LOAD-FLOW CALCULATIONS IN LARGE MULTILOOP NETWORKS ON DIGITAL COMPUTER

By

L. RÁCZ, Z. REGULY and A. BERTALAN

Department of Electric Power Transmission and Distribution, Technical University, Budapest

(Received November 10, 1970)

Presented by Prof. Dr. O. P. GESZTI

1. Introduction

For the design of network operation and for the effective operation control of power systems it is indispensable to determine the absolute value (and angle) of the node voltages as well as the active and reactive power flowing in the individual branches.

Because of the extremely large number of the mathematical operations it is unimaginable to solve the problem by manual calculation, so digital computer programs were developed for this purpose. There are numerous a.c. load-flow calculation programs in use, several such programs were developed in this country as well; these yield better and better solutions of the problem (less demand of computation time, capable for greater network, more universal) in the sequence of their elaboration. In this paper a program already used in practice is described the specific computation time of which is the least, at a maximum number of analyzed nodes among the similar programs used at present in our country.

2. The mathematical model of the program

The a.c. load-flow calculation problem is the following: a network containing n nodes (substations or power plant bus-bars) arbitrarily looped is given, a part of the nodes are feeding nodes while other parts are of consuming character. ($S_i = P_i + jQ_i$, where P_i is the active, Q_i the reactive power, the feeding can be taken into consideration as a negative consumption). The complex values of the power in the individual nodes as well as the absolute value and the angle (to a reference axis) of the voltage at a given node are known. This latter node is called reference point because the voltages of the other nodes are related to this one. Besides — in accordance with the practice of operative control — a part of the feeding nodes may be of constant voltage, in which the absolute value of the voltage must be kept constant at a defined tolerance by the appropriate regulation of the reactive power fed in. Naturally, it is only possible within the range of reactive power feeding of the analyzed node limited by stability analysis and by power plant capacity. Outside this range the reactive feeding of this node becomes also determined.

The mathematical model of the problem described above and based on Appendix II can be written as:

$$\mathbf{Y}_c \cdot \Delta U_c = (U_0 \cdot \mathbf{E} - \langle \Delta \hat{\mathbf{U}}_c \rangle)^{-1} \cdot \hat{S}_c \tag{1}$$

$$\mathbf{Y}_c = \mathbf{A} \cdot \mathbf{Y} \cdot \mathbf{A}_t \tag{2}$$

$$U = \mathbf{A}_t \cdot \Delta U_c \tag{3}$$

$$I = \mathbf{Y} U$$

In Appendix I the explication of the notations is given, while Appendix II contains the derivation of the equations.

Transforming the matrix equation (1), the following is obtained:

$$U_0 \cdot \mathbf{Y}_c \cdot \Delta U_c - \langle \Delta \hat{\mathbf{U}}_c \rangle \cdot \mathbf{Y}_c \Delta U_c - \hat{S}_c = 0 \tag{1a}$$

Matrix equation (1a) is nonlinear because of the presence of the product $\langle \Delta \hat{\mathbf{U}}_c \rangle \cdot \mathbf{Y}_c \cdot \Delta U_c$, so it can only be solved by iterative methods. Appendix III briefly describes the types of the commonly used iterative methods, which can be divided into two main groups:

a) inversion — iterative methods,
b) pure iterative methods.

The algorithm of the program described below belongs to the group a).

3. The principle of the program algorithm

The method used in the present program is not pure iterative, so it can be classified to the group a); but it cannot be considered as a method of inversion in its original sense, because the inverse of the node admittance matrix ($\mathbf{Y}_c$) does not appear in matrix form. To clear up the basic principles let us consider the inital equation of the calculation according to Appendix II:

$$\mathbf{Y}_c \cdot \Delta U_c = I_c \tag{5}$$

The basis of n—1 dimensions (namely the rank of the matrix $\mathbf{Y}_c$ is also n—1) in which this matrix equation is valid — i.e. where the column matrix I_c and the column matrices of $\mathbf{Y}_c$ are given — is defined by the unit matrix

$\mathbf{E}_{n-1}$ (trivial basis). If a transformation is made to a new basis which is obtained by changing the first column of $\mathbf{E}_{n-1}$ to the first column of $\mathbf{Y}_c$, then this new basis is defined by the following matrix:

$$\mathbf{Y}_{c_1} = \left[\begin{array}{ll} Y_{11} & O_t \\ Y_{21} & \\ \cdot & \\ \cdot & \mathbf{E}_{n-2} \\ Y_{n-1,1} & \end{array}\right] \tag{6}$$

where the symbol O_t denotes the transposed vector of the zero vector of n—2 dimension. If the components of the current vector $\mathbf{I}_c$ are given by the elements of $I_c^{(1)}$ in the new basis, the following can be written:

$$I_c = \mathbf{Y}_{c_1} \cdot I_c^{(1)} \qquad I_c^{(1)} = \mathbf{Y}_{c_1}^{-1} \cdot I_c \tag{6a}$$

where

$$\mathbf{Y}_{c_1}^{-1} = \left[\begin{array}{ll} \dfrac{1}{Y_{11}} & O \\ -\dfrac{Y_{21}}{Y_{11}} & \\ \cdot & \\ \cdot & \\ -\dfrac{Y_{n-1,1}}{Y_{11}} & \mathbf{E}_{n-2} \end{array}\right] \tag{7}$$

