Welcome, everybody, to Scientific Computing Using Python. So the topic we are going to discuss today is the last topic of this course. So today we will see how we can solve boundary problems numerically. So let's get started. So what we have done so far in this course, in the previous lectures, we have solved initial value problems. So in initial value problems, we have a solution given at some initial point and we keep approximating the solution numerically iteratively at every next step. But what we are going to do today is called boundary problem. His problem with short form boundaries. So today we have to see how to solve this. So we generally write the boundary related problems in this form. So like y is. y is a function of x, so we will write

$$d^2y / dx^2 + p(x) \, dy / dx + Q(x) \, y = f(x)$$

and the differential equations that we are taking now in this differential equation boundary problem are linear. And what do I do with it? I multiply it by a constant k. And our x is defined in some close interval from a to b. So this is our second order linear differential equation. What do we do now? In this, I will apply boundary conditions. So we will impose the boundary conditions. I am going to impose a general boundary condition. That is

$$\alpha_0 \, y_a \; + \; \beta_0 y'_a \; = \; \gamma_0$$

$$\alpha_1 \, y_b \; + \; \beta_1 y'_b \; = \; \gamma_1$$

ok so what does y at a mean, here y_a means y at a and y' means took the derivative and we found out its value on a, similarly y_b means y at b and y' is the derivative b of b. So the condition that we have taken, the boundary condition, has been taken in this manner. So this boundary condition is what we call mixed boundary condition. Boundary conditions. So you will see what we will become if we do it differently. Now and we are assuming that p(x) q(x), f(x) two, this function is a continuous function. So this means that the solution of this differential equation, which is a boundary problem, exists and it is a unique solution. So now we have to find this out. what shall we do now? I will take case one. If $\beta_0$ is 0 $\beta_1$ is 0 $\alpha_0$ is one and $\alpha_1$ is one, if we take this, then what will happen, then our boundary conditions will become y at a that is ya and that is gamma 0 and y at b that is y(b) we will call it gamma one. So if such boundary conditions are given to us, then we call it Dirichlet boundary conditions. Ok? So we will take case two. What are we doing in that? Suppose if alpha becomes 0, beta becomes 0, gamma becomes 1, oh sorry $\alpha_1$ becomes zero and $\beta_1$ becomes one, then what will happen in this case, in this case our boundary conditions will become y'(a) = y'_a , which we have written like this, y(a) will become gamma 0 and then we have written y'b, it will become one. So if we are given this type of boundary conditions in which derivatives are getting involved, then this type of boundary conditions are called Neumann type boundary

conditions. So this is a Dirichlet-type boundary condition. This is a Neumann-type boundary condition.

# Num. ODE — Boundary Value Problem (BVP)

$$K \frac{d^2 y}{dx^2} + P(x) \frac{dy}{dx} + Q(x) y = f(x) \qquad a \le x \le b$$

mixed B.C
$$\begin{cases} \alpha_0 y_a + \beta_0 y'_a = r_0 \\ \alpha_1 y_b + \beta_1 y'_b = r_1 \end{cases} \qquad y_a = y(a), \quad y'_a = y'(a) \\ \qquad\qquad\qquad\qquad y_b = y(b) \quad y'_b = y'(b)$$

Case ① If $\beta_0 = 0$, $\beta_1 = 0$, $\alpha_0 = 1$, $\alpha_1 = 1$

B.C
$$\left. \begin{array}{l} y(a) = y_a = r_0 \\ y(b) = y_b = r_1 \end{array} \right\} \text{ Dirichlet B.C}$$

Case ② If $\alpha_0 = 0$, $\beta_0 = 1$, $\alpha_1 = 0$, $\beta_1 = 1$

$$\Rightarrow \text{ B.C} \quad \left. \begin{array}{l} y'(a) = y'_a = r_0 \\ y'(b) = y'_b = r_1 \end{array} \right\} \text{ Neumann type B.C}$$

NPTEL

(6:11)

And if we combine both of these three, like it is, then we call it mixed boundary condition. And in some books it is also called Robin type boundary condition. But this is a mixed boundary condition. So if we solve this equation with the mixed boundary conditions. So there are different cases which we can find out from this. So what to do now in this case? We have to solve the boundary problem. So the other methods we use to solve this are analytical methods. You must have solved these with power series and if p(x) q(x) will become constant. So we can solve this by finding it out from the electrical equation. So but what we have to do is that we take a general equation, so I will name it one. And this boundary condition is our boundary condition. Mix is the boundary condition. So we'll take this. If we take case one case two we'll take that.
 Now we have to solve this. So what do we need to do to solve this ? This has to be solved numerically. So what's the first thing to solve numerically? That we have to see its domain first. We are doing its domain in One D. So these a and b, here we are given the value of the solution or the value of its derivative at this point and at this point. What do we do now? The solution in between this has to be found out. So suppose its solution is something like this or there is a solution of this which is like this or else I take some third solution. Any solution is like this. But our values are given here and here. What do we do now ? In between we have to see what the session will be? So what do we do for this? I will divide this into n sub intervals. Ok? So what did I do? The length of b - a, I divided it by n. And that's my h here it is. The width has arrived. So we divided it into n number of sub intervals. And this point, my a, is my x0, this is my x1, x2, x3 and at the last, this is my xn. Ok? So this means total is n + 1 points. So this which we have, we will say that this is x0 < x1 < x2 and it will go on like this. In the last there is xn, so this is our grid and mesh and the mesh width is we are making it uniform, so in this case it is this. So what did we do? This is how we divided it. Now we have

to approximate the solution. So we will see what solution comes at this point? What will happen at this point ? What will happen at this point? This is how we will find out the solution. Now, in this first initial value problem, the solution was given here, so we calculated it here, then calculated it here, then calculated it here, but in the boundary problem, one solution is non here and one is non here, so we will find out all the solutions in between in one go. So what will happen in this is that in these problems, we will find out the solution in one go on the entire mesh that we have generated. So now what we have to do is our equation one, so what is step one, we have to discretize equation one, so what is the meaning of discretization, our equation for the boundary problem is given in the form of continuous, but we have to discretize it above this match point. So what do we have to do now? The first derivative needs to be approximated. The second derivative needs to be approximated. Ok? So this is what we have defined above. So what do we do now? I have to see its approximation. I have to see its approximation. And we know that. So what we are doing, we are doing this with the help of finite difference method, we are using FDM, there are many different methods to solve this. There is also the finite element method. There are spectral methods. Ok? There are many methods. But we are making it a finite difference. So look at us now, what did we do? Now that we know what the first derivative is, how do we approximate the first derivative ? So this means I am approximating it. So this has to be approximated at any rectangle at the ith mesh point, okay so this means I can write it like this $y'$ at some $x_i$, its approximate value, if I write it like this $\{y_{(i+1)} - y_i\}/h$ then we know that this is the order of h, so I can do it like this. Here comes our forward. I can also do this backwards. $\{y_i - y_{(i-1)}\}/h$ + order of h This is done backwards. I can do this from the centre also. That is $\{y_{(i+1)} - y_{(i-1)}/2h$ this is the order of hv. And this is the centre.



