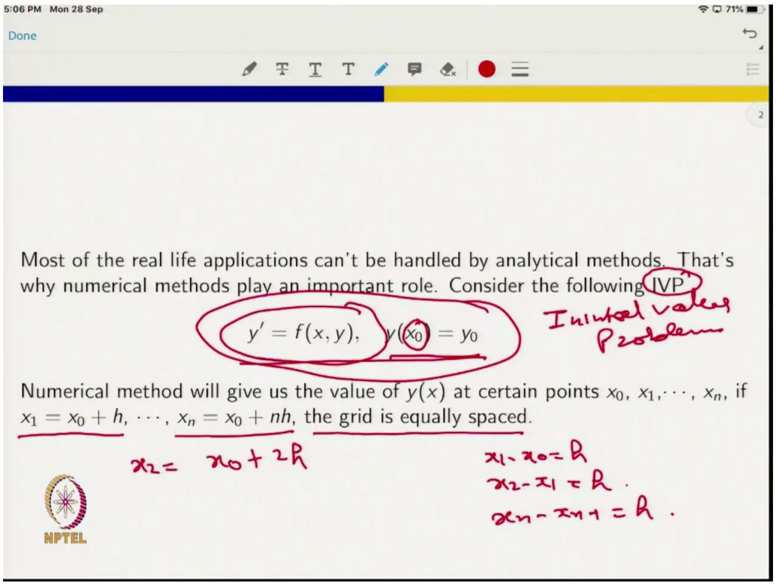
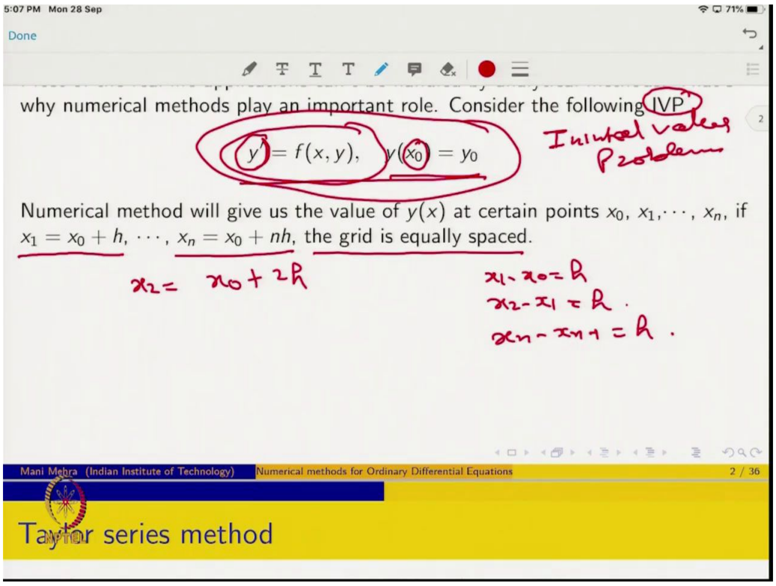
Scientific Computing Using MATLAB
Professor Mani Mehra
Department of Mathematics
Indian Institute of Technology, Delhi/DTU
Lecture - 61
Taylor Series Method for Ordinary Differential Equations

Welcome to the class of Scientific Computing Using MATLAB. I am a second instructor of this course and you have already seen a lot of topics in numerical analysis, approximation, etcetera.

So now, I will be teaching you numerical methods for ordinary differential equations. At the same time, there will be a lot of concepts, which will be used which you have already done in the previous part of this course.

(Refer Slide Time: 00:59)





So just let me start with a very simple differential equation, which is

$y' = f(x, y), y(x_0) = y_0$. This is first order differential equations and so, most of the real-life applications cannot be handled by analytical methods; that is what we all know. That is why numerical methods play an important role.

So first of all, we should understand why differential equations are useful. In fact, I do not need to explain to you much about it because we already know the application of differential equations. There are many physical quantities in real-world applications, which we want to trace.

So with the help of this differential equation also we are looking at the behavior of y with respect to x. If you look at it as a real-world problem, y is some physical quantity and x is someone another physical quantity. If you look at it from a mathematical point of view, y and x are, y is a dependent variable and x is an independent variable.

So in real-world applications, whenever we want to see the behavior of many physical quantities with respect to some other quantity, which is x here, so, in that case, we model real-life applications with the set of differential equations.

Of course, all of us know to solve differential equations. We need initial conditions or some set of conditions. If this is a first order differential equation and only one condition is prescribed at one point, which is an initial point. If it would have been a second order differential equations, we need 2 conditions to solve that.

And, if those 2 conditions are prescribed at the one point, that is called initial value problem, and if those two conditions are prescribed at two different points, they are called boundary value problems.

So, I have made a clear distinction between initial value problem and boundary value problem for a second order differential equation. But here just for simplicity, we are considering the only first order differential equations together with initial condition. That is why, altogether, this is called initial value problem, initial value problem and in a short form, it is called IVP.

We have seen different variants of a differential equation as well, if $f = 0$, we call it as a homogeneous differential equation; if it is non-zero, then we call it as a non-homogeneous differential equation. And depending on $f(x,y)$, we can also call it a linear or nonlinear differential equation; that is altogether a different ball game.

But right now, our aim is to solve this initial value problem by numerical method. But, before going to the domain of a numerical method, we should, first of all, make sure, whether we can solve this initial value problem analytically or not. We should solve means, whether there exists a solution, as well as this solution, is unique.

So all of you must have seen the existence and uniqueness theorem for the initial value problem. So under that theorem, we put some conditions on f, and under those conditions of f, we guarantee that the solution will exist as well as will be unique. So before going to the domain of a numerical solution, we should make sure that solution exists as well as solution is unique.

After that, we try to solve it numerically. Because if there does not exist a solution and we are trying to solve it numerically, it is a waste of time. At the same time, if there are infinitely many solutions and we are trying to solve it. So sometimes, we will get one solution, other times we will get another solution, and we will be not sure, which is the solution.

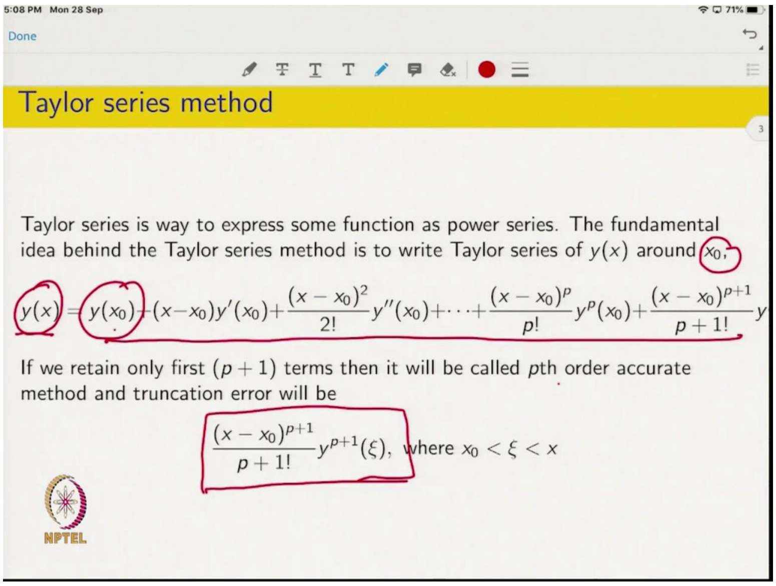
So numerical solutions play an important role, once you are sure that this problem has a solution. So because there are many theorems in numerical analysis as well as functional analysis. With the help of those theorems, we can predict whether a solution exists and, we do not know the exact form of a solution, but at least, we can predict whether a solution exists and the solution is unique also.

So now, we are considering the following initial value problem. So what we do initially, we discretize, because the initial point is given to us, x_0 is an initial point. From x_0, we have to find the solution at x_1, which is at . And then, we have to find a solution at x_2, which is again $x_0 + 2h$, and $x_n = x_n + nh$.

Again, just for simplicity, we are considering that the grid is equally spaced. Means, $x_1 - x_0 = h$; $x_2 - x_1 = h$, and $x_n - x_{n-1} = h$. Means that the difference between each grid point is the same. We can also work for a non-equally spaced grid but that is, we will see later on. Right now, for simplicity, we are considering an equally spaced grid.

