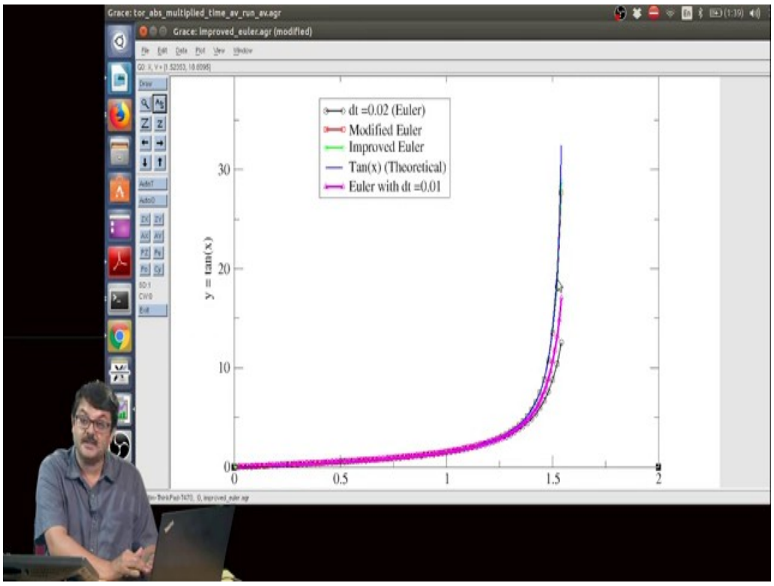
Computational Physics
Dr. Apratim Chatterji
Dr. Prasenjit Ghosh
Department of Physics
Indian Institute of Science Education and Research, Pune

Lecture - 27
Differential Eqns. Euler and Runge Kutta Part 02

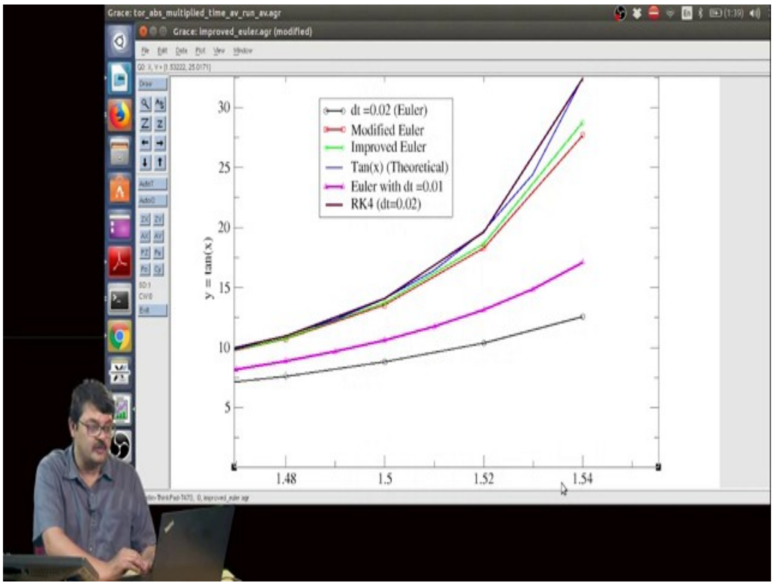
(Refer Slide Time: 00:16)



What happens if you had used Improved Euler? If you have improved Euler Modified Euler and Improved Euler which I have already implemented and plotted here and of course, you must also implement and plot it and compare these things and also for different values of d h. You see that of course, at small values of x you have a good agreement hardly surprising even Euler was giving a good match for small values of x, but at large values of x both modified and improved Euler is much better you see that this is nearly matching right much better than Euler.

There is no I mean hardly any comparison and of course, I have maintained d x equal to 0.02 to be able to compare right which is for Euler was this black curve. But, let us see let us blow up the data and see whether there are any deviations at least here.

