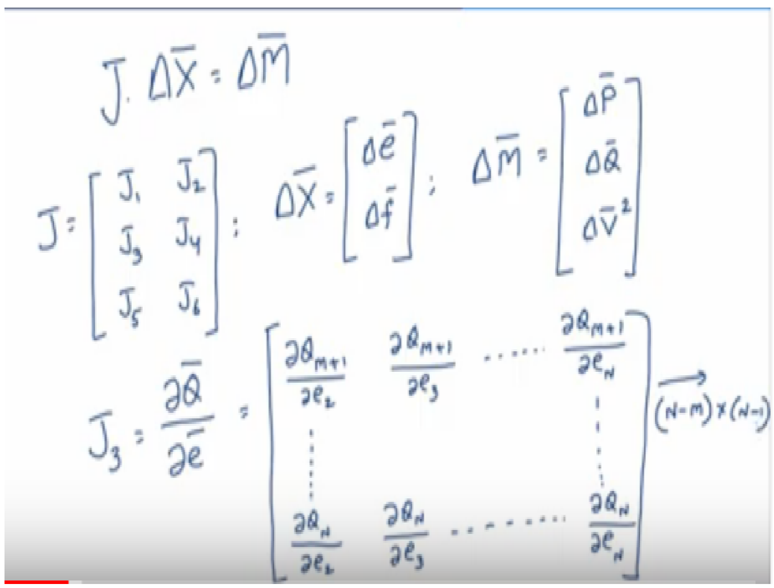
Computer Aided Power System Analysis
Prof. Biswarup Das
Department of Electrical Engineering
Indian Institute of Technology-Roorkee

Lecture - 18
NRLF in Rectangular Coordinate (Contd..)

Welcome to the another lecture of this course computer aided power system analysis. We have been discussing the Newton – Raphson load flow in the rectangular coordinate. So, so far we have been discussing.

(Refer Slide Time: 00:46)



We have discussed that in the Newton – Raphson rectangular coordinate the equations of load flow can be represented by this delta X delta M where matrix J is constituting of 6 sub matrices J 1, J 2, J 3, J 4, J 5, J 6. Vector X that is the correction vector is constituting of two sub vectors delta e and delta f. And delta M that is the mismatch vector is constituting of 3 sub vectors that is delta P, delta Q, and delta V square.

Now we have been also looking at the expressions of all the elements of this matrices J 1, J 2, J 5, and J 6. So to define this matrix J completely we also need to find out the expressions of the elements of the matrices J 3 and J 4. So today initially we would be doing that. Now matrix J 3 is essentially we know that it is actually del Q/del e because from here we can see that matrix J 3 is actually connecting delta Q vector with the delta e vector.

So it would be delta Q M + 1/del e 2, delta Q M + 1/delta e 3…delta Q M + 1/delta e N. And then up to we will go up to this delta Q N/delta e 2, delta Q N/delta e 3 then …delta Q N/delta e N. So this is the structure of the matrix and we have also seen that the dimension of this matrix would be (N – M) * (N – 1). So this is a rectangular matrix. So then therefore according to our own parlance it will have something called self-term.

(Refer Slide Time: 03:47)

Self-term: $\frac{\partial Q_i}{\partial e_k}$; $i = M+1, \cdots N$; $k = 2, \cdots N$; $k = i$

Non-self-term: $\frac{\partial Q_i}{\partial e_k}$; $i = M+1, \cdots N$; $k = 2, \cdots N$; $k \neq i$

We have,

$$Q_i = \sum_{k=1}^{N}\left[f_i\left(e_k g_{ik} - f_k b_{ik}\right) - e_i\left(e_k b_{ik} + f_k g_{ik}\right)\right]$$

$$= -\left(e_i^2 + f_i^2\right) b_{ii} + \sum_{\substack{k=1\\ \neq i}}^{N}\left[f_i\left(e_k g_{ik} - f_k b_{ik}\right) - e_i\left(e_k b_{ik} + f_k g_{ik}\right)\right]$$

Non-self-term

$$\boxed{\frac{\partial Q_i}{\partial e_k};\ k \neq i = \left(f_i g_{ik} - e_i b_{ik}\right)} \rightarrow \text{non-self term for } J_3$$

Self-term, where it is delta Q i/delta e k for i = varying from M + 1 to N, k varying from 2 to N and k = i. And we say that it is non-self-term. It is again denoted as this delta Q i/del e k. The conditions are same. The range of this different indices are same. Only difference is that now k is not equal to i. So now therefore you have to now find out this expressions.

