Numerical Analysis
Prof. Dr. S. Baskar
Department of Mathematics
Indian Institute of Technology-Bombay

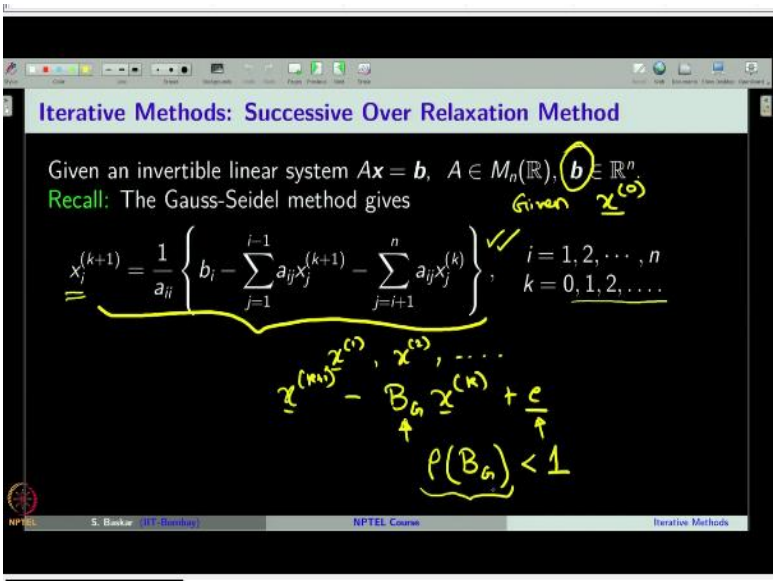
Lecture-21
Linear Systems: Iterative Methods (Relaxation Method)

Hi, we are learning iterative methods for linear systems. In this, we have introduced two main iterative methods namely Jacobi method and Gauss-Seidel method. We have also learned the convergence theorems for these methods. In fact in the last class, we have proved the necessary and sufficient condition for the convergence of general iterative method written in a particular form and this condition depends on the spectral radius of the iterative matrix of the method.

From this theorem we learned that if the spectral radius is very close to 1 but less than 1 then the corresponding iterative method may converge rather very slowly. We have also illustrated these phenomena in Jacobi method and in fact this can happen with Gauss-Seidel method also. In this lecture we will learn a slight modification of an iterative method through a relaxation technique.

We will do this with Gauss-Seidel method and see how to accelerate the convergence speed of the Gauss-Seidel method.

(Refer Slide Time: 01:43)



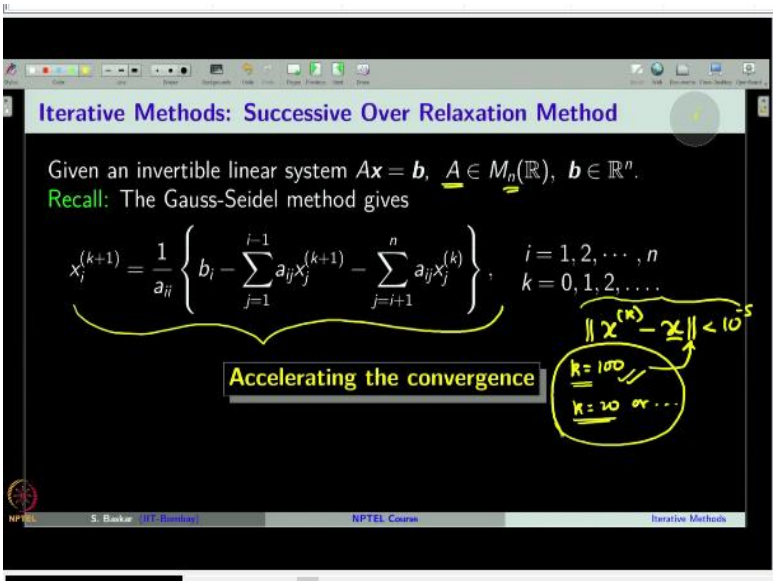
Assume that we are given a linear system $Ax = b$, where A is an $n \times n$, real matrix which is invertible and b is a given n-dimensional vector. Recall, that we can write the Gauss-Seidel

iterative procedure to approximate the solution of the given linear system $Ax = b$ and component wise the Gauss-Seidel formula is given like this, where x_i is the *i*th component of the iterative vector $x^{(k+1)}$.

And *k* runs from 0 to infinity. For *k* equal to 0, we are given the initial condition $x^{(0)}$ and from there for each *k* = 0, 1, 2 and so on we can get $x^{(1)}$ and then $x^{(2)}$ and so on whose components are computed using this formula and this is the Gauss-Seidel method. In fact, we can write this method in the form $x^{(k+1)} = B_G x^{(k)} + c$, where B_G is the iterative matrix for the Gauss-Seidel method and *c* is a vector which is derived from the right-hand side vector i.

We have seen that the Gauss-Seidel method will converge if and only if the spectral radius of the matrix B_G is strictly less than 1. If the spectral radius is very close to 1 then the convergence will be very slow in the sense that the sequence may take many terms to achieve the required accuracy of the solution.

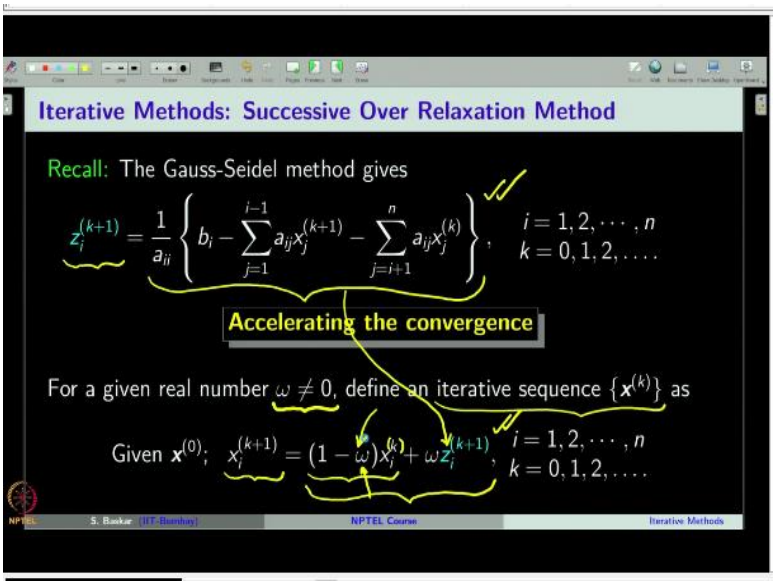
(Refer Slide Time: 04:03)



In such slow convergence situations, we are interested in looking for some way to accelerate the convergence. Say for instance, suppose we want the accuracy say our computed solution should be very close to the exact solution. In such a way that this error is less than say 10^{-5} and suppose it takes 100 iterations that is *k* = 100 to achieve this accuracy. Then it may be computationally very costly especially when the matrix is very large in its dimension.

