INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 37, 3685–3705 (1994)

A SEGREGATED CFD APPROACH TO PIPE NETWORK ANALYSIS

G. P. GREYVENSTEIN

Department of Mechanical Engineering, Potchefstroom University for Christian Higher Education, Private Bag X6001, Potchefstroom, 2520 South Africa

D. P. LAURIE

Department of Mathematics and Applied Mathematics, Potchefstroom University for Christian Higher Education, P.O. Box 1174, Vanderbijlpark, 1900 South Africa

SUMMARY

The most popular pipe network algorithms fall into three categories depending on whether node, loop or element solving equations are considered. Although node methods have some advantages over the other two methods, some authors have found the node methods to be more unreliable than the other two classes of methods. Node methods are also used in Computational Fluid Dynamics (CFD) to solve the Navier–Stokes equations. Since significant progress has been made in this field in the recent past it was felt that this should have some bearing on the development of more reliable node methods for pipe network problems. In this paper the well-known SIMPLE algorithm of Patankar and Spalding,[1] which is known in CFD as a segregated method, is extended to deal with pipe network problems. The method can deal with both compressible and incompressible flows. Special attention is given to the solution of the pressure correction equation, the stability of the algorithm, sensitivity to initial conditions and convergence parameters. It is shown that the present method is not very sensitive to initial conditions. The method is very reliable and it deals more effectively with compressible flows than the conventional Newton–Raphson method for incompressible flows.

1. INTRODUCTION

The steady-state analysis of flow and pressure in pipe networks is required when designing water, gas or compressed air distribution systems; chemical process plants or airconditioning systems. Pipe network analysis is therefore of great importance in engineering.

The first methods for solving pipe network problems were published by Hardy Cross in 1936.[2] Cross described two methods: one for solving the loop equations based on adjusting flow rates to individually balance the pressure drops around a closed loop and one for solving the node equations by adjusting the pressure at each node so that the continuity equation is satisfied. With these methods, adjustments are computed independent from each other, with the result that convergence was often slow or non-existent in some cases.[3–5] This shortcoming has triggered the search for more efficient procedures and many improvements to the original Hardy-Cross method have been proposed through the years.

With the advent of the digital computer, interest in network solving algorithms has grown not only because it has become possible to solve large networks on computers, but also because of the possibility to optimize designs.[6]

The most popular network solving algorithms fall into three categories depending on whether node, loop or element solving equations are considered.[4,7] Methods based on the loop solving

CCC 0029–5981/95/213685–21$9.00

Received 15 September 1992
Revised 18 February 1994

equations appeal to many people because they require less computer storage than methods based on the node and element equations.[4] A disadvantage of these methods is that the initial flow distribution must satisfy continuity. Also, the setting up of the different loops in the most efficient way for the solution algorithm is not a trivial task.[8] The advantage of methods based on the element solving equations is that it is not necessary to specify initial flows that satisfy continuity. These methods also converge very fast.[4] A disadvantage of the element methods is that they require more computer storage than the node and loop methods. Element methods also require the specification of a number of loops.

The advantage of the methods based on the node equations is that they require less computer storage than the element methods and it is also not necessary to specify loops or initial flows as is the case with the loop methods. The node methods can also deal with mixed boundary conditions (pressure or flow), which adds to the flexibility of the method. Wood and Rayes,[3] who have investigated the reliability of five methods, have found the node methods to be more unreliable than the loop and element methods. The unreliability of the loop methods stems from the following:

(a) the sensitivity of the methods to initial pressures, especially when line losses vary greatly or pumps operate on a very steep curve;
(b) the inability of the methods to deal with low resistance lines.

Wood and Rayes,[3] however, reported that when the simultaneous node method is successful, a highly accurate solution is obtained in relatively few trials.

Node methods are also used in CFD to solve the Navier–Stokes equations. Since significant progress has been made in the last couple of years in this field it was felt that these developments should have some bearing on the development of improved node methods for pipe network analysis.

One of the most well-known node methods is the SIMPLE algorithm developed by Patankar and Spalding.[1] In CFD this method is also described as a segregated method.[9] Although Patankar[10] has proposed that the SIMPLE algorithm be applied to pipe network problems the authors are not aware of any published work in this field.

The original SIMPLE algorithm applies to incompressible flows. An attractive feature of the algorithm is that it can be extended to deal with compressible flows as well.[9] (In this paper compressible flow means subsonic flows with varying densities.) The ability to handle compressible flows in pipe network is important when dealing with compressed air distribution networks.

In this paper the adaptation of the SIMPLE algorithm to compressible flow in pipe networks is discussed. It is shown that for incompressible flow the method is nearly identical to the simultaneous node Newton–Raphson method.[11] Special attention is given to the solution of the pressure correction equation, the stability of the algorithm, the sensitivity to initial conditions and convergence parameters.

The developed method is very robust and the authors have not experienced any convergence problems during the analysis of a wide variety of networks.

2. COMPUTATIONAL MESH

A network consists of a number of oriented elements connected to nodes as shown in Figure 1. Elements are connected to one another by nodes and each element is associated with an upstream and a downstream node. Numbers are used to identify each node and element. For the purpose of the computational scheme the network is divided into a number of meshes. A mesh is a node with all its branch elements and neighbouring nodes as shown in Figure 2.

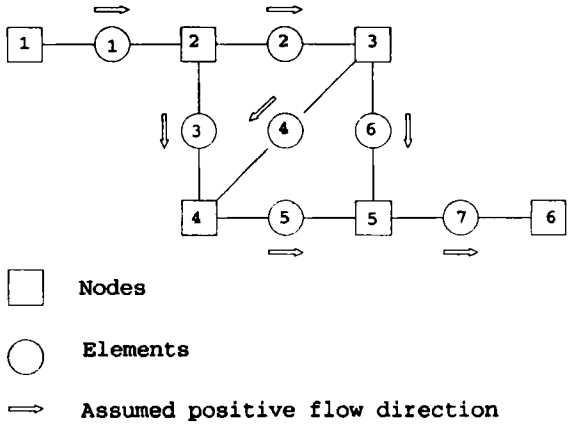

Figure 1. Graphical presentation of a network

In Figure 2 node i is connected through elements e_{ij} to neighbouring nodes n_{ij}, with $j = 1, 2, \ldots, J$. J is the number of branches associated with node i. The indices i, n_{ij} and e_{ij} are global indices while j is a local index. If the assumed positive flow direction for element e_{ij} is towards node i, e_{ij} is assigned a positive value in the element connectivity matrix **E**. Otherwise it is assigned a negative value.

The element and node connectivity matrices for the network shown in Figure 1 are as follows:

$$\mathbf{E} = [e_{ij}] = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & -3 \\ 2 & -4 & -6 \\ 3 & 4 & -5 \\ 5 & 6 & -7 \\ 7 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{N} = [n_{ij}] = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 3 & 4 \\ 2 & 4 & 5 \\ 2 & 3 & 5 \\ 4 & 3 & 6 \\ 5 & 0 & 0 \end{bmatrix}$$

The matrices **E** and **N** uniquely define the structure of the network. In the rest of this paper the subscript ij is used to refer to quantities that apply to element e_{ij}.

3. ALGEBRAIC EQUATIONS

The basic equations that must be satisfied in a network are the continuity equation which applies to every node and the correct relationship between pressure drop and flow rate for every element.

The continuity condition can be expressed as

$$\sum_{j=1}^{J} \rho_{ij} Q_{ij} s_{ij} = -d_i, \quad i = 1, 2, \ldots, I \tag{1}$$

where $s_{ij} = 1$ if $e_{ij} > 0$, $s_{ij} = -1$ if $e_{ij} < 0$ and d_i is the external mass flow into node i.

