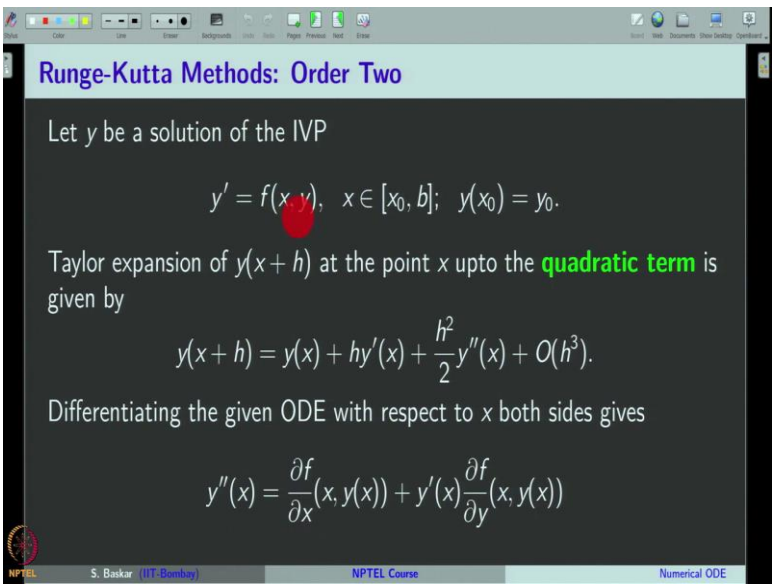
Numerical Analysis
Prof. S. Baskar
Department of Mathematics
Indian Institute of Technology - Bombay

Lecture - 58
Numerical ODEs_ Runge-Kutta Methods

Hi, we are discussing numerical methods for initial value problems, involving first order ordinary differential equations. In the last 2 classes, we have developed Euler's method and studied the convergence analysis also. Although, Euler method is easy to implement, this method is not so efficient in the sense that, to get a better approximation one needs a very small step size. This is mainly because Euler method is a first order method. One way to get better accuracy with less step size *h*, is to go for higher order methods.

Recall that in Euler's method, we considered approximation of the derivative by taking the Taylor expansion with degree 1 and truncated all terms of order 2 and higher. In order to achieve higher order accuracy, we will have to include higher order terms in the approximation of y'. This will naturally bring in higher order total derivatives of y, which is not preferred computationally. Runge-Kutta methods are focused to achieve higher order accuracy without involving higher order total derivatives of *y*.

(Refer Slide Time: 02:14)



This makes the methods more popular, till to date. Our interest in this method is to understand the rigorous derivation of the Range-Kutta method of order two. Let us recall that we are interested in devising numerical method for the initial value problems of the form $y' = f(x, y)$, for all *x* in some interval $[x_0, b]$, where $y(x_0) = y_0$. Now, let us use the Taylor approximation

for $y(x+h)$ about the point *x* where we now, retain the terms up to order 2 in *h* and truncate the terms of order 3 and above.

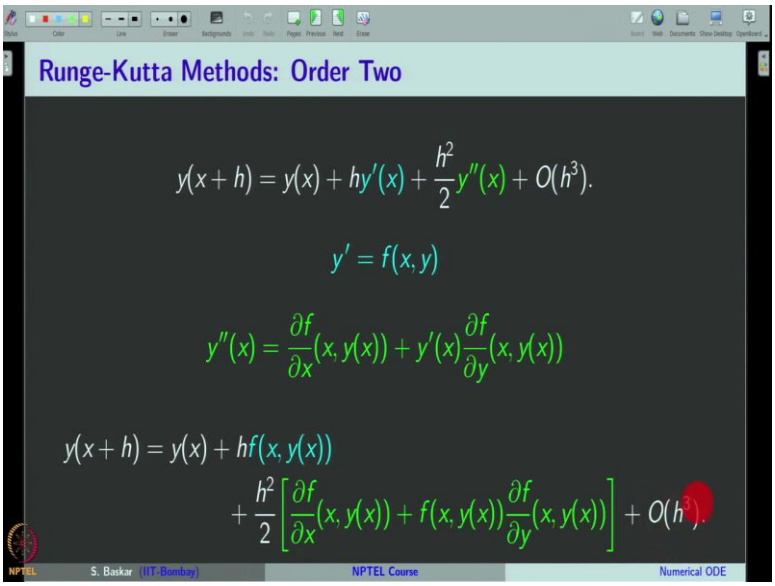
Well, we have to write the remainder term at the third order but it is just comfortable to simply write the $O(h^3)$, because anyway at the end of the derivation we are going to neglect these terms. Therefore, it is more comfortable for us to simply write these terms in the big O notation to just indicate that, whatever expression we have, after this term is going to involve something of order h^3.