That is *n* is a very large integer. In such cases we look forward to have some way to accelerate this convergence. That is achieve this accuracy but for some lesser value of *k*. Say for instance *k* = 20 or something like that then that will be very nice when compared to achieving the accuracy with large *k*. Now the question is; is it possible to somehow modify your original iteration in such a way that you can accelerate the convergence. I hope you understood what I meant by accelerating the convergence.

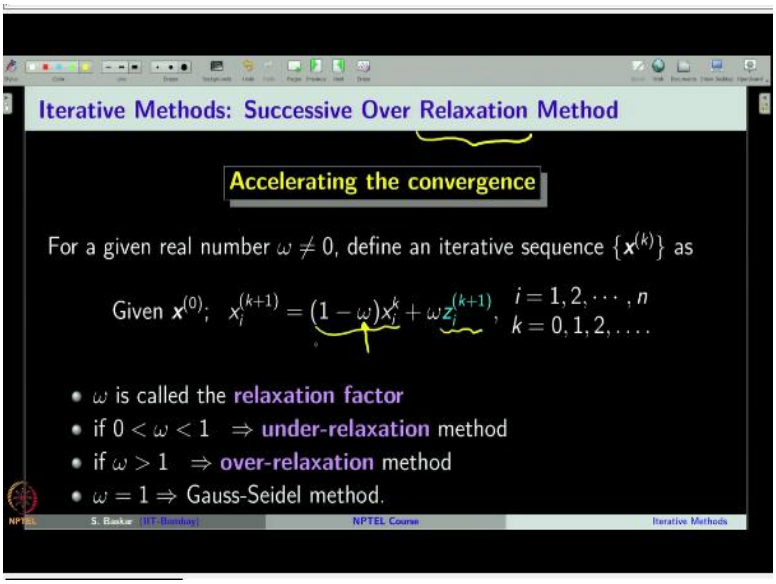
(Refer Slide Time: 05:31)



So, that is a nice idea what you do is just denote the Gauss-Seidel formula by *z* instead of *x* and now for a given non-zero real number omega we will now define our iterative sequence $x^{(k)}$ component wise as $x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega$ into this expression. So, this is just a slight modification of the Gauss-Seidel method. We are introducing a parameter ω here and writing the iterative formula in this form.

Now the question is what is the advantage of rewriting our original iterative formula in this form by introducing a parameter here?

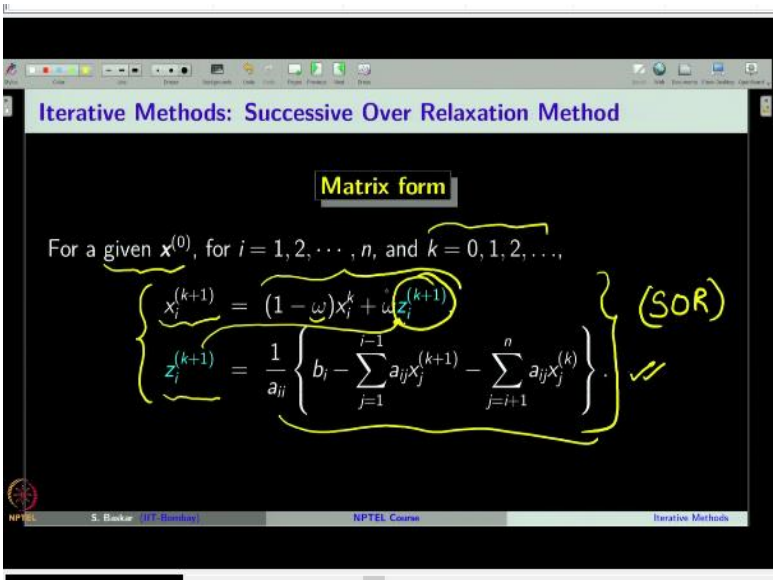
(Refer Slide Time: 06:45)



Before answering this question let us just name this parameter ω. In literature ω is called the relaxation factor. If ω is less than ,1 generally we call it as under relaxation and when ω is greater than 1, we call the method as the over-relaxation. You can observe that when ω = 1 then this term goes off and you will have $x_i^{(k+1)}$ is just equal to $z_i^{(k+1)}$ and $z_i^{(k+1)}$ is precisely the Gauss-Seidel formula.

Therefore, when ω = 1 our method just reduces to Gauss-Seidel method and in general we call this method as Successive Over Relaxation method. So, here the troublesome part is when the convergence is rather slow, so our purpose of introducing ω into this formula is to make the convergence little faster. In that way generally we call ω as a relaxation factor.

(Refer Slide Time: 08:13)

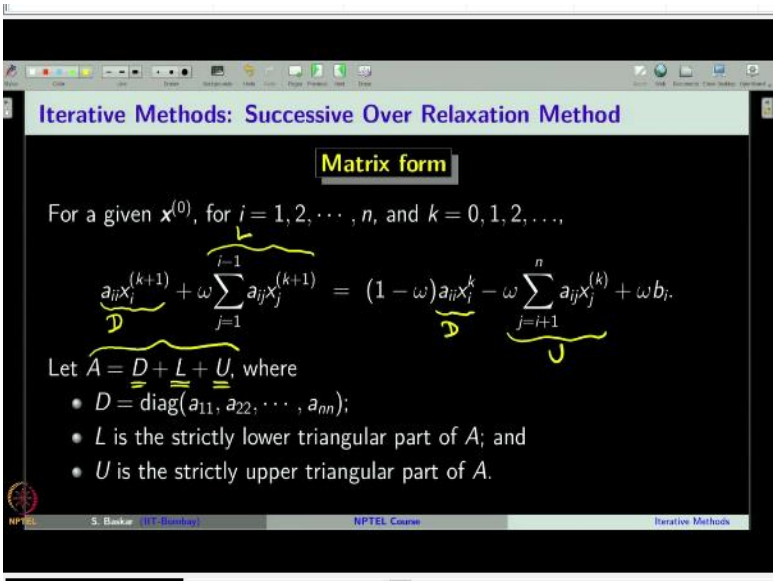


So, what we have done so far is, we have defined an iterative sequence in this way which is just an extrapolation of the Gauss-Seidel method in this way and it is called the Successive Over Relaxation method. Here we have to choose a ω and based on that ω we define the successive over-relaxation iteration process. In short, we generally call it as SOR. Therefore, for a given initial guess $x^{(0)}$ for every k first you will find $z^{(k+1)}$.

