Calculation of Smoke Movement in Building in Case of Fire

TAKAYUKI MATSUSHITA, HIROSHI FUKAI, and TOSHIO TERAI
Department of Architecture
Kyoto University, Japan

Calculation methods of smoke movement by using graph theory are presented. If both the routes of smoke movement and of evacuation are to be represented by the same method, the analysis of interaction is very simplified.

The main features of this programs are :

1. Flow circuit is expressed by incidence matrix, or loop matrix. If the data such as the incidence matrix, the geometry of branch and the initial conditions are given, no other modification is necessary to the program.

2. In order to facilitate analysis of the interaction between evacuation and smoke flow, methods are proposed to select the tree which embeds the evacuation route into part of the smoke flow tree, and to number the nodes and branches to simplify the incidence matrix.

3. The flow rate assuming method has been shown to be more efficient than the pressure assuming method .

KEYWORDS : graph theory, incidence matrix, loop matrix, smoke movement, evacuation, pressure assuming method, flow rate assuming method, Newton's method

1. INTRODUCTION

The safety of evacuation is usually estimated without taking into account smoke movement. When doors of staircases are opened during evacuation, it is possible that smoke flow through them may trap those evacuating. These effects have not been considered.

In large buildings, the longer the evacuation time, the higher is the chance for smoke to flow through evacuation routes. To analyze interaction between smoke flow and evacuation, it is convenient to formulate both routes by the same method using graph theory[1] . Mathematically there are many possibility to fix the graph, but when taking into account the safety, tree of the graph should be coincident with the route of the evacuation, therefore the tree is fixed.

In addition to the evacuation routes which are the main route causing the smoke movement, other main smoke flow routes, e.g. elevator shafts and ducts, have to be selected as a part of the tree in the incidence matrix. After routes of evacuation and smoke movement are represented by identical mathematical expression, the analysis of the system is very simplified. The incidence matrix is easily transformed to the loop matrix, both matrices could be used to calculate smoke flow. The one uses pressure as independent variables and uses incidence matrix, and the other loop flow rate and loop matrix. In this paper, two methods of formulation and the advantages of latter formulation are explained.

In order to compute smoke movement, there are some published programs based on the same method as in ventilation which assumes uniform mixing in nodes. Wakamatsu [2] has used regula-falsi method from node to node relaxation. This program has to be changed with network. Klote and Fothergill [3] improved the input of network data, but they also used regula-falsi method. Yoshida et al.[4] solved simultaneous linear equations by using one dimentional Newton's method from equation to equation. The method constructing equations is not flexible to other networks, as Wakamatsu's. In this report graph theoretical formulation is used. To solve the system iteratively, all nodes are relaxed simultaneously by using multi-dimensional modified Newton's method.[5,6] The regula-falsi method sometimes shows poor convergence for the building consisted of small and large openings, however modified Newton's method showed faster and better convergence.

2. SYMBOLS

( ) : matrix
{ } : column vector

Upper case letters represent variables at nodes.

A : floor area
C_p : specific heat
H : height to ceiling
P : room pressure at floor level
Q : strength of heat source in node
T : node temperature
W : strength of mass flow source in node
Y : smoke boundary height from floor
ΔP : error of node pressure
ΔW : error of mass flux (hypothetical source in node)
ΔV : error of node volume
Γ : density in node

Lower case letters represent variables at branches.

p : branch pressure difference
p_0 : pressure source in branch
w^+, (w^-) : flow rate to specified (opposite) directions
w : net mass flow rate, $w = w^+ - w^-$
$\tilde{w}$: mass flow rate of loop (mass flow rate of the co-tree is the same as that of the loop, while mass flow rate of the branch of the tree is the sum of that of the loops relating to this branch.)

Δp : error of branch pressure (hypothetical source in branch)
$\Delta\tilde{p}$: error of pressure around loop (hypothetical pressure source in branch of co-tree)
Δw : error of mass flow rate (hypothetical source in branch)
$\Delta\tilde{w}$: error of mass flow rate around loop (hypothetical source in branch of co-tree)
γ : branch density difference

Suffix

Lower case roman letters are used to specify the node, while lower case greek letters to branch. When the flow direction specified to branch λ is from i to j, then $p_\lambda = P_i - P_j$, $w_\lambda = w_\lambda^+ - w_\lambda^-$ and $\gamma_\lambda = \Gamma_i - \Gamma_j$.

s : smoke
a : air
t : tree
l : loop (co-tree)

$[I] = (I_t, I_l)$: reduced incidence matrix, order $(n-1) \times \beta$ where n is total number of node and β is that of branch.
$[L] = (L_t, E)$: loop(tie-set) matrix, order $(\beta-n+1) \times \beta$. From $[I][L']=[0]$, we get $(L_t) = -[I_l'][I_t']^{-1}$, where prime shows transpose of matrix.
$[E]$: unit matrix

3. FORMULATION OF GRAPH OF SMOKE MOVEMENT AND EVACUATION ROUTES

Usually it is convenient to select open air (final egress place) as the reference node (root).

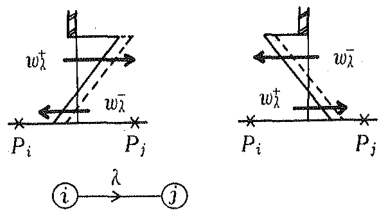
It is better to number the nodes and branches so as to simplify the incidence matrix. This can be done if

(1) the number of nodes should increase from the root to the end of branches,
(2) the branches should have the flow direction from nodes with greater number to smaller number, that is, the same direction to escape,
(3) the number of a branch of the tree should be equal to the upstream node number,

but the numbering of branches on the corresponding co-tree is arbitrary.

It is easy to check the incidence matrix, because for each column the sum of elements must be zero. $[I_t]$ thus obtained is upper triangular matrix.

The difference between the method in calculating the ventilation and the smoke flow is that the former supposes temperatures in all nodes as uniform and steady, and the latter as different and unsteady. When the temperature in node i is different from node j, a pair of fluxes in opposite direction, w_λ^+ and w_λ^-,



Direction of Branch and Net Flow

Fig.3.1 Relation between direction of branch and net flow rate.

would exist between these nodes as shown in Fig.3.1. Because it is sufficient to consider the net flow rate to evaluate conservation of flow in nodes, only one specified direction between nodes would be sufficient. When the direction of branch is from i to j, $p_\lambda = P_i - P_j$, $w_\lambda = w_\lambda^+ - w_\lambda^-$. p_λ and w_λ could be positive or negative, but derivative $\partial w_\lambda / \partial p_\lambda$ is always positive.

4. COMPUTATIONAL METHOD

4.1 Steady Flow (perfect mixing flow)

1) Pressure Assuming Method (PAM)

a) mass conservation in node

$$[I]\ \{w\} - \{W\} = \{0\} \tag{4.1}$$

b) relation between branch pressure and node pressure

$$\{p+p_\sigma\} = [I']\ \{P\} \tag{4.2}$$

c) relation between mass flow rate and pressure in branch

$$w = f(p, \Gamma, opening\ geometries) \tag{4.3}$$

There are many formulas to eq.(4.3) according to their precision. The most simple and common formulation is shown in Appendix A .

d) relation between errors of mass conservation and of pressure in node

$$([I]\ [\frac{\partial f}{\partial p}]\ [I'])\ \{\Delta P\} = \{\Delta W\}\ . \tag{4.4}$$

2) Flow rate Assuming Method (FAM)

a) tree mass flow rate, from eq.(4.1) by using the relation $[I][L']=[0]$

$$\{w_t\} = [L_t']\ \{\tilde{w}\} + [I_t]^{-1}\ \{W\} \tag{4.1'}$$

b) relation to zero loop pressure, from eq.(4.2)

$$[L]\ \{p+p_\sigma\} = \{0\} \tag{4.2'}$$

c) inverse relation to eq.(4.3)

$$p = g(w, \Gamma, opening\ geometries) \tag{4.3'}$$

This is usually not expressed explicity. Instead, eq.(4.3) should be solved iteratively or be substituted by approximate relations.

d) relation between errors of loop pressure and of loop mass flow rate

$$([L]\ [\frac{\partial g}{\partial w}]\ [L'])\ \{\Delta \tilde{w}\} = \{\Delta \tilde{p}\} \tag{4.5}$$

where $\partial g / \partial w = 1/(\partial f / \partial p)$. (4.6)

The derivative eq.(4.6) can be calculated as the reciprocal of the derivative of eq.(4.3), therefore it is not necessary to define eq.(4.3') explicitly.