Different relationships can be used to express the pressure drop as a function of flow rate. Examples are the Darcy–Weisbach equation, the Hazen–Williams equation, the Manning equation, exponential equations or empirically determined relationships.[7] In general, the pressure drop

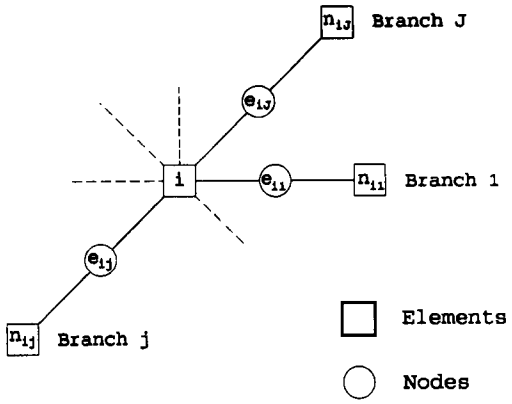

Figure 2. Notation used to define the topology of an arbitrary mesh

flow rate relationship for any of the branch elements of node i (see Figure 2) can be expressed as

$$\Delta p_{ij} = p_{n_{ij}} - p_i = s_{ij} H_{ij} g_{ij} f_{ij} \tag{2}$$

where $g_{ij} = g_{ij}(\rho_{ij})$ and $f_{ij} = f_{ij}(|Q_{ij}|)$ are functions of ρ_{ij} and the absolute value of Q_{ij}, respectively, and $H_{ij} = Q_{ij}/|Q_{ij}|$.

In the case of compressible flow, the element densities ρ_{ij} can be related to the element pressures using an equation of state. For an ideal gas, the relationship is

$$\rho_{ij} = p_{ij}/(RT_{ij}) \tag{3}$$

where T_{ij} is the element temperature and R the gas constant. If dynamic pressures are much smaller than the total pressures, the element pressures will simply be the mean of the two node pressures. Equation (3) then becomes

$$\rho_{ij} = \frac{(p_i + p_{n_{ij}})}{2RT_{ij}} \tag{4}$$

or

$$\rho_{ij} = a_{ij}(p_i + p_{n_{ij}}) \tag{5}$$

where

$$a_{ij} = 1/(2RT_{ij}) \tag{6}$$

The temperature in (6) can be obtained by solving the energy equation.

In an overall iterative solution scheme for compressible flow one usually guesses the temperature field and then solves equations (1), (2) and (5) to obtain flow rates, pressures and densities. During the solution of these equations the temperature at a specific position in the network is assumed to be constant. Temperatures may, however, vary from element to element. After flow rates, pressures and densities have been determined, the energy equation is solved to obtain new updated temperatures. Equations (1), (2) and (5) are then solved again and the whole process repeated until convergence. In this paper the focus is on the simultaneous solution of (1), (2) and (5), given the temperature field, and not on the overall algorithm which includes the solution of the energy equation. The solution of the energy equation is therefore not discussed in this paper.

In the case of incompressible flow the density is constant and the link between the flow equations and the energy equation is very weak. In many cases it suffices to solve the energy equation only once after the flow equations have been solved.

In the next section the simultaneous solution of (1), (2) and (5), given a fixed temperature field, is discussed.

4. COMPUTATIONAL SCHEME

The first step in the computational scheme is to guess pressures at all nodes. These values are treated as tentative values and denoted by p_i^*. If the flow is compressible, the tentative pressures are used to calculate tentative densities ρ_{ij}^* using equation (5).

Next equation (2) is solved for Q_{ij} using the tentative pressures and densities determined previously. These values are also treated as tentative values and denoted by Q_{ij}^* because it is based on tentative pressures and densities.

We now wish to calculate a field of pressure, density and flow rate corrections in such a way that the corrected values better satisfy (1) and (2). The corrected values are calcualted as follows:

$$p = p^* + p' \tag{7}$$

$$Q = Q^* + Q' \tag{8}$$

$$\rho = \rho^* + \rho' \tag{9}$$

where p', Q' and ρ' are the pressure, flow rate and density corrections, respectively. First we have to determine a relationship between pressure correction and flow correction. If (2) is differentiated with respect to Q_{ij}, we get

$$\frac{\partial p_{n_{ij}}}{\partial Q_{ij}} - \frac{\partial p_i}{\partial Q_{ij}} = s_{ij} H_{ij} f_{ij} G_{ij} \frac{\partial \rho_{ij}}{\partial Q_{ij}} + s_{ij} g_{ij} F_{ij} \tag{10}$$

where

$$G_{ij} = \frac{\partial g_{ij}}{\partial \rho_{ij}} \tag{11}$$

and

$$F_{ij} = \frac{\partial f_{ij}}{\partial |Q_{ij}|} \tag{12}$$

If we replace $\partial\rho/\partial Q$ by p'/Q' and $\partial\rho/\partial Q$ by ρ'/Q', we get

$$\frac{p'_{n_{ij}}}{Q'_{ij}} - \frac{p'_i}{Q'_{ij}} = s_{ij} H_{ij} f_{ij} G_{ij} \frac{\rho'_{ij}}{Q'_{ij}} + s_{ij} g_{ij} F_{ij} \tag{13}$$

It follows from (5) that

$$\rho'_{ij} = a_{ij}(p'_i + p'_{n_{ij}}) \tag{14}$$

If (14) is substituted into (13) and the equation is then solved for Q'_{ij}, we get

$$Q'_{ij} = p'_{n_{ij}} \left[\frac{1}{s_{ij} g_{ij}^* F_{ij}^*} - \frac{H_{ij} a_{ij} f_{ij}^* G_{ij}^*}{g_{ij}^* F_{ij}^*} \right] - p'_i \left[\frac{1}{s_{ij} g_{ij}^* F_{ij}^*} + \frac{H_{ij} a_{ij} f_{ij}^* G_{ij}^*}{g_{ij}^* F_{ij}^*} \right] \tag{15}$$

where the superscript $*$ is used to indicate that tentative values of ρ and Q are used to evaluate the

functions, g, G, f and F. We now turn our attention to the continuity equation. The following equation is obtained if Q and ρ in equation (1) are replaced by (8) and (9):

$$\sum_{j=1}^{J} (\rho_{ij}^* Q_{ij}^* + \rho_{ij}' Q_{ij}^* + Q_{ij}' \rho_{ij}^* + \rho_{ij}' Q_{ij}') s_{ij} = -d_i$$

If the term $\rho_{ij}' Q_{ij}'$ is neglected in the above equation, it reduces to

$$\sum_{j=1}^{J} (\rho_{ij}^* Q_{ij}^* + \rho_{ij}' Q_{ij}^* + Q_{ij}' \rho_{ij}^*) s_{ij} = -d_i \tag{16}$$

If (14) and (15) are substituted into (16) we obtain an equation which can be solved for p_i'. The result is

$$p_i' = \left[\sum_{j=1}^{J} (c_{ij} p_{n_{ij}}') + b_i \right] \Big/ c_{ii}, \quad i = 1, 2, \ldots, I \tag{17}$$

where

$$c_{ii} = \sum_{j=1}^{J} \left[\frac{\rho_{ij}^*}{g_{ij}^* F_{ij}^*} + s_{ij} H_{ij} a_{ij} \left(\frac{\rho_{ij}^* f_{ij}^* G_{ij}^*}{g_{ij}^* F_{ij}^*} - |Q_{ij}^*| \right) \right] \tag{18}$$

$$c_{ij} = \frac{\rho_{ij}^*}{g_{ij}^* F_{ij}^*} - s_{ij} H_{ij} a_{ij} \left(\frac{\rho_{ij}^* f_{ij}^* G_{ij}^*}{g_{ij}^* F_{ij}^*} - |Q_{ij}^*| \right) \tag{19}$$

$$b_i = d_i + \sum_{j=1}^{J} (\rho_{ij}^* Q_{ij}^* s_{ij}) \tag{20}$$

The fact that the term $\rho_{ij}' Q_{ij}'$ is omitted in (16) does not affect the accuracy of the solution as both the quantities ρ_{ij}' and Q_{ij}' approach the value zero when convergence is reached.

