SOLUTION OF THE COMPLEX PIPE NETWORK PROBLEM

by

PAUL MIDDLETON

A thesis presented at the University
of Aston in Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

October 1968.

Summary

The main purpose of the work described in this thesis is the development of digital computer methods for the analysis of complex pipe networks systems. For the solution to this problem the technique of diakoptics has been proposed. A new development of the theory has been shown which it is hoped is more easily understandable to chemical engineers. A computer program has been written and tested with example networks from the literature and a test network derived by the author. The results show that the program is easier to use than existing methods. The method converges to a solution more rapidly and is very insensitive to the initial guess. The initial guesses do not have to conform to either of Kirchoffs Laws. Small changes in the network can be solved automatically with a minimum of extra input data. Very large systems can be analysed with only moderate demands on the fast access storage of the computer. It has been shown by using the theory underlying the method how the designer can quickly check to see if networks are under or over specified and when changing, for example, some parameter what design variables can remain at their present values and which must be relaxed.

Diakoptics can be applied to other branches of chemical engineering and it has been suggested how it can be used for the solution of finite difference approximation of partial differential equations and the solution to systems containing mixed linear and non linear elements. Finally it has been suggested how the method can be used to form automatically the describing equations of highly complex systems.

ACKNOWLEDGEMENTS

The Author gratefully acknowledges the assistance of the following:-

Professor G.V. Jeffreys

For providing facilities for the work to be carried out in his department.

Dr. B. Gay

For his supervision of this project and the many suggestions and constructive comments on the preparation of this thesis.

Science Research Council

For their research studentship grant.

Miss J.Y. Bishop

For her diligent typing of this thesis.

Chapter 1

Introduction

Introduction

The design, optimisation, and analysis of fluid distribution systems is of considerable engineering and economic importance; the obvious examples being the gas and water distribution industries. In process plant design however between 30% to 50% of the capital cost is taken up by piping (1) and so considerable savings in capital and running costs could be achieved by optimisation. The very large amount of calculation required however has until recently been prohibitive. The widespread availability today of computers and the growth of computer orientated techniques has drastically changed the situation and it is now possible for such analysis to be undertaken.

The purpose of this thesis, is to describe the application of modern computational techniques to the problem of pipe network analysis. This problem is related in its essential details to the analysis of electrical systems (2), stress-strain analysis in frames (3), and diffusion processes (4).

It is therefore not surprising that the first systematic approach to the problem appears to have been by the civil engineer Hardy Cross, and that the method proposed below has been developed from the method of the electrical engineer Gabriel Kron for solving large electrical power distribution systems.

Now for any computational technique, it should not be a requirement that the persons using the program have a detailed knowledge of computers or any specialized branch of mathematics.

Therefore the data for the program should be easy to prepare with no precalculation required. The data format should also be completely unambiguous and it should not be possible to affect adversely the rate of convergence, by any unfortuitous selection of input parameters. In operation the program should be efficient in time and storage required. Most important for the designer, small changes in the specification should be capable of rapid analysis so that a large number of possibilities can be tried for an optimum solution to be found. It is with these considerations in mind that the present study has been carried out.

Chapter 2.

Literature Survey.

A. Introduction

The Hardy Cross method of analysis (5) will be considered in detail first. This is because it appears that all the methods for pipe analysis so far reported are based on this technique, with only minor modifications to include, for example, more realistic friction factors.

Hardy Cross based his analysis on Kirchoffs Laws i.e. for any solution:

1) The sum of the flows at any node (pipe junction) is zero.

2) The sum of the potential (pressure) drops around any closed mesh (loop) is zero.

A pressure drop-flow relationship is also required.

The two laws lead to different iteration schemes. If the flows are taken as the unknowns, one iterates until the requirement of the second law is satisfied, starting with an initial guess of the flow distribution which satisfies the first law. Conversely if the pressures are the unknowns one iterates until the requirements of the first law are realised.

B. The Iteration Schemes

It is assumed that the head lost in any pipe can be expressed by:

$$h = rQ^{n'}$$

For a solution to the problem Σh around any mesh must be zero

$$\text{i.e. } \Sigma\, rQ^{n'} = 0$$

Then for any pipe in the mesh with an initial guess Q_o for the flow

$$Q = Q_o + \Delta$$

and

$$rQ^{n'} = r(Q_o + \Delta)^{n'}$$

expanding

$$rQ^{n'} = r(Q_o^{n'} + nQ_o^{n'-1}\Delta + n'(n'-1)\ Q_o^{n'-2}\ \Delta^2/2! + \ldots\ldots)$$

Then if Δ is small and $\Sigma rQ^{n'} = 0$ we can write that for any given mesh

$$-\ \Sigma rnQ_o^{n'-1}\Delta = \Sigma rQ_o^{n'}$$

and if the correction factor is assumed constant for any given mesh

$$\Delta = -\Sigma rQ_o^{n'}/\Sigma rnQ_o^{n'-1} \qquad (2)$$

i.e.

$$\Delta = -\Sigma h/\Sigma R$$

where Σh is with due reference to the direction of flow

and ΣR is without due reference to flow direction.

For a given network Hardy Cross selected his meshes by eye based on experience (it will be shown later* that the number of basic meshes of any network equals the number of branches plus one minus the number of nodes i.e. m = b - n + 1)

Then knowing the inputs to the system he assumed a flow distribution which satisfied Kirchoffs first law.

For the first mesh he calculated the pressure drop h for each branch in the mesh from his simplified flow relationship.

$$h = rQ^2$$

Then having calculated Δ for the first mesh, the new branch flows are found by the addition of this term to each branch flow. This process is repeated for all basic meshes. The whole cycle is repeated

* Section on Topology and Graph Theory.

by returning to the first mesh, until the second law is satisfied on all meshes.

In the second scheme nodal pressures are assumed and the resultant branch flows calculated. From this data the flows incident at the first node are summed and the excess or deficiency found. This is then distributed to the incident pipes in an inverse proportion to the resistance ($R = nrQ_0^{n'-1}$). The process is then repeated until the first law is satisfied.

Hardy Cross draws attention to the fact that the truncation of the binomial expansion is justified only if Δ is small and that the exponent n' is less than one.

At the start of the process however, Δ can be very large and of course n' is always greater than one. In general n' lies between 1.0 and 2.0 depending on the Reynolds number. However since some branches are members of more than one mesh or incident to more than one node they are corrected a number of times per complete iteration cycle. He therefore maintained that the convergence was sufficiently rapid for practical purposes.

Hardy Cross developed his method for hand calculation which implies small networks. It will be shown that the methods are critically dependent on either the choice of basic meshes or the order in which the nodes are taken.

Both methods can be classified as relaxation techniques; the speed of convergence being determined by the experience of the calculator who develops a 'feel' for the problem. For large networks involving a computer solution this experience or feel is exceptionally difficult or even impossible to program and so a pre-determined solution pattern must be followed which can lead to very long and inefficient convergence.

A more sophisticated relaxation technique based on the second law has been reported by van der Berg (6). He developed a system of correction factors that operated on the nodal pressures in such a way that the flow residues were eventually reduced to zero. The order of calculation being determined by a numerical criterion.

C. The Method of Van der Berg

It was stated above that the rate of convergence depended on the order in which the nodes were taken. Van der Berg constructed an integral, the value of which determined which node was to be corrected next to obtain maximum convergence.

He plotted a graph of the residue

$$r_i^{(o)} = \sum_j Q_{ij}^{(o)} + I_i'$$

against pressure (see fig. 1.)

The node to be corrected first has the maximum value for the integral

$$\mathcal{I}_i = \int_{p_i^{(o)}}^{p_i^{(1)}} r_i dp_i \quad i = 1,2,3, \ldots\ldots\ldots$$

when each node is considered in isolation (i.e. all other nodal pressures are held constant).

Now $\mathcal{I}_i$ can be seen to be approximately equal to the area of a triangle (see fig.1.)

$$\text{i.e. } \mathcal{I}_i \approx \tfrac{1}{2}\, r_i^{(o)} \, (\bar{p}_i - p_i^{(o)})$$

Where $\bar{p}_i$ is the approximate value of the nodal pressure which

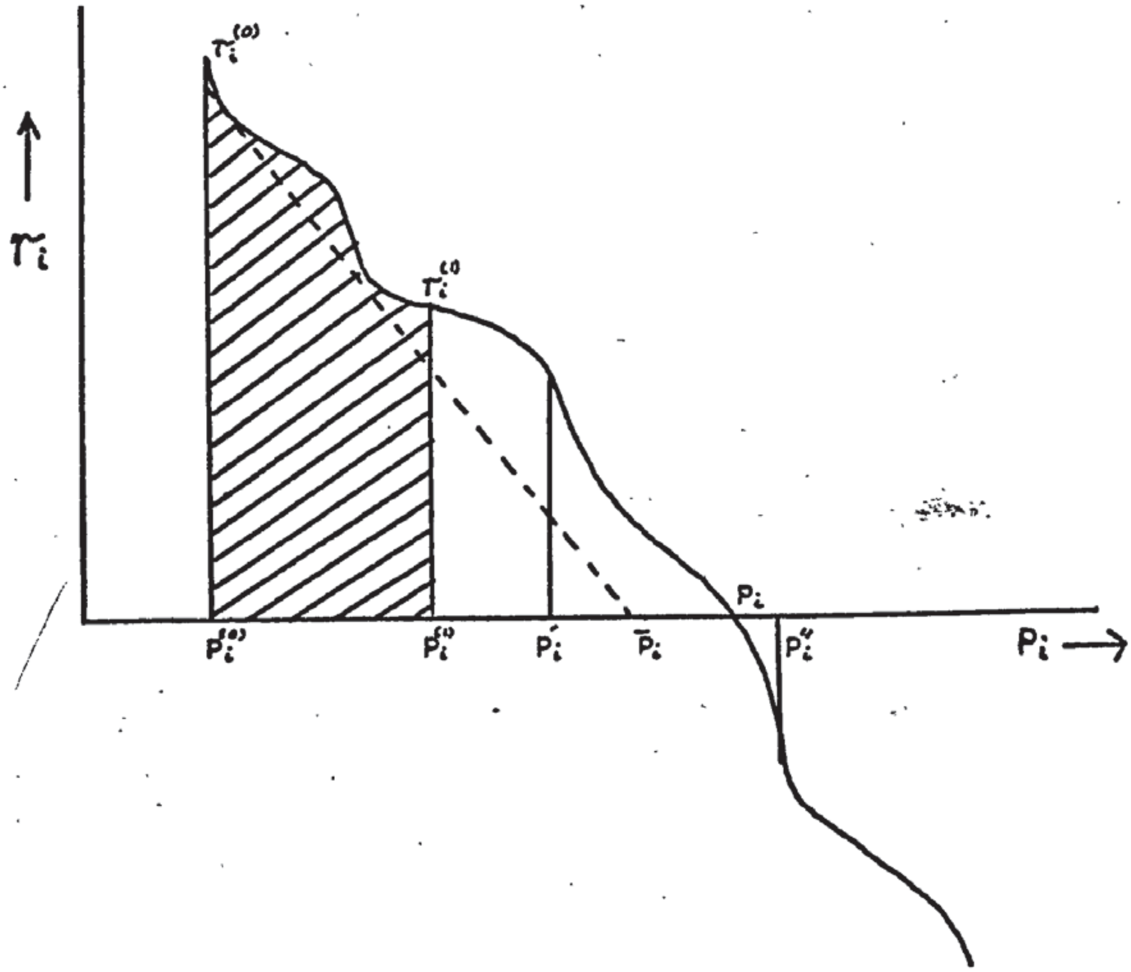


FIG I

RESIDUE AS A FUNCTION
OF PRESSURE
from van der Berg

reduces the residue to zero. $\bar{p}_i$ can be determined fairly quickly by putting $p_i^{(o)}$ equal to the pressures of the adjacent nodes $p_j^{(o)}$ and determining any two pressures p' and p'' for which the value of the residue changes sign. Then $\bar{p}_i \approx \frac{1}{2}(p' + p'')$

Now as the calculation proceeds the value of the residue drops and the values of $p_i^{(o)}$ move into the interval p' to p''. Van der Berg derived a more accurate expression for the value of the new pressure that reduces the residue to zero.

$$\Delta p_i^{(o)} = n' r_i^{(o)} / (\sum_j Q_{ij}^{(o)} / p_j^{(o)} - p_i^{(o)})$$

and

$$p_i^{(1)} \approx p_i^{(o)} + \Delta p_i^{(o)}$$

This method has the advantage that recalculation after small changes can be speeded up to some extent, because of the knowledge of where to start the corrections. However, for large systems one cannot follow a true optimum strategy for node selection. Not only because of the approximation inherent in the method, but also because having changed one node the integral values in the area surrounding the first chosen node have changed considerably, thus necessitating their recalculation.

Van der Berg maintained however that it is possible to overcome this feature by letting the new nodal pressure leave a residue that has some value greater or less than zero. This accelerating factor being determined by the users experience.

The next section of the literature survey is concerned with the mechanisation of the basic process due to Hardy Cross, into a suitable form for computer use. It would appear that no further work has been published on the method due to van der Berg although it would seem at least to limit the arbitrariness of the iteration pattern.

D. Survey of Computer Applications

Kniebes and Wilson (7) were amongst the first to report a computer solution. They used a straightforward Hardy Cross mesh analysis using a value for n of 1.8 . The data were presented in the form of tables of pipe data and tables of loop members. They found that the solution was efficient for large error criterion but the number of iterations increased markedly if greater accuracy was required. They also found that the program was most efficient for systems of the order of 250-400 pipes.

i) The Program due to Hunn and Ralph.

The program due to Hunn and Ralph (8) is a more sophisticated version allowing for the inclusion of pumps or other non-pipe elements that have a pressure drop or rise vs flow relationship that can be expressed as a polynomial.

Their program was written and run in sections because of machine size limitations; the computer used was an I.B.M. 650 with 2000 word memory and five segments or drum loads.

One interesting feature of the program is a section which calculates a feasible solution i.e. one that is in material balance at each node. This is accomplished by extra input data in the form of a trace. The trace is a sequence of nodes starting at the datum node which runs through all the nodes in the system at least once. Then starting at the datum node with a given input flow, branch flows are assigned to each pipe in the trace.

This trace is also used at the end of the calculation to assign to nodes their appropriate pressures from a knowledge of the individual pipe pressure drops.

Hunn and Ralph state that the construction of this trace is probably the most critical operation of the entire data preparation phase. However it has been shown by Daniel(13) and the author that the initial guess will affect to a certain extent the rate of convergence but that the loop formation is the only input data which is critical for convergence.

The large amount of data to be prepared and punched onto a suitable input medium for the computer, presents a considerable problem because mistakes can easily occur, if these are not detected the program may not converge so wasting valuable time or more seriously it could converge to the wrong solution.

Hunn and Ralph's program included data checks, to test for example that all loops are closed paths and that no more than two branches are incident at one node in any loop.

The input data format was however very complicated and needed a skilled coder. This is not so much a property of the program as of the very limited input capability of the early machine which was used.

ii) Ingels and Powers

It has been shown however by Ingels and Powers (9) that calculations based on the Hazen-Williams equation with a constant value for the exponent n' can be seriously in error. A typical error being about 20% for flows of approximately 10^5 lb/h in 6" pipes.

They used a more realistic flow equation developed earlier by Ingels (11) which approximated the relationship of friction factor versus Reynolds number of the Moody (10) diagram.

For Re>2,100 a power series of the following form was used

$$\phi = a \pm c\theta + d\theta^2$$

where $$\theta = \left[-b + \log Re\right]^{-1}$$

and a,b,c,d are polynomial functions of the relative roughness.

The friction factor was then used in the Darcy-Weisbach equation (12)

$$h = 8\phi L\, Q^2/g_c \pi^2 D^5 \qquad (3)$$

They also show that truncating a Taylor series expansion of the expression $h = f(Q)$

the resistance term R in the Hardy Cross expression for the correction factc is equivalent to $\partial h/\partial Q$.

∴ R can be written

$$R \equiv \partial h/\partial Q = (8L/g_c \pi^2 D^5)\ \left[2\phi Q + Q^2\ \partial\phi/\partial Q\right]$$

Substituting for $\partial\phi/\partial Q$ and simplifying

$$R = 2h/Q - \left[(KQ\theta^2)\ (\log e)\ (c + 2d\theta)\right]$$

where $$K \equiv 8L/g_c \pi^2 D^5$$

The empirical relationship works well for turbulent flow in rough pipes but varies considerably from the Moody diagram for low values of roughness at high Reynolds numbers.

It is felt however that these expressions are more complex than is necessary and a better method for the calculation of realistic friction factors will be proposed below.

The initial estimates of flow were produced by a separate program, based on the assumption that the individual pipe segments had been sized on economic considerations (14) i.e.

$$D = 2z\rho^{0.14}\,Q^{0.45}$$

$$\therefore Q = 0.17\,D^{2.22}/\rho^{0.31}$$

or

$$Q \approx 0.17\,D^{2}/\rho^{0.31}$$

So that starting from the major source of inflow and with a knowledge of pipe diameter and loads the program proportions the flow down each pipe by a simple second power relationship. These flows are then used as inputs to the main program.

Using equation (3) they analyse three networks previously reported. The largest of these, due to Dolan (15), will be discussed below in the results section together with the results obtained by the author.

iii) <u>Knights and Allen</u>

A computer solution based on Hardy Cross's second method has been reported by Knights and Allen (17) This method was chosen by them because in a preliminary analysis of the methods available they thought that its advantages of simpler data preparation, and programming together with more certainty of result seemed to outweigh the fact that convergence was slower

The main criticism of their method apart from the arbitary node numbering is the calculation of the friction factor using the Drew and Genereaux correlation (16).

i.e. $\phi = 0.0351 \, Re^{0.152}$

Their equation suffers from the usual errors inherent in the straight line plots when compared with a Moody diagram. The results given below however show that in the range of Reynolds numbers encountered in their test network agreement in the main is quite good.

iv) The Computer Solution of Daniel

The most comprehensive application of the Hardy Cross technique in respect of accuracy and ability to handle compressible as well as incompressible flow systems seems to be due to Daniel (13). In his treatment he adds another cycle to the basic Hardy Cross iteration scheme. This outer cycle is entered when convergence has been reached, and recalculates the resistance factors by accurate determination of friction factors from $\phi = 1/\{0.86859 \ln[\epsilon/3.7D + 2.51/Re\sqrt{\phi}]\}^2$ and in the case of compressible flow, the values of density and viscosity are also calculated from their respective polynomials. The inner cycle is then re-entered.

This has the advantage that, although accurate friction factors are used, the iterative procedure needed to calculate ϕ for each branch is only used outside the main basic cycle.

Daniel also systematises the calculation of the correction factor and its sign by the use of a branch-mesh incidence matrix. A full description of the properties of this matrix and other topological relationships is included in a later section so that only the equations will be given here.

$$Q_i^{(r+1)} = Q_{ij}^{(r)} + C_{mk} \Delta_m^{(r)}$$

where C_{mk} is an element of the branch-mesh incidence matrix.

and

$$\Delta_m^{(r)} = \frac{-\sum_{k=1}^{b} C_{mk} \operatorname{sign}(Q_{ij}^{(r)}) R (Q_{ij}^{(r)})^{n'}}{\sum_{k=1}^{b} n' \left| C_{mk} R (Q_{ij}^{(r)})^{n'-1} \right|}$$

The use of the branch-mesh incidence matrix has the advantage that the sign of the mesh correction factor is obtained automatically. However there are two serious disadvantages which can be illustrated with reference to fig. 4.

The matrix has the dimensions of branch x mesh so that for any real system very large amounts of storage are required, much of which is set to zero. The size of the matrix also increases the computation of the correction factors as the summation terms have to cycle branch times for each mesh. The other programs described above use list processing i.e. the information on shape is input as a list or vector and not as a matrix and the author having tried both methods has found the latter not only much more economical on storage but also computationally much more efficient.

E Discussion of Computer Solutions

i) Introduction

Before discussing this work it will be useful to re-examine the network problem in such a manner that the relative ease of executing each step by hand and by computer can be compared. Such analysis will show better the shift in emphasis required when moving to a computer solution. Assuming that the problem has been specified i.e. the network has been given together with the size of each branch and the properties of the fluid, the solution steps can then be broadly stated as:-

1) Presentation and assimilation of data.

2) Choice of the iteration pattern i.e. having numbered the branches and the nodes, the fundamental loops and the order in which these will be used are chosen; or the order in which the nodes will be taken is chosen.

3) The actual arithmetic of the iteration cycle is executed.

4) A decision on whether the system has converged is taken.

5) If it has not converged then the calculation returns to step (2)

For a hand calculation steps (1) and (2) may be to some extent time consuming but the real problem is the calculation. This is because one can look at the system as a whole and therefore decisions on loop formation or numbering are fairly easy and in the light of experience one can easily change the order of the calculation or even the shape of the loops.

A computer however is a sequentially operating machine 'looking' at only one number at a time. One of the main problems therefore is the format of the data which tell the machine the structure of the network and its constituent loops. This sequential nature also precludes any change due to 'feel' which one obtains from considering the system as a whole. Once a pattern is established then the machine must rigidly adhere to it. In fact even the format of the meshes must be input as data as these cannot be formed by the machine without a crippling additional computational load. Steps (3) and (4) are however no problem since arithmetic operations are easy to program and efficient in operation.

ii) Discussion of Daniels Solution

The importance of good data handling and easy presentation in computer solutions of network problems can now be more easily understood. In the Hardy Cross mesh analysis however the data preparation is complicated by having to choose the basic meshes. Maximum convergence is achieved by a choice of meshes which has the property of minimum overlap. That is the number of branches per mesh is a minimum. The logical method of mesh selection is through the use of trees and links. A tree is any path through the network which contains all the nodes so that it is possible to move along the tree between any two nodes. A link is any non tree branch which, when added to the tree forms a mesh. Daniel uses this method for defining meshes, but this just transfers the problem from selecting the meshes to finding a defining tree. The automation of this selection is a considerable computational problem and was not included in Daniel's method. It is relatively simple to find defining trees and therefore sets of basic meshes but, for example, the test network fig 16 has about 350 million trees each defining a set of meshes and to analyse them all for minimum overlap would be prohibitive. The easiest tree to find automatically is the 'trunk'. This tree passes in sequence through the nodes and therefore contains no side branches. Unfortunately this tree has the property of maximum overlap.

A program was written following Daniel and used for the test network fig 16. The two extreme cases i.e. minimum and maximum overlap were run. The minimum overlap condition converged to a solution in thirty minutes. The other case had not converged but was oscillating around the convergence criterior after two hours when it was stopped. For more detailed description of Daniel's program see Appendix G and results section.

The survey of the basic development is now complete. In the work to follow a completely new approach to the problem will be proposed. For this reason the above survey is not a complete record of all the published work, but includes only those papers which give a history of the problem and how it has developed.

It was suggested at the start of this particular research project that, as electrical engineers had most experience in solving network problems, some of their techniques could be adopted to the pipe flow problem. This line of inquiry lead almost immediately to a study of linear graph theory and the technique of Diakoptics.

Chapter 3.

Network Analysis and Development of the Diakoptics Method.

A. Introduction

The development of Diakoptics will start with an introduction to the basic topology and linear graph theory of networks. From the properties and relationships discussed in this section the classical electrical network relationships will be developed; this section being based on the work of Branin (18) and Roth (19). Having discussed these, Kron's original view of the same problem, which lead him to the development of Diakoptics will be outlined. The development itself is different from previous work, and it is hoped that in its present form will be more easily understood by most engineers.

It will be realised as the development progresses that Diakoptics is not only a powerful numerical technique but a completely new approach to the way in which engineers can think about and express problems. It is felt that the importance of this new approach to model building and analysis could be even more valuable than the methods undoubted power as a computational tool.

B. Topology and Graph Theory.

i) Definition of terms

A graph of a network is a diagrammatic representation of the network. It consists of branches, which correspond to the individual pipes, and nodes between which the branches run. A branch is said to be incident at its terminal nodes. A graph is said to be directed if assumed directions of flow (for instance) are indicated. The graph of a small network is shown in fig. 2. Note that a graph describes only the topology of a network. No information such as physical dimensions, hydrodynamic resistances, flows etc. is provided. The graph in fig. 2. is also said to be connected i.e. it is possible to move along the branches between any two nodes. Any connected graph contains at least one tree. A tree of a connected graph is any set of branches which connect all the nodes but does not form any meshes (closed loops). Hence if a connected graph has n nodes then any tree will contain n-1 branches. The term basic mesh is used to describe any closed path formed by a non-tree branch (or link) between the terminal nodes of any part of the tree. For example in Fig. 2. we may select the tree formed by branches 1,3,5,6 and 7 (shown as heavy lines) Consequently this tree forms three basic meshes containing the branches 1-3-4, 2-3-5, 5-6-7-8. It follows that for any graph the number of basic meshes is given by $m = b - n + 1$

In a directed graph the meshes are also orientated and it is convenient to define the mesh direction as that of its defining link.

ii) Matrix Representations

For computational purposes the graph is conveniently described by certain matrices. Fig. 3. shows the augmented incidence matrix $\underline{A}'$ for the graph in Fig. 2. The rows of $\underline{A}'$ correspond to the branches and the columns to the nodes of fig. 2. An element a_{ij} is +1, -1, or 0 if the i^{th}

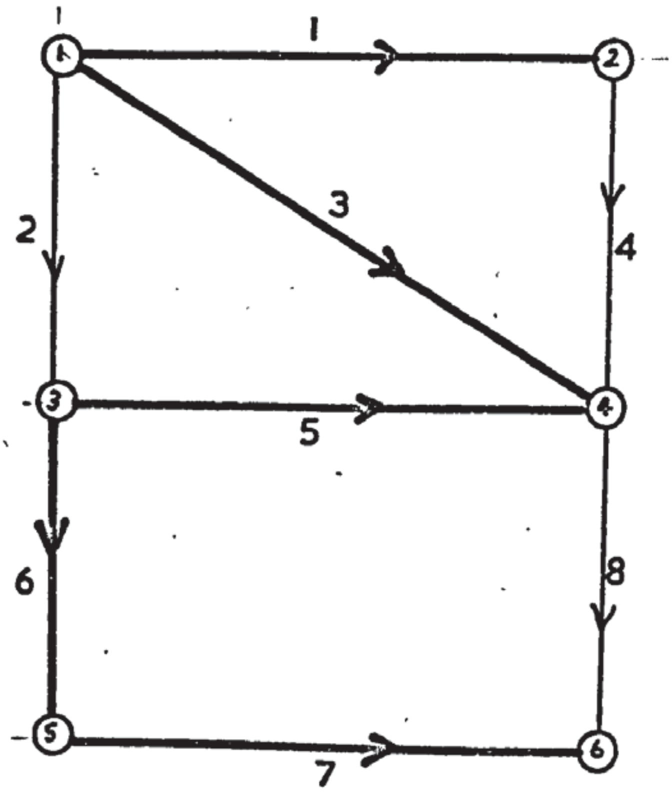


FIG 2

THE GRAPH OF A NETWORK

	1	2	3	4	5	6
1	-1	+1	0	0	0	0
2	-1	0	+1	0	0	0
3	-1	0	0	+1	0	0
4	0	-1	0	+1	0	0
5	0	0	-1	+1	0	0
6	0	0	-1	0	+1	0
7	0	0	0	0	-1	+1
8	0	0	0	-1	0	+1

Fig. 3 Augmented incidence matrix for Fig. 2

Branch	2	4	8
1	0	+1	0
2	+1	0	0
3	-1	-1	0
4	0	+1	0
5	+1	0	+1
6	0	0	-1
7	0	0	-1
8	0	0	+1

Fig. 4 Branch mesh matrix for Fig. 2

branch is positively negatively or not incident at node j. Clearly the sum of the elements in any row is zero and the columns are therefore not linearly independent. Hence we may delete any one column. The node corresponding to this column is then called the datum node and the matrix formed constitutes the incidence matrix $\underline{A}$ of the graph.

The basic meshes of a graph are conveniently described by its branch-mesh matrix $\underline{C}$ whose columns correspond to the links and the rows to the branches of the graph. Fig. 4. shows the branch-mesh matrix for the tree shown in heavy lines in Fig. 2. Any element C_{ij} is +1, -1, or 0 if the i^{th} branch is positively , negatively or not included in the j^{th} basic mesh.

It is readily shown that $\tilde{\underline{A}}\,\underline{C} = \underline{0}$ and that $\tilde{\underline{C}}\,\underline{A} = \underline{0}$

iii) Relationships between Node, Branch and Mesh Quantities

Now in general we can associate certain quantities or variables with the nodes, branches and meshes of any graph. The function of the matrices $\underline{A}$ and $\underline{C}$ is then to inter-relate these quantities. These relations are termed transformations and the matrix $\underline{A}$ will transform nodal quantities to branch quantities and $\tilde{\underline{A}}$ will transform branch quantities to nodal quantities. Similarly $\underline{C}$ transforms mesh quantities to branch quantities and $\tilde{\underline{C}}$ branch to mesh quantities. Note that the variables in the expressions below are only expressed as flows and pressures by way of example and for the sake of clarity.

If for example we assign arbitary quantities e' to the nodes and denote these by the vector $\underline{e}'$ then premultiplication by $\underline{A}$ assigns a vector $\underline{e}$ to the branches.

$$\underline{e} = \underline{A}\,\underline{e}' \qquad (4)$$

if $\underline{e}'$ is the vector of node-to-datum pressures then it is easily seen the $\underline{e}$ is the vector of pressure rises across the branches. In the same way branch quantities may be assigned to the meshes by the matrix $\underline{\tilde{C}}$ However if this transformation is applied to the vector $\underline{e}$ we find

$$\underline{\tilde{C}}\,\underline{e} = \underline{\tilde{C}}\,\underline{A}\,\underline{e}' = \underline{0} \qquad (5)$$

But, if additional arbitary branch quantities are represented by the vector $\underline{E}$, then $\underline{\tilde{C}}$ assigns non-zero quantities $\underline{E}'$ to the meshes.

$$\underline{E}' = \underline{\tilde{C}}\,\underline{E} \qquad (6)$$

Similarly one may assign quantities to the meshes and relate these to the branches and nodes. If vector $\underline{i}'$ represents a set of mesh quantities it is transformed by $\underline{C}$ into corresponding branch quantities.

$$\underline{i} = \underline{C}\,\underline{i}' \qquad (7)$$

The transformation of $\underline{i}$ by $\underline{\tilde{A}}$ into nodal quantities again yields a null vector.

$$\underline{\tilde{A}i} = \underline{\tilde{A}}\,\underline{C}\,\underline{i}' = \underline{0} \qquad (8)$$

However additional quantities $\underline{I}$ associated with the branches may be transformed into non-zero nodal quantities:

$$\underline{I}' = \underline{\tilde{A}}\,\underline{I} \qquad (9)$$

These topological transformations are summarised by the upper and lower halves of the algebraic diagram fig. 5. which is due to Roth (19)

Note however that having transformed quantities in one direction it is impossible due to the shape of the matrix i.e. because they are not

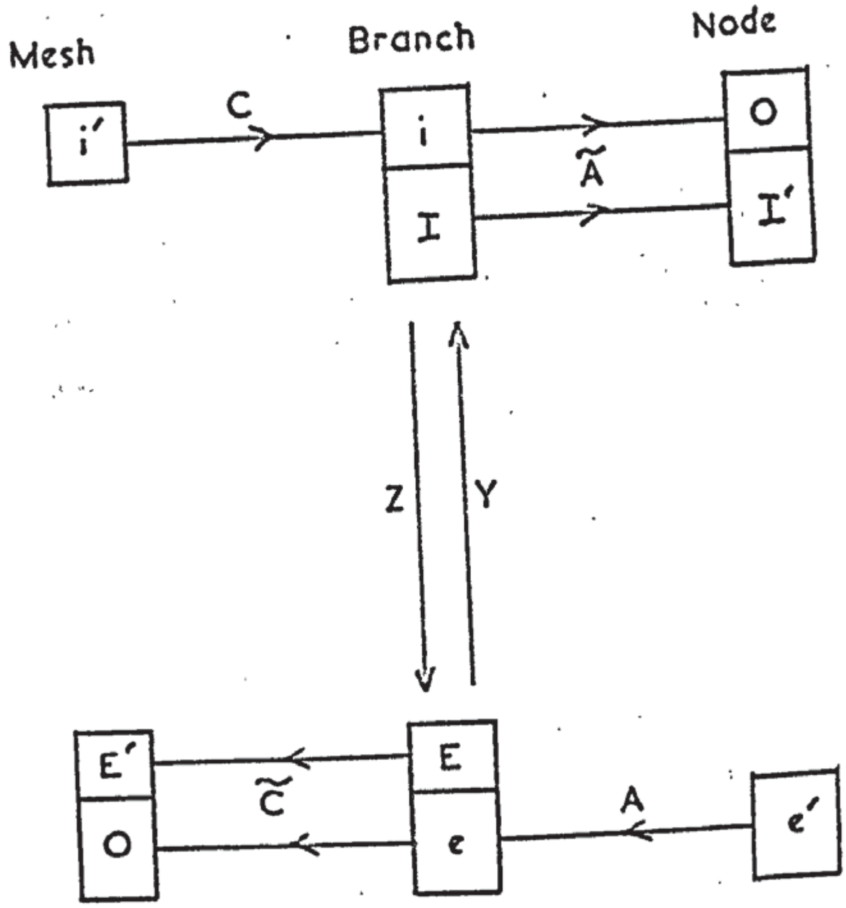


FIG 5

ALGEBRAIC DIAGRAM

due to Roth

square, to reverse the process. The development of square non-singular matrices and their importance will be discussed below.

C Network Relationships

In applying these transformations to networks the above quantities can be identified with physical quantities. I.e. $\underline{e}$ and $\underline{i}$ correspond to the potential (pressure) rises and currents (flows) in branches. E and I are the potential sources (pumps) and current sources or demands on the branch when treated in isolation.

Therefore each branch may contain three distinct elements: an impedance or admittance element, a potential source and a current source (see fig. 6.) By convention as can be seen from fig. 6. the potential source is orientated such that

$$V_r = E_r + e_r \quad \text{or} \quad \underline{V} = \underline{E} + \underline{e} \tag{10}$$

and

$$J_r = I_r + i_r \quad \text{or} \quad \underline{J} = \underline{I} + \underline{i} \tag{11}$$

Now V_r and J_r are the potential across and current in the impedance element and hence are related by an Ohm's Law type of equation.

$$V_r = Z_r J_r$$

and

$$J_r = Y_r V_r$$

Consequently the vectors $\underline{V}$ and $\underline{J}$ are related by the equation:

$$\underline{V} = \underline{Z}\,\underline{J} \tag{12}$$

and

$$\underline{J} = \underline{Y}\,\underline{V} \tag{13}$$

where $\underline{Y} = \underline{Z}^{-1}$

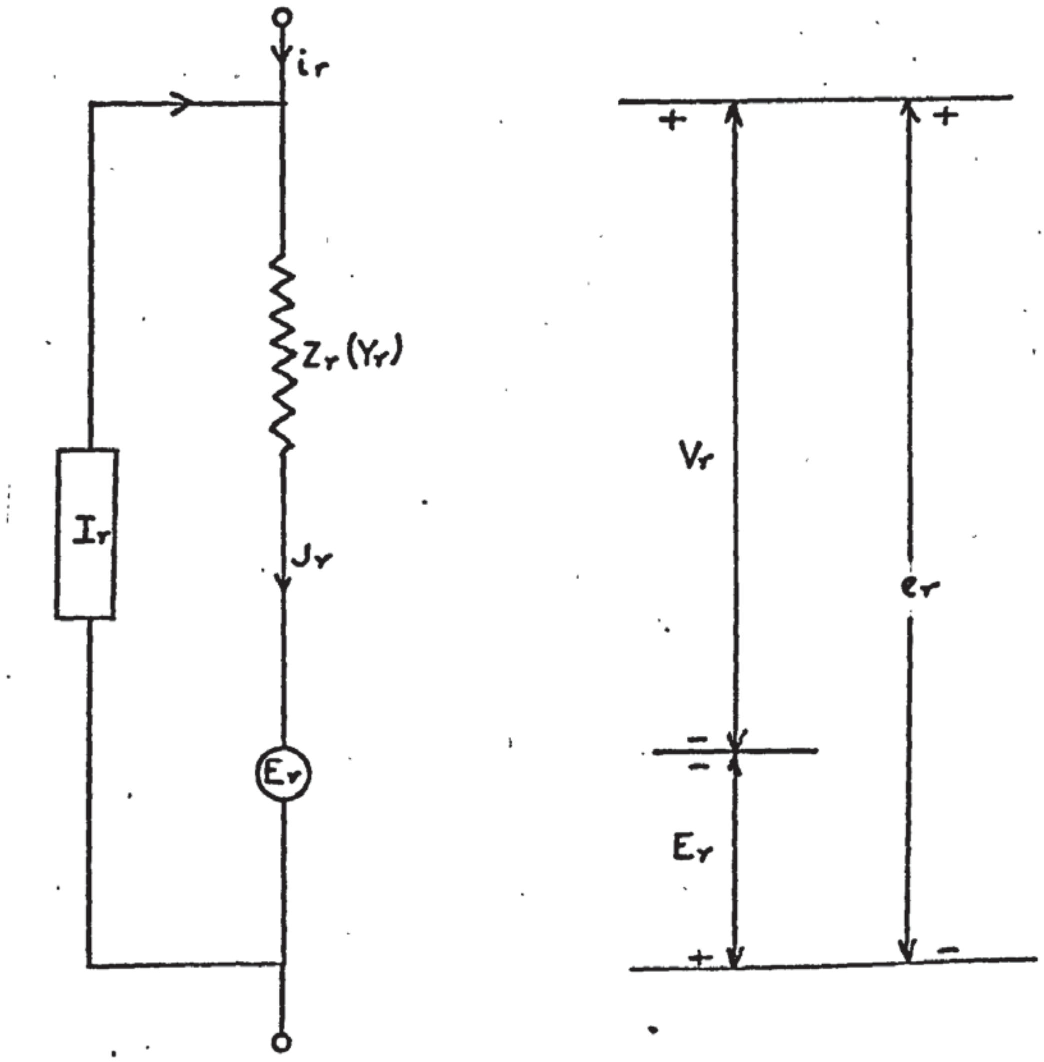


FIG 6

STRUCTURE OF THE r^{th} BRANCH OF A NETWORK

We are now in a position to combine the two sets of relationships i.e. the transformations and Ohm's Law for a solution to the network problem thus completing Roth's algebraic diagram. In addition note that equations (5) and (8) constitute a statement of Kirchoffs first and second laws.

D. Classical Network Analysis

i) The Connected Network.

It is very important to note at this stage that the network relationships equations (12) and (13) related to individual branches. The network is said to be in its primitive state and the matrices $\underline{Z}$ and $\underline{Y}$ are the primitive impedance and admittance matrices. They therefore only contain elements on the main diagonal.

