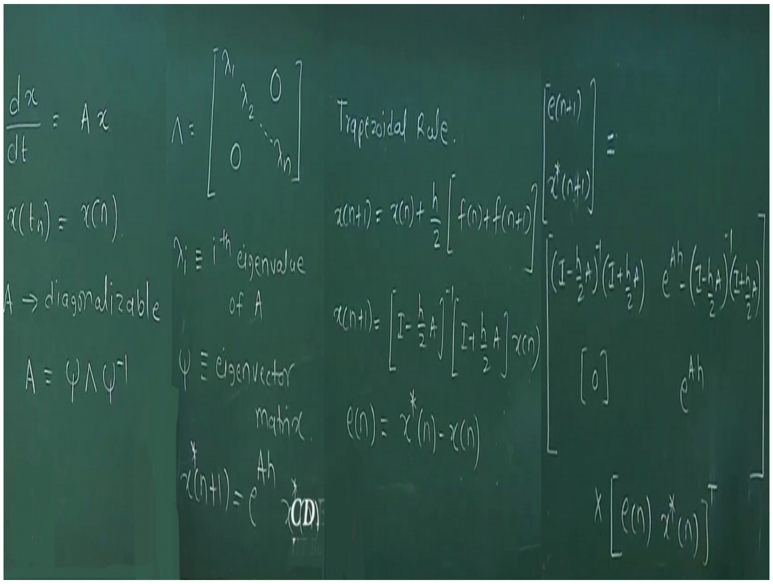
Advanced Numerical Analysis
Prof. Sachin Patwardhan
Department of Chemical Engineering
Indian Institute of Technology - Bombay

Lecture - 47
Solving ODE-IVPs: Convergence Analysis of Solution Schemes (contd.) and Solving ODE-BVP using Single Shooting Method

So we have been looking at convergence behaviour of numerical schemes for solving ODE initial value problems, and we graduated from scalar case to the vector case. We have looked at now general linear differential equations, multi-dimensional linear differential equations, and analyzed stability of some of the well-known integration methods, particularly we looked at explicit Euler, implicit Euler.

(Refer Slide Time: 01:04)


$\frac{dx}{dt} = Ax$
$x(t_n) = x(n)$
$A \rightarrow$ diagonalizable
$A = \psi \Lambda \psi^{-1}$
$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \ddots \lambda_n \end{bmatrix}$
$\lambda_i \equiv i^{th}$ eigenvalue of A
$\psi \equiv$ eigenvector matrix.
$x^*(n+1) = e^{Ah} x^*(n)$
Trapezoidal Rule.
$x(n+1) = x(n) + \frac{h}{2}[f(n)+f(n+1)]$
$x(n+1) = [I - \frac{h}{2}A]^{-1}[I + \frac{h}{2}A]x(n)$
$e(n) = x^*(n) - x(n)$
$\begin{bmatrix} e(n+1) \\ x^*(n+1) \end{bmatrix} = \begin{bmatrix} (I-\frac{h}{2}A)^{-1}(I+\frac{h}{2}A) & e^{Ah} - (I-\frac{h}{2}A)^{-1}(I+\frac{h}{2}A) \\ [0] & e^{Ah} \end{bmatrix} \times [e(n)\ x^*(n)]^T$

And then, so the way to go about of course is to use at least to get an insight is to use diagonalization, and we have considered a special case where dx/dt =A x diagonalizable matrix, and A can be written as psi lambda psi inverse, and lambda is a matrix which has lambda 1 lambda 2 lambda n, where lambda i respond to ith eigenvalues of A, and psi corresponds to the eigenvalue vector matrix. So that is columns are eigenvectors of A okay.

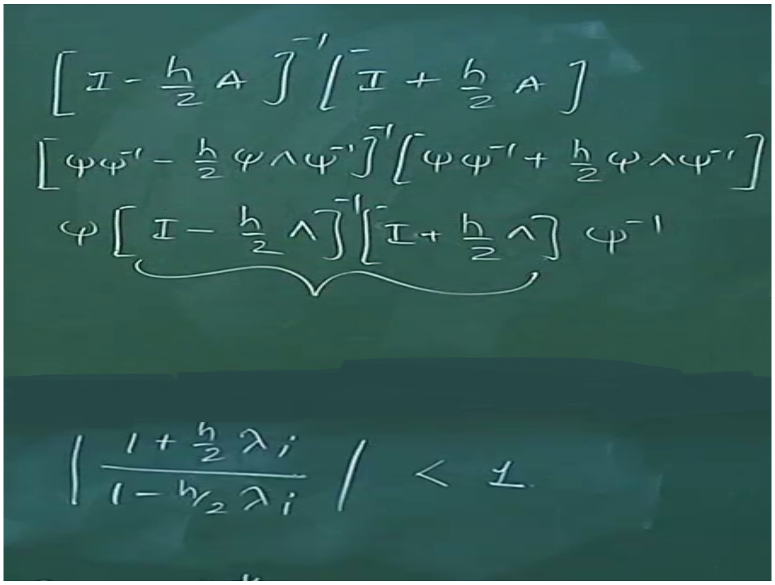
And then using this diagonalizable matrices, we actually got some insight into how different methods behave. See for example if I write down the equation for a trapezoidal rule, then trapezoidal rule is x n+1 =x n +h/2 f n+1, and for linear multivariable systems it turns out to be this difference equation. And then we define e n that is difference between x star n-x n, where x star n is the true solution, let me just move back and the state here the true solution.