(12:52)

So we can approximate our derivative in any way. Now let me do the second derivative like this. Ok? Ok? So if I want to approximate the second derivative, then I want to approximate it. Now the second derivative means $y''$ at xi. I want to approximate it. So there are different methods to approximate this. So for example, if I write this as $y_i - 2y_{(i+1)} + y_{(i+2)}/h^2$ then this will be the order of h and this is forward. I can do this backwards also. $y_i - 2y_{(i-}$

1)+ y_(i - 2) / h²  and its approximation is also of the order of h. This is backwards and if we wanted to do it from the center, it would be y_(i – 1) - 2y_i + y_(i + 1) / h² and this is the order of s², cent. Ok? So what do we do now? With its help we have to approximate it. So what we do with this is we want a little higher order approximation, we know that it is better. So we approach it with center. So we will call this central final difference. So now see what ? So equation one can be written as Firstly, we will write equation one like this. y' at xi is constant k okay? Ky''(x_i) + P(x_i)y'(x_i) +q(x_i) y(x_i) = f(x_i) . And whatever i is we will take  1 ,2, 3, up to N - 1 Why? Because we already know about the boundaries. So, we have to take it up to this point. Now the approximation that we have to do is to approximate this second derivative to the first derivative. So now let me do this from the center. So is this constant k any? Now look, I am applying it. I wrote it like this: y_(i – 1) - 2y_i + y_(i + 1) / h². Plus now p at xi this is no for us. I can also write this as pi. No problem. Ok ? So I wrote it as pi. So p at xi is pi. Now what did I do to it? I also moved y' from the center. So this will come as yi - 1, let's write it as plus. - yi - 1 by 2h plus qi yi = fy where q at xi is qi and f at xi is fi so what we did was we approximated it, discretized it on the mesh, now we will rearrange it so how will I rearrange, see what we have now, the terms. is in the form i − 1. is in the form of i. is in the form i + 1. That's all there is. So this means that I try to collect all these terms. Now look, we have k here. I first take the value of yi - 1. So k / h² comes from here. And from here we get minus pi / 2h, this is the quotient of yi - 1. Ok? Now let's see what will come from yi? look from yi this is coming and this is coming. So I can write from here yi to - 2k / h² + qi, okay. And this is our yi plus now we have to do yi + 1. So what will be the result? k / h² and this will come out to be plus just pi / 2h yi + 1 = fi and we get it. And now look at our i, we will find it out on the interior points 1 ,2 ,3,…, n - 1.



(18:42)


 Now this is the main question. This has become ours. I give it to him. Now look, if we want to simplify it a little then we can do so. Here we can make it a little easy by multiplying both sides by h². Ok? Because whatever h will be, it will be very small. So if hv happens it will become smaller. And if we do division with the help of some very small numbers, then what happens is that if division is done with small numbers, then sometimes it eliminates the significant digits. Ok? So he reduces it. So that is why we avoid division by the smaller numbers. So this is it, we can write it like this also. I will take the LCM of both. So the LCM

of both will be $2h^2$. So if you see, I can write it like this. $(2k - htp\_i)$ yi - 1 okay then I will do this division here by $2h^2$, here it is $x^2$. Right? So now I will write it like this plus $(-2k + x^2 qi)$ yi / $h^2$ plus and this will become the same $2k + h / 2h^2$ yi + 1 and f we will refine it a little more. So I'll move $2s^2$ over there. So basically this becomes we have $(2k - hp\_i) - 1$ okay? In this, sometimes we also bring out the negative sign to show the positive. So I can write it like this also. Two will come. So here I write this -2 and it will come here, I took the minus sign out, so $2k - h^2 qi$ okay yi plus and this became ours, so let's say it became the same $2x^2 fi$ this is ours. So, our values will come out from here. Ok? So I'll give this equation number three. Now how to deal with boundary conditions. Now we have to see the boundary conditions and how to solve it. So what should we do? Now there are two ways to solve the boundary condition. Ok ? Two ways. Two ways we can solve it. So I will calculate both. First one, I can write this. What should I write for this also ? Let me write down case one. Now we have this condition. If we look at the above, what will we do now? They will solve this directly, this mixed boundary. Ok? And then depending upon whether their coefficients are to be taken as zero or one. Those cases will come to us separately. So, we have alpha 0, y 0 and we want to calculate it like this. Now see how to do it. Now what do we have in this case ? alpha 0 y et a okay? So look at y at a and what is a? That is x0, if we assume it to be y0, then we will write it as y0, so here we have written y0 plus beta0, now let us see, what does y' at a mean, y' at x0, this is ours, now we cannot go back from this, okay brother, the derivative we have is on x0. Now we don't have any approximation behind x0. There is no mash point. x0 will be followed by x1. so what does that mean? What are we going to do here? I will approach this through forward. And what will be the forward? brother what are we doing ? I'm taking y' at x0. So what does one have to do to forward? I'll go to the next step in approximation. - y0 / h It will be done. Ok? So here I will write this as y1 - y0 / h and equal to gamma 0 now what will be the second one? alpha1 will become yn. Why? Because what we have is y at b and what is that? y is at xn and we will write that as yn, so I have written yn plus beta1 now see, we have given y at d at b so that is equal to y' at xn so if we have any point beyond xn, then we do not have a match point, so what will we have to do, here we will have to go backward and what will be backward, its approximate value will be yn – y(n – 1) / h so this will become yn – y(n – 1) / h and this will become gamma, so we will simplify this also a little. So if you look from here, what do we have ? Let me simplify this a little more. So see from here, we will take h as common, I will multiply it, I will take the terms of both sides y0, so if we look carefully, what will these terms become, h0 so alpha0 h and this will come to be $\alpha 0 * h -$ beta0, this will come to be the coefficient y0, okay plus beta0 * y1 and this becomes, this is what it becomes. What did I do? Multiplied both sides by h. So this will get cancelled. That will bring h here. So h0-beta0 is the coefficient of y0 of y0 + y1 and it arrived. It will come from here like this. If I multiply it by the same h, what terms will we get? yn - will come to us - beta1 yn - 1 + h1 + beta1 yn = h gamma1 It will come. Ok. So what do we have? This became a boundary condition. From here, I named it Four. So what do we do now? We got these equations by applying the final difference. Now let's see how to solve this. So now what do we need to do to solve this? 3 & 4 First, we have to see how the system was created. Now look, if I take i as one, what will come from here? From here we will get $(2k - hp\_1)$ y0 - 2 $(2k - h^2 q\_1)$ y_1 + $(2k + hp\_1)$ y2 = $2h^2$ f1. This is the first equation. then look for i = 2 what happens now? The forest will come. So this is going to be $2k - hp\_2$ and here comes y1, okay? - 2 $2k - h^2 q\_2$ This will come to y2 + $2k + hp\_2$ y3 And this becomes our $2h^2 f\_2$. So we will keep calculating like this here, ultimately we have to go till i which is $(n - 1)$, so let us see what will come in n - 1, so this will come as $2k - h\ p\_{(n-1)}$ Now y which is i - 1 will come as $n - 2(2k - h^2 q\_{(n-1)})$ y_{(n-1)} + $(2k + h\ p\_{(n-1)})$ this will become yn and this will become $2h^2 f\_{(n-1)}$, so if we look at it, we have n - 1 equations. So there are n - 1 equations.

