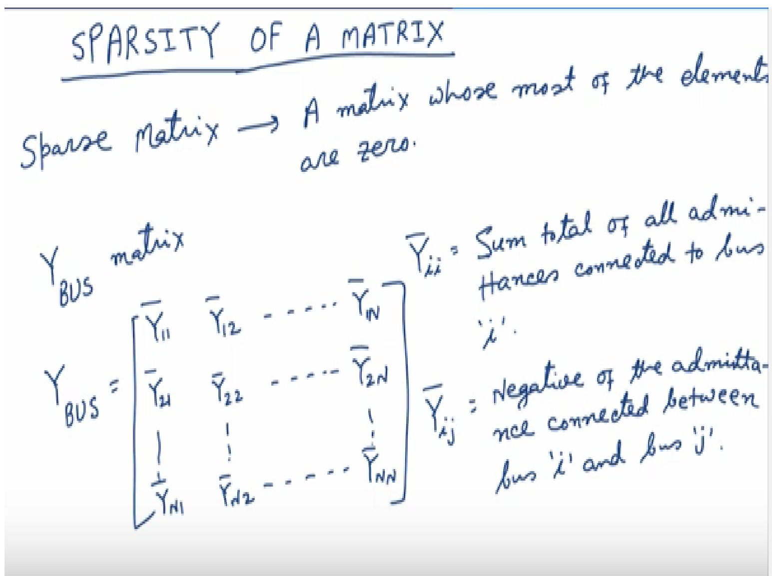
Computer Aided Power System Analysis
Prof. Biswarup Das
Department of Electrical Engineering
Indian Institute of Technology-Roorkee

Lecture - 26
Sparsity and Gaussian Elimination

Hello, welcome to this lecture on the course of computer aided power system analysis. Till the last lecture we have covered various methods of the load flow analysis. From this lecture and for the next couple of lectures, some lectures we would be looking into the concept of sparsity and essentially the solution of the linear equations. So to start with let us look at what is meant by sparsity.

(Refer Slide Time: 00:58)

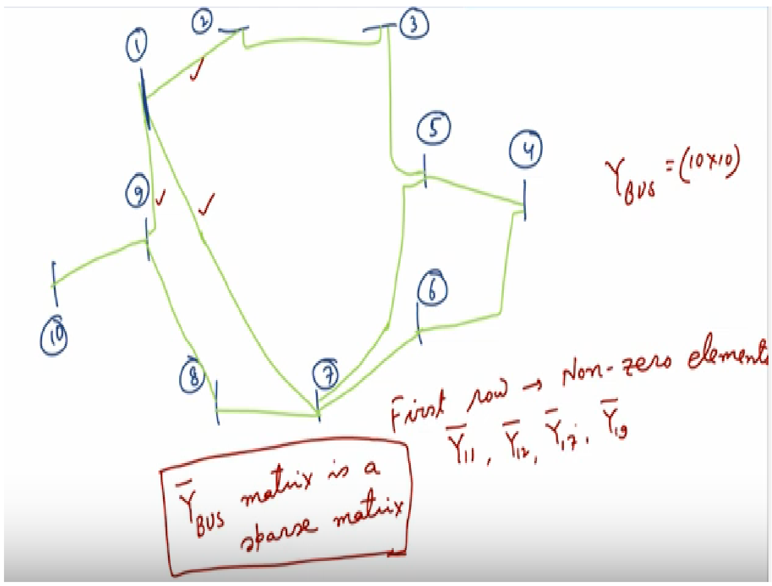


So we are looking at the concept of first, sparsity of a matrix. When you say sparsity, it is sparsity of a matrix. Now when we call that a matrix is sparse matrix, so sparse matrix is defined as a matrix, it is a matrix whose most of the elements are 0. Now the question is how this sparse matrix is really relevant to our course or to this particular course or to our discussion. So to understand that let us again revisit the Y- Bus matrix.

What is Y- Bus matrix? It simply shows the connectivity of the entire system. So Y- Bus matrix, we know that if there is N- Bus system, this is given as Y 11, Y 12….Y 1N. Then Y 21, Y 22…Y 2N and then …Y N1, Y N2…Y NN. So these are all known. And these are of course all complex quantities that we have already seen. So these are all complex quantities, that is fine. Now let us recollect that what these diagonal elements are.

