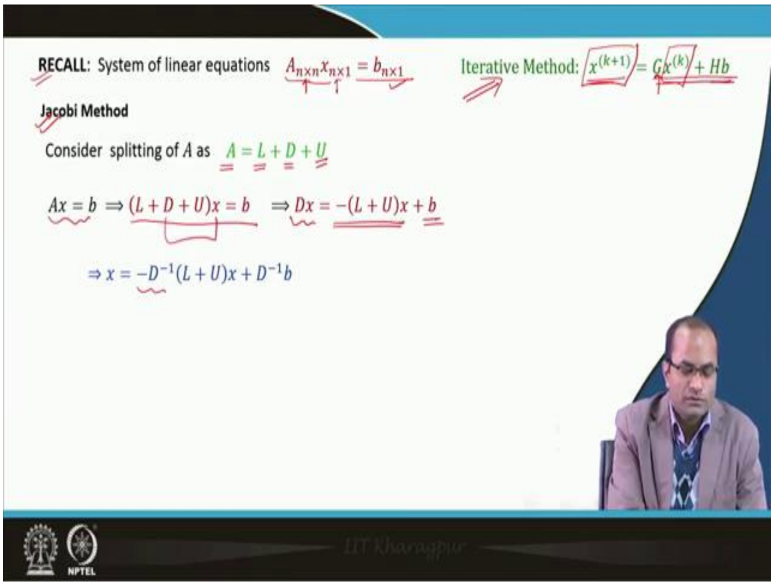
**Engineering Mathematics-II**
**Professor Jintendra Kumar**
**Department of Mathematics**
**Indian Institute of Technology, Kharagpur**
**Lecture - 22**
**Iterative Methods for Solving System of Linear Equations (Contd.)**

So, welcome back to lectures on Engineering Mathematics 2 and this is lecture number 22 and we will continue with iterative methods for solving system of linear equations.

(Refer Slide Time: 00:23)


So, in the last lecture we have already developed or derived two methods or two iterative methods, one was the Jacobi iteration method, another one was Gauss-Seidel iteration method. And now, we will use these methods to solve some system of linear equations. So, basically using Jacobi method and also using Gauss-Seidel method we will see how to get the solution of system of linear equations.

(Refer Slide Time: 01:00)

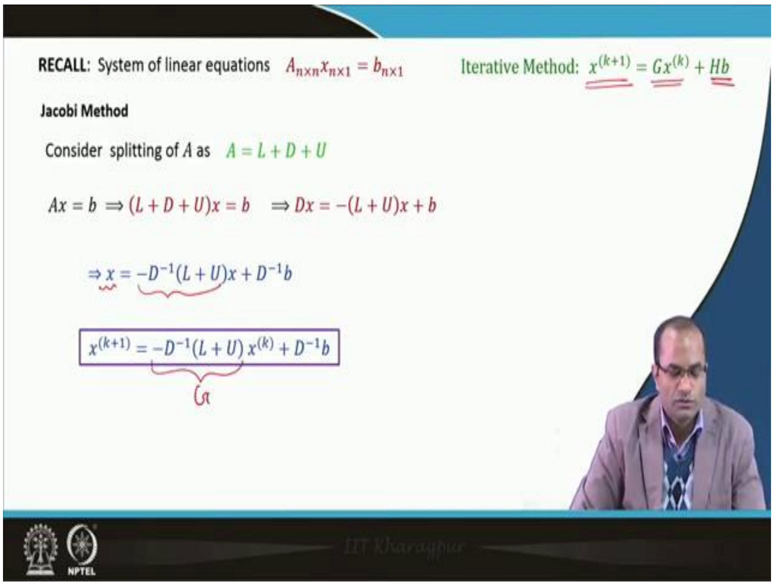



So, just to recall, if we have a system of linear equations, let us say Ax is equal to b, where this A is an n cross n matrix and x is n cross 1 or the column vector and b is also a column vector with n components. In, that case what we have seen that iterative methods, both the iterative methods which we have developed or in general the idea of iterative method is to have such a general form where left hand side we have x evaluated after this k plus 1th iteration.

And the right hand side we have G which is called the iteration matrix and then this xk which this is the vector evaluated or used at kth iteration and then we have some other matrix H into b. So, coming to the Jacobi method, we consider the following strip, the splitting that if we write A as L the lower triangular matrix plus D the diagonal matrix having entries in the diagonal from the coefficient matrix A and U is the upper part of the coefficient matrix.

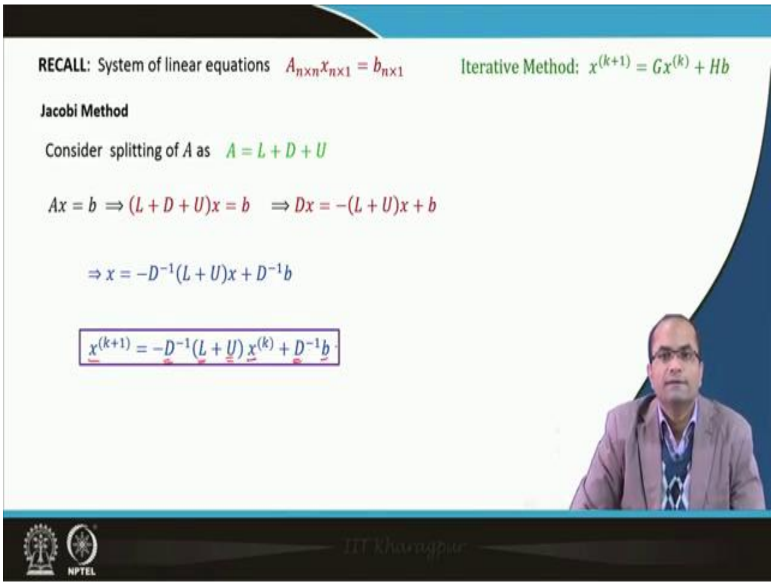
In that case, this Ax is equal to b can be written as L plus D plus U x is equal to b and from here we will derive the form of this iterative method getting this Dx together here and taking this L plus U to the right hand side, plus this b and if assuming that D inverse exist so we can multiply by this D inverse, so the right hand side we have D inverse L plus U and D inverse b.

(Refer Slide Time: 2:51)



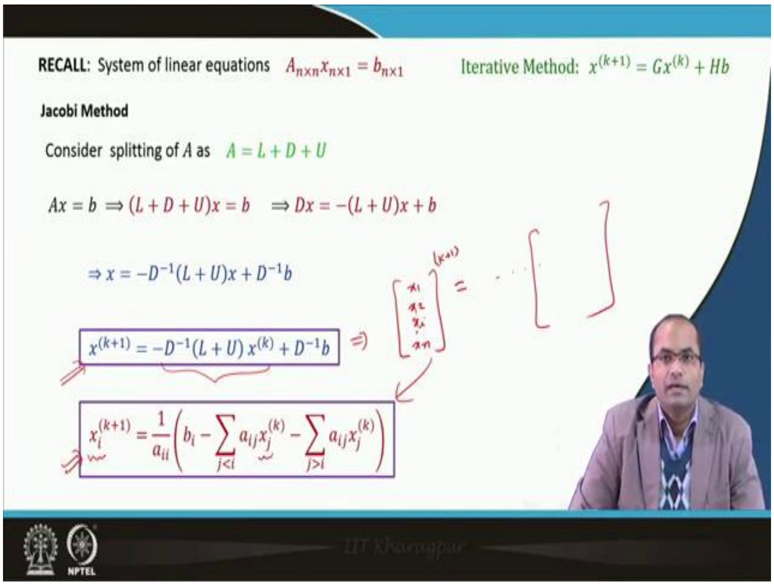

And this precisely the form which we are looking for an iterative method, so x k plus 1th iteration is equal to Gxk and Hb. So, here we can now set up the iteration. So, this x we will replace by this xk then the right hand side we have minus D L plus U which is the iteration matrix G, for this Jacobi method and then this D inverse is like H there are in the general setting.

(Refer Slide Time: 3:20)




So, this is the Jacobi method written in this matrix form were this x is a vector and then D is a matrix here L is matrix, U is a matrix, x is a vector, D is again a matrix and b is a vector.

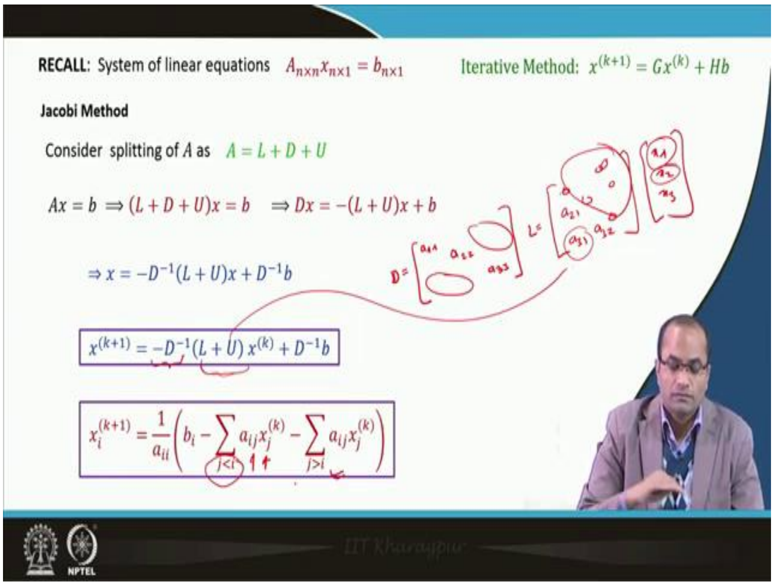
(Refer Slide Time: 3:39)




Sometimes it is convenient and we will also see in some numerical calculations that writing instead of this vector matrix form, this component form is much more convenient. So, this component form is exactly coming from this vector form, because here we have this x having for instance x1, x2, x3 and xn components at this k plus 1th level.

And similarly, the right hand side also after this multiplication we will have a vector of same components, number of components that is n. So, here what we have done, we have just written or compared each component equal, so the general xi component which is somewhere here let us say, so xi component is written now directly in terms of the coefficient and the other components of x.

(Refer Slide Time: 4:40)

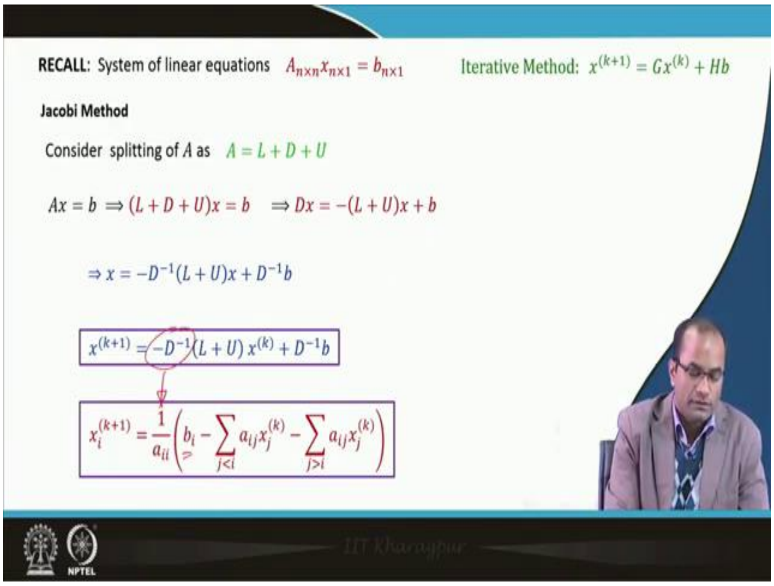


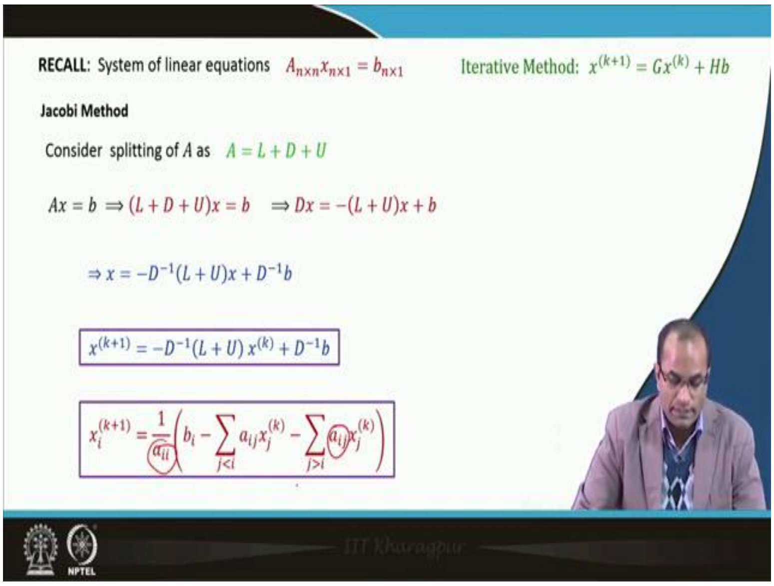

So, which can be seen directly from here because this D inverse, so D is the diagonal matrix having these entries a11, a22, a33 for instances we have 3 by 3 matrix and then all these components are 0 that is our D and this L and U are the lower and the upper ones. So, L is the lower, so here we have a 0 already on the diagonal and also on the off diagonal we will have 0 entries only these terms will survive that is a21 and here we have a31 and a32.

And similarly we have the upper triangular matrix were the next to the diagonal terms will survive, all other will set to 0 so that is the upper triangular matrix. So, when we multiply this L to x for instance if we take here x1, x2, x3, what will happen? You have the first term 0 basically, then a21 will be multiplied by this x1 and that is all in the third case we have the a31 and a32 will be multiplied to this one.

So, what is happing now here after this when we take this component? So, here when, whenever this j is less than this i then only this summation will take place, because otherwise this will became 0. And similarly, here when j is greater than i, it will be the other way round when we take the upper triangular matrix. So, this sum will be for j greater then i. And then for each row we will compare, so we will get that the ith component equal to this expression which is bi.

