Matrix Solvers

Prof. Somnath Roy
Department of Mechanical Engineering.
Indian Institute of Technology, Kharagpur

Lecture – 32
Basic Iterative Methods: Jacobi and Gauss-Siedel

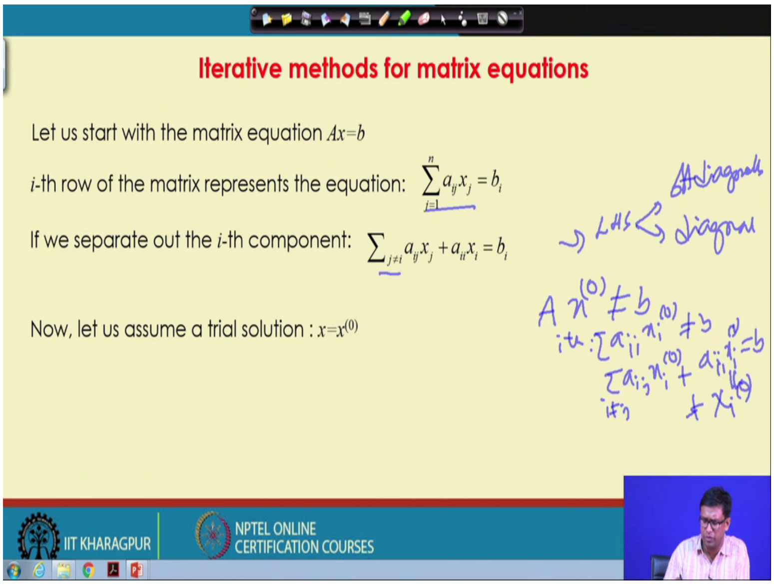
Hello. From this class we will start our discussions on Iterative Methods. Iterative methods are basically the solution method, it is not only for matrix solvers for any type of equations where we start with a guess value and we iterate on the guess value. That means, we keep on making better guesses using certain relations certain formula some correlations type of things. We keep on improving the guesses and use an updated value which is close to the actual solution and then and then sometime at some point of time we appear to the exact solution and we call that the solution has been converged to the exact solution.

So, we will start iterative methods for matrixes from this class. Earlier also we have discussed about one type of iterative methods if you can remember when we are discussing about tri diagonal matrix algorithm and thought like this these are applicable only for 1D differential equations. But thought that how this can be applicable for two dimensional or three dimensional geometries.

And we have seen that we can start using a guess value; so, that the 2D problem becomes 1D matrix equation problem. We put guess values for the for that perpendicular direction and then keep on iterating and change the direction go in a go in one direction with the guess values and then, we will go in another direction with the guess values. We will keep on changing the direction. And finally, yeah and I got a converse solution.

So, this class, but we will we will look there are a plethora of iterative solver techniques and this class we will start looking into very basic iterative solution techniques which are maybe couple of century old methods. And you has been implemented and utilized by many researchers till date for solving matrix equations and these are very robust methods for the class of matrix. They are supposed to work, they work very well and even today, we see lot of advanced computer programs which are still using this type of matrix solvers.

(Refer Slide Time: 02:46)



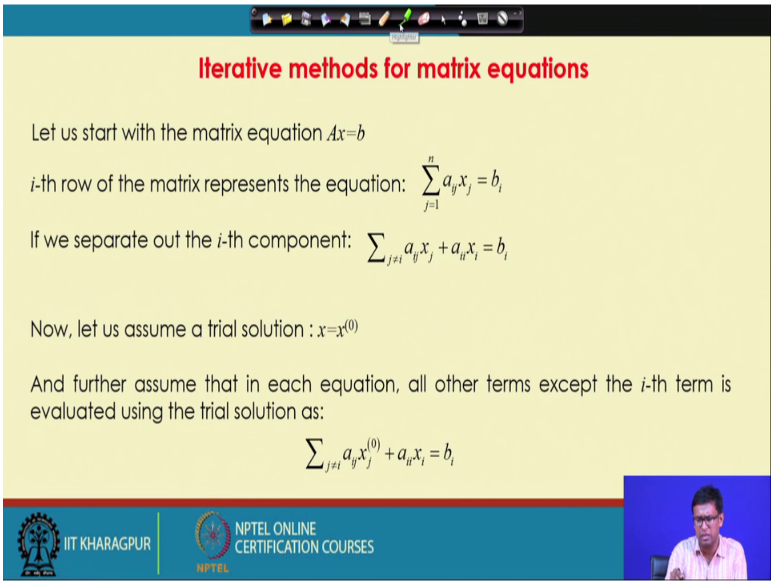
So, we will start our discussion on basic iterative methods for matrix solutions. Let us start with the matrix equation Ax is equal to b, now Ax is equal to b if we look into one particular row of this matrix this is, one row is gives us one equation of this equation system which is j is equal to 1 to n a ij x k is equal to b i. Now, if we take the diagonal component out of it which is if we separate out the ith component from the left hand side, from the left hand side we separate the ith component. This is same as a ij x j where j is not is equal to i plus a ii x i is equal to b.