Recall, in Euler method we have taken only the first order term and make this term with a remainder. That is how we got only the first order accuracy in Euler method. Now to achieve the second order accuracy, we are also including the second order term now into our expression and we are truncating the Taylor expansion only from order 3, that is the main idea of achieving order 2 now in the numerical method.

Let us see, how to go ahead with this. You can see that by including the higher order term, we have $y''(x)$. If you recall, we have substituted $y' = f(x,y)$, when we were deriving the Euler method. Now we have to similarly handle this second order term also. For that, what we will do is, we will differentiate the given ODE with respect to *x* and we get $y'' = \frac{\partial f}{\partial x}$ into $\frac{dx}{dx}$, that is equal to 1, + $\frac{\partial f}{\partial y}\frac{dy}{dx}$, that is y'.

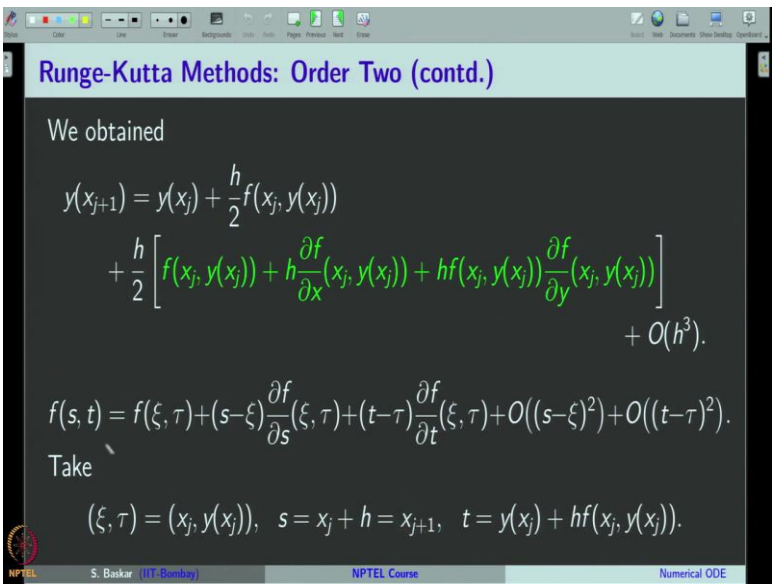
This is simply the chain rule that is applied on *f*. Remember, *f* is a function of 2 variables *x* and *y*, where *y* is a function of *x*. Therefore, we get this expression if you use the chain rule.

(Refer Slide Time: 05:41)



Therefore, we have the expression for y' which is coming from the equation and we have y'', which we obtained by differentiating the ODE once with respect to x. Now, we will substitute these 2 expressions into the Taylor expansion and see what happens. We get $y(x+h)$ is equal to $y(x)$ which is already there, $+ h$ into instead of y' will put $f(x,y) + \frac{h^2}{2}$, which is already there, into instead of y'' we put this expression here and then finally we are going to neglect all other terms which are of $O(h^3)$.

(Refer Slide Time: 06:40)



So, this is what we have achieved so far, using Taylor expansion and then substituting y' and y''. Now, let us write this expression in a slightly different way. What we will do is, we will split this term into 2 equal parts, $\frac{h}{2}f(x,y) + \frac{h}{2}f(x,y)$. One part we will retain separately and merge the other part into this third term and we get this expression, now as the third term.