And what about those unknowns that are ours? What is Anonymous? y0, y1, y2 ,….yn So what do we have? We have to find out all the values of y0, value of y1, value of y2. Because the boundary conditions which have been given to us are given to us. So we do n't know what the exact value of y0 is. Ok? So now it depends on how we will calculate this. Ok? OK, so we have a total of n + 1 unknowns. Now we have got this as n - 1 equation. So what do we do? Now we need two more equations. So we will get two equations from here. Ok? So the first equation, now I am writing it in red. So, we will write this first. Ok? So the equation that will come from here will be this (α0 - beta0) y0 + beta_0 *y1 = beta_0 y1. So here comes our boundary condition. From here, I am writing it here. Inside, the boundary condition will come to us - beta1. I should not write it here, I will write it here. Here - beta1 y_(n − 1) + (h1 + beta1) yn = h*gamma_1 So now we have n + 1 equations and n + 1 unknowns. So if we have this, then we can calculate it now. Now it depends on what kind of system we will have, if we have to solve this. So what will happen in this?



(32:20)

We will have a matrix. Now see what is happening in the matrix ? The first terms here we have a vector. The vector is y0, y1, y2,… yn equal to the right hand side. So now see, in the first I am writing here (α0 h - beta0) y0 Here the second term will come beta0 and this will come h * gamma0. Okay, after that see, the first boundary condition which we used has come, after that see what is coming, it is coming here (2k – hp_1)  -2( 2k - h² q_1) The next thing that will come is (2k + h p_1), this has come and here 2h² f1 will come then what will happen in the next? It will shift a little and come here. So this will become (2k - hp2), next will come -2 (2k - h² q2) and next will come (2k + h p_2), and here 2h² f2 will come and it will go on like this. Now we will see that there are only three entries in each row here. So one entry is on the diagonal. One is below him and one is above him. Ok ? And what will we have here in the end ? This will become (h*alpha_1 + beta_1) and this is - beta1 so this becomes our h*gamma_1 the entry that comes above it will be (2k + h* p_(n − 1)) here this -2(2k - h² q_(n − 1)) and here this will come (2k + h*p_(n − 1)) and here 2h² f_(n − 1) will come ok so what do we have? Now this becomes a system of equations. So the matrix that we have in it is a special matrix. If we look at this matrix carefully, it has only three entries on

the main diagonal. One above the main diagonal and one below the main diagonal. So this type of matrix is called tri diagonal. So this is our tri diagonal matrix. So what will you do with it now? We have matrix A. This is y here, and we have our right hand side function here. r I will write it down. Ok? So what will we do with this? We will solve it. So if you look carefully at this matrix, in this case it will be a diagonal dominant matrix. Ok? So this matrix is diagonally dominant. So it has a solution and we can find out a unique solution for it. So we will find this out with the help of the method that we have developed, Gauss elimination. So we will do this by Gauss elimination. Because the gas elimination method is quite expensive. The order of h occurs. But for a triangular matrix it remains just the order of h. So that's why we can find this out by Gauss elimination. If you don't want to do Gauss elimination then you can solve this system by any method like Gauss seidel, Gauss jacobi and we will get the solution that we have. Ok? So this is our case one. Now in this case, you see, now look at the observation here, what if in this case we have alpha, beta 0, which is beta and gamma becomes 0, then what will happen if beta 0 and beta 1 both are 0 and alpha becomes 0 and alpha 1. So from here we will have these conditions. See, this condition has become zero. Alpha 0 became 1. So if you see, here we will have y0 is gamma 0 and yn is gamma_1 and what is this? Dirichlet Boundary Condition So what happens to us in this ? The solution is non-existent on the boundaries. Ok? So what will happen if we do this ? If beta 0 and beta 1 both become zero then see what will happen here? Beta 0 this became 0. If I show this even a little, it will become zero. Alpha 0 will become 1. So only h will be left here. This will also become zero. And what is this gamma 0 ? This is known to us and h. So h and h will get cancelled and this will come from here. What will you find here? If $h \cdot y0 = h \cdot gamma\_0$ then h cancels out. So from here ultimately we get $y0 = gamma\_0$. So look, it has come here. Same thing will happen here. If beta1 here becomes zero, then beta1 also becomes zero. This also became zero. This left h1 and equal to so h1 will come here and $y\_n = gamma\_1$ which is one so h gets cancelled so from here we get yn =1 so ultimately it has satisfied the boundary condition, okay so what will happen in this is that this will ultimately happen. In this we know the values of the solution initially and final on that. So we can calculate from this also and we can also do this matrix in $n / (n-1) \cdot (n-1)$ . Ok? So, we solved it accordingly.



(38:05)

