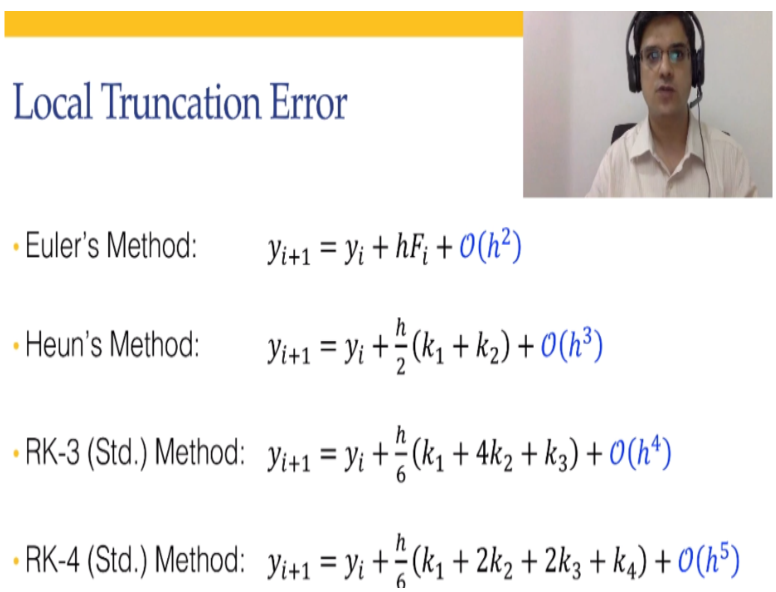
MATLAB Programming for Numerical Computation
Dr. Niket Kaisare
Department of Chemical Engineering
Indian Institute of Technology, Madras

Module No. #07
Lecture No. #7.5
Ordinary differential equations – Initial value problems – Error analysis of Runge Kutta Methods

Hello and welcome to MATLAB programming for numerical computations. We are in week number 7. In module 7 we are covering ordinary differential equations initial value problem. We are in lecture 7.5. This is the last lecture of this module. In this lecture we are going to present some error analysis of Runge-Kutta methods.

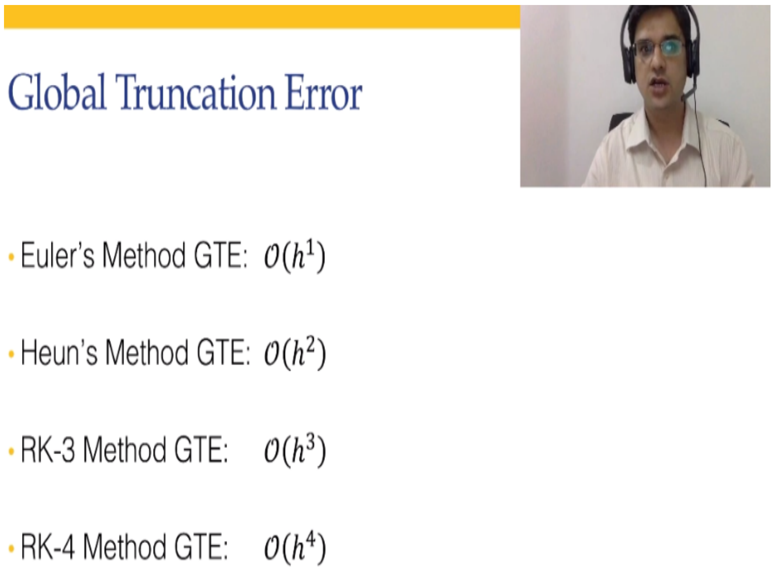
So far we have covered Euler's method which can be considered to be first order Runge-Kutta method. We have covered a couple of RK-2 methods. Specifically, Heun's method and midpoint method. And in the previous lecture, lecture 7.4 we have covered RK-4 method, standard RK-4 method. Today we are going to present error analysis of RK methods okay.

(Refer Slide Time: 01:02)



So first let us talk about the local truncation error. We will not demonstrate this in today's lecture, what I will show you is, with the global truncation error.

