Matrix Solvers

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

Lecture - 35
Numerical Experiments on Convergence

Welcome. In the last few sessions, I try to introduce Jacobi and Gauss-Seidel methods, which are the basic iterative solvers. And then, we also discussed about the properties of the matrix A for which this solvers should work. And we have seen that the matrix has to be diagonally dominant or irreducibly diagonally dominant that means, diagonal dominant means all the diagonal terms must be greater than sum absolute value of all the diagonal terms must be greater than sum of absolute value of the off diagonals for each particular row.

And diagonal irreducibly diagonally dominance means, the that sum the all absolute value of diagonal term must be greater than equal to sum of the absolute values of the off diagonal terms for each particular row. And there should be at least one row for which it is not greater than equal to rather it is greater than the sum of the off diagonal terms, that I was absolute value of the diagonal term is greater than the sum of the absolute value of the off diagonals.

And for this cases, Jacobi and Gauss-Seidel works. And Jacobi and Gauss-Seidel works for these cases, because we looked into the iteration matrix G, iteration matrix G will have a spectral radius less than 1, which ensure we will ensure that the matrix norm of G will be less than 1. And as we will do a number of iterations, the initial error which will be multiplied by matrix norm of G to the power k; k is a large number, which this will be a very small number. Therefore, it should converge the error should be very close to 0, and it should converge to the right solution.

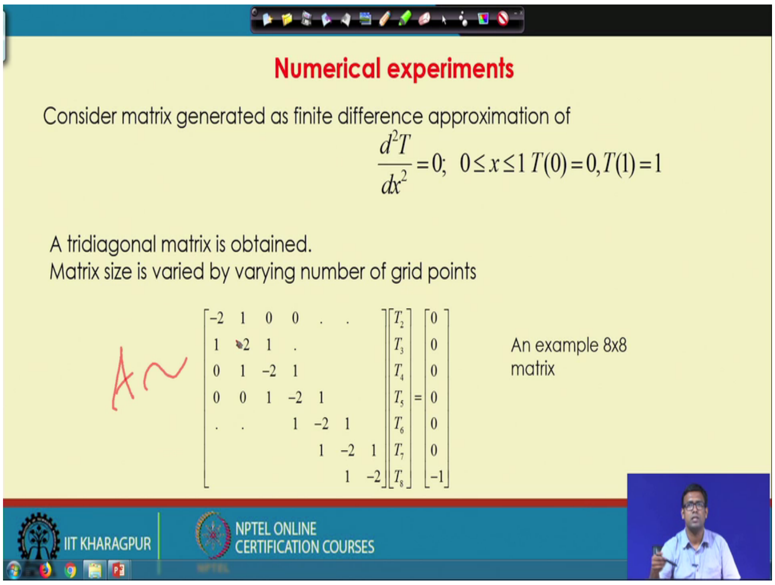
And if it is converges that means, if the guess after now a large number of iteration, the solution converges that means, the guess an updated value has very small difference in between them. It is must converge to the right solution that is also property of Jacobi and Gauss-Seidel.

So, now what we will do, we will do some numerical experiments using Jacobi and Gauss-Seidel. We also introduce the terms convergence rate, convergence factor, and I have seen that it depends on the matrix G, it depends how is splitting the main matrix say the only condition for the we are solving A x is equal to b. The only condition on the main matrix A is that, it has to be diagonally dominant or irreducibly diagonally dominant.

Now, it depends how are we splitting the matrix A to get the matrix G. And based on that, we will get the eigenvalues of G the largest eigenvalue is the spectral radius, which is also a measure of the convergence factor as well as convergence rate of the iterative scheme. So, and we will tell us how fast the iterations will converge. So, what will do, we do is called numerical experiments that means, we will take few matrices and do experiment with these matrices. So, we will try to ran Jacobi and Gauss-Seidel on this matrices.

As well as we will see, how are there how are there G matrices, what are the spectral radius of G matrix, what is the convergence factor or convergence rate, so that we can say a priory looking into the G matrix, we should be able to tell that whether they will converge fast, whether they will at all converge or not etcetera.

(Refer Slide Time: 03:42)



Numerical experiments

Consider matrix generated as finite difference approximation of

$$\frac{d^2T}{dx^2}=0;\quad 0\le x\le 1\ T(0)=0, T(1)=1$$

A tridiagonal matrix is obtained.
Matrix size is varied by varying number of grid points

$$A \sim \begin{bmatrix} -2 & 1 & 0 & 0 & . & . & \\ 1 & -2 & 1 & . & & & \\ 0 & 1 & -2 & 1 & & & \\ 0 & 0 & 1 & -2 & 1 & & \\ . & . & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{bmatrix} \begin{bmatrix} T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

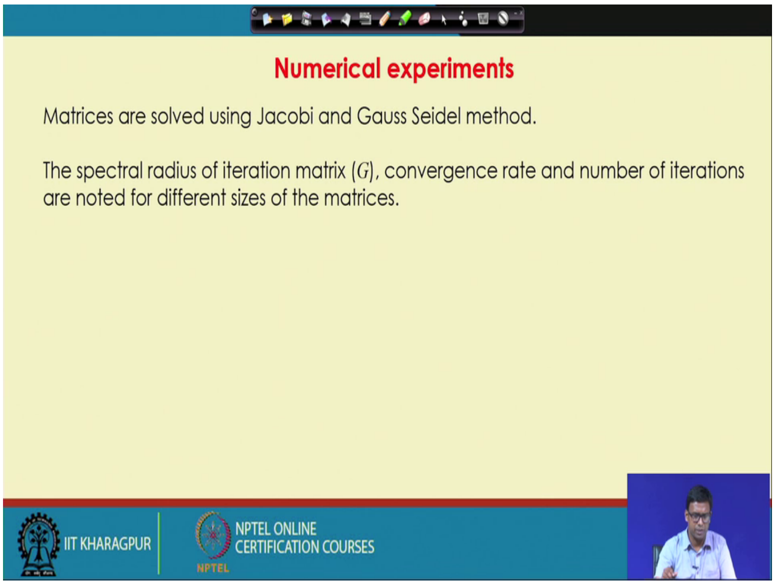
An example 8x8 matrix

So, for this numerical experiments, we have chosen particular matrix, which is the matrix generated as finite difference approximation of d 2 T d x square is equal to 0. We have discussed about how to convert a differential equation into a matrix equation, exactly data has been done here. And the boundary condition T x is 0 to 1, boundary condition is T is equal to 0, it is 0, T is equal to 1 that value is 1. And we now, so this will give essentially a tridiagonal system, we varied the number of intervals and got large number of points, and got larger matrix systems.

