Power Network Analysis

Dr. Abheejeet Mohapatra

Department of Electrical Engineering

IIT Kanpur

Week-08

Lecture-37

Hello everyone, welcome to lecture 2 of week 8 of the course Power Network Analysis, in which we are continuing our discussion on the third-to-last module, which is power flow analysis. In this discussion, we will understand the first general method of solving power flow equations, which we call the Gauss-Seidel method. In the previous lecture, we have seen at length the nature of the power flow equations, and in order to have consistency between the number of power flow equations and the number of power flow variables, which are bus voltage magnitude and phase angles at all buses in polar coordinate version or the real and imaginary parts of bus voltage phasor for all buses in rectangular version, in order to have that consistency, the number of equations and the number of unknowns, we also have understood the reason for the classification of buses into PQ buses, PV buses, and slack bus, the three major categories in which the buses need to be categorized so that power flow equations become solvable. So, what we understood in the previous lecture was that the power flow equations are non-linear equations. In fact, when we choose the polar version of coordinates for voltage phasors, the power flow equations have transcendental terms.

What are these transcendental terms? Each power flow equation would have a term that would consist of a form of this order: sin $(\theta_k - \theta_i) \cdot |V|_i \cdot |V|_k$ or cos $(\theta_k - \theta_i)$, where θ_k is the voltage phase angle of bus k, $|V|_k$ is the voltage magnitude of bus k , and a similar notion applies for bus i with respect to $|V|_i$ and θ_i. So, these terms, since they are sine and cosine functions, are called transcendental terms. In case the rectangular version of voltage phasors is considered, the power flow equations essentially have quadratic terms; that is, you would have terms like $e_i e_k$, or $f_i f_k$, or $e_i f_k$, or sort of $e_k f_i$, and often you can also have e_i^2, f_i^2 terms also present in the expression for bus i. So, because there are these transcendental equation terms or quadratic equations, the overall power flow equations are non-linear.

They cannot be solved directly the way the circuit equations based on KCL and KVL were discussed at length in the first lecture of the power flow analysis module. Since we have non-linear equations, the usual method of solving simultaneous equations, which requires the number of equations to match the number of unknowns, does not work. So far in the literature, researchers have still not been able to satisfactorily find closed-form solutions to the non-linear power flow equations. Closed form solutions, what do they mean? Closed form solutions indicate that if a set of linear equations like $Ax = b$, and A is well defined, then x can be simply obtained as $A^{-1}b$, where A is a full rank matrix. And that is how the factor of A would come in.

So, this form of the solution for $Ax = b$ indicates the closed form solution. The same, however, have not been satisfactorily done or achieved for nonlinear power flow equations, and these power flow equations must be solved for unknown voltage phasors with known data specifications such as network parameters and network topology, both of which are obtained from a parameter estimation exercise that is part of state estimation, load and generation injections, tap positions, capacitors, etc. So all those input information is known, which again we have discussed in the basis or assumptions involved in power flow. And since these are nonlinear equations, the techniques that are commonly available tend to be either iterative techniques or recursive techniques. And each of these techniques starts with a good estimate or a good guess of what these unknown voltage phasors would be.

Without an initial guess, these iterative or recursive techniques cannot work. Now, what is the difference between iterative and recursive techniques? Just to state it from a very layman perspective, iterative techniques often tend to find the values of all unknowns in one single go, whereas recursive techniques try to find the solution for each of these unknowns one at a time. We will understand this difference between iteration and recursion when we discuss the Gauss-Seidel method and other iterative techniques at length. But just to summarize, iterative techniques try to find all unknowns, which refer to all voltage phasors together in terms of voltage magnitude and phase angle. Iterative techniques try to find all unknowns in a single step, whereas recursive techniques try to find one unknown at a time; with one unknown known, the second unknown can be determined based on the first unknown, and that's how the phase-wise recursive process continues.

