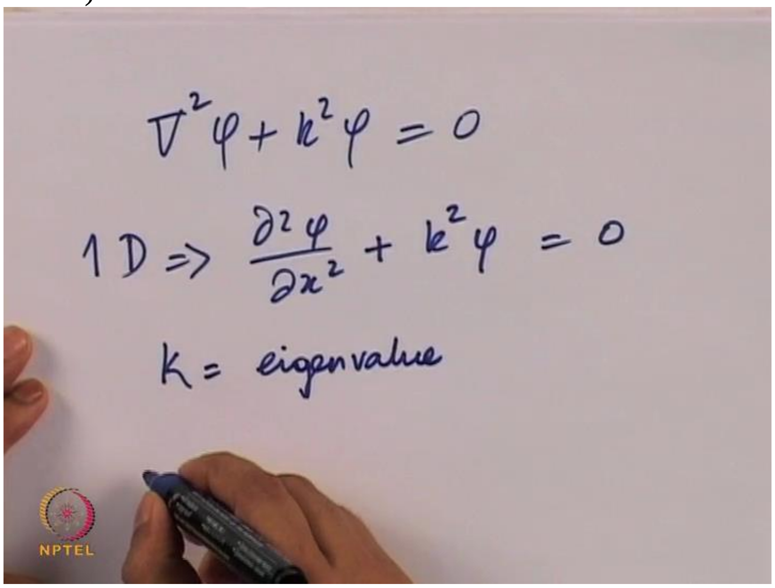
Computational Electromagnetics and Applications
Professor Krish Sankaram
Indian Institute of Technology, Bombay
Exercise No 4
Finite Difference Methods-II

So now we are going to look into one of the other simple equations that one will confront in electromagnetics, which is Helmholtz equation. So in this particular module of discussing exercises using Matlab simulation. So we are going to start with Helmholtz equation. So we will simplify Helmholtz equation to one dimensional case, and we will look into certain routines which we will use to compute Eigen values and Eigen vectors for that problem. So let us look into the basic equation for Helmholtz in one dimension.

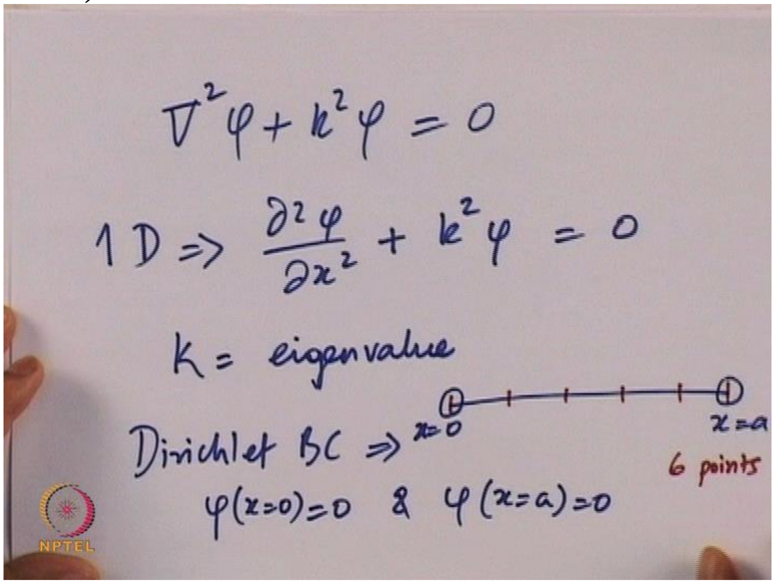
(Refer Slide Time: 00:54)



So what we have is Del square Phi plus k square Phi is equal to 0. And in a one dimensional case we can write it in a simple form for a 1D case this will imply Doe square Phi by Doe x square plus k square phi is equal to 0. And here k is the Eigen value and we can compute analytically the solution for this particular problem, but our focus will be on the numerical solution.

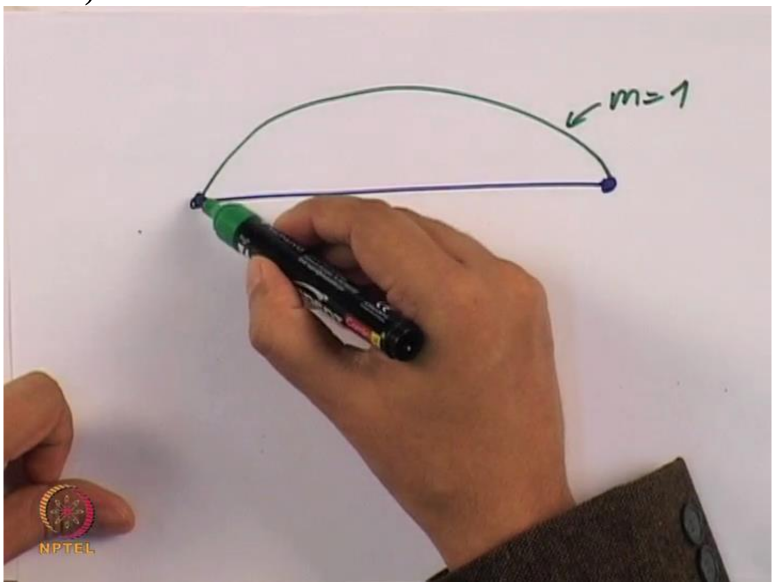
So let us force certain boundary condition so we are going to take Dirichlet Boundary condition which means we are going to set the problems boundary.

