LOAD FLOW METHOD FOR RADIAL DISTRIBUTION NETWORKS

2.1 Introduction

Load flow is very important and fundamental tool for the analysis of any power system and is used in the operational as well as planning stages. Certain applications, particularly in the distribution automation and optimization of a power system, require repeated load flow solutions. In these applications, it is very important to solve the load flow problem as efficiently as possible.

Distribution networks, with radial configurations and wide ranging resistance and reactance values are inherently ill-conditioned, and conventional Newton Raphson [4] and fast decoupled load flow [7] methods are not giving solution for solving such networks. Special load flow solution methods and modified versions of the conventional methods have, however, been suggested for solving such ill-conditioned networks [18,48]. In view of the topological specialty of distribution networks, and non- applicability of the simplifying assumption of a decoupled Jacobian matrix normally applicable to the transmission networks, researchers associated with the distribution systems have proposed several special load flow solution techniques [12, 50].

Kersting and Mendive [12] and Kersting [31] have developed a load flow method for solving radial distribution networks by converting ladder network theory into a working algorithm. Baran and Wu [54] have obtained the load flow solution of radial distribution networks by the iterative solution of the three fundamental equations representing real power, reactive power and voltage magnitude. They have computed system Jacobian matrix using chain rule in their method. They have also proposed decoupled and fast-decoupled distribution load flow algorithms. Chiang [80] has also proposed three different algorithms for solving radial distribution network based on the method of Baran and Wu [54]. He has proposed decoupled, fast decoupled and very fast-decoupled distribution load flow algorithms. In fact decoupled and fast-decoupled distribution load flow algorithms proposed by Chiang are similar to that of Baran and Wu [54]. However, very fast decoupled distribution load flow proposed by Chiang is very attractive because it does not require any Jacobian matrix construction and factorization but more computations are involved because it solves three fundamental equations representing real power, reactive power and voltage magnitudes.

In this chapter a simple method of load flow technique is proposed for the solution of radial distribution networks. The proposed method involves only the evaluation of a simple algebraic expression of receiving end voltages. Loads, in the present formulation, have been represented as constant power loads.

Assumptions:

1. It is assumed that the three-phase radial distribution networks are balanced and are represented by their equivalent single-phase representation.
2. Line shunt capacitance is negligible at the distribution voltage levels as is found in most practical cases (shunt capacitor banks are taken as loads).

In section 2.2, it is dealt with mathematical formulation of the proposed load flow method. Methodology of identification of nodes and the algorithm for the proposed method are described in Section 2.3. A simple method of load flow calculation has been proposed for radial distribution networks and discussed in Section 2.4. The above algorithm is illustrated with three examples in Section 2.5. Conclusions of this chapter are presented in section 2.6.

2.2 Mathematical Formulation

In any radial distribution network, the electrical equivalent of a branch jj, which is connected between nodes m1, m2 having a resistance $r_{(jj)}$ and inductive reactance $x_{(jj)}$ is shown in **fig.2.1**.

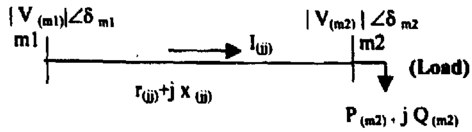

Fig.2.1: Electrical equivalent of a typical branch jj

From **fig.2.1**, the current flowing through above branch $I_{(jj)}$ is given by,

$I_{(jj)} = [\, |V_{(m1)}| \angle\delta_{m1} - |V_{(m2)}| \angle\delta_{m2} \,] / [\, r_{(jj)} + j\, x_{(jj)} \,]$ -------(2.1)

$= [\, P_{(m2)} - jQ_{(m2)} \,] / V^*_{(m2)} \angle\delta_{m2}$ -------(2.2)

Where

$P_{(m2)}$ = sum of the real power loads of all the nodes beyond node m2 plus the real power load at node m2 itself plus the sum of the real power losses of all the branches beyond node m2.

$Q_{(m2)}$= sum of the reactive power loads of all the nodes beyond node m2 plus the reactive power load at node m2 itself plus the sum of the reactive power losses of all the branches beyond node m2.

$|V_{(m1)}| \angle\delta_{m1}$, $|V_{(m2)}| \angle\delta_{m2}$ are the voltage magnitudes and corresponding phase angles at nodes m1 and m2 respectively.

