Hybrid method for the solution of piping networks

Y. M. Hamam, B.E.E., M.Sc., and A. Brameller, Ph.D., A.F.I.M.A., C.Eng., M.I.E.E.

Indexing terms: Hydraulic systems, Nonlinear equations

Abstract

A method is described for the analysis of general fluid-flow problems in which there is a nonlinear relationship between pressure and flow. The practical application is illustrated by solving gas-distribution networks, and a comparison of different methods of ordered elimination as applied to the problem is also included. In general piping networks, the solution based on loop analysis produces the best convergence characteristics, while nodal analysis is more convenient for the formulation of equations with maximum sparsity. Using electrical theory, the method described combines the mesh and nodal formulation into an efficient hybrid method.

List of principal symbols

Q_i = flow through branch i
Q_β = vector (i.e. single-column matrix) of the flow through the pipes, dimension β
ΔP_i = pressure drop across branch i
ΔP_β = vector of the pressure drop across the branches, dimension β
K_i = coefficient of friction in branch i
L_k = load at node k
L_α = vector of the loads, dimension α
P_α = nodal-pressures vector, dimension α
$D_{\gamma\beta}$ = loop-incidence matrix
$D^T_{\beta\gamma}$ = transpose of $D_{\gamma\beta}$
$C_{\alpha\beta}$ = nodal-incidence matrix
$C^T_{\beta\alpha}$ = transpose of $C_{\alpha\beta}$
$J_{\gamma\gamma}$ = loop Jacobian matrix
r_i = resistance of branch i in the equivalent electrical network
I_i = current of branch i of the electrical analogue
I_γ = vector of the currents in the cotree branches of the electrical analogue
$R_{\beta\beta}$ = diagonal matrix of the branch resistances in the electrical analogue
F_γ = error in satisfying the loop equations
α = number of nodes in the network excluding the reference
β = number of branches
γ = number of loops

1 Introduction

The choice of a method for network analysis depends on many factors and is usually based on either the loop or nodal formulation. The important factors are the convergence characteristics, storage requirements and computation time.

The equations of gas networks (and piping networks in general) are nonlinear and therefore require an iterative solution such as Newton's method or the Newton–Raphson method. Newton's method does not result in any storage difficulty because the equations may be solved sequentially within the iterative scheme. The Newton–Raphson method possesses better convergence properties but, as a matrix method, requires excessive storage and computation unless efficient sparsity techniques are employed. The use of sparsity techniques with the Newton–Raphson method provides the means for the efficient solution of large networks.

In general, it is easier to formulate the nodal equations which inherently possess maximum sparsity, i.e. the number of nonzero elements is a minimum. Because of the nonlinearity of the flow equations, the convergence properties of the nodal formulation are not as good as those of the loop. Although there are fewer equations in the loop formulation, maximum sparsity is not inherent unless special measures are taken to ensure that this is so.

The paper describes a general method of solution of fluid-flow problems which have a nonlinear pipe-flow relationship, with particular reference to low-pressure gas-distribution networks.

The term 'hybrid' is used to describe the method because it is intended to utilise the advantageous properties of both loop and nodal formulations. In doing so, an equivalent electrical network is set to represent the linearised equations of the Jacobian matrix, which is then solved by nodal analysis. This method has been thoroughly tested, and the results for actual gas systems have been obtained. With the collaboration of the North-Western Gas Board (NWGB) a commercial program is now being implemented.

2 Formulation

Any network, whether electrical, structural, gas, or other fluid, can be represented in terms of topological matrices derived from the structure of the network. To illustrate this, consider the gas-distribution network shown in Fig. 1.

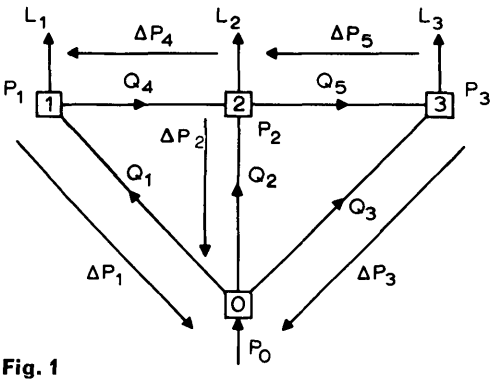

Fig. 1
Distribution network

From Kirchhoff's first law (the algebraic summation of the flow at any given node is zero) the nodal equations of the network are as follows:

$$\left.\begin{array}{rrrrrl} -Q_1 & -Q_2 & -Q_3 & & & = L_0 \\ +Q_1 & & & -Q_4 & & = L_1 \\ & +Q_2 & & +Q_4 & -Q_5 & = L_2 \\ & & +Q_3 & & +Q_5 & = L_3 \end{array}\right\} \quad . \quad . \quad (1)$$

These equations are dependent, and hence one of them is redundant. By eliminating the equation for the reference node, the nodal equations take the following general form:

$$L_k = \sum_{i=1}^{\beta} C_{ki} Q_i \quad (k = 1, \ldots, \alpha) \quad . \quad . \quad . \quad . \quad . \quad (2)$$

where L_k = load at node k
Q_i = flow in branch i
α = number of nodes excluding the reference
β = number of branches