Plug in that here and then get $x^{(k+1)}$. Once you have $x^{(k+1)}$ then again you will repeat this process and in that way, you can generate a iterative sequence $x^{(k)}$. Now our interest is to write this formula in the matrix notation. For that first what we will do is we will substitute this expression into the first equation. And get this equation for $x^{(k+1)}$. Now let us rewrite this equation.

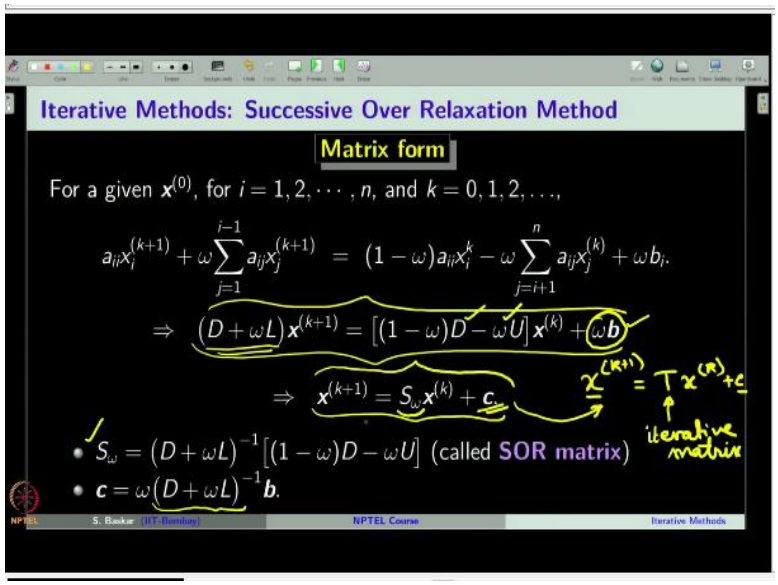
For this we will first multiply both sides by a_{ii} and then take this term to the left hand side and write this expression in this form. Now you see it is more or less very clear. From here how to write this equation in the matrix notation because this is the formula for computing one particular component of the vector $x^{(k+1)}$.

(Refer Slide Time: 10:19)



So, in order to write it in the matrix form first let us write A as the diagonal path of A plus the strictly lower triangular part of A plus strictly upper triangular part of A. Then you can see that this term is having the diagonal part, this term is having the strictly lower triangular path and you can see again this has the diagonal path and this has the strictly upper triangular path. Once you understand this it is more easy for us to write this expression in the matrix notation.

(Refer Slide Time: 11:07)

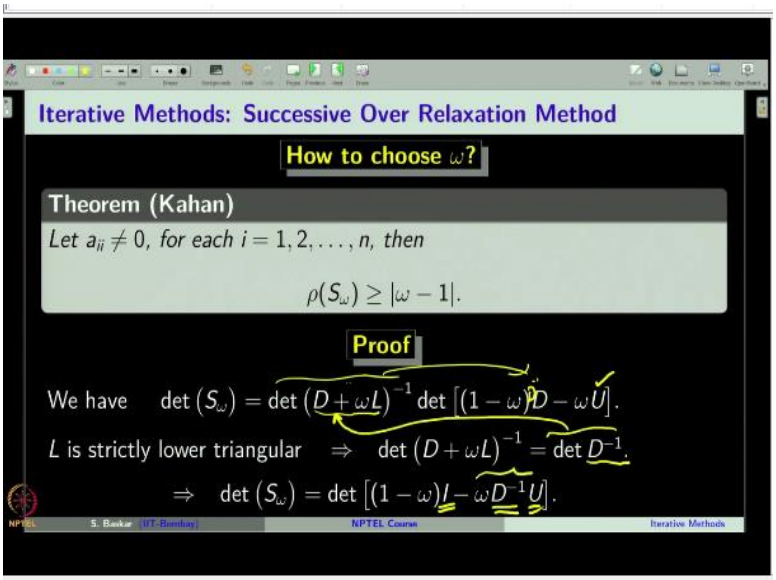


You have *D*, *L*. Again, you have *D* here and *U*. Therefore, you can combine these two and write $D + \omega L$ and now we are making everything as vector. Therefore, this is vector $x^{(k+1)} = [(1-\omega)D - \omega U]x^{(k)} + \omega b$. So, that is the vector form of our successive over-relaxation method. You can now multiply both sides by $(D + \omega L)^{-1}$.

And you can see that this equation now can be written in this form. If you recall in the last class, we have proved the necessary and sufficient condition for the convergence of any iterative method that can be written in the form $x^{(k+1)} = Tx^{(k)} + c$, where T is the general form of an iterative matrix. Now you can see that the SOR method is also written in this form with the iterative matrix as S_ω which is precisely $D^{-1} + L$ because we are now multiplying both sides of this equation by $(D + \omega L)^{-1}$.

Therefore, $S_\omega = (D + \omega L)^{-1}[(1-\omega)D - \omega U]$. That is what is the expression for the iterative matrix S_ω in the SOR method and the vector *c* is given by ω which is already there in this term into you are multiplying $(D + \omega L)^{-1}$. Therefore, that will come here into *b* which is already there in the term. So, this is how we can write the SOR iterative formula in the matrix notation.

(Refer Slide Time: 13:55)



Next our interest is to see how to choose the relaxation parameter ω, so that the convergence of the sequence is accelerated. The idea is to look for the spectral radius of the iterative matrix S_ω and try to make it as small as possible. That is the idea. The first theorem towards this is by Kahan.

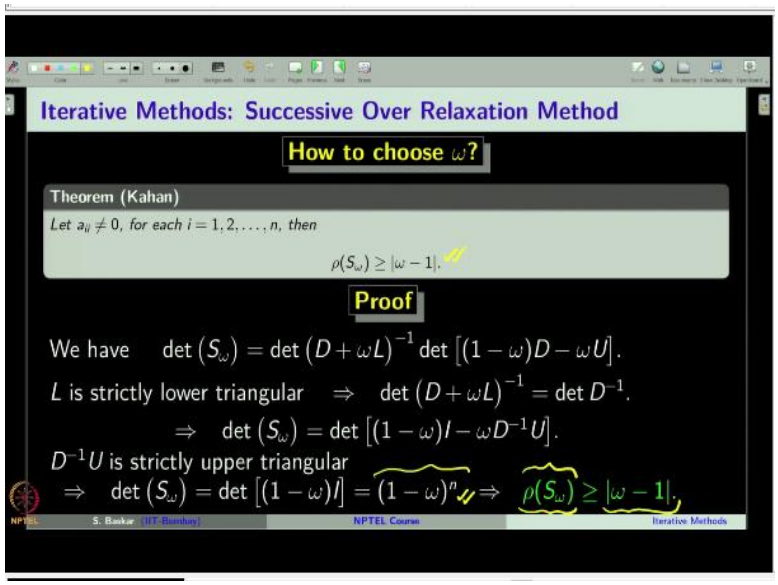
The theorem says that if you are working with the coefficient matrix *A* whose diagonal elements are non-zero. Then the spectral radius of the iterative matrix S_ω will be surely greater than or equal to $|\omega - 1|$. The proof of this theorem is very simple. So, let us prove it, go back to the expression of *S*, S_ω is given by the product of two matrices. This is one matrix and you have another matrix and S_ω is written as the product of these two matrices. Therefore, the determinant of S_ω can be determinant of this first matrix into the determinant of the other matrix.