So, LHS has broken into two parts: off diagonals and the diagonal and their product with the x vector. Left hand side has been decomposed into two parts. Now let us I assume a trial solution or a guess solution x is equal to x 0. This can be anything of course this is not a right solution. So, you need not get A x 0 is equal to b r. Our check will be because I told that in an iterative method and we; obviously, know by this time. That in an iterative method, we will start with a guess solution we will improve the guess or we will update the guess based on certain conditions or using some inductive logics of relation.

And then, we will try to achieve in the final solution. So, the guess solution is not the exact solution. Therefore, if we substitute x 0 into the equation, we will not get Ax 0 is equal to b. And once we will reach to the final solution or the exact solution, we will get Ax is equal to b. So, that check is that if we are getting x is equal to b is or not. So, we start with the trial solution x is equal to x 0 and of course, we can write that A x 0 is not is equal to b x x 0 is the vector.

So, ith row will give me that a ii x i 0 sum is not is equal to b we know that. But as this is a trial solution, let us write that some a ij x i 0; i is not is equal to j plus we say perhaps separated out the diagonal part. We are not using the trial solution for the diagonal part, let us see that diagonal part has some value which should give me satisfy the equation. So, plus a ii x i is equal to b and this x i is not x i is equal to 0 that, then we could not have written this equation. But now as we are breaking it this is not this is not is equal to x i 0. This is not the trial solution this is a some different value of x i and we write it the as the first case of x i.

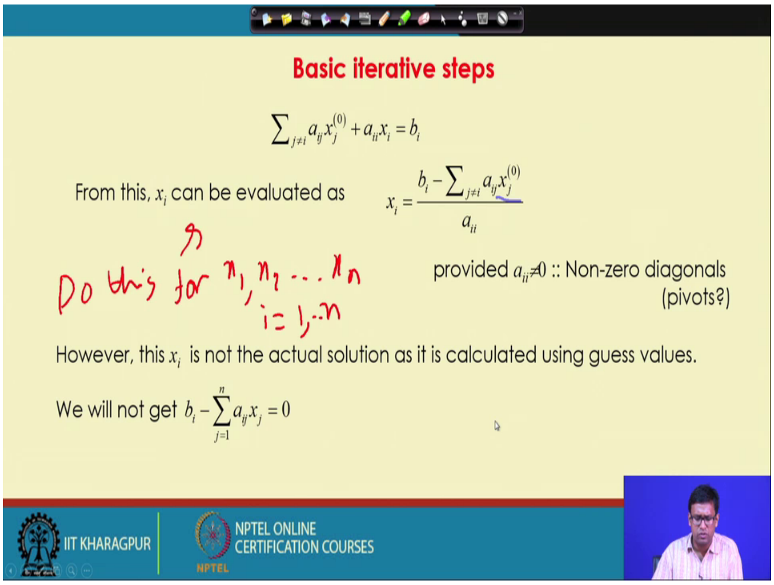
(Refer Slide Time: 06:04)



So, what we will do is do that we further assume that in each equation, all other terms except the ith term except the term corresponding to the diagonal value which will be multiplied with the diagonal term is evaluated using the trial solution.

So, we can write that j is not is equal to i a ij x j plus a ii x i. This is not had the this is not the trial solution, this is separate from trial solution is equal to b i because if I put the trial solution, I cannot write a ij x i x j is equal to b i. It is not trial solution is something else it will not it will be an inequality. But let us put an unknown here which is x i which is not known to us and we can therefore, write it. So, what we will do? We will solve this it with this is one equation, one variable; we will solve it we use the guess value of x 0 we will solve it and get some value x i.

