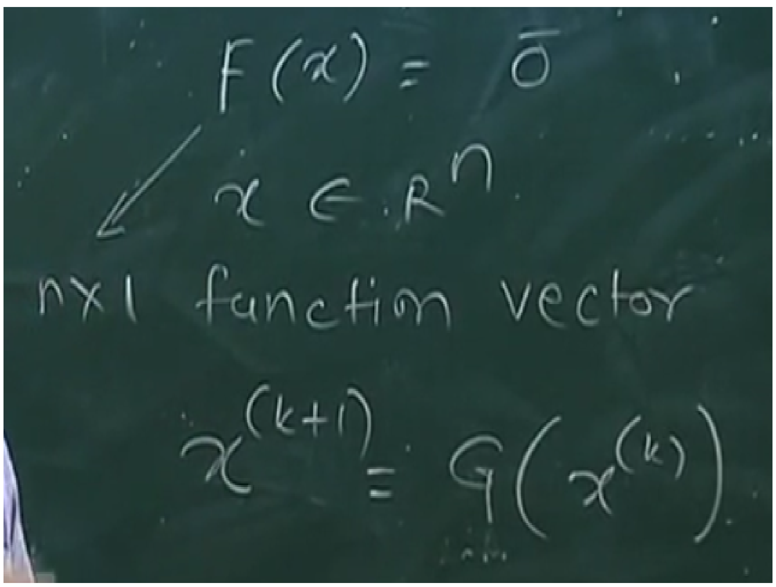
Advanced Numerical Analysis
Prof. Sachin Patwardhan
Department of Chemical Engineering
Indian Institute of Technology – Bombay

Lecture - 39
Solving Nonlinear Algebraic Equations: Introduction to Convergence analysis (Contd.) and Solving ODE-IVPs

In the last lecture we were looking at how to analyse convergence of non-linear, procedures for solving nonlinear algebraic equations. Iterative procedures and we said that in general we could write any iterative method for solving nonlinear algebraic equations as 1 equation.

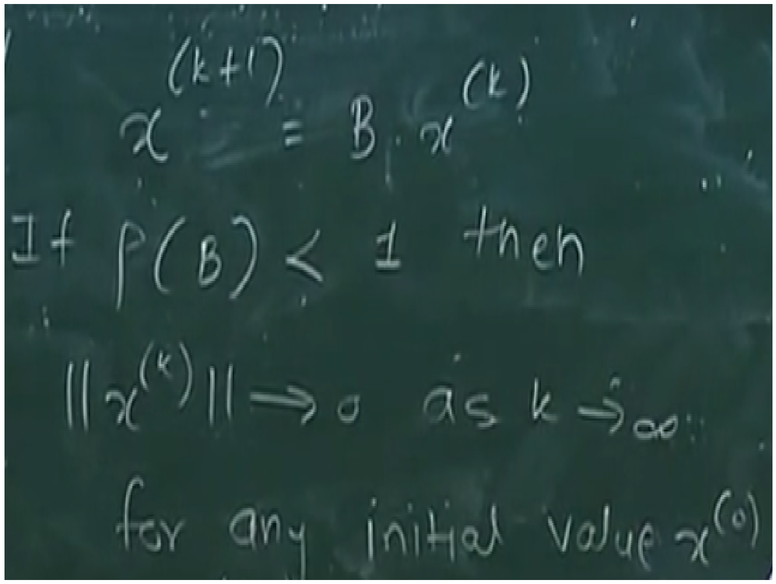
(Refer Slide Time: 00:53)



I want to solve for F of x=0, x belongs to Rn and F is a nx1 vector, this is nx1 function vector. Any iterative method to solve this problem numerically can be written as xk+1=G of xk. So the old guess generates a new guess and this process is continued till differences between 2 successive solutions become negligible or norm of F of x goes close to 0. If you look carefully, this is a nonlinear difference equation.

The index here is iteration index k. So the guess is generated from the old guess G is the transformation. I showed you that all the methods that we are looking at iterative methods can be expressed in this form. Now just like we had conditions for analyzing linear difference equations.

(Refer Slide Time: 02:41)

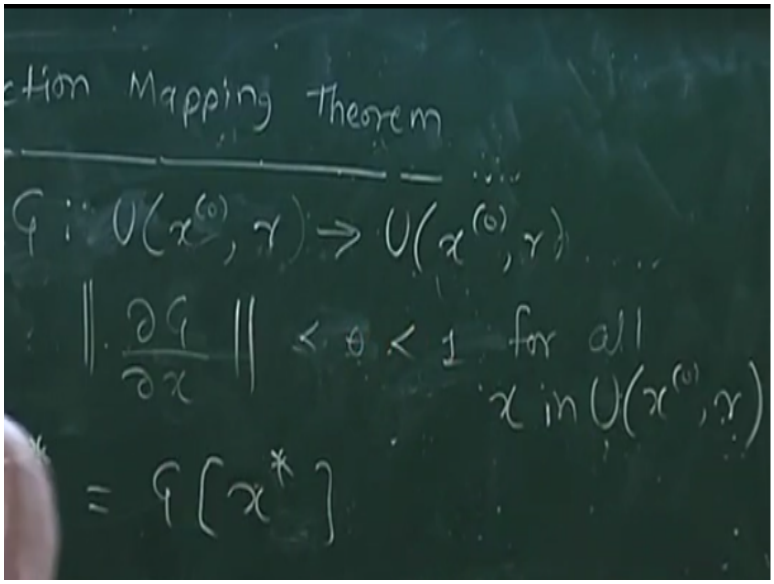


Earlier we had looked at equations of this type=B xk and for this particular case, we had derived necessary and sufficient condition for norm xk to go to 0 as k goes to infinity. In this case, we had a very, very powerful result that is spectral radius of B is strictly less than 1. This was the situation for the linear difference equation. We had got this kind of a generic form by analyzing iterative methods for solving linear algebraic equations.