Secondly, if our alpha is 0 and alpha1 is also zero and beta 0 beta1, both of them are one, okay, so both of them have become zero, this has come, so this has become purely Neumann type, okay, so what happens from here, there will be purely Neumann type boundary conditions and according to that, we will put $alpha\_0$ and $alpha\_1 = 0$ here, accordingly we will get its value. So even if we have these given separately we can do it and even if we have mixed givens we can do it. So generally speaking, what is the problem with what we are doing? So these are the methods, now see what we did, we applied the central fine difference here, so I applied the central fine difference at both the places, here as well as here, so what does it mean here and here both of them will give the approximation of the order of $h^2$. Ok? So this is the approximation of the order of $h^2$. Now we have done this work but what is the problem ? This approximation that we took here was of the order of H. This was also the Order of H. We know this, forward and center were placed. So what happened is that we used this and combined both of them, so the combined effect that we got from this method will be of order of h approximately. Why? Because on the boundary the order was of h and in the middle the order was of $h^2$. So if we take the composite, combine all of them , and look at the combined effect then the order of h will come. So, what happened with this? Our approximation has become a little less. Meaning it will come with some accuracy. If we have a differential equation and its solution is a linear function then it will give the exact solution but as soon as a little non-linearity comes in, we will get an error. So this problem which will arise, this problem occurs in this case. So what do we do to avoid this problem ? Now whatever methods we took, what did we do? I took case one. Ok? Second derivative First derivative Yes, we took this and this was ours, after that we followed these steps and we solved it. Right? Now what we have is the next thing I write here is to improve accuracy up to the order of $h^2$ so if I want accuracy of the order of $h^2$ what do I do? This approximation that we made, I will also do it with second order approximation. What does it mean? I will apply central final difference here also. So if I want to apply the central final difference in this case, then look here, if I am applying it, then if you understand from here, then we have the value y' at x0 now if I want to do this from the center then what I will have to do is y1 - I will go ahead one because x0 is here. Here's x1. So one goes forward but one also goes backward. So backwards means I will have to go backwards from y0 and here I will have to write 2h. so what does that mean? First, we need to create a fictitious point here. So that we can approximate it by central fine difference only. Ok? So there are methods of this type in which we will increase its accuracy. So if we want to increase the accuracy then what will we have to do? A Fictitious point or also known as Ghost point. He will have to be introduced. Ok. So, what am I doing now ? See, my x was 0 and this was $x\_n$. We already have this. So, what am I doing now? I am going back one. I am writing x - 1 and we will have to go one further because we had a boundary here also. So, here I will write $x\_(n + 1)$ and these points are this one and this one was closed, right. Ok? So if we look at these points, we will show that this and this are our fictitious points. There are ghost points. Ok? So we introduced this just to increase its accuracy. Now the question arises that here the solution is unknown. We don't know anything. Is it okay? So how to find it out now? So this is the main problem. Ok? So let us now see how we can use it to increase its accuracy. So you see, first let me look here. What am I doing now? Here 's the x1. So what am I doing? Here it was h, so I go back that much, h, so let me take the x - 1, that is $x\_0 - h$, okay and here we took $x\_n + 1$, that is $x\_n + h$, whatever it is, it will be the same in a uniform match, even if it is non-uniform, then also whatever it is, we will take that much here, now see, this was our boundary condition. So if we look at the boundary conditions, what did we have? alpha0 y0 + beta 0 and this becomes $\{y1 - y\_(- 1)\} /2h = gamma\_0$. So from here we will calculate, I can write this from here, look, from here I will multiply both the sides by 2H so it will come to $- beta\_0 y\_{- 1}$

+ (2 h\_0 + sorry here it comes to -1, okay? After that this will come 2 h\_0 *y\_0 + β0* y1 = 2h*gamma\_0 this will come. So in this we have y0 also coming , y1 also coming, y\_{- 1} also coming. Ok? So now we have named it equation. Suppose I gave it. Now what to do next? Solving the differential equation at x this was our differential equation above now I have to solve this at x0 so x 0 because see we started with i which is one. Now what happens if I take i to be 0, what does this become? y will become - 1. Is it okay? So if we pay attention, I am writing this directly from here. So you take i as 0. So what will come? (2k – h*p\_0) will come from here (2k – h*p0) y\_{- 1} - 2 what else was there? (2k - h² *q0)y0 + (2k + h p0) y1 = 2h² f0 Okay, this was on the right side 2h² f0 So here we have this equation, I will give it the name b So now you will see that we have two equations a and b y\_{- 1} ,y0 ,y1 ,y\_{- 1} y0 y1 Like this now what we will do is now eliminate y\_{- 1} from equation a and b from these two equations. We will eliminate it from here. Why? Because the value we have y \_{- 1} is our fictitious point, we took it. So we never know these values. So what will we do with this point, this thing ?



I will eliminate you. So what will you do now for elimination? Very simply we can do this? What can I do ? Let me multiply this and add it. So what can we do? I can do this. See, a is multiplied by 2k - hp0 plus b is multiplied by beta0, this is done by b with beta0, this is 2k - and both are added, so what will happen after adding, see, 2k – h*p0 will come here and beta0 will come here, okay and if both the adds are added then it will get eliminated. So what does the equation that we have become? We will have it (2h alpha\_0)( 2k – h*p0) okay. - 2 beta0 (2k - h²* q0) this comes to y0 okay plus [beta0 (2k – h*p0) because it was multiplied and here beta0 (2k + h*p0)] because I multiplied b not from here. Here we have y1 equal to and this becomes our 2h*gamma0( 2k – h*p0) + beta0 and this becomes our (2h² +f0) so we have this, okay so now let us look at this, you can simplify it a bit so if you see from here, beta0, this will cancel out so this and this will cancel out so 2k + 2k= 4k beta0 will come similarly we will calculate it from here also so whatever terms we get, they will come so ultimately what we have is that we have a coefficient y0 and one coefficient and we get y1 = something so now whatever things we have are starting from y not and we should start from

there because we don't know anything beyond that. Ok? So, I can name this equation as, let me name it. I will give it to you. It has come to us. Now same similarly at $x_{(n+1)}$, now we have to see what will happen to this? This is a fictitious point. So if you look at the fictitious point, what will we have? What do we have? What will happen to the second boundary which is the right side boundary condition ? So on that we have $\alpha 1\ y_n + beta1$, Now look what will come here? $y_{(n+1)} - y_{(n-1)}/\ 2h = gamma1$ This will come out. Why? Because the values that we have, now look at where this is, now we have to do it from the center, so n was here, then n + 1 will go ahead, one will go behind it and it will become 2h. Ok? So this quantity will come here. So now look what has happened here ? Now I multiply both sides by 2h. So this comes to $- \beta 1\ y_{(n-1)} + 2h*alpha1\ y_n + \beta 1 * y_{(n+1)} * 2h*gamma1$. This is what we have. Ok? So here we have the boundary condition on the right hand side. Similarly, now we have to solve this. C and I'll take this D. Now another equation is also needed to eliminate these. So what should we do? Using this same equation that we had, I will now apply it to i = n. So if do it for i = n then it will come out to be $(2k - h*p\_n)\ y_{(n-1)}$ okay? It was -2. What happened next? $(2k - h^2 *q\_n)y\_n + (2k + h\ p\_n)\ y_{(n+1)} = 2h^2\ f\_n$ Now, I will take this. Now you will see that here this thing that we have, this quantity, this one and this one, both are unknown. Ok? We do not know what their value will be because these are fictitious values. So what do we do? Now we have to eliminate $y_{(n+1)}$ from equations d and e. We will have to eliminate it from both of them. How will you eliminate them now? what shall we do now? what do I do to d? I will multiply it by $(2k + h*p\_n)$, I will multiply d and subtract e by beta1 because it is beta1 and its coefficient is this and we will subtract both of them from each other so this will get eliminated, so if we see from here, what do we have - beta1 , I multiplied it by $(2k + h*p\_n)$ minus, I multiplied it by beta1 and it came out $(2k - h*p\_n)$, this is our coefficient of $y_{(n-1)} + 2h\ alpha1$, I multiplied it by $(2k + h *p\_n)$, okay minus and multiplied this coefficient by beta1, okay so how much did we get here, this $-2\ (2k - h^2\ q\_n)$ and this quantity that we got, we got of y_n, okay why because I will multiply it by beta1 and then we will subtract both of them from it. So if we subtract then it is obvious that the minus sign here will come and this coefficient is definitely there. And what will this equal to two become? This becomes, we have $2h*gamma\_1\ (2k + h*p\_n) - beta1\ (2h^2 * fn)$. Right? So we did it like this. So, we will calculate all these quantities. Some part of this will be eliminated. Like this is what is happening. It's a minus, so -2k and -2k is getting cancelled from here. Just like this we can find it out. And because we will have to be a little careful. There should not be any mistakes in this. But if there is any, you can understand that

somewhere some negative sign or positive sign is happening,



$$\boxed{C} \Rightarrow \left[(2-h\alpha_0)(2k-h\rho_0) - 2\rho_0(2k-h^2q_0)\right]y_0 + \left[\rho_0(2k-h\rho_0) + \rho_0(2k+h\rho_0)\right]y_1$$
$$= 2hr_0(2k-h\rho_0) + \rho_0(2h^2\rho_0)$$