(Refer Slide Time: 01:58)



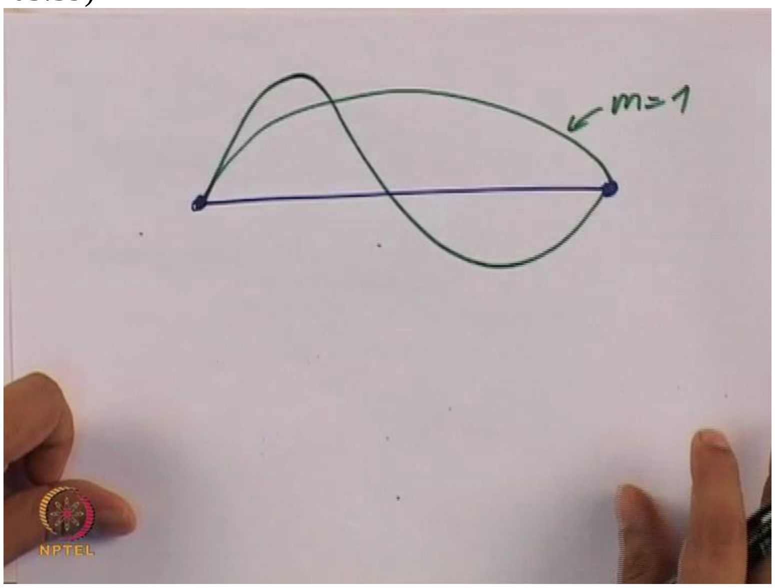
So let us define the problem, let us say we have a domain which is running from 0 x equal to 0 to x equal to a, and this domain is being discretized in certain manner. So let us say we have 1, 2, 3,4,5,6 points, so we have 6 points, and in these 6 points the initial point and the terminal point are going to be the boundaries of this domain. So we are going to set Phi at x equal to 0 is equal to 0 and Phi at x equal to a is equal to 0. So the solution goes to 0 in this particular manner at the boundaries. So regardless of what the solution will look like in the middle area at the boundaries they are going to come to 0.

(Refer Slide Time: 03:14)



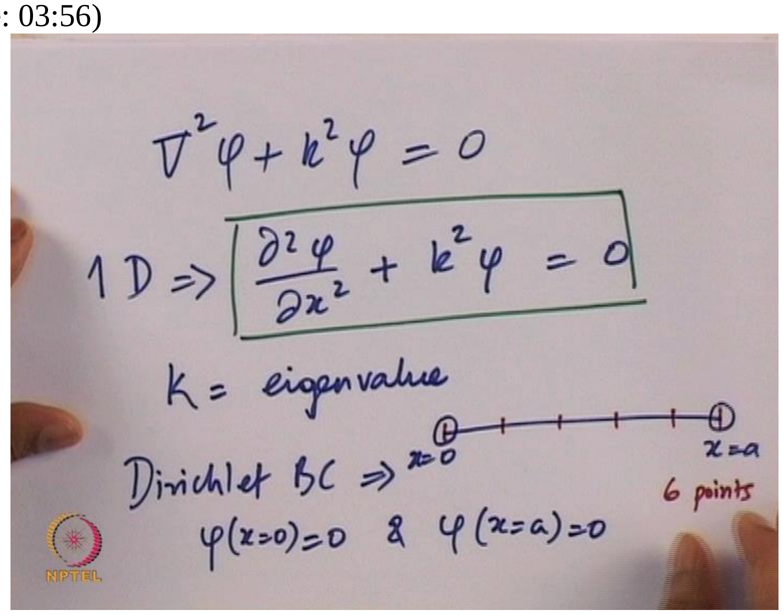
So if you see for this particular problem, let us say we can write it like this there are many solutions possible. One of them will be a very simple straight forward solution which will start from 0 will have a maximum at the middle and goes to 0. So let us say this is going to be our m equal to 1.

(Refer Slide Time: 03:39)



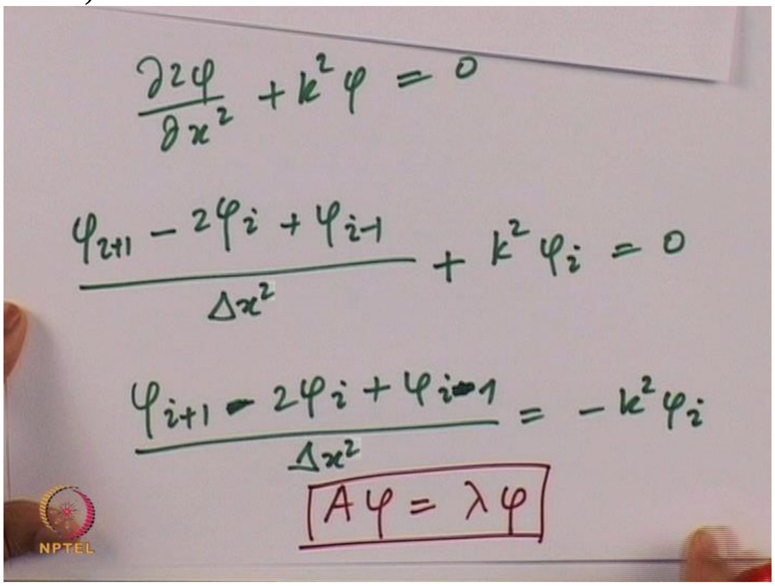
And there are other solution possible. So for example it can go to 0 it can go to high and it can go to 1 so on and so forth. So different possibilities exists.

(Refer Slide Time: 03:56)



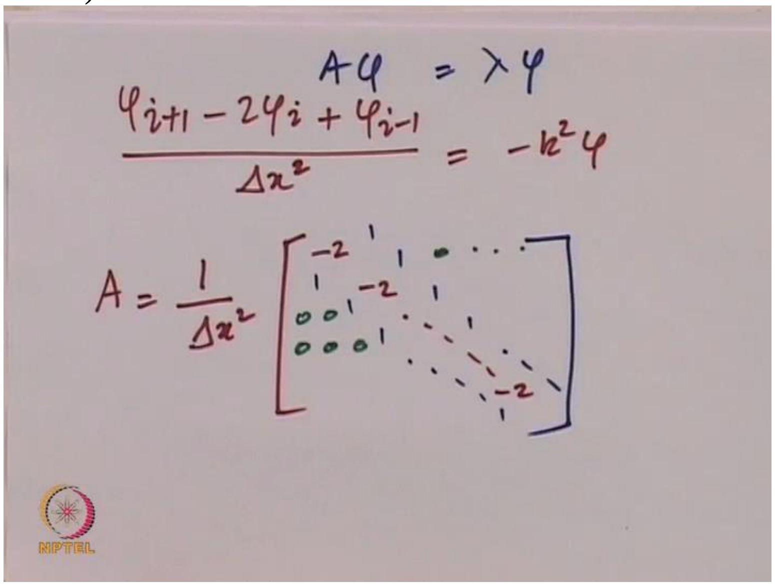
And right now we are going to simulate it for these various possibilities. And what we are going to see is our solution is going to be based on this particular equation and we have to discretize this particular equation using central differencing method for the first term. So we are going to write it in the manner as follows.