So to do that we need to look at the expressions of Q what we have already defined or rather derived. So the expression of Q is given by f i(e k g ik – f k b ik). So we have Q i is k = 1 to N f i e k sorry e k g ik – f k b ik. Then (e k g ik – f k b ik) – e i (e k b ik + f k g ik). So this is the expression. Now as before we will take out the ith term separately. So then when I make k = i so what becomes?

This becomes f i e i g ii and this becomes – f i square b ii and this becomes – e i square b ii and this becomes – e i f i g ii. So then therefore this f i e i g ii and this – e i f i g ii will cancel out. So what will remain is basically – f i square b ii – e i square b ii. So if I am to take them together I will get that is e i square * f i square * b ii + k = 1 to N not equal to i. then f i (e k g ik – f k b ik) – e i that entire expression (e k b ik + f k g ik).

So this is the expression. So now we first do the non-self-term. So non-self-term would be delta Q i/delta e k; k is not equal to i. And we have already written the ranges of i and k here. So then therefore we are not really repeating them unnecessarily. So that should be equal to so now of course this particular term would be 0. That is if I do take the derivative of this term with respect to k that would be 0.

And in this entire expression there will be only one term which will have e k. So it would be f i g ik, this would be 0 and – e i b ik. So it would be f i g ik from this f i g ik – e i b ik. So that would be the expression. This would be 0, this would be 0. So non-self-term is this. Now we go to the self-term.

(Refer Slide Time: 09:17)

Self-term: $\frac{\partial Q_i}{\partial e_k}$; $k = i$; $\Rightarrow \frac{\partial Q_i}{\partial e_k} = -2e_i b_{ii} + \sum_{\substack{k=1 \\ \neq i}}^{N} -(e_k b_{ik} + f_k g_{ik})$

$$\Rightarrow \boxed{\left.\frac{\partial Q_i}{\partial e_k}\right|_{k=i} = -2e_i b_{ii} - \sum_{\substack{k=1 \\ \neq i}}^{N} (e_k b_{ik} + f_k g_{ik})} \rightarrow \text{self-term for } J_3$$

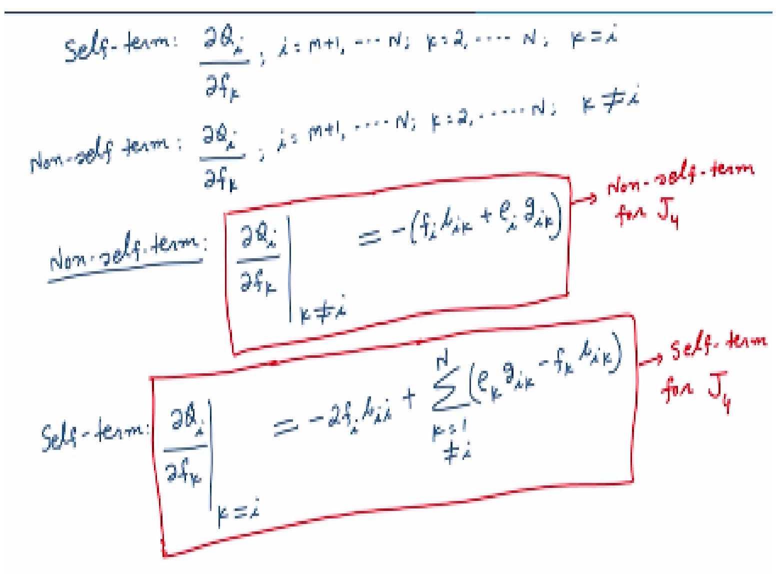
$$J_4 = \frac{\partial \bar{Q}}{\partial \bar{f}} = \begin{bmatrix} \frac{\partial Q_{m+1}}{\partial f_2} & \frac{\partial Q_{m+1}}{\partial f_3} & \cdots & \frac{\partial Q_{m+1}}{\partial f_N} \\ \vdots & & & \vdots \\ \frac{\partial Q_N}{\partial f_2} & \frac{\partial Q_N}{\partial f_3} & \cdots & \frac{\partial Q_N}{\partial f_N} \end{bmatrix} \rightarrow (N-m) \times (N-1)$$