We could derive a very, very powerful result here based on the Eigen value of matrix B. We wanted all Eigen values of matrix B to be inside the unit circle. Now coming to nonlinear equations it is not possible to prove so strong result. We can only give sufficient conditions. It is not possible to come up with necessary and sufficient conditions for a general nonlinear difference equation of that form. We have to come up with some kind of local condition.

These local conditions I described through contraction mapping theorem or contracting mapping principle, which forms the foundation of analyzing iterative schemes and 1 special that we saw was the operator G.

(Refer Slide Time: 05:21)



G is something, G maps a ball around x not of radius r to where r was a special radius, it should be >= a certain number that we had defined yesterday. So if is a mapping which maps a ball of radius r*itself and if G is a contraction map, 1 simple way of finding out whether G is contraction map over u, was to see whether dou G/dou x was strictly <1 or <= theta which is <1 for all x. If the partial derivative of G with respect to x has any induced norm strictly <1 everywhere, then we know that map G is the contraction.

If map G is contraction, then in neighborhood of x not of radius r, we were assured of existence of a solution. We are assured that any sequence starting from any point in this region would converse to the solution. So solution of this problem is x*=G of x*, x* is the solution and if this condition is met everywhere in this ball, then it is sufficient condition to say that any sequence generated by this difference equation will converse to this solution.

Solution is x*=G(x*). Just to draw the parallel, I am writing this just to draw parallel. We had a sufficient condition here that if norm of B is strictly <1, then also this condition holds that xk goes to 0 as k goes to infinity. So we said this is the weaker condition than this necessary and sufficient condition, but this condition helped us to analyse to come up with diagonal dominance and all kinds of other theorems, which were used to analyse iterative schemes.

Likewise, analogous to this, when I come here, this contraction mapping principle tells us very, very important things, 1 is that if G is a contraction map if its local derivative has. If you G (x) to be dx, then local derivative of G with respect to x will be matrix B and any induced norm of matrix B being strictly less than 1 is the condition that we are looking for there, so they coincide. This particular equation, only difference there was the solution, the point where we wanted to reach was 000 origin.

In this case, we want to reach a solution $x^*=G(x^*)$. It is possible to make everything in terms of 000, if you redefine or shift the origin to x^*, then you can make the 2 problems almost equivalent, but that is not important. It is just matter of shifting the origin. What is important is that there is an analogous sufficient condition here for nonlinear difference equations. It does not help us here to look at the spectral radius of this matrix.

It does not help here, the reasons which are difficult to explain as a part of this course, but we have to use only norm and any induced norm. If any induced norm is strictly < 1 in some region, then you are guaranteed that there exists a solution to this difference equation in that region. The solution is unique and the third point, which was very, very important, start from any initial guess you will converse to that solution.

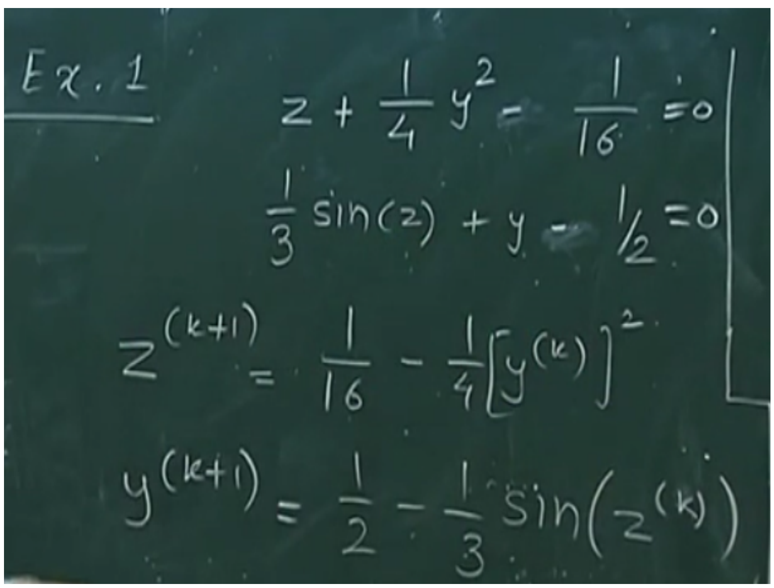
Start from any initial guess in that region, you will converse to the solution $x^*=G(x^*)$. So these are very, very important findings of this particular theorem. In general, it is more difficult to apply this theorem for a complex real problem. Nevertheless, it gives us some insights. For example, you can try and make the sufficient conditions meet by ensuring that dou G/dou x has induced norm <1. You can try to do this.

If there is some problem in solving some nonlinear equations, we can, these are sufficient conditions, remember that. If this conditions are violated even, then the conversions can occur. These are not necessary conditions, but this happens convergence will occur. Just like in this case, when we were talking about linear algebraic equations, if norm of these <1, spectral radius is <1, it is a sufficient condition.