From eqns. (2.1) and (2.2)

$$\frac{P_{(m2)} - j\, Q_{(m2)}}{|V_{(m2)}| \angle -\delta_{m2}} = \frac{|V_{(m1)}| \angle\delta_{m1} - |V_{(m2)}| \angle\delta_{m2}}{r_{(jj)} + j\, x_{(jj)}}$$

$$|V_{(m1)}|\, |V_{(m2)}| \angle(\delta_{m1} - \delta_{m2}) - |V_{(m2)}|^2 = \{P_{(m2)} - j\, Q_{(m2)}\}\{r_{(jj)} + j\, x_{(jj)}\}$$

Separating the real and imaginary parts,

The real part is

$|V_{(m1)}|\, |V_{(m2)}| \cos(\delta_{m1} - \delta_{m2}) = |V_{(m2)}|^2 + \{P_{(m2)}\, r_{(jj)} + Q_{(m2)}\, x_{(jj)}\}$ ------(2.3)

and the imaginary part is

$|V_{(m1)}|\, |V_{(m2)}| \sin(\delta_{m1} - \delta_{m2}) = \{P_{(m2)}\, x_{(jj)} - Q_{(m2)}\, r_{(jj)}\}$ ------(2.4)

Squaring and adding eqns. (2.3) and (2.4), and then simplifying,

$$|V_{(m2)}|^4 - b_{(jj)}|V_{(m2)}|^2 + c_{(jj)} = 0 \qquad \text{------(2.5)}$$

Where

$$b_{(jj)} = |V_{(m1)}|^2 - 2\,P_{(m2)}\,r_{(jj)} - 2\,Q_{(m2)}\,x_{(jj)} \qquad \text{------(2.6)}$$

$$c_{(jj)} = \{P^2_{(m2)} + Q^2_{(m2)}\}\,\{r^2_{(jj)} + x^2_{(jj)}\} \qquad \text{------(2.7)}$$

The four possible solutions for the receiving end voltage $|V_{(m2)}|$ from eqn.(2.5) are

(i) $[\frac{1}{2}[b_{(jj)} - \{b^2_{(jj)} - 4c_{(jj)}\}^{1/2}]]^{1/2}$

(ii) $-[\frac{1}{2}[b_{(jj)} - \{b^2_{(jj)} - 4c_{(jj)}\}^{1/2}]]^{1/2}$

(iii) $-[\frac{1}{2}[b_{(jj)} + \{b^2_{(jj)} - 4c_{(jj)}\}^{1/2}]]^{1/2}$

(iv) $[\frac{1}{2}[b_{(jj)} + \{b^2_{(jj)} - 4c_{(jj)}\}^{1/2}]]^{1/2}$

It is found for realistic systems, when P, Q, r, x and V are expressed in p.u., $b_{(jj)}$ is always positive because the term $2\{P_{(m2)}\,r_{(jj)} + Q_{(m2)}\,x_{(jj)}\}$ is extremely small as compared to $|V_{(m1)}|^2$. In addition the term 4 $c_{(jj)}$ is negligible compared to $b^2_{(jj)}$. Therefore $\{b^2_{(jj)} - 4c_{(jj)}\}^{1/2}$ is nearly equal to $b_{(jj)}$ and hence the first two solutions of $|V_{(m2)}|$ are nearly equal to zero and the third solution is negative and hence not feasible. The fourth solution of $|V_{(m2)}|$ is positive and hence it is only the possible feasible solution. Therefore, the possible feasible solution of eqn. (2.5) is

$$|V_{(m2)}| = [\frac{1}{2}[b_{(jj)} + \{b^2_{(jj)} - 4\,c_{(jj)}\}^{1/2}]]^{1/2} \qquad \text{------(2.8)}$$

The corresponding phase angle of $V_{(m2)}$ can be calculated from eqn.(2.3) and eqn. (2.4) as

$$\delta_{m2} = \delta_{m1} - \tan^{-1}\frac{(P_{(m2)}\,x_{(jj)} - Q_{(m2)}\,r_{(jj)})}{|V_{(m2)}|^2 + P_{(m2)}\,r_{(jj)} + Q_{(m2)}\,x_{(jj)}} \qquad \text{------(2.9)}$$