Starting from the primitive equation (13) for example

$$\underline{I} + \underline{i} = \underline{Y}\,(\underline{E} + \underline{e})$$

Re-arranging and premultiplying by $\underline{\tilde{A}}$

$$\underline{\tilde{A}}(\underline{I} - \underline{Y}\,\underline{E}) + \underline{\tilde{A}}\,\underline{i} = \underline{\tilde{A}}\,\underline{Y}\,\underline{e}$$

Hence by equation (4) and (8)

$$\underline{\tilde{A}}\,(\underline{I} - \underline{Y}\,\underline{E}) = \underline{\tilde{A}}\,\underline{Y}\,\underline{A}\,\underline{e}'$$

$$\therefore\ \underline{e}' = (\underline{\tilde{A}}\,\underline{Y}\,\underline{A})^{-1}\ \underline{\tilde{A}}(\underline{I} - \underline{Y}\,\underline{E}) \qquad (14)$$

Equation (14) represents the nodal method of solution. All quantities on the right hand side of (14) are known from the specification of the problem (note that by equation (9) $\underline{\tilde{A}}\,\underline{I} = \underline{I}'$ the nodal vector of external currents) and hence the vector of nodal potentials $\underline{e}'$ may be found. The branch vectors $\underline{e}$ and $\underline{i}$ may then be calculated.

Alternatively, by a similar derivation from equation (12) the following relationship is obtained:

$$\underline{i}' = (\tilde{\underline{C}}\ \underline{Z}\ \underline{C})^{-1}\ \tilde{\underline{C}}(\underline{E} - \underline{Z}\ \underline{I}) \qquad (15)$$

Equation (15) constitutes the mesh method of solution. The vectors $\underline{i}$ and $\underline{e}$ being calculated directly from the vector $\underline{i}'$.

A worked example can be found in appendix A. for the network fig. 7.

ii) Large Networks

For large networks equation (14) has two serious computational disadvantages. Firstly the computer storage requirements increase markedly, data storage required being approximately $n^2 + 3n$ locations, where n is the number of nodes. The second and more important limitation is the time required to invert the matrix $\tilde{\underline{A}}\ \underline{Y}\ \underline{A}$.

In 1958 Kron (20) proposed a method of analysis called Diakoptics, which overcomes these difficulties. However the method was not widely used until, following the work of Roth (19) Branin (18) and Brameller (21), one aspect of this powerful analytical toolwhich could be said to be a logical extension of the classical methods outlined above, was developed.

E. Further Network Transformations and Diakoptics

i) Introduction

Kron's contribution to network theory is his application of the concept of invariance to networks subject to transformations. In this case the invariant property in power i.e. $\tilde{\underline{V}}\ \underline{J}$ or as Roth (19) has proved, that

a

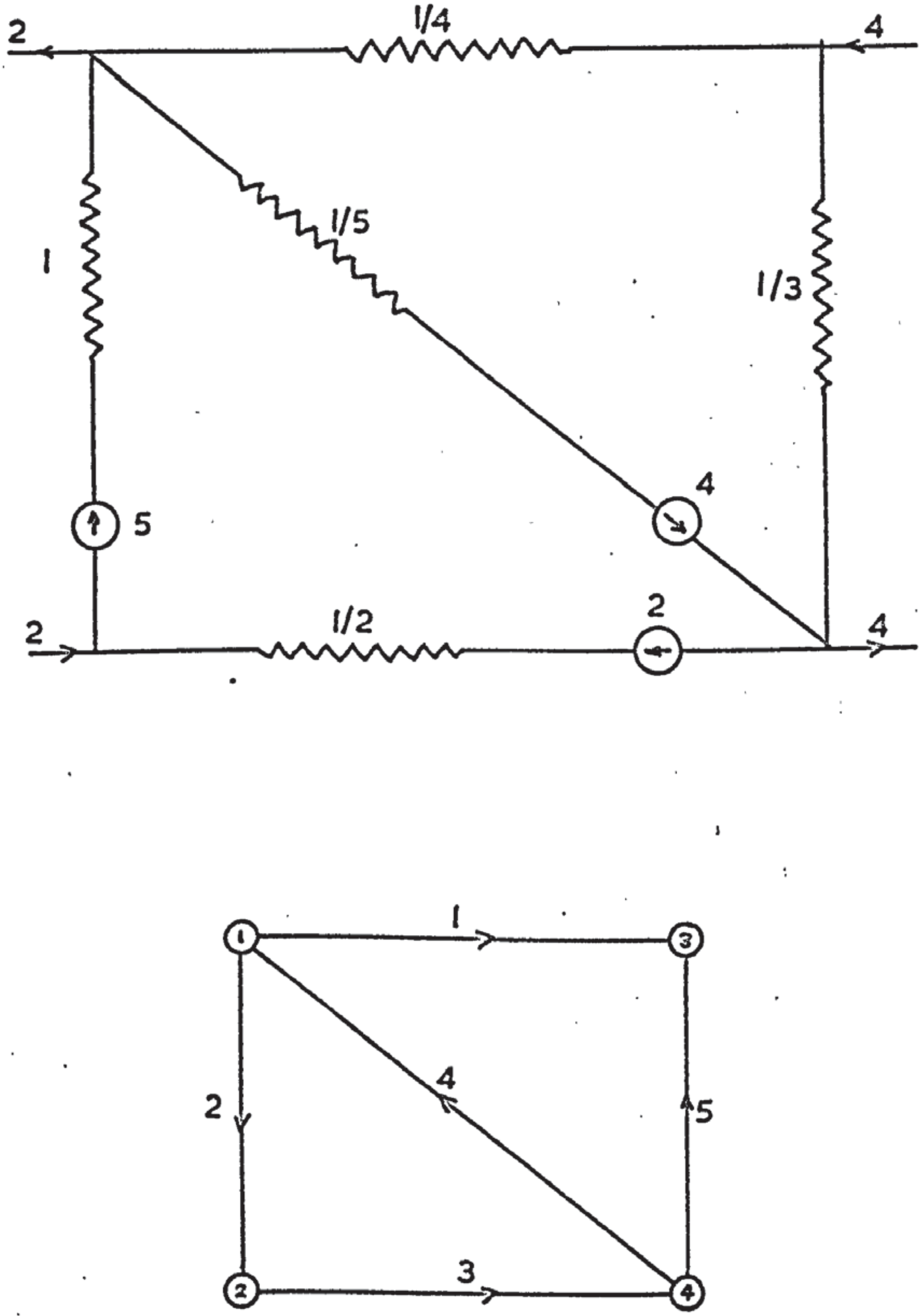


FIG 7

EXAMPLE NETWORK AND IT'S GRAPH

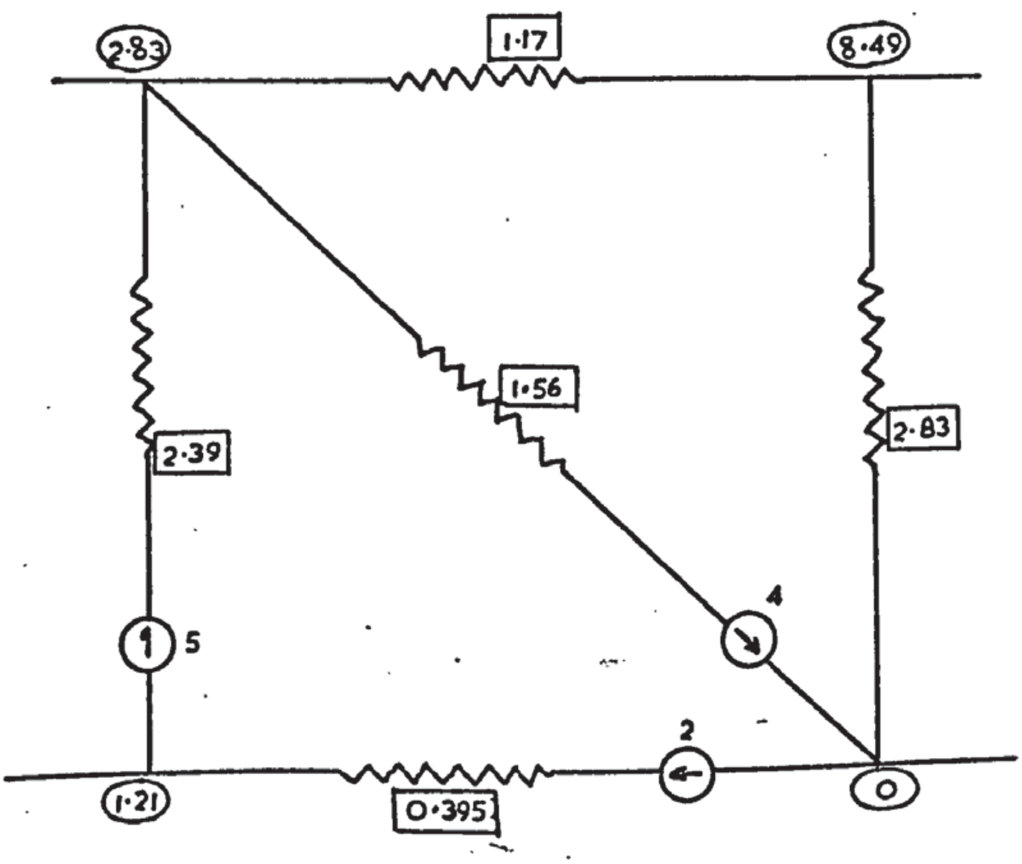


FIG 8

SOLUTION OF Fig 7 SHOWING NODAL POTENTIALS (in circles) AND BRANCH CURRENTS (in squares) a network exhibits 'ohmicness' The transformations considered are 'tearing' and 'reconnecting'.

The basic idea of Diakoptics is very simple: one tears the network into smaller pieces, solves each piece separately then reconnects for a solution. The storage requirements are then only those of the largest torn piece and since the matrices are smaller the time required for inversion is significantly reduced. Roth (19) has strikingly demonstrated the efficiency of the method. He tabulates the number of multiplications needed to solve a sixteen node linear network by various standard methods and by Diakoptics, table 1.

Diakoptics	368
K - Partitioning	618
Standard Partitioning	1647
Standard Inversion	4096

Table 1.

The key to Diakoptics is Kron's approach (22) to the original network problem outlined above. Instead of looking at the network from either a nodal or a mesh point of view which means the transformation matrices $\underline{A}$ and $\underline{C}$ have the dimension b x n and b x m, Kron's orthogonal network concept looks at the network from both points of view simultaneously. This means that his transformation matrices $\underline{A}_1$ and $\underline{C}_1$ are square and non singular It is this property that allowed Kron to develop a whole series of additional transformations, one of which will be explained in detail. A grasp of this orthogonal network concept is therefore essential for a complete understanding of Diakoptics and so an outline will now be given.

ii) The Orthogonal Network Concept.

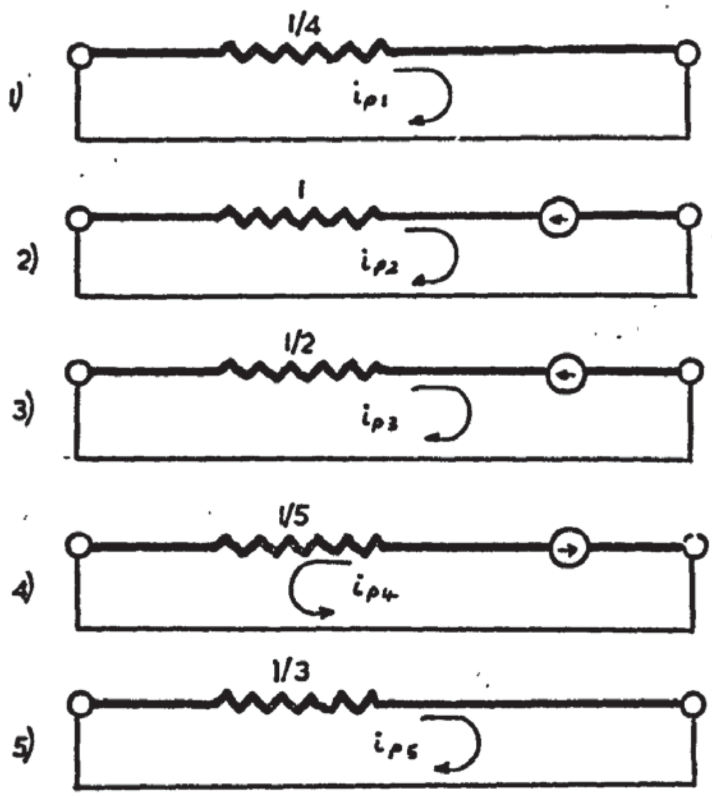
The transformation matrices $\underline{C}_1$ and $\underline{A}_1$ are formed by conversion of the given network to an all mesh or an all node-to-datum system.

Conversion to an all mesh network is accomplished by the addition of as many fictitious branches as there are non-datum nodes. Each fictitious branch is orientated from its associated node towards the datum node. Correspondingly an all node network is produced by opening the meshes, thereby producing extra nodes.

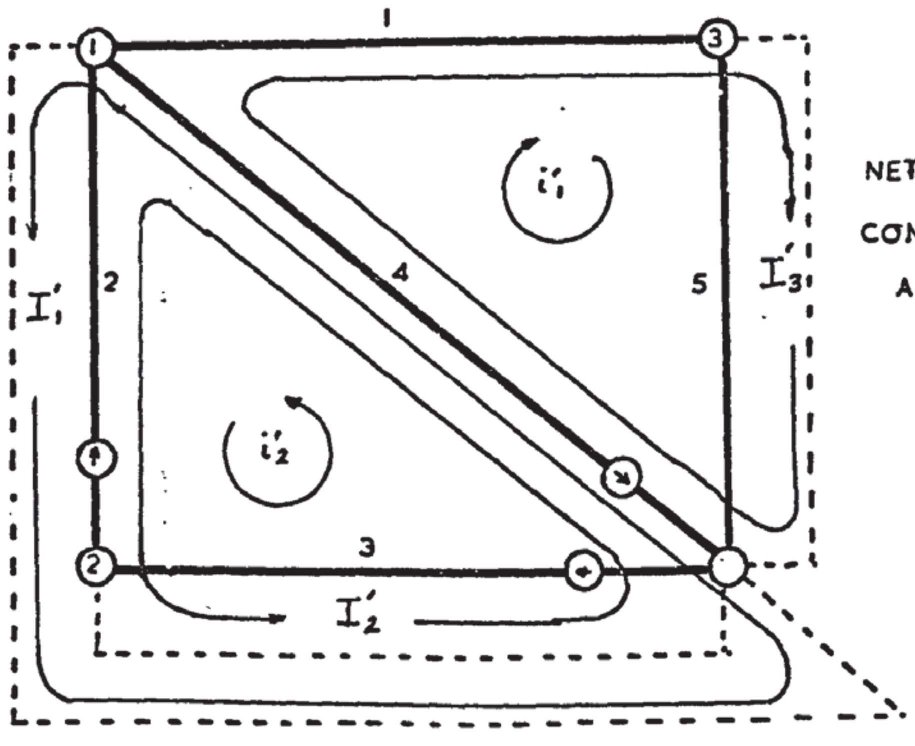
The conversion of the network in Fig. 7 to its primitive and all mesh forms is shown in Fig. 9. The nodal demands are now considered as mesh flows. The choice of paths through the network of these equivalent nodal flows can be taken in any arbitary manner. However the simplest method is to constrain them to flow along the branches of any tree of the graph as shown in Fig. 9. It is important to note that the directions of mesh flows are not defined by the links but, for the case of the nodal mesh flows, by the orientation of the fictitious branches and for the actual mesh flows, in any arbitary direction.

To form $\underline{C}_1$ one equates the branch currents in the primitive system to the branch currents in the all mesh network. The justification for this procedure is as follows.

Each coil or branch in the primitive system is short circuited. Now by the addition of the fictitious branches from the nodes to the datum (ground) point each coil in the connected network is in effect also short circuited. Therefore the branch flows in the primitive system are the same as those in the connected network and we can write:-



PRIMITIVE NETWORK OF FIG 7



NETWORK OF FIG 7 CONVERTED INTO AN ALL MESH NETWORK

FIG 9

$$i_{\rho 1} = i'_1 + I'_3$$

$$i_{\rho 2} = i'_2 + I'_2$$

$$i_{\rho 3} = i'_2$$

$$i_{\rho 4} = i'_1 + i'_2 + I'_1 + I'_2 + I'_3$$

$$i_{\rho 5} = -i'_1$$

or in matrix form

$$\underline{i_\rho} = \underline{C_1}\,\underline{J'} = \underline{C_1}\begin{bmatrix}\underline{I'}\\ \underline{i'}\end{bmatrix}$$

where

$$\underline{C_1} = \begin{array}{c|ccc|cc|} & I'_1 & I'_2 & I'_3 & i'_1 & i'_2 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 & 1 \\ 4 & 1 & 1 & 1 & 1 & 1 \\ 5 & 0 & 0 & 0 & -1 & 0 \\ \hline \end{array} \equiv \underline{C_{1j}}\;\underline{C_{1m}}$$

Note that $\underline{C_1}$ can also be formed by inspection as before by defining the meshes by their circulating currents taking no account of the fictitious branches e.g. I'_3 flows positively in branches 1 and 4 as indicated in the third column of $\underline{C_1}$. $\underline{C_{1m}}$ can also be seen to be identical with the $\underline{C}$ of the classical methods.

Kron also shows that $\underline{A_1} = \hat{\underline{C_1}}^{-1}$ is the transformation matrix for the corresponding all node network such that

$$\underline{e_\rho} = \underline{A_1}\begin{bmatrix}\underline{e'}\\ \underline{E'}\end{bmatrix}$$

$$\therefore \underline{A}_1 = \begin{array}{c} \begin{array}{ccccc} e_1 & e_1 & e_1 & E_1 & E_2 \end{array} \\ \left[\begin{array}{ccc|cc} -1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \end{array}\right] \end{array} = \underline{A}_{1j} \; \underline{A}_{1m}$$

$\underline{A}_1$ can also be formed by inspection as a consequence of restricting the nodal flows to the branches of a tree. It can be seen that $\underline{A}_{1j}$ is identical with the $\underline{A}$ of the classical methods and the elements of $\underline{A}_{1m}$ are entered in the link branches only and are positive or negative according to the direction of the assumed mesh flows and assumed direction of the link branch flows.

That $\underline{A}_1 = \hat{\underline{C}}_1^{-1}$ can also be proved directly from the proposition of power invariance as is shown in appendix D

The transformations developed previously are still valid but because the above transformation matrices are non-singular a different development of the equations of solution can be made.

For example assigning quantities $\underline{I}$ to the branches then

$$\hat{\underline{A}}_1 \, \underline{I} = \underline{I}' \quad \text{and} \quad \underline{I} = \underline{C}_1 \, \underline{I}'$$

note therefore that $\underline{I}'$ has the dimensions branch x 1. Also as before we can write

$$\underline{C}_1 \, \underline{i}' = \underline{i} \text{ and } \tilde{\underline{A}}_1 \, \underline{i} = \underline{i}'$$

and therefore $\underline{i}'$ has dimensions branch x 1.

From equation (13)

$$(\underline{I} + \underline{i}) = \underline{Y}\ (\underline{E} + \underline{e})$$

Then $\underline{C}_1(\underline{I}' + \underline{i}') = \underline{Y}\,\underline{A}_1\ (\underline{E}' + \underline{e}')$

$$\therefore \quad \underline{I}' + \underline{i}' = \underline{\tilde{A}}_1\ \underline{Y}\ \underline{A}_1\ (\underline{E}' + \underline{e}')$$

Now each of the above vectors has the dimensions branch x 1, but one can consider each as containing a nodal contribution and a mesh contribution since the number of non-datum nodes plus the number of basic meshes is equal to the number of branches in the system.

For example the vector $\underline{i}'$ has only values for the basic mesh currents therefore the nodal contribution is zero.

i.e. $\underline{i}' = \begin{bmatrix} \underline{o} \\ \underline{i}_2 \end{bmatrix}$ where $\underline{o}$ has dimensions n-1x1

$\underline{i}_2$ has dimensions m x 1

Therefore partitioning along this node-mesh axis we obtain

$$\begin{bmatrix} \underline{I}_1' & + & \underline{o} \\ \underline{I}_2' & & \underline{i}_2 \end{bmatrix} = \begin{bmatrix} \underline{Y}_1' & \underline{Y}_2' \\ \underline{Y}_3' & \underline{Y}_4' \end{bmatrix} \begin{bmatrix} \underline{E}_1 & + & \underline{e}_1 \\ \underline{E}_2 & & \underline{o} \end{bmatrix}$$

$$\therefore \quad \underline{I}_1' = \underline{Y}_1'\ (\underline{E}_1' + \underline{e}_1') + \underline{Y}_2'\ \underline{E}_2'$$

$$\underline{I}_2' + \underline{i}_2' = \underline{Y}_3'\ (\underline{E}_1' + \underline{e}_1') + \underline{Y}_4'\ \underline{E}_2'$$

solving for node-to-datum potentials

$$\underline{e}_1' + \underline{E}_1' = \underline{Y}_1'^{-1}(\underline{I}_1' - \underline{Y}_2'\underline{E}_2') \qquad (16)$$

$$\therefore \quad \underline{I}_2' + \underline{i}_2' = (\underline{Y}_4' - \underline{Y}_3'\ \underline{Y}_1'^{-1}\underline{Y}_2')\ \underline{E}_2' + \underline{Y}_3'\ \underline{Y}_1'^{-1}\ \underline{I}_1'$$

In terms of impedances by a similar development we get

$$\underline{e}_1' + \underline{E}_1' = (\underline{Z}_1' - \underline{Z}_2'\ \underline{Z}_4'^{-1}\underline{Z}_3')\underline{I}_1' + \underline{Z}_2'\ \underline{Z}_4'^{-1}\ \underline{E}_2' \qquad (17)$$

and $\underline{i}_2' + \underline{I}_2' = \underline{Z}_4'^{-1}(\underline{E}_2 - \underline{Z}_3' \underline{I}_1')$

Note that in the special case of constraining the nodal flows to a tree $\underline{I}_2' = 0$ This and other formulations of $\underline{I}'$ will be discussed further below.

The interesting feature of this development is that both sets of equations yield an expression for the node-to-datum potentials and the mesh currents. This is because the transformation matrices are non-singular and exemplifies what is meant by"looking at the network from a nodal and mesh point of view simultaneously."

Equations (16) and (17) are equivalent and it follows for example that $\underline{Y}_1'^{-1} \equiv (\underline{Z}_1' - \underline{Z}_2' \underline{Z}_4'^{-1} \underline{Z}_3')$

Now $\underline{Y}_1' = \tilde{\underline{A}} \, \underline{Y} \, \underline{A}$ so that from equation (14) we could write

$$\underline{e}_1' = (\underline{Z}_1' - \underline{Z}_2' \underline{Z}_4'^{-1} \underline{Z}_3')(\underline{I}_1' - \tilde{\underline{A}}_{13} \, \underline{Y} \, \underline{E}) \qquad (18)$$

We have therefore from a knowledge of $\underline{I}_1'$, $\underline{Y}$ and $\underline{E}$ two routes for calculating $\underline{e}_1'$, which do not involve the same amount of computation. For the example shown in Appendix A the route involving equation (17) requiring less calculation for it is necessary only to invert $\underline{Z}_4'$ which is a 2 x 2 matrix. For large networks however this is still not a great advance

iii) Extension of the Transformations

Another property of the non-singular transformation is that it is possible not only to transform the primitive system to a given connected system but by exactly the same procedure it is possible to construct a transformation matrix between any two systems containing the same number of branches. Mathematically, given two networks A and B containing the same number of branches we can write as above

$$\underline{i}_p = \underline{C}_{pA}\,\underline{J}'_A$$

and $$\underline{i}_p = \underline{C}_{pB}\,\underline{J}'_B$$

$$\therefore \quad \underline{J}'_A = \underline{C}_{pA}^{-1}\,\underline{C}_{pB}\,\underline{J}'_B = \underline{C}_{AB}\,\underline{J}'_B$$

$\underline{C}_{AB}$ will therefore transform any vector or matrix associated with network A to that of B

In particular if network A contains the same number of nodes as network B the transformation matrix C_{AB} will have the general form

$$\underline{C}_{AB} = \begin{bmatrix} \underline{U} & \underline{S}' \\ \underline{0} & \underline{U} \end{bmatrix}$$

These relationships, which are discussed in more detail in appendix D, lead directly to the technique of Diakoptics.

F. Diakoptics

The object of Diakoptics can now be restated as transforming the network into an intermediate network, whose solution can be found, then transforming (reconnecting) this solution into the solution of the given network. This process having the computational advantages of speed and small storage requirements.

It will now be shown that if, by a process of tearing, the intermediate network contains the same number of nodes, not only can the transformation matrix $\underline{C}_{AB}$ between the two networks be constructed by inspection, but also the actual mathematics of transformation or reconnection are inherently simpler.

Fig. (10) shows the previous example with the proposed cuts. These are chosen such that the subnetworks shown in Fig. (11) contain at least one ground point, with the exception of the cut branch subnetwork (4) which is in its primitive state.

Now as before, equating the branch currents in the two networks figs 10 and 11 we can write

$$i_{p1} = I'_{A1} = I'_{B1} - i'_{B1} - i'_{B2}$$

$$i_{p2} = I'_{A2} = I'_{B2} - i'_{B2}$$

$$i_{p3} = I'_{A3} = I'_{B3} + i'_{B1}$$

$$i_{p4} = i'_{A1} = i'_{B1}$$

$$i_{p5} = i'_{A2} = i'_{B2}$$

or in matrix form

$$\begin{bmatrix} I'_{A1} \\ I'_{A2} \\ I'_{A3} \\ i'_{A1} \\ i'_{A2} \end{bmatrix} = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} I'_{B1} \\ I'_{B2} \\ I'_{B3} \\ i'_{B1} \\ i'_{B2} \end{bmatrix}$$

$$\therefore \begin{bmatrix} \underline{I}'_A \\ \underline{i}'_A \end{bmatrix} = \begin{bmatrix} \underline{U} & \underline{S}' \\ \underline{0} & \underline{U} \end{bmatrix} \begin{bmatrix} \underline{I}'_B \\ \underline{i}'_B \end{bmatrix}$$

i.e. $\underline{J}'_A = \underline{C}_{AB} \underline{J}'_B$.

From equation (14) a solution for the node-to-datum potentials of

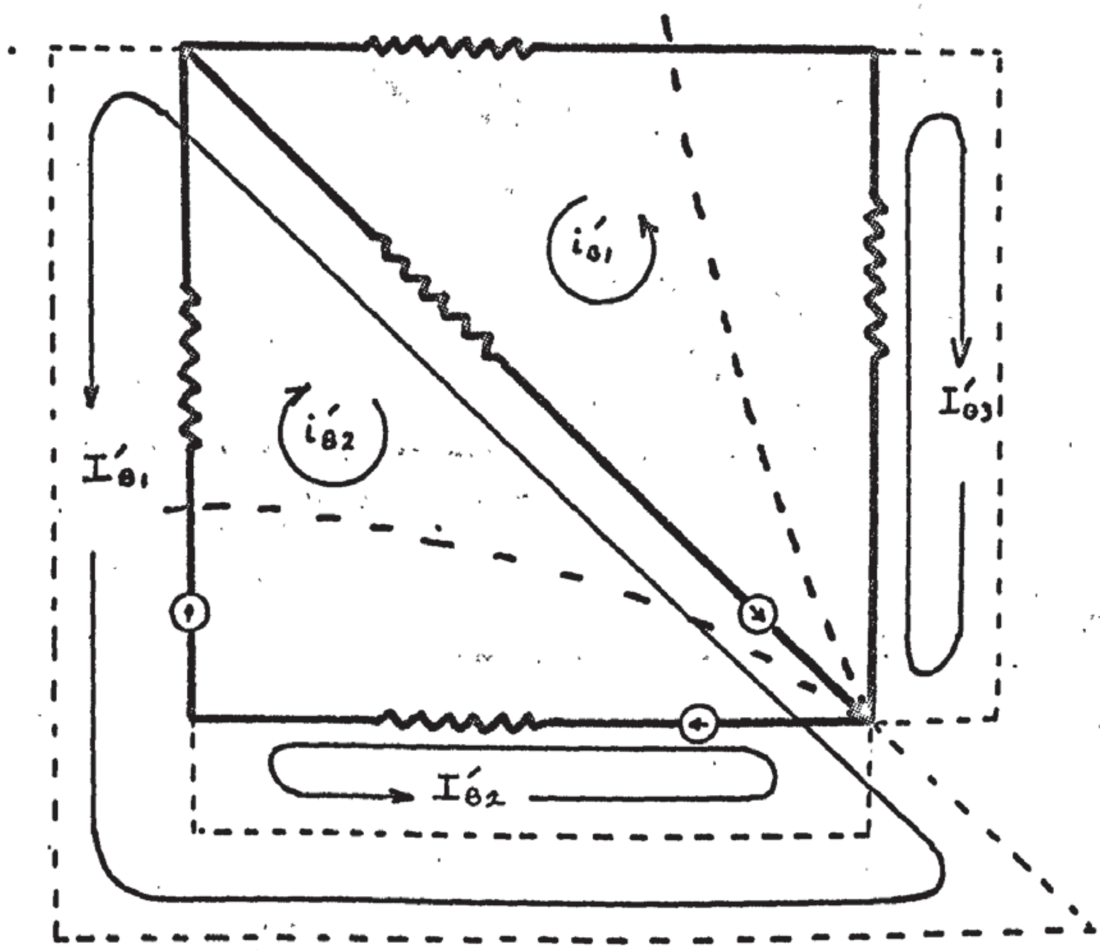


FIG 10

ORTHOGONAL NETWORK

SHOWING PROPOSED CUTS

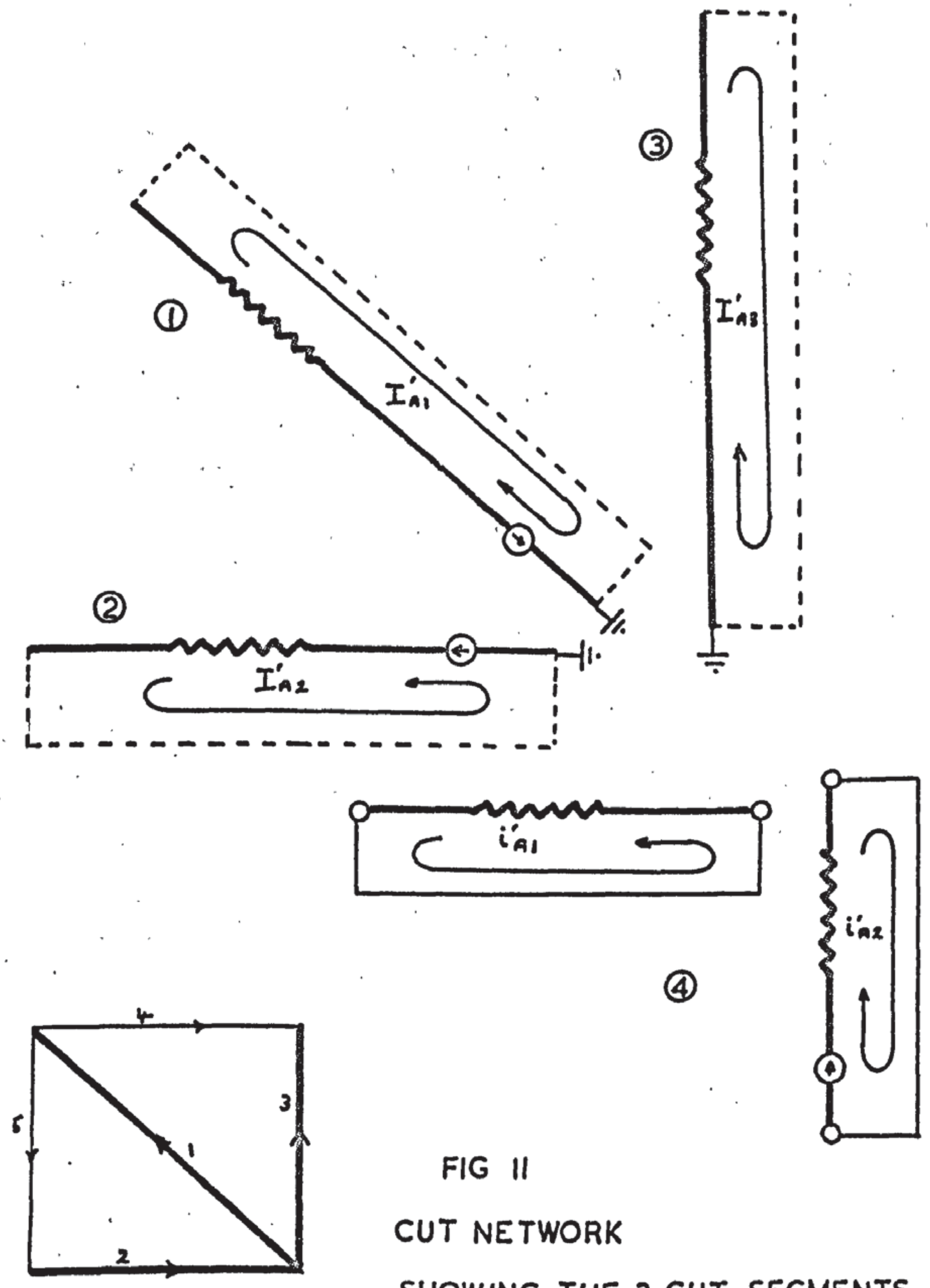


FIG II

CUT NETWORK

SHOWING THE 3 CUT SEGMENTS

AND THE CUT BRANCH SEGMENT 4

PRODUCED BY THE PROPOSED CUTS

segments 1 to 3 of fig. 11 can be obtained.

Let $\underline{I}'_{1Ai} = \underline{I}'_{1i} - \underline{A}_{JAi}\ \underline{Y}_i\underline{E}_i \quad i = 1 \ldots 3$ ....(19)

and if the corresponding vector for fig. 11. $\underline{I}'_{1Bi}$ contains only the same additional nodal demands due to the pump terms as $\underline{I}'_{1Ai}$ i.e. no contributions from pump terms in the cut branches, then the above identities are still true (e.g. $I'_{1A1} = I'_{1B1} - i'_{b1} - -i'_{b2}$)

$\therefore \quad \underline{J}''_{1A} = \underline{C}_{AB}\ \underline{J}'_{1B}$

for a solution to fig. 11. we can write from equation (14)

$$\underline{e}'_{Ai} = (\hat{\underline{A}}_{jAi}\ \underline{Y}_{Ai}\ \underline{A}_{jAi})^{-1}\ \underline{I}'_{Ai} \qquad i = 1 \ldots\ldots 3$$

and for the cut branch system

$$\underline{E}'_{A4} = \underline{Z}_4\ \underline{i}'_{A4}$$

i.e.

$$\begin{bmatrix} \underline{e}'_{A1} \\ \underline{e}'_{A2} \\ \underline{e}'_{A3} \\ \underline{E}'_{A4} \end{bmatrix} = \begin{bmatrix} \underline{Z}'_{A1} & & & \\ & \underline{Z}'_{A2} & & \\ & & \underline{Z}'_{A3} & \\ & & & \underline{Z}'_{A4} \end{bmatrix} \begin{bmatrix} \underline{I}'_{1A1} \\ \underline{I}'_{1A2} \\ \underline{I}'_{1A3} \\ \underline{i}'_{A4} \end{bmatrix} \begin{matrix} \uparrow \\ \alpha \\ \downarrow \\ \beta \end{matrix}$$

or

$$\begin{bmatrix} \underline{e}'_A \\ \underline{E}'_A \end{bmatrix} = \begin{bmatrix} \underline{Z}_\alpha & \\ & \underline{Z}_\beta \end{bmatrix} \begin{bmatrix} \underline{I}'_{1A} \\ \underline{i}'_A \end{bmatrix}$$

Transforming this solution i.e.

$$\underline{J}'_{1A} = \underline{C}_{AB}\ \underline{J}'_{1B}$$

Therefore we can write

$$\underline{V}'_B = \tilde{\underline{C}}_{AB}\ \underline{Z}'_A\ \underline{C}_{AB}\ \underline{J}'_B$$

i.e. $\underline{Z}_B \equiv \underline{\tilde{C}}_{AB}\ \underline{Z}'_A\ \underline{C}_{AB}$

$$\therefore\ \underline{Z}_B = \begin{bmatrix} \underline{\hat{U}}\, Z_\alpha\, \underline{U} & \underline{\hat{U}}\, Z_\alpha\, \underline{S}' \\ \underline{\hat{S}}'\, Z_\alpha & \underline{\tilde{S}}'\, Z_\alpha\, \underline{S}' + Z_\beta \end{bmatrix}$$

or omitting the multiplications by unit matrices.

$$\begin{bmatrix} \underline{e}'_B \\ \underline{E}'_B \end{bmatrix} = \begin{bmatrix} Z_\alpha & Z_\alpha \underline{S}' \\ \underline{\tilde{S}}'\, Z_\alpha & \underline{\tilde{S}}'\, Z_\alpha\, \underline{S}' + \underline{Z}_\beta \end{bmatrix} \begin{bmatrix} \underline{I}'_{1B} \\ \underline{i}'_B \end{bmatrix}$$

Therefore the solution for the node-to-datum notentials $\underline{e}'_B$ of the given network is

$$\underline{e}'_B = (Z_\alpha - Z_\alpha\, \underline{S}' \left[\underline{\hat{S}}'\, Z_\alpha\, \underline{S}' + \underline{Z}_\beta\right]^{-1} \underline{\hat{S}}'\, Z_\alpha)\ \underline{I}'_{1B}$$

$$+ Z_\alpha\, \underline{S}' \left[\underline{\hat{S}}'\, Z_\alpha\, \underline{S}' + \underline{Z}_B\right]^{-1} \underline{E}'_B \qquad (20)$$

Where, because of the equation (19), $\underline{I}'_B$ is the nodal input-output vector minus the assumed nodal currents produced by the potential sources in the subnetworks, $\underline{E}'_B$ is the potential source vector of the cut branches as these are in their primitive state.

Note that the only part of $\underline{C}_{AB}$ needed is $\underline{S}'$ which is easily formed as it contains as many columns as there are cut branches and shows between which two nodes any two cut branches run. Also as equation (20) is in its factorised form only a simple series of matrix-vector multiplications is needed to arrive at the vector $\underline{e}'_B$ as shown below

Let $Y_\beta = (\underline{\tilde{S}}'\, Z_\alpha\, \underline{S}' + \underline{Z}_\beta)^{-1}$

$$\underline{e}'_A = Z_\alpha\, \underline{I}'_B$$

$$\underline{E}_1 = -\underline{\tilde{S}}'\, \underline{e}'_A$$

$$\underline{E}_2 = \underline{E}_1 + \underline{E}'_B$$

$$\underline{i}_1 = \underline{Y}_B \underline{E}_2$$

$$\underline{I}'_1 = \underline{S} \underline{i}_1$$

$$\underline{e}'_2 = \underline{Z}_\alpha \underline{I}'_1$$

then $$\underline{e}'_B = \underline{e}'_A + \underline{e}'_2$$

The example previously considered is solved using the above method in Appendix B.