So, to solve this differential equation numerically, what is the first step we should take? This is, y' is a derivative. So we have to approximate this derivative first. We have to approximate this derivative first. You must have seen a lot of methods to approximate derivatives earlier in this course, like forward Euler, backward Euler, central difference to approximate the derivative, etcetera.

(Refer Slide Time: 07:50)

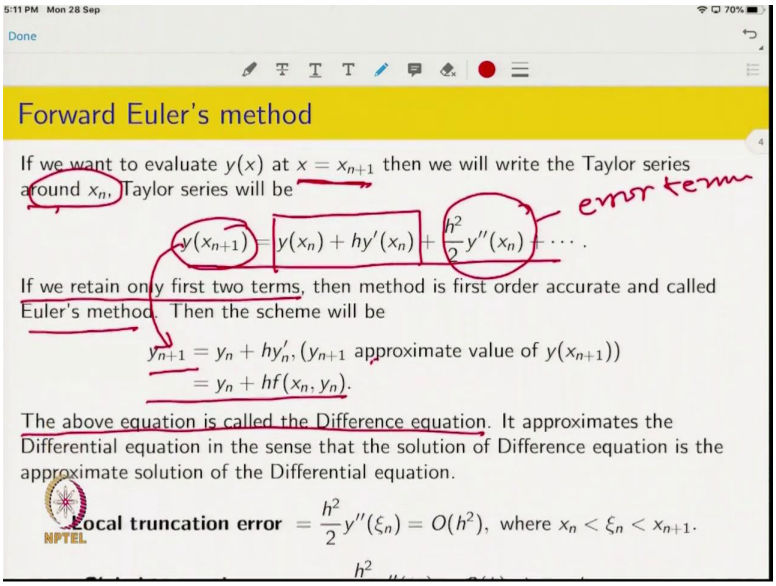


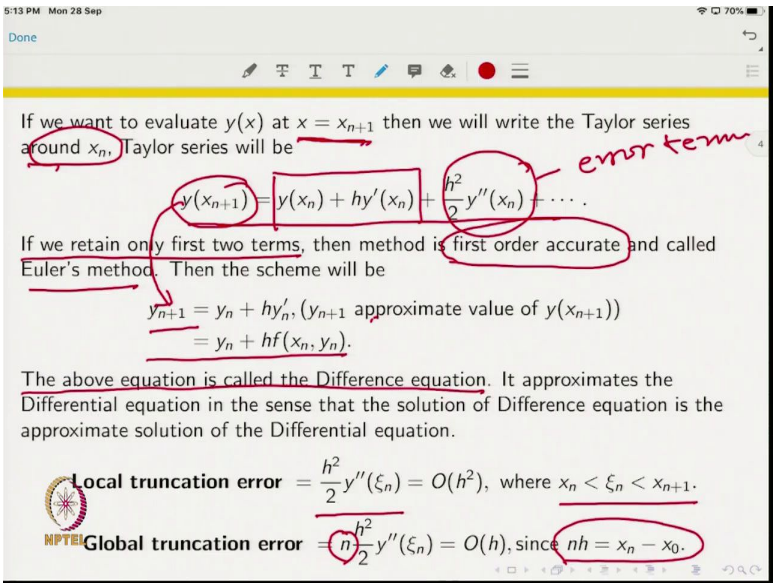
So we will be recalling those concepts in due course of time. But before that, let me start with the Taylor series method, which is a, which one could see as the oldest method to solve an initial value problem.

So in the Taylor series method, what do we do? We express y(x) with the help of Taylor series. We write Taylor series of a function y(x) around x_0. We write the Taylor series around y(x) around point x_0. So if, that is the fundamental idea behind the Taylor series.

If we retain only first initial terms, then it will be called pth order accurate, in not initial term, if we retain only first p+1 term, then it is called first pth order accurate method and truncation error will be this. Then it will be called pth order accurate methods, where ξ lies between x and x_0. I am not going into the detail of explaining you, Taylor series because all of you know Taylor series already.

(Refer Slide Time: 09:05)





So let me start with the Forward Euler method, which is a special case of a Taylor series method. How is it a special case of a Taylor series method? So in forward Euler method, what we are doing basically if we want to evaluate y(x) at $x = x_{n+1}$, then we are writing the Taylor series around point x_n, and this Taylor series will be this.

So basically, just if you have seen the previous version of a Taylor series, we have substituted $x = x_{n+1}$ here. So in forward Euler method, we are retaining only initial 2 terms, which means we want to retain only these 2 terms. If we retain only the first 2 terms, then the method is called first order accurate.

Why is this method called first order accurate? That I will explain to you in a while and this is called Euler methods. Then the scheme will be, you know, there is a lot of difference between going from here to here. Because from here to here, what is the difference you observe? We have dropped this error term. We have dropped this error term. So $y(x_{n+1})$ is an exact solution of a differential equation at point $x = x_{n+1}$, while y_{n+1} is a solution of this difference equation.

There is a difference between differential equation and difference equation because we obtain the difference equation after neglecting this error term. So basically, the solution of the difference equation approximates the solution of a differential equation. That is the whole idea. So y_{n+1} is a solution of a difference equation, while $y(x_{n+1})$ is a solution of differential equations. So that is what I wanted to say to you.

The above equation is called the difference equation. It approximates the differential equation in the sense that the solution of the difference equation is the approximate solution of a differential equation. Here, we have kept only the initial 2 terms, this and this, and we have neglected this term. After neglecting this term, we come to this step, which is a difference equation. So that is the whole point behind forward Euler methods.

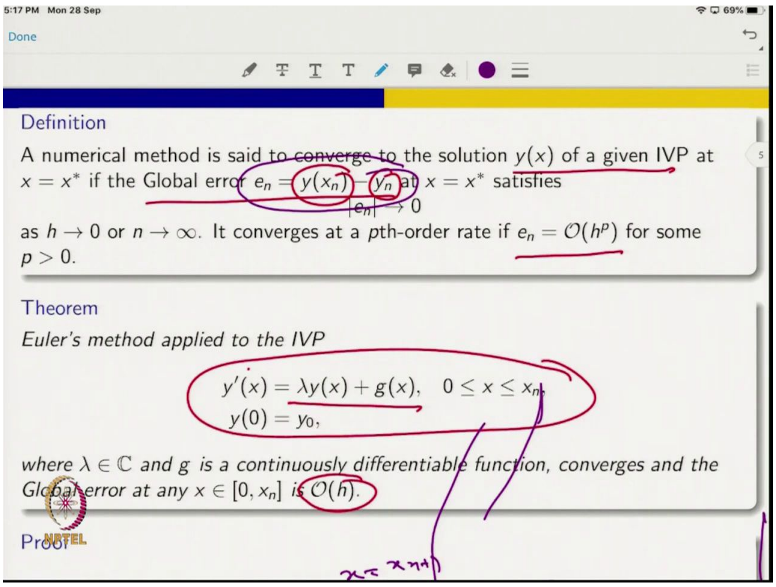
So what is the local truncation error? Local truncation error will be this. Where ξ_n lies between x_n to x_{n+1}. And global truncation error, because we have started from x_0 and we wanted to compute till x_n. So we have accumulated this local truncation error n times. That is why we are multiplying this global truncation error with that number n.

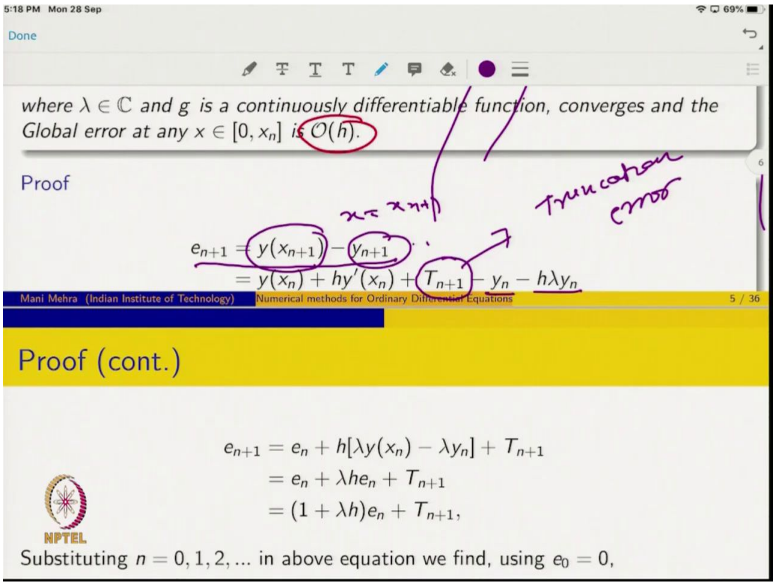
So nh is again a constant, which you can observe from here. So that is why global truncation error is $O(h)$. So basically, always the order of global truncation error will be 1 less than the order of local truncation error. And method is always known with the order of a global truncation error, not with the local truncation error. That is why it is called a first order accurate method.