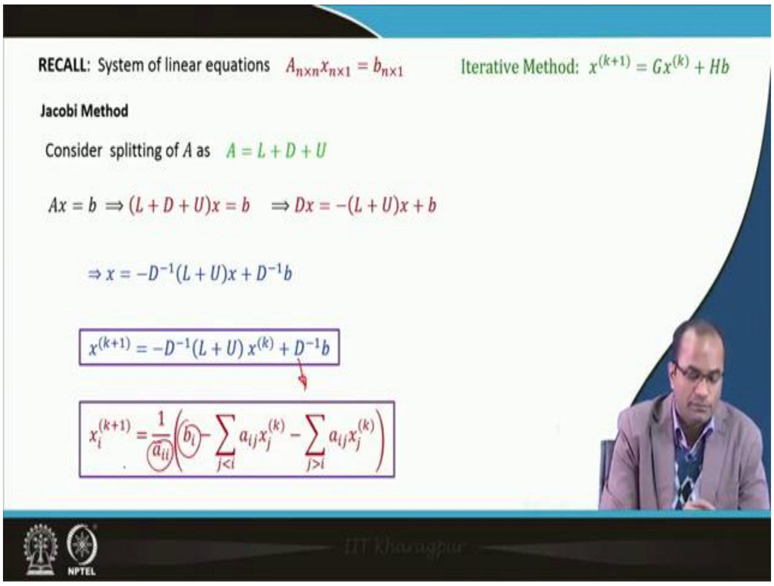
(Refer Slide Time: 6:36)

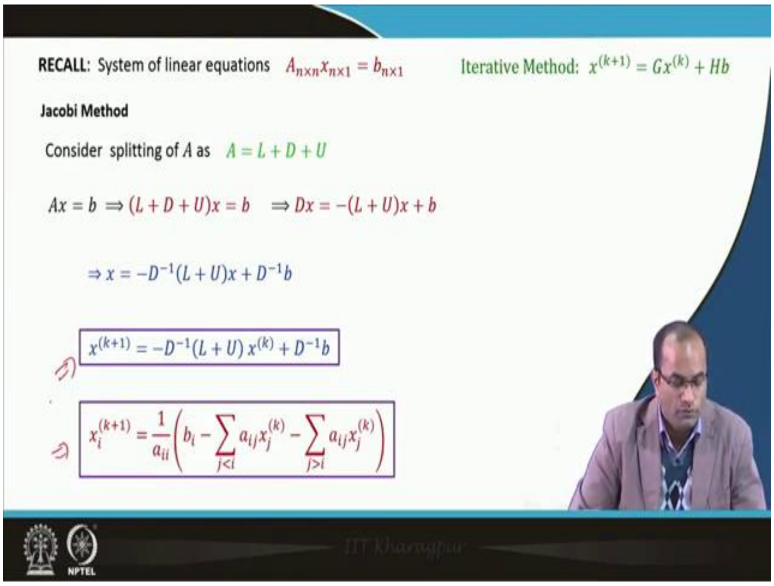



And this Di will just divide each number by the diagonal entries by ii, so here also we have this ii and also at this place we will get this ii.

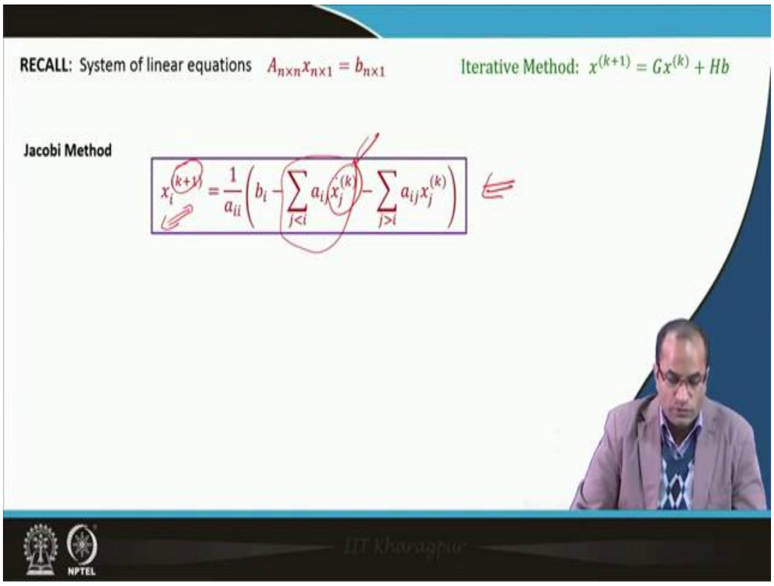
(Refer Slide Time: 06:49)

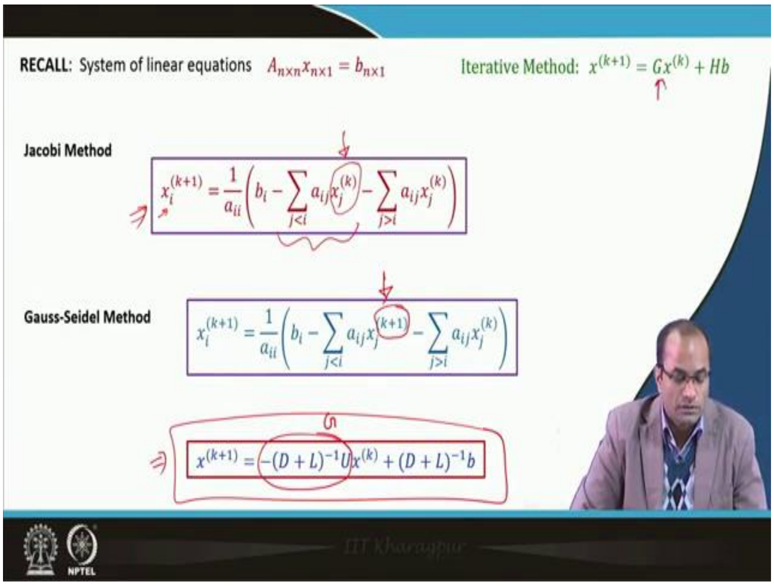


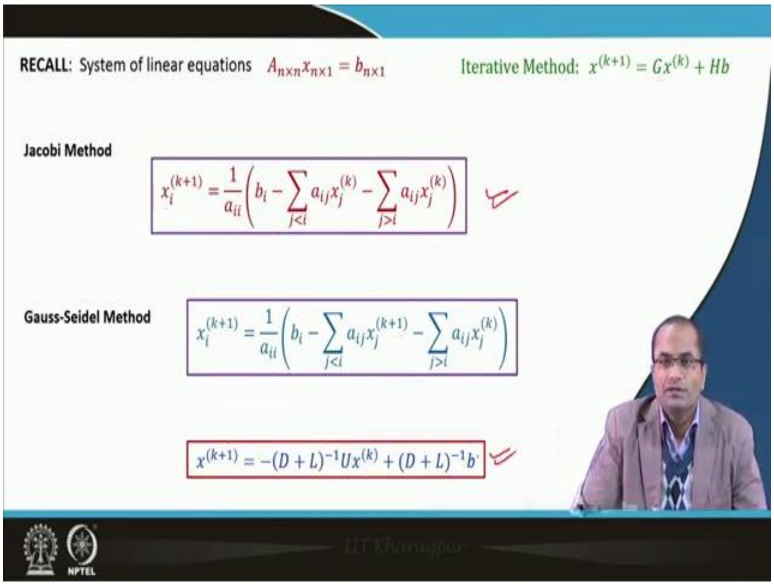




And this D inverse b will be again this bi vector divided by these diagonal entries aii. So, this is the component form and here we have the vector form of the or vector matrix form of the Jacobi method.

(Refer Slide Time: 07:11)





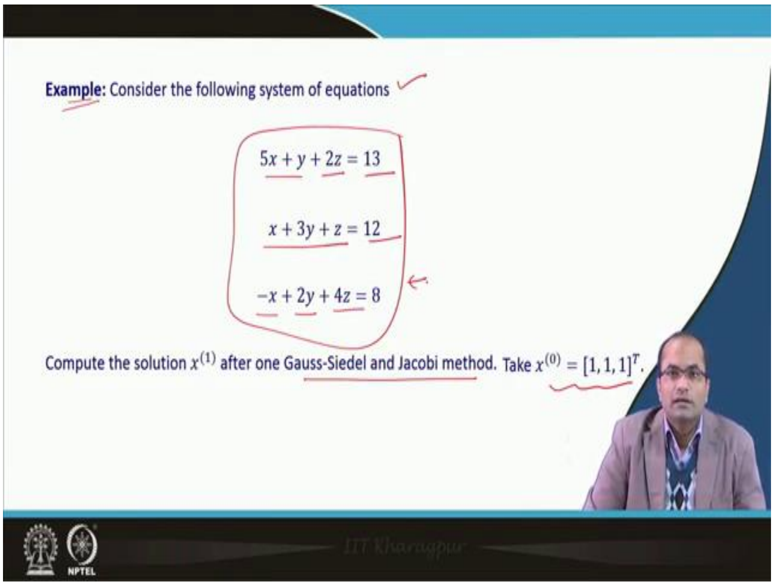

And then we have also discussed that once we have this Jacobi method for instances return in this component form. And from this component form it is very easy now to write the Gauss-Seidel method because the only differences in the Gauss-Seidel method, instead of using this xi's from the kth iteration from the previous iteration, because we are evaluating now at k plus 1th iteration and we are using all the values of x1, x2, x3, xn from the previous iteration.

So, the idea is that instead of using the values from the previous iteration, we can use the recently evaluated values and that means in this sum basically whenever this j is less than i by that time these xj's are available from the same iteration, because we are computing, let us say first x1, then x2, so when we compute x2, x1 is available which can be used here in this first summation.

So, the only difference is now for the Gauss-Seidel this k will be replaced by k plus 1 and as a result we have also seen in the previous lecture that this vector matrix form will take place now in this following manner that xk plus 1 is equal to minus D plus L inverse multiplied by U, so that will be the matrix G or the iteration matrix G for Gauss-Seidel method.

And then here also this D plus L inverse will come with b. So, these are the forms which we can use, either we can use the component form or we can use the matrix form whichever is convenient.

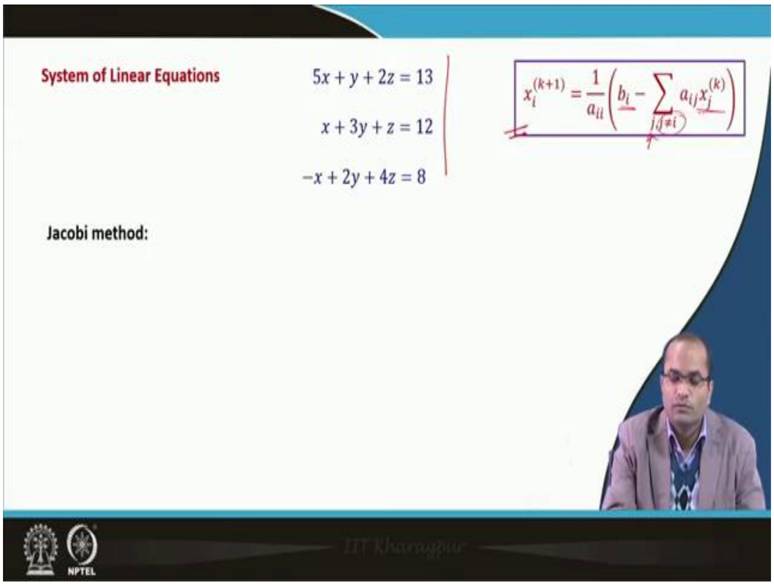
(Refer Slide Time: 09:01)




So, if we take this example now, we consider the following system of equations. 5x plus y plus 2z is equal to 13, x plus 3y plus z is equal to 12 and minus x plus 2y plus 4z is equal to 8. And we want to compute just x1 the first after this one iteration using Gauss-Seidel and the Jacobi method. What would be the approximation of Jacobi and the Gauss-Seidel method after one iteration. If, we start with the initial guess 1, 1, 1.

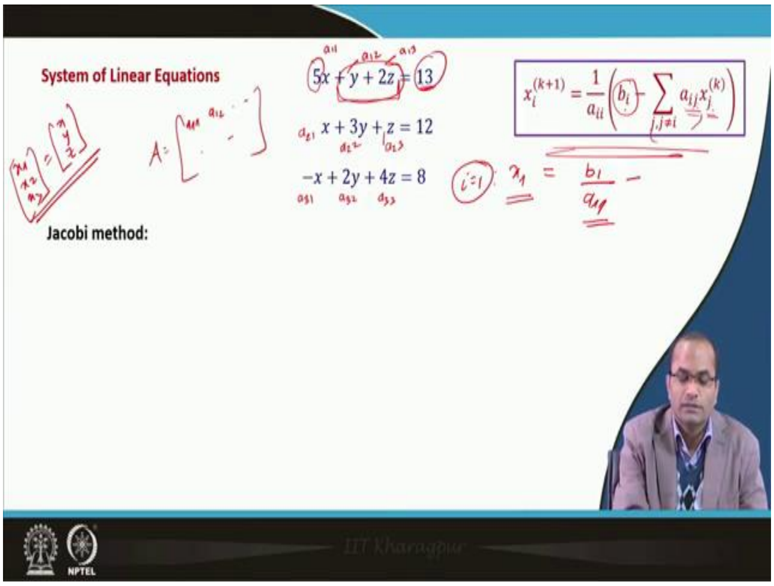
So, regarding the conversions of these methods for this given system, we will discuss a little later in the next lecture. So, in this lecture we will just evaluate these approximate values for instance in this case after just after one iteration.

(Refer Slide Time: 09:57)



So, these are the given equations the system of linear equations and we know that the Jacobi method has this form that one over aii and then we have bi minus and here the sum is over j where j is not equal to i and it is summed for all other values.