(Refer Slide Time: 01:25)

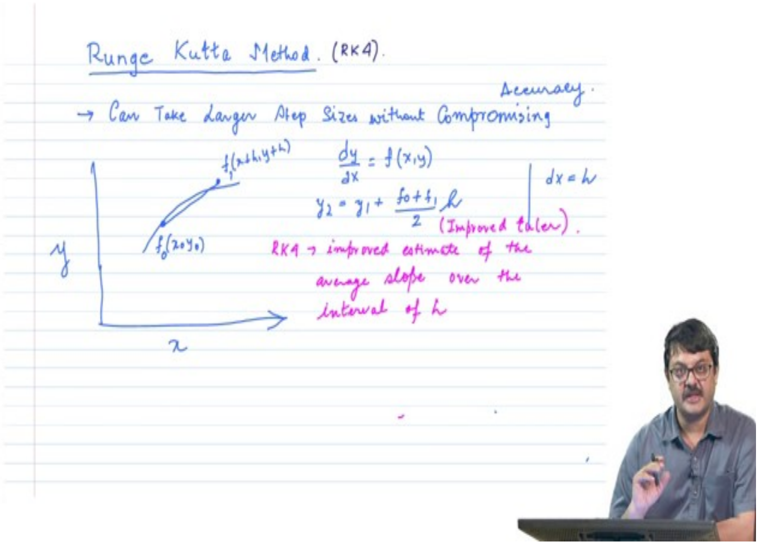


As you zoom in on the data, this is your blue curve which is theoretical analytical tan x and modified Euler has results closer to the analytical value, but improved Euler is even better right, it is going more closer and closer to this thing at the with the analytical value. But still there are deviations; deviations are being picked up as tan x rapidly goes towards infinity, not good enough. Now the next thing you will be learning is the Runge Kutta method and if you use the Runge Kutta method with the same value of dx in this purple curve is the data is the Runge Kutta RK4 integration scheme which we shall discuss.

You see how good the matches with the analytical one even at 1.54, you can also extend it to 1.55 as I was saying it in the lectures, but definitely this Runge Kutta is much better, no doubt than the Euler keeping dx the same. This is the Euler, this is the modified Euler, this is the improved Euler and you can see for yourself what is the value of the Runge Kutta 1 with it really matches very well, much better than the others as x approaches 1.57 pi by 2. Of course, these are not drawn for a pi by 2 exactly because it will be infinite I can plot it, but even close sums even at x values close to 1.57, it is significantly better.

So, next let us learn about the Runge Kutta method which is distinctly much better and typically for real problems you use the Runge Kutta method for solving differential equations right. Euler's methods is just an introduction to see how we are adding more terms to increase the accuracy starting from the Euler method next. So, that is what we discuss next.

(Refer Slide Time: 03:57)



So, coming to the Runge Kutta method and to discuss what is the algorithm of the Runge Kutta method. So, we will be discussing one version of it which is called Runge Kutta 4 RK4, why 4 because though I will not be talking about the derivation of it. But this RK4 method is accurate two orders of h to the power 4 h being the integration dx; I mean the step size basically.

So, if you take a small dx so, say dx equal to 0.01, then the accuracy of your solution how accurate it will be, how much will it match with the actual solution will be 0.01 to the power 4 right. So, that is the accuracy of it. And you see that in the Runge Kutta method, this is significantly more accurate than the Euler method and the improved Euler method where the accuracy is order of h square right which is this where h is the step size. Now before we discuss the Runge Kutta method, let me tell you what is the principle of it; I mean what is the broad idea of it rather than the steps.

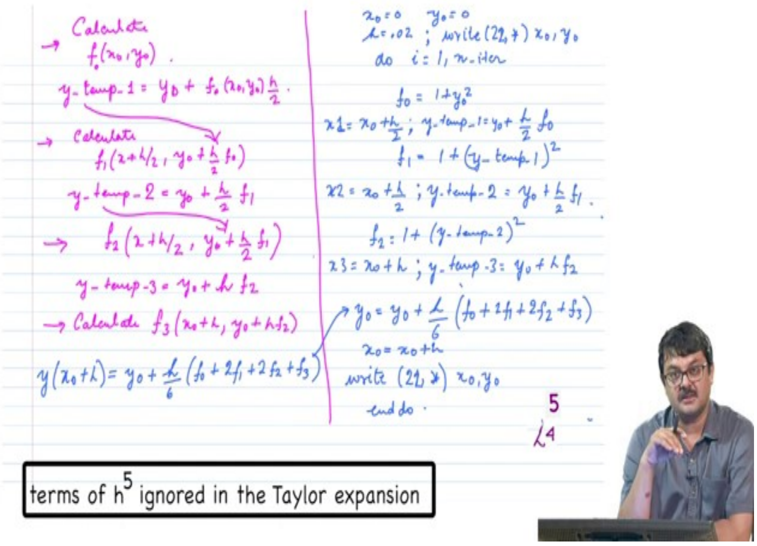
For that let us look back upon the improved Euler method and this was y and this was x and you had you have to integrate out dy dx equal to f xy. And suppose this curve; this curve is the y curve right and you have to start from sum x 0 and y 0. So, you have to tell the range of the over which we are going to integrate and to get the different solution to the differential equation. And at x 0 and y 0 suppose you can calculate the value of fx y. Now what you did in improved Euler was you did a simple Euler step and updated the value of x 0 to x 0 plus h. Similarly correspondingly you updated the value of y. With this at this updated point you calculated f 1 again right so, which is a different point. And then finally, to calculate the actual value of y 2 at the next step, you use the average of this point and this point right the average of these two slopes dy dx equal to fx y.