The true solution is x n+1 star =e to the power A h x star n this is the true solution, and then we want to find out a difference between the true and the approximate, the approximate solution for my trapezoidal rule is given by this particular matrix difference equation. And then if we want to look at e n+1 x star n+ 1 then you get this matrix difference equation, this particular matrix*this vector okay, so this is the difference equation which we should evolve.

We have one more assumption, we had made two assumptions. A is diagonalizable, the other assumption we had made was real part of all eigenvalues of A is strictly <0, so we have assumed that real part of lambda i is strictly <0, then x star goes to 0 as n goes to infinity okay. So we are looking at a stable multivariable system all eigenvalues are on the left of plane okay, and because of that x star goes to 0, our focus is in understanding how this particular equation behaves this part of the equation that is this matrix.

Eigenvalues of this particular matrix block matrix, this is a block matrix, this is the matrix, this is a matrix, this is a matrix, this is null matrix. So this is a block matrix, and what we want is all eigenvalues of this matrix should be inside unit circle, in the order that the approximation error asymptotically goes to 0 okay.

(Refer Slide Time: 07:35)



$$\left[I - \frac{h}{2}A\right]^{-1}\left[I + \frac{h}{2}A\right]$$

$$\left[\varphi\varphi^{-1} - \frac{h}{2}\varphi\Lambda\varphi^{-1}\right]^{-1}\left[\varphi\varphi^{-1} + \frac{h}{2}\varphi\Lambda\varphi^{-1}\right]$$

$$\varphi\underbrace{\left[I - \frac{h}{2}\Lambda\right]^{-1}\left[I + \frac{h}{2}\Lambda\right]}\varphi^{-1}$$

$$\left|\frac{1 + \frac{h}{2}\lambda_i}{1 - h/_2\,\lambda_i}\right| < 1$$

So for this to happen, we have to do this jugglery, we have to look at this matrix I-h/2 A inverse I +h/2 okay, and then I am going to use the same trick psi psi inverse h/2 psi lambda psi inverse inverse psi psi inverse +h/2 psi okay, this can be written as so psi =I-h/2 lambda inverse i+h/2 lambda into psi inverse okay, this is a diagonal matrix because lambda is a diagonal matrix, I is a diagonal matrix, I+h/2 lambda is a diagonal matrix, and this is a diagonal matrix.

So with you this this particular matrix any elements of this matrix, the diagonal element of this matrix will be of the form 1+h/2 lambda i/1-h/2 lambda i that will be the element of this matrix diagonal element of this matrix, and then we want this to be strictly <1 for all I okay. So that is the stability characteristic that is the convergence characteristics, we call this method to be asymptotically stable or a stable provided all the eigenvalues are, these eigenvalues of this matrix I-h/2 A inverse into i+h/2 A.

So all these eigenvalues should be inside within circle, we can show that if real part of each eigenvalue is negative then this condition is satisfied that is all the eigenvalue are always inside unit circle okay. So trapezoidal rule will or implicit Euler or a stable okay you will get solutions which are converging to the true solutions, so these methods are better than explicit Euler or similar some other explicit methods let us say second order Runge-kutta method.

So the stability characteristics of these, so one can actually draw what are called as stability envelopes of different methods, and you can see that the stability depends upon 2 things. One is eigenvalues, and another is h, how do you choose h okay. And the way you choose h is you apply this condition such h condition for all eigenvalues and choose most conservative h smallest h, what I meant most conservative is h smallest h okay that satisfies.

So there are certain general conclusions that have been drawn which compared different methods, so see there are plethora of methods, there are so many methods. So which one do you use, so as I said you will develop preferences when you start actually using them, some of you might use gear's method, some of you might use Runge-kutta method, and we know how to make it work, but to make it work you should know all this theory, otherwise it is hard to make it work.

So for example implicit Euler and Crank Nicolson method are asymptotically stable or a stable, but a higher order methods have predictor-corrector predictor methods have restricted regions of stability, so Crank Nicolson and implicit Euler are nice methods in terms of stability. But as you move onto higher order okay predictor-corrector method the region of stability shrinks, accuracy improves okay, a higher order predictor-corrector method we will have better accuracy, but a small smaller region of.

So when you move to the higher order or multi-step method higher order in terms of polynomial approximation multi-step method, the stability region actually shrinks, so accuracy improves stability shrinks. So one has to choose h very, very carefully if you want better accuracy okay, so there is a trade-off and you have to understand this when you. Then things like explicit Runge-kutta methods have better region of stability or larger region of stability than explicit Euler.

