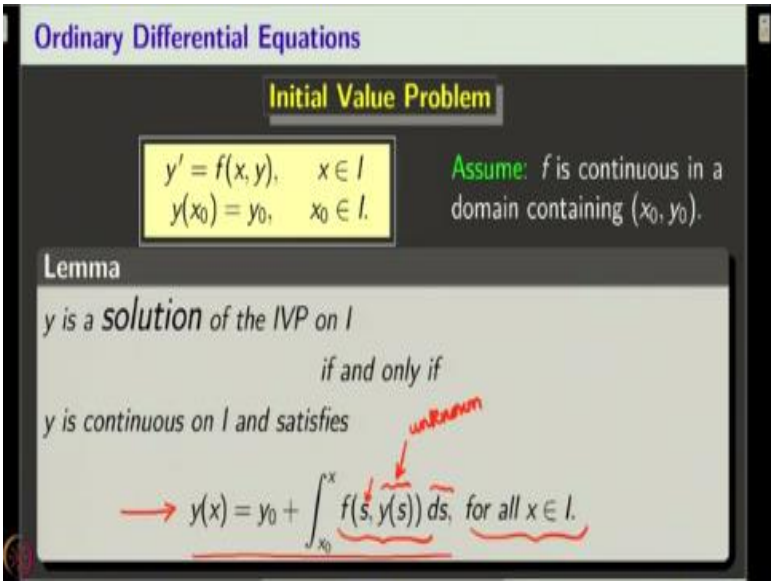
Numerical Analysis
Prof. S. Baskar
Department of Mathematics
Indian Institute of Technology, Bombay

Lecture - 55
Numerical ODEs: Euler Methods

Hi, in today's class we will start a new chapter and this is on numerical ordinary differential equations. In this chapter we will learn to develop numerical methods for some ordinary differential equations, with initial and boundary conditions. In this lecture, we will learn a very basic method called Euler method. Let us first see, what is the model that we are interested in approximating.

(Refer Slide Time: 00:48)



First let us see what is mean by an initial value problem? We are given a first order ordinary differential equation in the normal form, that is, $y' = f(x, y)$, where f is a function of two variables x and y. x is the independent variable and it is taken in an interval I and y is the dependent variable. It depends on the independent variable x, that is y is a function of x and y' denotes the differentiation of y with respect to x.

And this equation is posed on an interval I, for instance you can take this interval as a closed and bounded interval $[a, b]$ and we are also given an extra information, that y at some point x_0 in the interval I is given by y_0. So, it means the value of the unknown function y at one point is specified,

such a condition is called an initial condition. If you are specified the unknown variable at two different points, then it is called the boundary condition.

Here we are given initial condition and therefore this problem, that is the ordinary differential equation $y' = f(x, y)$ and the initial condition $y(x_0) = y_0$, posed on the interval *I* with x_0 in *I*, is what we generally call as the initial value problem. And we often denote it by IVP, here the function *f* is given to us and we are supposed to find the unknown variable *y* for all *x*, in the interval *I* such that *y* satisfies this equation and also, it satisfies this initial condition.

We generally assume that *f* is a continuous function in a domain in $\mathbb{R}^2$, that contains the point (x_0, y_0). If you would have gone through a basic course on ordinary differential equations, you will know that under this condition, we can say that this initial value problem has a solution in a small neighbourhood of the point (x_0, y_0). This is what is the statement of a well-known theorem called, Peano existence theorem.

We assume this condition, because we are going to develop some numerical methods to approximate a solution of the initial value problem. In order to ensure that our initial value problem has a solution at least in a small neighbourhood of (x_0, y_0), we have to assume this condition. Otherwise, we will be devising and trying to capture something which does not exist, that makes no sense. Therefore, this assumption is very important.

Well, there is a nice characterization for a solution of this initial value problem, in terms of a solution of an equivalent integral equation and this is especially very important and useful for us to generate many numerical methods for our problem. So, let us recall this result from our basic course on ODE. Here the theorem says that *y* is a solution of the given initial value problem, that is this problem, posed on an interval *I* if and only if *y* is a continuous function on the interval *I*.

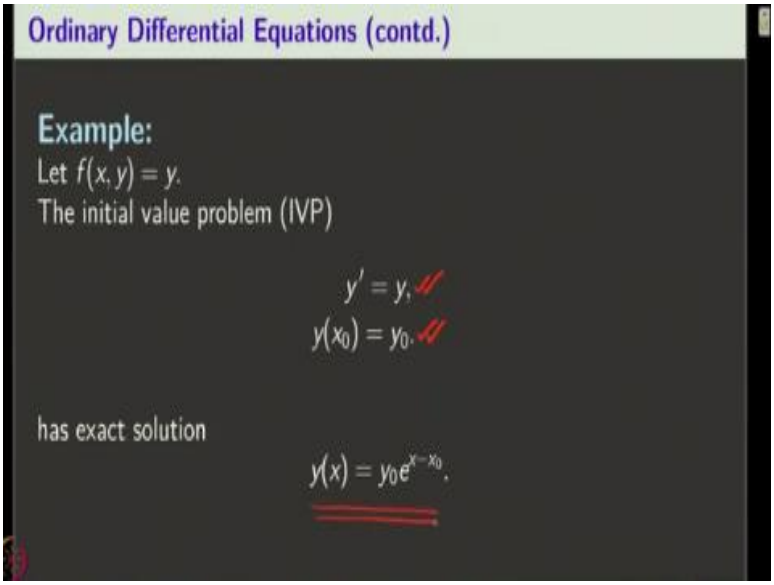
And it satisfies this integral equation. What is this integral equation? It is $y(x) = y_0 + \int_{x_0}^{x} f(s, y(s)))ds$, this is the dummy variable in the integral and the integration is done with respect to that variable. $y(s)$, this is unknown to us. Therefore, in general we cannot explicitly evaluate

this integral unless the function *f* is independent of *y* and this equation should hold for all $x \in I$, that is the integral equation.

You can see that it is also an equation for which $y(x)$ is an unknown. You can see that a direct integration of this will give rise to this equation under suitable conditions on *f* and *y*. So, that is what the Lemma says. It is not very difficult for us to prove, you can use the fundamental theorem of calculus to prove this. But this is a very useful result for us to devise many numerical methods.

Let us see, how to devise numerical methods to approximate the values of the unknown function *y*. This is the aim of this chapter.

(Refer Slide Time: 06:44)

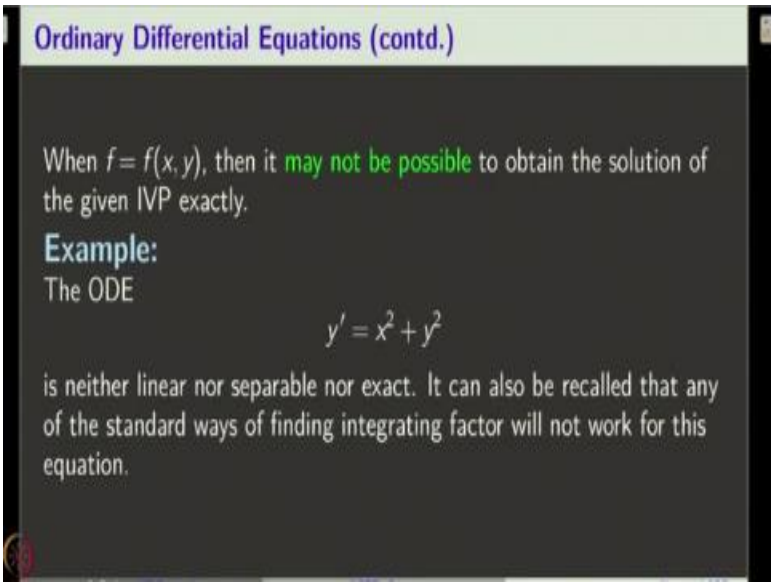


Before getting into the construction of some numerical schemes, let us try to recall some simple examples. Let $f(x, y) = y$, then our initial value problem is given by $y' = y, y(x_0) = y_0$. This is the initial condition and this is the equation given to us and this problem is posed on some interval. Since it is a very simple problem, I have not specified or restricted our problem to any particular interval.