So, you took the average of this slope and this slope right and then got the actual update to the next value. Now in Runge Kutta what you do is use the slope or f 0 so, as to say at this point moreover calculate the slopes at multiple points between the interval of x 0 and x 0 plus h ok. So, you try to have a better and better estimation of the value of the slope at different points in the interval and remember RK4 is accurate to h to the power 4 right; remember 4 times 4 orders. Hence you can easily take larger step size without compromising the accuracy of your calculation.

So, you can take largest step size and if you have a large step size, then over that step size basically the value of f x y is going to change. Hence, you are trying to have an average a more accurate estimate of this value of f x y at different points over the interval take an average. Take a weighted average over it and then finally, calculate the value of y at the next time step. So, here in the improved Euler we were doing one step one Euler step, then taking an average over the slope here and the slope here the value of fx y at this point and at this point. In Runge Kutta, you want to do it do this step multiple times you are going to calculate it over different points over the interval.

So, this is the basic principle. So, that you have an even better average a better estimate of f x y the average value over the interval ok. So, the derivation of Runge Kutta why and how it is accurate to h to the power 4, you can look up the book of has been and defies which are which is already listed, but here we shall more focus on the implementation of the code. Just like RK4 is accurate to h to the power 4, you can also have RK is 8 which will be more accurate to order of h to the power 8, but then you have to basically average over this f x y what larger number of points .

(Refer Slide Time: 09:42)



So, let us go to the details of the algo and what you have to do in RK4 of course, start and calculate the f 0 at x 0 y 0 that is the initial point, you have the function put in the values of x 0 and y 0 calculate the value of f 0. After that you have to update the value of x 0 and y 0 by an interval h by 2. So, that you calculate a temporary value this is not your final value, this is basically an exploratory value is just doing an Euler step so, y temp 1.

So, some new value of y, but it is a temporary variable hence I have called it y temporary 1; 1 because we are going to do this multiple times equal to y 0 plus f 0 which is calculated at x 0 and y 0 into h by 2. Note the interval is half the interval h is the total interval. So, you want to calculate the value of y finally, at 0 h 2 h 3 h 4 h and so on so forth over the entire range over, you want to calculate over. But here you are doing an Euler step over half of the interval.

So, you have an estimate of what the value of y could be right at half of the interval and with this point with this y temp 1, this estimate of the value of y you again calculate the function f ok. But now I am calling it f 1 because previously when you calculated it at x 0 and y 0, I called it f 0, but now at f 1 because you have a temporary estimate of y temp 1, it is using that right and x 0 plus h by 2, you calculate f 1 ok. Now using this f 1, you again update the value of y.

So, this is again a temporary estimate right. So, you are starting out with so, from here you basically try to I have an estimate at the interval h by 2, then you calculate the f x y at this point which I you call f 1. Now with this f 1 again you make an estimate so, previously you used this f x f 0. Now you are using f 1 this average a better estimate of the slope to again get an estimate of y 1 over an interval h by 2 starting from here.