(Refer Slide Time: 04:15)



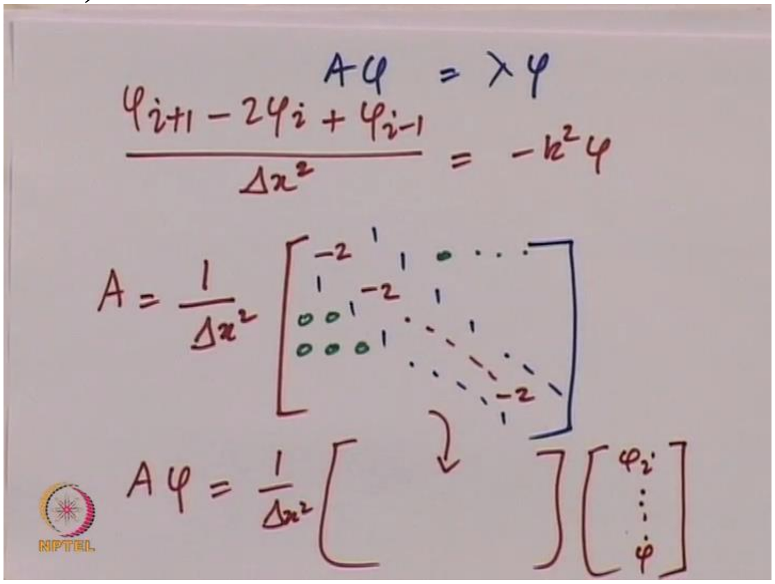
So doe square Phi by doe x square plus k square Phi is equal to 0 what we are going to do is we are going to say Phi of i plus 1 minus 2Phi of i plus Phi of i minus 1 divided by delta x plus k square phi of i is equal to 0. So what i am going to do now is I am going to take the term on the second term on the left hand side to the right hand side. And going to write the same equation in this manner Phi i plus 1 minus 2Phi i plus Phi i minus 1 divided by delta x square is equal to minus k square Phi of i. So this is a equation which is of the form A multiplied by Phi is equal to lambda multiplied by Phi. So what is this A and Lambda going to be is something that we will look now.

(Refer Slide Time: 05:46)



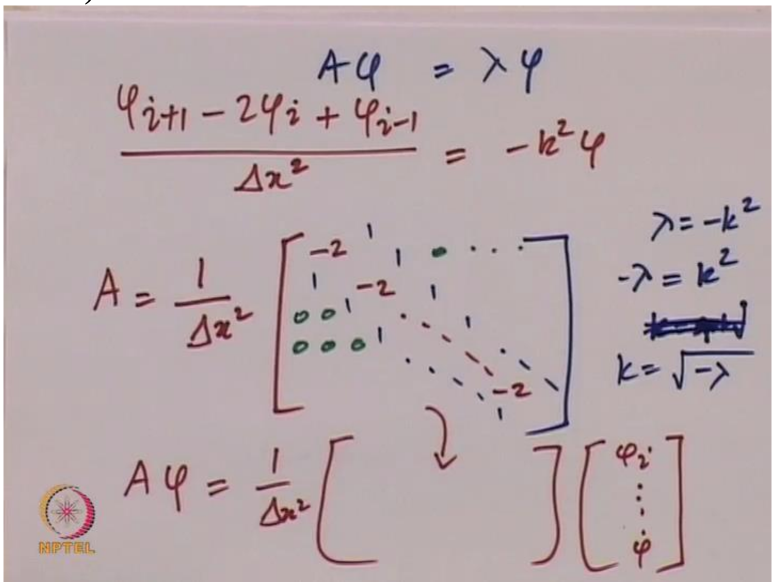
So if you look at that equation we have got I am writing it again Phi i plus 1 minus 2Phi i plus Phi i minus 1 divided by delta x is equal to minus k square Phi. I said this is going to be equal to A multiplied by Phi is equal to Lambda by Phi. So if I expand this for various i what I will get is essentially A equal to 1 by delta x square and we will have terms on the center this will be minus 2, minus 2 so on and so forth minus 2 and I will have terms as 1, 1, 1, so on and so forth and then 1,1,1,1 So on and so forth 1. So this will be a tri diagonal matrix where the other terms will be categorically 0, so other terms will be 0 we can fill them in. So it is going to be a sparse matrix with only three diagonal values and this comes directly from this particular equation that you can see, the first term is the term that is 1s which are here, the second are the leading diagonal terms and the third term are the next diagonal term. And when you write in this manner what you get is a series of equation.

