Flow in Pipes Simulation Using Successive Substitution Method

Dr. Ismail Al-Rawi
Arab Open University, (Kuwait Branch), P.O 830 Al Ardia, P.C 92400, Kuwait

Abstract

The aim of this paper is to present the implementation of mathematical model to simulate the flow of fluid and gas in pipes network consisted of a number of horizontal pipes of specified diameters and length, joined at a number of nodes, and to compute the pressure, the flow rate, and the direction of flow at each node. Such simulation allows predicting the behavior of the fluid network system under different conditions. Such prediction can be effectively used to give decisions regarding the design and the operation of the real system. This model adopt the successive substitution method as a numerical technique to solve non-linear equation using a fixed-point iteration.

Keywords: Flow in Pipes Network, Successive Substitution Method, Numerical Methods, Fixed Point Iterations.

1. Introduction

Fluid flow in pipes is much encountered in our daily life and industry. The hot and cold water that we use in our homes is pumped through pipes. Water in a city is distributed by extensive piping networks. Oil and natural gas are transported hundreds long distance by large pipelines. In our body, blood is carried out by arteries and veins. The cooling water in an engine is transported by pipes in the radiator where it is cooled as it flows. Thermal energy is transferred to the desired locations through pipes. [1]

There are many good modules around to simulate the flow in pipes, but our aims here is to find a module, which depends on a simple and efficient mathematical concepts that can be programmed and computed easily. The modules has to counts for time-dependent change in the state of the system and counts all parameters in a continuous manner. [2]. choosing the iterative technique is the key point of the solution methods, and always should answer two basic questions: first, does the iteration converge?. Second, if so, how fast does it converge?. The first question is more important to be answered, in one sense it can be answered easily, and in another too hard to be answered. It is easy to be answered simply because there is a little difficulty in showing that if the initial approximation(s) α of $F(x) = 0$ are sufficiently close to the root, the iteration will converge to α. In few cases, the iteration will converge to the root independent of the initial approximation. The hard part of answering the question comes from the phrase 'sufficiently close' or how close to the root that is generally depends on the derivative value of $f(x)$ at some unknown points on the interval spanned by the initial approximations [3].

In practice, to ensure the convergence, enough priori knowledge of the desired root is needed. When the prior knowledge is poor, it is recommended to use a method, which converges independent of the starting value until a good approximation is obtained and then switch over to a more rapidly converging method.

Solving a system of nonlinear equations is a problem that is avoided when possible, customarily by approximating the nonlinear system by a system of linear equations. When this is unsatisfactory, the problem must be tackled directly. The most straightforward approaches to adapt the methods, which approximate the solutions of a single nonlinear equation in one variable, to apply when the single-variable problem is replaced by a vector problem that incorporates all the variables. [4].

The principal tool to solve the non-linear equations was found in many methods mainly Newton-Raphson method, a technique that is quadratically convergent. This technique has a number of drawbacks. It is quite costly to apply. Poor convergence characteristics when division by zero occurs for some values of the roots. Divergence at inflection points, the method may start to diverge away from the root and it may start converging again to the root. In some case where the function $f(x)$ is oscillating and has a number of roots, one may choose an initial guess close to a root. However, the guesses may jump and converge to some other root. (5). The Other well-known method to solve the non-linear equation is the bisection method, which is One of the first numerical methods developed to find the root of a nonlinear equation (also called binary-search method). It is based on finding the root between two points; therefore, the method falls under the category of bracketing methods. The advantage of this method is always convergent. Since the method brackets the root, the method is guaranteed to converge. As iterations are conducted, the interval gets halved, so one can guarantee the error in the solution of the equation. A number of drawbacks are also found. The convergence of the method is slow as it is simply based on halving the interval. If one of the initial guesses is closer to the root, it will take larger number of iterations to reach the root. [6].

Other methods like false-position and secant method are also used as alternate methods to solve a non-linear equation, but also a number of drawbacks are exist. [7][8].

2. Method of solution

Our engineering principles are simulated by solving a non-linear functional equation consisted of n real functions with n unknown real variables such as:

$$f_1 (x_1, x_2, x_3 \ldots x_n) = 0$$
$$f_2 (x_1, x_2, x_3 \ldots x_n) = 0$$
$$.$$
$$.$$
$$.$$
$$f_n (x_1, x_2, x_3 \ldots x_n) = 0$$

If we say $x = [x_1, x_2, x_3 \ldots x_n]^t$ then we shall write $f_i (x) = fi (x_1, x_2, x_3 \ldots x_n)$ when $1 \leq i \leq n$.
This will lead to solve equation in the form of $x_i = F_i (x)$ such that $x = g(x)$.
The method of solving $x = g(x)$ which we shall investigate is that of SUCCSSIVE SUBSTITUTING or Fixed point Iteration Method [9].