Similarly at $\boxed{x_{N+1}}$

$$\alpha_1 y_N + \beta_1 \frac{y_{N+1} - y_{N-1}}{2h} = r_1$$

$$\Rightarrow \boxed{-\beta_1 y_{N-1} + 2h\alpha_1 y_N + \beta_1 y_{N+1} = 2h\,r_1} \quad - \boxed{D}$$

$$(2k - h\rho_N)y_{N-1} - 2(2k - h^2 q_N)y_N + (2k + h\rho_N)y_{N+1} = 2h^2 f_N \quad - \boxed{E}$$

Eliminate $y_{N+1}$ from Eq. $\boxed{D}$ & $\boxed{E}$

$$(2k + h\rho_N)\boxed{D} - \beta_1 \boxed{E}$$

$$\Rightarrow \left[-\beta_1(2k+h\rho_N) - \beta_1(2k-h\rho_N)\right]y_{N-1} + \left[2h\alpha_1(2k+h\rho_N) - \beta_1(-2(2k-h^2 q_N))\right]y_N$$

$$= 2h\,r_1(2k+h\rho_N) - \beta_1(2h^2 f_N)$$

(59.47)

but what we have done is we have multiplied it by this and multiplied it by beta1, that's fine. So we will calculate this. Ok ? So from here we will name it. Like I have done above C de and this gives us F. From here we get this equation. So now you'll see what we have in this case is that this is the quantity coefficient. This is y of n - 1. This coefficient is of yn and is equal to the right hand side, this is. And everything on the right hand side is non. So after reducing and eliminating what we have, we get this. So if we pay attention to what we have now? Now we have (n + 1) cross (n + 1) cross this becomes a system and here our quantities are y0, y1, …..yn and equal to the right hand side. Ok? So what will be the first value that will come on the right hand side ? This quantity will come in the first place. Why? Because we will first take this equation at y0. So, what do we do? The coefficient of y0 is to be written here. We have to substitute the coefficient of y1 here. And the right hand side vector that is coming, we have to write it here. After that, it will continue in the same way as our process was, the same values will come there and whatever will happen in the end, what will happen in the end? This is the quantity of yn that will come here. This coefficient, the coefficient of $y_{(n-1)}$, will come here, this one and the right hand side vector will come here. So this will come in between. So we will have this tri diagonal system. And with the help of our iterative methods, or direct method, or by Gauss elimination, we will solve it and we will get our solution. So this is the theory we have just presented.

$$(2k - hp_N)y_{N-1} - 2(2k - h^2 q_N)y_N + (2k + hp_N)y_{N+1} = 2h^2 f_N \quad - (E)$$

Eliminate $y_{N+1}$ from Eq. $(D)$ & $(E)$

$$(2k + hp_N)D - \beta_1(E)$$

$$\Rightarrow \boxed{F} - \left[\beta_1(2k + hp_N) - \beta_1(2k - hp_N)\right]y_{N-1} + \left[2h\alpha_1(2k + hp_N) - \beta_1(-2(2k - h^2 q_N))\right]y_N$$

$$= 2h\gamma_1(2k + hp_N) - \beta_1(2h^2 f_N)$$

$$\begin{bmatrix} \cdot \cdot \cdot \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vert \\ \vert \\ \\ y_N \end{bmatrix} = \begin{bmatrix} \cdot \\ \\ \\ \\ \\ \cdot \end{bmatrix}$$

$(N+1) \times (N+1)$

Tridiagonal system

(1:02:37)

So now let us implement it a little in Python. How can we find this out in Python? So look, this is what we have done. We have solved the problem with second order boundary for this equation. Well, instead of k I have written epsillon. No problem. So we have written p(x), q(x) r(x) f as r. And these quantities are given to us. Now see what to do in this? If we find it numpy then we have to use it. Matplot also has to be used. So I defined a function there called solve boundary problem. In this we will input the coefficient. p q r a b alpha0 beta 0 gamma 0 alpha1 beta1 gamma and we will have to input all these because all these things are known, we will input it here now what do we have, we have to find out h, so in h I had told b - a / n so what we will do is I will find a mesh x, I will divide it from a to b n + 1 into points, whose initial point will be a and the last point will be b. Because all n are intervals, so there will be n + 1 points. What are we doing now? The matrix I am generating has dimensions n + 1 and has all zero elements in it. Because mostly if you see this matrix and tri diagonal, all the elements in it are zero except the three main diagonals, only those in the middle are non-zero. So this type of matrix is called sparse matrix. So what happens in this is that we can solve it very easily. So what did we do? We defined it all from zero. And I have also defined the right hand side vector whose all elements are zeros. This is just to define it.

```python
def solve_bvp(eps, p, q, r, a, b, alpha0, beta0, gamma0, alpha1, beta1, gamma1, N):
    """
    Solves the BVP eps y'' + p(x)y' + q(x)y = r(x) using finite differences.
    """
    h = (b - a) / N
    x = np.linspace(a, b, N + 1)

    A = np.zeros((N + 1, N + 1))
    B = np.zeros(N + 1)

    # Interior points
    for i in range(1, N):   # it will caluclate from 1 till N-1
        xi = x[i]
        A[i, i-1] = eps - p(xi)*h/2   # Left neighbor
        A[i, i] = q(xi)*h**2-2*eps          # Center
        A[i, i+1] = eps + p(xi)*h/2   # Right neighbor
        B[i] = r(xi)*h**2

    # Boundary conditions
    A[0, 0] = alpha0*h - beta0     # Left
    A[0, 1] = beta0
    B[0] = gamma0*h

    A[N , N-1] = -beta1            # Right
    A[N , N ] = alpha1*h + beta1
    B[N] = gamma1*h

    # Solve the system
```

(1:04:16)