So in some suspense especially Euler you can say is you know very vulnerable to mistakes, you make small mistakes in choice of h explicit Euler can give you. But then if you see engineering literature there are some people will still use explicit Euler okay, they will choose integration steps as very, very small and make it work, so it is not that you cannot make explicit Euler work okay, you have to understand how to make it work.

So with Runge-kutta you might be able to take larger steps, with explicit Euler you probably have to take many, many small steps, now and they are situations where you have to use explicit Euler okay. For example, right now one of my student in systems and control PhD student is working on, we are working on some algorithms for control of a motor okay, now we have to do online integration okay in milliseconds okay, because the motor is fast right.

And you want to do some mathematical model calculation online in the microprocessor in fractions of a second, I cannot do it do iterative calculations, I cannot afford to iterative equations, I have to do very, very fast in computing. We still do not have computers or microprocessors even with so much this thing, which can you know do implicit method implementation in fraction of a second okay, because there are floating point calculations.

So the simplest method that you can implement is explicit Euler, and if you chose integration step size very, very small it works okay, we want to use mind you I am trying to use a model for a motor which is a set of 5 differential equations, I want to solve them online on a chip okay, and this is where I have to use explicit Euler because that is simplest in terms of computation okay. So I have to make it work rather than trying to go for a more complex method.

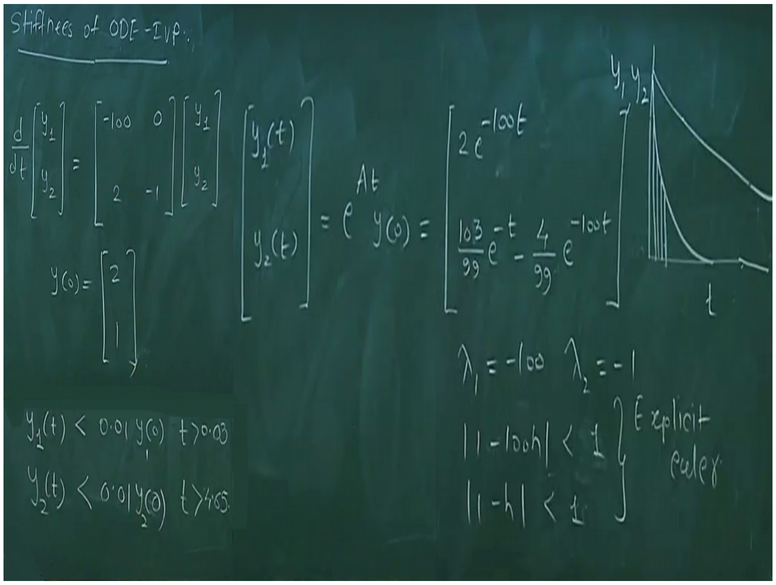
Where you know convergence see convergence may occur if you are using iterative calculations implicit methods, for some steps convergence may occur but for some one particular step suppose it gets stuck, what you do okay, so I am trying to control my motor using a model and then if suddenly my calculations get stuck I have a problem. So I have to use the method which is guaranteed to give me the next value.

So which method you use when is something which is you know there is no formula for that you have to develop expertise, and sometimes you have to use explicit Euler, sometimes you have to use predictor-corrector, sometimes you have to use Runge-kutta depending upon, sometimes you have to use collocation. So it depends upon the situation, upon the problem at hand, upon the solvers that you have, kind of computing power that you have, limitations that you have.

So it is a function of many things, there is no 1 prescription, so do not think that after doing all this well I will never use explicit Euler, you will use explicit Euler in some situations okay, and you should know how to make it work. Now there is one concept which I want to state here, before I do one more small application of ODE-IVP differential algebraic solvers probably I will give a introduction in the next class, if there is a time we will start today.

So as I said there is much more to the stability analysis, and you should be aware of these aspects, when you compare different methods that they are very, very useful for comparing different methods of solving initial value problems okay. One last thing which will connect to DAE solver is the concept of stiffness of ODE okay.

(Refer Slide Time: 17:35)



Stiffnees of ODE-IVP:

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -100 & 0 \\ 2 & -1 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$y(0) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = e^{At}\, y(0) = \begin{bmatrix} 2e^{-100t} \\ \frac{103}{99}e^{-t} - \frac{4}{99}e^{-100t} \end{bmatrix}$$

$\lambda_1 = -100 \quad \lambda_2 = -1$

$$\left.\begin{array}{l} |1-100h| < 1 \\ |1-h| < 1 \end{array}\right\} \text{Explicit euler}$$

$y_1(t) < 0.01\, y_1(0) \quad t > 0.03$

$y_2(t) < 0.01\, y_2(0) \quad t > 4.65$

Now stiffness is a concept that is related to the eigenvalues of local Jacobian if it is a non-linear system, or if it is a linear system it is relate relative eigenvalues that is the largest eigenvalue and the smallest eigenvalue okay. I will just talk about this through an example, I am going to use this particular example d/dt y1 y2 simple problem you know dy1/dt and dy2/dt =given by this matrix differential equation two variables okay, now this problem can be solved analytically.