But if norm of B is >1, even then convergence can occur. Because convergence depends upon the spectral radius. Spectral radius can be <1. Similarly, contraction mapping principle gives us a sufficient condition for convergence. It is not a necessary condition. If you meet the sufficient condition, you are guaranteed to converge. So this gives at least some handle to understand how the convergence occurs. From that view point, this is important.

Those of you who are solving large algebraic equations as a part of your research M. Tech. or Ph.D. and hit into problems, you should look at the norm of the Jacobian. I mean at least that much you should remember, look at the norm of the Jacobian. I try to see whether you can make the norm of the Jacobian <1, you have good chances of convergence. Just to illustrate this idea of contraction map, I just give you 1 example here.

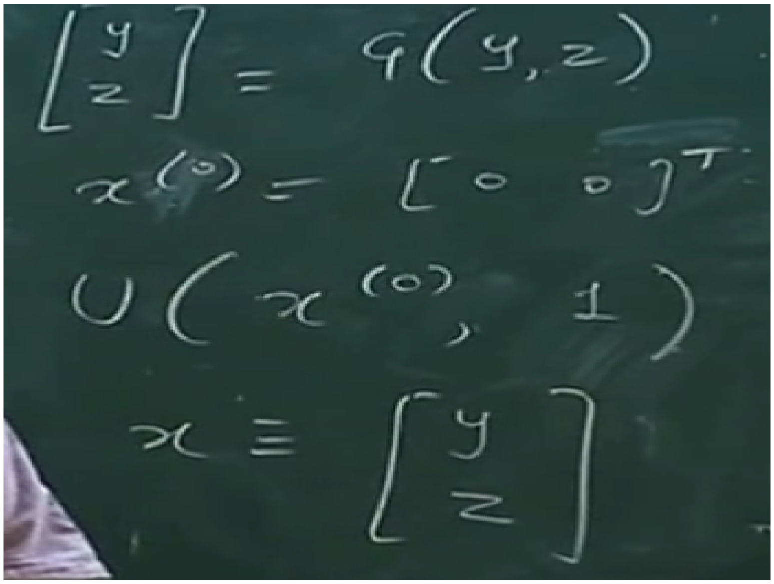
(Refer Slide Time: 13:19)



I want to solve simultaneously, these are 2 nonlinear algebraic equations, which I want to solve simultaneously. If I write this – this=0 and this-this=0, then this is F(x)=0. There are 2 functions, F1zy and F2zy. I want to find out a solution for this particular problem. I am formulating an iteration scheme here, zk+1=1/16-1/4 yk square and I have just formed 1 iteration scheme. This is not the only way to form iteration scheme.

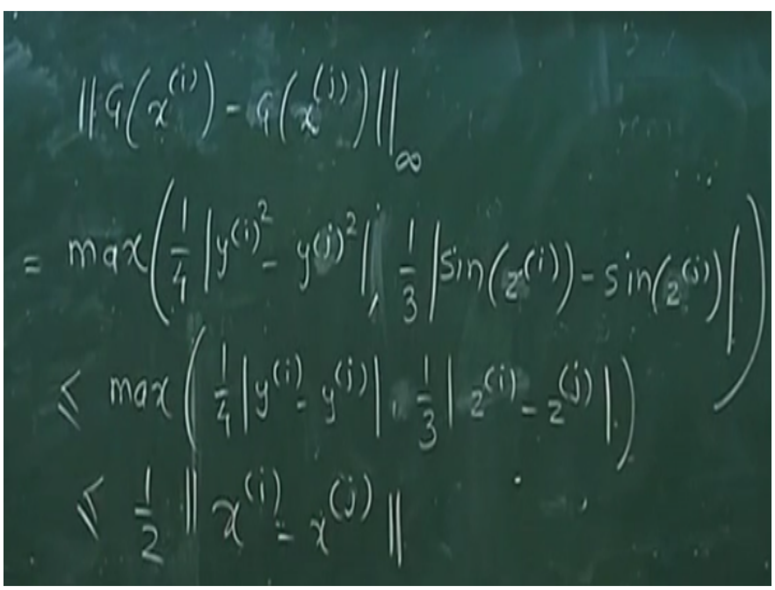
I am showing you 1 possible way of forming the iteration scheme. This is a Jacoby type iteration scheme. What would be the Gauss-Seidel kind of iteration scheme. If I were to use zk+1 here, it will become Gauss-Seidel type iteration scheme. This is the Jacoby type iteration scheme. Now what I am going to do here is.

(Refer Slide Time: 15:44)



I have this scheme which is yz=G(yz), where G I this right hand side function. I am considering this unit ball, let us say x not, my initial guess is 0, 1, no, no. My initial guess is x0=0, 0 and I am considering this unit ball of radius 1 in the neighborhood of 0, 0. So I am looking at.

(Refer Slide Time: 17:18)



Now what is this infinite norm, I am taking some point xi and some point xj, x here is x consist of y and z, x is the vector consisting of 2 elements y and z. Now I am looking at this. What is the infinite norm? Infinite norm is maximum of the absolute value of the elements. What I am doing is, I am taking xi-Gxz, it has 2 elements, I am just taking the maximum of these 2 absolute values will be the norm. I am just using definition of infinite norm, nothing else.

Just this is definition of infinite norm. So you can show that this is <= max of, I am skipping in between steps, you should fill them up. Just go back and look at why this step comes from this. You can prove this in equalities that is this particular difference, infinite norm of this difference is <xi-xj 1/2*xi-xj. Actually the contraction constant is half. I just wanted to show that in this particular case, you can show that.