Correctness of (7) can be proved by checking the equation $Y_{c_1} \cdot Y_{c_1}^{-1} = \mathbf{E}_{n-1}$. In the new basis Eq. (5) can be written as

$$\mathbf{Y}_c^{(1)} \cdot \Delta U_c = I_c^{(1)} \tag{8a}$$

where evidently:

$$\mathbf{Y}_c^{(1)} = \left[\begin{array}{llll} 1 & \dfrac{Y_{12}}{Y_{11}} & \cdots & Y_{1,n-1} \cdot \dfrac{1}{Y_{11}} \\ 0 & & & \\ \cdot & & & \\ \cdot & & \mathbf{A}_1 & \\ 0 & & & \end{array}\right] \tag{86}$$

Here $\mathbf{A}_1$ is a nonsingular quadratic matrix of n—2 order, while the general element of matrix $\mathbf{Y}^{(1)}_{c_1}$ is

$$Y^{(1)}_{i,j} = Y_{i,j} - \frac{Y_{1,1}\cdot Y_{1,j}}{Y_{11}} \tag{8c}$$

Obviously, Eqs (8) are correct if, after multiplying column matrices of $\mathbf{Y}^{(1)}_{c}$ by $\mathbf{Y}_{c_1}$, the original column matrices of $\mathbf{Y}_c$ result. Based on Eq. (6a):

$$I^{(1)}_c = Y^{-1}_{cl}\, I_c = \begin{bmatrix} \frac{1}{y_{11}}\cdot i_{cl} \\ i_{c_2} - \frac{Y_{21}}{Y_{11}}\cdot i_{cl} \\ \cdot \\ \cdot \\ i_{c,n-1} - \frac{Y_{n-1}}{Y_{11}}\cdot i_{cl} \end{bmatrix} \tag{9}$$

If now the second column matrix of the new basis is changed by the second column matrix of $\mathbf{Y}^{(1)}_c$, then, after transformations formally similar to that of Eqs (6) through (9), the original equation appears in the following form in the new basis, determined now by matrix $\mathbf{Y}_{c_2}$

$$\mathbf{Y}^{(2)}_c \cdot \Delta U_c = I^{(2)}_c \tag{10}$$

where the first and second column matrices of $\mathbf{Y}^{(2)}_c$ are already the e_1 and e_2 unit vectors, respectively, and

$$I^{(2)}_c = \mathbf{Y}^{-1}_{c_2}\cdot \mathbf{Y}^{-1}_{c_1}\cdot I_c \tag{11}$$

Performing the change of bases one after the other in this way, Eq. (5) becomes after the n—1th step:

$$\mathbf{E}_{n-1}\cdot \Delta U_c = I^{(n-1)}_c = (\mathbf{Y}^{-1}_{c,n-1}\cdot Y^{-1}_{c,n-2}\ \dots\ \mathbf{Y}^{-1}_{c_1})\cdot I_c \tag{12}$$

According to (5)

$$\Delta U_c = \mathbf{Y}^{-1}_c\cdot I_c \tag{5a}$$

so from (5a) and (12)

$$Y^{-1}_c = (\mathbf{Y}^{-1}_{c,n-1}\ \dots\ \mathbf{Y}^{-1}_{c_1}) \tag{12a}$$

that is, the inverse appears in product form.

It is to be mentioned that not all the elements of the $\mathbf{Y}_c^{(1)}$, $\mathbf{Y}_c^{(2)}$, etc. matrices and those of the $\mathbf{Y}_{c1}$, $\mathbf{Y}_{c2}$, etc. basic matrices are of admittance dimension, as it is seen from Eqs (5) and (8), respectively. In spite of this, for the sake of continuity it seems to be expedient to use the notation $\mathbf{Y}$. The same is valid for the column matrices I_c^1, I_c^2, etc., the elements of which are of different dimensions. According to Eq. (12), all the elements of the column matrix I_c^{n-1} obtained in the last step are of voltage dimension. Nevertheless, the notation I is kept.

From Eq. (12) it appears that the column matrix ΔU_c formed of the relative voltage drops of the nodes can be calculated by making the series of basis transformations characterized by the matrices $\mathbf{Y}_{cl}^{-1} \ldots \mathbf{Y}_{c,n-1}^{-1}$. By generalizing Eq. (9) it can be stated that the basis transformation done in the i-th step and characterized by the equation $I_c^{(i)} = Y_{cl}^{(-1)} \cdot I_c^{(i-1)}$ contains the following steps:

a) the value of the i-th element of the column matrix $I_c^{(i-1)}$:

$$I_{ci}^{(1)} = \frac{1}{Y_{ii}(i-1)} \cdot i_{ci}^{(i-1)} \tag{13a}$$

b) the new value of the arbitrary jth element of the column matrix I_c^{i-1}

$$i_{cj}^{(i)} = i_{cj}^{(i-1)} - \frac{Y^{(i-1)}}{Y_{ii}^{(i-1)}} \because i_{ci}^{(i-1)} = i_{cj}^{(i-1)} - Y_{j1}^{(i-1)} \cdot i_{ci}^{(i)} \tag{13/b}$$

where the numbers $Y^{(i-1)}$ denote the elements of matrix $\mathbf{Y}_c^{(i-1)}$. From Eq. (13b) it is obvious that an arbitrary vector element $i_{c_k}^{i-1}$ changes during the basis transformation only if the corresponding $Y_{j.l}^{(i-1)}$ element is not zero. Then the modification can be done with the aid of the following three data: i, j and $Y_{j,1}^{i-1}$;
if $i = j$, then according to (13a) two data are sufficient: i and $Y_{ii}^{(i-1)}$.