The techniques available for power flow analysis can be well classified into, I would say, three broad categories. The first basic one is the Gauss-Seidel method, which is a recursive method. The second most common method available in textbooks, at least, is the Newton-Raphson method, which is an iterative technique, and from an industry-grade perspective or from the application perspective, the well-known techniques to date used are the fast decoupled method, which is an extension of the Newton-Raphson method. Of late, from the research perspective, at least a decade ago, there has been this method known as the holomorphic embedded method, which is actually an extension of the Gauss-Seidel method, and this method has shown computational efficiency in terms of time saving compared to the Newton-Raphson method and the fast decoupled method, although the holomorphic embedded method has its own demerits that still have not been addressed satisfactorily from the research perspective. So from the discussion perspective, we would discuss the first three topics at length because these are well-known techniques available in different textbooks, references, and literature papers.

The holomorphic embedded method is more of a research perspective technique that still needs some maturity for its actual application, and as I mentioned, most of our industry-grade solvers are based on the fast decouple method, which we will probably discuss in next week's discussion as the time comes in. So, coming to the Gauss-Seidel method, there is a uniqueness to the Gauss-Seidel method. We have discussed these power flow equations in transcendental form or second order form, depending on the polar or rectangular version of voltage coordinates. Coincidentally, the Gauss-Seidel method does not need these transcendental or quadratic expressions. In fact, it rearranges or tries to solve the entire equation in its complex form.

Remember when we were discussing these power flow equations? I had mentioned that often these equations are segregated into real and imaginary components, specifically the complex power injection equation. From where we get real equations, we can try to solve for real variables, where voltage magnitude and phase angle are again real variables. On the other hand, the Gauss-Seidel method has this unique property that there is no need to separate out the real and imaginary components of a given complex equation. It can very well handle or solve the complex equation as it is. So, the GaussSeidel method, irrespective of the voltage phasor in polar or rectangular coordinates, rewrites, rearranges, and then applies the Gauss-Seidel method to solve the entire complex power flow equation together.

Before we apply this Gauss-Seidel method, the classification of buses into slack, PV , and PQ needs to be done a priori so that we can have consistency in terms of knowns and unknowns. So if there is an N bus network, we number those buses as $1,2,3$, till N , and for the sake of discussion that we will have, we will assume bus 1 to be the slack bus, which is a unique bus and is acting as both the angle reference and the voltage reference. For all other bus voltages, we are choosing bus 1 to be the slack bus; although the discussion would remain the same, the bus numbers may be changed as per convenience. So, if I have to understand the Gauss-Seidel method, let us look at what our complex power flow injection equation was. The basic complex power flow injection equation was based on the fact that the complex power at bus i , which is S_i, is the product of the voltage phasor at bus i and the conjugate of the current phasor at bus i.

So, if I rearrange the conjugate term, I would essentially have $S_i^* V_i$ is equal to $V_i^* I_i$, where I_i can be further expanded in terms of individual Y bus elements and all N bus voltage phasors, which again can be rewritten in a fact that. So, if I look at my complex power injection and focus on the boxed terms, which are these two terms, the first boxed term being over here and the other one over here, I can, I basically have to find voltage phasors that tend to satisfy this complex injection, and S_i^*, S_i is $P_i + jQ_i$, where j is the complex operator, the square root of minus 1 . If I rearrange this a bit in the sense that, for example, if I focus on these two boxes, then I can rewrite the entire equation as:

$$P_i - jQ_i = Y_{ii} V_i^* V_i + \sum_{\substack{k=1 \\ k \neq i}}^{N} Y_{ik} V_i^* V_k$$

I divide the entire equation by the conjugate of V_i because the conjugate of V_i is common in both of these terms. So, if I do that, then I have V_i^* divided over here, and the denominator V_i^* gets removed from here.

