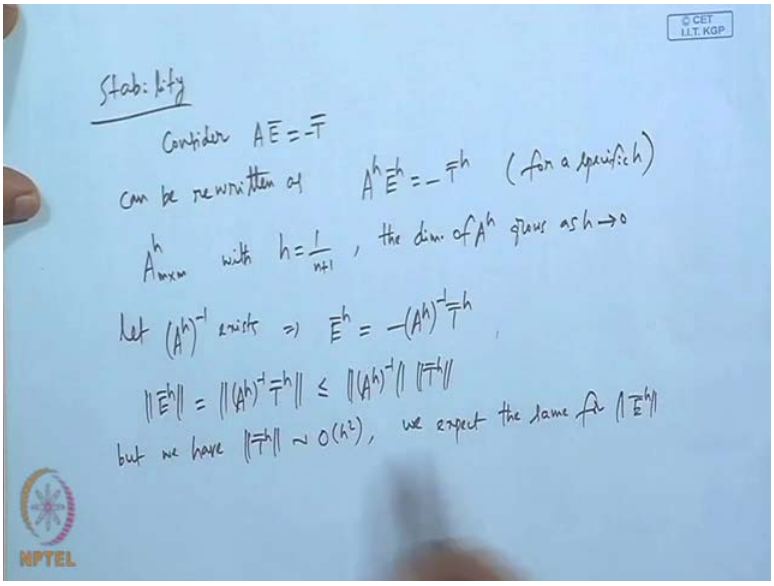
# Numerical Solutions of Ordinary and Partial Differential Equations

**Prof. G. P. Raja Sekhar**
**Department of Mathematics**
**Indian Institute of Technology, Kharagpur**

## Lecture - 17
## Linear/Non -Linear Second Order BVPs

Hi, so we have discussed in the last class, about the finite difference schemes and the corresponding error estimates. So, let us proceed further little bit on that, with respect to the stability aspects. So, and then we generalise the second order method.

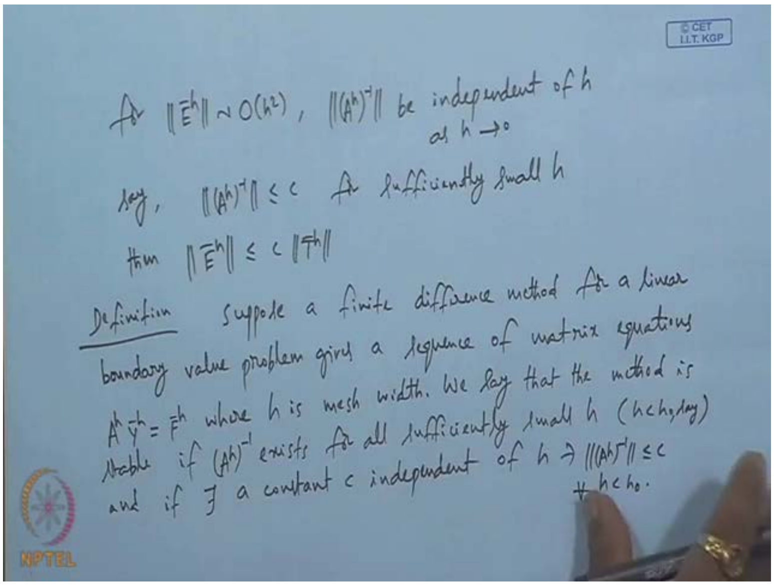
(Refer Slide Time: 00:40)




So, stability, consider so this equation so which we have derived so where A is the corresponding tri diagonal matrix and E is the error and error, global error matrix and then, this is the local truncation error matrix. So, this can be, this can be rewritten as A h, E h so this is just to know that for a specific h.

So, definitely the matrix A h so this is with h equals to 1 by and plus 1 for 0 1 case and the dimension of Ah, grows as this happens. Let Ah inverse exists, this implies Eh is then, the nor is equals to, but we have this, as order of h square. Therefore, we expect, we expect the same for. But, from this if you see for this, to have this order, this must be bounded by a constant.