(Refer Slide Time: 10:24)



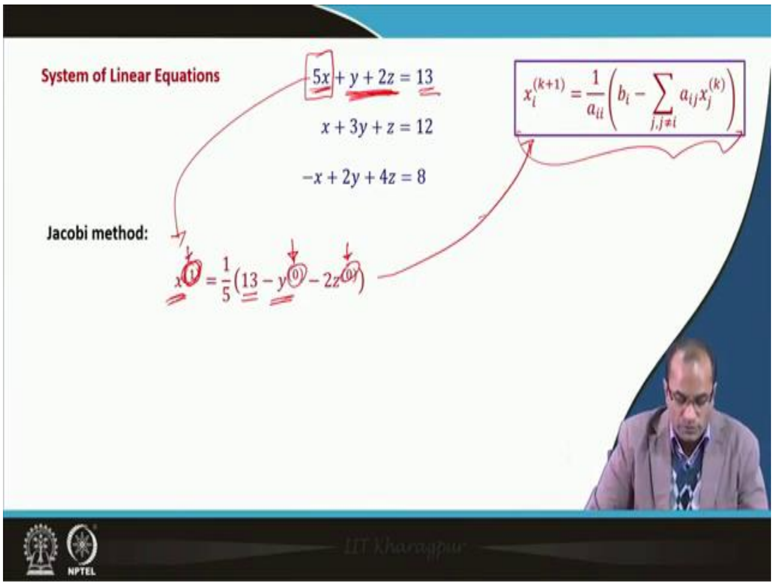

To write down the Jacobi method indeed we do not have to remember this formula because what this formula says that the right hand side. For example, we are writing the x1, so for x1 we say that b1 that is here already divided a11, this is a11, these are the coefficients.

So, this coefficient is a11, this is here a12 and this 2 is a13 and so on, here again we have a21, this is a22 and then here one this coefficient is a23 and then here we have a31 and then a32 and then a33. So, these are the coefficient matrix which can be written also in this form as this a11, a12 etcetera.

So, having this formula what we are doing actually this x1 is equal to b1 divided by this a11 and minus again we are dividing by this a11 that is this 5, when we sum with this aij to xj and this is nothing exactly it is a12 and a13, because the sum is over j so we have a1 so 1 is fix now, i is 1 for instance to write this equation.

So, what we are doing here, a12 x1, so x1, x2, x3 in this particular case, x1, x2, x3 in our component settings here represent just simply x, y, z, because here the notation for unknown is taken as x, y, z, not x1, x2, x3. So, here this getting this x1, the first component x it is like b1 over a11, so we have to divide by this 5, so we will get here 13 by 5 and then minus here the sums, so a12 with x1 and a13 with x3. So, a12 and a13 because when i is 1, j will not be equal to 1. So, precisely this is the number here now, so 1 into this component and plus this 2 into this 3rd component of this x is use.

(Refer Slide Time: 12:49)

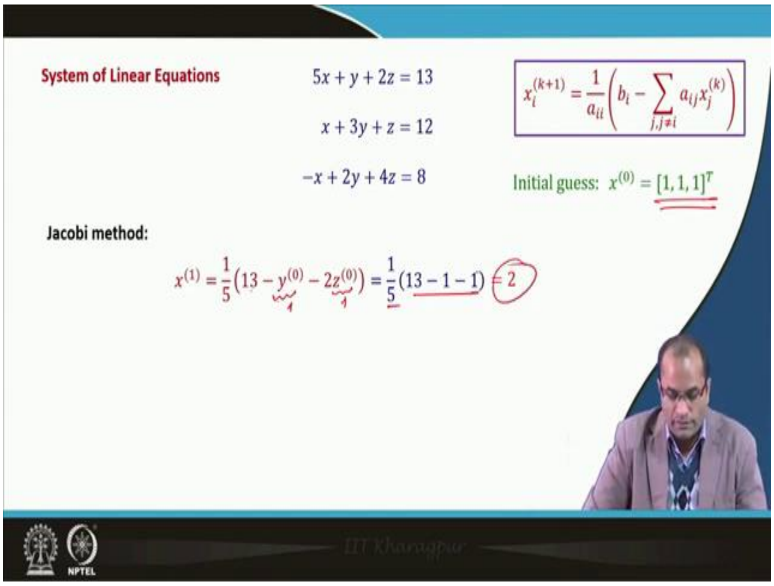



So, without remembering this formula what we can do? We can write the Jacobi iteration method from the first equation we will take x to this side and the rest these two y and z the other components will go to the right hand side.

So, right hand side we have 13 there minus this y obviously at previous iteration, so here if we are talking about the values at this after one iteration then these will be the values at 0th iteration which we will plug into this equations. So, here then minus 2z and this 0th iteration value, so this is the first equation which is exactly the equation we will get from this formula.

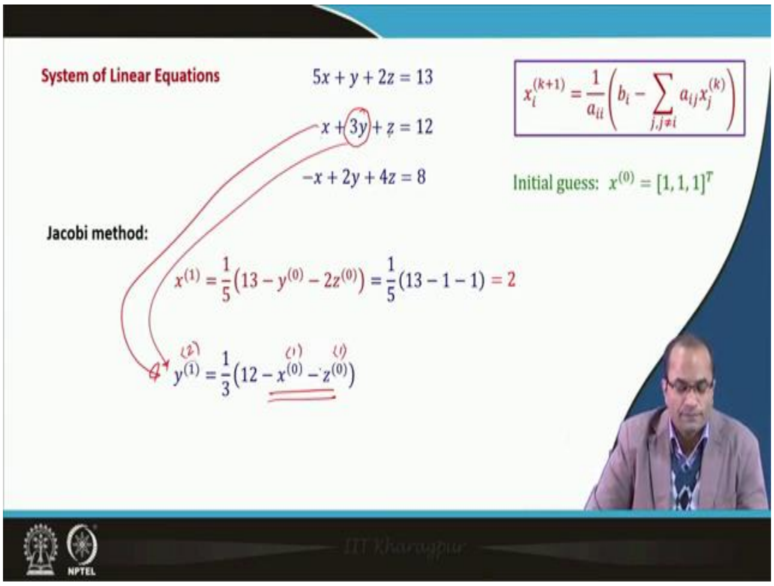
So, without remembering also what we will do? We will, in the first equation we will set for x, so x equal to everything will go to the right hand side and then we will set up this iteration the left hand side will put 1, the right hand side we will put the previous iteration that is 0 in this case.

(Refer Slide Time: 13:52)



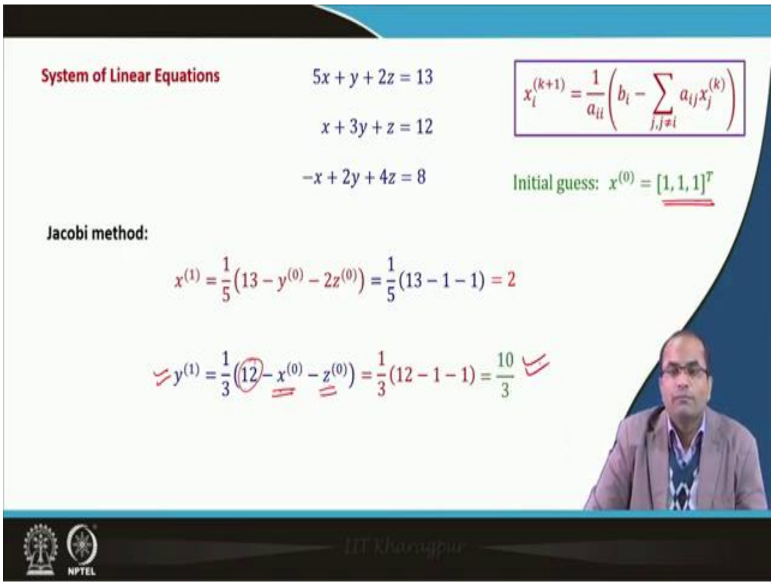

Initial guess was taken as 1, 1, 1, so if we take the initial guess so we have to substitute 1 here and we have to also substitute 1 there, that means 13 minus 1 and then minus 2 that is 10 and divided by 5, so we will get this value as 2.

(Refer Slide Time: 14:18)



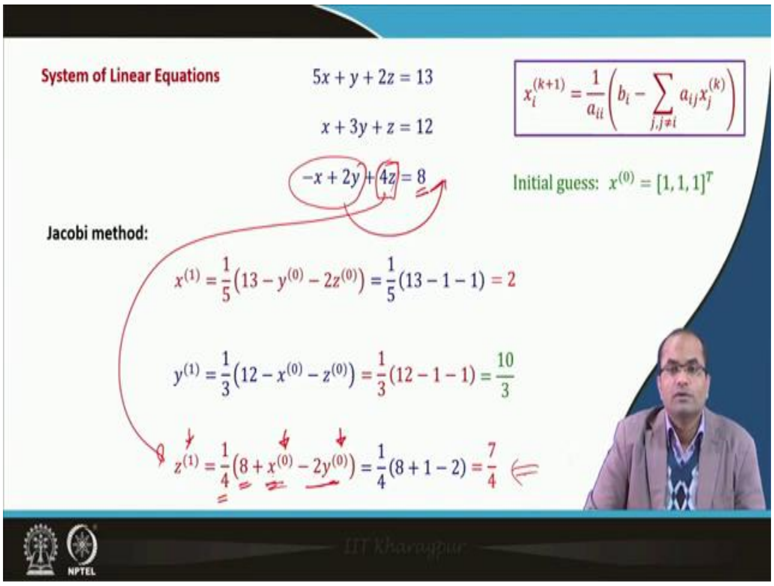

Regarding the second equation and again we do not have to really go through this formula directly we can write. So, from the second equation now we have to take y this side and the rest everything so x and z will go to the right hand side and then we will setup the iteration, so here we will put a 0, then we will put here 1. And the later iterations also for example the second iteration this will be 2 and this will be just the iteration before this that is 1.

(Refer Slide Time: 14:48)




Well so having this now for y we can again here we have x0, z0 that is the starting values which are or 1 1 in our case, so we have 1 and 1 that is 2 and 12 minus 2 that is 10. So, we have this 10 by 3 the second component, the approximate value of y after iteration.

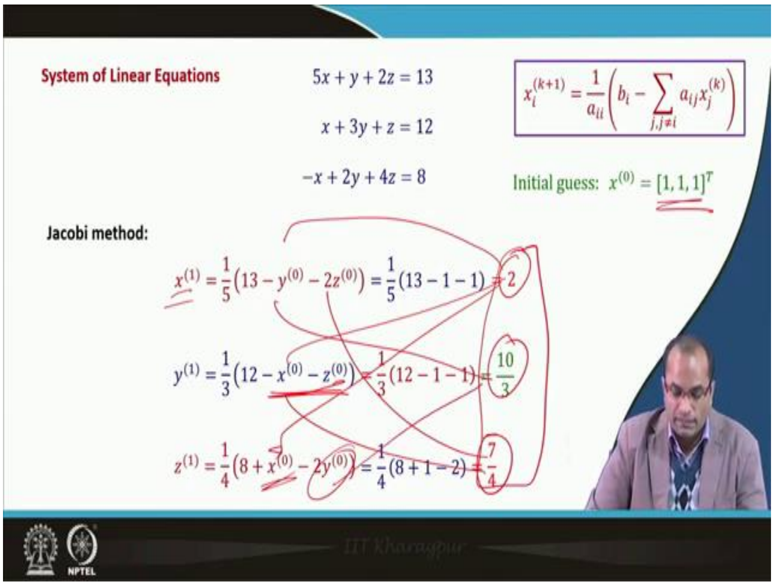
(Refer Slide Time: 15:14)




In the third one again, so in the third equation now we will write for z. So, that means this 4z will remain here and minus x plus 2y will go to the right hand side and then we will divide by 4 also. So, we have 1 by 4 then the right hand was this 8, then we have plus this x and minus this 2y and then we have set up these iterations, so the values are $0^{th}$ level and the left hand side we have the values after first iteration.

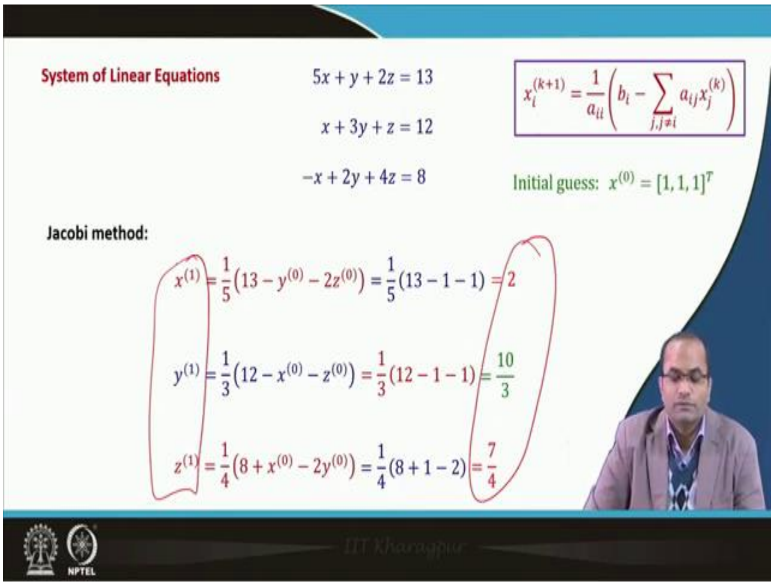
So, these are the initial guess which are known to us, so here we have 1 and minus 2, so this is minus 1 and then we have a plus 8, so this is all about we are, what we are getting 7 by 4. So, these are the values after 1 iteration.