So the analytical solution of this particular problem is y1 t, well we have seen how to compute analytical solutions it is e to the power A t*y0 okay, so this will be e to the power A t y0 which is nothing but if you do this calculations of e to the power A t, it will be 2 e to the power-100 t, and now this is 103/99 e to the power-t, and so this is the solution, this is y1 t and this is y2 t okay.

Now suppose I want to solve this problem using some numerical integration okay, what will it depend upon, the choice of h will depend upon, what are lambda 1 lambda 2? What are eigenvalues here? The eigenvalues here lambda 1=-100 and lambda 2=-1 right. Now suppose I am using explicit Euler okay, then you know the condition right 1-100 h<0 strictly inside unit circle are strictly <1-h right, there are two conditions one is this condition, one is this condition.

If I were using explicit Euler, this is for making sure that our explicit Euler does not blow up okay, but if you want accurate solution just look at the relative time scales, what is happening here is that y1 t is decaying 100 time faster than y2 t, because y2 t has this +this, this term is going to decay very, very slowly, e to the power-t decays much, much slower than e to the power-100 t okay.

So if you want to capture the behaviour of e to the power-100 t you should choose integration step size to be very small, if you do not choose very small okay, you will miss out on these dynamics okay. But this if you choose it based on -100 t okay, you have to do too many steps of integration okay, so your computation increases okay but nevertheless, you are forcing to look at these eigenvalues when you choose integration step size.

Otherwise, you will miss out what is happening for y1 okay, you will be able to only capture y2 correctly okay, so if you look at the way this is I mean if you plot y1 and y2 versus t, one will decay like this, and other well decay like this okay. So if you want to capture this, you better choose small step size okay, but you know this one is very, very slow, and unfortunately these kind of effects are very, very common in chemical engineering systems, pressure dynamics.

I mean this is a representative hypothetical 2 differential equations, look at real distillation column, in the distillation column dynamics of temperature on a tray or dynamics of composition liquid composition okay is very, very slow compared to dynamics of pressure and vapor compositions okay. Dynamics of pressure changes very fast okay across the column, but the liquid composition on the tray changes very, very slowly.

If you want to write a differential equation that governs the pressure dynamics okay, we normally make an assumption that pressure is constant when you do design, actually pressure is not constant that is simplifying assumption, pressure is varying inside a discussion column otherwise they will not be a flow right, the vapor has to flow from bottom to top so there is a pressure gradient. Pressure transients are extremely fast; concentration transiting are extremely.

I mean on the relative scale concentration transients are something like e to the power-t, pressure transients are e to the power-100 t very fast. If you want to capture pressure transiting you have to choose integration step size to be very small okay maybe 0.1 second, but concentration change on a tray might take you know 50 minutes okay. So one is very slow one is very fast, and then if you insist that you want a dynamics of pressure to be captured.

You have to choose integration step size with reference to the transient of the pressure and not transient of the concentration okay, such systems are called as stiff systems okay. So the way to mathematically quantify stiffness is through eigenvalues or local eigenvalues, you linearize, find the Jacobian, find the local eigenvalues, and you define something called as stiffness ratio okay. so here is just quantification of the time scales, before I moved to stiffness ratio you know.

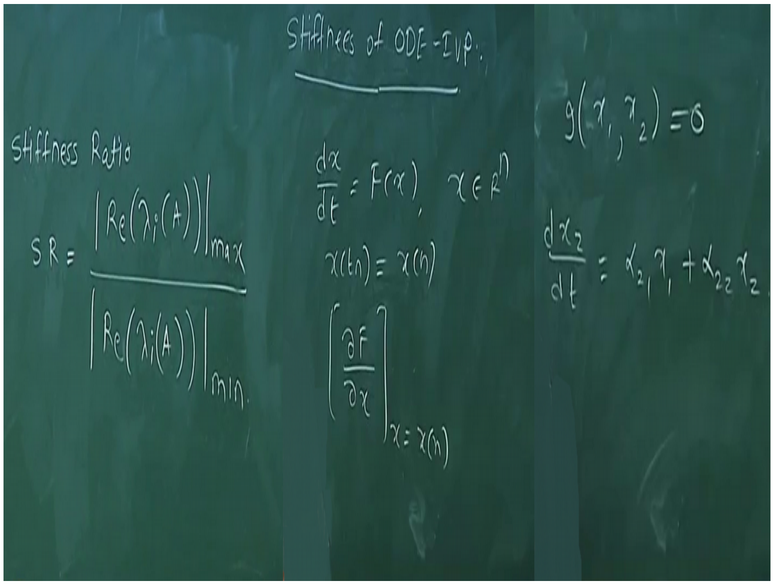
For this particular system y1 t will become <0.01 times y 0, when t is >0.03 in a very short time okay, y1 t will reduce 100 of its original value okay, whereas same thing to happen for okay. So y1 t comes to 100 of its original value in just 0.03 seconds or 0.03 hours whatever time unit you want to take, whereas y2 t comes to 100 of its value in 4.65 hours or 4.65 units minutes or seconds or whatever you want to take okay.