In the next step in the solution algorithm equation (17) is solved simultaneously for all nodes in the network, after which new updated values for pressure, flow rate and density are calculated using (7), (15), (8), (14) and (9). The new updated values are now treated as new tentative values and the process is repeated until convergence.

5. STABILITY

Stability requires that c_{ij}, given by (19), must always be positive, which means that

$$\frac{\rho_{ij}^*}{g_{ij}^* F_{ij}^*} \geqslant a_{ij} \left(|Q_{ij}^*| - \frac{\rho_{ij}^* f_{ij}^* G_{ij}^*}{g_{ij}^* F_{ij}^*} \right) \tag{21}$$

Consider the following general pressure drop/flow rate relationship:

$$\Delta p = K \rho^{\beta} Q^{\alpha} \tag{22}$$

For the above relationship the functions f and g are given by

$$f = K Q^{\alpha} \tag{23}$$

$$g = \rho^{\beta} \tag{24}$$

The derivatives of f and g are given by

$$F = \alpha K Q^{\alpha - 1} \tag{25}$$

$$G = \beta \rho^{\beta - 1} \tag{26}$$

Substitution of (23)–(26) into (21) leads to the following condition:

$$\rho_{ij}^* \geqslant a_{ij} K_{ij} \rho_{ij}^{*\beta} Q_{ij}^{*\beta} (\alpha - \beta) \tag{27}$$

After substitution of (6) and (22) into (27) and rewriting, we get

$$\frac{\Delta \rho_{ij}^*}{\rho_{ij}^*} \leqslant \frac{2}{\alpha - \beta} \tag{28}$$

where

$$\Delta \rho_{ij} = \Delta p_{ij} / (R T_{ij})$$

In the case of the Darcy–Weisbach equation $\alpha = 2$ and $\beta = 1$. Substitution of these values of α and β into (28) leads to

$$\frac{\Delta \rho_{ij}^*}{\rho_{ij}^*} \leqslant 2 \tag{29}$$

The mean density in an element is calculated as

$$\rho_{ij}^* = (\rho_i^* + \rho_{n_{ij}}^*)/2 \tag{30}$$

and the density difference as

$$\Delta \rho_{ij}^* = \rho_{n_{ij}}^* - \rho_i^* \tag{31}$$

which means that condition (29) will always be satisifed if the pressure drop is calculated with the Darcy–Weisbach equation. It follows from (28), (30) and (31) that stability will always be ensured if $(\alpha - \beta) \leqslant 1$. This is usually true for most of the other pressure drop/flow rate relationships used in practice.

6. SIMILARITY BETWEEN THE PRESENT METHOD AND THE NEWTON–RAPHSON METHOD

According to the Newton–Raphson method, when there are n equations to be satisfied $[h_1(x_1, \ldots, x_n) = 0, \ldots, h_n(x_1, \ldots, x_n) = 0]$ and n unknown $(x_1, \ldots, x_n)$ to be solved for, the set of improvements $(x_1', \ldots, x_n')$ are the solution of the set of n simultaneous linear equations

$$h_i(x_1, \ldots, x_n) + \sum_{j=1}^{n} \frac{\partial h_i}{\partial x_j} x_j' = 0, \quad i = 1, \ldots, n \tag{32}$$

Consider the following form of the continuity equation, equation (1):

$$h_i = \sum_{j=1}^{J} \rho_{ij} Q_{ij} s_{ij} + d_i = 0, \quad i = 1, \ldots, I \tag{33}$$

If the Newton–Raphson method is applied to (33), the following equation is obtained:

$$h_i(p_i, p_{n_{i1}}, \ldots, p_{n_{iJ}}) + \sum_{j=1}^{J} \frac{\partial}{\partial p_i} (\rho_{ij} Q_{ij} s_{ij}) p_i' + \sum_{j=1}^{J} \frac{\partial}{\partial p_{n_{ij}}} (\rho_{ij} Q_{ij} s_{ij}) p_{n_{ij}}', \quad i = 1, \ldots, I \tag{34}$$

Consider the term $\partial / \partial p_{n_{ij}} (\rho_{ij} Q_{ij} s_{ij})$ in (34). This term can be written as

$$\frac{\partial}{\partial p_{n_{ij}}} (\rho_{ij} Q_{ij} s_{ij}) = \rho_{ij} s_{ij} \frac{\partial Q_{ij}}{\partial p_{n_{ij}}} + s_{ij} Q_{ij} \frac{\partial \rho_{ij}}{\partial p_{n_{ij}}} \tag{35}$$

It follows from (2) that

$$1 = s_{ij}\left(g_{ij}F_{ij}\frac{\partial Q_{ij}}{\partial p_{n_{ij}}} + H_{ij}f_{ij}G_{ij}\frac{\partial \rho_{ij}}{\partial p_{n_{ij}}}\right) \tag{36}$$

where F_{ij} and G_{ij} are given by (11) and (12).

It also follows from (5) that

$$\frac{\partial \rho_{ij}}{\partial p_{n_{ij}}} = a_{ij} \tag{37}$$

If (37) is substituted into (36) and the equation is then solved for $\partial Q_{ij}/\partial p_{n_{ij}}$ the following result is obtained:

$$\frac{\partial Q_{ij}}{\partial p_{n_{ij}}} = \frac{1}{s_{ij}g_{ij}F_{ij}} - \frac{H_{ij}f_{ij}G_{ij}a_{ij}}{g_{ij}F_{ij}} \tag{38}$$

Substitution of (37) and (38) as well as $Q_{ij} = H_{ij}|Q_{ij}|$ into (35) leads to

$$\frac{\partial}{\partial p_{n_{ij}}}(\rho_{ij}Q_{ij}s_{ij}) = \frac{\rho_{ij}}{g_{ij}F_{ij}} + s_{ij}H_{ij}|Q_{ij}|a_{ij} - \frac{s_{ij}H_{ij}a_{ij}\rho_{ij}f_{ij}G_{ij}}{g_{ij}F_{ij}} \tag{39}$$

It can also be shown that

$$\frac{\partial}{\partial p_i}(\rho_{ij}Q_{ij}s_{ij}) = -\frac{\rho_{ij}}{g_{ij}F_{ij}} + s_{ij}H_{ij}|Q_{ij}|a_{ij} - \frac{s_{ij}H_{ij}a_{ij}\rho_{ij}f_{ij}G_{ij}}{g_{ij}F_{ij}} \tag{40}$$

Substitution of (39) and (40) into (34) leads to an equation which is identical to (17).

