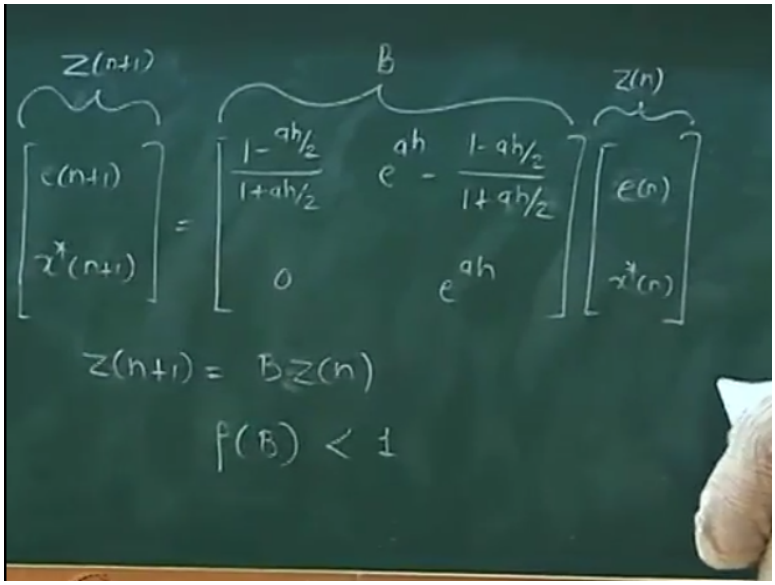
Advanced Numerical Analysis
Prof. Sachin Patwardhan
Department of Chemical Engineering
Indian Institute of Technology - Bombay

Lecture - 46
Solving ODE-IVPs: Convergence Analysis of Solution Schemes (contd.)

So we have been looking at stability of a stability asymptotic stability of integration schemes numerical solvers for ordinary differential equation initial value problem and then we looked at certain cases right. We looked at explicit Euler, implicit Euler, trapezoidal rule okay and we carried out analysis of under what conditions the difference between the true and approximate will vanish as at least asymptotically okay.

(Refer Slide Time: 01:09)

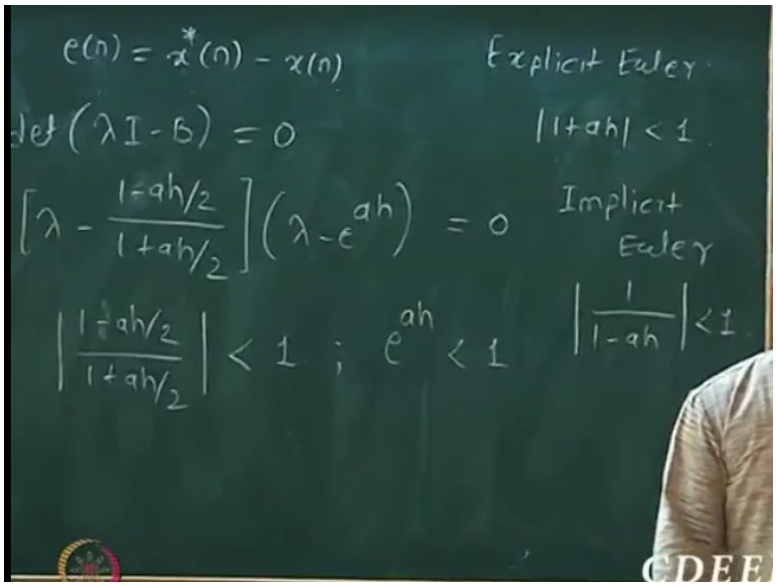


So for example for trapezoidal rule, I considered a simple system dx/dt=ax a<0 and xn is the initial condition okay and then we wrote this difference equation. When we use trapezoidal rule, we use this difference equation and we finally found that the dynamics is governed by en+1 so this is actually Pade approximation of e to the power ah okay and a way pointed out that this is nothing.

But if you call this vector as z n+1, call this matrix as B and call this as zn okay. Then we have this linear difference equation z n+1=B zn. We have looked at equations of this type earlier very, very similar equations except that the context was different. We looked at iterative schemes. There was no time involved, here n represents the time instant, earlier it

was k+1=B*ek and the criteria was you know the spectral radius of B should be strictly<1 okay.

(Refer Slide Time: 04:11)



Now in this particular case, what are the Eigen values of B? Okay if you do this, you have to solve for determinant of lambda I-B=0 okay and with this you will get lambda-there are 2 Eigen values. Spectral radii should be strictly<1 okay. If the spectral radii should be<1 then absolute of each Eigen value also should be<1. Spectral radius is the maximum magnitude Eigen value okay.

So I need 1+1-ah/2/1+ah/2 to be strictly<1 and anyway since a is<0 okay we know that e to the power ah is strictly<1 so this condition is satisfied because a is<1 this has to be satisfied in order that asymptotically error goes to 0. Error between the true and approximate okay. So our en is nothing but x star n, x star n is the true solution and xn is the approximate solution using trapezoidal rule. In trapezoidal rule, it turned out to be this okay.