So you can see one is a fast mode, and one is a slow mode such systems where some modes are very fast some modes are very slow are called as strip systems, and then you have to use what are called as stiff solvers okay. Well one now in non-linear systems there is one more difficulty, the stiffness eigenvalues can change because it is local Jacobian okay, and eigenvalues can change as you move in the state space okay.

So in some region out state space system can be highly stiff, in some region it can be less stiff, so well best method is to use variable step size integration, if you can afford to do it. I gave you an example of electrical motor where I cannot afford to use a variable stuff size you know my calculations will go berserk, if I use variable step size. So if you can use variable step size if you do not know, if you have luxury of using variable step size.

And if you do not know anything about stiffness that will solve your problem to a large extent, but sometimes you are forced to use fixed size, and then you have to worry about the relative timescales.

(Refer Slide Time: 28:34)



Stiffness of ODE-IVP:

Stiffness Ratio

$$SR = \frac{|Re(\lambda_i(A))|_{max}}{|Re(\lambda_i(A))|_{min}}$$

$$\frac{dx}{dt} = F(x), \quad x \in R^n$$

$$x(t_n) = x(n)$$

$$\left[\frac{\partial F}{\partial x}\right]_{x = x(n)}$$

$$g(x_1, x_2) = 0$$

$$\frac{dx_2}{dt} = \alpha_{21} x_1 + \alpha_{22} x_2$$

So how do you quantify this okay? So we call this as stiffness ratio, and stiffness ratio is defined as mod of real part of lambda i A max/mod of real part of, well the eigenvalues when you find out they can be complex okay, so we only look at the real part. The complex part typically gives, if you look at a solution of exact solution of linear ordinary differential equations, then probably you remember from your course on control that you know roots of the characteristic polynomial have real part and complex okay.

The complex part gives rise to sinusoidal bounded behaviour, the real part actually decides the rate of convergence okay, so we only look at the real part, the maximum magnitude real power/minimum magnitude real part, this ratio is called as stiffness ratio. If this ratio is large typically >10, 20 then you have a problem okay 100 is very bad. **"Professor - student conversation starts"** yeah, (()) (30:12) so that is variable step size right.

You cannot have step change in h value, but you can have variable step size implementation, if you can afford to. So if the system is stiff, one way is use variable step size with tight accuracy monitoring that will you know take off your worry about the stiffness. (()) (30:39) then you can yeah one has to built-in some super intelligence, which looks at you know if one value is not changing you can start increasing the.

So if y1 has decreased okay already decreased and going to 0, then y2 you can increase the step size, but then you will have to have some you know if then else rule base or something like that, which you know a priori that now pressure has stabilized okay, now let me look at. But for a complex plant okay it is not always possible, for some simple system with 2, 3 variables you might be able to do what you are saying okay.

But for a chemical plant simulation that is difficult okay, but then you should know about the stiffness part, and then you know decide upon how to implement your scheme **"Professor - student conversation ends."** And then of course the stiffness ratio in this particular case there is a constant A matrix if you have a non-linear differential equation then you locally linearize, and then look at Jacobian.

So if you have if you are solving equation here of type dx/dt =f of x, x belongs to Rn, and you are at x t n =x n, then well we of course look at dou f/dou x at x=x n, we look at these Jacobian eigenvalues of this Jacobian, will tell you local stiffness you cannot define a global stiffness, the way local stiffness and eigenvalues of this matrix can change as exchanges okay. So in the situation where you do not know anything, and you are luxury to implement variable step size that is a better option okay.

Now before we move to differential algebraic systems, so one way is you know one way to deal with the situation is to say that well I am not really interested in the dynamics of the slower variable okay. I could argue that well pressure you know transient if it happens in seconds, I am not so much concerned, I am just worried about the steady state pressure behaviour okay, and I want to only study the dynamics of the slow system okay.

So one way see I have this differential equation, I will go back to the simpler differential equation, we let us say we have this differential equation dx1/dt =sum alpha 1 x1+alpha 12 x2 and dx2/dt =alpha 21 x1+alpha 22 x2 this is my differential equation. I know from the physics of the problem that dx1/dt is a fast mode, and dx2/dt is a slow mode okay. I can decide to approximate make a pseudo steady state approximation that this=0 okay.

Now there is a trouble because this is fine you can do the approximation, I do not want to worry about pressure transients I mainly concerned about concentration transients in the distillation column okay. We want to make this approximation this is no longer an ODE initial value problem, this is a differential algebraic system okay, now here you might say well that is not so difficult, you know you just eliminate x1 right x1 in terms of x2 and you have only 1 differential equation okay.

This is true, it is easy to solve when this is a linear differential equation these coefficients are nice, and you can solve. What if I say well this is not the case, but it is some g of x1, x2 =0 okay, likely to be the case in chemical engineering system some non-linear function of these kind of algebraic constraints in chemical engineering system arise because of thermodynamic equilibrium, concentration between or concentration of vapor phase and liquid phase are you know Euler thermodynamic equilibrium.

