USBM Contract No. P0241054

SPECIALIZED FORTRAN COMPUTER PROGRAMMING AND ANALYSIS SERVICES TO UPGRADE CAPABILITY OF MFIRE PROGRAM

Xinquan Zhou   Rudolf E. Greuer

Department of Mining Engineering
Michigan Technological University

DEPARTMENT OF THE INTERIOR
BUREAU OF MINES
WASHINGTON, D.C.

FORWARD

This report, titled "Specialized Fortran computer Programming and Analysis Services to Upgrade Capability of MFIRE Program", was prepared by the Department of Mining Engineering, Michigan Technological University, Houghton Michigan, under USBM Contract No. P0241054 . It was administered under the technical direction of the Twin Cities Research Center with Mr. Linneas W. Laage acting as the project administrator.

This report is a summary of the work accomplished during the performance of this project. This report was submitted by the author in December 1994.

Author: Dr. Xinquan Zhou, Associate Professor
Department of Mining Engineering
Beijing Graduate School
China University of Mining & Technology
Beijing, 100083 P. R. China

Advisor: Dr. Rudolf E. Greuer, Professor
Department of Mining Engineering
Michigan Technological University
Houghton, Michigan 49931

ACKNOWLEDGEMENT

The investigation for the convergence behavior of program MFIRE made by Mr. Roger A. Pierce, mining engineer in MSHA, and Mr. Chris Pritchard, senior mining engineer in Tg Soda Ash, Inc. is gratefully acknowledged.

Table of Contents

Introduction 1

1. A Brief Introduction of the Different Versions of MFIRE 3

2. Case Study Summary on the Investigation of Divergence Behavior and Computing Problems in Numerical Solution of Program MFIRE 5

3. The Investigation and Analysis of Divergence Behavior in Numerical Solution of Program MFIRE 8
   - 3.1. The iteration process in ventilation state simulation 8
   - 3.2. The analysis on Hardy Cross method 9
   - 3.3. The boundary region of fan curve 10
   - 3.4. The analysis on computing convergency with the applied fan curve fitting method 12
   - 3.5 Case study on the relation between computing convergency and the fan curve fitting or boundary region treatment of fan curve. 16

4. The Investigation, Analysis and Modification on the Iteration Divergence Caused by Programming Error. 18
   - 4.1 The fan behavior change in dynamic state simulation part (data file -- cote93.mfi) 18
     - 4.1.1 The investigation of computing results 18
     - 4.1.2 The result analysis and modification 21
   - 4.2 The error caused by fan characteristic changed (data files brushy.mfi) 22
     - 4.2.1 Case E (data file brushy1.mfi) 22
     - 4.2.2 Case F (Data file brushy3.mfi) 22
   - 4.3 The error caused by the data handling in the time table (data file vibbase.mfi) 23
     - 4.3.1 Case G (date file vibbase1.mfi) 24
     - 4.3.2 Case H (date file vibbase2.mfi) 24
     - 4.3.3 The analysis and modification 27
     - 4.3.4 The computing results of the modified program MFIRE V1.30 28

5. The Investigation, Analysis of the Other Problems in State Simulation And Source Code Modification 30

5.1. The wrong distribution of airflow rates, temperature, and contaminant in quasi-equilibrium state part when airflow reversal is happening: 30
5.2. The mistakes in the part of dynamic state simulation (data file cote93.mfi) 34
5.2.1 The fume concentration in the end junction 21 of airway 23 with fire source 34
5.2.2 The problem on the determination of average airflow rate at the time interval of airflow reversal: 36
5.2.3 The output error for airways in which airflow reversal happened 37

6. The Investigation, Analysis, and Modification for MFIRE V2.01/2.0 41
6.1 The error in the part of input data check 41
6.1.1 The elevation difference of two end junctions in an airway 41
6.1.2 The cross sectional area of airway 41
6.1.3 The reference temperature TR, the temperature T(K) and the relative humidity HJN(K)in junction K 42
6.2 The error in the part of fan curve fitting 42
6.3 The error in the predetermined iteration times in airflow distribution balancing 43
6.4 The error caused by the handling of the data in the user specified time table 46

6.5 The error in the number of segments of fan curve MFA in subroutine SPLINE. 47

7. The Summary of Modification in MFIRE v1.30 and v2.10 48
7.1 The modification in MFIRE v1.30 48
7.2 The modification in MFIRE v 2.10 49

8. The Answers for Some User's Questions 51

9. Conclusion 54

Reference 55

APPENDIX 1 Input Data File Template 56

APPENDIX 2 The Modified Subroutines of Program MFIRE version 2.10
. . . . 61

Introduction

MFIRE, written in Fortran 77, is a computer program for ventilation state simulation in the steady state or transient state condition. It can be applied for the calculation of normal ventilation system planning or the analysis of the dynamic state variation in a ventilation system under thermal or mechanical disturbance.

The state simulator MFIRE is a useful tool for mine fire fighting. In the case of a mine fire, mine ventilation engineers want to get some help to obtain more information, such as what is happening in the underground mine and how to control the mine fire, for correct decision making. By use of program MFIRE, forecasts can be made on the variation of airflow, temperature, and contaminant distribution in a ventilation system during a mine fire. The paths which the contaminant takes at each time interval can be traced. The program MFIRE can be of assistance in the preplanning of escape routes. It can also be helpful during a mine fire emergency and mine recovery operation.

Besides of offering the information of state simulation during a mine fire, MFIRE can be applied for state control by pretesting the suitability of the selected state control measures. It is to simulate the effects of the state control measures on the airflow distribution of a dynamic ventilation system. The testing result can be used for fire fighting in a similar real case of a mine fire. By employing empirical methods or a qualitative analysis like Budryk approach, ventilation engineers can choose different combinations of locations and sizes of control devices as samples of state control for an underground mine fire. Program MFIRE can simulate the different combination samples, that consist of a certain mine fire, the corresponding different control measures, and obtain the control effects. After the different control results are compared, the state control measure, which creates the best effect among the different state control measures, will be chosen for that fire case. When a mine fire, which is similar to one of the pretested cases, breaks out, the corresponding choice of the state control measure is applied to the ventilation system. It means that the simulator MFIRE has the function of indirect state control[10].

To achieve the above two functions, state simulation and state control, the mine ventilation system condition, sometimes, should be changed in a wide range to meet the requirement of the fire fighting. These wide system condition changes, such as adding operating fans, stopping or reversing fans, changing fan characteristics, adding regulators and making airflow shortcuts are represented in the time table of the input data set. When the data files are changed in a wide range, however, an iteration divergence and some other computing errors in the distribution of airflow, temperature, and contaminant may occur. More serious logical programming and flexible selection on the algorithms are wanted to meet the request of the different data sets, to achieve correct state distribution simulation and improve the iteration convergence. To enhance the reliability and common suitability of program MFIRE, it is necessary to make investigation, analysis, and modification for the formal versions of MFIRE. The version 1.29, permitting dry calculation, and versions 2.01/v2.0, permitting dry and humidity calculation, are widely applied in the mining industry today.

The data sets used for the investigation have a wide range of system condition change. The investigation reveals the new computing problems on the simulation of the airflow, temperature, or contaminant distribution, and the algorithm divergence. Programming error, improper selection of the fan curve fitting method and the treatment of the boundary region of fan curve can cause the above problems. The modification of the program may induce new problems because of the complication of the program. Avoiding new problem emergence during source code modification is a principle that was being followed by the authors of the report.

This report offers the investigation on the computing results of the data sets and the existing computing problems on the state distribution, the fan curve fitting, the handling of the boundary region of operating range of fan characteristic. The causes of the problems are analyzed and the modification of source code of MFIRE is also presented. The investigation is based on the application of MFIRE v1.29 and v2.0/v2.01. The new data file sample and modified subroutines of program MFIRE are shown in the appendix A and B respectively. The modifications have been marked to make them easily to be found.

1. A Brief Introduction of the Different Versions of MFIRE

In the recent years, the Bureau of Mines has released several versions of MFIRE. Fig.1 shows the relation among these versions. Based on the former version 1.27 {1}, Yang made some modifications to develop the version 1.29 to improve the algorithm convergence in 1992 [9]. The outline of the modification of version 1.29 is following:

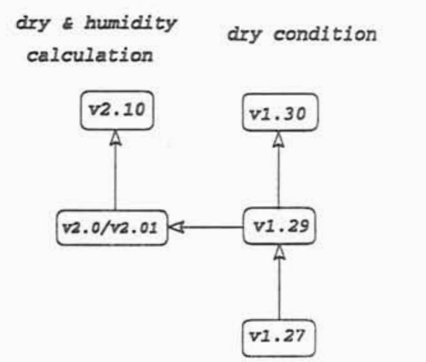

Fig.1 The Different Versions of MFIRE

i). A new cubic spline method is employed to achieve smooth fan curve fitting and improve the iteration convergence of network balancing.

ii). Apply actual airflow direction instead of initial assumed airflow direction in the input data to determine the temperature of starting and ending junctions in an airway and calculate the average rock temperature of airway wall.

iii). Correct a programming error. In the subroutine ITR, the fan curve's derivative is converted into the unit of IN. WG/10^5 CFM, as in the derivative of a mine airway to match the units in the iteration formula of network balancing.

iv). The subroutine NVP1 is modified by applying a more reasonable natural ventilation method as in the subroutine NVP2.

In 1993, based on former version 1.29, Chang made some source code modification and function enhancement to develop new versions of MFIRE v2.0/v2.01. The version 2.0 released in August 1993 and the version 2.01 released in April 1994 are similar so that they are put together [7]. The modification in this version is as follows:

i). Add the calculation of humidity condition.
ii). To simplify the structure of the program, the three parts of source code of MFIRE, MFIRE0.FOR, MFIRE1.FOR, and MFIRE2.FOR, have been combined because the functions and capacity of computers are greatly enhanced in the present time. It is no longer necessary to calculate separately for saving computer memory.
iii). Make some simplifications on the data input and programming structure.

Consequently, the investigation, analysis, and modification of the program MFIRE in this report are divided into two steps:

i). Upgrade the MFIRE V1.29 to MFIRE V1.30, which is applied in dry case, to improve the algorithm convergence and the program suitability to different data files. The chapters 3 to 5 will discuss this problem.

ii). Debug the program MFIRE V2.01, which is applied for both dry and wet cases, to correct the existed errors and add whatever the modification has been made in the version v1.30 to develop new version 2.10. Chapter 6 will deal with the problem.

2. Case Study Summary on the Investigation of Divergence Behavior and Computing Problems in Numerical Solution of Program MFIRE

There are seven groups of data sets, most of them from operating mines in the US, being applied by this computing investigation. They are i.testdata.dat (51 airways and two fans), ii. Mult-lev.dat (49 airways and one fan -- Waldo Experiment Mine NM), iii. Cote93.mfi (38 airways and one fan --Fig.3), iv.xt101.dat (70 airways and two fans Dongpong Coal Mine, Xintai Mining Administration, China), v. f94-10.bas (56 airways and two fans -- Tg Soda Ash, Inc. WY), vi. Vibbase.mfi (74 airways and eight fans -- The Viburnum N0.29 Mine,MO), and vii. Brushy.mfi (60 airways and five fans -- The Brushy Creek Mine, MO).

Based on above data sets, there are three functions of program MFIRE being investigated: i. steady state simulation, network and temperature calculation, ii. unsteady state simulation, and iii. quasi-equilibrium state simulation. The cases A, B, C, and D in data sets cote93.mfi and brushy.mfi, the cases A, B, in data sets vibbase.mfi and f94-10 has described the variation in ventilation system condition.

The summary of investigation in table 1 shows that the iteration divergence or wrong computing results were induced by the variation of ventilation system condition. The convergence behavior may get ever worse in a multi-fan system. The problems revealed by the investigation of program MFIRE can be divided into two groups:

A. the computing divergence during iteration

i). the computing divergence in network balancing calculation

ii). the computing divergence in natural ventilation calculation

iii). the fan operating point calculated converging to the outside of the operating region of fan curve. It may be induced by two reasons: iteration divergence or an unsuitable fan installed. The problem caused by the divergence should be analyzed and solved in this report. in the case of a wrong fan installed, which is not caused by programming error, a suitable fan should be selected and the input data file should be modified.

B. the error in programming

i). give the wrong distribution of airflow rates, temperature, and contaminant in quasi-equilibrium and dynamic state calculation.

ii). When one fan is stopped, the operating fans with higher sequence numbers may pick up wrong fan characteristics.

iii). the error in output results

		variation in system structure	steady state simulation	dynamic state simulation	quasi-equilibrium state simu.	comment
testdata.dat 2 fans			correct	correct	correct	two files have been applied for development and modification of MFIRE for years.
mult-lev.d at 1 fan			correct	correct	correct	
xt101.dat 2 fans			correct	correct	correct	
f94-10. bas 2 fans	A		DNB	DNB DNVP	DNB DNVP	old version < v1.29
	B		correct	correct	correct	new version > = v1.29
cote93 .mfi 1 fan (case)	A	blower in AW 1 then stop, an exhaust fan in AW5	correct	DNB, OPTOUT	WRONG STATE,***	*** if outer-level iteration is one time, it will causes problem when airflow reverse
	B	no fan in steady state then blower fan in AW 1	DNB*	DNB, OPTOUT	correct	*caused by cyclic pattern in every 2 iterations because no fan
	C	AW 1 has fixed Q, then fan stop an exhaust fan Aw 5	correct	DNB,DNVP, OPTOUT	WRONG STATE***	*** if outer level iteration is one time. it causes problem when airflow reverse
	D	dynamic part, a blower fan in AW 1, stop, an exhaustt fan in AW 5	DNB*	DNB, OPTOUT	correct	*caused by cyclic pattern in every 2 successive iterations because no fan operation
vibbase. mfi 8 fans	A		correct	correct	correct	
	B	fan 4 reverse	correct	WRONGFAN, DNB	correct	error in time table data handled
brushy .mfi 5 fans (case)	A	fan 4 reverse	correct	DNB,OPTOUT WRONGFAN	correct	
	B	fans 4,5 reverse	correct	correct**	correct	** but if fan 3,4 reverse, curve of fan 5 will be wrong fitted
	C	same as A fan 3 change	DNB,DNVP	DNB,DNVP, OPTOUT	DNB,DNVP	the data points of fan 3 are changed from (10000,16600, 27000) to (20000, 26600, 27000) cause the divergence problem.
	D	sane as B fan 3 change	DNB,DNVP	DNB,DNVP, OPTOUT	DNB,DNVP,	

Table 1 The Investigation of Simulator MFIRE (v.1.29)

To table 1

DNB: divergence in network balancing calculation,
DNVP: divergence in natural ventilation pressure calculation,
OPTOUT: Operating point in the outside of operating range of fan curve,
WRONGFAN: Picking up a wrong fan characteristic,
WRONGSTATE: Wrong temperature, fume, and airflow distribution.

3. The Investigation and Analysis of Divergence Behavior in Numerical Solution of Program MFIRE

The iteration convergence behavior is tied, in many ways, to various aspects involved in the numerical analysis algorithm for network calculation, programming of source code, fan curve fitting, fan operating point finding, and the handling of the boundary region of fan characteristic curve. The modification of fan curve fitting by applying a new cubic spline interpolation approach has reduced the possibility of iteration divergence [9]. The application of some data sets, however, has still resulted in iteration divergence as shown in table 1. Consequently, in this chapter, a brief analysis is introduced to reveal the necessity of selecting flexible methods for the fan curve fitting and the treatment of the boundary region of fan characteristic curve. A comprehensive investigation for the relations between the convergence behavior and the system condition changes, programming error, fan curve fitting method, and the treatment of the boundary region of fan curve is offered.

3.1. The iteration process in ventilation state simulation

During dynamic state simulation, the data, such as the distribution of airflow rates, temperature, and contaminant must be updated in each time interval. An interval-oriented simulation method is applied in simulator MFIRE. Each airway, affected by a mine fire, is divided into several segments, control volumes, along airways[3]. The parameters in each segment, such as location, temperature, concentration of fume and methane, are updated and recorded. By advancing the air segments, the parameters in junctions and airways are calculated. A nested iteration is employed because of the complicated nature of dynamic state simulation. The inner iteration for network balancing is the same as steady state simulation. The criterion of airflow rate correction in airways of one mesh is taken to be 10 CFM in the average increment summary. Its convergence behavior of iteration is related to the applied methods in network solving and fan curve fitting.

The outer iteration is for updating natural ventilation pressure. The variation of natural ventilation drafts during the refining process of iteration is the key factor that induces the parameter changes in the ventilation system. It is naturally chosen as the criterion to judge if the computing accuracy is satisfied. In program MFIRE, the criterion, being the absolute value of difference of the natural ventilation pressure between two successively performed iterations, is taken to be 0.0002" W.G. per mesh. Its convergence behavior, not like the inner iteration depending on the specific algorithms for fan curve fitting and the network solving, may relate to the pattern of state variation during iteration and the influence of network calculation results. The divergence of network balancing (inner iteration) will cause the large error of the pressure balancing in each mesh so that the divergence in the natural ventilation pressure calculation, outer iteration, is induced. The convergence in the inner iteration becomes the prerequisite for the computing convergence of the outer iteration. The reversal of the airflow in some airways between two successively preformed iterations is often another reason for divergence in the outer iteration. Consequently, the investigation and analysis of divergence behavior should be considered on the solution method of ventilation network, fan curve fitting method, treatment of boundary region of fan curve, and programming error. Seven groups of data sets have been employed for the experiment investigation to check the correction of formal and modified programs on above items with intermediate and final output. The results of investigation are combined with the theoretical analysis to reveal the soundness of various algorithms and programming.

3.2. The analysis on Hardy Cross method

The comments on the solution method of a ventilation network have been carried on many years. They can be easily found from many technical papers about mine ventilation network calculation around the world. The Hardy Cross method, which is the most popular approach among various algorithms, has been proved to be one of the good solution methods for the ventilation network.

The Hardy Cross method omits all terms of the second and higher-order derivative from the Tailor's expansions of the ventilation network equations. It disregards the first order derivative terms but those with respect to the airflow rates in the primary branches of each mesh. In this way, the simultaneous equation set of a ventilation network is converted into a set of equations that are independent of each other. The algorithm becomes very simple but the convergency of this approach suffers from an intricate network divided into independent meshes.

Obviously, if the term in the first order derivative with respect to the airflow rate in the primary airway, the remainder term, is much larger than the other terms of the first order derivative, the omitted terms, the influence of those terms omitted on the algorithm convergency can be reduced. The term RQ, the product of resistance R and airflow rate Q, is the first order derivative of RQ^2 with respect to the airflow rate in a primary airway, It should be as large as possible so that the convergence of the Hardy Cross method can be improved. That is why the airways with large RQ should be selected as primary airways. The pattern of meshes formed from a ventilation network, which is based on the selection of primary airways, becomes an important factor of iteration convergence for Hardy Cross method. If the state distribution of a ventilation system is changed by the airflow reversal, fan added or stopped, or regulator added, the pattern of meshes should be alternated by finding new primary airways from the renewed values of the RQ in airways. In some programs for mine ventilation calculation, the airways with large airway resistance R are chosen as primary airways, because the program developers think that the initial airflow rate Q is assumed and the product of RQ is no meaning to be the selected standard for finding primary airways. In fact, the Q will converge to the true value gradually after several iterations. The RQ choice will offer faster convergency than the R choice after reorganizing the pattern of meshes according to the upgraded value of RQ.