So, a tridiagonal matrix is obtained. Matrix size is varied by number varying the number of grid points or number of intervals, and we got different matrices. But, the matrices will essentially look like a tridiagonal matrix for this is an example of a the 8 by 8 matrix. However, here we consider 10 by 10, 20 by 20, 40 by 40 matrices. And then with this matrix, we ran a Gauss-Seidel or Jacobi program. Later, we will also discuss how to write this program. So, we have to implement a Gauss-Seidel or Jacobi into a computer program, we ran a program.

And noted, what are the number of iterations to solve it. Also noted into the G matrix, which comes out of so, this is basically how the A matrix looks like here. But, this is 8 by 8, we have taken several matrices. So, seen how is the G matrix, and then try to find out the eigenvalues of the G matrix from which we can get the spectral radius, and see how is the convergence property.

(Refer Slide Time: 05:27)



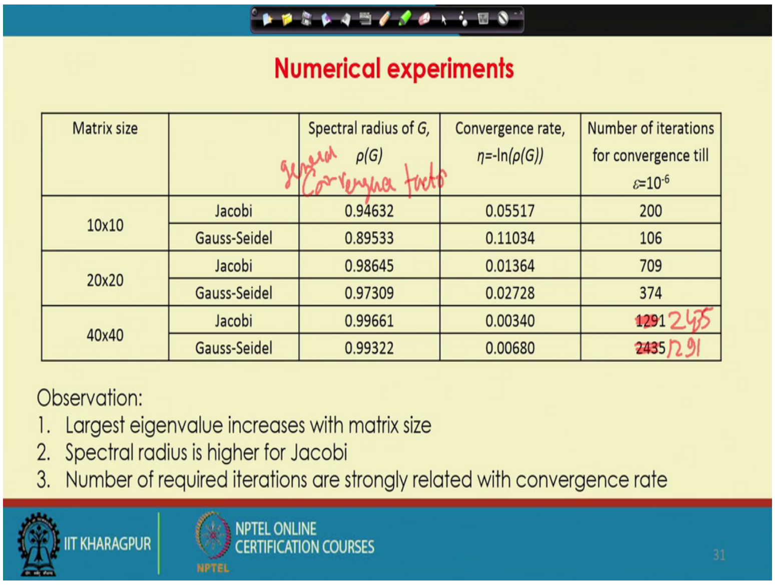
Numerical experiments

Matrices are solved using Jacobi and Gauss Seidel method.

The spectral radius of iteration matrix (G), convergence rate and number of iterations are noted for different sizes of the matrices.

So, matrices are solved using Jacobi and Gauss-Seidel method, because this is doing experiment with the matrices and the method And but, this is not experiment in hardware, we call it numerical experiment. The spectral radius of matrix G, convergence rate convergence rate is log minus logarithm of the convergence factor of the spectral radius, and the number of iteration, and noted for different sizes of the matrices.

(Refer Slide Time: 05:53)



Numerical experiments

Matrix size		Spectral radius of G, $\rho(G)$ general Convergence factor	Convergence rate, $\eta=-\ln(\rho(G))$	Number of iterations for convergence till $\varepsilon=10^{-6}$
10x10	Jacobi	0.94632	0.05517	200
	Gauss-Seidel	0.89533	0.11034	106
20x20	Jacobi	0.98645	0.01364	709
	Gauss-Seidel	0.97309	0.02728	374
40x40	Jacobi	0.99661	0.00340	~~1291~~ 2435
	Gauss-Seidel	0.99322	0.00680	~~2435~~ 1291

Observation:

1. Largest eigenvalue increases with matrix size
2. Spectral radius is higher for Jacobi
3. Number of required iterations are strongly related with convergence rate

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES NPTEL 31

So, for 10 by 10 matrix for Jacobi iteration, the spectral radius is 0.94632; for Gauss-Seidel, the spectral radius is 0.89533. So, for Gauss-Seidel, the spectral radius is smaller. The convergence rate is 0.05517 for Jacobi, and for Gauss-Seidel is 0.11034. So, convergence rate is almost double in case of Gauss-Seidel than Jacobi. And this is due to the fact, that the spectral radius is smaller.

So, if the convergence is double, the rate will be double, speed will be faster; so, the number of iterations will be small, faster it will converge. And we can see that this is almost have, if we need 200 iterations for Jacobi for Gauss-Seidel, we need 106 iterations. So, there is a relation between convergence rate and number of iterations. Roughly if we multiply convergence rate with number of iterations, there that is roughly constant, at least in this case. In other cases, we will also see at least for same size of matrix.

So, once we have a method in a Gauss-Seidel, we have a laser spectral radius, therefore the convergence will be faster the number of iterations so, rho G is also convergence factor right rho G is also convergence factor general convergence factor. So, if this is smaller, the iterations will we need less number of iterations, the scheme will be faster.

Now, we go to 20 into 20, we can see similar thing. The spectral radius of Jacobi is 0.98645, cell in not lot more iterations spectral radius is very close to 1. The convergence rate is 0.101364. The Gauss-Seidel spectral radius is 0.973, convergence rate is 0.027. And the number of iterations are in almost close to half of each other 709 in Jacobi, and 374 in Gauss-Seidel.

And now, we can also see the also you can see that for convergence rate in 10 into 10 Jacobi is 0.055, it added in Gauss-Seidel is 0.013 nearly four times, convergence rate is reduced by nearly four times. And iteration number is increased by 23.5 times. So, there is a relation between convergence rate and iteration number for similar type of matrices. If we increase 40 into 40, the Jacobi converse spectral radius is very high 0.99661, convergence radius is 0.0034. Gauss-Seidel 0.99324, convergence radius is 0.0068, because the iteration number is very high. I think this is swiped, this is swiped this will be well change that in the later slide, this is 2435 this is 1291 this is (Refer Time: 09:02) while making the table.

So, because the iteration spectral radius is very high very close to 1, we need a very large number of iterations to 2435 iterations here. And in Gauss-Seidel, we need it is also high, we need 1200 iterations. So, as the spectral radius is increasing, we are also needing large number of iterations. However, Jacobi consistently give larger spectral radius than Gauss-Seidel, and it consistently takes more number of iterations than Gauss-Seidel. And the convergence rate is reversed in both the cases.