So diagonal elements, if we just recollect that diagonal elements Y ii is basically sum total of all admittances connected to bus i that we already know. And Y ij, what is Y ij? Y ij is nothing but the negative of the admittance connected between bus i and bus j. So this admittance which is connected between bus i and bus j is nothing but the line admittance connected between bus i and bus j.

(Refer Slide Time: 04:28)



Now to appreciate this, let us consider a system. I am taking just an arbitrary system say, very arbitrary system I am taking, very arbitrary 1, 2, 3, 4, 5, 6, 7, 8 say 9 and 10 let us say. So there are all 10 bus system. So there are let us say this is a 10 bus system. So these are all 10 bus system, just taking an example. It is just an example. So all these are buses and the lines are let us say connected something like this. So 1 and 2. So this is it.

And then let us say from here another one, then here another one and here to here. Then here to here. Maybe here to here. Maybe here to here etc. So something like this. So now if we look at any particular bus, any bus for example let us say bus 1 if we look at bus 1, so then what is happening at bus 1. So if we look at bus 1 so there are actually 4 lines connected to it. So at bus 1 4 lines are connected to it.

This is between 1 and 2 and then between 1 and 7, then between 1 and 9. Only 3 lines are connected. So then therefore in the first row, so in the first row how many, so in the first row the nonzero elements would be Y 11. Y 11 would be obviously the nonzero elements because it would be nothing but the sum total of all this admittances; this, this and this.

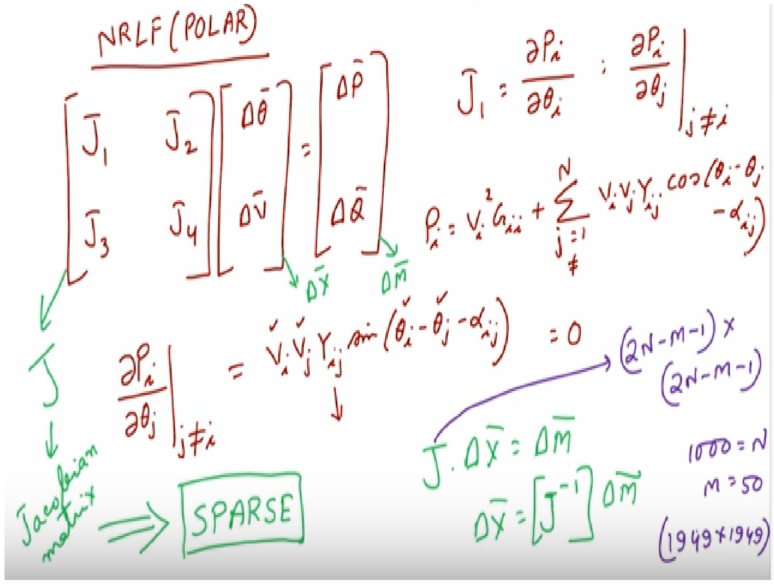
And then Y 12, it will be nothing but the negative of this. Then Y 17 and then Y 19. Now because this is a 10 bus system so then Y- Bus is basically 10 * 10. So then therefore this matrix has got 10 row and 10 columns but in the first row there are only 4 nonzero elements. Rest 6 are 0. Similarly, if we look at bus 2, there would be only 3 nonzero elements in that row. Similarly, for bus 3 there would be only 3 nonzero elements.

For bus 4 there would be 3, for bus 5 there would be 3, bus 6 3, bus 7 would be probably 5 because 1, 2, 3, 4 lines are connected. So 5 and etc. So then what we are seeing is that although that it is a 10 * 10 matrix but then in each row not even 50% of the elements is 0, right? So then therefore in this matrix basically there is a so then therefore in this matrix more than 50% elements are 0.