(Refer Slide Time: 01:20)



Now if we were to compare global truncation error instead of local truncation error, all of the odes solvers they drop bit overall order of accuracy by 1 power of h. What that means is Euler's method, the global truncation error is h ^ of 1, Heun's is h ^ 2, RK-4 is h ^ 4. Today's lecture, this is what we are going to demonstrate and compare. We are going to see how the error of Euler's method compares with that of Heun's method and RK-4 method. Specifically we are talking about the global truncation error.

(Video Starts: 02:02) Let us now go to MATLAB not to do this okay. What we will do is, we will bring up the codes that we had written earlier. We had written RK-2 Heun's method code, RK-4 standard method code and Euler's explicit method code okay. So RK-2 Heun's method which uses myFun, which we had done in lecture 7.4.

And this was the function myFun okay. And then we have, we had the RK-4 standard method okay. Which also use myFun. What I will do also is, open Euler's explicit method okay. So this was the Euler's explicit method RK-2 and RK-4 methods. Euler's explicit method also we are going to change this instead of -2 t * y, I am going to use myFun (ti, yi) okay.

Exactly in the same way as we had done before, as we had done over here. That is what we have done again with our Euler's explicit method okay. Save this, okay. And what we will do is, we will do a comparison of Euler's explicit, RK-2 Heun's and RK-4 standard method. Edit, compareODEs. Compare global truncation error of Euler, Heun and RK-4 methods okay. What I will do, I will copy this part, t0 is 0, y0 is 1.

Let us take tEnd = let us say 1 also. h we are going to keep varying so h, HALL, let us say is 0.1, 0.05, 0.01 and 0.005. Let us say these are the 4 h values that we have okay. for i = 1 :4, h = HALL(i), n equal to what we had earlier tEnd - t0 / h okay. With this we are going to call Euler's explicit, RK-2 Heun's and RK-4 standard method okay.

So what we are going to do is we will call the RK Euler's and Heun's method but we do not require everything. We only require the Yend at tEnd, so Yend myEuler okay. And we are going to call it with our y0, t0, h and N okay. So Yeuler is this Yheun equal to myHeun (y0, t0, h, N) and Yrk4 = myRK4 (y0, t0, h, N) okay.

So let us go to Euler's explicit and convert this into a function. Remember what we had said in module 1 where, we discussed functions versus scripts. Scripts are something that we are going to use in order to complete a particular task in MATLAB. We are going to use functions if we want to compute something as a function of certain input variables or if we want to repeat a particular task multiple number of times using parenthesis.

That is why we are going to convert all of these 3 odes solvers. Our Euler's method Heun's method and RK-4 standard method into a corresponding functions okay. The first function is myEuler okay. Function yEnd = myEuler sorry, compared odes myEuler (y0, t0, h and N). So we do not need t0, we do not need y0, we do not need tEnd, do not need h and we do not need n. So everything that was in problem definition.

We did not need. Why because the problem definition was given in the compare ode script. So there is just 1 script which is the compare ode script myEuler, myRK2 and myHeun and myRK4 are going to be 3 functions okay. We will need to calculate tEnd also. So tEnd is going to be t0+ N*h okay. So that is something that we have added okay. We do not want plot or anything of that sort. We want yEnd is nothing but Yend and that is the only thing that we need to do okay.

Likewise we are going to do the same thing with RK-2 Heun's and RK-4 standard as well. Before doing that let us go to compare odes and see whether this works or not. And ERR (1, i) = abs (Yeuler-Ytrue). And the true value which I will again quoted over here, yTrue=exp (-tEnd^2) okay. So let us just run this and see whether this works or not okay. We do get certain error. Ytrue is undefined. Let us go and see why we are getting that Ytrue okay.

That is again, it is a matter of MATLAB being case sensitive and I was not careful about the cases. I need to initialize ERR=zeroes (3, 4), 3 methods and we have our 4 errors okay. So let us save and run this okay. So now we have our errors, the first row is the errors in Euler's method. So let us look at ERR, we only focus on row number 1 and these are the errors in Euler's method.

So let us actually have HALL up to 0.001 as well okay. And let us now do this for Heun and RK-4 and likewise error (2, i) is for Heun, Yheun and RK-4 okay. I will quickly go to RK-2Heun and save as myHeun okay. We remove this part and replace it with function yEnd= myHeun and let us just copy paste this okay. And RK-4 standard also save as myRK4 okay. Function yEnd = myRK4 and just paste this over here and delete this part okay.