G. Advantages of the Diakoptics Approach

From the mathematical development it is perhaps difficult to see the wider implications or the radically different approach to problems that underlies the method. These will be discussed later in more general terms. The specific advantages of the proposed route to solution for the complex pipe network problem must however now be outlined as they form an integral part of the computer programme developed. A description of these programmes follows this section.

It was stated in the introduction to this thesis that, for any design aid, small changes in the shape or input parameters of the system must be capable of rapid calculation. The parameters here will be classified as:-

1) The vector of nodal demands.

2) The branch impressed pressure vector that is the value of the pressure rise of pumps or the calculated pressure drop through a piece of plant in the line (e.g. strainers, valves, heat exchangers etc.,)

The diameter, length and position of any pipe constitute the shape of the network.

Now after a solution is obtained for a given network, the inverted admittance matrices of the subnetworks will be contained in the appropriate backing store of the computer. The solution for a change in the input parameters can then be arrived at by just the matrix-vector multiplication outlined above. If a major change is contemplated then the process must be allowed to iterate to an accurate solution but of course the number of iteration cycles is reduced.

The effect of the addition or removal of pipes can be handled with similar ease and speed. Consider for example the addition of a new pipe to Fig. (10) say running between nodes 2 and 3. This pipe can be considered to be cut and would appear as another isolated segment in part (4) of Fig. (11). The new solution would then be just the reconnection process with a new $\underline{S}'$ matrix. See appendix C for a worked example.

Branch removal can be considered in exactly the same way e.g. removal of branch (1) is the same as the addition of a new branch running between nodes 1 and 3 with an impedance of minus the value of the calculated impedance of branch (1)

Note however if a branch to be removed forms part of the cut branch set then its removal is accomplished merely by leaving it out.

This illustrates the general point, that it is easier to change factors in the cut branch set than those associated with the subnetworks. Therefore as a general point of policy it is more efficient to put those parts of the network whose design is uncertain in the cut segment.

It can also be seen that the sub-systems can be interconnected in any arbitary way by changing $\underline{S}'$. This means that the effect of connecting isolated distribution systems together or an optimum policy for reconnecting an existing system can be found. This process can be carried one stage further by the interconnection of existing systems into super systems without any increase in the direct access storage required.

H. Summary of the Calculation Steps

(i) The steps in a full calculation of a new network can be summarised as follows.

1) Form $\underline{\tilde{A}}_i \, \underline{Y}_i \, \underline{A}_i$ and invert forming $\underline{Z}_i$ for i = 1,2 ...... ω

2) Form $\underline{I}'_{Bi} = (\underline{I}'_i - \underline{\tilde{A}}_i \, \underline{Y}_i \, \underline{E}_i)$ for i = 1,2 .... ω

3) Form $\underline{e}'_{Ai} = \underline{Z}_i \, \underline{I}'_{Bi}$ for i = 1,2 .... ω

4) Form $\underline{E}_1 = -\underline{\tilde{S}}' \, \underline{e}'_A$

5) Form $\underline{Y}_\beta = \left(\sum_{i=1}^{\omega} \underline{\tilde{S}}'_i \, \underline{Z}_i \, \underline{S}'_i + \underline{Z}_\beta \right)^{-1}$

6) Form $\underline{E}_2 = \underline{E}_1 + \underline{E}'_B$

7) Form $\underline{i}_1 = \underline{Y}_\beta \, \underline{E}_2$

8) Form $\underline{I}'_1 = \underline{S}' \, \underline{i}_1$

9) Form $\underline{e}'_{2i} = \underline{Z}_i \, \underline{I}'_{1i}$

10) Form $\underline{e}_B = \underline{e}'_A + \underline{e}'_2$

The number of multiplications involved in the calculation of each step will be shown. It is assumed that the approximate number of multiplications needed to invert a symmetric matrix is $n^3/2$ where n is the dimensions of the matrix. Let there be ω segments, each segment containing n nodes and let there be P cut branches. The approximate number of multiplications for a solution to a 200 node network cut into 8 segments of 25 nodes and with 20 cut branches is also shown. The time taken for the addition and subtraction is not taken into account as this will be negligible compared with the multiplication time.

1) $\omega \times n^3/2$	125,000
2) n	25
3) $\omega \times n^2$	4,200
4) 0	0
5) $P^3/2$	4,000
6) 0	0
7) p^2	400
8) 0	0
9) $\omega \times n^2$	4,200
10) 0	0
Total number of operations	141,825

i.e. Inversion 125,000

Connection Process 16,825

Total number of operations for inversion of full matrix 4,000,000.

(ii) The steps for serious modifications of a network are as follows:-

Having obtained a solution to the full problem

(1) To change the nodal demands or branch pressure rises. Start from step (2) and execute 2,3,4,6 to 10.

(2) To add or remove branches start from step (4) and execute 4 to 10

(3) To change cut branch pump terms start from step 6 execute 6 to 10.

(4) To interconnect segments in different manner start from step 4 and execute 4 to 10.

The development so far has assumed a linear relationship between current and voltage i.e, $\underline{J} = \underline{Y}\,\underline{V}$. Now the pressure drop-flow relationships for fluid networks are non-linear and so an iteration scheme based on diakoptics has to be used.

I The Iteration Scheme for Fluid Networks

For a single pipe the pressure drop Δp can be found from

$$\Delta p = 4\phi\, L/D\, \rho\, u^2/g_c$$

i.e. $$\Delta p = 4\phi\, L/D\, \rho\, Q^2 / A^2\, g_c$$

or $$Q = \left(\frac{D\, A^2\, g_c}{4\phi\, L\, \rho\, Q}\right) x \Delta p$$

i.e. $$J = Y\, V$$

It also follows that

$$\phi^{\frac{1}{2}}\, Re = \sqrt{\frac{\Delta p\, D^3\, \rho}{4\, L\, \mu^2}}$$

and, from the work of Colebrook and White it is known that the friction factor relationshop for turbulent flow in smooth and rough pipes is given by

$$\phi^{\frac{1}{2}} = -2.5\, \ln\left(\frac{\varepsilon/d}{3.7} + \frac{1}{1.13\, Re\, \phi^{\frac{1}{2}}}\right)$$

At the start of the computation a guess is made of the individual branch flows and friction factors. The branch admittance Y and the admittance matrix for each segment are found and the node to datum pressure vector $\underline{e}'$ calculated.

Then knowing the branch pressure drops the values of ϕ and Q are recalculated. The whole process then being repeated to convergence. The convergence criterion being,

$$\sqrt{\left(\sum_{i=1}^{n} \Delta' e_i'\right)^2} \leq \text{limit}$$

where $\Delta' e_i' = e_i^{(o)} - e_i^{(1)}$

Note that no precalculation is necessary as the initial guesses do not have to obey Kirchoffs Laws.

Chapter 4.

Description of the Computer Program

A. Description of the National Elliott 803 Computer

The machine used in this study (a National Elliott Series 803) is a second generation computer with an 8K core store. Each word is capable of holding two machine code instructions or one integer or one floating point number. The instruction code has hardware floating point. There are two tape readers, two punches, one on-line teleprinter (output only) and a lineprinter. The backing store consists of three film handlers each film holding 4K blocks of 64 words per block. The rate of data transfer between these films and the core store is very slow, being a maximum of 5 blocks per second and the efficiency of the program may be impaired if the transfers are not well organised.

B. General Description of the Computer Program

i) Introduction

The Algol language in which the program is written does not specify any input/output format so that all blocks containing such statements particularly Procedure Resultsprint must be regarded as specific to the 803 machine. Three other procedures not mentioned below also come into this catagory. These deal with the film transfers i.e. Procedures Filmwrite. Filmread and Locate. Although their function is self explanatory they are also specific to the 803. In fact although data transfers form an integral part of the method the configuration of the backing store varies so much from machine to machine that no discussion in general terms can be attempted.

The program has been written as a series of self contained Procedures. This method has the advantage that the different parts can be written and tested separately, the logical paths for the different options open to the user are easier to organise and special procedures for the calculation of specific items can be included without changing the basic configuration. There is also a set of basic matrix procedures to execute the relevant matrix manipulations.

ii) Matrix Procedure List and Functions

ZERO (A);

Sets elements of Array A to zero

*MXSUM (A,B,C)

Sets A equal to the sum of matrices B and C

*MXPROD (A,B,C)

Sets A to the matrix product of B times C.

CHOLESKI (A)

Inverts matrix A by Choleski method putting result in A.

*READMX (A)

Reads a set of data and writes this in stores assigned to A.

*PRINTMX (A)

Prints values of Matrix A (used as a check routine.)

*Programs from the 803 computer library.

iii) Basic Operation of Blocks in Overall Flow Sheet

In the general flow sheet Fig. (12) and from the description which follows, it will be seen that only blocks 8-20 are concerned with Diakoptics steps summarised on page (36). The rest are basic housekeeping operations which organise the calculation procedures into the required order so as to solve, for example, a new problem or one with a change in any of the network parameters, or in the shape of an existing problem for which the solution has already been obtained. Certain others are considered to be self-explanatory and no further description will be given.

BLOCK (1) INPUT Number of individual segments

The three starting block addresses for film handlers

Total number of nodes in System (excluding reference node.)

Number of cut branches

π

Fluid density

Viscosity

Convergence Criterion

(2) INPUT Number of branches minus one and number of nodes (excluding reference node) for each cut segment in order.

(3) INPUT This block sets the Flags which are boolean identifiers to control the calling sequence of the procedurees. In the setting procedure a number is input and compared with a data value and the flags are set to true if a comparison is obtained i.e. in Algol,

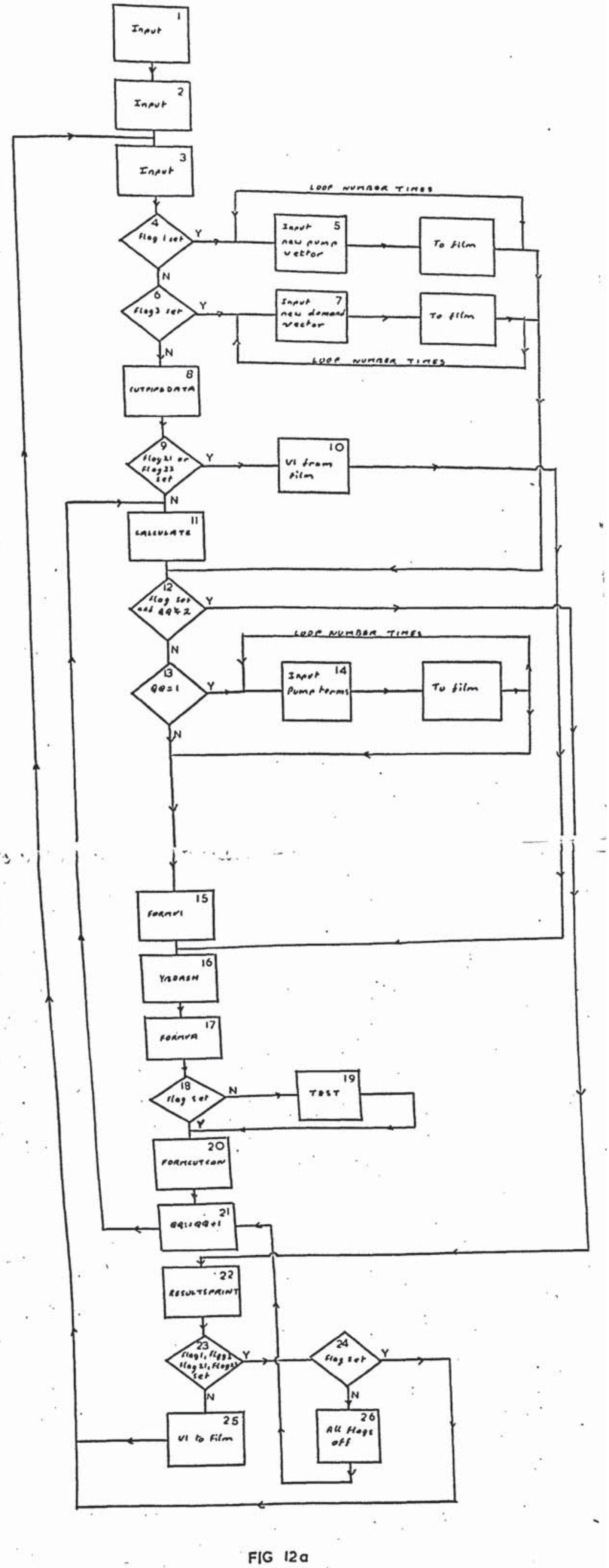


FIG 12a

GENERAL FLOW SHEET OF DIAKOPTICS PROGRAM

READ N′

FLAG: = N = 0′

Integer Input

N ≠ 0 FLAG not set: program will accept completely new data. QQ: = 1 for procedure CALCULATE to read data for each segment:

0 FLAG set: Program will behave as if converged. Will now go on to set the four flags for changes in shape or parameters of the network QQ: = 2

1 FLAG 1 set: Pump terms for individual pipes in segments will be changed

21 FLAG 21 set: enables branches to be added to system, branches to be removed from cut branch segment, pump terms to be changed in cut branches, length or diameter of pipes in cut segment to be changed.

22 FLAG 22 set: enables branches to be removed from segments.

3 FLAG 3 set: Nodal demand vector to be changed

If FLAG 21 or FLAG 22 set: reads new dimension for the cut branch set

8) CUTPIPEDATA

If FLAG 22 set this procedure reads branch and segment numbers for pipes to be removed.

Reads assumed values of resistance, diameter, lengths, roughness and pump terms for the cut branches.

Reads connection list 1 ≡ $\underline{S}'$

10) If FLAG 21 or FLAG 22 set then vector $\underline{e}'_A$ called V1 in program is read off film so that the calculation can begin at step 4. (See chapter 3 Section H.)

11) Calculate

This procedure is a set of procedures, for a new problem when the program is entered for the first time. It reads:

The diameters, lengths, assumed branch flows and roughnesses of the pipes.

The connection list two i.e. GRAP ≡ A_{1j} for the first segment, calculates the branch admittances, forms the admittance matrix, inverts it and repeats this procedure for all the segments.

After the first time round it calculates the friction factors, flow and branch admittances from the individual calculated branch pressure drops, before forming and inverting the admittance matrix.

12) If FLAG is set and QQ = 2 then there is a modification to the network and the new nodal pressure vector and branch flows must be calculated. If QQ ≠ 2 and FLAG set then system has converged and control passes to the procedure RESULTSPRINT.

13) If QQ = 1 then it is a new problem on its first iteration cycle, therefore the segment pump terms must be input to complete the data.

15) FORM V1

This procedure executes steps 2 and 3 outlined above.

16) YB DASH

This procedure executes steps 4 and 5 outlined above.

17) FORM VA

This procedure executes steps 6,7,8,9 and 10 outlined above.

19) TEST

If the square root of the sum of the squares of the pressure differences is less than the value specified FLAG is set.

20) FORMCUTCON

From a knowledge of the final pressure vector, calculates the pressure drop across the cut branches and hence in a similar manner to PIPECONSTANTS calculates the flow and friction factors for the cut branches

24) After a change has been made to the original network and the results of the first iteration have been printed, FLAG is set in a similar manner to block 3. If it is set true then control passes back to block 3 for a new change to be input. If not set then all flags are turned off and program iterates till the accurate solution to the new problem is found.

C. Discussion of Procedures in Detail

The Matrix Procedure Choleski

This inversion routine was chosen for its speed. It is applicable only to symmetric matrices but is at least twice as fast as the standard Gaussion elimination methods. The calculation proceeds in two passes. The first pass operates on the elements of the upper triangle, including the main diagonal, one row at a time.

The diagonal term is evaluated first followed by the elements in its row.

Diagonal evaluation

$$b_{ii} = \sqrt{a_{ii}} \qquad i = 1$$

$$b_{ii} = \sqrt{\left(a_{ii} - \sum_{k=1}^{i-1} b^2_{ki}\right)} \qquad i = 2,3 \ldots\ldots n$$

Off diagonal element evaluation

$$b_{ij} = a_{ij}\, b_{ii}^{-1} \qquad i = 1$$

$$b_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} b_{ki}\, b_{kj} \right) b_{ii}^{-1} \quad i = 2,3, \ldots n$$

Note that each diagonal element evaluation must be checked to see that the quantity under the square root is always positive i.e. the matrix is non singular.

The second pass forms the final elements of the lower triangle (including the main diagonal) and so because the matrix is symmetric these elements can be reflected. The order of the elements calculated is the mirror image of the first pass, that is starting from the last element $b_{n\,n}$ and working back along the row.

Diagonal evaluation

$$c_{ii} = \left(b_{ii}^{-1} - \sum_{k=i+1}^{n} c_{ki}\, b_{ik} \right) b_{ii}^{-1}$$

Off diagonal element evaluation

$$c_{ij} \equiv c_{ji} = \left(- \sum_{k=n}^{j+1} C_{ik} b_{jk} \right) b_{jj}^{-1}$$

OUTPIPEDATA

Format of connection List S′

Each branch has two nodes and an assumed direction associated with it. In the original matrix these were represented by plus or minus one.

In the program however the matrix is not input as such.

A list is input containing the relevant information as node numbers, each branch having a pair of node numbers and the direction of assumed flow being from the first node to the last node mentioned. This choice is arbitary but if a pipe has a pump in it the direction of flow must be considered in assigning the sign of the pressure rise in the pump i.e. if the pressure rise in the pump has the same direction as the assumed flow the sign of this pressure rise is negative in the pump rise pressure vector, otherwise it is positive.

CALCULATE

The flow sheet for this procedure is shown in Fig. (12b)

List of Procedures in CALCULATE

FORMDELTP

This procedure is used in procedure PIPECONSTANTS to calculate the individual branch pressure drops across the impedance element.

FORMADMIT

This procedure which is used by INVADMIT forms the admittance matrix from the calculated branch admittance i.e. Diagonal terms a_{ii} equal to the sum of the admittances of branches incident at node i and the off diagonal elements a_{ij} equal to minus the admittance of the branch running between nodes i and j.

PIPEDATA

This is essentially the same as CUTPIPEDATA except in the format

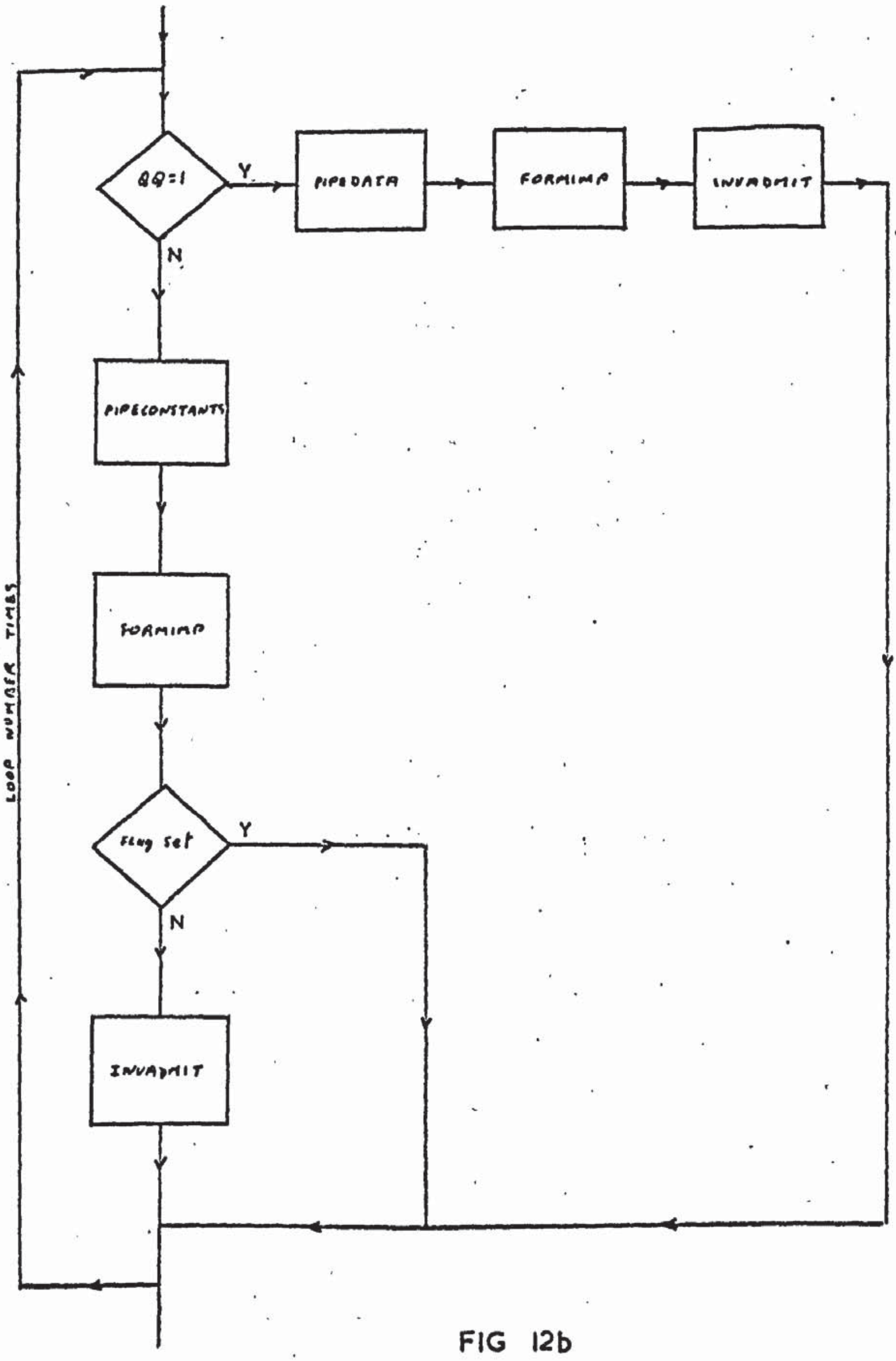


FIG 12b

PROCEDURE CALCULATE

of the correction list grap. As the pump terms for the segments are all taken as pressure increases the branches must be orientated in the opposite direction to the pressure rise.

A check procedure is incorporated after the input of Grap. This is a list of the number of branches incident at the nodes. The connection list is then checked. For example if node i has three branches incident at it then i must appear three times in the connection list.

FORMIMP

This is self explanatory but one further section is included so that if Flag 22 is set the resistance of the branch or branches to be removed is copied into the appropriate position of the Z_β matrix.

PIPECONSTANTS

This procedure from a knowledge of the individual pressure drops calculates a value of Re $\phi^{\frac{1}{2}}$ and checks to see whether the flow is laminar. If this is the case it calculates the friction factor and flow from the appropriate equation and prints the branch and segment number of this pipe. If not it then calculates the Reynolds number and checks to see if the flow is transitional again printing out the branch and segment number. It then calculates the friction factor and flow as outlined above.

INVADMIT

This procedure calls FORMADMIT, inverts this matrix by CHOLESKI and writes the resulting matrix on a film handler.

D. Example of Data Preparation and Results

As a further aid in the understanding of the program a full description of the method of data preparation will now be given. It will demonstrate the approach favoured by the author for the compilation of such data for the test network with the proposed cuts. As the test network has no real datum nodes such as reservoirs, river or cooling tower pools a node is first selected as a datum. The cut branches are then selected so that the cut segments are completely isolated from each other but all cut segments are connected by at least one branch to the chosen datum fig (23). More cut branches can be chosen than are needed for isolation as shown in fig (24) The network is redrawn with the cut branches shown in dotted lines in Fig(27)

The segments are then renumbered and the nodes and branches of each cut segment are allocated sequential reference numbers, fig (27). In addition starting at segment 1 each node is given an absolute reference number. Each node in the system now has two reference numbers, an absolute number and a segment number. The absolute number is used for the final pressure vector $\underline{e}'_B$ and also to form the cut branch connection list.

The segment node numbers are used to form the segment connection lists.

The cut branches can now be drawn and numbered showing the two absolute node numbers to which they are incident.

From Fig (27) the data is drawn up as shown in Fig. (13)

The final printout of results for this problem is presented in fig (15)

The branches of the test network all have the same dimensions i.e. length 100ft. diameter 0.5 ft. roughness 0 There are no pumps in the system.

2 800 0 0 21 3 3.142 62.4 2.42 20 INPUT (1)

16 10
17 11 INPUT (2) Dimensions of cut segments

1 INPUT (3) FLAG not set

CUT PIPE DATA

1 1 1 Assumed resistance
0.5 0.5 0.5 Diameter
100 100 100 Length
0 0 0 Roughness
0 0 0 Pump terms

3 11 4 11 10 14 Cut pipe connection list

SEGMENT 1

Diameters

.5 .5 .5 .5 .5 .5
.5 .5 .5 .5 .5
.5 .5 .5 .5 .5 .5

Lengths

100 100 100 100 100 100
100 100 100 100 100
100 100 100 100 100 100

Assumed flows

50 50 50 50 50 50
50 50 50 50 50
50 50 50 50 50 50

Roughness

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

Connection list for segment 1

1 2 2 3 3 4 2 4 4 5 1 5
1 6 6 5 6 7 7 8 8 5 5 10
8 9 7 9 9 10 10 11 9 11

3 3 2 3 5 3 3 3 4 3 2 Connection list check

120 0 0 0 0 0 210 0 0 0 Nodal demands

FIG 13

P.T.O.

SEGMENT 2

Data	Description
·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5 ·5	Diameters
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100	Lengths
50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50	Assumed Flows
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Pipe roughness
1 3 1 2 3 2 3 4 4 2 2 6 2 5 5 6 6 7 7 8 5 8 5 10 5 9 9 10 8 11 10 11 9 11 11 12	Connection List for segment 2
2 5 3 2 5 3 2 3 3 3 4 1	Connection list check
-120 0 0 0 0 0 -240 0 0 -60 0	Nodal demands
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Pump terms Segment 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	" " " 2

FIG 13

SPECIMEN COMPUTER INPUT DATA

Two examples of the data format for changing the system after it has converged to a solution can be seen in Figs (14)

The first row can be seen to set the flags in order i.e. Flag, Flag 1, ... etc. The other information being new data for the problem. The last digit input allows the program to iterate to a complete solution after printing out the results for the first cycle.

0 0 21 0 0 *Flag and Flag 21 set*

7 *Number of branches in new cut branch set*

1.585 1.7048 1.0389 0.90107 1.93 1.434 0.8	*Resistance*
.5 .5 .5 .5 .5 .5 .5	*Diameter*
100 100 100 100 100 100 100	*Length*
0 0 0 0 0 0 0	*Roughness*
0 0 0 0 0 0 0	*Pump Terms*
3 11 4 11 5 10 14 13 10 14 12 15 21 19	*Connection List*

1 *Iterate to solution*

Input data for example a) section D chapter 5

0 0 0 0 3 *Flag and Flag 3 set*

0 0 0 0 0 0 210 0 0 0 *New demand vector segment 1*

-120 0 0 0 0 0 -240 0 0 -60 0 *" segment 2*

1 *Iterate to solution*

Input data for example a) section E chapter 5

FIG 14

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	88.341274	1	2	.59539765
2	50.880716	2	3	.93009425
3	31.471074	4	3	1.3656721
4	37.556811	2	4	1.1862349
5	83.928039	5	4	.62071512
6	45.557090	1	5	1.0163948
7	13.982532	6	1	2.5862665
8	48.528338	6	5	.96614773
9	62.464712	7	6	.78843061
10	69.481835	7	8	.72342606
11	38.696676	8	5	1.1582428
12	48.761505	5	10	.96243597
13	30.858944	8	9	1.3871642
14	77.957367	7	9	.65902714
15	54.452180	9	10	.88069816
16	1.7393838	11	10	12.415593
17	54.383788	9	11	.88158961

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	5.5235464	1	3	5.2778389
2	46.905377	1	2	.99289155
3	46.325954	3	2	1.0028428
4	40.825392	4	3	1.1097033
5	64.086691	4	2	.77227674
6	84.495379	2	6	.61732783
7	72.719968	2	5	.69723750
8	38.156469	5	6	1.1713237
9	122.52645	6	7	.45598174
10	117.15084	8	7	.47302042
11	50.318345	5	8	.93843994
12	1.7960240	5	10	12.131174
13	17.512838	9	5	2.1692074
14	17.692175	9	10	2.1519579
15	66.958890	11	8	.74539401
16	40.559362	11	10	1.1155215
17	35.171692	11	9	1.2499858
18	142.44441	12	11	.40305242

FIG 15

P.T.O.

CUT PIPE RESULTS

PIPE NO	FLOW	FROM NODE	TO NODE	RESISTANCE
1	82.250681	3	11	1.5848506
2	89.979339	4	11	1.7048295
3	104.77526	10	14	1.9298801

NODE NO	PRES
1	95.346816
2	-53.026751
3	-107.73165
4	-84.687269
5	50.524578
6	100.75327
7	179.97992
8	83.934391
9	61.688326
10	-.14009672
11	-238.08670
12	-285.32789
13	-239.13325
14	-202.34379
15	-389.62516
16	-422.20067
17	-690.90979
18	-443.24430
19	-381.55178
20	-389.77321
21	-353.41411

END OF FILE BLOCKNUMBERS
HANDLER 1 804HANDLER 2 6HANDLER 3
6

FIG 15

COMPUTER PRINTOUT OF SOLUTION
TO CASE I

Chapter 5.

Results

A. Hardy Cross Method.

Following Daniel (13) a program was written the listing of which is shown in Appendix G. A test network was devised, shown in Fig. (16) It contains 22 nodes and 38 branches, therefore one needs to form 17 basic loops.

Three different loop formations shown in Fig. (17,18,19) were tried, Fig. (17) shows a case of minimum overlap, Fig. (18) is an arbitary case and Fig. (19) shows a trunk (maximum overlap). It is to be remembered that the loops are defined by the non-tree branches and the appropriate defining trees are shown in double lines. The first seventeen branches are therefore the links and the rest are numbered in any arbitary manner. It can be seen that the trunk is the easiest to form.

Table 2* shows the time taken to converge to a solution for the above cases, and the number of iterations for convergence in the inner cycle.

The time dependence of the convergence on the choice of basic meshes is well demonstrated. These are of course extreme cases but the choice of basic meshes has to be made by the user and it does require a certain amount of trial and error to pick a defining tree. The data preparation is also tedious and time consuming as it has to contain the assumed direction of flow in each branch.

The actual results are presented for comparison with each other and the results from the diakoptics program in table 3.

* This and all subsequently referred to tables will be found in Appendix E.