Because not every bus is connected to all the other bus. At best any particular bus is connected to only hardly maximum 3 – 4 buses. So then therefore in any row most of the elements are 0. So then therefore because more than 50% elements are 0 in this Y- Bus matrix so then any Y- Bus matrix of any system is always a sparse matrix. So we write, so we note that Y- Bus matrix is a sparse matrix.

Now the question is well even if this Y- Bus matrix is a sparse matrix how really does it affect us in our calculation. So that is the question. How does it really affect us in our calculation? Let us accept this fact that this Y- Bus matrix is a sparse matrix. But then how does it really affect us. So to understand that let us look again back a Newton – Raphson NRLF polar.

(Refer Slide Time: 08:55)



In fact the same observation would be true in the case of Newton – Raphson rectangular or even for FDLF but we are taking it NRLF polar as an example. So in the case of NRLF polar we have this equation J 1, J 2, J 3, J4 and then we have got delta theta, delta V. Then we have got delta P, delta Q. So these are all known. Now what is J 1? J 1 is essentially del P i/del theta i and del P i/del theta j with the rider j is not equal to i.

And if we look at the expression of these two, for example if we look at this expression of del P i/del theta j now we have got P i = the standard V i square * G ii + j = 1 to N not equal to i V i V j Y ij cosine (theta i – theta j – alpha ij). If this is the expression so then we have seen that del P i/del theta j the rider is j is not equal to i would be given by V i V j Y ij sin (theta i – theta j – alpha ij).

Now in this expression V i is nonzero because this is the bus voltage. V j is nonzero, theta is nonzero, theta j is nonzero, right? But then Y ij can be 0. Y ij is nothing but the magnitude of the ijth element of the Y- Bus matrix and we have just now seen that depending upon the fact that whether any line is really connected to another bus or not, this element Y ij may be 0, maybe nonzero.

That is if there is any line directly connected between bus i and bus j, this element Y ij would be nonzero. On the other hand if there is no direct element connected between bus i and bus j this Y ij would be 0. So then therefore depending upon the connectivity of the system, this Y ij can be either 0 or nonzero. If this Y ij is 0 then automatically this would be 0 irrespective of the value of V i, V j, theta i, theta j.

Or in other words the pattern of 0 and nonzero elements would be directly reflected into this matrix J 1, right. So then therefore if any term G ij in the Y- Bus matrix is 0 right corresponding to that term the element in this J 1 matrix would also be 0. Similarly, if we also write down this expression of del P i/del theta i we will find that depending upon the case whether Y ij would be 0 or not, that particular element would be 0, right?

Similarly, if we look at the expressions of J 2, J 3, and J 4 we will again find that depending upon the value of Y ij, that means whether there is any direct connection between bus i or bus j or not, this matrix J 2, J 3, J 4 can have many 0 elements. So then therefore because matrix J 1, J 2, J 3, J 4 individually can have many 0 elements so then overall this matrix J, this is the big matrix J, Jacobian matrix, this is the Jacobian matrix, this matrix will also have many 0 elements.

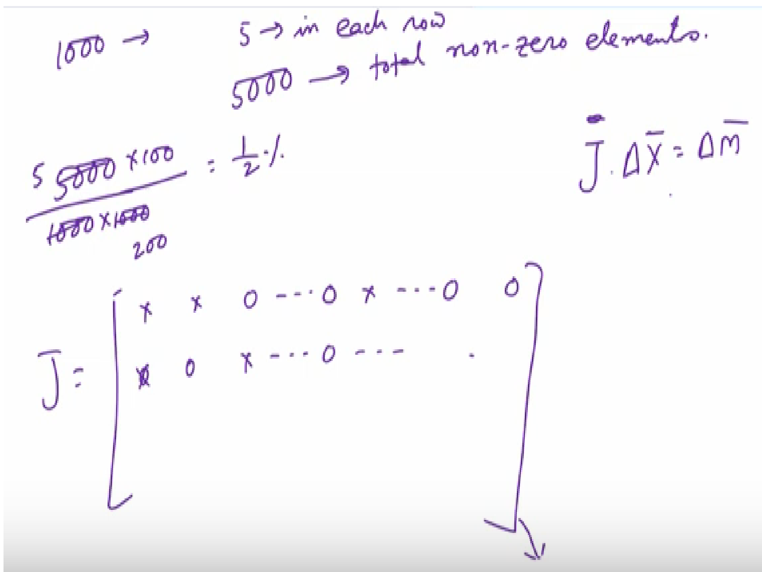
Or in other words this Jacobian matrix will also be sparse matrix. This set of equations we know that we said that this is Jacobian. This is delta X vector and this is delta M vector. So then this entire equation broadly can be written as J * delta X = delta M vector, right? Now what we have said earlier that delta X should be equal to J inverse.

