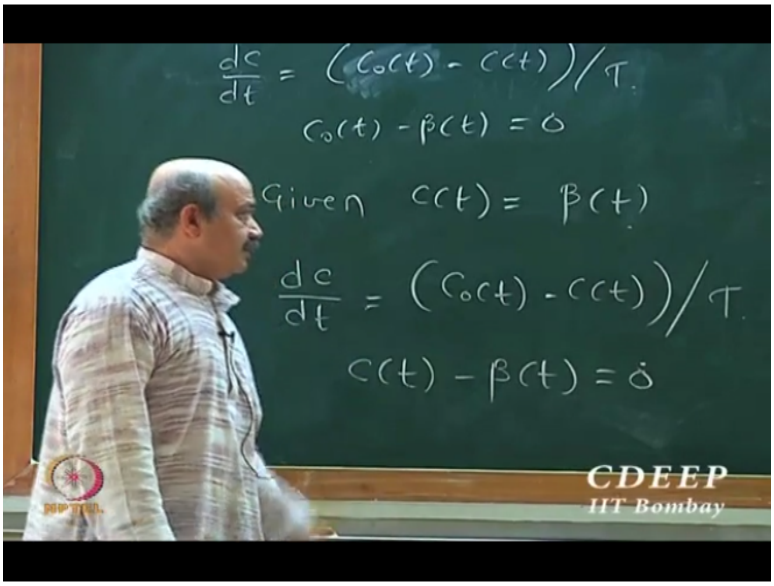
(Refer Slide Time: 41:11)



So, V is the volume here okay and one differential equation okay. And for this particular problem, I will just show you 2 phases of this problem okay. In one case, you can solve it as an ODE, in other case, it turns out to be a differential algebraic system with index 2 okay. So, one way is c0t, inlet concentration = some function beta t is specified okay. And then initial value of c0 is specified of concentration inside the tank is specified.

Simple ODE problem okay, you substitute this is a DAE, but then you can substitute for beta t here and you are given initial condition, it becomes one ODE, you can solve it. It is a very, very easy problem okay. The second one actually the other problem. This one, the first one is very, very simple DAE.

(Refer Slide Time: 43:50)

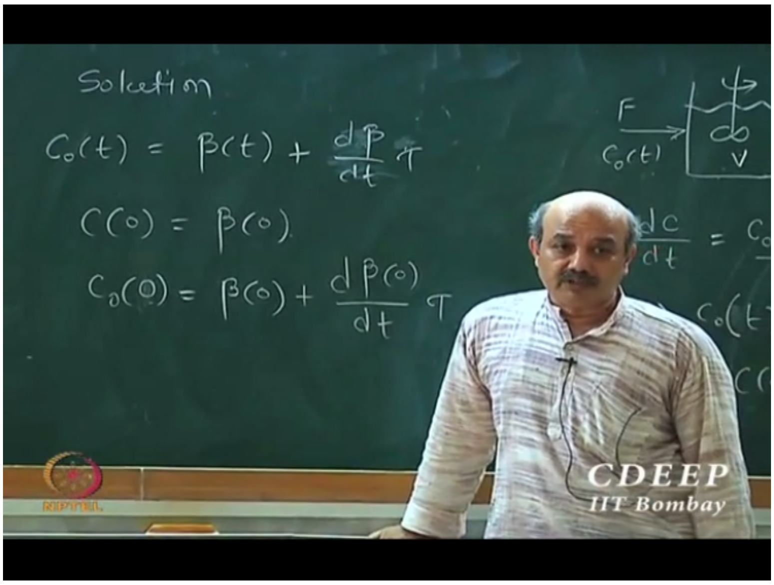


That is dc/dt=c0 t-ct/tau and c0t-beta t=0, so this is a DAE and substitute for beta, you can solve this using in the ODE solver, not at all an issue. Simple index one system, not at all difficult to solve. The second one here is earlier I had specified c0, inlet condition and able to do simulation okay. Now, I want to specify c of t okay as a function of beta t as some specified function of time = beta t.

Beta t generic way of writing a specified function. I am just putting this as beta t okay. See now, I have to solve for dc/dt=okay. It is very, very important how I choose the initial condition here. This is an example in which I have specified ct okay. This problem, this differential algebraic system, it turns out to be an index 2 system and this is much more difficult to solve.

So well, I will just quickly write down the solution and then we will also look at why it is index 2 systems that may be we have to do it in the next class.

(Refer Slide Time: 46:29)



But a solution for this problem is. You can see that solution to this problem will not be ct, ct has been specified. Solution to this problem will be c0 t, the inlet profile okay. Though, this is a (()) (47:35) problem okay, it actually tells you something that is intrinsic difficult about DAEs. Well, you might wonder where do you get this problem. For example, when I am doing control, I am specifying how should the output behave, I want to find out an input okay.

In a control problem, I am specifying how the control variable should behave and then I want to reconstruct the input, that will give me that particular controlled output okay. So, we will continue analyzing this particular system, we will see that, these particular systems turn out to be an index 2 system and then we have to solve it by doing some more manipulations. It is very, very important that the initial conditions are consistent. If you give a wrong initial condition, you can get into trouble here.