(Refer Slide Time: 16:01)



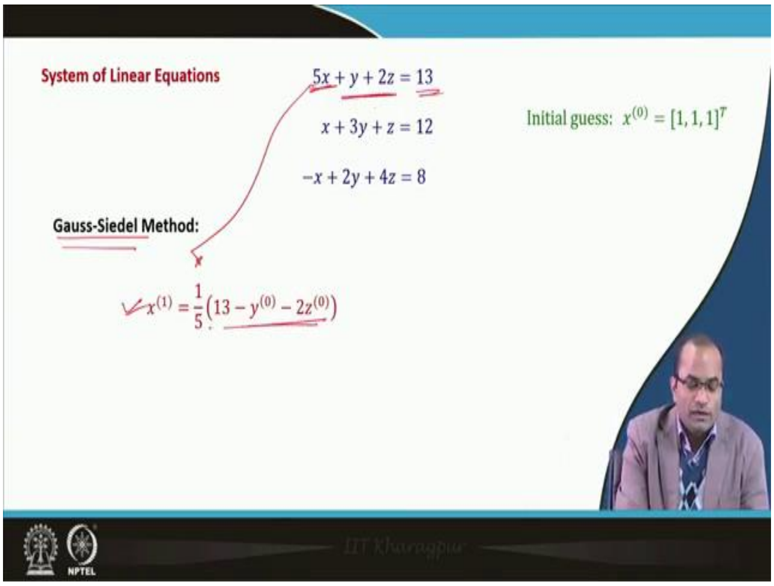

What will be now done if you want to get iteration, if you want to approximate for these two iterations or we want to get values after next iteration? So, these values here now instead of this 1, 1, 1, we will use these values in the next iteration, so for y for instance this will be used, for z this will be used and then we will get the new value after the next iteration for x. Again here this x and z values will be used x and z from here, again in this case x will be used as 2 and y will be used as 10 by 3.

(Refer Slide Time: 16:45)




So, we can apply or we can apply or we can use these values which we have computed now to forward our iteration to get the values after second iteration so these are the values after first iteration.

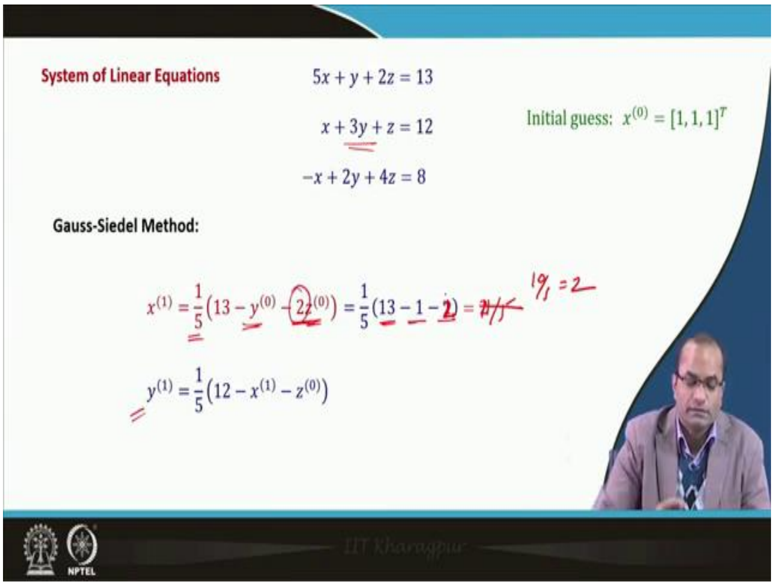
(Refer Slide Time: 16:58)

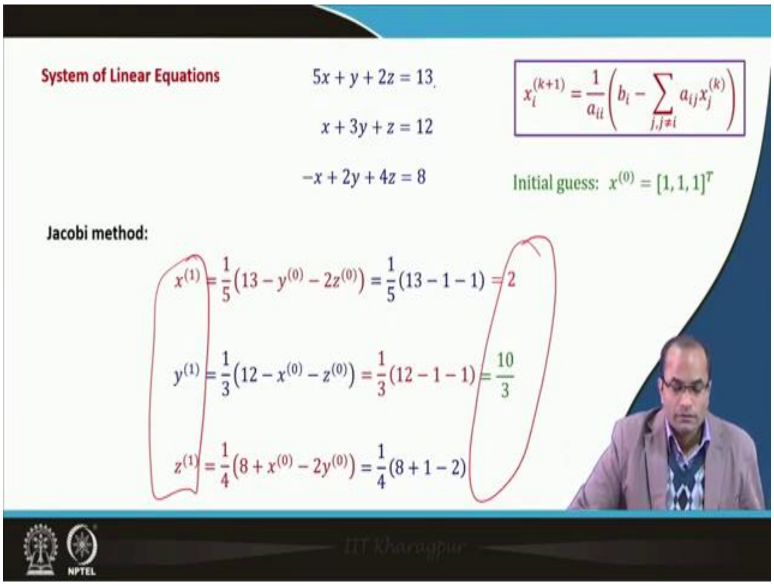



So, If you want to get for the Gauss-Seidel method, the only difference again we do not have to remember the formula for the Gauss-Seidel also, the only difference on this Gauss-Seidel and Jacobi would be, in Jacobi we were using for the complete evaluation after this first iteration for instance in each for the evaluation of each component we were using the initial values.

Now, here what we will do of x1 for instance the formula will remain the same as the Jacobi there will be no difference. So, from this equation we will write this as x1 is equal to 1 by 5 and this y minus 2z will go to the right hand side, so we will have this iteration scheme.

(Refer Slide Time: 17:52)

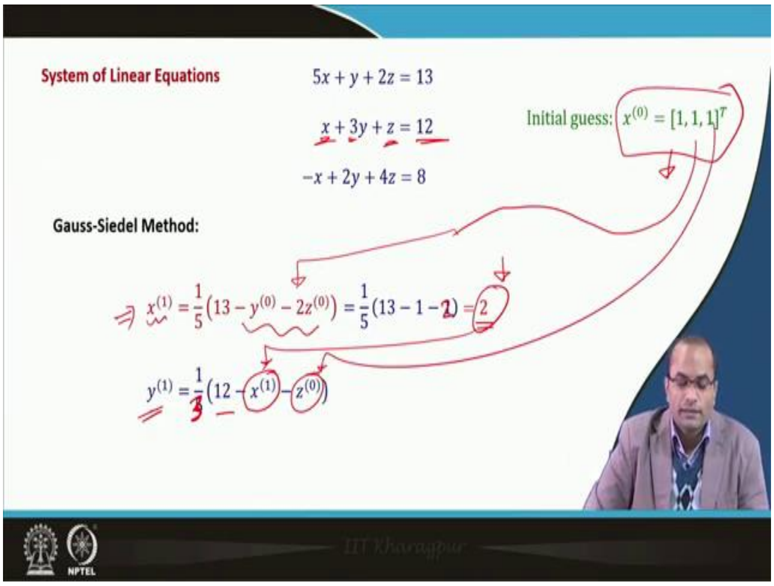



And then we have this 1 by 5 then 13 and since now we have only available values as the starting values, starting guess, so that we can plug here 1 and then minus 1, so we have this value as 13 minus 2.

So, this is actually 12, so if we go to the, if we go to the previous step, the value here 13 minus 2, so this is 11 by, 11 by 5 and there also it will be as 11 by 5. So, here also we have 13 minus this 2 that is 11 and then by 5. Coming to the next one, when we have this equation for y and there exactly the difference comes, so here we have this 2 sorry, so this is 2 here, so that's what it said 10 by 5 only, so the value is 2. So, this was 2 which was missing there in the earlier case also it is the same.

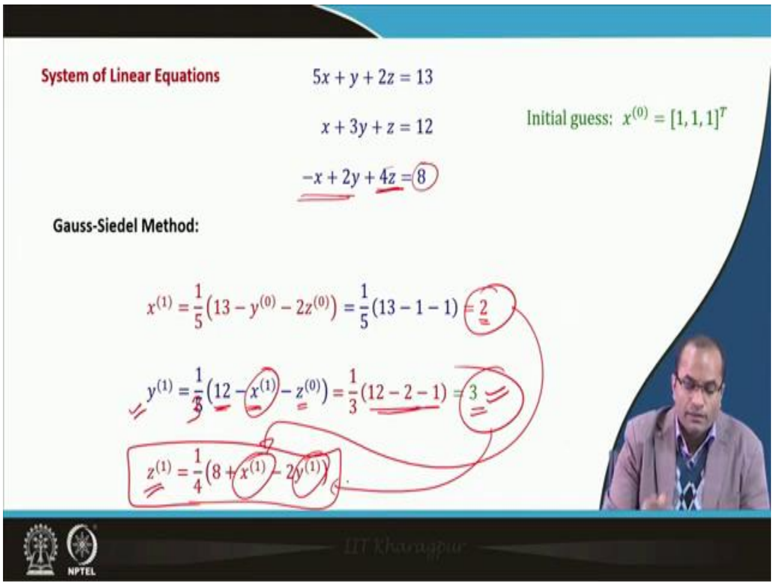
(Refer Slide Time: 18:54)




So, after this first iteration the value using the Gauss-Seidel as well as using the Jacobi method we have the same value, because at this stage when we are evaluating this x1, we do not have any other value then this initial guess. So, if the initial guess is same in both the methods then naturally this first component will be the same in both the cases.

Now, when we come to this y we have 12 minus, so we have 12 minus this x and minus z and divided by this 3 there. So, here we have slight difference because now we will be using x1 which we have just computed here, we will not use these values which are used here for instances in the first evaluation, because x1 is now known, the x1 we have a new value. So, that is the difference that this value will be used in this evaluation now and since z0 is not available, so we will use of course from the initial guess.

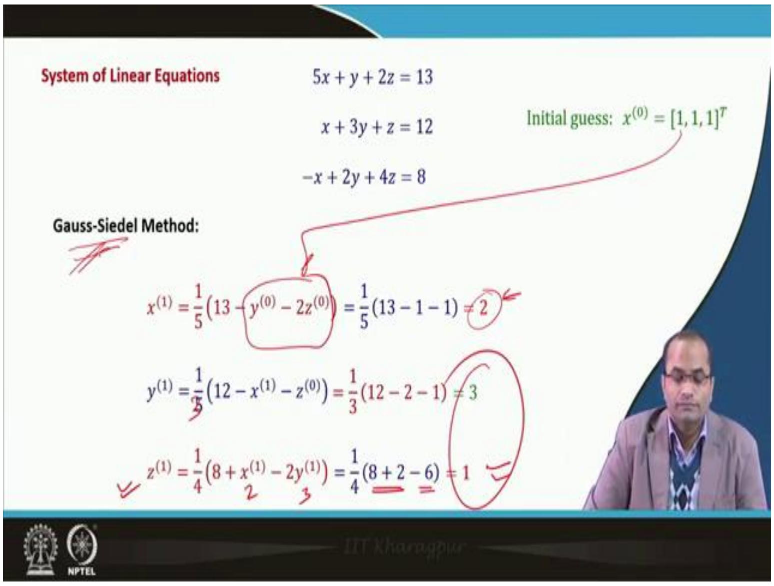
(Refer Slide Time: 20:06)




So, having this now we have 12 and then this is 2 there and then we have 1 there, so we have the values here 9 by 3, so this is the number 3 we getting for y1 which is different now form the earlier calculations form the gauss Jacobi method. Because in Jacobi method this x1 was not used, the x0 was use so that was 1 there and we got the 10 by 3, but now we are getting here as 3.

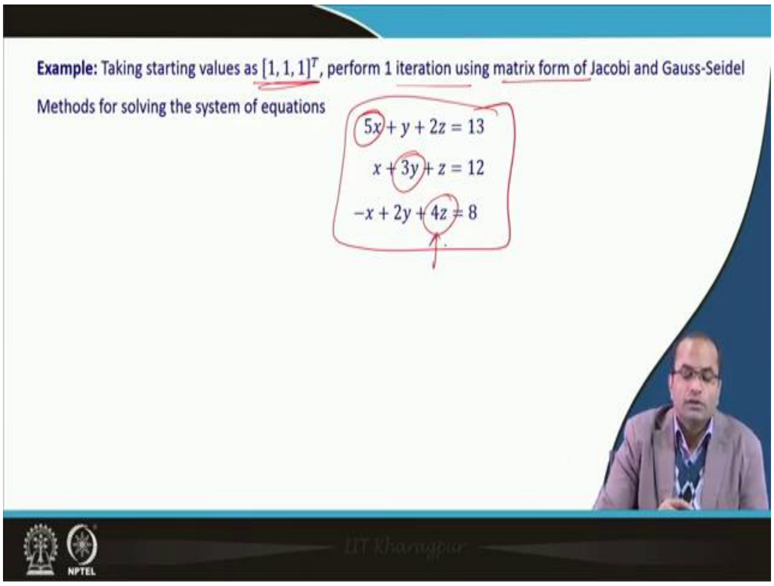
Coming to the fourth, third equation, so we will naturally get this 4z equal to 8 minus or plus this x and minus 2y and then this 4 we will divide, so this will be the setup for the iteration scheme. And now x1 is available and also the x2, y1 is available. So, x1 is available and here y is also available. So, we will be using these values now which we have just computed.

(Refer Slide Time: 21:15)



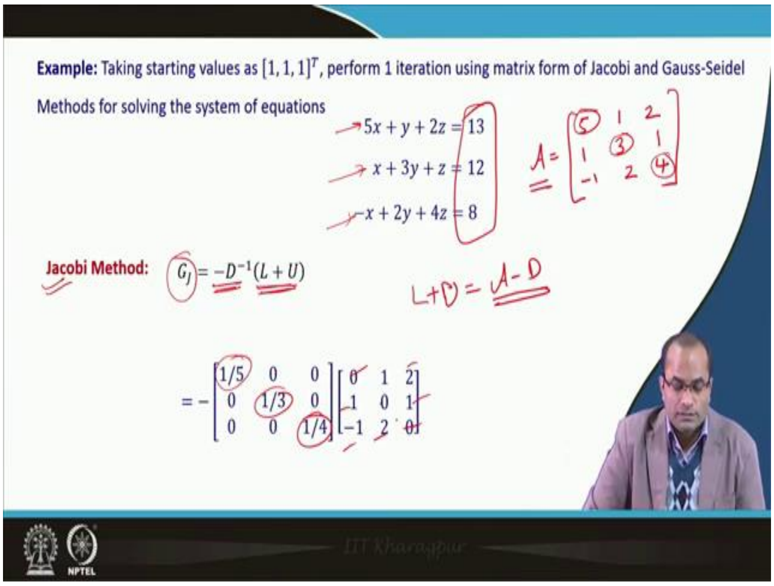

So, we will not use the values from the beginning. But we will use now the newly evaluated values as 2 here and 3 there. So, that means it is a 2 and then we have minus 6 there, so 8 plus 2 10 minus 6, we have 4 and by 4, so we have the value 1. So, these are, these two values are different then the Jacobi method whereas the first value, because here nothing is available we have to use the initial guess only, otherwise the formulation is same, so that is the slight difference that in Gauss-Seidel method we are using whatever we have available recently evaluated values available with us, whereas in Jacobi method we finish the complete iteration with the values from the previous iteration.