(Refer Slide Time: 07:40)



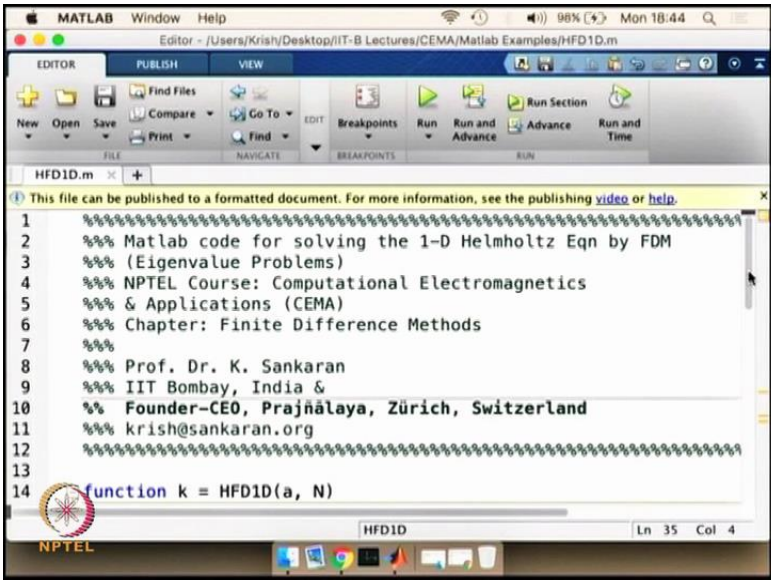
And of course when you multiply A times Phi what you see is this particular equation 1 by del x square and this particular equation like the way we have written and we have Phi i so on and so forth until Phi. So these are the Phi s. So this is the way we are going to understand this particular problem in the discrete space.

(Refer Slide Time: 08:16)



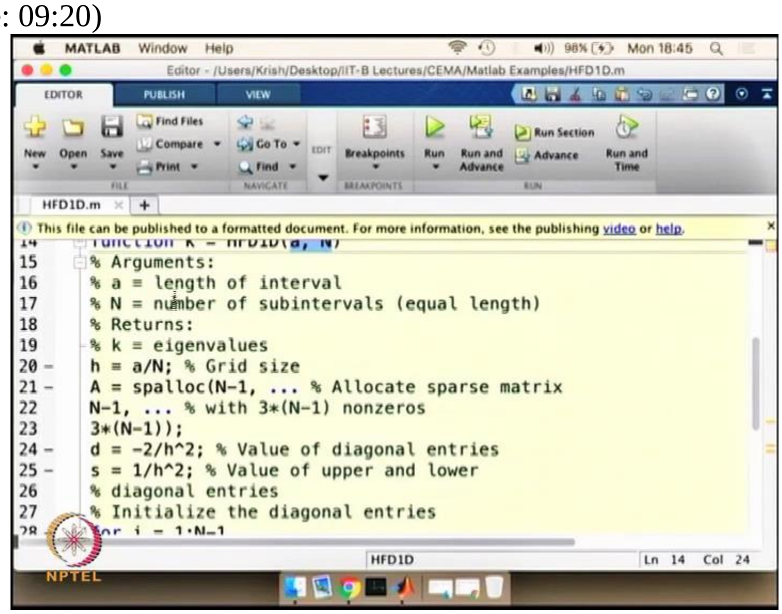
And what is also important to know is what we get as Lambda is nothing but Lambda is going to be equal to minus k square. So when you want to compute the value of k what you have to do is take minus on the other side. So when you are computing k, so you say k is equal to square root of minus Lambda. That is what you will have in the equation as well in the program.

(Refer Slide Time: 08:50)



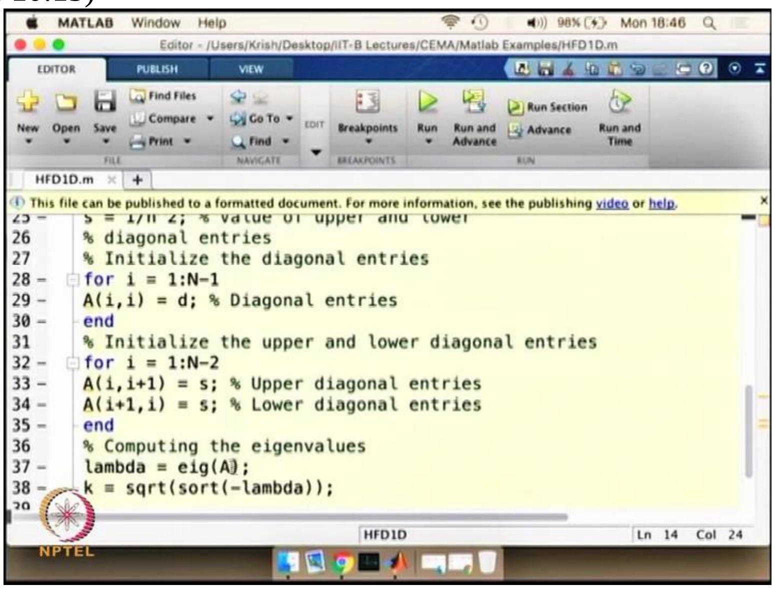
So let us go in the program itself and we will take it one by one as we are discussing. So this is a classical program we are going to use. And this program is basically from the source what we have given here. You can have a look at the source later on. In all the course what we have taken wherever possible we have given the source information already in the code, so you can get also some more background information of the code.

(Refer Slide Time: 09:20)



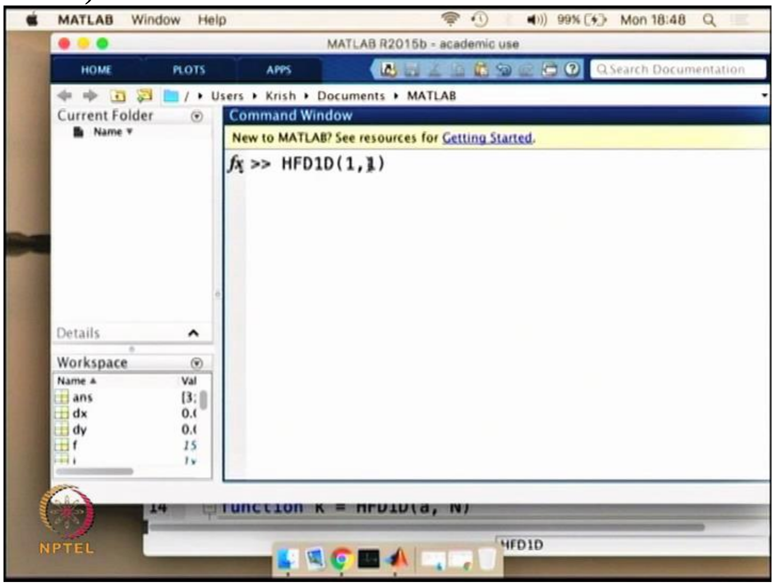
So let us look into this function that we have written, the function is going to give the value of k but we have to give some inputs to this function which is a and N. So let us look into the argument itself, a is the length of the interval so this is going to be our domain length and N is the number of sub intervals and for simplicity we are using equal length that means each of the sub intervals will be of same size. And we are going to get the value k when we give the value of a and N. And it is going to do that in a systematic manner by defining the grid size as h, it is a delta x term which we are using in our equation and the diagonal values are computed accordingly once we know the dimensions of the grid space.