The present method can therefore be classified as a node orientated Newton–Raphson method. It can also be classified as a simultaneous node adjustment (S-NODE) method according to the classification scheme of Wood and Rayes.[3] The present method, however, differs from Newton–Raphson methods described by other workers (i.e. the methods described by Shamir and Howard[11] and Jeppson[7]) in the following respects:

1. Density variations are taken into consideration during the derivation of the pressure correction equation, equation (17). The present method should therefore be able to deal more effectively with compressible flows.
2. With the methods described by Shamir and Howard[11] and Jeppson,[7] the flow is solved explicitly using relationships similar to (2) and then substituted into (34) before the pressure derivatives are evaluated. This presents difficulties when the function f in (2) is a complex function as, for example, in the case of fan or pump curves. With the present method this does not pose a problem as the flow rate can initially be guessed and then updated using (15) and (8).

7. SOLUTION OF THE PRESSURE CORRECTION EQUATION

Wood and Rayes[3] have found the S-NODE methods to be unreliable. They also reported that the methods are sensitive to initial values and that the methods are unable to deal with low resistance lines. For low resistance lines small errors in the calculation of pressure may produce very large errors in the flow rate. According to Wood and Rayes, this is due to the fact that solution algorithms for the node equations do not incorporate an exact continuity balance.

Similar problems are experienced with the SIMPLE algorithm in CFD when the coefficients in the pressure correction equation vary greatly in different co-ordiante directions. A remedy for this

problem is to solve the pressure correction equation very well before one moves on to the next iteration of the overall solution algorithm.

It is important to notice that the linearized continuity equation (16) will be exactly satisfied if (17) is solved exactly. It will therefore greatly enhance the reliability of the S-NODE methods if (17) is solved exactly.

The method used in the present paper to solve (17) is the envelope method as described by George and Liu,[12] which is mathematically identical to the familiar Gaussian elimination method. When (17) is written in matrix form,

$$\mathbf{M}\mathbf{p}' = \mathbf{b} \tag{41}$$

with

$$m_{ii} = c_{ii} \tag{42}$$

$$m_{i,n_{ij}} = -c_{ij} \tag{43}$$

and the other elements of **M** are 0.

Gaussian elimination without row interchanges is equivalent to factoring **M** as

$$\mathbf{M} = \mathbf{L}\mathbf{R} \tag{44}$$

where **L** and **R** are lower and upper triangular matrices, respectively, and then computing the solution in two stages by back-substitution in two triangular systems

$$\mathbf{L}\mathbf{x} = \mathbf{b} \tag{45}$$

and

$$\mathbf{U}\mathbf{p}' = \mathbf{x} \tag{46}$$

Substantial savings in storage and computing time can be achieved if we avoid computing or using quantities that are known in advance to be zero. Much has been written on the topic, and the best algorithms are exhaustively discussed and given as FORTRAN code by George and Liu.[12] A fundamental point is that the closer one tries to get to the most economical storage scheme, the more expensive the analysis of the matrix structure becomes. For example, it is possible to design a scheme that takes advantage of all provable zeros in **L**,[12,6] at the expense of some extra pointer storage and a complicated code. We have chosen the envelope method (a refinement of the band elimination method used by Lemieux[13]) as offering the best trade-off between storage economy and algorithmic overhead.

The essential idea is to notice that all the leading zeros in a particular row of **M** are preserved in **L**. Therefore, only the elements between the first non-zero and the diagonal need be considered. Some of these elements may also actually be zero, but the envelope method treats them as significant. One may think of the envelope method as a band method in which the bandwidth is allowed to vary. In Figures 3 and 4 we show the non-zero pattern of a typical example of a matrix arising in a pipe network problem, and of its lower triangular factor. In conventional Gaussian elimination, the whole lower triangle (3160 elements in the case of an 80×80 matrix) would be computed and stored, but in the envelope method only 663 elements are needed.

Although this is already a substantial improvement, in many cases (including this example) storage and computation time can be further reduced by numbering the nodes in a different way. Again, there is a law of diminishing returns when one tries to find an optimal ordering. Three renumbering schemes are discussed by Chandrashekar and Steward:[6] the method actually recommended by these authors is extremely costly (they quote a computer run on a network with 191 nodes and 287 pipes where the reordering takes 30 times as much as the actual solution of the

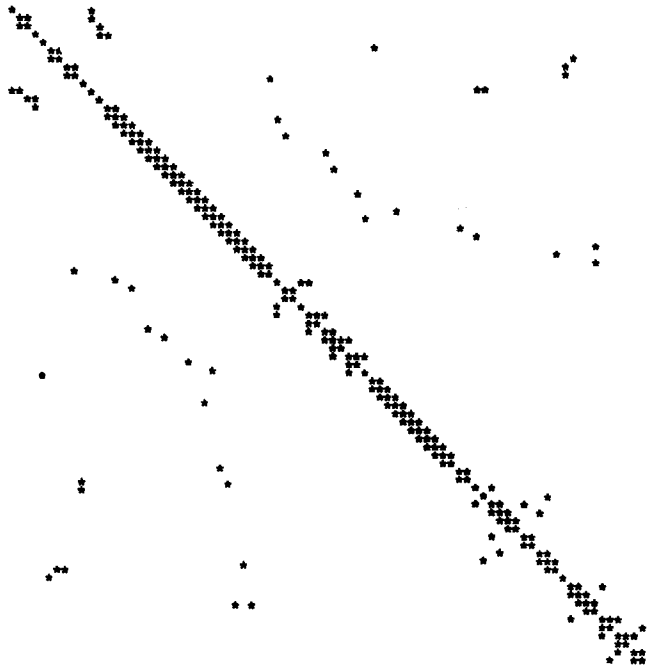

Figure 3. Structure of the matrix arising from 80 nodes and 88 pipes

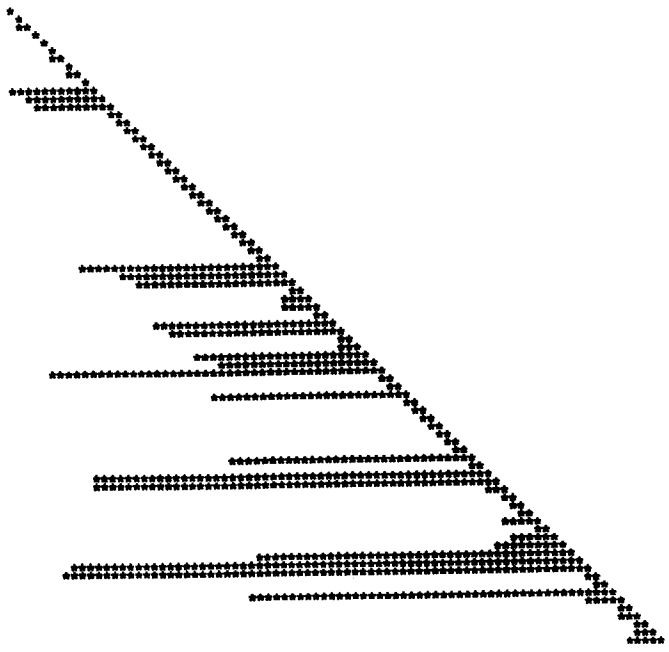

Figure 4. Structure of the lower triangular factor when the envelope method is applied to the matrix in Figure 3. There are 663 significant entries

linear system!). We have followed George and Liu[12] in choosing the reverse Cuthill–McKee algorithm, which can be summarized as follows:

1. Start at any node, and find the node furthest removed from it, i.e. with the largest number of other nodes along the simplest path connecting these nodes. Repeat from the new node until the path length stops increasing. Choose the last of these nodes as number 1. As yet, all other nodes are unnumbered, and all nodes are unvisited.
2. Visit the lowest-numbered unvisited node, as follows: order all its unnumbered neighbours in increasing order of the number of pipes attached to each node, and give the next available numbers to these nodes. Repeat until all nodes are numbered.
3. Reverse the ordering, i.e. the last numbered node is the new number 1 and the old number 1 is now last.