Paper 6537 P, first received 19th April and in revised form 21st July 1971
Dr. Brameller and Mr. Hamam are with the Department of Electrical Engineering & Electronics, University of Manchester Institute of Science & Technology, PO Box 88, Sackville Street, Manchester M60 1QD, England

and

$$C_{ki} = \begin{cases} +1 \text{ if the flow in branch } i \text{ enters node } k \\ -1 \text{ if the flow in branch } i \text{ leaves node } k \\ 0 \text{ if branch } i \text{ is not incident to node } k \end{cases}$$

Eqn. 2 can be represented in matrix form as

$$L_\alpha = C_{\alpha\beta} Q_\beta \quad \ldots \quad (3)$$

where L_α = load vector of dimension α
Q_β = vector of the flow in the branches
$C_{\alpha\beta}$ = nodal connection matrix of α rows and β columns.

The above equations relate the flow in the branches of a network to the loads at the various nodes. In a similar manner, the equations relating the pressure drop across the branches to the nodal pressures are

$$C^T_{\beta\alpha} P_\alpha = \Delta P_\beta \quad \ldots \quad (4)$$

where P_α is the nodal-pressure vector
ΔP_β is the vector of pressure drop across the branches
$C^T_{\beta\alpha}$ is the transpose of $C_{\alpha\beta}$

and the pressure at the reference node is taken to be zero (i.e. atmospheric pressure).

Eqns. 3 and 4 are derived from pure topological considerations, and subsequently they are applicable to any network. Hence, if the variable P_α represents the nodal voltages and L_α the nodal currents, they can be used for electrical networks also.[1]

Another set of equations necessary for the solution of a network are the branch-flow equations. In a fluid-flow system, these equations relate the pressure drop across a given branch to the flow in that branch, while in the electrical network they relate the voltage drop to the current in the branch.

In general,

$$\Delta P_i = \phi(Q_i) \quad i = 1, \ldots, \beta \quad \ldots \quad (5)$$

or, in vector form,

$$\Delta P_\beta = \phi_\beta(Q_\beta) \quad \ldots \quad (6)$$

Conversely, Q_β can be expressed as a function of ΔP_β, which yields

$$Q_\beta = \phi'_\beta(\Delta P_\beta) \quad \ldots \quad (7)$$

For an electrical network this equation is linear and obeys Ohm's law. For gas networks and other fluid-flow problems, the equation is nonlinear.

2.1 Nodal analysis

The nodal-analysis equations of the solution are derived by substituting eqns. 4 and 7 into eqn. 3, which gives

$$L_\alpha = C_{\alpha\beta} \phi'_\beta(C^T_{\beta\alpha} P_\alpha) \quad \ldots \quad (8)$$

There are various methods of solution, such as the Gauss–Seidel[3] method, which is commonly used in the electrical load flow. In general, for a fluid system, the method of solution depends on the complexity of the equations. In gas systems, the solution by nodal analysis is possible using the Newton–Raphson[2, 6] method, but can result in convergence difficulties owing to the presence of square-root terms in the equations. However, it has the advantage of using a highly sparse Jacobian matrix. An alternative approach is to use the loop method.

2.2 Loop analysis

The equations for loop analysis are derived from Kirchhoff's second law (the pressure drop around any given loop is zero). For the example shown in Fig. 1, the two loop equations are

$$\Delta P_3 + \Delta P_2 - \Delta P_1 = 0$$
$$\Delta P_4 - \Delta P_5 + \Delta P_1 = 0$$

In general form, the loop equations can be written

$$\sum_{j=1}^{\beta} D_{ij} \Delta P_j = 0 \quad (i = 1, \ldots, \gamma) \quad \ldots \quad (9)$$

where $D_{ij} = \begin{cases} 1 \text{ if branch } j \text{ is in the same direction as loop } i \\ -1 \text{ if branch } j \text{ is in the opposite direction} \\ 0 \text{ if branch } j \text{ is not in loop } i \end{cases}$

$\gamma = \beta - \alpha$ = number of loops in the network

In matrix form, this can be expressed as

$$D_{\gamma\beta} \Delta P_\beta = 0_\gamma \quad \ldots \quad (10)$$

Substituting eqn. 6 into eqn. 10, the loop equations are expressed in terms of the branch flows as

$$D_{\gamma\beta} \phi_\beta(Q_\beta) = 0_\gamma \quad \ldots \quad (11)$$

The equation of the solution for the loop analysis is obtained by combining eqns. 3 and 11. Consider eqn. 3, which can be partitioned as

$$L_\alpha = [C_{\alpha\alpha} \mid C_{\alpha\gamma}] \begin{bmatrix} Q_\alpha \\ \hline Q_\gamma \end{bmatrix} \quad \ldots \quad (12)$$

or

$$L_\alpha = C_{\alpha\alpha} Q_\alpha + C_{\alpha\gamma} Q_\gamma \quad \ldots \quad (13)$$