I am using here the fact that the elements are drawn from the unit ball, so that is why these types have been written and essentially using these inequalities what you can show is that Gi-Gj. Uing these inequalities, you can also do analysis using the derivative of this and taking this infinite norm. You can also do analysis using derivative of this right hand side, Jacobian matrix and infinite norm of the Jacobian matrix, that analysis is also possible.

In this particular case, we have found that, if we apply G on any xi and xj, then this inequality holds. If this inequality holds, what it means is that this constant on the right hand side is <1. So this is strictly <1. So this G map is a contraction. If G map is a contraction, I am guaranteed that there exists a solution in this unit ball. The solution is unique and starting anywhere in this unit ball, this is in reference to the infinite norm.

It will be a square, it will look like a square. We have seen this, how does the unit ball look like in different norms. Starting from any initial guess within this, the iterations will converge to the solution. So this we are guaranteed because we are able to prove this in equality here, for this particular x=G(x). What is important here is that just looking at or just developing this inequality, this is infinite, I am guaranteed that a solution exist in the ball.

I am guaranteed that I start from anywhere and I will reach the solution and this iteration scheme is going to work, that is what I know from this analysis. Just do not bother about these in between steps. Assume that this sequence is true, because our aim is not to do this algebra. You can work on this algebra later. More important is that by doing this algebra, I can show that infinite norm of Gi-Gj/xi-xj. For any i, j, I can prove this.

I take any 2 points in this ball, apply G on both the points, the new points will have a distance which is closer than the original 2 points. That is the main thing. If that happens, we are assured that the solution exists. We are assured that starting from x not, we will reach the solution. Moreover, from any initial guess in this region, if we start, we will still reach the solution. That is the important point. It is difficult to do this analysis for a very large scale nonlinear system.

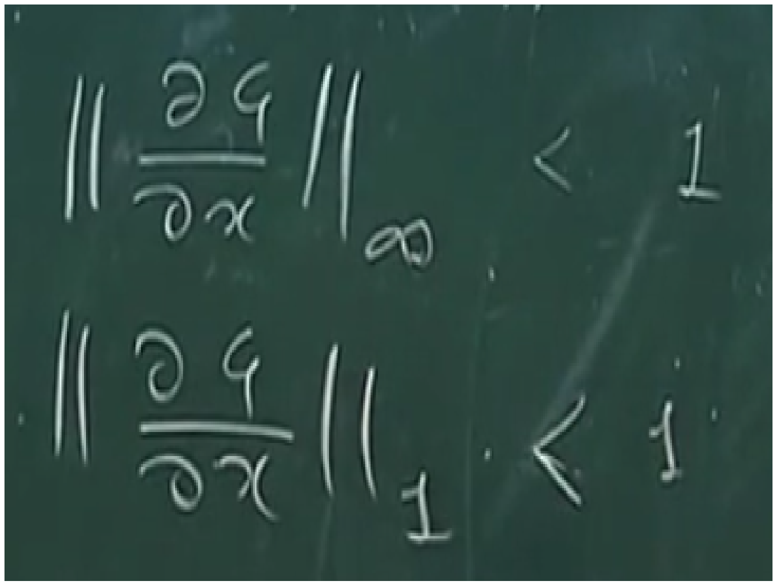
Nevertheless, it is important to get this insight that how does 1 look at analysis of convergence of iterative schemes for solving nonlinear algebraic equations, because most of the times you will be actually dealing with nonlinear algebraic equations, large scale in your computation work, because most of the chemical engineering problems, 99.9% of them are nonlinear problems, reactions of heavy transfer occur and turbulence and always things will make the life very, very complex.

We have to work with a set of nonlinear algebraic equations. What is it that governs the convergence? We can get some clues if you can show that the iteration scheme that you have formed actually is a contraction map, difficult to show in general for large scale system, but this does give you insight, which is very, very important. That is what you should carry. I want to stop here. I do not want to get into too much details.

In the notes, I have given some more detailed discussion on Newton's method. So there are special theorems for convergence of Newton's method and more than the proof and the theorem statement, I have tried to give some qualitative insights as to how to interpret those theorems. I have not included the proof. The proof can be found in any of the text books on nonlinear systems like (()) (26:48), 1 of the very well known textbooks.

So you can find proofs there, but the interpretation is quite important as to how do you make convergence occur. So typically if you have formed iteration scheme, in this case, I worked with, I did not take a derivative, but you could also try to see for this particular system, you can work this out.

(Refer Slide Time: 27:33)

