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

Module No. #08 Lecture No. #8.4 Practical aspects of ODEs – Some practical Examples

Hello and welcome to MATLAB programming for numerical computations. We are in module 8. In this module we are covering practical aspects of ordinary differential equations initial value problem. We have looked at a several examples of ordinary differential equations initial value problems from practical interest. This is the last lecture of this course and we are going to finish of this course by taking a couple of practical examples that we started of this course with. (Refer Slide Time: 00:46)





• Damped spring-mass system:

$$system: mx'' + cx' + kx = 0$$

$$\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} v \\ -\frac{cv + kx}{m} \end{bmatrix}$$

Before we do that let us go to the type of problems that we have considered in this module. The first example that we considered is a damped spring-mass system model for which is given over here okay. What we did was we wrote v = dx / dt and then we obtained the 2 odes of the form dx / dt = v and dv / dt equal to this guy. This we wrote as a multivariate ode.

(Refer Slide Time: 01:16)

Examples Considered



.

Van der Pol Oscillator

$$x'' - \mu(1 - x^2)x' + x = 0 \qquad \frac{d}{dt} \begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} v \\ \mu(1 - x^2)v - x \end{bmatrix}$$

٠

Thereafter we looked at the Van der Pol Oscillator. Again as before we wrote dx / dt = v and we got 2 equations in 2 variables. These 2 equations we were able to solve for a low value of mu such as mu = 1 we were able to solve using ode45. However for a fairly large value of mu such as mu = 1000 this model becomes very stiff. And we need to resort to ode15s.

We saw what stiff systems means stiff system basically means a system where there are slow dynamics and fast dynamics mixed together in the same ode set sequence of odes.

(Refer Slide Time: 02:05)

Examples Considered



•

.

Transient heat conduction

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} - \beta (T - T_a)$$

Then in lecture 8.3 we considered a transient heat conduction problem which results in a parabolic partial differential equation. The partial differential equation was written in this form.

We used what is known as method of lines in order to convert the pde into ode in time. We did that by writing $d^2 t / dx^2$ as a central difference formula.

(Refer Slide Time: 02:34)

Examples Considered



•

$$\frac{d}{dt} \begin{bmatrix} T_2 \\ T_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \alpha \frac{T_3 - 2T_2 + T_1}{h^2} - \beta (T_2 - T_a) \\ \alpha \frac{T_4 - 2T_3 + T_2}{h^2} - \beta (T_3 - T_a) \end{bmatrix}$$

Transient heat conduction

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} - \beta (T - T_a)$$

As a result of this when we obtained the central difference formula at various locations in the rod. Let us say from going from t2 to t10 these are the 9 equations that we got. The first equation is dt2/dt = alpha multiplied by the central difference formula - beta * t2- ta. Likewise we did the same thing for t3, t4, t5 up to tn. That is the example that we considered in the previous lecture. So this is what we covered in the previous lecture, lecture 8.3.

(Refer Slide Time: 03:07)

Examples Considered



Damped spring-mass system:

$$mx'' + cx' + kx = 0$$

· Van der Pol Oscillator

$$x'' - \mu(1 - x^2)x' + x = 0$$

Transient heat conduction

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} - \beta (T - T_a)$$

So, so far the practical examples that we have considered is the damped spring mass system, the van der pol oscillator and transient heat conduction problem. These 2 were 2 equations, 2 odes whereas, the transient heat conduction problem resulted in 9 odes. I also showed you that going an increasing n can result in a larger number of odes which ode45 has no problems at all solving. (Refer Slide Time: 03:37)

"Back to the Beginning"



Indian captain, Mahendra Singh Dhoni, hits a ball with initial velocity of 35 m/s and angle of 45°. If the boundary is at a distance of 75 m, will he score a six?

So finally in this lecture, we are what I call this is back to the beginning. Right in lecture 1.1, I had given a motivating example. The example was Indian captain Mahendra Singh Dhoni hitting a ball with an initial velocity of v0 at an angle given by theta0. And we need to find out where the ball is going to land okay. So to setup the problem what we have is we have initial velocity and initial angle and therefore we have the velocity in the horizontal component and in the vertical component.

We are starting at x = 0, y = 0 with a velocity of 35 in this direction okay. In the vertical direction we have gravity acting and in the horizontal direction we have air drag that is acting okay. (Refer Slide Time: 04:29)

Setting up the Problem



Acceleration is known: gravity in y-direction and air gray in x-

$$\frac{d^2x}{dt^2} = -\kappa u, \qquad \frac{d^2y}{dt^2} = -g$$

Velocity and initial location are given:

$$v_{net} = 35$$
, $u_0 = v_{net} \cos(\pi/4)$, $v_0 = v_{net} \sin(\pi/4)$

System of ODEs:

- x' = u

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ u \end{bmatrix} = \begin{bmatrix} u \\ v \\ -\kappa u \end{bmatrix}$$

The acceleration due to gravity in y direction is given by $d^2y / dt^2 = -g$. And at air drag in x direction will result in $d^2x / dt^2 = - \text{ kappa * velocity u.}$ Where u is the x component of velocity and v is the v component of velocity. The initial velocity and initial location are given the initial angle is also given because the angle is 45 degrees or pi /4 radiance, we get the horizontal component of velocity is 35 * cos pi /4 and the vertical component is 35 * sin pi / 4.