So, this is a very simple property of determinant and now once you have this look at the determinant of $(D + \omega L)^{-1}$. Since L is strictly lower triangular you can see that determinant of $(D + \omega L)^{-1}$ equal to determinant of D^{-1}. Once you have this you can plug in this instead of this expression and then again use the property, the determinant of the product of two matrix is equal to the product of the determinants.

So, once you use that you can write the resulting equation as determinant of S_ω equal to determinant of when you take this inside you can see that it becomes 1 – ω. Remember now $(D + \omega L)^{-1}$ is nothing but D^{-1} when you take the determinant. So, when it goes inside you

will have $1 - \omega D^{-1}D$ that becomes identity matrix $-\omega D^{-1}$ that is coming here into U which is already there. Now let us try to understand how this will be when you are taking determinant.

(Refer Slide Time: 16:54)

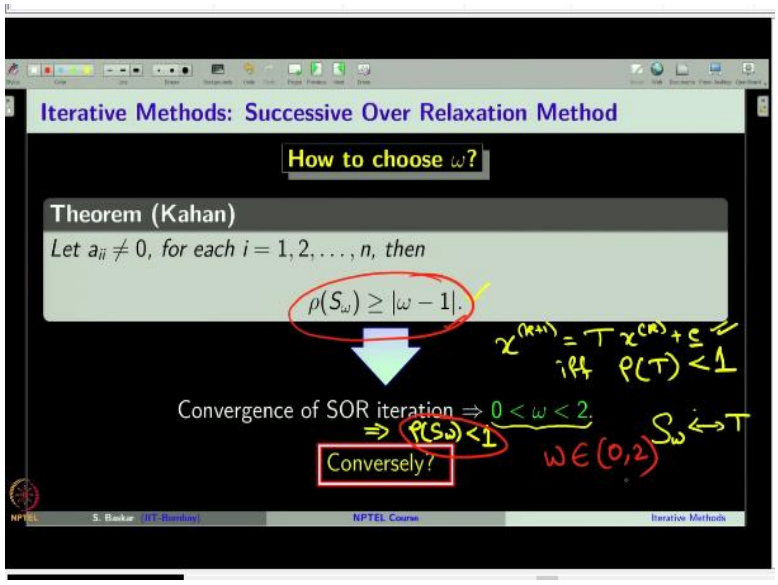


Again, you use the fact that $D^{-1}U$ is the strictly upper triangular matrix. Therefore, you can join these two and look at the determinant of that matrix and that will happen to be simply determinant of $1 - \omega$ into the identity matrix. Therefore, the determinant of S_ω which is given like this is finally reduced to this. Now since the determinant of a diagonal matrix is simply the product of all the diagonal elements, you can see that this is precisely equal to $(1 - \omega)^n$.

If you recall the definition of the spectral radius, you can see now that when determinant of S_ω is given like this which is also the product of all the eigenvalues of S_ω. You should keep this property of determinant also in mind and look at this you can immediately see that the spectral radius.

Again, you have to recall the definition of spectral radius and keep in mind the determinant of a matrix is equal to the product of the eigenvalues of that matrix. You use these two conditions together to see that the $\rho(S_\omega) \geq |1 - \omega|$. Therefore, this property is proved.

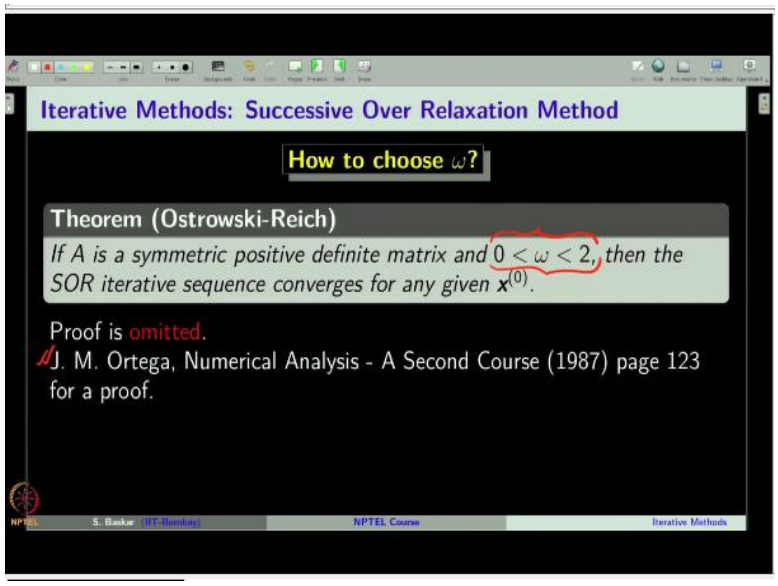
(Refer Slide Time: 18:49)



Now what this property tells us? This property tells us that if SOR is converging, it means you have chosen your omega in the interval 0 to 2. That is what it says, because from the theorem that we have proved in the last class if you recall any iterative method that can be written in the form $x^{(k+1)} = Tx^{(k)} + c$ will converge if and only if the spectral radius of this iterative matrix is strictly less than 1.

And we have written SOR in this form and we have used the notation S_ω instead of T. Therefore, SOR iteration is converging means $\rho(S_\omega)$ is less than 1, but theorem says that the spectral radius of S_ω is greater than or equal to $\omega - 1$. You combine these two conditions and you can see that convergence of SOR iteration implies ω belongs to the open interval 0 to 2. Now what about the converse of this statement?