So self-term is again delta Q i/delta e k; k = i. And we have already written the range of this i and k. So then therefore we are not really repeating them. So that would be equal to delta Q i/delta e k is, so from this – 2e i bii; so – 2e i b ii + k = 1 to N not equal to i and I am trying to do it with respect to e i. So then therefore this term would be all cancel out. So only this term will remain. So e i, so – e k b ik – f k g ik.

So it would be – e k b ik + f k g ik. So then therefore what we have is delta Q i/delta e k for k = i. So that would be equal to – 2 e i b ii – k = 1 to N not equal to i e k b ik + f k g ik. So this is the self-term. So this is the self-term for J 3. And this is the non-self-term for J 3. So now only one Jacobian sub matrix is now left J 4. So let us do that, J 4. J 4 = del Q/del f.

So it would be again del Q M + 1/del f 2, del Q M + 1/del f 3…del Q M + 1/del f N. It would go up to del Q N/del f 2, del Q N/del f 3 and it would be del Q N/del f N. So again its dimension is again (N – M) * (N – 1). So again it is rectangular matrix. So it will have a self-term.

(Refer Slide Time: 13:23)



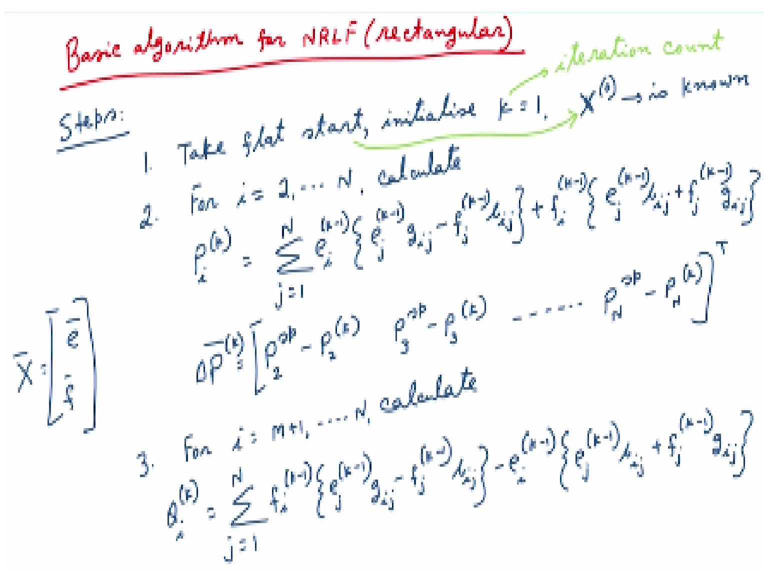
It would be delta Q i/delta f k i varies from M + 1 to N; k varies from 2 to N and k = i and then as usual non-self-term is again delta Q i/delta k. Again i varies, the range of variation of i and k are same. Only difference is that k is not equal to i. So then therefore we now derive this. First we derive the non-self-term. So we derive first non-self-term. So non-self-term is del Q i/del f k with the condition that k is not equal to i.

So that would be equal to if we go to this expression we are not trying to take the partial derivative of this expression Q i with respect to f k where k is not equal to i. So this term would be 0 and we will get 1 term from this that is – f i b ik and one term from this that would be – e i g ik. This term also would be 0, this term also would be 0. So it is – f i b ik and – e i g ik. So it would be – (f i b ik + e i g ik). So this would be the non-self-term.