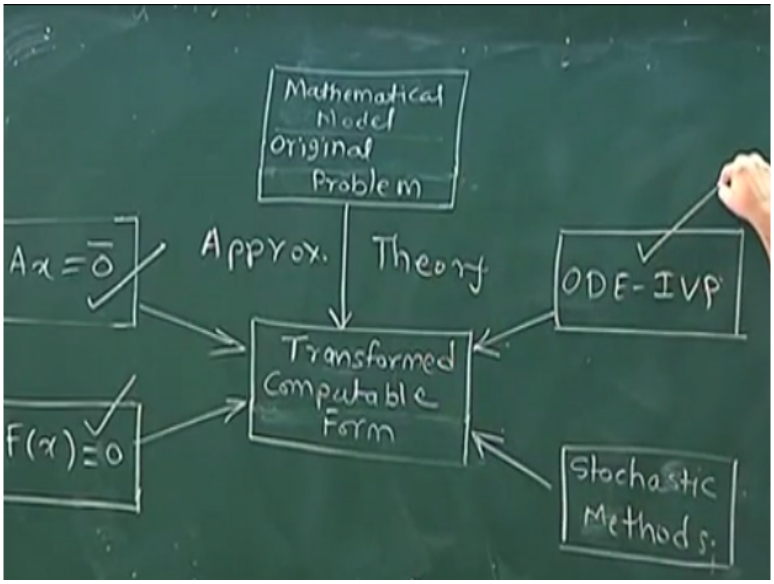


You can try to see whether dou G/dou x infinite norm. if this is strictly <1 in the region where you are trying to operate or trying to solve the problem or dou G/dou x, 1 norm is strictly <1. If these conditions are met, then we are guaranteed that the solution exist and we will reach the solution. These are some, why infinite norm and why 1 norm, because they are easy to compute. Infinite norm and 1 norm are easy to compute.

Other norms like 2 norms will require Eigen value computation. Other than that 1 norm and infinite norm are easy to compute. So you can quickly make a judgment what is going wrong when you are solving the problem. This brings us to an end of methods for solving nonlinear algebraic equations. We have looked at different concepts. We have looked at how to solve them using different algorithms, we just briefly touched upon idea of condition number.

Also we very, very briefly touched upon the idea of convergence of iterative schemes. We have not gone deep into it, but at least you know about what is the tool or what is the machinery that is used for actually looking at this problem. Let us move on to solving ordinary differential equations initial value problems. Now what I want to do next is before I proceed again, we go back to our global diagram.

(Refer Slide Time: 29:38)

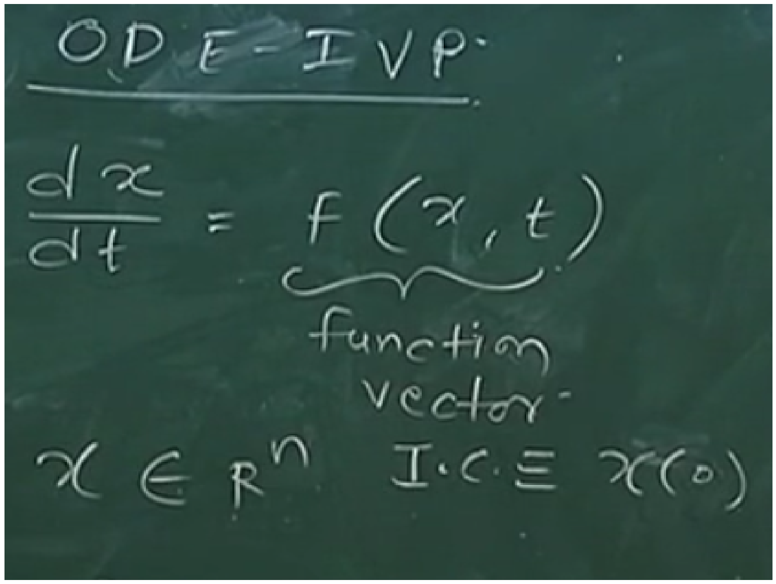


So our global diagram was, so we have this original problem then we use approximation theory to come up with transformed problem. So we have been calling it transformed computable forms and then we said there are 4 tools, 1 is Ax=0, this tool set which we will be using and the other tool set was f(x)=0. So solving nonlinear algebraic equations, solving linear algebraic equations, this is the second tool set that we have.

The third tool set that I am going to look at is OD-IVP because in many cases, the transformed problem is an OD initial value problem. I talk about a method later on how do you transform a boundary value problem into initial value problem. Actually not just one initial value problem, a series of initial value problems, which are then solved iteratively. The fourth tool is stochastic methods, but we are not going to get into this.

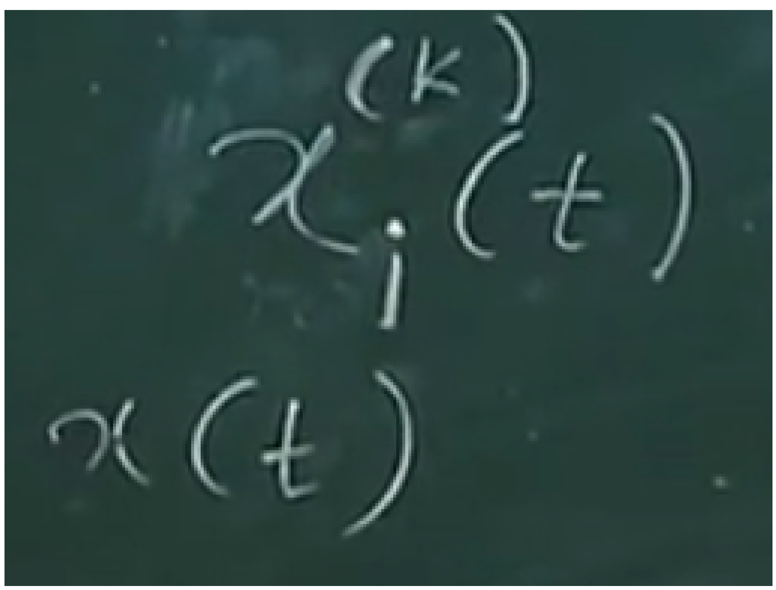
So right now we have done this, how to solve Ax=B, we looked at many, many methods. We looked at many issues that are associated with this. we have looked at F(x)=0 and now we are moving to OD-IVP, all these after all is going to give us approximate solution. This is going to give an approximate solution to the original problem. So moving on to solving ordinary differential equations, initial value problem.