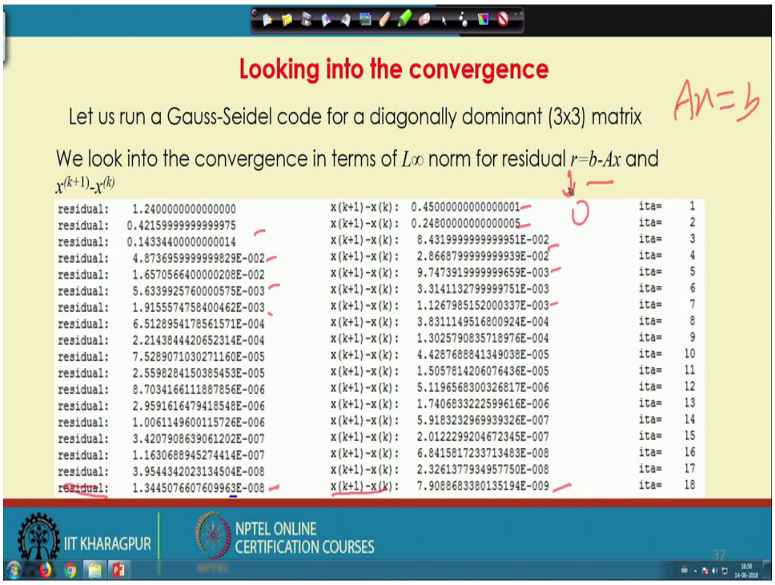
So, the observations are, largest eigenvalues or the spectral radius increases with matrix size. Matrix 20 into 20, eigenvalue largest eigenvalue is more. And spectral radius is and because largest eigenvalue increases, spectral radius also increases, and number of iterations are more. Spectral radius is higher for Jacobi, then than Gauss-Seidel. And we when proposing Gauss-Seidel, almost intuitively you propose that if we can use some of the iterative values some of the updated values, the iteration should converge faster. So, this is an almost intuitive all actually intuitive proposition, however we can see that it is actually giving faster convergence. And why is it giving faster convergence, because in the iteration matrix level, the spectral radius has been reduced well.

The 3rd observation is that number of iterations required for convergence is strongly related with convergence rate. Almost if we have like here convergence rate is almost twice for Jacobi and Gauss-Seidel, and the number of iterations is again almost half in Gauss-Seidel compared to Jacobi. So, larger its convergence rate and number of iterations are related with each other. For a same matrix, Jacobi gives higher spectral radius, therefore slower convergence rate compared to Gauss-Seidel. And as we will increase matrix size, the largest eigenvalue increases, convergence rate decreases.

This is actually very important the first observation, because we will come maybe at the very end of this course, when we will discuss about what is called multi grid, we will look into large matrix size issues with larger matrix size. But, if we are increasing larger matrix size, we are taking lot of grid points. You were taking lot of grid points, you are doing less numerical errors in terms of the deceleration error.

However, the number of iterations increase, and the computational cost increase. Why the number of iterations increase, because the eigenvalue of the largest eigenvalue of the G matrix or spectral radius of the G matrix increases, if we take large matrix. So, large matrix many number of grid points will give us accurate result, but it will be a slow process, and the computational cost will be high.

(Refer Slide Time: 12:04)



Looking into the convergence

$Ax=b$

Let us run a Gauss-Seidel code for a diagonally dominant (3x3) matrix

We look into the convergence in terms of $L\infty$ norm for residual $r=b-Ax$ and $x^{(k+1)}-x^{(k)}$

```
residual:   1.2400000000000000          x(k+1)-x(k):  0.45000000000000001          ita=    1
residual:  0.42159999999999975          x(k+1)-x(k):  0.24800000000000005          ita=    2
residual:  0.14334400000000014          x(k+1)-x(k):   8.4319999999999951E-002     ita=    3
residual:   4.8736959999999829E-002     x(k+1)-x(k):   2.8668799999999939E-002     ita=    4
residual:   1.6570566400000208E-002     x(k+1)-x(k):   9.7473919999999659E-003     ita=    5
residual:   5.6339925760000575E-003     x(k+1)-x(k):   3.3141132799999751E-003     ita=    6
residual:   1.9155574758400462E-003     x(k+1)-x(k):   1.1267985152000337E-003     ita=    7
residual:   6.5128954178561571E-004     x(k+1)-x(k):   3.8311149516800924E-004     ita=    8
residual:   2.2143844420652314E-004     x(k+1)-x(k):   1.3025790835718976E-004     ita=    9
residual:   7.5289071030271160E-005     x(k+1)-x(k):   4.4287688841349038E-005     ita=   10
residual:   2.5598284150385453E-005     x(k+1)-x(k):   1.5057814206076436E-005     ita=   11
residual:   8.7034166111887856E-006     x(k+1)-x(k):   5.1196568300326817E-006     ita=   12
residual:   2.9591616479418548E-006     x(k+1)-x(k):   1.7406833222599616E-006     ita=   13
residual:   1.0061149600115726E-006     x(k+1)-x(k):   5.9183232969939326E-007     ita=   14
residual:   3.4207908639061202E-007     x(k+1)-x(k):   2.0122299204672345E-007     ita=   15
residual:   1.1630688945274414E-007     x(k+1)-x(k):   6.8415817233713483E-008     ita=   16
residual:   3.9544342023134504E-008     x(k+1)-x(k):   2.3261377934957750E-008     ita=   17
residual:   1.3445076607609963E-008     x(k+1)-x(k):   7.9088683380135194E-009     ita=   18
```