Iteration is a fundamental principle in computer science. A process is repeated until an answer is reached. Iterative techniques are used to find roots of equations, solutions of linear and nonlinear systems of equations, and solutions of differential equations. To start the iteration, a function g (x) for computing successive terms is needed, together with a starting value P_0. Then a sequence of values $\{p_k\}$ is obtained using the iterative rule $p_{k+1} = g (p_k)$. The sequence has the pattern of the following:

P_0 (starting value)
$p_1 = g (p_0)$
$p_2 = g (p_1)$
$p_k = g (p_{k-1})$

What can we learn from an unending sequence of numbers? If the numbers tend to a limit, we suspect that it is the answer [10]. A fixed point of a function g (x) is a number p such that p = g (p), so it is not a root of the equation x = g (x), it is a solution of the equation x = g (x).
Figure-1 show the fixed points of the function g (x) as the intersection of the curve y = g(x) and the line y = x.

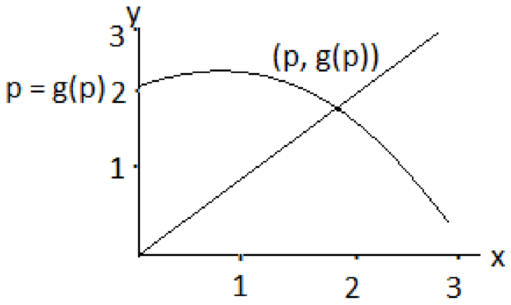


Figure-1

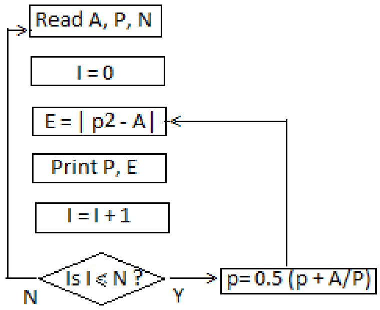


Figure-2

Figure-2 shows how the Fixed Point Iteration works.
Let p_0 approximate a solution for the equation x = g (x). Generate the sequence $\{p_n\}$ recursively by the relation $p_n = g (p_{n-1})$, n= 1, 2, 3.... .that may converge to the root α, and may be shown by Figure-3.
We also notice that the root of x = F(x) can be expressed by lim $x_n = x$.

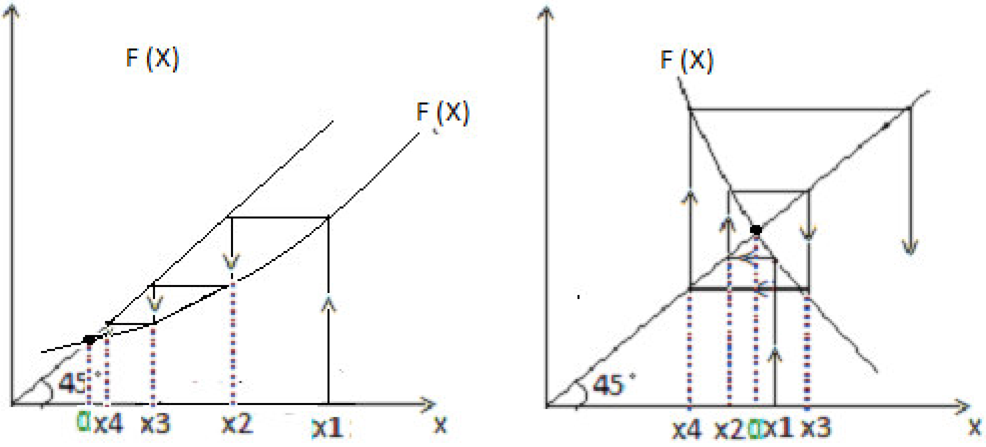


Figure-3

3. Problem statement

A network consists of a number of horizontal pipes, of specified diameters and lengths, that are joined at n nodes, numbered I = 1, 2, … n. The pressure is specified at some of these nodes. There is at most a single pipe connected directly between any two nodes.

We wrote a C program that will accept the above information, and will proceed to compute: (a) the pressures at all remaining nodes, and (b) the flow rate (and direction) in each pipe.

4. Simulation Principles

For flow of a liquid from point, *I* to point *j* in a horizontal pipe the pressure drop is given by the Fanning equation:

$$p_i - p_i = \frac{1}{2} f_M \, p \, u_m^2 \, \frac{L}{D} \text{ .... } (1)$$

Here f_M is the dimensionless *Moody friction factor*, *P* is the liquid density, u_m is the mean velocity, and *L*and *D* are the length and diameter of the pipe, respectively. Since the volumetric flow rate is $Q = (\pi D^2/4)u_m$, equation *(1)* becomes:

$$p_i - p_j = \frac{8 f_M \; pQ^2 L}{\pi^2 D^5}$$

Here, all quantities are in consistent units. However, if P_i and P_j expressed in psi (Ib_f / sq in.), *p* in *(Ib_m / cu ft)*, Q in (*gallons / min)*, *L* in ft, and D in inches, we obtain:

$$P_i - P_j = \frac{CLQ2}{D5} \text{ .... } (2)$$

Where

$$c = \frac{8 \times 12^5 \, f_m p}{\pi^2 x144 \times 32.2 \times (7.48 \times 60)^2} \text{ .... } (3)$$