(Refer Slide Time: 32:49)



General form that the types of equation that I am going to look at is of this type, dx/dt=F(x, t) where F is the function vector and x belongs to Rn. What I am given apart from this differential equation model, I am also given initial condition at time=0. Now before I move on, let me explain one notational difference that we will have. In this case, if you are dealing with vectors, we will have to deal with 3 different attached indices with the vector.

(Refer Slide Time: 34:18)



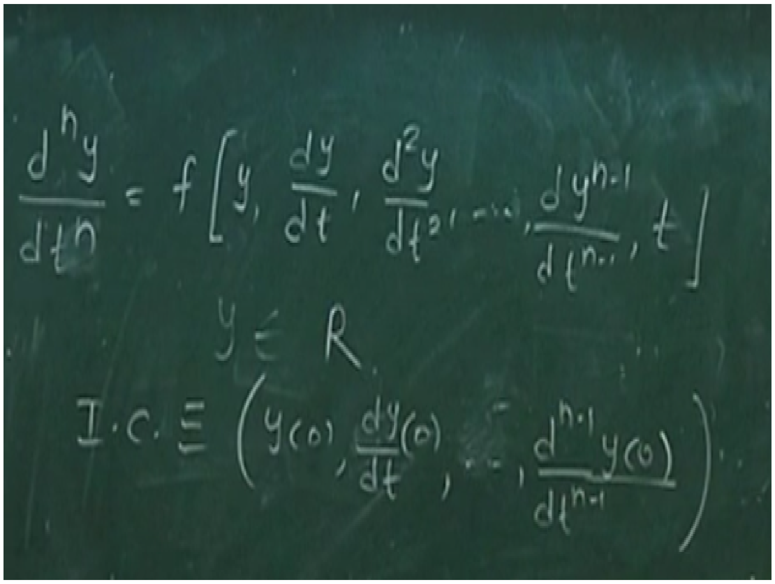
Suppose x is my vector, here i-th element of the vector will be given by xi. This notation we have been using even earlier. Bracket k will indicate k-th iteration. Now additional complexity comes in. We have time, so time will come here. So there are 3 things attached to the vector. In some cases, you will have i-th component of the vector. You will have time t appearing here and you may have k-th iteration.

In some cases, we do not need i and k, we just might work with x, t. x, t means vector x at time t. So now a third dimension comes into picture. Here when you write in the notation. Sometimes, there are schemes which are iterative and you will need index, sometimes you need to prefer to i-th component, so you need xi and t is time. Now what kind of equations I am worried about, what kind of equations I am going to look at.

You might say that well what is written here is only a first order vector matrix equation, $dx/dt=F(x)$. I am writing only a first order equation, only first order derivatives and in your engineering problems, you often come across models which are second order, third order, fourth order and when you did your first course in the differential equations, you had n-th order differential equations and then you had methods of solving n-th order differential equation.

So why am I doing things only for the first order differential equation, though the difference here is the vector differential equation. Earlier we were looking at scalar differential equation. What I am going to show that any n-th order differential equation can be converted into n first order differential equations. So this form which I have written here is very, very generic. So let us begin by looking at this conversion. Let us say you have this.

(Refer Slide Time: 36:54)

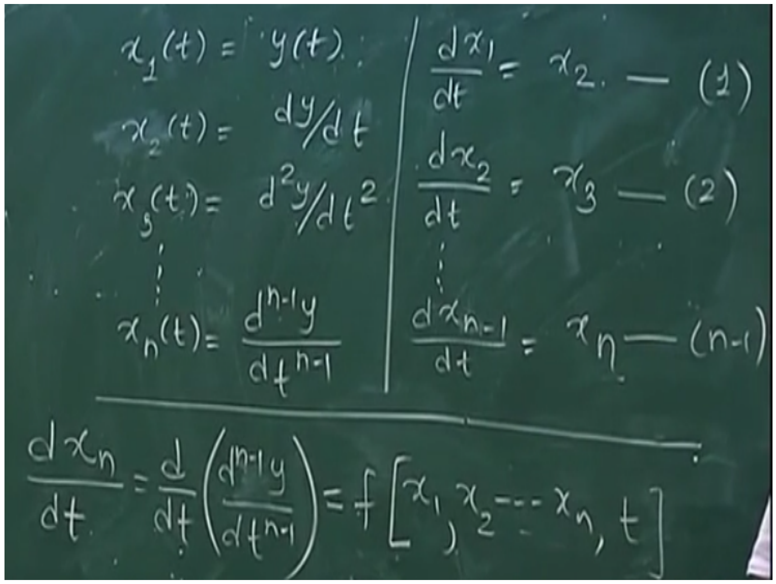


Let us say I have this differential equation in the scalar variable y, so y is a scalar. Y is some mass fraction or some temperature or whatever is the case. You have some differential equation. Let us say this is n-th order differential equation. In general, nonlinear differential equation, we do not know, I am just writing a generic form, could be anything. This is in one variable. An independent variable is time.