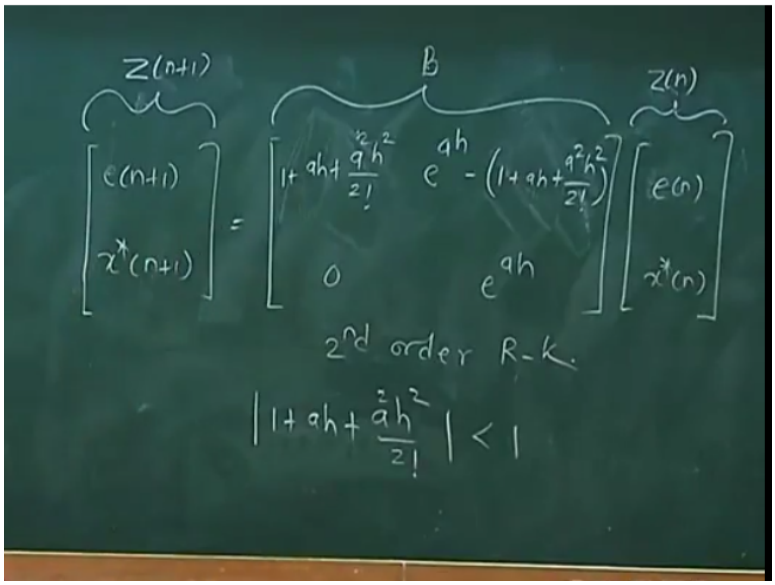
We have derived similar conditions for other cases. For explicit Euler, we got condition 1+ah strictly<1 right and this tells you how to choose h because you are given a problem, so a is given to u. You are asked to solve particular problem, a is given to you, you have to choose h so this tells you how to choose h, h has to satisfy this condition. For implicit Euler, we got condition 1/1-ah strictly<1 okay.

Now the question is how do I take this analysis to a higher level that is to multivariate case? Okay we are still talking about a scalar case. I want to graduate to the multivariate case that is

again for the multivariate case, the situation where I know the solution is when you have linear ordinary differential equations okay given an initial condition. Just now, we looked at those kind of multivariate equations.

So I want to extend this analysis to that case okay. Now we will have some more complications there because you have this A matrix okay and it is a full matrix. Here in this particular case, A is a scalar and life was simple okay. How do you deal with the case where A is a matrix okay? Are these ideas clear first of all for the scalar case? Okay likewise if you go or doing deriving these things, you know let us say if I derive it for the second order Runge-Kutta method what will I get?

(Refer Slide Time: 08:15)

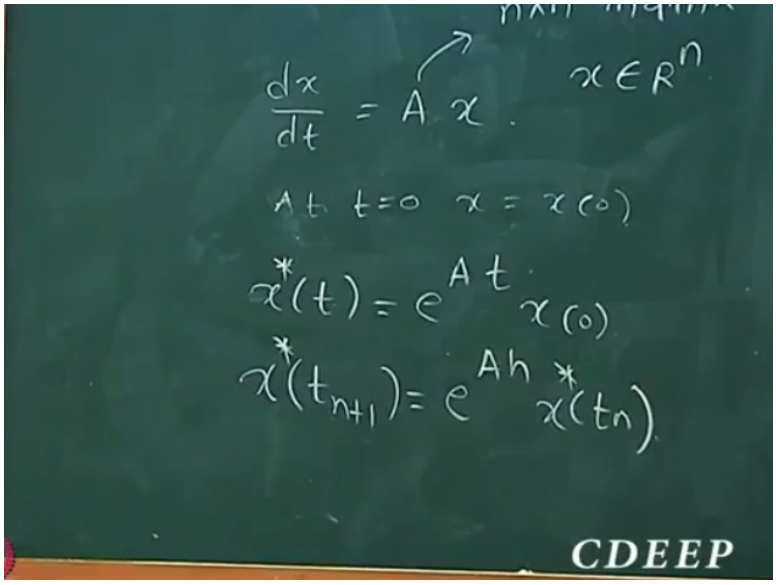


I will just write down that what you get for second order Runge-Kutta method. Second order Runge-Kutta method you will get here 1+ah+a square h square/2 factorial and here you will get 1+ah+a square h square/2 factorial and ah so this will be case for the second order Runge-Kutta methods okay so this will be the matrix equation for the second order Runge-Kutta method.

And the condition will reduce to mod of 1+ah+a square h square/2 factorial should be strictly<1 okay and so on. See can you guess now what will be for the third order Runge-Kutta method? You can see the pattern, first order Runge-Kutta method is explicit Euler, second order Runge-Kutta method is we have looked at Heun's rule and so on so that will give you this.

Third order will give you 1+ah+a square h square/2 factorial+a cube h cube/3 factorial, mod of that should be strictly<1 okay. If those conditions are satisfied, then only you get approximation error asymptotically going to 0 otherwise it will not happen okay. So one can derive for scalar case okay. Is this clear?

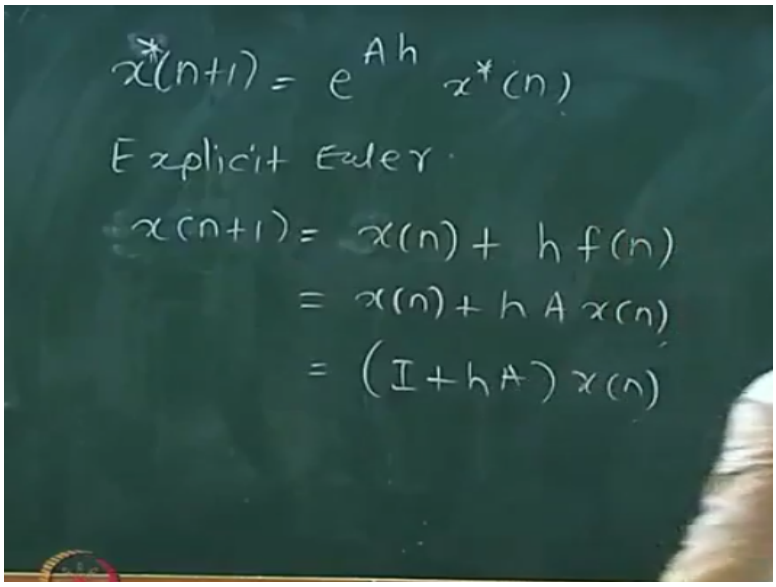
(Refer Slide Time: 10:17)



Okay let us graduate to the vector case. In a vector case, there are 1 or 2 different variants the way we can introduce this. One way is you know let us take a case where we know that for a vector case if xn is the initial condition okay, we know that the true solution xt=e to the power or we can write a general solution starting from 0 let us do that. At t=0 x=x0 this is my initial condition.