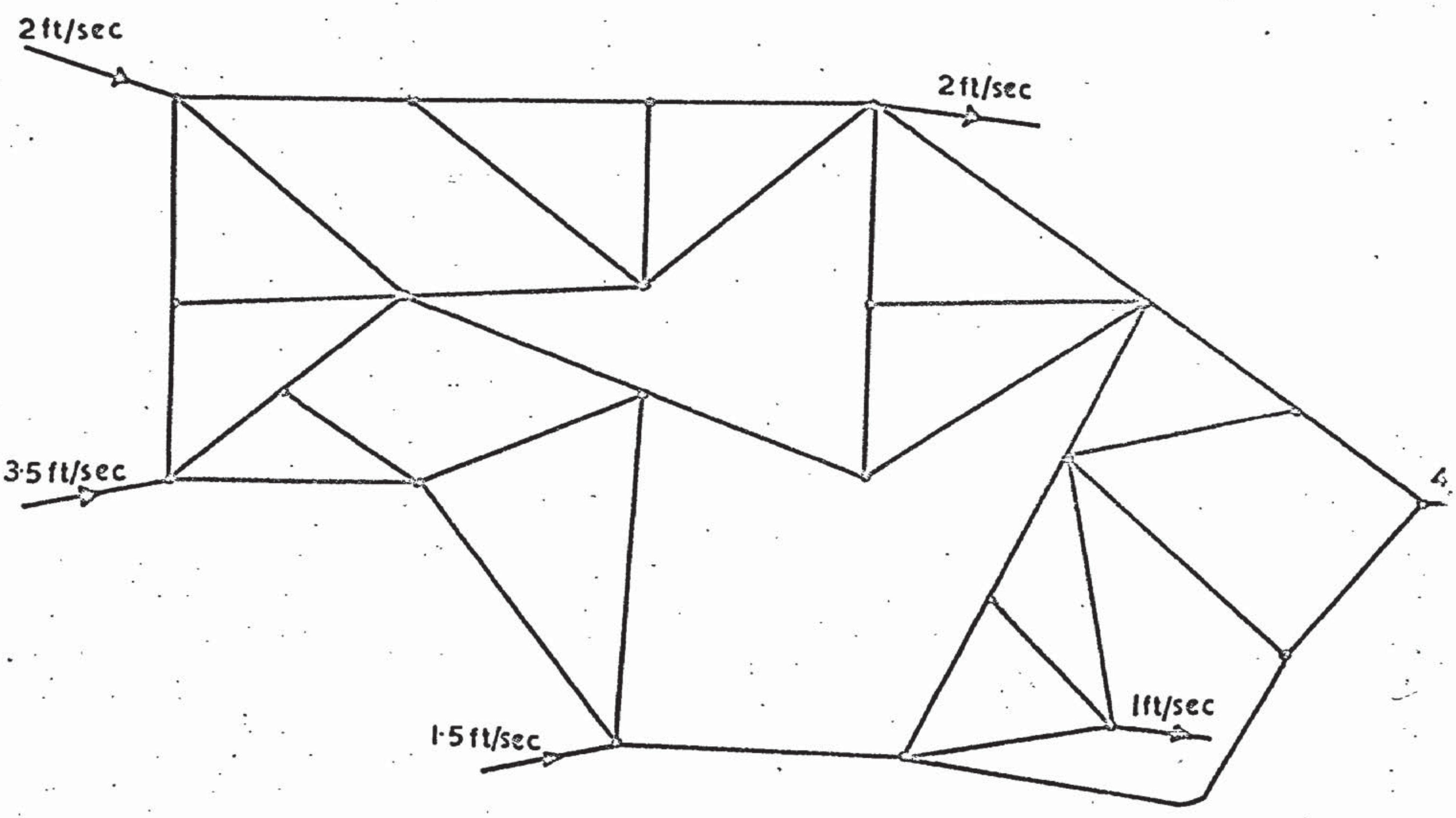


FIG 16
TEST NETWORK

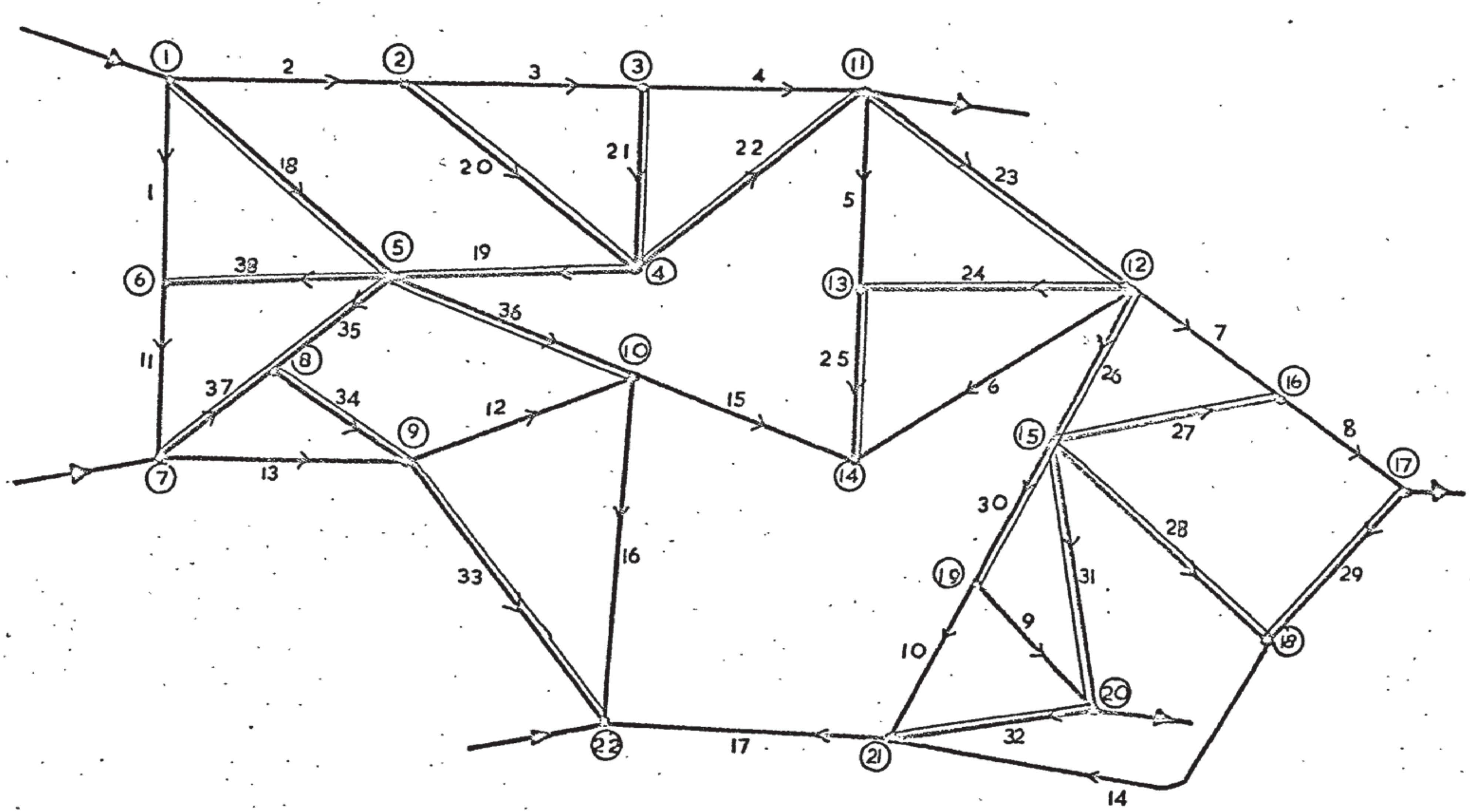


FIG 17

HARDY CROSS CASE I.
MINIMUM OVERLAP

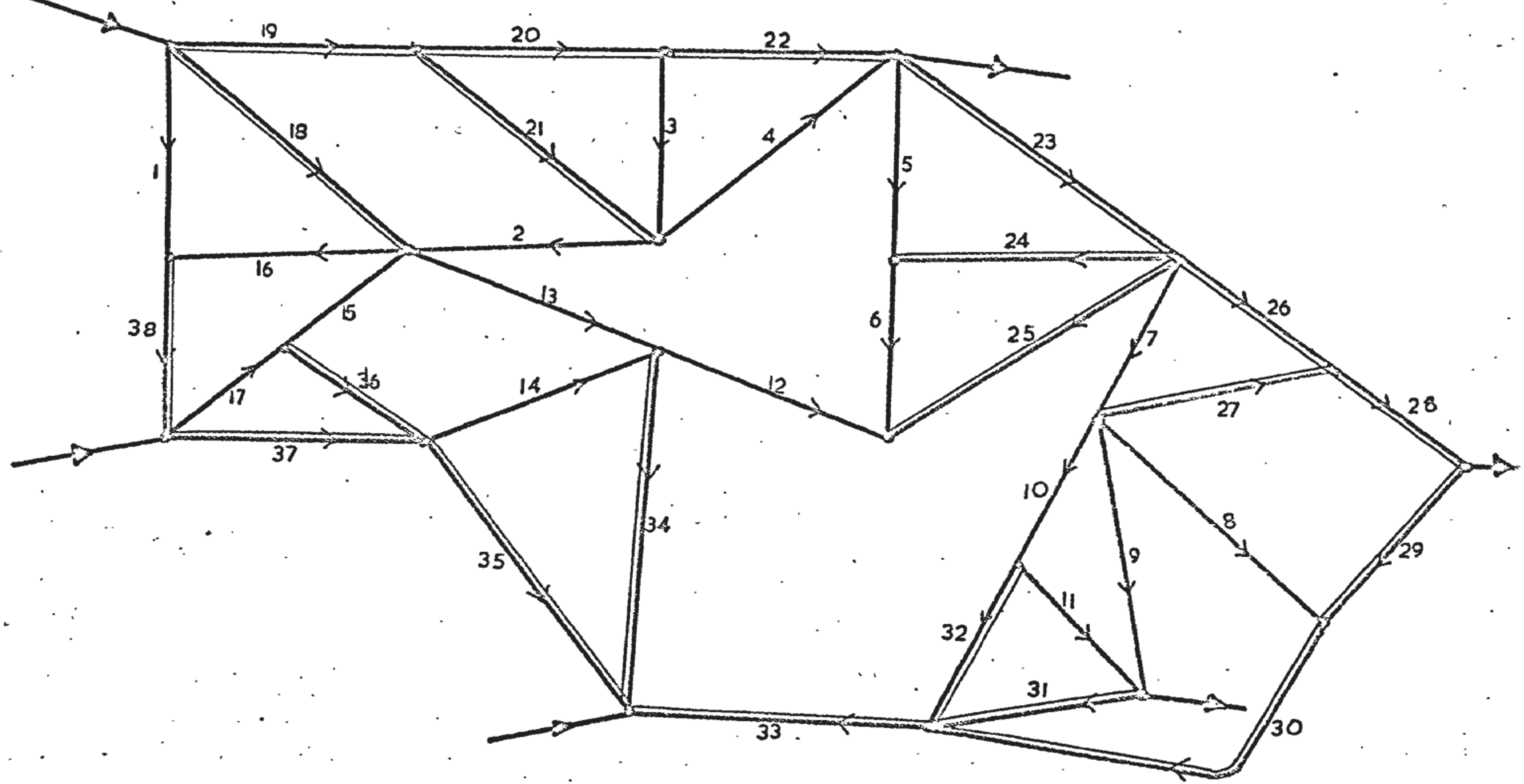


FIG 18
HARDY CROSS CASE 2

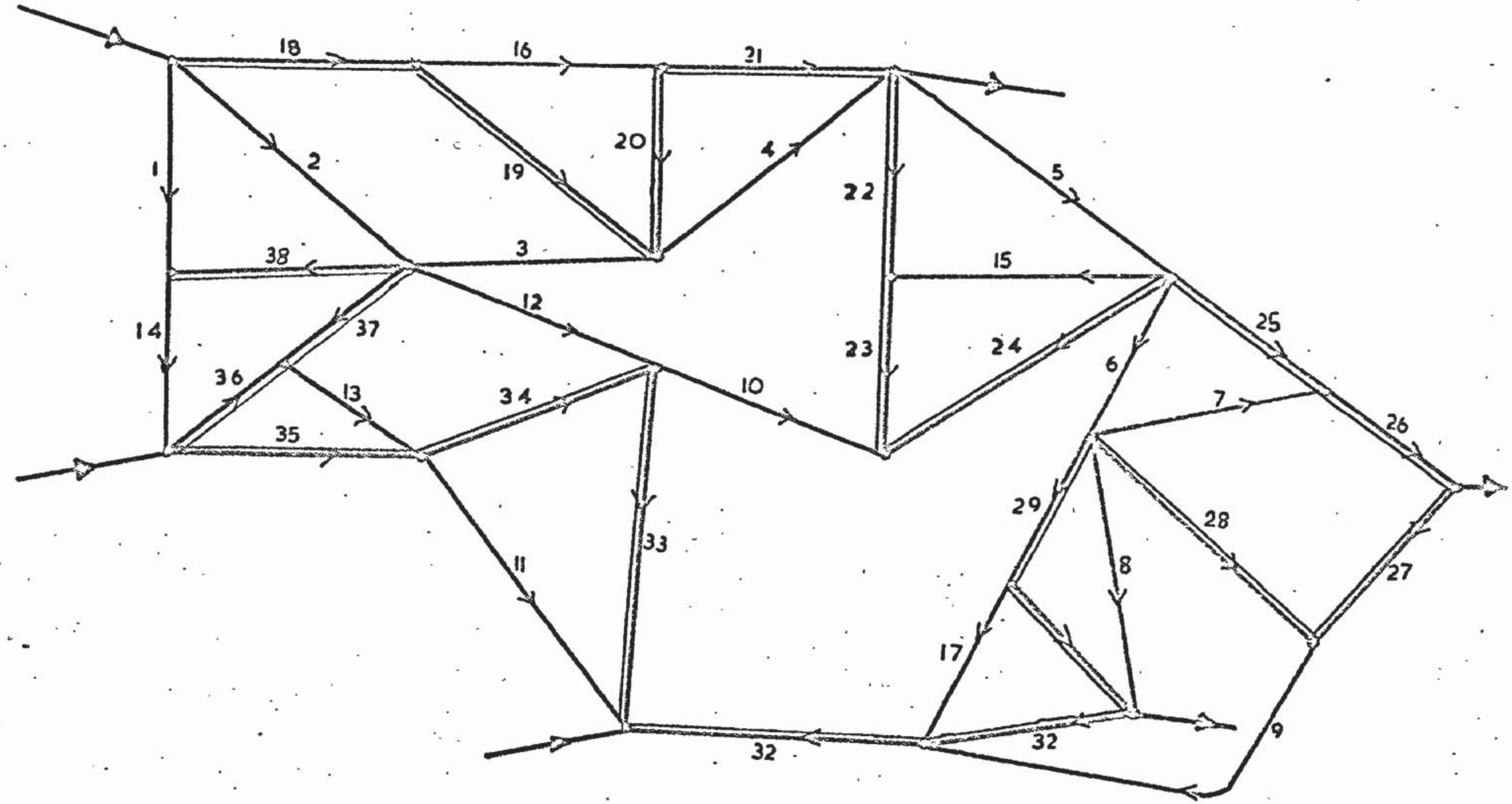


FIG 19
HARDY CROSS CASE 3
TRUNK

B. Comparison of Diakoptics Results for Networks reported in the Literature

A comparison of three networks reported in the literature was attempted It has been found however that the value of these networks as valid comparisons is somewhat limited.

i) Results of the Network due to Knights and Allen

Tables 4 and 5 show the dimensions of the individual branches and the demands at the nodes of the network shown in Fig (20) Unfortunately the properties of the Towns gas used in the analysis had to be taken from Perry (14) as the viscosity and density used by Knights and Allen were not reported.

Two analyses of the network were undertaken, one of the whole network and one with cut branch numbers 4,5,6, and 7 removed.

Tables 6 and 7 show the results obtained for the complete network compared with those reported, with percentage differences of flow and nodal pressure based on the results of the diakoptics method. Tables 8 and 9 are a similar analysis of the network with the given branches removed.

The results for the individual branch flows can be seen to be in good agreement, large percentage errors occuring only in branches which have small flows. The agreement is much better than that reported by Ingels and Powers. They compared the percentage difference from their results and those from Dolan who used a straight line approximation for the friction factor and found that the difference was an average about 20% to 30%.

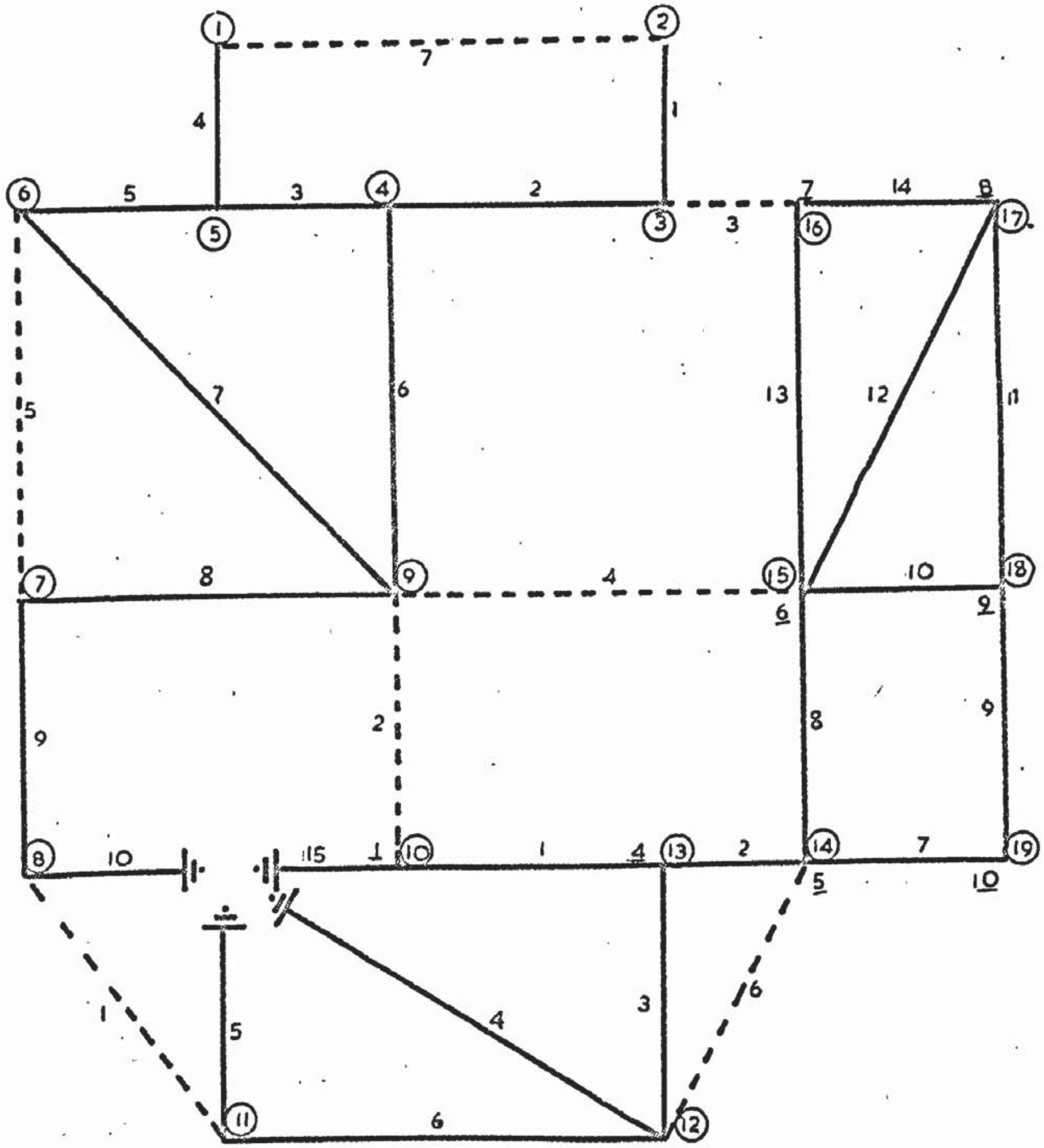


FIG 20

NETWORK DUE TO KNIGHTS ALLEN

In the region of most flows in the above network the straight line plot has obviously been chosen such that agreement is good i.e. $N_{Re} \simeq 5 \times 10^4$ to 5×10^7

The error in the nodal pressures can be seen in most cases to be a constant and of the order of 6 to 7%, this is attributed to the difference in the viscosity and density data.

ii) Comparison with the Results of Ingels and Powers

Tables 10 and 11 show the dimensions and nodal demands of the network shown in Fig(21) This network is due to Dolan (15) and was used by Ingels and Powers as a comparison with their calculation. Dolan calculated his flows as a percentage of the total input since he used a simple power law flow relation. Ingels and Powers set as an input an arbitary quantity of fluid at 780,000 lb/hr.

Table 12 shows the results obtained by Dolan , Ingels and Powers and shows the percentage difference based on Diakoptics and the Reynolds numbers for the branches.

It can be seen that although some of the differences are very large these are associated with branches carrying very small flows. The large number of such branches suggests that the input to the system has been chosen about an order of magnitude too small. This is somewhat surprising since Dolan's original analysis was for the performance under a firefighting flow from node (2)

No comparison of nodal pressures can be given because these were not reported.

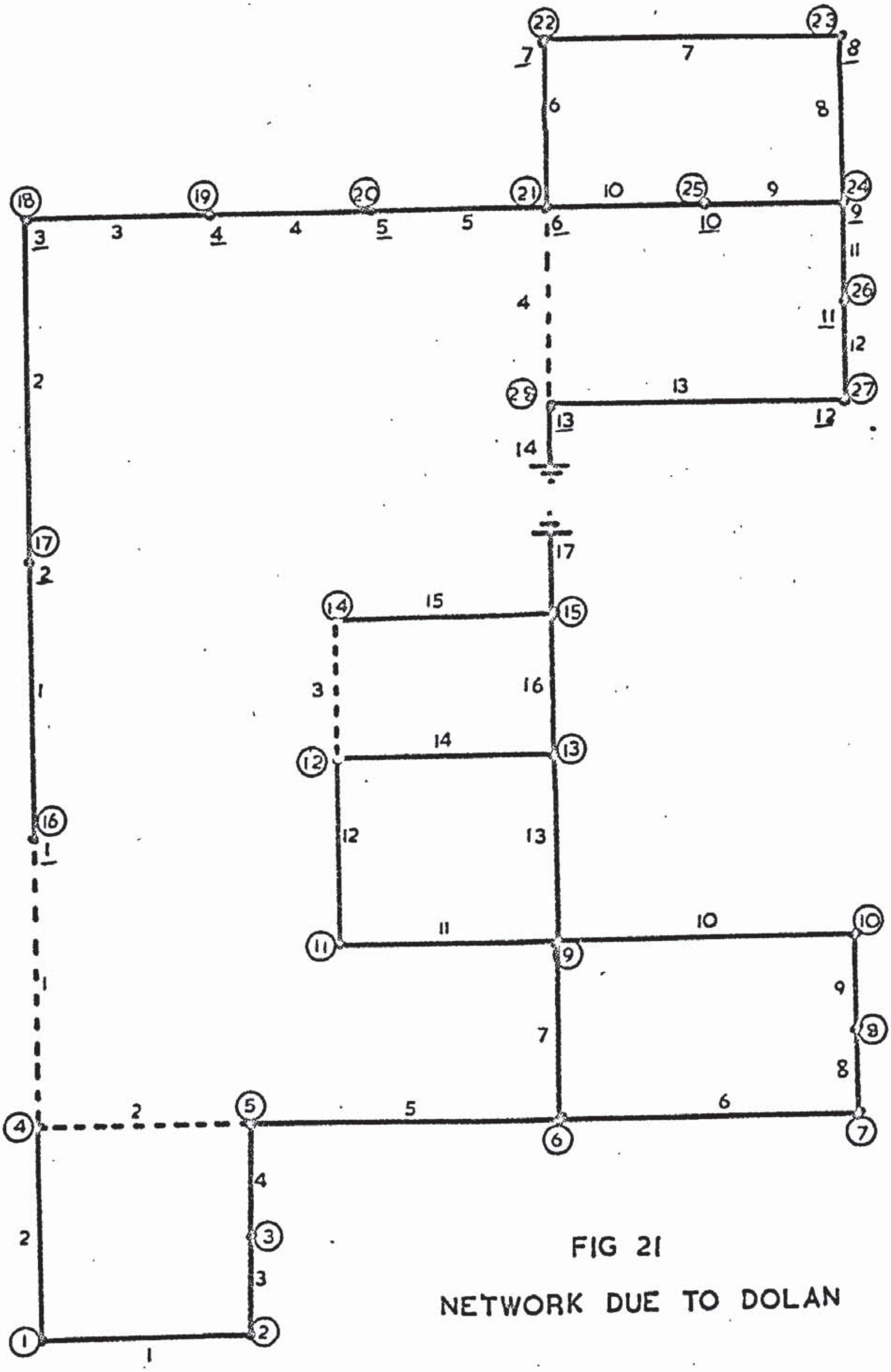


FIG 21

NETWORK DUE TO DOLAN

iii) The Network of Hunn and Ralph

Tables 13,14 give the dimensions of this network shown in Fig (22) Unfortunately no direct comparison of their results can be attempted as their pipe resistance factors bear no relation to actual values. Their inputs to the system are also approximately an order of magnitude too high. This results in for example, a velocity of 180 ft/sec with a pressure drop of 7ft water in pipe (9) segment 1 the dimensions of which are diameter 12" length 2000ft . The network was analysed however because it illustrates two further points in the programs use.

Firstly the network contains pumps which feed water from a river into the network. Secondly, the river can be considered as a datum node. Therefore no artificial datum is required and the network can be cut in any arbitary manner as long as each cut segment contains a pipe connected to the river.

The results are presented in Appendix F.

C. General Performance of the Diakoptics Program

i) Effect of different cutting patterns

Figs (23 to 30) show the test network fig (16) with four different cutting patterns, for which the relevant data are summarised in table 15 All four cases converged in eight iterations but the time per iteration varied and is shown in table 16