That is what I said that I will explain to you later, how you define it is a first order accurate method. Because we define the order of the method as the order of a global truncation error. So this is called the forward Euler method. Why is it called the forward Euler method? Because Euler is the name of a scientist, who invented this method. Why is the forward word coming? Because we are going one step in the forward direction. We have started our things from x_0 and we are computing at x_1.

And then, once we know the solution at x_1 then we will go to the x_2. So we are moving in the forward direction; that is why it is called the forward Euler method. I have also explained to you, how you will define the local truncation error, how you will define the global truncation error.

(Refer Slide Time: 14:01)





So now, let me give you the formal definition of this. A numerical method is set to converge to a solution of y(x) of a given initial value problem at some point. If global error is this, how do you define the global error?

So, global error $e_n = y(x_n) - y_n$ That is how we define the global error. So basically, global error, the maximum of global error should tends to 0 as $h \to 0$.

It converges at a pth-order rate if we say $e_n = O(h^p)$. So basically, $|e_n| \to 0$ as $h \to 0$ or $n \to \infty$. It converges at a pth-order rate if $e_n = O(h^p)$ for some p > 0. So this is a more formal definition, how you define the convergence of a numerical method.

So again, we look at this definition and using this definition, we will prove when the Euler method is applied to the following initial value problem. It converges and the order of convergence is first order because it converges with the first order. That is what you can see here.

So basically, the theorem is Euler methods apply to the following initial value problem

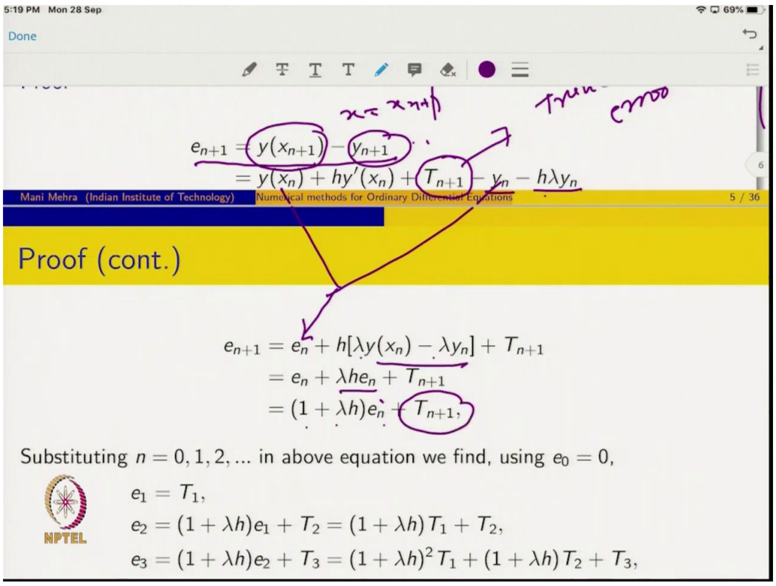
$y' = \lambda y + g, 0 \le x \le x_n, y(0) = y_0,$ where $\lambda \in \mathbb{C}$ and g is a continuously differentiable function, that is the conditions we are putting converges and global error at any x from 0 to x_n will be $O(h)$.

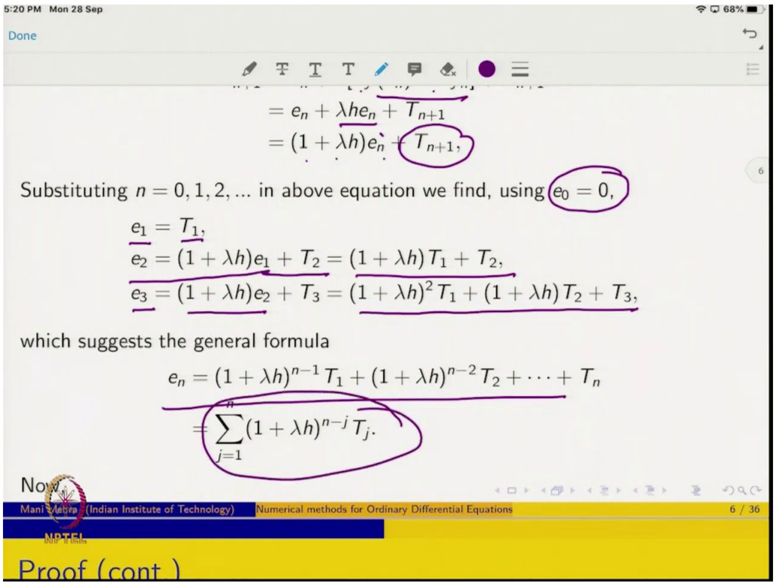
So basically, that we have seen earlier also that here, Euler method means forward Euler method, converges the order of method is first order. That is why global error will be order of

h. So here we will see more formally. Here, we have also taken just for simplicity, we have taken the specific value of f(x,y), which is $\lambda y + g$. So the proof goes in the following manner.

We define $e_{n+1} = y(x_{n+1}) - y_{n+1} = y(x_n) + hy'(x_n) + T_{n+1} - y_n - h\lambda y_n$. So here, basically, truncation error T_{n+1} plays a role. Truncation error decides the order of the method that we have already seen. So here, you see, I am expanding this y(x) as a Taylor series function around point x_n. And I also write what are the difference equations I have written.

(Refer Slide Time: 18:13)





So then, we do some specific calculation, $y(x_n) - y_n$ can be, so this, this end, this term jointly can be written as e_n and then I substitute the value of $y'(x_n)$, which is, will be $\lambda y(x_n)$. And then, I club this term with this.

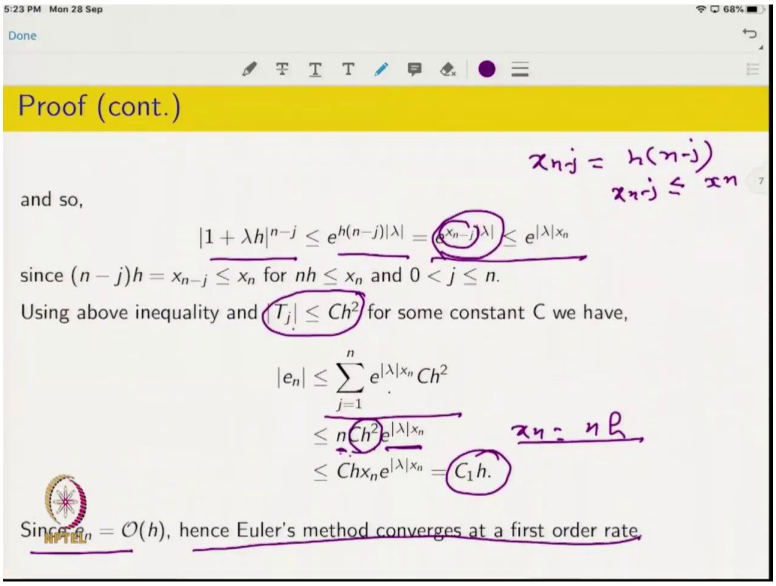
So again, by rearranging the terms, I get $e_{n+1} = e_n + h(\lambda y(x_n) - \lambda y_n) + T_{n+1} = e_n + \lambda h e_n + T_{n+1} = (1 + \lambda h)e_n + T_{n+1}$

So substituting, this is an iterative procedure because as I told you, we will start from x_0. So x_0 corresponds to n = 0. And then, we will compute x_1. So e_1 is basically T_1. And e_0 is because, in initial conditions, we are not making any approximation, therefore, $e_0 = 0$, $e_2 = (1 + \lambda h)e_1 + T_2 = (1 + \lambda h)T_1 + T_2$.

So in a similar way, we write $e_3 = (1 + \lambda h)^2 T_1 + (1 + \lambda h)T_2 + T_3$. And then, we substitute the value of e_2 from the previous steps and we do, we get the following terms.

So basically, if we start doing this in a recursive manner, we get e_n in this way which can also be written this way. So till this point, I hope it should be clear to everyone.