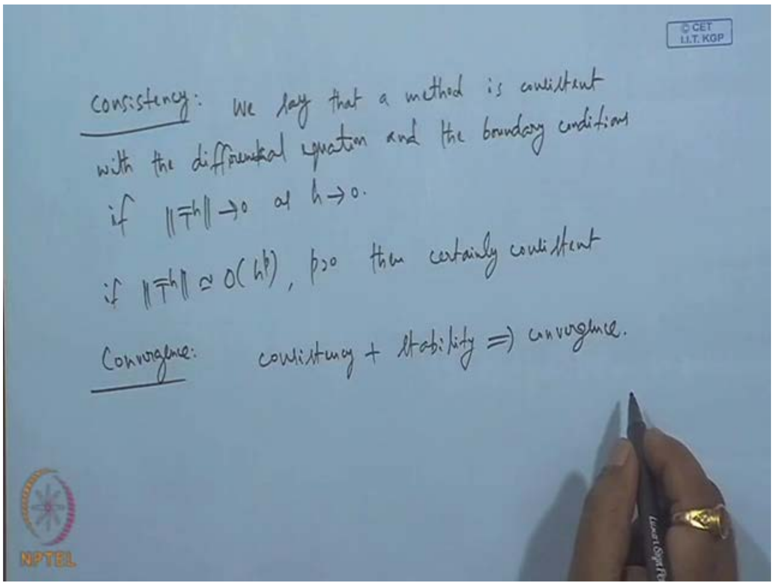
(Refer Slide Time: 03:41)




So, for this to behave in this order, this be independent of h as h goes to 0 say, less than or equals to c, for sufficiently small h then, we get. So, this suggests a definition that means so global error is bounded so we know local error is order of h square and global error is also bounded like that. Hence, the stability is ensured. Suppose a finite difference method, for a linear boundary value problem gives a sequence of matrix equations where, h is mesh width. We say that, the method is stable if this exists for all sufficiently small h so that is say h less than h naught say. And if there exists constant c, independent of h such that is less than equal to c, for every h. This is, this will suggest the stability condition.

So, once stability is done we need to, consistency so we have all talked before in the context of multi-step methods. So, we say that a method is consistent with the difference, method is consistent with the differential equation and boundary conditions, if the local truncation error goes to 0. As h goes to 0 and in general, this is order h power p, p greater than 0 then, then certainly consistent then what next, convergence.

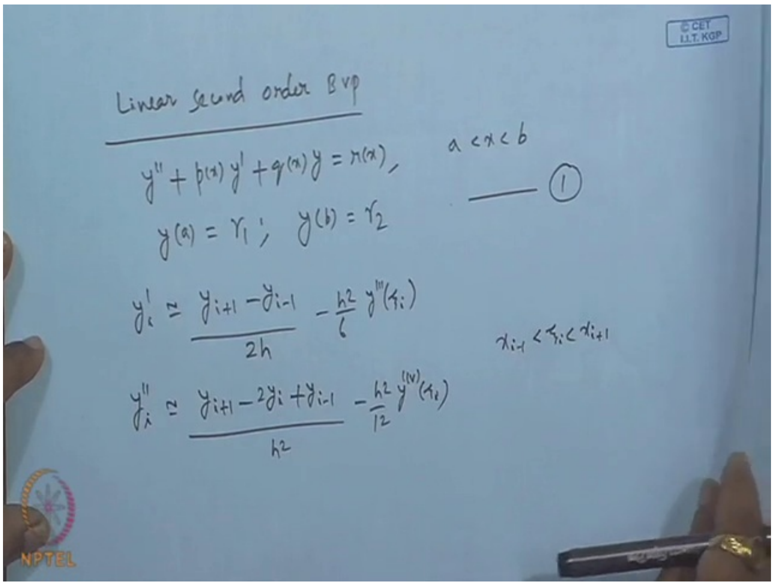
(Refer Slide Time: 07:40)




So, since we talked on, with respect to multi step method I do not want to repeat just consistency plus stability implies convergence. So, with this we have some idea of really how a particular approximation is giving you sensible results. Because, you have taken a differential equation and then pick up the derivatives, approximate and then we try to solve system and get the corresponding solutions. But what kind of errors are introduced locally and then whether, really these errors are bounded so that globally the error is bounded.