Rearrangement of the above equation gives

$$Q_\alpha = C^{-1}_{\alpha\alpha} L_\alpha - C^{-1}_{\alpha\alpha} C_{\alpha\gamma} Q_\gamma \quad \ldots \quad (14)$$

Thus it is possible to solve for Q_α provided that the inverse of $C_{\alpha\alpha}$ exists. This can be ensured by forming a tree (i.e. minimum number of branches connecting all the nodes) for the system, where $C_{\alpha\alpha}$ is the nodal-incidence matrix for the tree branches and $C_{\alpha\gamma}$ is that for the cotree, i.e. the loop-forming branches.

Combining eqns. 14 and 11 (see Appendix 11.1) yields the loop equations

$$D_{\gamma\beta} \phi_\beta(Q^0_\beta + D^T_{\beta\gamma} Q_\gamma) = 0 \quad \ldots \quad (15)$$

where $Q^0_\beta = \begin{bmatrix} -C^{-1}_{\alpha\alpha} L_{\alpha\gamma} \\ \hline 0_\gamma \end{bmatrix}$ $\ldots$ (16)

and $D^T_{\beta\gamma} = \begin{bmatrix} -C^{-1}_{\alpha\alpha} C_\alpha \\ \hline U_{\gamma\gamma} \end{bmatrix}$ $\ldots$ (17)

Eqn. 15 is the equation of the loop-analysis formulation. In gas networks, the equations are quadratic, resulting in good convergence properties.

2.3 Low-pressure gas networks

Low-pressure gas networks consist of supply and load nodes interconnected by pipes, with the pressure at the supply nodes fixed and independent of the flow. Thus, a supply point can be represented by a branch connecting the supply node to the reference node (atmosphere with zero pressure) with the flow equation

$$\phi(Q_i) = \Delta P_i = \text{constant (for all supplies)} \quad \ldots \quad (18)$$

This is shown schematically in Fig. 2*a*.

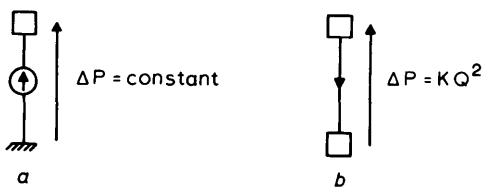

Fig. 2
Network elements

The loads are fixed and are assumed to be independent of the nodal pressures. The pipe-flow equations used in the gas-network analysis can be approximated to the form

$$\Delta P_k = P_i^2 - P_j^2 = K_k Q_k^n \text{ (for medium pressure)}$$
$$\Delta P_k = P_i - P_j = K_k Q_k^n \text{ (for low pressure)}$$

or in general form as

$$\phi(Q_k) = \Delta P_k = K_k Q_k^n \text{ (for all pipes)} \quad \ldots \quad (19)$$

where n is a constant; for the low-pressure gas network $n = 2$ is usually considered as a good approximation.

To illustrate the various system parameters, a schematic diagram is given in Fig. 3.

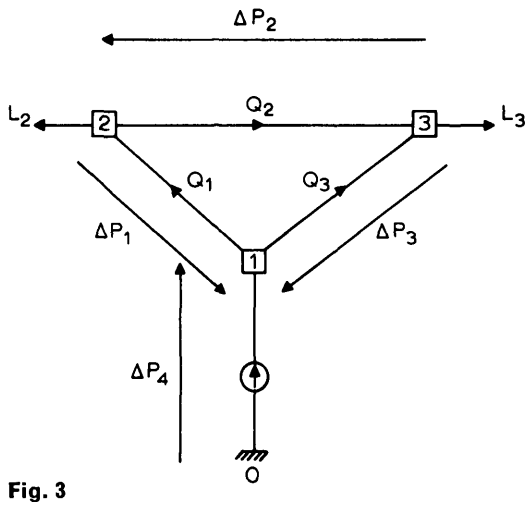

Fig. 3
Typical gas network

3 Method of solution

The procedure of solution used is the Newton–Raphson method, of which the theory is well known.[3] It consists essentially in solving a set of nonlinear equations, $\boldsymbol{F}_\gamma(\boldsymbol{Q}_\gamma) = \boldsymbol{0}$, by an iterative scheme. Thus at the rth iteration, the next estimate for $\boldsymbol{Q}_\gamma^r$ is obtained from

$$\boldsymbol{Q}_\gamma^{r+1} = \boldsymbol{Q}_\gamma^r - \mathscr{J}_{\gamma\gamma}^r \boldsymbol{F}_\gamma(\boldsymbol{Q}_\gamma^r) \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (20)$$

or $$\Delta\boldsymbol{Q}_\gamma^r = \mathscr{J}_{\gamma\gamma}^r \boldsymbol{F}_\gamma(\boldsymbol{Q}_\gamma^r) \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (21)$$

where $\mathscr{J}_{\gamma\gamma} = \boldsymbol{J}_{\gamma\gamma}^{-1}$

$$\Delta\boldsymbol{Q}_\gamma^r = \boldsymbol{Q}_\gamma^r - \boldsymbol{Q}_\gamma^{r+1} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (22)$$

and $\boldsymbol{J}_{\gamma\gamma}$ is the Jacobian matrix of the first partial derivatives of $\boldsymbol{F}_\gamma(\boldsymbol{Q}_\gamma)$ with respect to $\boldsymbol{Q}_\gamma$.