(Refer Slide Time: 20:30)

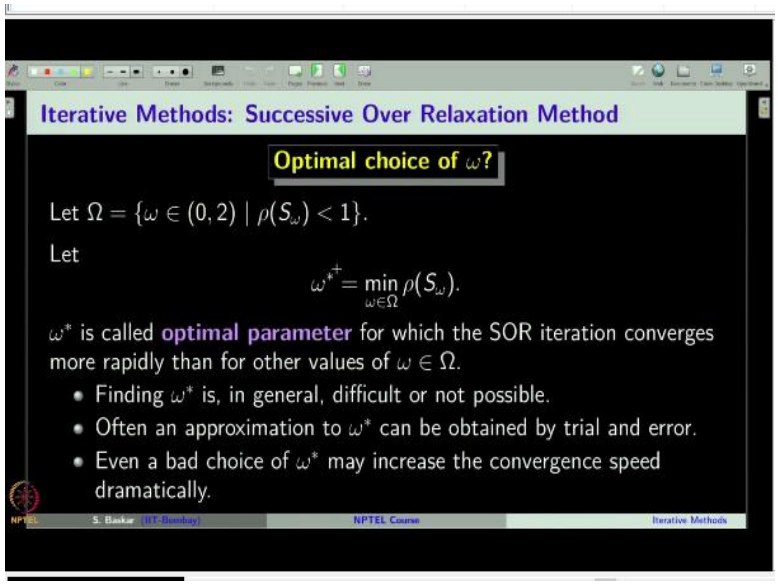


We cannot in general say anything about the converse of the statement. However, if A is a symmetric positive definite matrix and if ω belongs to the open interval $(0, 2)$ then you can say that the SOR iteration sequence will converge for any given initial guess $x^{(0)}$. The proof is little lengthy. Therefore, we will omit the proof. Interested students can learn the proof from this classical book.

From these two theorems what we understand is when we are working with SOR method it is always important that we should choose ω in this interval, whether it will lead to a convergence sequence or not that we do not know. But definitely you cannot go away from this interval and choose an ω and hope that your sequence will converge. That is not possible.

Rather you have to choose your ω in this interval but then whether the convergence will happen or not still you do not know it will depend on the spectral radius of S_ω for the chosen ω.

(Refer Slide Time: 22:08)

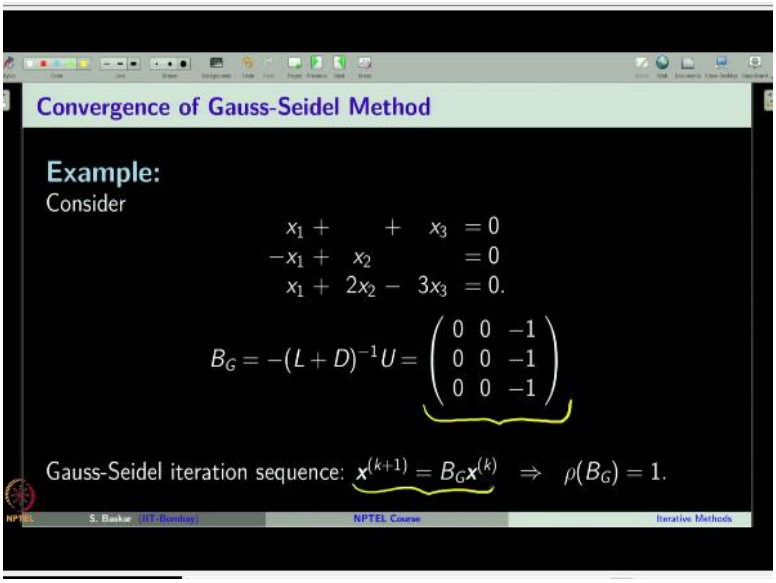


Let us now define what is called optimal choice of ω. For that let us first consider the set which we will denote as Ω and it is the set of all ωs in the open interval $(0, 2)$ such that the spectral radius of $S_\omega < 1$. Let us call this as the admissible set because for all such ωs in the Ω our corresponding SOR sequence will surely converge.

Now what you do is, you take the spectral radius S_ω for all ω in the admissible set Ω and take the minimum of all these numbers. We will denote it by ω^* and call it as optimal parameter for the SOR iterative sequence. Why it is so, because for this choice of ω we will achieve the fastest convergence. That is why this parameter is called optimal parameter.

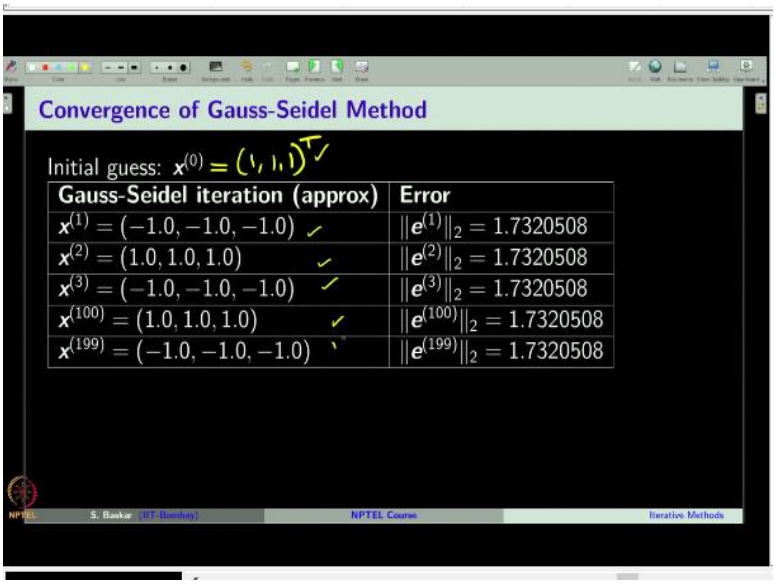
Now the question is for any given matrix A can we compute ω^*? It is in general very difficult to obtain ω^* but often one can just get this ω^* with few trial and error, it may be an approximation but sometimes it works dramatically well in terms of speed of convergence then the Gauss-Seidel method.