From the Hardy Cross method, the correction increment of each iteration is given as following

$$\Delta Q_i = -\frac{\sum_{k=1}^{N} C_{ik} R_k Q_k^{(n)} \left| Q_k^{(n)} \right| - F_i\left(Q_i^{(n)} \right)}{2\sum_{k=1}^{N} (C_{ik})^2 R_k \left| Q_k^{(n)} \right| - \frac{dF_i(Q_i)}{dQ_i}} \qquad i = 1,2,...m \tag{1}$$

where: ΔQ_i is the correction increment of the airflow rate in mesh i,
m is the total number of meshes in a ventilation network,
N is the total number of branches in the ventilation network,
C_{ik} is the fundamental mesh matrix,
R_k is the resistance of branch k in mesh i,
$Q_k^{(n)}$ is the airflow rate in branch k in mesh i at the n-th iteration,
$F_i(Q_i^{(n)})$ is the fan pressure function in mesh i with airflow rate Q_i being its argument at the n-th iteration.

The investigation shows that the correction increment of airflow rate ΔQ should be large enough for fast convergence. If the ΔQ is too large, however, it will cause a divergence problem because the large fluctuation may be induced. In the denominator of equation (1), the first derivative of regular airway characteristics $R_k|Q_k|$ is always positive. The slope of the normal operating range of the fan characteristic curve is always negative. The denominator remains positive during the iteration process in network calculation. If the fan curve is flat (the slope of fan curve is small), the derivative of the fan curve has small influence on the airflow correction and computing convergence. If the slope of the fan curve is very large and the fan curve rises steeply, the very large value of the denominator will cause the divergence problem [2].

The other factor affecting the computing convergence is the numerator in equation (1). The intermediate calculating results of the case study show that the divergence will occur sometimes such as at the large value of the numerator, or at the mutual change of positive and negative values between two successively preformed iterations. The correction increment of airflow rate ΔQ cannot be reduced to achieve computing convergence.

3.3. The boundary region of fan curve

Normally, the operating point, the intersection of the fan characteristic curve and network characteristic curve, is in the operating region of the fan curve. Sometimes, if the fan is not suitable for the ventilation network or there is thermal draft disturbance during mine fire, the operating point may converge outside the operating region. On the other hand, in the initial stage of iteration, the upgraded operating point is always starting from the outside the working region, then gradually converging to the operating region because the initial airflow distribution is assumed. Consequently, it is important to investigate and analyze the influence of the boundary

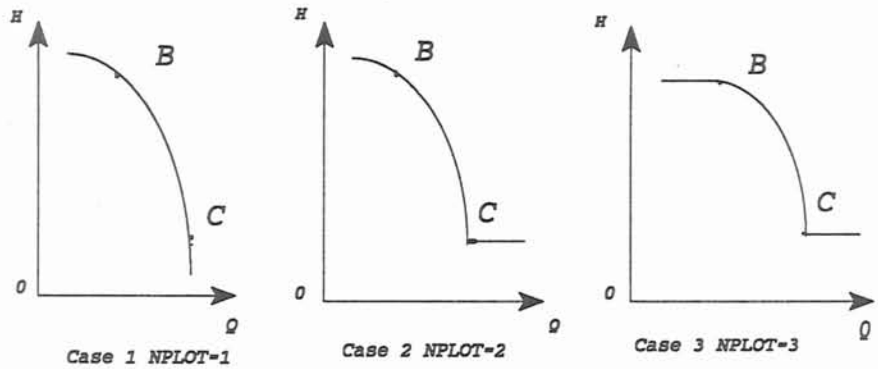

Fig.2 Handling boundary region of fan curve

region of the fan curve on the convergence behavior in network simulation. Fig.2 shows the treatments for the boundary region that are represented by NPLOT=1,2, and 3 in the new version of program MFIRE v1.30 and v2.10.

A data set Vibbase1.mfi (NSWT=2 --- cubic spline interpolation) is applied for the investigation. At case 1, (NPLOT=1), the network calculation gets convergency. At case 2, if the airflow rate passing through fan is larger than that of the most right data point of the fan characteristic, the fan pressure is equal to that of the data point. It means that the slope of the right side of the outside region of the fan curve is equal to zero. The convergency cannot be achieved at 200 iterations because the discontinuity of the first order derivative at point C. The intermediate results show that the influence of the discontinuity is not equal shared on each fan curve fitting. The divergence of network calculation is caused by three out of the total eight fans (fan 2, 6 and 8). The ΔQ_2, ΔQ_6, and ΔQ_8 cannot be reduced during iteration and the airflow rates (CFM) of the three meshes in the iteration have large fluctuation as table 1 shown:

fan No.	Iteration times								
	3	4	5	6	....	197	198	199	200
2	37862	25465	38144	25569	....	38284	25000	38285	25001
6	58686	38363	59332	38678	....	59134	39398	59132	39450
8	53193	34280	53256	34303	....	53103	34862	53099	34854

Table 2. The fluctuation of airflow rates during iteration

The data in Table 2 shows that i. the convergency cannot be achieved even the maximum iteration time is increased because of the data recirculation. ii. The fluctuation of data will cause a big problem in calculation results. The final outputs will shows the airflow rates for fans 2, 6, and 8 are 38285, 59132, and 53099 CFM but 25001, 39450, and 34854 if one chooses the maximum iteration number 199 instead of 200. Consequently, the results are much different only because of the different iteration time chosen. This is why the divergence sometimes may cause very large error in the distribution of airflow rate, temperature, and contaminant. In the data recirculation situation, the iteration error of network balancing is always larger than 10000 CFM per mesh, compared to the iteration criterion 10 CFM per mesh.

In Case 3 (NPLOT=3), the computing convergency is achieved. The calculation results are similar as case 1. The analysis of the intermediate printout shows 169 iterations needed to achieve the convergence for network calculation and there is large fluctuation of computing airflow rates before the 160-th iteration as case 2. The reliability of convergency here should be questioned. The discontinuity of the first order derivative at the turning points B and C cause this problem. The influences of the discontinuity on different data files are different. The investigation shows that the divergence is only caused by a few data files. The pattern of case 1 may be the best method for most data files. It may, however, gain the negative pressure when the operating point is at the right outside region of fan curve as the cases marked (1)* in table 3. In order to meet the convergence requirements of the fan curve fitting for various data sets, the flexible selection of boundary region handling is offered in the modified versions of program MFIRE v1.30 and v2.10.

3.4. The analysis on computing convergency with the applied fan curve fitting method

From given data points of fan characteristics, the fan curve fitting is to search some approaches to offer an approximating function. There are two basic schemes, the first scheme requires that the approximating function passing through every data point (interpolation method such as Lagrange or cubic spline interpolation). The alternate approach is to find a simple function that applies over the total range of known data points of the fan characteristic but does not necessarily to satisfy every data point (curve fitting method such as least squares method)[8].

For the Lagrange interpolation, the n+1 data points of the fan characteristic can interpolate a polynomial of degree n. This method allows simple programming and gives a smooth fit of the fan curve. It is applied for fan curve fitting at the early stage of ventilation network calculation. It should be noted, however, that polynomial interpolation of this type can be dangerous toward the center of regions where the independent variable is widely spaced. Although the polynomial is "tied down" at the data points, it is free to wonder, possibly excessively, between widely spaced data points. To improve the accuracy of fan curve fitting, the number of data points is often increased. Large fluctuations may be induced between the data points because of the higher degree of the formed polynomial, caused by the large number of data points. Therefore, the order of polynomials, formed by applying the least squares fitting or cubic spline interpolation, can be determined by user. The two methods become the most popular approaches for fan curve fitting in ventilation network calculation.

The method of least squares is to seek an approximating function such that the sum of the squares of the difference between the function and the actual data is a minimum. The key for the application of the least squares approach is the choice of the degree of polynomial to be used for the fitting of data. It is a difficult job. The best situation is one in which it is already known that the data should fall on a polynomial of a given degree. This degree of the polynomial is then the obvious choice. Qualitative judgements can often be made by examining the data. For example, if the data appear to contain one inflection, then a cubic is the obvious choice. The degree of polynomial generated by least square interpolation is always chosen from three to five to prevent the fluctuation of fitted curve. The least squares approach can reduce the influence of a few wrong data points on curve fitting because the fitted curve does not necessarily pass through those data points as interpolation method does. This is the important advantage for the experiment data handling. If the data points of fan characteristic, however, are picked up from the fan curve offered by manufacturer or fan test in the field with the picked data correction, the data points of the fan curve are picked up correctly. The above advantage of least squares method is not obvious.

Among the spline interpolation methods, the cubic spline interpolation is the most popular approach applied in engineering. Technically, the failure of smoothness is in the pronounced discontinuity of first derivative. Thus, at each data point, the slope of the spline can change abruptly from one value to another. One joins cubic polynomials together so that the resulting spline function S has two continuous derivative S' and S" everywhere. At each data point, three continuity conditions will be imposed. Since S, S', and S" are continuous, the graph of the function will appear smooth to eye. Discontinuities, of cause, will occur in the third derivative but cannot be detected visually, which is one reason for choosing degree three. Experience has shown, moreover, that seldom is any advantage gained by using spline of degree greater than three [6,8].

The investigation shows that no fan curve fitting method being perfect for every data file. The modified source code of MFIRE offer 4 kinds of choice to make the selection of fan curve fitting more flexible. i. least squares fitting (NSWT=1); ii.cubic spline interpolation (NSWT=2); iii.automatical selection by program following some standards of curve fitting selection (NSWT=3); iv. combination of automatical and user selection (NSWT=4). In the fourth case, the program offers the selection as reference then user make choice according to the reference and actual fitting results from above methods.

The practical application of program MFIRE reveals that the cubic spline interpolation is sensitive to the uneven distribution of data points. The spline method will have more interpolation error than least square method if the data points show the tendency of forming a straight line approximately. The modified program selects the least squares method automatically if the proportion of maximum and minimum spread between data points of a fan characteristic is larger than four or the data points show the fitting curve being about a straight line.

As Fig.3 shows, in the modified MFIRE, the fluctuation DIFH of each two successive data point is added to gain THSUM after applying the cubic spline and least square methods respectively. The two fitting errors of THSUM from above two methods, which are calculated from equation (2), are compared. The curve fitting method offering less fitting error is selected for the corresponding fan. If user switches to automatical selection (NSWT=3), the results of comparing fitting error THSUM is applied for choice of fan curve fitting method. If the user prefers to choose the combination of automatic and manual selection (NSWT=4), the comparing results are applied as reference. The results of fan curve fitting by employing least square and cubic spline methods are outputted on the screen (there are five data points being printed out between two input data points). The user make final decision from the reference and the data of fan curve fitting results are printed on the screen.

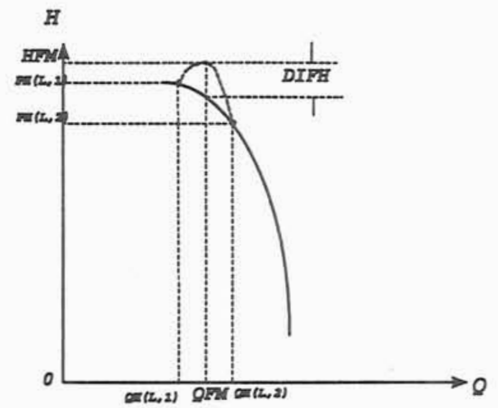

Fig.3 Fluctuation calculation in fan curve fitting

$$THSUM = \sum_{i=1}^{n-1} |DIFH(i)| = \sum_{i=1}^{n-1} \left| HFM - \frac{PH(L,i)+PH(L,i+1)}{2} \right| \tag{2}$$

where: n: the number of data points

i: the data points of fan characteristic

HFM: the pressure, which is given by the fan curve fitting, corresponds to the airflow rate QFM, which is the airflow rate of the middle point between the two successive data points of fan characteristic.

DIFH(i): the fluctuation of fan curve fitting for the middle point in the i-th section of fan curve.

THSUM: the summary of the fluctuation of the fan curve fitting.

		former program **	NSWT=1***			NSWT=2			NSWT=4		
			NPLOT=			NPLOT=			NPLOT=		
			1	2	3	1	2	3	1	2	3
testdata .dat 2 fans	A	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	B	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	C	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
mult-lev .dat 1 fan	A	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	B	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	C	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
code93. mfi 1 fan	A	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	B	no	yes	yes	yes	yes	yes	yes	yes	yes	yes
	C	no	yes	yes	yes	yes	yes	yes	yes	yes	yes
xt101. dat 2 fans	A	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	B	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	C	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
f94-10. bas 2 fans	A	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	B	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	C	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
vibbase1 .mfi 8 fans	A	yes	yes	yes	yes	yes	no	yes	yes	no	yes
	B	yes	yes	yes	yes	yes	no	yes	yes	no	yes
	C	yes	yes	yes	yes	yes	no	yes	yes	no	yes
vibbase2 .mfi 8 fans 1 fan reverse	A	yes	yes	yes	yes	yes	no	no/yes	yes	no	yes
	B	no	yes	yes	yes	yes	no	yes	no	no	no
	C	no	yes	yes	yes	yes	no	yes	no	no	no
brushy1. mf1 5 fans i fan reverse	A	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	B	yes	(1)*	yes	yes	(1)	yes	yes	(1)	yes	yes
	C	yes	(1)	yes	yes	(1)	yes	yes	(1)	yes	yes
brushy2. mfi 5 fans 2 fans reverse	A	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
	B	stop	yes	yes	yes	yes	yes	yes	yes	yes	yes
	C	———	yes	yes	yes	yes	yes	yes	yes	yes	yes
brushy3.mfi as 1 but fan 3 changed	A	no	yes	yes	no	yes	yes	no	yes	yes	no
	B	no/stop	yes	yes	no/yes	yes	yes	no	yes	yes	no/yes
	C	———	yes	yes	yes	yes	yes	no	yes	yes	yes
brushy4. mfi as 2 but fan 3 changed	A	no	yes	yes	no	yes	no/yes	no	yes	yes	no
	B	no/stop	yes	yes	no/yes	yes	yes	no/yes	yes	yes	no/yes
	C	--------	yes	yes	yes	yes	yes	yes	yes	yes	yes
brushy5. mfi as 4 but 3 fans reverse	A	no	yes	yes	no/yes	yes	no/yes	no	yes	yes	no
	B	no	yes	yes	yes	yes	yes	no/yes	yes	yes	no/yes
	C	no	yes	yes	yes	yes	yes	yes	yes	yes	yes

Table 3. The Investigation of the computing Convergence

Explanation to table 3

(1)*. The fan operating point is converged to negative pressure.

** The former program means the program MFIRE (v.1.29) with NSWT=2, NPLOT=3.

*** NSWT is the array in the modified program MFIRE for the selection of fan curve fitting method:

1--least square fitting method, 2--cubic Spline interpolation,

3--automatic choice, 4--automation+user choice.

The no/yes means that the computing convergence is not achieved in the first part but achieved in the second part. In the dynamic simulation B. the time interval of the system condition change (adding or stopping fan) is often the demarcation line for the two parts. In the steady state simulation, the first and second parts are network balancing and temperature part respectively.

NPLOT is the switch of handling the boundary region of fan curve as shown in Fig.2.

Cases A, B, and C are steady state, nonsteady state, and quasi-equilibrium state simulation respectively.

3.5 Case study on the relation between computing convergency and the fan curve fitting or boundary region treatment of fan curve.

Table 3 shows the investigating results of the computing convergency when different data sets are applied. The results focus on the influence of system condition change, fan curve fitting and treatment of boundary region of fan curve on the computing convergence but not on the influence of programming error. The summary of programming error influence has been introduced separately in the table 1 and the error analysis will be introduced at the later part of this report.

From the investigation of the relation among the computing convergency with the system condition change, fan curve fitting methods and the treatment of the boundary region of fan curve in table 3, following conclusion can be given:

i. The computing convergency can be achieved easily in a ventilation system with a few fans.

ii. The large state variations (fan stop and reverse) in a multi-fan system may induce a divergence problem more easily.

iii. No fan curve fitting method or the treatment of the boundary region of fan curve is perfect for every data set. Usually, either the cubic spline or the least squares methods can give convergence computing results. For applying data file vibbase2.mfi, the least squares method offers better convergence than the cubic spline does. The cubic spline method, however, offers better convergence than the least squares method when applying data file brushy2.mfi.

vi. The new version of Program MFIRE has included the least squares and cubic spline methods, and offers three treatments for the boundary region of the fan curve to offer more flexible choice of approach for improving the computing convergency and simulating accuracy. The cubic spline interpolation and NPLOT=1 are suggested for the fan curve fitting of the fans added in time table.

v. The formal treatment of the boundary region of fan curve in MFIRE (v.1.29) makes the slope of that region equaling to zero to reduce the denominator of equation (1) and increase the correcting increment ΔQ. The increase of ΔQ can speed the convergency for some data sets but may cause calculating fluctuation for other data sets because of the discontinuity of the first order derivative at the turning points B and C in Fig.2.

vi. In data file brushy2.mfi, the computing result from the former program is even better than that of the modified program in some selections of fan curve fitting methods (NSWT=1 and NSWT=4). It means that the spline method is better than the least square method for this data file and the auto - selection of fan curve fitting method in the program is not always reliable for any data set.

4. The Investigation, Analysis and Modification on the Iteration Divergence Caused by Programming Error.

Besides of the methods of the fan curve fitting, treatment of boundary region of fan characteristic curve, the programming error, of course, can induce the iteration divergence too. Data sets cote93.mfi. brushy.mfi, and vibbase.mfi are used for the investigation and reveal that the fan behavior change (fan added, stopped, or reversed) may cause computing divergence.

4.1 The fan behavior change in dynamic state simulation part (data file -- cote93.mfi)

The condition change in fan behavior of data file cote93.mfi offer four cases (case 1 to case 4) for the investigation.