The above procedure can be used to solve eqn. 15, i.e.

$$\boldsymbol{D}_{\gamma\beta}\boldsymbol{\phi}_\beta(\boldsymbol{Q}_\beta^0 + \boldsymbol{D}_{\beta\gamma}^T\boldsymbol{Q}_\gamma) = \boldsymbol{0}$$

The Jacobian matrix for the above equation is

$$\boldsymbol{J}_{\gamma\gamma} = \boldsymbol{D}_{\gamma\beta}\frac{\partial\boldsymbol{\phi}_\beta}{\partial\boldsymbol{Q}_\beta}\boldsymbol{D}_{\beta\gamma}^T \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (23)$$

with $$\boldsymbol{Q}_\beta = \boldsymbol{Q}_\beta^0 + \boldsymbol{D}_{\beta\gamma}^T\boldsymbol{Q}_\gamma \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (24)$$

and

$$\frac{\partial\boldsymbol{\phi}_\beta}{\partial\boldsymbol{Q}_\beta} = \begin{bmatrix} \frac{d\phi_1}{dQ_1} & & 0 \\ & \ddots & \\ 0 & & \frac{d\phi_\beta}{dQ_\beta} \end{bmatrix} \quad . \quad . \quad . \quad . \quad . \quad . \quad (25)$$

Substituting eqns. 23 and 24 into eqn. 21, and rearranging the terms, yields

$$\boldsymbol{D}_{\gamma\beta}\frac{\partial\boldsymbol{\phi}_\beta}{\partial\boldsymbol{Q}_\beta}\boldsymbol{D}_{\beta\gamma}^T\Delta\boldsymbol{Q}_\gamma^r = \boldsymbol{F}_\gamma^r \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (26)$$

As mentioned previously, it is desirable when using this approach to define the loops so that the Jacobian matrix has maximum sparsity. However, to find the optimum loops is a difficult task which may require a great deal of computing time. After a series of investigations, a combination of the nodal and loop approach has been developed whose algorithm is described in Section 5.

The pipe-network equations are inherently better conditioned when expressed in terms of loop analysis, and therefore for large networks it is imperative to preserve this property. However, to enable an efficient formulation of the equations to give maximum sparsity and hence minimum storage and computation time in solving the equations, a nodal analysis is utilised. This is done by transforming the loop equations of the Newton–Raphson method to equivalent nodal equations using the electrical model.

4 Electrical analogy

By analogy, it is possible to build a d.c. electrical network whose loop equations are identical to eqns. 26. This is achieved by replacing every tree branch by a resistive branch and every cotree branch by a resistance in series with an e.m.f. Since the flow equation for the electrical network obeys Ohm's law, the following equations are evident:

(*a*) for tree branches

$$\phi(I_i) = R_i I_i \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (27)$$

(*b*) for cotree branches

$$\phi(I_j) = E_j + R_j I_j \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (28)$$

Furthermore, since the injected currents are zero (i.e. $\boldsymbol{L}_\alpha = 0$) eqn. 15 reduces to

$$\boldsymbol{D}_{\gamma\beta}\boldsymbol{\phi}_\beta(\boldsymbol{D}_{\beta\gamma}^T\boldsymbol{I}_\gamma) = \boldsymbol{0} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (29)$$

By substituting eqns. 27 and 28 into eqn. 29, the loop equations of the electrical network become

$$\boldsymbol{D}_{\gamma\beta}\boldsymbol{R}_{\beta\beta}\boldsymbol{D}_{\beta\gamma}^T\boldsymbol{I}_\gamma = \boldsymbol{E}_\gamma \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (30)$$

If $\boldsymbol{R}_{\beta\beta}$ is set equal to $\partial\boldsymbol{\phi}_\beta/\partial\boldsymbol{Q}_\beta$, and if $\boldsymbol{E}_\gamma$ is set equal to $\boldsymbol{F}_\gamma$ of eqn. 26, then $\boldsymbol{I}_\gamma$ becomes equivalent to $\Delta\boldsymbol{Q}_\gamma$. The resistance of any given branch i in the electrical analogue takes the value

$$r_i = \frac{\partial\phi_i}{\partial Q_i} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (31)$$

Hence, for the supply branches,

$$r_i = 0 \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (32)$$

and, for the pipe branches,

$$r_i = 2K_i|Q_i| \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (33)$$

The e.m.f. in any given cotree branch is equal to the error in pressure around the corresponding loop. Furthermore, since the resistance between any supply node to ground of the electrical analogue is zero (eqn. 32), all the supply nodes are connected directly to the reference node. Subsequently, the number of the nodes in the equivalent network will be reduced.