If $C_{ij} = CL_{ij} \, l \, D_{ij}^5$, where the subscripts *ij* now emphasize that we are concerned with the pipe joining nodes *i* and *j*. The flow rate Q_{ij} between nodes *i* and *j* is then given by:

$$| P_i - P_j | = C_{ij} \, Q_{ij}^2 \text{ .... } (4)$$

in which Q_{ij} is plus or minus for flow from *i* to *j* or *vice versa*, respectively. In the following version, Q_{ij} will automatically have the correct sign:

$$Q_{ij} = (P_i - P_j) \sqrt{\frac{1}{C_{ij} \, | P_i - P_j |}}$$

At any *free* node *j*, where the pressure is not specified, the sum of the flows from neighboring nodes *i* must be zero:

$$\sum_i Q_{ij} = \sum_i (P_i - P_j) \sqrt{\frac{1}{C_{ij} \; | P_i - P_j |}} = 0 \text{ .... } (5)$$

When applied at all free nodes, equation *(5)* yields a system of nonlinear simultaneous equations in the unknown pressures. We shall solve this system by the method of successive - substitution. First, we note that *(p_i - p_j) is* more sensitive than *(|p_i - p_j|)*$^{1/2}$ to variations in p_j. Thus an appropriate version, analogous to equation $x_i = F(x)$ is:

$$Pj = \frac{\sum a_{ij} P_{ij}}{\sum a_{ij}} \quad ....(6)$$

In which

$$a_{ij} = (C_{ij} \ | P_i - P_j |)^{-1/2} \quad .... (7)$$

Equation *(6)* is applied repeatedly at all free nodes until either each computed pressure p_j does not change by more than a small amount ϵ from one iteration to the next, or a preassigned number of iterations, *itmax,* has been exceeded. The most recently estimated values of p_i will always be used in the right-hand side of equation (6).

In order to implement the above, we also introduce the following:

(a) A vector of logical values, p_1, p_2 ..... p_n (PGIVEN in the program), such that p_j is true (T) if the pressure is specified at node *j* and false (F) if it is not.

(b) A matrix of logical values, I_{11} ... I_{nn} (the *incidence* matrix INCID in the program), such that I_{ij} is true if there is a pipe directly joining nodes *i* and *j*, and false if not.

Since the incidence, diameter, and length matrices are symmetric (for example $D_{ij} = D_{jj}$) we need to supply only the lower triangular portions of such matrices as data. The input data will also include a complete set of pressures, $p_1, p_2, ... p_n$; some of these will be the known pressures, and the remainder will be the starting guesses at the free nodes.

5. Simulation Methods and Programming Notes

Figure-5 represents the flowchart that show the simulation process that are implemented in C-Program which reads a description of the topology of an arbitrary n node pipe network with pressures specified at particular nodes, and then computes the pressures at the remaining nodes and the inter-nodal flow rates using a method of SUCCESSIVE SUBSTITUTIONS.

List of used variables and Program Symbol Definition

A	Matrix, whose elements a_{ij} are defined by equation (7).
C	Matrix, whose elements c_{ij} relate the flow rate to the pressure drop in the pipe joining nodes *i and j.*
CONV	Logical variable used in testing for convergence, *conv.*
D. L	Matrices, whose elements D_{ij} and L_{ij} give the diameter (in.) and length (ft) of the pipe joining nodes *i* and *j*.
DENOM	Storage for the denominator of equation (6), *den.*
NUMER	Storage for the numerator of equation (6), *num.*
EPS	Tolerance, ϵ, used in testing for convergence.
F	Moody friction factor, f_m (assumed constant).
FACTOR	The constant, c, in equation (3).
I.J	Indices for representing the nodes *i* and *j*.
INCID	Matrix of logical values, *I*; if I_{ij} is T, there is a pipe joining nodes *i* and *j;* if F, there is not.
ITER	Counter on the number of iterations, *iter.*
IPRINT	Logical variable, which must have the value T if intermediate approximations to the pressures are to be printed.
ITMAX	Upper limit on the number of iterations, *itmax.*
N	Total number of nodes, *n.*
P	Vector of pressures, p_i (psi), at each node.
PGIVEN	Vector of logical values, p_i at each node. If P_i is T, the pressure is specified at node *i*; if F, it is not.
Q	Matrix, whose elements Q_{ij} give the flow rate (gpm) from node *i* to node *j*.
RHO	Density of the liquid in the pipes, *p* (lb $_m$ / cu ft).
SAVEP	Temporary storage for old pressure *p*, during convergence testing, *p*.