The real and reactive power losses in any branch jj are

$$LP_{(jj)} = I^2_{(jj)}\,r_{(jj)} = \frac{r_{(jj)}\,\{P^2_{(m2)} + Q^2_{(m2)}\}}{|V_{(m2)}|^2} \qquad \text{----(2.10)}$$

$$LQ_{(jj)} = I^2_{(jj)}\, x_{(jj)} = \frac{x_{(jj)}\{P^2_{(m2)} + Q^2_{(m2)}\}}{|V_{(m2)}|^2} \quad \text{-----(2.11)}$$

The total real and reactive power loss are

$$TLP = \sum_{jj=1}^{LN1} I^2_{(jj)}\, r_{(jj)} \quad \text{------ (2.12)}$$

$$TLQ = \sum_{jj=1}^{LN1} I^2_{(jj)}\, x_{(jj)} \quad \text{------(2.13)}$$

Initially, if $LP_{(jj)}$ and $LQ_{(jj)}$ are set to zero for all jj, then initially estimate of $P_{(m2)}$ and $Q_{(m2)}$ will be the sum of loads of all the nodes beyond node m2 plus the load at node m2 itself.

2.3 Identification of Nodes Beyond all the Branches

Before giving the detailed algorithm, it is discussed here regarding the methodology of identifying the nodes and branches beyond a particular node, which will help in finding the total load feeding through that particular node. The method is illustrated with a sample network shown in **fig.2.2.** Branch number, sending-end and receiving-end nodes of this network are given in **Table 2.1**.

Consider the first branch i.e., jj=1, the receiving end node of branch 1 is 2 (from **Table 2. 1**), therefore, IB(1, ip+1) and IE(1,ip+1) will help to identify all the branches and nodes beyond node 2 and node 2 itself.

This will help to find the exact load feeding through node 2. Similarly consider branch 2, i.e., jj=2,the receiving end node of branch 2 is3. Therefore IB(2,ip+1) and IE(2, ip+1) will identify nodes and branches beyond node 3 and node 3 itself. This will help to compute the exact load feeding through node 3. For each node and branch identification 'ip' will be incremented by 1. Note here that before identification of nodes and branches beyond a particular node, 'ip' has to reset to be zero.

For jj=1, (first branch in **fig.2.2, Table 2.1**) IR(jj)= IR(1)=2 check whether IR(1)=IS(i) or not, for i=2,3,....., LN1. It is seen that IR(1)=IS(2)=2, IR(1)=IS(5)=2, IR(1)=IS(7)=2, corresponding receiving end nodes are IR(2)=3, IR(5)=9 and IR(7)=6.

Therefore IB(1, 1)=1, IE (1, 1)=2, IB(1, 2)=2, IE(1, 2)=3, IB(1, 3)=2, IE(1, 3)=9, IB(1, 4)=2, IE(1, 4)=6.

The node IB(1, 1) will be missed out, because it is required to identify the nodes and branches, which are beyond node IR(1) and store the receiving end node in the name of a variable, say kk(ip), i.e., kk(1)=2, kk(2)=3, kk(3)=9 and kk(4)=6.

Note that there should be no repetition of any branch or node while identifying nodes and branches and this logic has been incorporated in the proposed algorithm (**Algorithm 2. 3.1**).

From the above discussion, it is seen that node 2 is connected to nodes 3, 9 and 6 and the corresponding branches of branch 2, (2 $\rightarrow$ 3), branch 5, (2 $\rightarrow$ 9) and branch 7, (2 $\rightarrow$ 6). Similarly the proposed logic will identify the nodes and branches, which are, connected to nodes 3, 9 and 6. First, it will check whether node 3 appears in the left hand column of **Table 2.1**. It is seen that node 3 is connected to nodes 4 and 11 (branch 3 and 10). Therefore IB(1, 5)=3, IE(1, 5)=4, IB(1, 6)=3, IE(1, 6)=11, and kk(5)=4, kk(6)=11. Then it will check whether the node 9 appears in the left hand column of Table 2.1. It is seen that node 9 is connected to node. Therefore, IB(1, 7)=9, IE(1, 7)=10, kk(7)=10. Similarly node 6 is connected to node 7 and node 8. Therefore IB(1, 8) =6, IE(1, 8)=7, IB(1, 9)=6, IE(1, 9)=8, kk(8)=7, and kk(9)=8.