Then the true solution is xt=e to the power At where A is the matrix, here A is n cross n matrix and x belongs to Rn, x is n cross 1 vector, A is the n cross n matrix and you have given initial condition which is a vector. The true solution at any time t is written by e to the power At x0 okay. Using the same trick that we did earlier okay for the scalar case, it is not difficult to show that in the new notation xt n+1=e to the power Ah x tn.

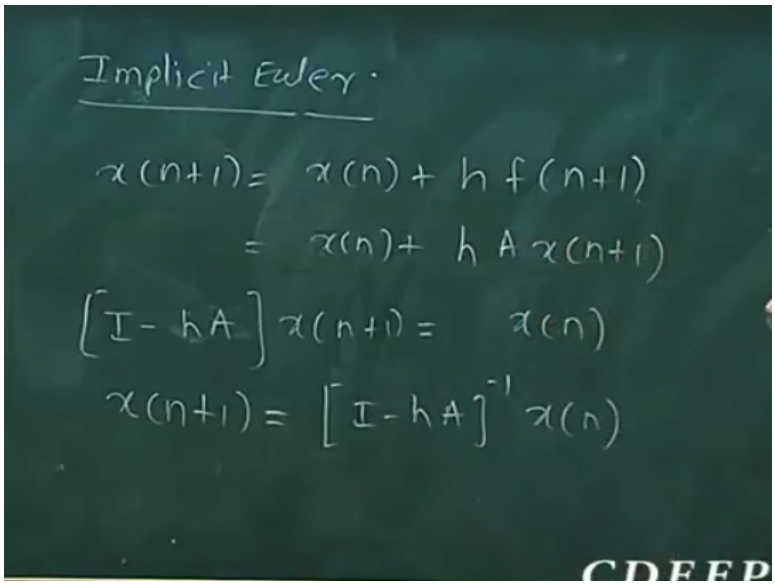
(Refer Slide Time: 12:33)



This is true solution, this is star okay, it is possible to show that or in other words the true solution evolves according to x star n+1=e to the power Ah x star n okay. The true solution evolves according to this. What about the explicit Euler method? Or what about implicit Euler method okay? What about explicit Euler method? Explicit Euler method is xn+1=xn+h fn right, so this is nothing but xn+hA xn right.

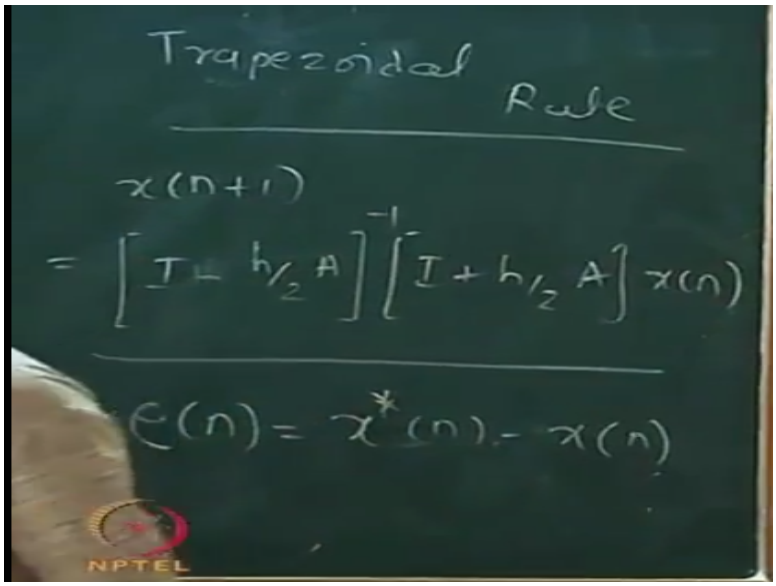
So this is nothing but I+hA xn, now we are not dealing with scalars, we are dealing with vectors and matrices. Pre-multiplication, post multiplication these are very, very important, you cannot take it lightly whether xn should be before or after you should be very, very careful when you write okay. What would be the case for implicit Euler? Can you write down, just try and trapezoidal rule?

(Refer Slide Time: 14:18)



For implicit Euler, what we are going to do is something like this xn+1=xn+hf n+1 so this will give me xn+hA x n+1 okay so if I rearrange I get I-h times A this matrix*x n+1=xn right. I am just taking this on the left hand side okay, what I get here is xn+1 is I-hA inverse x n. Mind you I cannot write divided by I have to write pre-multiplied by inverse that is the only correct way okay.

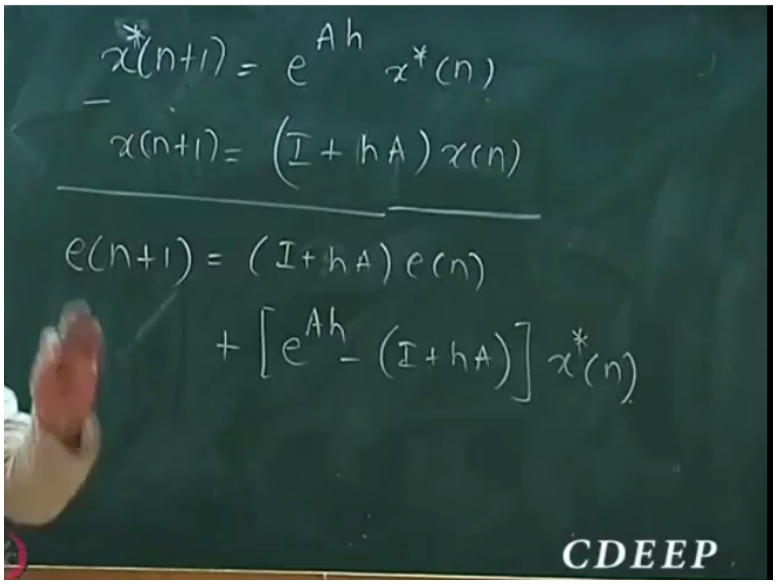
(Refer Slide Time: 15:50)



Similarly, in trapezoidal rule what will it be? Trapezoidal rule will turn out to be xn+1=I-h/2 inverse I+h/2A*xn, just I follow the same thing, for trapezoidal rule I will get this thing here okay. Now the question is how do I do analysis of evolution of error okay? I want to do analysis which is very, very similar to the previous case okay. So I want to analyze en which is x star n-xn.