And I take up this second term, minus it over here, and then divide the entire expression by Y_{ii}; I would have this as my equation, which can be used for the Gauss-Seidel method:

$$S_i^* = P_i - jQ_i = V_i^* I_i = V_i^* \{Y_{i1} V_1 + Y_{i2} V_2 + \cdots + Y_{ii} V_i + \cdots + Y_{iN} V_N\}$$

$$\Rightarrow V_i = \frac{1}{Y_{ii}} \left[\frac{P_i - jQ_i}{V_i^*} - \sum_{\substack{k=1 \\ k \neq i}}^{N} Y_{ik} V_k \right]$$

So, how does the Gauss-Seidel method work? It tries to find the latest bus voltage at bus I using the previous estimates or previously known values of remaining bus voltage phasors. It includes the information about bus I itself as well as bus K. Obviously, the Y bus elements have to be known, the network topology is known, parameters are known, and these injection values are also known; then only would this equation become solvable, and that is where the premise or assumptions of power flow analysis come in. So, I have bus 1 as my slack bus; I can use this equation or solve this equation in a recursive manner, and before that, I have to find the initial choice or guess of all voltage phasors.

So, usually we call or choose the bus voltage to be a voltage that is practically realizable, practically achievable, or practically observable. In per unit analysis or the per unit system, the way we choose our base voltages and base power is expected to ensure that the voltage magnitude of a bus will typically be close to or around 1 , because that is how the base kV has been chosen. And in fact, we were discussing high voltage solutions and low voltage solutions in the previous lecture. So, the high voltage solution, which refers to a voltage value being around 1 per unit, is the practically realizable voltage value, and essentially that itself can serve as a good initial choice for voltages of all PQ buses and angle values or phase angles of all bus voltage phasors with respect to the chosen reference, where we usually choose 0 degrees or 0 radians as the angle reference for the angle reference bus, so it's common that we can choose other phase angles also as 0 degrees. For PV buses, the voltage specified for the PV bus itself can be chosen, and it is better to choose that value because this value has been obtained as per some active power loss minimization exercise, which indicates that for the PV bus, the voltage at the terminal of a synchronous generator can be controlled by proper reactive power control or power factor control.

So we call this initial choice a flat start. The flat term here refers to the fact that since the number of PV buses in the network would be very limited, not all buses would have generators connected to them. So comparatively or usually we would have a lot of PQ buses, and for all such PQ buses, we would often choose one at an angle of zero as the good initial choice for the unknown voltage phasor. Since the voltage profile is uniform with the phase angle being zero, the voltage profile itself is mostly flat, and that notion of having a flat voltage profile is associated with this flat start of the initial choice. So, given this classification and initial choice, if we want to find or make use of this Gauss-Seidel technique to find the solution to this equation in a recursive manner, what would we do? Let us start.

Since we have chosen bus 1 as the slack bus, its voltage value is known. For the slack bus, no power flow equation needs to be solved. So exclusively for bus 1, this equation need not be solved. That means for the slack bus, there is no need to solve this because the voltage for the slack bus is known.

So, what we would do is start solving this equation for the next bus onwards, which is bus number 2 .

So, if I write the equation or expand this for bus number 2, what I would observe is that V_2 would be equal to $\frac{1}{Y_{22}}$ multiplied by $\frac{P_2-jQ_2}{V_2^*}-Y_{21}V_1-Y_{23}V_3\cdots-Y_{2N}V_N$, and while I am trying to solve this equation, essentially I am trying to find the value on the left-hand side. And this left-hand side value can be known when I know the initial values of the right-hand side quantities with a flat start as my initial choice of values; I very well know what V_2, V_3, and V_N are; these need not be the final solutions. So, for a kth step or for the first step, if I associate the initial choices with a superscript, let us say 0 . Then V_1^0 doesn't make sense because V_1 refers to the slack bus.

Its voltage will always remain constant. So V_1 doesn't need any subscripts or superscripts like 0 or 1 indicating the recursion number. V_1 would remain fixed as it is chosen as the slack bus. We know what V_3^0 and V_N^0 are, which are essentially these initial choices that I have marked. And that is the reason why the 0 superscript appears there.