Table 2.1: Branch number (jj), sending-end ($m_1=IS_{(jj)}$) node, and receiving end ($m_2=IR_{(jj)}$) node for fig. 2.2

Branch number (jj)	Sending-end node $IS_{(jj)}$	Receiving end node $IR_{(jj)}$
1	1	2
2	2	3
3	3	4
4	4	5
5	2	9
6	9	10
7	2	6
8	6	7
9	6	8
10	3	11
11	11	12
12	12	13
13	4	14
14	4	15

From the above discussion, again it is seen that node 3 is connected to nodes 4 and 11, node 9 is connected to node 10 and node 6 is connected to nodes 7 and 8. Similarly the proposed logic checks whether nodes 4, 11, 10, 7 and 8 are connected to any other nodes. This process will continue unless all the nodes and branches are identified beyond node 2.

Table 2.2 gives the nodes and branches beyond branch 1. The algorithm (**Algorithm 2.3.1**) will skip the node and branch before node 2, i.e., node 1 and branch 1.

Total load fed through node 2 is sum of the loads of all the nodes beyond node 2 plus the load of the node 2 itself plus the sum of the losses of all the branches beyond node 2.

Similarly it is to be considered the receiving end nodes of the branch 2, branch 3, branch 4, ..., branch LN1 in **fig.2.2**, and in a similar way to that discussed above, the nodes and branches have to be identified beyond this receiving end nodes. **Tables 2.3 and 2.4** also give the nodes and branches beyond branches 2 and 3.

Note that if the receiving end node of any branch in **fig.2.** is an end node of the particular literal, then the total load fed through this node is load of this node itself. For example, consider node 5 in **fig.2.2.** This is an end node, and therefore the total load fed through node 5 is a load node 5 only. Similarly nodes 7, 8, 10, 13, 14 and 15 are end nodes in **fig.2.2.** The proposed computer logic will algorithm identifies all the end nodes.

The concept of identifying the nodes and branches beyond a particular node that helps in computing exact load feeding through particular node has been realized using an algorithm (**Algorithm 2.3.1**). Based on the above ideas, the algorithm for node identification beyond all the branches is presented in the next section.

Table 2.2: Nodes and branches beyond branch 1 for fig.2.2

Branch number (Beyond branch 1)	Sending end nodes	Receiving end nodes
2	2	3
3	3	4
4	4	5
5	2	9
6	9	10
7	2	6
8	6	7
9	6	8
10	3	11
11	11	12
12	12	13
13	4	14
14	4	15

Table 2.3: Nodes and branches beyond branch 2 for fig. 2.2

Branch number (Beyond branch 2)	Sending end nodes	Receiving end nodes
3	3	4
10	3	11
4	4	5
13	4	14
14	4	15
11	11	12
12	12	13

Table 2.4: Nodes and branches beyond branch 3 for fig. 2.2

Branch number (Beyond branch 3)	Sending end nodes	Receiving end nodes
4	4	5
13	4	14
14	4	15

2.3.1 Algorithm for node identification

Following is the algorithm which explains in detail about the methodology of identify the nodes and branches beyond a particular node which will help in finding the exact load feeding through that particular node.

Identification of nodes and branches beyond a particular node:

Step 1 : Read system sending end and receiving end nodes and total number of nodes and branches

Step 2 : jj = 1

Step 3 : RE(i) = IR(jj)
K = jj + 1

Step 4 : ip = 0; iq = 0

Step 5 : i= K

Step 6 : nc = 0
if { RE(jj) = IS(i)} go to step 7
Otherwise go to step 15

Step 7 : if { ip=0 } go to step 13
Otherwise go to step 8

Step 8 : in =1

Step 9 : if { IS (i) = LL (ip) and IR (i) = KK (ip) } or
if { IR(i) = KK(ip) } go to step 10
Otherwise go to step 11

Step 10 : nc =1

Step 11 : in =in + 1
if { in ≤ ip } go to step 9
otherwise go to step 12

Step 12 : if { nc =1} go to step 15
otherwise go to step 14

Step 13 : IE(jj,ip+1) = IR(jj)
IB(jj,ip+1) = IS(jj)

Step 14 : ip = ip + 1
IK(ip) = i
LL(ip) = IS(i)
KK(ip) = IR(i)
IE(jj,ip+1) = IR(i)
IB(jj,ip+1) = IS(i)
N(jj) = ip + 1

Step 15 : i = i + 1
if { i ≤ LN1} go to step 6
otherwise go to step 16

Step 16 : if { ip = 0} go to step 17
otherwise go to step 18

Step 17 : IE(jj,ip+1) = IR (jj)
IB(jj,ip+1) = IS (jj)
N(jj) = ip + 1
go to step 20

Step 18 : iq =iq + 1
If { iq > ip} go to step 20
otherwise go to step 19

Step 19 : RE(jj) = KK(iq)
K = IK(iq) + 1
if { iq ≤ ip} go to step 5

Step 20 : jj = jj + 1
if { jj ≤ LN1-1} go to step 3
other wise go to step 21

Step 21 : IE(LN1,1) = IR(LN1)
IB(LN1,1) = IS(LN1)
N(LN1) = 1

Step 22 : stop.