(Refer Slide Time: 22:06)



Well, now the question is that if we take this starting guess 1, 1, 1, again and perform 1 iteration using matrix form of Jacobi and Gauss-Seidel method for solving system of equations. In the previous example, we have used the component form which was directly possible from the given equation, from the first equation we have written in term of x equal to something, then from 3y equal to and then 3 was dividend from the fourth, from the third equation we got z from here.

(Refer Slide Time: 22:49)



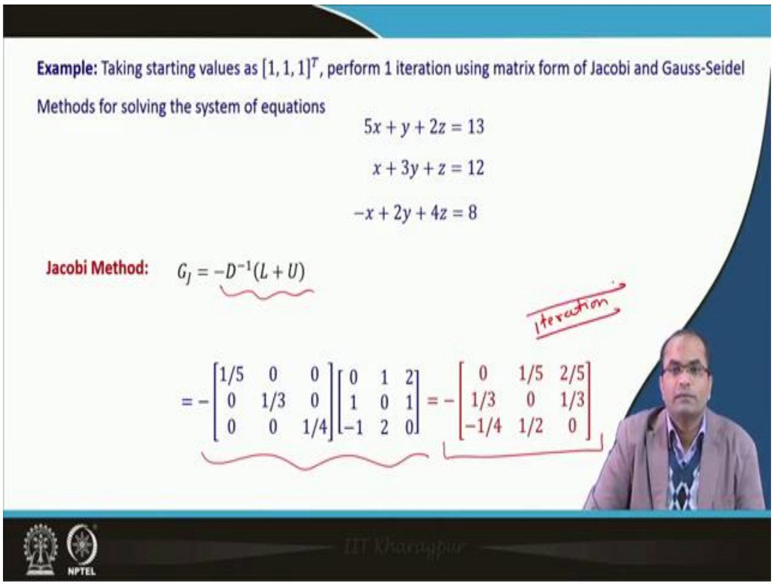

So, this was easy indeed to get these iterations for each component for x we got from first equation, the iteration for y from the second one and for z we got from the third equation. But, now we will write everything in the matrix form because sometimes using matrix form is much easier in many programing languages once we have these matrix form the computation is going to be very faster.

So, for instances if we use the Jacobi method where we need to get this iteration matrix and the iteration matrix for Jacobi method is minus D inverse and then L plus U, L is the lower diagonal matrix, triangular matrix, U is the upper triangular matrix. So, our A in this case is 5, 1, 2, 1, 3 the coefficient matrix and then we have minus 1, 2 and 4 and the right hand side b vector is 3, 12, 8.

So, having this matrix A now the diagonal, the D will be having only 5, 3 and 4 in the diagonal entries rest everything 0, so that means and that to inverse, so for the diagonal matrix getting inverse is very easy we have to just inward the diagonal entries. So, instead of 5 we have 1 by 5, then we have 1 by 3 and then we have here 1 by 4.

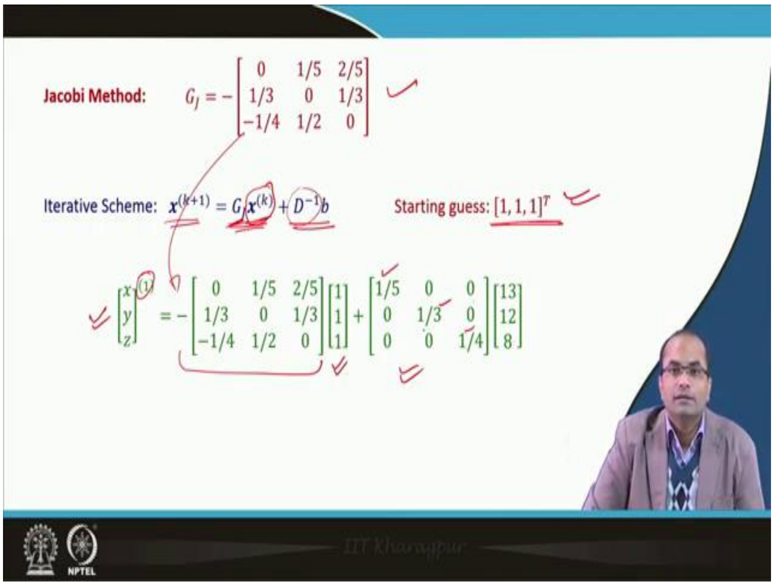
So, this was the diagonal entries were (4, 3, 4) 5, 3, 4 so we got this inverse 1 by 5, 1 by 3 and 1 by 4 with this minus sign and L plus U, so L is the lower triangular, U is the upper triangular, when we are adding this basically it is nothing but A minus D. So, L plus U is nothing but matrix A minus we remove the diagonal entries. So, removing the diagonal entries, so we have set here 0, 0, 0 and the rest everything same. So, 1, 2, 1 here we have 1 minus 1 and 2. So, the rest everything is same.

(Refer Slide Time: 24:51)




So, we got this minus D inverse and this L plus U we can multiply these two matrices and we will get a new matrix here which is our iteration matrix.

(Refer Slide Time: 25:10)

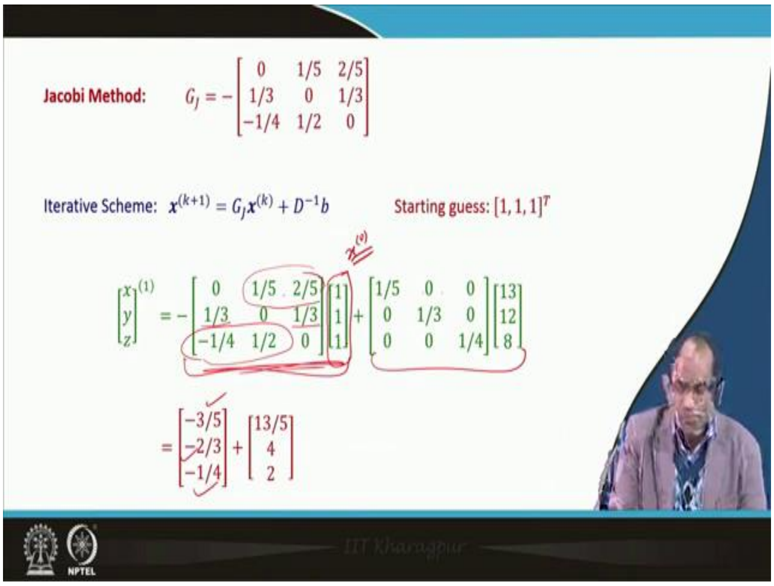


So, first we need to compute the iteration matrix and then having this iteration matrix for the Jacobi method the iteration scheme is that xk plus 1 is equal to Gxk plus this D inverse b this was the iteration scheme written in the matrix form.

So, that means if we take the starting values now this that is this xk replaced by this 1, 1, 1, the benefit now here in the in this matrix form that after just matrix vector multiplication, we will get the all components the value of all components, we do not have to compute 1 by 1. So, here for example we have the x, y, z these are the 3 components after this first iteration, how do we get Gj that is the iteration matrix which we have just evaluated, so this is Gj.

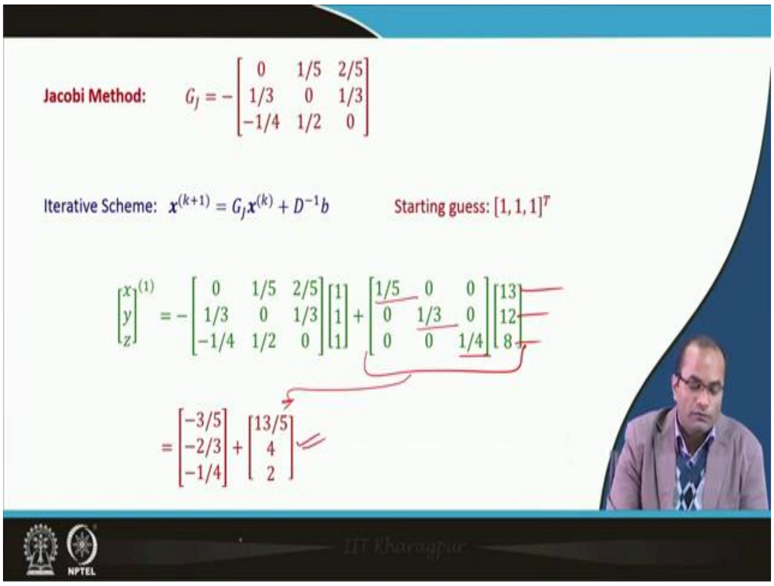
Xk that is now the starting values we want to use so x0, so we have this 1, 1, 1 plus we have here again this D inverse, D inverse means the diagonal entries will be just inverted, so 1 by 5, 1 by 3 and 1 by 4. And then we have 13, 12 and 8, so that is the right hand side b of our system of our linear system.

(Refer Slide Time: 26:30)



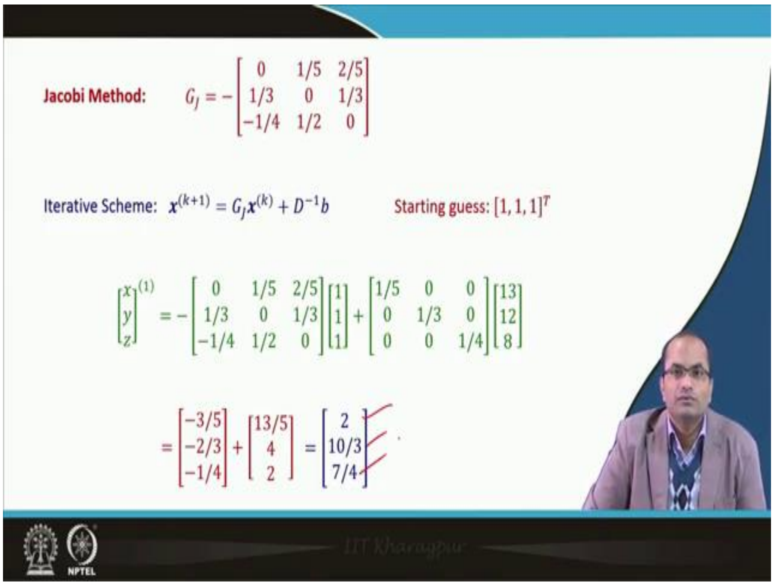

So, having this now we have substituted these x naught here, so x naught the starting values. And then we have to just to perform this multiplication, this multiplication and then we have to add it. So, this multiplication here for this matrix and this vector, it is just the addition of these rows so 1 by 5 plus 2 by 5 will minus 3 by 5, then here also 1 by 3 and then 1 by 3 that will be 2 by 3 with the minus sign, here also we can add. So, 2 by 4 and then minus 1, so 1 by 4 with the negative sign.

(Refer Slide Time: 27:05)



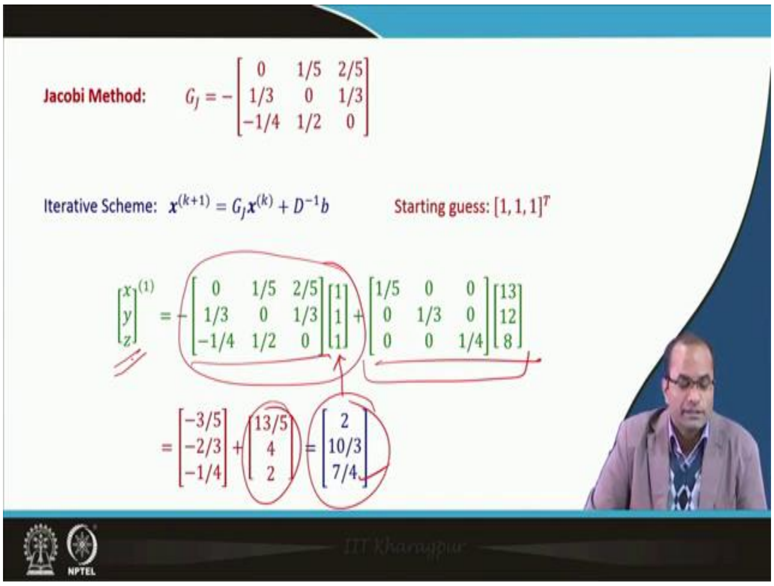

Similarly, here this is just 13 by 5, 12 by 5 and 8 by 4 and 12 by 3 and then 13 by 5. So, this is the vector coming from this product.