What I am going to do now, I am going to define new state variables. So my state variable and what I am given together here to solve this problem, say initial value problem, so what do I need to solve this problem. I need a differential equation and I need the initial conditions, initial conditions are given for y(0) dy/dt at 0. So we are given initial condition, we are given initial y0, initial derivatives up to order n-1. These are required to solve this differential equation.

With this, differential equation together with this initial condition will be initial value problem, solving ordinary differential equation initial value problem, this is what I get. Now what I am going to do now is to start defining a new set of variables.

(Refer Slide Time: 39:35)



My new variable x1t-yt, x2t=dy/dt, x3t=d2y/dt square up to xnt=dn-1/dt n-1-th derivative. I am defining new variable x1 to xn. Now you can see that these variables are related to first order differential equations. I can very easily say that dx1/dt=x2, dx2/dt=x3, so I have such n-1 equations. This is my equation number 1, equation number 2, and this is my equation number n-1. I have n-1 such relationships between the variables.

All of them are first order differential equations. The last 1 is now just the equation that we have. So the last equation n-th equation. This is dxn/dt, this is nothing but d/dt of d n-1y, this is my definition, this is = Fx1, x2, ... xnt. I have an n-th order differential equation, which got converted into n first order differential equations. This is my first equation, second equation, n-1-th equation and the last equation came from the original n-th order differential equation.

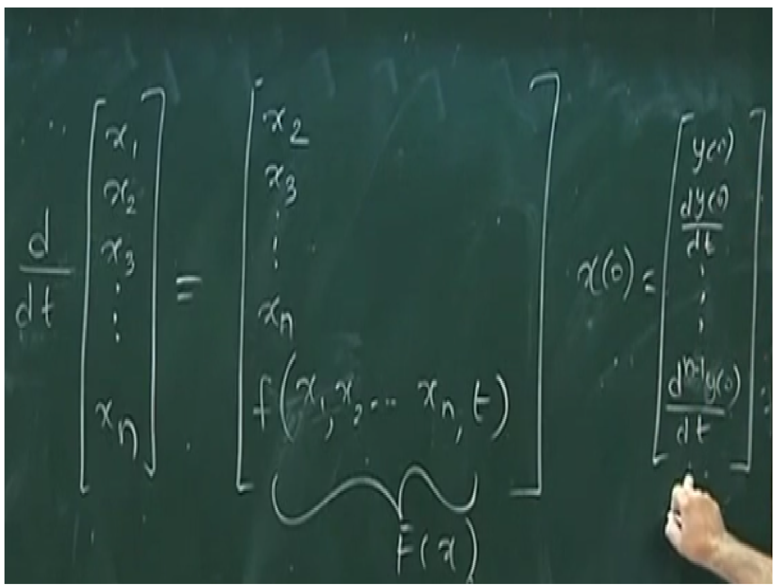
X1, x2, x3, … xn are the new state variables that we have defined. So what I have actually done is a scalar n-th order differential equation. I have converted into n first order differential equations in new variables. So if I have n-th order equation, I can convert it into n first order equations. If I have 2 simultaneous equations, 1 n-th order in 1 variable, other m-th order in other variable. First 1 will give me n first order equations.

Second 1 will give me m first order equations, you can stake them together into a bigger vector, you will still get this form. So this is the very, very generic form. I am not doing any compromise. Any n-th order equation or any set of n-th order equations, n-th, m-th order equations can be combined into finally this form. This is the very, very generic form. So do not worry about why are we looking at only first order vector differential equation.

So all the advanced books on nonlinear differential equations, will worry about this generic form, because anything can be converted to the generic form, that is the first thing to understand. So all the methods that we will develop are for this. If you have n-th order equations, you know how to convert them into n-th first order equations and write it like this. So what will be F(x). In this particular case what will be the F vector.

Let us go back and write that. In this particular case, my F vector after a transformation actually.

(Refer Slide Time: 45:02)



My equations are d/dt of x1, x2, x3, … xn=x2, x3, …xn and F(x1, x2, … xnt). This is my F(x). This is the transform problem. This is my F(x) and I am given the initial condition. So I am given initial condition, x not, which is whatever. This is y0, dy0/dt all these are given to me. This is my x not. This is given to me. This is my F(x). The original equation will appear as 1 scalar nonlinear function in a function vector. This my function vector.

This is a transform problem. I do not have to worry about n-th order equations. I am not going to do separate methods. In the first course of differential equation, you have second order differential equations, 1 chapter on second order differential equations, then you will look at n-th order equations. We are not going to separate. We are just going to look at n differential equations, which are coupled.

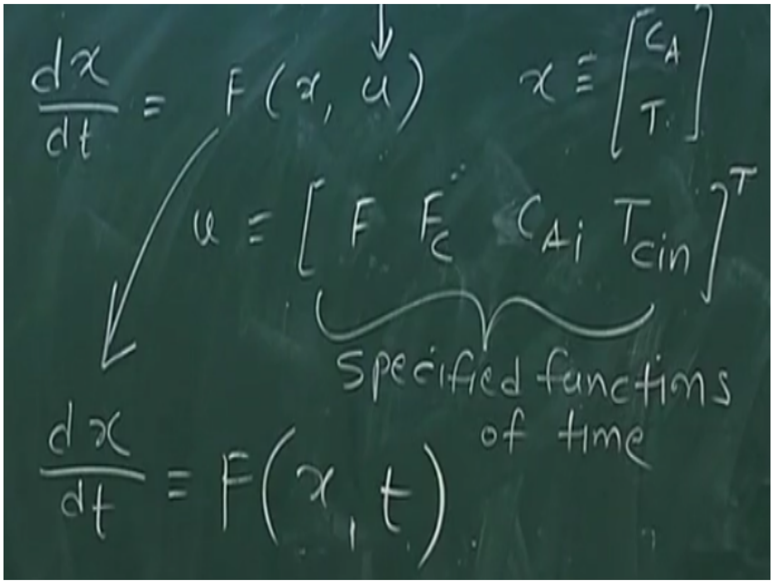
If you are trained to solve dynamic simulation of a chemical plant, there will be 1000s of differential equations which are solved simultaneously together. In fact, they might be differential and algebraic equations, not differential equations. So we are worried about right now to begin with solving large number of differential equations simultaneously together in 1 shot, that is my aim. This form is very generic, applicable to any set.