(Refer Slide Time: 24:03)



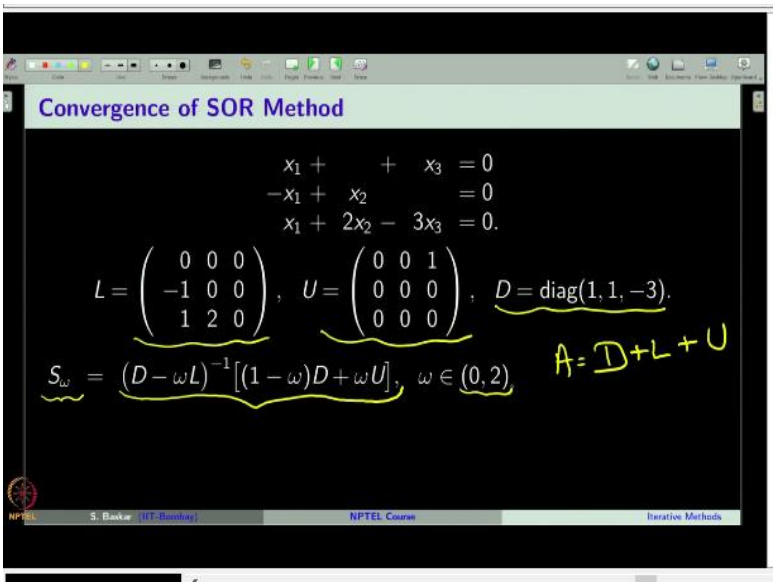
Let us take an example. Let us consider this system. If you recall we have already considered this system in our previous lectures. Let us take the Gauss-Seidel method. If you recall the iterative matrix for Gauss-Seidel method in this example is given like this and the Gauss-Seidel iterative sequence is given like this. You can immediately see that the spectral radius of the Gauss-Seidel iterative matrix is equal to 1. In fact, you can also directly see that this sequence will lead to an oscillating sequence.

(Refer Slide Time: 24:51)



Just to see this let us take the initial guess as (1, 1, 1) and apply the Gauss-Seidel iteration for this initial guess. You can see that the Gauss-Seidel iteration sequence will be an oscillating sequence and it will never converge. We have seen this behaviour in one of our previous lectures itself. Now let us see how we can choose ω and with that ω how the resulting SOR sequence will behave for this particular example.

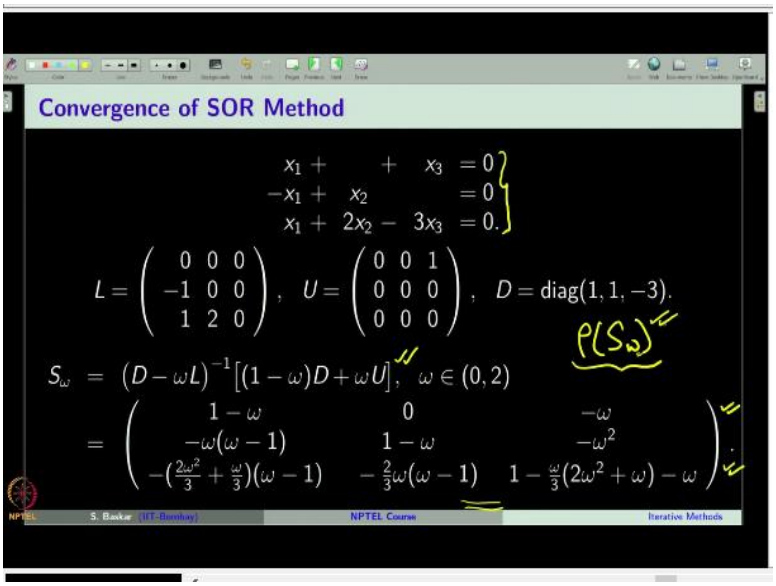
(Refer Slide Time: 25:36)



You can immediately see that A can be written as the diagonal matrix D given like this plus the strictly lower triangular matrix L given like this plus the strictly upper triangular matrix U given like this. Once you have this, you can immediately form the iterative matrix for the SOR method for any non-zero ω and if you recall we have derived this matrix and its formula is given like this.

For any ω, now is restricted to 0 to 2 because we have seen that if you go away from this interval the SOR method will not converge.

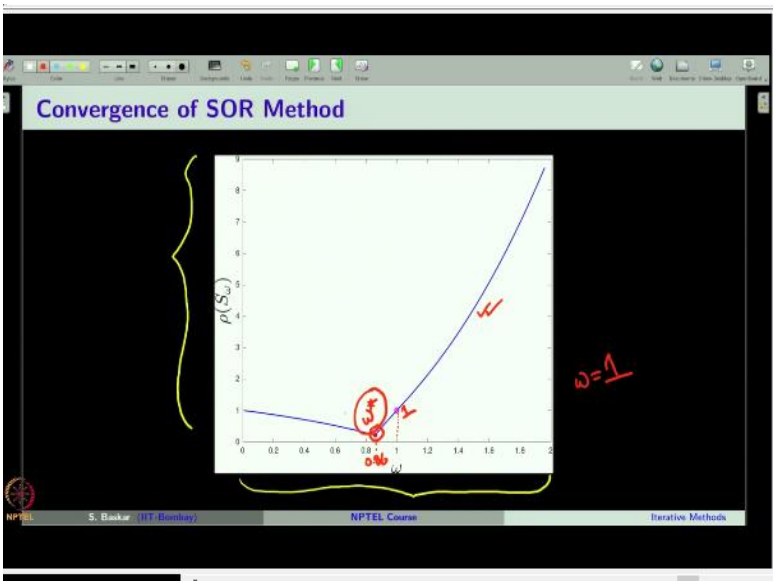
(Refer Slide Time: 26:35)



With little difficulty you can explicitly write this matrix and it is given like this. Now you see if you want to get the spectral radius of S_ω as a function of ω you will have to find the eigenvalues of this matrix which itself will be very difficult. Remember you are only with 3×3 system. For this itself the spectral radius expression will become very complicated. That is why we told that finding an optimal ω is in general very difficult.

It may be possible for 2×2 case but even for 3×3 it becomes very difficult because you have to find all the eigenvalues of this matrix and then you have to take the modulus of that and then take the square root of that eigenvalues and take the maximum. And that will give you a function of ω and now you have to find the minimum of that function. So, that is theoretically very complicated.

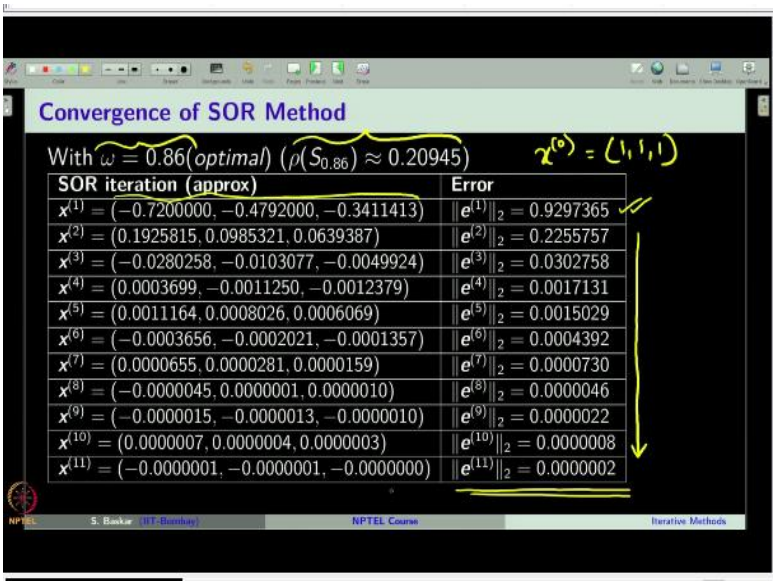
(Refer Slide Time: 27:56)



Rather you can just compute the spectral radius numerically and you can plot it and see where the minimum of the spectral radius of S_ω is achieved as the function of ω. Here I am plotting the graph of $\rho(S_\omega)$ where the *x*-axis represents the values of ω and the *y*-axis represents the values of $\rho(S_\omega)$ and the blue line is the graph of the function $\rho(S_\omega)$ and the optimal ω that is ω^* is achieved at this point which is roughly 0.86.