If *INCID(I, J)* is true, then nodes I and J are connected by a pipe segment of diameter *D(I,J)* inches and length *L(I,J)* feet. If *PGIVEN(I)* is true, the pressure at node *I, P(I)* psi, is fixed. Otherwise, *P (I)* assumes successive estimates of the pressure at node *I.* rho is the fluid density in lb/cu ft and f is the pipe friction factor, assumed identical for all pipe segments. *ITER* is the iteration counter. Iteration is stopped when *ITER* exceeds *ITMAX* or when no nodal pressure changes by an amount greater than *EPS* psi between two successive iterations. *Q(I,J)* is the flow rate in gal/min between nodes *I* and *J.* when *Q(I,J)* is positive, fluid flows from node I to node J. When IPRINT is true, intermediate approximations of the nodal pressures are printed.

6. Discussion of Results

The initial data used in the simulation is shown in Figure-6 and relate to the network that shown in the Figure-7, with $f_m = 0.056$, $p=50$ Ib m / cu ft, and two pressures fixed $p1 = 50$, $p3 = 0$ psi.

Results of simulation Figure-8 show that, although the method is computationally straight forward, it needs many iterations to give a reasonable degree of convergence. In addition, referring to equation (7), we can see that a starting guess of $p_i = p_j$ for any two nodes that are directly connected would be unfortunate.

We also note that most of the pressure drop occurs in the pipe 2-3, and that the flow in the branch 1-4-5-2 is appreciably greater than that in the pipe 1-2, even though the latter is much shorter Figure-8. Both these observations can be reconciled by noting that pressure drop is proportional to Q^2 / D^5, and that pipe 2-3 must take the combined flows along 1-4-5-2 and 1-2.

7. Conclusions

The difficulty of using successive substitution method in solving a non-linear equations depend on the nature of the problem, there is no general recognized “best” method of solution. In our simulation, the method proved to be a powerful method for solution if the approximate locations of the root is known. The success of the method depends on arranging matters so that large errors in an unknown tend to produce smaller ones in the next cycle of iteration. Thus, the successive substitution method is more suitable only when the situations carefully analyzed and details should be easily worked out for any particular situation. The method can be extended to more complex situations, in which we could allow for (a) f_m being a function of Reynolds number and pipe roughness, instead of being treated as a constant, and (b) pumps and valves in some of the branches, etc. In addition, the logical arrays used above could find a similar application in solving for the currents in a network of resistors, with known voltages applied at some of the nodes (although this would lead to a set of simultaneous linear equations).

References

[1] Flow in Pipes. Cen72367-ch08.qxd, Page 321.
[2] Bader. E.A.: An Introduction to Mathematical Modeling. N.Y. Dover. ISBN 0-486-41180-X. 2000.
[3] Anthony Ralston: A first Course in Numerical Analysis. Dover Publication INC, 2001 (Second Edition).
[4] J. D. Faris, R. Burden: Numerical Methods. 4th Edition, ISBN 918-0-495-11476-5.
[5] A. Kaw: Newton-Raphson Method of Solving Nonlinear Equation. Text Book, December 2009. www. numericalmethods.eng.usf.edu.
[6] A. Kaw: Bisection Method of Solving Nonlinear Equation. Text Book, January 2012. www. numericalmethods.eng.usf.edu.
[7] D. Nguyen: FALSE-Position Method of Solving Nonlinear Equation. Text Book, September 2012. www. numericalmethods.eng.usf.edu.
[8] A. Kaw: Secant Method of Solving Nonlinear Equation. Text Book, December 2009. www. numericalmethods.eng.usf.edu.
[9] Al-Rawi: Finding the flow of fluid in pipes Network. Al-Mansour University College – Baghdad-Iraq, May-1996.
[10] Mathews: Module for Fixed Point Iteration. California State University Fullerton. Mathfaculty.fullerton.edu/Mathews/n2003/fixedpointmod.html.
[11] Brice C., H.A. Luther: Applied Numerical Methods. Krieger Publishing Company, Malabar, Florida, 1990.