We have not done anything we have just split this second term into 2 parts and merge the second part into the third term, that is all we did. What is the reason to do this small adjustment? This will be clear if you look at the Taylor expansion of a 2 variable function up to degree one. It will happen to be of this form but before doing this, let us fix our notation at par with the numerical method. For that, let us replace *x* by x_j and write the above expansion.

Now replacing *x* by x_j therefore $x + h$ will be $x_j + h$, and wherever *x* comes, we will put x_j and we are just writing the same expression which is written here, now here with *x* replaced by x_j. So, we have this expression for $y(x_{j+1})$ with us. Let us now see, how to handle this term because this term involves partial derivative of *f*, which are not so comfortable from the implementation point of view.

Well, we have to provide all this information into our code, in order to compute the solutions and that is not very comfortable, in general. Let us see if we can somehow replace this expression with something, which can be handled computationally. For this, let us first recall the Taylor representation of *f* about the points (ξ, τ). Let us take some point (ξ, τ) and let us recall how to write the Taylor representation of a 2 variable function *f* about this point (ξ, τ).

The value of *f* at some point (s, t), which is very close to (ξ, τ) can be written as $f(\xi, \tau) + (s - \xi)\frac{\partial f}{\partial s}(\xi, \tau) + (t - \tau)$, now this is coming for the second argument of $f\frac{\partial f}{\partial t}(\xi, \tau)$. You can observe that this is just a straight forward generalization of 2 two dimensional Taylor expansion. We just want the Taylor expansion up to order one.

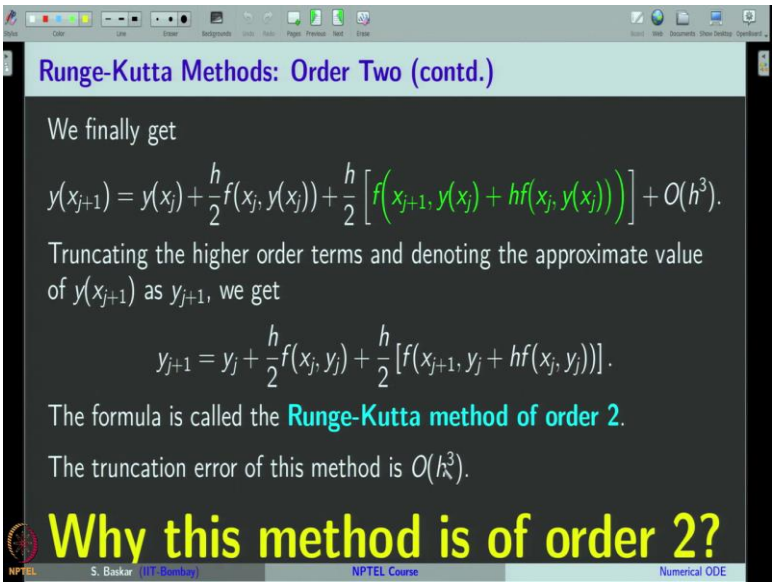
Therefore, from the second order term onwards we will simply write in the big O notation, because at the end, anyway we will try to absorb this second order term into the term which we already have. Why it is so? We already have a *h* here. Now we will try to see that $(s - \xi)$ is of order *h*. Therefore, its square will be of order h^2 into *h*, will be of order h^3. Already we made our mind to neglect all the terms which are of order h^3.

Therefore, whatever may be the terms which are sitting, after this will finally be neglected in our formula. Therefore, we will not write them explicitly or write as remainder but we will

simply write them in the big O notation. Now let us take (ξ, τ) equal to $\left(x_j, y(x_j)\right)$ and look at this expression in the bracket and compare it with the expression here. We can see that if you take *s* is equal to x_{j+1} and *t* is equal to $y(x_j) + hf\left(x_j, y(x_j)\right)$ then this term will match exactly with the term in the bracket.