So, what you are doing is y temp 2 is y 0 again the initial point plus h by 2 into f 1 ok. So, you have a second estimate for what could the value of y b at an interval after an interval h by 2 using this y temp 2, again you calculate the local slope the local value of f. So, you are calculating f 2 x plus h by 2 so, the value of f 2 at x plus h by 2 and y 0 plus h by 2 f 1. So, this has been calculated here this has been calculated here, now you are updating this value of f 0. So, you are basically calculating the slope at different points in the interval getting better and better estimates right with this f 2, again calculate an estimate of the value of y; this is not your final y, it is an estimate of the value of y.

So, y 0 plus now you see that basically the interval is now over h, but this is not your final y, this is still an estimate. So, we are using this f 2 right which was calculated here to update y and to have an update of the value of y. This is again temporary. So, this y 0 plus h the interval is over the entire interval right. So, basically what you are doing using the value of f 2, you are trying to have an estimate of the value of y over the entire interval so, somewhere here. So, similar to improved Euler, but now you are using better estimate of the value of f x y right.

So, your actual value the final updated value of y at x 0 plus h is y 0 plus h by 6 right and weighted average of f 0 plus 2 f 1 plus 2 f 2 plus 2 f 3. So, f 0 is the one calculated at right at the beginning, then you using f 0 you updated the y 1 by h by 2. Using this y 1 using this new y temp 1, you calculated f 1 right here at interval h by 2. Now using this slope using this f 1, you again updated y 1 you again updated y over an interval h by 2 to calculate y temp 2. With this value calculate f 2 and you are storing your f 0 f 1 f 2 each time. Using this f 2 update another temporary value of y, but over at interval h right and then at that point calculate f 3.

So, you basically at different values of y you are having an estimate of f. You taking a weighted average and this is your final y. Taking this you can again calculate it at y at x plus 2 h, but again you have to do these 4 steps. So, this seems to be much more involved than your improved Euler where you have to do two steps, here you are doing 4 steps, but what you gain right though this is computationally more expensive what you gain is you can use higher values of h. So, suppose you had to use for a particular complex function which fastly varying f x y a complicated f x y you had to use suppose h equal to 0.01 right.

And you would get an accuracy in the improved Euler case of point of h square right whereas, here you have an accuracy of h to the power 4 which is much better right. And so, even if you use h equal to 0.1 you have the similar accuracy or rather and more accuracy right. So, h if you took h equal to 0.01, then your accuracy with the improved Euler would be 0.0001. So, like x square whereas, for improved with Runge Kutta even if you use h equal to 0.1 so, you are using a large interval you would have a similar accuracy.

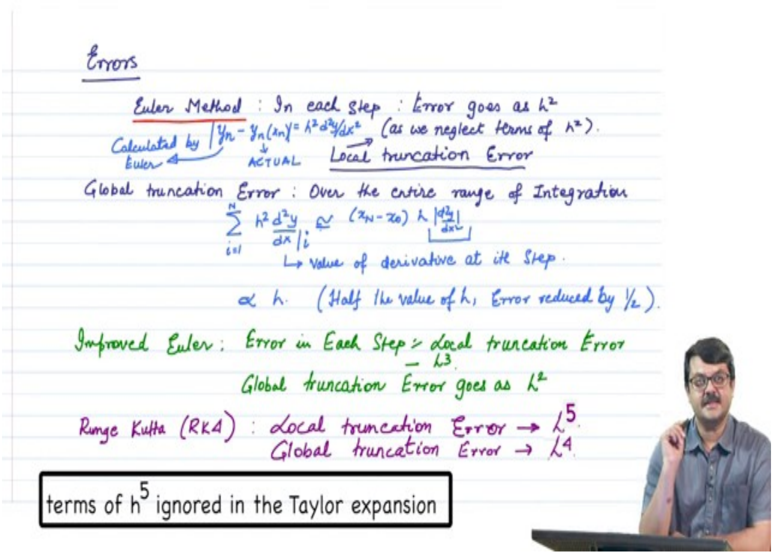
So, you are saving 4 around 10 updates in that Runge Kutta you are just going from steps of 0.1. How would you implement it? So, basically I am showing the raw code and you should just try it out and this is suppose x 0 equal to 0, y 0 equal to 0; we use the same function which is 1 plus y square right. And suppose you want to integrate it from 0 to 1.55 or 1.54 just like that and you want to plot the function the solution of the differential equation. So, h equal to 0.02 say and write 22 the initial values. So, the value of x and y right later you are going to plot x versus y.

