MATLAB Programming for Numerical Computation
Dr Niket Kaisare
Department of Chemical Engineering
Indian Institute of Technology, Madras

Module No. #08
Lecture No. #8.2
Practical aspects of ODEs – Stiff systems and solutions using ode15s

Hello and welcome to MATLAB programming for numerical computations. We are in module 8 and this module we are covering practical aspects of solving ode initial value problem. In the previous lecture we extended ode45 to a general multivariable case. In this lecture we are going to take up examples where ode45 is going to find it very difficult to solve the problems. The reason is ode45 is a Runge-Kutta solver which is an explicit solver.

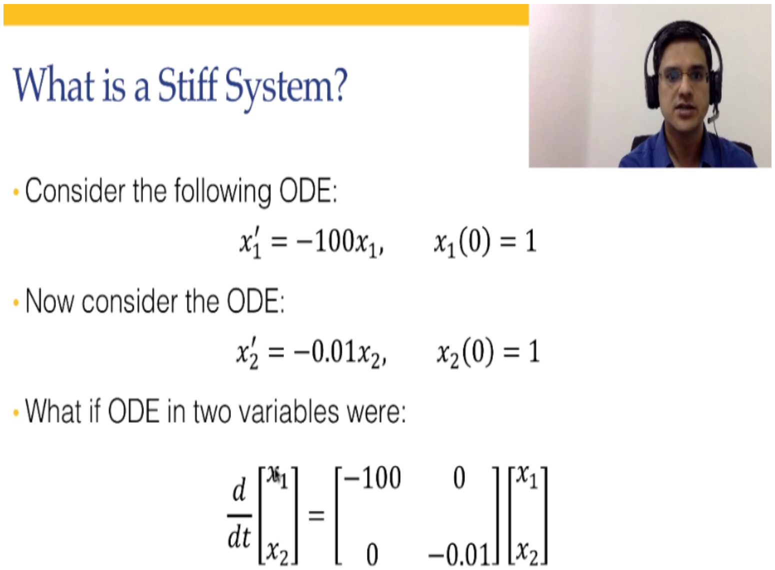
An explicit solver will have numerical difficulties when face with what is known as stiff systems. So what I am going to cover in this lecture is what stiff systems are and if ode45 fails how you can instead use ode15s. What we are going to is we will first start of with the same problem that we looked at in the previous lecture. So let us go to MATLAB and do that okay.

(Video Starts: 01:12) So this is the solveMassSpring example that we took earlier. What we have done in the previous lecture is this. What we did was, we said that the way we are going to write our functions that we are going to use for ode45 in multivariable case is going to be exactly same as we did in the single variable case. The only difference is that 1 our y is an n by 1 vector, n rows and a single column and our fval is also an n/1 vector, n rows and a single column again. We solve this using ode45.

Now if we want to solve it using ode15s, I can show you how simple it is to solve using ode15s. If we were to solve this using ode15s all we need to do is change ode45 to ode15s.That is all that we need to do okay. I click on run okay and I get the results the same as what we had seen before. So, take home message over here is that ode15s for an external user like us works in the same manner as ode45.

If ode45 works then why do we need ode15s and the reason for this is ode45 may not work for all the cases. So let us go on to power point and see okay. (Video Ends: 02:52)

So the cases where we need to look at ode15s is what is known as stiff systems.
(Refer Slide Time: 03:01)



So let us consider the first consider the ode x`= -100 * x1 okay. (Video Starts: 03:12) We can solve this problem as (tSol, xSol) = ode45 @ (t, x) -100 * x1, negative 100 * x. Let us take the tSpan for this as say (0:1). And let us and take the initial value as 1. Let us also plot (tSol, xSol). And let us see what we get okay. When we solve (tSol, xSol), we see that the value of x or rather the value of x1 drops rapidly and goes on to equal to 0 okay.

So within pretty much .01 seconds the xSol value has dropped to its steady state value of 0 okay. So let us hold this. Now what we will do is will consider the other ode -100 * x. So (tSol, xSol) sorry, not -100 * x, -0 .01 * x okay. So -0.01 * x is what we need to solve okay. And we will plot this result as well. And let us look at the graph. As you can see in the red color line over here this particular value remains nearly constant at 1.

And let us now solve the same problem again but instead of tSpan being from 0 to 1 let us make tSpan as 0 to 3000 and then plot this and see what we are going to get. Let us look at that figure okay. As you can see here the original the earlier graph goes from 1 to 0 within 0.1 seconds.

Whereas, the second system goes from 1 to 0 takes almost 500 seconds in order to do that. (Video Ends: 05:30)
So when this coefficient is a very large value the system responds very quickly, if this coefficient is a very small value then the system responds very slowly. Now the question is what if the 2 variables were together in the same equation. If we have dx in the same vector equation rather. If we have dx1/ dt = -100*x1 and dx2/ dt = -0.01*x 2 what we are going to get is this.

So if we have an explicit solver such as Runge-Kutta solver and we have this system and this system together as shown over here okay. We need to take really small steps because of this particular equation. The steps that we need to take would be of the order of 0.001. But in order to get the evolution of this particular equation. We need to solve this problem for a large number of times which means we need to take a step of .001 but need to go almost to 100 or 1000 steps.