And after I put in all these right-hand side values with the known subscripts, I can solve the right-hand side, and the value that I would get would be related to the first recursive value of V_2. So $V_2^{(1)}$ would indicate the first recursion value of V_2 in the Gauss-Seidel method. And the same would continue for V_3, V_4, until $V_N^{(1)}$ has been obtained. All these voltages for all N voltages, when combined together the way they are being solved, we are trying to solve each voltage one at a time. So, let me show you the next slide.

So, for V_2, I would get $V_2^{(1)}$ based on the prior values of voltages known on the right-hand side. Once the first recursive value of V_2 is known, I can move on to the next bus number, which is V_3, and when I find $V_3^{(1)}$, remember $V_3^{(0)}$; it is the previous value whose initial choice has been made, but now if I look at the V_2 term, the latest value of V_2 has been chosen here. That is the difference; similarly, while for all other voltages they are still retained at their 0th recursive value, for $V_4^{(1)}$ I will be using the latest values of V_2 and V_3, and not their previous values. For other buses to come, I will still be using their previous value, and this sort of recursive process continues when I reach or find $V_N^{(1)}$.

Eventually, when I find $V_N^{(1)}$. I would have found voltages or first recursive voltage updates for all voltages, and hence essentially in $V_N^{(1)}$ on the right-hand side, I will not have any term with a subscript zero sitting on top because all voltages would have been evaluated, so the subscript would become one for all of them. With this, the first recursion would continue. So, in general, if I have to find the kth recursion solution for bus i for the kth recursion of bus i. For all prior voltages before bus i, which is bus number 1 to i minus 1, the latest bus voltage phasors to be used, which is the kth recursion itself, whereas for the following buses from i plus 1 to n, the kth minus 1 recursive value would be obtained, and all this is known or possible only when these injection values are known, so I will come to that aspect a little later. To further speed up or improve the convergence property or reduce the number of recursions, an accelerator factor can also be used as per this equation, where typically alpha is between 1 .

4 and 1.5, which dramatically reduces the number of recursions. And all this is true only when, as I said, the injections are known and P_i and Q_i are known. For a slack bus, since there is no equation involved for the PV for the slack bus, there is no problem associated with the Gauss-Seidel method for the slack bus. For the PQ bus, these P's and Q's are also known, so the same equation can be used to find the voltages. For PV buses, however, there is an issue because for PV buses, Q as such is not known explicitly.

So how do we apply the Gauss-Seidel method for a PV bus? The next few slides will explain that specifically because, for this equation to be solved, Q has to be known, and Q is not known explicitly for the PV bus. So how do we go about that? As I mentioned yesterday, if a PV bus Q is not known explicitly, the best reasonable logic would be, okay, fine, we do not know what Q is, but we definitely know for sure what our latest voltage values are for all buses. Can we not use these latest voltage values and evaluate the reactive power expression from which we get the reactive power injection, and probably that injection value could be used in this voltage equation if the PV bus remains as a PV bus? So that complexity lets us see what our complexity is. The same equation means that the way of solving this GaussSeidel method is not applicable for PV buses. So, Q is not known as a priority for PV buses.

What do we do to evaluate it in the best possible way at the latest voltage phasor values? Use the latest voltages and evaluate this expression. So, cal refers to the calculated value for a PV bus to remain a PV bus; its reactive power injection or generation should be within a minimum and maximum limit as defined by the capability curve of the synchronous generator connected to the PV bus. Using this calculated injection, we find what the generation at the PV bus is if Q_g is within its limit as per the capability curve. The PV bus remains a PV bus, which means its voltage value remains fixed; the magnitude remains fixed. So how do I find the voltage magnitude? Remaining fixed doesn't essentially mean that the phase angle should not change; the phase angle can still change in the Gauss-Seidel method.

To ensure this, what we do is directly use the Q_i^{cal} value into this expression here, solve the right-hand side, and get the latest voltage update for the PV bus. This voltage that we have obtained need not have the same voltage specification that was there for the PV bus. PV bus remains the PV bus as long as Q is within limits. If Q is within a limit, voltage magnitude should not change. So we, in a way, correct the voltage magnitude according to this correction factor.