4. The arithmetic part of the programs

4.1. *Determination of the favorable node sequence*

It is known that the node admittance matrix $\mathbf{Y}_c$ of power transmission networks contains a lot of zero elements: in general, only 5—6 per cent of the elements are not zero. This fact can be used to reduce the number of the operations necessary for the basis transformations according to Eqs (6 to 8) to a least value by suitably choosing the sequence of nodes. This is possible by the following considerations:

According to Eq. (8c), the general i, j element of matrix $\mathbf{Y}_c^{(k)}$ can be calculated as:

$$Y_{i,j}^{(k)} = Y_{i,j}^{(k-1)} - \frac{Y_{i,k}^{(k-1)} \cdot Y_{k,j}^{(k-1)}}{Y_{kk}^{(k-1)}} \tag{8d}$$

$\mathbf{Y}_{kk}^{k-1}$ is the generating element of the kth basis transformation.

From Eq. (8d) it appears that during the kth transformation the logical condition of the change of element $Y_{i,j}^{(k-1)}$ is:

$$Y_{i,k}^{(k-1)} \neq 0 \wedge Y_{k,j}^{(k-1)} \neq 0 \tag{14}$$

On the other hand, provided that (14) holds, the most unfavorable is if $Y_{i,j}^{(k-1)} = 0$, because in this case the i, j element, which was originally zero, gets a non-zero value; namely the matrix part not involved in the basis transformation (the quadratic matrix $\mathbf{A}_k$ according to 8b) is filled gradually by non-zero elements, increasing thereby the operation demand of the remaining basis transformations. From all these the following conclusions can be drawn:

a) In case the generating element of the kth basis transformation belongs to a node of radial, end-point character and being connected to the node of number l, then in the kth row there are only two non-zero elements: Y_{kk} and Y_{kl}; while Y_{kk} and Y_{lk} are the non-zero elements in the kth column. Then during the basis transformation, taking into consideration Eqs (8d) and (14), only the element Y_{ll} changes; no new element is formed and element Y_{lk} is zeroed [to the analogy of (8d), hence in the residual matrix $\mathbf{A}_k$ the number of the non-zero elements diminishes by one. If the node l had only one joint (e.g. at the mth node) apart from the joint at the kth node, then the original non-zero elements of the lth row are: Y_{lk}, Y_{ll} and Y_{lm}. From these Y_{lk} becomes zero during the kth transformation — as we have already seen — so in the further steps the lth node behaves like and end point as it was written above.

b) If the generating element of the basis transformation belongs to a node in which only two branches (loop branches) are connected, then in the row and in the column of the generating element there are three elements. In this case during the transformation — according to similar ideas — two new elements are formed and two elements are zeroed; thus the number of the elements do not change, neither does the saturation of the remaining matrix $\mathbf{A}_k$.

c) Evidently, if the generating element belongs to a node in which three or more branches are connected, then during the transformations the saturation of the remaining matrix increases with the number of branches connected to the node in question. The "three branch" nodes are exceptions if among the branches connected to them there is at least one radial branch, because at the previous transformation done with the generating element of this joint radial node the corresponding element became zero in the row of this three-branch node. Thus this node has a character similar to those in item b).

From these it follows that the necessary quantity of operations of the transformation in item 3 can intensively be reduced, if the sequence of operations is chosen in the following manner: the radial end-point nodes are numbered beginning with 1;

the "two-branch" nodes are getting the following numbers and from these the ones with one radial branch must precede the others;

thereafter the three-, four-, etc. branch nodes will be numbered; here it must be taken into consideration that the nodes with radial branches must always precede the other ones.

The reference point will get the last number; this point represents generally a power plant with large generating reserve and several connected branches (transmission line, transformer).

The transmission lines and their data have to be given in the sequence of the enumeration so that the data of the outgoing transmission lines must be enumerated for each node in the increasing sequence of the nodes of the other end of the lines.

In the case of large networks it is very difficult or even impossible to compile the data in this way and it may cause a lot of data errors. Therefore it is absolutely necessary to prepare a sub-program for this work. This program part produces a data system satisfying the conditions described above from a simply given data system.

4.2. *The algorithm of the data transformation*

The algorithm will be presented on an example and general conclusions will de drawn of it.

Let us consider the network in Fig. 1. Noting the serial number of reading, the input sequence will be the following:

1	170
2	180
3	220
4	112
5	101
6	123

The node transformation vector t contains these data. By means of this vector the original number of the ith read node can be determined.

From now on, the nodes will be noted by the number of the input (Fig. 2). Now the matrix **H** of the network topology must be produced. In our example:

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0. \end{bmatrix}$$

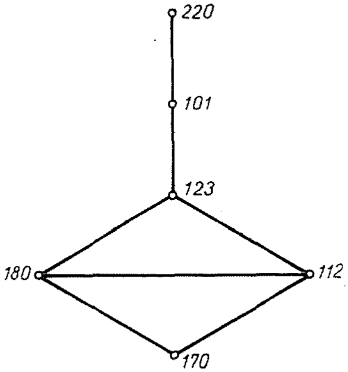

Fig. 1

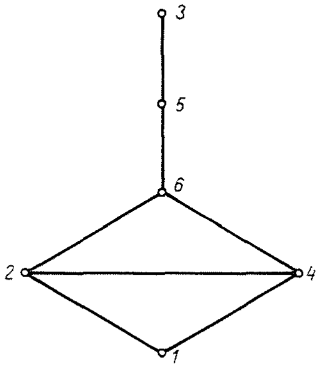

Fig. 2

This matrix is often called a vertex matrix. As our purpose is to find the radial branches with end vertex in first step we can find them by producing the *row sums* of matrix **H**:

$$c = \mathbf{H}e$$

where c is the row sum vector

e is the summarizing vector

The row sum vector designates the end branches. Hereupon the remaining radial branches must be found by iteration. Now these and the other looped branches must be arranged in rows so that first the end branches, then the radial elements and at least the looped elements (according to their looping grades) follow.