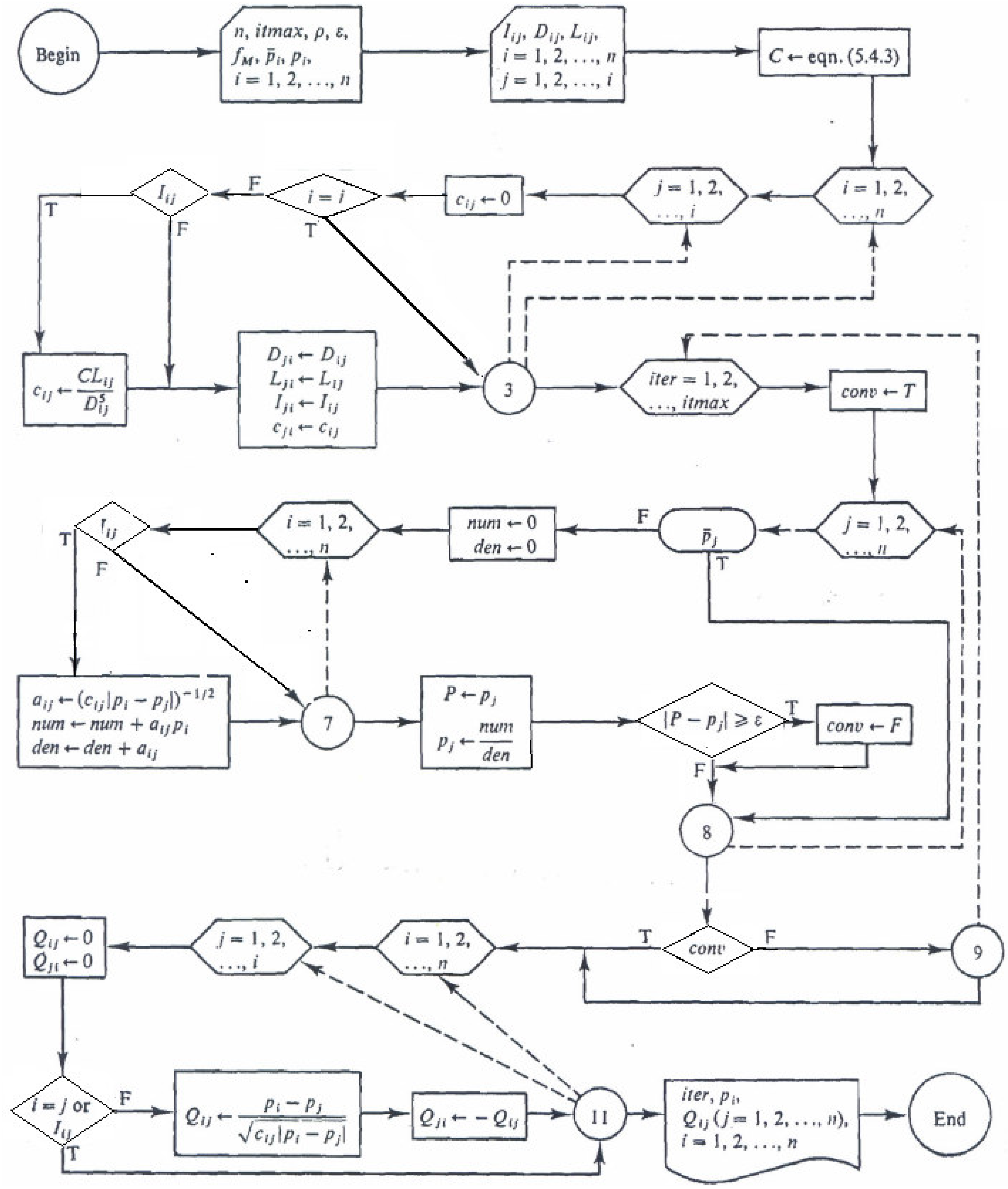
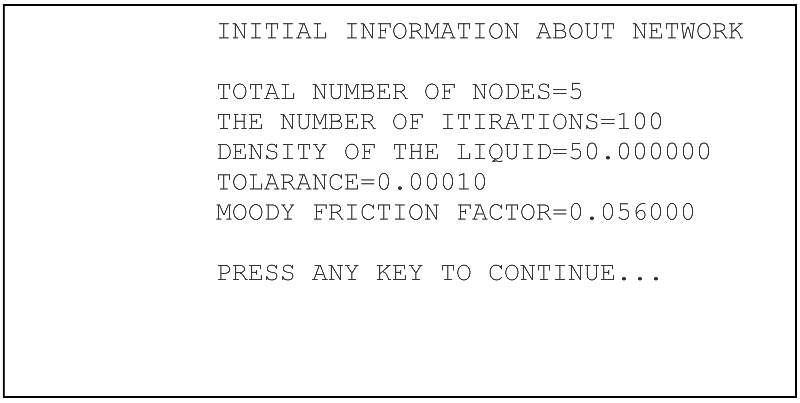