When so many heuristics appear in an algorithm, the proof of the pudding is in the eating thereof. In Figures 5 and 6 we show the non-zero pattern of the same matrix as before, and of its lower triangular factor, after reordering by the above algorithm. There are now only 274 elements in the lower triangular factor.

Exactly the same numbers are calcualted as in standard Gaussian elimination, and therefore the envelope method in theory solves (17) exactly. In practice, the solution is as accurate as is possible with the finite arithmetic used when the matrix **M** is symmetric and positive definite.[14] This is the case here when the terms involving a_{ij} in (18) and (19) are ignored. Since these terms are usually numerically small relative to the other terms we can assume that the accuracy is not significantly impaired by the lack of symmetry.

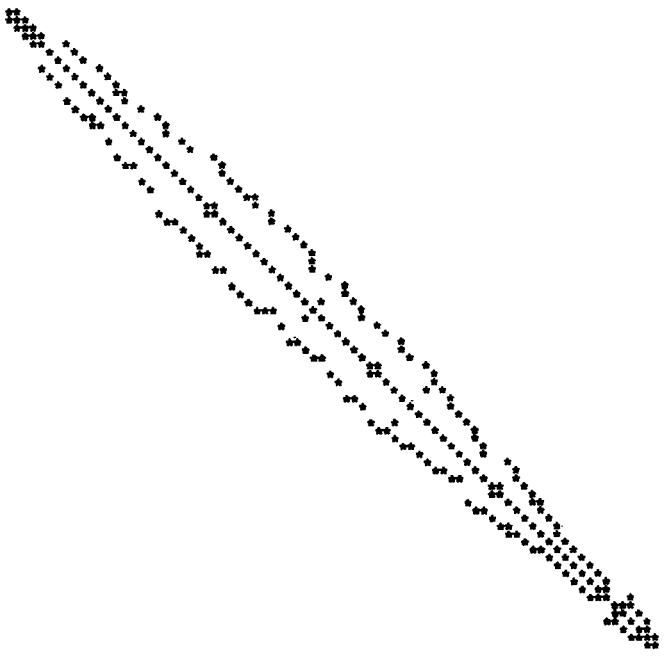

Figure 5. Structure of the matrix in Figure 3 after reordering by the reverse Cuthill–McKee algorithm

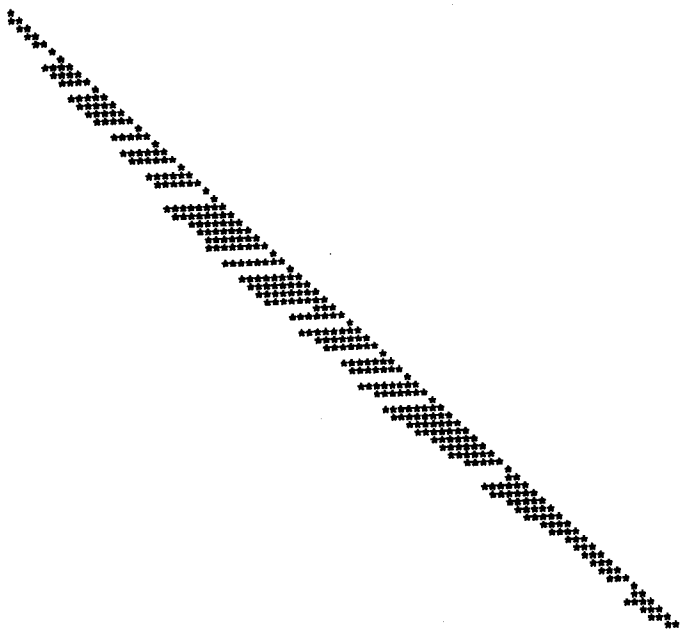

Figure 6. Structure of the lower triangular factor when the envelope method is applied to the reordered matrix in Figure 5. There are 274 significant entries

8. CONVERGENCE PARAMETERS

Wood and Rayes[3] have found that the S-NODE methods often give inaccurate results when the following convergence criterion is used:

$$\frac{\sum(Q^k - Q^{k-1})}{\sum Q^k} \leqslant 0{\cdot}005 \tag{47}$$

where Q^k is the flow rate for the present iteration and Q^{k-1} is the flow rate for the previous iteration. The accuracy problems of the S-NODE methods do not stem from an inherent flaw in the methods but rather from an inappropriate choice of the convergence parameter.

In the present method two convergence parameters are used: one that checks the accuracy of the continuity equation and one that checks the accuracy of the pressure drop/flow rate relationship.

The convergence parameter for the continuity equation is defined by

$$\varepsilon_m = \frac{|h_i|_{\max}}{m_{\text{mean}}} \tag{48}$$

where m_{mean} is the mean of the absolute values of all the mass element flows and $|h_i|_{\max}$ is the maximum of all the absolute node residual mass flow values. The residual mass flow for a node is given by

$$h_i = \sum_{j=1}^{J} (\rho_{ij} Q_{ij} s_{ij}) + d_i \tag{49}$$

The convergence parameter for the pressure drop equation is defined by

$$\varepsilon_p = \sum_{e=1}^{E} \left| \frac{\Delta p_1 - \Delta p_2}{\Delta p_1} \right| \tag{50}$$

where Δp_1 is the pressure drop across an element calculated with (2) and Δp_2 is the difference of the pressures of the two nodes associated with the element.

An accurate solution is ensured when both ε_m and ε_p become sufficiently small.

The convergence criterion used in the present study is

$$\varepsilon_p \leqslant 10^{-3} \quad \text{and} \quad \varepsilon_m \leqslant 10^{-3} \tag{51}$$

9. EXAMPLES AND COMPARISONS

Three example networks will be analysed to demonstrate the characteristics of the present method. The first is a water network which was analysed by Wood and Charles[15] with the linear method. The results and number of iterations of the linear method are reported in this paper. The layout of the network is shown in Figure 7 and the pipeline data are given in Table I. The Hazen–Williams equation was used to compute the heat loss in metres. This equation is given by

$$h_L = \frac{10{\cdot}78\, L Q^{1{\cdot}852}}{C^{1{\cdot}852} D^{4{\cdot}87}} \tag{52}$$

in which C is the pipe roughness coefficient, L is the length in m, D is the pipe diameter in m and Q is the discharge in m^3/s. With the analysis the pressure at node 11 was taken as the reference pressure. Initial pressures were set equal to the reference pressure and all initial flows were set equal to zero.

The results, shown in Table II, are virtually identical to those of the linear method given by Wood and Charles.[15]

The present method needed seven iterations to reach convergence as opposed to the four iterations for the linear method. Although the present method requires nearly twice as many iterations as the linear method, according to Wood and Charles,[15] the linear method requires substantially more time per iteration than the Newton–Raphson method. It can therefore be assumed that the speed of the present method compares favourably with that of the linear method. The calculation was performed in 1·5 s on an IBM PC 286. In order to investigate the sensitivity of the present method to initial conditions, three different cases of initial conditions were considered. In the first case initial pressures were set equal to the highest pressure in the network. In the second case they were set equal to the lowest and in the third case they were set equal to the mean of the highest and lower pressures in the network. The number of iterations for the three cases were, respectively, 10, 8 and 9. This shows that although there is some degree of sensitivity of the present method to initial conditions there is not a serious deterioration in speed if bad initial conditions are used.