4.1.1 The investigation of computing results

a. Case 1 (cote9305.mfi):

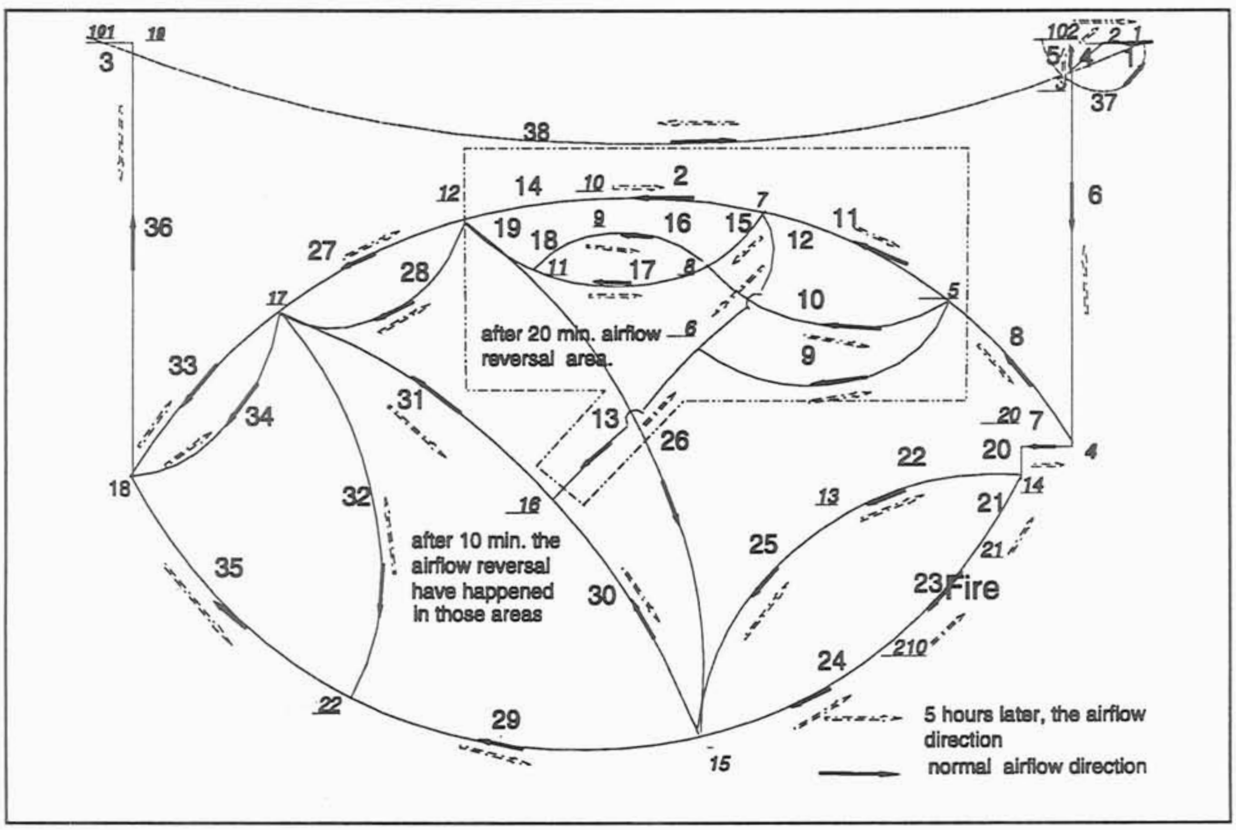

Fig.4 The ventilation network of the data file cote93.mfi

At dynamic simulation part, a blower fan in airway 1 stops and another exhaust fan in airway 5, which is parallel to airway 1, is in operation. Airway 1 is changed to an ordinary airway, fan stops, with resistance equal to 10 (1.E-10 IN.W.G./(CFM)2) at 10 MIN. The resistance of airway 1 is changed to 3755.102, while an exhaust fan start operation in airway 5 at 15 MIN. with fan characteristics specified by four data points as 90000 (CFM), 4.25 (IN.W.G.), 101750,3.025, 113700,1.5, 120000,.30. This change is shown in the following time table of input data set:

10,1,1,10.
15,1,1,3755.102
15,2,5,4,90000,4.25,101750,3.025,113700,1.5,120000,.30

The computing results for the state distribution in 1200 seconds after event are as follows. The calculation did not get convergence in two-level iteration: In the natural ventilation pressure calculation: ten cycles of data preparation, the outer iteration, cannot achieve convergence for natural ventilation pressure calculation (error 1 larger than the criterion $2*10^{-4}$ IN.W.G.). In the airflow distribution calculation iteration: 500 iterations for each cycle cannot achieve convergence for network balancing and the error 2 is larger than the criterion 10 CFM per mesh. The current errors for each preparation cycle at the time interval of 1200 seconds are following:

cycle	1	2	3	4	5	6	7	8	9	10
error1	larger than $2*10^{-4}$ IN.W.G.									
error2	23416	17356	26862	17779	26856	17790	26856	17790	26856	17790

Another problem of the computing result is that the fan operation point (123037 CFM, .298 IN.W.G.) is out of the operating region of fan characteristic.

b. Case 2 (cote9311.mfi)

There is no fan in steady state simulation, then having blower fan in airway 1 in dynamic state simulation: It means that the changes are made as following in data file cote93.mfi:

```
********************************************************************************
********************************************************************************

$NB,NFNUM,INFLOW,NVPN,NETW,T,MADJ,ITN,NTEMP,TINC,SPAN,IOUT,TOUT, CONCT,DR
  38, 0  ,  0 , 0 ,  0 ,90,   ,  , 0 ,300 , 60 , -2 , 5 , 0  .072

 $-------- NO,JS,JF,NWTYP,R,      Q,      KF,LA, A,O,HA,HK,CH4V,CH4PA,TROCK--$
          1   2   0  .000000  203500,,      50 58 30,,,,, 90

$  TIME TABLE FOR CONDITION CHANGE
$-- (IT MUST END WITH A ONE-LINE VALUE LARGER THAN "SPAN" IN CARDI) --$
```

5,3,23,2000,50,5000.
0,2,1,4,180000,8.5,203500,6.05,227500,3.00,235000,1.6

****

From the output, one can see the algorithm could not get convergence at each time interval (in the network calculation, the error is about 20888 CFM per mesh, which is much larger than the convergent criterion of network balancing 10 CFM per mesh. The fan operating point, 263706 CFM,1.6 IN.W.G., is out of the working region of fan characteristic.

c. Case 3 (cote9303.mfi)

It is the same as case 2 except airway 1 being the fixed quantity airway, the added fan being shut down at 10 MIN. and a reverse fan starting operation at 15 Min. in dynamic simulation part.

****
****

```
$-------- NO,JS,JF,NWTYP,R,      Q,KF,    LA,A,  O,HA,HK,CH4V,CH4PA,TROCK---$
          1  2  -1  .000000  203500,,   50  58   30,,,,,90
```

```
$--(10) TIME TABLE FOR CONDITION CHANGES: ---------------- $
5,3,23,2000,50,5000.
10,1,1,10.
15,1,1,3755.102
15,2,5,4,90000,4.25,101750,3.025,113700,1.5,120000,.30
```

****

The computing results are following:

The calculation did not get convergence in the two-level iteration: In the natural ventilation pressure iteration: 10 data preparation cycles cannot achieve convergence. Error 1 larger than the iteration criterion $2*10^{-4}$ IN.W.G. per mesh. In network calculation iteration: 500 iterations for each cycle cannot achieve convergence. Error 2 larger than the requirement of computing criterion for convergence 10 CFM per mesh.

cycle	1	2	3	4	5	6	7	8	9	10
error1	larger than $2*10^{-4}$ IN.W.G.									
error2	23258	26943	17650	26918	17652	26917	17652	26917	17652	26917

Another problem is that the fan operating point (155722 CFM, .298 IN.W.G.) is out of the operating region of fan characteristic curve.

d. Case 4 (cote9302.mfi)

There is no fan in steady state part, then installs a blower fan in airway 1, stops in 10 MIN. and starts a exhaust fan in airway 5 while the resistance of airway 1 changed to 3766.102 (1.E-10 IN.

W.G./(CFM)**2), described as following time table:

5,3,23,2000,50,5000.
0,2,1,4,180000,8.5,203500,6.05,227500,3.00,235000,1.6
10,1,1,10.
15,1,1,3755.102
15,2,5,4,90000,4.25,101750,3.025,113700,1.5,120000,.30
$50,3,23,2000,50,40000.

There are two problems found from the computing result of above data file (at 600 seconds after the event): a). network calculation has not converged. The error is 20889 CFM (comparing with the required criterion 10 CFM per mesh; b). the operating point (263706 CFM, 1.6 IN.W.G.) is in the outside the working region of fan characteristics.

The summary of the influence investigation for ventilation system condition changes (fan added, stopped, and reversed) on the computing convergence is shown in Table 4.

	divergence		1200 sec.*	1500 sec.	1800 sec.	2100 sec.	....	3600 sec.
case1	iteration error (CFM/per mesh)		17790	25618	23816	23778	....	23856
	opt. point	P inwg	0.298	0.300	0.300	0.300	....	0.300
		Q CFM	123037	149205	146129	146156	....	145993
case2	iteration error (CFM/per mesh)		20889	20889	20889	20889	....	20888
	opt. point	P inwg	1.600	1.600	1.600	1.600	....	1.600
		Q CFM	263706	263706	263706	263706	....	263705
case3	iteration error (CFM/per mesh)		26917	21289	23581	23546	....	23626
	opt. point	P inwg	0.298	0.300	0.300	0.300	....	0.300
		Q CFM	155722	149543	146630	146655	....	146488
case4	iteration error (CFM/per mesh)		(600 sec.) 20889					
	opt. point	P inwg	1.600					
		Q CFM	263706					

*: the iteration error and operation point of fan in the time interval of 1200 seconds after fire breaks out

Table 4. The influence investigation of system condition change on computing convergence

4.1.2 The result analysis and modification

The case study of case 1 to case 4 shows that the fan behavior change induces some large different distributions of airflow rates, temperature, and contaminants. The former versions of programs MFIRE v1.29 and v2.01/v2.0 will reform the meshes pattern for ventilation network calculation only in airflow reversal case but not in the case of the large variation of airflow distribution. The primary airway selection and meshes formation were based on the product of the resistances R and the airflow rates of airways Q which is the former airflow distribution but not the changed airflow distribution. From the application principle of Hardy Cross method, the former selected primary airways and corresponding pattern of meshes cannot be suitable to the new airflow distribution, and therefore, cannot improve the convergence of the algorithm. The programming error on the meshes reformation induces the computing divergence for case 1 to case 4.

In the modified program MFIRE v1.30 and v2.10, if some system condition changes happen in a ventilation system (such as fan added, fan stop and reverse, regulators installed) as time table shown, The flag INIFAN is sent to 1 at the subroutine CDCH used to handling the time table data. The airways rearrangement and meshes reformation will be started again to improve the computing convergence for network calculation. A new subroutine ITR1 in version 1.30, which adds the mesh reformation part to subroutine ITR, is appended. In version v2.10, the mesh reformation part is added to ITR directly. The modified program MFIRE offers correct computing results and achieves the algorithm convergence as Table 5 shown which is applied same data set as Table 4.

	divergence		1200 sec.	1500 sec.	1800 sec.	2100 sec.	....	3600 sec.
case 1	iteration error (CFM/per mesh)		< 10	< 10	< 10	< 10	....	< 10
	opt. point	P inwg	1.228	1.316	1.329	1.335	....	1.331
		Q CFM	115195	114756	114688	114659	....	114671
case 2	iteration error (CFM/per mesh)		< 10	< 10	< 10	< 10	....	< 10
	opt. point	P inwg	5.959	5.959	5.959	5.959	....	5.959
		Q CFM	204370	204370	204370	204370	....	204370
case 3	iteration error (CFM/per mesh)		< 10	< 10	< 10	< 10	....	< 10
	opt. point	P inwg	1.227	1.316	1.329	1.335	....	1.332
		Q CFM	115197	114757	114688	114659	....	114671
case 4	iteration error (CFM/per mesh)		(600 sec.) < 10					
	opt. point	P inwg	5.959					
		Q CFM	204371					

Table 5 The computing results of modified program MFIRE

4.2 The error caused by fan characteristic changed (data files brushy.mfi)

The data set brushy.mfi is a file about a mine ventilation system with five operating fans and the characteristic of fan 3 is changed to offer cases E and F.

4.2.1 Case E (data file brushy1.mfi)

The iterations in network balancing and natural ventilation calculation are converging

4.2.2 Case F (Data file brushy3.mfi)

The fan characteristic of fan 3 in airway 3 of data file brushy1.mfi is changed

from 3,3

10000,8.1,16600,1.27,17000,.10

to 33

20000,8.1,26600,1.27,27000,.10

The computing result shows that the network calculation cannot get convergence at the steady state, dynamic, and quasi-equilibrium state parts and the computed operating points of the fans are wrong:

		steady state	dynamic state simulation					quasi-equilibrium state
			120.S	240. S	360.S	480.S	600.S	
case F	error CFM	1347	2535	2534	2569	2577	< 10	271
	Outside operating range ?	yes	yes	yes	yes	yes	no	yes

Table 6 The former computing results for case F

From the investigation of table 6 and table 3, in the column of the former program - brushy3.mfi and the column of the modified program, the case of NSWT-2 and NPLOT=3, they give same divergence results. It means that the divergence is not caused by the programming error. Selecting suitable methods for the treatment of boundary region of fan curve (NPLOT) and fan curve fitting (NSWT) improve the convergence behavior as follows:

		steady state	dynamic state simulation					quasi-equilibrium state
			120.S	240. S	360.S	480.S	600.S	
case E	error CFM	< 10	< 10	< 10	< 10	< 10	< 10	< 10
	Outside operating range ?	no	no	no	no	no	no	no

Table 7 The new computing results for case F

In the row of brushy3.mfi of table 3, the other choices except NSWT=2 and NPLOT=3 can achieve the computing convergence as shown in Table 7.

4.3 The error caused by the data handling in the time table (data file vibbase.mfi)

In data file vibbase.mfi, eight fans are operated in a mine ventilation system. One of the eight fans is reversed to form two different cases G and H for investigation.

4.3.1 Case G (date file vibbase1.mfi)

There is no fan behavior change in the data set. The computing convergence is achieved as table 8 shows.

		steady state	dynamic state simulation					quasi-equilibrium state
			720.S	840. S	960.S	....	1800.S	
case G	error CFM	< 10	< 10	< 10	< 10	....	< 10	< 10
	Outside operating range ?	no	no	no	no	....	no	no

Table 8 The former computing results for case G

4.3.2 Case H (date file vibbase2.mfi)

There are eight fans operating in the steady state part, then, fan in airway 1 stop and a fan starts running in airway 10 that is in parallel with airway 1, the case of fan 1 reversed:

The problems in following results are i). network calculation cannot get convergence; ii). after fan in airway 1 stop, the fan curve fitting for the fans in airways 2,3,4,5,6,7,8 are wrong. The fan with sequential number n picks up the data points in characteristic of a fan with sequential number n-1. iii). some operating points of fans are in the outside the working range and the computing state distribution is wrong. The wrong computing results are shown as the table 9 and 10.

a. The computing divergence and operating point being the outside the operating region of fan curve

		steady state	dynamic state simulation					quasi-equilibrium state
			720.S	840. S	960.S	....	1800.S	
case H	error CFM	< 10	33750	18481	25979	....	48178	< 10
	Outside operating range ?	no	yes	yes	yes	....	yes	no

Table 9 The former computing results for case H

b. Pick up a wrong fan characteristic curve

Because of the wrong handling in system condition change in the time table (dynamic part), the picked fan characteristics are totally wrong:

There are eight fans in the ventilation system.

```
1,3
30000,2.75,31800,1.25,32500,.5
2,3
30000,8.0,35500,2.7,36500,1.0
3,3
35000,7.1,52400,0.5,53000,.20
4,3
30500,4.00,33500,0.6,34000,.1
5,3
22500,6.0,25500,2.3,27000,0.2
6,3
40000,6.5,52900,1.27,54000,.5
7,3
27900,6.0,31500,1.05,33000,0.80
8,3
40000,5.4,48900,1.3,50000,0.4
```

The time table shows that the fan in airway 1 will stop and reverse (same fan starts at airway 10) in 10 minutes:

```
4,3,74,200,50,5000
10,1,1,10.0
10,2,10,3,30000,2.75,31800,1.25,32500,.5
```

The output of the dynamic part is following:

****
****

TIME AT 720. SEC. AFTER EVENT

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	DELTAQ	AIRFLOW	AVE.T	T AT END	FUMES	CH4	HEADLOSS
1	21	3	-2783.	29059.	57.60	57.59	0.0000	0.00	0.837
2	23	5	-12575.	22965.	57.66	57.65	0.0000	0.00	2.762
3	26	9	26254.	78704.	57.07	57.06	0.0000	0.00	1.006
4	27	13	171372.	204890.	59.25	59.23	0.0000	0.00	0.200
5	28	14	-596.	24947.	57.57	57.56	0.0000	0.00	4.018
6	33	20	-31014.	21947.	57.79	57.78	0.0000	0.00	6.025
7	30	17	57808.	89385.	57.66	57.65	0.0000	0.00	0.502
8	31	18	-6620.	42334.	57.77	57.76	0.0000	0.00	0.803
9	21	1	68188.	38326.	56.40	55.00	0.0000	0.00	1.142
10	3	21	65405.	67385.	57.47	57.36	0.0000	0.00	0.502

****

One will find that the picking up of fan characteristic curve is wrong. Fan 2 in airway 2, because of NOF(N) equal to NO(N) and N, has the operating point (22965, 2.762) from above output results. The operating point is, however, the left boundary of the characteristic of fan 1 in data file vibbase2.mfi. Fan 3 get the operating point (78704, 1.006) from above output results. The operating point is at the right boundary of characteristic of fan 2 in the data file vibbase2.mfi. For same reason, the fan n gets the fan characteristic of fan (n-1). It is worth to notice that the pressure drafts (2.762 or 1.006) have a little difference from the corresponding pressure drafts of boundary region of fan characteristics 1 and 2(2.75 or 1.000) because of the density difference.

Same mistake is discovered in the calculation of data file brushy1.mfi: There are five fans in the ventilation system:

```
1,3
50000,8.35,74000,4.06,81100,.9
2,3
12000,6.0,33100,1.17,35300,.3
3,3
10000,8.1,16600,1.27,17000,.10
4,3
140000,5.90,275000,1.30,292000,0.20
5,3
65000,9.25,96700,2.20,100000,.5
```

The computing results by former program MFIRE (v1.29) are as follows:

****
****

TIME AT 720. SEC. AFTER EVENT

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	DELTA Q	AIRFLOW	AVE. T	TAT END	FUMES	CH4	HEADLOSS
1	29	28	-3756.	71103.	54.32	54.33	0.0000	0.00	5.126
2	27	26	-5015.	28840.	55.05	55.07	0.0000	0.00	2.631
3	21	22	6842.	23523.	59.48	59.47	0.0000	0.00	0.100
4	6	8	41306.	28185.	64.01	64.01	0.0000	0.00	2.167
5	31	17	-20383.	126629.	54.41	54.41	0.0000	0.00	6.174

****

Looking on above results, the operating point in airway 5 is with q=126629 CFM and h=6.174 INCH. W.G.. Checking the data points of the fan characteristic in data set brushy1.mfi, one will find that it is not in the 5th fan characteristic but in the 4th fan characteristic curve. This mistake will spread to following time interval until reach 3600. S. The operating points are always in the outside region of fan curve.

4.3.3 The analysis and modification

Subroutine CDCH is applied to handling the time table in input data set for system condition changes. If flag ICODE equal to 1, it means that there are a fan stopping operation and the corresponding airway changing to an ordinary airway. For the time table in brushy1.mfi, the fan in airway 4 will be stopped and airway 4 is changed to be an ordinary airway in eight minutes after mine fire occurs. In the former program MFIRE (v1.29), the subroutine CDCH makes the number of fan in operation reduced by 1 (5-1=4). The related arrays, such as NOF(4) -- airway number installed fan, MPTS(4) -- number of data points defining fan characteristic, and NSKP(4) -- marker indicating performance of spline method, are replaced by the former values of NOF(5), MPTS(5), and NSKP(5). The array NOF(5), MPTS(5), and NSKP(5) are sent to zero to end the input of system condition change. The former program MFIRE (v1.29) has not made the corresponding changes for input data points of fan characteristic (PF, QF) and the coefficients of fan equation array FKQ. That is why the wrong data points of fan characteristic are picked up to fit a wrong fan curve, and then induce the computing divergence in above cases.