Begin
n, $itmax$, ρ, ε, f_M, $\bar{p}_i$, p_i, $i = 1, 2, \ldots, n$
I_{ij}, D_{ij}, L_{ij}, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, i$
$C \leftarrow$ eqn. (5.4.3)
$i = 1, 2, \ldots, n$
$j = 1, 2, \ldots, i$
$c_{ij} \leftarrow 0$
$i = i$
F
T
I_{ij}
T
F
$c_{ij} \leftarrow \frac{CL_{ij}}{D_{ij}^5}$
$D_{ji} \leftarrow D_{ij}$
$L_{ji} \leftarrow L_{ij}$
$I_{ji} \leftarrow I_{ij}$
$c_{ji} \leftarrow c_{ij}$
3
$iter = 1, 2, \ldots, itmax$
$conv \leftarrow T$
$j = 1, 2, \ldots, n$
$\bar{p}_j$
F
T
$num \leftarrow 0$
$den \leftarrow 0$
$i = 1, 2, \ldots, n$
I_{ij}
T
F
$a_{ij} \leftarrow (c_{ij}|p_i - p_j|)^{-1/2}$
$num \leftarrow num + a_{ij}p_i$
$den \leftarrow den + a_{ij}$
7
$P \leftarrow p_j$
$p_j \leftarrow \frac{num}{den}$
$|P - p_j| \geq \varepsilon$
T
F
$conv \leftarrow F$
8
$conv$
T
F
9
$i = 1, 2, \ldots, n$
$j = 1, 2, \ldots, i$
$Q_{ij} \leftarrow 0$
$Q_{ji} \leftarrow 0$
$i = j$ or I_{ij}
F
T
$Q_{ij} \leftarrow \frac{p_i - p_j}{\sqrt{c_{ij}|p_i - p_j|}}$
$Q_{ji} \leftarrow -Q_{ij}$
11
$iter$, p_i, Q_{ij} ($j = 1, 2, \ldots, n$), $i = 1, 2, \ldots, n$
End

Figure-5



```
INITIAL INFORMATION ABOUT NETWORK

TOTAL NUMBER OF NODES=5
THE NUMBER OF ITIRATIONS=100
DENSITY OF THE LIQUID=50.000000
TOLARANCE=0.00010
MOODY FRICTION FACTOR=0.056000

PRESS ANY KEY TO CONTINUE...
```