So this correction factor, if you see V_i^{sp}, is the specified voltage magnitude of the PV bus. We are dividing the voltage phasor by the actual value of the voltage magnitude that we obtained. and multiplying by the V_i^{sp} value. So, in a way, the phase angle between V_i^{cor} refers to the corrected voltage magnitude; the phase angle of the corrected voltage and the evaluated voltage phasor do not change; only the magnitude is changing. Now this is all good when PV remains a PV bus if the condition in blue is correct.

What happens if this condition is not true? In that case, the generation or generator cannot or should not generate above its maximum limit or below its minimum limit. For the purpose of maintaining safety and security. So if Q_g is trying to violate its limit, we limit its value to the capped amount because, from a safety perspective, all control over expectations is lost. So all of it has been utilized. In case Q_g is trying to violate the lower limit according to this condition, we cap it at its lower value.

If it is trying to violate its maximum limit, we cap it to the maximum value. Once the Q_g value has been capped, we can find, depending on whether the demand is connected to the PV bus or not, the injection at this bus, which is trying to become a PQ bus, as the PQ bus

Q has to be known. So we tend to call this a violated term because the reactor generation has been violated, so PV is now becoming PQ, and we use this Q_i^{violated} as it is over here in the Gauss-Seidel update equation and then try to find the voltage phasor for this converted PV to PQ bus. Why does this conversion happen? Because the PV bus has lost its reactive power generation capability, the handling limit PQ bus is called because Q is now becoming specified for this converted PV to PQ bus. In such a conversion, voltage need not remain fixed; voltage magnitude can change, so no voltage magnitude correction is required.

The generator is not able to control or regulate its terminal voltage. The terminal voltage is now being dictated by the power flows occurring in the power network according to the lines connected to this PV to PQ converted bus. Now, once the PV bus is connected to the PQ bus, it is not necessary for it to remain a PQ bus. We always have to check whether this converted PV to PQ bus can revert to the PV status. For already converted PV to PQ buses, the conversion can only revert when the PV bus has regained its capability of maintaining the terminal voltage via reactive power control or excitation control.

So, what we do for this converted PQ bus is change its voltage magnitude first to the specified value, correct the value of the voltage phasor, use this voltage phasor to evaluate the expression of Q_i as shown over here with the corrected voltage, and find the reactive power generation from this Q expression. If this reactive power generation comes back within its limit as per the correction, then this converted PQ bus, which was actually PV , can again regain its PV status. Once PV status has been regained, voltage magnitude remains fixed. The phase angle can change according to the correction provided in slide number 8, which is this particular slide over here.

Sorry, the slide number is incorrect. This has to be slide number 10, not 8. I'll correct this when I change the slide. And if Q_g doesn't remain within its limit for this converted PV to PQ bus, the status remains as PQ, and we evaluate the voltage magnitude and phase angle as per the information given on slide number 11 , not slide number 9 . So all this goes on when there is a check of when this whole process can stop. All this process stops when the voltages do not change.

The voltage changes have now saturated. So a measure of doing that is to ensure or check whether the actual injections being calculated match their actual specifications in terms of generation and load specification. The differences are very small. The mod infinity here refers to the infinity norm. This is the infinity norm of a vector, which in a way is also the same as the maximum absolute value of a vector.

This vector here is a vector sitting inside this one. If the maximum absolute value of the vector is less than a particular tolerance, which is a numeric tolerance, we say that the method has converged. The solutions have now been obtained. And for different buses, the corresponding specifications are all defined. The same convergence can also be used for making a check whether the difference in voltage magnitude of a particular bus between two recursive iterations is less than some epsilon or not. And this needs to be checked for all buses in order to ensure these convergence criteria.