Let us consider the vector c as the first element of the iteration

$$c^{(i)} = c$$

Let us form now a subsidiary vector

$$v^{(0)}$$

where $v_j \begin{cases} = 1000 \text{ if the } j\text{th node is the reference point} \\ = \ \ 100 \text{ in every other case.} \end{cases}$

Suppose for the previous calculations that

$$c^{(0)} = 0$$

In our example the 180th node will be the reference point, so

$$c^{(1)} = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 3 \\ 2 \\ 3 \end{bmatrix} \qquad v_{(o)} = \begin{bmatrix} 100 \\ 1000 \\ 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}$$

In the kth step of the iteration the value of $v^{(k)}$ must be determined:

$$v_j^{(k)} = \begin{cases} k, & \text{if } c_j^{(k)} = 1 \wedge c_j^{(k-1)} - c_j^{(k)} \neq 0 \\ v_j^{(k)} \text{ else} \end{cases}$$

Thus, in our example:

$$v^{(1)} = \begin{bmatrix} 100 \\ 1000 \\ 1 \\ 100 \\ 100 \\ 100 \end{bmatrix}$$

In the following steps of the iteration the further radial branches must be found. For this purpose two previous subsidiary vectors have to be calculated, needed to be defined only from the second step.

$r^{(k)}$ is the subsidiary vector of the kth step and it shows the found end branches; its arbitrary element is:

$$r_j^{(k)} = 0 \text{ if } v^{(k-1)} - v^{(k-2)} = 0$$

1 in every other case.

In our example:

$$r^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The subsidiary vector $s^{(k)}$ must be calculated in the following way:

$$S^{(k)} = S^{(k)}\,(\mathbf{H} \cdot v^{(k)}, c^{(k-1)})$$

where

$$S_j^{(k)} = \begin{cases} 0 \text{ if } c_j^{(k-1)} = 1 \\ \sum_l h_{lj}^{(k)} \cdot v_j^{(k)}, \text{ if } c_j^{(k-1)} \neq 1 \end{cases}$$

In our example

$$S^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

The value of the vector $c^{(k)}$:

$$c^{(k)} = c^{(k-1)} \qquad S^{(k)}$$

In our example:

$$c^{(2)} = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 3 \\ 1 \\ 3 \end{bmatrix} \qquad \text{and} \qquad v^{(2)} = \begin{bmatrix} 100 \\ 1000 \\ 1 \\ 100 \\ 2 \\ 100 \end{bmatrix}$$

The iteration has to be continued until $r^{(k)} = 0$, where there is no more radial branch.

Let us do the iteration in the example

$$r^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad S^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad c^{(3)} = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 3 \\ 1 \\ 2 \end{bmatrix}$$

and at last:

$$v^{(3)} = \begin{bmatrix} 100 \\ 1000 \\ 1 \\ 100 \\ 2 \\ 100 \end{bmatrix}$$

In the next step $r^{(4)} = 0$, thus the iteration has come to the end. Where there is no more radial branch, the nodes can be arranged in rows according to their looping grade.

For the determination of the final sequence, the vector v gives the numbers characterizing the individual nodes:

$$v = v^{(n)} + c^{(n)}$$

where n is the number of the last step where vector $r^{(n)}$ is not yet zero. In our example:

$$v = v^{(3)} + c^{(3)} = \begin{bmatrix} 100 \\ 1000 \\ 1 \\ 100 \\ 2 \\ 100 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \\ 1 \\ 3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 102 \\ 1003 \\ 2 \\ 103 \\ 3 \\ 102 \end{bmatrix}$$

After this the optimal node sequence is obtained by arranging the elements of v according to their magnitude.

$$v_{\text{row}} = \begin{bmatrix} 2 \\ 3 \\ 102 \\ 102 \\ 103 \\ 1003 \end{bmatrix}$$

and the corresponding sequence of node numbers:

$$\begin{bmatrix} 3 \\ 5 \\ 1 \\ 6 \\ 4 \\ 2 \end{bmatrix} \quad \text{that is with original numbers} \quad \begin{bmatrix} 220 \\ 101 \\ 170 \\ 123 \\ 112 \\ 180 \end{bmatrix}$$

Fig. 3 shows the network of Fig. 1 with the numbers of the optimal node sequence.

With the use of the data arranging program described above, the data must be given in the following form:

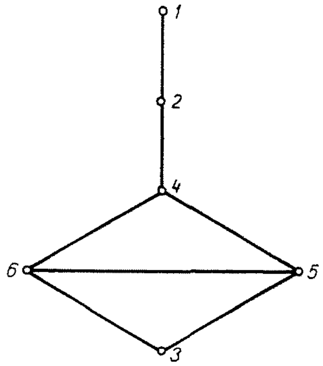

Fig. 3

— data of the nodes (serial number, type, load) in arbitrary sequence;

— the lines and their data (node number of their end-points, type, resistance, inductivity, capacity) in arbitrary sequence.

The program will produce the topology matrix and the optimal sequence of these data. Fig. 4 shows a data tape as an example.

4.3. *The structure of the main part of the program*

Appendix 3 deals with the inversion-iteration method of the solution of the non-linear matrix equation (1a), which is the mathematical model of the problem. Based on these and taking into consideration the ideas in Chapter 3, the following algorithm was elaborated:

a) The program determines the optimal node sequence according to point 4.2.