That is the reason why stiff systems are going to be difficult to solve. The reason is we need to take small steps in order to ensure stability but we want to go to a large amount of time. (Video Starts: 06:56) Let us see what happens when we try to solve this using ode45 and ode15s. edit, multiVarFun. Let us create this function fval = multiVarFun (t, y). We will do a = (-100, 0) (0, -0.01) that is our A matrix. And fval = a * y okay. So we have done this.

So now let us go and run this using ode45. (tSol, ySol) is going to be equal to ode45 @ (t , y) multiVarFun(t, y). For tSpan from 0 to let us say 0 to 200 that should be sufficient for us to show our show what we have. And for initial condition of 1 and 1 okay. And we have run this using ode45 okay. Now MATLAB has solved as using ode45. Look at the number of time steps MATLAB has taken.

It has taken 24157 times steps in order to solve this rather simple problem using ode45. So let us plot (tSol, ySol) and see what we are going to get okay. This particular guy responds very quickly whereas this guy is responding extremely sluggishly. Let us look at our vector tSol and see what we get okay. So as you can see our solver is taking very, very small steps it started with steps in the order of 10 ^ -4 but as things progress okay it could not take large steps.

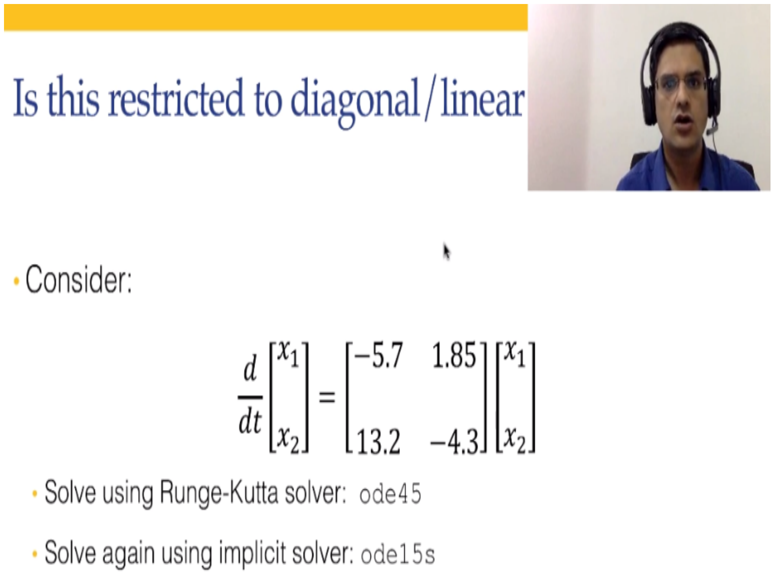
It is still taking steps of the order of 10 ^ -3 as you can see over here. The reason why it needs to take steps in the order of 10 ^ -3 is because an explicit method will become unstable if we take larger steps. On the other hand ode15s is an implicit method and there is no restriction on the r. Typically there is there is a very small restriction on that step sizes that ode15s will take. So let us try to solve the same problem using ode15s.

Let us call this as t2 and ySol 2 = ode15s multiVarFun from (0 200) and initial condition is (1; 1) okay. And let us solve this and let us plot (t2, y2) okay. As you will see (t2, y2) has the exact same behavior as (tSol, ySol), using ode15s we are going to get the exact same behavior. However, when you look at the Val, the number of points at which ode15s has run it has run only at 88 points instead of 24000 points.

Let us click on our t2 and see the points at which it has run okay. Initially when our first guy our x1 is still evolving. It the ode solver needs to take small steps in order to maintain accuracy. So in order to maintain accuracy ode15s has taken small steps 10 ^ -4, 10 ^ -4, 10 ^ -4. Now, it has started taking steps of the order of 10 ^ -3 but as we go further okay.

Now it has started taking steps of the order of 10 ^ -2. We go even further beyond say time 2 it has started taking steps in the order of 1. Now it has started taking steps in the order of 10 accuracy dictates that the steps need to be taken in the order of 10 ^ -3 till the response of x1 stabilizes. But after the response of x1 stabilizes we do not have to worry about x1 we should be able to take larger steps.

We are unable to take larger steps in an explicit solver because this is a stiff system but we are able to take larger steps in an implicit solver which is ode15s. So the question is what the definition of stiff system is. (Video Ends: 12:08) The working definition of stiff systems for this course is going to be a stiff system is a system where you have a highly fast dynamics and highly slow dynamics couple together in the same ode.
(Refer Slide Time: 12:25)