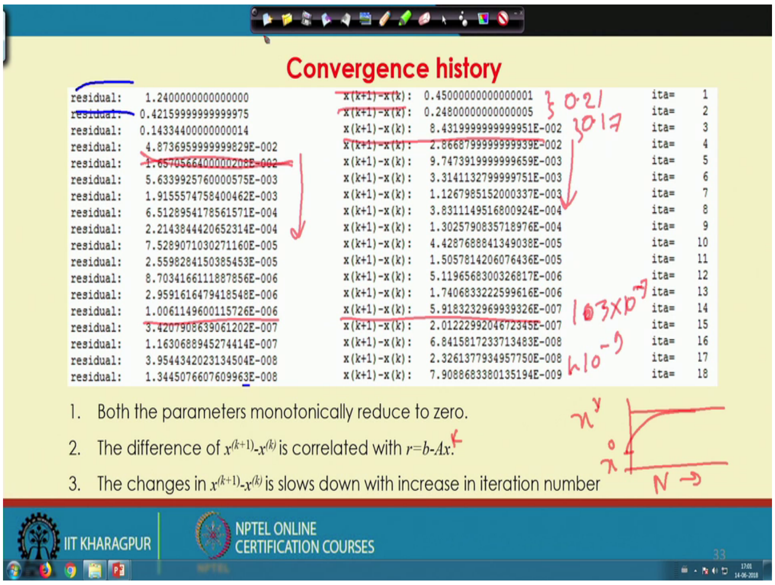
0

So, now let us look into the convergence in detail. Let us run a Gauss-Seidel code for a diagonally dominant a small matrix 3 by 3 diagonally dominant matrix. We look into the convergence in terms of L infinity norm of the residual b minus A x, and we will also look into the difference of x k plus 1 minus x k.

The residual initially start with some guess x, x is equal to 0. First iteration, residual is 1.24, and x k plus 1 minus x k is 0.45. The next case, we use the updated x, the residual becomes 0.42, and the difference between x x k plus 1 minus x k becomes 0.248.

So, as this difference decreases we can see as this difference decreases, the residual also decreases. And this difference decreases asymptotically at the end, x k plus 1 minus x k is 10 to the power minus 9 and the residual is 10 to the power minus 8. We will say that the difference this has converged. So, A x minus the difference between b minus A x which should be actually 0 is also practically 0, because we are solving A x is equal to b. So, this should also come to 0, as well as this also should go to 0 for convergence and accuracy. So, as a solution becomes accurate, the difference goes to 0.

(Refer Slide Time: 12:42)



Convergence history

```
residual:   1.2400000000000000        x(k+1)-x(k):  0.45000000000000001        ita=    1
residual:  0.42159999999999975        x(k+1)-x(k):  0.24800000000000005        ita=    2
residual:  0.14334400000000014        x(k+1)-x(k):   8.4319999999999951E-002  ita=    3
residual:   4.8736959999999829E-002   x(k+1)-x(k):   2.8668799999999939E-002  ita=    4
residual:   1.6570566400000208E-002   x(k+1)-x(k):   9.7473919999999659E-003  ita=    5
residual:   5.6339925760000575E-003   x(k+1)-x(k):   3.3141132799999751E-003  ita=    6
residual:   1.9155574758400462E-003   x(k+1)-x(k):   1.1267985152000337E-003  ita=    7
residual:   6.5128954178561571E-004   x(k+1)-x(k):   3.8311149516800924E-004  ita=    8
residual:   2.2143844420652314E-004   x(k+1)-x(k):   1.3025790835718976E-004  ita=    9
residual:   7.5289071030271160E-005   x(k+1)-x(k):   4.4287688841349038E-005  ita=   10
residual:   2.5598284150385453E-005   x(k+1)-x(k):   1.5057814206076436E-005  ita=   11
residual:   8.7034166111887856E-006   x(k+1)-x(k):   5.1196568300326817E-006  ita=   12
residual:   2.9591616479418548E-006   x(k+1)-x(k):   1.7406833222599616E-006  ita=   13
residual:   1.0061149600115726E-006   x(k+1)-x(k):   5.9183232969939326E-007  ita=   14
residual:   3.4207908639061202E-007   x(k+1)-x(k):   2.0122299204672345E-007  ita=   15
residual:   1.1630688945274414E-007   x(k+1)-x(k):   6.8415817233713483E-008  ita=   16
residual:   3.9544342023134504E-008   x(k+1)-x(k):   2.3261377934957750E-008  ita=   17
residual:   1.3445076607609963E-008   x(k+1)-x(k):   7.9088683380135194E-009  ita=   18
```

1. Both the parameters monotonically reduce to zero.
2. The difference of $x^{(k+1)}$-$x^{(k)}$ is correlated with $r=b-Ax^{k}$.
3. The changes in $x^{(k+1)}$-$x^{(k)}$ is slows down with increase in iteration number

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES

Now, then make few observations from that. That both the parameters x k plus 1 minus x k x k plus 1 minus x k and residual, both these parameters monotonically reduced to 0. These are the L infinity norm that is the largest value of the both these are vectors, so largest component of the vector. Both these parameters monotonically reduce to 0, there is no oscillation. It is never increasing, it is continuously reducing to 0.

So, for Gauss-Seidel or Jacobi or all something like these direct basic iterative methods, the errors or the residuals must not oscillate, they must monotonically go to 0. If there is something different, if it is oscillating, then there is some problem in the implementation.

Why, because every time this residual is being multiplied by G to the power k, which is reducing the norm of the vector.

The difference x k plus 1 minus x k is correlated with r is equal to b minus A x k b minus A x k. As the residual and the difference is correlated in a sense, if this is reducing, this is also reducing. Like as the smaller the value of x k plus 1 minus x k is being like, this is smaller than this value. Therefore, the residual is also smaller than this value, so they are related.

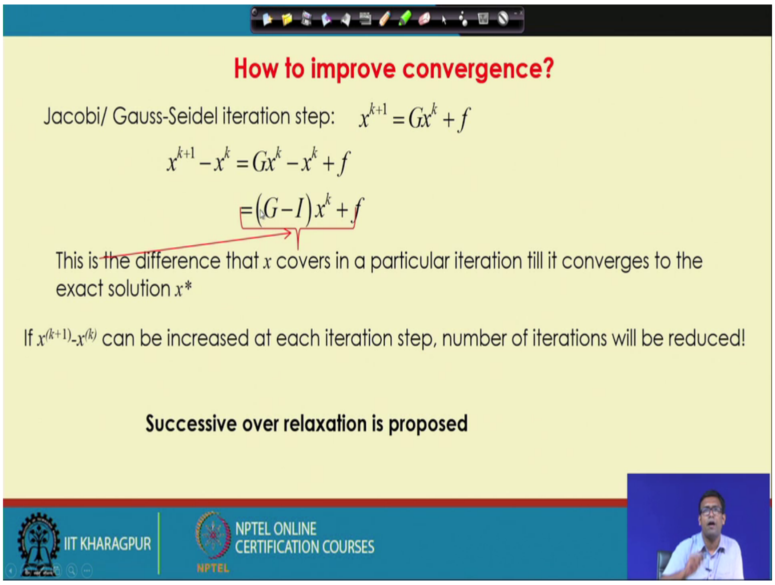
Interestingly, the residual as well as this difference, the rate of slowing down is not reduces with time. For example, here this difference reduces by 0.21, in the next step this difference reduces by 0.17 something like that. Here this difference reduces by 0.3 into 10 to the power minus 7. Here this difference reduces by something order of 10 to the power minus 9. So, the rate at which the differences are reducing is also slowing down with number of iterations.