The modification for this problem in subroutine CDCH is shown and marked as follows:

```
          IF (NFNUM.GT.0) THEN
             DO 8 J=1,NFNUM
                IF (NOF(J).EQ.NBR) THEN
                   IF (NFNUM.GT.J) THEN
                      DO 14 K=J,NFNUM-1
                         NOF(K)=NOF(K+1)
                         MPTS(K)=MPTS(K+1)
                         NSKP(K)=NSKP(K+1)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
                DO 7 KK=1,IMX
                   QF(K,KK)=QF(K+1,KK)
                   PF(K,KK)=PF(K+1,KK)
7               CONTINUE
                IF (NSWT(K).EQ.2) THEN
                   MF=MPTS(K+1)
                   DO 12 II=1,MF-1
                      FKQ(K,II,1)=FKQ(K+1,II,1)
                      FKQ(K,II,2)=FKQ(K+1,II,2)
                      FKQ(K,II,3)=FKQ(K+1,II,3)
                      FKQ(K,II,4)=FKQ(K+1,II,4)
12                 CONTINUE
                ELSEIF (NSWT(K).EQ.1) THEN
                   NCOF(K)=NCOF(K+1)
                   N1A=NCOF(K)
                   DO 13 II=1,N1A
                      COF(K,II)=COF(K+1,II)
13                 CONTINUE
                ENDIF
14           CONTINUE
          ENDIF
          INIFAN=1
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC C
        NOF(NFNUM)=0
          MPTS(NFNUM)=0
          NSKP(NFNUM)=0
          NFNUM=NFNUM-1
          GO TO 9
        ENDIF
     CONTINUE
  ENDIF
```

4.3.4 The computing results of the modified program MFIRE V1.30

The computing convergence of modified program is shown as follows:

		steady state	dynamic state simulation					quasi-equilibrium state
			720.S	840. S	960.S	....	1800.S	
case F	errorCFM	< 10	< 10	< 10	< 10	....	< 10	< 10
	outside of operating range ?	no	no	no	no	....	no	no

The mistake at the picking up of fan data points have been corrected as follows:

TIME AT 720. SEC. AFTER EVENT

****

****

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	DELTA Q	AIRFLOW	AVE.T	T AT END	FUMES	CH4	HEADLOSS
1	21	3	0.	22517.	57.60	57.59	0.0000	0.00	0.502
2	23	5	0.	35672.	57.66	57.65	0.0000	0.00	2.433
3	26	9	0.	52627.	57.07	57.06	0.0000	0.00	0.389
4	27	13	0.	33684.	59.25	59.23	0.0000	0.00	0.414
5	28	14	0.	25655.	57.57	57.56	0.0000	0.00	2.101
6	33	20	0.	53188.	57.79	57.78	0.0000	0.00	1.077
7	30	17	0.	32004.	57.66	57.65	0.0000	0.00	0.871
8	31	18	0.	49145.	57.77	57.76	0.0000	0.00	1.111
9	21	1	0.	30350.	56.40	55.00	0.0000	0.00	0.716
10	3	21	0.	52867.	57.47	57.36	0.0000	0.00	0.502

****

Comparing above two computing results (former and modified results at 720 seconds after event of mine fire), one can find the former computing distributions of pressure drafts and airflow rates are wrong. The operating points being at the boundary region of fan curve is caused by the error in the former program but not a unsuitable fan installed to the ventilation system.

The modified program MFIRE gives correct computing results for the data file brushy1.mfi too. The fan in airway 5 picks up the correct fan characteristic and the operating point is in the operating region of fan characteristic.

****
****

TIME AT 720. SEC. AFTER EVENT

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	DELTAQ	AIRFLOW	AVE. T	T AT END	FUMES	CH4	HEADLOSS
1	29	28	-3925.	71375.	54.32	54.33	0.0000	0.00	5.039
2	27	26	-5232.	29209.	55.05	55.07	0.0000	0.00	2.519
3	21	22	6206.	22797.	59.48	59.47	0.0000	0.00	0.100
4	6	8	49080.	27049.	64.01	64.01	0.0000	0.00	1.996
5	31	17	-4145.	92943.	54.41	54.41	0.0000	0.00	3.904

****

5. The Investigation, Analysis of the Other Problems in State Simulation And Source Code Modification

Besides of inducing divergence problem and offer wrong results, program error can cause the problems on the distribution of airflow rate, temperature, and contaminant. The influence will be investigated at quasi-equilibrium or dynamic state parts respectively. The reasons of programming error, source code modification and the computing results of the modified program are shown as follows:

5.1. The wrong distribution of airflow rates, temperature, and contaminant in quasi-equilibrium state part when airflow reversal is happening:

As shown in table 1 (cote93.mfi cases A*** and B***), Program MFIRE give following output results for quasi-equilibrium state simulation:

****
****

OUTPUT OF THE QUASI-EQUILIBRIUM SIMULATION PART

TIME AT 5.0 HOUR(S) AFTER EVENT

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	AIRFLOW	AVE. T	T AT END	FUMES	METHANE	HEADLOSS
3	101	19	112738.	86.57	86.57	0.48268	0.00	0.0000
4	2	3	1888.	90.14	90.27	0.00000	0.00	0.0001
5	3	102	114626.	90.12	89.98	0.00000	0.00	1.3386
6	4	3	112738.	93.10	95.67	0.00000	0.00	1.0673
7	20	4	42453.	95.71	95.75	0.00000	0.00	0.0865
20	14	20	42453.	96.07	96.39	0.00000	0.00	0.0035
21	21	14	33734.	96.33	96.28	0.00000	0.00	0.0104
23	210	21	33734.	101.29	101.29	1.67404	0.00	0.0000
24	15	210	33734.	99.78	98.46	1.67404	0.00	0.0245
26	15	12	16115.	93.75	93.68	0.00000	0.00	0.0012
29	22	15	24257.	95.64	94.98	0.93964	0.00	0.0033
30	16	15	34311.	95.72	95.10	0.93964	0.00	0.0003
31	17	16	34862.	94.75	94.45	0.92673	0.00	0.0013
32	22	17	37738.	93.19	93.13	0.35814	0.00	0.0016
33	18	17	28190.	93.11	93.03	0.35814	0.00	0.0080
34	18	17	22554.	93.08	93.01	0.35814	0.00	0.0080
35	18	22	61995.	93.65	93.48	0.58460	0.00	0.0064
36	19	18	112738.	89.65	86.56	0.48268	0.00	0.2660

PARAMETERS OF AIR IN JUNCTIONS

JUNCTION	TEMP.	FUMES	METHANE	JUNCTION	TEMP.	FUMES	METHANE
1	90.00	0.00000	0.00	2	90.00	0.00000	0.00
3	90.27	0.00000	0.00	4	95.67	0.00000	0.00
13	96.08	0.00000	0.00	14	96.39	0.00000	0.00
15	96.44	0.93964	0.00	16	95.09	0.92673	0.00
17	93.26	0.35814	0.00	18	93.27	0.48268	0.00
19	86.57	0.48268	0.00	20	95.75	0.00000	0.00
21	96.28	0.00000	0.00	22	93.85	0.58460	0.00
101	89.00	0.00000	0.00	102	90.00	0.00000	0.00
210	101.29	1.67404	0.00				

IN THE FOLLOWING AIRWAYS EXIST CRITICAL CONDITIONS

AIRWAY	FROM	TO	CH4 %	FUMES %	TEMPERATURE	HEADLOSS
			> 1.000	> 0.05000	> 100. F	< 0.01000 IN.WG.
3	101	19	0.00	0.48268	86.6	0.00000
4	2	3	0.00	0.00000	90.3	0.00012
9	6	5	0.00	0.00000	93.7	0.00931
12	7	6	0.00	0.00000	93.2	0.00279
13	16	6	0.00	0.00000	94.3	0.00646
14	12	10	0.00	0.00000	94.0	0.00217
15	8	7	0.00	0.00000	93.2	0.00031
16	9	8	0.00	0.00000	92.7	0.00031
17	11	8	0.00	0.00000	92.6	0.00155
18	11	9	0.00	0.00000	93.0	0.00124
19	12	11	0.00	0.00000	92.1	0.00031
20	14	20	0.00	0.00000	96.4	0.00347
23	210	21	0.00	1.67404	101.3	0.00000
24	15	210	0.00	1.67404	98.5	0.02453
26	15	12	0.00	0.00000	93.7	0.00117
27	17	12	0.00	0.00000	92.4	0.00283
28	17	12	0.00	0.00000	92.7	0.00283
29	22	15	0.00	0.93964	95.0	0.00326
30	16	15	0.00	0.93964	95.1	0.00033
31	17	16	0.00	0.92673	94.4	0.00133
32	22	17	0.00	0.35814	93.1	0.00160
33	18	17	0.00	0.35814	93.0	0.00804
34	18	17	0.00	0.35814	93.0	0.00804
35	18	22	0.00	0.58460	93.5	0.00644
36	19	18	0.00	0.48268	86.6	0.26605
37	102	1	0.00	0.00000	90.0	0.00000
38	1	101	0.00	0.00000	90.0	0.00000

IN THE FOLLOWING JUNCTIONS EXIST CRITICAL CONDITION

```
JUNCTION CH4  % FUMES % TEMP. F     JUNCTION CH4  % FUMES % TEMP. F
         .00 >     0.050   > 100.                  > 1.00   > 0.050   >  100.

  15     0.00  0.93964    96.4        16     0.00  0.92673    95.1
  17     0.00  0.35814    93.3        18     0.00  0.48268    93.3
  19     0.00  0.48268    86.6        22     0.00  0.58460    93.9
 210     0.00  1.67404   101.3

*************************************************************************
```

From above output, one can see the fume and temperature distribution in airways and junctions are totally wrong: the airflow reversal has happened five hours ago and the airflow direction is from junction 101 to 19 to 18 to 22 to 15 to 210 to 21 to 14 to 4 to 3 to 102 in Fig.4. The distribution of airflow rates, temperature and fume concentration in airways and junctions, however, is same as the case of the reversal having not happened. The airways and junctions in the upstream area (airway 3, airways 29 - 36) of fire sources (airway 21) contains hot fume, but the airways and junction at the downstream area do not contain any fume.

The reason is that if the quasi-equilibrium state needs once only at the outer-level iteration, for natural ventilation calculation, to achieve the convergence ((SUMFNV / MNO) .LE.2. E-4), the statement, IF (Q(I) .LT. 0.0) GO TO 40, in quasi-equilibrium part will make the calculation jump over the correction of data. When the airflow reversal happens in an airway, the parameters, such as starting and ending junctions of that airway cannot be changed and cause the wrong computing results. After the distribution of airflow rates, temperature, and contaminant have been calculated, they will not be calculated repeatedly as the calculation for each time interval in dynamic state simulation part. The computing error will be kept in the final output of quasi-equilibrium state part. Any data sets will cause the similar problem during airflow reversal if only one iteration is needed for natural ventilation pressure calculation. The statement (IF(MADJC.EQ.1) GO TO 124) means that if the counter of iteration time (MADJC) is equal to 1, then the computing should be forced back to the subroutine FWCT for data rearrangement of the airways in which airflow reversals are happening.

The corresponding modified part of source code is shown as follows:

```
C    TRANSIENT-STATE SIMULATION PROGRAM FOR MINE VENTILATION
C               UNDER THE INFLUENCE OF FIRES
C
C               VERSION OF MAY 1992 V 1.29
C                      (PART 1)
C
C<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
    IF  (MADJC.LT.MADJ) THEN
      MADJC=MADJC+1
      ITCT=0
      CALL ITR (MARKX,NSFLOW,1,MADJC,ITCT)
      WRITE (6,870) MADJC,SUMFNV/MNO
      IF (MADJC.EQ.1) GO TO 124
      IF ((SUMFNV/MNO).LE.2.E-4) GO TO 130
```

```
124     IF (MADJC.GE.(MADJ-9)) THEN
          J=MADJC-MADJ+10
          HRM(J)=SUMFNV/MNO
          DO 125 I=1,NB
            QRCD(I,J)=ABS(Q(I))
            IF (JSB(I).NE.IABS(JS(I)).AND.Q(I).GT.0.0)
     .        QRCD(I,J)=-ABS(Q(I))
            IF (JSB(I).EQ.IABS(JS(I)).AND.Q(I).LT.0.0)
     .        QRCD(I,J)=-ABS(Q(I))
            TMRCD(I,J)=TMRD(I)
125       CONTINUE
        ENDIF
        ITCT=0
        DO 127 I=1,NB
          IF (Q(I).LT.0.0) GO TO 40
127     CONTINUE
        NSFLOW=1
        GO TO 70
      ENDIF
130   CALL FWCT (NSFLOW,0)
      IF (MARKY.LE.0) WRITE (8,620)

*  *  *  *  *  *  *  *  *  *  *  *  *  *
```

The modified program give correct fume and temperature distribution as follows:

TIME AT 5.0 HOUR(S) AFTER EVENT

THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS PER MESH < 2.E-4 IN.W.G.) SATISFIED. CURRENT SUMFNV PER MESH 0.000000 IN.W.G., ITERATIONS 2

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	AIRFLOW	AVE. T	T AT END	FUMES	METHANE	HEADLOSS
1	1	2	3415.	90.00	90.00	0.00000	0.00	4.3782
2	10	7	36904.	91.44	91.46	0.00000	0.00	0.0000
3	101	19	82782.	89.00	89.00	0.00000	0.00	0.0000
4	2	3	3415.	90.12	90.24	0.00000	0.00	0.0004
5	3	102	86197.	88.96	88.83	1.04496	0.00	4.2500
6	4	3	82782.	91.88	89.04	1.08557	0.00	0.5689
7	20	4	41134.	98.69	98.53	2.43953	0.00	0.0821
8	5	4	41649.	92.54	92.68	0.00000	0.00	0.0336

9	6	5	30032.	92.39	92.67	0.00000	0.00	0.0025
10	8	5	4988.	91.67	91.72	0.00000	0.00	0.0116
11	7	5	6630.	91.38	91.29	0.00000	0.00	0.0087
12	7	6	29778.	91.76	91.97	0.00000	0.00	0.0008
13	16	6	253.	92.85	92.38	0.00000	0.00	0.0014
14	12	10	36904.	91.86	91.42	0.00000	0.00	0.0007
15	7	8	496.	91.51	91.46	0.00000	0.00	0.0001
16	9	8	1512.	91.46	91.44	0.00000	0.00	0.0003
17	11	8	2980.	91.78	91.61	0.00000	0.00	0.0008
18	11	9	1512.	91.73	91.47	0.00000	0.00	0.0010
19	12	11	4491.	92.13	92.03	0.00000	0.00	0.0002
20	14	20	41134.	99.99	98.85	2.43953	0.00	0.0033
21	21	14	31253.	103.98	103.76	3.23666	0.00	0.0092
22	13	14	9880.	93.41	93.39	0.00000	0.00	0.0134
23	210	21	31253.	104.21	104.20	3.23666	0.00	0.0000
24	15	210	31253.	94.22	94.19	0.00000	0.00	0.0206
25	15	13	9880.	93.77	93.43	0.00000	0.00	0.0312
26	12	15	5210.	92.31	92.31	0.00000	0.00	0.0001
27	17	12	33927.	92.82	92.26	0.00000	0.00	0.0024
28	17	12	12680.	92.96	92.43	0.00000	0.00	0.0016
29	22	15	21405.	95.05	94.98	0.00000	0.00	0.0025
30	16	15	14518.	93.41	93.44	0.00000	0.00	0.0001
31	17	16	14771.	93.57	93.39	0.00000	0.00	0.0002
32	22	17	20006.	95.03	94.95	0.00000	0.00	0.0005
33	18	17	23469.	93.86	93.21	0.00000	0.00	0.0056
34	18	17	17902.	93.59	93.05	0.00000	0.00	0.0051
35	18	22	41411.	95.20	95.12	0.00000	0.00	0.0029
36	19	18	82782.	92.52	95.28	0.00000	0.00	0.1465
37	102	1	86197.	90.00	90.00	0.00000	0.00	0.0000
38	1	101	82782.	90.00	90.00	0.00000	0.00	0.0000

5.2. The mistakes in the part of dynamic state simulation (data file cote93.mfi)

5.2.1 The fume concentration in the end junction 21 of airway 23 with fire source

a. case study:

****
****

TIME AT 900. SEC. AFTER EVENT

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY FROM TO DELTA Q AIRFLOW AVE. T T AT END FUMES CH4 HEADLOSS

```
23  210  21  67835.    10776.  107.68  107.68  3.6569  0.00  0.000
```

PARAMETERS OF AIR IN JUNCTIONS

```
JUNCTION   TEMP.  FUMES   METHANE      JUNCTION TEMP.  FUMES   METHANE
   21      96.27  0.0000   0.00           22    93.33   0.0000   0.00
```

****

From above output, one can see that the airflow direction in airway 23 (fire source) is from junction 210 to 21. From the parameters of air in junctions, however, the fume concentration in junction 21 is zero.

b). The program modification and new output:

In subroutine DTTR, the following statements are modified to calculate the fume concentration in time.

```
      ......
      DZRD(I)=-DZRD(I)
      FRNVP(I)=-FRNVP(I)
      FFRNVP(I)=-FFRNVP(I)
      QTP(I)=-QTP(I)
      QQ(I)=-QQ(I)
      Q(I)=-Q(I)
      NUMCT=0
C ADD FOLLOWING STATEMENT:
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 60 J=1,INFLOW
        IF (NO(I).EQ.NCENT(J)) THEN
          DO 50 JJ=1,NB
            IF (JF(JJ).EQ.JNO(JU)) THEN
              NUMCT=NUMCT+1
              SFUME=SFUME+RDPROP(JJ)
            ENDIF
50         CONTINUE
          PROP(JU)=SFUME/NUMCT
        ENDIF
60     CONTINUE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

The correct computing result is given:

****

TIME AT   900. SEC. AFTER EVENT

THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS PER MESH < 2.E-4 IN.W.G.)
SATISFIED. CURRENT SUMFNV PER MESH 0.000048 IN.W.G., ITERATIONS 2

TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY	FROM	TO	DELTA Q	AIRFLOW	AVE. T	T AT END	FUMES	CH4	HEADLOSS
23	210	21	67812.	10718.	105.71	105.71	3.6511	0.00	0.000

PARAMETERS OF AIR IN JUNCTIONS

JUNCTION	TEMP.	FUMES	METHANE	JUNCTION	TEMP.	FUMES	METHANE
21	96.27	3.6511	0.00	22	93.89	0.0000	0.00

As above output shows, the fume concentration in junction 21 has been calculated.

5.2.2 The problem on the determination of average airflow rate at the time interval of airflow reversal:

The average airflow rate QQ in the time interval is the prerequisite condition for calculating the amount of heat released from the fire source, afterwards, is related to the calculation of the airflow rate, temperature, contaminant distribution and other parameters. The formulas in the former version of program MFIRE offer the lower value of the QQ. It is suffered by the offsetting influence of the positive and negative values in following formulas at subroutine DISP, if airflow reverse in airway 1:

QQA = 0.55*Q(I)+0.45*QTP(I) and QQ(I)=0.85*QQA+0.15*QQ1(I). (3)

where: QTP(I) is the initial airflow rate in airway I

Q(I) is the airflow rate in airway I at the end of time interval

QQ(I) is the average airflow rate in the time interval

QQ1(I) is the average airflow rate in the time interval at the last iteration

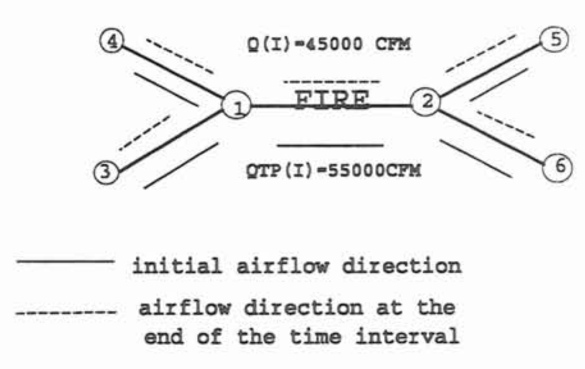

Fig.5 Average airflow rate calculation

The amount of heat released is related to the absolute average value of airflow rate passing through the fire source rather than the algebraic average value of the airflow rate while the airflow reversal is happening. The common formula of the algebraic average value is not suitable for obtaining the average airflow rate. The too small value of the average airflow rate offered by former formula (3) will give unrealistic state simulation for the end junction of an airway with fire source in the time interval and, unfortunately, the error will spread to the airways down stream of the fire source at following time intervals.

Fig. 5 shows the calculation error of former formulas. For example, if QTP(I) and Q(I) ,with opposite direction, are equal to 55000 and 45000 CFM respectively, formula (3), gives QQA=0 and QQ(I)=0.15 * QQ1(I). The value of QQ(I) may be reduced during iteration. It may induce the divergence problem or wrong state simulation. In fact, the initial airflow direction is from junction 1 to 2 and is changed from junction 2 to 1 at one moment of this time interval. The heat release depends on the airflow rate (55000 CFM) and the hot fume flew from junction 2 to 5 and 6 at the beginning of time interval. The heat release then depends on the airflow rate (45000 CFM) and the hot fume flowed from junction 1 to 3 and 4. Consequently, the amount of heat released is closer to the absolute average value of airflow rate in the time interval with airflow reversal happening.. The absolute average value of airflow rate in the airway should be considered to calculate the interval average airflow rate QQ2 for getting the heat release and the fume or the temperature distribution in that time interval. The modified formulas for average airflow rate are as follows:

QQA2=0.55*|Q(I)|+0.45*|QTP(I)| and QQ2(I)=0.85*QQA2(I)+0.15*QQ1(I) (4)

When applying the data set vibbase1.mfi and stopping fan 1 in 10 minutes, the program, version 1.29, stops running at 900 second time interval and the following error message is printed out:

```
 * ERROR *  ABNORMAL TEMPERATURE WAS RESULTED IN AIRWAY    74

   AIRWAY      VART          HEATAD (BTU)    QQ (CFM)       TJS(°F)
     74     8.62173E+08       65000.0         94.8021       19462.8
```

The fire source is in airway 74. Because the algebraic average airflow rate QQ in the airflow reversal time interval at airway 74 is too small, it gives a large value of the variable VART, and finally gives an extra-large value of the varable TJS, the starting junction tempearture of airway 74.

In subroutine DIP: the source code of this part is made following change:

```
DO 60 I=1,NB
  TJS(I)=TTJS(I)
  TRD(I)=TTRD(I)
  RDPROP(I)=RDOP(I)
  RDCH4(I)=RCH4(I)
  QQA=0.55*Q(I)+0.45*QTP(I)
  QQ(I)=0.85*QQA+0.15*QQ(I)
  QQA2=0.55*ABS(Q(I))+0.45*ABS(QTP(I))
  QQ2(I)=0.85*ABS(QQA2)+0.15*ABS(QQ(I))
  IF (CONCT.EQ.1) QQ(I)=QQQ(I)
  NSAC(I,1)=NSACB(I,1)
  NSAC(I,2)=NSACB(I,2)
  IQ=IBTN(I)
```

```
      IF (IQ.GT.0) THEN
        TAUXC(IQ)=TAUXD(IQ)
        DO 50 J=1,10
          TAUXA(IQ,J)=TAUXB(IQ,J)
          DO 50 JJ=1,5
            DPPA(IQ,J,JJ)=DPPB(IQ,J,JJ)
50          CONTINUE
      ENDIF
60    CONTINUE
```

The array QQ2 replaces the array QQ in the most location in program MFIRE. The distribution of airflow rates, temperature and contaminant will be changed by corresponding formulas in the subroutine CDJN.

5.2.3 The output error for airways in which airflow reversal happened

At first, one can compare the following airflow direction in the periods of 300. sec. and 1200. sec.

```
*****************************************************************
*****************************************************************

                    TIME AT   300. SEC. AFTER EVENT

THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS PER MESH<2.E-4 IN.W.G.)
 SATISFIED. CURRENT SUMFNV PER MESH 0.000000 IN.W.G.,  ITERATIONS    2
     TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY FROM  TO DELTA Q  AIRFLOW AVE. T T AT END FUMES  CH4  HEADLOSS

  1    1   2     0.   202781.  90.00  90.00  0.0000  0.00  6.125
  2    7  10     0.   115859.  93.37  93.35  0.0000  0.00  0.000
  3   19 101     0.   199250.  86.47  86.47  0.0000  0.00  0.000
  4    2   3     0.   202781.  90.14  90.27  0.0000  0.00  1.441
  5    3 102     0.     3531.  90.12  89.98  0.0000  0.00  4.684

                    TIME AT  1200. SEC. AFTER EVENT

THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS PER MESH<2.E-4 IN. W.G.)
 NOT SATISFIED. CURRENT SUMFNV: 0.002706 IN.W.G., ITER.    8

     TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS
```

AIRWAY FROM TO DELTA Q AIRFLOW AVE. T T AT END FUMES CH4 HEADLOSS

```
1   2   1 213041.     -3753.  93.94  90.00  0.0000  0.00  -5.365
2  10   7-113389.     56325.  93.37  93.39  0.0000  0.00   0.000
3 101  19 210048.    119341.  86.47  86.47  0.0000  0.00   0.000
4   3   2 213041.     -3753.  91.99  90.00  0.0000  0.00   0.000
5   3 102  -2993.    123037.  93.57  93.19  2.2986  0.00   0.298
```

****

Comparing the airflow distribution in above two time intervals, the airflow reversal has happened in airways 1,2,3,4,5 at 1200 second after event. The following output, however, has not shown airflow reversal happened in airways 2. From the output of the time interval at 1200 sec. after event, the airflow in airway 1 is from junction 1 to 2 and in airway 4 is from junction 2 to 3 (airflow rates are both -3753). The output, however, is still showing as follows:

```
*************************************************************************************************
************************************************************************************
REVERSAL OF AIRFLOW HAS OCCURRED IN THE FOLLOWING PLACES

 AIRWAY    1 IS NOW CARRYING AIR FROM    2 TO    1
 AIRWAY    3 IS NOW CARRYING AIR FROM  101 TO   19
 AIRWAY    4 IS NOW CARRYING AIR FROM    3 TO    2
.

*************************************************************************
```

It means that the outputs are conflict and show a wrong airflow direction in airway 1 and 4, because the negative value of airflow rate in airways 1 or 4 and leads to output error.

After make some change in subroutine OUTPUT as follows:

```
       IF (IOUT.LT.0.OR.MARKX.EQ.5) THEN
       KTL=0
       DO 90 I=1,NB
         NNREVA=NNREV(I)
         IF ((NNREVA/2)*2.NE.NNREVA) THEN
           IF (KTL.EQ.0) THEN
             WRITE (8,470)
             KTL=1
           ENDIF
           IF (Q(I).GT.0.0) THEN
            ================
             WRITE (8,480) NO(I),IABS(JS(I)),IABS(JF(I))
           ENDIF
           =====
         ENDIF
90      CONTINUE
```

```
      IF (KTL.EQ.0) WRITE (8,482)
    ENDIF
```

This problem has been solved. The airflow reversal in airway 2 has been shown in the following output results. The airflow in airways 1 and 4 has been changed to correct direction and no longer occur in the output results of "reversal of airflow has occurred in the following places".

```
            TIME AT  1200. SEC. AFTER EVENT




1**********************************************************************************
************************************************************************

 THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS PER MESH < 2.E-4 IN.W.G.)
 SATISFIED. CURRENT SUMFNV PER MESH 0.000120 IN.W.G.,   ITERATIONS    4


      TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN AIRWAYS

AIRWAY FROM TO DELTA Q AIRFLOW AVE. T T AT END FUMES CH4  HEADLOSS

   1    1   2  12056.     1796.  93.61  93.64  2.4869  0.00   1.227
   2   10   7  60250.    57776.  93.37  93.39  0.0000  0.00   0.000
   3  101  19 102600.   113398.  86.47  86.47  0.0000  0.00   0.000
   4    2   3  12056.     1796.  93.84  94.05  2.4869  0.00   0.000
   5    3 102 114656.   115195.  93.64  93.25  2.4869  0.00   1.228
   6    4   3 102600.   113398.  97.09  94.05  2.4869  0.00   1.092

 REVERSAL OF AIRFLOW HAS OCCURRED IN THE FOLLOWING PLACES

 AIRWAY    2 IS NOW CARRYING AIR FROM    10 TO    7
 AIRWAY    3 IS NOW CARRYING AIR FROM   101 TO   19
 AIRWAY    6 IS NOW CARRYING AIR FROM     4 TO    3

****************************************************************************
```

6. The Investigation, Analysis, and Modification for MFIRE V2.01/2.0

The modification for MFIRE v2.01/2.0 includes 2 parts: i). add the revised parts of v1.30, the newest version for dry case, into the version 2.01/2.0 to improve the iteration convergence and offer more correct results because the versions 2.01/2.0 are not based on version 1.30 but the version 1.29. ii). correct the errors which are occurred when the version 1.29 was upgraded to version 2.01/2.0. The newest version of MFIRE, for both dry and wet condition, is v2.10.

6.1 The error in the part of input data check

6.1.1 The elevation difference of two end junctions in an airway

In subroutine CCDATA, in order to check the elevation difference of two end junctions of an airway, it is compared with the input length of that airway. In the versions 2.01/2.0, the statements in the wet case part are as follows:

```
DZRD(J)=EF-ES
IF (ABS(DZRD(J)).GT.LA(J)) THEN
  LA(J)=DZRD(J)
  WRITE (8,320) NO(J),LA(J)
ENDIF
```

If the elevation difference of two end junctions of airway j, DZRD(J), is a negative value, airway j is a decline airway. The length of airway j LA(J), however, is given a negative value by the statement LA(J)=DZRD(J). The negative value of airway length may cause some computing mistakes in the successive computing process.

This problem can be shown from an example. When a data file (F94-10.bas a mine of Tg Soda Ash Inc. WY) is apllied, the length of airway 202 is 5 feet only, but the elevation difference of two end junctions 10 and 20 is 10 feet. The unreasonable input data should be corrected by assigning the elevation difference 10 feet to the length of airway 202. Above statement, unfortunately, makes a wrong correction by assigning a -10 feet to the length of airway 202. The negative value of the airway length will cause problem at the successive calculation.

The following modification has been made in version 2.10 and the length of an airway is given an absolute value.

```
DZRD(J)=EF-ES
IF (ABS(DZRD(J)).GT.LA(J)) THEN
  LA(J)=ABS(DZRD(J))
  ******************
  WRITE (8,320) NO(J),LA(J)
ENDIF
```

6.1.2 The cross sectional area of airway

In order to save computing time, in subroutine CDJN, the statement

```
IF(A(NW).GE.1000.0) GT TO 50
```

is used to make the evaluation of thermal state being not applied on the surface airways and junctions which areas are assigned some values larger than 1000 ft^2. It has no problem for most data files for coal mines or many other mines. In some mines, such as in the example file cote93.mfi, the cross sectional area of some airways may be larger than 1000 ft^2. The above statement will make those airways and relative junctions failing in the evaluation of thermal state variation and, of course, giving a wrong state distribution.

Consequently, the modified statement

```
IF (A(NW).GE.100000.0) GO TO 50
```

is employed to make the program being suitable for various mines. It is worth to notice that the cross sectional area of surface airways in the input data file should be equal to or large than 100000 ft^2 otherwise the evaluation of thermal state variation will be employed on the surface airways.

6.1.3 The reference temperature TR, the temperature T(K) and the relative humidity HJN(K)in junction K

In subroutine INPUT, the statement

```
IF (TR.LT.(-100.0).OR.TR.GT.150) TR=75.0
```

is used to correct the unreasonable reference temperature TR given in the input data file. The statement does not consider the case of the value of TR being not given in the input file. In this situation, the value of TR is equal to zero and TR will keep the zero value through all computing process because above statement fail to assign the value of 75.0 to TR. It will cause a large computing error. An assigning statement is added to solve above problem.

```
IF (TR.LT.(-100.0).OR.TR.GT.150) TR=75.0
IF (ABS(TR).LT.0.01) TR=75.0
```

In version 1.29, above problem does not exist. The statement for the correction of TR including the case of TR equal to zero in input data as follows:

```
IF (ABS(TR-60.).GT.50.) TR=75.0
```

Sometimes, the zero value in computer will become a very small value which is not equal to zero exactly so that the correct statement, ABS(TR).LT.0.01, is applied. For the same reason, the correction of temperature and relative humidity, T(K) and HJN(K), should be considered in the same way. The following two statements are added in corresponding parts.

```
IF (T(K).LT.(-100.0).OR.T(K).GT.1500.0) T(K)=75.0
IF (ABS(T(K)).LT.0.01) T(K)=75.0
*********************************
IF (HJN(K).LT.0.0.OR.HJN(K).GT.100.0) HJN(K)=80.
```

```
IF (ABS(HJN(K)).LT.0.01) HJN(K)=80.
*************************************
```

6.2 The error in the part of fan curve fitting

It is always to increase 10^{10} times of the value for the resistance of airway R for computing convenience because the value of airway resistance is very small. To show the influence of the unit change, the iteration equation of network calculation (1) can be changed as following equation:

$$\Delta Q_i=-\frac{\sum_{k=1}^{N} C_{ik}R_k*10^{10}Q_k^{(n)}\left|Q_k^{(n)}\right|*10^{-10}-F_i\left(Q_i^{(n)}\right)}{\left(2\sum_{k=1}^{N}(C_{ik})^2R_k*10^{10}\left|Q_k^{(n)}\right|*10^{-5}-\frac{dF_i(Q_i)}{dQ_i}*10^5\right)*10^{-5}} \quad i=1,\ldots,m \tag{5}$$

The terms R_k*10^{10} in the numerator and denominator are the input resistances of airways which have been increased 10^{10} times as stated above. The 10^{-10} in the term $Q_k^{(n)}|Q_k^{(n)}|*10^{-10}$ is used to keep the unit balancing, which is represented by the statement in subroutine ITR

```
DP=R(N)*Q(N)*ABS(Q(N))*1.E-10
```

Checking the term $R_k*10^{10}|Q_k^{(n)}|*10^{-5}$ in the denominator of equation (5), it is shown on the statement

```
RQ2=R(N)*ABS(Q(N))*2.E-5
```

the unit is not balancing because the value of that term is 10^5 times larger than it should be. The number 10^{-5} outside the bracket is required to keep that term correct. The fan curve's derivative term dPF_i/dQ_i should be multiplied a 10^5 to offset the influence of the number 10^{-5} outside the bracket. Consequently, the statement, for the term $(dFi(Qi)/dQi)*10^5$, should be modified as follows:

```
RQSUM=RQSUM-(RGRAD(L)*100000)
                      ******
```

The modification has made by Yang in the version 1.29, but in the version 2.01/2.0, the number of 100000 is missing. Yang analyzed the consequences if the term dPF/DQ would be 10^5 times smaller than it should be[9].

6.3 The error in the predetermined iteration times in airflow distribution balancing.

In the network balancing involving the temperature or dynamic state calculation, the iteration of network calculation consists of inner and outer parts. The inner iteration, in subroutine ITR, is for the airflow distribution balancing. Its convergence criterion is DQ, the sum of average correction of airflow rates in airways of per mesh, and is, say, 10 CFM. The outer iteration, in the main part of the program, is for the natural ventilation calculation. The prefixed accuracy criterion is SUMFNV/MNO, the sum of average natural ventilation pressure correction per mesh in successive intervals, and is 0.0002 IN.WG in the version 2.01/2.0.

In the subroutine ITR of the former version v1.29 of MFIRE, The maximum inner iteration number is assigned a certain number such as 100, 200, or 500 as following statement

```
ELSE IF (ITCT.GT.500) THEN
  NSFLOW=0
  WRITE (8,200) DQSUM/FLOAT(MNO)
  WRITE (8,205) MNO
  RETURN
```

Mark ITCT is the counter of inner iteration in network balancing. The variables ITN and MADJ are the maximum number of outer iteration in dynamic and temperature simulation respectively. They are prefixed in input data file by user. Marks ITT and MADJC are the counter of outer iteration in dynamic and temperature simulation respectively. When the inner iteration achieves airflow distribution convergence or the iteration time is larger than the prefixed maximum inner iteration number, the outer iteration for natural ventilation calculation starts one more time.

In subroutine ITR of the versions 2.01/2.0, the prefixed maximum inner iteration number, which is often given a value between 100 to 500, is wrongly assigned the prefixed variable ITN of input data file by following statements, which is the maximum outer iteration number and always given a value from 10 to 15 in the input data set.