Before we go into the corresponding example, I will conclude my discussion today. I would like to spend some time understanding what the nature of convergence of the Gauss-Seidel method is. The Gauss-Seidel method can be used; it is a recursive method, but what type of convergence does it attribute? Is it linear convergence? Is it quadratic convergence? Typically, methods, numeric methods, and their assessment or performance assessment are based on what type of convergence they provide. So, to do that, I will take a very basic approach. So, basically in Gauss-Seidel, what I am trying to do is equate this term with respect to this term, and I am assuming that all on the LHS, which is the first ticked box, is known, and on the right-hand side, I am trying to find these voltages which, if found, should equate these two boxes.

And the same equation in Gauss-Seidel form is rewritten in this fashion. So, if I were to generalize it in terms of a recursive property. So, given that at kth recursion of Gauss-Seidel method. If I write a generic equation and choose my notation of variables to be x now.

So, $x^{(k+1)}$. Which is the term that states that $x^{(k+1)}$ is in a way equal to $p\left(x^{(k)}\right)/x^{(k)} + -q\left(x^{(k)}\right)$? What are $x^{(k+1)}$ and $x^{(k)}$? $x^{(k+1)}$ is identical to $V_i^{(k+1)}$, the kth plus 1 recursive update of voltage phasor i, $p\left(x^{(k)}\right)$, small $p\left(x^{(k)}\right)$, or maybe let me use a different notation. Let us say that if I use $r\left(x^{(k)}\right)$ and $t\left(x^{(k)}\right)$, okay. So, $r\left(x^{(k)}\right)$ is actually $\frac{P_i - jQ_i}{Y_{ii}}$; Q_i can also be a function of $V_i^{(k)}$, $x^{(k)}$ itself is V_i in some form of a conjugate, and $t\left(x^{(k)}\right)$ is the summation term $\frac{\Sigma\, Y_{ik}V_k}{Y_{ii}}$, the term sitting over here. So, I can correlate or rewrite this Gauss-Seidel iterative equation into this particular form. Overall, if I see and try to equate these two boxes into a form of the nature that I want to solve for x , where g is the equation that should be equal to 0 , then essentially I am trying to convert g of x into x is equal to. $r(x)/x - t(x)$. This is the reoriented form of the g of x expression, where g of x refers to the two equated boxes over here. And on this equation, if I apply the Taylor series expansion at some initial value or some value of x, y is equal to $x^{(k)}$. What do I get? I would get $x^{(k)} - \frac{r\left(x^{(k)}\right)}{x^{(k)}} + t\left(x^{(k)}\right)$, or maybe if I simplify this a bit, if I take together the series of g of x at x equal to $x^{(k)}$, what do I get? I would get $g\left(x^{(k)}\right) + \left.\frac{dg}{dx}\right|_{x^{(k)}} \left(x - x^{(k)}\right) + \left.\frac{d^2g}{dx^2}\right|_{x^{(k)}} \frac{\left(x - x^{(k)}\right)^2}{2!}$ and so on should be equal to 0 for timing if I assume the effect of first and second order terms to be absent. Even if I retain them, I can always rewrite this as $g\left(x^{(k)}\right)$ being the same as this term, which is equal to 0 , so I can substitute this as $x-x^{(k)}-\frac{r\left(x^{(k)}\right)}{x^{(k)}}+t\left(x^{(k)}\right)$.

So this is the term over here; then I have $x-x^{(k)}$, which is the term over here, multiplied by the first derivative of g at $x^{(k)}$, which would result in $1-\frac{r'\left(x^{(k)}\right)}{x^{(k)}}+\frac{r\left(x^{(k)}\right)}{\left(x^{(k)}\right)^2}+t'\left(x^{(k)}\right)$, so on, almost equal to 0 . I am neglecting the higher-order terms for the sake of simplicity in our discussion. Now, if I look at this term here, the term sitting over here, and compare it with the Gauss-Seidel update method, this is nothing but $x^{(k)}-x^{(k+1)}$. Pardon me for the subscript or superscript here; it should actually be $x^{(k+1)}+\left(x-x^{(k)}\right)$, multiplied, then I have plus $\left(x-x^{(k)}\right)\cdot\frac{r\left(x^{(k)}\right)}{\left(x^{(k)}\right)^2}-\frac{r'\left(x^{(k)}\right)}{x^{(k)}}+t'\left(x^{(k)}\right)$ should be equal to zero; $x^{(k)}$ and $x^{(k)}$ get cancelled here, so I have essentially the term like, and if I define the error at k plus one recursion to be $x-x^{(k+1)}$, then the error at the k -th recursion is $x-x^{(k)}$.