(Refer Slide Time: 27:23)



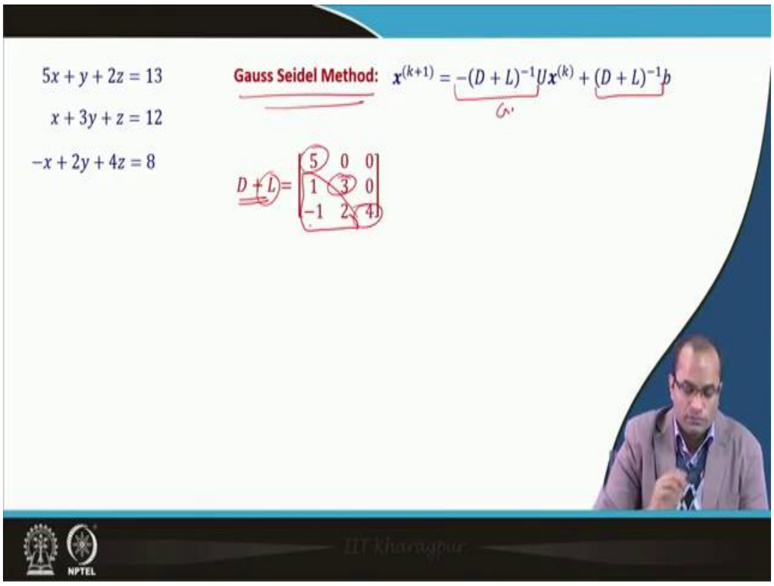

And now we can add the 2 and we got 2, 10 by 3, 7 by 4 and these are exactly the values which we got earlier also using the component wise a calculation.

(Refer Slide Time: 27:33)



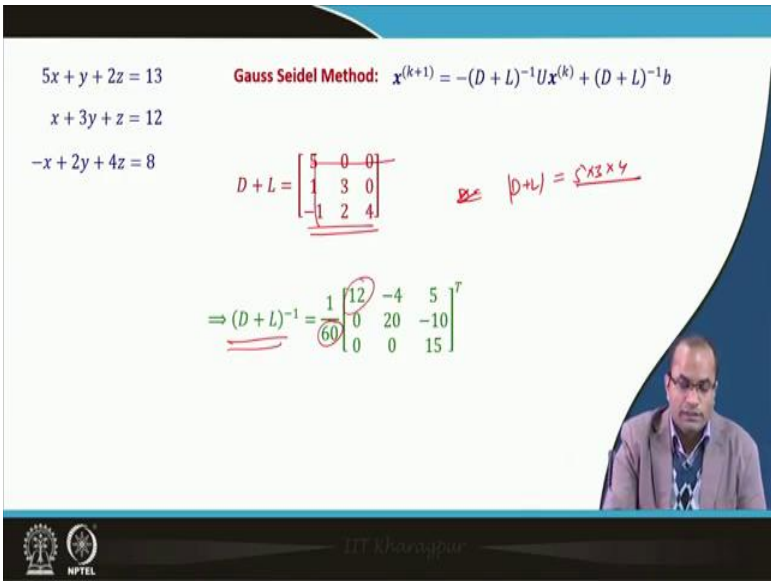

Here the benefit is once we have this one, we just need to plug this instead of this 1, 1, 1 and then again these multiplications we will get the next value. Indeed in this D inverse b there will be no change, so this vector we can keep as it is, the only change would be in the first multiplication, because this 1, 1, 1 will be replaced by 2, 10 by 3 and 7 by 4 and then we can get the next update. And so on we can proceed with this algorithm to get values, our values updated after each iteration.

(Refer Slide Time: 28:09)



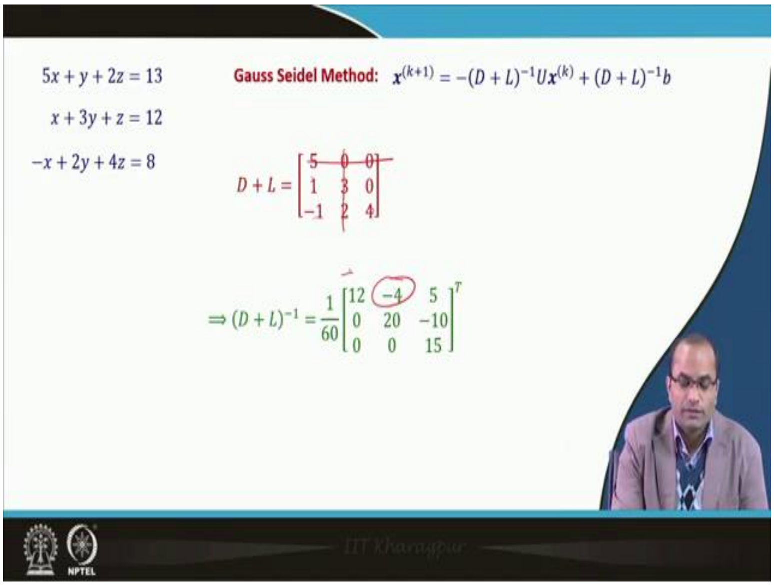

Coming to the Gauss-Seidel method we have this matrix form minus D plus L inverse U and D plus L inverse, this is the iteration matrix in this case. So, D plus L now so basically the lower triangular, so this is the lower triangular part and plus this diagonal also, also with the diagonal entries. So, this is D plus L.

(Refer Slide Time: 28:34)




And then we can get this D plus L inverse, so here we have (())(28:40) in the 1 over determinant, so 5 3 into so the determinant will be or the determinant of this D plus L will be 5 into 3 into 4, so that is 60. And then this is the adjoint of this matrix which can easily evaluated for instance this 12 is coming from this evaluation 4 into 3 and minus 2 into 0 that is 12.

(Refer Slide Time: 29:07)



$5x + y + 2z = 13$

$x + 3y + z = 12$

$-x + 2y + 4z = 8$

**Gauss Seidel Method:** $\boldsymbol{x}^{(k+1)} = -(D+L)^{-1}U\boldsymbol{x}^{(k)} + (D+L)^{-1}b$

$$D + L = \begin{bmatrix} 5 & 0 & 0 \\ 1 & 3 & 0 \\ -1 & 2 & 4 \end{bmatrix}$$

$$\Rightarrow (D+L)^{-1} = \frac{1}{60}\begin{bmatrix} 12 & -4 & 5 \\ 0 & 20 & -10 \\ 0 & 0 & 15 \end{bmatrix}^T$$



$5x + y + 2z = 13$

$x + 3y + z = 12$

$-x + 2y + 4z = 8$

**Gauss Seidel Method:** $\boldsymbol{x}^{(k+1)} = -(D+L)^{-1}U\boldsymbol{x}^{(k)} + (D+L)^{-1}b$

$$D + L = \begin{bmatrix} 5 & 0 & 0 \\ 1 & 3 & 0 \\ -1 & 2 & 4 \end{bmatrix}$$

$$\Rightarrow (D+L)^{-1} = \frac{1}{60}\begin{bmatrix} 12 & -4 & 5 \\ 0 & 20 & -10 \\ 0 & 0 & 15 \end{bmatrix}^T$$

NPTEL IIT Kharagpur

$5x + y + 2z = 13$

$x + 3y + z = 12$

$-x + 2y + 4z = 8$

**Gauss Seidel Method:** $\boldsymbol{x}^{(k+1)} = -(D+L)^{-1}U\boldsymbol{x}^{(k)} + (D+L)^{-1}b$

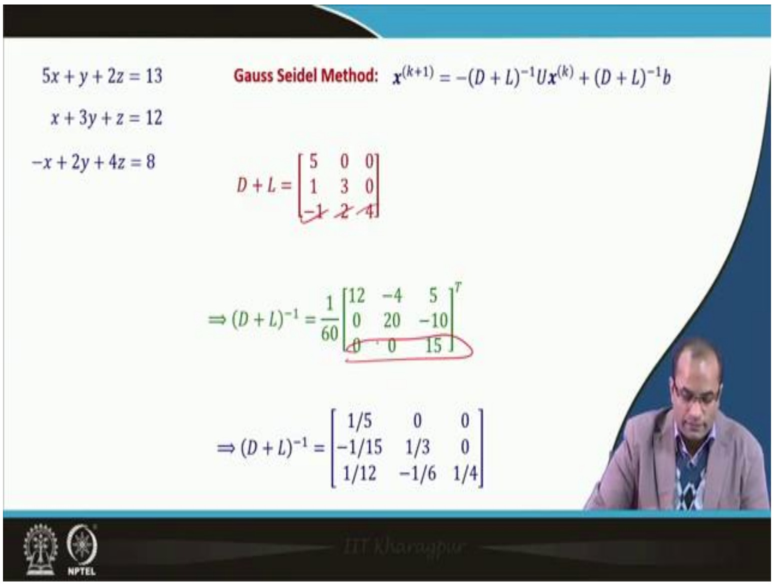
$$D + L = \begin{bmatrix} 5 & 0 & 0 \\ 1 & 3 & 0 \\ -1 & 2 & 4 \end{bmatrix}$$

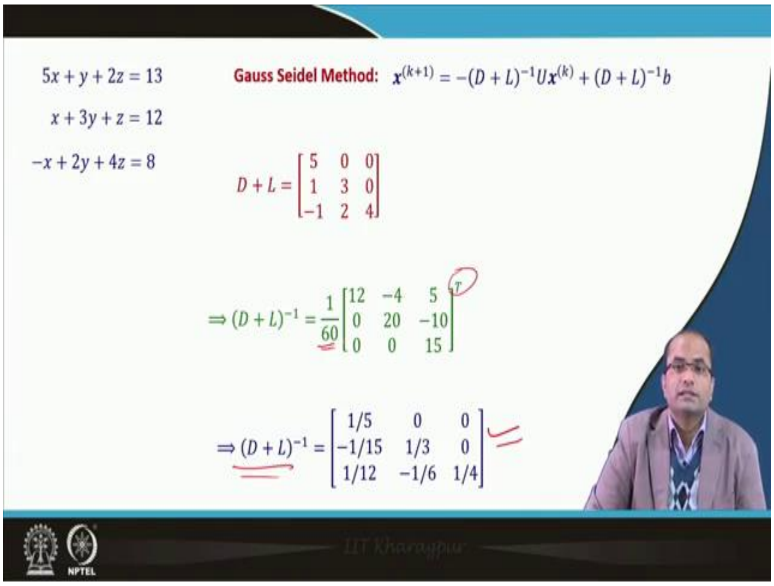
$$\Rightarrow (D+L)^{-1} = \frac{1}{60}\begin{bmatrix} 12 & -4 & 5 \\ 0 & 20 & -10 \\ 0 & 0 & 15 \end{bmatrix}^T$$

NPTEL IIT Kharagpur

And then for instance this one here will be coming as 4 minus 1 0 and then the negative sign, because we have to consider plus minus plus minus and so on. And for the third one we have 2 and plus this 3 that is the 5 there and then for instance for this one we have 0, so this is 0, for the next one when we compute for this we have 20 this exactly there then this one we will compute 10 with minus sign.

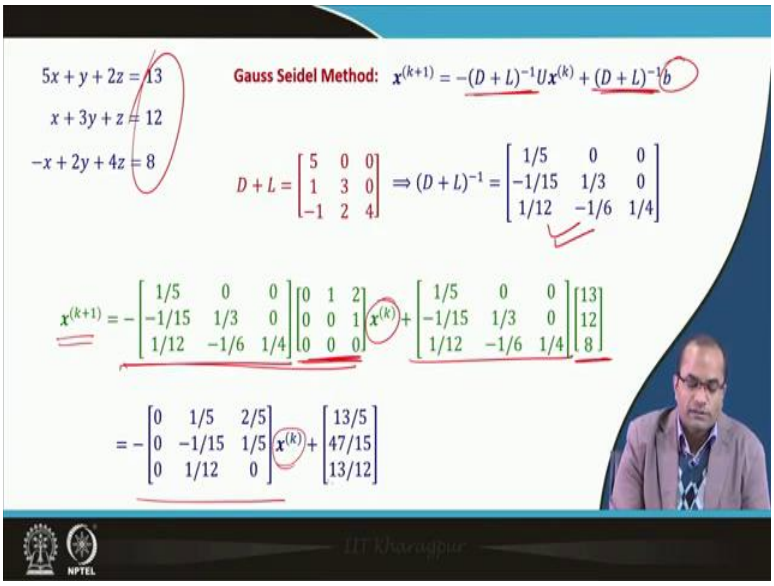
(Refer Slide Time: 29:45)



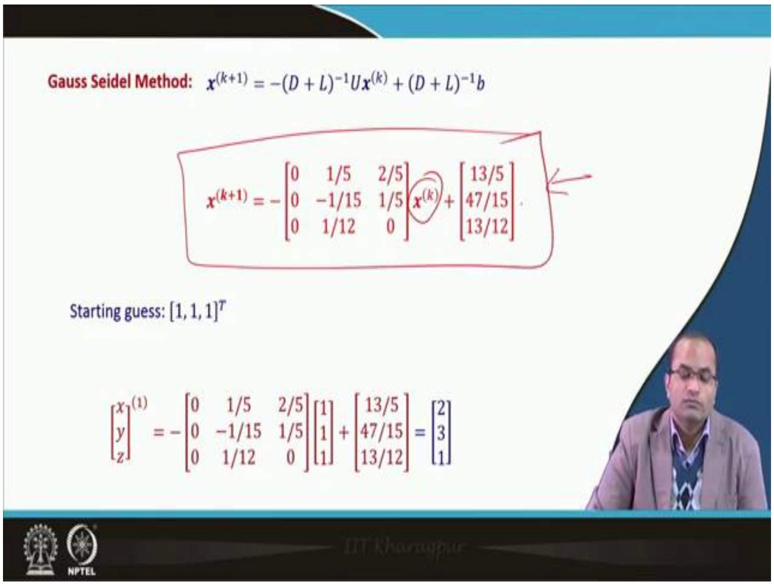

And then similarly we can compute for this and this, so we will get this adjoint matrix, the transpose has to be done. So, the D plus L inverse is just after taking transpose of this and dividing by 60 we are getting this D plus L inverse.