```
ELSE IF (ITCT.GT.ITN) THEN
  MARKN=1
  RETURN
```

This wrong modification is not a misprint. In the versions 2.01/2.0, the term explanation in the beginning of the main program has shown:

ITN: MAX. NO. OF ITERATION IN NETWORK BALANCE.

In the temperature part, the explanation is also shown:

ITN: USER-SPECIFIED MAX. NUMBER 0F ITERATION IN NETWORK BALANCINiG (SUB. ITR) AND DATA PREPARATION (TEMP. PART).

That is why the authors of the report checking it carefully to avoid the misunderstanding of the change made by Chang in the version 2.01/2.0. In the input data file template of the versions 2.01/2.0, inputw.fmt, the definition of ITN is still as in the version 1.29:

ITN: MAX. NO. OF ITERATION IN DYNAMIC PART, DEFAULT 15.

In the subroutine ITR, the default number of ITN is assigned same as v1.29

```
IF (ITN.LT.10.OR.ITN.GT.50) ITN=15
```

In the main program, main.f, the application of ITN in dynamic state part is same as v1.29

```
      IF (ITT.GE.(ITN-9)) THEN
        J=ITT-ITN+10
        DO 260 I=1,NB
          QRCD(I,J)=ABS(Q(I))
          IF (JSB(I).NE.IABS(JS(I)).AND.Q(I).GT.0.0)
     .      QRCD(I,J)=-ABS(Q(I))
          IF (JSB(I).EQ.IABS(JS(I)).AND.Q(I).LT.0.0)
     .      QRCD(I,J)=-ABS(Q(I))
          TMRCD(I,J)=TMRD(I)
260     CONTINUE
      ENDIF
  IF (ITT.GE.ITN.OR.(SUMFNV/MNO.LE.2.E-4.AND.ITT.GE.2)) THEN
```

Consequently, ITN serves the both iteration criterion for the inner iteration in network balancing and outer iteration in dynamic part in versions 2.01/2.0 but serve iteration criterion for the outer iteration only in dynamic part in version 1.29.

The inner iteration needed to achieve the airflow distribution calculation convergence varies to a wide range, say less than 10 to over several hundred times. It depends on the dimension and structure of a ventilation network and the problem difficulty. The confusion between the maximum inner and outer iteration times will cause a big problem. The inner iteration for airflow distribution calculation will stop after the counter ITCT is larger than ITN even if the iteration time is only 15 and convergence has not achieved yet. The iteration of airflow distribution in many mine ventilation networks can not convergence at less than 15 times. The temperature and contaminant distribution calculation, and the state variation evaluation are based on the airflow balancing calculation. Consequently, the divergence of airflow distribution iteration will cause the wrong airflow distribution and give wrong state simulation results usually. In the new version 2.10, this part has changed back as that of the version 1.29.

data files	v1.30	v2.01/2.0	v2.10	
			NHM=0 dry	NHM=1 WET
testw2.dat	*yes	yes	yes	yes
multlev.dat	yes	yes**	yes	yes
cote931.dat	yes	no	yes	yes
f94-101.bas	yes	no	yes	yes
brushy1.mfi	yes	no	yes	yes
vibbas1.mfi	yes	no	yes	yes
vibbas2.mfi	yes	no	yes	yes

Table 10 Convergence Comparison for different versions of MFIRE

*: the YES means that the algorithm iteration is converging and the computing results are correct.

**: the computing gets convergence and results are correct at the network and quasi-equilibrium state calculation part but did not get convergence at the dynamic part.

If user gives ITN a value as 200 in order to achieve convergence in network balancing calculation as in the data set f94-101.bas, it will be assanged a default value 15 by the following statement in the subroutine INPUT

IF (ITN.LT.10.OR.ITN.GT.50) ITN=15

and may cause the divergence. If above statement is deleted, the maximum iteration time 200 will make the outer iteration run 200 time then return. It will cause a large amount computing time being wasted. Consequently, it is wrong to give the mark ITN two functions at the same time in version 2.01/2.0.

The case study in table 10 shows that the version 2.01/v2.0 of MFIRE cannot achieve iteration convergence in the network balancing and give wrong computing results for the most of the given examples. Only in some simple ventilation networks, such as in data sets testw2.dat and mult-lev.dat, a fewer iterations can achieve the convergence for airflow distribution calculation in steady state condition. The computing results for the airflow distribution calculation is correct.

6.4 The error caused by the handling of the data in the user specified time table

The user specified time table, used for more realistic state simulation and state control, is applied for the simulation of the ventilation system variation. In the former version v1.29, if user needs to stop a fan for the state control after the mine fire breaks out, the parameters of that fan stored in relative arrays, such as NOF, MPTS, NSKP, NWTYP, and so on, should be replaced by that of the fan with higher successive sequential number in the array NOF. The parameter transfer process will continue by the fans with higher sequential numbers in the array NOF. For example, in the data file vibbase2.mfi, in ten minutes after mine fire occurs, fan 1 in airway 1 will stop and a same type of fan will be installed at airway 10, which is parallel with airway 1. The subroutine CDCH inputs the variation of ventilation system condition from the time table of input data file. The parameter transfer starts to make NOF(1)=NOF(2), NOF(2)=NOF(3), ..., NOF(7)=NOF(8) and NOF(8)=0 if no fan is added. The added fan ID number 10 is assigned to NOF(8) to represent a fan being added to airway 10. The other arrays of the parameters of fans will be given similar change.

In the new version of MFIRE v1.30, for dry case, some parameters which did not make above transfer, say RGRAD, NFREG, QF, PF, are added as stated in section 4.3. In subroutine CDCH of the version of 2.01/2.0, the only parameter change is to send the type of airway NWTYP(I), in which airway the fan stop, equal to zero. It is no necessary to make above parameter transfer for the parameters of successive fans as version 1.29 does. This part in version 2.01/2.0 is much simpler than the former versions of MFIRE. The unchanged parameters of the stopped fan, unfortunately, may cause problem. It can be explained by an example.

In the time table of data file vibbase2.mfi, there are eight fans operated in a mine. When fan 1 is stopped in 10 minutes after fire occurs, the array, type of airway, NWTYP(1) is assigned to zero. The total number of fans, NFNUM, is still equal to eight. The other corresponding arrays keep unchanged too. It will cause problem in airway arrangement in the network balancing part. In subroutine ARR, the airways are arranged to form array INU as following statements.

```
     IF (NFNUM.GT.0) THEN
       DO 25 K=1,NB
         DO 20 J=1,NFNUM
           IF (NOF(J).EQ.NO(K)) THEN
             RGRAD(J)=0.0
             NFCW(J)=0
             NFREG(J)=K
             INU(NBU)=K
             NBU=NBU-1
             GO TO 25
           ENDIF
20       CONTINUE
25     CONTINUE
     ENDIF
```

The unchanged NOF(1) will make the airway 1 being still a fan branch and give a wrong variable NBU, the sequential number for fan branch, and array INU, arranging airways following airway types or the products RQ of resistances R and airflow rates Q of the airways. It will cause some problems in the computing process and data output results. Consequently, the unchanged array of fan ID number, NOF, should be controlled by adding the changed array, NWTYP, as the underlined part of following statement:

```
    IF (NOF(J).EQ.NO(K).AND.NWTYP(K).EQ.1)THEN
                        *********************
```

6.5 The error in the number of segments of fan curve MFA in subroutine SPLINE

In the subroutine SPLINE of former version MFIRE, the variable MFA, the number of segments in fan curve, is assigned a value by the statement:

```
          MFA=MF-1
```

where variable MF is the number of data points of fan characteristic. It is changed in version 1.29. The variable MFA is never assigned any value. The statement

```
       M = MF-1
45     DQKI=QKI-QF(J,M)
       RKI = FKQ(J,M,1)+FKQ(J,M,2)*DQKI+FKQ(J,M,3)*DQKI**2
     .        +FKQ(J,M,4)*DQKI**3
```

will give a zero value to variable M which is the M-th segment of fan curve and may cause problem.

7. The Summary of Modification in MFIRE v1.30 and v2.10

7.1 The modification in MFIRE v1.30

7.1.1 More flexible selection for the fan curve fitting and the handling of the boundary region of the fan curve (sections 3.3 to 3.5, & 4.2)

For the selection of fan curve fitting, the cubic spline and least squares methods are available. For the handling of the boundary region of the fan curve, there are three choices: a. the slopes of the both right and left boundary region of fan curve are same as that of the most right and left data points of input fan characteristic curve. b. the slope of the right boundary region of fan curve is sent to zero. c. the slopes of the right and left boundary region of fan curve are sent to zero. The flexible choices for fan curve fitting method and the handling of boundary region of fan curve improve the common suitability and calculation convergence of program MFIRE for various input data sets.

7.1.2 The wrong airflow, fume distribution after airflow reversal happens (section 5.1)

If airflow reverses in an airway, it is necessary to change the airflow direction and make the mutual exchange of the starting and ending junctions in the airway. In quasi-equilibrium state simulation part, if one cycle of data preparation, the outer iteration for natural ventilation calculation, can achieve convergence, the former version of MFIRE v1.29 cannot change the corresponding airway parameters in time when airflow reversal happens. It gives a wrong airflow, temperature, and fume concentration distribution. The modification in new version v1.30 make the single iteration doubled and achieve the airway parameters, such as the starting and ending junctions, change in time.

7.1.3 Wrong fan characteristic assignment (section 4.3)

If stopping a fan for state control in a certain time after mine fire breaks out, the arrays for storing relative parameters of that stopped fan, such as NOF, MPTS, and NSKP, should be replaced by that of the fan with higher successive sequential number in array NOF. The parameter transfer will continue one by one through the fans with higher sequential numbers. The former version 1.29 misses the replacement for some parameters of the stopped fan, such as the input data point of fan characteristic PF, QF, and the coefficient of fan equation array FKQ. The changed arrays NOF, MPTS, and NSKP conflict with unchanged arrays PF, QF, and FKQ. The fan NOF(n), pick up the input data points of fan characteristic of fan NOF(n-1). The parameter transfer of arrays PF, QF, and FKQ is added in the subroutine CDCH of new version 1.30.

7.1.4 The iteration divergence of dynamic state simulation part. (section 4.1)

If the ventilation system condition is changed, the state distribution will change also. The mesh pattern formed from former airflow distribution is not suitable to the new state distribution, It causes the divergence of Hardy Cross method in the former version 1.29. The modification is applied on the new version of MFIRE 1.30. If the ventilation system condition changes, the new mesh pattern will be reformed for iteration calculation in subroutine ITR.

7.1.5 Wrong fume concentration for the end junction of an airway with fire source (section 5.2.1)

The fume concentration of the ending junction of an airway with fire source should not be zero after mine fire breaks out. The former version of MFIRE did not make fume concentration calculation in time so that the fume concentration is delayed on one time interval. In subroutine DTTR of new version 1.30, the modification corrects the fume concentration calculation.

7.1.6 The determination of the average airflow rate in the time interval of airflow reversal (section 5.2.2)

The formula in subroutine DIP of the former version v1.29 offers the average airflow rate in the time interval of airflow reversal by the algebraic time interval average value of beginning and ending airflow rates of the time interval. The former formula offers a lower average airflow rate and causes larger error on the state simulation because of the offsetting influence of positive and negative value when airflow reversal happens in the time interval. The new formula in version 1.30 offers the absolute average value of the beginning and ending airflow rates in the time interval.

7.1.7. In the subroutine SPLINE of version 1.29, the variable MFA, the number of the fan curve segments, is directly replaced by MF-1 so that it is not assigned any value. The variable MFA, however, still exist in statement

M=MFA

to assign zero to the variable M which is the m-th segment of fan curve and cause problem. It will cause the error of RKI and RGRAD in a certain cases.

7.1.8 The output error for the airways in which airflow reverses (section 5.2.3)

In the dynamic state simulation part, when airflow reverses, the starting and ending junctions should be mutually exchanged and airflow direction should be corrected in time. The output of some data set shows that the airflow direction has been corrected but the airflow rates in some airways still keep negative value. The modified version makes the correction of airflow direction and airflow rate in time.

7.2 The modification in MFIRE v 2.10

7.2.1 Add all the modified parts in version 1.30 to version 2.10 except section 7.1.3 because the error in handling of the input data in the user specified time table has not existed in version 2.01/2.0.

7.2.2 The error in handling the input data of the used specified time table (section 6.4)

In the version 2.10/2.0, the error shown in section 7.1.3 is eliminated because it is not necessary to make the parameter transfer for every related arrays. It induces, however, a new error. When a fan stops, in the subroutine CDCH of version v2.01/2.0, only the type of airway NWTYP, in which the stopped fan is installed, is send to zero. The other parameters of that fan in arrays NOF, RGRAD, NFREG, QF, PF, and variable NFNUM, the number of operating fans, are unchanged. The simplification causes a new conflict between the unchanged variable NFNUM, array NOF and the changed array NWTYP. The conflict will cause problem. The airways arranged by following their types form array INU in subroutine ARR. The wrong type of airway will give a wrong array INU and variable NBU and then cause the problems at later calculation and final result output. The modified part offers the combination of unchanged array NOF and changed array NWTYP to avoid above mistake.

7.2.3 The error in input data check (section 6.1)

a. In input data set, if an airway is a decline airway and the input length of the airway is smaller than the elevation difference of two end junctions of the airway, the correction of input data in subroutine CCDATA will wrongly assign the negative elevation difference to the length of that airway and cause a computing problem. In the new version 2.10, the absolute value of elevation difference of two end junctions of the airway is applied for the correction of airway length.

b. In the subroutine CDJN of former version of MFIRE, Airways with cross sectional areas larger than 1000 ft^2 is considered to be surface airways so that the evaluations of thermal state variation of those airways are ignored, In some mines, however, it is possible the cross sectional areas of some underground airways are larger than 1000 ft^2, such as in the example data set cote93.mfi. These airways are wrongly treated as surface airways and ignored in the calculation of thermal state variation. In the new version of v2.10, the criterion of cross selection area of a surface airway is increased to 10^5 ft.

c. If the reference temperature, TR, is not given a value in input data file, the input data correction in subroutine INPUT of former version v2.01/2.0 will fail to assign a default value to the TR. The reference temperature TR with zero value will cause large error. In the new version, the correct statement for the TR with zero value is added. For the same reason, the input data correction of junction temperature T(K), and relative humidity HJN(K) are added the zero value correction statements.

7.2.4 The error in fan curve fitting (section 6.2)

In the former version 1.29, the number 10^5 is in the statement of fan curve derivative of subroutine ITR:

```
RQSUN=RQSUM-(RGRAD(L)*100000)
```

In version 2.01/2.0, the number 10^5 is missed in above statement so that it is added in new version v2.10.

7.2.5 The maximum outer iteration number ITN (section 6.3)

In version 2.01/2.0, the variable ITN, maximum outer iteration number for natural ventilation calculation, which is usually used in the main program, is wrongly applied as the maximum inner iteration number for airflow distribution calculation, which is usually used in subroutine ITR. The small error causes a big problem. The iteration for airflow distribution calculation, the inner iteration, is forced stopped when the iteration time counter ITCT is larger than ITN, the maximum outer iteration time and normally given value about 10 to 15 in input data set. The ITN is often too small to achieve convergence of airflow distribution calculation, inner iteration, in the most data sets. It causes the iteration divergence and the unbalancing airflow distribution. Finally, it causes total wrong state simulation results. The maximum inner iteration number is replaced by 500 in subroutine ITR in the new version 2.10. The problem does not exist in former version 1.29 because the number 500, not ITN, is used as the maximum inner iteration number.

8. The Answers for Some User's Questions

8.1 Why is the heat added to airflow from the fire source different in the different time intervals while the heat released from the fire source in input data set is constant?

Some users questioned the computing correction of the amount heat added on the airflow from a fire source. The computing investigation of applying data set cote93.mfi shows that the heat added to airflow from the fire source at each time interval as follows when the heat released from the fire source in input data is keeping 5000 btu/min:

time interval Sec.	300	600	900	1200	1500	1800	2100	2400	....	3600
heat added (BTU/Min.)	5000	5000	5000	8850	8840	5120	5110	5110	....	5110

In the subroutine CDJN

```
VART=(9900.0+TK)**2+2.0*HEATAD/(QQ2(NW)*2.4E-5*DR
 *****                 ********
IF (VART.LT.9.3E7) THEN
   WRITE (8,100) NO(NW)
   NSTOP
   RETURN
ENDIF
TJS(NW)=-9900.0+SQRT(VART)
***************************
IF (CONCT.EQ.1) TJS(NW)=TJSS(NW)
IF (NWTYP(NW).EQ.10) THEN
   TRD(NW)=TJS(NW)
   IF (TRD(NW).LT.(-200.0).OR.TRD(NW).GT.5000.0) THEN
      WRITE (8,100) NO(NW)
      NSTOP=6
      RETURN
   ENDIF
   RDPROP(NW)=(PROP(K)*QQ2(NW)+CONTAM)/(QQ2(NW)+CONTQ)
   RDCH4(NW)=PRCH4(K)
   HTAD(L)=(TJS(NW)-T(K))*0.24*QQ2(NW)*DR
   *****************************************
   GO TO 50
ENDIF
```

Array HTAD is used to represent the heat added on airflow from the fire source. It depends on the input heat released from fire source HEADAD which is calculated from the array HEAT in the fixed quantity fire case or other parameters calculated in the oxygen or fuel rich fire cases. The value of HTAD depends on the former air temperature at the last time interval, and the airflow rate which gives the values of the arrays TJS, T, and QQ2 respectively. Consequently, the HTAD, that is printed out as heat input, will be changed with the variation of TJS or T even if the array HEAT or HEATAD keeps constant.

8.2 the state simulation of airflow cooling plant

Program MFIRE has the function to handle state simulation of air conditioning. The users may find that the program operation may stop and an error message about the computing of junction temperature, which is in impossible negative value (> -250 °F), is printing out. It may be caused by two possibilities:i. user selecting an unrealistic value of air cooling capacity for sample testing, ii. The airflow reversal causes the junction temperature near cooling station changing dramatically. Two formulas for the junction temperature calculation are presented for the analysis of the state simulation of airflow cooling plant.

The formula suggested by Greuer [4] for the temperature calculation in the junction near fire source or air cooling station is as follows:

$$t = -4950+\frac{t_1}{2}+\sqrt{(4950-\frac{t_1}{2})^2+9900*t_1+\frac{HEATAD}{Q*2.4E-5*DR}} \tag{6}$$

where: t is the temperature of a junction after heat addition °F

t_1 is the temperature of a junction before heat addition °F,

DR: reference density ib/ft^3,

Q is the airflow rate passing through the fire source CFM,,

HEATAD: The heat released from the fire source btu/min.

The formula used in program MFIRE (v.129) is as follows:

$$t = -9900 + \sqrt{(9900+t_1)^2+\frac{2*HEATAD}{Q*2.4E-5*DR}} \tag{7}$$

The calculation results of above formulas are similar. In the state simulation for air cooling plant, the array HEATAD or HEAT is given negative value. If the absolute value of HEATAD or HEAT is very large (> -650000 btu/min.), the junction temperature t (in equation 6 or 7) may be larger than a predetermining criterion (see -250 °F). an error message will be printed out and program running will stop. It does not mean that there are some programming errors but the unsuitable data input. That is why the printout reminds the user changing the data of the capacity of air cooling plant. On the other hand, if user offers the value of the heat released from the fire source is small (see 5000 btu/min. as in data file cote93.mfi), the state change, of course, is very small.

When airflow passes through the cooling station, the air temperature of the junction in the immediate downstream of the station will be reduced. If the airflow reversal happens, the airflow with the reduced temperature will pass through the air cooling station again and the airflow temperature will be reduced one time again. If the initial data of the air cooling station is large enough, the two-time cooling action will reduce the air temperature in a junction near the cooling station to a value lower than the temperature criterion (see -250 °F) and stop the program running. The situation of temperature double reduction mirrors the temperature variation in the practical case.

Consequently, there is no programming mistake in this situation even if the error message is printed out. The only thing needed to do for the user is to change the capacity of cooling station at the output's reminder. It does often happen during the sample test of the program by using the data file which is not from a ventilation survey but from the user's assumption.

8.3 prepare the data file carefully

The data file should be changed carefully when the variation of the ventilation system condition is wanted. If user changes the data file improperly, the computing divergence or computer problems may be induced, which is not caused by the programming error or improperly selecting the methods of fan curve fitting and handling the boundary region of fan curve. For example, when adding a fan into a ventilation system (not by employing time table of dynamic state part), one needs to make corresponding change in the data file: i. add 1 to the number of operating fans in the control card I; ii. change the type of airway in which the fan is installed from 0 to 1 in airway card; iii. add the data points of that fan characteristic to fan cards I and II. The miss of any one of above changes will cause the computing problem.

When switching the calculation from dry case to wet case, the input data should be made corresponding change in following parts of an input data file: (3). airway data group 16 arrays, (4). junction data group, 5 arrays, (8). contamination data lines, 11 arrays or variables, (9). control data II, 13 variables, (11) time table for condition change, 4,). add a fire source 11 variables. If the input data of above groups are less than they should be, the program will pick up wrong data from next line and computing problems will be induced. Consequently, it is necessary to prepare enough data, at least leave the blank at the corresponding position. If preparing input data for dry case based on the wet condition data, user just need to send the last number of first line, NHM, to zero. The section 5.3 in the MFIRE USERS MANUAL VERSION 2.0 [7] presents the introduction of input file change.

8.4 The difference of the unsuitable selection of fan curve fitting method and the programming error in fan curve fitting.

In the former version of program MFIRE, the cubic spline interpolation is the only method for fan curve fitting. Some users put the blame of iteration divergence on the application of cubic spline method. In fact, most divergence problems are caused by programming errors in the part of fan curve fitting or the handling of the boundary region of fan curve. The investigation at table 1 shows that the cubic spline interpolation method works well if the ventilation system condition has no large change. It means that the computing divergence is not caused by the fan curve fitting method itself mostly but the system condition change which is not handled correctly in programming. The proper selection of fan fitting method, of course, can improve the fan curve fitting accuracy sometimes, but it is not involved to the convergence behavior in most cases after the modified cubic spline method [9] is employed.

Program MFIRE has been developed and modified to enhance its functions for years. More and more marks and switches are added to guide the program going to the desired direction. These marks and switches, however, should be made corresponding correction to meet the need of a new version of the program. Any omission in the correction of those flags or switches will cause new programming error. It is a basic principle for the program modifiers to pay more attention to those flags, switches, and the modified parts not causing new computing problems.

9. Conclusion

As a state simulator, the program MFIRE has been applied for steady state or dynamic state simulation in industry. State control, however, is another important function of program MFIRE too. It can simulate the effects of the state control measures on a dynamic ventilation system. Consequently, it is helpful on the development of the emergency planning for mine fire fighting.

The wide range variation of ventilation system condition (such as fans add, stop or reverse, regulator added, and airflow short cut), is always needed in making state control measures to meet different situations of mine fire and different ventilation system. It requires more reliability on programming and algorithm convergence. The investigation of the application of the data files, in which a wide range changes are happening, reveals the problems in the former version of program MFIRE versions 1.29 and v2.01/2.0.

In the modified versions of MFIRE, v1.30 for dry case and v2.10 for both dry and wet cases, the mistakes on computer programming have been corrected and more reliable calculation results will be given even if the various data files are employed for testing. The modification of the new version of program MFIRE is as follows:

i. Programming error correction:

Improve the algorithm convergence and offer correct calculating results on the distribution of airflow rates, temperature and contaminant in quasi-equilibrium state and dynamic state simulation parts.

ii. suitable method selection for different data files

offer flexible selection methods on the fan curve fitting and the treatments of the boundary region of fan characteristic curve to meet the various data files and to improve the computing accuracy and iteration convergence.

Reference

1. Chang, X, Laage,W.L., & Greuer, R.E. , "A User's Manual for MFIRE: A Computer Simulation Program for Mine Ventilation and Fire Modeling", BuMines IC9245, 1990

2. Chang, X, "Investigation on Ventilation Network Calculation Techniques", M.S. Thesis, Michigan Technological University, 1983.

3. Chang X. "The transient State Simulation of Mine Ventilation System", Ph.D. Thesis, Michigan Technological University, 1987.

4. Greuer, R.E., "Study of Mine Fires and Mine Ventilation; Part I, Computer Simulation of Ventilation System under the influence of Mine Fires", (U.S. Bureau of Mines Contract No. SO241032 Michigan Tech. Univ.), 1977.

5. Greuer, R.E., "Influence of Mine Fires on the Ventilation of Underground Mines", (US Bureau of Mines Contract No. PO122095), 1973.

6. James, M.L., Smith, G.M., & Wolford, "Applied Numerical Methods for Digital Computation", Third Edition, Harper & Row, Publishers, New York, pp. 299-383, 1985.

7. Laage, L.W., Greuer R.E., & Pomroy, W.H., "MFIRE Users Manual Version 2.0", US, Bureau of Mines, 1994.

8. Robert W.H., "Numerical Methods", Quantum Publishers, Inc., Net York, pp47-50, pp121-144, 1975.

9. Yang, H. & Greuer, R.E. "Investigation of the convergence Behavior in Computer Modeling of Mine Ventilation and Fires ", (US Bureau of Mines Contract No. PO210154 Michigan Tech. Univ.), 1992

10. Zhou, X. "Optimal Control on Underground Mine Fire', Ph.D. Thesis, Michigan Tech. University, pp. 1-10, 1988.

APPENDIX 1 Input Data File Template