So we write it, so this is non-self-term for matrix J 4. And the last one is self-term. Self-term is del Q i/del f k with the condition that k = i. So that would be equal to so first is that we have to take the derivative with respect to f i because we are actually taking the partial derivative of Q i with respect to f i. So from here I will get 2 F i b ii minus that is. So it is – 2f i b ii + k = 1 to N not equal to i. So we are trying to do with f i.

So only this much; e k g ik – f k b ik. So e k g ik – f k b ik. So that would be the self-term. **So we,** so this is actually self-term for J 4. So now we have essentially derived all the elements of this Jacobian matrix J and now we are in a position to describe the very basic skeleton algorithm of the Newton – Raphson rectangular technique.

(Refer Slide Time: 18:14)



So therefore we write basic algorithm of, once you understand the basic algorithm then we would be in a position to understand the algorithm which converts everything for

NRLF rectangular. So the steps would be, steps: 1. Take flat start. We have already discussed what is meant by flat start. So then therefore we need not discuss it and initialize let us say k. This is the iteration count; k is here.

We must say that it is iteration count. So this is the iteration count. Now here only one very minor point which needs to be discussed. We have already seen that when we say that we are taking flat start. So for this PV buses we take the magnitudes as the specified voltage and for the PQ buses we take the magnitude as 1.0 per unit. But for all these buses we take the angles to be 0.

So then therefore in this rectangular coordinate for this PV buses e i would be equal to nothing but the specified voltage magnitude; f i would be 0 because this angle is 0 and for all this PQ buses e i would be equal to 1.0 and the f i would be 0. So this is the only thing we need to understand. Now then for i = 2 to N calculate P i k = j = 1 to N. So e i(k) then e j(k) g ij – f j(k) b ij – f i(k) sorry + f i(k) e j(k) b ij + f j(k) g ij.

There is some small mistake here. Everywhere there is a mistake. This should be all (k – 1). So when we are evaluating P i(k), so then therefore we need to use the previously known value, the last known value of all this voltage real and real part in this expression. Then calculate delta P vector as P 2 specified – P 2 (k), P 3 specified – P 3(k)…P N specified – P N(k) T. Then what we do? For i = M + 1 to N calculate Q i(k) = j = 1 to N.

Then we, f i * e j g ij – f j b ij. So f i e j g ij – f j b ij. Then – e i (k – 1) e j(k – 1) b ij + f j(k – 1) g ij. So again the same principle. We are calculating Q i(k) using the last known values of this real and imaginary part of the voltages.

(Refer Slide Time: 24:56)

$$\Delta \bar{Q}^{(k)} = \left[Q_{M+1}^{sp} - Q_{M+1}^{(k)} \quad Q_{M+2}^{sp} - Q_{M+2}^{(k)} \quad \cdots\cdots \quad Q_N^{sp} - Q_N^{(k)} \right]^T$$

4. For $i = 2, \cdots M$, calculate

$$\{V_i^{(k)}\}^2 = \{e_i^{(k-1)}\}^2 + \{f_i^{(k-1)}\}^2$$

$$\{\Delta \bar{V}^{(k)}\}^2 = \left[(V_2^{sp})^2 - \{V_2^{(k)}\}^2 \quad (V_3^{sp})^2 - \{V_3^{(k)}\}^2 \quad \cdots\cdots \quad (V_M^{sp})^2 - \{V_M^{(k)}\}^2 \right]^T$$

5. $$\Delta \bar{M}^{(k)} = \begin{bmatrix} \Delta \bar{P}^{(k)} \\ \Delta \bar{Q}^{(k)} \\ \{\Delta \bar{V}^{(k)}\}^2 \end{bmatrix} \rightarrow (2N-2) \times 1$$

Then we form delta Q vector as Q M + 1 specified – Q M + 1 (k). Then Q M + 2 specified – Q M + 2 (k)…Q N specified – Q N (k) transpose. Then in the next step what we do, for i = 2 to N calculate V i (k) square = e i (k – 1) whole square + f i (k – 1) whole square and form the delta V square vector as V 2 specified square – V 2 (k) square. Then V 3 specified square – V 3 (k) square…V M.