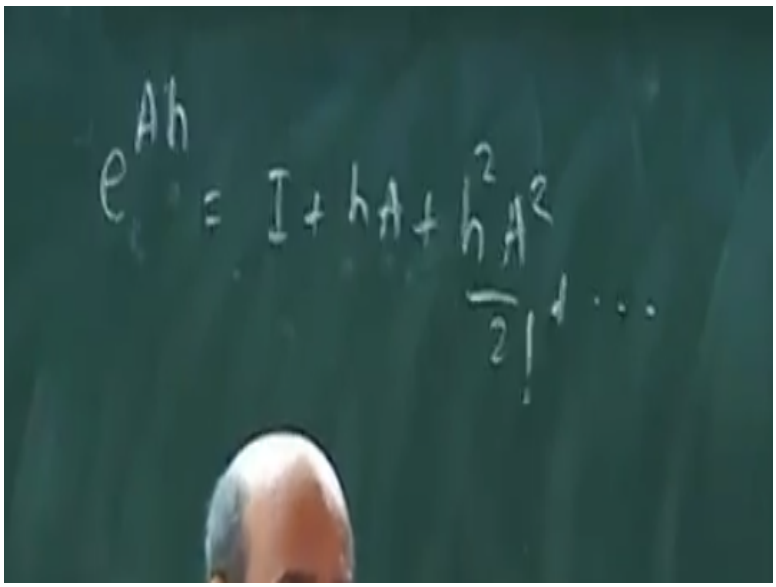
I want to analyze how this error behaves okay. So let us see whether we can get some insights into this. Now we will visit trapezoidal rule and implicit Euler a little later. Let us begin with the simplest one explicit Euler method okay.

(Refer Slide Time: 17:29)



What is my explicit Euler method? xn+1=I+h A*xn okay. Now I want to find out well if I subtract these 2 okay, I will get en+1 okay, I will get en+1=I+hA en+e to the power Ah-I+hA* x star n. If I subtract and rearrange, I will get an equation which looks very, very similar to the scalar case, no difference okay. For the time being well how do you define e to the power Ah?

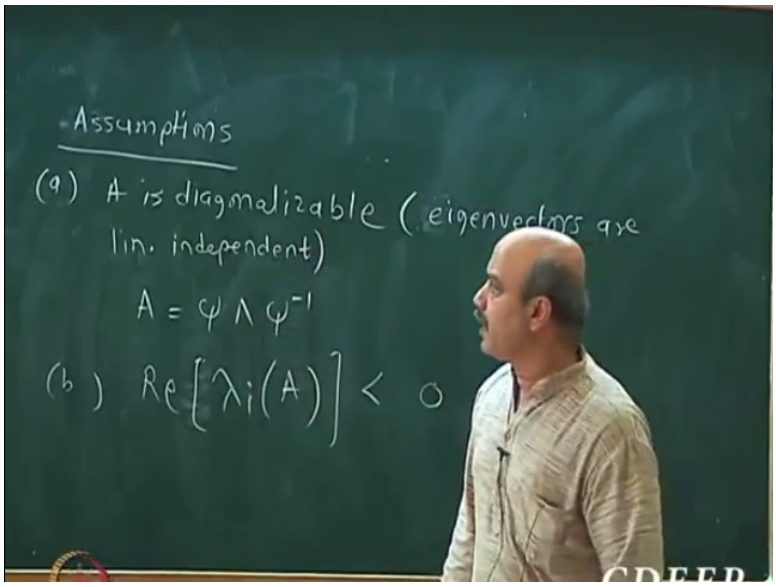
(Refer Slide Time: 18:43)



E to the power Ah is defined as I+hA+h square A square/2 factorial+ and so on right. So if h is very, very small which is the condition for Euler integration right, h is very, very small then you can think that first 2 terms will suffice in terms of approximating e to the power Ah and the later on the terms afterwards can be neglected okay. So for the sake of analysis, let us assume that this difference is very, very small just to get initial insights okay.

We will do the formal analysis little later so formal analysis in the sense by combining the 2 difference equations and the way we wrote in terms of a matrix that will do later. So right now let us assume that this difference is negligible and this is what dominates. Let us look at this part of the equation okay. Now what I am going to do is I am going to use this idea of diagonalization.

(Refer Slide Time: 20:04)