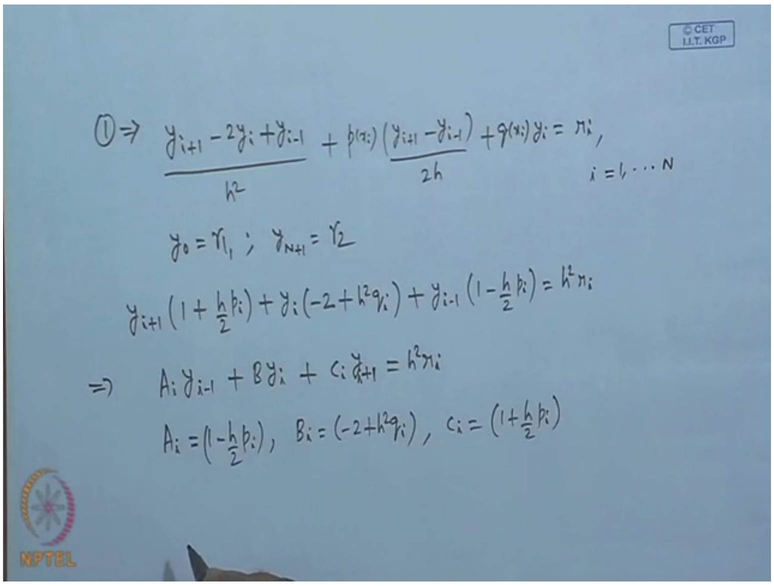
So, they are not magnified so that, the method is stable. So, these on the conditions, on the matrix are really necessary. Now, let us we discussed earlier for a simple second order, that is y double prime equals to f of x. So, now let us generalise little bit more, where we have a general set up of linear second order boundary value problem.

(Refer Slide Time: 10:52)




Second order BVP so we consider so this is our BVP, then the approximations. So, the first derivative is done with the central derivative approximations.

(Refer Slide Time: 12:29)

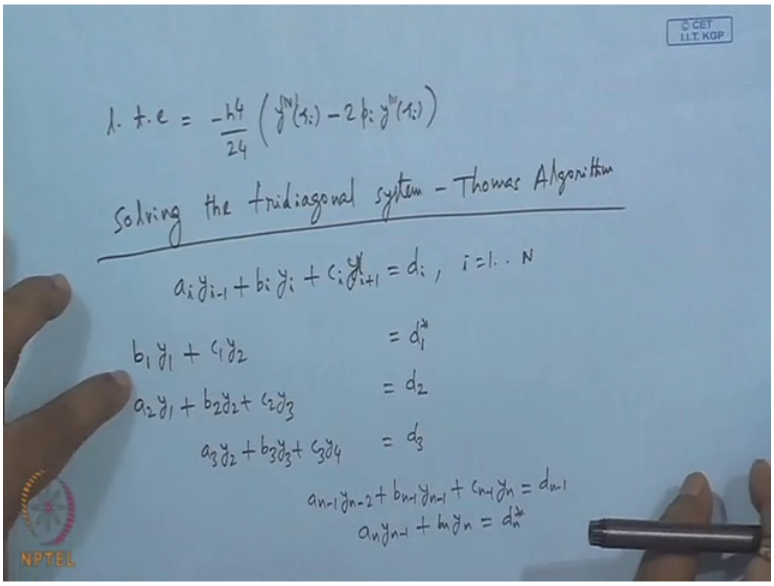



So, in view of this 1 becomes so y0 is gamma 1 and y n plus 1 is gamma 2. Now, we collect the coefficients of similar terms, so y i plus 1, 1 plus, 1 plus so if you multiply by h square there so h, h gets cancelled, so 1 h by 2 and p so we get plus minus 2 and y i, h square q 1 minus h by 2. So, this implies where so I started with y i minus 1 so this will come here.

Now with this set up we expect a tri diagonal system where, A is B1, C1. Because, when we run the system, when i is 1 y0 so A1 does not exist because, y0 is known to us. So, A1 gets transferred to the right hand side therefore, we have this. And similarly, when we run for i n so we have, when we run for i equals to n, we have n plus 1. So, this term gets transferred to the right hand side so we have, this is Bi so we have A and B.

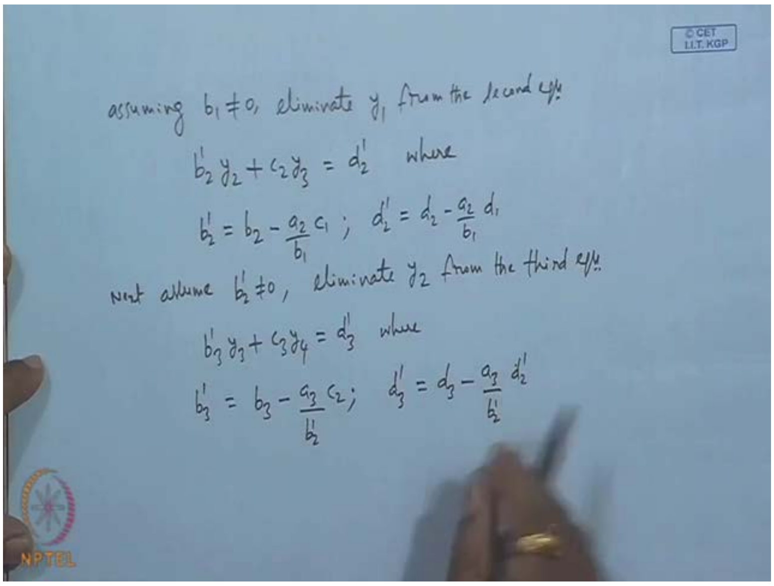
So, and b bar look at that, when we run this for i equals to 1, h square r1 and y0 is gamma 1 so A1 gamma 1 gets transferred to the right hand side. So, we have h square r1 minus then we have h square r2, h square r3 then h square r n minus so i equals to n. So, this will be y n plus 1 so we have c n times gamma t.