Figure-6

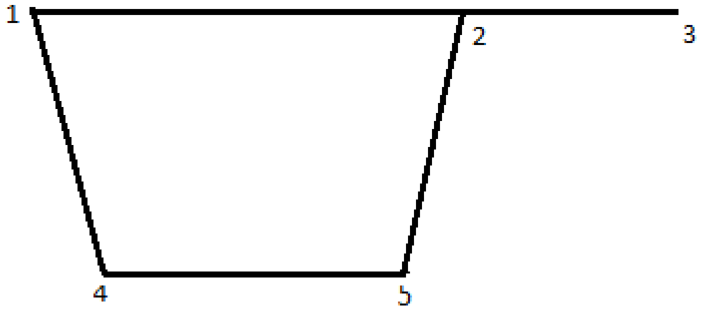
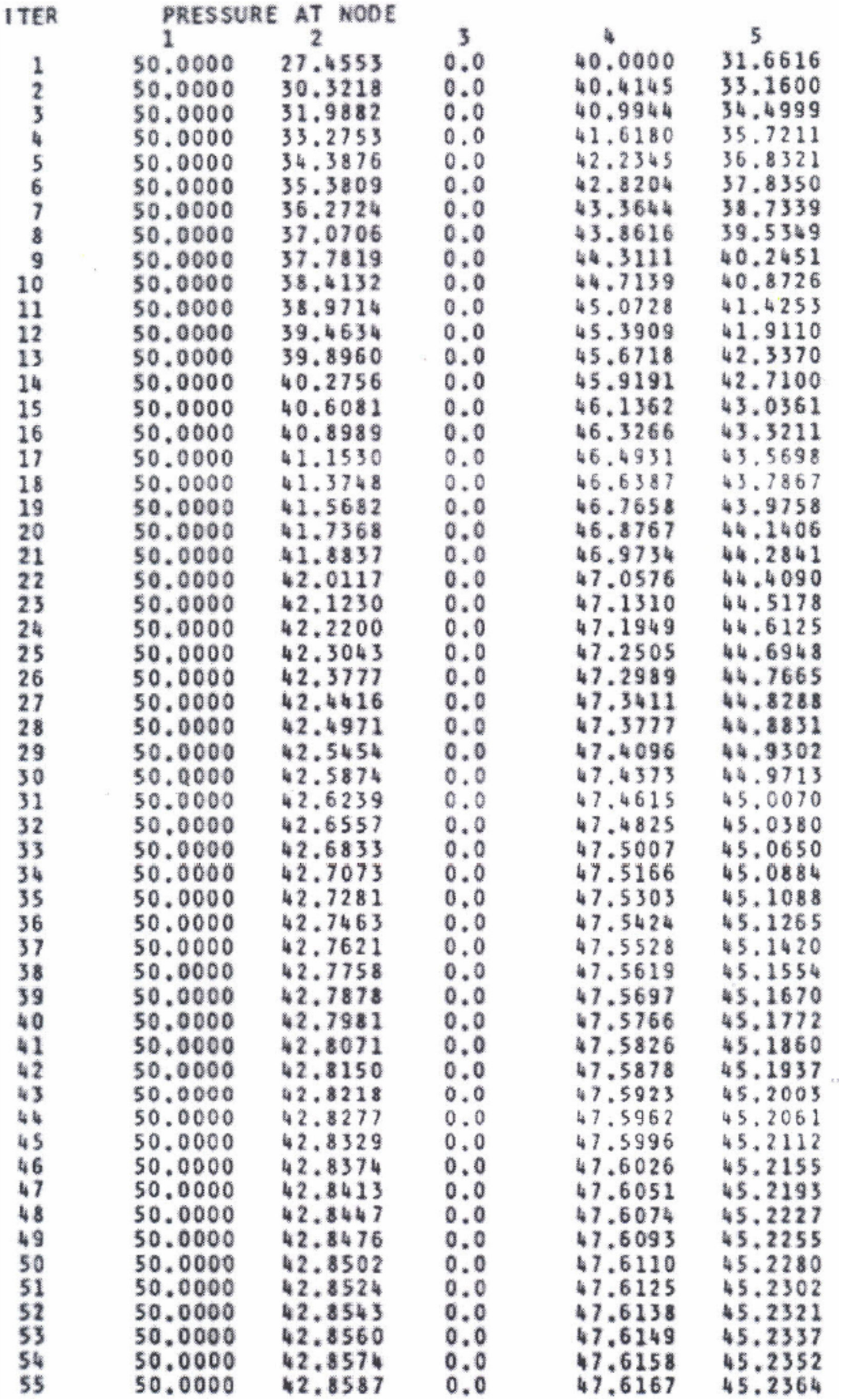
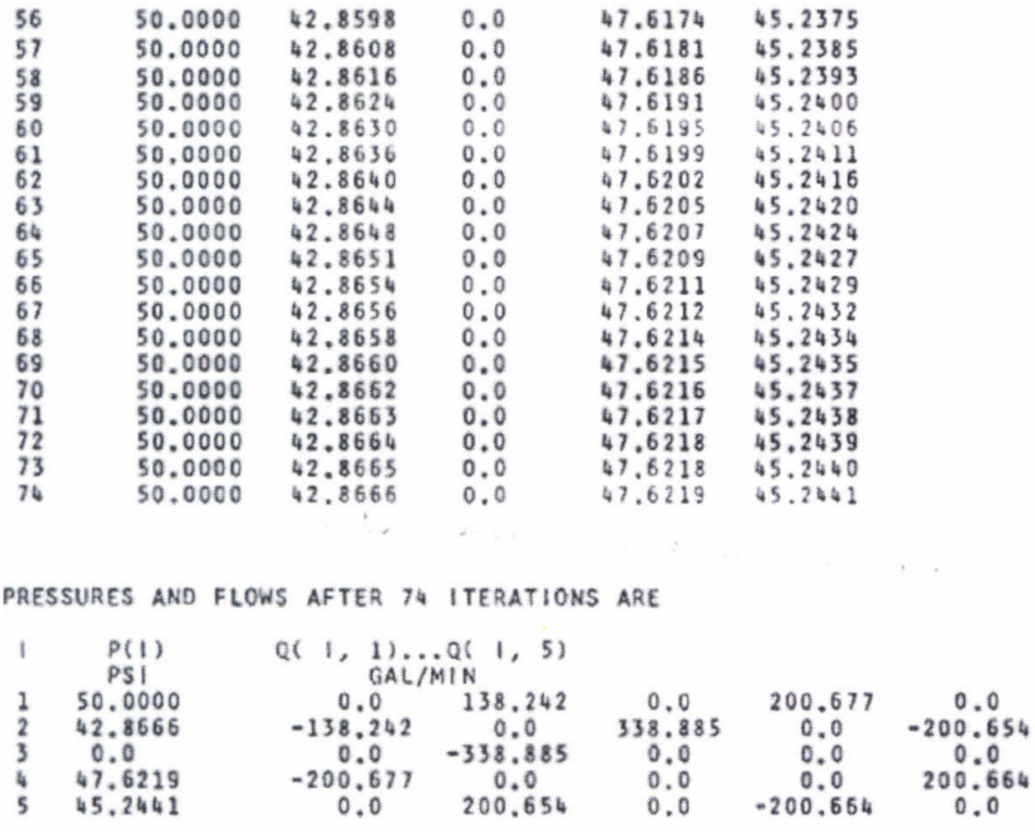