That is what we have said, that J inverse * delta M. But then we need to understand something very interesting here. We have just now understood that J matrix is a sparse matrix, right? now depending upon the number of bus, this J matrix can have any dimension. For example this dimension of this J matrix we know that this dimension of J matrix is 2N – (M – 1) * 2N – (M – 1).

So this dimension we know, so then this dimension of this matrix is 2N – (M – 1) * 2N – (M – 1). So then therefore depending upon the value of N and M this J matrix can be very large. For example if N is 1000 and M is let us say just 50 so then it would be 2000 – 51. So 1949 * 1949, so many elements.

And if there is let us roughly 4 on an average, for example let us say if in this system roughly if there is let us say 4 lines are connected at each bus, so then therefore this Y-Bus matrix is roughly more than 80% sparse because in each row, because it is an N * N matrix so then therefore in the Y- Bus matrix there will be in each row is 1000 row, basically 1000 column.

(Refer Slide Time: 15:50)



And if any bus is connected to let us say at the maximum 4 or an average, so then therefore in each row there will be 5 in each row. So then therefore total 5000 total nonzero elements in the nonzero elements, right? So then the percentage nonzero elements is 5000/1000 * 1000 * 100. So this is 5 and this is 200, just half percent. So you see so then therefore this is more than 99% of this matrix is 0.

So then therefore the same thing will also be reflected here in the case of J matrix. So then therefore for this J matrix also more than 99% elements will be 0. So then we have got a matrix here where more than 99% of the elements is 0. And we are trying to store all the elements first thing. So it will require lot of memory space which is totally unnecessary because for the purpose of calculation we only need to store the nonzero elements, right? I mean 0 elements need not be stored.

Secondly, even if this matrix J is highly sparse that is even if this matrix J has got more than 99% elements to be 0 but then when we are taking the inverse of a sparse matrix this inverse of this J matrix is not at all sparse. It would be a full matrix. Full matrix means that almost all the elements would be nonzero. So then therefore we have to store this complete matrix, this particular J inverse matrix in our memory.

So then here we have got two issues, one issue is that we have to unnecessarily store the J matrix where more than 99% elements is 0 which is not necessary. Secondly, we have to also invert this big matrix which requires an highly computation intensive task and to again subsequently store this completely full matrix. Now let us understand also one very important thing.

At each and every iteration this matrix J is evaluated with the current values of V i, V j, theta i and theta j. So then therefore after we evaluate this matrix J so then therefore what we have got, we have got basically the numerical values of this J matrix. So then therefore for this J matrix, so then J matrix will have got some numerical value, some numerical value which are constant values and maybe there will be some 0 values which are constant values.

So then all these are let us say so then there will be some 0 value, some constant values etc. But then all these values are either 0 or some constant values, right? So then therefore it is a constant matrix at any particular iteration. So then what we are actually trying to do? So when we are actually trying to solve this case that J * delta X = delta M what we are actually trying to do? We are trying to solve a set of linear equation.