In order to investigate the capability of the present method to deal with low resistance lines, the lengths of two of the lines, i.e. lines 17 and 5 in Figure 7, were changed to 0·1 mm. No problems were encountered with convergence and seven iterations were required to reach convergence. For this calculation the initial pressures were set equal to the reference pressure, i.e. the pressure at node 11.

The second network that was considered was one that was analysed by Lui[16] using the Hardy-Cross, Newton–Cross and Newton–Raphson methods and by Wood and Charles[15] using the linear method. This network is shown in Figure 8 with inflows, outflows and pipeline constants

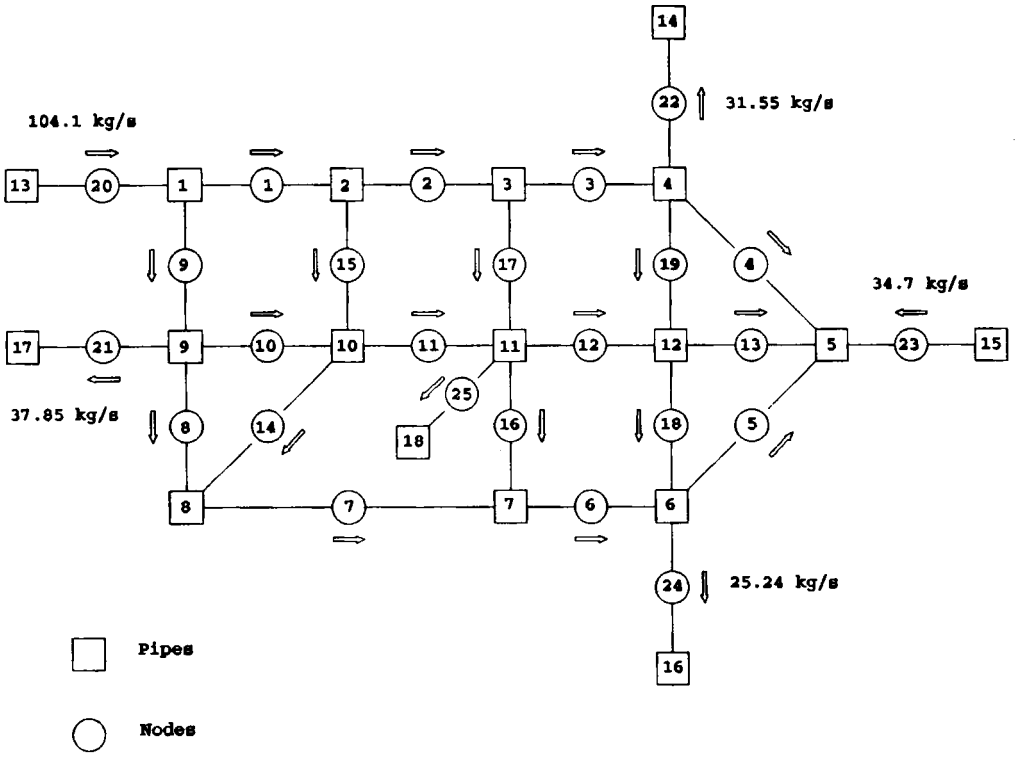

Figure 7. Water network for Example 1

Table I. Pipeline data for Example 1

Element number	Diameter (m)	Length (m)	Roughness
1	0·305	457·2	130
2	0·203	304·8	130
3	0·203	365·8	120
4	0·203	609·6	120
5	0·203	853·4	120
6	0·203	335·3	120
7	0·203	304·8	120
8	0·203	762·0	120
9	0·203	243·8	100
10	0·152	396·2	100
11	0·152	304·8	100
12	0·254	335·3	130
13	0·254	304·8	130
14	0·152	548·6	120
15	0·152	335·3	120
16	0·152	548·6	120
17	0·254	365·9	130
18	0·152	548·6	120
19	0·152	396·2	120

Table II. Results for Example 1

Element results		Node results	
Element number	Volumetric flow rate (l/s)	Node number	Pressure (bar)
1	60·66	1	1·4386
2	44·15	2	1·3323
3	17·15	3	1·0463
4	− 9·83	4	0·9773
5	− 8·84	5	1·0183
6	12·11	6	0·9711
7	13·54	7	1·0043
8	8·17	8	1·0415
9	43·44	9	1·0779
10	− 2·59	10	1·0909
11	8·55	11	1·0000
12	− 7·17	12	1·0036
13	− 16·03	13	1·4814
14	5·37	14	0·0310
15	16·51	15	2·1473
16	− 1·44	16	0·3452
17	27·00	17	− 0·2484
18	4·29		
19	− 4·56		
20	104·10		
21	37·85		
22	31·55		
23	34·70		
24	25·24		

K for each pipe. The network contains small resistance lines and according to Wood and Charles it is a much more severe test of convergence than the previous network.

With the present method the initial pressures were set equal to the pressure at the reference node and the solution was obtained in 12 iterations. Lui and Wood and Charles stated that the following numbers of iterations were needed for the methods that they have investigated:

Method	Number of iterations
Hardy-Cross	635
Newton–Cross	151
Newton–Raphson	24
Linear method	4

The present method therefore requires more iterations than the linear method, but less than the Newton–Raphson method that Liu has used.

The last network that will be analysed is a compressed air network. The layout of the network with boundary conditions is shown in Figure 9 and the pipeline data are given in Table III.