(Refer Slide Time: 18:49)



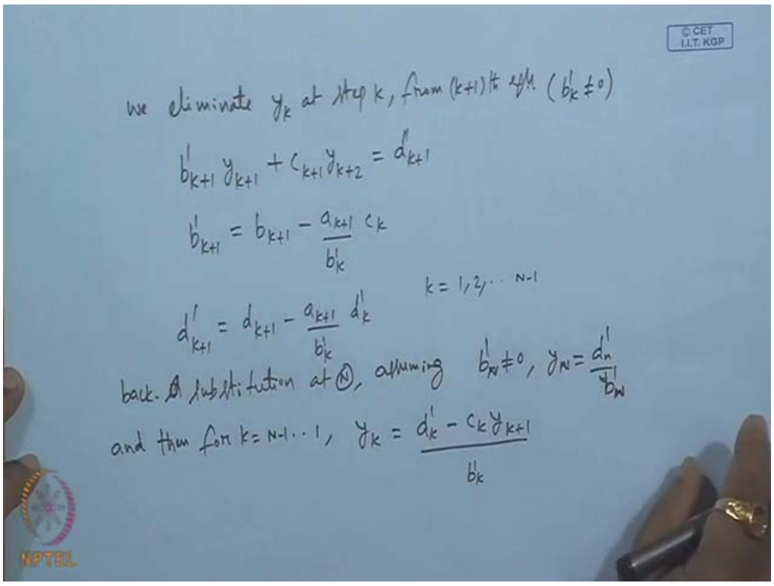

So, this will be the tridiagonal system and the corresponding local truncation error. So, this will be the corresponding local truncation error so we have system like this. So, which is a tridiagonal system and the same can be solved solving the... So, generally it is an elimination process, but for this particular case since it is a tridiagonal. So, there is an algorithm called Thomas algorithm. So, let us look at it so let the given system be in this form. So, which means b1 y1 plus c1 y2 equals d1 star then a2 similarly, a3 y2 then n minus 1. So, this will like our tridiagonal system expansion.

(Refer Slide Time: 22:23)




Then, assuming b1 is non-zero eliminate naturally y1, assuming b1 non-zero we eliminate y1, from the second equation. So, we get b2 prime that is modified b2 is no longer b2, where b2 prime is b2 minus this. Next, assume b2 prime non zero eliminate y2 from the third equation of course, using this equation. Then we have v3 prime y3, where.

(Refer Slide Time: 24:44)

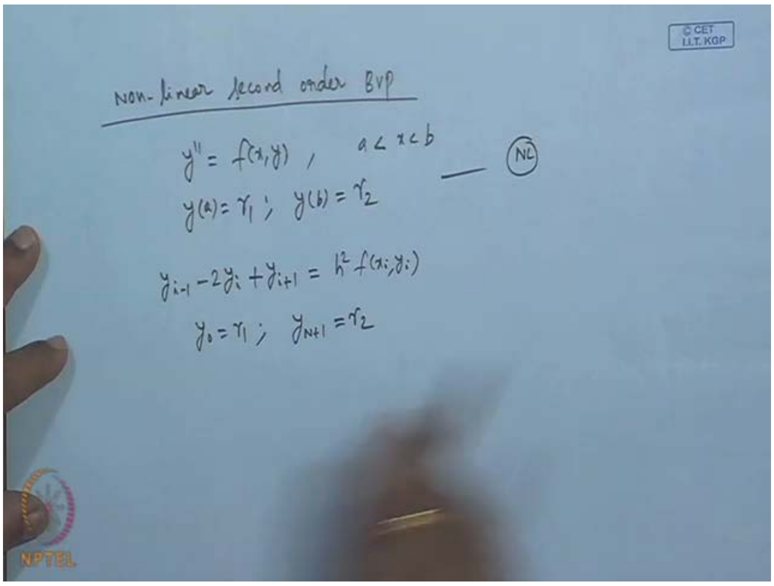



So, we continue like this, we eliminate y k at step k from k plus 1th equation of course, assuming this is non-zero, where k is. Now, what we have to do so having obtained up to here, we back substitute, back substitution at n, assuming b n prime non-zero, y n is b n prime.