2.4 Load Flow Calculation

Once all the nodes and branches are identified, it is very easy to calculate voltage magnitudes and phase angles of all the nodes by using eqns. (2.8) and (2.9). In this algorithm, it is necessary to obtain the exact load feeding through all the receiving end nodes and the voltage magnitudes and phase angles of all these nodes by using equations (2.8) and (2.9) as the voltage of substation is known (V(1)). Then compute the branch current, branch losses using equations (2.1), (2.10) and (2.11).

Convergence criterion of this load flow method is the maximum difference in voltage magnitude (DVmax) should be less than 0.0001p.u. in successive iterations.

2.4.1 Algorithm for load flow calculation

Step1: Read line and load data
Initialize V(i) = VV(i) = 1.00000
LP(jj) = LQ(jj) = 0.00
δ(i) = 0.000
Iteration count = IT =1

Step 2: TLP = 0.0; TLQ = 0.0

Step 3: jj =1

Step 4: LK = N(jj)

Step 5: i = 1

Step 6: if {IB(jj,i) = IS(jj)} go to step 8
Otherwise go to step 7

Step 7: L1 = IB(jj,i)
L2 = IE(jj,i)
m1 = IS(jj)
m2 = IR (jj)
P(m2) = PL(L2)
Q(m2) = QL(L2)
go to step 13.

Step 8: L1 = IB(jj,i)
L2 = IE (jj,i)

Step 9: in = 1

Step 10: if {L1 = IS(in) and L2 - IR(in)} go to
otherwise go to step 12

Step 11: P(m2) = P (m2) + PL(L2) + LP(in)
Q(m2) = Q(m2) + QL(L2) + LQ(in)

Step 12: in = in +1
If {in ≤ LN1} go to step 10
Otherwise go to step 13

Step13: i = jj+i
If { i ≤ LK} go to step 6

Step 14:

a) Compute voltage magnitude by using eqn. (2.8), i.e., compute b(jj) and c(jj)by using eqns. (2.6) and (2.7) and phase angle by using eqn. (2.9)
b) Compute branch real and reactive power losses by using eqns. (2.10) and (2.11).

c) TLP = TLP + LP(jj)
TLQ = TLPQ+ LQ(jj)

Step15: Calculate absolute changes in voltage at node m2
DV(m2)=ABS{|V(m2)|-|VV(m2)|}
If {DV(m2)>DVmax} go to step 16
otherwise go to step 17

Step 16: DVmax = DV(m2)

Step 17: jj = jj +1
If { jj ≤ LN1} go to step 4
Otherwise go to step 18

Step 18: If { DVmax < ε } go to step 20
Otherwise go to step 19

Step 19: IT = IT + 1
VV(m2)=V(m2) for m2=2,3,....LN1.
Go to step 2.

Step 20: Solution has converged, print voltage magnitudes and its phase angle, Power losses, etc..

Step 21: Stop.

2.5 Application of Proposed Algorithm-Examples

To demonstrate the effectiveness of the proposed algorithm, three examples are considered.

2.5.1 Example-1

Fig.2.3 is a 28-node, 11KV radial distribution network. The line and load data of this system are given in **Appendix A1**. **Table 2.5** gives the load flow results of a 28-node radial distribution network. The total real- and reactive power losses of this system are **68.87 KW** and **46.55KVAr** respectively.