The results confirm what would be expected from the nature of the

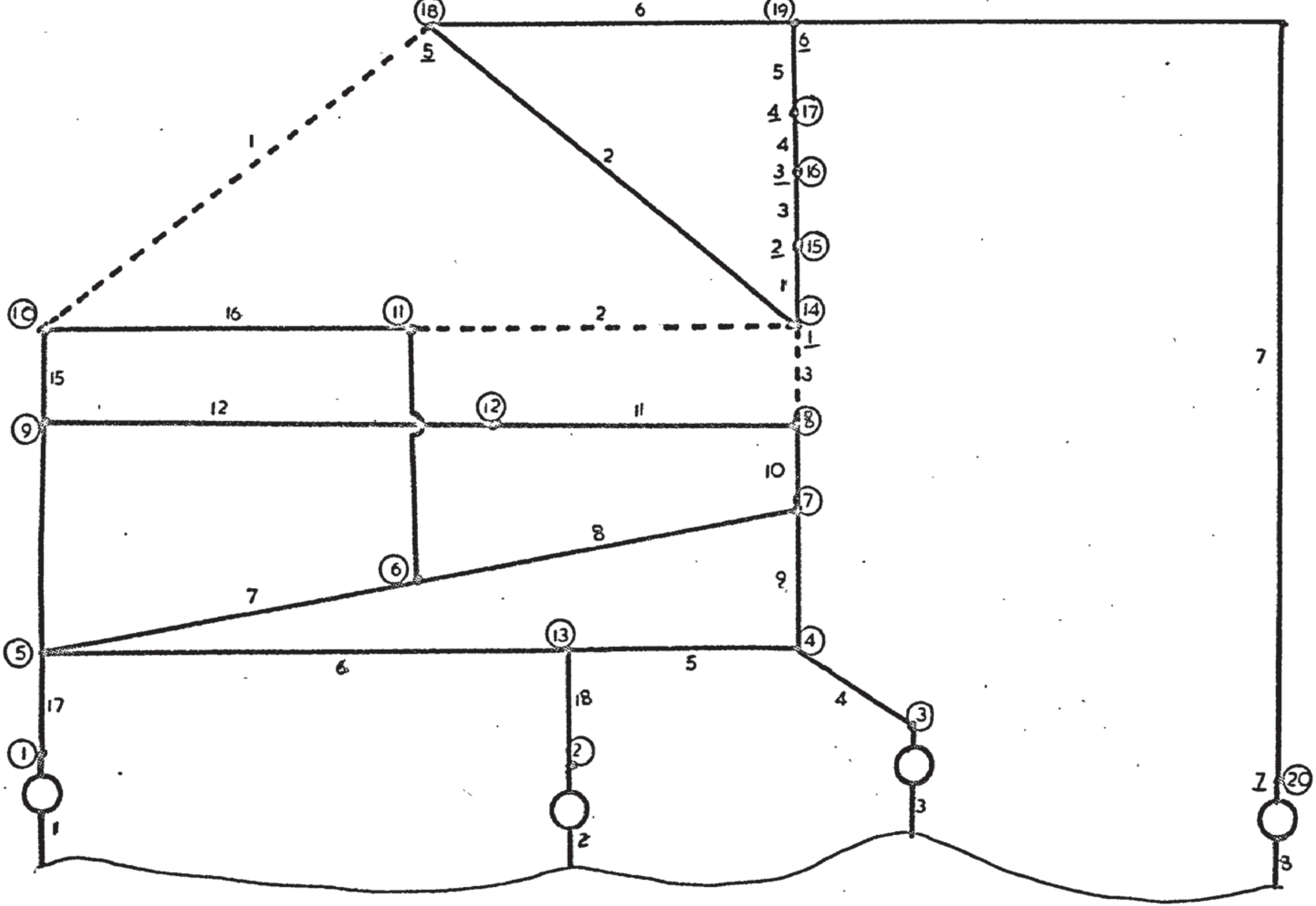


FIG 22

NETWORK DUE TO HUNN AND RALPH

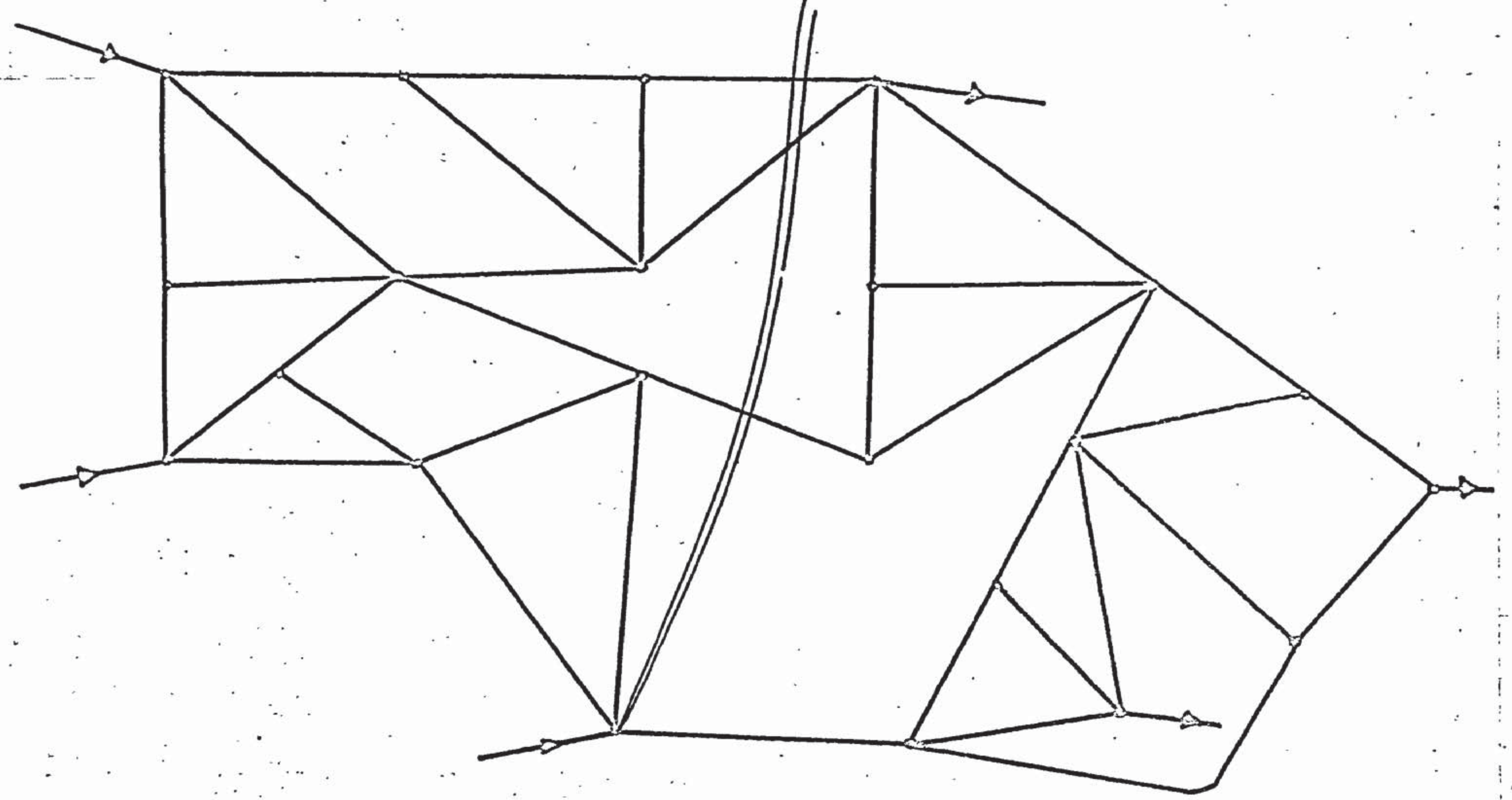


FIG 23
DIAKOPTICS
PROPOSED CUTS
CASE I

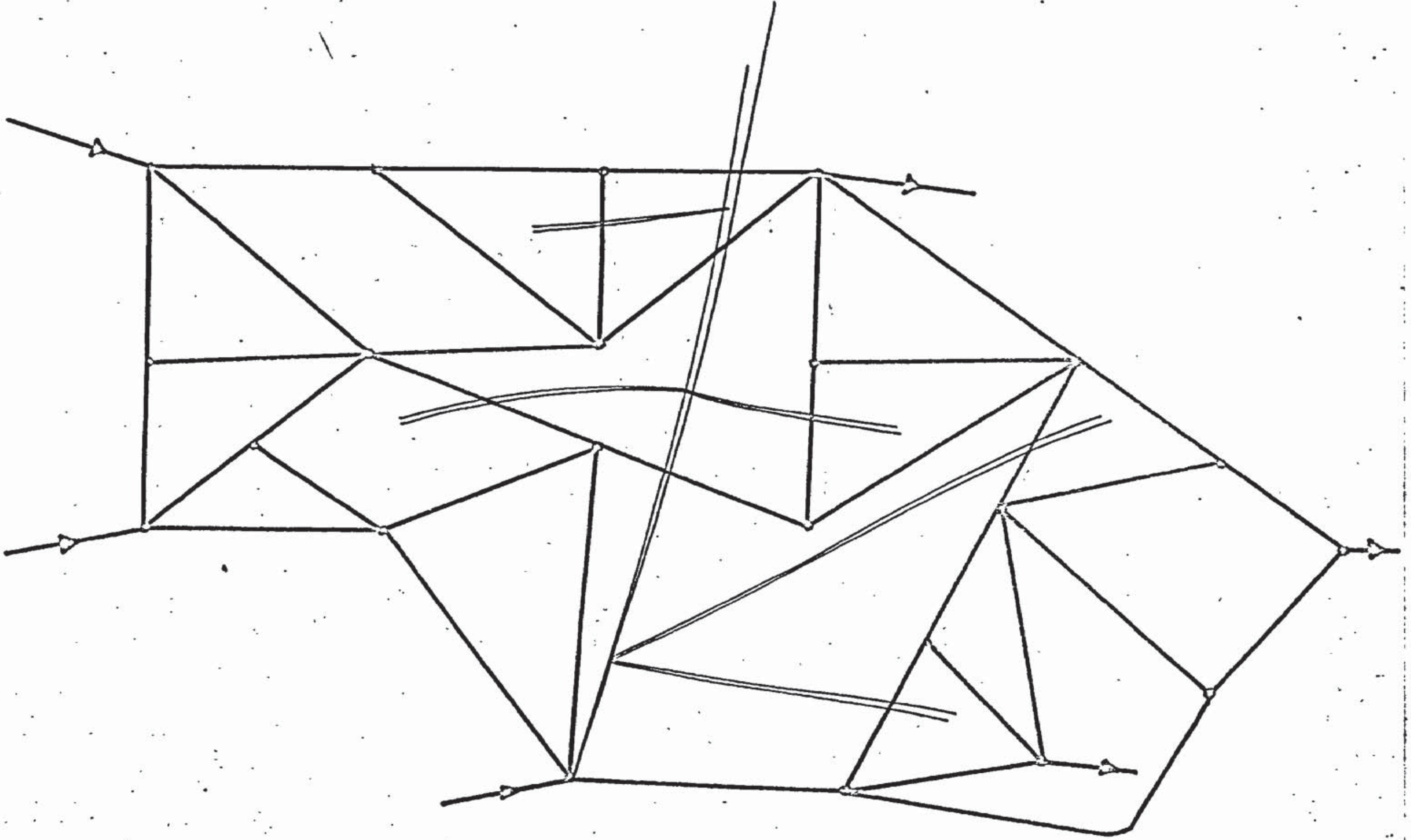


FIG 24
PROPOSED CUTS
CASE 2

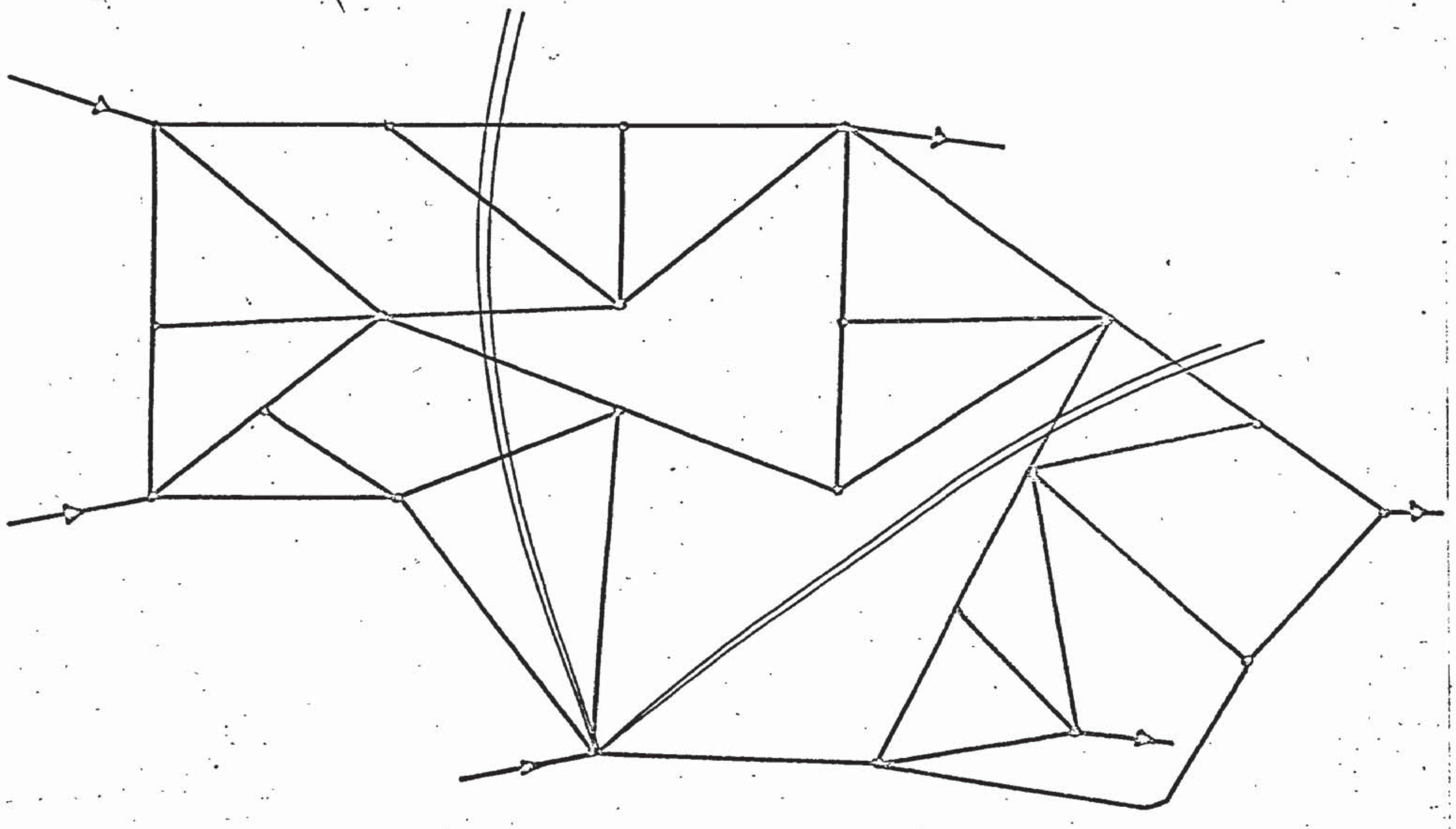


FIG 25
PROPOSED CUTS
CASE 3

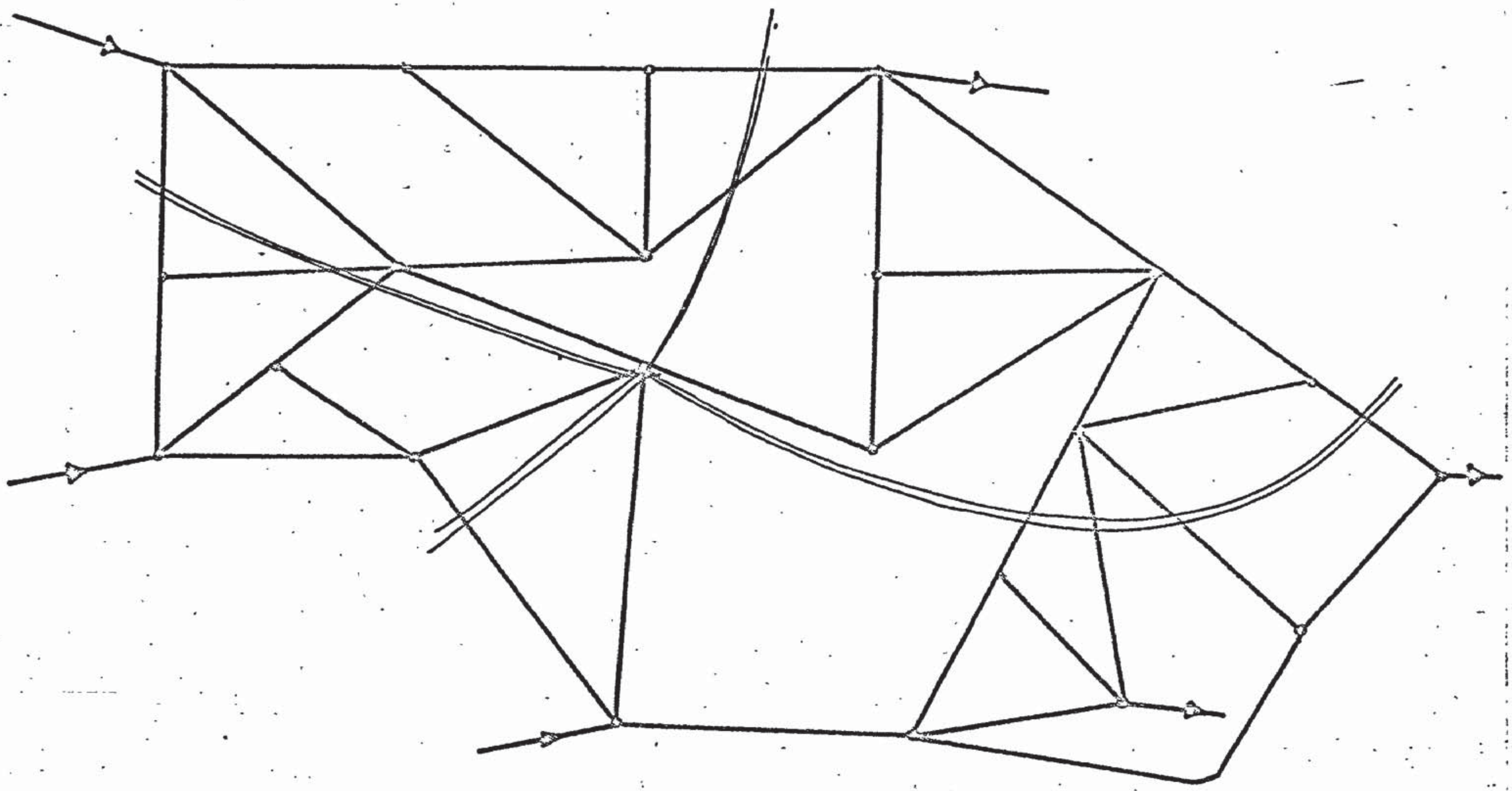


FIG 26
PROPOSED CUTS
CASE 4

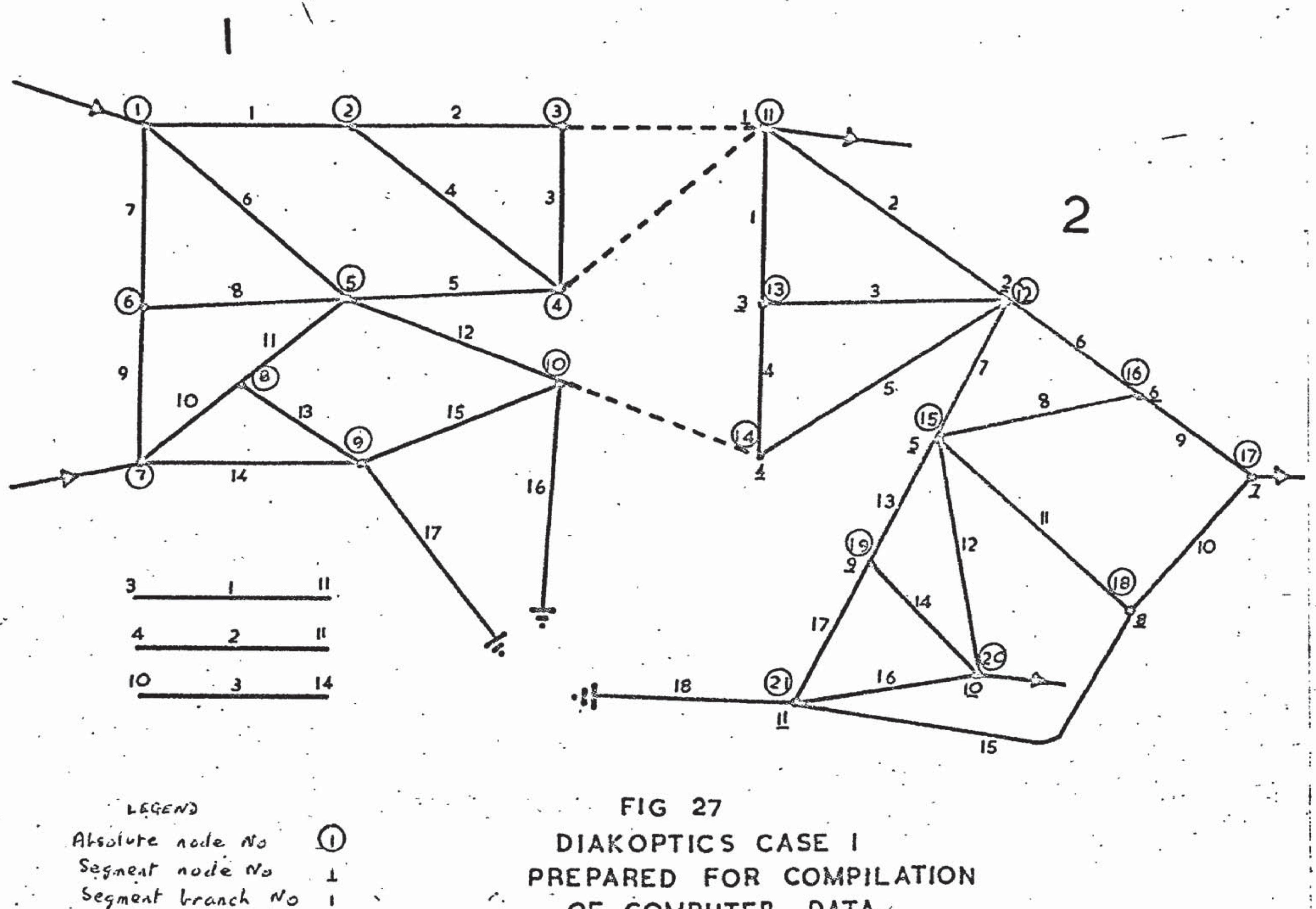


FIG 27
DIAKOPTICS CASE I
PREPARED FOR COMPILATION
OF COMPUTER DATA

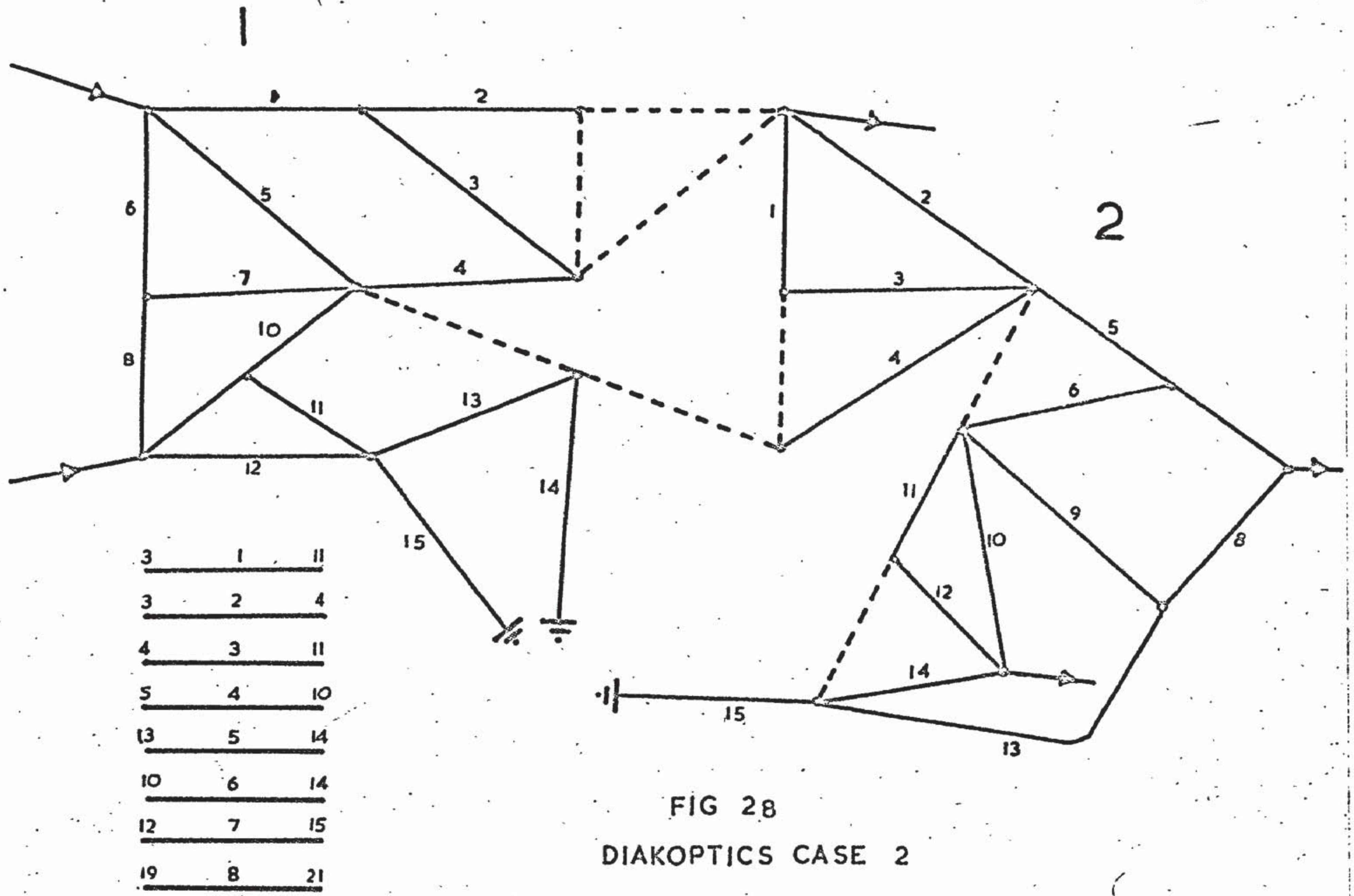


FIG 28

DIAKOPTICS CASE 2

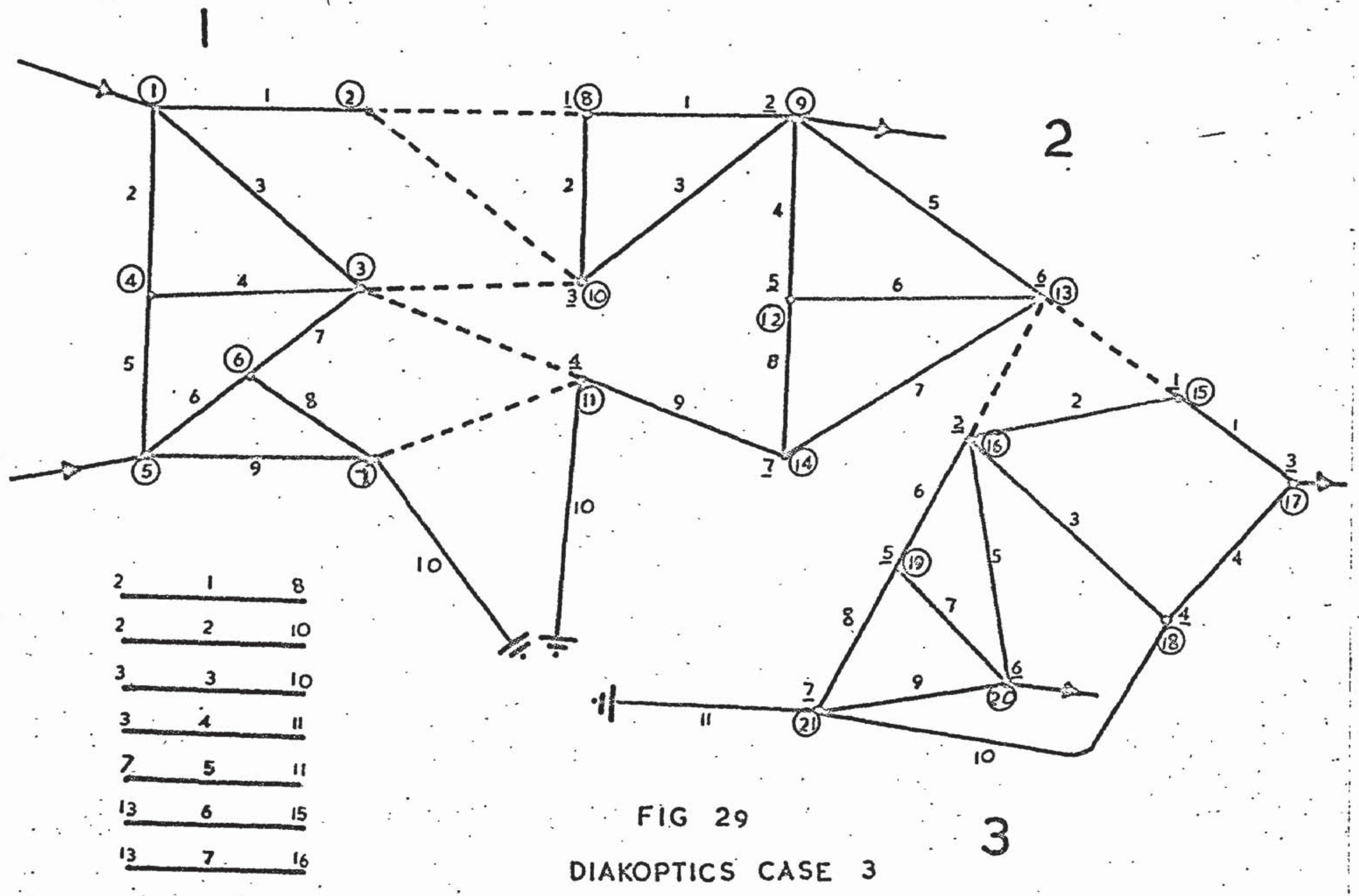


FIG 29

DIAKOPTICS CASE 3

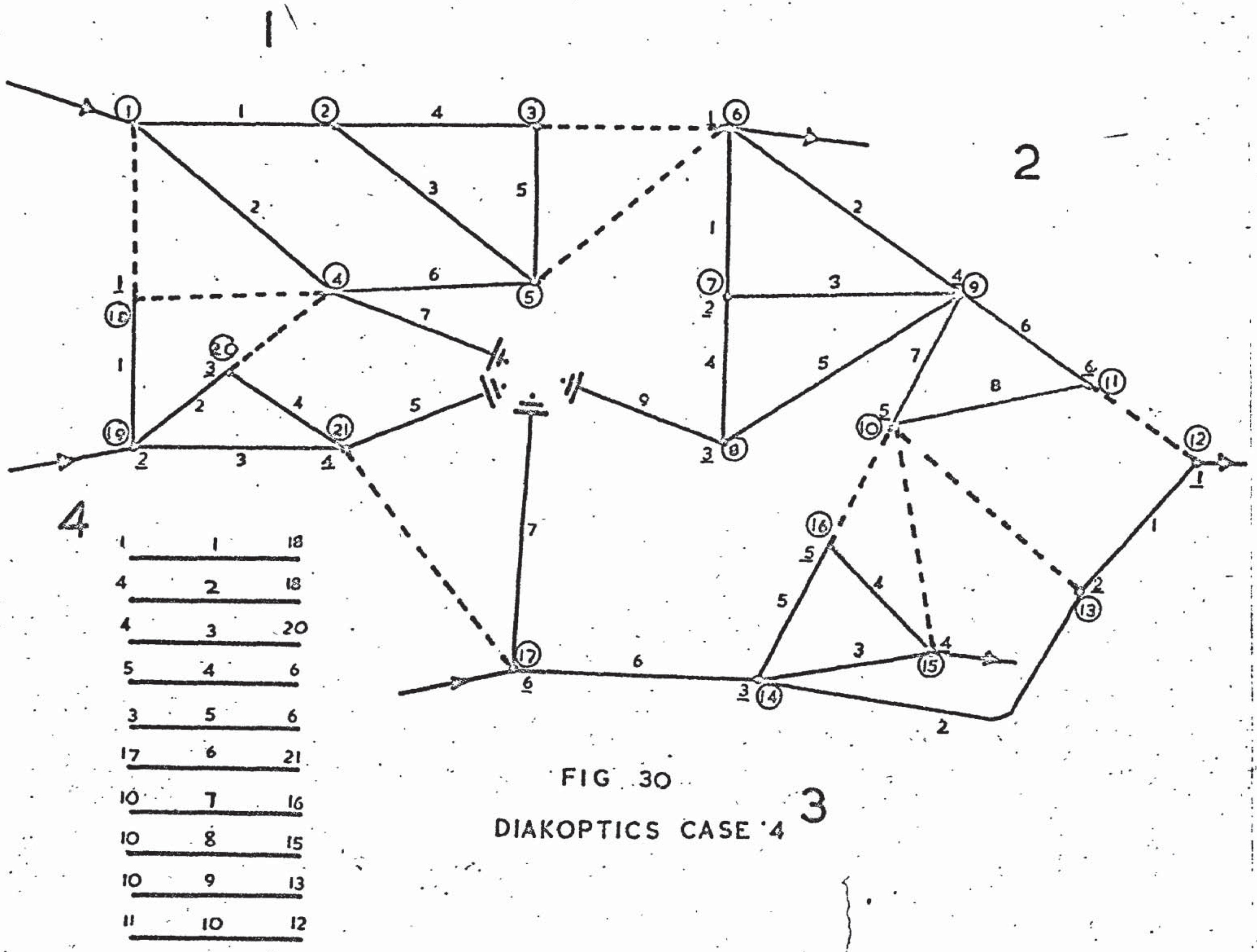


FIG 30

DIAKOPTICS CASE 4

method (see summary Chapter 3 section H) That is since the calculation involves the inversion of the admittance matrix for each segment and the inversion of the cut pipe resistance matrix the minimum number of operations is required when the number of nodes per segment is the same and the number of segments is as large as possible with the restriction that the number of cut branches be not greater than the number of nodes per segment.

ii) Convergence

To show the rate of convergence, the pressure vector $\underline{e}'_B$ was printed out after each iteration cycle. The absolute error for selected nodal pressures from their final value was plotted against iteration number Fig (31) on a linear scale and in Fig (32) as log (error) against iteration number. The percentage error was also calculated and plotted as shown in fig (33)

Case 1 was also run with initial guesses 1ft^3/min and 2500 ft^3/min for the flow in the individual pipes. The number of iterations required for convergence of the three cases are presented in Table 17.

In an attempt to explain the rapid convergence and its stability with widely differing inputs the pressure drops across certain of the branches of case 1 with an initial guess of 50ft^3/min for the branches are presented in Table 18

Now whatever the calculated pipe admittances for the first iteration the inputs and demands at the nodes dictate that these calculated branch flows will be of the correct order. For the next iteration therefore the pipe resistances will be a fair approximation as the change in friction

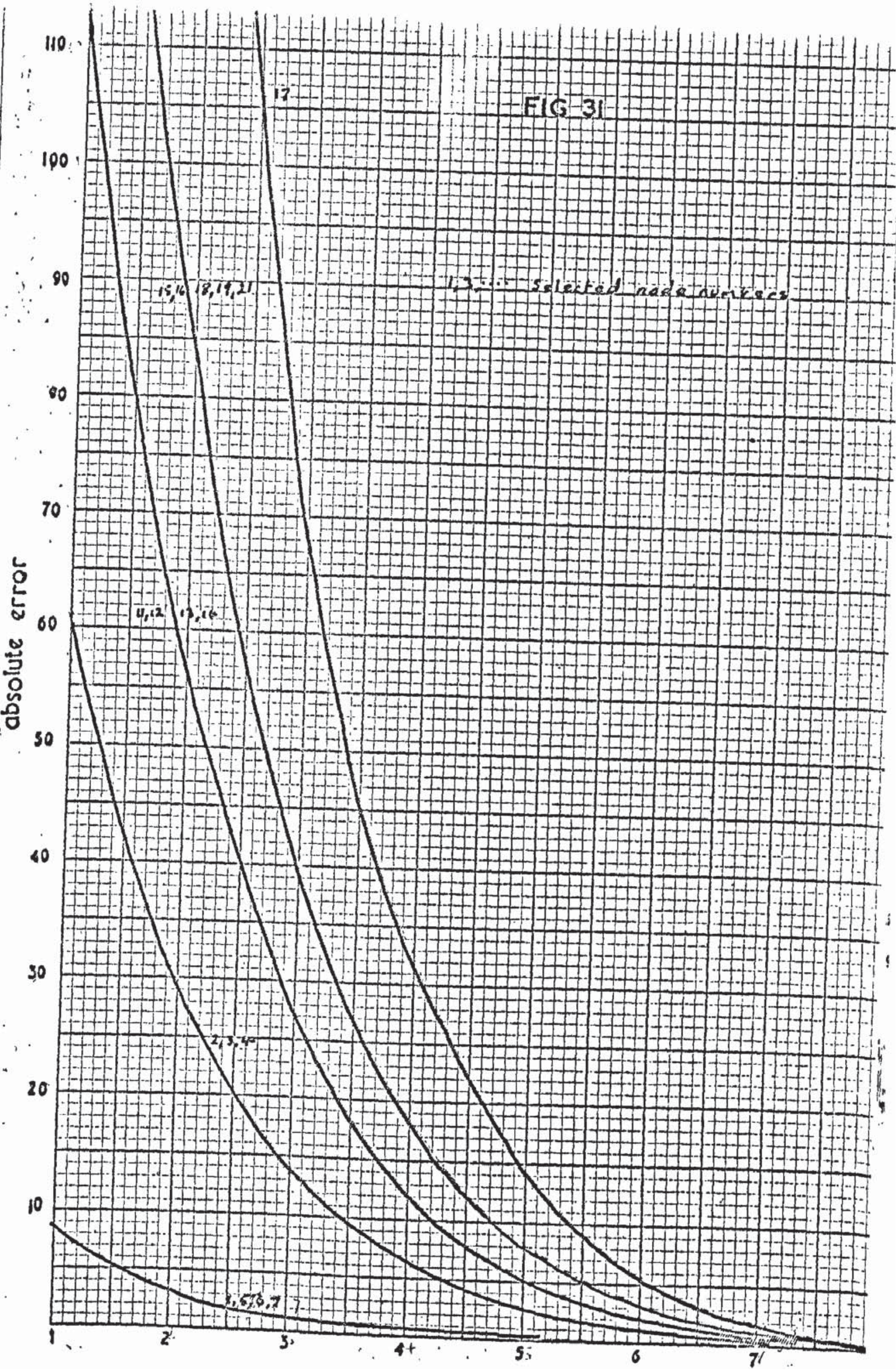

absolute error
110
100
90
80
70
60
50
40
30
20
10
1
2
3
4
5
6
7
17
15,16
18,19,21
11,12
13,14
2,3,4
Selected node numbers

FIG 31

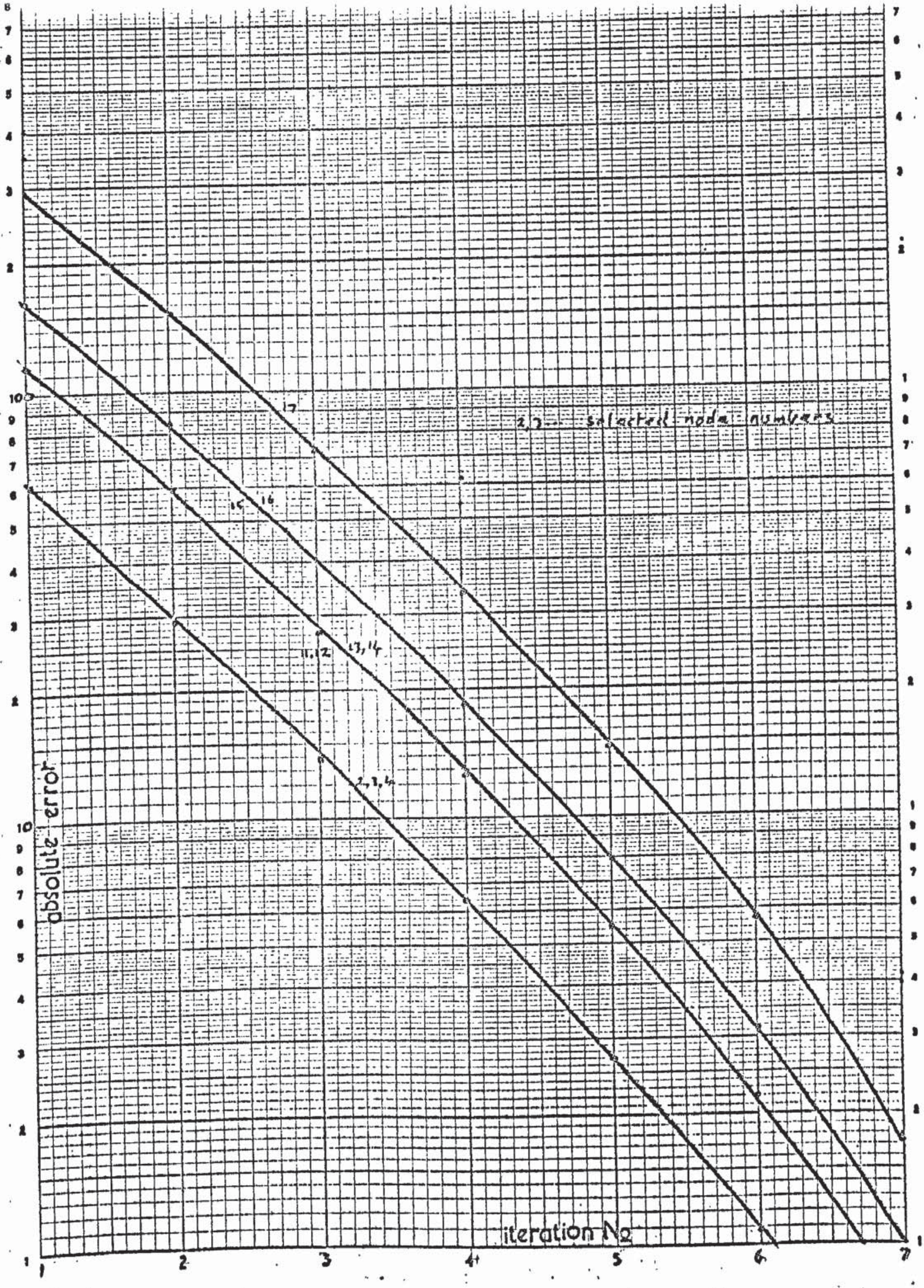
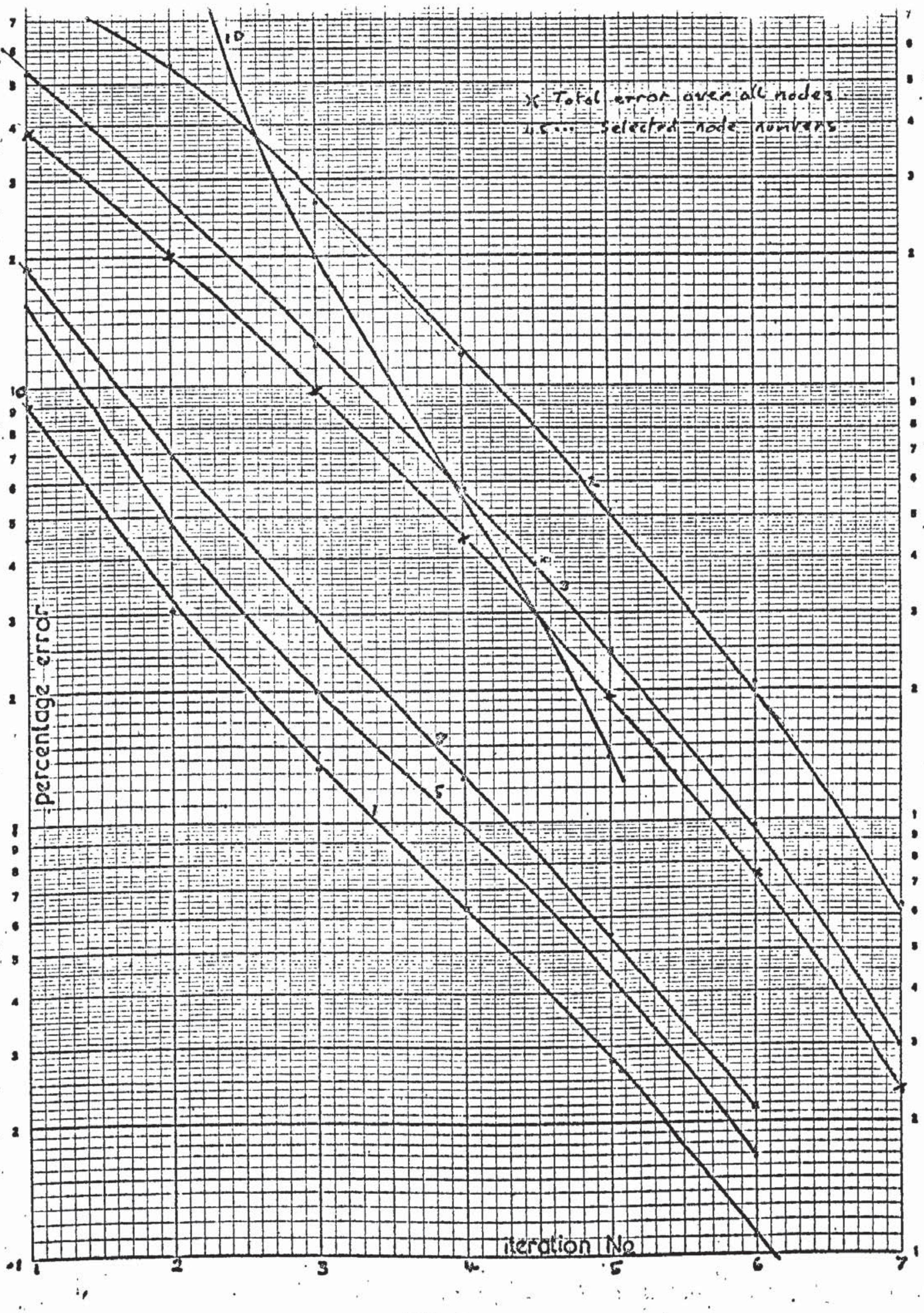

absolute error
iteration No
selected node numbers
17
15
16
11,12
13,14
100
10
1
2
3
4
5
6
7

FIG 32



x Total error over all nodes
4,5 ... selected node numbers
percentage error
iteration No

FIG 33

factor with flow is relatively small. One would then expect that the pressure drops would rapidly converge to their true values although the absolute nodal pressures could still change.

D. Performance of the Program when the Shape of the network is changed

For this analysis case 2 of thetest network was taken. Six different changes in shape were attempted. A seventh case taken from the analysis of the network due to Knights and Allen Fig 20 is also given. These are summarised below:-

a) From the full network cut branch 8 was removed

b) From the full network cut branch 2 was removed

c) From case b cut branch 3 was removed

d) From case c cut branch 6 was removed

e) From case d cut branches 2,3, and 6 were replaced thus reforming the full network

f) From the full network cut branches 2,3,6, and 8 were removed

g) From the second case of the network due to Knights and Allen cut branches 4,5,6, and 7 were added thus reforming full network.

The results are summarised in table 19 showing the number of iterations to reach a solution to the new problem and the percentage difference in the change of nodal pressures.

It can be seen that if small changes are made then the solution is rapid. However certain pipes in the system are critical and their removal drastically changes the flow pattern and the nodal pressures, these therefore need the larger number of iterations shown. It can be seen that changes c) and f) in fact change the nodal pressures by a greater percentage than the change from the inital guess of 50ft^3/min to the final solution of case 1 as shown in fig (2) however the number of iterations needed remains the same.

E. Performance when Changing the Nodal Demands

Case 2 was again taken, the changes in the nodal demands being

a) Input at node 1 changed to 0ft^3/min

b) Input at node 1 restored to 60ft^3/min

c) Output to node 20 increased to 90ft^3/min

d) Output to node 20 restored to 60ft^3/min

Fire fighting flow of 240ft^3/min taken from node 12

e) From original network Input at node 1 increased to 126ft^3/min

f) From e) new demand of 6ft^3/min taken from node 16.

g) Original network demands restored.

Table 10 shows the number of iterations to reach a solution of the new problems outlined above. It can be seen that they have the same pattern as the results of changing the shape i.e. small changes are executed rapidly, large changes take up to a maximum of eight iterations.

F. Discussion of Results

The results clearly show that the advantages claimed for the method are borne our in the actual computation of problems.

The method is at least as efficient in time as the Hardy Cross approach and much more efficient in its storage requirements. The efficient use of fast access storage is becoming of increasing importance with the wide-spread use of multiprogramming facilities, as the smaller the storage requirements of each program the greater the number of programs that can be run simultaneously. From the engineer's standpoint the data are much easier to compile and changes in the system are quick and simple to execute and whole series of changes can be attempted in one run automatically.

If the same changes as shown in section D for example were run with a Hardy Cross solution a new set of basic meshes would have to be found together with a new set of initial guesses as to the individual branch flows which satisfied Kirchoffs first law for each example. Changes similar to those in section E would also entail recalculation of the individual branch flows so that the nodes where new demands were applied would obey Kirchoffs first law.

The method can be seen to be very insensitive to guesses as to the individual branch flows and it is ~~becoming~~ probably not worth the effort of the engineer to even attempt any estimations, but to have the program set some value for them, so that they are never input as data.

An optimum policy for cutting the network has also been proposed in that one should endeavor to cut the network up into the largest number of equi-nodal pieces while keeping the number of cut branches at a minimum.

Chapter 6.

Further Discussion on the Network Concepts

A. The Solution of Design Problems

The above development has assumed that the problem has been completely specified and that the only unknowns in the system have been the node to datum potentials and the mesh currents. In Design problems in general however not only do the problems tend to be underspecified or, more rarely, overspecified but the unknown quantities are not confined to the above two vectors. In such systems there is then the extra problem of applying constraints in such a manner that a numerical solution can be attempted. It is the purpose of this section to show some of the systematic ways in which such constraints can be chosen.

Consider the basic orthogonal equation.

$$\begin{bmatrix} \underline{I}'_1 + \underline{0} \\ \underline{I}'_2 \quad \underline{i}'_2 \end{bmatrix} = \underline{Y} \begin{bmatrix} \underline{E}'_1 + \underline{e}'_1 \\ \underline{E}'_2 \quad \underline{0} \end{bmatrix}$$

Now as the dimensions of each vector is branch xl the total number of variables not counting the admittance matrix $\underline{Y}$ is

$$\underline{I}'_1 = n-1$$

$$\underline{I}'_2 = m$$

$$\underline{i}'_2 = m$$

$$\underline{E}'_1 = n-1$$

$$\underline{E}'_1 = m$$

$$\underline{e}'_1 = n-1$$

$$\text{Total} = 3b$$

Since there are only b equations then for any solution 2b variables must be specified as data.

It will be necessary to discuss the composition of the vector $\underline{I}'_1$. In the original problem it was explained that a demand vector $\underline{I}$ can be associated with the branches and the transformation $\underline{\tilde{A}}\,\underline{I}$ assigned a corresponding vector to the nodes. This transformation is still valid for:

$$\begin{bmatrix} \underline{I}'_1 \\ \underline{I}'_2 \end{bmatrix} = \underline{\tilde{A}}_1 \underline{I}$$

In the formation of the branch demand vector $\underline{I}$ the individual branch terms can be assigned in any arbitary manner as long as they sum to the individual nodal demands. If we assigned individual branch demands to only the branches of the tree, then for the original problem,* noting that a positive branch flow is opposite to the assumed direction.

$$I = \begin{array}{c|c|} 1 & 4 \\ 2 & 2 \\ 3 & 0 \\ 4 & 4 \\ 5 & 0 \end{array}$$

and

$$\underline{\tilde{A}}_1 \underline{I} = \begin{array}{c|c|} 1 & -2 \\ 2 & 2 \\ 3 & 4 \\ \hline 4 & 0 \\ 5 & 0 \end{array}$$

* See fig..7.

Therefore $\underline{I}_2' = 0$

Note however that for the same example

$$\underline{I} = \begin{array}{c|c|} 1 & 0 \\ 2 & 2 \\ 3 & 0 \\ 4 & 0 \\ 5 & 4 \end{array}$$

is just as valid, then

$$\underline{\tilde{A}}_1 \underline{I} = \begin{array}{c|c|} 1 & -2 \\ 2 & 2 \\ 3 & 4 \\ 4 & -4 \\ 5 & 0 \end{array}$$

Such an arrangement can be useful in certain problems as will be shown below.

We can now summarise with the aid of suitable examples how any network problem can be quickly checked to see what additional information is needed, or which design variable must be released so a solution can be obtained.

For example the original network has to be changed so that the demand from node 2 be increased to 4. The problem is to determine what can be left constant in the old system. For instance can all the nodal pressures remain at their present value if additional pumps are added to the system?

Now $\underline{I}_1' = \underline{Y}_1' (\underline{E}_1' + \underline{e}_1') + \underline{Y}_2 \, \underline{E}_2'$ and the solution to this problem requires only these three equations.

Total number of variables in problem

= n-1 + n-1 + n-1 + m

= 11

Known quantities $\underline{I}'_1$ = 3

$\underline{e}'_1$ = 3

= 6

Number of unknowns in problem = 3 only 2 pump terms can remain unchanged, for example.

In a more general case suppose that for the above network we know 1 branch flow, 2 nodal demands and two nodal pressures, what additional information is required for a solution?

In such totally mixed problems it is a great advantage to start from the individual branch equations derived via the transformation matrices. For the example we can write

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} I_1 + i'_1 \\ I_2 + i'_2 \\ I_3 + i'_2 \\ I_4 + i'_1 + i'_2 \\ I_5 - i'_1 \end{bmatrix} = \underline{Y}_p \begin{bmatrix} E_1 + e'_3 - e'_1 \\ E_2 + e'_2 - e'_1 \\ E_3 + e'_2 \\ E_4 + e'_1 \\ E_5 + e'_3 \end{bmatrix}$$

Consider two cases for which

e'_1	=	3.82		e'_1	=	3.82
e'_3	=	8.49		e'_3	=	8.49
i_5	=	2.83	and	i_4	=	1.56
I'_1	=	-2		I'_1	=	-2
I'_2	=	2		I'_2	=	2
E_1	=	0		E_1	=	0

Case 1

Assigning individual branch demands to the tree then $I_5 = 0$ as above

$\therefore i_5 = -i'_1$ so that $i'_1 = -2.83$ and $I_1 = I_4 = I'_3$

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} I_1 - 2.83 \\ 2 + i'_2 \\ 0 + i'_2 \\ I_4 - 2.83 + i'_2 \\ 0 + 2.83 \end{bmatrix} = \underline{Y}_p \begin{bmatrix} 0 + 8.49 - 3.82 \\ E_2 + e'_2 - 3.82 \\ E_3 + E'_2 \\ E_4 + 3.82 \\ E_5 + 8.49 \end{bmatrix}$$

It can be seen that as E_5 and I_1 are completely determined by the data the number of simultaneous equations is reduced to 3. In these we have 5 unknown quantities. ∴ two extra pump terms must be specified for a solution. For the second case if is more helpful to use the second $\underline{I}$ vector outlined above, page 61, constraining the nodal demands to flow in branches 2 and 5. This gives zero entries in branch 1 for example so that i'_1 can be determined uniquely.

Case 2

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{bmatrix} 0 & + & i_1' & \\ 2 & + & i_2' & \\ 0 & + & i_2' & \\ 0 & + & i_1' & + i_2' \\ I_5 & - & i_1' & \end{bmatrix} = Y_p \begin{bmatrix} 0 & + & 8.49 - 3.82 \\ E_2 & + & e_2' - 3.82 \\ E_3 & + & e_2' \\ E_4 & + & 3.82 \\ E_5 & + & 8.49 \end{bmatrix}$$

From 1, i_1' can be uniquely determined. With the knowledge of $i_4 = 1.56$

Then $1.56 = i_1' + i_2'$

$$\therefore i_2' = 0.39$$

We have remaining then 6 unknowns and 4 equations. .two pump terms must be specified.

To summarise the approach to mixed problems

1) Check number of known quantities

2) Check to see if any vector is completely known and if only part of the orthogonal equations are needed for a solution e.g. case 1 above

3) If equations of solution cannot be partitioned, reduce to the primitive system.

4) Remove equations, if any, that are completely specified and form new set of simultaneous equations for solution.

B. Partial Differential Equations

The finite difference technique for the numerical solutions of the heat and mass transfer equations is easily amenable to the diakoptics approach. Fig (34) shows an operational calculus diagram similar to

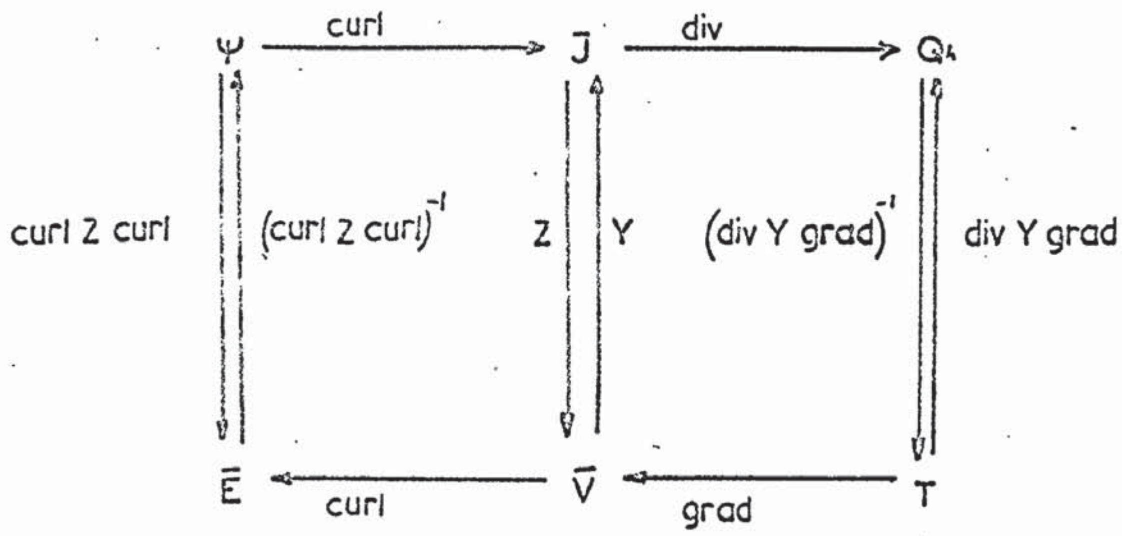


FIG 34

OPERATOR DIAGRAM FOR VECTOR CALCULUS

the algebraic diagram of Roth. It can be seen therefore in converting a problem such as the three dimensional heat conduction equation i.e.

$$C_p \frac{\partial T}{\partial t} = \text{div } K \text{ Grad } T$$

to a finite difference solution the operator grade is equivalent to $\underline{A}$. and the operator div equivalent to $\underline{\tilde{A}}$ where $\underline{K}$ is the primitive matrix of conductivity. In the simplified case of K being invariant with temperature if the cut segments are chosen such that they are of equal dimensions, then with the exception of the segments containing the boundary conditions they will be numerically similar. In which case having inverted one such segment the resulting matrix can be used for all others. Thus not only is the computation required reduced but only one such matrix need be stored. It is realised that as the set points are in general regular in space then the number of cut branches can become very large. However the whole system need not be connected together in one operation, but each cut segment can be connected together sequentially thus keeping the cut branch matrices small.