So if I use that, this term here can be written as $e^{(k+1)}+e^{(k)}$ multiplied by the, let us say if I use a form of the fact that, let us say if it is some $w\left(x^{(k)}\right)$, where $w\left(x^{(k)}\right)$ is the entire term sitting over here, equal to 0 . This indicates that $e^{(k+1)}$ is $-w\left(x^{(k)}\right)e^{(k)}$, or indirectly, $e^{(k+1)}$ is proportional to $e^{(k)}$. This proportionality, in terms of the errors in the current recursion and the previous recursion, is responsible for or indicative of the Gauss-Seidel method's linear convergence. When we discuss the Newton-Raphson method, we would see that the Newton-Raphson method has $e^{(k+1)}$ proportional to $\left(e^{(k)}\right)^2$.

This would indicate that the convergence is quadratic and not linear. So basically, the convergence is happening in a much slower fashion. It's all happening linearly. Whereas in the Newton-Raphson method, we would see this to be in a. Uh, quadratic convergence saves time; that is the benefit. So, the Gauss-Seidel method, in general, is recursive and has linear convergence; therefore, it has a slow convergence property.

Convergence of the Gauss-Seidel method becomes very difficult when the network loading is extremely high, and the voltage stability or instability problem might arise in those cases. The Gauss-Seidel method has difficulty in converging. Lines have a high R by X ratio, specifically in distribution systems. The Gauss-Seidel method need not give you the proper high-voltage solution. For networks that are highly radial and weakly meshed, as distribution systems are, the Y bus tends to become very sparse.

So basically, the Y_{ii} term becomes significantly lower compared to the off-diagonal terms. and that also results in difficulty in the convergence of the Gauss-Seidel method. Series compensation, if it has very high line reactance, can become negative, which may also lead to problems in Gauss-

Seidel convergence. We will conclude the discussion today with an example in which we take a three-bus network.

Bus 1 is designated as the angle reference bus, and bus 2 is a PQ bus. Bus 3 is a PV bus with a voltage magnitude of 1.05 and a power of 1 PU at bus 3 ; reactive regeneration should be within minus 1 and plus 1 PU. So how do we solve the Gauss-Seidel method for one recursion, choosing unknown flat states as unknown voltages for the initial guess? Since bus 1 is the angle reference bus, its voltage magnitude is given as 1 p.u. We first find the Y bus for this 3-bus matrix. The three bus reactances, all of which are given on a common PU. So we can evaluate the Y bus and get this thing confirmed. Bus specifications, as per the classifications, are also known. Since bus 1 is a slack bus, no equation needs to be solved for it.

We can start with bus 2 as the iterative update for bus 2 itself. So if you put in those initial values of flat stat, 1.05 PU remains the same for V_3 because V_3 is a PV bus. Then we get the first update of the voltage at bus 2 as this complex number. When we go for bus 3 , since it is a PV bus, we need to check whether its generation, which is based on the latest voltage values, is within its limits or not. The generation value is not within its limit because the maximum value is 1 pu , so we cap Q_g at one PU and recalculate the corresponding injection, which turns out to be equal to, since there is no load, so Q_3 in itself becomes one PU; we use this one PU over here, find the voltage value, and the magnitude is no more equal to 1.05 ; no correction is needed, and the bus has become a PQ bus. And this process, when it continues, eventually shows that after 19 recursions, the solution process converges, which means the mismatches have turned down to zero. The voltage of V_2 and V_3 is in this order, where V3 bus 3 is still not a PV bus. In the next discussion, we will take up our next technique, a popular technique known as the Newton-Raphson method, with applications for voltages in polar coordinates, and we will try to understand this more in terms of examples. Thank you.