(Refer Slide Time: 20:08)



Now, we are using some inequalities. One of the inequalities, I will be using $|1+\lambda h|^{n-j} \le e^{h(n-j)|\lambda|} = e^{x_{n-j}|\lambda|} \le e^{|\lambda| x_n}$ because $x_{n-j} \le x_n$.

So you, and T_j is our, we also want to bound this term T_j, which is $T_j \le Ch^2$. Because T_j is the truncation error involving in one step, that is what you could see. T1, T2, T3, T4, these are the, we are bringing truncation error in each step. So that is Ch^2 for some constant C.

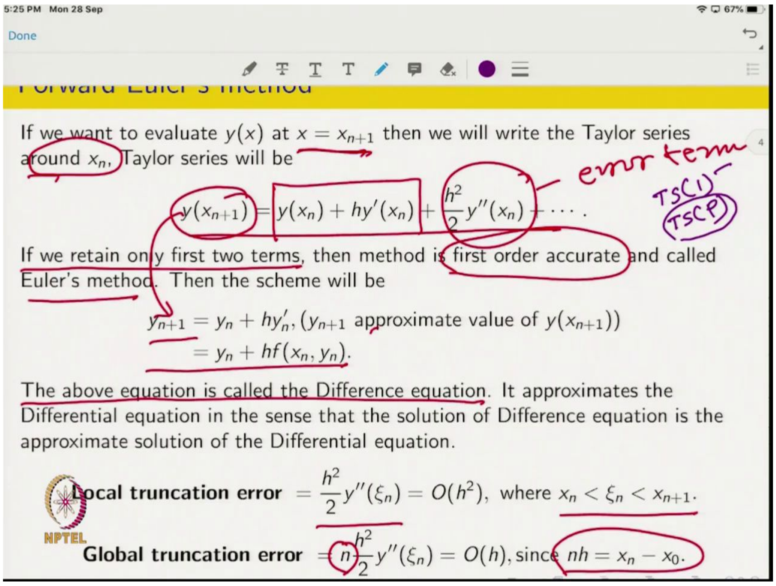
So when we substitute this here, we substitute the bounds for T_j as well as the bounds of this inequality here. And this term is Ch^2, we can take it common; Ch^2. And this is again independent of j. So that is why n and this term has come. So Ch, again we can right Ch and nh. xn is basically nh, so using the value of this variable, we can formally write this as a C1 h.

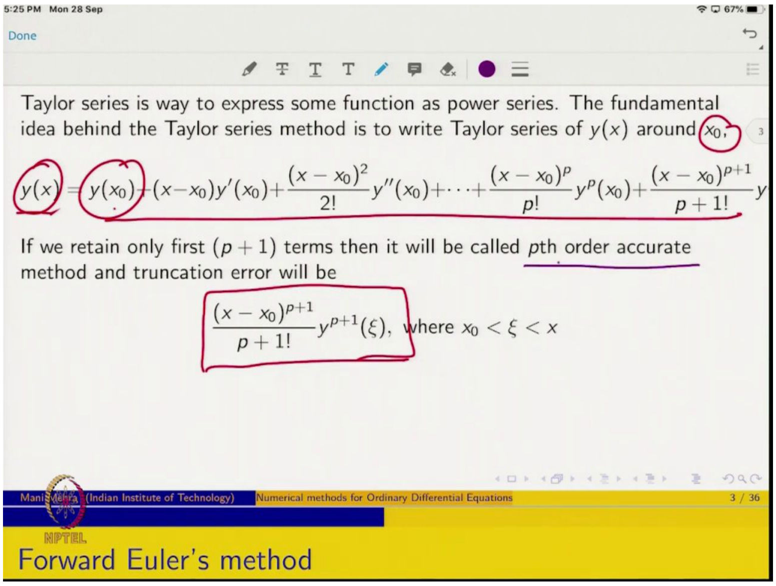
So this is a more formal way of proving the convergence because we are seeing, as $h \to 0$ and $|e_n| \to 0$. So since e_n is $O(h)$. Hence Euler method converges with the first order rate. So in this theorem, just for simplicity, we have chosen a specific value of f(x,y).

You can also work for general f(x,y) but in that case, you have to use some inequalities. Because here, we are also making one condition that g is a continuously differentiable function. But you can do for general f(x,y) also, in the similar way.

So finally, we have proved that the forward Euler method converges. That is one thing and converges with which order. The rate of convergence is the first order rate. The rate of a convergence is first order or the order of the convergence is first order.

(Refer Slide Time: 23:24)





So now, in a similar way, you can ask me that I can retain more terms in the Taylor series method if I wanted to achieve a high order method, yes. So the answer is yes. I can work out a second order method, third order method, fourth order method, similar way.

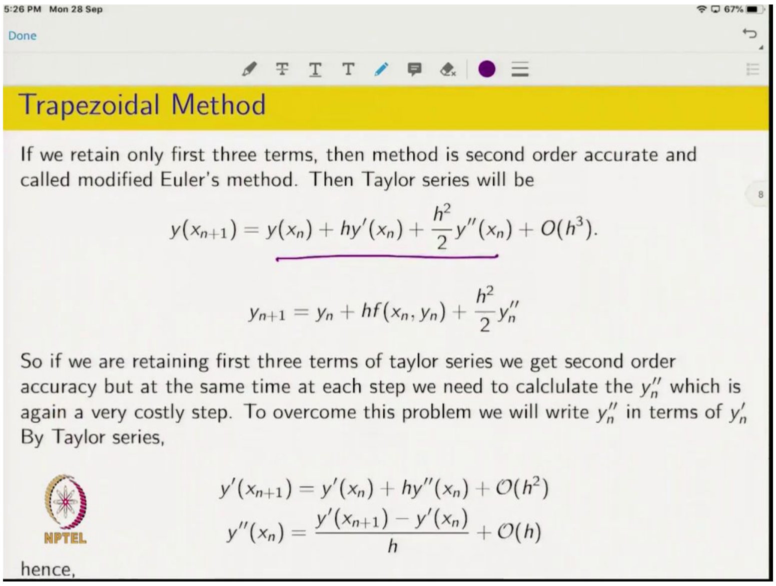
But what will be the disadvantage you feel if I keep adding this way? Because if I am retaining 2 terms, I have to calculate the approximation of y'. If I retain 3 terms, I have to calculate the approximation of y''.

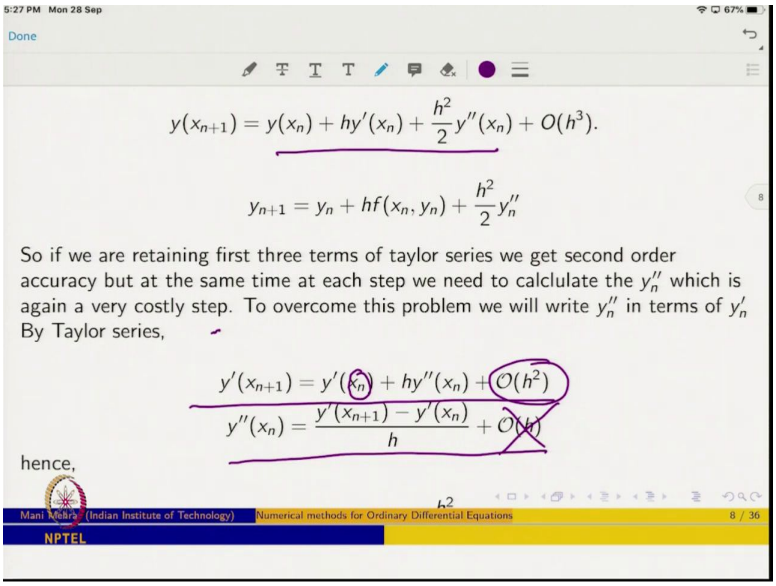
So in a similar way, which is a very complicated step because for some initial value problem, even analytically, you cannot compute the derivative very easily. So that is why the Taylor series method is acceptable to the community who do numerical methods only up to first or second order. It is not acceptable beyond that because the computation of approximation of high order derivatives is a tedious task.

So some people also, if we retain 2 terms, this is called TS1 method. If we retain p+1 term, this is called the pth order method. Because here, we are retaining 2 terms and it is called a first order method and if we are retaining first p+1 terms, then it is called the pth order method.

That is what we have seen here also. If we retain p+1 terms, then it is called pth order methods. But as I told you, it is not very practical to retain more and more terms. I have also told you the reason.