C. Systems with mixed linear and non-linear admittances

The solution of systems in which the admittance elements are a mixture of linear and non-linear quantities can cause serious computational problems as the standard non-linear numerical methods sometimes fail to converge. However for a solution of network problems if the non-linear elements are all confined to the cut branch set then they are isolated from the linear system. This results in a much faster iteration cycle as the inverted linear systems are invariant and only the cut segment terms change so that the process has only to cycle through part of the connection process.

Chapter 7

Conclusions and Further Work

The computer program described in this work can be seen to conform closely to the criteria set out in the Introduction. The method not only converges to a solution more quickly than the Hardy Cross approach but is very insensitive to the error of the initial guesses. The data are simple to compile and therefore take less time with less opportunities for error. Simple rules have been formulated to enable a network to be cut so as to ensure an efficient solution.

It is not necessary to input a feasible solution to start the iteration cycle. This means that if the network is to be analysed under a set of small changes a good approximation exists in the machine which does not need to be modified by the user to conform to Kirchoffs Laws. Also when changing the shape of the network branches can be added to or removed from the network by just changing the composition of the cut branch set. No new data on loop formation have to be input. It has been shown that for the above reasons both man and machine time are greatly reduced. For example, the two cases reported by Knights and Allen took 40 to 35 iterations to converge whereas the diakoptics program executed the change automatically and converged in half the time of the original solution

The information on shape and the solution of each segment is treated and stored as a separate entity. This enables not only the solution of very large systems to be attempted but the larger the system the more efficient the method becomes compared with the Hardy Cross approach. Each segment can be connected to any other segment in any arbitary manner and so it is possible to build up a library of segment shapes and solutions which can be reformed with any system by only the addition of cut pipe data.

It has also been demonstrated how by using the theoretical basis of the program any system containing mixed known and unknown quantities can be quickly checked for under or over specification and how such systems can then be solved. No analysis of this type can be attempted by the Hardy Cross technique. Diakoptics therefore is not only a method of solution but provides a logical framework through which the designer can easily find what constraints must operate in any system given its design specification.

Now it was realised at an early stage in the work when the theory of diakoptics was being investigated and as the above development was formulated that it had a much wider application to chemical engineering than just the complex pipe network problem. The technique can be applied to finite differences approximations of partial differential equations, and because the matrices formed in certain cases are equivalent, large amounts of computation time and storage can be saved as only one of these matrices need be inverted and stored.

It has also been suggested how in systems with mixed linear and non-linear elements the two classes can be separated so that the iterations required for solution need only cycle through the non-linear elements.

One other aspect of diakoptics has been found by the electrical engineers to be so useful that is has become at least as important as the computational advantages. The above development showed how from a knowledge of the individual branch admittances through the use of the connection matrices the segment admittance matrix was formed. These matrices or tensor admittances were then connected to form the complete system. Put in another way, from the basic elements of the system either described by scalars or tensors, the equations describing the total system were formed automatically. This automatic generation of the describing equations of highly complex systems has been demonstrated by Kron (20). Now it is a feature of modern chemical engineering to consider chemical plant from a systems point of view. The concept of breaking the system down into 'ultimate building blocks' whose equation or equations are known and through the use of connection matrices to form automatically the total describing equations has an immediate application therefore, to this way of thinking about chemical plant.

Further sets of transformations have been developed by Kron (20) for the solution of large systems containing only one ground point so that all the cut segments except one have a singular admittance matrix. The potential applications of these further extensions of diakoptics to chemical engineering are not clear, at the present time. One example may be the solution of the heat conduction equations where the boundary conditions vary in such a way that the nodes representing the boundary conditions all have a different temperature.

Appendix A

Worked example nodal analysis

Consider the network and its graph shown in Fig 7. The admittances of the impedance elements, the magnitude of the potential sources, their directions and the external current sources are shown.

The graph shows the node numbers, the branch numbers and the orientation of the branches. Note that the orientation of a branch containing a potential source is chosen to be opposite to the direction of the source, the other directions being assigned arbitarily.

The problem is to solve for the nodal potentials and the branch flows. Node 4 is taken as the datum node.

$\underline{A}$ =

	1	2	3
1	-1	0	1
2	-1	1	0
3	0	-1	0
4	1	0	0
5	0	0	1

In constructing the admittance matrix $\tilde{\underline{A}}\ \underline{Y}\ \underline{A}$ it is not necessary for simple impedances to perform the indicated matrix multiplications, since it can be formed from two simple rules:

a) The diagonal elements are the sum of the admittances of the branches incident at the node.

b) Each off-diagonal element is the negative of the admittance of the branch running between the nodes concerned.

$$\tilde{\underline{A}}\,\underline{Y}\,\underline{A} = \begin{array}{c|ccc|} & 1 & 2 & 3 \\ \hline 1 & 1 + \frac{1}{4} + \frac{1}{5} & -1 & -\frac{1}{4} \\ 2 & -1 & 1 + \frac{1}{2} & 0 \\ 3 & -\frac{1}{4} & 0 & \frac{1}{4} + \frac{1}{3} \\ \hline \end{array}$$

∴ by inversion

$$(\tilde{\underline{A}}\,\underline{Y}\,\underline{A})^{-1} = \begin{array}{c|ccc|} & 1 & 2 & 3 \\ \hline 1 & 1.497 & 0.986 & 0.634 \\ 2 & 0.986 & 1.323 & 0.423 \\ 3 & 0.634 & 0.423 & 1.986 \\ \hline \end{array}$$

Next, the vector $\tilde{\underline{A}}(\underline{I} - \underline{Y}\,\underline{E})$ is formed

$\hat{\underline{A}}\,\underline{I} = \underline{I}'$ the nodal input and demand current vector

$$\tilde{\underline{A}}\,\underline{I} = \begin{array}{c|c|} 1 & -2 \\ 2 & 2 \\ 3 & 4 \\ \hline \end{array}$$

Note that currents leaving a node are negative and currents entering a node are positive.

$\tilde{\underline{A}}\,\underline{Y}\,\underline{E}$ is the sum at each node of the connected potential source times the branch admittance i.e. the external current produced by this potential The sign of this current depending on the orientation of the source. At node 1, for example the potential source in branch 2 will produce a current leaving node 1 and entering at node 2.

$$\therefore \tilde{\underline{A}}\,\underline{Y}\,\underline{E} = \begin{array}{c|c|} 1 & -5 \times 1 + 4 \times \frac{1}{5} \\ 2 & 5 \times 1 - 2 \times \frac{1}{2} \\ 3 & 0 \\ \hline \end{array} = \begin{array}{|c|} -4.2 \\ 4.0 \\ 0 \\ \hline \end{array}$$

$$\therefore \tilde{\underline{A}}(\underline{I} - \underline{Y}\,\underline{E}) = \begin{array}{c|c|} 1 & 2.2 \\ 2 & -2.0 \\ 3 & 4.0 \end{array}$$

Premultiplying by $(\tilde{\underline{A}}\,\underline{Y}\,\underline{A})^{-1}$ we obtain the node to datum potential vector with respect to node 4.

$$\underline{e}' = \begin{array}{c|c|} 1 & 3.82 \\ 2 & 1.21 \\ 3 & 8.49 \end{array}$$

This vector constitutes the solution to the first part of the problem.

Hence from equation (4) the branch potential rise vector is

$$\underline{e} = \begin{array}{c|c|} 1 & 4.67 \\ 2 & -2.61 \\ 3 & -1.21 \\ 4 & 3.82 \\ 5 & 8.49 \end{array}$$

Now by equation (10)

$$\underline{V} = \begin{array}{c|c|} 1 & 0 \\ 2 & 5 \\ 3 & 2 \\ 4 & 4 \\ 5 & 0 \end{array} + \begin{array}{|c|} 4.67 \\ -2.61 \\ -1.21 \\ 3.82 \\ 8.49 \end{array} = \begin{array}{|c|} 4.67 \\ 2.39 \\ 0.79 \\ 7.82 \\ 8.49 \end{array}$$

By equation (13) $\underline{J} = \underline{Y}\,\underline{V}$ where $\underline{Y}$ is the primitive admittance matrix

$$\underline{Y} = \begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \frac{1}{4} & & & & \\ 2 & & 1 & & & \\ 3 & & & \frac{1}{2} & & \\ 4 & & & & \frac{1}{5} & \\ 5 & & & & & \frac{1}{3} \\ \hline \end{array}$$

$$\therefore \underline{J} = \begin{array}{c|c|} 1 & 1.17 \\ 2 & 2.39 \\ 3 & 0.395 \\ 4 & 1.56 \\ 5 & 2.83 \\ \end{array}$$

which is the vector of branch flows.

Note that since $\underline{V}$ is the potential rise vector in the direction of the orientated graph, positive flows in $\underline{J}$ implies a flow opposite to assumed direction, negative flows in $\underline{J}$ implies a flow in the assumed direction. Fig. (8) shows the nodal potentials and the branch flows obtained above which may be seen to satisfy Ohm's and Kirchoffs Laws.

Appendix B

Worked example diakoptics

For the given network it has been shown that for fig. 7.

$$\underline{S}' = \begin{array}{c|cc|} & 1 & 2 \\ \hline 1 & -1 & -1 \\ 2 & 0 & 1 \\ 3 & 1 & 0 \\ \hline \end{array}$$

From equation (19) we have elected to convert the segment pump terms into their equivalent nodal impressed loads. Note that just as $\underline{Z}_A$ need not be formed as a full matrix, $\underline{I}_B'$ is never used in practice in its complete form. It is necessary only to premultiply just the nodal demands of each cut segment by the admittance matrix for that segment. However the full vector will be formed here for sake of clarity.

$$\underline{I}_B' = \begin{bmatrix} -2 & -4 \times \frac{1}{5} \\ 2 & -(-2 \times \frac{1}{2}) \\ 4 & - \; 0 \end{bmatrix} = \begin{bmatrix} -2.8 \\ 3 \\ 4 \end{bmatrix}$$

The admittance and impedance matrices of each cut segment are given below. Note that since segment 4 is in its primitive state its matrix has only diagonal elements.

$$\underline{Y}_{A1} = \begin{bmatrix} \frac{1}{5} \end{bmatrix} \quad \underline{Z}_{A1} = \begin{bmatrix} 5 \end{bmatrix}$$

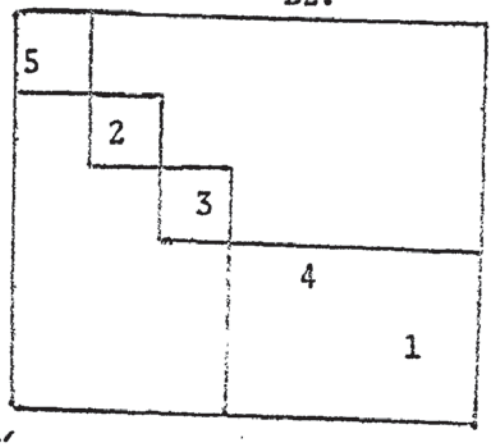
$$\underline{Y}_{A2} = \begin{bmatrix} \frac{1}{2} \end{bmatrix} \quad \underline{Z}_{A2} = \begin{bmatrix} 2 \end{bmatrix}$$

$$\underline{Y}_{A3} = \begin{bmatrix} \frac{1}{3} \end{bmatrix} \quad \underline{Z}_{A3} = \begin{bmatrix} 3 \end{bmatrix}$$

$$\underline{Y}_{A4} = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & 1 \end{bmatrix} \quad \underline{Z}_{A4} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

Thus there is a considerable saving in computer storage and computation time since each subnetwork's admittance matrix is formed and inverted separately. The full matrix $\underline{Z}_A$ is never in fact formed. However in this example for completeness $\underline{Z}_A$ will be used.

$\therefore \underline{Z}_A =$



$\underline{e}'_A = \underline{Z}_\alpha \underline{I}'_B$

$$\therefore \underline{e}'_A = \begin{bmatrix} -14 \\ 6 \\ 12 \end{bmatrix}$$

$$\underline{E}_1 = -\underline{\underline{\tilde{S}}}' \underline{e}'_A = \begin{bmatrix} -26 \\ -20 \end{bmatrix}$$

$$\underline{E}_2 = \underline{E}_1 + \underline{E}'_B = \begin{bmatrix} -26 \\ -25 \end{bmatrix}$$

$$\underline{Y}_\beta = (\underline{S}' \; \underline{Z}_\alpha \underline{S}' + \underline{Z}_\beta)^{-1}$$

$$= \begin{bmatrix} 12 & 5 \\ 5 & 8 \end{bmatrix}^{-1}$$

$$\underline{Y}_\beta = \begin{bmatrix} 0.1126 & -0.0704 \\ -0.0704 & 0.169 \end{bmatrix}$$

$$\underline{i}_1 = \underline{Y}_\beta \underline{E}_2 = \begin{bmatrix} -1.160 \\ -2.40 \end{bmatrix}$$

$$\underline{I}'_1 = \underline{S}' \underline{i}_i = \begin{bmatrix} 3.560 \\ -2.40 \\ -1.169 \end{bmatrix}$$

$$\underline{e}'_2 = \underline{Z}_\alpha \underline{I}'_1 \quad \begin{bmatrix} 17.800 \\ -4.80 \\ -3.507 \end{bmatrix}$$

$$\therefore \underline{e}'_B = \underline{e}'_A + \underline{e}'_2$$

$$= \begin{bmatrix} 3.800 \\ 1.20 \\ 8.493 \end{bmatrix}$$

To calculate the branch flows we have for each cut segment

$$\underline{e}_i = \underline{A}_i \, \underline{e}'_{Bi} \qquad i = 1, 2, 3$$

where $\underline{A}_i$ is the incidence matrix for the cut segments.

$$\therefore \underline{e} = \begin{bmatrix} 3.8 \\ -1.2 \\ 8.493 \end{bmatrix}$$

adding the vector of cut segment branch pressure rises

$$\underline{V} = \begin{bmatrix} 3.8 + 4 \\ -1.2 + 2 \\ 8.49 + 0 \end{bmatrix} = \begin{bmatrix} 7.38 \\ 0.8 \\ 8.49 \end{bmatrix}.$$

$\therefore$ branch flows =

$$\underline{J} = \underline{Y}\,\underline{V} = \begin{bmatrix} 1.56 \\ 0.4 \\ 2.83 \end{bmatrix}$$

For the cut branches

$$\underline{e}_c = \underline{\tilde{S}}' \, \underline{e}'_B \qquad \therefore \underline{e}_c = \begin{bmatrix} 4.69 \\ -2.61 \end{bmatrix}$$

adding the cut branch pressure rises

$$\underline{V}_c = \begin{bmatrix} 4.69 + 0 \\ -2.61 + 5 \end{bmatrix} = \begin{bmatrix} 4.69 \\ 2.39 \end{bmatrix}$$

$$\therefore \underline{J}_c = \underline{Y}_c \underline{V}_c$$

$$= \begin{bmatrix} 1.17 \\ 2.39 \end{bmatrix}$$

These results can be seen to be in agreement with the previous calculation (appendix A.)

Appendix C.

Worked example branch addition to network

In this example it is proposed to add to the system a new branch running between nodes 2 and 3 with an admittance of ½. see fig. 7.

For this calculation it is only necessary to start at step (4) in the procedure outlined in Chapter 3 section H with a new $\underline{S}'$ matrix which now includes nodes of the new branch. Therefore $\underline{I}'_B$ and $\underline{e}'_A$ remain the same as in the previous calculation (Appendix B.)

As in the example of Appendix B

$$\underline{I}'_B = \begin{bmatrix} -2.8 \\ 3 \\ 4 \end{bmatrix}$$

$$\underline{Z}_A = \begin{bmatrix} 5 & & & & & \\ & 2 & & & & \\ & & 3 & & & \\ & & & 4 & & \\ & & & & 1 & \\ & & & & & 2 \end{bmatrix} \quad \begin{matrix} \updownarrow \alpha \text{ (rows 1–3)} \\ \updownarrow \beta \text{ (rows 4–6)} \end{matrix}$$

Note that $\underline{Z}_\alpha$ is unchanged but $\underline{Z}_\beta$ includes the new resistance term for the additional branch.

$$\therefore \underline{e}'_A = \begin{bmatrix} -14 \\ 6 \\ 12 \end{bmatrix}$$

Now however $$\underline{S}' = \begin{bmatrix} -1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & 1 \end{bmatrix}$$

which includes the new branch running between nodes 2 and 3

$$\underline{E}_1 = -\underline{\tilde{S}}' \, \underline{e}'_A = \begin{bmatrix} -26 \\ -20 \\ -6 \end{bmatrix}$$

$$\underline{E}_2 = \underline{E}_1 + \underline{E}'_B = \begin{bmatrix} -26 \\ -25 \\ -6 \end{bmatrix}$$

$$\underline{Y}_\beta = (\underline{\tilde{S}}' \, \underline{Z}_\alpha \, \underline{S}' + \underline{Z}_\beta)^{-1}$$

$$= \begin{bmatrix} 0.164 & -0.129 & -0.107 \\ -0.129 & 0.2365 & 0.123 \\ -0.107 & 0.123 & 0.224 \end{bmatrix}$$

$$\underline{i}_1 = \underline{Y}_\beta \, \underline{E}_2$$

$$= \begin{bmatrix} -0.388 \\ -3.290 \\ -1.631 \end{bmatrix}$$

$$\underline{I}'_1 = \underline{S}' \, \underline{i}_1$$

$$\begin{bmatrix} 3.628 \\ -1.659 \\ -2.019 \end{bmatrix}$$

$$\underline{e}'_2 = \underline{Z}_\alpha \underline{I}'_1$$

$$= \begin{bmatrix} 18.390 \\ -3.318 \\ -6.057 \end{bmatrix}$$

$\underline{e}'_B = \underline{e}'_A \, \underline{e}'_2$

$$= \begin{bmatrix} 4.390 \\ 2.682 \\ 5.943 \end{bmatrix}$$

which compares with classical analysis for the vector $\underline{e}'_B$

$$\underline{e}'_B = \begin{bmatrix} 4.392 \\ 2.682 \\ 5.946 \end{bmatrix}$$

Note that for any real system most of the computing time is taken up by the calculation of $\underline{e}'_A$ which remains the same when a new branch is added or one removed by the above method.

Appendix D.

Development of transformations between networks

Part 1

To establish the connection matrix between any two networks, for example networks A and B Fig (35), one starts by constructing the primitive transformation matrices $\underline{C}_{pA}^{-1}$ ($= \underline{\tilde{A}}_{pA}$) and $\underline{C}_{pB}$

$\underline{A}_{pA} =$

	e_1'	e_2'	e_3'	E_1'	E_2'	E_3'
1	-1	1	0	1	0	0
2	-1	0	0	0	1	0
3	0	0	1	0	0	0
4	-1	0	1	0	0	0
5	0	-1	1	0	0	0
6	0	-1	0	0	0	1

$\underline{C}_{pB} =$

	I_1'	I_2'	I_3'	I_4'	i_1'	i_2'
1	0	0	0	0	1	0
2	0	-1	0	0	-1	0
3	-1	0	0	0	1	1
4	0	0	0	0	0	-1
5	1	1	1	1	0	-1
6	-1	-1	-1	0	0	1

$$\underline{C}_{AB} = \underline{\tilde{A}}_{pA} \; \underline{C}_{pB}$$

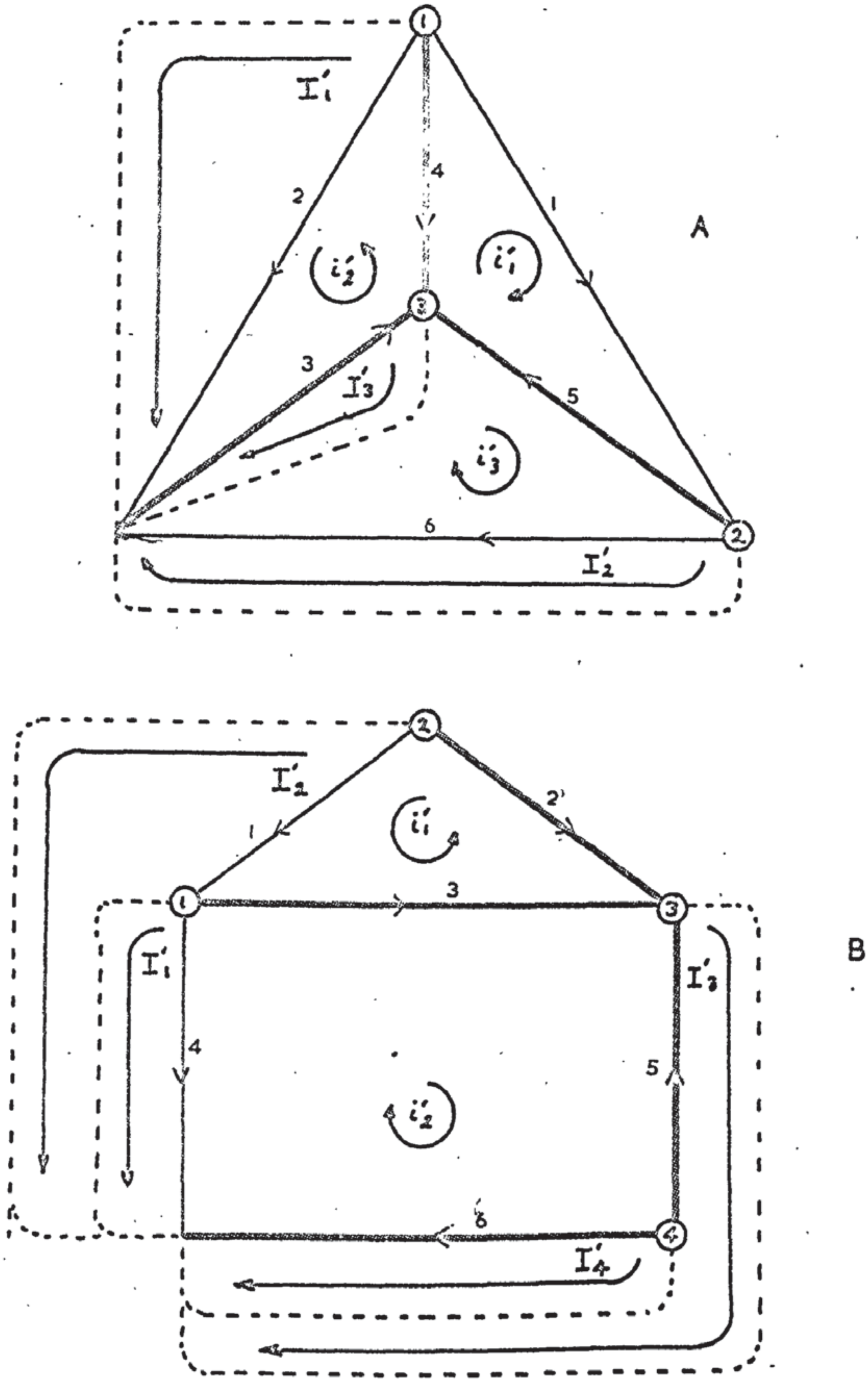


FIG 35

NETWORKS A AND B OF APPENDIX D

$$\therefore \underline{C}_{AB} = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & -1 & 1 & 0 \\ 3 & 0 & 1 & 1 & 1 & 1 & -1 \\ 4 & 0 & 0 & 0 & 0 & 1 & 0 \\ 5 & 0 & -1 & 0 & 0 & -1 & 0 \\ 6 & -1 & -1 & -1 & 0 & 0 & 1 \\ \hline \end{array},$$

$$\text{and } \underline{A}_{AB} = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 2 & -1 & 0 & 1 & -1 & 0 & 0 \\ 3 & -1 & 0 & 1 & 0 & 0 & 0 \\ 4 & -2 & 1 & 0 & -1 & -1 & -1 \\ 5 & 0 & -1 & 2 & 0 & 0 & 1 \\ 6 & -1 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array},$$

Now it can be seen by comparing these matrices with the diakoptics transformation matrices that the restriction of the same number of nodes in the two networks greatly simplifies the transformation of the matrices $\underline{Z}_B$.

For example partitioning the matrices as in the text.

$$\begin{bmatrix} \tilde{C}_1 & \tilde{C}_3 \\ \tilde{C}_2 & \tilde{C}_4 \end{bmatrix} \qquad \begin{bmatrix} Z_{1A} & Z_{2A} \\ Z_{3A} & Z_{4A} \end{bmatrix} \qquad \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix}$$

Then the transformed $\underline{Z}_{1B}$

$$= \underline{\tilde{C}}_1 \underline{Z}_{1A} \underline{C}_1 + \underline{\tilde{C}}_1 \underline{Z}_{2A} \underline{C}_3 + \underline{\tilde{C}}_3 \underline{Z}_{3A} \underline{C}_1 + \underline{\tilde{C}}_3 \underline{Z}_{4A} \underline{C}_4$$

For the diakoptics transformations which have the same number of nodes in each network $\underline{C}_1 = \underline{U}$ and $\underline{C}_3 = \underline{0}$

$\therefore \underline{Z}_{1A} = \underline{Z}_{1B}$

This however is not the case for a generalised transformation.

Part 2

Proof of $\underline{C}_1^{-1} = \tilde{\underline{A}}_1$ from the proposition of Power Invariance

Now from the proposition of power invariance for any two networks containing the same number of branches then

$$\tilde{\underline{V}}_A \; \underline{J}_A = \tilde{\underline{V}}_B \; \underline{J}_B$$

Now as shown above one can write that $\underline{J}_A = \underline{C}_{AB} \, \underline{J}_B$

and $\underline{V}_A = \underline{A}_{AB} \, \underline{V}_B$

$$\tilde{\underline{V}}_B \, \tilde{\underline{A}}_{AB} \, \underline{C}_{AB} \, \underline{J}_B = \tilde{\underline{V}}_B \; \underline{J}_B$$

$$\therefore \; \tilde{\underline{A}}_{AB} \; \underline{C}_{AB} = \underline{U}$$

$$\therefore \; \tilde{\underline{A}}_{AB} = \underline{C}_{AB}^{-1}$$

Appendix E

Table of Results

	Case 1	Case 2	Case 3
	11	31	50
Number of iterations	3	3	never converged
in inner cycle	1	1	———
Time	30min	1hr 16min	2hrs (Stopped)

Table 2

Table 3

Comparison of Results for Hardy Cross Solution

Segment 1

BRANCH NO Hardy Cross Case 1	BRANCH NO Hardy Cross Case 2	BRANCH NO Diakoptics Case 1	Flow ft^3/min. HC 1	Flow ft^3/min. HC 2	Flow ft^3/min. Diakoptics (Case 1)
2	19	1	88.56	88.98	88.34
3	20	2	50.94	51.18	50.88
21	3	3	31.56	31.74	31.47
20	21	4	37.62	37.8	37.56
19	2	5	83.88	84.78	83.93
18	18	6	45.78	46.20	45.56
1	1	7	14.4	15.24	14.98
38	16	8	48.3	46.38	48.53
11	38	9	62.7	61.56	62.46
37	17	10	69.48	70.56	69.48
35	15	11	38.58	41.58	38.70
36	13	12	48.78	49.38	48.76
34	36	13	30.9	28.98	30.86
13	37	14	77.82	77.88	77.96
12	14	15	53.88	53.52	54.45
16	34	16	2.874	1.356	1.739
33	35	17	54.84	53.34	54.38

Table 3 Part 2

Segment 2 Cut Branch Set

5	5	1	5.429	6.354	5.525
23	23	2	47.04	47.39	46.91
24	24	3	46.52	46.75	46.33
25	6	4	41.09	40.40	40.83
6	25	5	64.44	63.84	64.09
7	26	6	84.84	84.54	84.50
26	7	7	73.20	73.50	72.72
27	27	8	39.93	38.23	38.16
8	28	9	122.8	122.8	122.5
29	29	10	117.2	117.2	117.2
28	8	11	50.60	50.06	50.32
31	9	12	2.063	1.942	1.796
30	10	13	17.40	16.76	17.51
9	11	14	17.56	17.92	17.69
14	30	15	66.60	67.2	66.96
32	31	16	40.38	40.14	40.56
10	32	17	34.96	34.69	35.17
17	33	18	142.0	142.0	142.4
4	22	1	82.5	82.92	82.25
22	4	2	90.00	90.84	89.98
15	12	3	105.5	104.2	104.8

Branch Number	Length in Feet	Diameter in Feet
1	16,840	0.67108
2	5,280	0.67108
3	10,560	0.67108
4	5,280	0.51042
5	10,560	0.67708
6	21,120	1.02083
7	31,680	0.854167
8	42,240	0.854167
9	10,560	0.34375
10	5,280	1.02083
1	10,560	0.67708
2	5,280	0.51042
3	21,120	0.67708
4	31,680	1.28125
5	10,560	1.02083
6	26,400	0.34375
7	5,280	0.51042
8	10,560	1.02083
9	21,120	0.854167
10	26,400	1.02083
11	47,520	0.51042
12	36,960	0.67708
13	21,120	1.02083
14	7,920	0.34375
15	10,560	1.02083

Table 4 Part 1

Branch Number	Length in feet	Diameter in feet
1	31,680	0.51042
2	29,040	1.02083
3	21,120	1.02083
4	42,240	1.28125
5	26,400	0.51042
6	15,840	0.66708
7	10,560	0.34375

Table 4

Branch dimension for network due to Knights and Allen.

Node Number	Demand ft^3/min
1	-166.67
2	-250
3	0
4	-250
5	0
6	-500
7	-333.3
8	0
9	0
1	-500
2	-500
3	-1666.7
4	0

Table 5

Nodal demands of the Network due to Knights and Allen

Node Number	Demand ft^3/min
5	-833.4
6	3333
7	0
8	-500
9	-666.7
10	1333.3

Table 5 continued

32

Branch Number	Diakoptics Flow ft^{3}m^{-1}	Knights Allen Flow ft^3/m^{-1}	Absolute Difference	Percentage Difference
1	266.10000	267.30000	1.2000000	.45095828
2	121.54000	121.23333	.30667000	.25232022
3	259.18000	258.41666	.76334000	.29452118
4	150.58000	151.35000	.77000000	.51135608
5	108.36000	105.53333	2.8266700	2.6085917
6	374.45000	388.25000	13.800000	3.6854052
7	357.75000	375.40000	.35000000	.97833682+ 1-
8	279.00000	278.76666	.23334000	.83634408+ 1-
9	87.880000	93.716666	5.8366660	6.6416317
10	128.90000	174.76666	45.866660	35.583134
1	224.92000	237.56666	12.646660	5.6227369
2	111.97000	113.56666	1.5966600	1.4259712
3	122.49000	122.56666	.76660000+ 1-	.62584700
4	1455.0000	1456.4333	1.4333000	.98508591+ 1-
5	502.43000	459.00000	43.430000	8.6439902

Table 6 Part 1

Branch Number	Diakoptics Flow ft^3m^{-1}	Knights Allen Flow $ft3/m^{-1}$	Absolute Difference	Percentage Difference
6	43.510000	38.100000	5.4100000	12.433923
7	222.35000	227.75000	5.4000000	2.4286035
8	996.80000	998.70000	1.9000000	.19060995
9	110.58000	107.21666	3.3633400	3.0415445
10	856.06000	851.21666	4.8433400	.56577109
11	78.710000	77.416666	1.29	1.643
12	315.54000	313.45000	2.0900000	.66235659
13	496.87000	500.30000	3.4300000	.69032141
14	105.74000	109.06666	3.3266600	3.1460752
15	1079.8600	1094.1666	14.306600	1.3248569
1	41.140000	79.900000	38.760000	94.214876
2	352.86000	355.98333	3.1233300	.88514708
3	391.60000	389.78333	1.8166700	.46390960
4	690.00000	669.21666	20.783340	3.0120782
5	35.140000	37.583333	2.4433330	6.9531388
6	47.610000	50.500000	2.8900000	6.0701533
7	16.680000	16.083333	.59666700	3.5771402

Table 6

Comparison of results for the branch flows of network due to Knights and Allen.

Node Number	Diakoptics Pressure lb/ft^2	Knights Allen Pressure lb/ft^2	Absolute Difference	Percentage Difference
1	-165.70000	-153.53520	12.16480	7.341604
2	-158.90000	-148.34820	10.551800	6.6405286
3	-100.50000	-94.922100	5.5779000	5.5501492
4	-105.00000	-98.553000	6.4470000	6.1400000
5	-139.80000	-130.19370	9.6063000	6.8714592
6	-146.90000	-135.89940	11.000600	7.4884955
7	-137.50000	-127.60020	9.8998000	7.1998545
8	-.66000000	-1.0374000	.37740000	57.181818
9	-87.000000	-80.917200	6.0828000	6.9917241
10	-64.800000	-59.650500	5.1495000	7.9467592
11	-15.500000	-42.014700	26.514700	171.06258
12	-109.20000	-102.18390	7.0161000	6.4250000
13	-91.500000	-86.104200	5.3958000	5.8970491
14	-106.50000	-99.590400	6.9096000	6.4878873
15	-50.830000	-48.239100	2.5909000	5.0971867
16	-81.100000	-87.141600	6.0416000	7.4495684
17	-226.30000	-205.92390	20.376100	9.0040212
18	-155.30000	-144.71730	10.582700	6.8143593
19	-160.10000	-148.86690	7.0163023	7.0163023

Table 7.

Comparison of nodal Pressures of network due to Knights and Allen.

Branch Number	Diakoptics Flow ft^3/m^{-1}	Knights Allen Flow ft^3/m^{-1}	Absolute Difference	Percentage Difference
1	250.31000	250.60000	.29000000	.11585633
2	512.75000	508.58333	4.1666700	.81261238
3	308.40000	312.28333	3.8833300	1.2591861
4	167.20000	168.00000	.80000000	.47846889
5	142.10000	113.95000	28.150000	19.809992
6	44.640000	54.150000	9.5100000	21.303763
7	358.20000	357.65000	.55000000	.15354550
8	234.50000	229.55000	4.9500000	2.1108742
9	98.630000	104.80000	6.1700000	6.2557031
10	139.27000	185.66666	46.396660	33.314181
1	43.120000	34.450000	8.6700000	20.106679
2	236.00000	233.95000	2.0500000	.86864406
3	192.60000	198.73333	6.1333300	3.1844911
4	1421.9000	1431.8166	9.9166000	.69741894
5	500.73000	458.03333	42.696670	8.5268847
6	42.340000	37.050000	5.2900000	12.494095
7	182.96000	186.91666	3.9566600	2.1625819
8	1243.0000	1222.5000	20.500000	1.6492357
9	151.14000	146.48333	4.6566700	3.0810308
10	894.58000	899.91666	5.3366600	.59655480
11	81.860000	80.333333	1.5266670	1.8649731
12	329.00000	327.93333	1.0666700	.32421580
13	849.52000	853.18333	3.6633300	.43122351
14	90.000000	91.616666	1.6166660	1.7962955
15	1087.8000	1109.0833	21.283300	1.9565453

Table 8 Part 1

Branch Number	Diakoptics Flow ft^3/m^{-1}	Knights Allen Flow ft^3/m^{-1}	Absolute Difference	Percentage Difference
1	40.830000	79.616666	38.786666	94.995508
2	645.97000	64285000	3.1200000	.48299456
3	774.60000	760.20000	14.400000	1.8590240

Table 8

Comparison of branch flows of network due to Knights and Allen with cut branches 4,5,6 and 7 removed.

Node Number	Diakoptics Pressure lb/ft^2	Knights Allen Pressure lb/ft^2	Absolute Difference	Percentage Difference
1	-213.40000	-198.14340	15.256600	7.1492970
2	-123.40000	-117.22620	6.1738000	5.0030794
3	-71.440000	-70.543200	.89680000	1.2553191
4	-134.00000	-125.00670	8.9933000	6.7114179
5	-182.00000	-169.61490	12.385100	6.8050000
6	-193.50000	-180.50760	12.992400	6.7144186
7	-170.60000	-157.16610	13.433900	7.8745017
8	-.75700000	-1.0374000	.28040000	37.040951
9	-133.50000	-124.49000	9.0520000	
10	-65.700000	-61.206600	4.4934000	6.8392694
11	-15.370000	-41.496000	26.126000	169.98048
12	-104.50000	-98.553000	5.9470000	5.6909090
13	-64.330000	-60.687900	3.6421000	5.6615886

Node Number	Diakoptics Pressure lb/ft^2	Knights Allen Pressure lb/ft^2	Absolute Difference	Percentage Difference
14	- 4.4200000	-10.892700	6.4727000	146.44117
15	- 80.230000	-64.837500	15.392500	19.185466
16	- 2.1600000	-9.3366000	7.1766000	332.25000
17	- 109.50000	-99.590400	9.9096000	9.0498630
18	- 33.280000	-36.827700	3.5477000	10.660156
19	- 41.590000	-43.570800	1.9808000	4.7626833

Table 9

Comparison of Nodal pressures of Networks due to Knights and Allen with cut branches 4.5.6 and 7 removed.

Segment 1

Branch Number	Length ft	Diameter ft
1	3900	1
2	8800	1
3	2100	1
4	3300	1
5	4000	1
6	3000	1
7	4500	1
8	2000	0.5
9	2000	1
10	1200	1
11	2600	1.333
12	2500	0.667
13	14100	1.333
14	1200	0.667
15	5300	0.833
16	8000	1.333
17	2200	1

Table 10 Part 1.

Segment 2

Branch Number	Length ft	Diameter ft
1	1000	1.333
2	3000	1.667
3	4100	1
4	5000	1
5	1500	1
6	1000	0.667
7	5000	0.833
8	2500	0.833
9	2000	0.5
10	1000	0.5
11	1600	0.667
12	1500	0.667
13	2200	0.833
14	2000	1
	Cut Branches	
1	3300	1.167
2	9300	1
3	4500	0.667
4	3400	1

Table 10

Dimensions of Network Due to Ingels and Powers

Node Number	Demand ft^3m^{-1}
1	-2.083
2	-2.083
3	206.25
4	-8.333
5	-6.25
6	0
7	0
8	-12.5
9	0
10	-4.167
11	-2.083
12	-4.167
13	0

Table 11 Part 1

Node Number	Demand ft^3m^{-1}
1	-2.0833
2	-93.75
3	-20.3
4	-6.25
5	0
6	0
7	-14.58
8	0
9	-2.0833
10	-6.25
11	0
12	-12.5
13	0
14	0
15	-4.167

Table 11

Nodal Demands for Network Due to Ingels and Powers.