The system of odes that we want to solve as before we are going to write dx / dt = u and dy / dt = uv. So we have dx / dt = u, dy / dt = v, d^2x / dt^2 which is du / dt = - kappa u and dv / dt = - g. So these are 4 odes that we need to solve okay. And the overall odes will look in the vectorial form as shown over here. We will call this guy as vector capital y okay.

(Video Starts: 05:56) So let us go on to MATLAB and try to solve this problem. CricketFun and as we have been doing function fval = cricketFun (t, Y) okay. Get variables small x is y1, small y is y2, small u is y3, and small v is y4. Constants gravity is 9.81, the value of kappa we choose as 0.006 okay. This was the value that we had taken in lecture 1.1. So we will stick to that okay.

Describing dy / dt okay. So fval we are going to initialize that as a column vector we have 4 rows in a single column dx/dt, dy/dt, du/dt, and dv/dt. So fval we will initialize as zeros (4, 1). fval (1) =, fval (2) =, fval (3) =, and fval (4) = sorry, fval in brackets4 = okay. So what is dx/dt, dx/dt is

nothing but u, dy /dt is nothing but v okay. du / dt is, du /dt is -kappa * u,- kappa * u and dv / dt is - g okay. And that is all there is to it in cricketFun. So we have saved this and now we will edit 1 more script file that will run this file for us.

So we will call as edit, let us call this as dhoniSixers okay. To solve trajectory of a ball hit with initial velocity okay. And v let us see our vNet was 35. vNet = 35, theta= pi / 4. So our u0 is vnet * cos theta and v0 is vNet * sin theta okay. We will also need to write x0 = 0 and y0 = 0 as well okay. So we have all of these and let us initialize. The initialization is our y0 is going to be equal to x0, y0, u0, v0 okay. So let us write this over here.

Initialization, executing ode. So we do not know what tSpan we need to use so let us say we are going to use tSpan as $(0\ 10)$ okay. So let us say tSpan is $(0\ 10)$ and our (tSol, ySol) is going to be equal to ode45 (@ $(t\ ,y\))$, the function name is cricketFun sorry cricketFun $(t\ ,y)$ again our tSpan is $(0\ 10)$ and y0 okay. Verifying the results. Verifying the results so what we want to do is we want to plot (y,x) versus time. So let us do plot (tSol,ySol).

Let us plot all of that first and then we will see what is going to happen. So let us run this oops! Undefined variable ySol because we need to have Y over there. So now let us run this and see what happens okay. So the blue line is our x, the red line is our y. So y is indeed going to 0 or going below 0. So what exactly is happening over here so what exactly is happening over here is this.

Is that the ball trajectory it goes through a maxima and then it falls. Now in reality what happens is when the ball is going to hit the ground the model is no longer going to be applicable because that ball is going to bounce back. That particular bouncing motion is not incorporated in our model. So model is correct only till the point that y > 0.

So it looks like at approximately 5 or so y value has been gone less than 0 right. So what we are going to do next is we will just solve it for tSpan from 0 to 5 and let us run this okay. Yes, so now we have got all the plots. Let us find out what is the value of ySol at the end, the value of vertical

location at the end. So ySol (end, 2). So that is at the last time, the second row the second column, the second column represents this location y from in the vertical direction.

So when we see this okay that value is still 1.11. So let us go and do this solve this until tspan = 0 to 5.5. We will run this okay and we will again find out ySol so, ySol has gone below 0 so it is gone substantially below 0. We do not want that to happen. So we will let us just try it up to this point okay. And this is negative. So the ball has finally reached the ground at this particular time okay.

So in order to verify the results what we want to check is we want to plot our x location and our y location. So what we will do is to plot all the rows that means at all times the x location. And we will also want to plot the y location. So 1 to 2 okay. So this is what we are going to plot. So what that is going to give us is how the x location changes and how the y location changes. What we will do further is we will show the property editor and we will ensure that the y axis is plotted from 0 to the highest value okay.

Okay, so this is how the y location changes and this is how the x location is changing. So the ball has indeed reached a point which is beyond 75 meters okay. So this is the way we are plotting our x location and our y location. Let us say we wanted to and we have plotted this against the time. So let us say we want to plot what the trajectory that ball takes.

And in order to do that we want to plot y in vertical axis and x in the horizontal axis. So what we will do is that we will say xlabel time and ylabel location in meters okay. We want to make another figure. So plot tSol sorry, not plot tSol plot (ySol). So that is the x location we want to plot at all times. So all the rows and the first column will be the x location and we want to that x location will be plotted on the x axis, y location will be plotted on the y axis.

So that is going to be (:, 2) and we want to plot that as circles okay. And xlabel is x in meters and ylabel is y also in meters okay. Let us save this and let us run this okay. And this is how the ball trajectory has taken. So this is how the ball goes when it was hit by Mahendra Singh Dhoni okay.

(Video Ends: 16:33) So let us go back to power point and see what we have done so far. So what we have done today is we have taken the example that we started this particular course with.

This particular course we started with showing an example of Dhoni hitting the ball at a particular angle, at a particular given velocity and we want to find the trajectory that the ball takes okay. At that stage in lecture 1.1, I had said that I will cover this particular problem at a later stage and today what we are doing is we are finishing of this particular course by taking up the same problem and solving it with our ode solver ode45.

So with that I come to the end of this particular lecture and indeed into the end of this particular lecture series. I hope you have enjoyed this lecture series and you have learned how to do numerical computations using MATLAB. Thank you and see you later bye.