(Refer Slide Time: 25:50)





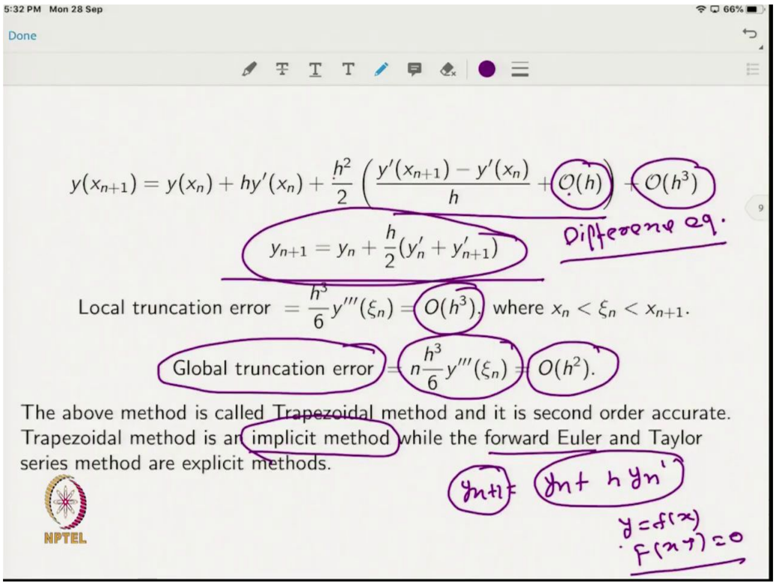
That is why, so now, the question is if it is a first order accurate method, how can we achieve high order methods? So for that reason, I am going to explain to you a Trapezoidal method. What does this method say? This method says, if we retain only the first 3 terms, the method is second order accurate and called a modified Euler method. Then the Taylor series will be this.

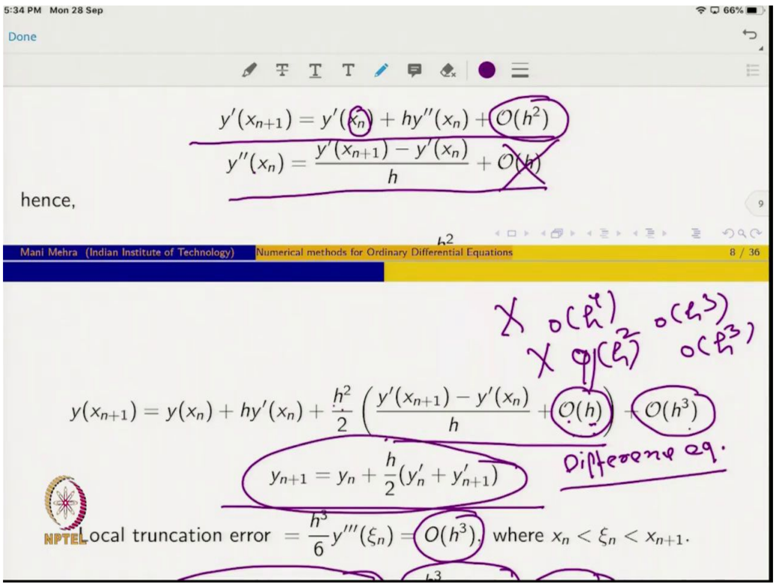
So basically, I am saying, if you retain the first 3 terms, it will be called Taylor series method. First 3 terms of Taylor series, so if we are returning first 3 terms of a Taylor series, we get a second order equation. But at the same time, at each step, we need to calculate the y'_n, which is again a very costly step. That is what I have already explained to you.

So to overcome this problem, what do we do? We write y''_n in terms of y'_n by Taylor series. So we are doing some manipulation in the Taylor series. We are, this is the Taylor series of y' around point x_n. This is the error term, truncation. So if by rearranging the terms, we get the following.

So you must have already seen these things in the, when you must be doing a numerical approximation of the derivatives. So now, we are writing $y''(x_n)$ in terms of $y'(x_n)$. So if I neglect this term, this will be an approximation..

(Refer Slide Time: 27:28)





So now, I am substituting the value of y'' here in the Taylor series. Means, I am substituting this value of y'' here. And after neglecting the error term which is $O(h^3)$ from here, and again this is $O(h^3)$ because h^2 with h and this will be the $O(h^3)$.

So after neglecting this term we get the following equations. So again, what is this? This is a difference equation. So in numerical method, basically, we are not solving the differential equations. In numerical methods, we are solving the difference equation, which is the solution of the difference equation, and approximate the solution of differential equations. That is what we have seen earlier also.

So the local truncation error of this method will be $O(h^3)$. That is what we have already seen. And global truncation error will be the same way, which I have explained to you earlier. It will be multiplied with n.

You can also prove it in a more formal way. The way I have done it is a forward Euler method, but this is easier. So that is why, because I do not need to prove the theorem in each and every case. So this is $O(h^2)$.

So the Trapezoidal method is a second order accurate because the order of the method is governed by the order of the global truncation error. That is what we have already said. So the above method is called the Trapezoidal method and it is second order accurate. Trapezoidal method is an implicit method.

So here, there is a point to explain. What do you mean by implicit method or what do you mean by explicit method? So this implicit and explicit you must have seen, when you were, what is the meaning of an explicit function and what is the meaning of an implicit function.

So if you keep those distinctions in your mind, I can explain to you what is the meaning of explicit and implicit method. In the case of a forward Euler, the difference equation which we have used is $y_{n+1} = y_n + hy'_n$.

So in the left-hand side, there was a term y_{n+1}, while in the right-hand side, there was no term involving y_{n+1}. So that is also the meaning of explicit function when we write y explicitly as a function of x; while implicit functions are defined in the following way. When you cannot explicitly write y in terms of x.

So the same thing goes here. y_{n+1}, you can write explicitly in the form of a y_n; while in the case of a Trapezoidal method, in which I use the following difference equation, y_{n+1} term is also involved in the right-hand side. That is why this method is called an implicit method, contrary to the forward Euler method, which is an explicit method.

So now, you can think of doing some examples or you can think of doing some situations, where an explicit method is more beneficial or an implicit method is more beneficial. Or means what is the advantage and disadvantage of explicit and implicit methods. We have, so far, seen Taylor series methods, which the order of the Taylor series method is defined by how much term you keep in the Taylor series.

So the forward Euler is a special case of a Taylor series method and it is also called TS1. Forward Euler method, we have derived by retaining just the initial 2 terms of the Taylor series. But the Trapezoidal method, we have not derived just by retaining the initial 3 terms.

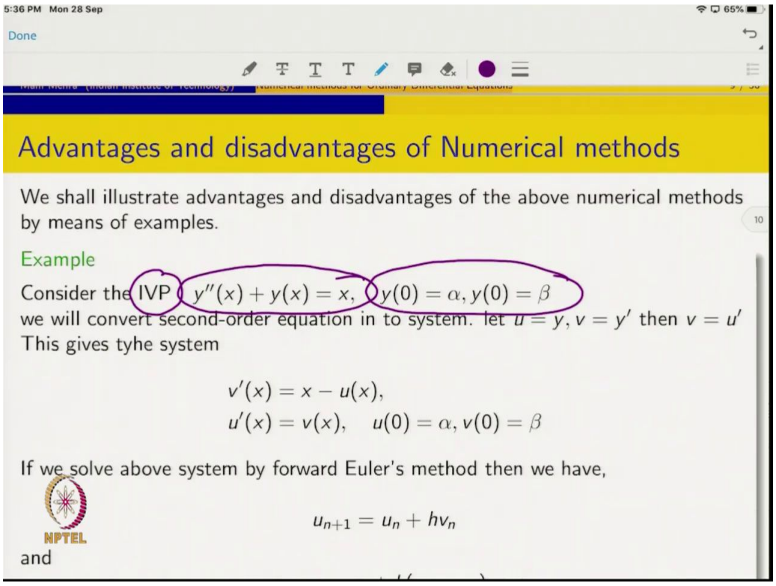
We have done some extra because we do not, we want to avoid the approximation of the second derivative. That is why we have done some extra calculation and that is why this is not a TS2 method. Trapezoidal is not a TS2 method. We have done something extra and that is why it is with a different name, which is a Trapezoidal method.