PAM and FAM are both iterative methods. The PAM starts from initial guess of node pressures and iterates by solving eq.(4.4) until eqs.(4.1),(4.2) and (4.3) are satisfied within prescribed error, and the FAM takes the same process with loop flow rates and eqs.(4.1'),(4.2'),(4.3') and (4.5). In addition to PAM, FAM is tested by using Newton's Method in this paper. In most cases iteration by using modified Newton's method with constant step length correction, could give a stable result, even though the circuit consists of branches with large and small flow coefficients.

4.2 Unsteady Flow

1) Common relations to PAM and FAM

A) in case of perfect mixing flow

a) change of temperature

$$\{AH\Gamma\frac{dT}{dt}\} = -[I]\{\overline{wT}\} + \{\frac{Q}{C_p}\} \tag{4.7}$$

$$\text{where} \quad \overline{wT} = \begin{cases} w^-(T_i - T_j) & for\ (i)\ node \\ w^+(T_i - T_j) & for\ (j)\ node. \end{cases} \tag{4.8}$$

B) in case of two layers flow

a) change of temperature in smoke zone

$$\{A(H-Y)\Gamma_s\frac{dT_s}{dt}\} = -[I]\{\overline{w_sT_s}\} + \{\frac{Q_s}{C_p}\} - \{W_{as}(T_s - T_a)\} \tag{4.7'}$$

where W_{as} is the flow rate from air zone to smoke zone .

b) change of smoke boundary height

$$\{A\Gamma_a\frac{dY}{dt}\} = -[I]\{w_a\} - \{W_{as}\}\ . \tag{4.9}$$

Considering the meaning of the net flow in branch, it is not difficult to extend from perfect mixing to two layers.

2) Pressure Assuming Method (PAM)

a) relation of volume flow in node (By taking into account the volume change in node, conservation of volume flow rate is relevant.)

$$[I]\{(\frac{w}{\Gamma})\} - \{\frac{Q}{C_p\Gamma T}\} = \{0\} \tag{4.10}$$

where (w/Γ) is the net volume flow rate in branch, $(w/\Gamma) = w^+/\Gamma_i - w^-/\Gamma_j$.

For two layers flow, $w^+/\Gamma_i = w_s^+/\Gamma_{si} + w_a^+/\Gamma_{ai}$ (positive total volume flow of smoke and air) and $w^-/\Gamma_j = w_s^-/\Gamma_{sj} + w_a^-/\Gamma_{aj}$ (negative total volume flow of smoke and air).

b) relation between branch pressure and node pressure

$$\{p+p_a\} = [I']\{P\} \tag{4.2}$$

c) relation between volume flow rate and pressure in branch

$$(\frac{w}{\Gamma}) = \overline{f}(p, \Gamma, Y, opening\ geometries) \tag{4.11}$$

d) relation between errors of volume conservation and of pressure in node

$$([I][\frac{\partial \overline{f}}{\partial p}][I'])\{\Delta P\} = \{\Delta V\}\ . \tag{4.12}$$

3) Flow rate Assuming Method (FAM)

a) from eq.(4.10)

$$\{(\frac{w}{\Gamma})_t\} = [L_t']\{(\frac{\widetilde{w}}{\Gamma})\} + [I_t]^{-1}\{\frac{Q}{C_p\Gamma T}\} \tag{4.10'}$$

b) relation to zero loop pressure

$$[L]\{p+p_a\} = \{0\} \tag{4.2'}$$

c) inverse relation to eq.(4.11)

$$p = g((\frac{w}{\Gamma}),\Gamma,Y,opening\ geometries). \qquad (4.11')$$

d) relation between errors of loop pressure and of loop volume flow rate

$$([L]\ [\frac{\partial \bar{g}}{\partial (w/\Gamma)}]\ [L']\ )\{\Delta(\frac{\tilde{w}}{\Gamma})\} = \{\Delta \tilde{p}\} \qquad (4.13)$$

where in the same as eq.(4.6), elements of diagonal matrix in eq.(4.13), $\partial \bar{g}/\partial (w/\Gamma)$, are the reciprocal of $\partial \bar{f}/\partial p$.

In unsteady case, at each time step eq.(4.10) should be used instead of eq.(4.1) for PAM, or eq.(4.10') instead of eq.(4.1') for FAM. Procedure of iteration is the same as that of the steady case. This implies the conservation of volume flow should be satisfied at each time step even in the calculation of unsteady case. After the norm of errors becomes smaller than the convergence limit at each time step, temperatures in nodes for next time step are obtained by solving eq.(4.7) for perfect mixing in node. In case of two layers flow, temteratures in smoke zone and smoke boundary heights are calculated by using eq.(4.7') and (4.9).

5. COMPARISON BETWEEN LOOP FLOW RATE ASSUMING METHOD AND NODE PRESSURE ASSUMING METHOD

The computing times of flow rate assuming method (FAM) and of pressure assuming method (PAM) are compared. The main computing time is used to solve the systems of equations repeatedly. Therefore it depends mainly on the dimensions of the variables. Dimensions of eq.(4.1) or (4.10) for PAM is $(n-1)$ and of eq.(4.1') or (4.10') for FAM is $(\beta-n+1)$. Generally in tall building, there are many cases that $(n-1)>(\beta-n+1)$.

The conservation of flow in node is satisfied always for FAM, but has an error for PAM. This means that FAM is more favorable in stability than PAM when the temperature or concentration in nodes are calculated, because they are based on the conservation of flow in nodes. In case of PAM, pressures are given by eq.(4.2) and the flow rates of branches are calculated by eq.(4.3) or (4.11) explicitly, while in FAM, the pressure of branch is expressed by eq.(4.3') or (4.11'), which is the inverse relation to eq.(4.3) or (4.11), and can not be calculated explicitly. But this difference of the computing time between PAM and FAM is not significant compared to time to solve the systems of equations.

In order to combine the evacuation to smoke flow, the change of effective area αA of door in each time should be taken into account. This could be implemented to PAM as well as FAM by assigning appropriate small αA for closed door, and need not change incidence matrix and loop matrix.

Programs by PAM and FAM in steady case are shown in Appendix B.

Example of calculation in steady flow is shown in Fig.5.1 and Table 5.1. This building has n=100 nodes and β=153 branches. Therefore numbers of loops $\beta-n+1$=54 for FAM and numbers of reduced nodes $n-1$=99 for PAM. Computing time is 0.24 sec for FAM and 1.19 sec for PAM by FACOM-M382. The calculations are carried at Kyoto Univ. Data Processing Center.

6. DISCUSSION

In order to consider interaction between smoke movement and evacuation, the method of calculation is proposed. When graph theory is used, the formulation is simple and no modification of program is necessary.