Therefore, we will take these expressions and put them in this and then replace $f(s,t)$ where *s* is x_{j+1} and *t* is this expression. We will replace this entire expression by *f* of this. So, that is the idea, this is how you are getting rid of these partial derivatives of *f*, which we saw that they are not very comfortable from the computational point of view and they are just going off from our scene by simply replacing this expression by $f\left(x_{j+1}, y(x_j) + hf\left(x_{j+1}, y(x_j)\right)\right)$.

(Refer Slide Time: 12:59)



If you do that then you will have $y(x_{j+1}) = y(x_j) + \frac{h}{2} f(x_j, y_j)$, up to this we already have here. Now what we are doing is, only in the third term we are simply replacing the expression inside the bracket by *f* of this using this, Taylor's expansion of a bivariate function. And that gives us the third term as $\frac{h}{2}\left[f\left(x_{j+1}, y(x_j)\right) + hf\left(x_j, y(x_j)\right)\right]$ +, of course, we have some higher order terms which we anyway made our mind to neglect them.

Therefore, you just simply freeze them into this notation and keep and finally we will simply neglect this higher order terms and write the formula. And just to denote that, by doing this we only get an approximation to this solution. We will use the notation y_{j+1} in order to denote the approximate value of the exact solution $y(x_{j+1})$. Therefore, we have this formula y_{j+1} which

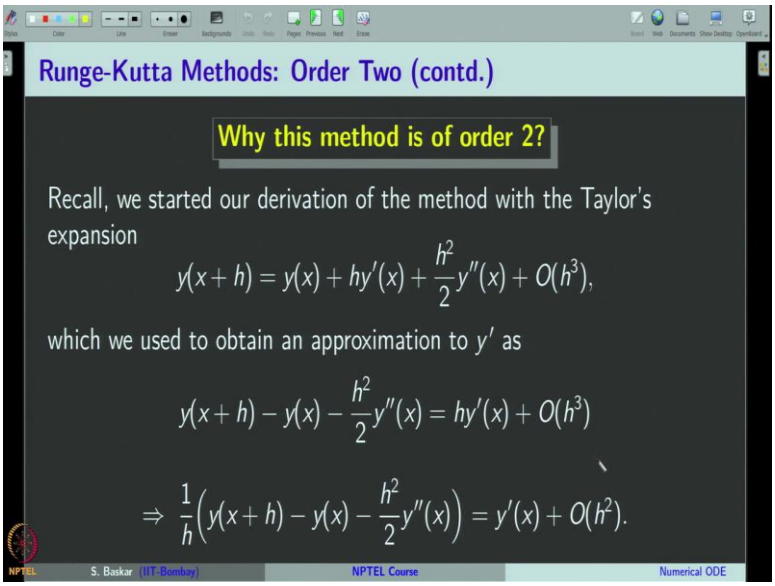
is the approximate value of $y(x_{j+1})$, obtained using this formula where all the exact values are now replaced by the approximate values of the solution.

Otherwise, this expression is exactly the same as this expression. Just we have neglected all the higher order terms and this is precisely the Runge-Kutta method of order 2. It is very nice and interesting to see how we have achieved the higher order without using the partial derivatives of *f* with respect to *x* and *y*. We have cleverly eliminated these partial derivatives, simply using the Taylor's expansion of a bivariate function up to order one.

So, that is the idea behind the Runge-Kutta method of order 2. What is the truncation error of this method. Well, the truncation error of this method is something of order 3, that is what we can see. In fact, you can easily write this expression. It is simply the remainder term in the Taylor formula. Now the question is, why this method is of order 2 when the truncation error of the method is of order 3.

Well, you can easily answer this question if you have understood why we had Euler method as of order one. If you recall, the truncation error in Euler method is of order 2 but the Euler method is of order 1. As far as the formulas that we derived so far for the first order ODEs are concerned, the order of the method will be 1 less than the order of the truncation error. Let us see why it is so, in the present case.