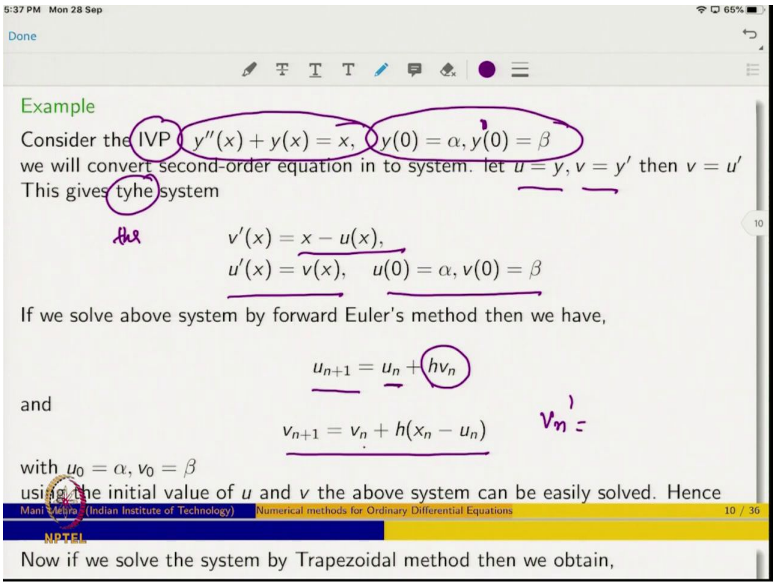
So now, the next question which can come to your mind is, why I have used only, why here I have kept here 2 terms? Because I also want that the error which I am neglecting in this part, means specifically in this part, that should also be $O(h^3)$. Because if this is order of, if after multiplying, if it would have been $O(h^2)$ and after multiplying with h^2, it would have been $O(h^4)$. And here, it is $O(h^3)$.

Or means this error can be $O(h^4)$ and this is $O(h^3)$ or if it would have been $O(h)$ and this would have been order of, sorry, $O(h^2)$ and $O(h^3)$. We want to retain the same order; not this, not this. I will explain to you later why not. But right now, I wanted to retain the same order; that is why we are here, we have retained only 2 terms of the Taylor series, and after that I will neglect this as an error.

So basically, we have seen Taylor series methods, forward Euler method, and Trapezoidal method. So all Taylor series methods, you can see they will be explicit methods. Though in detail, we have seen only one special case of a Taylor series method, which is a forward Euler method. But if you retain 2 terms, 3 terms, 4 terms without doing any manipulation, then also it will be an explicit method.

(Refer Slide Time: 34:36)



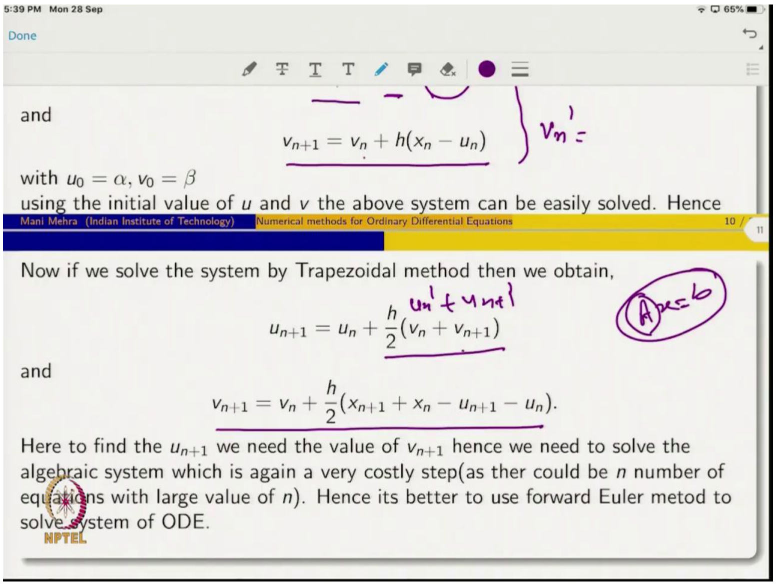


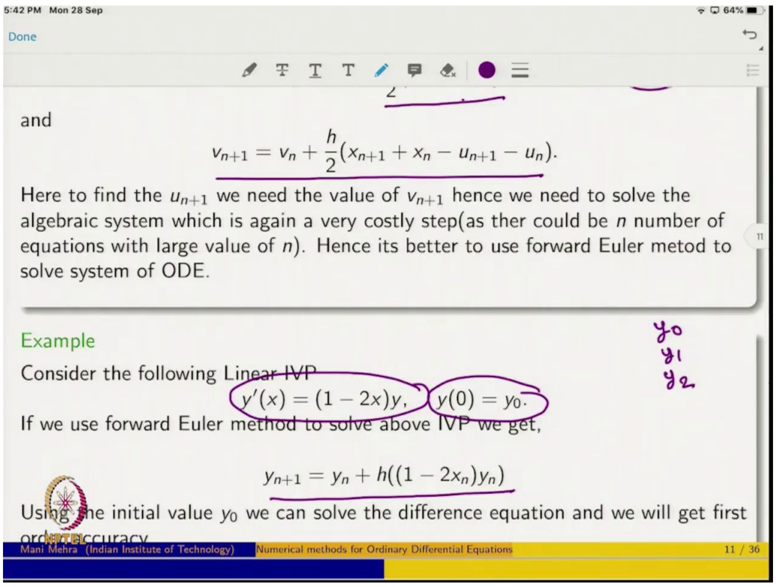
So now, let me go to the next slide, which is an advantage and disadvantage of the numerical method. So here, we will take a few examples. By those examples, you will understand what is the advantage and disadvantage of explicit and implicit methods.

So in this, now we are considering the following example, which is a second order differential equation, together with 2 conditions. Both the conditions are prescribed at one point 0, and that is why it is called the initial value problem. The abbreviation, which we have used for the initial value problem is IVP.

We will convert this second order equation to a system, that we all know how we can convert high order differential equations into a system of differential equations, first order differential equations. So I substitute this, $u = y, v = y'$. Then this solves the system. So $v' = x - u, u' = v, u(0) = \alpha, v(0) = \beta$. So if we solve the above system by forward Euler methods, what will be the difference equation of a forward Euler method we have seen? $u_{n+1} = u_n + hv_n$ and $v_{n+1} = v_n + h(x_n - u_n)$ with $u_0 = \alpha, v_0 = \beta$. Using the initial value of u and v in the above system can be easily solved, which we all can understand.

(Refer Slide Time: 37:10)





Hence, now, if we solve the system by the Trapezoidal method. So this is the idea I have given you, how you can solve a system of differential equations using the forward Euler method. And now, let me come to the Trapezoidal method.

So in the Trapezoidal method,

$$u_{n+1} = u_n + \frac{h}{2}(v_n + v_{n+1})$$
$$v_{n+1} = v_n + \frac{h}{2}(x_{n+1} + x_n - u_{n+1} - u_n)$$

So here, to find the value of u_{n+1}, we need the value of v_{n+1}, that is what you can see from here. Hence, we have to solve the algebraic system, which is again a very costly step as there could be a n number of equations with large values of n. Hence, it is better to use the forward Euler method to solve the system of ODE.

If for a system of ODE, n is very large or you can say, here we have a start, why we have ended with a 2 by 2 algebraic system because we started with second order differential equations. If there are, the order is more, the order of this system will also be more. That you must have seen in some theoretical course of differential equations.

So, solving an algebraic system of high order is a costly step. That all of us know and you must have also observed in the earlier part of this course that solving Ax = b is a costly step once the order of A is large. 2 by 2 system is anyway, you can solve it by hand also but we are developing numerical methods, not for 2 by 2 cases. This is just for simplicity, we are taking this problem but our motivation is for a general and complicated case.

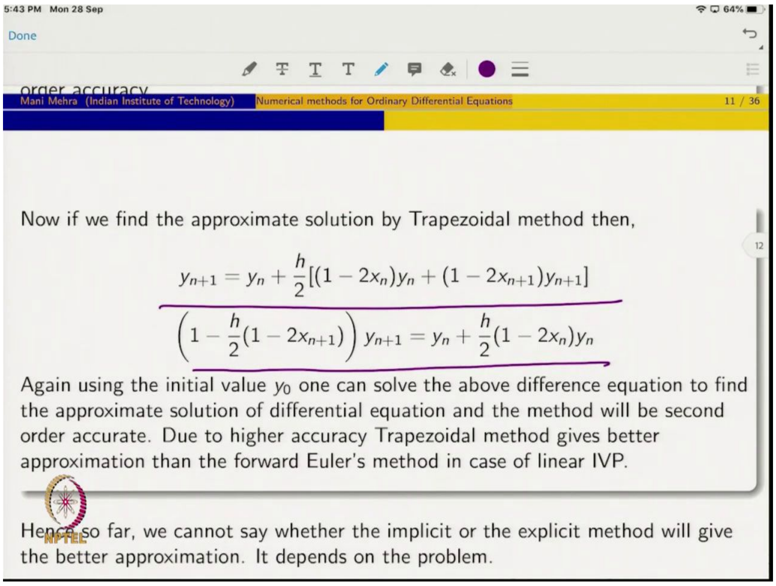
So if you look at this example and you solve it by forward Euler and you solve it by Trapezoidal method. So the first difference, which I can observe between forward Euler and Trapezoidal method is, forward Euler is a first order accurate method, while Trapezoidal method is a second order accurate.