And also, you can see that if you take ω = 1 we get back our Gauss-Seidel method. And the spectral radius of Gauss-Seidel method as we have seen explicitly its value is 1. Now let us apply the SOR method with ω is equal to the optimal value ω^*.

(Refer Slide Time: 29:14)



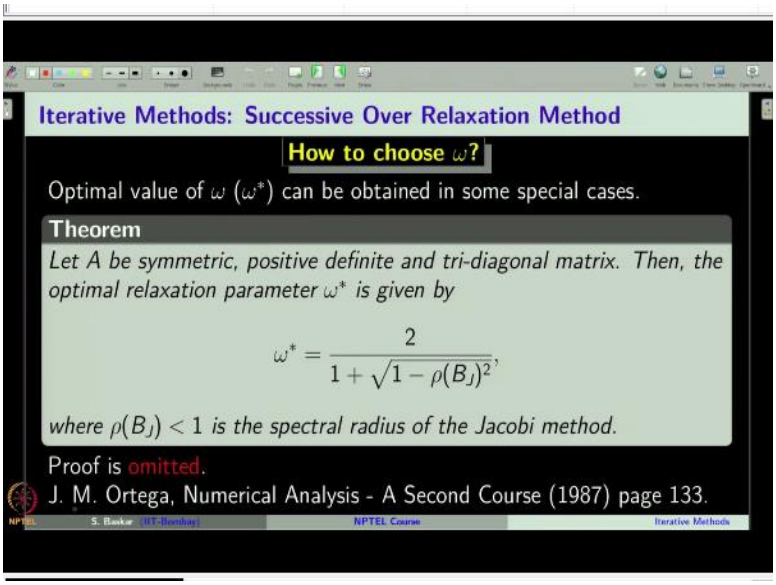
And see how the iteration goes on. As I told let us take ω is equal to the optimal value which is roughly 0.86. For this ω the spectral radius of the corresponding iterative matrix of the SOR

method is around 0.21 which is pretty close to 0. Therefore, we expect a rather faster convergence in the SOR method for omega = 0.86. Let us see how the iterations go. We again take $x^{(0)} = (1,1,1)$ and see how the iteration goes.

The first iteration is given like this and the L_2 error involved in the first iteration is around 0.92 and as we go on with the iterations you can see that the error falls down quite rapidly and within 11 iterations we got a pretty accurate solution for our system. If you recall we have also computed Jacobi iteration for this system. And we have seen that after 200 iterations the error was around 0.00002 which means we captured the solution up to around 5 significant digits only after 200 iterations.

Whereas, you can see that we captured the exact solution up to around 7 significant digits within 11 iterations using SOR method with omega = 0.86 which is the optimal value of ω in this particular example. Also recall that the Gauss-Seidel method gave an oscillating sequence and therefore it is not a convergent sequence at all. So, in that way SOR method with the proper choice of ω may often lead to a faster convergence. However, getting the optimal parameter ω^* is in general very difficult.

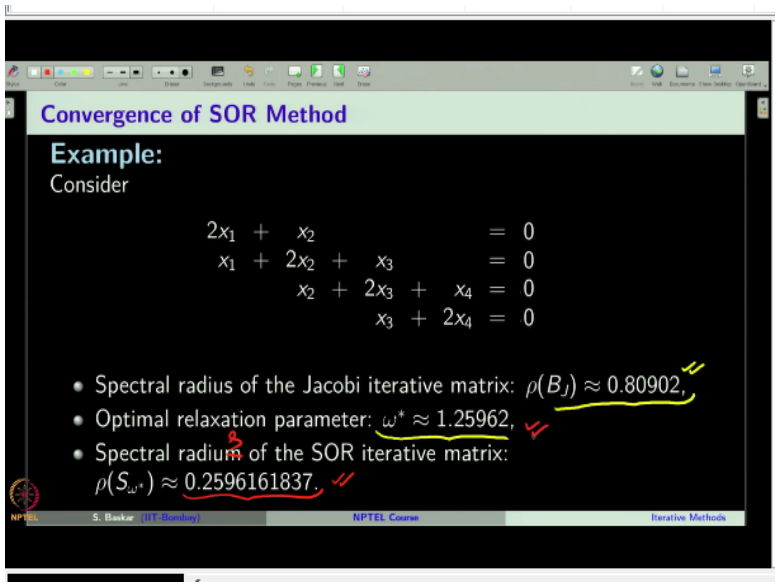
(Refer Slide Time: 31:54)



However, in certain particular cases we can obtain the optimal value of ω. For instance, if your matrix is a symmetric positive definite and tridiagonal matrix then the optimal relaxation parameter ω^* can be obtained using this formula. That is $\omega^* = \frac{2}{1+\sqrt{1-\rho(B_J)^2}}$. Remember, this is the notation we used for the iterative matrix for Jacobi method.

So, first you have to compute the Jacobi iterative method and find the spectral radius of that matrix. Once you have this you can get the optimal relaxation parameter for SOR method. If the given matrix is a symmetric positive definite and tridiagonal matrix and the formula is given like this. The proof of this theorem is very difficult, so we will omit it in our course. However, interested students can get the proof of this theorem from a classical book authored by J. M. Ortega.