(Refer Slide Time: 10:15)



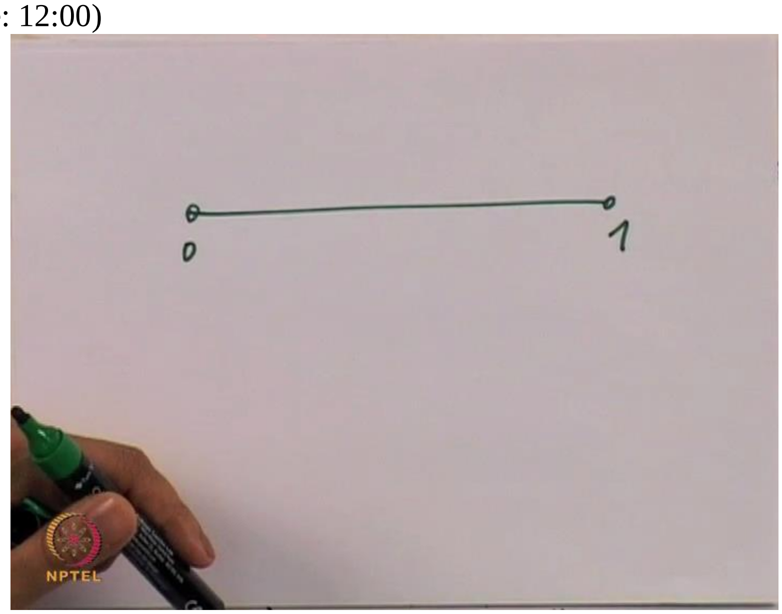
What we are using is a simple Matlab function called eig which works very well in one dimensional case and also in a two dimensional case it might be useful. But once you go into a three dimensional problem it will not be useful where we are going to use a different function which is called as eigs with gs at the end. But for this one dimensional problem we can still use eig as in built function and we can compute the value of Lambda using the Eigen value of the Matrix what we have here, As you can see once you try to compute the value of the Eigen value of this matrix you will directly get the lambda values and the Lambda value is nothing but the Eigen value of A matrix and once you have that Lambda value you can compute the value of k as i explained taking the square root of minus Lambda. We are sorting the Lambda in the order of increasing value and we are taking the square root.

(Refer Slide Time: 11:17)



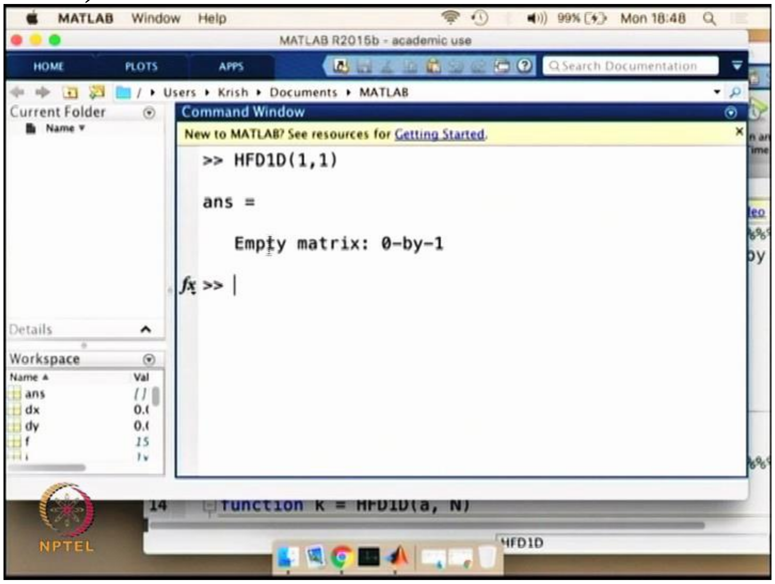
So let us save this one and when let us go into the main window of the Matlab and let us clear the screen. So now we can call the same function for certain value of a and N. So remember the first one is going to be the length of the interval which is going to be 1. And of we say that the entire domain is having only 1 sub interval that means in simple terms we have very veryvery coarse grid.

(Refer Slide Time: 12:00)



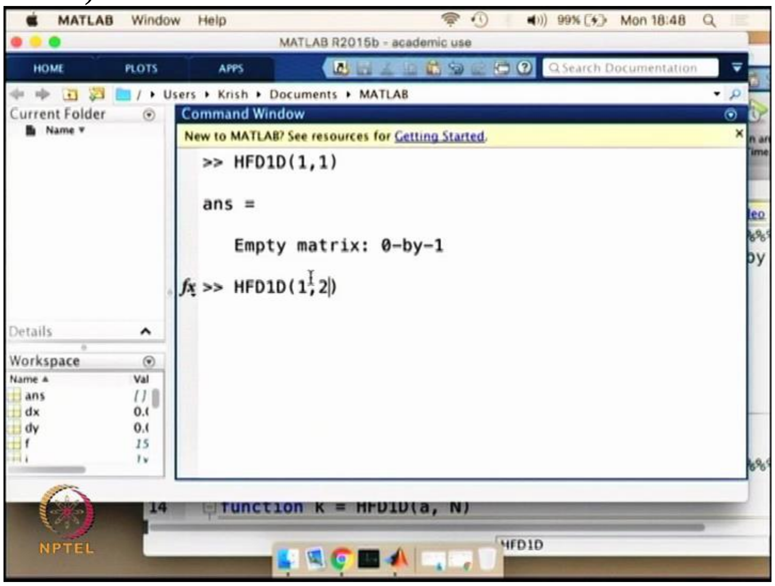
So when we have a very very coarse grid and we do not have any grid points. So if we say 0 to a is going to be 0 to 1 and if we say the entire grid is going to be one sub interval that means there is no points in between the grid to resolve, so in that sense we will not be able to simulate this problem because we do not have any sample points in between.