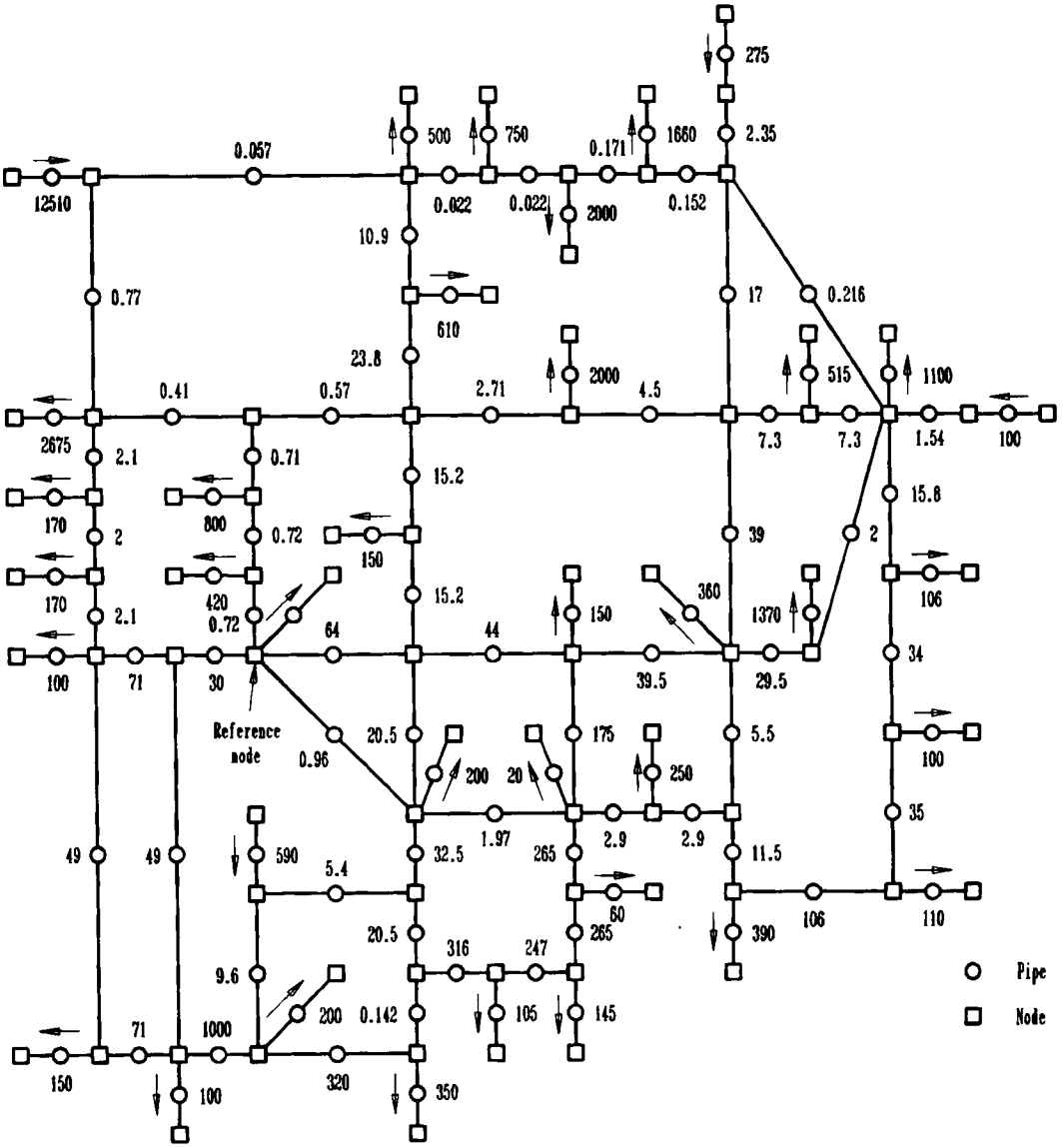

Figure 8. Water network for Example 2 (flow rates in gal/min, line constants in British units)

The pressure loss is calculated with the Darcy–Weisbach equation

$$\Delta P = \frac{fL}{D}\frac{1}{2}\rho V^2 \tag{53}$$

in which f is the friction factor, L is the length in m, D is the diameter in m and V is the velocity in m/s. The friction factor is a function of the Reynolds number, which is in turn a function of the discharge. The friction factor therefore introduces a non-linearity into the equations which can

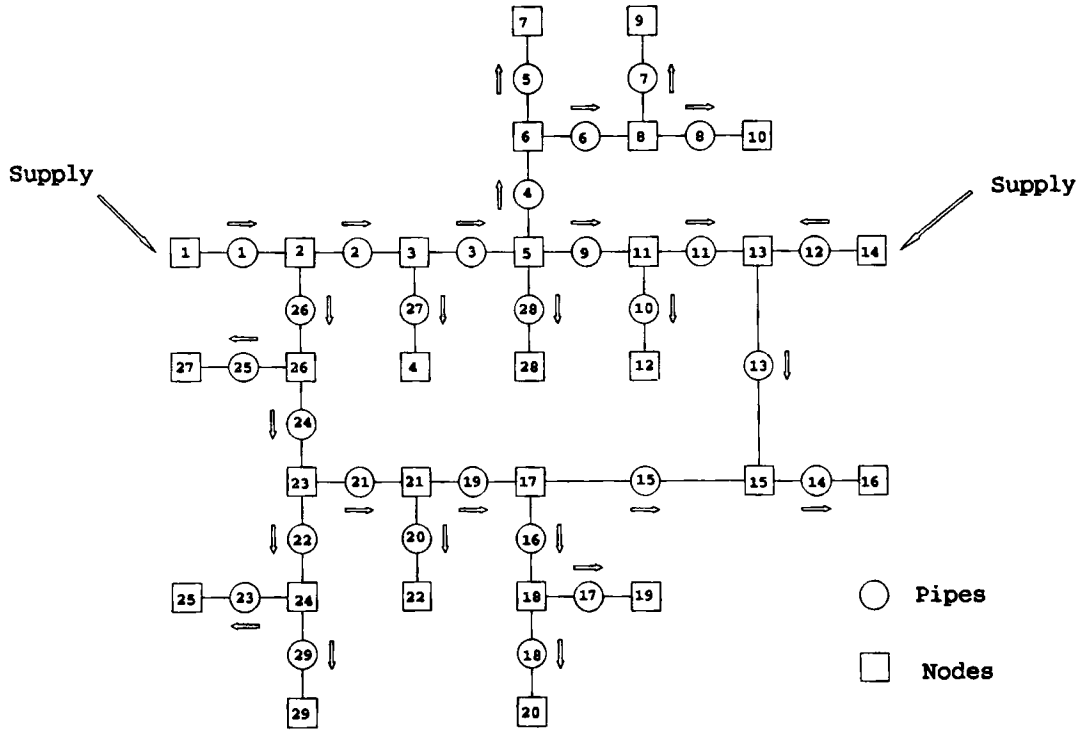

Figure 9. Compressed air network for Example 3

slow down the solution. We have chosen to eliminate this additional complexity for the purpose of demonstrating the basic algorithm by assuming f is constant and equal to 0·03.

The network was analysed using both the compressible and incompressible versions of the present method. The compressible version is the method as described in Section 4 while the incompressible version is obtained by setting a_{ij} in (18) and (19) equal to zero. However, a_{ij} in (5) and (14) is not set equal to zero. The density is therefore updated in the same way as in the case of the compressible method.

An improtant difference between the compressible and incompressible versions is that the coefficient matrix **M** in (41) is unsymmetric for the compressible version and symmetric for the incompressible version of the solution algorithm. The solution of (41) is faster and less computer storage is required if **M** is symmetric than when it is unsymmetric. It is therefore important to investigate if any benefits are gained by using the compressible rather than the incompressible solution algorithm.

Calculations were also repeated for different supply pressures to investigate how the two versions of the method compare as density variations increase. The temperature was assumed constant in all cases with a value of 15·0°C assigned to all nodes and elements. The results for a supply pressure of 6·0 bar are shown in Table IV.

Figure 10 shows the convergence parameter ε_p, given by equation (50), as a function of the number of iterations for the two versions. It can be seen that the rate of convergence of the incompressible version is much more influenced by the magnitude of density variations than the compressible version. The compressible version performs better than the incompressible version. The greater the density variations, the greater the difference between the convergence rates of the

Table III. Pipelne data for Example 3

Element number	Upstream node	Downstream node	Diameter (m)	Length (m)
1	1	2	0·019	200
2	2	3	0·01588	400
3	3	5	0·01588	400
4	5	6	0·010	100
5	6	7	0·010	100
6	6	8	0·010	100
7	8	9	0·010	100
8	8	10	0·010	100
9	5	11	0·01588	400
10	11	12	0·010	100
11	11	13	0·01588	400
12	14	13	0·019	200
13	13	15	0·01588	400
14	15	16	0·010	100
15	17	15	0·01588	400
16	17	18	0·010	100
17	18	19	0·010	100
18	18	20	0·010	100
19	21	17	0·01588	400
20	21	22	0·010	100
21	23	21	0·01588	400
22	23	24	0·010	100
23	24	25	0·010	100
24	26	23	0·01588	400
25	26	27	0·010	100
26	2	26	0·01588	400
27	3	4	0·010	100
28	5	28	0·010	100
29	24	29	0·010	100

Table IV. Results for Example 3 for a supply pressure of 6 bar

Element results			Node results	
Element number	Mass flow rate (g/s)	Density (kg/m^3)	Node number	Pressure (bar)
1	16·461	8·56	1	6·0000
2	8·033	7·19	2	5·2151
3	3·596	3·77	3	4·1131
4	3·375	10·07	4	3·0000
5	1·782	6·05	5	3·8546
6	1·593	5·37	6	3·2057
7	0·797	2·78	7	3·0000
8	0·797	2·78	8	3·0423
9	− 3·596	− 3·77	9	3·0000
10	4·437	13·14	10	3·0000
11	− 8·033	− 7·19	11	4·1131