You can immediately see that the exact solution of this initial value problem is $y_0 e^{x-x_0}$. How will you check this? Well, you can differentiate the function $y(x)$ and see that it satisfies this equation and also put $x = x_0$ and see that the initial condition is also satisfied. Therefore, this is clearly a

solution of our initial value problem. Well in this simple example, we got the exact solution very nicely.

(Refer Slide Time: 07:58)



But in general, it may not be possible for us to get exact solution for any given initial value problem. That is for any given right hand side function $f(x, y)$ and for any given initial condition, sometimes we may know that there exists a solution. Like for instance if f is continuous in a domain in $\mathbb{R}^2$ that contains (x_0, y_0), that is our data, then Peano theorem says that we always can find a solution at least in a small neighbourhood of (x_0, y_0).

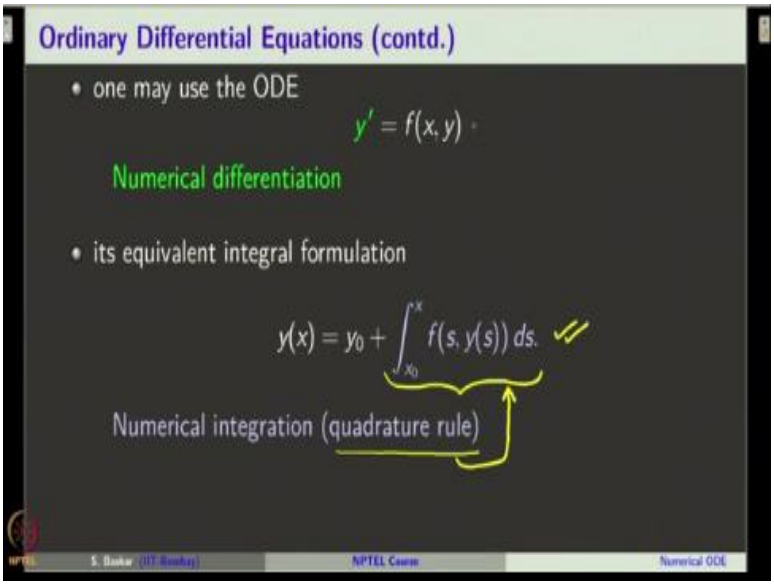
Like that we also have Lipschitz theorem, which says that if f is lipschitz with respect to y in a domain containing (x_0, y_0). Then again you can say that there exists a unique solution for our initial value problem in a smaller neighbourhood of (x_0, y_0). So, these results are qualitative, you cannot construct the solution and show these results. Therefore, in general there are equations for which you know that there exists solution. In fact, we may know also that there exists a unique solution in a small neighbourhood of your initial data.

But you may not know how to obtain these solutions explicitly, something like what we got in the previous example, we had the solution explicitly. But this may not be the case always. For instance, you consider this equation $y' = x^2 + y^2$, you give any initial condition to this equation and define

the initial value problem. In fact, you can show that it has a unique solution by properly defining an initial condition and restricting yourself to a suitable domain in $\mathbb{R}^2$.

But you can see that none of the methods that you learned in your basic ordinary differential equation course, will help you to actually construct the solution explicitly, for this particular equation. Now if you ask the question then how will I compute or construct the solution for such equations. Well, one way to do it is to go for certain numerical methods. In which case you can at least get an approximate solution for this initial value problem. This is the purpose of our chapter.

(Refer Slide Time: 10:47)



Let us learn how to device numerical methods for a given initial value problem of the form $y' = f(x, y)$ and the initial condition is $y(x_0) = y_0$. That is, we know the initial condition and we want to devise a numerical method for this ordinary differential equation. There are two ways that we can construct a numerical method for this differential equation. One is using the numerical differentiation formulas.

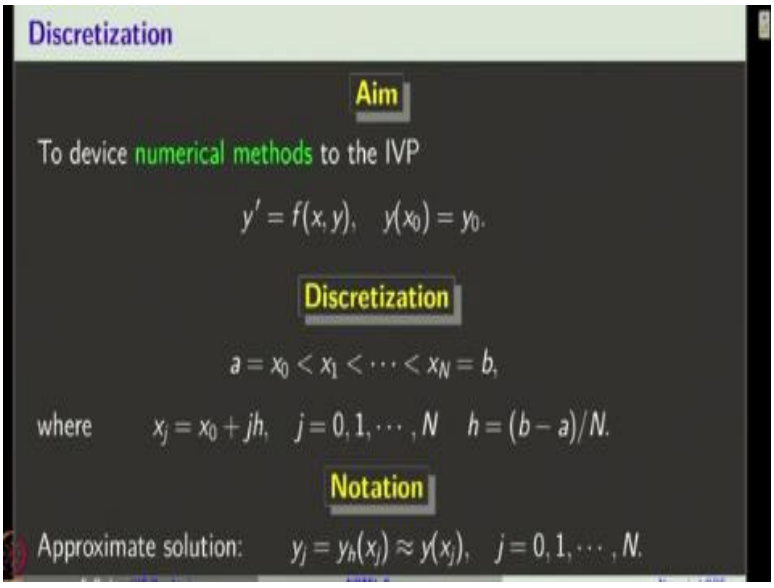
In the last chapter, we have learned to develop numerical differentiation formulas. We call them as finite difference formulas for any order derivative of a function. Here y is a function of x and y' is the derivative of y with respect to x. Therefore, we can use the ideas that were introduced in our last chapter, to approximate the derivative y' using some finite difference formulas. So, that is one way to develop numerical methods for this equation.

The next approach is go to the lemma, that we stated in the first slide of this class. We had an equivalent equation for our initial value problem. That is given as this integral equation. The lemma says that any continuous solution of this integral equation, is also a solution of this equation. Now look at this integral equation. We can also approximate this integral by some quadrature formula. Again, recall we have learned already to develop quadrature formulas for a given integral.

We can use one or the other quadrature rule to develop a numerical method for our initial value problem. By first considering the given ODE in the form of this integral equation and then approximate this integral by a suitable quadrature formula. Of course, different finite difference methods in this form or different quadrature formulas in this form, can lead to different methods for our initial value problem.

So, let us see how to device different numerical methods for our initial value problem. First let us understand a general framework for these methods.

(Refer Slide Time: 13:34)



To devise a numerical method for the given initial value problem, first you have to discretize your interval. Remember this initial value problem is posed on an interval $x \in I$, that is our interest is to obtain the unknown function *y* on the interval *I*, such that both this equation is satisfied as well

as this initial condition is satisfied. So, the first step of our method is to introduce a partition to the interval *I*.