Now we will come to the interior points. So let's look at the interior points, we're going to take this range from one to n - 1. So in Python you have seen that the range will not go till n at the end. It goes before that. Ok? So it will go from one to n - 1. So we have made xi into xi. Ok? So x1, x0, x1, x2, x3, we have just changed its notation. Now see, these values will come in the ith row and i - 1 column of the matrix. Ok? epsillon- $p(xi) * h / 2$ if we pay attention. Ok. Now, I have taken the place of K. So if we pay a little attention here, these are the quantities that we have. If I divide this by h then the values we have are basically these. So this is the place of K and what is H is what I am doing here because it depends on us how we write it. So I am taking Epsilon $– p*x\_i*h/2$. So I just moved $h^2$ to the right hand side. Ok? So if I see this here, I can eliminate this from the two as well because there is a two here as well. There is two here as well, so I can take two common from here as well. So what will happen if we take two commons ? $k - h / 2$ this will come. And this is what is written there. Ok? So I can take two as common from here also. Whatever unnecessary things are there, we will take them in common. And look here on the right hand side, there will be just $h^2$. So we'll go this way. So we have defined the first left neighbor. What was that thing in the middle? We have divided this also by two. So here it is $q*h^2 - 2epsilon$ look it is coming here. here comes the. Ok? ok so if I take the negative sign inside then it will come here $h^2 q – 2*epsilon$ so the same quantity has come, ok and the right hand side, which will be above the main diagonal, is just minus, here it will become plus and see on the right side I have written $h^2$ so I cancelled the two as well now what will we do, boundary condition, so for the boundary condition, we are doing the first case. When we approximated it from forward or backward, we did it from forward. Look, a0, which is the quantity, which is the first element of the matrix, will be replaced by alpha0 h - beta 0. We have already seen that. Where is this from? Look at this, alpha0 h - beta0 and then beta0 here it comes. So alpha0 h - beta0 and beta0 and the right side was gamma0 * h so we have the boundary condition on the left. What will come in last? In A[N,N-1]= - beta1 and in A[N,N] = alpha1 h* beta1, this is the quantity we will get there. This - beta1 and this quantity is h + beta1, so see h alpha1 + beta1, so this is our right hand side. So our entire matrix is solved. So now if we want to solve this matrix, then we will write solve in the module of np in linear algebra, give a and b, then it will be returned from there, we have y and x which we have calculated from here, which is our mesh and y is there, okay?

So now we can apply this to different examples. So suppose I took p(x) as 2x, I took q(x) as 2 and r(x) as 5x, okay? So we took this quantity. So if we look at what am I doing now ? I am trying to solve this. I will solve these equations. So the first example that I will take is y" plus y(x) and the x that I am taking is 0 to 0.5. This is a given to us and its boundary conditions are a given. 1.5 It's here. So what this means is that it is in dirichlet type boundary condition. Ok? So the boundary condition that we have in this is the dirichlet type. We will define the value of h. So I defined it here. So I have one because its coefficient is one. Y" of a and b are 0 to 5. We have defined alpha 0 is one beta 0 is 0 gamma 0 is 0 alpha1 is 1 beta1 is 0 and gamma1 in this case is 1.5. So I will write this as 1.5, so gamma1 is 1.5.



```
B[N] = gamma1*n

    # Solve the system
    Y = np.linalg.solve(A, B)
    # print(A,B)
    return x, Y

# Example problem: eps y'' +2x y'+2y = 5x, 0 <= x <= 0.5, with boundary conditions y(0) = 1, y(0.5) = 1.5
def p(x):
    return 2*x  # No first derivative term

def q(x):
    return 2  # y'' - y = x

def r(x):
    return 5*x

eps = 1
a, b = 0, 0.5
alpha0, beta0, gamma0 = 1, 0, 0  # y(0) = 0
alpha1, beta1, gamma1 = 1, 0, 0  # y(1) = 1
N = 40  # Number of internal points

x, Y = solve_bvp(eps,p, q, r, a, b, alpha0, beta0, gamma0, alpha1, beta1, gamma1, N)

# Plot the results
plt.figure(figsize=(8, 5))
plt.plot(x, Y, label='Finite Difference Solution', marker='o')
plt.xlabel('x')
```

(1:09:18)

 Okay and after that I took n from here, I took 40, so I called it. From there we will have x and y returned. After that I will plot it and calculate it. What did I do just like that? Took another example. So the second example we take is like this y' + sinx y' - xy = 4 exponential x is between 0 to 1 and this is the condition given to us. So what will I do now? I defined p. I wrote it as p2. That's sin x this - x taken on the right hand side times 4 exponential. Ok ? a2, b2 is between 0 to 1. And alpha 0 is 1 beta 0 and gamma 0 so 1 ,0 ,0 is given. What do we have in the other? The derivative is given. Y' , y' at 1 is 0 so what's in it? On the left side we have the radical type and on the right side we have the derivative. It is Newman type. So we have this mixed. So alpha 2 which is alpha 1 is zero. Beta1 is one and gamma2, which is gamma, is also zero. So here we get the quantity. Ok? So we are taking this, here y' at one is