(Refer Slide Time: 16:43)



Recall that we started our derivation of the method with the Taylor's expansion given like this with the truncation part being order h^3 and this is finally used in obtaining an approximation of

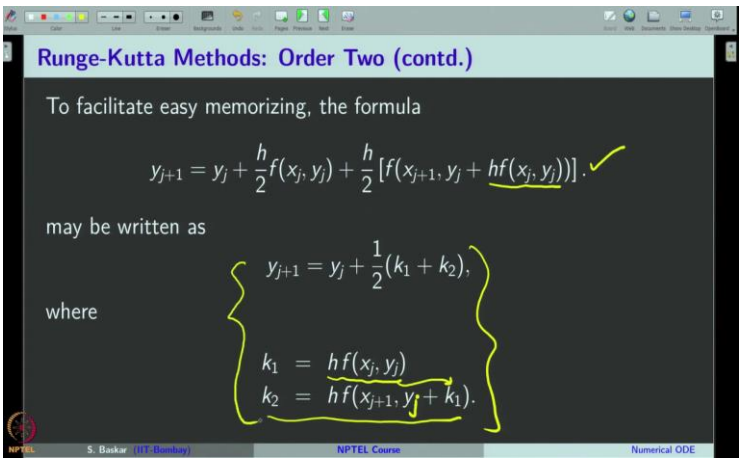
y'. So, let us write this expression in this form where you bring all the terms on the left hand side, keeping only the y' term on the right hand side plus of course the terms which you are going to neglect.

Now in order to get y' on the right hand side, what you have to do, you have to divide both sides by h. That is what we are doing. So, in order to get y' on the right hand side, from here we are dividing both sides by *h* and that takes away one order from your truncated part. So, that is why the method is order 1 less than the truncation error order. This happens with the methods that we have derived so far, that is, it happened with Euler method also and also it happened with the Runge-Kutta method that we have derived so far.

Because we are approximating y' and in the Taylor's formula, y' appears with *h*. Therefore, in order to get an approximation for y', finally you have to divide both sides by *h*. That will naturally reduce one order from the truncation error. Therefore, you should always remember that in such methods, if your truncation error is of order something say *n* then the order of the method will be one less, that is it will be $n - 1$.

Here, the truncation error is 3 therefore the order of the method is 2. Similarly, in the Euler method the truncation error was of order 2. Therefore, the method as such is of order one.

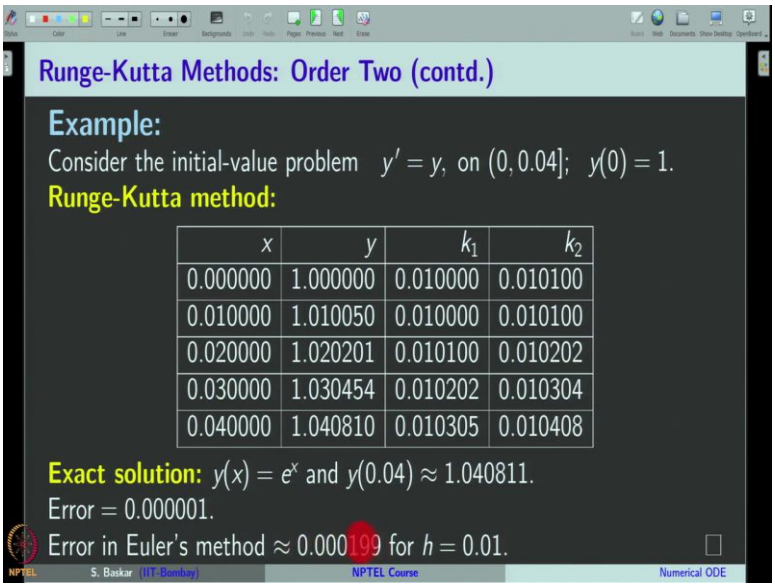
(Refer Slide Time: 19:03)



Now let us put the Runge-Kutta method of order 2 in a slightly simple form, which is more easy for us to remember generally. How to write it? Recall, the Runge-Kutta method of order 2, that we have derived in our previous slide, is given like this. Now what we will do is, we will write it in a different form which is easy for us to remember. How to do that? Well, you can write this expression as $y_{j+1} = y_j + \frac{1}{2}(k_1 + k_2)$, where $k_1 = hf(x_j, y_j)$.