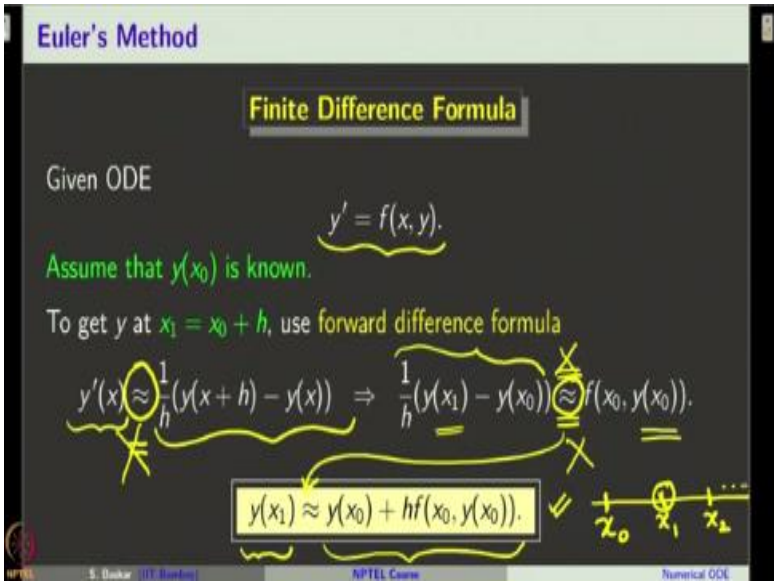
Let us consider this interval as a closed and bounded interval $[a, b]$ and first introduce a partition to this interval, which we call as discretization of our space. Here the space is the interval for our problem and all this node points are called grids. In the context of interpolation, we call them as nodes and in the context of numerical methods for differential equations, generally people call these points as grids.

And our numerical methods will give approximate values of our function *y* only at these grid points. Remember your numerical method will not give exact solution, it only gives approximate values of your solution *y* and not for all *x* in the interval, only at this grid points which you introduced at the beginning. And how you are generating this grid points?

Well, you can have non uniform grids but for the sake of simplicity, more precisely notational simplicity, we will consider this partition as equally spaced partition. It means what? The length of each sub interval of this partition has an equal length denoted by *h* and it is given by $\frac{b-a}{N}$. If we have *N* + 1 grid points in our partition. And this grid points are obviously given by this. Now, our aim is to find and approximate value of the solution of our initial value problem. Generally, we denote this approximate solution by $y_h(x_j)$.

This suffix *h* is to denote that we have generated these values with *h* as the length of the partition. We also often use the notation y_j to denote this approximate value. These are some notations that you have to keep in mind.

(Refer Slide Time: 16:50)



Now let us learn to devise Euler methods. We are interested in solving the ordinary differential equation $y' = f(x, y)$. But our function f is too complicated, just like we have shown in the example, that we do not know how to obtain an exact solution. However, we know that there exists a solution, at least locally around the initial condition that we specified. These are something that we have to make sure before coming to develop a numerical method or to work with some numerical method.

Once, these are given, then let us see how to device the so called Euler method. Note that, we are also given the value of y at the point x_0, through the initial condition in our initial value problem. Now our aim is to find the value of y at the point x_1 which is given by $x_0 + h$. That is, we know the solution at the point x_0, that is given by $y(x_0)$ and we want to find the solution at the point x_1, that is our aim.

For this, we will put a finite difference formula for the derivative of y in our equation. Let us take the forward difference formula. If you recall, we have introduced this formula in the last chapter, when we were discussing the finite difference formulas for derivatives of a given function. In that, we have introduced forward difference formula and we have also learned the mathematical error expression for this formula.

Let us use that formula, it is given by $\frac{y(x+h)-y(x)}{h}$. So, we will try to replace this formula in the place of y' in our equation and see what happens. So, instead of $y' = f(x, y)$, what we are doing is, we are putting this approximate formula and we are evaluating an approximate value of *y* at the point x_0 by taking $x = x_0$ here. That is what we are putting here, and since we are taking $x = x_0$ in the equation, also you have to put $(x_0, y(x_0))$.

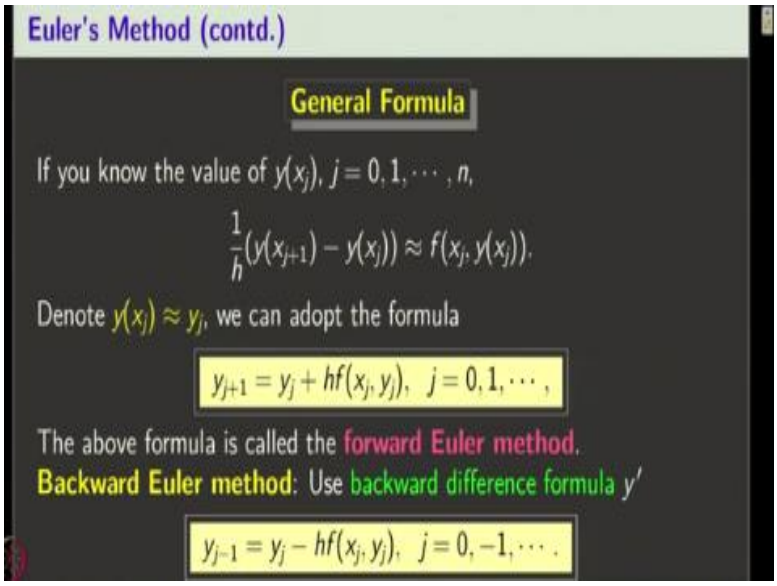
Remember, this is a function that depends on *x* although we write like this, it means $f\big(x, y(x)\big)$, you have to keep that in mind. That is what we meant by writing like this and also this means $y'(x)$ that is nothing but $\frac{dy}{dx}$, all this should be kept in mind always. And you see from here, we can write $y(x_1)$ as $y(x_0) + hf\big(x_0, y(x_0)\big)$. In all this, one thing should be carefully observed. The equation is given like this, we have put an approximation to y'.

Therefore, the left hand side is now not equal to the right hand side. It is only approximately equal to the right hand side. This is a notational discipline, that you have to keep in mind. Do not ever put, this is equal to this, that is wrong. You have to put approximately equal to, here also we are approximating this quantity by this formula. Therefore, you have to put approximately equal to. If you put exactly equal to, then that is wrong.

And that, approximately equal to, is also put here because the left hand side is the notation for exact value of *y* at the point x_1, whereas the right hand side is obtained by putting the finite difference formula. Therefore, they are approximately equal not exactly equal. So, this is what we got and with this formula, you can compute the value of *y* approximately at the point x_1, provided you are given the value of the function *y* at the point x_0.

Similarly, once you get the value of *y* at the point x_1, well approximately, then you can go to get the value of *y* at the point x_2, again approximately and so on. So, you can simply repeat this idea by replacing x_0 by x_1 and x_1 by x_2 in order to approximate $y(x_2)$ and similarly you can approximate $y(x_3)$ and so on, that is what I am trying to say.

(Refer Slide Time: 22:32)



Let us put this idea in general, we are known or given the value $y(x_j)$ for some $j = 0, 1, 2, \cdots, N$ and our aim is to find an approximate value for $y(x_{j+1})$. For that what we do is, we replace the derivative of y in our equation by the corresponding forward difference formula evaluated at the point x_j and that gives us this approximate relation. And from there you can get $y(x_{j+1}) \approx y(x_j) + hf\left(x_j, y(x_j)\right)$.