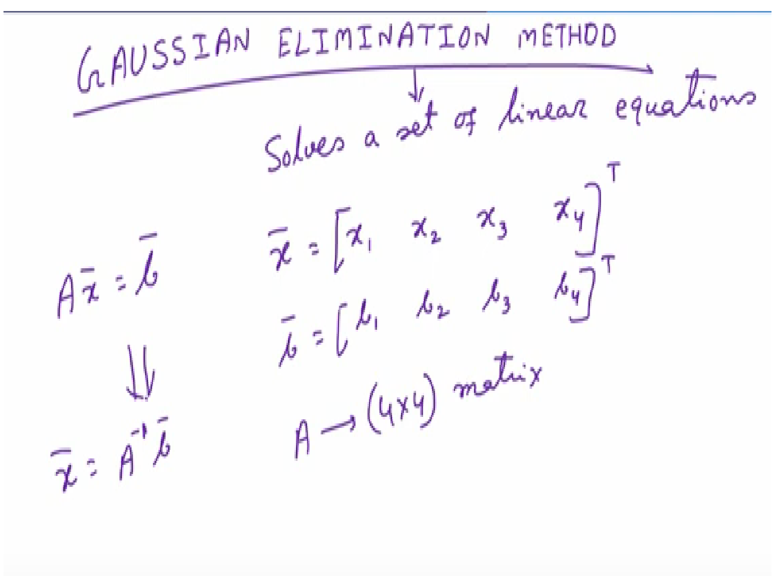
We are simply trying to solve a set of linear equation and these linear equations are basically a set of highly sparse set of linear equations, right? Now so far we have discussed that to solve this linear equation we have to take the inverse of this J matrix.

But however, if we can exploit this sparsity of this matrices and then try to solve this linear equations without involving this Jacobian matrix sorry without involving the inverse of the Jacobian matrix then our computation burden can be reduced to a large extent.

So then and precisely that is what which is being done in each and every commercial software that no commercial software really undertakes the inverse of a Jacobian matrix. Rather this set of linear equation is solved by some other method without involving this inverse of this Jacobian matrix, right?

So in this course for the next couple of lectures or more we would be looking at some method of solving a set of linear equation without involving the inverse of this Jacobian matrix and where we can also exploit the sparsity of this equations. Now one of the technique for I mean solving a set of linear equations is called, a very popular technique, Gaussian elimination method.

(Refer Slide Time: 21:32)

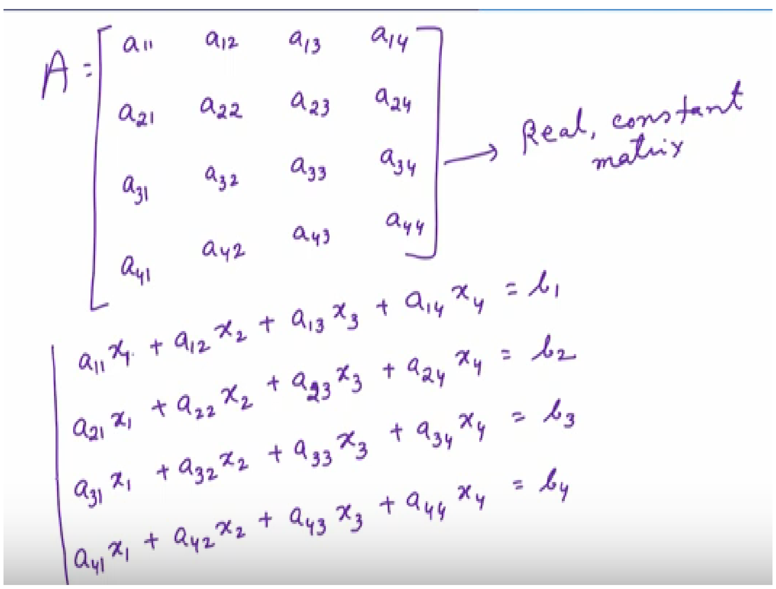


So this Gaussian elimination method is actually solves a set of linear equations, right and it is applicable to any large set that it is. Now let us look that how this particular Gaussian elimination method works. So to understand that we will look into one simple example and essential feature of this method is that it does not require the inverse of the Jacobian matrix. Now suppose now to understand that, so let us take an example.

Suppose I do have a set of equations let us say A x = b where and let us take that x is a vector of unknown quantity. So let us look at x, let us take a 4 * 4, so 4 unknown quantities; b is the vector of known quantity say let us say this is given as b 1, b 2, b 3, b4. So this is a vector of known quantities and A is a 4 * 4 matrix, constant matrix. So by the normal routine we would say that x = A inverse b.