The relationship between the voltage drop and the current in the branches of the electrical network is obtained from eqns. 27 and 28. Thus, for the tree branches,

$$\Delta\boldsymbol{V}_\alpha = \boldsymbol{R}_{\alpha\alpha}\boldsymbol{I}_\alpha \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (34)$$

or $$\boldsymbol{I}_\alpha = \boldsymbol{R}_{\alpha\alpha}^{-1}\Delta\boldsymbol{V}_\alpha \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (35)$$

For the cotree branches,

$$\Delta\boldsymbol{V}_\gamma = \boldsymbol{R}_{\gamma\gamma}\boldsymbol{I}_\gamma + \boldsymbol{E}_\gamma$$

On rearranging, this becomes

$$\boldsymbol{I}_\gamma = \boldsymbol{R}_{\gamma\gamma}^{-1}(\Delta\boldsymbol{V}_\gamma - \boldsymbol{E}_\gamma)$$

or $$\boldsymbol{I}_\gamma = \boldsymbol{R}_{\gamma\gamma}^{-1}\Delta\boldsymbol{V}_\gamma - \boldsymbol{I}_\gamma' \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (36)$$

where $\boldsymbol{I}_\gamma' = \boldsymbol{R}_{\gamma\gamma}^{-1}\boldsymbol{E}_\gamma$

Combining eqns. 35, 36 and 8, the nodal equation of the electrical network becomes

$$\boldsymbol{Y}_{\alpha\alpha}\boldsymbol{V}_\alpha = \boldsymbol{I}_\alpha' \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (37)$$

where $\boldsymbol{Y}_{\alpha\alpha} = \boldsymbol{C}_{\alpha\beta}\boldsymbol{G}_{\beta\beta}\boldsymbol{C}_{\beta\alpha}^T$

and

$$\boldsymbol{G}_{\beta\beta} = \left[\begin{array}{c|c} \boldsymbol{R}_{\alpha\alpha}^{-1} & \\ \hline & \boldsymbol{R}_{\gamma\gamma}^{-1} \end{array}\right] = \begin{bmatrix} \frac{1}{r_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{r_\beta} \end{bmatrix}$$

and

$$\boldsymbol{I}_\alpha' = \boldsymbol{C}_{\alpha\beta}\left[\begin{array}{c} \boldsymbol{0}_\alpha \\ \hline \boldsymbol{I}_\gamma' \end{array}\right] \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (38)$$

The solution of eqn. 31 gives the voltages and currents of the equivalent electrical network, and hence the corrections to the flows in the original gas network can be obtained.

The three steps of the analogy are illustrated in Fig. 4. It is important to stress at this point that the steps of the solution are those of the Newton–Raphson loop analysis (i.e. the correction of the flows are those that would be obtained if pure loop analysis has been used). It is worth noting that eqns. 30 and 36 are only a restatement of Norton's theorem for converting voltage sources to current sources.

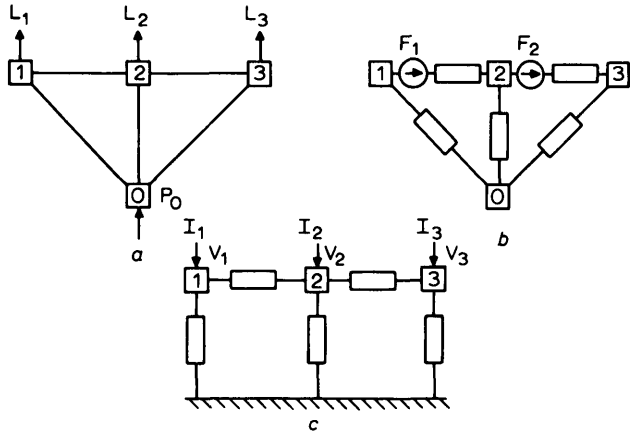

Fig. 4

Equivalent electrical representation of gas network

a Original network

$D_{\gamma\beta}\phi_\beta(Q^0_\beta + D^T_{\beta\gamma}Q_\gamma) = 0_\gamma$

b Incremental network

$J_{\gamma\gamma}\Delta Q_\gamma = F_\gamma$

c Electrical analogue

$Y_{\alpha\alpha}V_\alpha = I'_\alpha$

5 Initial estimate

The Newton–Raphson method is well known to give speedy convergence provided a good initial estimate is available. However, it is generally difficult to have a close initial estimate to start the solution. One way of finding the initial values is to define a tree for the system (a method for finding trees is outlined in Appendix 11.2). Initial values are obtained by equating to zero the flows in all the cotree branches, and then flows in the tree branches are obtained from eqn. 14

$$Q^0_\alpha = C^{-1}_{\alpha\alpha}L_\alpha \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (39)$$