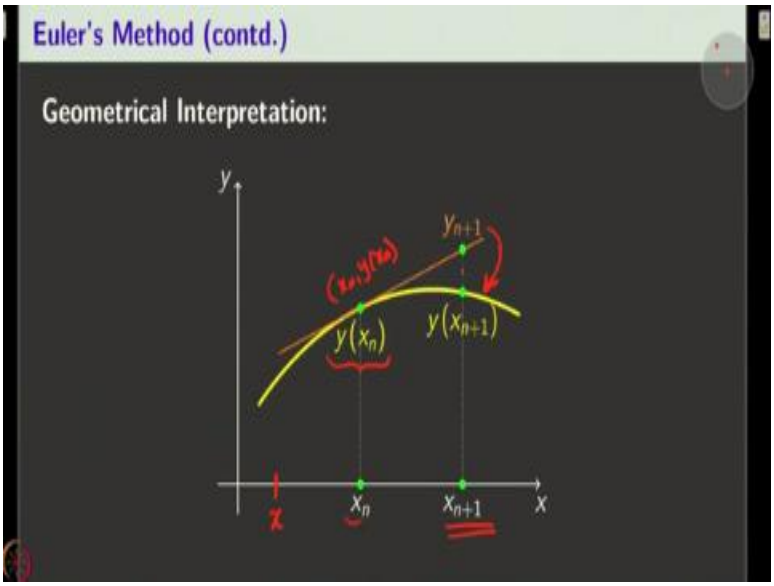
Remember, you have to put approximately equal to here. If you do not want to always remember to put approximately equal to, what you can do is, you can fix a notation for this approximate value. That is coming out of this relation by y_{j+1} and also you replace this by y_j and then you use this notation to indicate the approximate value computed through this formula. Then you can put equal to sign here, because this value is precisely obtained by evaluating this expression with their appropriate values.

Therefore, you can put equal to sign here, with this notation but here you should not put equal to sign, its only approximately equal to. This is just a notational discipline that you have to keep in mind always. This formula is called the forward Euler method, what this formula gives us is, you first know the value of the solution at the point x_0 through the initial condition. Now you want to find the value of your solution at $x_1, x_2, \cdots, x_N$.

In general, we take x_0 as a and x_n as b. This is again an unnecessary restriction, you can have $x_0, x_1, \cdots, x_N$ anywhere even in the interior of your interval. But generally, we will take either x_0 as a or x_0 as b. For forward Euler method, we will take x_0 as a and then we go forward in the space to get the value of our function at other points in the interval, which are lying on the right side of x_0. Then we have to use the forward Euler method.

Similarly, you can also devise backward Euler method by replacing y' in the given ordinary differential equation by its backward difference formula. And that leads to this expression for computing an approximate value to our function at a point x_{j-1} given y_j.

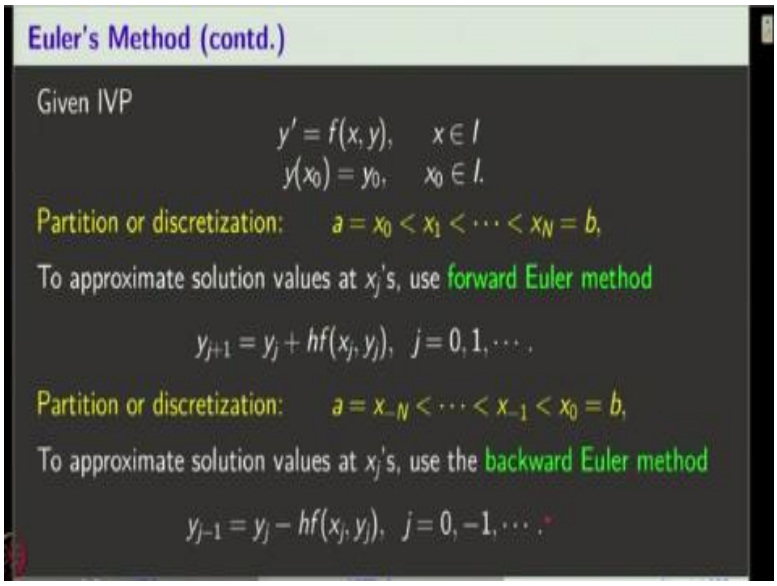
(Refer Slide Time: 26:03)



What is the geometrical interpretation for Euler method, let us see. Suppose, this is the solution that we want to obtain that is $y(x)$. Assume that we know the exact value of the unknown function y at the point x_n and it is given like this, then what the Euler method does, it goes along the tangent line at the point $(x_n, y(x_n))$ and takes the value of this tangent line at the point x_n and it gives that value as the approximate value of your exact solution $y(x_{n+1})$.

And this is the geometrical interpretation of the forward Euler method. Similarly, you can give the geometrical interpretation for the backward Euler method also. If you are given the value of the function at x_n and if you want to find the value of your unknown at x_{n-1}, then you have to go for the backward Euler method.

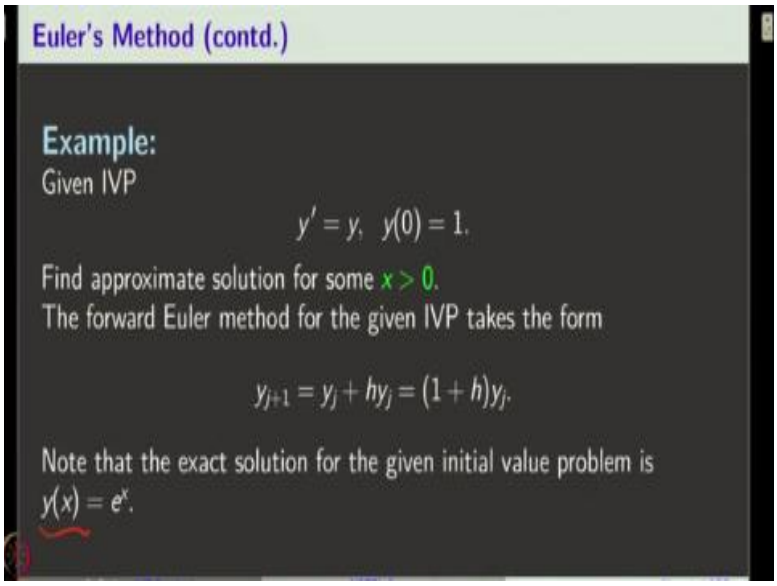
(Refer Slide Time: 27:24)



Let us summarize what we have done so far. We want to find an approximation to the given initial value problem. For that, first we have discretized our space interval $[a, b]$ by introducing a partition to our interval. So, these are called grid points. If you are given the value at some point x_0 and you want to find the values of your function at points that are lying on the right side of x_0, then you have to go for the forward Euler method and the formula is given like this.

On the other hand, if you are given the initial condition at x_0 and you are interested in finding an approximate value of the points that are lying on the left side of x_0, then you have to go for the backward Euler method. So, this is what you have to keep in mind, where the initial condition is given and where the approximation is needed. Depending on this we have to decide, whether we have to go for forward Euler method or backward Euler method.