And then, 1, y k is so these are capital n we are using so this is Thomas algorithm. So, one can write a nice programme and then try to solve it so having done a linear system so the next task is to try out with a non-linear system. So, what is the big deal in it. Well, for the linear case we have obtained a system of equations and then it happened to be tridiagonal and then, we have a simple algorithm called Thomas algorithm.

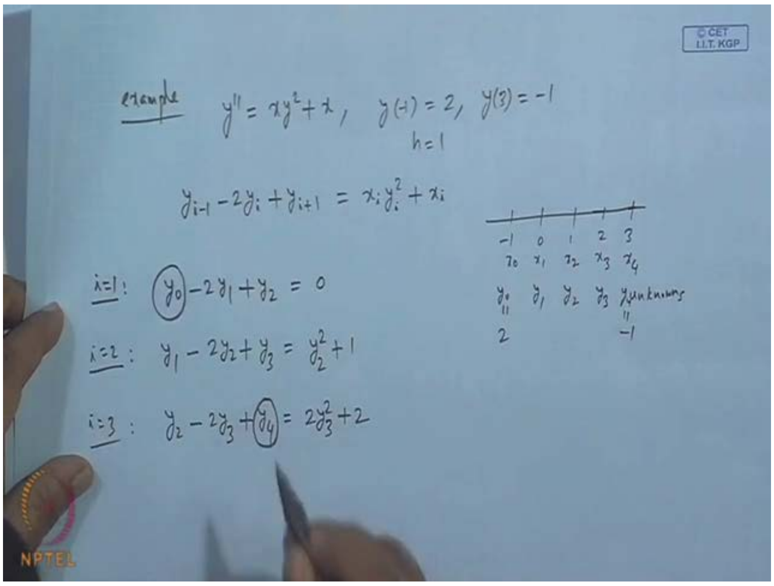
And then one can obtain the solution, even otherwise using elimination etcetera. But in case of non-linear, one straight forward thing one could expect is, probably we expect that the system is non-linear so this is one thing. So maybe it is very trivial guess, but then the next task is how do we solve corresponding non-linear system of equations. So, let us look into non-linear system.

(Refer Slide Time: 28:30)



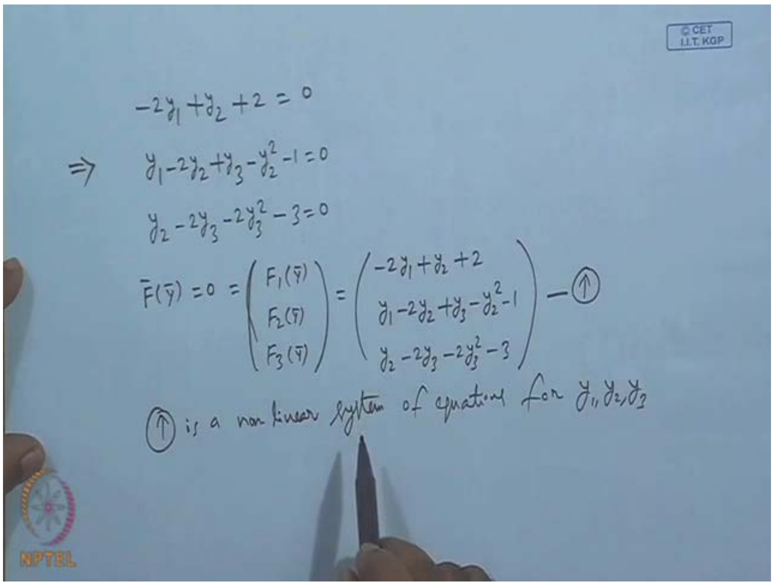

So, non-linear second order BVP so we consider say so if we discretise, if we discretise. Now, as far as derivative is concerned it is linear, but we have the right hand side part is non-linear. So, let us see how this works out with reference to an example.

(Refer Slide Time: 30:09)




Suppose this is the example, I have taken h for simplicity, now the discretised version because, h is 1. So, this our grid, so essentially so these are the unknowns because, y0 is 2, y4 is minus 1. So, i equals to 1, x 0, i equals to 1 so this will be x 1, x 1 is 0 so unfortunately this is 0. So, this will be x 2 is 1 so this is 1 and y 2 square, so because x 3 is 2 so 2 y 3 square plus 2.