So of course, using Trapezoidal, I am getting more accuracy, that is one thing. But at the same time, I am involving one extra step which is costly, so means, computation is more. So if you ask me, in terms of computational efficiency, forward Euler is more simpler; the computational efficiency is more because you are not solving an algebraic system.

But in case of a Trapezoidal method, to solve a system of differential equations, you have to solve an algebraic system. So that is the one difference you could observe between explicit and implicit methods in general. So you can say, Forward Euler is a one category of an explicit method. Similar thing, Trapezoidal method is also a category of implicit methods.

Now, let me consider another example where we are considering the following initial value problem and with the following initial conditions. So this time, this is a first order, if we use the forward Euler method to solve, the difference equation will be this. And so, y0 is given to me and then I can compute y1; then from y1, I can compute y2. So we can solve the difference equation and we get the first order accuracy.

(Refer Slide Time: 41:38)



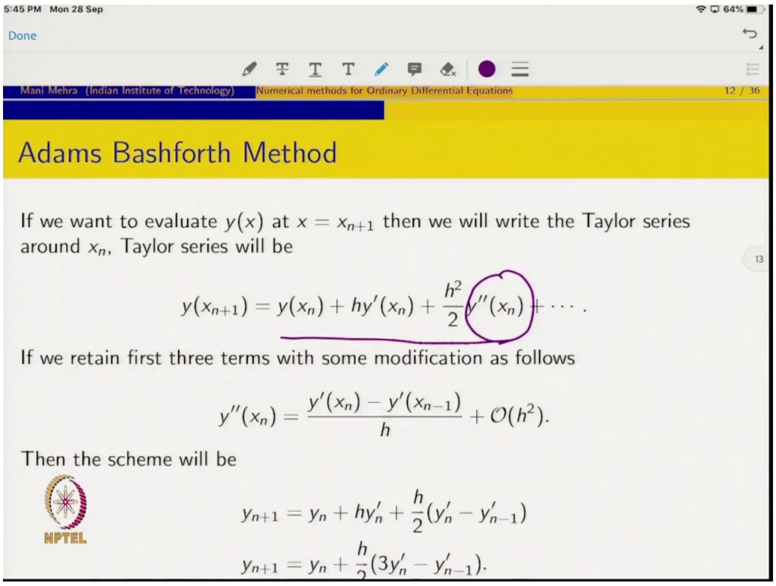
And similar things, if we approximate it with the Trapezoidal method, we use the following difference equations. And so, for a linear problem, we are not taking any extra step. So for a linear problem, as well as this is just a first order. So again, using the initial values, we can solve the above difference equation to find the approximate solution of a differential equation and the method will be second order accurate.

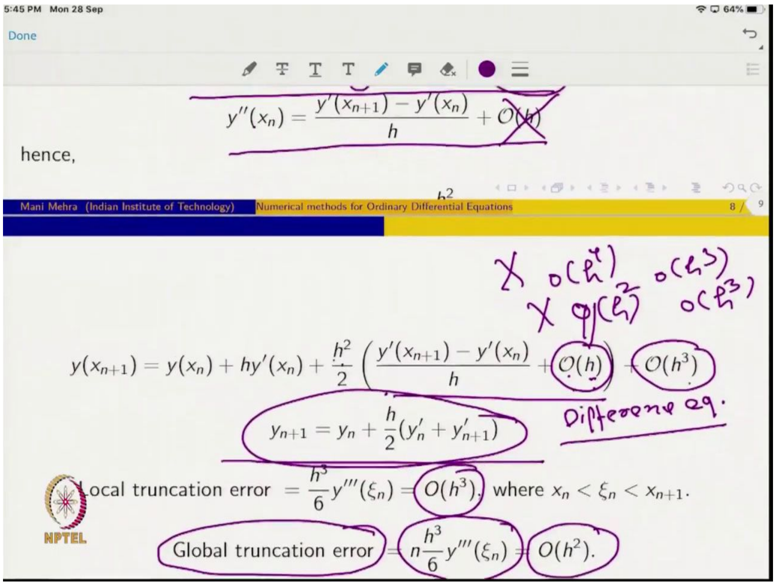
Due to higher accuracy, the Trapezoidal method will give a better approximation than the forward Euler. That is what we have expected in case of a linear IVP. Hence, so far, we cannot say whether the implicit or the explicit method will give the better approximations.

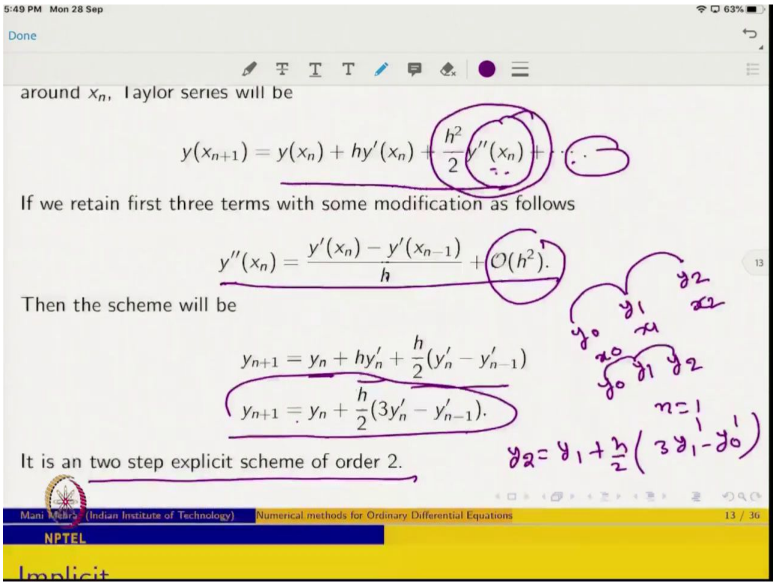
Because in one case, of course, I am saying the Trapezoidal method will give the better approximation because it is a high order method. But if I compare implicit and explicit methods of the same order, then which one will give you a better approximation that you cannot say because both are under same order accurate methods.

In that case, depending on the problem you have to choose or depending on the means, how much computational efficiency you want from the problem. So that, later on, I will take one example of a nonlinear problem also and I will explain to you how it works?

(Refer Slide Time: 43:15)







So now, let me start with the Adam Bashforth method. What is Adam Bashforth? Again, Adams is also the name of the scientist, Adam Bashforth. So they are basically, under this category, we learn how to develop high order explicit methods.

Because so far, we have seen the forward Euler method, which was a first order but how to develop high order methods, but explicits. Because the Trapezoidal Method was implicit. So again, of course, if we retain these terms as such this will be called the Taylor series method of second order.

If we do some modifications here, that is what we have seen in the Trapezoidal method. We develop the Trapezoidal method. If we retained, first 3 with some other modifications like this, $y'(x_n)$, I am writing $y'(x_n)$ in the following way.

So this again, you, that getting this step is not very tedious. The same way, like you have done earlier, you can write a difference equation and you can get it like you have done here.

So then, the scheme will be, so I am retaining this $O(h^2)$. So again, why I am retaining $O(h^2)$, that should be clear to you.