(Refer Slide Time: 07:13)

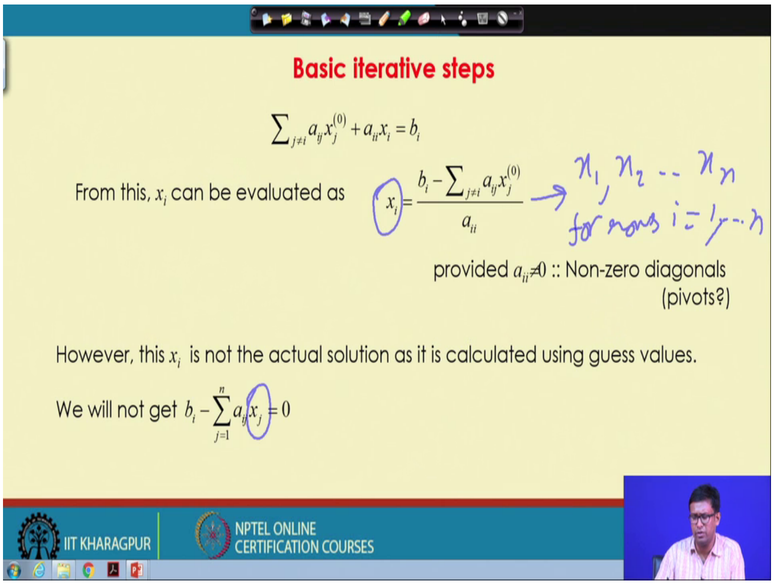


So, we will get an updated x i using this. A ij x k 0 plus a ii x i is equal to b from this x i can be evaluated as x i is equal to b i minus j is not is equal to 1 a ij x j 0 by a ii. Now this can be done in one particular case only that a ii is non-zero. So, this is possible provided a ii is the diagonal term of a is non-zero a has non-zero diagonals or non-zero pivots himself if we keep on doing some of the row operation it should have non-zero pivots.

So, this is very important that A must have non-zero pivots or non-zero diagonals, otherwise this operation would have given us something infinitely large wrong value; however, x i is also not the actual solution because when you are calculating x i, we are calculating it based on the trial solution which are not the exact solution. So, x i cannot be also the actual solution x 1 is calculated based on x 2 x 3 x 4 which is the trial solution.

Similarly, x 2 will be calculated bits based on x 1 x 3 x 4 x 5 which at the trial solution. So, we will do this; for we will do this for all do all x do this particular step for x 1 x 2 up to x n; i is equal to 1 to n. For all the rows, you we do this step. We will get a get some value of x 1, x 2, x 3 which are an updated value, but they are not the actual solution because each value is calculated based on the guess values. So, we can get b i minus a ij x j and this is this x j is basically this x j is basically the updated value.

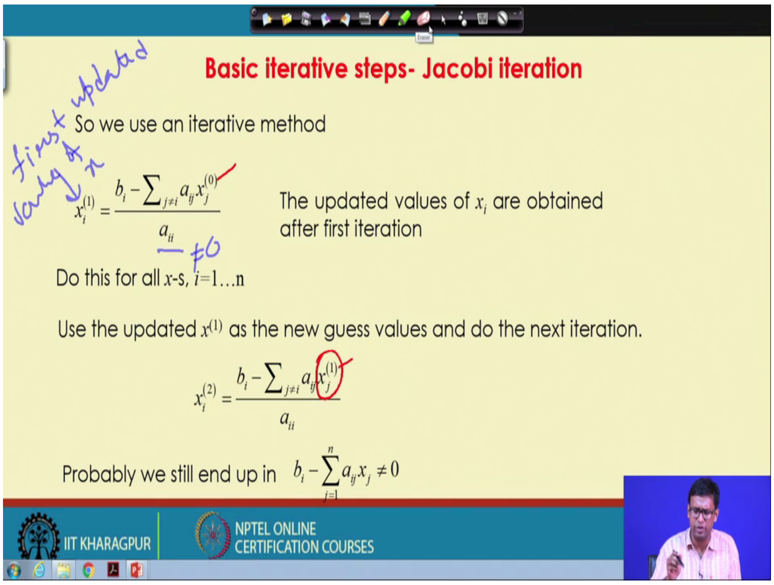
(Refer Slide Time: 09:14)



We will still not get bi minus a ij x j is equal to 0, because this is not the actual solution. So, what we will do? We got some updated value which is not that actual solution is now you use this updated value from x 1. So, we got here using this we got x 1 x 2 x n for rows i is equal to 1, for i is equal to 1 we got x 1 updated i is equal to 2 do the same operation got x 2 updated. So, the similar operation like that.