```
$-----------------------------------------------------------------------------------------------------$
$                                                                                                     $
$    *** INPUT FORMAT INSTRUCTION FILE FOR PROGRAM WFIRE ***                                          $
$                                                                                                     $
$  GENERAL RULES FOR DATA INPUT:                                                                      $
$                                                                                                     $
$  * NO NEED TO DISTINGUISH INTERGER AND REAL NUMBERS. SYMBOLS USED                                   $
$  IN NUMERICAL INPUT COMPRISE THE FOLLOWING LIST:                                                    $
$         1 2 3 4 5 6 7 8 9 0 . , - + BLANK                                                           $
$  * FREE FORMAT ALLOWED. COMMA AND BLANK SPACE CAN BOTH BE DELIMINA-                                  $
$  TORS. SEQUENTIALLY DISTRIBUTED BLANK SPACES ACT AS ONE DELIMINATOR.                                $
$  BLANKS ADJACENT TO A COMMA HAS NO EFFECT IN SEPARATING INPUT TERMS                                .$
$  * ZERO INPUT CAN BE OMITTED BUT ITS DELIMINATOR CAN NOT.                                           $
$  * ALL LINES CONTAINED IN THIS INPUT FORMAT FILE MUST BE ERASED                                     $
$  BEFORE USING IT AS AN INPUT DATA FILE FOR WFIRE.                                                   $
$-----------------------------------------------------------------------------------------------------$

$-----------------------------------------------------------------------------------------------------$
$ (2) CONTROL DATA I:                                                                                 $
$   NB,NFNUM,INFLOW,NVPN,NETW,TR,MADJ,ITN,TINC,SPAN,IOUT,DR,NHM                                       $
$                                                                                                     $
$   NB: NUMBER OF AIRWAYS IN NETWORK, (NB+INFLOW) < NMX.                                              $
$   NFNUM: NUMBER OF FAN CHARACTERISTICS, MAX. 10.                                                    $
$   INFLOW: NUMBER OF CONTAMINATION SOURCES, MAX. 10.                                                 $
$   NVPN: MARK FOR PRESENCE OF JUNCTION DATA, 0: PRESENT.                                             $
$   NETW: MARK, 1: ONLY NETWORK PART WILL BE PERFORMED.                                               $
$   TR: REFERENCE TEMPERATURE OF AIR, DEG. F. DEFAULT VALUE 75.                                       $
$   MADJ: MAX. NO. OF ITERATION IN TEMP. PART. DEFAULT 15.                                            $
$   ITN: MAX. NO. OF ITERATION IN DYNAMIC PART, DEFAULT 15.                                           $
$   TINC: TIME INCREMENT IN DYNAMIC SIMULATION PART, SEC, DEFAULT 15                                  $
$   SPAN: TIME SPAN OF DYNAMIC SIMULATION, MIN, DEFAULT 5.                                            $
$   IOUT: OUTPUT, 0: BRIEF; 1: NORMAL; -1: DETAIL; -2; MORE DETAILED.                                 $
$   DR: REFERENCE DENSITY, LBM/FT3.                                                                   $
$   NHM: MARK, 0: DRY CONDITION, 1: WATER PHASE CHANGE CONSIDERED.                                    $
$                                                                                                     $
$        *** INPUT DATA FOR CONTROL CARD I HERE ***                                                   $
$-- NB,NFNUM,INFLOW,NVPN,NETW,TR,MADJ,ITN,NTEMP,TINC,SPAN,IOUT,DR,NHM                                -$

$-----------------------------------------------------------------------------------------------------$
$  (3) AIRWAY DATA GROUP                                                                              $
$   NO,JS,JF,NWTYP,R,Q,KF,LA,A,O,HA,HK,CH4V,CH4PA,TROCK(,PSI)                                         $
$                                                                                                     $
$  * NB LINES OF AIRWAY DATA MUST BE PRESENT. EACH DATA LINE MUST                                     $
$  CONTAIN ALL THE ABOVE VARIABLE INPUT VALUES. IF NHM IS 0, PSI                                      $
$  SHOULD BE OMITTED.                                                                                 $
$  * KF,LA,A,O,HA,HK HAVE THEIR AVERAGE OR DEFAULT VALUES WHICH ARE                                   $
$  ACTIVATED BY INPUTING ZERO FOR THE RESPECTIVE TERM.                                                $
$  * TROCK CAN BE INPUT AS ZERO TO EXCEPT THE CALCULATED VALUE.                                       $
$                                                                                                     $
$   NO:   AIRWAY ID NUMBER (1 TO 9950).                                                               $
```

```
$    JS:   JUNCTION ID NUMBER OF AIRWAY BEGINNING (1 TO 950).                  $
$    JF:   JUNCTION ID NUMBER OF AIRWAY END (1 TO 950).                        $
$    NWTYP: AIRWAY TYPE, 1: FAN; 0: ORDINARY; -1: FIXED-Q.                     $
$    R:    RESISTANCE OF AIRWAY 1.E-10 IN. W.G./(CFM)2. (NWTYP=0)              $
$          OR FAN PRESSURE IN IN.W.G. (NWTYP=1)                                $
$    Q:    AIRFLOW RATE IN CFM.                                                $
$    KF:   FRICTION FACTOR IN 1.E-10 LBM*MIN2/FT4.                             $
$    LA:   AIRWAY LENGTH IN FT.                                                $
$    A:    AIRWAY CROSS SECTIONAL AREA IN FT**2.                               $
$    O:    AIRWAY PERIMETER IN FT.                                             $
$    HA:   THERMAL DIFFUSIVITY OF ROCK, FT2/HR.                                $
$    HK:   THERMAL CONDUCTIVITY OF ROCK, BTU/HR*FT*F.                          $
$    CH4V: METHANE EMISSION RATE IN AIRWAYS FT3/MIN.                           $
$    CH4PA: METHANE EMISION RATE PER UNIT SURFACE AREA, FT/MIN.                $
$    TROCK: AVERAGE ROCK TEMPERATURE IN AIRWAY, DEGREE F.                      $
$    PSI:  AIRWAY WALL WETNESS, 0 TO 1.                                        $
$                                                                              $
$         ***  INPUT DATA FOR AIRWAY CARDS HERE  ***                           $
$----- NO,JS,JF,NWTYP,R,Q,KF,LA,A,O,HA,HK,CH4V,CH4PA,TROCK,PSI ----------------$

$------------------------------------------------------------------------------$
$  (4) JUNCTION DATA GROUP:                                                    $
$    JNO,T,Z,CH4C ,HJN                                                         $
$                                                                              $
$  * IF NVPN IS 0 OR LESS, EACH JUNCTION SPECIFIED IN JS AND JF  MUST          $
$  HAVE  A ONE-LINE INPUT HERE. OTHERWISE JUNCTION DATA GROUP MUST BE          $
$  OMITTED.                                                                    $
$                                                                              $
$    JNO:  JUNCTION ID NUMBER (1 TO 950), 900 TO 950 ARE SURFACE               $
$          JUNCTIONS.                                                          $
$    T:    TEMPERATURE OF AIR IN JUNCTIONS, DEGREE F.                          $
$    Z:    ELEVATION OF JUNCTIONS, FT.                                         $
$    CH4C: METHANE CONCENTRATION IN JUNCTIONS, PERCENTAGE.                     $
$    HJN: RELATIVE HUMIDITY IN JUNCTIONS, PRECENTAGE                           $
$                                                                              $
$         ***   INPUT JUNCTION CARDS HERE   ***                                $
$------------------------ JNO,T,Z,CH4C,HJN-------------------------------------$

$------------------------------------------------------------------------------$
$   FAN LINES I , II and III COMPRISE A DATA GROUP WHICH MUST REPEAT NFNUM     $
$  TIMES FOR NFNUM FAN CURVES. IF NFNUM IS 0 OR LESS, FAN DATA GROUP           $
$  MUST BE OMITTED.                                                            $
$                                                                              $
$  (5) FAN DATA LINE I:                                                        $
$    NOF,MPTS                                                                  $
$    NOF:  ID NUMBER OF FAN BRANCH.                                            $
$    MPTS: NUMBER OF DATA PAIRS DEFINING FAN CHAR, MAXIMUM 10.                 $
$    NSWT: MARK, TO SELECT FAN CURVE FITTING METHOD                            $
$            1 LEAST SQUARE 2.SPLINE 3.AUTOMATIC 4.AUTO+MANUAL                 $
$  (6) FAN DATA LINE II ( IF NECESSARY, TAKE MORE THAN 1 LINE )                $
$    QF,PF,QF,PF,QF,PF,QF,PF,QF,PF                                             $
$    QF:   AIRFLOW AT GIVEN POINT OF FAN CHAR, CFM.                            $
$    PF:   PRESSURE AT GIVEN POINT OF FAN CHAR, IN.W.G.                        $
$  (7) FAN DATA LINE III                                                       $
$    NPLOT: MARK, TREATMENT OF THE BOUNDARY REGION OF FAN CURVE                $
$         1: EXTEND FAN CURVE BY FOLLOWING GRADIENTS OF TWO ENDS.              $
```

```
$        2: EXTEND FAN CURVE AS ABOVE AT LEFT BROUNDARY REGION AND        $
$          SEND GRADIENT OF RIGHT BOUNDARY REGION TO ZERO.                 $
$        3. SEND GRADIENT OF BOTH SIDES OF BOUNDARY REGION TO ZERO.       $
$                                                                          $
$         ***   INPUT FAN CARDS I AND II HERE   ***                       $
$------------------------ NOF,MPTS,NSWT------------------------------------$
$---------------- QF,PF,QF,PF,QF,PF,QF,PF,QF,PF ---------------------------$
$---------------- NPLOT----------------------------------------------------$

$-------------------------------------------------------------------------$
$ (8) CONTAMINATION (HEAT SOURCE) DATA LINES.                             $
$   NCENT,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,QCENT,TPR,(CN,HN)              $
$ * EACH HEAT SOURCE MUST HAVE  A ONE-LINE INPUT HERE. THE NUMBER OF      $
$ TOTAL HEAT SOURCES WAS SPECIFIED BY "INFLOW" IN CONTROL DATA I.         $
$ * IF NHM=0, CN AND HN INPUT SHOULD BE OMITTED.                          $
$                                                                         $
$   NCENT: ID NUMBER OF AIRWAYS WHERE FIRE SOURCES LOCATE.                $
$   CONT:  VOLUME FLOW RATE OF CONTAMINATED GAS INFLOW IN CFM.            $
$   CONC:  CONCENTRATION OF CONTAMINANT IN GAS INFLOW, %.                 $
$   HEAT:  HEAT ENTERING AIRWAY IN BTU/MIN.                               $
$   O2MIN: OXYGEN CONCENTRATION OF AIR CURRECT LEAVING FIRE ZONE, %.      $
$   SMPO2: CONTAMINANT PRODUCTION PER FT**3 OF OXYGEN DELIVERY.           $
$   HTPO2: HEAT PRODUCTION (BTU) PER FT**3 OF OXYGEN DELIVERY.            $
$   QCENT: REF. Q DEFINING FIRE CHARACTERISTICS, CFM.                     $
$   TPR: LEADING TIME PERIOD IN MIN. FOR FIRE TO REACH FULL STRENGTH     .$
$   CN:  AVERAGE CARBON CONTENT IN FUEL.                                  $
$   HN:  AVERAGE HYDROGEN CONTENT IN FUEL.                                $
$                                                                         $
$       ***   INPUT DATA FOR FIRE SOURCES HERE      ***                   $
$----- NCENT,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,QCENT,TPR,(CN,HN) ----------$

$-------------------------------------------------------------------------$
$ (9) CONTROL DATA II                                                     $
$ NAV,JSTART,TSTART,TIME,CRITSM,CRITGS,CRITHT,WRNPR,WRNSM,WRNGS,          $
$ WRNHT,HSTART,CRITW                                                      $
$                                                                         $
$ * A SET OF DEFAULT VALUES ARE AVAILABLE. THEY CAN BE ACTIVATED BY       $
$ INPUTING ZERO FOR THE RESPECTIVE TERM.                                  $
$                                                                         $
$   NAV: MARK. 1 OR LARGER: AVERAGE-VALUE CARD IS PRESENT.                $
$   JSTART: STARTING JUNCTION (ATMOSPHERE).                               $
$   TSTART: TEMPERATURE AT THE START JUNCTION, DEGREE F.                  $
$   TIME: TIME SPAN TO ASSUME QUASI-EQUILIBRIUM, HOUR. DEFAULT 10.        $
$   CRITSM: ACCURACY IN FUME CALCULATION, %. DEFAULT 0.005.               $
$   CRITGS: ACCURACY IN METHAME CALCULATION, %, DEFAULT .01.              $
$   CRITHT: ACCURACY IN TEMPERATURE CALCULATION, F. DEFAULT 0.1.          $
$   WRNPR: PRESSURE DROP WARNING CRITERIA, IN.W.G. DEFAULT 0.01.          $
$   WRNGS: METHANE CONCENTRATION WARNING CRITERIA, %. DEFAULT 1.0.        $
$   WRNSM: FUME CONCENTRATION WARNING CRITERIA, %. DEFAULT 0.05.          $
$   WRNHT: HIGH TEMPERATURE WARNING CRITERIA, F. DEFAULT 100.0.           $
$   HSTART: REL HUMI. IN START JUNCTION JSTART, PERCENTAGE.               $
$   CRITW: CRITERIA IN WALL TEMP. TO ASSUME DRY CONDITION, DEGREE F.      $
$        DEFAULT 122 F.                                                   $
$                                                                         $
$      ***   INPUT DATA FOR CONTROL CARD II HERE   ***                    $
$ NAV,JSTART,TSTART,TIME,CRITSM,CRITGS,CRITHT,WRNPR,WRNSM,WRNGS,WRNHT $
```

```
$ HSTART,CRITW                                                          $

$---------------------------------------------------------------------------$
$ (10) AVERAGE-VALUE DATA LINE:                                             $
$   HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR,TAVR                                  $
$                                                                           $
$   IT MUST EXIST IF NAV IN CONTROL DATA II IS 1 OR LARGER. A DEFAULT       $
$ VALUE CAN BE ACTIVATED BY INPUTING ZERO FOR THE RESPECTIVE TERM.          $
$ WHEN NAV IS ZERO OR LESS, A WHOLE SET OF DEFAULT VALUES WILL BE           $
$ TAKEN.                                                                    $
$                                                                           $
$   HAAVR: AVERAGE THERMAL DIFFUSIVITY OF ROCK, FT2/HR. DEFAULT 0.1.        $
$   HKAVR: AVERAGE THERMAL CONDUCTIVITY OF ROCK, BTU/HR*FT*F.               $
$         DEFAULT 3.0.                                                      $
$   KFAVR: FRICTION FACTOR, 1.E-10 LBM*MIN2/FT4. DEFAULT 100.               $
$   LAAVR: AVERAGE LENGTH OF AIRWAY, FT. DEFAULT 500.                       $
$   AAVR: AVERAGE SECTIONAL AREA OF AIRWAY, FT2. DEFAULT 100.               $
$   OAVR: AVERAGE PERIMETER, FT. DEFAULT 40.                                $
$   TAVR: AVERAGE ROCK TEMPERATURE, DEGREE F, DEFAULT 75.                   $
$                                                                           $
$     ***  INPUT AVERAGE-VALUE CARD HERE IF NEEDED  ***                     $
$------------- HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR,TAVR ----------------------$