(Refer Slide Time: 33:14)

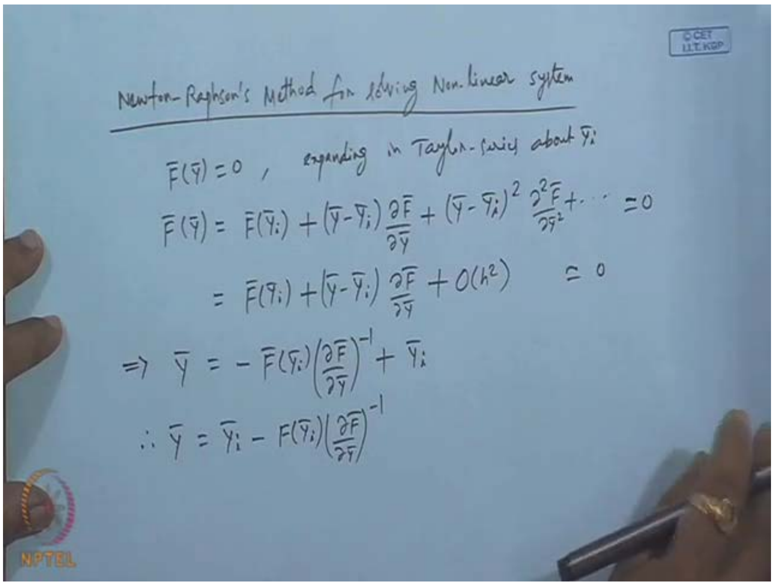



Now, in this system these are known so the final system we get it as follows. If you observe I did not put this as a matrix system x equals to b because, this is a non-linear.

So, let us call this as f of y bar equals to 0, where we have as components of so that means this is, this is also vector. So, this is nothing but so these are our f1, f2, f3 and what is it, is a non-linear system of equations for y1, y2 and y3. So, this is a non-linear system for y1, y2, y3.

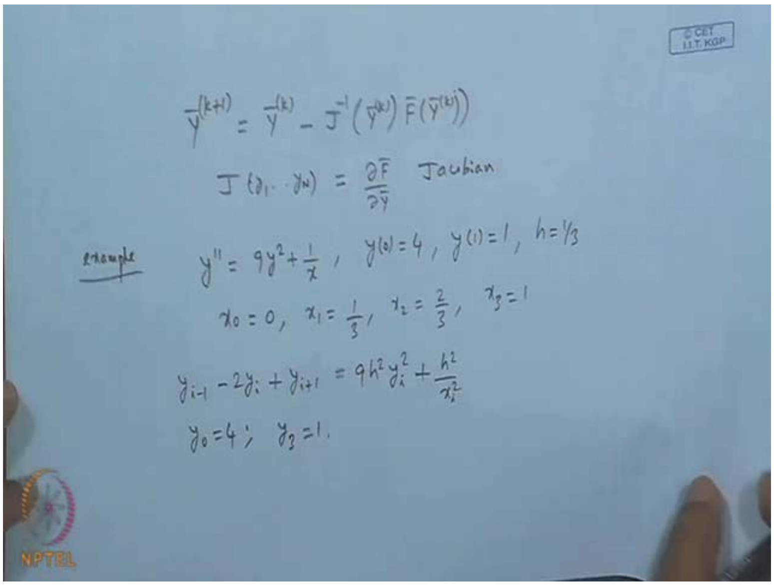
So, how do we solve ((Refer time: 36:14)). So, I think you have heard for solving non-linear algebraic equations, we have lot of methods and one popular method is Newton Raphson method. Now for non-linear system, we should try Newton Raphson method. So, what is the motive?

(Refer Slide Time: 36:41)