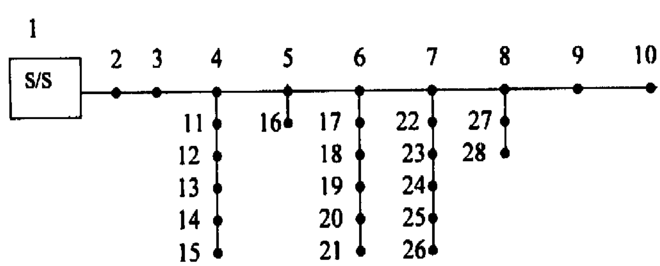

Fig. 2.3: 28 node radial distribution network

Table. 2.5: Load flow results of 28-node radial distribution network

Node number	|V| (p.u)	δ (p.u)
1	1.000000	0.000000
2	0.986312	0.002400
3	0.966675	0.005946
4	0.952669	0.008564
5	0.937160	0.011545
6	0.926464	0.013650
7	0.918431	0.015256
8	0.916770	0.015594
9	0.916281	0.015694
10	0.916038	0.015744
11	0.948346	0.010436
12	0.947050	0.011001
13	0.946390	0.011289
14	0.946141	0.011397
15	0.945902	0.011501
16	0.936036	0.012041
17	0.922517	0.013350
18	0.921101	0.013982
19	0.919120	0.014869
20	0.917985	0.015377
21	0.915444	0.016522
22	0.915449	0.016594
23	0.913472	0.017486
24	0.912236	0.018045
25	0.911702	0.018099
26	0.911538	0.018174
27	0.916575	0.015682
28	0.916526	0.015704

2.5.2 Example-2

Consider a 69-node [54] radial distribution network **[fig.2.4]**. The line and load data of this system is given in **Appendix A2**. Load-flow results of this network are given in **Table 2.6**. The total real and reactive power losses of this network are **224.99 KW** and **102.16 KVAr** respectively. These results are shown in **Table 2.6.** Compared with magnitudes of the voltages at all nodes with an existing method [125], it can be observed that the magnitudes of voltages are almost equal. However in their method [125] only magnitudes are computed, where as in the proposed method both magnitude as well as phase angle at each node are also obtained.

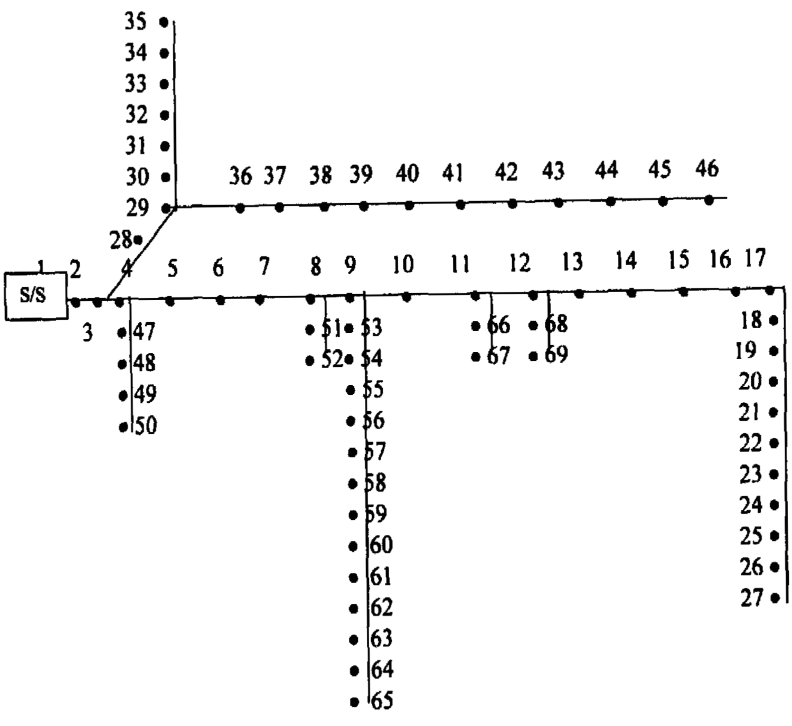

Fig. 2.4: 69 node radial distribution network

Table. 2.6: Load flow results of 69-node radial distribution network

Proposed Method		
Node No.	|V| (p.u)	δ(p.u)
1	1.00000	0.000000
2	0.99997	-0.000029
3	0.99993	0.000058
4	0.99984	-0.000139
5	0.99902	-0.000440
6	0.99009	0.001101
7	0.98079	0.002743
8	0.97858	0.003138
9	0.97744	0.003344
10	0.97245	0.005380
11	0.97134	0.005833
12	0.96818	0.007117
13	0.96526	0.008270
14	0.96236	0.009396
15	0.95950	0.010489
16	0.95896	0.010693
17	0.95808	0.011031
18	0.95808	0.011034
19	0.95761	0.011238
20	0.95731	0.011370
21	0.95683	0.011582
22	0.95682	0.011585
23	0.95675	0.011617
24	0.95660	0.011686
25	0.95643	0.011761
26	0.95636	0.011792
27	0.95634	0.011801
28	0.99993	-0.000075
29	0.99985	-0.000298
30	0.99973	-0.000015
31	0.99971	0.000035