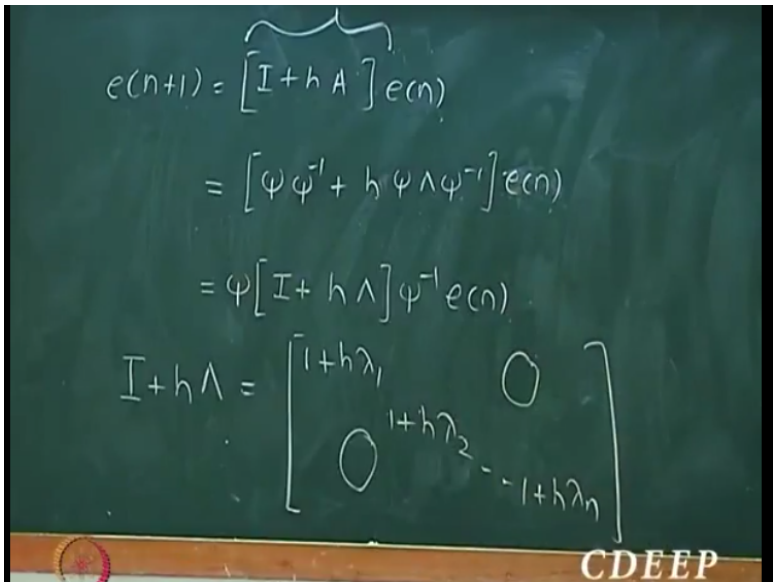
So I am going to make an assumption that to do analysis I am going to do 2 assumptions, one assumption is Eigen vectors are linearly independent. If A is diagonalizable Eigen vectors are linearly independent then I can write A as psi lambda psi inverse where psi is a Eigen vector matrix with columns as Eigen vectors, lambda is a diagonal matrix with all the Eigen values appearing on the diagonal okay lambda 1, lambda 2.

So second assumption I am going to make is that real part of lambda i of A that means strictly<0, due to the analysis I am going to make one more assumption that all the Eigen values of A are in the left half complex plane. Why I do talk about complex numbers? Eigen values need not be real. Given the matrix which has all real entries, I can have Eigen values which are complex okay.

I can have all the Eigen values which are complex. I am talking about the systems, which have all the Eigen values with negative real part okay. Advantage of this is that such systems solution will decay to 0 as t goes to infinity. You can show this very, very easily so I am considering a special class to get insights okay and using this special class of system we are going to get insights into what is happening okay.

How does this help me okay? So let us go back and start looking at our error dynamics. Let us start looking at our error dynamics.

(Refer Slide Time: 22:26)

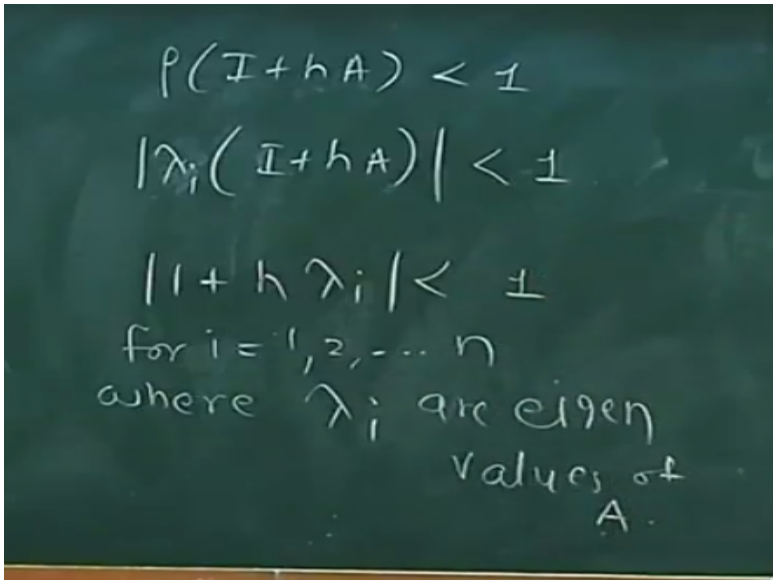


Here you know en+1=I+hA let us look at this part okay. I am going to write this as psi psi inverse, psi psi inverse is I okay+h psi lambda psi inverse en okay which is nothing but psi I+h lambda psi inverse en. Is everyone with me on this? What is psi? psi is a Eigen vector matrix and what is lambda? Lambda is a diagonal matrix with Eigen values appearing on the diagonal okay.

What will be I+h lambda okay? I+h lambda is nothing but 1+h lambda 1 okay 1+h lambda 2 1+h lambda n. This I+h lambda is also a diagonal matrix because lambda is a diagonal matrix, I+h lambda is also a diagonal matrix with 1+h lambda I appearing on the main diagonal, it is fine right. This is appearing on the main diagonal. Now error equation this particular equation is same as something that we have encountered earlier okay.

When will you say that you know error will asymptotically go to 0? What is the condition? Spectral radius of this I+Ah should be strictly<1.

(Refer Slide Time: 25:17)



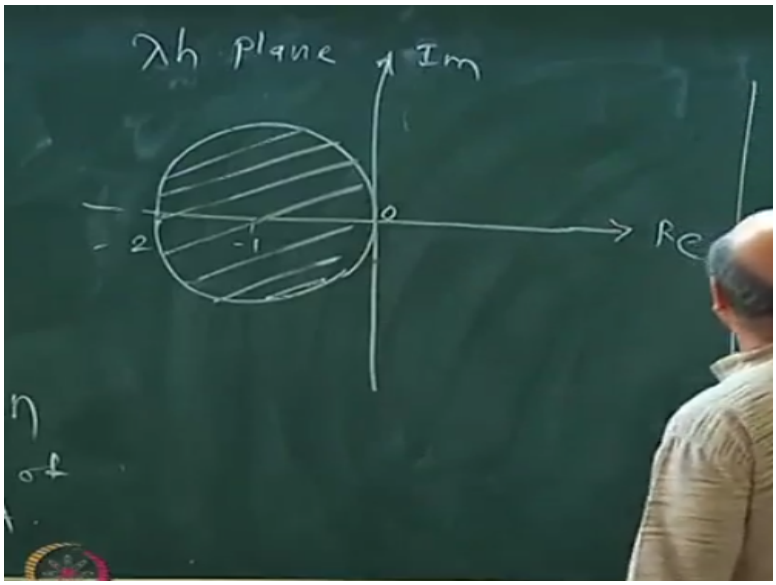
We can translate that condition to you know spectral radius of I+h A should be strictly<1 okay. I can also say that lambda of I+hA lambda i that means each Eigen value of this matrix okay should be strictly<1. Now the question is what are the Eigen values of I+Ah? If you look here this actually is nothing but diagonalization of this matrix. What are the Eigen vectors of I+Ah?