In actual programming, it is not necessary to invert $C_{\alpha\alpha}$, but Q_α can be obtained by tracing the tree from each load node towards the reference. Nevertheless, the branches with zero flow will result in zero resistance in the equivalent electrical network, and hence its admittance is infinite (eqn. 33). One way of overcoming this difficulty is to assign arbitrarily large values to represent the branch admittances. In practice, these arbitrarily large values may cause an ill-conditioned admittance matrix which gives large rounding errors during matrix inversion. An alternative approach is to establish an initial equivalent electrical network whose branch resistances are all equal to the coefficient of friction K of the corresponding pipes and with nodal currents equal to the loads. The branch currents thus calculated are taken as the initial values. In order to eliminate the possible rounding errors, the flows in the tree branches are then recalculated from eqn. 14

$$Q_\alpha = C^{-1}_{\alpha\alpha}(L_\alpha - C_{\alpha\gamma}Q_\gamma)$$

The above approach gives a good initial estimate, and so far no divergence of the algorithm has been encountered.

6 Solution algorithm

The various steps of the algorithm are:

(*a*) define a tree for the network (Appendix 11.2)
(*b*) obtain the initial estimates as discussed in Section 5
(*c*) by tracing the tree starting from the reference node, find the pressures at the load nodes
(*d*) the errors in the loop equation can then be calculated by considering the cotree branches. For a pipe k, corresponding to a cotree branch, connected to nodes i and j, the error around the corresponding loop is

$$F_k = P_i - P_j - K_kQ^2_k$$

The e.m.f.s E_γ are then obtained from

$$E_\gamma = F_\gamma$$

Hence, the nodal injected currents are

$$I'_\alpha = C_{\alpha\beta}\begin{bmatrix} 0_\alpha \\ \hline R^{-1}_{\gamma\gamma}E_\gamma \end{bmatrix}$$

(*e*) eqn. 37, $Y_{\alpha\alpha}V_\alpha = I'_\alpha$, is then solved to obtain the nodal voltages of the equivalent electrical network. The nodal-admittance matrix is highly sparse, and hence a compact storage scheme using ordered elimination is used. The method of ordering adopted in the program is the one suggested by Zollenkopf.[4] A comparison of various methods of ordering is given in Section 7.

7 Results

Several actual low-pressure gas networks have been solved on the Atlas, 1905F and NWGB 4–70 computers. The results obtained at the Atlas computer for five different networks are shown in Table 1.

Table 1

COMPARISON OF SOLUTION TIME AND STORAGE FOR DIFFERENT NETWORKS USING THE ATLAS COMPUTER

System	1	2	3	4	5
	Size number				
Pipes	135	597	784	1017	1260
Nodes	113	482	621	782	915
Loops	22	115	163	235	345
Governors (supply nodes)	4	5	4	14	8
	Computation time				
	s	s	s	s	s
Reading data	—	—	2	3	3
Compacting and initial estimates	—	—	1	1	2
Simulation and elimination	—	—	7	10	14
Total iteration time	—	—	11	15	16
Total computing time	2	12	19	26	32
Printing results	8	35	40	56	70
Storage: real words	1100	5000	6600	8400	10400
integer words	1800	8000	10800	13500	16800

The method gives speedy convergence, since none of the systems solved required more than 7 iterations to give a loop error of less than 0·001% of the pressure at the main supply node.

Fig. 5*a* shows the total computation time for the solution of different sizes of gas-distribution networks using dynamic ordering, and illustrates an almost linear relationship. The total computing time for different system sizes obeys, for the Atlas computer, the approximate relationship

$$t = 0{\cdot}0034b^{1{\cdot}28}$$

where t = time, s

b = total number of branches

This linearity is partly due to the dynamic ordering method[4] used and partly due to the independence of the number of iterations required for convergence for different system sizes.

Fig. 5 shows also the effect of different ordering approaches on the total computation time for network analysis. Curve *a* shows the use of dynamic ordering method, curve *b* the static ordering and curve *c* the natural (the order of the elimination depends on the original numbering of the nodes). It has been shown[5] that other algorithms of higher order do not give significant extra reduction of nonzero elements when compared with Zollenkopf's technique, and are expensive in the

computation time needed for the ordering. From these results, it is the opinion of the authors that the dynamic ordering used is the most suitable for large gas networks.

The program developed is general and has been used also for the solution of a nuclear reactor gas-cooling system. This system is characterised by regular mesh structure consisting of 122 branches and 72 nodes and has been solved in under 5s on the Atlas computer.