Existing Method [125]	
Node No.	|V| (p.u)
1	1.00000
2	0.99997
3	0.99993
4	0.99984
5	0.99902
6	0.99009
7	0.98079
8	0.97858
9	0.97745
10	0.97245
11	0.97135
12	0.96819
13	0.96526
14	0.96237
15	0.95950
16	0.95897
17	0.95809
18	0.95808
19	0.95761
20	0.95731
21	0.95683
22	0.95683
23	0.95675
24	0.95660
25	0.95643
26	0.95636
27	0.95634
28	0.99993
29	0.99985
30	0.99973
31	0.99971

Continued...

32	0.99961	0.000286
33	0.99935	0.000878
34	0.99901	0.002029
35	0.99895	0.002894
36	0.99992	0.002882
37	0.99975	0.002730
38	0.99959	0.002672
39	0.99954	0.002656
40	0.99954	0.002655
41	0.99884	0.002393
42	0.99855	0.002282
43	0.99851	0.002267
44	0.99850	0.002264
45	0.99841	0.002221
46	0.99840	0.002220
47	0.99979	-0.000184
48	0.99854	-0.001275
49	0.99470	-0.004685
50	0.99415	-0.005198
51	0.97854	-0.005192
52	0.97853	-0.005188
53	0.97466	0.003849
54	0.97141	0.004439
55	0.96694	0.005263
56	0.96257	0.006074
57	0.94010	0.015604
58	0.92904	0.020542
59	0.92476	0.022530
60	0.91973	0.025107
61	0.91234	0.026816
62	0.91205	0.026883
63	0.91166	0.026974
64	0.90976	0.027418
65	**0.90918**	0.027552
66	0.97129	0.005860
67	0.97129	0.005860
68	0.96785	0.007261
69	0.96785	0.007261

32	0.99961
33	0.99935
34	0.99901
35	0.99895
36	0.99992
37	0.99975
38	0.99959
39	0.99954
40	0.99954
41	0.99884
42	0.99855
43	0.99851
44	0.99850
45	0.99841
46	0.99840
47	0.99979
48	0.99854
49	0.99470
50	0.99415
51	0.97854
52	0.97853
53	0.97466
54	0.97142
55	0.96694
56	0.96257
57	0.94010
58	0.92904
59	0.92476
60	0.91974
61	0.91234
62	0.91205
63	0.91166
64	0.90976
65	**0.90919**
66	0.97129
67	0.97129
68	0.96786
69	0.96786

2.5.3 Example-3

Next consider a 85-node [102] radial distribution network **[fig.2.5]**. The line and load data of this system are given in **Appendix A3**. Load flow results of this system are given in **Table 2.7**. The total real- and reactive power losses are **212.45 KW and 184.93 KVAr** respectively.