	Flow ft^3/min		Percentage Difference; on Diakoptics		
	Dolan	Ingels & Powers	Dolan	Ingels & Powers	Reynolds Number
1	54.487178	58.974357	7.2968932	.3374958	115,764
2	56.570511	61.057690	7.0252099	.34956035	119,839
3	39.262819	34.775640	13.230912	.29023792	68,297
4	60.096152	55.608972	9.565684	1.3670901	108,049
5	8.1730767	4.3269229	60.665946	14.941558	23,548
6	18.696580	8.3333331	171.87116	21.176866	13,544
7	4.0598289	1.9230768	54.424834	26.851395	13,544
8	4.0598289	1.9230768	54.483595	26.823561	10,365
9	10.309828	8.1730767	16.127821	7.9401137	5,176
10	10.256409	1.2553418	404.24823	38.282114	17,485
11	10.256409	1.2553418	404.24823	38.282114	3,004
12	20.833332	17.334401	31.948394	9.7878333	6,008
13	14.529914	9.9091877	65.923421	13.157333	23,325
14	8.2264955	3.8461537	42.154752	33.538038	25,870
15	35.363246	27.243589	44.227929	11.112153	13,787
16	47.756408	35.256409	38.729979	2.4181065	36,220
1	116.82691	129.32691	9.2853127	.42078658	190,224
2	118.91025	131.41025	9.1698812	.37829889	154,704
3	87.339741	74.839741	17.577024	.74948642	146,306
4	97.006408	66.506408	19.686730	.75049309	130,020
5	72.756408	60.256408	21.647925	.74806132	117,800
6	15.544871	13.221153	14.099170	2.9568922	40,248
7	15.544871	13.221153	14.099170	2.9568922	32,201

8	3.0448717	.24038460	169.69634	78.708184	2,666
9	5.2350425	3.0982905	23.906331	26.667680	16,648
10	9.4017o91	7.2649570	12.031805	13.429969	33,057
11	8.2799143	2.8579089	147.01414	14.740277	15,811
12	6.1965810	.77457262	89.729975	76.283753	9,648
13	2.0299144	3.3920939	125.05342	226.07613	2,131
14	49.839742	37.339742	34.585607	.83101641	72,937
1	114.74358	127.24358	9.8565637	.36467907+ 1-	214,893
2	51.923075	59.909186	13.262044	.78824629+ 1-	117,903
3	8.2264955	3.8461537	42.548873	33.353774	17,048
4	47.809827	40.731836	25.815334	7.1890421	78,844

Table 12

Comparison of Results of Diakoptics Program with Ingels and Powers

Branch Number	Length ft^2	Diameter ft
1	1	1
2	1	1
3	1	1
4	1000	1
5	5200	1
6	10000	1
7	8850	0.833
8	7600	0.833
9	2000	1
10	1000	0.833
11	1000	0.5
12	1000	0.5
13	7700	0.833
14	1000	0.5
15	300	0.5
16	5000	0.833
17	1000	0.833
18	1000	0.833
1	1000	0.5
2	1000	0.5
3	1000	0.5
4	1000	0.5
5	1000	0.5

Table 13

Dimensions of Network Due to Hunn and Ralph

Branch Number	Lenght ft^2	Diameter ft
6	3000	1.667
7	33200	1
8	1	1

Node Number	Demand ft^3m^{-1}
1	-1.6026
2	-1.6026
3	-1.6026
4	0
5	0
6	-168.3
7	-80.13
8	-224.4
9	-64.1
10	-8.11
11	-51.28
12	-97.76
13	0

Table 14

Nodal Demands /10

due to Hunn and Ralph.

Case Number	Number of Segments	Dimensions of Segments		Number of cut Branches
		Node	Branch	
1	2	10	17	3
		11	18	
2	2	10	15	8
		11	15	
3	3	7	10	7
		7	10	
		7	11	
4	4	5	7	10
		6	9	
		6	7	
		4	5	

Table 15

Basic dimensions of the four cutting patterns of test network fig 16

Case	Time per iteration	Number of Iterations	Total Time
1	2m 56sec	8	23min 12sec
2	3m 31sec	8	28min 8sec
3	2m 10sec	8	17min 20sec
4	3m 40sec	8	29min 20sec

Table 16

Computation time required for the four cases of different cutting patterns

Initial Branch FLOWS ft^3/min	Number of Iterations
1	9
50	8
2500	10

Table 17

Number of iterations for case 1 with different initial guess of branch Flows.

Iteration Number	Pressure Drop lb/ft^3				
	Pipe No.18 Segment 2	1 1	11 2	13 1	1 2
1	176	96	40	30	4.85
2	261	121	47	26	2
3	309	135	52	24	1.37
4	333	142	53	23	1.17
5	344	145	53	22.5	1.1
6	350	147	54	23	1.05
7	352	148	54	22	1.04
8	353	148	54	22	1.04

Table 18

Pressure drop on iteration for selected pipes of case 1

Case	Number of Iterations	Percentage changes from Initial to Final Pressure
a	3	6.3
b	3	6.3
c	6	42
d	8	106
e	5	62
f	8	179
g	4	25

Table 19

Number of iterations and percentage change in final pressure vector for cases in Chapter 4 Section D

Case	Number of Iterations
a	6
b	6
c	5
d	8
e	2
f	2
g	2

Table 20

Number of iterations needed for convergence for the cases Chapter 4 Section E.

Appendix F

Detailed Results of Networks analysed summmarised in Results Section

RESULTS FROM NETWORK DUE TO KNIGHTS AND ALLEN

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	266.09563	3	2	4.5723515
2	121.53638	3	4	27.976159
3	259.17974	4	5	7.4576697
4	150.58021	5	1	5.8207952
5	108.35982	5	6	15.338742
6	374.45418	9	4	20.788717
7	357.74615	9	6	5.9793499
8	279.00187	9	7	5.5264459
9	87.884637	8	7	.64220946
10	128.90296	10	8	195.35405

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	224.92053	1	4	8.4129924
2	111.97216	4	5	7.4731638
3	122.49107	4	3	6.9500175
4	1454.9887	11	3	13.328303
5	502.42588	11	2	32.479422
6	43.512889	2	3	.46440355
7	222.35467	5	10	4.1526080
8	996.79715	6	5	17.897563
9	110.58448	9	10	23.286506
10	856.06176	6	9	8.1926451
11	78.705881	9	8	1.1094159
12	315.54479	6	8	1.7986419
13	496.86710	6	7	16.393757
14	105.73602	7	8	.72857723
15	1079.8608	11	1	16.663008

CUT PIPE RESULTS

1	41.136926	8	11	.35999797
2	352.85690	10	9	.06296157
3	391.60129	16	3	.04992997
4	689.98592	15	9	.05245449
5	35.140910	7	6	.26594112
6	47.606207	14	12	.05548303
7	16.679526	2	1	.40591178

NODE NO	PRES
1	-165.65747
2	-158.88705
3	-100.69038
4	-105.03467
5	-139.78812
6	-146.85257
7	-137.50716
8	-.65984276
9	-87.022293
10	-64.805871
11	-15.469053
12	-109.16534
13	-91.540773
14	-106.52401
15	-50.829433
16	-81.137745
17	-226.26446
18	-155.32093
19	-160.06980

RESULTS OF NETWORK DUE TO KNIGHTS AND ALLEN WITH BRANCH REMOVED

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	250.31484	3	2	4.8175610
2	512.74945	3	4	8.2019038
3	308.39822	4	5	6.4214895
4	167.20144	5	1	5.3212778
5	142.09784	5	6	12.321665
6	44.637310	9	4	107.72590
7	358.20662	9	6	5.9728301
8	235.45843	9	7	6.3595240
9	98.630041	8	7	.58082219
10	139.26953	10	8	183.99927

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	43.118476	4	1	31.273398
2	236.03918	5	4	3.9397243
3	192.60796	4	3	4.7922548
4	1421.8739	11	3	13.603203
5	500.72792	11	2	32.572885
6	42.338255	2	3	.47489754
7	182.96304	5	10	4.9230673
8	1242.9695	6	5	14.682716
9	151.14192	9	10	18.207223
10	894.57715	6	9	7.8804515
11	81.864981	9	8	1.0745430
12	329.02871	6	8	1.7344287
13	849.52421	6	7	10.310226
14	90.004540	7	8	.83874881
15	1087.8017	11	1	16.554033

CUT PIPE RESULTS

1	40.832161	8	11	.35794420
2	645.97117	10	9	.10500663
3	774.61386	16	3	.08943822

NODE NO	PRES
1	-213.40506
2	-123.40073
3	-71.441894
4	-133.95780
5	-181.98376
6	-193.51612
7	-170.56798
8	-.75690261
9	-133.54344
10	-65.712185
11	-15.372538
12	-104.52494
13	-64.333427
14	-4.4208107
15	80.234472
16	-2.1618061
17	-109.46991
18	-33.284044
19	-41.585251

RESULTS OF NETWORK DUE TO INGELS AND POWERS

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	58.776531	1	2	.48449708
2	60.845106	4	1	.20810115
3	34.675846	3	2	1.4382313
4	54.859282	5	3	.60935403
5	5.0869257	5	6	3.6164019
6	11.956782	6	7	2.4637829
7	6.8769483	9	6	2.5478145
8	2.6293765	8	7	.42446489
9	2.6282882	+ 10	8	11.845083
10	8.8782226	9	10	7.8172815
11	2.0344819	11	9	42.678128
12	2.0345380	12	11	1.6553448
13	15.789322	13	9	1.6961420
14	8.7572988	13	12	1.0902713
15	5.7877712	15	14	1.0254053
16	24.519583	15	13	2.0933785
17	34.424550	16	15	1.3815991

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	128.78519	2	1	3.9591900
2	130.91579	3	2	4.0111152
3	74.283129	3	4	.37235371
4	66.014374	4	5	.34010559
5	59.809583	5	6	1.2399901
6	13.624262	6	7	.89956608
7	13.624127	7	8	.54325257
8	1.1285044	8	9	7.3177409
9	4.2253364	10	9	.28988617
10	8.3924300	6	10	.32525588
11	5.3523550	9	11	1.2230816
12	3.2667276	11	12	1.9249109
13	.90197271	13	12	9.7174468
14	37.032115	13	14	1.4257303

CUT PIPE RESULTS

1	127.29116	16	4	1.5450314
2	59.862878	4	5	5.0040734
3	5.7712434	14	12	2.4432810
4	38.004744	21	28	1.2198155

NODE NO	PRES
1	-40.054997
2	-161.36952
3	-137.25945
4	252.32737
5	-47.230869
6	-48.637496
7	-53.490513
8	-47.295946
9	-45.938340
10	-47.074057
11	-45.890669
12	-44.661597
13	-36.629378
14	-30.560828
15	-24.916453
16	448.99620
17	481.52437
18	514.16262
19	314.66648
20	120.56683
21	72.332912
22	57.187542
23	32.108734
24	31.954519
25	46.53000366
26	27.578397
27	25.881317
28	25.974137

RESULTS FOR NETWORK DUE TO HUNN AND RALPH

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	247.60545	1	14	545.63214
2	178.28531	2	14	550.18973
3	374.05206	3	14	541.65872
4	372.44990	3	4	.50412889
5	94.307992	13	4	.30157665
6	82.378645	13	5	.17492474
7	151.17986	5	6	.04666365
8	94.280017	7	6	.08156900
9	466.75154	4	7	.20816977
10	364.37624	7	8	.21104647
11	48.573944	8	12	.08860382
12	49.222246	9	12	.08759711
13	177.20323	5	9	.04667214
14	77.469595	6	11	.05893379
15	63.988942	9	10	.23246669
16	73.649745	10	11	.15271485
17	246.00341	1	5	.26862527
18	176.68324	2	13	.38818447

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	6.1575087	2.	1	.47691017
2	74.884505	5	1	.06073024
3	54.254778	3	2	.08052558
4	1.3578452	4	3	1.4741856
5	49.355176	6	4	.08739363
6	105.44574	6	5	5.4724349
7	154.79996	7	6	.03153566
8	156.40201	7	8	550.73917

CUT PIPE RESULTS

1	17.760859	18	10	4.1313996
2	99.922432	11	14	6.7794672
3	91.422190	8	14	15.546006

NODE NO	PRES
1	9035.5539
2	9035.6788
3	9035.3261
4	8277.5802
5	8119.3591
6	4879.5811
7	6035.4125
8	4308.8911
9	4322.5924
10	4047.3317
11	3565.0620
12	3760.6761
13	8590.2967
14	2887.6412
15	2900.5524
16	3574.3107
17	3575.2318
18	4120.7089
19	4139.9775
20	9035.7182

RESULTS OF TEST NETWORK

CASE 2

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	88.340671	1	2	.59540098
2	50.879142	2	3	.93011736
3	37.557927	2	4	1.1862067
4	83.927488	5	4	.62071843
5	45.557914	1	5	1.0163801
6	13.982792	6	1	2.5862290
7	48.529216	6	5	.96613370
8	62.465935	7	6	.78841815
9	69.482019	7	8	.72342451
10	38.699238	8	5	1.1581815
11	30.856396	8	9	1.3872552
12	77.956335	7	9	.65903421
13	54.450486	9	10	.88072022
14	54.382093	9	11	.88161172
15	1.7393765	11	10	12.415630

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	5.5208409	1	3	5.2797956
2	46.906810	1	2	.99286722
3	46.327903	3	2	1.0028089
4	64.085557	4	2	.77228779
5	84.490734	2	6	.61735541
6	38.160929	5	6	1.1712144
7	122.52604	6	7	.45598299
8	117.15124	8	7	.47301909
9	50.320256	5	8	.93841130
10	1.7965207	5	10	12.128747
11	17.514209	9	5	2.1690743
12	17.693620	9	10	2.1518202
13	40.556163	11	10	1.1155919
14	66.958246	11	8	.74539979
15	142.44276	12	11	.40305625

CUT PIPE RESULTS

1	82.250960
2	31.467466
3	89.978064
4	48.755769
5	40.821596
6	104.77824
7	72.712076
8	35.167269

NODE NO	PRES
1	95.337531
2	-53.034197
3	-107.73605
4	-84.696406
5	50.513833
6	100.74416
7	179.97361
8	83.927626
9	61.684857
10	-.14009571
11	-238.09190
12	-285.33568
13	-239.13755
14	-202.35424
15	-389.61247
16	-422.19483
17	-690.90232
18	-443.23528
19	-381.53796
20	-389.76059
21	-353.40665

3	11	1.5848550
4	3	.73217344
4	11	1.7048099
5	10	1.0389320
14	13	.90107477
10	14	1.9299249
12	15	1.4341055
21	19	.79992885

TEST NETWORK CASE 3

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	88.338383	1	2	.59541351
2	13.981369	4	1	2.5864340
3	45.559125	1	3	1.0163584
4	48.529843	4	3	.96612369
5	62.465444	5	4	.78842314
6	69.482087	5	6	.72342393
7	38.699159	6	3	1.1581834
8	30.856654	6	7	1.3872460
9	77.956518	5	7	.65903295
10	54.376400	7	8	.88168604

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	82.272474	1	2	.63083859
2	31.472853	3	1	1.3656107
3	90.000338	3	2	.58645761
4	5.5267836	2	5	5.2754998
5	46.906282	2	6	.99287618
6	46.326270	5	6	1.0028373
7	64.084469	7	6	.77229839
8	40.821836	7	5	1.1097806
9	104.79274	4	7	.51809638
10	1.7415827	8	4	12.404262

CUT SEGMENT NO 3

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	122.52662	1	3	.45598124
2	38.155166	2	1	1.1713557
3	50.317945	2	4	.93844595
4	117.15069	4	3	.47302090
5	1.7937611	2	6	12.142248
6	17.513312	5	2	2.1691615
7	17.692257	5	6	2.1519501
8	35.172224	7	5	1.2499708
9	40.559877	7	6	1.1155101
10	66.959048	7	4	.74539258
11	142.44566	8	7	.40304952

CUT PIPE RESULTS

1	50.871750	2	8	1.0750076
2	37.543797	2	10	.84277012
3	83.917029	3	10	1.6108734
4	48.749953	3	11	1.0388325
5	54.444947	7	11	1.1353414
6	84.477804	13	15	1.6196111
7	72.700862	13	16	1.4339263

NODE NO	PRES
1	95.328478
2	-53.036282
3	50.502632
4	100.73413
5	179.96246
6	83.916301
7	61.673200
8	-107.72380
9	-238.14142
10	-84.677072
11	-.14040196
12	-239.18906
13	-285.38426
14	-202.40536
15	-422.20545
16	-389.63194
17	-690.91524
18	-443.25031
19	-381.55817
20	-389.77967
21	-353.41973

TEST NETWORK CASE 4

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	88.315062	1	2	.59554133
2	45.566156	1	4	1.0162327
3	37.650685	2	5	1.1838728
4	51.017152	2	3	.92809424
5	31.564102	5	3	1.3624705
6	83.943992	4	5	.62061931
7	48.766266	4	6	.96236050

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	5.5180027	1	2	5.2818503
2	46.901105	1	4	.99296413
3	46.322665	2	4	1.0028999
4	40.826740	3	2	1.1096740
5	64.085100	3	4	.77229224
6	84.493230	4	6	.61734058
7	72.711291	4	5	.69730488
8	38.167029	5	6	1.1710649
9	104.79860	7	3	.51807276

CUT SEGMENT NO 3

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	117.14217	2	1	.47304901
2	66.953397	3	2	.74544344
3	40.566899	3	4	1.1153557
4	17.693620	5	4	2.1518202
5	35.179317	3	5	1.2497699
6	142.45380	6	3	.40303062
7	1.7362557	6	7	12.431753

CUT SEGMENT NO 4

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	62.460217	2	1	.78847639
2	69.483779	2	3	.72340968
3	77.960207	2	4	.65900766
4	30.861219	3	4	1.3870829
5	54.448529	4	5	.88074569

CUT PIPE RESULTS

1	13.961631	18	1	.38620762
2	48.529252	18	4	1.0350540
3	38.688010	20	4	.86322234
4	89.994916	5	6	1.7050695
5	82.224814	3	6	1.5844460
6	54.380347	21	17	1.1342568
7	17.514949	16	10	.46104142
8	1.7922357	10	15	.08230638
9	50.304803	10	13	1.0653678
10	122.50751	11	12	2.1927929

NODE NO	PRES
1	95.511905
2	-52.781851
3	-107.75165
4	50.673595
5	-84.584833
6	-238.03242
7	-239.07713
8	-202.28548
9	-285.26586
10	-389.54061
11	-422.13233
12	-690.76594
13	-443.13372
14	-353.31685
15	-389.68812
16	-381.46549
17	.13966298
18	100.90399
19	180.12034
20	84.069950
21	61.820943

RESULTS FOR CHAPTER 5 SECTION D

EXAMPLE a)

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	90.031449	1	2	.58629280
2	68.948573	2	3	.72794940
3	20.763684	2	4	1.8977160
4	79.122556	5	4	.65114431
5	44.513228	1	5	1.0354592
6	14.554719	6	1	2.5066978
7	47.755191	6	5	.97868607
8	62.306473	7	6	.79004674
9	69.393865	7	8	.72416797
10	37.670999	8	5	1.1833630
11	31.728619	8	9	1.3568494
12	78.290884	7	9	.65674933
13	55.026412	9	10	.87329132
14	54.994294	9	11	.87370194
15	1.1252565	11	10	16.953043

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	3.2626071	1	3	7.8313086
2	45.872253	1	2	1.0107908
3	45.638082	3	2	1.0149481
4	64.625686	4	2	.76706933
5	84.083761	2	6	.61978131
6	38.500873	5	6	1.1629464
7	122.57273	6	7	.45584092
8	117.39788	8	7	.47220664
9	50.196127	5	8	.94027572
10	1.1093897	5	10	17.124174
11	17.770793	9	5	2.1444925
12	17.848542	9	10	2.1371666
13	41.034311	11	10	1.1051812
14	67.213430	11	8	.74311084
15	143.83798	12	11	.39984618

CUT PIPE RESULTS

1	70.283936
2	98.822647
3	50.806246
4	42.340717
5	106.95883
6	72.048346
7	35.614477

3	11	1.3952099
4	11	1.8400195
5	10	1.0738948
14	13	.92781128
10	14	1.9626293
12	15	1.4234956
21	19	.80802924

NODE NO	PRES
1	97.483066
2	-56.077492
3	-150.79365
4	-67.018899
5	54.494188
6	103.28940
7	182.15368
8	86.328036
9	62.943998
10	-.06637490
11	-248.85449
12	-294.23703
13	-249.27110
14	-209.98691
15	-396.79754
16	-429.90386
17	-698.79750
18	-450.18201
19	-388.51083
20	-396.86232
21	-359.73329

EXAMPLE b)

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	89.222830	1	2	.59061017
2	51.379247	2	3	.92283222
3	37.844174	2	4	1.1790369
4	85.182043	5	4	.61328226
5	45.245346	1	5	1.0220063
6	14.469764	6	1	2.5181629
7	48.414933	6	5	.96796452
8	62.885541	7	6	.78416793
9	69.461363	7	8	.72359857
10	39.213516	8	5	1.1460278
11	30.245109	8	9	1.4094879
12	77.651547	7	9	.66113116
13	54.189964	9	10	.88412693
14	53.714625	9	11	.89041856
15	5.2801385	11	10	5.4607649

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	6.0694131	1	3	4.9140468
2	48.332467	1	2	.96929044
3	47.664697	3	2	.98017731
4	65.640484	4	2	.75747270
5	85.804606	2	6	.60966455
6	35.519763	5	6	1.2402098
7	121.33884	6	7	.45962904
8	118.66046	8	7	.46809533
9	42.252381	5	8	1.0796090
10	.90461179	10	5	19.766488
11	.60028774	9	5	26.244468
12	.60058515	10	9	26.235593
13	60.469776	11	10	.80935696
14	76.344036	11	8	.67029679
15	138.42124	12	11	.41263433

CUT PIPE RESULTS

1	83.286667
2	31.875000
3	91.113399
4	47.612881
5	41.583635
6	107.11901
7	75.677027

3	11	1.6010395
4	3	.73970449
4	11	1.7222860
5	10	1.0193332
14	13	.91450831
10	14	1.9650275
12	15	1.4813024

NODE NO	PRES
1	91.837570
2	-59.231331
3	-114.90694
4	-91.328861
5	47.566468
6	97.583728
7	177.77770
8	81.783366
9	60.325141
10	-.96692288
11	-248.25219
12	-298.11595
13	-249.48730
14	-211.45872
15	-410.21651
16	-438.85664
17	-702.84962
18	-449.35326
19	-410.19364
20	-410.17075
21	-335.45739

EXAMPLE c)

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	77.413905	1	2	.66277640
2	116.20164	2	3	.47617601
3	38.981292	4	2	1.1514806
4	39.156123	5	4	1.1473700
5	47.769954	1	5	.97844329
6	5.1675886	6	1	5.5503412
7	48.273128	6	5	.97024699
8	53.457242	7	6	.89386694
9	69.709325	7	8	.72151524
10	22.809014	8	5	1.7623021
11	46.873009	8	9	.99344168
12	86.827840	7	9	.60381970
13	66.469446	9	10	.74982808
14	67.234254	9	11	.74292475
15	7.6410693	10	11	4.1232560

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	26.504580	3	1	1.5650752
2	23.161442	1	2	1.7411088
3	36.608646	3	2	1.2106948
4	75.296126	4	2	.67784803
5	76.297573	2	6	.67062766
6	44.366643	5	6	1.0382002
7	120.66613	6	7	.46172366
8	119.33352	8	7	.46593567
9	47.298206	5	8	.98626853
10	9.6495568	10	5	3.4470086
11	23.183921	9	5	1.7397763
12	20.319172	9	10	1.9303049
13	49.347720	11	10	.95324219
14	72.034334	11	8	.70260665
15	164.87575	12	11	.35743479

CUT PIPE RESULTS

1	116.25435	3	11	2.1008423
2	79.614854	5	10	1.5435054
3	63.166791	14	13	1.2798405
4	138.44871	10	14	2.4238475
5	58.803824	12	15	1.2080201
6	43.499719	21	19	.94809547

NODE NO	PRES
1	173.56152
2	56.759080
3	-187.27178
4	90.612272
5	124.73912
6	174.49256
7	234.29704
8	137.68186
9	90.499411
10	1.8531640
11	-431.50385
12	-444.80654
13	-414.56882
14	-333.72541
15	-515.84274
16	-558.57694
17	-819.91533
18	-563.79947
19	-502.51694
20	-513.04335
21	-461.27505

EXAMPLE d)

SYSTEM CONVERGED

CUT SEGMENT NO. 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	107.22736	1	2	.50847910
2	175.21238	2	3	.33999090
3	67.987886	4	2	.73625825
4	67.988364	5	4	.73625407
5	33.797088	1	5	1.2903330
6	21.023733	6	1	1.8792144
7	41.188190	6	5	1.1018765
8	62.219789	7	6	.79093518
9	68.165209	7	8	.73470886
10	31.640530	8	5	1.3598527
11	36.518452	8	9	1.2130802
12	79.613596	7	9	.64788424
13	33.975125	9	10	1.2849443
14	82.144110	9	11	.63163877
15	72.454131	10	11	.69930846

CUT SEGMENT NO. 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	25.057023	1	3	1.6362009
2	30.204237	1	2	1.4110028
3	14.952902	3	2	2.4544856
4	10.122645	4	2	3.3221378
5	57.011134	2	6	.84871898
6	57.077916	5	6	.84791872
7	114.09290	6	7	.48335520
8	125.90679	8	7	.44594000
9	29.898275	5	8	1.4224597
10	40.388287	10	5	1.1192989
11	48.259593	9	5	.97046547
12	23.492395	9	10	1.7217184
13	76.906496	11	10	.66631976
14	96.000241	11	8	.55645191
15	244.65367	12	11	.25808618

CUT PIPE RESULTS

1	175.15708	3	11	2.9404908
2	38.611233	5	10	.86185341
3	10.118925	13	14	.30092564
4	1.7541967	15	12	.08103909
5	71.742921	21	19	1.4186078

NODE NO	PRES
1	163.07801
2	-47.800579
3	-563.14491
4	44.541863
5	136.88548
6	174.26552
7	252.93163
8	160.15310
9	130.04919
10	103.60826
11	-1078.1927
12	-1099.5989
13	-1093.5068
14	-1096.5519
15	-1099.4567
16	-1166.7721
17	-1402.8157
18	-1120.4755
19	-1049.7284
20	-1063.3732
21	-947.95338

EXAMPLE e)

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	109.99473	1	2	.49801312
2	180.22728	2	3	.33218119
3	70.359039	4	2	.71611883
4	70.399002	5	4	.71578980
5	32.388987	1	5	1.3347963
6	22.386286	6	1	1.7884975
7	40.820743	6	5	1.1098044
8	63.209693	7	6	.78091911
9	68.064968	7	8	.73558386
10	33.072699	8	5	1.3127878
11	34.989944	8	9	1.2551591
12	78.724735	+ 7	9	.65381217
13	34.007016	9	10	1.2839843
14	79.688564	9	11	.64738967
15	69.705894	10	11	.72154399

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	27.378655	1	3	1.5253604
2	33.000567	1	2	1.3150715
3	16.353220	3	2	2.2887163
4	11.074894	4	2	3.0992375
5	56.094567	2	6	.85987396
6	55.770580	5	6	.86389518
7	111.87167	6	7	.49118004
8	128.12727	8	7	.43960379
9	4.9302736	5	8	5.7506770
10	33.538200	10	5	1.2982593
11	22.783158	9	5	1.7638797
12	22.783354	10	9	1.7638677
13	116.31161	11	10	.47580804
14	123.13521	11	8	.45413675
15	239.49582	12	11	.26267859

CUT PIPE RESULTS

1	180.28479	3	11	3.0111966
2	35.883677	+ 5	10	.81289653
3	11.071920	13	14	.32259299
4	4.3146134	12	15	.15729985

NODE NO	PRES
1	150.04141
2	-70.825718
3	-613.38287
4	27.424790
5	125.77629
6	162.55822
7	243.50091
8	150.96901
9	123.09211
10	96.606575
11	-1156.2558
12	-1181.3499
13	-1174.2048
14	-1177.7765
15	-1182.0286
16	-1246.5857
17	-1474.3468
18	-1182.8860
19	-1169.1121
20	-1156.1954
21	-911.74472

RESULTS FOR CHAPTER 5 SECTION E

EXAMPLE a)

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	71.238421	1	2	.70895523
2	43.092924	2	3	1.0627139
3	28.141222	2	4	1.4924995
4	83.743556	5	4	.62182543
5	25.076048	5	1	1.6352191
6	45.752410	6	1	1.0129137
7	36.336063	6	5	1.2179360
8	82.183440	7	6	.63139337
9	63.862408	7	8	.77446674
10	61.723730	8	5	.79606223
11	1.9661584	8	9	11.360500
12	63.936520	7	9	.77374157
13	60.114797	9	10	.81321037
+ 14	5.5981173	9	11	5.2245470
15	59.630912	11	10	.81852905

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	5.2168933	3	1	5.5106914
2	40.169957	1	2	1.1241613
3	40.757830	3	2	1.1111745
4	63.746270	4	2	.77560618
5	79.764244	2	6	.64689128
6	41.796726	5	6	1.0890198
7	121.56264	6	7	.45893682
8	118.44163	8	7	.46880224
9	48.733179	5	8	.96288520
10	5.1600497	10	5	5.5564613
11	20.468201	9	5	1.9192402
12	19.450416	9	10	1.9977658
13	45.695594	11	10	1.0139236
14	69.706605	11	8	.72153802
15	155.31644	12	11	.37542075

CUT PIPE RESULTS

1	73.523467
2	30.430469
3	81.457755
4	10.840535
5	45.943150
6	109.68689
7	64.911007
8	39.918667

3	11	1.4470533
4	3	.71292970
4	11	1.5724367
10	5	.31736732
14	13	.99055049
10	14	2.0033902
12	15	1.3083113
21	19	.88509973

NODE NO	PRES
1	-91.626717
2	-192.11038
3	-232.66026
4	-210.96548
5	-76.291739
6	-46.457607
7	83.704418
8	1.2445726
9	1.0715029
10	-72.851308
11	-339.05264
12	-374.78591
13	-338.10596
14	-292.59695
15	-459.70972
16	-498.08985
17	-762.96867
18	-510.32134
19	-449.04498
20	-458.78106
21	-413.71298

EXAMPLE c)

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	91.454204	1	2	.57886288
2	52.807275	2	3	.90270833
3	38.648168	2	4	1.1594041
4	88.405033	5	4	.59504854
5	44.379299	1	5	1.0379629
6	15.839524	6	1	2.3465392
7	48.139986	6	5	.97240092
8	63.979614	7	6	.77332058
9	69.295807	7	8	.72499696
10	40.744963	8	5	1.1114552
11	28.545994	8	9	1.4756891
12	76.724243	7	9	.66760270
13	53.998567	9	10	.88664840
14	51.270218	9	11	.92440967
15	14.030132	11	10	2.5794309

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	7.7477646	1	3	4.0797498
2	52.113875	1	2	.91235698
3	51.146247	3	2	.92621059
4	69.580851	4	2	.72259300
5	90.547310	2	6	.58357467
6	32.704883	5	6	1.3245263
7	123.25992	6	7	.45376080
8	116.74005	8	7	.47438042
9	48.535999	5	8	.96602528
10	16.879720	5	10	2.2326693
11	15.970331	9	5	2.3315058
12	24.219792	9	10	1.6807484
13	+ 48.507388	11	10	.96648280
14	68.203273	11	8	.73437721
15	156.90280	12	11	.37229934

CUT PIPE RESULTS

1	85.858030
2	33.051847
3	94.002318
4	44.855923
5	43.400320
6	112.97848
7	82.290014
8	40.198814

3	11	1.6410760
4	3	.76135540
4	11	1.7665921
5	10	.97171020
14	13	.94635963
10	14	2.0523495
12	15	1.5854659
21	19	.89006303

NODE NO	PRES
1	80.903874
2	-77.085535
3	-135.58425
4	-110.42004
5	38.147723
6	87.654038
7	170.38766
8	74.806838
9	55.462658
10	-5.4392354
11	-276.48380
12	-333.60385
13	-278.38288
14	-237.31057
15	-464.07186
16	-488.76362
17	-760.40436
18	-514.31485
19	-457.22207
20	-471.63219
21	-421.44259

EXAMPLE d)

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	123.30379	1	2	.45362872
2	72.665039	2	3	.69766430
3	50.640595	2	4	.93363747
4	130.88804	5	4	.43199631
5	31.492431	1	5	1.3649356
6	34.799153	6	1	1.2606411
7	48.767976	6	5	.96233341
8	83.568992	7	6	.62288009
9	63.694824	7	8	.77611211
10	72.130853	8	5	.70184529
11	8.4808540	9	8	3.8067063
12	62.732349	7	9	.78571372
13	77.354175	9	10	.66319136
14	23.154237	11	9	1.7415364
15	82.104152	11	10	.63188832

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	31.961409	1	3	1.3489832
2	102.21735	1	2	.52871660
3	95.112334	3	2	.56067788
4	117.87079	4	2	.47065743
5	59.264802	2	6	.82260466
6	56.026460	5	6	.86071584
7	115.30080	6	7	.47921401
8	124.69876	8	7	.44947233
9	35.258892	5	8	1.2475207
10	33.254290	10	5	1.3070773
11	41.907995	9	5	1.0867049
12	22.985621	9	10	1.7516102
13	70.266515	11	10	.71688186
14	89.435282	11	8	.58946907
15	224.59384	12	11	.27701488

CUT PIPE RESULTS

1	121.02284
2	48.359808
3	133.16930
4	21.493497
5	63.148241
6	181.01643
7	16.104216
8	64.898532

3	11	2.1710324
4	3	1.0321511
4	11	2.3478319
5	10	.54147883
14	13	1.2795369
10	14	3.0212588
12	15	.43171542
21	19	1.3081082

NODE NO	PRES
1	-95.223851
2	-367.04039
3	-471.19512
4	-421.28049
5	-118.29632
6	-67.619521
7	66.545936
8	-15.523165
9	-13.295293
10	-129.93459
11	-733.93962
12	-927.27069
13	-757.63258
14	-676.83208
15	-934.22313
16	-999.31599
17	-1239.9200
18	-962.48630
19	-895.65885
20	-908.78141
21	-810.76455

EXAMPLE e)

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	89.436503	1	2	.58946252
2	51.333931	2	3	.92348714
3	38.081072	2	4	1.1731757
4	84.050021	5	4	.61998338
5	47.658444	1	5	.98028055
6	11.562066	6	1	2.9976706
7	49.719711	6	5	.94750840
8	61.329974	7	6	.80018463
9	69.867453	7	8	.72019360
10	37.850315	8	5	1.1788842
11	31.993262	8	9	1.3479148
12	78.846900	7	9	.65299035
13	55.353860	9	10	.86912907
14	55.483632	9	11	.86749165
15	2.5411846	10	11	9.4184614

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	6.1699238	1	3	4.8531476
2	47.280999	1	2	.98655659
3	46.580976	3	2	.99843577
4	63.990414	4	2	.77321516
5	84.728595	2	6	.61594721
6	38.024944	5	6	1.1745585
7	122.78555	6	7	.45519448
8	117.31968	8	7	.47246390
9	50.413561	5	8	.93701525
10	2.0396010	5	10	11.061510
11	17.360085	9	5	2.1841466
12	17.584758	9	10	2.1622526
13	40.334909	11	10	1.1204834
14	66.870424	11	8	.74619116
15	142.21962	12	11	.40357490

CUT PIPE RESULTS

1	82.820195
2	31.534052
3	90.535248
4	51.195718
5	40.401570
6	104.29943
7	73.061070
8	34.981747

3	11	1.5937544
4	3	.73340513
4	11	1.7133911
5	10	1.0805076
14	13	.89365138
10	14	1.9227287
12	15	1.4396776
21	19	.79656300

NODE NO	PRES
1	104.20432
2	-47.521194
3	-103.10825
4	-79.981015
5	55.587167
6	108.06133
7	184.70611
8	87.694066
9	63.958691
10	.26980889
11	-235.10330
12	-283.02858
13	-236.37463
14	-200.26971
15	-388.21297
16	-420.58679
17	-690.32984
18	-442.01525
19	-380.26474
20	-388.39735
21	-352.39957

EXAMPLE f)

SYSTEM CONVERGED

CUT SEGMENT NO 1

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	90.054020	1	2	.58617328
2	51.727098	2	3	.91783908
3	38.324150	2	4	1.1672284
4	84.862091	5	4	.61516000
5	47.601559	1	5	.98122072
6	11.750700	6	1	2.9603234
7	49.723822	6	5	.94744546
8	61.483148	7	6	.79857538
9	69.840097	7	8	.72042186
10	38.126623	8	5	1.1720561
11	31.708553	8	9	1.3575322
12	78.685752	7	9	.65407491
13	55.198500	9	10	.87109837
14	55.195593	9	11	.87113532
15	.26566484	11	10	44.093085

CUT SEGMENT NO 2

PIPE NO	FLOW	FROM NODE	TO NODE	IMPEDANCE
1	6.6324072	1	3	4.5936129
2	48.281412	1	2	.97011333
3	47.499364	3	2	.98291484
4	65.015676	4	2	.76334964
5	85.850455	2	6	.60939998
6	36.964317	5	6	1.2013890
7	122.82650	6	7	.45507030
8	117.19477	8	7	.47287548
9	49.911794	5	8	.94457751
10	.47699915	5	10	30.670214
11	17.975032	9	5	2.1253669
12	17.993553	9	10	2.1236515
13	41.428236	11	10	1.0967649
14	67.274703	11	8	.74256355
15	144.67367	12	11	.39795021

CUT PIPE RESULTS

1	83.560213
2	31.834270
3	91.350268
4	50.616909
5	40.860573
6	105.85923
7	74.931436
8	35.972878

NODE NO	PRES
1	102.70092
2	-50.929456
3	-107.28694
4	-83.762919
5	54.188328
6	106.67032
7	183.66136
8	86.718021
9	63.360526
10	-.00602509
11	-241.42686
12	-291.19569
13	-242.87069
14	-206.02414
15	-401.30473
16	-432.07272
17	-701.97937
18	-454.14507
19	-392.84735
20	-401.32028
21	-363.54716

3	11	1.6053085
4	3	.73895262
4	11	1.7259275
5	10	1.0706769
14	13	.90176293
10	14	1.9461516
12	15	1.4694638
21	19	.81450791