So, if I say that the that this is the x star and I start up started with x 0, and this is the number of iterations, this is x star this is x 0. This kind of asymptotically approaches it, the rate slows down with time. It is an important observation here. However, so this is what were we gating using Gauss-Seidel or this particular case is I think for (Refer Time: 16:28) this particular case is for a Gauss-Seidel solver, these for a Gauss-Seidel solver.

For Jacobi it will give similar result, but we will need a more number of iterations the rate at which the residual falls down is smaller because G has a larger at the spectral radius. However, so what do you see that they are following certain pattern and this pattern is determined by how is G what are the eigenvalues of G being a largest eigenvalue of G in Gauss-Seidel or Jacobi method.

Now, our question becomes that can we improve it, can we increase the spectral radius through some splitting will not be intuitively we found out 1 splitting which is Gauss-Seidel can we have some other splitting through which we can improve the convergence rate.

(Refer Slide Time: 17:23)



How to improve convergence?

Jacobi/ Gauss-Seidel iteration step: $x^{k+1} = Gx^k + f$

$$x^{k+1} - x^k = Gx^k - x^k + f$$

$$= \underbrace{(G-I)x^k + f}$$

This is the difference that x covers in a particular iteration till it converges to the exact solution x^*

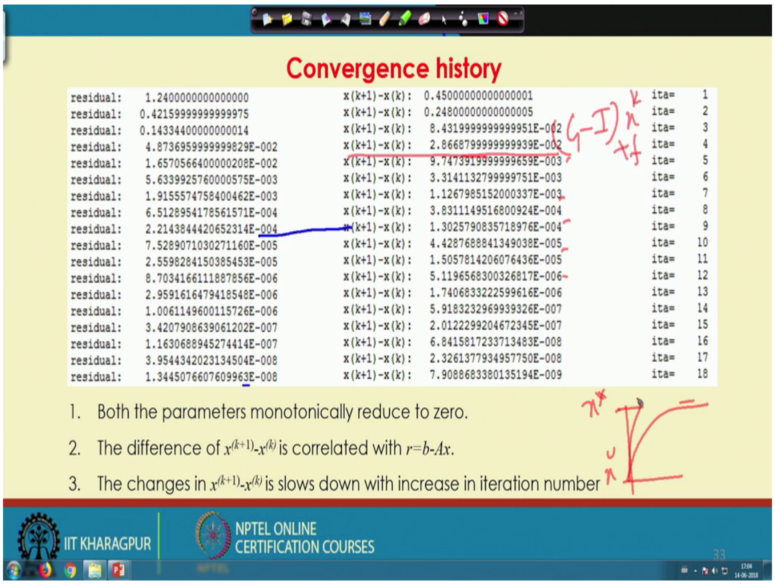
If $x^{(k+1)}$-$x^{(k)}$ can be increased at each iteration step, number of iterations will be reduced!

Successive over relaxation is proposed

How to improve convergence? Jacobi or Gauss-Seidel gives x k plus 1 is equal to G x k plus f we will think of some and this G comes out splitting of a is equal to a is equal to a minus n which is a minus d minus d minus n minus f or d minus n minus f something like that and then forming m inverse n as G. Now, we will try to A is equal to m minus n m inverse n is equal to G that is the idea we followed here. Now, if we try to improve convergence that means, if we try to take m and n in some different way, so that the G has a smaller spectral radius can we do that.

So, again we will start with some intuitive idea here and we will see the effect in the spectral radius later. So, what we do at every iteration step is that x k plus 1 minus x k is equal to G x k minus x k plus f. So, after every iteration step x k plus 1 this is the iteration step is modified this is the error be difference between consecutive guesses updates. The x k plus 1 is modified by this term G minus I x k plus f.

(Refer Slide Time: 18:54)



Convergence history

```
residual:    1.2400000000000000              x(k+1)-x(k):  0.45000000000000001            ita=   1
residual:   0.42159999999999975              x(k+1)-x(k):  0.24800000000000005            ita=   2
residual:   0.14334400000000014              x(k+1)-x(k):   8.4319999999999951E-002        ita=   3
residual:    4.8736959999999829E-002         x(k+1)-x(k):   2.8668799999999939E-002        ita=   4
residual:    1.6570566400000208E-002         x(k+1)-x(k):   9.7473919999999659E-003        ita=   5
residual:    5.6339925760000575E-003         x(k+1)-x(k):   3.3141132799999751E-003        ita=   6
residual:    1.9155574758400462E-003         x(k+1)-x(k):   1.1267985152000337E-003        ita=   7
residual:    6.5128954178561571E-004         x(k+1)-x(k):   3.8311149516800924E-004        ita=   8
residual:    2.2143844420652314E-004         x(k+1)-x(k):   1.3025790835718976E-004        ita=   9
residual:    7.5289071030271160E-005         x(k+1)-x(k):   4.4287688841349038E-005        ita=  10
residual:    2.5598284150385453E-005         x(k+1)-x(k):   1.5057814206076436E-005        ita=  11
residual:    8.7034166111887856E-006         x(k+1)-x(k):   5.1196568300326817E-006        ita=  12
residual:    2.9591616479418548E-006         x(k+1)-x(k):   1.7406833222599616E-006        ita=  13
residual:    1.0061149600115726E-006         x(k+1)-x(k):   5.9183232969939326E-007        ita=  14
residual:    3.4207908639061202E-007         x(k+1)-x(k):   2.0122299204672345E-007        ita=  15
residual:    1.1630688945274414E-007         x(k+1)-x(k):   6.8415817233713483E-008        ita=  16
residual:    3.9544342023134504E-008         x(k+1)-x(k):   2.3261377934957750E-008        ita=  17
residual:    1.3445076607609963E-008         x(k+1)-x(k):   7.9088683380135194E-009        ita=  18
```