What are the Eigen vectors? They are same as Eigen vectors of A okay. The columns which are appearing here in this matrix are nothing but Eigen vectors of I+Ah. They happen to be same as Eigen vectors of A. We have just proved that okay. This is the diagonal matrix. When you do diagonalization, what appears on the diagonal matrix? Eigen values. So this must be the Eigen values of this matrix okay.

So my condition for stability translates to 1+h lambda i should be strictly<1 for i=1, 2 up to n where lambda i are Eigen values of A okay. So I+h lambda i should be strictly<1 for every i, for every Eigen value this should happen okay. This is the condition under which the error will asymptotically go to 0 okay. We are looking at right now approximate error dynamics; we have neglected one component.

So this is what will happen. This is the condition under which so now the condition that we had earlier is a subset of this condition because earlier we considered A to be a real negative number right. Now I am expanding it and saying that we may get complex numbers and then the stability condition in this particular case, it translates to 1+h lambda i where lambda i are Eigen values of A, this should be strictly<1.

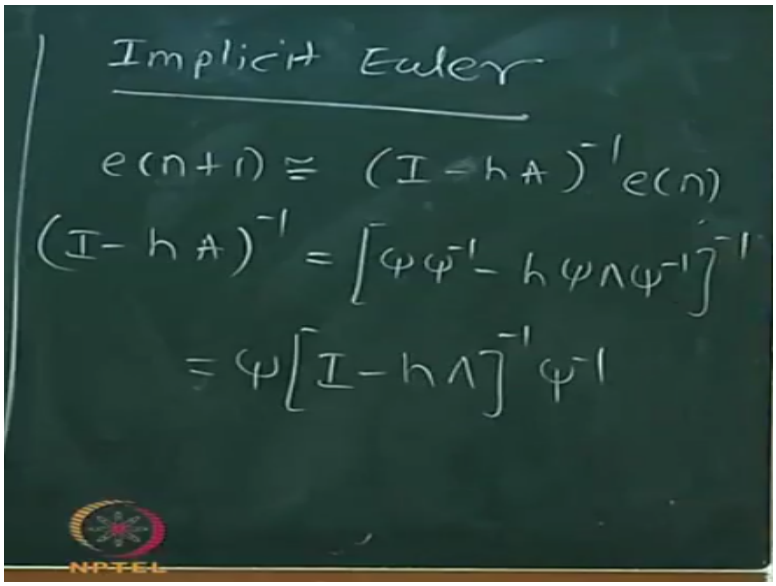
(Refer Slide Time: 28:39)



So actually what is typically done is in the complex plane you draw stability envelope or the region for values for which you get stable dynamics of the error. In this particular case, this will be -2, -1 0 and there will be a circle here. This is lambda h plane, this is imaginary, this is real.

Only in this region okay only in this region if lambda h value lambda i times h value falls in this region okay we can translate this to our, we can draw region in which this condition will be satisfied in the complex plane okay and this is called as stability envelope for explicit Euler method. Likewise, one can derive stability envelops for implicit Euler for you know trapezoidal rule and so on.

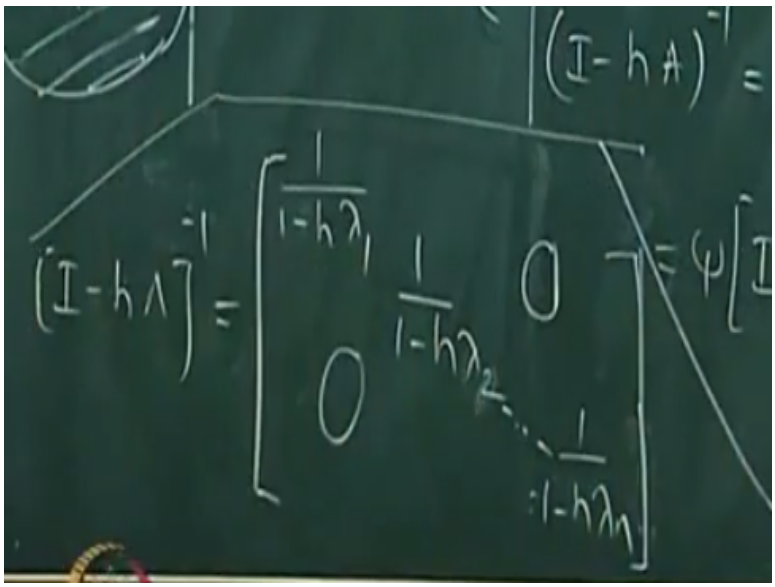
(Refer Slide Time: 30:05)



What will be the case? How will you analyze this case? In the implicit Euler, you have en+1 will turn out to be I-or approximate because we are neglecting one small part, I-hA inverse en okay. How will you write this? What will this turn out to be? See this is I-hA inverse, using the same trick okay that is psi psi inverse-h psi lambda psi inverse inverse okay. I want to take out psi, I want to take out psi inverse.