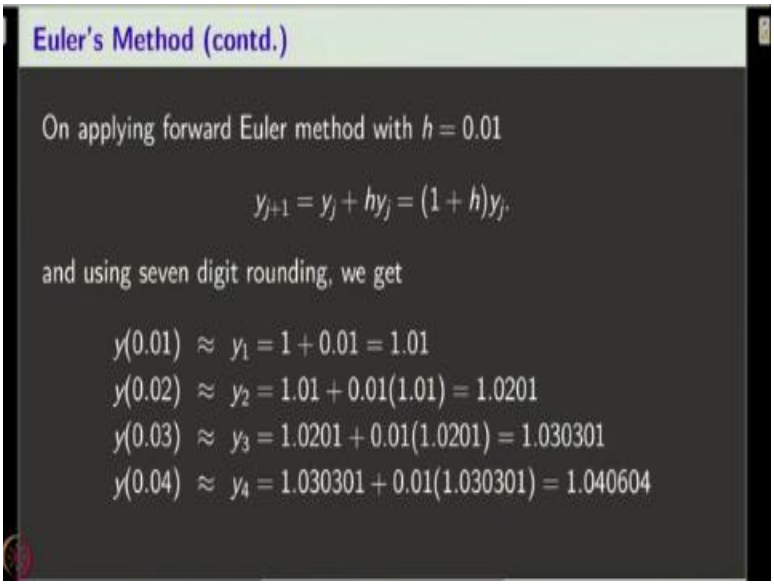
(Refer Slide time: 28:47)



Let us take the example, that we have seen at the beginning of this lecture. Our initial value problem is $y' = y$ and the initial condition is specified at x_0 which is equal to 0 and y_0 is taken as 1. With that, we know what is the exact solution for this initial value problem. Just to see how our Euler method works, let us also compute the solution of this initial value problem using Euler method.

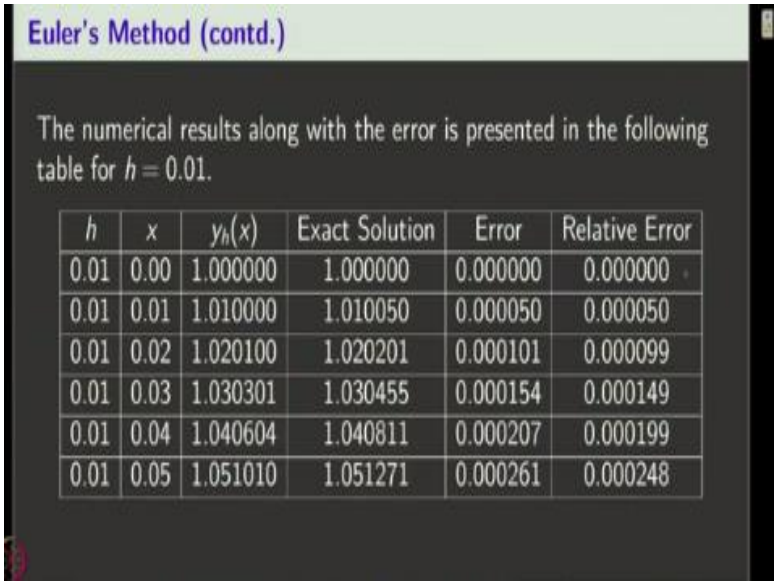
Here we are interested in finding an approximate solution for this initial value problem. For some point $x > 0$, it means what? Our domain lies on the right side of the point where we are given the initial condition. Therefore, we have to use the forward Euler method and forward Euler method in this particular example is given like this. You can easily check that, and the exact solution is given by $y(x) = e^x$.

(Refer Slide Time: 29:54)



Let us take h = 0.01 and compute the approximate values of the solution at different node points. y_1 is an approximation to $y(0.01)$ and it is given by 1.01. Similarly, you can find an approximate value of $y(0.02)$ denoted by y_2 and it is given by this number. Similarly, you can find $y(0.03)$ approximately and so on.

(Refer Slide time: 30:33)



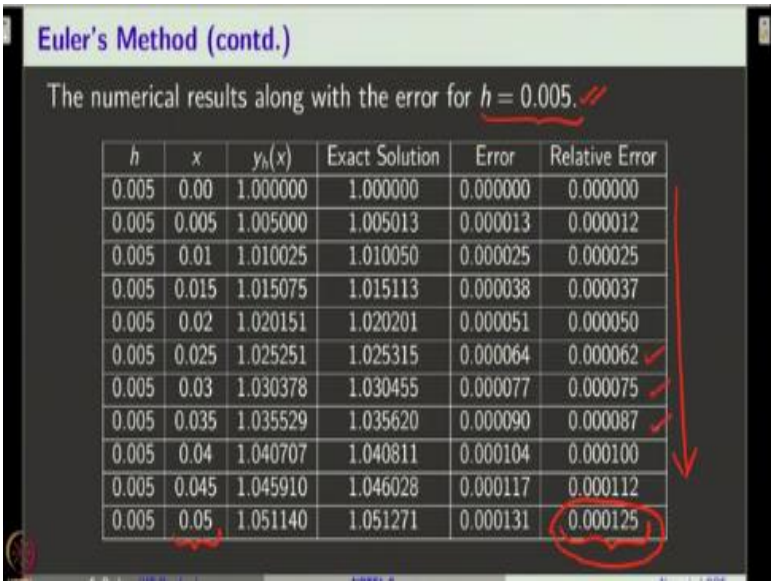
h	x	$y_h(x)$	Exact Solution	Error	Relative Error
0.01	0.00	1.000000	1.000000	0.000000	0.000000
0.01	0.01	1.010000	1.010050	0.000050	0.000050
0.01	0.02	1.020100	1.020201	0.000101	0.000099
0.01	0.03	1.030301	1.030455	0.000154	0.000149
0.01	0.04	1.040604	1.040811	0.000207	0.000199
0.01	0.05	1.051010	1.051271	0.000261	0.000248

Let us now see, how the error looks like. In this table we are giving exact value, approximate value, then we are also showing the error, that is exact value minus approximate value and also the relative error. For the approximate solution obtained using forward Euler method with h = 0.01, you can see from here the initial condition does not have any error, because we have taken this value from the initial condition itself.

Generally, in such cases we do not commit any mistake, unless there is some rounding error involved in the data. In this particular case, the initial condition is an integer therefore there is no rounding error involved in it. And that is why we see that the error is 0 at the first level, and as you go on computing the solution at later points of x, you can see that the relative error keeps on increasing.

That is, as you go away from the initial condition at every grid point when you calculate the solution, you see that the error is increasing. That is an interesting observation. When we are doing the error analysis, we will clearly see why we are accumulating the error, when we go from one grid point to the next grid point. So, we will keep this behaviour of error in mind and we will try to answer this question.

(Refer Slide Time: 32:19)



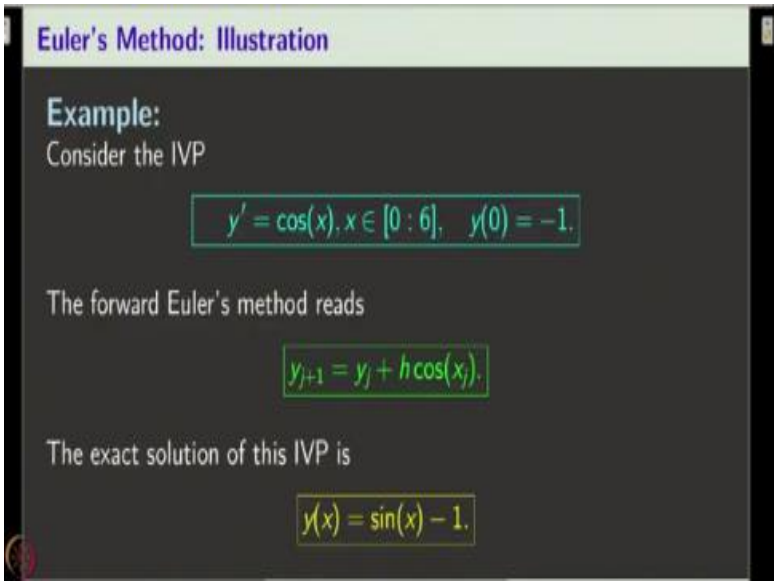
h	x	$y_h(x)$	Exact Solution	Error	Relative Error
0.005	0.00	1.000000	1.000000	0.000000	0.000000
0.005	0.005	1.005000	1.005013	0.000013	0.000012
0.005	0.01	1.010025	1.010050	0.000025	0.000025
0.005	0.015	1.015075	1.015113	0.000038	0.000037
0.005	0.02	1.020151	1.020201	0.000051	0.000050
0.005	0.025	1.025251	1.025315	0.000064	0.000062
0.005	0.03	1.030378	1.030455	0.000077	0.000075
0.005	0.035	1.035529	1.035620	0.000090	0.000087
0.005	0.04	1.040707	1.040811	0.000104	0.000100
0.005	0.045	1.045910	1.046028	0.000117	0.000112
0.005	0.05	1.051140	1.051271	0.000131	0.000125

Let us see how the errors look like, when we take $h = 0.005$ you can see for instance, in the previous case we have obtained the solution up to $x = 0.05$ and at this point the relative error was given like this. Now we have decreased h and thereby you can see at the point $x = 0.05$ the relative error is given like this, which is much lesser than what we got as the relative error with $h = 0.01$. This shows that as you go on decreasing h, you will get better and better approximation for your solution.