0, let's take this.



```
plt.show()

# Example 2: y'' + sinx y' -xy = 4exp(x), 0 <= x <= 1, with y(0) = 1 and y'(1) = 0
def p2(x):
    return np.sin(x)

def q2(x):
    return -x

def r2(x):
    return 4*np.exp(x)

a2, b2 = 0, 1
alpha0_2, beta0_2, gamma0_2 = 1, 0, 0  # y(0) = 1
alpha1_2, beta1_2, gamma1_2 = 0, 1, 0  # y'(1) = 0
N2 = 40

x2, Y2 = solve_bvp(eps, p2, q2, r2, a2, b2, alpha0_2, beta0_2, gamma0_2, alpha1_2, beta1_2, gamma1_2, N2)

# Plot the results
plt.figure(figsize=(8, 5))
plt.plot(x2, Y2, label='Finite Difference Solution', marker='o')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.title('BVP Solution using Finite Differences (Example 2)')
plt.grid()
plt.show()
```
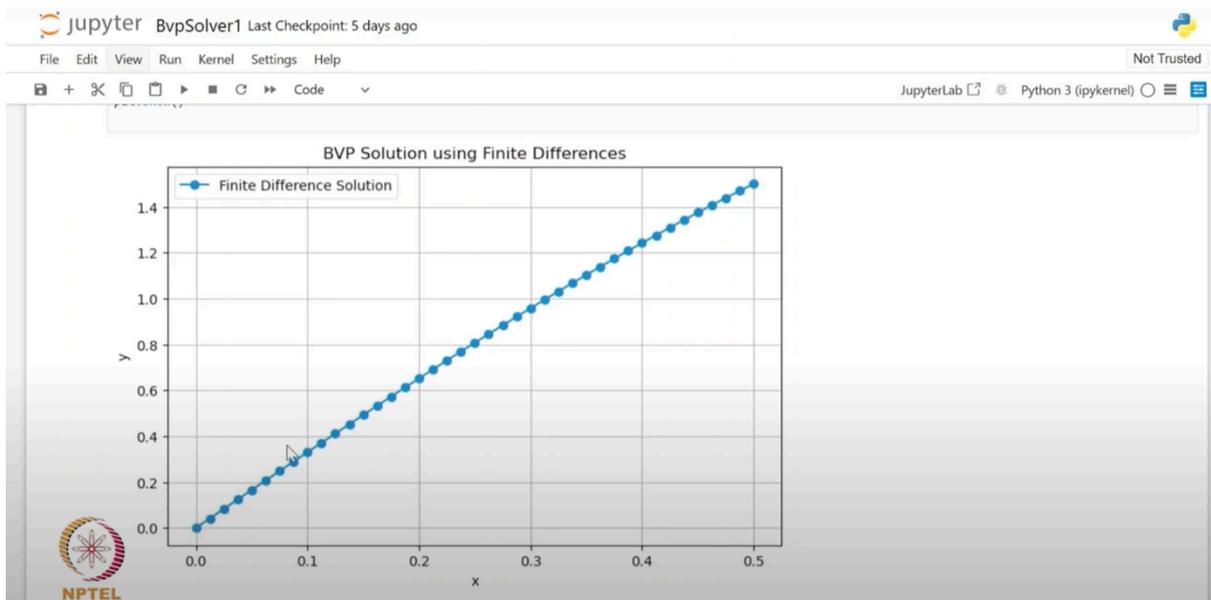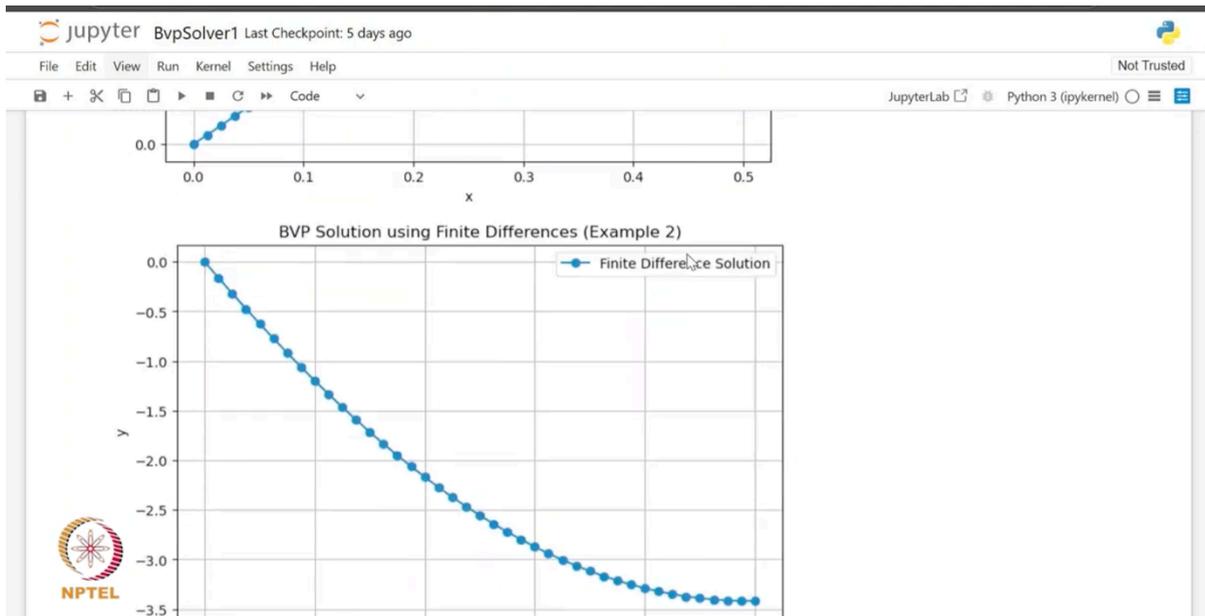
(1:12:41)

And then we called him also and he gave the plot to us. So let 's run this. So I ran.



(1:13:02)

 Look, it has come. So in the first case, the solution we have is of this type. Because the number of points that we have taken is 40 and the second solution is coming in this type.

Now let us see, let me not take too many points and take less. I will take 20. Look at this. Now look what is y0 here ? is zero over zero and what is y over 1.5 ? It is 1.5. The sessions in between the conditions that we have to impose are like this. What was happening in the other one ? y0 is 1 and y'1 is 0. So y0= 1 look this is y0. y 0 Yes, we have taken zero. Ok? So this is zero. So y 0 and y' 0 the derivative of that one is zero. What does derivative zero mean? The parallel to x axis which will come will become parallel to the x axis. So this is our solution. So in this way we can find out our different equations. Substitute this and the solution will appear. Now otherwise, if we see, if we solve this equation then it will take us a lot of time and it may be possible that we can solve it only through power series solution. We can solve both of these using the power series solution. So we have methods for that. But in that also we will get approximate sessions because power series are found in it. But if we write the code numerically once, then with its help we can always solve it. So this is our simple thing which we said. So this is the order of H. We can do the same thing from Ghost point or Fictitious point. Which we did the second part of. So we have to do the same work in that also. Everything is just the same. We will keep inputting all these p q r a b. We have approximated this value. a b x0 x_n Now see what I did? The value of x0 is x0, and x which is -1, I named it just x_n just. Ok? For us, we will have to give it some name in the computer. Apart from this we have done all the work the same. There is no problem with this here. Now the only problem was coming with a0. So we had defined a0, you might have noticed that this will come out h alpha0, we had multiplied it 2 x0 x0, by the way, I can also put this a here, is a okay? And x0 is what a is. We already know that. So here, I wrote a. Why? Because these are the quantities that will come to us. a b yes this is so we have y0 that we can cancel it out whatever terms it is. Write it in the form of y0 and y1. how much is y1 coming? 2k+ 2k= 4k h beta0 We did that with beta0 and two, see 2 k beta0 is in place of k because we had cancelled two on both sides, so this came and this quantity came and it came on the right hand side h 2k – h*p x0 I will again write x0 as a because a is the values we already have gamma zero beta0 ,a and b are already given to us, here we are inputting a and b from here and then we are defining it in it, so it will be none for us. What do we do now ? Have to go here. So the quantity will be -2epsilon beta1. Look, it has been cancelled. So it will become -beta1* h*p_n and -beta1*p_n  2 times. So its values will come here. So this is  - 2 epsilon *beta1 then the next term will come. We can write it as b. So we have p(b) q(b) because the