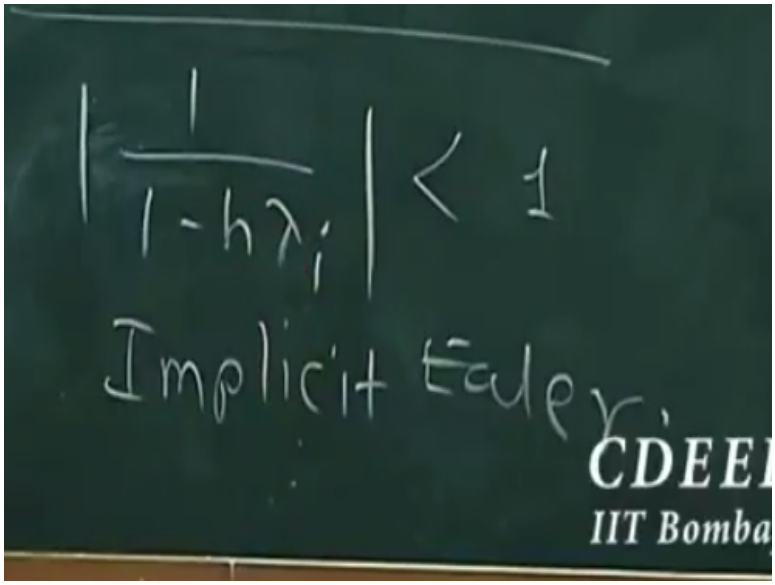
Do you remember the rule for inverse of multiplication of 2 matrices? What is A B whole inverse? B inverse A inverse, so we use rules properly, you will see that this turns out to be psi I-h lambda inverse psi inverse okay. I-h lambda inverse is very easy to compute okay. I-h lambda inverse is very, very easy to compute, it is a diagonal matrix. Inverse of a diagonal matrix is 1/each diagonal element okay.

(Refer Slide Time: 32:00)



So in this particular case, this matrix will turn out to be this matrix here I-h lambda inverse will turn out to be a diagonal matrix okay with 0 0 half diagonal elements and other diagonal element will be 1/1-h lambda 1 1/1-h lambda 2 and so on. What is the stability condition?

(Refer Slide Time: 33:07)



In this particular case, the stability condition for implicit Euler will turn out to be mod of 1/1-h lambda i should be strictly<1 okay and then you can show when Eigen values of A have negative real part okay, the region of stability is nothing but entire left half plane. For any value of h, you will get okay. That is not the case for explicit Euler. Explicit Euler there is a very small region.

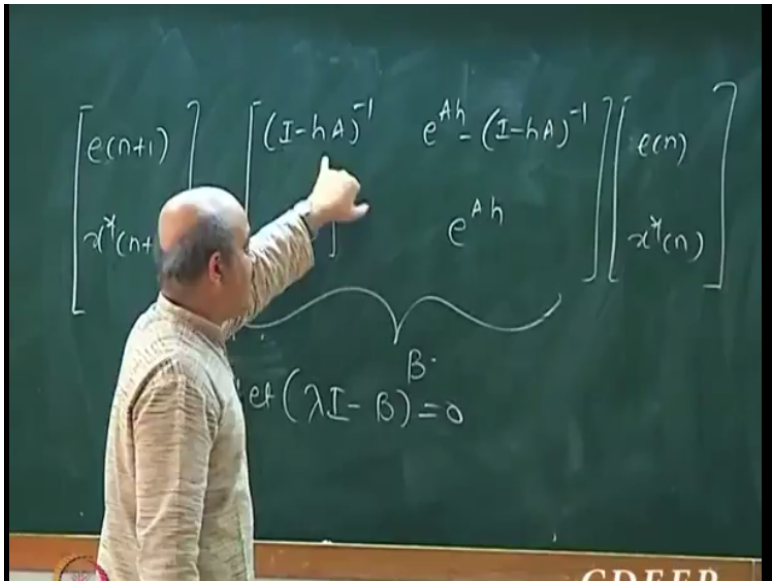
You have to be very, very careful when you choose the integration step size. Implicit Euler okay when if you make slight error or if you choose slightly larger integration or step size, it will not give you asymptotically wrong results okay. This is more to get insights. The insight that we get here is that even for a multivariable case okay if you use explicit Euler method you still have to be very, very careful.

First of all, how do you choose h? It depends upon the Eigen values of A matrix okay. How do you choose h depends upon the Eigen values of A matrix okay. Second thing is and then of course you should choose the one which is most conservative okay because you will get 1+h lambda 1 1+h lambda 2 you will get so many inequalities for i going from 1 to n. The most conservative value of h which you get that is what you should choose okay.

Because it may happen that you might choose h for lambda 1 and that may be a wrong h for lambda 2 okay. So the most conservative one you have to choose. The smallest h that you have to choose which satisfies all the inequalities has to be chosen that is very, very important okay. You do not need to worry so much when it comes to implicit Euler okay. Implicit Euler is relatively much more stable algorithm than explicit Euler okay.

Trapezoidal rule, can you guess if I do a similar manipulation okay. It will be for trapezoidal rule it will be 1+h lambda i/2/1-h lambda i/2, it will turn out to be very, very similar okay for the trapezoidal rule okay.

(Refer Slide Time: 36:11)



Well so I said that the exact analysis would be little more you will have to consider the entire matrix so you can write en+1 x star n+1=I will just directly write for the trapezoidal rule or for implicit Euler. For implicit Euler, it is I-hA inverse e to the power Ah-I-hA inverse. This is null matrix e to the power Ah*en x star n. Well you can show if all Eigen values of A have negative real part, then x star n will asymptotically go to 0.

This is not difficult to show so ultimately it will boil down to Eigen values of this matrix because you have to look at determinant. If you call this matrix as B matrix then you have to look at determinant of lambda I-B roots of determinant of this okay, which will turn out to be nothing but roots of this and roots of this. If Eigen values of A have negative real part, these will always be stable or these will asymptotically go to 0.

You have to worry about this part okay and this part will change depending upon whether you are using implicit Euler or explicit Euler, Runge-Kutta whatever. If you are using Runge-Kutta, you will get I+h+Ah square second order Runge-Kutta will give you that. If you are using third order Runge-Kutta this will change and this will change and so on okay. So how you approximate?

You know e to the power ah will change depending upon the method that you are using but the way of analysis finally remains same okay. We are going to look at. In literature, you can find out this stability envelops for different methods and that way you can compare different methods, which methods are you know where it is easier to choose integration interval, where it is difficult to choose integration interval.