Again from myEuler, we will take this tEnd and paste it in both of these functions okay. Remember that is what we had done earlier okay. Now the next task is going to be delete the output part and okay and save this. And likewise this also, we will do the same thing okay. So what we have done is we have converted, I have gone carefully over how to convert explicit Euler into myEuler.

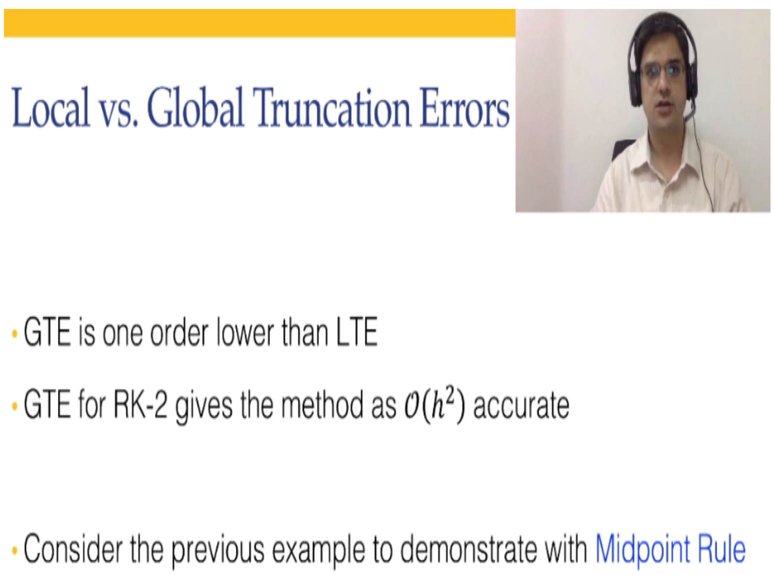
A script which was explicit Euler, we have converted into a function that was myEuler. The function takes the initial conditions step size and number of steps. And returns the value at the end point. That is what all the 3 functions are intending to do okay. We use all of these functions in our compare odes file for. And run it for different values of h from 10 ^ -1 to 10 ^ -3.

And using plot for comparison. And what we are going to do is we will plot log logarithm (H) against logarithm (ERR) okay. And let us run this. I have certain error okay. That it should be plot (HALL) and plot of log (HALL) and log (ERR). So these are 3 plots, this 1 is for RK-1 or Euler's method, this 1 is for RK-2 method, this 1 for RK-4 method.

All of these are straight lines though we are the way we are going to compute the slopes are LE = log (err) and LH = log (h) sorry, log (HALL) okay. So our denominator is always going to be denominator, is going to be LH (end) - LH 1 okay. Our numerator is going to be LE all the rows, last column-LE all the rows first column. The slope is going to be numerator divided by denominator okay.

So the slope for RK-2 method sorry, slope for Euler's method is 1.0, slope for RK-2 method is 2.0 approximately and the slope for RK-4 method 4.0. So this demonstrates the order of accuracy of the 3 RK methods from a global truncation error perspective. So we have the 3 straight lines with slope of 1, slope of 2 and slope of 4 okay. (Video Ends: 17:15) So this was the discussion about global truncation error.

(Refer Slide Time: 17:19)



Global truncation error is 1 order greater accurate than the global truncation error just like what we had seen earlier in numerical integration. The global truncation error of RK-2 method gives us the method as h ^ 2 accurate, RK-4 method is h2 ^ 4 accurate. What you can do as an assignment is you take the previous example and demonstrate it with the midpoint rule and see for yourself that like the Heun's method midpoint also gives us order of x square accurate global truncation error.

With that I come to the end of this lecture and indeed to the end of module 7. The module 7 we considered ode initial value problem. We first started with Euler's method considered both explicit and implicit Euler's method. Later we said that in this module we are going to focus on explicit methods. Only in lecture 7.2 we introduced ourselves to Runge-Kutta second order method and extended it to Runge-Kutta 4th order method in lecture in 7.4.

In today's lecture, lecture 7.5 we did an error analysis of all the 3 methods. In lecture 7.4 we covered ode 45 which is the most popular algorithm that is provided by MATLAB in order to solve the ode problem. With that I come to the end of this module. I will see in module 8. Thank you and bye.