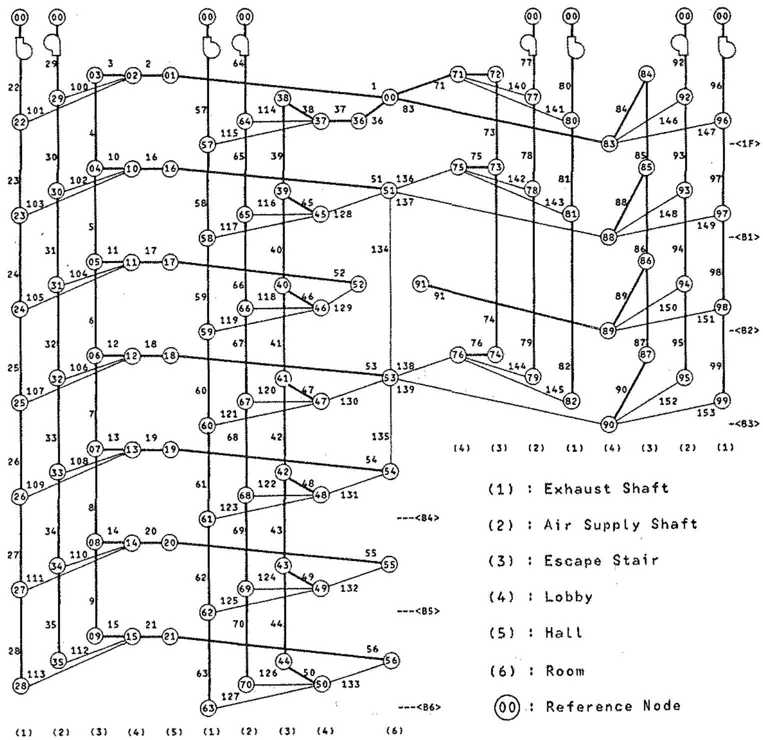


Fig.5.1 Graph of building. This is underground part of a building (B6-6F). Thick lines show the TREE.

Table 5.1 Data and results of example calculation.

number of nodes	n	100
number of branches	β	153
number of loops	$\beta-n+1$	54
dimension of variables	FAM	54
	PAM	99
computational time (sec)	FAM	0.24
	PAM	1.19

FAM : flow rate assuming method
PAM : pressure assuming method

This program is written taking into account the sparse properties of $[I]$ and $[L]$. This is the more efficient than the programs proposed in references[2,3,4].

The advantage of flow rate assuming method over the pressure assuming method is mainly a reduced computing times for solving of system of equations .

Net flow rates have some errors in PAM, while no errors in FAM. Therefore FAM is advantageous than PAM to calculate unsteady heat and concentration in its stability .

Reference

1. Busacker,R.G. and Saaty,T.L., Finite Graphs and Networks, McGraw-Hill, 1965
2. Klote,J.H. and Fothergill,J.W.Jr., Design of Smoke Control System for Buildings, ASHRAE and NBS, Sep. 1983
3. Wakamatsu,T., Calculation of Smoke Movement in Buildings, BRI Research Paper No.34,1968
4. Yoshida,H.,Shaw,C.Y. and Tamura,G.T.,A FORTRAN IV Program to calculate Smoke Concentrations in a Multi-Story Buildings, NRCC, DBR Computer Program No.45, June 1979
5. Terai,T., Calculation Method of Ventilation to apply Smoke Exhaust, Report of Sub-Committee of Architectural Institute of Japan, Kinki Branch, May 1971
6. Terai,T.,Matsushita,T. and Fukai,H., Effect of Pressure Change with Time on Smoke Movement and Difference between Pressure Assuming and Flow Assuming Methods, Trans. of Architectural Institute of Japan, Kinki Branch, Vol.24, June 1984

Appendix A

When the direction of branch is from i to j , the relations corresponding to eq.(4.3) are

$$w_\lambda = sgn.(p_\lambda)\ \alpha b(h_2-h_1)\sqrt{2g\Gamma|p_\lambda|}$$

for isothermal case $(\Gamma=\Gamma_i=\Gamma_j)$

$$\bar{w} = ||1-\bar{h}|^{3/2}-|\bar{h}|^{3/2}| \qquad if\ \gamma_\lambda<0,\bar{h}<0\ or\ \gamma_\lambda>0,\bar{h}>1$$

$$\bar{w} = -a||1-\bar{h}|^{3/2}-|\bar{h}|^{3/2}| \qquad if\ \gamma_\lambda<0,\bar{h}>1\ or\ \gamma_\lambda>0,\bar{h}<0$$

$$\bar{w} = (1-\bar{h})^{3/2}-a(\bar{h})^{3/2} \qquad if\ \gamma_\lambda<0,0\leqq\bar{h}\leqq1$$

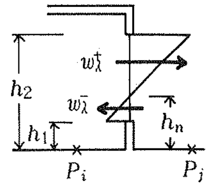
$$\bar{w} = (\bar{h})^{3/2}-a(1-\bar{h})^{3/2} \qquad if\ \gamma_\lambda>0,0\leqq\bar{h}\leqq1$$

for non-isothermal case
where

$$a = \sqrt{\Gamma_j/\Gamma_i}\ ,\bar{h} = \frac{h_n-h_1}{h_2-h_1}\ ,\bar{w} = \frac{w_\lambda}{w_0^+}\ ,$$

$$w_0^+ = \frac{2}{3}\alpha b\sqrt{2g\Gamma_i|\gamma_\lambda|}\,(h_2-h_1)^{3/2}\ ,p_\lambda = h_n\gamma_\lambda$$

and h_1,h_2,h_n are defined as showing in Fig.A.1, α is flow coefficient and b is width of opening.



$$\gamma_\lambda = \Gamma_i-\Gamma_j$$
$$p_\lambda = P_i-P_j$$
$$w_\lambda = w_\lambda^+-w_\lambda^-$$
$$h_n = p_\lambda/\gamma_\lambda$$

Fig.A.1 Definitions of h_1 , h_2 and h_n .