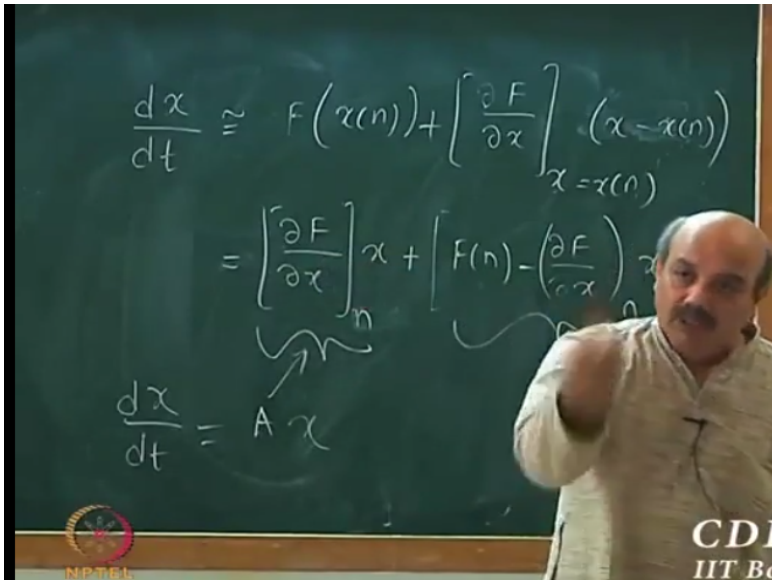
So one can actually compare methods based on these for the last part okay. So this is all fine. Spectral radius of this should be<1 and so on. What about the real problem? The real problems are never linear for linear ordinary differential equations and we are trying to get insights. Why we use linearity? Because for linear ordinary differential equations we know the true solution okay.

The true solution is e to the power Ah or e to the power At. This is not a case for a general nonlinear differential equation so what do I do? Okay I can do a local analysis using Taylor series approximation locally okay. So I can extend this idea to non-linear case by doing repeated local linearization okay. Again the analysis of stability of these methods is fairly involved topic.

And then I cannot do a justice in a course in which I have to pack many, many things so as to prepare you. Idea is to sensitize you that there exists something called stability analysis, stability envelopes and if you really get doubts whether this method is performing well whether I am choosing h step size correctly, go back and look at stability envelops okay.

Those are available in the literature and if you want to have a relative you know comparison of different methods one basis could be looking at these stability envelopes.

(Refer Slide Time: 40:12)



So if I have general non-linear differential equation, dx/dt=F of xt where x belongs to Rn and F is a function vector and F here is a n cross 1 function vector. Then way I could do analysis, I am currently at x tn=xn then what I can do is locally I can write dx/dt is approximately=F at xn+dou F/dou x evaluated at x=xn.

Locally I can differentiate okay and then rewrite this right hand side as dou F/dou x at x=or with the notation that we have used earlier dou F/dou x at x=xn*x+Fn-dou F/dou x n. Since xn is known, fn is known, this matrix is known, this is a constant vector okay. Your equation becomes similar to a linear differential equation. This is dx/dt=this is my A matrix, this is a Jacobian matrix okay and*x.

So this becomes similar to dx/dt=Ax, my A is this matrix local Jacobian okay. I can look at Eigen values of the local Jacobian and do the analysis okay. I can look at Eigen values of the local Jacobian and do the analysis but the problem with this Eigen value of the local Jacobian is that Eigen values will change as you move in time okay. So suppose you happen to choose you know integration step size based on time 0 Jacobian okay.

Jacobian Eigen values change drastically as you advance in time okay. What is giving you stable results in one region may give you unstable results in other region okay. That is where the fix of variable step size comes handy. When you do not know anything use variable step size method okay but if you want to understand, if you want to get insights, you could look at local Jacobian, Eigen values of the local Jacobian.

And if Eigen values of the local Jacobian do not change too much in the region in which you are integrating, you could choose one fixed step size and use it for solving your problem at hand okay. So the way to extend this analysis to the non-linear case is through repeated local linearization okay. So each Jacobian will have different Eigen value actually but typically Jacobian will be a smooth function of Eigen values.

Eigen values will change smoothly but they can change and if they change over a period of time, they change drastically then choosing integration step size becomes tough. In such cases, the fix that we describe right in the beginning variable step size integration is the best okay.

So MATLABS most popular solvers RK45 or RK23 are variable step size solvers okay. When you give some 0 to h internally it will you know keep dividing h till it gets and it will ask you for accuracy. So it keeps finding out h for which you get a desired accuracy and it proceeds okay. So all these things I presented because you should know how to analyze these methods.

How to have a critical look at these methods okay. Practically, when you actually solve problems later on you might use variable step size okay but you should know if you are getting stuck somewhere okay you probably should look at the Eigen values and when I am going to describe something in my next class differential algebraic systems, these Eigen values will play a very, very crucial role okay.

We have something called stiff systems. Stiff systems are one in which some Eigen values are large, some Eigen values are small okay. If some Eigen values are 10 to the power -5 and some Eigen values are 1000, 10,000 okay you know large differential equation this can happen. How do you choose integration step size with respect to 0.0001 or with respect to 10,000 okay?

In such cases, you get what are called as differential algebraic systems. You approximate certain derivatives to be 0 and you only solve for certain other derivatives and so on. So will look at differential algebraic systems but you have notion of stiff systems. Stiff systems arise from Eigen values of Jacobian and we look at the ratio of the real part of Eigen values. The ratio of you know real part of Eigen values of Jacobian maximum and minimum values that gives us what is called as stiffness ratio.

If a system is stiff, you have to use a stiff differential equation solver. MATLAB will give you stiff solver. It is called ODE 15s, s stands for stiff solver okay. So if you have this Eigen value you know disparity you have to go for stiff solvers and you should know about this business of stiff solvers and so basically this is touching tip of the iceberg. There is lot more to it but you can look at the references I have given and try to understand more about this.