$(G-I)x^k + f$

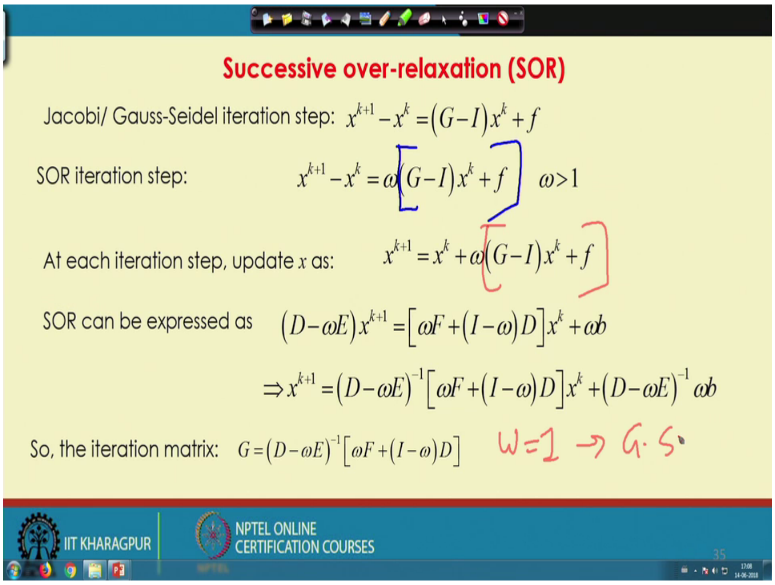
1. Both the parameters monotonically reduce to zero.
2. The difference of $x^{(k+1)}$-$x^{(k)}$ is correlated with $r=b-Ax$.
3. The changes in $x^{(k+1)}$-$x^{(k)}$ is slows down with increase in iteration number

Now, if we go to my previous slide what was looking here is that that this is this is G minus I x k plus f. Why is it taking so much time so convergence because this value is reducing with each iteration this value is not a large value this is a very small value. Had this been of larger value we could have achieve the solution we could have conversed to the right solution in a fast manner. Or I showed that this is approaching this the actual solution x star x 0 in something like this, instead if we can use some faster solution or some faster way to converge it.

So, what we will do here we will look into this difference and if you can increase this difference because the it every time we are trying to approach the final solution by adding some value with x k and getting x k plus 1. So, if what are we adding here is G minus I x k plus f in Gauss-Seidel and Jacobi. If we can increase this or if we can scale this so, this is the difference that x covers in a particular iteration till it convergence to the exact solution. If x k plus 1 minus x k can be increased at each iteration steps number of iterations will be reduced. This goes through a very very simple linear logic; if we can increase this value then x k plus 1 will be further updated.

So, it converges to the right result there can be some cons also it might not converge it may diverge. It is increased so much, there it is never coming back. Let us see what happens here. And this method is called a successive over relaxation that you over relaxed the solution you are updating the solution using some relation from the using gauge value you are updating the solution. The increase the amount of update it more over relaxation updating update.

(Refer Slide Time: 21:15)



Successive over-relaxation (SOR)

Jacobi/ Gauss-Seidel iteration step: $x^{k+1}-x^{k}=(G-I)x^{k}+f$

SOR iteration step: $x^{k+1}-x^{k}=\omega\left[(G-I)x^{k}+f\right]$ $\omega>1$

At each iteration step, update x as: $x^{k+1}=x^{k}+\omega\left[(G-I)x^{k}+f\right]$

SOR can be expressed as $(D-\omega E)x^{k+1}=\left[\omega F+(I-\omega)D\right]x^{k}+\omega b$

$$\Rightarrow x^{k+1}=(D-\omega E)^{-1}\left[\omega F+(I-\omega)D\right]x^{k}+(D-\omega E)^{-1}\omega b$$

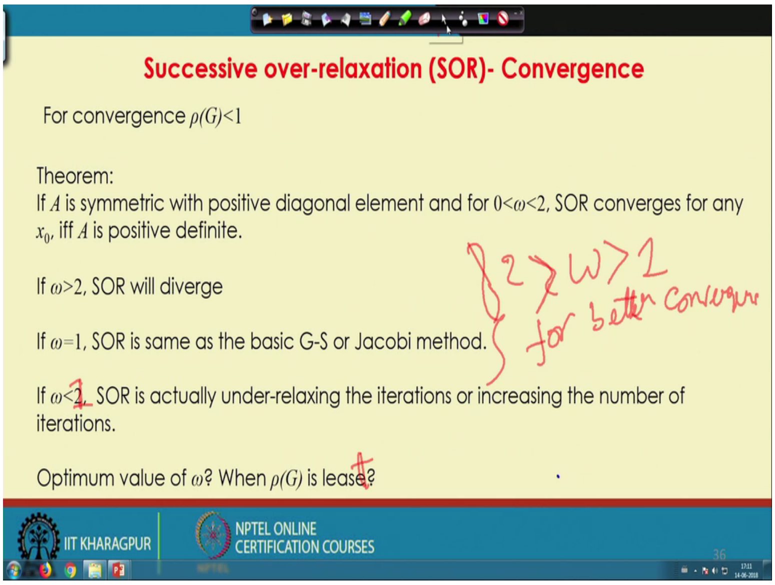
So, the iteration matrix: $G=(D-\omega E)^{-1}\left[\omega F+(I-\omega)D\right]$ $W=1 \rightarrow G.S.$

So, Jacobi Gauss-Seidel step we will look like this and over relaxation step will be like this value is multiplied into omega G I this is this is under a bracket omega into G minus i x k and omega greater than 1. So, at each iteration step update x as x k plus 1 is equal to x k omega G minus i x k sorry plus f this will be plus f (Refer Time: 21:59). The SOR if we write it down by D D and I SOR will be d minus omega e x k plus 1 is equal to omega f plus 1 minus w d x k plus omega b. And the iteration matrix or x k plus 1 is equal to D minus omega inverse w F plus 1 minus or 1 D x k plus D minus omega inverse omega b. So, it is nothing but they follow this step x k plus 1 is equal to x k G G minus I x k plus x.

So, find out the updated value, what is the difference between the updated value and gauges multiplied by some factor omega which is greater than 1. And then add this with the gauge value this is your new update value. And this is equivalent to writing the matrix like this. So, the G matrix iteration matrix is D minus omega E inverse omega F plus I minus w omega D. In case omega is equal to 1, this is not over relaxation this is becomes to the Jacobi or Gauss-Seidel step. We can check that this D minus E inverse f plus D this becomes the Jacobi step in case omega is equal to 1. This is this is sorry Gauss-Seidel step over relaxation is applied over Gauss-Seidel.