I want the approximation of this term, the whole term. The error which I neglected from this term and the error which the order should remain the same. It should not be different. Why should it not be different? That is what I will explain to you later. So with this

$y_{n+1} = y_n + hy'_n + \frac{h}{2}(y'_n - y'_{n-1})$ after simplification we will get $y_{n+1} = y_n + \frac{h}{2}(3y'_n - y'_{n-1})$

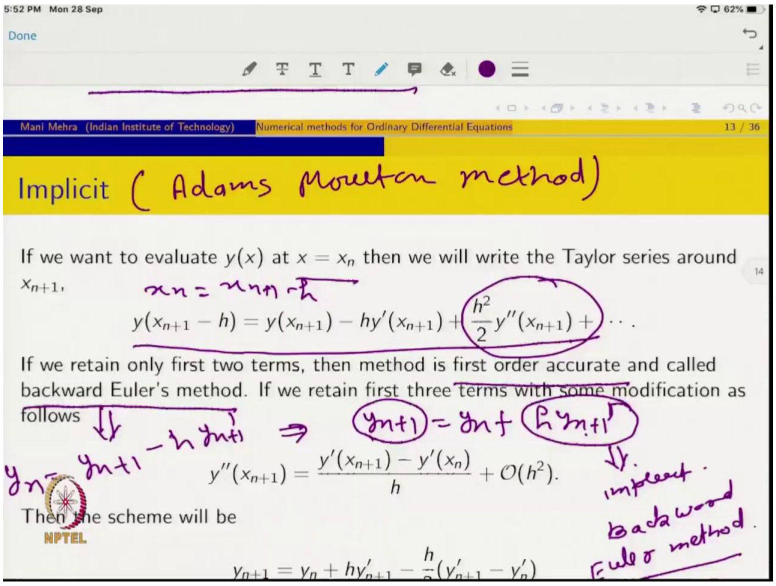
So it is a 2 step. So again, you will call it an explicit method or implicit method? In my opinion, it is again an explicit method because the left-hand side contains y_{n+1}, while the right-hand side does not contain any terms, which involves y_{n+1}. So it is again an explicit method.

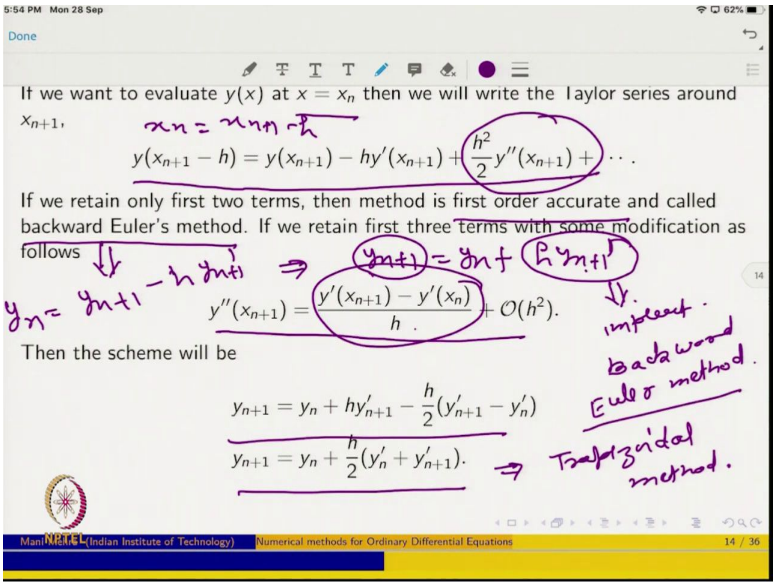
But here, what is the cost I am paying to achieve high order explicit method? That it is a two-step explicit scheme of order 2. What is the meaning of 2-step? Like if I am going from y0 to y1 in a one, this is called a single step, one step method. But here, to compute at the value of y2, I need the value of y naught as well as I need the value of y1.

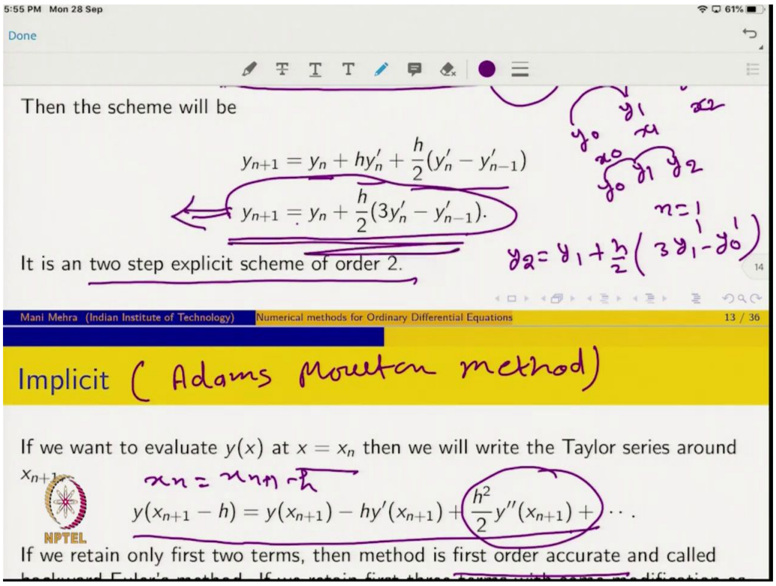
Because if I substitute n is equal to 1 here, so this will be y2 y1 plus h by 2 3 y1 dash minus y0 dash. So basically, to compute the value at the following grid, so this is at x naught corresponds to x1 corresponds to x2. So to compute y2, I need from this, I need from this.

So that is why it is called a two-step method, while forward Euler method was a one-step method. I need previous values only one step previous. In this case, I need two-step previous values. So, that is why it is a two-step explicit scheme of order 2. So now, this is an Adam Bashforth method, which is a general method to derive high order accurate explicit methods category.

(Refer Slide Time: 48:07)







Similarly, how to derive high order accurate methods but implicit? This time, I am saying implicit, which will become under the category of an Adams Moulton method. So if you, so here, again we will be playing with the Taylor series.

So here, I am writing the Taylor series in the following way. If you want to evaluate y(x) at x_{n+1}, so basically, xn is $x_{n+1} - h$. That is what we are doing. So here also, means Taylor series is written in the following way. If we retain only first 2 terms, again so if I am retaining first 2 terms, this term will be neglected; not this term exactly, the remaining terms will be neglected.

So the local truncation error will be order of h square and global truncation error will be order h. That is why, here itself I am writing this method is called first order accurate and called backward Euler methods because the order of a local truncation error will be 2, and that is why, again with the same calculations which we were doing in case of a Trapezoidal as well as in case of a forward Euler, the global truncation, the order of global truncation error will be order of h. That is why it is called the first order accurate method, backward Euler.

But why this time backward, that is what I will explain. So if, so here, the difference equation in case of a backward Euler will become, this is $y_n = y_{n+1} - hy'_{n+1}$. So basically, it means $y_{n+1} = y_n + hy'_{n+1}$. So why is it an implicit method? Because the term which is involved in the left-hand side is also involved in the right-hand side. That is why this is called the implicit backward Euler method. Okay, clear to everyone?

So in a similar way, with some modifications, again I can say how to develop high order implicit methods. How to develop high order implicit methods, and those methods will come under the category of Adams Moulton methods. Like in the previous case, we have seen how to develop high order accurate explicit methods. They come under the category of Adam Bashforth methods.

So basically, the whole idea is Adam Bashforth methods can be of different order, as well as Adams Moulton method can also be of different order, depending what type of calculations you do, what is the desired order you do. So now. if now I will, I wanted to retain 3 terms. That is the only idea with some modifications.

So I am writing, y dash in the following way. I will use the following approximation for the derivative. After substituting this value here, I get this which is basically the same difference equation which we have seen in case of a Trapezoidal method.

So and, so here, let me stop in this lecture here itself. In the next lecture, geometrically also I will explain to you why this is called the Trapezoidal method. So in a nutshell, in today's lecture, we have seen Taylor series methods, Adam Bashforth methods, Adams Moulton methods.

But we have seen only a few first order as well as second order Adam Bashforth formulas; some people also call it as a formula, as well as in Adam Moulton's case also, we have seen first order method, we have seen second order method. First order is backward Euler, second order this Trapezoidal.

In the case of Adam Bashforth methods, first ordered his forward Euler and second order is not, I am not specifying it with some special name and the difference equation of a second order method is this, which you can observe from here. So here also, if I compare the Trapezoidal method with the second order Adam Bashforth formula, the only difference is a two-step, this is a one-step.

So later on, I will also explain to you what is the advantage and disadvantage of one-step, two-step methods; like at least so far, I have shown you one disadvantage of implicit method over explicit methods in case of a system of differential equations. If you are solving just one first order linear differential equation together with initial value, then it is not a disadvantage but in case of a system of differential equation which you get when you solve second order differential equations together with initial conditions, in that case, it is a disadvantage.

So with these things, I am closing now. Thank you very much for your attention.