That is what we will say immediately. But then again it involves the inversion of this matrix A and when we are talking about inversion of a matrix A we have to calculate each and every element of this matrix. So it is really a computationally intensive task. So instead of that if we apply Gaussian elimination method we really do not have to invert the matrix A. So let us look at that how do we do this.

(Refer Slide Time: 24:13)

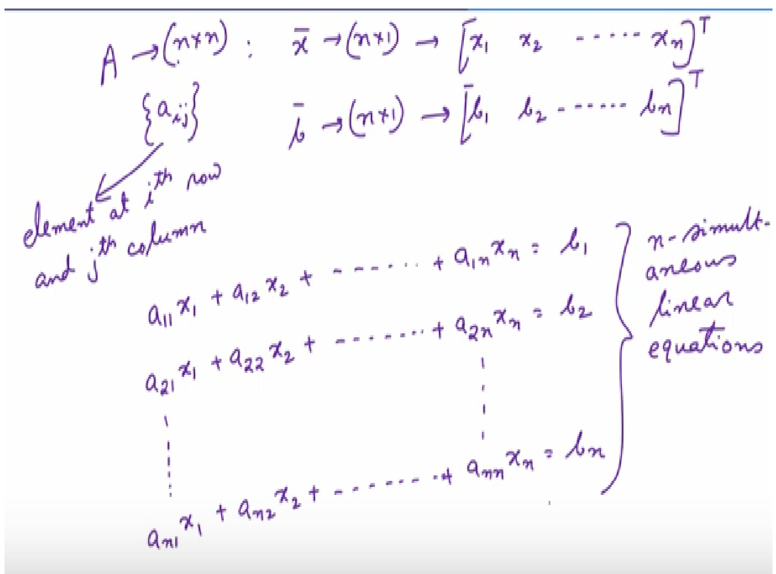


So what we do is so let us say we have A is a 11, a 12, a 13, a 14; a 21, a 22, a 23, a 24. These are the elements; a 31, a 32, a 33, a 34 and a 41, a 42, a 43, a 44. So this is the matrix A. Please note that these elements a 11 to a 44 these are all constant values, real values. So we are talking about a so we are basically saying that A is a real, this is a real constant matrix.

Real constant matrix means that where all the elements of this matrix A are real quantities. We are not talking about here imaginary quantities. So if we expand this equation so then what I will get? So then we have got a set of linear equations and those set of linear equations would be a 11 x 1 + a 12 x 2 + a 13 x 3 + a 14 x 4 = b 1. Then a 21 x 1 + a 22 x 2 + a 23 x 3 + a 24 x 4 = b 2.

Then we have got a 31 x 1 + a 32 x 2 + a 33 x 3 + a 34 x 4 that is = b 3. And the last is a 41 x 1 + a 42 x 2 + a 43 x 3 + a 44 x 4 = b 4. So if we expand all this set of equation a x = b so we get these set of equations. Now we will see that how to solve this set of equations without involving the inversion of this matrix.

(Refer Slide Time: 27:17)



Now just to let us say I mean generalize this, if let us say A is an n * n matrix, if A is an n * n matrix and x is an n * 1 vector where A is n * n matrix where this element are a ij where a ij is nothing but the element corresponding to ith row and jth column. So this is the element at ith row and jth column. And x is an n * 1 vector is x 1, x 2 to let us say x n transpose T.

And let us b is also an n * 1 vector which is given as b 1, b 2 to b n transpose T. This T basically stands for transpose. So then if we have got this so then if we expand this what I get? I will get a 11 x 1 + a 12 x 2 + …a 1n x n = b 1. Then a 21 x 1 + a 22 x 2 + … a 2n x n = b2. Then we keep on doing this; a n1 x 1 + a n2 x 2 + … + a nn x n = b n. So these are the set of n – simultaneous. So we have got n – simultaneous linear equations where n can be any value.

So these are n – simultaneous linear equations. So this set of n – simultaneous linear equations can also be solved by the Gaussian elimination method. So we will first in the next lecture, we will first look at this small example that how to solve this. And then we will simply try to generalize it for solving the set of n – simultaneous linear equations. Thank you.