Actually, when you say that thermodynamic equilibrium that is a simplification, this is transient when the concentration on the you know plate changes there is some transient between you know that concentration and the concentration of the vapor phase, but we make an assumption that it is instantaneously vapor phase composition comes from the equilibrium, it does not happen instantaneously we are only making a simplification we are making an approximation okay.

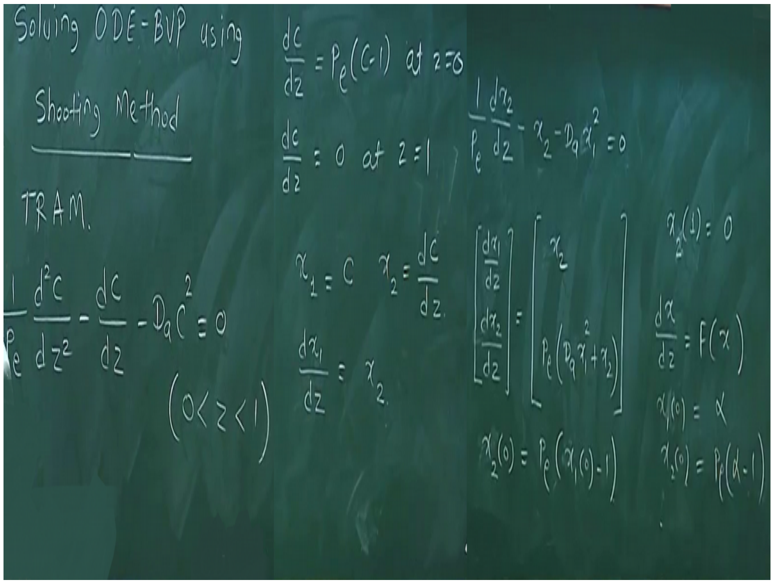
Because those timescales we are not so much concerned about those time scales, we assume that almost instantaneous there is for modeling purpose, in reality it is not going to happen there will be some time lag small time lag, we are willing to forgo the time lag okay, so you will get this kind of equations differential algebraic equations. So differential algebra equations of course belong to many times the stiff equations.

And then you need different solvers, different class of solver to handle this kind of problem, we will have a very, very brief look at these methods, how to handle this. Anyway before I move to that, so before we start talking about differential algebraic systems, I want to talk about one application of ODE solving in the different context, ODE initial value problem. I have a solver for ODE initial value problem okay, and I want to use this solver to solve the boundary value problem okay.

I want to solve a boundary value problem, now what is the difference between the boundary value problem and initial value problem. Initial value problem only at the initial point the conditions are specified and then you can go marching okay. Boundary value problem part of the conditions are specified at one boundary, part of the boundary conditions are specified at other boundary okay.

Suppose you have 2 differential equations you need 2 conditions, one condition might be specified at one boundary, one condition might be specified at other boundary okay, so now if you want to use marching algorithms, you have to do some more tricks, this class of methods are called as shooting methods okay.

(Refer Slide Time: 38:17)



Solving ODE-BVP using

Shooting Method

TRAM.

$$\frac{1}{P_e}\frac{d^2C}{dz^2}-\frac{dC}{dz}-D_aC^2=0$$

$$(0<z<1)$$

$$\frac{dc}{dz}=P_e(C-1) \text{ at } z=0$$

$$\frac{dc}{dz}=0 \text{ at } z=1$$

$$x_1=C \quad x_2=\frac{dc}{dz}$$

$$\frac{dx_1}{dz}=x_2$$

$$\frac{1}{P_e}\frac{dx_2}{dz}-x_2-D_ax_1^2=0$$

$$\begin{bmatrix}\frac{dx_1}{dz}\\ \frac{dx_2}{dz}\end{bmatrix}=\begin{bmatrix}x_2\\ P_e(D_ax_1^2+x_2)\end{bmatrix}$$

$$x_2(0)=P_e(x_1(0)-1)$$

$$x_2(1)=0$$

$$\frac{dx}{dz}=F(x)$$

$$x_1(0)=\alpha$$

$$x_2(0)=P_e(\alpha-1)$$

Now again I am not going to go too much deep into this, I just want to give you an introduction. So there are methods called single shooting, there are methods called multiple shooting, I am going to discuss only about a single shooting method, just to give you an idea of what is shooting method. Let us go back to our good old problem of tubular reactor with axial mixing okay, and then with that help of that example, I will show how I can use an ODE initial value problem solver to deal with the boundary value problem okay.

Now this is my TRAM problem 1/Pe and by now you are very, very familiar with this problem, you have solved this problem by multiple methods, so inside the domain 0 to 1 this is the differential equation which should hold second order differential equation. And at the boundaries you have dc/dz =Pe C-1 at z=0 and dc/dz=0 at z=1 a second order differential equation needs 2 conditions okay, one condition is at one boundary, the other condition at other boundary okay.