Appendix B

```
***** LIST OF VARIABLES *****

 (1) PRESSURE ASSUMING METHOD
       IP(J)    : POSITIVE NODE OF BRANCH J
       IM(J)    : NEGATIVE NODE OF BRANCH J
         INCIDENCE MATRIX (IP(J),J) = 1.0
         INCIDENCE MATRIX (IM(J),J) =-1.0
       IB(J)    : IDENTIFICATION OF BRANCH
         ABS(IB(J))= 0  HORIZONTAL OPENING BRANCH
         ABS(IB(J))= 1  VERTICAL OPENING BRANCH
         ABS(IB(J))= 2  FAN BRANCH
         ABS(IB(J))= 3  FAN BRANCH
         ABS(IB(J))= 4  WIND-PRESSURED BRANCH
         ABS(IB(J))= 5  WIND-PRESSURED BRANCH
       PP(I)    : NODE PRESSURE  ( mmAq )
       GG(I)    : NODE AIR SPECIFIC GRAVITY ( AIR DENSITY )  ( Kg/m**3 )
       TT(I)    : NODE ABSOLUTE AIR TEMPERATURE  ( K )
       TH(I)    : NODE AIR TEMPERATURE  ( C )
         TH(I)=TT(I)-273.16
       P(J)     : BRANCH PRESSURE DIFFERENCE  ( mmAq )
       PS(J)    : BRANCH PRESSURE SOURCE  ( mmAq )
       G(J)     : BRANCH AIR SPECIFIC GRAVITY DIFFERENCE  ( Kg/m**3 )
       GBP(J)   : BRANCH AIR FLOW SPECIFIC GRAVITY ( POSI-DIRECT. )
                  ( Kg/m**3 )
       GBM(J)   : BRANCH AIR FLOW SPECIFIC GRAVITY ( NEGA-DIRECT. )
                  ( Kg/m**3 )
       WP(J)    : BRANCH AIR MASS FLOW RATE ( POSI-DIRECT. )  ( Kg/sec )
       WM(J)    : BRANCH AIR MASS FLOW RATE ( NEGA-DIRECT. )  ( Kg/sec )
       VP(J)    : BRANCH AIR VOLUME FLOW RATE ( POSI-DIRECT. )  ( CMH )
       VM(J)    : BRANCH AIR VOLUME FLOW RATE ( NEGA-DIRECT. )  ( CMH )
       WPP(J)   : CHANGE OF BGANCH AIR FLOW RATE WITH PRESSURE DIFFERENCE
                  ( POSI-DIRECT. )
       WMP(J)   : CHANGE OF BRANCH AIR FLOW RATE WITH PRESSURE DIFFERENCE
                  ( NEGA-DIRECT. )
       DWDP(J)  : CHANGE OF TOTAL BRANCH AIR FLOW RATE WITH PRESSURE
                  DIFFERENCE
         DWDP(J)=WPP(J)+WMP(J)
       ON(J)    : NUMBER OF OPENINGS
       ALF(J)   : FLOW COEFFICIENT
       B(J)     : BREADTH OF OPENING  ( m )
       HH(J)    : TOP HEIGHT OF OPENING FROM FLOOR LEVEL  ( m )
       HL(J)    : BOTTOM H EIGHT OF OPENING FROM FLOOR LEVEL  ( m )
       HS(J)    : FLOOR LEVEL DIFFERENCE  ( m )
       DWW(I)   : NODE FLOW BALLANCE ERROR  ( Kg/sec )
       DWWMAX   : MAXIMUM OF DWW(I)  ( Kg/sec )
       ERR      : MAXIMUM OF RELATIVE ERRORS
       DPP(I)   : NODE PRESSURE CORRECTING VALUE  ( mmAq )
       AJ(I,J)  : COEFFICIENT MATRIX
       PFAC     : CORRECTIVE FACTOR
       VO       : OUTSIDE WIND VELOCITY  ( m/sec )
       IMAX     : TOTAL NUMBER OF NODES
       IMAXS    : NUMBER OF NODES EXCEPT BASE NODE
         IMAXS=IMAX-1
       JMAX     : TOTAL NUMBER OF BRANCHES
       IFIRE    : NODE NO. OF FIRE ROOM
       ISTEP    : ITERATION TIMES
       ERRLIM   : CRITICAL VALUE OF RELATIVE ERROR
       ISTLIM   : ALLOWABLE ITERATION TIMES
       VW(I)    : WORK SPACE
       IPP(I)   : WORK SPACE
       SUM(I)   : WORK SPACE
 (2) FLOW RATE ASSUMING METHOD
       AL(L)    : VALUE OF Lth ELEMENT OF LOOP MATRIX
       LL(L)    : NO. OF LOOP OF Lth ELEMENT
       LB(J+1)-LB(J)    : NUMBER OF LOOP RELATED TO Jth BRANCH
       W(J)     : BRANCH NET AIR MASS FLOW RATE  ( Kg/sec )
       DPDW(J)  : CHANGE OF BRANCH PRESSURE DIFFERENCE WITH NET AIR
                  FLOW RATE
       DPL(L)   : LOOP PRESSURE BALLANCE ERROR  ( mmAq )
       DPLMAX   : MAXIMUM OF DPL(L)  ( mmAq )
       ERR      : MAXIMUM OF RELATIVE ERRORS
       DWL(L)   : LOOP FLOW RATE CORRECTING VALUE  ( Kg/sec )
       AJ(L,J)  : COEFFICIENT MATRIX
       WFAC     : CORRECTIVE FACTOR
       LMAXS    : NUMBER OF INDEPENDENT LOOP
         LMAXS=JMAX-IMAX+1 (=JMAX-IMAXS)
       VW(L)    : WORK SPACE
       IPP(L)   : WORK SPACE
       ABSP(J)  : WORK SPACE
```

Main Routine of Pressure Assuming Method

```
**********************************************************************
*                                                                    *
*   CALCULATION OF SMOKE MOVEMENT IN BUILDING FIRE                   *
*                                                                    *
*   UNIFORMLY MIXED STEADY STATE , PRESSURE ASSUMING METHOD          *
*   USING REDUCED INCIDENCE MATRIX & NEWTON'S METHOD                 *
*                                                                    *
*                                                                    *
*                                                                    *
**********************************************************************
      CHARACTER MON*10,COM(4)*72
      DATA NO,VER,MON,ID,IY,COM/1,4.1,'  JANUARY,',13,1985
     *    ,'UNIFORMLY MIXED STEADY STATE , PRESSURE ASSUMING METHOD'
     *    ,'USING REDUCED INCIDENCE MATRIX & NEWTON''S METHOD'
     *    ,' ',' '/
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * :
      PARAMETER (NN=210,NB=380)
      DIMENSION IP(NB),IM(NB),IB(NB)
     *          ,PP(NN),GG(NN),TT(NN),TH(NN)
     *          ,P(NB),PS(NB),G(NB),GBP(NB),GBM(NB)
     *          ,WP(NB),WM(NB),WPP(NB),WMP(NB),DWDP(NB),VP(NB),VM(NB)
     *          ,ON(NB),ALF(NB),B(NB),HH(NB),HL(NB),HS(NB)
     *          ,DWW(NN),DPP(NN),AJ(NN,NN),VW(NN),IPP(NN),SUM(NN)
      DATA EPS,PFAC,ISTEP,IX/1.E-12,0.6,2*0/

*....... READ CONDITIONS & DATA FOR CALCULATION ......................
      CALL RCOND1(NW,IPRD,ERRLIM,ISTLIM)
      CALL RDATA1(COM(4),TT,TH,IP,IM,IB,ON,ALF,B,HH,HL,HS
     *            ,IMAX,IMAXS,JMAX,IFIRE,VO,NN,NB)

*....... PREPARATION FOR CALCULATION .................................
      CALL MGSET (1,6)
      CALL TITLE (NO,VER,MON,ID,IY,COM,4,NW)

*....... INITIALIZATION OF INDOOR PRESSURES ..........................
      CALL RANU2 (IX,PP,IMAXS,ICON)
      IF(ICON.EQ.30000) STOP 'CONDITION ERROR OCCURS IN RANU2'
      PP(IMAX)=0.0

*....... CALCULATION OF SPECIFIC GRAVITY & PRESSURE SOURCE ...........
      CALL CALGPS(TT,GG,GBP,GBM,G,HS,PS,IP,IM,IB,VO,IMAX,JMAX)

*::::::: PRINTOUT OF THE DATA :::::::::::::::::::::::::::::::::::::::::
      CALL WDATA1(TH,GG,IP,IM,IB,ON,ALF,B,HH,HL,G,PS,IMAXS,IMAX,JMAX
     *            ,IFIRE,VO,IPRD,NW)
*:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

*============================================== ITERATION LOOP ======
 1000 CONTINUE
      ISTEP=ISTEP+1

*....... CALCULATION OF DWW ..........................................
      CALL CALFLO(PP,P,PS,WP,WM,WPP,WMP,VP,VM,GBP,GBM,G,ON,ALF,B
     *            ,HH,HL,IP,IM,IB,IMAX,JMAX)
      CALL CALWP (WPP,WMP,DWDP,JMAX)
      CALL CALDW2(WP,WM,DWW,DWWMAX,ERR,IP,IM,IMAX,JMAX,SUM)

*""""""""""""""""""""""""""""""""""""""""""""" JUDGEMENT OF DWW """"""
      IF(ERR.LT.ERRLIM) GO TO 1100
*""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

*....... CORRECTION OF INDOOR PRESSURES ..............................
      CALL SETX  (DWW,DPP,IMAXS)
      CALL JAC2  (DWDP,AJ,IP,IM,NN,IMAX,JMAX)
      CALL LAX   (AJ,NN,IMAXS,DPP,EPS,1,IS,VW,IPP,ICON)
      IF(ICON.GE.20000) STOP 'CONDITION ERROR OCCURS IN LAX'
      CALL CORP  (PP,DPP,PFAC,IMAXS)

*""""""""""""""""""""""""""""""""""""""""""" JUDGEMENT OF ISTEP """"""
      IF(ISTEP.GE.ISTLIM) GO TO 1110
*""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

      GO TO 1000
*======================================== END OF ITERATION LOOP ======

*....... PRINTOUT OF THE RESULT ......................................
 1100 CALL WRELT1(PP,WP,WM,VP,VM,IMAXS,JMAX,ISTEP,ISTLIM,ERR,ERRLIM
     *            ,DWWMAX,IPRD,NW)
      STOP
 1110 CALL WMSG  (NW,IPRD,1)
      CALL WRELT1(PP,WP,WM,VP,VM,IMAXS,JMAX,ISTEP,ISTLIM,ERR,ERRLIM
     *            ,DWWMAX,0,NW)
      STOP 'TIMES OF STEP OVER THE MAXIMUM COUNT'
      END
```