Now you can see, that expression is sitting here. Therefore, k_2 can be written as $hf(x_{j+1} + y_j + k_1)$, this part is simply now named as k_1, therefore you can simply plug in k_1 here and get this expression. Well, this is the same as this. It is only written in a different form. Generally, it is also easy for us to implement as a computer code and also from the examination point of view this may be easy for us to remember.

(Refer Slide Time: 20:43)



Since the method that we have derived is of order 2, means its truncation error is of order 3. Remember that the Euler method is of order 1 and therefore its truncation error is of order two. So, we gained one order in the Runge-Kutta method we derived so far and therefore for a given *h*, we expect better accuracy from the Runge-Kutta method of order 2, when compared to the Euler method.

Let us see this by an example. Let us consider the initial value problem $y' = y$ on the interval (0, 0.04] with the initial condition as $y(0) = 1$. It is a very simple linear ordinary differential equation and therefore we of course know the exact solution of this initial value problem. Now, let us compute the numerical solution of this initial value problem using the Runge-Kutta method of order 2.

And let us present the numerical result in this table. We start with $x = 0$ and we are given that at $x = 0$, y takes the value 1. Therefore, we do not do any computation at $x = 0$. We simply put the initial condition here and now we have to find k_1 and k_2 using the formula shown in the

previous slide and they are given by 0.01 for k_1 and 0.0101 for k_2. You can check this and once you have k_1 and k_2, you can plug in that into the formula of y_1 and you can get it as 1.01005.

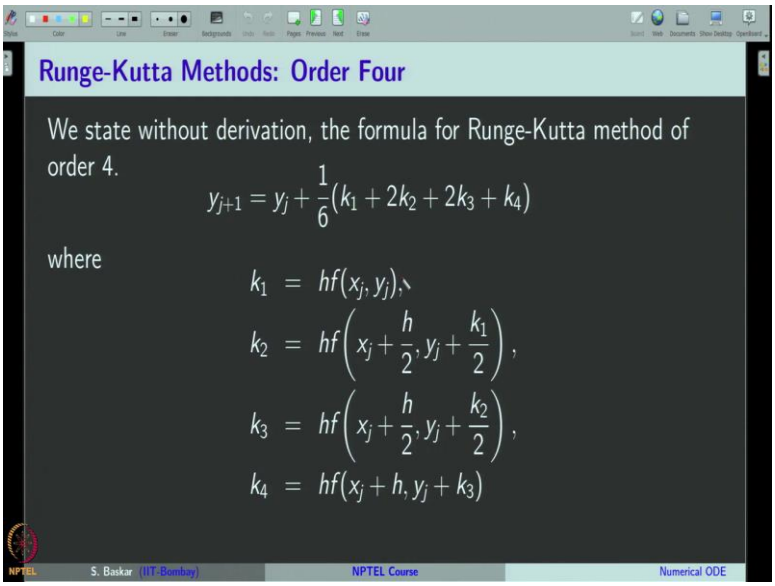
And once you have y_1, you can get again k_1 and k_2 and then you can obtain y_2 from there. Remember, you have to use the formula that we have displayed in the previous slide. I hope, I do not need to show you the calculation. You can do it easily. Once you have y_2 then again put y_2 and x_2 and get k_1 and k_2. Once you have k_1 and k_2, you can go to get y_3. So, like that you can keep on going till you reach the right end limit of the domain, that you are interested in.

Here we have taken it as 0.04 and we have taken *h* as 0.01. From this column, you can see that we have taken $h = 0.01$. What is the exact solution? The exact solution is $y(x) = e^x$ and the value of the exact solution at the point $x = 0.04$ is given by this number approximately, at least up to the six significant digits and what we obtained as the numerical solution from the Runge-Kutta method of order 2, we obtained this value.