(Refer Slide Time: 30:02)



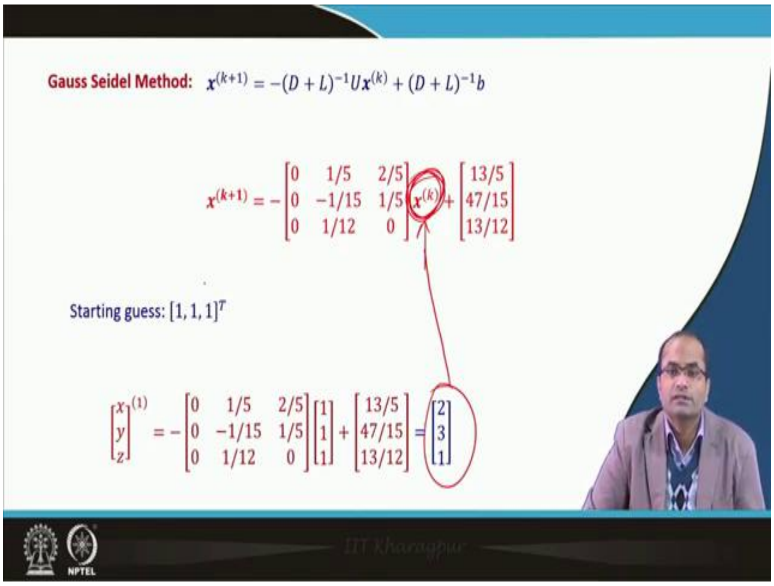

Having this D plus L inverse we can plug this into this matrix form, so we have D plus L inverse, we have this upper triangular matrix, we have D plus L inverse and then we have this b from the equations the 13, 12 and 8. And then we need to just multiply this one x0 we can take 1, 1, 1 and so we have this iteration scheme that xk plus 1 is equal to this matrix, multiplication with this vector plus this vector.

(Refer Slide Time: 30:33)



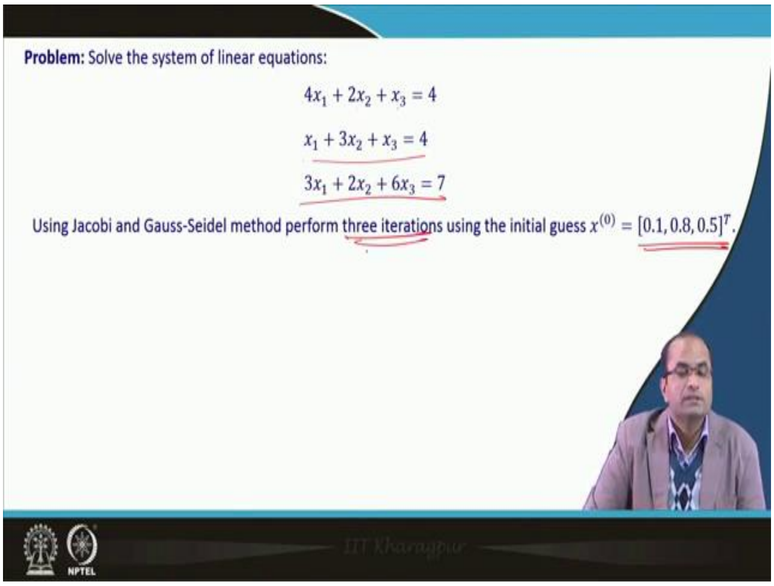
And if we plug this initial guess here for instance 1, 1, 1 and do this evaluation we will get 2, 3, 1. So, the advantage here in this matrix form is that we have to just do this once to get this form we have to do some calculations like inverse and so on multiplication.

(Refer Slide Time: 31:00)



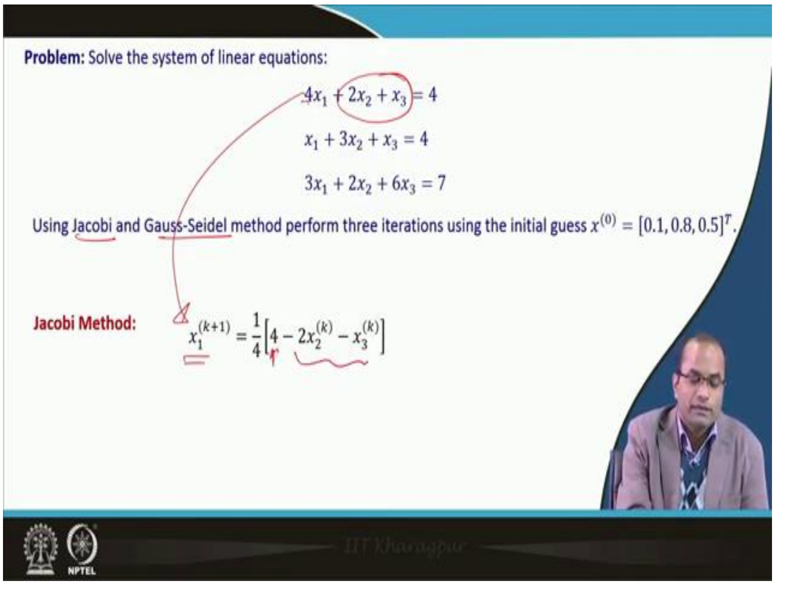

But once we have this form here we just need to plug the initial value here and then for instance this is the value after first iteration. Again we will use this there we will get the new value and so on. And iterations will be very fast.

(Refer Slide Time: 31:11)



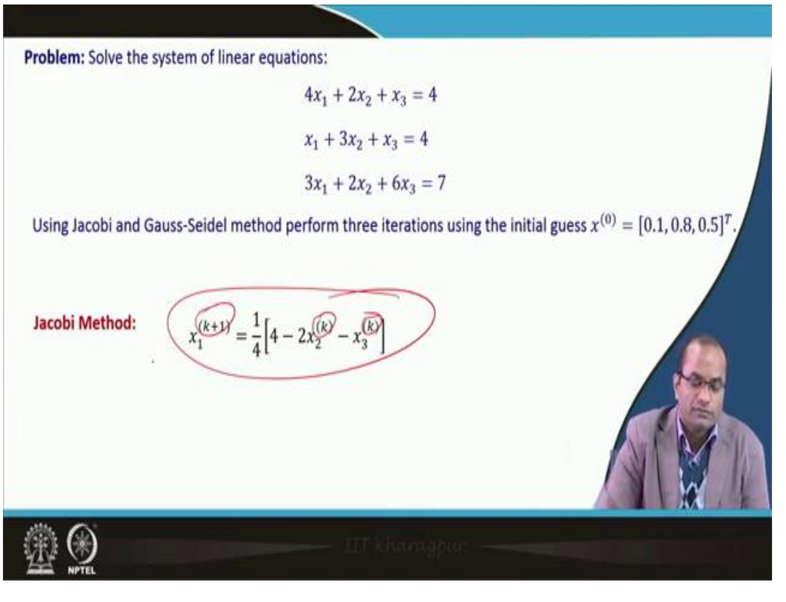
The last problem here we want to solve this one 4x1, 2x2 x3 and then we have the second equation we have the third linear equation. And we want to use for instance the initial guess 0.1, 0.8 and 0.5. And we want to perform three iterations using this initial guess.

(Refer Slide Time: 31:35)



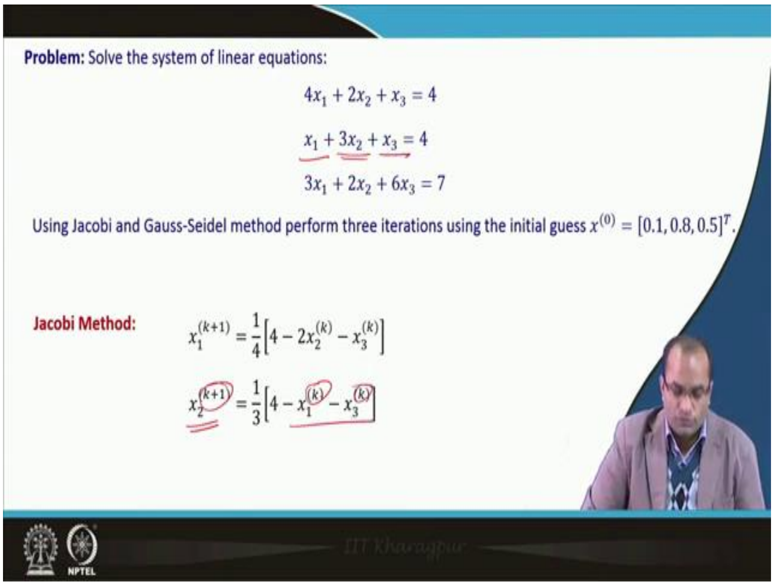

And we will use the Jacobian Gauss-Seidel method to do this. So, the Jacobi method as usual we discussed before that from the first equation we will write or we will setup the iteration for x1 taking this 2x2 plus this x3 to the right hand side and this 4 was already there and then we need by 4.

(Refer Slide Time: 31:59)



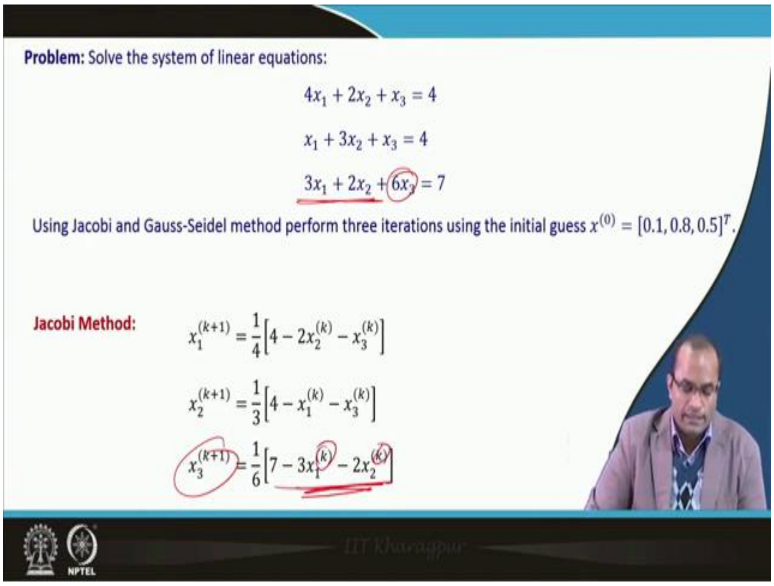

So, we have this iteration for x1, putting this k there and k plus 1 to the left hand side.

(Refer Slide Time: 32:05)



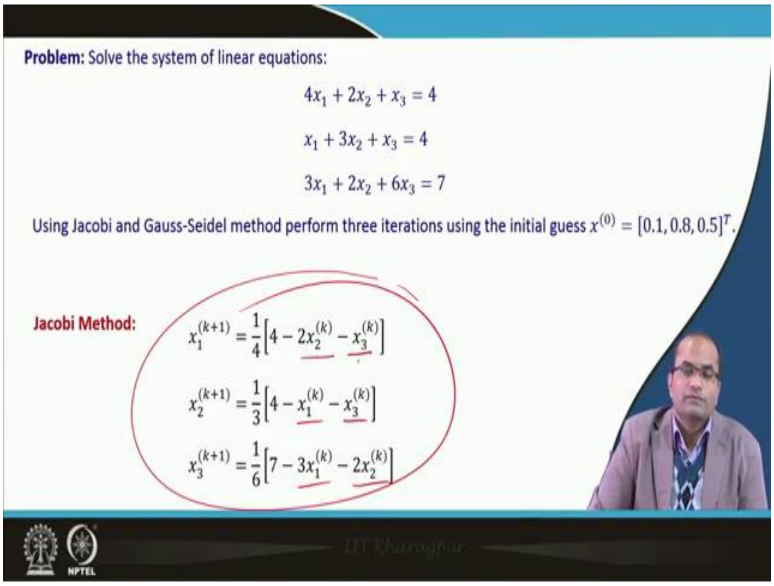

Similarly, from the second equation we can perform for x2 after this k plus 1th iteration taking x1 and x3 to the right hand side putting them at kth iteration.

(Refer Slide Time: 32:20)



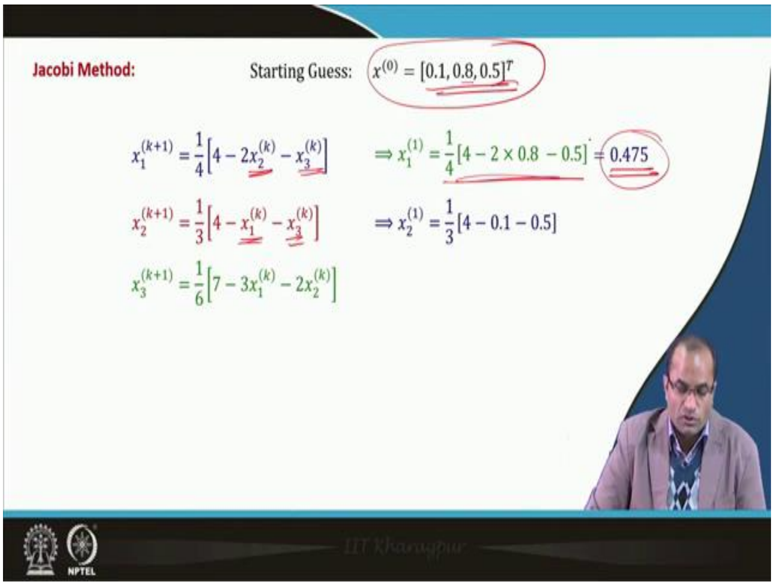

From the third equation we will get this x3 and this 3x1 and 2x2 will go to the right hand side with this kth iteration values and so on.