(Refer Slide Time: 12:22)



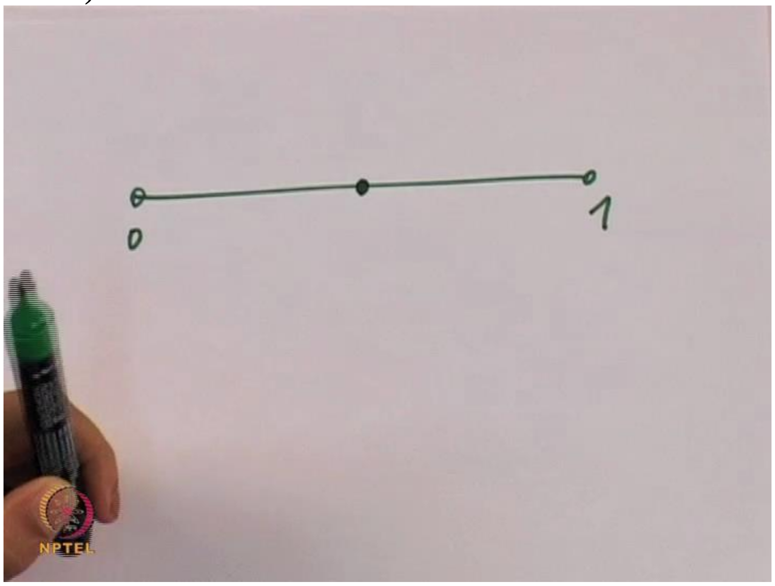
So this will give us an error like we see in this program, so if we simulate this program it will say empty matrix. It does not give the answer. So for that we need to increase the number of points.

(Refer Slide Time: 12:35)



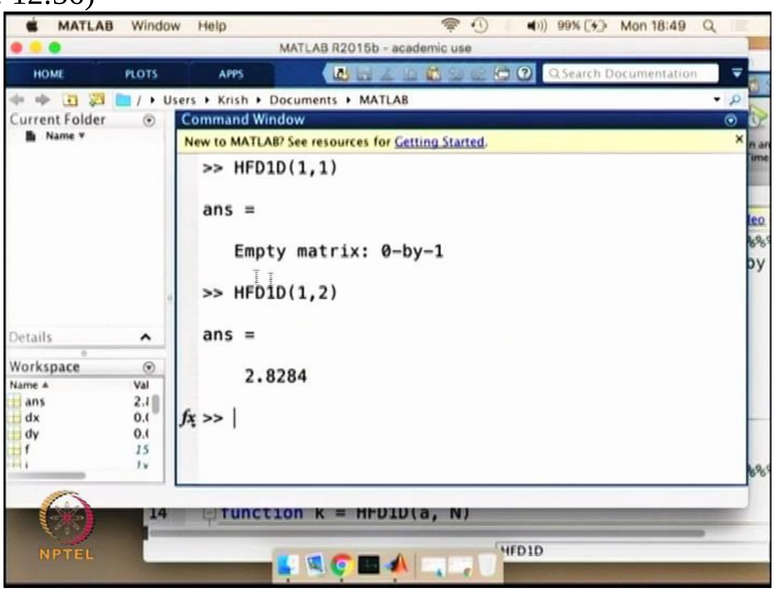
So what we are going to do is we are going to simulate for 2 points, 2 points means so there will be 2 cells in this particular one dimensional domain.

(Refer Slide Time: 12:48)



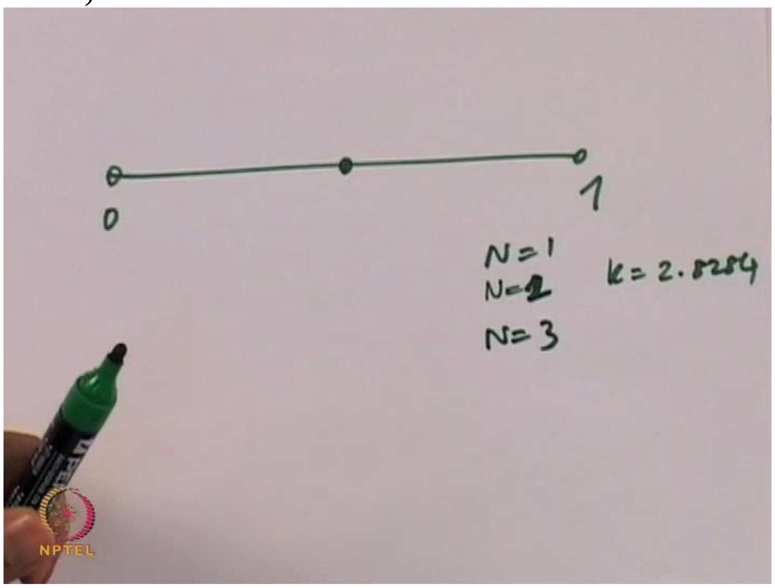
So there is going to be one sample point in between. Still very coarse but good enough to get some results.