So, you will have at different values of x different, you will basically write down and calculate and write down different values of y and you are going to plot x versus y that is what we did yesterday. So, here do I equal to 1 to number of iterations. Suppose if it is 1.55 or 1.54, then you can calculate the number of iterations will be 1.54 divided by 0.02. So, that is your increase in step size. So, f 0 calculated is 1 plus y square using that I am updating x 0 1. So, I am calculating x 1 essentially here, this is x 0 plus h by 2 and then updating y here calculating f 1 here using this value. We are not using x 1 in this function, but you could have a function where you have to use x 1 as well which is updated to x 0 plus h by 2.

Now, using this f 1 you have to update again you are updating x 2, but starting from x 0 its at this interval h by 2 which we are not using and updating y here right. It is the same as this line. Starting from y 0 and we are updating this then we calculate f 2 right using this value of y temp 2 that we have calculated, then we calculate in the next step using f 2; we calculate y temp 3 which is nothing, but this step and also update x 3 equal to x 0 plus h. Though we do not use it to calculate f in for this particular function and finally, you your final corresponding to this step is y 0 equal to your new value of y 0 is y 0 plus this weighted average right.

Update x 0 to x 0 plus h. So, write down suppose in some file 22 x 0 and y 0 and this goes on and on. So, x 0 keeps on changing from 0.02 0.04 0.06 0.08 and every time using this step, you have an updated value of y 0. The value of y 0 at 0.02, 0.04, 0.06, 0.08 and so on so forth and use this to do the entire step. So, this is your new value of y 0 thereby you can doing these 4 steps; you can calculate y 0 at x equal to 0.04 right.

(Refer Slide Time: 21:33)



So, we shall end this class with a discussion of the errors involved in each of these methods. So, in the Euler method if you remember that we are neglecting terms order of h square; so, d 2 x d 2 y dx 2 into h square the higher order terms, the second order terms we are neglecting right. So, in each step; so, in each step we are neglecting errors of h square. So, that is the error that is being introduced. So, in each step when we calculate the new value of y right, we neglect this step and so this is the so called local truncation error in each step right.

That is the order of and that is the and it goes as h square into d 2 y dx 2. Of course, at each point when you integrate each point when you calculate the value of y at each point the value of d 2 y dx 2 will be different, but the pre factor remains h square. And so, thereby the local truncation error is h square; however, to calculate the global truncation error that is after n steps of when you have basically calculate the calculated the value of y over n steps h plus h plus h plus h plus h plus h after n th steps, what will be the total error?

Now in each step you are making an error of h square d 2 y d x 2, but now we have to sum over all these errors over n steps right and when you multiply n h basically the error becomes the global truncation error over the entire interval becomes h into d 2 y d x 2 where d 2 y d x 2 ideally should be at each step, but I have written it as an average over different steps, but the global error becomes the difference in interval x and minus x 0 because you are summing it n h if you like into h into d 2 x d 2 y d 2 y d x 2 right.

So, the error over the entire interval goes as h this is pretty high right. But if you half the value of h your global truncation error will also reduced by half because basically when you have sum over rate this is n into h is this entire interval and h remains and so on so forth. Now you can look up the book for the derivation of improved Euler, but in improved Euler the local truncation error the error made and each step goes as h cube for Euler it was going as h square for improved Euler, it is going as h cube. And when you sum over the entire interval when you sum over the errors over the accumulated errors over the entire interval, it goes as h square.

And Runge Kutta was giving such good results even for relatively high values of h because for the Runge Kutta the local truncation error goes as h 5 again y h 5. I mean that is why you call it RK4, but the derivation is there in the box here we are focusing more on the implementation it goes as h 5s. And the global truncation error just as before goes as h 4 when you sum over the errors over the entire interval. So, of course, it is the margin of error. Here it goes as for Euler goes as h for improved Euler global truncation, it gives as h square and in RK 4 you see much more accurate values closer values to the actual function even with higher values of h.

So, with that we come to the end of this lecture. We have basically discussed Euler modified Euler improved Euler and Runge Kutta for practical purposes you shall always be using Runge Kutta. So, next class what we shall discuss is how to do coupled equations, the simple coupled equations and then we will slowly move to the non-linear case; how do we go about it.

Thanks.