So, we got a updated state of value. So, we got an updated x vector, we will use this x vector as the new trial solution and that is the iterative method that you iterate with that the with the updated values.

(Refer Slide Time: 10:19)

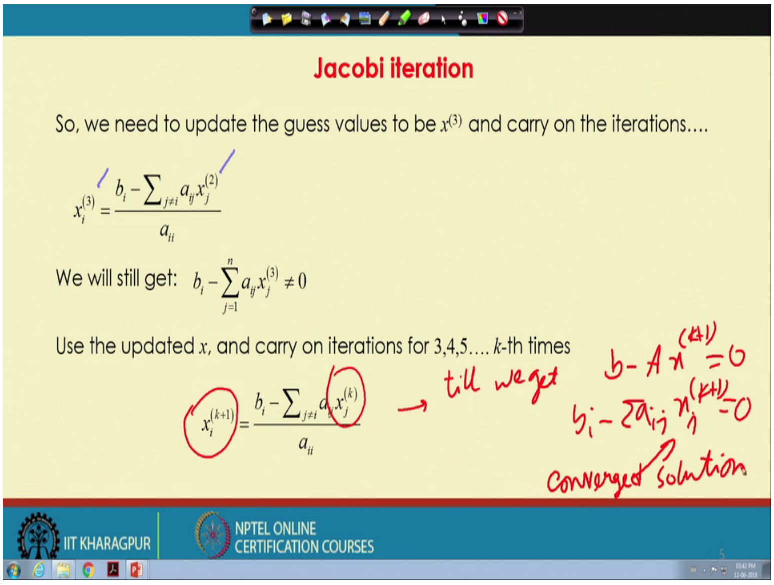


So, we use an iterative method that x 1 one which is the x i 1 which is the first updated value. So, this can be written as the first updated value of x. So, this is coming after the first iteration, this is coming from the first iteration: x i is equal to b i minus x 0 j is not equal to a ii and again this is doable only when a ii is non-zero.

The updated values of x I; these, this is the updated values of x i which are obtained after the first iteration. So, we will do this for all x s x i is equal to 1 to n and we will use the updated x 1 updated vector x 1 which was x 1 1 x 2 1 to x n 10 as the new guess values and do the next iteration which is x 1 at the x i at the second iteration is bi minus a ij x j ij is not equal to i and x j 1 x j 1. This is the, here this was x x j 0 and x j 1.

So, here we are using the updated value and probably we will still end up in b i minus a ij x i x j is not equal to 0, at least there will be some i for which we will get that will most likely. So, we have to keep on iterating, till we get this value to be 0.

(Refer Slide Time: 12:07)

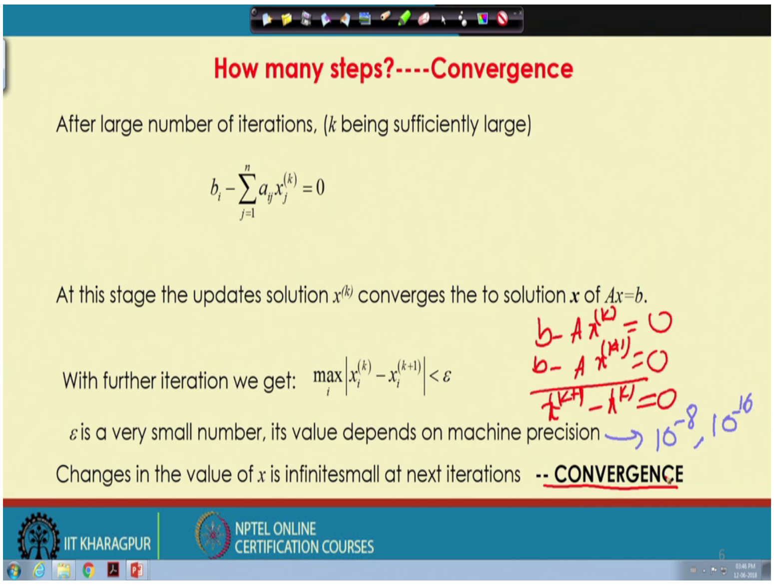


So, you need to update guess values to be x 3 and carry on the iterations that x third level like x i 3 is equal to bi minus a ij x 2 x j will be using its second level iteration value here by a ii.