But you should keep in mind, that we have seen a very dangerous behaviour of the finite difference formulas which are also true for quadrature formulas. If you go on decreasing *h*, do not think that your error will also decrease and tends to 0. There is an optimal *h*, beyond which if you tend to decrease *h*, your error may tend to increase because of the arithmetic error. The same behaviour will also be seen in this kind of methods.

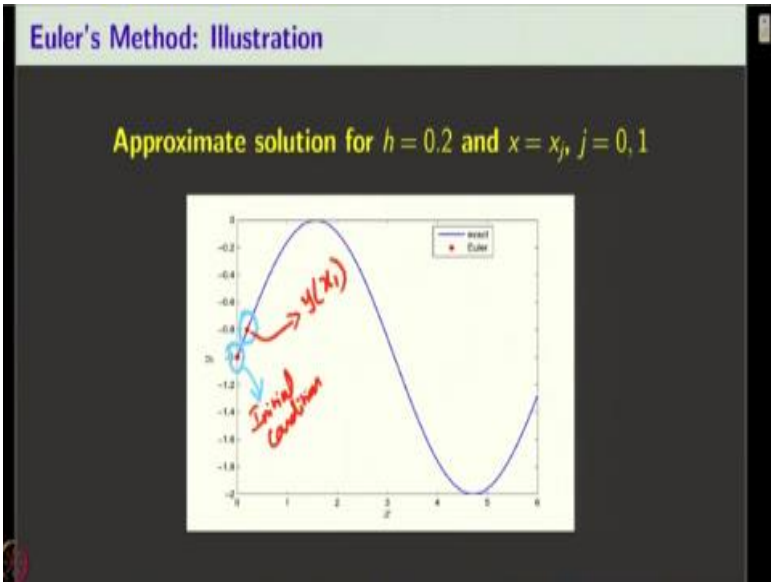
Because we are basically using the finite difference approximations. However at least from h = 0.01 to h = 0.005, you see that the error is pretty good. However, you can see that as you go on with your x_i, your error gets accumulated here. You can see from one step to the next step, to the next step the error is accumulating. We will study this behaviour when we are doing the error analysis.

(Refer Slide Time: 34:25)



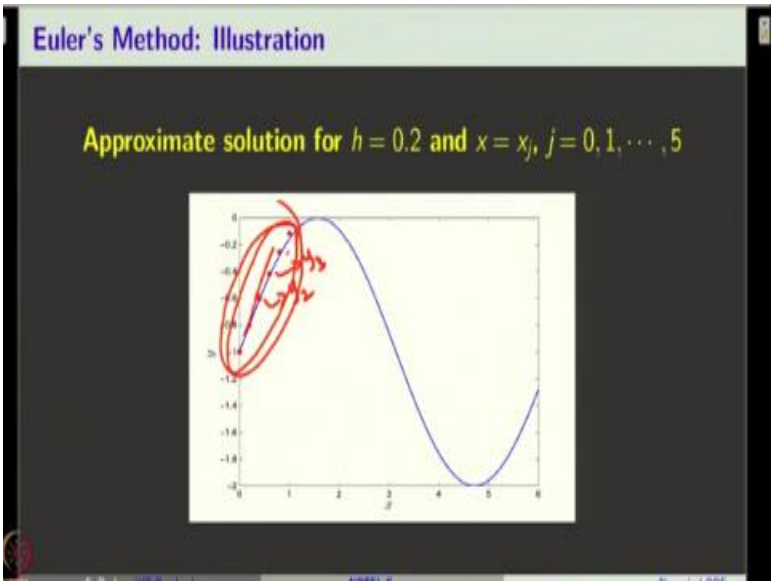
Let us see another example, here we will take $y' = cos\, x$ and we are interested in solving this initial value problem $y' = cos\, x$ and $y(0) = -1$ in the interval [0,1]. The Euler forward formula gives us the expression like this, and we know that the exact solution is like this. Let us try to visualize the solution of this initial value problem, along with the computed solution from the forward Euler method.

(Refer Slide Time: 35:02)



The exact solution of this initial value problem is shown in solid blue line and the Euler solution is shown in red dotted lines, which are shown here for the initial condition. This is the initial condition. And then this is the solution obtained as an approximation to $y(x_1)$.

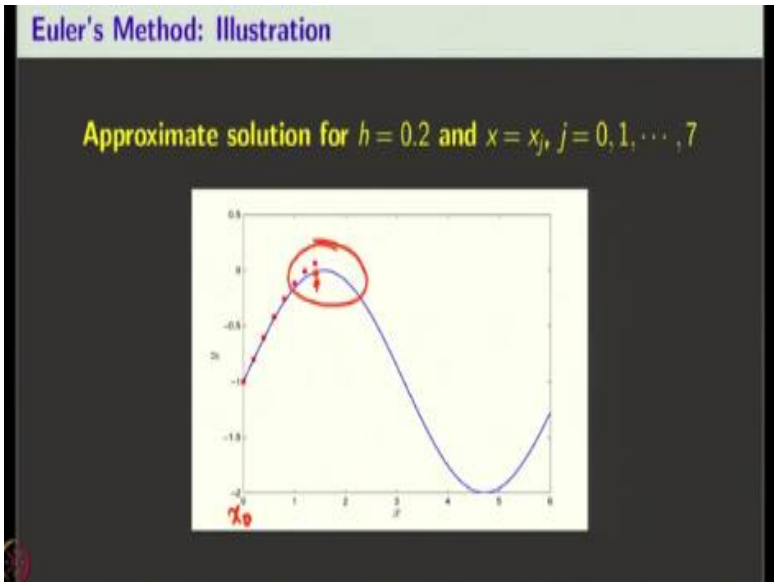
(Refer Slide Time: 35:35)



Similarly, this is $y(x_2)$, which we call as y_2, this is y_3 and so on. You can see that the solution is nicely coinciding with the exact solution, this is because from the geometrical interpretation of Euler method. We have seen that the Euler method takes the tangent line and tries to approximate the solution value along the tangent line. Therefore, on those regions where the solution is almost linear, something which is happening here, you will see that the Euler method gives better approximation.