Main Routine of Flow Rate Assuming Method

```
*********************************************************************
*                                                                   *
*   CALCULATION OF SMOKE MOVEMENT IN BUILDING FIRE                  *
*                                                                   *
*   UNIFORMLY MIXED STEADY STATE , FLOW ASSUMING METHOD             *
*   USING REDUCED LOOP MATRIX & NEWTON'S METHOD                     *
*                                                                   *
*                                                                   *
*                                                                   *
*********************************************************************
      CHARACTER MON*10,COM(4)*72
      DATA NO,VER,MON,ID,IY,COM/2,4.1,'  JANUARY,',13,1985
     *    ,'UNIFORMLY MIXED STEADY STATE , FLOW ASSUMING METHOD'
     *    ,'USING REDUCED LOOP MATRIX & NEWTON''S METHOD'
     *    ,' ',' '/
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      PARAMETER (NN=210,NB=380,NL=182)
      PARAMETER (NNL=NN*NL)
      DIMENSION IP(NB),IM(NB),IB(NB),AL(NNL),LL(NNL),LB(NB)
     *          ,GG(NN),TT(NN),TH(NN)
     *          ,P(NB),PS(NB),G(NB),GBP(NB),GBM(NB)
     *          ,W(NB),WP(NB),WM(NB),DPDW(NB),VP(NB),VM(NB)
     *          ,ON(NB),ALF(NB),B(NB),HH(NB),HL(NB),HS(NB)
     *          ,DPL(NL),DWL(NL),AJ(NL,NL),VW(NL),ABSP(NL)
     *          ,AW(NN,NN),IPP(NN)
      DATA EPS,WFAC,ISTEP,IX/1.E-12,1.0,2*0/

*....... READ CONDITIONS & DATA FOR CALCULATION .......................
      CALL RCOND1(NW,IPRD,ERRLIM,ISTLIM)
      CALL RDATA1(COM(4),TT,TH,IP,IM,IB,ON,ALF,B,HH,HL,HS
     *            ,IMAX,IMAXS,JMAX,IFIRE,VO,NN,NB)
      LMAXS=JMAX-IMAXS

*....... PREPARATION FOR CALCULATION .................................
      CALL MGSET (1,6)
      CALL LOOP21(IP,IM,AL,LL,LB,NNL,NN,LMAXS,IMAXS,JMAX,AW)
      CALL TITLE (NO,VER,MON,ID,IY,COM,4,NW)

*....... INITIALIZATION OF LOOP FLOW RATE ............................
      CALL RANU2 (IX,W(IMAX),LMAXS,ICON)
      IF(ICON.EQ.30000) STOP 'CONDITION ERROR OCCURS IN RANU2'

*....... CALCULATION OF SPECIFIC GRAVITY & PRESSURE SOURCE ...........
      CALL CALGPS(TT,GG,GBP,GBM,G,HS,PS,IP,IM,IB,VO,IMAX,JMAX)

*:::::: PRINTOUT OF THE INPUT DATA :::::::::::::::::::::::::::::::::::
      CALL WDATA1(TH,GG,IP,IM,IB,ON,ALF,B,HH,HL,G,PS,IMAXS,IMAX,JMAX
     *            ,IFIRE,VO,IPRD,NW)
*:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

*============================================= ITERATION LOOP ======
 1000 CONTINUE
      ISTEP=ISTEP+1

*....... CALCULATION OF PRESSURE BALLANCE ERROR ......................
      CALL CALW21(W,AL,LL,LB,NNL,IMAXS,JMAX)
      CALL CALPRE(W,WP,WM,VP,VM,P,DPDW,GBP,GBM,G,ON,ALF,B,HH,HL
     *            ,IB,JMAX)
      CALL CDPL21(P,PS,DPL,DPLMAX,ERR,AL,LL,LB,NNL,LMAXS,IMAXS,JMAX
     *            ,ABSP)

*"""""""""""""""""""""""""""""""""""""""""""" JUDGEMENT OF ERR """"""
      IF(ERR.LT.ERRLIM) GO TO 1100
*"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

*....... CORRECTION OF LOOP FLOW RATE ................................
      CALL SETX  (DPL,DWL,LMAXS)
      CALL JACW21(DPDW,AJ,AL,LL,LB,NNL,NL,LMAXS,JMAX)
      CALL LAX   (AJ,NL,LMAXS,DWL,EPS,1,IS,VW,IPP,ICON)
      IF(ICON.GE.20000) STOP 'CONDITION ERROR OCCURS IN LAX'
      CALL CORW  (W(IMAX),DWL,WFAC,LMAXS)

*"""""""""""""""""""""""""""""""""""""""""" JUDGEMENT OF ISTEP """"""
      IF(ISTEP.GE.ISTLIM) GO TO 1110
*"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

      GO TO 1000
*======================================= END OF ITERATION LOOP ======

*....... PRINTOUT OF THE RESULT ......................................
 1100 CALL WRELT2(W,WP,WM,VP,VM,JMAX,ISTEP,ISTLIM,ERR,ERRLIM
     *            ,DPLMAX,IPRD,NW)
      STOP
 1110 CALL WMSG  (NW,IPRD,1)
      CALL WRELT2(W,WP,WM,VP,VM,JMAX,ISTEP,ISTLIM,ERR,ERRLIM
     *            ,DPLMAX,0,NW)
      STOP 'TIMES OF STEP OVER THE MAXIMUM COUNT'
      END


      SUBROUTINE RCOND1(NW,IPRD,ERRLIM,ISTLIM)

      WRITE (6,900)
      READ  (5, * ) IANS
      WRITE (6,600)
      IF(IANS.EQ.1) WRITE (6,910)
      READ  (5, * ) NW,IPRD,ERRLIM,ISTLIM
      RETURN

  600 FORMAT(' ',T5,'INPUT (NW),(IPRD),(ERRLIM),(ISTLIM)')
  900 FORMAT(' ',T5,'DO YOU NEED EXPLANATION OF FOLLOWING INPUT ? '
     *             ,'(1)=YES OR (0)=NO')
  910 FORMAT(' ',T8,'(NW)      : OUTPUT UNIT IDENTIFIER'