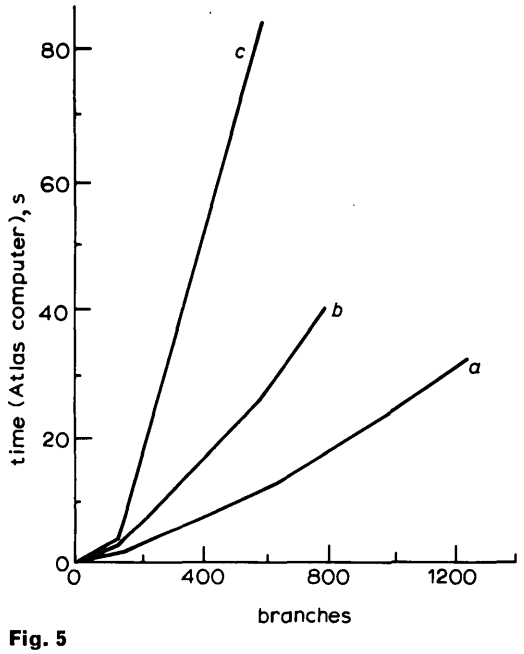

Fig. 5

Total computation time for the solution of gas networks using different ordering methods

a With dynamic ordering
b With static ordering
c With natural ordering

8 Conclusions

In recent years, the solution of gas networks has increasingly made use of methods which are based on the loop formulation. This is principally because this formulation results in equations which are better conditioned, but also because good initial estimates are most readily obtained within the solution algorithm when using loop methods.

The nodal formulation does not lend itself readily to obtaining good starting values, nor does it possess the convergent properties of the mesh equations. It does, however, inherently ensure maximum sparsity, even though there are more equations.

The hybrid method developed in this paper is intended to combine

(*a*) the good convergent properties of the Newton–Raphson method
(*b*) the better conditioning properties and starting values of the loop formulation
(*c*) the inherent sparsity of the nodal formulation.

The success of the method depends directly on taking advantage of the sparsity in an efficient manner. Tests on comparatively large, actual systems have shown stable and fast convergence and no systems have been encountered which have failed.

The method is general and suitable for other fluid networks with nonlinear flow properties.

9 Acknowledgments

The authors wish to thank the North-Western Gas Board for facilities, interest and support, and for permission to publish the details of the investigations. They are also grateful for the facilities provided by the Power System Laboratory at UMIST.

They also wish to extend their thanks to the Nuclear Power Group for providing the data for the nuclear reactor gas-cooling system.

10 References

1 BRAMELLER, A., JOHN, M. N., and SCOTT, M. R.: 'Practical diakoptics for electrical networks' (Chapman & Hall, 1969)
2 HAMAM, Y. M.: 'Gas flow analysis'. M.Sc. dissertation, University of Manchester Institute of Science & Technology, 1970
3 STAGG, G. W., and AL-ABIAD, A. H.: 'Computer methods in power systems analysis' (McGraw-Hill, 1968)
4 ZOLLENKOPF, K.: 'Bi-factorisation'. Conference on large sets of linear equations, Oxford, April 1970
5 STOTT, B., and HOBSON, E.: 'Solution of large power-system networks by ordered elimination: a comparison of ordering schemes', *Proc. IEE*, 1971, **118**, (1), pp. 125–134
6 BRAMELLER, A., CHANCELLOR, V. E., HAMAM, Y. M., and YALCINDAG, C.: 'Recent advances of mathematical analysis for gas industry applications'. Institution of Gas Engineers communication 837, 1970

11 Appendixes

11.1 Derivation of the loop equations

Starting from eqn. 4,

$$C^{T}_{\beta\alpha}P_{\alpha} = \Delta P_{\beta} \quad \ldots\ldots\ldots\ldots \quad (40)$$

$C^{T}_{\beta\alpha}$ is partitioned in the same way as its transpose ($C_{\alpha\beta}$) in eqn. 12, giving

$$\begin{bmatrix} C^{T}_{\alpha\alpha} \\ \hline C^{T}_{\gamma\alpha} \end{bmatrix} P_{\alpha} = \begin{bmatrix} \Delta P_{\alpha} \\ \hline \Delta P_{\gamma} \end{bmatrix} \quad \ldots\ldots\ldots\ldots \quad (41)$$

i.e. $$C^{T}_{\alpha\alpha}P_{\alpha} = \Delta P_{\alpha} \quad \ldots\ldots\ldots\ldots \quad (42)$$

$$C^{T}_{\gamma\alpha}P_{\alpha} = \Delta P_{\gamma} \quad \ldots\ldots\ldots\ldots \quad (43)$$

Substituting eqn. 42 into 43 and letting

$$B_{\alpha\gamma} = -C^{-1}_{\alpha\alpha}C_{\alpha\gamma}$$

gives $$B^{T}_{\gamma\alpha}\Delta P_{\alpha} + \Delta P_{\gamma} = 0_{\gamma}$$

or $$[B^{T}_{\gamma\alpha} \mid U_{\gamma\gamma}] \begin{bmatrix} \Delta P_{\alpha} \\ \hline \Delta P_{\gamma} \end{bmatrix} = 0_{\gamma} \quad \ldots\ldots\ldots\ldots \quad (44)$$

or $$D_{\gamma\beta}\Delta P_{\beta} = 0_{\gamma} \quad \ldots\ldots\ldots\ldots \quad (45)$$