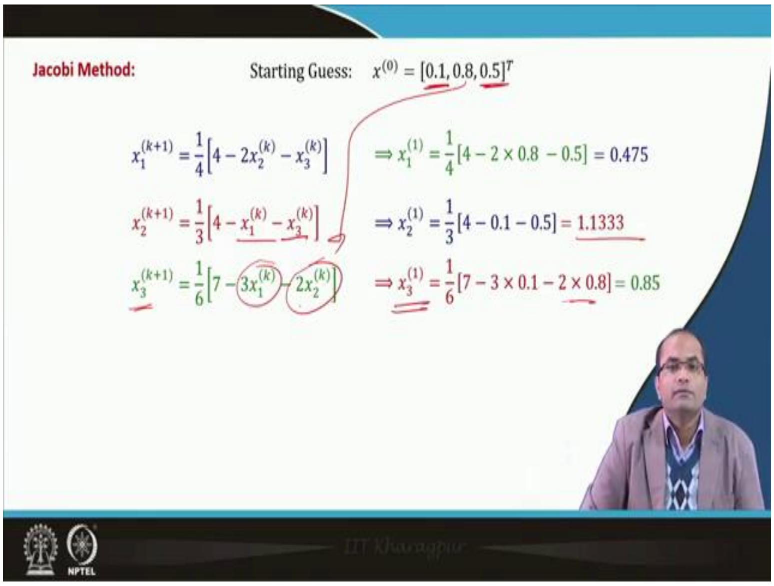
(Refer Slide Time: 32:33)




So, this is the iterative method the Jacobi iteration method we will plug these values the initial values here by putting k equal to 0 and we will get the values after first iteration. Again we can move to the next iteration by putting by using these values after first iteration to get the values after the second iteration.

(Refer Slide Time: 32:53)

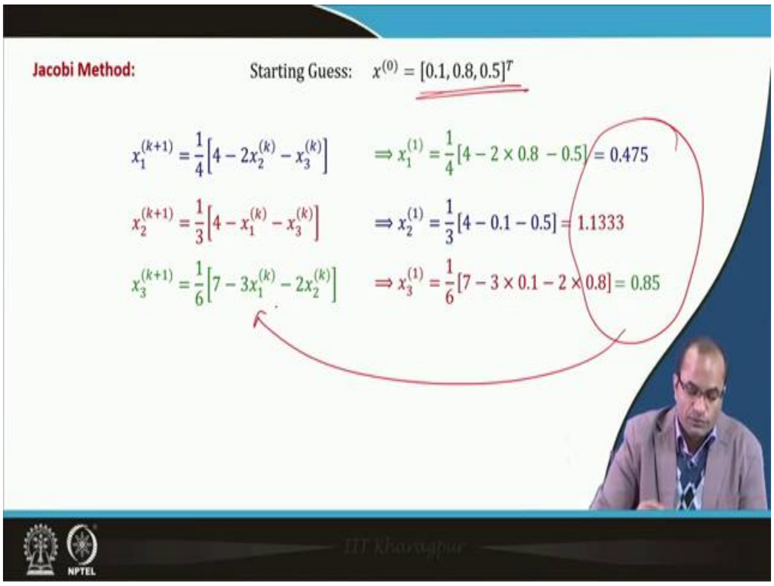





So, having this iterative scheme, if we start with this initial guess, so x1, so we will put here for x2 like 0.8 here we will put 0.5 and do this calculation we will get 0.475. For x2 we will plug here x1 and x3 from the all values will be used from this x0 until we get this complete step or we compute all the components in this iteration. While in the Gauss-Seidel we will use recently evaluated values as well.

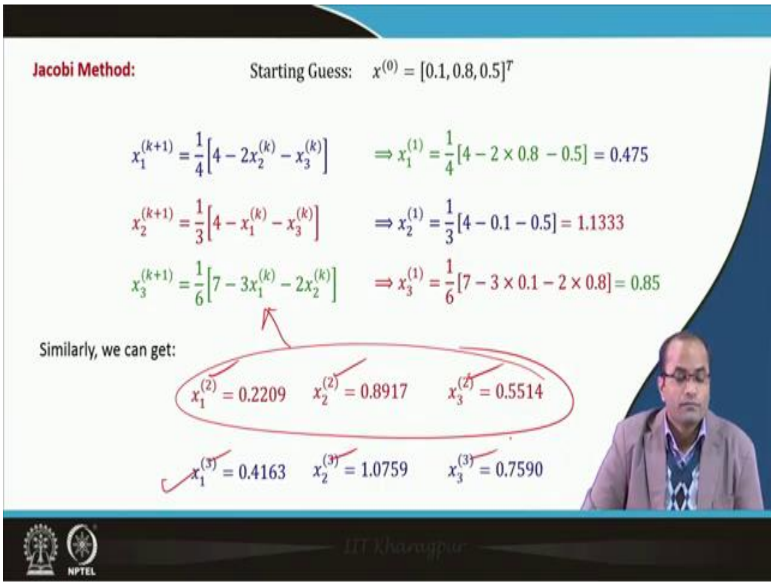
So, this is the Jacobi one, we will use again x1 and x3 from this initial guess only, that is 1 and this 0.5. So, this value is 1.133 and then to compute this x3 also we will use the values from the initial guess itself. So, x1 was 0.1 and then x2 was 0.5 there, 0.8 there and then this will give 0.85.

(Refer Slide Time: 33:59)



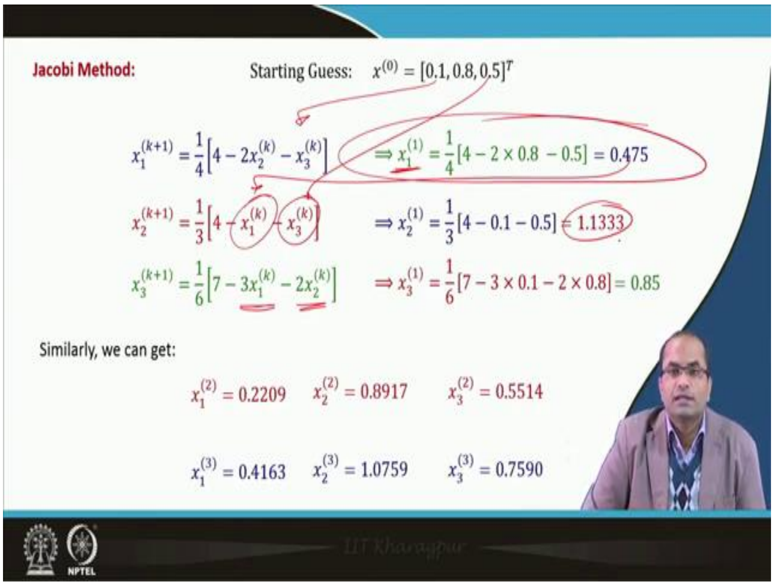
And now in the next iteration these values will be used instead of these 0.8, 0.8, 0.5.

(Refer Slide Time: 34:10)



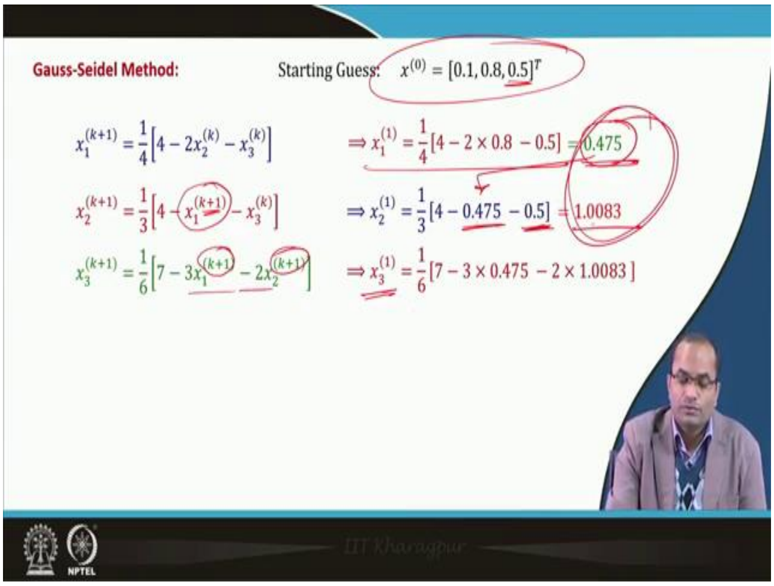

So, if we use these values here we will get the next values of x1 after second iteration, x2 after second iteration, x3 after second iteration and then if we use these values in this scheme again then we will get x1 after third iteration, x2 after third iteration and then x3 after third iteration.

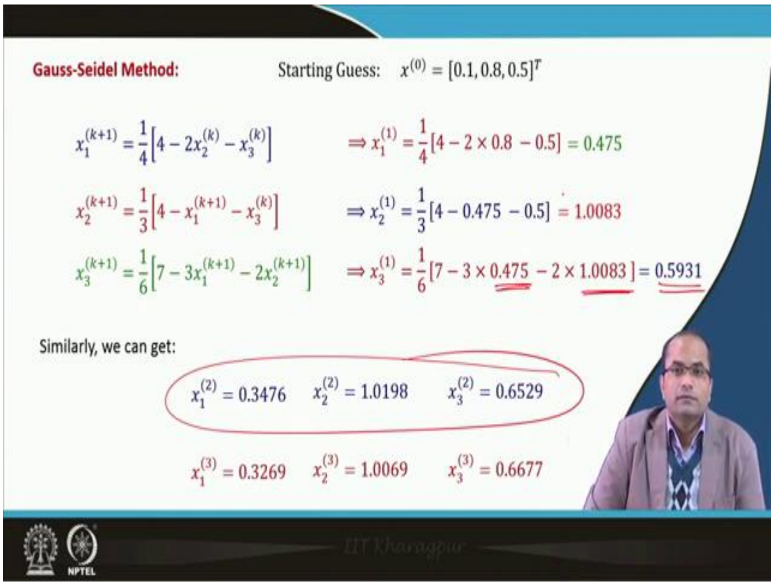
(Refer Slide Time: 34:34)



Well, so this was a Jacobi method, the only difference in the Gauss-Seidel will come, that the first evaluation will be the same, because we will be using exactly the initial guess, while in the second one x1 this will be used, the recently computed values because now x1 is available, x3 is not available, so we will use still from the previous one and we will get some different value there. Similarly, in the third component x1 and x2 both are available now and we will be those recently evaluated values.

(Refer Slide Time: 35:02)



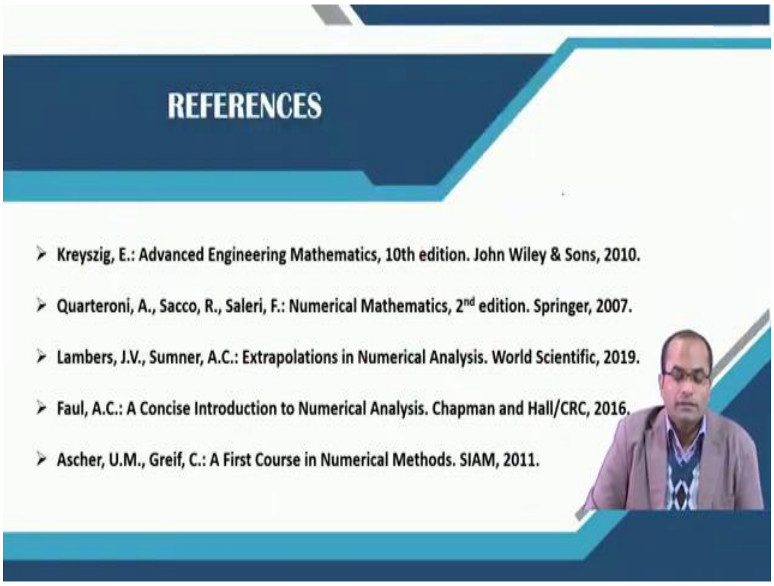




So, coming to the Gauss-Seidel that is the only difference that this x1 will be here now k plus 1 and this x1 and x2 both will be using the k plus 1th values. So, starting with this guess the same evaluation as in the Jacobi 0.475. And then the next one there will be change here that x1 now will use this value instead of the value form the $0^{th}$ iteration or the initial guess.

And this 0.5 naturally will come from this initial guess, because it is not available yet. And the value is 1.10083, the x3, x1 and x2 these both the values will be used now from this iteration which we have already evaluated, so like 0.475 and this 1.10083 will be used now. And we will get a new value 0.5391 and similarly, we can proceed for the second iteration, so all these values from the second iteration and then we will get the values after the third iteration.

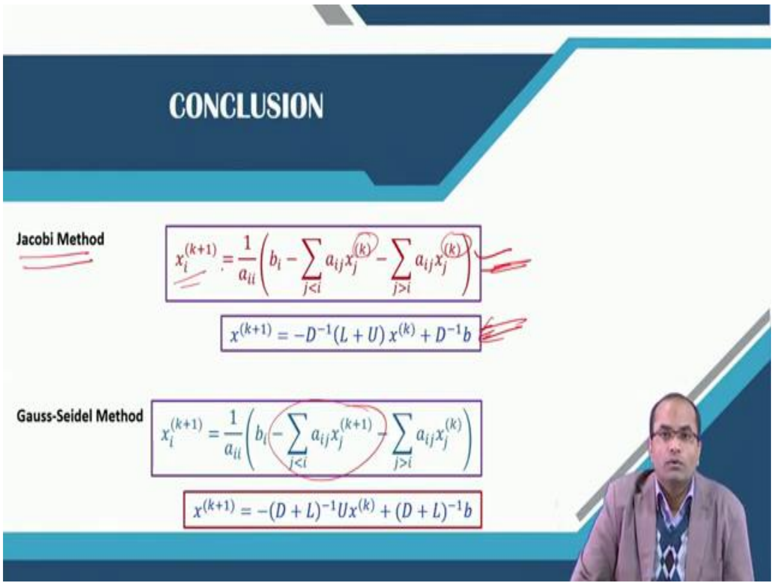
So, here there was a difference that whatever we have evaluated that value will be used in in the current iteration current evaluation. Well, so that was the evaluation part in the next lecture.

(Refer Slide Time: 36:28)




So, this are the references.

(Refer Slide Time: 36:31)




In the next lecture we will continue with this how to, I mean what is the convergence issues of these two methods which converges faster, etc. So, here just to conclude we have a evaluated or we have approximated the solution of a system of linear equation using the Jacobi method and also using the Gauss-Seidel method and we have seen that either we can use the matrix vector form or we can use the component form depending on the convenience and the major difference in both the methods is the evaluation of this term for instance so here we are using until the iteration is complete that particular iteration to get all the components we are using the values from the previous iteration whereas here we are using the recently evaluated values. So, that is all for this lecture and I thank you very much for your attention.