     *      /' ',T8,'(IPRD)    : PRINTOUT INPUT DATA (1)=YES OR (0)=NO'
     *      /' ',T8,'(ERRLIM) : LIMIT OF RELATIVE ERROR'
     *      /' ',T8,'(ISTLIM) : LIMIT OF ITERATIVE TIMES')
      END

      SUBROUTINE RDATA1(COM,TT,TH,IP,IM,IB,ON,ALF,B,HH,HL,HS,M,MS,N,IF
     *                  ,VO,NN,NB)
      CHARACTER*(*) COM
      DIMENSION     TT(NN),TH(NN)
     *             ,IP(NB),IM(NB),IB(NB),ON(NB),ALF(NB),B(NB)
     *             ,HH(NB),HL(NB),HS(NB)

      READ(50,500) M,IF,VO
      READ(50,510) (TH(I),I=1,M)

      READ(51,520) COM
      READ(51,530) N
      READ(51,540) (IP(J),IM(J),IB(J),ON(J),ALF(J),B(J),HH(J),HL(J)
     *             ,HS(J),J=1,N)

      DO 100 I=1,M
      TT(I)=TH(I)+273.16
  100 CONTINUE

      MS=M-1
      RETURN

  500 FORMAT(2I4,F4.0)
  510 FORMAT(F8.1)
  520 FORMAT(A)
  530 FORMAT(I4)
  540 FORMAT(3I4,F4.0,4F8.4,F8.3)

      END


      SUBROUTINE LOOP21(IP,IM,AL,LL,LB,KL,KM,LS,MS,N,A)
      DIMENSION IP(N),IM(N),AL(KL),LL(KL),LB(N+1),A(KM,MS)

      DO 100 I=1,MS
      DO 101 J=1,MS
  101 A(I,J)=0.
      A(I,I)=1.
  100 CONTINUE
      DO 110 I=MS,2,-1
      DO 110 J=I,MS
      A(IM(I),J)=A(IM(I),J)+A(I,J)
  110 CONTINUE

      LB(1)=1
      LT=0
      DO 120 J=1,MS
      DO 121 L=1,LS
      B=0.
      I=MS+L
      IF(IP(I).LE.MS) B=B-A(J,IP(I))
      IF(IM(I).LE.MS) B=B+A(J,IM(I))
      IF(B.EQ.0.) GO TO 121
      LT=LT+1
      LL(LT)=L
      AL(LT)=B
  121 CONTINUE
      LB(J+1)=LT+1
  120 CONTINUE

      DO 130 J=MS+1,N
      L=J-MS
      LT=LT+1
      LL(LT)=L
      AL(LT)=1.
      LB(J+1)=LT+1
  130 CONTINUE

      RETURN
      END


      SUBROUTINE CALW21(W,AL,LL,LB,KL,MS,N)
      DIMENSION W(N),AL(KL),LL(KL),LB(N+1)

      DO 100 J=1,MS
      W(J)=0.
      DO 100 L=LB(J),LB(J+1)-1
      W(J)=W(J)+AL(L)*W(MS+LL(L))
  100 CONTINUE

      RETURN
      END


      SUBROUTINE CALGPS(TT,GG,GBP,GBM,G,HS,PS,IP,IM,IB,VO,M,N)
      DIMENSION TT(M),GG(M),GBP(N),GBM(N),G(N),HS(N),PS(N)
     *         ,IP(N),IM(N),IB(N)

*....... CALCULATION OF NODE SPECIFIC GRAVITY ........................
      DO 100 I=1,M
      GG(I)=353.25/TT(I)
  100 CONTINUE

      DO 110 J=1,N

*....... CALCULATION OF BRANCH SPECIFIC GRAVITY ......................
      GBP(J)=GG(IP(J))
      GBM(J)=GG(IM(J))
      G(J)=GBP(J)-GBM(J)

*....... CALCULATION OF PRESSURE SOURCE OF BUOYANCY ..................
      IF(HS(J)) 1,2,3
    1 PS(J)=(GBM(J)-GG(M))*HS(J)
      GO TO 10
    2 PS(J)=0.0
      GO TO 10
    3 PS(J)=(GBP(J)-GG(M))*HS(J)

*....... CALCULATION OF PRESSURE SOURCE OF WIND ......................
   10 IF(IB(J).EQ.-4) PS(J)=PS(J)-0.7*GG(M)*VO*VO/19.61
      IF(IB(J).EQ. 4) PS(J)=PS(J)+0.7*GG(M)*VO*VO/19.61
```

```
      IF(IB(J).EQ.-5) PS(J)=PS(J)+0.4*GG(M)*V0*V0/19.61
      IF(IB(J).EQ. 5) PS(J)=PS(J)-0.4*GG(M)*V0*V0/19.61

  110 CONTINUE
      RETURN
      END

      SUBROUTINE CDPL21(P,PS,DPL,DPLMAX,ERR,AL,LL,LB,KL,LS,MS,N,ABSP)
      DIMENSION P(N),PS(N),DPL(LS),AL(KL),LL(KL),LB(N+1),ABSP(LS)
      DPLMAX=0.
      ERR   =0.

      DO 100 L=1,LS
      DPL(L)=0.
      ABSP(L)=ABS(P(MS+L))
  100 CONTINUE
      DO 110 J=1,N
      DO 110 L=LB(J),LB(J+1)-1
      DPL(LL(L))=DPL(LL(L))+AL(L)*(P(J)+PS(J))
  110 CONTINUE
      DO 120 L=1,LS
      IF(ABSP(L).LE.1.) ABSP(L)=1.
      DPLMAX=AMAX1(DPLMAX,ABS(DPL(L)))
      ERR=AMAX1(ERR,ABS(DPL(L))/ABSP(L))
  120 CONTINUE

      RETURN
      END

      SUBROUTINE JACW21(DX,AJ,AL,LL,LB,KL,KN,LS,N)
      DIMENSION DX(N),AJ(KN,LS),AL(KL),LL(KL),LB(N+1)

      DO 100 L1=1,LS
      DO 100 L2=1,LS
      AJ(L1,L2)=0.
  100 CONTINUE
      DO 110 J=1,N
      I1=LB(J)
      I2=LB(J+1)-1
      DO 110 L1=I1,I2
      DO 110 L2=I1,I2
      AJ(LL(L1),LL(L2))=AJ(LL(L1),LL(L2))+AL(L1)*DX(J)*AL(L2)
  110 CONTINUE

      RETURN
      END

      SUBROUTINE JAC2(DX,AJ,IP,IM,K,M,N)
      DIMENSION DX(N),AJ(K,M),IP(N),IM(N)

      DO 100 I2=1,M
      DO 100 I1=1,M
      AJ(I1,I2)=0.
  100 CONTINUE
      DO 110 J=1,N
      AJ(IP(J),IP(J))=AJ(IP(J),IP(J))+DX(J)
      AJ(IP(J),IM(J))=AJ(IP(J),IM(J))-DX(J)
      AJ(IM(J),IP(J))=AJ(IP(J),IM(J))
      AJ(IM(J),IM(J))=AJ(IM(J),IM(J))+DX(J)
  110 CONTINUE

      RETURN
      END

      SUBROUTINE CALWP(WPP,WMP,DWDP,N)
      DIMENSION WPP(N),WMP(N),DWDP(N)

      DO 100 J=1,N
      DWDP(J)=WPP(J)+WMP(J)
  100 CONTINUE

      RETURN
      END


      SUBROUTINE CALPRE(W,WP,WM,VP,VM,P,DPDW,GBP,GBM,G,ON,ALF,B,HH,HL
     *                  ,IB,N)