Now what is the error? Error is something nearby 10^{-6} and that is pretty good, when compared to the Euler method. You may recall that in Euler method, we computed the numerical solution of the same initial value problem with the same $h = 0.01$ and we got the error in the numerical solution of $y(0.04)$ as something nearby 2×10^{-4}. Therefore, you can see that we have a good improvement in the numerical method when it is of order 2, when compared to a numerical method of order 1.

At least in this particular example, we can clearly see the advantage of going for the higher order method when compared to the lower order methods.

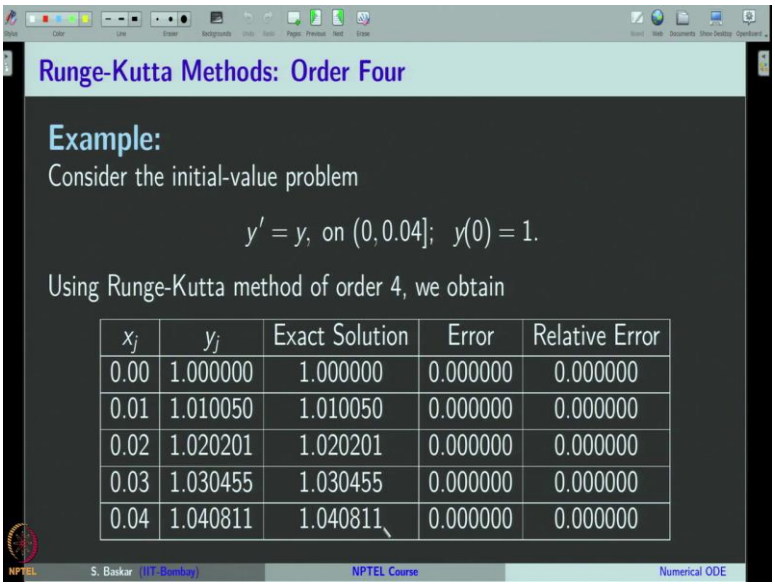
(Refer Slide Time: 25:27)



Once you understand the derivation of the Runge-Kutta method of order 2, the same idea can be adopted to derive the Runge-Kutta method of any higher order. The idea is to retain adequate number of terms in the Taylor's expansion and then you have to use the bivariate Taylor's approximation accordingly. For instance, the Runge-Kutta method of order 4 is given by this expression where $y_{j+1} = y_j + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$, where k_1 is given like this, k_2 is given like this, k_3 is given like this and k_4 is given like this.

Well, the idea of the derivation of the Runge-Kutta method of order 4 is very much similar to what we did with the Runge-Kutta method of order 2. However, the derivation is very lengthy but it is straightforward. We omit the derivation of Runge-Kutta method of order 4 and just take this formula alone, for our computation. Let us compute the solution for the same initial value problem that we did in our previous example and see how the accuracy is in the present case.

(Refer Slide Time: 27:03)



Let us consider the initial value problem $y' = y$ on the interval (0, 0.04] with the initial condition as $y(0) = 1$ and now we will use the Runge-Kutta method of order 4. You can see that at the initial point, you have the exact value because it is given to us and therefore there is no error involved in the numerical solution, when compared to the exact solution at the initial point.

Then when you go to compute the solution at the later points, we are expected to accumulate some error. At $x = 0.0$, remember, again we are taking $h = 0.01$. Numerical solution is given like this and it is nice to see that exact solution is also the same, at least up to the six significant digits, that is shown here therefore the error is zero. Now let us go to the next grid point 0.02. Here also, you can see that up to six significant digits, we do not have any error

Similarly, for $x = 0.03$ and even for $x = 0.04$, we got almost the exact solution, at least up to six significant digits. This is very nice to see and also gives us a clear understanding that the Runge-Kutta method of order 4 is better that the Runge-Kutta method of order 2. In fact, it is in general true that as you go on for higher order methods, you tend to get better results. But this may not hold for all the problems, for all the methods. But this is a general idea. With this note, let us finish this lecture, thank you for your attention.