In case omega is equal to 1, this goes to Gauss-Seidel. What happens in case omega is equal to 0 and check that. So, this is D inverse f plus D right and f plus D is again D is equal to d minus c minus f is equal to a. So, iteration does not progress it stops there is no update. So, what should be at omega is equal to 0 1, it is Gauss-Seidel; omega less than 1 it is under relax the solution is become slower. So, omega must be greater than 1 what should be the right value of omega for which we will.

(Refer Slide Time: 24:18)



Successive over-relaxation (SOR)- Convergence

For convergence $\rho(G)<1$

Theorem:
If A is symmetric with positive diagonal element and for $0<\omega<2$, SOR converges for any x_0, iff A is positive definite.

If $\omega>2$, SOR will diverge

If $\omega=1$, SOR is same as the basic G-S or Jacobi method.

$2 > \omega > 1$ for better converge

If $\omega<1$, SOR is actually under-relaxing the iterations or increasing the number of iterations.

Optimum value of ω? When $\rho(G)$ is least?

For convergence we need the spectral radius of G less than 1. The theorem if A is symmetric with positive diagonal element an omega is in between 0 to 2 successive over relaxation will converge for any x 0, if A is positive definite, this is a theorem like that. So, if omega greater than 2 successive efficient will not converge the theorem is not satisfied. So, it will diverge.

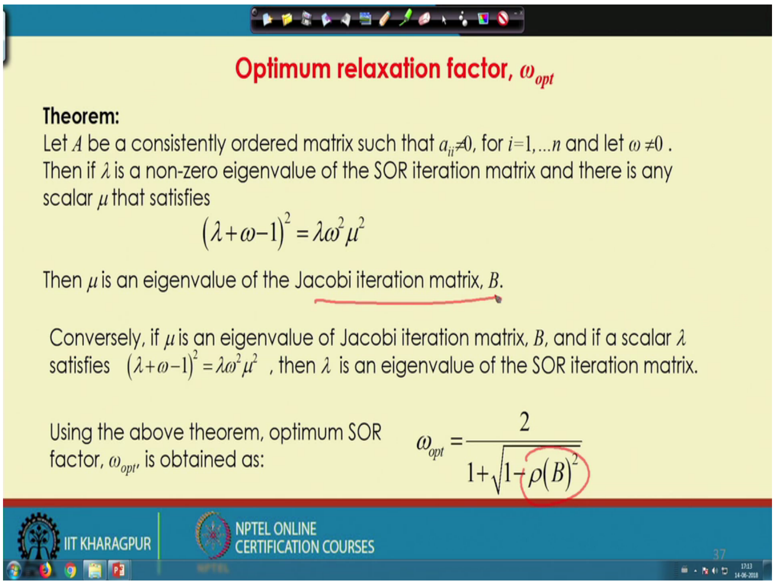
If omega is equal to 1 successive it is same as basic Gauss-Seidel or Jacobi method. If it is applied over Gauss-Seidel it is usually it is applied over Gauss-Seidel because Gauss-Seidel is better method than Jacobi. And we want to further improve convergence. So, we imply successive discussion over the Gauss-Seidel.

If omega is less than 2, successive omega is less than 1 I am sorry, omega is less than 1 successive relaxation is actually under relaxing the iterations are increasing the number of iterations. So, the question is that omega should be between: 1 to 2 for better convergence omega for better convergence.

So, the question is what should be the value of the omega. What is the optimum value of omega? And in that in which case and what is the optimum value of omega when rho G is least will have best convergence rate. So, to either look into the G matrix and c with omega c when it is having the least over relaxation factor least spectral radius.

So, the so we said that that is the best omega is in between 1 to 2 in 2 omega will start diverging in 1 it is same as Gauss-Seidel. So, best omega should be in between 1 to 2. However, there is 1 more theorem which helps us to determine what should be the value of optimum omega.

(Refer Slide Time: 26:37)



Optimum relaxation factor, ω_{opt}

Theorem:

Let A be a consistently ordered matrix such that $a_{ii}\neq 0$, for $i=1,\dots n$ and let $\omega\neq 0$. Then if λ is a non-zero eigenvalue of the SOR iteration matrix and there is any scalar μ that satisfies

$$(\lambda+\omega-1)^2=\lambda\omega^2\mu^2$$

Then μ is an eigenvalue of the Jacobi iteration matrix, B.

Conversely, if μ is an eigenvalue of Jacobi iteration matrix, B, and if a scalar λ satisfies $(\lambda+\omega-1)^2=\lambda\omega^2\mu^2$, then λ is an eigenvalue of the SOR iteration matrix.

Using the above theorem, optimum SOR factor, ω_{opt}, is obtained as:

$$\omega_{opt}=\frac{2}{1+\sqrt{1-\rho(B)^2}}$$

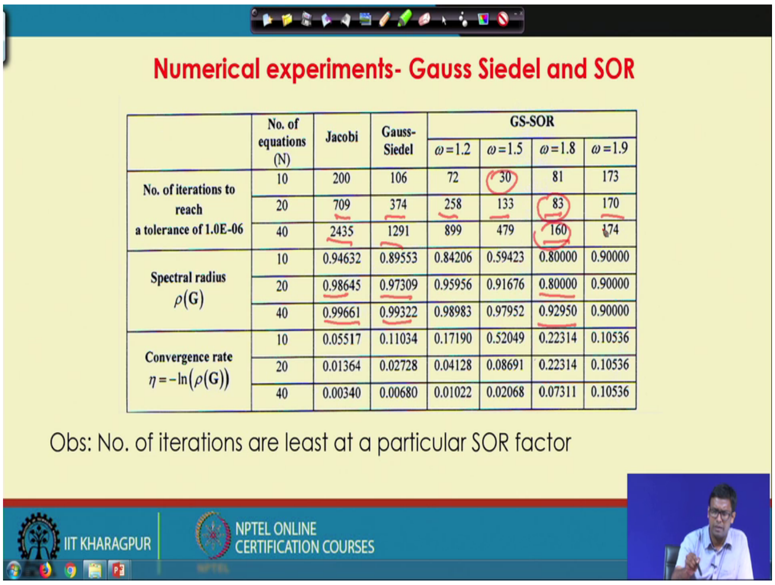
If A be a consistently ordered matrix such that the diagonal element is non-zero there is a definition of consistently ordered matrix will not come here, but all these matrices we deal with is usually consistently ordered matrix. And omega is non-zero and if lambda is a non-zero eigenvalue of the SOR iteration matrix c, there is a scalar mu which satisfies lambda plus omega minus 1 whole square is equal to lambda square omega square mu square then mu is the eigenvalue of the iteration matrix Jacobi iteration matrix B.