And we will probably still get vi minus x i is not equal to 0. So, now from 0, we came to 2, 1 from 1 we came to 2, from 2 we came to 3 and we have to continue these iterations use the updated x and cannot carry on the iterations 3, 4, 5 up to the k th time k is the large number. And the k th iteration will be we get k plus 1 th we get k plus 1 th updated value of x i using k th value of x k th iteration value of x. And this we will do till we get b minus ax is equal to 0 or ax k plus 1 is equal to 0 when we get b minus Ax k plus 1 is equal to 0 or bi minus sigma a ij x j k plus 1 is equal to 0 this is the converse solution. We will say that we have converged to the final solution.

Now, this x will be the exact solution of this equation because the minus x will be exactly 0. So, we will keep on increasing the number of k and after sufficiently large number of k if the method is right, we should right and rightly implemented then we should must get a value x k for which we even if they update it x k plus 1 will give me a converse solution. And this convergence we will see it is asymptotically it is slowly converges to the right solution with the b minus Ax slowly reduces and finally, it becomes 0 or very close to 0.

(Refer Slide Time: 14:38)



So, after large number of iterations, k being sufficiently large we get b i minus a ij x j is equal to 0 at k th iteration is equal to 0. At this stage updated solution x k converges to the solution x of ax is equal to b. Now, once b minus a x j k is equal to b we have the equation that b minus Ax k is equal to 0. Now, if we still use the iteration technique, we will get A x k plus 1, but x k plus 1 is obtained assuming x k s which already satisfied.

So, for x k plus one we will also see that b minus Ax k plus 1 and as we will keep on iteration, we will get the later A k plus 2 k plus 3 etcetera that is also satisfying the equation. They are not exactly satisfying the equation because there will be round off error always they 0 is never 0; there is no leading to the problem and we will see what you how should you how small should we go. Because for a round off error, we cannot get exactly 0 as a real number here or as a floating number here 0 will be 10 to the power minus 7, 10 to the power minus 8, 10 to the power minus nine. So, its nearly equal to 0 right.

So, if we subtract this 2 we get x k plus 1 minus x k is also 0. So, if we if the solution, when the solution converges to x is equal to b, the difference between x k plus 1 minus x k will also be very small. So, if we further iterate the solution will not improve because it has already converge to 0 although all the further iterations will also give us converge solution or the exacts or something very close to exact solutions or solutions will not improve.

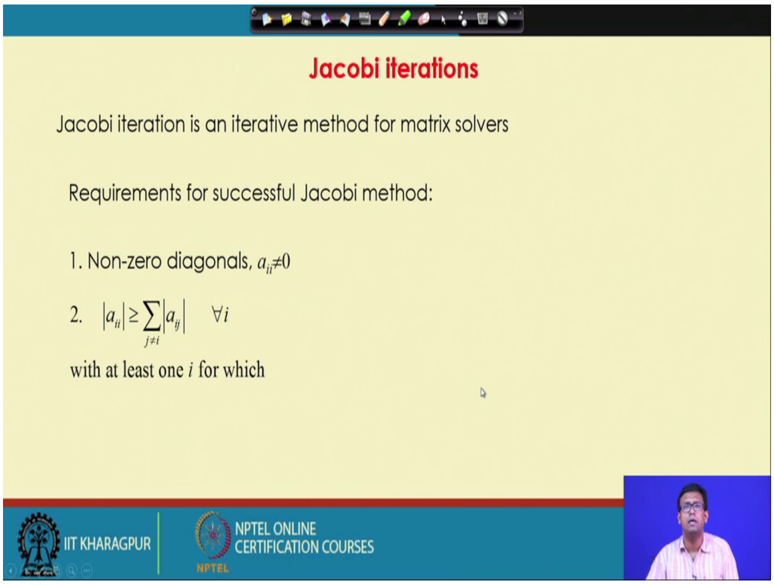
So, with further iteration we get that x k and x k plus 1 is very close to each other or maximum value of the difference because x is a vector it has multiple component; x i k minus x i k plus 1 and its maximum absolute value is less than epsilon. Where epsilon is a very small number and its value we decide what is the value of epsilon where when writing a program and we decide the value based on the machine precision.