To control the further operations and for the sake of compact storage, a two-dimensional integer array is needed which contains the topological data of the network. Hereafter we shall show the construction of the topological array on a simple example. The rows of this array correspond to the nodes in the sequence of the inner numbering. Thus the topological array of the network of our example — taking into consideration the node numbers in Fig. 3 — is shown in Table 1.

Table 1

1	2	0	0	0
1	2	4	0	0
3	5	6	0	0
2	4	5	6	0
3	4	5	6	0
3	4	5	6	0

CS. P	KOORDI- NAATA	JELLEG	TELJESITMEENYEK MW	 MVAR	FESZ	MEDDOE HATAA ROK ALSO	 FELSOE:
43	26.035	0	19	3.5			
81	8.020	0	34	−20			
96	8.060	0	9	3.5			
93	8.090	−1	−220	−20	128		
293	8.115	0	0	0			
44	16.060	0	51	20			
80	26.090	0	45	32.2			
280	26.115	0	0	0			
98	32.010	0	13	4			
17	40.035	0	37	9			
45	40.080	0	19	6.5			
53	46.060	0	31	15			
57	58.060	1	−45	−44	126	−10	−80
9999							

AAG ELEJE	AAG VEEGE	KOORDI- NAATA	TIP. SZ.	R	X	C:
81	93	3.035	23	7.090	24.41	0.536
81	96	8.035	23	1.144	8.4	0.172
81	43	16.024	23	8.74	20.03	0.384
81	44	12.035	23	6.9	11.84	0.212
96	93	8.075	23	2.46	18.1	0.37
93	293	8.100	12	0.6	9.5	0
43	17	32.035	23	14.25	32.5	0.623
17	98	36.020	23	6.85	23	0.5
17	53	40.046	23	7.79	17.66	0.34
53	57	52.060	73	1.8	4.1	0.317
57	45	52.071	23	11.67	26.6	0.51
45	80	32.086	22	10.77	25.0	0.463
80	280	26.100	12	0.61	9.5	0
80	44	21.075	22	12.53	21.51	0.385
280	293	16.115	29	2.8	20.4	0
9999						

Fig. 4

It is seen that on the end of each row at least one zero element is found, indicating that no further elements follow.

b) After these the computation and storage of the elements of matrix $\mathbf{Y}_c$ is performed, so that in one array the real elements, in an other the imaginary ones will be stored. These arrays contain the elements of matrix $\mathbf{Y}_c$ in compact form, that is the zero elements — which constitute most part of the elements — are not stored; the length of the rows is determined by the number of the non-zero elements in the very row of $\mathbf{Y}_c$ containing the most non-zero elements.

The arrays containing the real and imaginary part of matrix $\mathbf{Y}_c$ of the network in our example are similar in construction to the array containing the topological data. For example, in the 4th row of these arrays the first element represents the transfer admittance 2—4, the second — where in the topological array the number of the row in question can be found — the driving point admittance of the node 4, the third one is the 4—5, and the fourth one is the 4—6 transfer admittance.

c) The next part of the program performs the series of basis transformations described in item 3. This is controlled by the topological matrix so that the multiplications with zero elements need not to be executed; by this method the operation demand of the program is intensively reduced. The two-dimensional array — the rows of which contain the data necessary for the elementary transformations on the current vector I_c according to Eqs 13a and 13b in Chapter 3 — will be filled during the basis transformations.

d) At the end of the transformations the matrix $\mathbf{Y}_c$ will take the form of a unit matrix according to relationship (12). At the same time the two-dimensional array mentioned in c) contains the whole information on the matrix product $\mathbf{Y}_{c_{n-1}}^{-1}\ldots\mathbf{Y}_{c_1}^{-1}$. After these, for starting the iterations, the program sets the voltage of every node equal with that of the reference point and then calculates the starting value of the node currents from the equation:

$$I_c = (U_0 \cdot \mathbf{E})^{-1} \cdot S_c \tag{13}$$

Thereafter the series of transformations on the vector I_c according to Eq. (12) will be executed. It results in the first approximation of the voltage difference ΔU_c. Then the new approximation of I_c is:

$$I_c = (U_0\mathbf{E} - \langle \Delta U_c\ )^{-1} \cdot S_c \tag{14}$$

Hereupon the iteration will be continued again by the series of transformations according to (12) until the change of each element of the voltage difference ΔU_c diminishes to a given value.

The main advantage of the algorithm described above is that the operation demand of the individual iterations is essentially less than for common inversion-iteration methods. It is worth to mention as an example that in the calculation of the Hungarian multilooped network of 100 nodes, the equivalent operation demand of the series of transformations, responsible for 80 per cent of the operation demand of an iteration — corresponds to about 600 complex multiplications (the operation demand of a complex multiplication is four multiplications + two additions). At the same time, in a network of similar size, the equivalent operation demand of the matrix product $\mathbf{Y}_c^{-1} \cdot I_c$ calculated in each iteration — as the node impedance matrix $\mathbf{Y}_c^{-1} = \mathbf{Z}_c$ does not contain zero elements — is about 10 000 complex multiplications i.e. it is larger by an order of magnitude.

4.4. *Output*

Output may appear either
— in tabulated or
— in map form.