Appendix G

i) Program listing for Daniels Solution

ii) Program listing for Diakoptics Program

```
 HARDY CROSS METHOD'
 BEGIN INTEGER MESH,BRANCH,N,QQ,M,MAX'
 READ MESH,BRANCH,MAX'
BEGIN REAL SUM1,SUM2,ERROR,PI,RHO,MU,ESPI,X'
 ARRAY L,D,REL(1:BRANCH,1:1),R,PHI,Q1,Q2,Q3(1:BRANCH)'
INTEGER ARRAY C(1:MESH,1:BRANCH),CON(1:MESH,1:MAX),NUM(1:MESH)'
SWITCH S:=L1,L2'

PROCEDURE FINDPHI'
 BEGIN REAL ARRAY RE(1:BRANCH),DUM,LAM(1:2)'
       SWITCH S:=NOW,NEW,AGAIN,L1,L2,L3'
      FOR N:=1 STEP 1 UNTIL BRANCH DO
      BEGIN M:=1'
IF Q1(N)=0 THEN GOTO L2'
      RE(N):=CHECKR(4*D(N,1)*RHO*3600*ABS(Q1(N))/(MU*PI*D(N,1)**2))'
      IF RE(N) LESS 2100 THEN GOTO NOW ELSE IF RE(N) LESS 4000
       THEN BEGIN PRINT PUNCH(3),££L?CRITICAL FLOW IN PIPE?,SAMELINE,
      N' IF REL(N,1)=0 THEN GOTO AGAIN ELSE GOTO NEW END ELSE IF
      REL(N,1)=0 THEN GOTO AGAIN ELSE GOTO NEW'
      NOW:PHI(N):=8/RE(N)' GOTO L2'
       AGAIN:LAM(M):=0.316*RE(N)**(-0.25)'
       IF RE(N) LESS 1.005 THEN BEGIN PHI(N):=LAM(M)/8' GOTO L2 END'
     L1:LAM(M+1):=(1/(0.87*LN(RE(N)*SQRT(LAM(M)))-0.8))**2'
       IF ABS((LAM(M+1)-LAM(M))/LAM(M)) GR 0.005 THEN BEGIN LAM(M)
      :=LAM(M+1)' GOTO L1 END'
      PHI(N):=LAM(M+1)/8' GOTO L2'
      NEW:DUM(M):=CHECKR(0.87*LN(3.7*D(N,1)/REL(N,1)))'
      IF RE(N) GR DUM(M)*200*D(N,1)/REL(N,1) THEN BEGIN PHI(N):=1/(8*
      DUM(M)**2)' GOTO L2 END'
      L3:DUM(M+1):=CHECKR(-0.87*LN(REL(N,1)/(3.7*D(N,1))+2.51/RE(N)*DUM(M)))
       IF ABS(1/DUM(M+1)-1/DUM(M)) LESS 0.0001 THEN BEGIN
      PHI(N):=CHECKR(1/(8*DUM(M+1)**2))' GOTO L2 END'
       DUM(M):=DUM(M+1)' GOTO L3'
      L2:END'
PRINT££L?RE=?,SAMELINE,RE(1),££S6??,RE(2)'
 END'
```

```
PI:=3.142'

READ ESPI,X,RHO,MU'
FOR N:=1 STEP 1 UNTIL MESH DO
FOR M:=1 STEP 1 UNTIL BRANCH DO
C(N,M):=0'
FOR N:=1 STEP 1 UNTIL MESH DO
BEGIN READ NUM(N)'
FOR M:=1 STEP 1 UNTIL NUM(N) DO
BEGIN READ CON(N,M)'
IF CON(N,M) LESS 0 THEN BEGIN CON(N,M):=CON(N,M)*(-1)'
C(N,CON(N,M)):=-1'
END ELSE C(N,CON(N,M)):=1'
END
END'
FOR N:=1 STEP 1 UNTIL BRANCH DO
READ Q1(N)'
FOR N:=1 STEP 1 UNTIL BRANCH DO
READ L(N,1)'
FOR N:=1 STEP 1 UNTIL BRANCH DO
READ D(N,1)'
FOR N:=1 STEP 1 UNTIL BRANCH DO
READ REL(N,1)'
QQ:=1'
L2:IF QQ=1 THEN FOR N:=1 STEP 1 UNTIL BRANCH DO
PHI(N):=0.05 ELSE FINDPHI'
QQ:=QQ+1'
FOR N:=1 STEP 1 UNTIL BRANCH DO
BEGIN
R(N):=2*PHI(N)*L(N,1)*RHO/(32.2*D(N,1)**5)'
Q3(N):=Q1(N)
END'
L1:FOR N:=1 STEP 1 UNTIL BRANCH DO
Q2(N):=Q1(N)'
FOR N:=1 STEP 1 UNTIL MESH DO
BEGIN SUM1:=SUM2:=0'

FOR M:=1 STEP 1 UNTIL NUM(N) DO
SUM1:=C(N,CON(N,M))*SIGN(Q1(CON(N,M)))*R(CON(N,M))*(ABS(Q1(CON(N,M)))
**X)+SUM1'
FOR M:=1 STEP 1 UNTIL NUM(N) DO
SUM2:=X*ABS(C(N,CON(N,M))*R(CON(N,M))*(ABS(Q1(CON(N,M)))**(X-1)))+
SUM2'
```

```
IF SUM2 NOTEQ 0 THEN BEGIN
ERROR:=-SUM1/SUM2'
 FOR M:=1 STEP 1 UNTIL NUM(N) DO
 Q1(CON(N,M)):=Q1(CON(N,M))+C(N,CON(N,M))*ERROR'
 END
 END'
 M:=0'
 FOR N:=1 STEP 1 UNTIL BRANCH DO
 IF ABS(Q2(N)-Q1(N)) LESS ESPI THEN M:=M+1'
 PRINT DIGITS(3),M'
 IF M NOTEQ BRANCH THEN GOTO L1'
 PRINT££L?PIPE   NO     FLOW?'
 FOR N:=1 STEP 1 UNTIL BRANCH DO
 PRINT££LS5??,DIGITS(3),N,SAMELINE,££S5??,Q2(N)'
 M:=0'
 FOR N:=1 STEP 1 UNTIL BRANCH DO
IF ABS(Q3(N)-Q1(N)) LESS ESPI THEN M:=M+1'
 IF M NOTEQ BRANCH THEN GOTO L2
 END
 END'
```

```
DIAKOPTICS PROGRAM'
BEGIN REAL MU,RHO,SUM1,PI,CON,NEG,LIMIT'
INTEGER NN,MM,N,M,Q,T,CUT,NUMBER,GR11,GR22,GR33,GR5,GR7,CC,CCC,
QQ,TOTNODE'
BOOLEAN FLAG,FLAG1,FLAG21,FLAG22,FLAG3'
READ  NUMBER,GR11,GR22,GR33,TOTNODE,CUT,PI,RHO,MU,LIMIT'
BEGIN INTEGER ARRAY BRAN,NOD,GR2,GR3,GR4,GR6(1:NUMBER),REM(1:3,1:2)'
SWITCH
 SSS:=START'
FOR N:=1 STEP 1 UNTIL NUMBER DO
READ BRAN(N),NOD(N)'
START:READ N' QQ:=1'
FLAG:=N=0'
IF FLAG THEN BEGIN QQ:=2' READ N'
FLAG1:=N=1' READ N' FLAG21:=N=21' READ N' FLAG22:=N=22'
READ N' FLAG3:=N=3'
IF FLAG21 OR FLAG22 THEN READ CUT'
END ELSE FLAG1:=FLAG21:=FLAG22:=FLAG3:=FALSE'
SUM1:=20000'
BEGIN  INTEGER ARRAY C(1:2*CUT)'
INTEGER NODE,BRANCH'
REAL ARRAY V1,VA(1:TOTNODE,1:1),YB(1:CUT,1:CUT),
DC,PUMPC,LC,RELC(1:CUT,1:1)'
SWITCH SS:=L1,L2,L3,L4,L5,L6,L7,L8,L9,L10'&


PROCEDURE MXAUX(A,B,C,D,E)'
VALUE D,E' BOOLEAN D,E' ARRAY A,B,C'
COMMENT THIS PROCEDURE IS USED IN MXSUM, MXDIFF,MXCOPY,
MXNEG AND MXQUOT AS AN AUXILIARY PROCEDURE'
BEGIN INTEGER AA,AB,AC,SA'
AA:=ADDRESS(A)' AB:=ADDRESS(B)'
AC:=ADDRESS(C)' SA:=SIZE(A)'
IF SA NOTEQ SIZE(B) OR SA NOTEQ SIZE(C) THEN
BEGIN PRINT PUNCH(3),££L?MXAUX ERROR?'
STOP
END'
SA:=SA+AA-1'
FOR AA:=AA STEP 1 UNTIL SA DO
BEGIN LOCATION(AA):=
IF D THEN (IF E THEN LOCATION(AB) ELSE -LOCATION(AB))
ELSE IF E THEN LOCATION(AB)+LOCATION(AC)
ELSE LOCATION(AB)-LOCATION(AC)'
ELLIOTT(2,2,AB,0,2,2,AC)
END
END MXAUX'
```

```
PROCEDURE MXSUM(A)BECOMES :(B)PLUS :(C)'
ARRAY A,B,C'
MXAUX(A,B,C,FALSE,TRUE)'


PROCEDURE MXPROD(A) BECOMES :(B) TIMES :(C)'
ARRAY A,B,C'
COMMENT A MUST NOT EQUAL B OR C'
BEGIN INTEGER AA,AB,AC,RA2,RB2,J,JSTOP,L,LSTOP,M,MSTART,SA'
REAL SUM'
AA:=ADDRESS(A)' SA:=SIZE(A)+AA-1'
AB:=ADDRESS(B)'AC:=ADDRESS(C)'
RA2:=RANGE(A,2)' RB2:=RANGE(B,2)'
IF AA=AB OR AA=AC OR RANGE(C,2) NOTEQ RA2
OR RANGE(C,1) NOTEQ RB2 OR RANGE(A,1) NOTEQ RANGE(B,1) THEN
BEGIN PRINT PUNCH(3),££L?MXPROD ERROR?'
STOP
END'
FOR AA:=AA STEP RA2 UNTIL SA DO
BEGIN JSTOP:=AA+RA2-1' MSTART:=AC-1'
FOR J:=AA STEP 1 UNTIL JSTOP DO
BEGIN M:=MSTART:=MSTART+1'
LSTOP:=AB+RB2-1' SUM:=0'
FOR L:=AB STEP 1 UNTIL LSTOP DO
BEGIN SUM:=SUM+LOCATION(L)*LOCATION(M)'
ELLIOTT(3,0,RA2,0,2,4,M)'
END'
LOCATION(J):=SUM
END'
AB:=AB+RB2
END
END MXPROD'&

PROCEDURE PRINTMX(A)' ARRAY A'
BEGIN INTEGER I,J,RA2,SA,AA'
AA:=ADDRESS(A)'
SA:=SIZE(A)+AA-1'
RA2:=RANGE(A,2)'
SAMELINE'
FOR AA:=AA STEP RA2 UNTIL SA DO
BEGIN PRINT ££L2??'
I:=AA+RA2-1'
FOR J:=AA STEP 1 UNTIL I DO
PRINT LOCATION(J)'
END
END'
```

```
PROCEDURE READMX(A)'ARRAY A'
BEGIN INTEGER AA,SA'REAL X'
AA:=ADDRESS(A)' SA:=SIZE(A)+AA-1'
FOR AA:=AA STEP 1 UNTIL SA DO BEGIN READ X' LOCATION(AA):=X'
END '
END'


PROCEDURE CHOLESKI (B)'
ARRAY B'
BEGIN REAL X,D'
INTEGER A,AA,BB,CC,P,SA,Q,RA,N,M,T,J,QQ'
SWITCH S:=L1,L2,L3,L4,L5,L6'
A:=AA:=ADDRESS (B)' RA:=RANGE (B,1)'
FOR P:=1 STEP 1 UNTIL RA DO
BEGIN A:=AA+(P-1)*RA+P-1'
FOR Q:=P STEP 1 UNTIL RA DO
BEGIN X:=LOCATION (A)'
IF Q=1 THEN GOTO L4'
IF P=1 THEN GOTO L2'
IF P NOTEQ Q THEN BEGIN BB:=AA+P-1' GOTO L3 END'
BB:=AA+Q-1'
FOR J:=2 STEP 1 UNTIL P DO
BEGIN X:=X-LOCATION (BB)*LOCATION  (BB)'
BB:=BB+RA'
END'
L4:IF X LESSEQ 0 THEN BEGIN PRINT££L?MATRIX SINGULAR AT ROW?,P'
GOTO L1 END'
D:=1/SQRT(X)' BB:=BB-RA'
GOTO L2'
L3:FOR J:=2 STEP 1 UNTIL P DO
BEGIN CC:=BB+Q-P'
IF LOCATION(CC)=0 THEN GOTO L6'
X:=X-LOCATION(BB)*LOCATION(CC)'
L6:BB:=BB+RA'
END'
L2:LOCATION(A):=X*D'
A:=A+1
END
END'
A:=AA+SIZE(B)-1'
Q:=0' SA:=RA'
L5:P:=A' QQ:=A-Q'
D:=X:=1/LOCATION(QQ)'
```

```
CC:=QQ+Q*SA' BB:=QQ+Q'

FOR N:=1 STEP 1 UNTIL Q DO
BEGIN X:=X-LOCATION(BB)*LOCATION(CC)'
LOCATION(BB):=LOCATION(CC)'
BB:=BB-1' CC:=CC-SA'
END'
LOCATION(QQ):=X*D'
Q:=Q+1' T:=A-SA'
FOR M:=Q STEP 1 UNTIL SA-1 DO
BEGIN X:=0'
FOR N:=1 STEP 1 UNTIL M DO
BEGIN X:=X-LOCATION(P)*LOCATION(T)'
P:=P-1' T:=T-1'
END'
LOCATION(P):=X/LOCATION(T)'
P:=P+M' T:=T+M-SA
END'
A:=A-SA'
IF Q LESS SA THEN GOTO L5'
L1:END'&


PROCEDURE ZERO(A)'
ARRAY A'
BEGIN INTEGER AA,SA,N'
AA:=ADDRESS(A)'
SA:=SIZE(A)+AA-1'
FOR N:=AA STEP 1 UNTIL SA DO
LOCATION(N):=0'
END'


PROCEDURE CUTPIPEDATA'
BEGIN
ZERO(YB)'
ZERO(REM)'
IF FLAG22 THEN
FOR N:=1 STEP 1 UNTIL 3 DO
FOR M:=1 STEP 1 UNTIL 2 DO
READ REM(N,M)'
FOR N:=1 STEP 1 UNTIL CUT DO
READ YB(N,N)'
READMX (DC)'
READMX (LC)'
READMX (RELC)'
READMX (PUMPC)'
```

```
M:=2*CUT'
FOR N:=1 STEP 1 UNTIL M DO
READ C(N)'
END'&


PROCEDURE CALCULATE'
BEGIN
CC:=CCC:=1'
&FOR NN:=1 STEP 1 UNTIL NUMBER DO
BEGIN BRANCH:=BRAN(NN)+1'
NODE:=NOD(NN)+1'
BEGIN ARRAY DELTP,REL,FLOW,D,L,IMP(1:BRANCH,1:1),PRES(1:NODE,1:1)
,PHI,REQ(1:BRANCH),
      ADMITT(1:NODE-1,1:NODE-1),B,E(1:2)'
      INTEGER ARRAY GRAP(1:2*BRANCH)'
      SWITCH SSS:=NEW,S1,S2,S3'


PROCEDURE FORMDELTP'
BEGIN ARRAY PUMP(1:BRANCH)'
N:=SIZE(D) DIV 64'
M:=IF SIZE(D)=N*64 THEN N ELSE N+1'
LOCATE (GR5+(NN-1)*M,2)'
FILMREAD (PUMP,2)'
LOCATE (GR22+3*M*NN,2)'
M -:=BRANCH*2'
FOR N:=2 STEP 2 UNTIL M DO
DELTP(N DIV 2,1):=PRES(GRAP(N),1)-PRES(GRAP(N-1),1)+PUMP(N DIV 2)'
END'


PROCEDURE FORMADMIT'
BEGIN SWITCH S:=AGAIN,NOW'
M:=2*BRANCH'
FOR N:=2 STEP 2 UNTIL M DO
BEGIN  Q:=N DIV 2'
T:=GRAP(N)'
IF T GR NOD(NN) THEN GOTO AGAIN'
ADMITT(T,T):=ADMITT(T,T)+IMP(Q,1)'
```

```
AGAIN:T:=GRAP(N-1)'
 IF T GR NOD(NN) THEN GOTO NOW'
ADMITT(T,T):=ADMITT(T,T)+IMP(Q,1)'
NOW:END'
FOR M:=1 STEP 1 UNTIL BRANCH DO
BEGIN  N:=GRAP(2*M)'
Q:=GRAP(2*M-1)'
 IF N LESSEQ NOD(NN) AND Q LESSEQ NOD(NN) THEN
ADMITT(N,Q):=ADMITT(Q,N):=-IMP(M,1)'
END
END'&


PROCEDURE PIPEDATA'
BEGIN
ARRAY INLET(1:NOD(NN),1:1)'


READMX (D)' FILMWRITE (D,2)'
READMX (L)' FILMWRITE (L,2)'
READMX (FLOW)'READMX (REL)' FILMWRITE (REL,2)'
GR5:=BLOCKNUMBER+1'
FOR N:=1 STEP 1 UNTIL 2*(BRAN(NN)+1) DO
READ GRAP(N)'
M:=2*(BRAN(NN)+1)'
FOR N:=1 STEP 1 UNTIL NOD(NN)+1 DO
BEGIN READ T' MM:=0'
FOR Q:=1 STEP 1 UNTIL M DO
IF GRAP(Q)=N THEN MM:=MM+1'
IF MM NOTEQ T THEN PRINT PUNCH(3),££L?
ERROR IN DATA GRAP AT NODE?,N'
END'

 READMX (INLET)'
FILMWRITE (INLET,1)' GR3(NN):=BLOCKNUMBER+1'
FILMWRITE (GRAP,1)' GR4(NN):=BLOCKNUMBER+1'
ZERO (ADMITT)'
FOR N:=1 STEP 1 UNTIL BRANCH DO
PHI(N):=0.002'
END'&
```

```
PROCEDURE FORMIMP'

BEGIN
 FOR N:=1 STEP 1 UNTIL BRANCH DO

IMP(N,1):=1811.25*D(N,1)**5*PI**2/(PHI(N)*RHO*L(N,1)*FLOW(N,1))'
FILMWRITE (IMP,3)'
GR2(NN):=BLOCKNUMBER+1'
IF FLAG THEN BEGIN LOCATE (4096-GR2(NN),3)' FILMWRITE (FLOW,3)'
LOCATE (GR6(NN),3)'
END'
M:=0'
FOR N:=1 STEP 1 UNTIL 3 DO
IF REM(N,2)=NN THEN BEGIN T:=CUT-M'
YB(T,T):=-1/IMP(REM(N,1),1)'
M:=M+1'
END'
END'


PROCEDURE INVADMIT'
BEGIN
FORMADMIT'&
CHOLESKI (ADMITT)'
FILMWRITE (ADMITT,3)'&
GR6(NN):=BLOCKNUMBER+1'
END'


PROCEDURE PIPECONSTANTS'
BEGIN
FILMREAD (D,2)' FILMREAD (L,2)' FILMREAD (REL,2)'
ZERO (ADMITT)'

T:=0'
LOCATE (GR3(NN),1)' FILMREAD (GRAP,1)'

FOR M:=CCC STEP 1 UNTIL NOD(NN)+CCC-1 DO
BEGIN T:=T+1'
```

```
PRES(T,1) :=VA(M,1)'
END'
CCC:=CHECKI(NOD(NN)+CCC)'
PRES(NODE,1) :=0'
FORMDELTP'
BEGIN ARRAY RE(1:BRANCH)'

FOR N:=1 STEP 1 UNTIL BRANCH DO
BEGIN
REQ(N):=SQRT(ABS(DELTP(N,1))*D(N,1)**3*RHO*1.296@7*32.2/(4*L(N,1)
*MU**2))'
IF REQ(N) LESS 126.49 THEN BEGIN
PRINT PUNCH(3),££L?LAMINAR FLOW IN PIPE NO?,SAMELINE,
N,£SEGMENT?,NN'
RE(N):=REQ(N)**2/8 END ELSE BEGIN
RE(N):=-2.5*REQ(N)*LN(REL(N,1)/(D(N,1)*3.7)+1/(1.13*REQ(N)))'
IF RE(N) LESS 3000 THEN PRINT
  PUNCH(3),££L?TRANSITIONAL FLOW IN PIPE NO?,N,£SEGMENT?,NN END'
PHI(N):=(REQ(N)/RE(N))**2'
FLOW(N,1):=RE(N)*PI*D(N,1)*MU/(4*60*RHO)'
END'
END'
END'

%1
CC:=CCC:=1'
IF QQ=1 THEN BEGIN
 PIPEDATA' FORMIMP' INVADMIT'
GOTO S1
END'
PIPECONSTANTS'
FORMIMP'
IF FLAG THEN GOTO S1'
INVADMIT'
S1:END
END
END CALCULATE'&


PROCEDURE FORMV1(V1)' ARRAY V1'
BEGIN
CC:=1'

FOR NN:=1 STEP 1 UNTIL NUMBER DO
BEGIN
ARRAY ADMITT(1:NOD(NN),1:NOD(NN)),INLET,SUM,PRES(1:NOD(NN),1:1),
IMP(1:BRAN(NN)+1,1:1)'
```

```
INTEGER ARRAY GRAP(1:2*(BRAN(NN)+1),1:1)'
ARRAY PUMP(1:BRAN(NN)+1,1:1)'
SWITCH S:=NOW,AGAIN'
ZERO(SUM)'
FILMREAD (INLET,1)' FILMREAD (GRAP,1)'
FILMREAD (IMP,3)' FILMREAD (ADMITT,3)'
FILMREAD (PUMP,2)'
M:=2*(BRAN(NN)+1)'
FOR N:=2 STEP 2 UNTIL M DO
BEGIN Q:=N DIV 2'
T:=GRAP(N,1)'
IF T GR NOD(NN) THEN GOTO AGAIN'
SUM(T,1):=SUM(T,1)+PUMP(Q,1)*IMP(Q,1)'
AGAIN:T:=GRAP(N-1,1)'
IF T GR NOD(NN) THEN GOTO NOW'
SUM(T,1):=SUM(T,1)-PUMP(Q,1)*IMP(Q,1)'
NOW:END'
FOR N:=1 STEP 1 UNTIL NOD(NN) DO
SUM(N,1):=INLET(N,1)-SUM(N,1)'
MXPROD(PRES,ADMITT,SUM)'
M:=0'
FOR N:=CC STEP 1 UNTIL NOD(NN)+CC-1 DO
BEGIN M:=M+1'
V1(N,1):=PRES(M,1)'
END'
CC:=CC+NOD(NN)'
END
END'&


PROCEDURE YBDASH'
BEGIN ARRAY TOTSUM(1:CUT,1:CUT)' ZERO(TOTSUM)'
CC:=0'
FOR NN:=1 STEP 1 UNTIL NUMBER DO
BEGIN ARRAY ADMITT(1:NOD(NN),1:NOD(NN)),SUM(1:CUT,1:CUT),A(1:
NOD(NN),1:CUT)' SWITCH S:=AGAIN,NOW,L1,L2'
LOCATE (GR2(NN),3)'
 FILMREAD (ADMITT,3)'
ZERO (A)'
 T:=0' M:=2*CUT'
FOR N:=2 STEP 2 UNTIL M DO
BEGIN
T:=T+1'
IF C(N-1) GR CC AND C(N-1) LESSEQ NOD(NN)+CC THEN
FOR Q:=1 STEP 1 UNTIL NOD(NN) DO
```

```
A(Q,T):=-ADMITT(Q,C(N-1)-CC)'
IF C(N) GR CC AND C(N) LESSEQ NOD(NN)+CC THEN
FOR Q:=1 STEP 1 UNTIL NOD(NN) DO
A(Q,T):=ADMITT(Q,C(N)-CC)+A(Q,T)'
END'
T:=0'
FOR N:=2 STEP 2 UNTIL M DO
BEGIN T:=T+1'
IF  C(N-1) GR CC AND C(N-1) LESSEQ NOD(NN)+CC THEN
FOR Q:=1 STEP 1 UNTIL CUT DO
TOTSUM(T,Q):=TOTSUM(T,Q)-A(C(N-1)-CC,Q)'
IF C(N) GR CC AND C(N) LESSEQ NOD(NN)+CC THEN
FOR Q:=1 STEP 1 UNTIL CUT DO
TOTSUM(T,Q):=TOTSUM(T,Q)+A(C(N)-CC,Q)'
END'
CC:=CC+NOD(NN)
END'
LOCATE (GR4(NUMBER),1)'

FOR M:=1 STEP 1 UNTIL CUT DO
TOTSUM(M,M):=TOTSUM(M,M)+YB(M,M)'&
CHOLESKI (TOTSUM)'
FILMWRITE (TOTSUM,1)'&
PRINTMX (TOTSUM)'
LOCATE (GR4(NUMBER),1)'
END'&



PROCEDURE FORMVA(V1)' ARRAY V1'
BEGIN ARRAY I2,V22(1:TOTNODE,1:1)'&
BEGIN  ARRAY VD,ID(1:CUT,1:1)'
REAL SUM2'
LOCATE (GR2(1),3)'

FOR N:=1 STEP 1 UNTIL CUT DO
BEGIN M:=2*N'
VD(N,1):=V1(C(M-1),1)-V1(C(M),1)'
END'
FOR N:=1 STEP 1 UNTIL CUT DO
VD(N,1):=VD(N,1)+PUMPC(N,1)'&

BEGIN ARRAY TOTSUM(1:CUT,1:CUT)'
FILMREAD (TOTSUM,1)' MXPROD (ID,TOTSUM,VD)'
```

```
END'
M:=2*CUT'
ZERO (I2)' CC:=0'
FOR N:=2 STEP 2 UNTIL M DO
BEGIN CC:=CC+1'
I2(C(N-1),1):=I2(C(N-1),1)-ID(CC,1)'
I2(C(N),1):=I2(C(N),1)+ID(CC,1)'
END'
CC:=1'
FOR N:=1 STEP 1 UNTIL NUMBER DO
BEGIN REAL ARRAY ADMITT(1:NOD(N),1:NOD(N)),V2(1:NOD(N),1:1),
CUTI2(1:NOD(N),1:1)'
LOCATE (GR2(N),3)'
FILMREAD (ADMITT,3)'
T:=1'
FOR M:=CC STEP 1 UNTIL NOD(N)+CC-1 DO
BEGIN CUTI2(T,1):=I2(M,1)'
T:=T+1'
END'
T:=1'
MXPROD (V2,ADMITT,CUTI2)'
FOR Q:=CC STEP 1 UNTIL NOD(N)+CC-1 DO
BEGIN V22(Q,1):=V2(T,1)'
T:=T+1'
END'
CC:=CC+NOD(N)'
END'
MXSUM (VA,V22,V1)'&
PRINTMX (VA)'
END
END'&


PROCEDURE TEST'
BEGIN
IF QQ GR 1 THEN BEGIN ARRAY VA2(1:TOTNODE,1:1)'
FILMREAD (VA2,3)'
SUM1:=0'
LOCATE (GR6(NUMBER),3)'
FOR N:=1 STEP 1 UNTIL TOTNODE DO
SUM1:=SUM1+SQRT((VA2(N,1)-VA(N,1))**2)'
FILMWRITE (VA,3)'
END ELSE FILMWRITE (VA,3)'
FLAG:=SUM1 LESS LIMIT'
END'
```

```
PROCEDURE FORMCUTCON'
BEGIN ARRAY PHIC,FLOWC,REQC,REC(1:CUT),DPC(1:CUT,1:1)'
SWITCH S:=L4'
FOR N:=1 STEP 1 UNTIL CUT DO
BEGIN
M:=N*2'
DPC(N,1):=VA(C(M-1),1)-VA(C(M),1)+PUMPC(N,1)'
END'
FOR N:=1 STEP 1 UNTIL CUT DO
BEGIN
REQC(N):=SQRT(ABS(DPC(N,1))*DC(N,1)**3*RHO*1.296@7*32.2/(4*LC(N,1)
*MU**2))'
REC(N):=-2.5*REQC(N)*LN(RELC(N,1)/(DC(N,1)*3.7)+1/(1.13*REQC(N)))'
PHIC(N):=(REQC(N)/REC(N))**2'
FLOWC(N):=REC(N)*PI*DC(N,1)*MU/(4*60*RHO)'
YB(N,N):=PHIC(N)*RHO*LC(N,1)*FLOWC(N)/(1811.25*DC(N,1)**5*PI**2)'
END'
IF FLAG THEN BEGIN LOCATE (GR4(NUMBER)+(TOTNODE DIV 62+2),1)'
FILMWRITE (FLOWC,1)'
END'
L4:END'&


PROCEDURE RESULTSPRINT'
BEGIN REAL DP'
CC:=1'
IF FLAG1 THEN PRINT££L?NEW PUMP CONFIGURATION TO FIRST DEGREE APROXI
MATION? ELSE IF FLAG21 OR FLAG22 THEN PRINT££L?NEW PIPE CONFIGURATION TO FI
 DEGREE APROXIMATION? ELSE IF FLAG3 THEN PRINT££L?NEW DEMAND VECTOR
 TO FIRST DEGREE APROXIMATION?'
IF FLAG THEN PRINT££L?SYSTEM CONVERGED?'
FOR NN:=1 STEP 1 UNTIL NUMBER DO
BEGIN INTEGER ARRAY GRAP(1:2*(BRAN(NN)+1))'
ARRAY FLOW,IMP(1:BRAN(NN)+1,1:1),PRES(1:NOD(NN)+1,1:1)'
M:=0'
LOCATE (GR3(NN),1)' FILMREAD (GRAP,1)'
FILMREAD (IMP,3)'
LOCATE (4096-GR2(NN),3)'
FILMREAD (FLOW,3)' LOCATE (GR6(NN),3)'
FOR N:=CC STEP 1 UNTIL NOD(NN)+CC-1 DO
BEGIN
M:=M+1'
PRES(M,1):=VA(N,1)'
END'
PRES(NOD(NN)+1,1):=0'
```

```
PRINT££L4?CUT SEGMENT NO?,SAMELINE,NN'
PRINT££L?PIPE NO     FLOW        FROM    NODE TO NODE     IMPEDANCE?'
M:=2*(BRAN(NN)+1)'
Q:=0'
FOR N:=2 STEP 2 UNTIL M DO
BEGIN Q:=Q+1'
DP:=PRES(GRAP(N-1),1)-PRES(GRAP(N),1)'
IF DP LESS 0 THEN PRINT££LS3??,DIGITS(4),Q,SAMELINE,££S3??,FLOW(Q,1)
,££S7??,GRAP(N),££S2??,GRAP(N-1),££S5??,IMP(Q,1) ELSE PRINT££LS3??,
DIGITS(4),Q,SAMELINE,££S3??,FLOW(Q,1),££S7??,GRAP(
N-1),££S2??,GRAP(N),££S4??,IMP(Q,1)'
END'
CC:=CC+NOD(NN)'
END'
BEGIN ARRAY FLOWC(1:CUT,1:1)'
LOCATE (GR4(NUMBER)+(TOTNODE DIV 62+2),1)'
FILMREAD (FLOWC,1)'
PRINT££L4?CUT PIPE RESULTS?,££L2??'
FOR N:=1 STEP 1 UNTIL CUT DO
BEGIN M:=N*2'
DP:=VA(C(M-1),1)-VA(C(M),1)'
IF DP LESS 0 THEN PRINT££LS3??,DIGITS(3),N,SAMELINE,££S3??,FLOWC(N,1),££S
7??,C(M),££S2??,C(M-1),££S4??,YB(N,N) ELSE PRINT££LS3??,DIGITS(3),N,SAMELI
,££S3??,FLOWC(N,1),££S7??,C(M-1),££S2??,C(M),££S4??,YB(N,N)
END'
PRINT££L?NODE NO    PRES?'
FOR N:=1 STEP 1 UNTIL TOTNODE DO
PRINT££LS3??,DIGITS(3),N,SAMELINE,££S3??,VA(N,1)'
PRINT££L4?END OF FILE BLOCKNUMBERS?'
PRINT££L?HANDLER 1?,SAMELINE,PREFIX(££S3??),GR4(NUMBER),£HANDLER 2?,GR5,£HA
?,GR6(NUMBER)'
END'
END RESULTSPRINT'&
```

```
IF FLAG1 THEN BEGIN LOCATE (GR5,2)'

FOR NN:=1 STEP 1 UNTIL NUMBER DO
BEGIN ARRAY PUMP(1:BRAN(NN)+1,1:1)'
READMX (PUMP)'  FILMWRITE (PUMP,2)'
END'
GR7:=BLOCKNUBER+1'
GOTO L3
END'
IF FLAG3 THEN BEGIN LOCATE (GR11,1)'
FOR NN:=1 STEP 1 UNTIL NUMBER DO
BEGIN ARRAY INLET(1:NOD(NN),1:1)'
READMX (INLET)' FILMWRITE (INLET,1)'
LOCATE (GR4(NN),1)'
END'
GOTO L3
END'
L1:LOCATE (GR11,1)' LOCATE (GR22,2)' LOCATE (GR33,3)'
CUTPIPEDATA'
IF FLAG21 OR FLAG22 THEN BEGIN LOCATE (GR7,2)' FILMREAD (V1,2)'
 GOTO L4 END'
L5:CALCULATE'
L3:LOCATE (GR11,1)' LOCATE (GR33,3)'&
IF FLAG AND QQ NOTEQ 2  THEN GOTO L6'

IF QQ=1 THEN BEGIN
FOR NN:=1 STEP 1 UNTIL NUMBER DO
BEGIN ARRAY PUMP(1:BRAN(NN)+1,1:1)'
READMX (PUMP)' FILMWRITE (PUMP,2)'
END'
GR7:=BLOCKNUMBER+1'

END'
LOCATE (GR5,2)'
FORMV1(V1)'
LOCATE (GR2(1),3)'
L4:YBDASH'
L2:FORMVA(V1)'

IF NOT FLAG THEN TEST'

FORMCUTCON'
L10:LOCATE (GR11,1)' LOCATE (GR22,2)' LOCATE (GR33,3)'
QQ:=QQ+1'
GOTO L5'
L6:RESULTSPRINT'
```

```
IF FLAG1 OR FLAG21 OR FLAG22 OR FLAG3 THEN GOTO L8'
LOCATE (GR7,2)' FILMWRITE (V1,2)'
WAIT'
GOTO START'
L8:WAIT'
READ N'
FLAG:=N=0'
IF FLAG THEN BEGIN WAIT'
GOTO START END'
L9:FLAG:=FLAG1:=FLAG21:=FLAG22:=FLAG3:=FALSE'
GOTO L10'
END
END
END
END
END
END
END OF PROGRAM'
```

REFERENCES

1) Pase H.F. Piping design for Process Plants Wiley N.Y. 1963

2) Le Corbeiller P. matrix analysis of electrical networks Wiley 1950

3) Livesley R.K. Analysis of rigid frames by an electronic computer Engineering (London) vol 176, August 21 1953 P. 230

4) Crank J. Mathematics of Diffusion Oxford Claredon Press

5) Cross H. Analysis of flow in networks of conduits or conduction. Bul 286 Univ. of Illinios Experimental Station Urbana Ill 1936

6) Van der Berg De Ingenieur JRG 75/ NR5D/ 13-11-63

7) Kneibes D.V. Wilson G.G. Chem Engng. Prog. Symp. Sep. 1960, 56, No 31 p. 49

8) Hunn R.J. Ralph J.I.A. I. Ch. E. Computer Program No. 10.

9) Ingels D.M. Powers J.E. Chem Eng. Prog. 60. No.2. Feb. 1964.

10) Moody L.F. Trans. Am. Soc. Mech. Eng. 1944, 66, 761.

11) Ingels D.M. M.Ch. E. Thesis Univ. of Oklahoma (1962)

12) Brown G.G. Unit Operations Wiley N.Y. 1960.

13) Daniel P.T. Trans Inst Chem. Eng. T77 Vol 44 1966

14) Perry J.H. Chem Eng. Handbook 3rd Ed. P. 385.

15) Dolan J.J. Eng. News. Record cxvii p. 475 1936

16) Drew T.B. Genereaux R.P. The Reynolds Number relationship Trans. A.I. Ch. E. July 1936.

17) Knights I.A. Allen J.W. Midland Junior Gas Association - 60th Session 1964.

18) Branin F.H. I.B.M. Technical Report T.R. 00.855 March 30th 1962 Data Systems Div. Poughkeepsie N. Y.

19) Roth P.J. Quart App. math. Vol XVII No.1. April 1959.

20) Kron G. Diakoptics Macdonald London 1963 (Reprinted from the Electrical Journal.)

21) Brameller A. Diakoptics A.E.I. Ltd., Systems Eng. Dept. Manchester 17.

22) Kron G. Tensor Analysis of Networks Wiley N.Y. 1939.

LIST OF SYMBOLS

b	Number of branches in system
e'	Node to Datum potential
e	Branch potential rise in direction of assumed flow
g_c	Gravitational acceleration
h	head lost in pipe
i'	Mesh flow
i	Component of branch flow due to assumed mesh flow
i_p	Total branch flow in promitive system
m	Number of basic meshes in system
n	number of nodes in system
n'	Hardy Cross Exponent
p	Fluid pressure
r	Hardy Cross resistance factoe
r_i	Van der Berg flow residue at node i
u	Fluid velocity
$\underline{A}$	Incidence matrix
$\underline{A}_1$	Square non singular incidence matrix
$\underline{C}$	Branch - mesh incidence matrix
$\underline{C}_1$	Square non singular incidence matrix
D	diameter of pipe
I'	Nodal demand or input
I	Assumed branch inpressed flow
J	Total flow in branch
L	length of pipe
Q	Fluid flow in pipe

R resistance

V Potential rise in branch due to impedemce element in direction of assumed flow

$\underline{U}$ Identity Matrix.

Y Branch admittance

$\underline{Y}'$ Derived or transformed admittance matrix

Z Branch impedence

$\underline{Z}'$ Derived or transformed admittance matrix

μ Fluid viscosity

ϵ Pipe roughness

ρ Fluid Density

$\underline{\ }$ Underlined Qunatities vectors

$\underline{\tilde{A}}$ Designates $\underline{A}$ transpose

Note on Subscripts

Double subscriptions e.g. I'_{A1} $\equiv$ flow at node 1 of network A

Triple subscripts e.g. $\underline{A}_{jAI}$ $\equiv$ Junction part of incidence matrix for the i^{th} segment of network A

$\underline{I}'_{1A2}$ $\equiv$ vector of nodal demands for segment 2 network A