If the machine cannot calculate anything less than 10 to the power minus 8; we say that the epsilon is 10 to the power minus 8 or so. So it is epsilon is you really of the order of 10 to the power minus 8 for single precision case is 10 to the power minus 16 for a double precision etcetera. It is a very small number changes in the value of x is infinite small at the next iterations and that is why we call it convergence because x has been converged to the exact solution.

Now, if we try to change the value of x by, the change will be infinite small at the next level of iteration. So, it is now converge to a particular value and this value is if b minus x is equal to 0. These value is the exact solution; however, later we will verify that if its converges change in x is very small, then x is converse to the right solution because they really do not ensure; it is it does not really ensure that if x k plus x k minus 1 is very small b minus x will also be Ax will also be 0 though I gave some hints here.

But, we will later verify it and this method is called a Jacobi iteration method. The iterative method I discussed is called a Jacobi iteration method and Jacobi iterative method, it is an iterative method for matrix solvers. The requirement for successive Jacobi method is that the diagonals must be non-zero and we have seen that because the formula at a each iteration that we have used is b minus a ij something if we can look into this formula there is a division by a ii. So, a ii must be non-zero if a ii is 0, then the equate then this method fails.

(Refer Slide Time: 19:53)

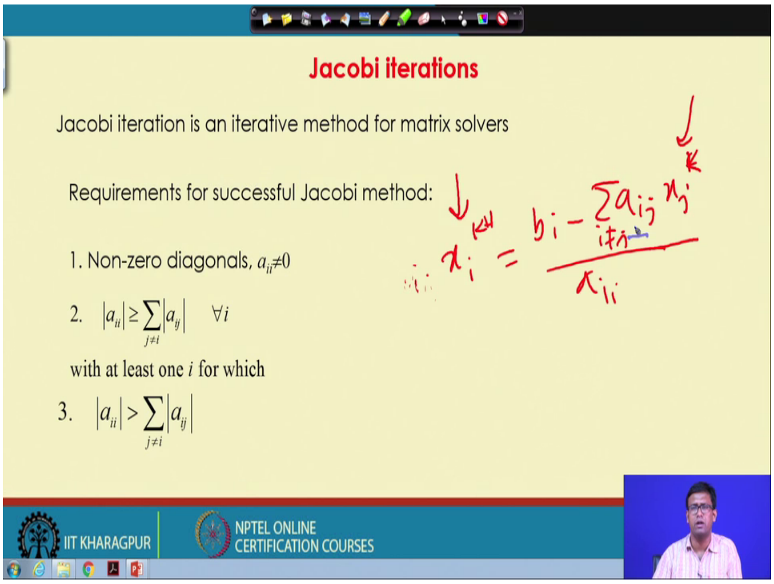


So, this is one of the requirements; the diagonals are non-zero. Further, we have seen that the diagonal must be greater than or equal to some of the all of all off diagonal terms, then only Jacobi method will work with at least 1 i. They should or for all is for all values for all rows the diagonal should be greater than equal to some of the off diagonals and there must be at least one i for which the diagonal is absolute value of the diagonal is greater than equal to some of the absolute value of greater than some of the absolute value of the all off diagonal terms.

For all the rows, the diagonal cannot be less than this absolute value of the diagonal cannot be less than the some of the absolute value of the off diagonals, but and there has to be one particular row for which the diagonal absolute value of the diagonal must be greater at least one row. There has to be at least one row for which the absolute the value of the diagonal term must be greater than the sum of absolute value of all the off diagonals, then and only then and only then Jacobi method will work.

So, it has to be it has to be a matrix with non-zero diagonals that; that means, diagonal is greater than sum absolute value of diagonal is greater than sum of all the off diagonals greater than equal to. So, if there is a non-zero diagonal if there is a 0 diagonal, the entire row has to be 0 which is not possible there has to be a non-zero diagonal term and the diagonal must be greater than or equal to sum of absolute value of off diagonals. There has to be at least one row for which the diagonals in it is absolute value is greater than the sum of the absolute values of all the off diagonals. So, it is called a diagonally dominant or irreducibly diagonally determinant matrices for which only for this matrices take a view at work.