The table of the outputs contains the nodes with their original numbers, in increasing order of magnitudes. It gives the resulting node data (voltage, angle, power) and the resulting data of the lines joining the node (number of the other node, active and reactive power, capacitive power). In addition it gives the loss data and the total capacitive power (reactive power production). Fig. 5 shows a table of the results. The output in map form permits data of the nodes and lines to be printed on the very place of a previously drawn map. It is realized so that the co-ordinates of the optimal place of printing will be measured on the scheme of the network drawn before on a paper, and these co-ordinates will be given on the data card after the nodes and after the data of the nodes indicating the end of the lines. Fig. 6 shows a print in map form, as an example.

The computer prints in the nodes the node voltage, the active and reactive power and on the lines the active and reactive power as well as the percentage of loading. The power data of the lines are positive if the power flows from the node of higher to that of lower number.

The algorithm of the program in map form will not be described here in detail. A description in detail of the program will be published later, containing the algorithm, the handling of the program, different output possibilities, the methods of the computation of the basic state and the modifications, etc.

5. Computation experiences

The effective digital program is written in ALGOL language corresponding to the RAZDAN computer representation. In view of the computer characteristics, the program in its present form is suitable for the calculation of a network containing 200 nodes and 300 lines. This capacity can be increased by refining the program. The control calculations were carried out on the Hungarian 120—220—400 kV co-operation network containing 100 nodes by a series of load-flow calculations. The computer time needed for the calculation of basic state in details is composed of:

a) data input from data tape: about 25 s,

b) data transformation: about 45 s,

c) total computer time of the setting up of the matrix $\mathbf{Y}_z$ and of the total series of basis transformation: about 40 s,

d) computer time of each step in the series of iterations described in item 4.2./*d*: about 6 s, VAALTOZAT SZAAMA: 1975 MUNKA SZAAM: 1002

CSOMOPONTI JELLEMZOK AAG JELLEMZOEK

SZAMA	FESZ. ABS	SZOG	TELJESITMENYEK		CSP. FESZ	CSP.	TELJESITMEENYEK		TOLTOTELJ	TERH.
	KV	RAD.	MW	MVAR	ELTERES	SZAMA	MW	MVAR	MVAR	
17	123.38	—14.3	37.0	9.0		98	13.05	1.987	1.20	
						43	—28.31	11.90	1.49	
						53	—21.67	—22.71	0.81	
43	123.47	—10.2	19.0	3.5		17	29.23	—12.67	1.49	
						81	—48.21	9.230	0.92	
44	121.51	—7.85	51.0	20.0		80	—6.657	—4.482	0.89	
						81	—44.31	—15.52	0.49	
45	124.79	—11.0	19.0	6.5		57	9.299	—28.49	1.25	
						80	—28.27	22.07	1.13	
53	127.89	—13.6	31.0	15.0		17	22.15	22.22	0.87	
						57	—53.07	—37.12	0.81	
57	129.80	13.0	—45.0	—64.0		45	—8.677	27.41	1.35	
						53	53.53	36.58	0.84	
80	122.83	—7.47	45.0	32.2		44	6.706	—2.788	0.91	
						280	—80.85	—12.86	0.00	
						45	29.19	—22.11	1.10	
81	125.53	—6.27	34.0	20.0		96	—60.10	1.751	0.43	74.8
						44	45.33	16.26	0.52	
						43	49.59	—7.884	0.95	
						93	—68.82	9.884	1.33	86.5
93	128.00	.000	—220.0	—20.0		96	70.10	6.420	0.95	85.9
						293	81.79	24.24	0.00	
						81	71.00	—5.067	1.38	86.9

```
                                                              81      60.36    -.6764    0.43    74.9
 96     125.99     4.44      9.0      3.5                     93     -69.36    -2.817    0.92    86.1

 98     122.10    -15.4     13.0      4.0                     17     -12.97    -3.932    1.17

                                                              17     -81.12   -17.07     0.00
280     217.17    -4.63      0.0      0.0                     17      81.12    17.07     0.00

                                                              17      81.53    20.03     0.00
293     219.94    -2.70      0.0      0.0                     17     -81.53   -20.03     0.00


     A REFERENCIAPONT WATTOS EES MEDDOE SZALDOJA:

        222.9  MW          25.60  MVAR


  OESSZES WATTOS EES MEDDOE VESZTESEEGEK
                                 MW          MVAR

     120 KV-ON                  9.123        5.048
     220 KV-ON                  0.943       11.385
     400 KV-ON                  0.000        0.000
     SZUMMA                    10.067       16.433


  OESSZES TOELTOE TELJESITMEENY:  23.64 MVAR


HA TEERKEEP SZERINTI KIIRATAAS NEM KELL, 1-ES KULCS LENYOMANDO
```