Figure-7



```
ITER          PRESSURE AT NODE
               1          2          3          4          5
  1        50.0000    27.4553     0.0       40.0000    31.6616
  2        50.0000    30.3218     0.0       40.4145    33.1600
  3        50.0000    31.9882     0.0       40.9944    34.4999
  4        50.0000    33.2753     0.0       41.6180    35.7211
  5        50.0000    34.3876     0.0       42.2345    36.8321
  6        50.0000    35.3809     0.0       42.8204    37.8350
  7        50.0000    36.2724     0.0       43.3644    38.7339
  8        50.0000    37.0706     0.0       43.8616    39.5349
  9        50.0000    37.7819     0.0       44.3111    40.2451
 10        50.0000    38.4132     0.0       44.7139    40.8726
 11        50.0000    38.9714     0.0       45.0728    41.4253
 12        50.0000    39.4634     0.0       45.3909    41.9110
 13        50.0000    39.8960     0.0       45.6718    42.3370
 14        50.0000    40.2756     0.0       45.9191    42.7100
 15        50.0000    40.6081     0.0       46.1362    43.0361
 16        50.0000    40.8989     0.0       46.3266    43.3211
 17        50.0000    41.1530     0.0       46.4931    43.5698
 18        50.0000    41.3748     0.0       46.6387    43.7867
 19        50.0000    41.5682     0.0       46.7658    43.9758
 20        50.0000    41.7368     0.0       46.8767    44.1406
 21        50.0000    41.8837     0.0       46.9734    44.2841
 22        50.0000    42.0117     0.0       47.0576    44.4090
 23        50.0000    42.1230     0.0       47.1310    44.5178
 24        50.0000    42.2200     0.0       47.1949    44.6125
 25        50.0000    42.3043     0.0       47.2505    44.6948
 26        50.0000    42.3777     0.0       47.2989    44.7665
 27        50.0000    42.4416     0.0       47.3411    44.8288
 28        50.0000    42.4971     0.0       47.3777    44.8831
 29        50.0000    42.5454     0.0       47.4096    44.9302
 30        50.0000    42.5874     0.0       47.4373    44.9713
 31        50.0000    42.6239     0.0       47.4615    45.0070
 32        50.0000    42.6557     0.0       47.4825    45.0380
 33        50.0000    42.6833     0.0       47.5007    45.0650
 34        50.0000    42.7073     0.0       47.5166    45.0884
 35        50.0000    42.7281     0.0       47.5303    45.1088
 36        50.0000    42.7463     0.0       47.5424    45.1265
 37        50.0000    42.7621     0.0       47.5528    45.1420
 38        50.0000    42.7758     0.0       47.5619    45.1554
 39        50.0000    42.7878     0.0       47.5697    45.1670
 40        50.0000    42.7981     0.0       47.5766    45.1772
 41        50.0000    42.8071     0.0       47.5826    45.1860
 42        50.0000    42.8150     0.0       47.5878    45.1937
 43        50.0000    42.8218     0.0       47.5923    45.2003
 44        50.0000    42.8277     0.0       47.5962    45.2061
 45        50.0000    42.8329     0.0       47.5996    45.2112
 46        50.0000    42.8374     0.0       47.6026    45.2155
 47        50.0000    42.8413     0.0       47.6051    45.2193
 48        50.0000    42.8447     0.0       47.6074    45.2227
 49        50.0000    42.8476     0.0       47.6093    45.2255
 50        50.0000    42.8502     0.0       47.6110    45.2280
 51        50.0000    42.8524     0.0       47.6125    45.2302
 52        50.0000    42.8543     0.0       47.6138    45.2321
 53        50.0000    42.8560     0.0       47.6149    45.2337
 54        50.0000    42.8574     0.0       47.6158    45.2352
 55        50.0000    42.8587     0.0       47.6167    45.2364
```

Computer output (continued)



```
56        50.0000    42.8598     0.0         47.6174    45.2375
57        50.0000    42.8608     0.0         47.6181    45.2385
58        50.0000    42.8616     0.0         47.6186    45.2393
59        50.0000    42.8624     0.0         47.6191    45.2400
60        50.0000    42.8630     0.0         47.6195    45.2406
61        50.0000    42.8636     0.0         47.6199    45.2411
62        50.0000    42.8640     0.0         47.6202    45.2416
63        50.0000    42.8644     0.0         47.6205    45.2420
64        50.0000    42.8648     0.0         47.6207    45.2424
65        50.0000    42.8651     0.0         47.6209    45.2427
66        50.0000    42.8654     0.0         47.6211    45.2429
67        50.0000    42.8656     0.0         47.6212    45.2432
68        50.0000    42.8658     0.0         47.6214    45.2434
69        50.0000    42.8660     0.0         47.6215    45.2435
70        50.0000    42.8662     0.0         47.6216    45.2437
71        50.0000    42.8663     0.0         47.6217    45.2438
72        50.0000    42.8664     0.0         47.6218    45.2439
73        50.0000    42.8665     0.0         47.6218    45.2440
74        50.0000    42.8666     0.0         47.6219    45.2441


PRESSURES AND FLOWS AFTER 74 ITERATIONS ARE

 I      P(I)          Q( I, 1)...Q( I, 5)
        PSI                GAL/MIN
 1   50.0000         0.0      138.242      0.0      200.677      0.0
 2   42.8666      -138.242      0.0      338.885      0.0     -200.654
 3    0.0            0.0     -338.885      0.0        0.0        0.0
 4   47.6219      -200.677      0.0        0.0        0.0      200.664
 5   45.2441         0.0      200.654      0.0     -200.664      0.0
```

Figure-8