(Refer Slide Time: 22:09)

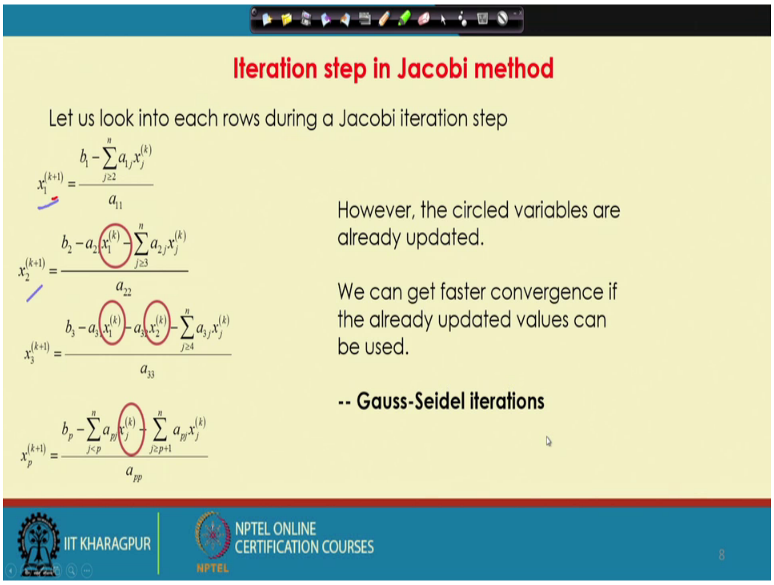


And we can only think a little bit also in a sense that one Jacobi step is written as x i at k plus 1th iteration is equal to b i minus sigma a ij x j i is not equal to j at k th iteration by a ii. So, we are is using a guess value for all the off diagonals and calculating the diagonal using that the value, we are using guess value to multiply with all the off diagonals and updating the x i based on this value.

In case they are very large, the effect of the guess values will always remain. The updated update in the x which is because the off diagonal is this update is the effect of the guess values is small because this is a large term here. If this terms are large, then this then the guess effect of guess values will be very very high so, that the update in the x will be not that significant. Therefore, with large number of strips probably we will never be able to reach the final value because I want to reach to the final value using the guess values only.

Out of the guess values we will have pulled me back if the coefficient that is being multiplied with the guess value is very large. So, that is one the there is a philosophy that is for which we need the diagonally dominant matrices whenever less this can be mathematically shown that this method is stable and converges only for diagonally dominant or irreducibility diagonal determinant. We will discuss these 2 definitions matrixes.

(Refer Slide Time: 24:16)



So, we can probably write that let us go to the next slide for that. So, now if we look into the iteration steps with one particular iteration step in a Jacobi method for the k th iteration step, the k plus 1 th value is coming as b 1 minus j is greater than 2 a 1 j x j k a 11 because j is equal to 1 is that I am going to term here. The x 2 value is b 2 minus a 2 1 x 1 k minus j is equal to 2 will not come here because we are calculating x 2 here j is greater than equal to 3.

So, j is equal to 3 j is equal to 4 or. So, on a 2 j x j k divided by a 2 2. Similarly x 3 will be b 3 minus all the terms except the a 33 term, a 31 x 1 k minus a 32 x 2 we doing the guess value. The third term is not here again starting from 4 a 3 j x j k and similarly x p k plus 1 will be b p minus a p j x j k j is less than p minus j is greater than p of j is greater than equal to p plus 1 starting from p plus 1 a p j x j k a pp.