Fig. 5

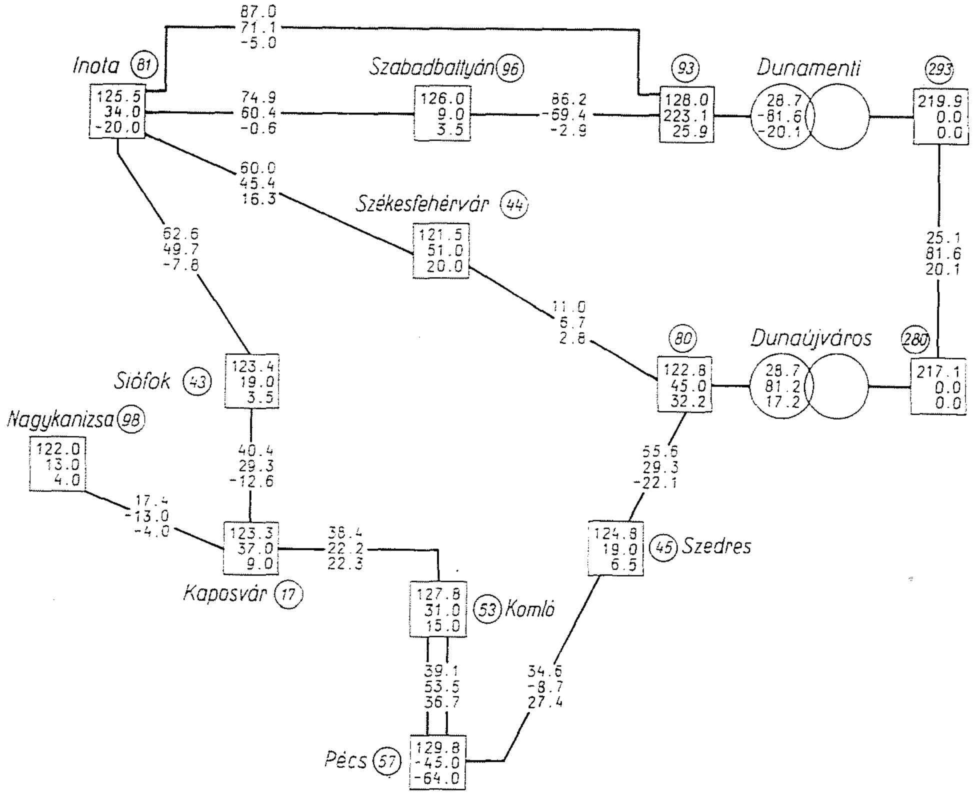

Fig. 6

e) printing of the results on line-printer in table form (the angle to the voltage of the reference point, the absolute value of the node voltages as well as the active and reactive load of each branch on both ends): about 90 s.

f) printing in map form the results at basis state: 2 minutes.

The iterations will be continued until the change of the elements of the voltage difference vector (both the active and the reactive component) related to its value in the previous iteration diminishes to a value given on the data tape. This value was fixed by test calculations at 0.1 kV: according to the experiences it would have been useless to give a lower value because the balance failures in the values of the active and reactive power of the individual nodes were throughout below 0.3 MVA (for most nodes below 0.1 MVA !).

In the peak load state of the analyzed network (in this case the iteration process converges more slowly than it would converge in low load period) generally 9—10 iterations were needed to reach the limit of error mentioned above.

Thus the total computer time of the calculation of the base state is about 6 minutes including the printing for the network in question. If calculations are needed for several networks and load states, then these may follow in turn the calculation of the basic state. If a switch on or switch off has to be analyzed then the calculation begins from the beginning; if there is only a load modification then the calculation will continue at the iterations in the program. The time needed for the iterations of a new state is generally less; 4—6 cycles are mostly sufficient. Considering all these if the calculations of several states are carried out in expedient groups one after the other, the average computer time demand is 2.5—3 minutes including the printing. (From this the printing takes about 2 minutes.) It must be mentioned that if in the individual states not all the results are needed, but only certain basic informations, then this can be reached by appropriate control -panel action and in this case much of the printing time can be saved.

Appendix I

Notations:

n number of nodes in the network;
e number of branches of the looped network.

Currents:

i_{c_i} load current of the *i*th node (positive if it flows out of the node),
I_c node current vector of n—1 elements,
I branch current column vector of e elements,
i_l *l*th branch current.

Voltages:

U_0 voltage of the reference node (line to ground voltage),
ΔU_c column matrix of order n—1 of the voltages related to the reference point,
$\langle \mathbf{U}_c \rangle$ node voltage (line to ground) diagonal matrix of order n,
$\langle \Delta \mathbf{U}_c \rangle$ diagonal matrix of order n—1 of the voltage related to the reference point,
U branch voltage column matrix of e elements,
U_l voltage drop of the *l*th branch.

Powers:

S_c column matrix of the apparent power of the node loads.
Impedances and admittances:

$\langle \mathbf{Y} \rangle$ diagonal matrix of the branch admittances,
$\mathbf{Y}_c$ node admittance matrix,
$\mathbf{Z}_c = \mathbf{Y}_c^{-1}$:
Y_l admittance of the lth branch (the lth diagonal element of $\mathbf{Y}$).

Other concepts:

$\mathbf{E}_n$ unit matrix of n dimensions
e_i ith unit vector
e summarizing vector
$\wedge$ sign of logical multiplication
$\hat{}$ conjugate value

$$e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \cdot \\ \cdot \\ 1 \\ 1 \end{bmatrix}$$

Appendix II

Three-phase networks of symmetrical structure and operation can be represented by their single-phase positive sequence scheme. In load-flow calculations the node currents are given, so a current generator must be inserted between the n—1 node of an n node network and the neutral busbar. As evidently $\sum_{i=1}^{n} i_{c_i} = 0$, so the current of the nth node is already determined by the others, and it cannot be chosen independently; between this point (so called reference point) and the zero busbar a voltage generator must be inserted which determines the voltage of the reference node. Drawing the graph of the network containing $n + 1$ nodes — taking into consideration that current and voltage generators represent a disconnection and a short circuit, respectively — a graph of n nodes is obtained. The nth node corresponds to the reference node. The rank of the incidence matrix $\mathbf{A}_a$ (vertex matrix) is n—1, so one row may be omitted; it is practical to omit the row corresponding to the reference row; thereby the matrix $\mathbf{A}$ to be used in the following is obtained. In this case, Kirchhoff's 1st law can be written in matrix form for each node of the network (except the reference node):