(Refer Slide Time: 12:56)



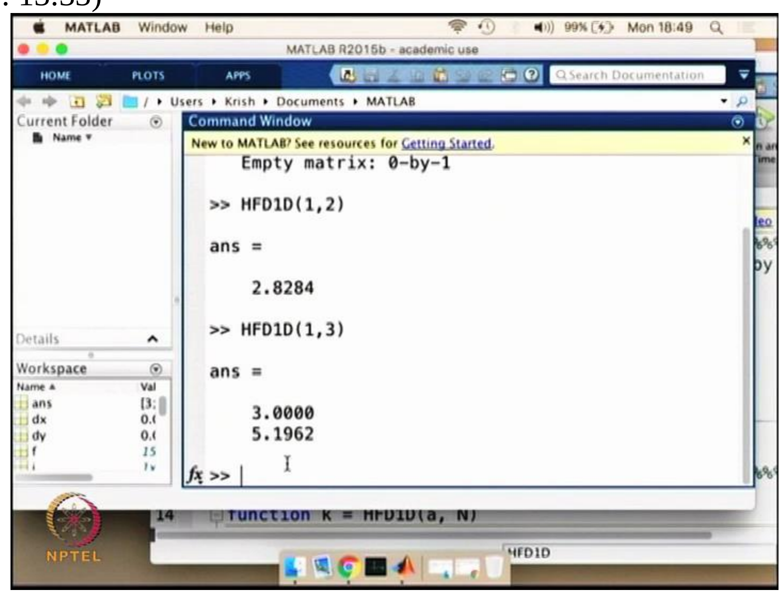
So when we run this code now it will give you the first value as 2.8284.

(Refer Slide Time: 13:06)



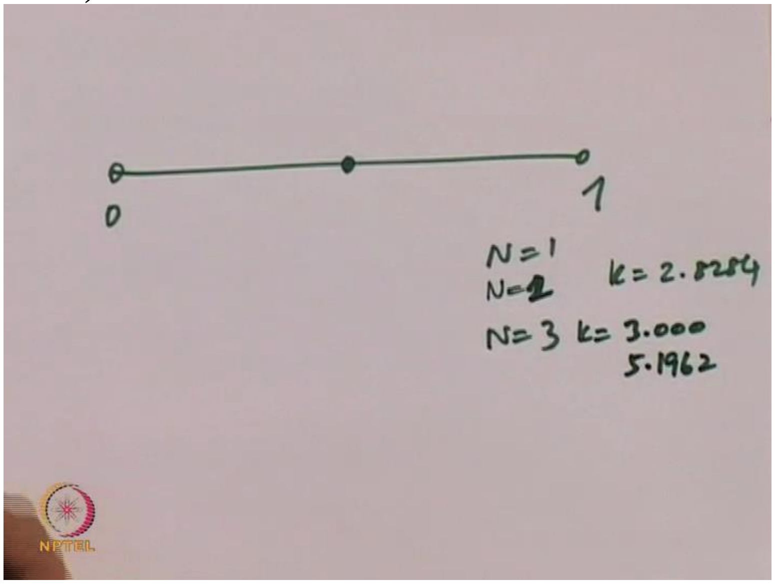
So let us write it down. So it is giving as k equal to 2.8284. So now we are going to run the code. So this is going to be for N equal to 2. So for N equal to 1 we did not get any value, so for N equal to 2 we are getting 2.8284. So let us say N equal to 3.

(Refer Slide Time: 13:33)



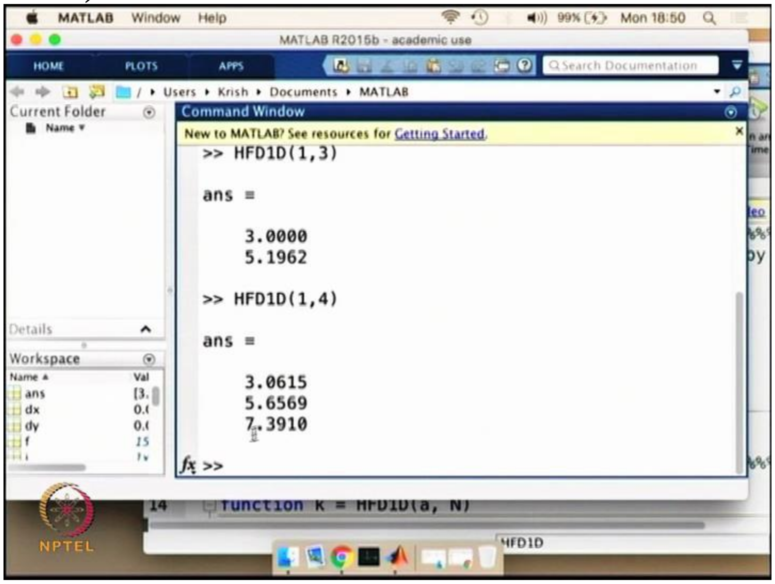
So if i say N equal to 3 in the code, I am running it for N equal to 3, what you see is you are getting 2 values.

(Refer Slide Time: 13:50)



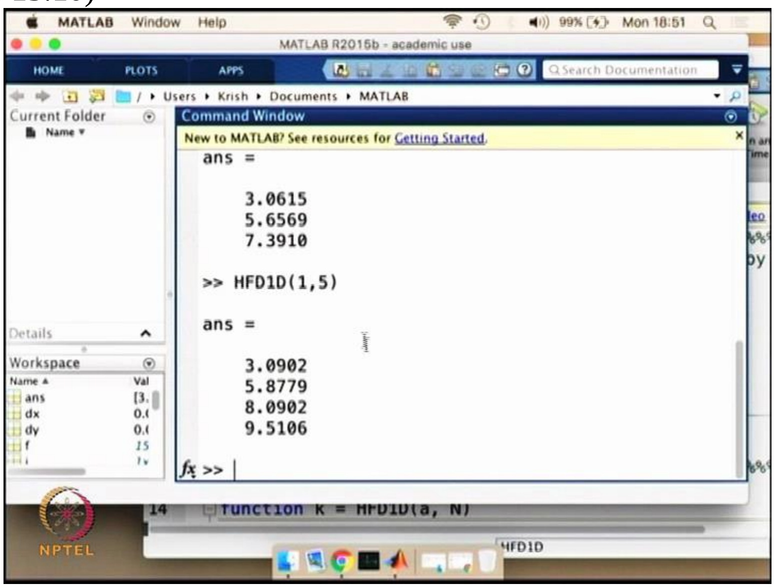
So what we are getting is k equal to 3.000 and k equal to 5.1962. As you can see the first case we got k equal to 2.8284 and in the second case we are getting k equal to 3.000 and 5.1962. We are able to get more solutions but also we are able to see that the first solution itself k equal to the first Eigen value is also changing it is going from 2.8284 to 3.0000. So this is going to improve in accuracy when you go higher up in the number of points.