(Refer Slide Time: 33:35)

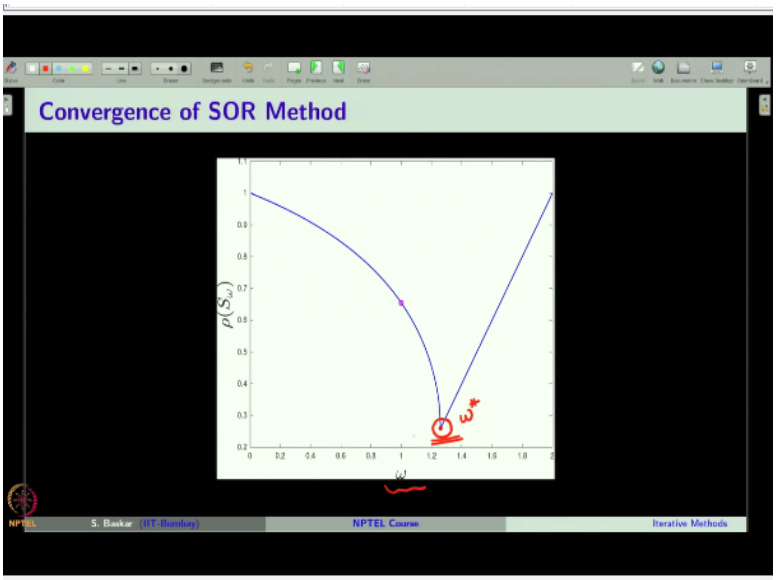


Let us take an example, let us consider this 4×4 tri-diagonal system. You can see that this system is a symmetric and positive definite system. Let us try to find the optimal ω for the SOR method to this system. For that you first have to find B_J and then obtain its spectral radius. How will you get? You have to find all the eigenvalues of this matrix B_J. Remember we have given the formula of B_J in one of our previous lectures.

You go back and see that formula then compute this matrix B_J and then find all the eigenvalues of that matrix B_J and from there you can get the spectral radius. Once you get the spectral radius you can simply use this formula to get the optimal value of the relaxation parameter and for this system the spectral radius of the Jacobi iterative matrix is given by 0.80902. And once you have this you can immediately get the optimal relaxation parameter using the formula given in our previous theorem and it is given by 1.25962.

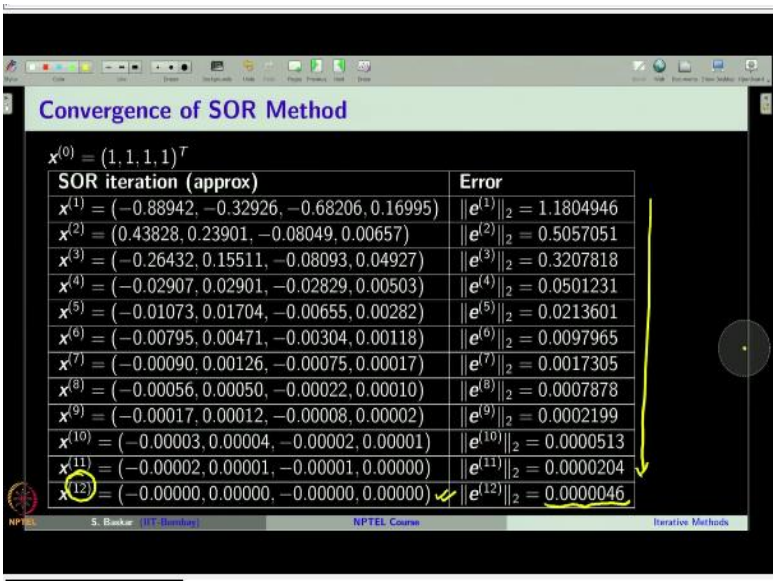
In fact from the explicit formula, you can clearly see that the optimal parameter in this case of the coefficient matrix will always lie in the interval 1 to 2. And the spectral radius is given by 0.2596. That is again pretty small. Therefore, we expect a faster convergence again for this system when we use the SOR method with ω is equal to this.

(Refer Slide Time: 35:59)



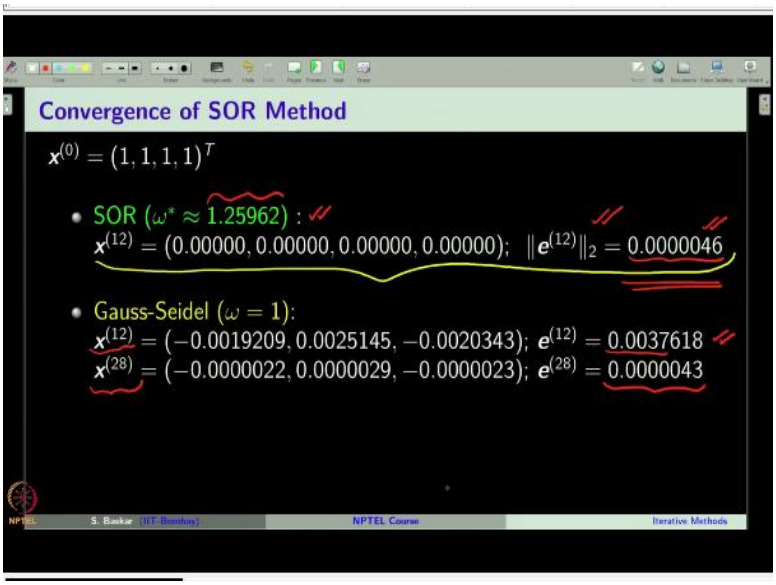
Let us see the plot of the spectral radius of S_ω as a function of ω. You can see that for any value of ω in the open interval (0 ,1) the spectral radius of the corresponding S_ω is strictly less than 1. That means for any choice of ω, SOR method will surely converges for this particular linear system and moreover you can also see that ω^* which we have computed using the formula given in the theorem is indeed the minimum of this function $\rho(S_\omega)$.

(Refer Slide Time: 36:48)



Again, we will start our initial guess with (1, 1, 1) and you can see that the convergence is pretty good and we got around 6 significant digits of the exact solution. Remember the exact solution is just the zero vector and we got the exact solution up to around 6 significant digits in just 12 iterations.

(Refer Slide Time: 37:21)



So, this is what finally we got in the SOR method when we took the ω as the optimal value of ω which is 1.2596 around and the same when we take ω equal to 1 as you can see in this plot at ω = 1 this spectral radius is way below 1, it is around 6.5. Therefore, even Gauss-Seidel method will converge. But as you can see that the Gauss-Seidel method is converging rather slowly because its spectral radius is around 0.65, whereas the spectral radius of SOR with this parameter is around 0.25.

Therefore, the SOR method converged quite faster that is in 12 iterations we got around 6 significant digits accurately, whereas after 12 iterations you can see that the Gauss-Seidel method give around only 3 significant digits. And in fact, Gauss-Seidel method took around 28 iterations to achieve the accuracy which is almost equivalent to the accuracy achieved by the SOR method.

So, in this way we could accelerate the convergence of the Gauss-Seidel method quite efficiently by introducing a relaxation parameter into the sequence and we derived a resulting method called SOR method. With this let us finish this lecture and thank you for your attention.