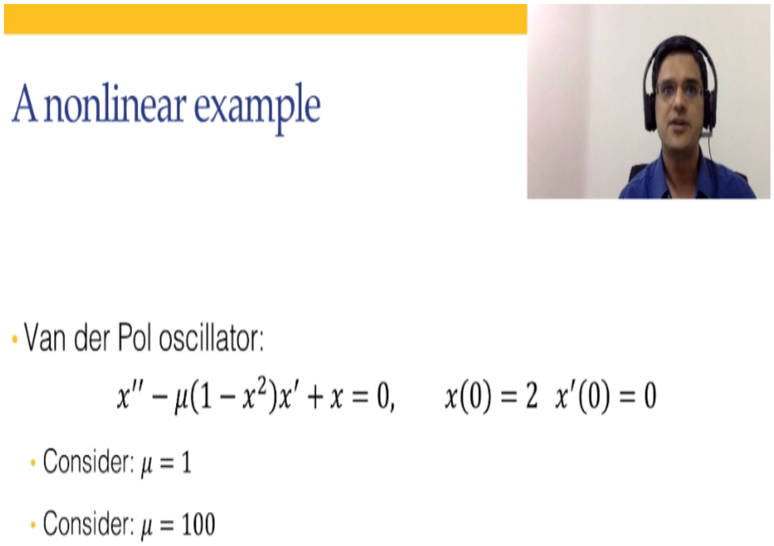


(Video Starts: 12:36) So question is does the stiff system needs to be diagonal or linear. So let us take this particular example where a is -5.7, 1.85 okay, 13.2 and -4.3. And then let we click evaluate selection so that now we have a okay. So now let us solve this using ode45 okay. Everything else remains the same we are going to solve it using ode45 okay. And when we solve it using ode45 we see that we take 2400 steps. We now solve it using so we now solve it using ode15s.

When we solve it using ode15s let us run this ode15s and see how many steps ode15s takes. Ode15s requires only 40 steps. Question is why is this happening? The reason why this is happening is let us take the Eigen values of our A matrix. The Eigen values of a matrix are of the order of 10 and of the order of 0.01 okay. So the ratio of the 2 Eigen values ans1/ans2 is a very large number.

It is of the order well it is actually not very large numbers it is reasonably large number the ratio of the Eigen values is of the order of thousand. So this matrix A is an NILL condition matrix. (Video Ends: 14:01) We are going to have stiff system and it might be necessary to use ode15s okay. Let us look at an example of a nonlinear system.

(Refer Slide Time: 14:12)



This is a solved example in MATLAB help files, this is known as a Van der Pol oscillator, this is the overall equation x``- mu *(1-x ^2)* x `+x = 0 and these are the starting points okay. (Video Start: 14:31) So let us go to MATLAB. Edit, myVDP okay. So function fval = myVDP (t, y). Let us call mu = 1 okay. Our first guy is going to be dx / dt = x`. So our fval (1, 1) is going to be nothing but so let us call x = y1 and v = y2 okay.

So, fval is nothing but dx/dt and dx/dt is equal to v and fval (2, 1) is going to be equal to square mu *(1-x^2)*v-x. And now let us run our (tSol, ySol) = ode45 @ (t , y) myVDP (t , y) okay. Let us run it from 0 to let us say 10 that should be sufficient with an initial condition as x0 = 2 and x' 0 = 0 so that is our initial condition okay. We solve this and let us also do plot (tSol, ySol) (1, :) okay (tSol : , 1). So let us do this okay.

So this our Van der Pol oscillators solution for mu = 1 for mu = 1 Van der Pol oscillation. Oscillator is not a very stiff oscillator and therefore we get the solution very quickly. So the first practical tip is that when you want, when you know that the system is non stiff or you are not sure about the stiffness of the system you should first use ode45.

If ode45 does not work or if you know that the system is a stiff system that is when you should use ode15s. So let us make the system stiff by changing our mu to 100 and let us try to solve this using ode45 okay. Let us solve it up to from 0 to 1000 and let us solve this. And let us see, we will see that this particular system behaves rather sluggishly it takes a fairly long time in order to solve okay.

I am still waiting for ode45 to solve and finally ode45 has solved this okay. It has taken almost 2 lakhs step. Let us now solve this using ode15s instead because this is the stiff system. As you will see that solving this using ode15s is going to be much faster and instead of 2 lakh points it is going to solved as in only 1725 points. And the result is the same as before.

Now if we were to increase the mu to 1000 and this is what your MATLAB help example gives you. If you increase this to 1000 you will be not able to solve this problem using ode45. So let us try to solve using ode45 and let us give this I have waited for nearly 1 minute and it has not yet solved. So let me just cancel this by pressing control c. As you can see in a reasonable amount of time ode45 has still not solved this problem okay.

Instead if we were solve this using ode15s, we are going to get the solution fairly quickly okay. Let me solve this say up to 40 span = 5000 and then I will show why this is an extremely stiff system. As you can see the system responds slowly at first till a certain point and at this point that the system response is very rapid okay. So this is because of the overall stiffness of the system.

Again you have slow time scale then a very rapid time scale, slow time scale then a very rapid time scale. So as you can see over here in this practical example also you have case where, a slow time scale is mixed with a fast time scale. (Video Ends: 19:40) So with that I come to the end of this lecture.

What I have covered in this lecture ode45 and ode15s. Ode45 is a Runge-Kutta explicit solver. If you know a priori that the system is the stiff system you want to use ode15s instead. If you do not know whether the system is stiff or not, it is always a good idea to first try ode45. If ode45 does not work then you would switch to ode15s. With that I come to the end of this lecture and I will see you in the next lecture.