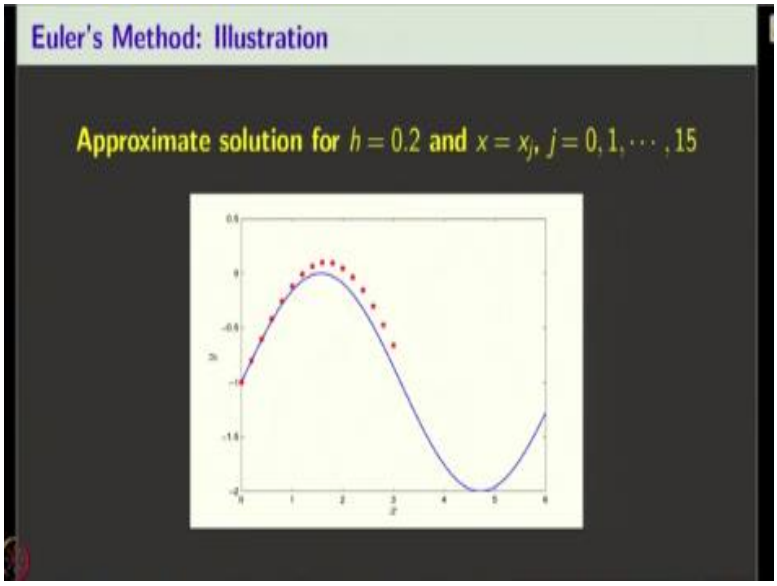
However, if your solution has too much of curvature, you will see that the Euler method gives poor approximation.

(Refer Slide Time: 36:30)



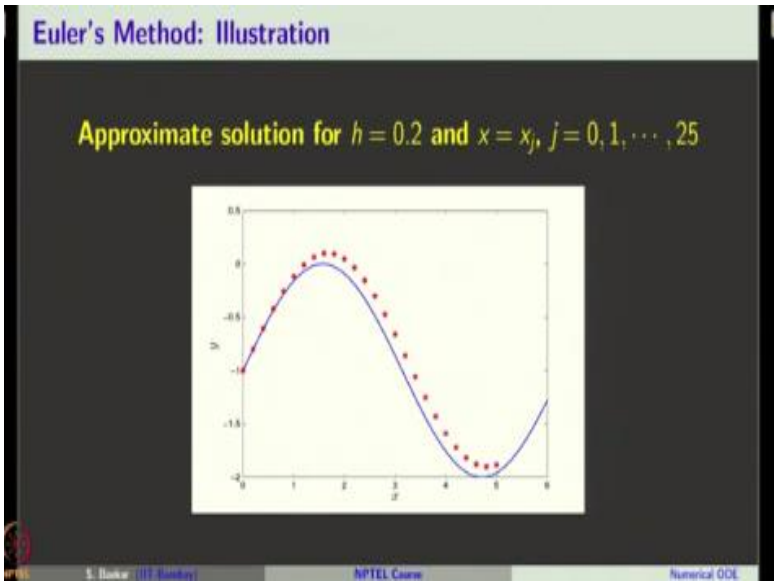
Let us go on, you can see that as you go on computing the solution for different x_i's lying on the right side of x_0, you see when you come to this region, the Euler method is giving rather a poor approximation. Means the exact solution, at least at this point, is this but your Euler method is giving this as the computed solution.

(Refer Slide Time: 36:57)



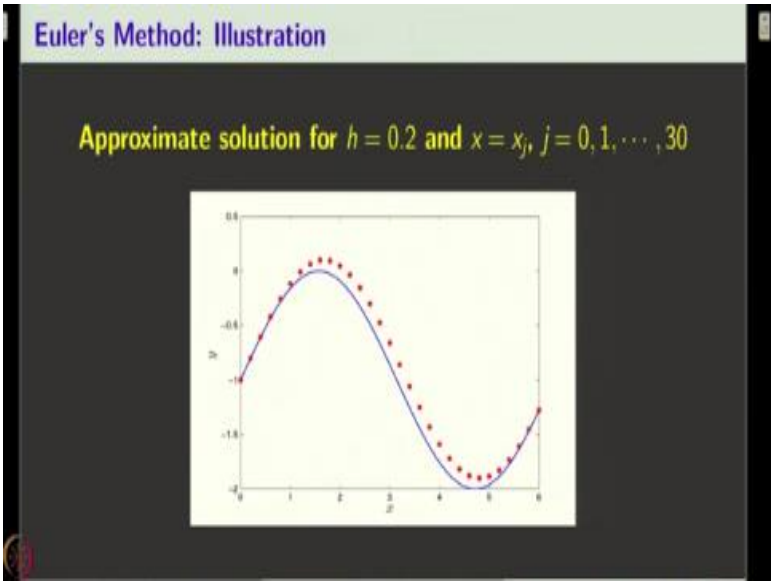
So, I am just going on with my computation with x_1, x_2, x_3 and so on.

(Refer Slide time: 37:06)



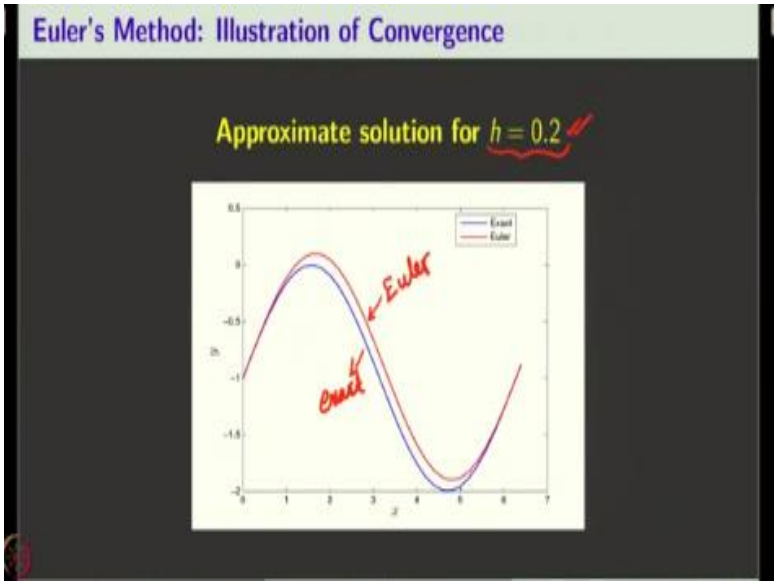
And you can see that the values obtained from the forward Euler method are shown in the red dots.

(Refer slide Time: 37:16)



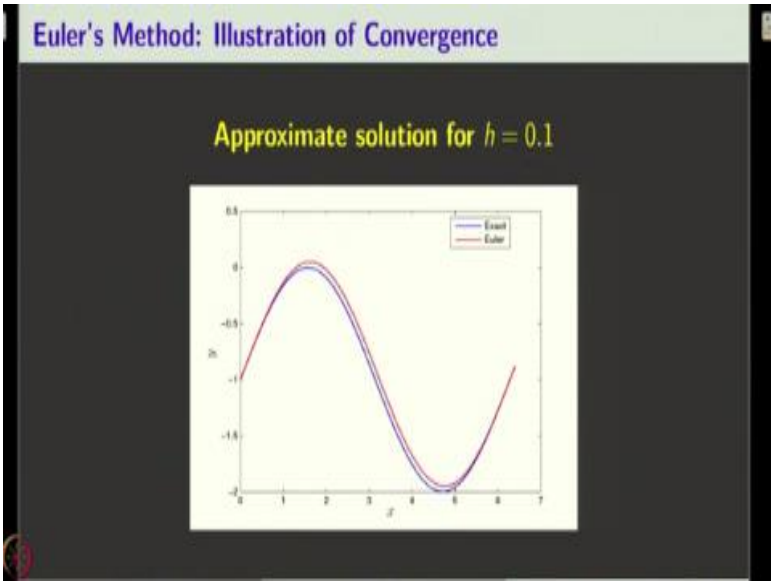
And you can see again when you come to this region your approximation is rather better, because again in this region you have an almost linear behaviour for the exact solution.