Other way of getting these kind of equations, we have already seen where do you get these kind of equations. In problem discretization, where did you find them. Finite difference method, orthogonal collocations of partial differential equations that involve time and space. We discretize in space, we got differential equation in time, we got n differential equations. They were first order. If those are all second order, you can convert them into 2 first order equations.

All that is possible, that is not difficult. So converting n-th order equation into first order equations is not a problem. We are going to look at the generic form. This could be arising from any of the sources. This could be arising from the 1 which we have done right now. It could be arising from discretization of a PDE. It might be arising from some other context. We already have studied about in what context this kind of problems will come.

We will look at only how to solve this abstract form of vector differential equation. The other thing which you might worry about is that where does this time t come into picture. Most of the times, the differential equations that you get, an exercise that I have given you to solve differential equations for 1 particular system and I had given you a program, which solves differential equations for a CSTR. I suppose you remember to submit assignment soon.

(Refer Slide Time: 49:06)



That equation is of this form, dx/dt=F(x, u). There are some free variables, x are dependent variables and there are some free variables, like feed flow, coolant flow, coolant temperature, inlet concentration, all these are these u variables. So in that particular problem, CSTR problem, x corresponds to concentration of a, and temperature and u corresponds to inlet flow rate, cooling water flow rate, inlet concentration, cooling water temperature at inlet and so on.

So these are the free variables, but if you go back and look at the problem statement, these manipulated variables or input variables have been defined as a function of time. This is sinusoidal, this is whatever. We have defined these as some functions of time. Once these are given as functions of time, we can substitute them here as some function of time and then once these are specified functions of time, then only we can solve the initial value problem.

For those specified functions of time, this problem has been transformed to dx/dt=F(x, t), because u will be function of only time, some specified function of time, a ramp function, step function, sinusoidal function, or whatever. Whatever you want to study the dynamics of the particular system. You are specified this free inputs and then this becomes a problem, which again is the generic form.

So this parameter or these input variables, we assume that we already know them and then we want to solve the problem for the known inputs, how does the dynamics evolves in time. That is what we want to solve. That is why we are looking at in general dx/dt=F(x, t). How this is specified as a function of time, let us not worry about that right now. It could be an operator who is giving these values, it could be a controller which is finding out these values.

It could be some environmental conditions, which define the cooling water inlet temperature, we do not bother about that right now. We want to solve the problem, when this is specified, how do you actually find out x as a function of time. I want to find out given these input trajectories in time, I want to find out x trajectory, that is concentration trajectory starting from time 0 to whatever final time you want and temperature trajectory as solution of this problem is going to be not 1 vector.

When you are solving nonlinear algebraic equations, you got 1 vector as a solution, the fixed point. Now the solution is going to be a trajectory in time. Trajectory in time over the finite, if we are solving over a finite time or whatever t goes to infinity, if you want to look at. Now linear differential equations of this type, you probably have already looked at in some other course, wherever we need them, we will visit them.

Those of you who have not done the other course on analytical methods in chemical engineering. I will briefly mention those results, which we need here. We are going to look at the problem when this F(x) on the right hand side is nonlinear, not when it is linear. That is very, very crucial. We will use the results for linear later on to get some insights into the convergence properties under what conditions the methods that you have proposed will converge.

That is why we will use some linear system results, but in general what we are going to look at is methods for solving nonlinear ordinary differential equations given initial conditions. How do you get trajectories in time or it could be trajectories in space? We have seen that for example, method of lines for converting Laplace equation, you discretize only in 1 spatial direction, the other 1 is stated as a differential equation, so you get instead of differential equations in time or space.

You want to integrate the differential equations. So t here in general need not be time alone t here is treated as independent variable, in some context it could be space. So maybe I should write a generic form that eta, so eta is some independent variable. It could be time on space depending upon the context and initial condition at eta=0 is given and you want to integrate this set of differential equations.

The way that we are going to proceed will briefly peak into the issue of existence of solution very, very briefly and then move on to the different methods of doing numerical integration. Again, what is going to help us Taylor series approximation and polynomial approximations. We are going to meet our old friends Taylor and Weierstrass again and use them repeatedly to solve these problems.

What I want to stress here is that the same ideas are used again and again to form the solution methods. There are few fundamental ideas which if you understand those ideas and if you know how to apply them you can almost do everything from scratch. Same idea is repeatedly used. If you get this viewpoint, then I think you have learnt a lot. Next class onwards we will begin with how to solve ordinary differential equations and algorithms.

And then finally we will move on to the convergence properties under what conditions, these converge, try to get some insights into relative behavior of different methods and so on.