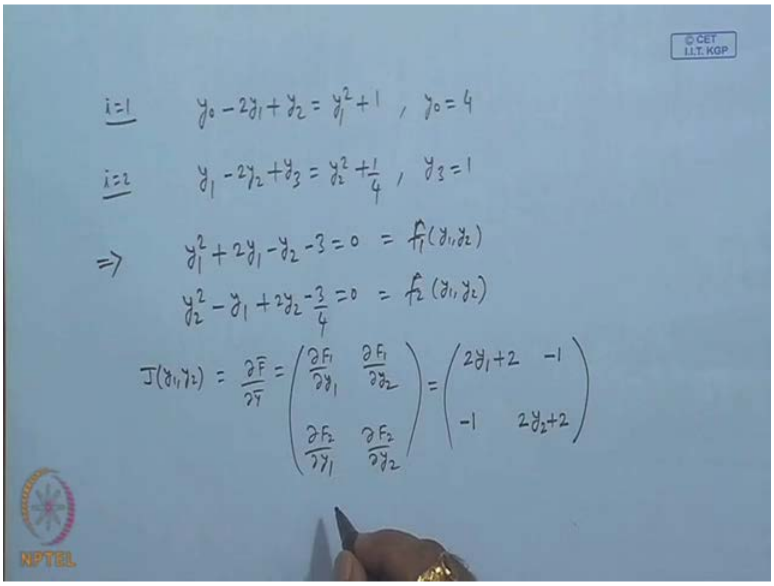

So, the motive behind so Newton Raphson’s method for solving non-linear system, so what we have is this, suppose we expand in Taylors series about y i plus. So, for a vector equation dau f bar by dau y bar so this is h square because, second order y minus y i square. So, y bar is minus f of and this must be approximately 0, so this must be approximately 0. So, from here y bar I am retaining then, we must transfer these things, I am transferring this, but then this is a matrix. So, it becomes inverse there and minus y i becomes this. So, therefore so this is now what is the inverse so this is the Jacobian dau f by dau y bar is the Jacobian and hence it is a inverse.

(Refer Slide Time: 40:16)




So, accordingly the solution is an iterative method is defined so where the Jacobian. So, let us look at an example, let us look at an example, these are the boundary conditions. So, x0 is 0, x1 is one-third, two-third so the discretised version is. And we have y0 and y3 so we have y0 is 4 and y3 is 1.

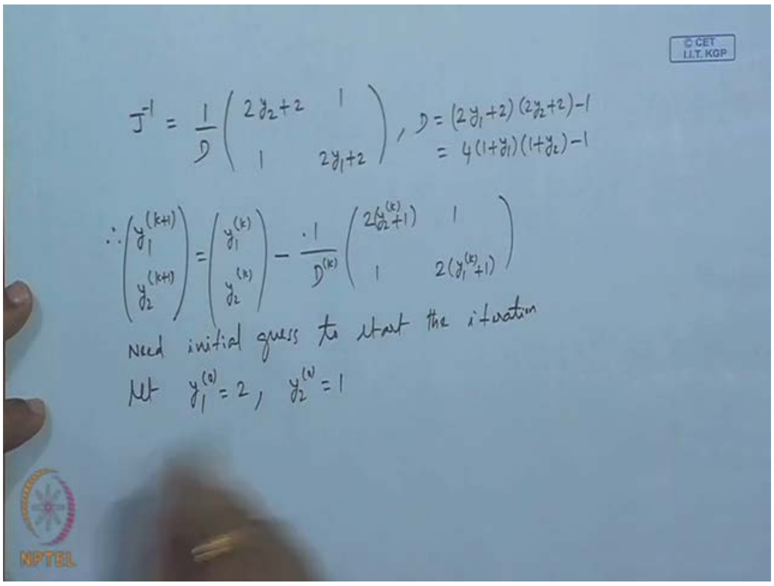
(Refer Slide Time: 42:59)




So, let us run the system at i equals to 1 and i equals to 2 so we have y0 is 4, 1. So, now it gets simplified, let us put it like this y0 is 4 so it gets transferred. So, this is our, now we need to compute the Jacobian. So, this the Jacobian so in this case treating this as f

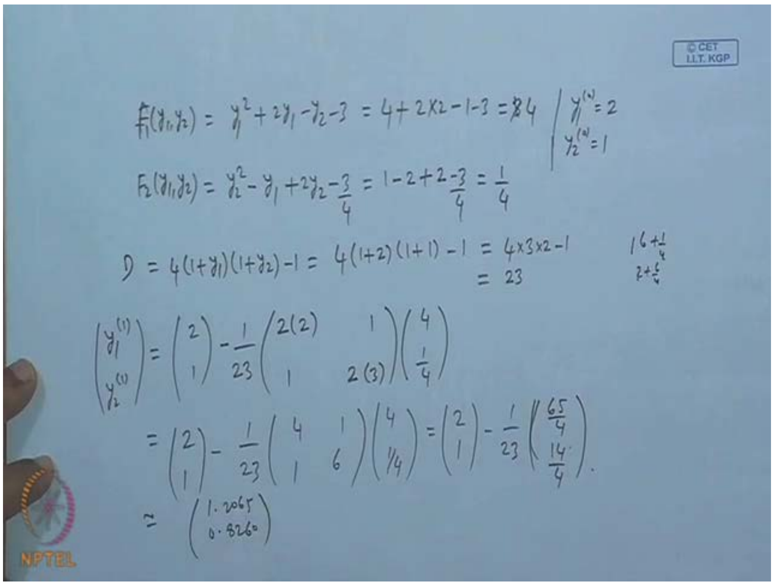
1th and this is f 2. So, this will be and with respect to 2 and here with respect 1, y1 and here so this a Jacobian.