Eqn. 45 is identical to eqn. 10, obtained from Kirchhoff's second law in Section 2.2, since

$$D_{\gamma\beta} = [B^{T}_{\gamma\alpha} \mid U_{\gamma\gamma}]$$

Hence

$$D^{T}_{\beta\gamma} = \begin{bmatrix} B_{\alpha\gamma} \\ \hline U_{\gamma\gamma} \end{bmatrix}$$

Furthermore, it is necessary to express ΔP_{β} in eqn. 45 in terms of the flow in the cotree branches. Using the flow equation, ΔP_{β} can be expressed in terms of Q_{β} as in eqn. 6. It remains to express Q_{β} in terms of Q_{γ}. Consider eqn. 13, i.e.

$$Q_{\alpha} = C^{-1}_{\alpha\alpha}L_{\alpha} + B_{\alpha\gamma}Q_{\gamma}$$

This can be augmented to include Q_{γ}, as

$$\begin{bmatrix} Q_{\alpha} \\ \hline Q_{\gamma} \end{bmatrix} = \begin{bmatrix} C^{-1}_{\alpha\alpha}L_{\alpha} \\ \hline 0_{\gamma} \end{bmatrix} + \begin{bmatrix} B_{\alpha\gamma} \\ \hline U_{\gamma\gamma} \end{bmatrix} Q_{\gamma} \quad \ldots\ldots \quad (46)$$

or $$Q_{\beta} = Q^{0}_{\beta} + D^{T}_{\beta\gamma}Q_{\gamma} \quad \ldots\ldots\ldots\ldots \quad (47)$$

where $$Q^{0}_{\beta} = \begin{bmatrix} C^{-1}_{\alpha\alpha}L_{\alpha} \\ \hline 0_{\gamma} \end{bmatrix} \quad \ldots\ldots\ldots\ldots \quad (48)$$

Thus, substituting for Q_{β} in eqn. 11 gives

$$D_{\gamma\beta}\phi_{\beta}(Q^{0}_{\beta} + D^{T}_{\beta\gamma}Q_{\gamma}) = 0_{\gamma} \quad \ldots\ldots \quad (49)$$

11.2 Tree construction

Different techniques exist for the tree construction, and the one used in the method described is as follows:

(*a*) A list is formed of the nonzero elements of the connection matrix $C_{\alpha\beta}$ with their row and column locations.
(*b*) The tree is started by connecting branches from the reference node (atmosphere), and the corresponding nodes are entered into the tree list.

(*c*) The branches connected to the nodes just entered into the tree are found (branches already entered in the tree are bypassed).
(*d*) The nodes at the other end of the branches just considered are found. The branches for which the nodes have already been connected to the tree go into the cotree. The branches for which the nodes have not been entered go into the tree.
(*e*) The new nodes are entered into the tree and step *c* is performed until all the nodes are in the tree.

In the actual program, the tree is constructed as a series of subtrees, with every subtree formed as shown above but with one supply node.

11.3 Ordered elimination

The object of the ordered-elimination technique is the solution of the sparse matrix equation

$$A_{\alpha\alpha}X_\alpha = b_\alpha \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (50)$$

For large systems the solution of this equation is not practicable if the sparsity property is not exploited. The value of X_α can be obtained by inversion of $A_{\alpha\alpha}$, i.e.

$$X_\alpha = A_{\alpha\alpha}^{-1}b_\alpha \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (51)$$

If the inverse of $A_{\alpha\alpha}$ is computed explicitly, it gives a full matrix. Fortunately, the sparsity can be retained if the inverse is expressed as a series of elementary transformation matrices. One method of doing this is the bifactorisation[4] method. In this method, the matrix $A_{\alpha\alpha}$ is reduced to a unit matrix as follows

$$L_nL_{n-1}\ldots L_2L_1A_{\alpha\alpha}R_1R_2\ldots R_{n-1}R_n = U_{\alpha\alpha} \quad (52)$$

$$L_k = \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \vdots \\ & & \ddots & & & & \vdots \\ & & & \frac{1}{a_{kk}} & & & \\ & & & \vdots & \ddots & & \\ & & & \vdots & & \ddots & \\ & & & \frac{-a_{nk}}{a_{kk}} & & & 1 \end{bmatrix} \qquad R_k = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ 0 & \ldots & 1 & \ldots & \frac{-a_{kn}}{a_{kk}} & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

It can be seen that the transformation matrices consist of one nonzero column or row. These columns and rows are, however, very sparse and thus, by using a compact storage scheme, the overall storage requirement is immensely reduced. However, during the reduction process, new elements are generated. These elements will increase in number enormously for large networks if the natural order of the elimination is adopted.

Many different alogrithms have been suggested for ordering the elimination. Two of these algorithms have been tested for large gas networks, and their results compared with the natural ordering.

The first is a preordering scheme where the order of elimination is established at the beginning from the degree of interconnection at every node. The nodes are numbered in ascending order, starting with the one which has the least number of branches incident to it.

The second algorithm is that put forward by Zollenkopf. In his scheme, at each reduction step the column with the least number of nonzero elements is selected as a pivotal column.