(Refer Slide Time: 14:20)



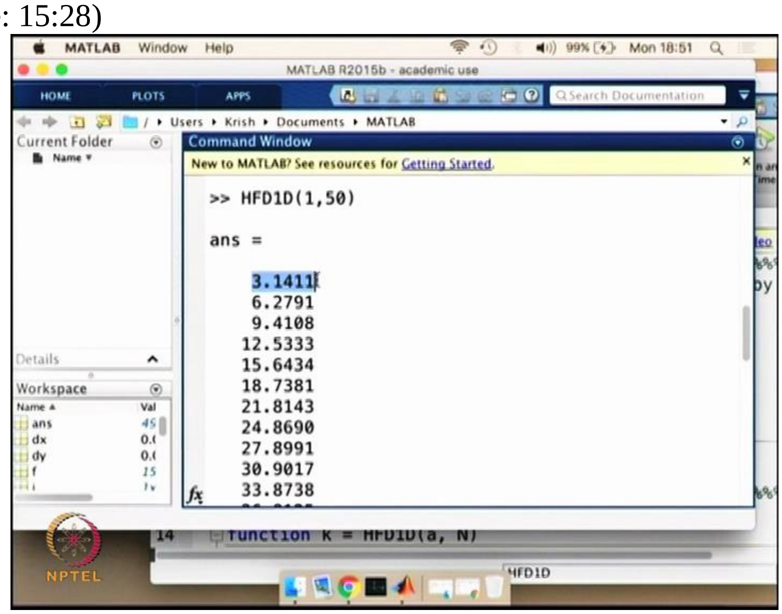
So if we say 4 you will see that the value is improving it is initially it was 2.8284 later on it went to 3.000 and then it is going to 3.06. As you can see the first case where we went from 1 to 2 points, the jump was quite drastic. We are almost point 2 differences. Whereas in the second case where we go from 3 to 4 points the jump is not that high it is only 0.06. And we will see that when we go from 4 to 5 this jump will be even lower. Not only that the second Eigen value will see that it is also not that big of a difference it is 0.5.

(Refer Slide Time: 15:10)



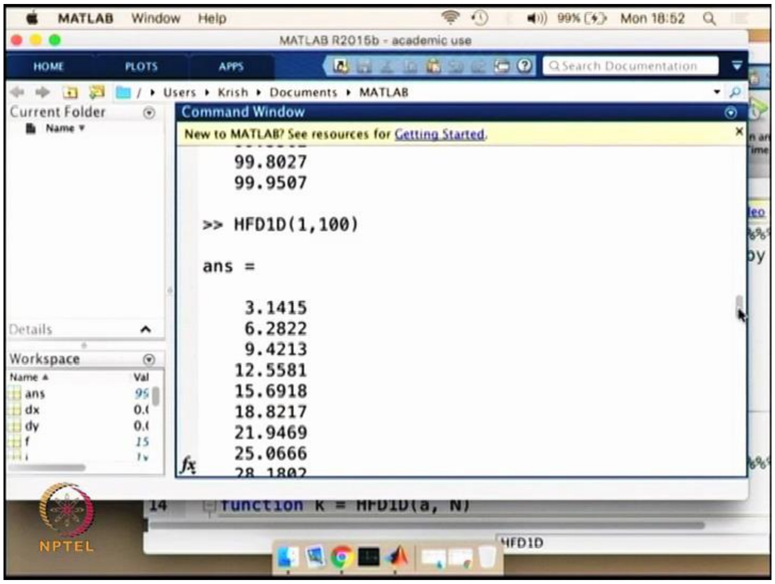
So now when we run the same code for higher number of points we go from 4 to 5. We see that the results are converging in the first case also in the second case.

(Refer Slide Time: 15:28)



So for fun let us say we are putting number of points as 50. When we do that what we see is different values coming from various cases, and the first value is almost 3.1411. So if we go from 50 to 100, this is not going to change much and that is what i am going to show you.

(Refer Slide Time: 15:43)



We go from 50 to 100 you see that the first value is still 3.1415 whereas in the previous case also it was 3.14 there is a small difference which 0.0004, which is demonstrating the solution is converging when we have more number if points between the interval.

So this is a classical example for us to showcase how we can use a simple central differencing method to study one dimensional Helmholtz equation. Of course when we go to second and third dimensions the Matlab code will be much more intense in terms of computational time and also in terms of the accuracy it will be more demanding. But the one dimensional case and two dimensional case is good enough in most cases to understand the physics of numerical method. How the physical solution should look like, in what matter in which we can adjust the accuracy of the solution like we have seen in this example.

So this particular example is also for you to try it out at a later stage. We will give you the code. Please look into the code and this is the function so you have to give the value. But in order to understand this matlab function you can go inside the Matlab function test a little bit varying certain parameters, so that you get a better understanding of how this function is working and also I would recommend you to manipulate the interval itself, for now we have chosen 0 to 1, may be you go from 0 to 10, 0 to 5, so on and so forth. So there will be more discretization possible. Again it will lead to more Eigen values and Eigen vectors but at least it will give you a rough idea how the code is working.

So with that being said i will stop here and we will come back in the later modules to look into other sides of the problems particularly using the accuracy aspect. This will be the focus later on. But for now it is a good point to stop and I request you try this problem yourself