(Refer Slide Time: 45:28)



Then, we need to compute this is, therefore, minus j inverse so where of course D is given by 2 y1. So, this is the iterative method now we have, we have to obtain y1 and y2 so this is Newton Raphson, need initial guess to start the iteration. So, let so let this be the initial guess.

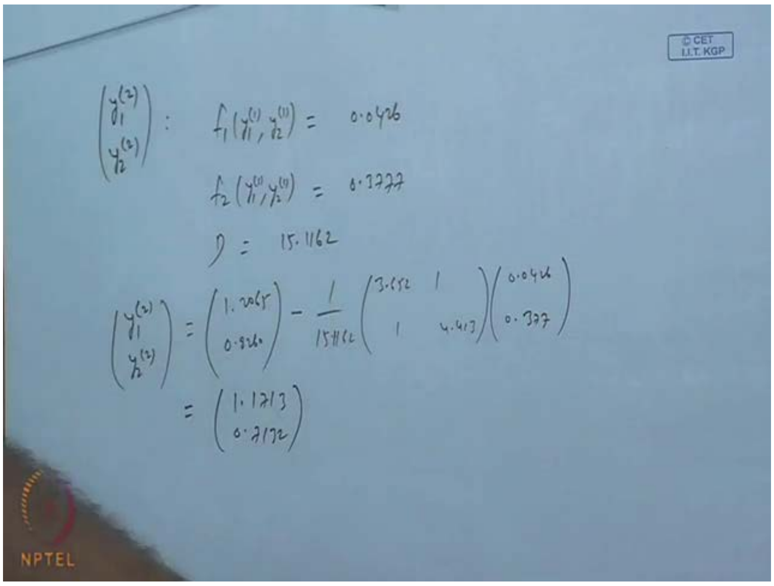
(Refer Slide Time: 48:02)



Then, f1 this is, our guess was y 1 0, 2 y 2 0 so this will be 4 plus 2 so y 1 square 2 minus 1 minus 3. So, this is so this is 8, this is 4 then, this will be y 2 square, this will be minus y 2, y 2 is 1, that is correct.

So, y 2 square minus y 1 so y 2 square y 1 plus 2 y 2 so this will be 1 by 4 and d so d was so this 4 into 1 plus y 1. So, 1 plus 2 1 plus 1 so this will be 4 into 3 into 2. So, accordingly y 1 0, y 2 0 minus 1 over D then, we need this terms two times y 2 plus 1 so 2 times y 2 plus 1 is 2 1 1 2 times y 1 plus 1. So, this multiplied by f1, f2 so here 4 and 1 on 4 so this is, so this is 2 1 so you will get 16 plus and here we get so we can compute and we get some value. So I guess I have done it, but this is subject verification.

(Refer Slide Time: 52:36)




So, then we have to iterate further y 1 2. So, in order to compute we need f1 of so this probably, please verify there could be a mistake in the numerical calculation. Then the corresponding d and then so the other values so we may get some value. So, this the next iteration then, we continue further so we stop when we have decide accuracy.

So, essentially with respect to the Newton Raphson method, we are trying to solve the non-linear system and then, this is just a solution like this. So, may be while calculating this is 0 so throughout we have to multiply by the inverse. So, may be I made a mistake so this will come here so this is the linearization in some sense. So, the guess is correct when you have non-linear equation, remember the non-linearity is only with respect to right hand side not in the derivatives, so far.

So, if that is a case we get system of equations and then, in this case the corresponding system of equations are non-linear and we use Newton Raphson method to solve these non-linear system of equations. So, far the stories for a simple boundary conditions suppose, your boundary conditions involve derivatives. So for example, one can classify the boundary conditions like, if the function value is given say like, type and derivatives are type then, a combination is given then robin type.

So, if the derivatives are involved what would happen whether, the same techniques work well, the same techniques work, but the derivatives need to be discretised right. So, may be they will bring in little complications so we have to discuss them with a special care, so until then bye.