      DIMENSION W(N),WP(N),WM(N),VP(N),VM(N),P(N),DPDW(N),GBP(N),GBM(N)
     *         ,G(N),ON(N),ALF(N),B(N),HH(N),HL(N),IB(N)

      DO 100 J=1,N
      WP(J)=0.0
      WM(J)=0.0
      VP(J)=0.0
      VM(J)=0.0
      IABSIB=IABS(IB(J))

*...... CALCULATION OF FAN PRESSURE .....................
      IF(IABSIB.EQ.2) THEN
      C1=1./3.
      C2=1./6.
      C3=-1.5
      P0=ON(J)
      Q0=ALF(J)

      IF(IB(J).LT.0) THEN
        WM(J)=-W(J)
        VM(J)=WM(J)*3600./GBM(J)
        X=VM(J)/Q0
        Y=(C1*X+C2)*X+C3
        DYDX=2.*C1*X+C2
        P(J)=-P0*Y
        DPDW(J)=P0*DYDX*3600./(Q0*GBM(J))
      ELSE
        WP(J)=W(J)
        VP(J)=WP(J)*3600./GBP(J)
        X=VP(J)/Q0
        Y=(C1*X+C2)*X+C3
        DYDX=2.*C1*X+C2
        P(J)=P0*Y
        DPDW(J)=P0*DYDX*3600./(Q0*GBP(J))
      END IF

      ELSE IF(IABSIB.EQ.3) THEN
      P0=ON(J)
      Q0=ALF(J)

      IF(IB(J).LT.0) THEN
        WM(J)=-W(J)
        VM(J)=WM(J)*3600./GBM(J)
        X=VM(J)/Q0
        IF(X.GE.0.6) THEN
          C1= 25./18.
          C2=-C1
          C3=-1.
        ELSE
          C1=-25./18.
          C2= 35./18.
          C3=-2.
        END IF
        Y=(C1*X+C2)*X+C3
        DYDX=2.*C1*X+C2
        P(J)=-P0*Y

        DPDW(J)=P0*DYDX*3600./(Q0*GBM(J))
      ELSE
        WP(J)=W(J)
        VP(J)=WP(J)*3600./GBP(J)
        X=VP(J)/Q0
        IF(X.GE.0.6) THEN
          C1= 25./18.
          C2=-C1
          C3=-1.
        ELSE
          C1=-25./18.
          C2= 35./18.
          C3=-2.
        END IF
        Y=(C1*X+C2)*X+C3
        DYDX=2.*C1*X+C2
        P(J)=P0*Y
        DPDW(J)=P0*DYDX*3600./(Q0*GBP(J))
      END IF

*....... ISOTHERMAL CASE ........................... .....
      ELSE IF(IABSIB.EQ.1.OR.G(J).EQ.0.) THEN
      CC=19.61*(ON(J)*ALF(J)*B(J)*(HH(J)-HL(J)))**2

      IF(W(J).LT.0.0) THEN
      WM(J)=-W(J)
      VM(J)=WM(J)*3600./GBM(J)
      P(J)=-WM(J)*WM(J)/(CC*GBM(J))
      DPDW(J)=2.*WM(J)/(CC*GBM(J))
      ELSE
      WP(J)=W(J)
      VP(J)=WP(J)*3600./GBP(J)
      P(J)=WP(J)*WP(J)/(CC*GBP(J))
      DPDW(J)=2.*WP(J)/(CC*GBP(J))
      END IF

*....... NON-ISOTHERMAL CASE .................:.... .....
      ELSE
      ABSG=ABS(G(J))
      HHL=HH(J)-HL(J)
      CC=ON(J)*ALF(J)*B(J)*SQRT(19.61*ABSG)
      C=2./3.*CC*HHL**1.5
      WBP=C*SQRT(GBP(J))
      WBM=C*SQRT(GBM(J))
      GMP=SQRT(GBM(J)/GBP(J))
      GPM=1./GMP

      IF(W(J).GT.WBP) THEN
      WP(J)=W(J)
      VP(J)=WP(J)*3600./GBP(J)
      CC=9.*ABSG*WP(J)/(CC*HHL)**2/GBP(J)
      P(J)=CC/8.*WP(J)+G(J)*(HH(J)+HL(J))/2.
      DPDW(J)=CC/4.

      ELSE IF(W(J).LT.-WBM) THEN
      WM(J)=-W(J)
      VM(J)=WM(J)*3600./GBM(J)
      CC=9.*ABSG*WM(J)/(CC*HHL)**2/GBM(J)
      P(J)=-CC/8.*WM(J)+G(J)*(HH(J)+HL(J))/2.
      DPDW(J)=CC/4.

      ELSE IF(G(J).GT.0.) THEN
      CP=CC*SQRT(GBP(J))
      CM=CC*SQRT(GBM(J))

      CC=3.*ABSG/(CC*SQRT(GBP(J)*HHL))
      AA=2.*(1.+GMP)
      P(J)=(CC*W(J)+G(J)*(GMP*HH(J)+HL(J))*2.)/AA
      DPDW(J)=CC/AA
      WP(J)=2./3.*CP*ABS(HL(J)-P(J)/G(J))**1.5
      WM(J)=2./3.*CM*ABS(HH(J)-P(J)/G(J))**1.5
      VP(J)=WP(J)*3600./GBP(J)
      VM(J)=WM(J)*3600./GBM(J)

      ELSE
      CP=CC*SQRT(GBP(J))
      CM=CC*SQRT(GBM(J))
      CC=3.*ABSG/(CC*SQRT(GBM(J)*HHL))
      AA=2.*(1.+GPM)
      P(J)=(CC*W(J)+G(J)*(GPM*HH(J)+HL(J))*2.)/AA
      DPDW(J)=CC/AA
      WP(J)=2./3.*CP*ABS(HH(J)-P(J)/G(J))**1.5
      WM(J)=2./3.*CM*ABS(HL(J)-P(J)/G(J))**1.5
      VP(J)=WP(J)*3600./GBP(J)
      VM(J)=WM(J)*3600./GBM(J)
      END IF

      END IF
  100 CONTINUE
```

```
      RETURN
      END

      SUBROUTINE CALFLO(PP,P,PS,WP,WM,WPP,WMP,VP,VM,GBP,GBM,G,ON,ALF
     *                 ,B,HH,HL,IP,IM,IB,M,N)
      DIMENSION PP(M),P(N),PS(N),WP(N),WM(N),WPP(N),WMP(N),VP(N),VM(N)
     *         ,GBP(N),GBM(N),G(N),ON(N),ALF(N),B(N),HH(N),HL(N)
     *         ,IP(N),IM(N),IB(N)
      DO 100 J=1,N

*....... CALCULATION OF BRANCH PRESSURE .....................
      P(J)=PP(IP(J))-PP(IM(J))-PS(J)

*....... CALCULATION OF FLOW RATE ..........................
      WP(J)=0.0
      WM(J)=0.0
      VP(J)=0.0
      VM(J)=0.0
      WPP(J)=0.0
      WMP(J)=0.0
      IABSIB=IABS(IB(J))

*....... CALCULATION OF FAN FLOW RATE ......................
      IF(IABSIB.EQ.2) THEN
      C1=1./3.
      C2=1./6.
      C3=-1.5
      P0=ON(J)
      Q0=ALF(J)

      IF(IB(J).LT.0) THEN
      D=AMAX1(0.0,C2*C2-4.*C1*(C3+P(J)/P0))
      SQD=SQRT(D)
      VM(J)=Q0*(SQD-C2)/(2.*C1)
      WM(J)=VM(J)*GBM(J)/3600.
      IF(D.EQ.0.) WMP(J)=2.E5