Because we are doing only up to the PV buses. V M specified square – V M (k) whole square transpose T. So in step 5 so what you have got? Different sub vectors of this mismatch vector. So then after that we form delta m vector at kth iteration as delta P vector at kth iteration we should also write k, we should also write here k and we should also write here k.

In fact we should write here more elegantly delta Q k and delta V (k) whole square. We know that this dimension of this matrix rather vector is (2N – 2) * 1.

(Refer Slide Time: 28:29)

6. Let $\Delta\bar{m}^{(k)} = \left[\Delta m_1^{(k)} \quad \Delta m_2^{(k)} \quad \cdots\cdots \quad \Delta m_{2N-2}^{(k)}\right]^T$

7. Calculate $e_i^{(k)} = \left|\Delta m_i^{(k)}\right| \quad \forall i = 1, \ldots (2N-2)$

8. Calculate $e_{rr}^{(k)} = \max\left[e_1^{(k)}, e_2^{(k)} \ldots\ldots e_{2N-2}^{(k)}\right]$

9. If $e_{rr}^{(k)} < \epsilon$, then stop; Else, go to step 10.

10. Evaluate $[J]\Big|_{\bar{e}^{(k-1)}, \bar{f}^{(k-1)}}$

11. Calculate $\Delta\bar{x}^{(k)} = [J]^{-1}\,\Delta\bar{m}^{(k)}$

12. update $\bar{X}^{(k)} = \bar{X}^{(k-1)} + \Delta\bar{x}^{(k)}$, $k = k+1$, and go to step 2.

So then therefore we can say that we can say let delta M (k) vector is given by delta M 1 (k) delta M 2 (k), all this component taken together delta M 2N – 2 (k) transpose. So then what we do? Then we calculate e i (k) as delta M i (k) for i = 1 to (2N – 2). Then we calculate e error at k as max of e 1 (k), e 2 (k). Please remember in this step everything is positive. So there will be some maximum quantity which would be indeed a positive.

So we take the maximum of them. If error (k) is less than some epsilon then stop else go to step 10. What we do in 10? Evaluate matrix J, evaluate matrix J with e vector of e and f known at (k – 1)th iteration. So evaluate this matrix j with this vector e and f known at (k – 1)th iteration. This is not minus, this is actually, this is not really minus.

This is actually, this is a vector notation here and this (k – 1) denotes that this vector e is known corresponding to (k – 1)th iteration and this (k – 1) denotes that this vector f is known at (k – 1)th iteration. And then what we do? Calculate delta X vector k as J inverse into delta M (k). Then update as X (k) = X (k – 1) + delta X (k) and go to step 2.

And here we should write initialize and when we do initialize then after taking flat start we actually know what is, X vector is known. And what is X vector? X vector is vector e and vector f. So then therefore we first take the flat start from here X vectors at the initial condition that is this 0th iteration is known and after that we start our operation. So after we take this initial condition this is known.

So then therefore after that we straightaway keep on doing this iteration. After that it is straightforward, we calculate delta P (k). we calculate delta Q (k), we calculate delta V (k) square, we form this delta M matrix. Then we check for convergence. If it is within some threshold value then we stop. We say that it is that my solution is obtained. Otherwise we evaluate this Jacobian matrix, take its inverse and calculate this correction vector.

At that correction vector with the previously known X vector, X vector is nothing but the solution vector. Remember when k = 1 it is 0. So then therefore we know this and we have just now calculated, so we know this. And then we go to step 2. And here also go to step 2. Here also you should write k = (k + 1) and go to step 2. So this is the very basic algorithm of this Newton – Raphson rectangular method.

In this algorithm you must have noticed that we have not considered the case when there would be some violation of the reactive power limits of the generator. So in the next lecture we would be considering that and then also we will be looking at a simple example for Newton – Raphson rectangular method. Thank you.