$$A \cdot I = I_c \qquad \text{(F.1)}$$

Considering an arbitrary tree, the corresponding fundamental loop set and its matrix $\mathbf{B}_f$ can be determined. The rank of this matrix (and the number of its rows as well) is equal to the nullity ($e - n + 1$) of the graph; each row belongs to a fundamental loop, then columns to the individual branches, so its transposed matrix can be multiplied by $\mathbf{A}$. It can be proved that

$$\mathbf{A}\cdot\mathbf{B}_f^{*}=0 \tag{F. 2}$$

Kirchhoff's 2nd law can be written for the independent loops determined by the fundamental loop set chosen:

$$\mathbf{B}_f\cdot U=0 \tag{F. 3}$$

Based on (F. 2) the bases formed by **A** and $\mathbf{B}_f$ are orthogonal, on the other hand, as the ranks of **A** and $\mathbf{B}_f$ are n—1 and $e-n+1$, respectively, so they determine together the e-dimensional space where the **U** and **I** vectors can be interpreted as well. Based on (F. 3), **U** is orthogonal to the basis formed by $\mathbf{B}_f$ and so it must lie in the n—1 dimensional subspace characterized by the basis **A**. Then **U** can be expressed by a linear combination of the column vectors of **A**:

$$U=\mathbf{A}^{*}\cdot\Delta U_c \tag{F. 4}$$

where the ΔU_c vector contains the coordinates of U in the basis **A**. The vector ΔU_c is obviously of n—1 dimension and as the rows of **A** belong to the individual nodes, so the elements of ΔU_c are the so-called node voltages referred to the reference point. (The asterisk superscript of the matrix indicates a transposed matrix.) It is evident that for any lth branch of the network:

$$i_l=y_l\cdot u_l \tag{F. 5}$$

and for the whole network $I=\mathbf{Y}\,U$. (F. 6)

Premultiplying Eq. (F. 6) by **A** and taking into consideration (F. 1) and (F. 4), it can be written that

$$\mathbf{A}\cdot\mathbf{Y}\cdot\mathbf{A}^{*}\cdot\Delta U_c=\mathbf{Y}_c\cdot\Delta U_c=I_c \tag{F. 7}$$

As the voltage of the reference point is U_0 so the diagonal matrix formed by the effective node voltages

$$\langle\mathbf{U}_c\rangle=U_0\mathbf{E}_{n-1}-\langle\Delta U_c\rangle \tag{F. 8}$$

Furthermore it is obvious that the vector of the node current can be calculated in the following way:

$$I_c=\langle\hat{\mathbf{U}}_c\rangle^{-1}\cdot\hat{S}_c \tag{F. 9}$$

Relationships (F. 7) and (F. 9) deliver the basic equation of the load-flow calculation:

$$\mathbf{Y}_c\cdot\Delta U_c=\langle\hat{\mathbf{U}}_c\rangle^{-1}\cdot S_c \tag{F. 10}$$

Appendix III

The matrix equation (F. 10) is nonlinear according to Chapter 2, and it can only be solved by iterative methods. According to literature data, the relevant methods can be divided into two main groups:

a) inversion-iterative methods,

b) pure iterative methods.

ad a) From Eq. (F. 10) premultiplied by $\mathbf{Y}_c^{-1}$ we obtain:

$$\Delta U_c = \mathbf{Y}_c^{-1} \cdot \langle \hat{\mathbf{U}}_c \rangle^{-1} \cdot \hat{S}_c \qquad \text{(F. 11)}$$

(F. 11) is the basic equation of the iteration. At the beginning of the calculation the vector ΔU_c is not known. Then, after the first iteration, based on (F. 8) and (F. 11):

$$\Delta U_{c(1)} = \frac{1}{U_0} \cdot \mathbf{Y}_c^{-1} \cdot \hat{S}_c \qquad \text{(F. 12)}$$

Generally after the *i*th step:

$$\Delta U_{c(i)} = \mathbf{Y}_c^{-1} \cdot \langle \hat{\mathbf{U}}_c \rangle_{(i-1)}^{-1} \cdot \hat{S}_c \qquad \text{(F. 13a)}$$

and

$$\mathbf{U}_{c(i)} = U_0 \mathbf{E}_n + \Delta \mathbf{U}_{c(i)} \qquad \text{(F. 13b)}$$

According to experiences, the iteration is divergent only in the limit cases of the steady state stability and in the cases near to network resonance (very long, unloaded transmission line, etc.), in other cases it converges.

ad b) A common feature of these methods is that the algorithm takes one after the other the equations of the non-linear equation system forming the mathematical model and it corrects the value of the node voltage of the proper number of every equation, and at the next equation already this corrected value is taken into consideration.

A big group of the solutions starts from the matrix equation (1a) which is of power dimension and performs the iterations according to Newton's method.

The other main group starts from Eq. (F. 10) and applies one of the approximate methods used for the solution of linear equations — for example the Gauss—Seidel method — for the calculation of the elements of ΔU_c.

In general, these methods converge essentially slower than that presented under a), thus more iteration steps are needed for the solution, there is, however, no inversion and generally the operation demand of the individual iterations is reduced.

Summary

A special load-flow algorithm and digital computer program based on inversion-iteration method is described, where the inverse of the nodal admittance matrix is not formed in the usual matrix form, but a so-called "product-formed" inverse is computed; the informations of this latter are stored in a very compact form by the algorithm, highly reducing thereby the operation demand of the iterations and the necessary machine store capacity. The algorithm for the computation of optimal node-sequence is also described.

Dr. László RÁCZ
Zoltán REGULY
Attila BERTALAN

Budapest XI., Egry József u. 18, Hungary