So, Jacobi iteration matrix eigenvalue and the S 1 matrix eigenvalue are related by the term omega. So, and using this we can find out for which omega if we know already the iteration matrix we can Jacobi iteration actual matrix we can find out the Jacobi iteration matrix eigenvalue. And find out for which omega we should have the least value of lambda or that will be the or that is what eigenvalue has least smallest spectral radius. Then we will say that this is the optimum omega.

The converse is also true if mu is an eigenvalue of Jacobi iteration matrix Ps and a scalar lambda is there which satisfies lambda plus 1 the omega minus 1 whole square is equal to lambda omega square mu square. Then lambda is an eigenvalue of SOR iteration matrix is the proofs are complex we are not been discussing here.

Using the above relation we can find out the optimum successful relaxation factor omega of 2 which is obtained as omega of this 2 by root over 1 minus 1 plus root over 1 minus rho b whole square which is the this is the spectral radius of the Jacobi iteration matrix B from the same problem.

(Refer Slide Time: 28:30)



Numerical experiments- Gauss Siedel and SOR

	No. of equations (N)	Jacobi	Gauss-Siedel	GS-SOR			
				$\omega=1.2$	$\omega=1.5$	$\omega=1.8$	$\omega=1.9$
No. of iterations to reach a tolerance of 1.0E-06	10	200	106	72	30	81	173
	20	709	374	258	133	83	170
	40	2435	1291	899	479	160	174
Spectral radius $\rho(G)$	10	0.94632	0.89553	0.84206	0.59423	0.80000	0.90000
	20	0.98645	0.97309	0.95956	0.91676	0.80000	0.90000
	40	0.99661	0.99322	0.98983	0.97952	0.92950	0.90000
Convergence rate $\eta=-\ln(\rho(G))$	10	0.05517	0.11034	0.17190	0.52049	0.22314	0.10536
	20	0.01364	0.02728	0.04128	0.08691	0.22314	0.10536
	40	0.00340	0.00680	0.01022	0.02068	0.07311	0.10536

Obs: No. of iterations are least at a particular SOR factor

IIT KHARAGPUR

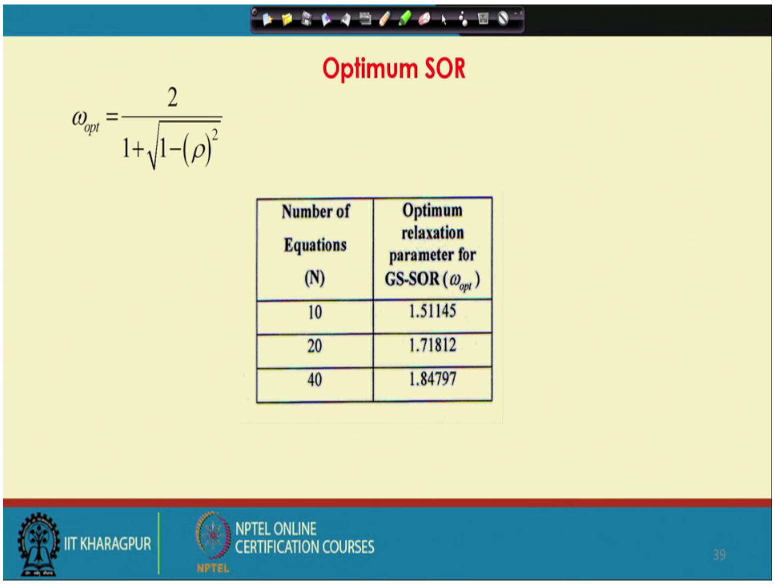
NPTEL ONLINE CERTIFICATION COURSES

Now, what we will do we will do continue the numerical experiment with what we did earlier with different value of omega for Gauss-Seidel. And what we can see is that the case for which Jacobi will 20 into 20 matrix, Jacobi needed 709 steps, Gauss-Seidel needed 374 step. For varying omega, omega 1.2, it is 258; omega 1.8 it is 133; omega 1.5 on it is 83 steps. From 374, it came down to 83 steps again omega started increasing here. So, omega optimum should be in between 1.5 to 1.9 here.

And what is happening in the spectral radius at 1.8. The cases we observed its giving least spectral radius like for Jacobi the spectral radius was 0.98 Gauss-Siedel the spectral radius is 0.97 omega 1.8 spectral radius is 0.8 and that is why convergence is faster.

We look into the larger matrix 40 into 40 matrix Gauss-Seidel is 0.99 322. And Jacobi is 0.99661 the number of iterations are 2400 or 1291. Omega 1.8 that spectral radius is 0.92 and the number of iteration is 1 6 2 is minimum. So, again if we look varying omega for a 10 by 10 matrix is the least comes at omega 1.5 for 20 by 20 the least value comes at omega 83. Here at omega 1 omega 1.8 again. So, there is an optimum SOR factor for which is number of iterations are least. And this we can calculate using the formula we have given earlier.

(Refer Slide Time: 30:20)



Optimum SOR

$$\omega_{opt} = \frac{2}{1+\sqrt{1-(\rho)^2}}$$

Number of Equations (N)	Optimum relaxation parameter for GS-SOR (ω_{opt})
10	1.51145
20	1.71812
40	1.84797

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
39

That for 10 it is 1.51 where we will get the least spectral radius for 20, it is 1.7; for 40 it is 1.84. And these are the values for which we get least spectral radius. After that the spectral radius will increase as well as number of iterations will increase. However, we can reduce the spectral radius significantly using successive over relaxation and that is how we can increase reduce the number of iterations also substantially.

So, we try to cover few important basic iterative methods. We will look into more complex iterative methods which gives faster solution. And also see in the iterative methods which work for non diagonally dominant matrices in later classes. Also another important thing which we will look is how to write computer programs using this iterative methods the problems actually which have been used for the numerical experiments in this purpose how to how can we reproduce these programs through these methods.

Thank you.