Now if I say that z=0 is the initial point okay, then to start using marching, I need 2 values in the beginning let us see how why we need 2 values in the beginning. Before that I am going to just transform this problem okay. I am going to define x1=C and x2 =dc/dz okay, this will give me 2 differential equations, one is dx1/dz=x2 okay, and then this differential equation I can write as 1/Pe dx2/dz right d2 C/dz square is dx2/dz okay -x2-Da x1 square=0 right.

So if I write it the standard form I will get dx1/dz dx2/dz, this is x2 and this is Pe times DA x1 square +x2, this is my differential equation dx1/dz and dx2/dz okay. Now my first condition is x2 0=this okay right, this is my differential equation, this is boundary condition one, this is boundary condition 2 okay. This equation is of the form dx/dz=f of x fine this is ordinary differential equation first order I have so many methods for solving this, you know Euler method, implicit Euler, explicit Euler, Runge-kutta whatever okay.

I have those methods for solving this, but to start solving okay, I need 2 things using initial value problem, I need x1 0 and x2 0 okay, I do not have 2 of them separately, I have one constraint okay. Suppose I decide to choose x1 0, x2 0 will get defined right, if I decide to choose x1 0, x2 0 will get defined. Then I can choose x1 0, let us say I call this alpha, then x2 0 will become Pe alpha-1 okay, this is my guess using this guess I get x2 which is this guess okay.

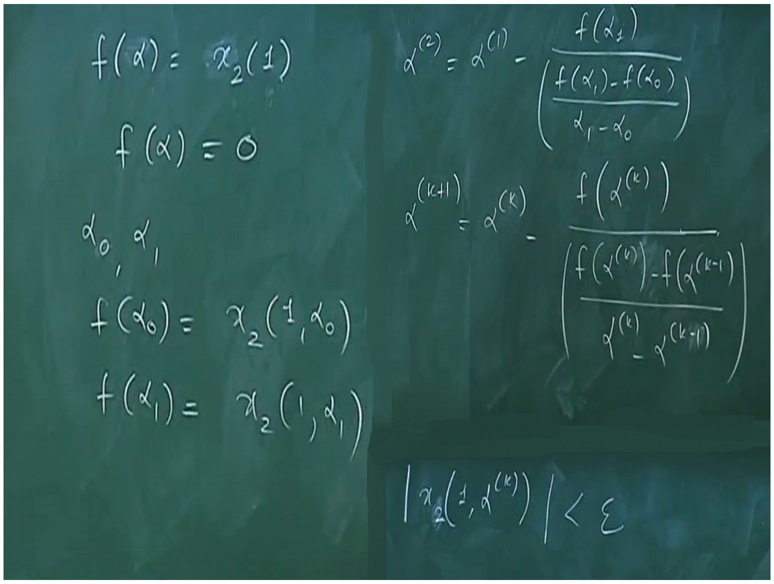
Now what I can do? See now this is a differential equation with 2 initial conditions specified, I can start marching in space now not in time I am marching in space, so I go from z=0 okay using say Euler method I start marching from z=0 to z=1 okay. If my guess alpha is correct okay what should happen this condition should be satisfied that is x2 1 should be =0 okay, if it is !=0 I can refine my guess okay, I can come back and change alpha okay again start marching.

So I can solve this problem iteratively okay, this is called shooting because you know we are trying to hit a bird which is hidden behind the cloud okay, so you just shoot you guess and shoot where the bird is okay, if it does not hit you start again take a new bullet and start. So you start shooting you guess the missing initial conditions at z=0, start shooting towards start marching towards z=1 okay, so guess is correct you know the conditions at z=1 will be satisfied.

This is a way I can use a ODE initial value problem solver to solve the boundary value problem okay, this is a boundary value problem being solved using repeated use of initial value problem solver okay. I am going to use again and again say Euler method or Runge-kutta method to march in space to reach from z=0 to z=1 keep checking this condition. Now the question is how do I form the iterations, can you tell me how will you form the iterations?

Now you have to merge Newton's method, see you should create a guess from the old guess, how will you do it x2 1=0 as an equation, x2 1=0 is actually function of this alpha okay, and so you can actually find the gradient.

(Refer Slide Time: 47:49)



$$f(\alpha) = x_2(1)$$

$$f(\alpha) = 0$$

$$\alpha_0, \alpha_1$$

$$f(\alpha_0) = x_2(1, \alpha_0)$$

$$f(\alpha_1) = x_2(1, \alpha_1)$$

$$\alpha^{(2)} = \alpha^{(1)} - \frac{f(\alpha_1)}{\left(\frac{f(\alpha_1) - f(\alpha_0)}{\alpha_1 - \alpha_0}\right)}$$

$$\alpha^{(k+1)} = \alpha^{(k)} - \frac{f(\alpha^{(k)})}{\left(\frac{f(\alpha^{(k)}) - f(\alpha^{(k-1)}}{\alpha^{(k)} - \alpha^{(k-1)}}\right)}$$

$$|x_2(1, \alpha^{(k)})| < \varepsilon$$

See suppose you choose some alpha 0, so what I am saying is f of alpha =x2 1 right, what is x2 1 is function of what is your guess initial guess alpha right, now you want to reach you want to solve the problem, see for the time being do not worry about what is the process of calculating this x2 1 okay, given alpha there is some mechanism by which you calculate x2 1 say Runge-kutta solver or whatever method that you are chosen, you have a way of solving this.