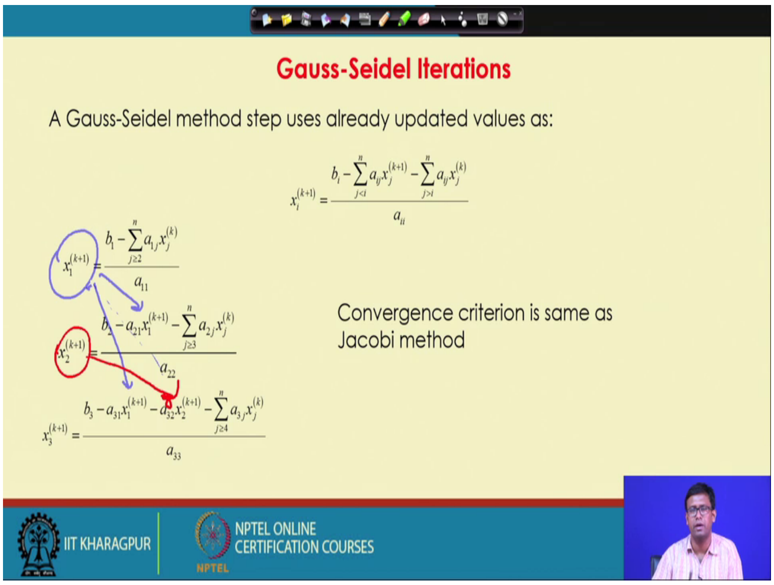
Now, if we relook into these steps. When we are calculating x 2, we already have an updated value for x 1, when we are calculating x 3 we already have an updated value for x 1 as well as an updated value for x 3. Similarly when we are calculated x p we already have updated values for x up to p minus 1 x p minus 1 k plus 1 is already available because we are trying to reach to a final solution by updating the values. What would happen if we use all the already updated values for which the way this values are already available to us.

For example if we are calculating for x 2 if we use already updated value of x 1. When we calculated for x 3 if we use already updated values for x 1 and x 2, this 1 and 2 are the components of the vector; not the iteration level at any iteration level. When you are calculating the third component of x vector, first and second components are already known to us if we follow the Jacobi steps.

So, if we use all the already updated values here. So, what our observation is that x 1 is already known when we are calculating for x 2, updated value of x 1 which is which is this one. Similarly when we are calculating for x 3 x 1 and x 2, updated values x 1 and x 2 is updated values are already known to us. When we are calculating x k plus 1, then all the updated values up to x x 1 x 2 up to x p all these values at k plus 1 th level are known to us.

So, the circle variables are already updated and we can get faster convergence; if we, if the already updated values can be used and this is called a Gauss Seidel iteration. These type of iteration can be modified to Gauss Seidel iteration when during one iteration step, we are use the already updated values of few variables which have been in that iteration level only well which have been already updated.

(Refer Slide Time: 28:08)



So, if we it will be more clear if we look into the algorithm. A Gauss Siedel step uses the already updated values as x x 1 k plus 1 is equal to b 1 minus a 1 chicks check a 11. Because nothing at this stage, when x 1 is being calculated we only have the guess values at the k th level x 1 to x n nothing is updated. Now, when we are calculating x 2 x 1 k plus 1 is available to me and I use it while calculating x 2.

So, x 2 k plus 1 is b 2 minus a 2 1 x 1 k plus 1 which is already available minus a 2 j to x 2 j. When we are calculating x 3 x 2 k plus 1 and x 1 k plus 1 is available to us. So, I use x 1 k plus sorry. So, I will use x 1 k plus 1 here and I will use x 2 k plus 1 here. So, if we carry on the k a general step for k plus 1, it will be for the terms which are already updated which are the previous from the previous rows of the equation, we will use the updated value for ith term. For j less than I, you will use the updated value for j greater than I, we will use the guess solution or the updated value which is coming from last iteration and divided by a ii.

And we can see that it has same convergence criteria as the Jacobi method however; that means, that it has to be a ii is non-zero. And it has to be diagonally dominant at least one row has to be diagonally dominant diagonal term is greater absolute value of diagonal term is greater than some of the of absolute value off diagonals at least for a one row and for all other rows. The absolute value of the diagonal cannot be less than the sum of the absolute value of the off diagonals. This has to be satisfied.

This is the same criteria as the Jacobi method has to be there in Gauss Seidel method; however, Gauss Seidel gives us first a convergence. Now in next few classes, we will do few analysis with Jacobi and Gauss Seidel method. The algorithm is straightforward at some point of time we will also see how we can easily translate this algorithm into a computer program which is an important step also.

But in the subsequent classes, we will look into try to do analysis of the Jacobi and Gauss Seidel method and see how it is working as a matrix operation. And from there we will try to estimate on the, how fluid it convergence, what should be the speed of convergence, what should be the rate of convergence in these 2 methods. And what are the properties of the matrix in terms of Eigen values of the matrix or spectral radius of the matrix which ensures that the convert, then the method should converge to the right solution in that smaller number of steps. We will look into these things in the subsequent classes.

Thank you.