(Refer Slide Time: 37:31)



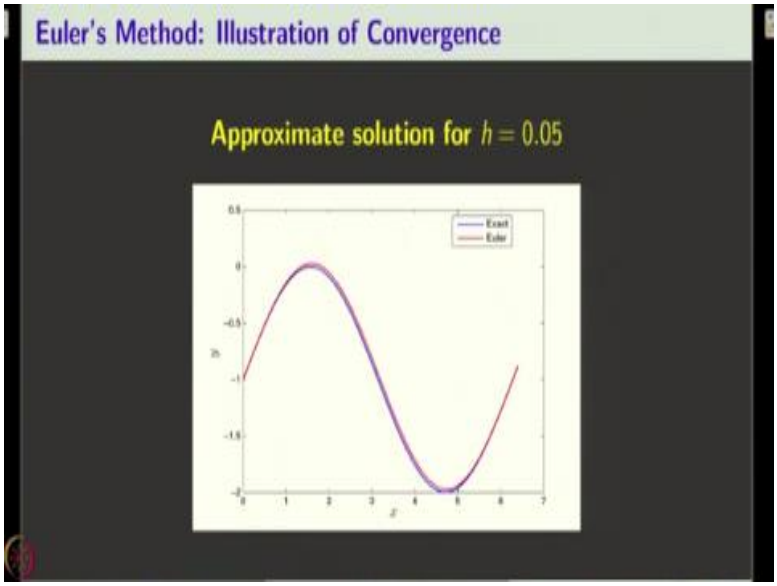
So, let us see how this Euler method approximates the solution for different values of *h*. Let us start with *h* = 0.2, the solid red line shows the Euler solution and blue line is the exact solution. Now I am going to decrease the value of *h* and see how the Euler solution looks like.

(Refer Slide Time: 38:00)



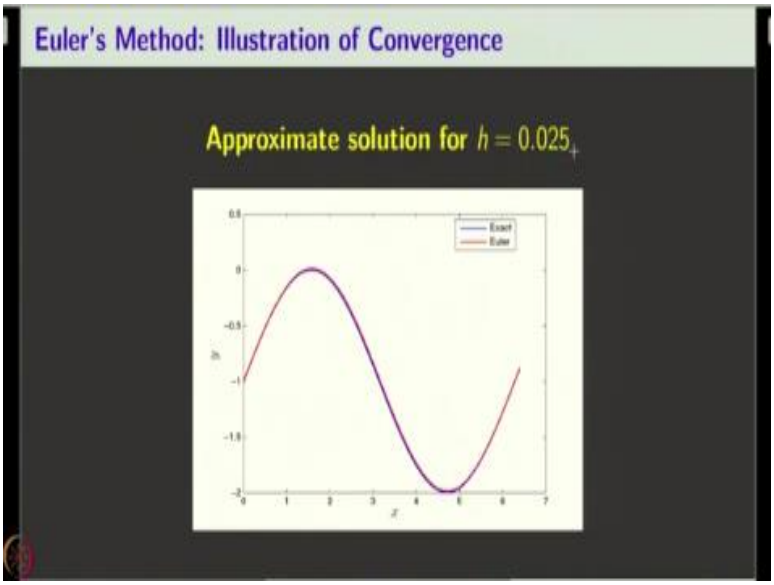
For *h* = 0.1, the Euler solution is little closer to the exact solution, than what we got when we took *h* = 0.2.

(Refer Slide Time: 38:15)



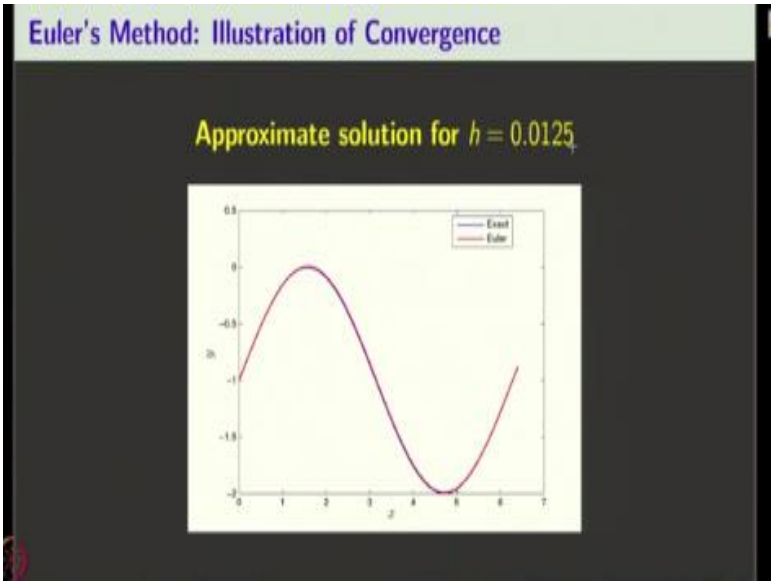
Then I will go on decreasing *h*, now I take *h* = 0.05. You can see that the approximation is significantly improved from *h* = 0.1 to *h* = 0.05.

(Refer Slide Time: 38:30)

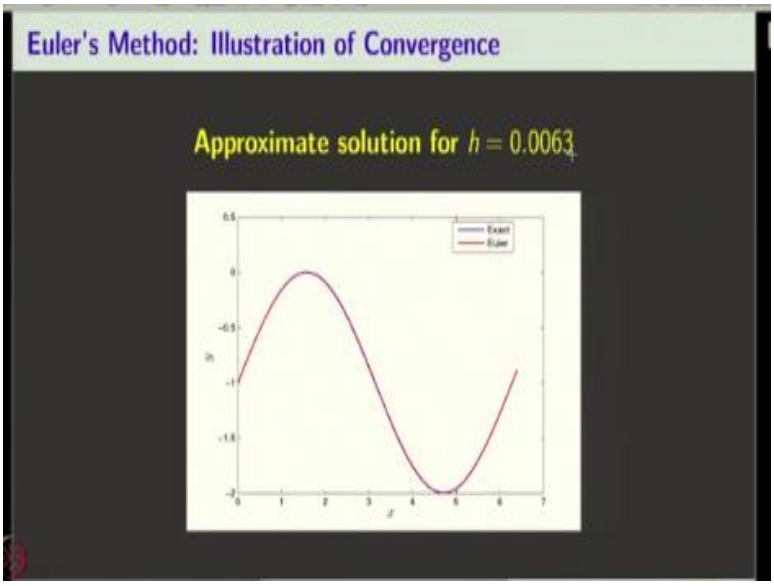


Further, let us reduce *h*. You can see that Euler method is giving better and better result as you go on decreasing *h*.

(Refer Slide Time: 38:36)



(Refer Slide Time: 38:38)



I went up to $h = 0.0063$, here at least visually you can see that the Euler solution is nicely coinciding with the exact solution. So, this is what we say that Euler method is point wise converging to the exact solution. That is, you take any particular point, your Euler solution is this whereas the exact solution is this and as you go on decreasing your h, they both coincides very well.

Although I show you in this plot, a continuous solid line for Euler solution, it is actually computed only at some discrete grid points. Rest all are linearly joined to show a continuous solid line. I hope you understood how to develop Euler method and how the Euler method works both geometrically and numerically. In the next class, we will learn how to analyse the error involved

in the Euler method, both mathematical error as well as the truncation error. Thank you for your attention.