      IF(D.NE.0.) WMP(J)=GBM(J)*Q0/(3600.*P0*SQD)
      ELSE
      D=AMAX1(0.0,C2*C2-4.*C1*(C3-P(J)/P0))
      SQD=SQRT(D)
      VP(J)=Q0*(SQD-C2)/(2.*C1)
      WP(J)=VP(J)*GBP(J)/3600.
      IF(D.EQ.0.) WPP(J)=2.E5
      IF(D.NE.0.) WPP(J)=GBP(J)*Q0/(3600.*P0*SQD)
      END IF

      ELSE IF(IABSIB.EQ.3) THEN
      P0=ON(J)
      Q0=ALF(J)
      X=P(J)/P0
      IF(IB(J).LT.0) X=-X

      IF(X.GE.-4./3.) THEN
      C1=25./18.
      C2=-C1
      C3=-1.
      D=AMAX1(0.0,C2*C2-4.*C1*(C3-X))
      SQD=SQRT(D)
      ELSE
      C1=-25./18.
      C2=35./18.
      C3=-2.
      D=AMAX1(0.0,C2*C2-4.*C1*(C3-X))
      SQD=-SQRT(D)
      END IF

      IF(IB(J).LT.0) THEN
      VM(J)=Q0*(SQD-C2)/(2.*C1)
      WM(J)=VM(J)*GBM(J)/3600.
      IF(D.EQ.0.) WMP(J)=2.E5
      IF(D.NE.0.) WMP(J)=GBM(J)*Q0/(3600.*P0*SQD)
      ELSE
      VP(J)=Q0*(SQD-C2)/(2.*C1)
      WP(J)=VP(J)*GBP(J)/3600.
      IF(D.EQ.0.) WPP(J)=2.E5
      IF(D.NE.0.) WPP(J)=GBP(J)*Q0/(3600.*P0*SQD)
      END IF

      ELSE
      WM(J)=2./3.*CM*ABS(HHN*SQHHN-HLN*SQHLN)
      VM(J)=WM(J)/GBM(J)*3600.
      WMP(J)=CM/ABS(G(J))*ABS(SQHHN-SQHLN)
      END IF

      ELSE IF(HN.GT.HH(J)) THEN
      IF(G(J).GT.0.) THEN
      WP(J)=2./3.*CP*ABS(HHN*SQHHN-HLN*SQHLN)
      VP(J)=WP(J)/GBP(J)*3600.
      WPP(J)=CP/ABS(G(J))*ABS(SQHHN-SQHLN)
      ELSE
      WM(J)=2./3.*CM*ABS(HHN*SQHHN-HLN*SQHLN)
      VM(J)=WM(J)/GBM(J)*3600.
      WMP(J)=CM/ABS(G(J))*ABS(SQHHN-SQHLN)
      END IF

      ELSE IF(G(J).LT.0.) THEN
      WP(J)=2./3.*CP*HHN*SQHHN
      WM(J)=2./3.*CM*HLN*SQHLN
      VP(J)=WP(J)/GBP(J)*3600.
      VM(J)=WM(J)/GBM(J)*3600.
      WPP(J)=-CP/G(J)*SQHHN
      WMP(J)=-CM/G(J)*SQHLN
      ELSE
      WP(J)=2./3.*CP*HLN*SQHLN
      WM(J)=2./3.*CM*HHN*SQHHN
      VP(J)=WP(J)/GBP(J)*3600.
      VM(J)=WM(J)/GBM(J)*3600.
      WPP(J)=CP/G(J)*SQHLN
      WMP(J)=CM/G(J)*SQHHN
      END IF
      END IF

  100 CONTINUE
      RETURN
      END

*....... ISOTHERMAL CASE ................................. ........
      ELSE IF(IABSIB.EQ.1.OR.G(J).EQ.0) THEN
      IF(P(J).LT.0.) THEN
      C=ON(J)*ALF(J)*B(J)*(HH(J)-HL(J))*SQRT(19.61*GBM(J))
      SQP=SQRT(-P(J))
      WM(J)=C*SQP
      VM(J)=WM(J)/GBM(J)*3600.
      WMP(J)=0.5*C/SQP
      ELSE IF(P(J).GT.0.) THEN
      C=ON(J)*ALF(J)*B(J)*(HH(J)-HL(J))*SQRT(19.61*GBP(J))
      SQP=SQRT(P(J))
      WP(J)=C*SQP
      VP(J)=WP(J)/GBP(J)*3600.
      WPP(J)=0.5*C/SQP
      ELSE
      WPP(J)=1.E5
      WMP(J)=1.E5
      END IF

*....... NON-ISOTHERMAL CASE ...................... ........
      ELSE
      CP=ON(J)*ALF(J)*B(J)*SQRT(19.61*GBP(J)*ABS(G(J)))
      CM=ON(J)*ALF(J)*B(J)*SQRT(19.61*GBM(J)*ABS(G(J)))
      HN=P(J)/G(J)
      HHN=ABS(HH(J)-HN)
      HLN=ABS(HL(J)-HN)
      SQHHN=SQRT(HHN)
      SQHLN=SQRT(HLN)

      IF(HN.LT.HL(J)) THEN
      IF(G(J).LT.0.) THEN
      WP(J)=2./3.*CP*ABS(HHN*SQHHN-HLN*SQHLN)
      VP(J)=WP(J)/GBP(J)*3600.
      WPP(J)=CP/ABS(G(J))*ABS(SQHHN-SQHLN)


      SUBROUTINE CALDW2(WP,WM,DWW,DWWMAX,ERR,IP,IM,M,N,SUM)
      DIMENSION WP(N),WM(N),DWW(N),IP(N),IM(N),SUM(M)
      DWWMAX=0.
      ERR   =0.

      DO 100 I=1,M
      DWW(I)=0.
      SUM(I)=0.
  100 CONTINUE
      DO 110 J=1,N
      DWW(IP(J))=DWW(IP(J))+WP(J)-WM(J)
      DWW(IM(J))=DWW(IM(J))-WP(J)+WM(J)
      SUM(IP(J))=SUM(IP(J))+WM(J)
      SUM(IM(J))=SUM(IM(J))+WP(J)
  110 CONTINUE
      DO 120 I=1,M-1
      IF(SUM(I).LE.1.) SUM(I)=1.
      DWWMAX=AMAX1(DWWMAX,ABS(DWW(I)))
      ERR=AMAX1(ERR,ABS(DWW(I)/SUM(I)))
  120 CONTINUE

      RETURN
      END

      SUBROUTINE SETX(Y,X,M)
      DIMENSION Y(M),X(M)

      DO 100 I=1,M
      X(I)=Y(I)
  100 CONTINUE

      RETURN
      END

      SUBROUTINE CORW(WL,DWL,WFAC,LS)
      DIMENSION WL(LS),DWL(LS)

      DO 100 L=1,LS
      WL(L)=WL(L)-DWL(L)*WFAC
  100 CONTINUE

      RETURN
      END

      SUBROUTINE CORP(PP,DPP,PFAC,M)
      DIMENSION PP(M),DPP(M)
      DO 100 I=1,M
      PP(I)=PP(I)-DPP(I)*PFAC
  100 CONTINUE
      RETURN
      END


***** Omitted Subroutine *****

      SUBROUTINE ; TITLE
      --- Subroutine to print title
      SUBROUTINE ; WDATA1
      --- Subroutine to print input data
      SUBROUTINE ; WRELT1, WRELT2
      --- Subroutine to print result
      SUBROUTINE ; WMSG
      --- Subroutine to print error message
      SUBROUTINE ; RANU2
      --- Subroutine to create random number
      SUBROUTINE ; LAX
      --- Subroutine to solve system of equations by
          LU decomposition algorithm
```