right hand side is the b value, we know it. So we will calculate it on that. And what will happen to the last right side? Gamma1 h *( 2epsilon + h*p(b)) this is gamma 2k + h*p_n okay h is already 2k + h*p_n this was it and - h²beta1 *r(b) so like this we define it as h² beta1 r(b) and this is outside so this is the quantity that we have. So that's it, we solved the conditions and then we defined the system and we got x and y in return. Ok?

```
        # Ghost point formulation for boundary conditions
        A[0, 0] = h*alpha0*(2*eps-h*p(a)) + beta0*(q(a)*h**2-2*eps)
        A[0, 1] = 2*eps*beta0
        B[0] = h*(2*eps-h*p(a))*gamma0 + h**2*r(a)*beta0

        A[N , N-1] = -2*eps*beta1
        A[N , N] = h*alpha1*(2*eps+h*p(b)) - beta1*(q(b)*h**2-2*eps)
        B[N] = gamma1*h*(2*eps+h*p(b)) - h**2*beta1*r(b)

        # Solve the system
        Y = np.linalg.solve(A, B)

    return x, Y

# Example: y'' + 2 y' - 3y = 9x, 0 <= x <= 1, with y(0) = 1 and y'(1) = 2
def p(x):
    return 2

def q(x):
    return -3

def r(x):
    return 9*x

eps = 1
a, b = 0, 1
alpha0, beta0, gamma0 = 1, 0, 1  # y(0) = 1
alpha1, beta1, gamma1 = 0, 1, 2  # y'(1) = 2
N = 40
```

(1:18:06)

So we can do different types of equations. For example, let us take a simple equation, y' + 2y' - 3y = 9x and thus we have defined the boundary conditions. So here we have taken Epsilon as one and we have taken a and b between 0 to one and this we are taking as Dirichlet type and this we are taking as Neuman type. So we took mixed and instead of taking n for 40, I will just make it 20 and from here we made the call. Ok? And we saw the session of this equation, so its session is also given to us. If the exact solution is given then it is great, we can find out the error. If it is not given then we cannot do it. Ok? So from here if we had the exact session given. We found out the exact solution at x.

```
x, Y = solve_bvp_ghost(eps, p, q, r, a, b, alpha0, beta0, gamma0, alpha1, beta1, gamma1, N)

# Exact solution for error analysis (if known, here assumed as an approximation for demonstration)
def exact_solution(x):
    return ((3*np.exp(1)-5)*np.exp(-3*x)+(5+9*np.exp(-3))*np.exp(x))/(np.exp(1)+3*np.exp(-3))-3*x-2

exact_vals = np.array([exact_solution(xi) for xi in x])
errors = np.abs(Y - exact_vals)

# Plot the results
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(x, Y, label='Finite Difference Solution', marker='o')
plt.plot(x, exact_vals, '--', label='Exact Solution')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.title('BVP Solution using Finite Differences with Ghost Points')
plt.grid()

# Error plot
plt.subplot(1, 2, 2)
plt.plot(x, errors, label='Error', marker='s', color='r')
plt.xlabel('x')
plt.ylabel('Absolute Error')
plt.legend()
plt.title('Error Plot')
```

(1:19:37)

See, the array has been defined, the exact solution at xi, which we had defined for xi in x and after that we found out the errors and called it, so see, this solution has come to us, our exact solution and the final difference solution both are matching and our error is this error has come. Ok?



(1:20:01)

So in this way we can calculate its values. We have written it like this, sometimes we have to check it, so we found the quantity. So with the help of this code, even if we do not have the exact session, we can still find out its values. For example, if I suppose I take some other equation. Let me take the second equation. I suppose I took this. P and I took this -3 okay? I took this and I took this np dot exponential x I took this. 9 *exponential x I got this. Ok? And everything else is assumed to be the same.

(1:21:15)

Let's see what happens. If this came, then the exact session was not known in it, then we know that it is unknown, so this is our main solution, this is the error, when we do not have the exact solution, then we have found it out, okay, why because in this case our exact solution that is unknown, so we will define it only when it is unknown, otherwise if it is unknown, then we will get only one solution, that will be our final difference solution, so we have calculated that.
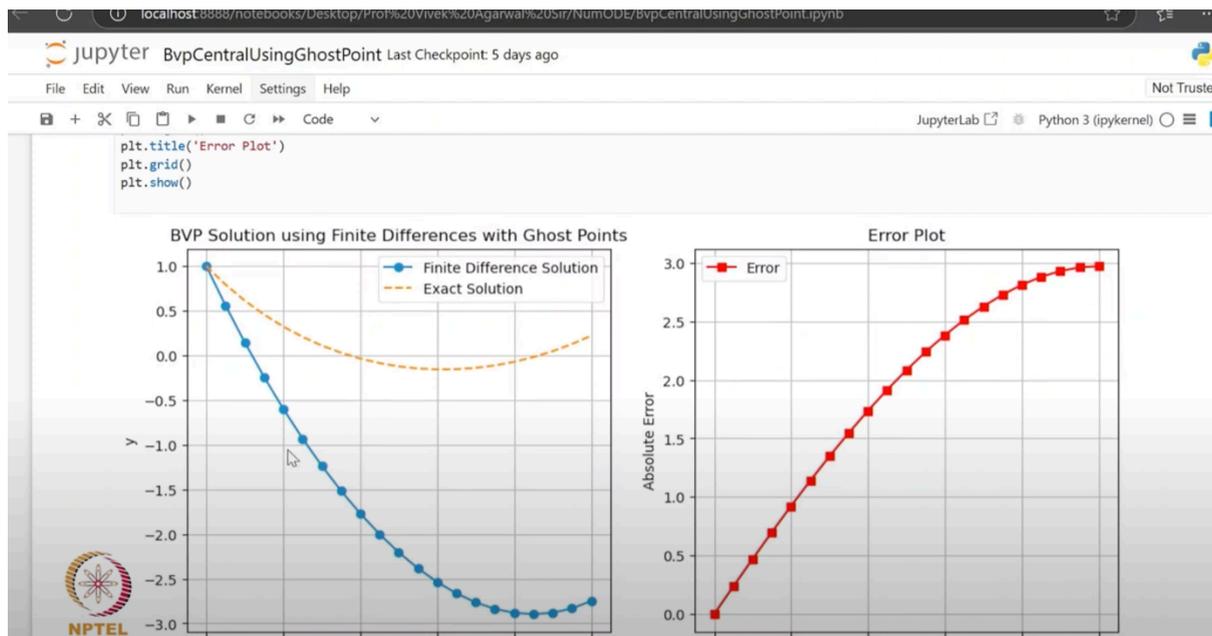


(1:21:47)

So if we have any differential equation and if its solution is known to us then it will show us the error. Otherwise you will not be able to show it. Ok? So in this way we will keep inputting different differential equations using different methods and we will keep getting their approximate solutions and we will keep solving it accordingly. Ok? So, we found this out from both of them. Either we go with the simple order of H or we go with the fictitious and ghost point. So because of this we can solve any boundary related problem which is second order OD with mixed boundary conditions. We can approximate it with the help of finite central difference operator. So I hope you have understood this lecture and thank you for watching this lecture. Hello.