Where you want to reach? You want to reach f of alpha=0 am I correct, you want to reach x2 1=0 this is where you want to reach, classic one variable problem or non-linear problem okay might be a little difficult to get derivatives, it is not impossible it is possible to do it I will discuss this in the notes. But let us use the simpler approach secant method okay, what I can do is I can start with some alpha 0 and alpha 1 okay, I can take 2 guesses alpha 0, alpha 1.

How do you give a good guess, you have to use your physics knowledge, this is a tubular reactor what should be you know initial value, you have to use your understanding of chemical engineering to give 2 initial guesses. Now you know using alpha 0 I do calculations, I compute f alpha 0 which is whatever x2 value at 1 which you will get this will not be=0, because your guess is not correct, if your guess was correct you would have reached the solution okay.

Then there was no problem of getting iterations, now this is !=0 okay, then you can start with f alpha 1 and you again get x2, so let us all this as x2 1 alpha 0 x2 1 alpha 1 okay. Now I want to create a new guess alpha 2 okay, so what I am going to do now. So what is alpha 2 this will be alpha 1-the derivative, the derivative is f alpha 1-f alpha 0/ we have to use derivative inverse actually, so if alpha 1/f alpha 1-f alpha 0, simple secant method okay.

So in general I can write that is alpha k+1 is alpha k-f of f alpha k, what is the f value? f is nothing but x1 evaluated sorry x2 evaluated at z=1 okay, you want to reach x2 at 1=0 okay, and you generate initial guess like this. If you have a differential equation boundary value problem where you know you do not have only 2 equations, but you have 4, 5, 6 equations it might happen if you have a temperature and concentration to be considered together okay.

Then you have more missing initial conditions, you can write Stein method, you can use multivariable secant method, you can use you can also use Newton-Raphson method. So this is just the idea as to how to do solving of boundary value problem using a shooting method, where the missing initial conditions are guessed, in this case there is only one missing initial condition, you guess the missing initial conditions.

You go to the final point find out the gradient generate a new guess and keep doing this ODE initial value problem solving till you reach the solution. How do you reach, how do you find whether you reach the solution? You look at whether f, so basically finally what I want to happen is this mod of x2 at 1 alpha this should be strictly <epsilon, we can specify epsilon to be 10 to the power-10, what if 0 it is difficult to say, so a small number okay, till these 2 come very, very close they are very, very small okay.

One minute this cannot be the condition here, because we want to reach x2 not the gradient, here I would have to check x2 =0 okay, so this should be <epsilon. I want the final value to go to 0, so I cannot say it goes to exactly=0, I can give a very, very small number here, and say that this

should be close to 0 okay, so if this should be very, very close to 0 whatever epsilon is specify, then you stop the iterations okay, till then till this condition is not satisfied you keep guessing alpha okay.

For every guess of alpha you use an initial value problem ODE IVP solver to integrate from z=0 to z=1 that is you do one shooting, well if you hit the target you stop the iterations, if you do not hit the target generate a new guess of alpha and keep doing this. See your combined initial value problem solver with the secant method, see one thing which I wanted to stress right in the beginning is that there are very few tools that we have.

And finally you know we have to make a bhelpuri of a solution is a bhelpuri, for a particular problem you have to come up with you know taking this and taking this and then you know mixing the 2 and creating a recipe. So some of the one of the well-known books in numerical methods is titled numerical recipes, is a book called numerical recipes in C or numerical recipes in FORTRAN very popular books, very nicely written books.

So I think the word recipe when I read it for the first time I was intriguing I mean why should somebody called a numerical recipe okay, but it is indeed if you start looking at it, it is a recipe you know you are concocting solution by using some basic tools basic ideas, and you are mixing the 2 same problem, you could have solved by converting into you know algebraic equations, but then again for solving the algebraic equations you have to use Newton's method Newton Raphson okay.

Here, you are not converting them into algebraic equations, you are converting them into an initial value problem using initial value problem repeatedly okay, and then it is a mixture of things. See, to solve let us say you convert into algebraic equations using orthogonal collocation the same problem okay, then to solve it using Newton's method you need A x=b solver okay, so solving A x=b or solving an ODE-IVP, these are some basic tools, if you have them you can concoct a solution to large class of problems okay.

You can concoct a recipe for solving many, so you have to understand this how these ideas are used to concoct recipe or a solution that is very, very important. So next class we will have a very brief peek at this differential algebraic systems, because in real chemical engineering problems you more often than not encounter differential algebraic systems, so let us at least touch upon the tip of the Iceberg and then close the show.