Table IV. (continued)

12	16·461	8·56	12	3·0000
13	8·429	7·65	13	5·2151
14	4·135	12·47	14	6·0000
15	− 4·293	− 4·73	15	3·9848
16	2·800	8·77	16	3·0000
17	1·400	4·81	17	3·5975
18	1·400	4·81	18	3·1286
19	− 1·493	− 1·75	19	3·0000
20	2·986	9·60	20	3·0000
21	1·493	1·75	21	3·5478
22	2·800	8·77	22	3·0000
23	1·400	4·81	23	3·5975
24	4·293	4·73	24	3·1286
25	4·135	12·47	25	3·0000
26	8·429	7·65	26	3·9848
27	4·437	13·14	27	3·0000
28	3·816	11·72	28	3·0000
29	1·400	4·81	29	3·0000

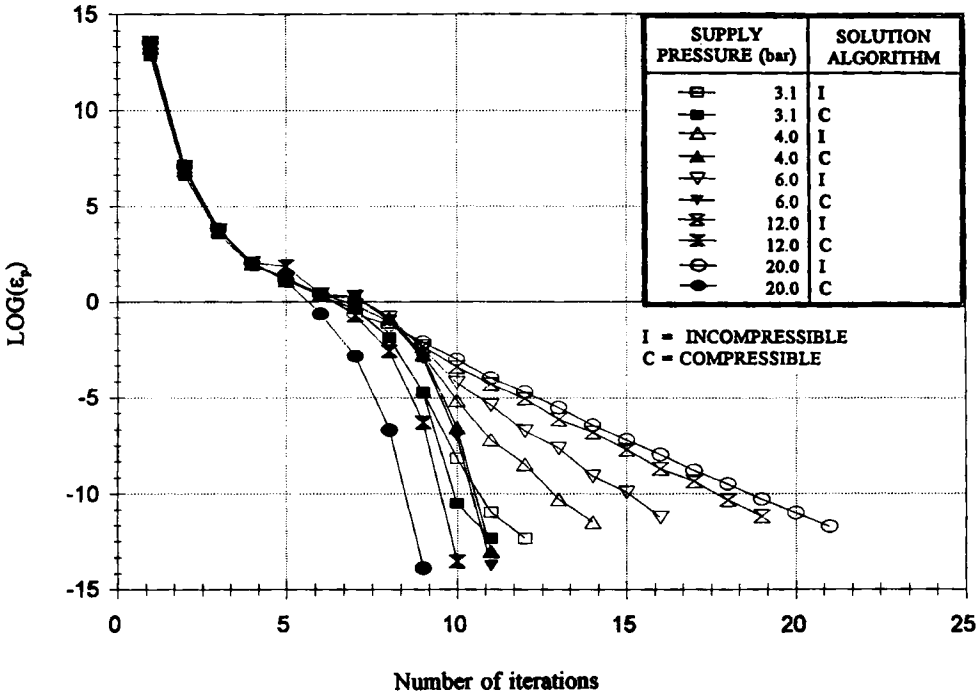

Figure 10. Convergence parameter as a function of number of iterations for Example 3

two versions. In the case of a supply pressure of 3·1 bar, there is virtually no difference in the rate of convergence between the two versions.

10. CONCLUSION

The SIMPLE algorithm, a segregated method used in CFD to solve the Navier–Stokes equations, has been extended to solve steady-state pipe network problems. The method which is based on the node equations can deal with both compressible and incompressible flows. An efficient direct method is used to solve the pressure correction equation. This enables the method to handle networks that include low resistance lines. Density variations are taken into account with the derivation of the algorithm and it is shown that in doing so a significant increase in speed is obtained when dealing with compressible flows.

The program has been used on a wide variety of networks and no convergence problems have been encountered.

APPENDIX

Nomenclature

a_{ij} $1/(2RT_{ij})$
C pipe roughness coefficient
d_i external mass flow into node i
D pipe diameter
e global element number
e_{ij} element associated with node i and branch j
E total number of elements
$\mathbf{E}$ element connectivity matrix
f $f(|Q|)$, used in equation (2)
F $\partial f/\partial|Q|$
g $g(\rho)$, used in equation (2)
G $\partial g/\partial\rho$
h_i mass flow error for node i
h_L head loss
H_{ij} $Q_{ij}/|Q_{ij}|$
i global node number
I total number of nodes
j local branch number
J number of branches associated with a specific node
L pipe length
$\mathbf{M}$ coefficient matrix
n_{ij} node associated with node i and branch j
$\mathbf{N}$ node connectivity matrix
p pressure
Q flow rate
R gas constant
s_{ij} 1 if element e_{ij} is directed towards node i, else -1
T temperature
ε_m convergence parameter, see equation (48)

ε_p convergence parameter, see equation (50)
ρ density
Δ difference

Superscripts

$*$ tentative values
$'$ corrections

Subscripts

ij values associated with element e_{ij}

REFERENCES

1. S. V. Patankar and D. B. Spalding, 'A calculation procedure for heat mass and momentum transfer in three-dimensional parabolic flows', *Int. J. Heat Mass Transfer*, **15**, 1787–1806 (1970).
2. H. Cross, 'Analysis of flow in networks of conduits or conductors', *University of Illinois Engineering, Experimental Station, Bulletino*, 286, 1936.
3. D. J. Wood and A. M. Rayes, 'Reliability of algorithms for pipe network analysis', *J. Hydraul. Div. ASCE*, **107**, 1145–1161 (1981).
4. P. Dupuis, J. L. Robert and Y. Ouellet, 'A modified element method for pipe network analysis', *J. Hydraul. Res.*, **25**, 27–39 (1986).
5. H. C. Ti, 'Mixed specification problems pipeline network analysis: partitioning methods', *Chem. Eng. J.*, **44**, 89–95 (1990).
6. M. Chandrashekar and K. H. Stewart, 'Sparsity oriented analysis of large pipe networks', *J. Hydraul. Div. ASCE*, **101**, 341–355 (1975).
7. R. W. Jeppson, *Analysis of Flow in Pipe Networks*, Ann Arbor Science, Ann Arbor, Michigan, 1977.
8. B. Gay and P. Middleton, 'The solution of pipe network problems', *Chem. Eng. Sci.*, **26**, 109–123 (1971).
9. J. P. Van Doormaal, G. D. Raithby and B. H. McDonald, 'The segregated approach to predicting viscous compressible flows', *J. Turbomachinery*, **109**, 268–277 (1987).
10. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1980.
11. U. Shamir and C. D. D. Howard, 'Water distribution system analysis', *J. Hydraul. Div. ASCE*, **94**, 219–234 (1968).
12. A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1981.
13. P. F. Lemieux, 'Efficient algorithm for distribution networks', *J. Hydraul. Div. ASCE*, **98**, 1911–1920 (1972).
14. G. H. Golub and C. Van Loan, *Matrix Computations*, 2nd edn, Johns Hopkins University Press, Baltimore.
15. D. J. Wood and C. O. A. Charles, 'Hydraulic network analysis using linear theory', *J. Hydraul. Div. ASCE*, **98**, 1157–1169 (1972).
16. K. T. H. Liu, 'The numerical analysis of water supply networks by digital computer', *Proc. 13th Congr. Int. Assoc. Hydraul. Res.*, **1**, 25–42 (1969).