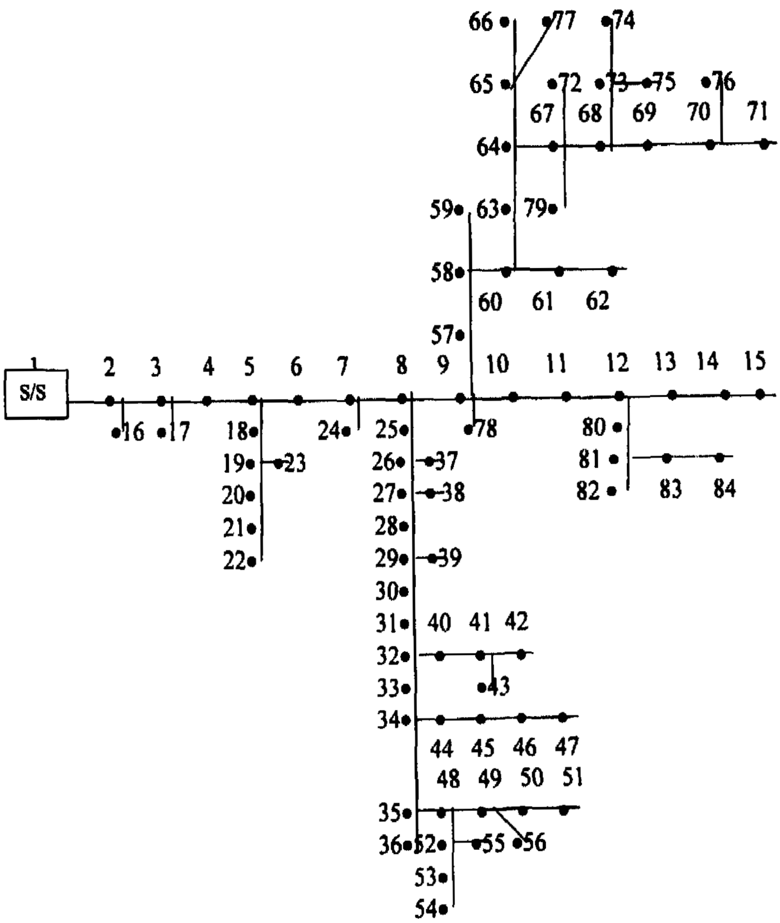

Fig. 2.5: 85 node radial distribution network

Table. 2.6: Load flow results of 85-node radial distribution network

Node No.	|V| (p.u)	δ (P.u)
1	1.000000	0.000000
2	0.995957	0.000711
3	0.989963	0.001805
4	0.982030	0.003278
5	0.978171	0.004007
6	0.964341	0.006670
7	0.955841	0.008354
8	0.937120	-0.003916
9	0.935365	-0.003578
10	0.932059	-0.003011
11	0.929414	-0.002485
12	0.927226	-0.002047
13	0.926274	-0.001856
14	0.925989	-0.001793
15	0.925815	-0.001759
16	0.995655	0.000836
17	0.989887	0.001837
18	0.975473	0.005145
19	0.973806	0.005852
20	0.973227	0.006098
21	0.972532	0.006393
22	0.971874	0.006672
23	0.973684	0.005904
24	0.955449	0.008526
25	0.931735	-0.001614
26	0.927593	0.000176
27	0.922160	0.002548
28	0.919640	0.003656
29	0.914990	0.005715
30	0.910734	0.007617
31	0.908731	0.008517
32	0.907480	0.009084
33	0.906478	0.009539
34	0.902108	0.011517
35	0.899705	0.012614
36	0.899621	0.012653
37	0.927336	0.000291
38	0.921448	0.002867
39	0.914599	0.005892
40	0.906859	0.009367

Continued

41	0.905946	0.009783
42	0.905821	0.009840
43	0.905738	0.009877
44	0.900546	0.012231
45	0.899544	0.012691
46	0.898960	0.012960
47	0.898857	0.013010
48	0.897595	0.013582
49	0.897343	0.013699
50	0.896905	0.013900
51	0.896572	0.014054
52	0.894963	0.014793
53	0.894419	0.015043
54	0.894020	0.015228
55	0.894563	0.014978
56	0.897243	0.013745
57	0.932946	-0.002526
58	0.926263	0.000396
59	0.926134	0.000453
60	0.922201	0.002197
61	0.921166	0.002661
62	0.920453	0.002982
63	0.921237	0.002628
64	0.917504	0.004293
65	0.917342	0.004367
66	0.917212	0.004426
67	0.915577	0.005156
68	0.912786	0.006411
69	0.910719	0.007347
70	0.910185	0.007589
71	0.909938	0.007702
72	0.915447	0.005215
73	0.911399	0.007040
74	0.911203	0.007129
75	0.911042	0.007352
76	0.909793	0.007768
77	0.917326	0.004374
78	0.931611	-0.002812
79	0.915331	0.005268
80	0.925742	-0.001386
81	0.925257	-0.001170
82	0.925193	-0.001140
83	0.924577	-0.000865
84	0.924399	-0.000786
85	0.925909	-0.001693

2.6 Conclusions

A simple load flow technique has been proposed for solving radial distribution networks. It completely exploits the radial feature of the distribution network. The proposed method always guarantees convergence for any type of radial distribution network with a realistic R/X ratio. The proposed method has been applied to various radial distribution networks. In particular, the results pertaining to 28, 69 and 85 node networks are presented in this chapter. This method can also handle different types of load characteristics with slight modification in basic equations.