$---------------------------------------------------------------------------$
$  (11) TIME TABLE FOR CONDITION CHANGES:                                   $
$                                                                           $
$   THIS TIME TABLE CAN CONTAIN THE FOLLOWING CONDITION CHANGES.            $
$   THE ORDER OF CHANGING CONDITIONS MUST BE ARRANGED                       $
$   ACCORDING TO THE RESPECTIVE OCCURING TIME.                              $
$   TS: OCCURING TIME OF THE CHANGE CONDITION, MIN.                         $
$   NBR: AIRWAY ID NUMBER.                                                  $
$                                                                           $
$  1) TS,1                                                                  $
$    NBR,R                                                                  $
$    THIS INPUT CHANGES BRANCH "NBR" TO AN ORDINARY AIRWAY WITH             $
$   AIRFLOW RESISTANCE EQUAL TO "R" AT "TS" MIN. AFTER TIME ZERO.           $
$    R: RESISTANCE OF AIRWAY 1.E-10 IN. W.G./(CFM)2.                        $
$  2) TS,2                                                                  $
$    NBR,INDEX                                                              $
$    QF1,PF1,QF2,PF2,QF3,PF3,QF4,PF4,QF5,PF5                                $
$    THIS INPUT CHANGES BRANCH "NBR" TO A FAN BRANCH WITH ITS               $
$   CHARACTERISTICS SPECIFIED BY QF AND PF (NUMBER OF QF-PF PAIRS           $
$   EQUALS INDEX)                                                           $
$    INDEX: NUMBER OF DATA POINTS TO DEFINE FAN CHAR. MAX. 5.               $
$    QF: AIRFLOW RATE IN FT**3/MIN.                                         $
$    PF: CORRESPONDING FAN PRESSURE IN IN. W.G.                             $
$  3) TS,3                                                                  $
$    NBR,RX                                                                 $
$    THIS INPUT CHANGES BRANCH NBR TO A FIXED PRESSURE FAN BRANCH           $
$   OF PRESSURE RX AT TS MIN AFTER EVENT.                                   $
$    RX: FIXED FAN PRESSURE, IN.W.G.                                        $
$  4) TS,4                                                                  $
$    NBR,CONT,CONC,HEAT,02MIN,SMPO2,HTPO2,QCENT,TPR(,CN,HN)                 $
$    THIS INPUT CHANGES BRANCH "NBR" TO A FIRE SOURCE BRANCH WITH           $
$   FIRE CHAR. SPECIFIED BY "CONT" TO "TPR", WHICH RETAIN THE SAME          $
$   MEANING AS IN FIRE SOURCE DATA (7). REFER BACK TO (7) FOR DETAIL        $
```

```
$   5) TS,5                                                          $
$     NBR                                                            $
$     THIS INPUT CHANGES FIRE BRANCH "NBR" TO AN ORDINARY AIRWAY AT  $
$   TIME "TS".                                                       $
$   6) TS,6                                                          $
$     TINC                                                           $
$     THIS INPUT CHANGES TIME INCREMENT IN SYNAMIC SIMULATION TO A   $
$   NEW VALUE "TINC" IN SEC.                                         $
```

APPENDIX 2 The Modified Subroutines of Program MFIRE version 2.10

```
C
C
      SUBROUTINE  ADPT (IS,IQ,MARKS,I)
C
C     ---------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) ADVANCING EXISTING C.V. IN AIRWAYS, UPDATING RELEVANT DATA.
C       2) UPDATING AIRWAY ENDING CONDITIONS.
C       3) CONDENSING DATA STORAGE ARRAYS IF THERE IS A C.V. EXCEEDING
C          THE AIRWAY ENDING.
C
C     ---------------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
C     IS: NUMBER OF C.V. IN AIRWAY NO(I).
C
      CP=0.24
      IF (NSAC(I,2).EQ.0) RETURN
      DO 5 K=1,NJ
        IF (JS(I).EQ.JNO(K)) GO TO 8
5     CONTINUE
8     IF (MARKS.GT.0.AND.JCH(K).EQ.0) RETURN
      IF (IS.LE.0) GO TO 50
      IF (DPPA(IQ,IS+1,1).GT.LA(I)) GO TO 50
C
C       "MARKS" ACTS AS A TIMER. WHEN MARKS=0, THIS SUB. ADVANCES ALL
C     EXISTING C.V., WHILE MARKS>0, IT SEARCHES FOR THOSE C.VS. WHICH
C     WERE GENERATED IN THE PRESENT INTERVAL AND TRAVELLED MORE THAN
C     ONE AIRWAY IN THE PRESENT INTERVAL THEN UPDATES THE RELEVANT DATA.
C
C  ** MARKS=0, ADVANCING ALL EXISTING DATA POINTS, STARTING WITH THE C.V
C     CLOSEST TO THE BEGINNING JUNCTION OF THE AIRWAY.
C
      DO 40 K=1,IS
        JA=IS-K+1
        NN=0
        MK=0
        SEG=DPPA(IQ,JA+1,1)-DPPB(IQ,JA+1,1)
        IF (ABS(SEG).LT.1.E-3) SEG=0.001
        IF (NHM.EQ.0) THEN
C
C     FOR PURELY DRY CASE: (REFER TO THE RELEVANT PART OF SUB CDJN.)
C
          FA=XNEW(I)*ABS(SEG)/CP
C
C       DPPA RECORDS UPDATED DATA. THE DATA CORRESPONDING TO THE END
C     OF LAST INTERVAL IS STORED IN ITS BACKUP ARRAY DPPB WHICH DOES
C     NOT CHANGE WITHIN EACH TIME INCREMENT.
C       SEG: C.V. TRVELLING DISTANCE IN THE PRESENT INTERVAL.
C
```

```
      VL=SEG/LA(I)
      T0=TAUXB(IQ,JA)
      T2=DPPB(IQ,JA,2)
      IRV=0
      IF ((TAUXB(IQ,JA).LT.DPPB(IQ,JA,2).AND.TAUXB(IQ,JA).GT.
     .   TROCK(I)).OR.(TAUXB(IQ,JA).GT.DPPB(IQ,JA,2).AND.
     .   TAUXB(IQ,JA).LT.TROCK(I))) THEN
         T0=DPPB(IQ,JA,2)
         T2=TAUXB(IQ,JA)
         IRV=1
      ENDIF
      FAS=0.0
      IF (FA.LT.25.0) FAS=EXP(-FA)
      T1=TROCK(I)+(T0-TROCK(I))*FAS-
     .   (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
      TAUXA(IQ,JA)=T1
      CALL COLUMN (T1,SEG1,I,IQ,JA)
      SEG=DPPA(IQ,JA+1,1)+SEG1-DPPB(IQ,JA,1)
      FA=XNEW(I)*ABS(SEG)/CP
      VL=SEG/LA(I)
      FAS=0.0
      IF (FA.LT.25.0) FAS=EXP(-FA)
      DPPA(IQ,JA,2)=TROCK(I)+(T2-TROCK(I))*FAS-
     .           (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
      IF (JA.EQ.1) TAUXC(IQ)=TROCK(I)+(TAUXD(IQ)-TROCK(I))*FAS-
     .           (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
      IF (IRV.EQ.1) THEN
         TBW=DPPA(IQ,JA,2)
         DPPA(IQ,JA,2)=TAUXA(IQ,JA)
         TAUXA(IQ,JA)=TBW
      ENDIF
      DPPA(IQ,JA,1)=DPPB(IQ,JA,1)+SEG
     ELSE
C
C    FOR WET CASE: (REFER TO THE RELEVANT SECTION IN SUB. CDJN.)
C
      HC=BI(I)*HK(I)*O(I)/(2.0*A(I))
      HLW=1050.0
      DRR=DR
      NNA=20
      IF (DCOAGE(I).LT.1.E-5) THEN
         DCOAGE(I)=0.6
         NNA=3
         MARKE=1
      ENDIF
C
C     NSP(I): NUMBER OF FIXED STATIONS IN AIRWAY NO(I) AT WHICH
C    WET CASE TEMP. CORR. ARE EVALUATED.
C
      JLE=NSP(I)
      T0=TAUXB(IQ,JA)
      IF (T0.LT.DPPB(IQ,JA,2)) T0=DPPB(IQ,JA,2)
      TQ=T0-5.0
      T2=0.0
      TTSS=0.7*T0
      IF (T0.LT.200.0) TTSS=0.9*T0
```

```
25    TXM=DCOAGE(I)*(TROCK(I)-TTSS)/BI(I)+TTSS
      CALL MTCOEF (T0,TQ,I,BETA)
      WC=0.0
      WD=0.0
      NC=1
      DO 22 JL=JLE,1,-1
        SDL=LA(I)*(JL-1)/JLE
        IF (MK.EQ.0) THEN
          DS=0.0
          IF (JA.LT.IS) DS=DPPB(IQ,JA+1,1)
        ELSE
          DS=DPPA(IQ,JA+1,1)
        ENDIF
        IF (SDL.LE.(DS+ABS(SEG))) THEN
          WC=WC+WMTR(IQ,JL)
          WD=WD+GFF(IQ,JL)
          NC=NC+1
          IF (DS.GE.SDL) GO TO 23
        ENDIF
22    CONTINUE
23    GADD=WC/NC
      GF=WD/NC
      IF (GF.LT.0.0) GF=0.0
      DRR=(DR+DR*GADD*18/28.98)*(TR+460)/((TTSS+460)*(1+GADD))
      GDP=GADD*DRR
      IF (GDP.LT.0.0) GDP=0.0
      CNT(I)=HC*WTCORR(IQ,JL)+HK(I)*DCOAGE(I)*TROCK(I)
     .      *O(I)/(2.0*A(I))+BETA*GDP*CP*(TXM-TQ)-
     .      HLW*QQ(I)*DR*GF/(O(I)*ABS(SEG))
      FAW(I)=HK(I)*DCOAGE(I)*O(I)*O(I)/(120.0*QQ(I)*A(I)*CP*DR)
      IF (MK.EQ.1) CALL COLUMN (T0,SEG,I,IQ,JA)
      IF (FAW(I)*SEG.LT.25.0) THEN
        EXPN(I)=EXP(-FAW(I)*SEG)
      ELSE
        EXPN(I)=0.0
      ENDIF
      T1=2*A(I)*(CNT(I)-(CNT(I)-HK(I)*DCOAGE(I)*O(I)
     .          *T0/(2*A(I)))*EXPN(I))/(HK(I)*DCOAGE(I)*
     .          O(I))-DZRD(I)*SEG/(LA(I)*778.26*CP)
      NN=NN+1
      IF (ABS(T1-T2).GT.1.0) THEN
        TSS=TTSS
        B=2*A(I)/(HK(I)*DCOAGE(I)*O(I)*FAW(I)*SEG)
        E=B*(CNT(I)-HK(I)*DCOAGE(I)*T0*O(I)/(2*A(I)))
     .   *(1.0-EXPN(I))
        TTSS=2*A(I)*CNT(I)/(HK(I)*DCOAGE(I)*O(I))-E-
     .      SEG*DZRD(I)/(2*778.26*LA(I)*CP)
        TQ=T1
        T2=T1
        IF (NN.LE.NNA) GO TO 25
        WRITE (8,*) 'SUB. ADPT  LINE 170',I,IS,MARKS
      ENDIF
      IF (MK.EQ.0) THEN
        MK=1
        NN=0
        TAUXA(IQ,JA)=T1
```

```
          T0=T1
          TQ=T1-5.0
          SEG=QQ(I)*DELTAT*(TTSS+460)/(A(I)*(TR+460)*60)-SEG
          IF (ABS(SEG).LT.1.E-3) SEG=0.001
          GO TO 25
        ENDIF
        DPPA(IQ,JA,1)=DPPA(IQ,JA+1,1)+SEG
        DPPA(IQ,JA,2)=T1
      ENDIF
      SE=ABS(DPPA(IQ,JA,1)-DPPB(IQ,JA,1))
      DPPA(IQ,JA,4)=(DPPB(IQ,JA,4)*QQ(I)+CH4V(I)*SE/LA(I))/
     .              (QQ(I)+CH4V(I)*SE/LA(I))
      IF (DPPA(IQ,JA,1).GE.LA(I)) GO TO 50
40    CONTINUE
      FA=XNEW(I)*ABS(DPPA(IQ,1,1)-DPPB(IQ,1,1))/CP
      FAS=0.0
      IF (FA.LT.25.0) FAS=EXP(-FA)
      VL=(DPPA(IQ,1,1)-DPPB(IQ,1,1))/LA(I)
      TAUXC(IQ)=TROCK(I)+(TAUXD(IQ)-TROCK(I))*FAS-(DZRD(I)*VL/
     .          (2.0*778.26*CP))*(1.0+FAS)
50    ISS=NSAC(I,2)
      DO 52 K=1,ISS
        JA=ISS-K+1
        IF (DPPA(IQ,JA,1).GE.LA(I)) GO TO 54
52    CONTINUE
      IF (NSAC(I,1).EQ.1) THEN
        IF (NHM.GT.0.AND.RDH2O(I).LT.1.E-10) THEN
          HU1=RDH2O(I)
          TU=TRD(I)
          TU=DCOAGE(I)*(TROCK(I)-TU)/BI(I)+TU
          IF (TU.GT.210.0) TU=210.0
          IF (TU.LE.122.0) THEN
            PVW=-1.608+0.07946*TU-0.00129*TU*TU+1.26E-5*TU**3
          ELSE
            PVW=-33.786+0.97*TU-9.305E-3*TU*TU+3.65E-5*TU**3
          ENDIF
          GAW=0.622*PVW/(101.35-PVW)
          GAWS=GAW*PSI(I)
          RDH2O(I)=(GAW+GAWS)/2.0
        ENDIF
        RETURN
      ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 51 K=1,ISS
        JA=ISS-K+1
        IF (DPPA(IQ,JA,3).NE.0) GO TO 53
51    CONTINUE
      RDPROP(I)=0.
      RETURN
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     IF NO C.V. EXCEEDS THE ENDING, THEN SUBROUTINE RETURNS AFTER
C     UPDATING AIRWAY ENDING CONDITIONS.
C
53    SEG=LA(I)-DPPA(IQ,1,1)
      SSL=LA(I)-DPPB(IQ,1,1)
```

```
      IF (SSL.LE.SEG) SSL=SEG
      IF (NHM.EQ.0) THEN
        FA=XNEW(I)*SEG/CP
        IF (FA.LE.25.0) THEN
          FAS=EXP(-FA)
        ELSE
          FAS=0.0
        ENDIF
        TT=DPPA(IQ,1,2)
        ENDT=TROCK(I)+(TT-TROCK(I))*FAS-(DZRD(I)*SEG/
     .      (2*778.26*CP*LA(I)))*(1.0+FAS)
      ELSE
        FA=FAW(I)*SEG
        IF (FA.LE.25.0) THEN
          FAS=EXP(-FA)
        ELSE
          FAS=0.0
        ENDIF
        TT=DPPA(IQ,1,2)
        ENDT=2*A(I)*(CNT(I)-(CNT(I)-HK(I)*DCOAGE(I)*O(I)
     .      *TT/(2*A(I)))*FAS)/(HK(I)*DCOAGE(I)*
     .      O(I))-DZRD(I)*SEG/(LA(I)*778.26*CP)
      ENDIF
      TRD(I)=TTRD(I)+(ENDT-TTRD(I))*(SSL-SEG)/SSL
      RDPROP(I)=RDOP(I)+(DPPA(IQ,1,3)-RDOP(I))*(SSL-SEG)/SSL
      RETURN
C
C ** C.V. EXCEEDS THE ENDING, UPDATE OF AIRWAY ENDING CONDITIONS
C    IS CALLED FOR.
C    TLEFT(J): TIME LEFT AT JUNCTION JNO(J) FOR WAVE FRONT (C.V.)
C            TO TRAVEL INTO THE DOWN-STREAM AIRWAY.
C
54    DO 56 LL=1,NJ
        IF (IABS(JF(I)).EQ.JNO(LL)) KL=LL
        IF (IABS(JS(I)).EQ.JNO(LL)) ML=LL
56    CONTINUE
      FL=DPPA(IQ,JA,1)-LA(I)
C***  FBN(I)=FL/(DPPA(IQ,JA,1)-DPPB(IQ,JA,1))
      TK=DELTAT*FL/((DPPA(IQ,JA,1)-DPPB(IQ,JA,1))*60.0)
      IF (TK.GT.TLEFT(KL)) TLEFT(KL)=TK
C
C    NSAC(I,1)=-1: AIRWAY NO(I) IS A FUME-FILLED ONE.
C
      IF (NSAC(I,1).NE.(-1)) NSAC(I,1)=-1
C
C    RDPROP(I): FUME CONCENTRATION AT AIRWAY ENDING.
C    RDCH4(I): CH4 CONCENTRATION AT AIRWAY ENDING.
C    RDH2O(I): HUMIDITY RATIO AT AIRWAY ENDING.
C    DPPA,DPPB: (1) DISTANCE, (2) TEMP. (3) FUME, (4) CH4,
C             (5) AIR MASS IN C.V.
C
      RDCH4A=DPPA(IQ,JA,4)-CH4V(I)*(DPPA(IQ,JA,1)-LA(I))/
     .      (LA(I)*QQ(I))
C
C    NSACB: BACKUP ARRAY OF NSAC. NSACB(I,2) INDICATES THE NUMBER
C         OF C.V. IN AIRWAY NO(I) IN THE LAST INTERVAL.
```

```
C
      IF (JA.EQ.NSAC(I,2)) THEN
        SECT=LA(I)
        TLAST=TJS(I)
        RDF=DPPA(IQ,JA,3)-PROP(ML)
        IF (NWTYP(I).EQ.10) RDF=0.0
        SSL=DPPA(IQ,JA,1)
      ELSE
        SECT=LA(I)-DPPA(IQ,JA+1,1)
        TLAST=TAUXA(IQ,JA)
        RDF=DPPA(IQ,JA,3)-DPPA(IQ,JA+1,3)
        SSL=DPPA(IQ,JA,1)-DPPA(IQ,JA+1,1)
      ENDIF
      RDPROP(I)=DPPA(IQ,JA,3)-RDF*FL/SSL
      RDCH4(I)=RDCH4A
      IF (NHM.EQ.0) THEN
C
C      AIR TEMP. AT AIRWAY ENDING IN DRY CASE.
C
        IF ((TAUXB(IQ,JA).LT.DPPB(IQ,JA,2).AND.TAUXB(IQ,JA).GT.
     .    TROCK(I)).OR.(TAUXB(IQ,JA).GT.DPPB(IQ,JA,2).AND.
     .    TAUXB(IQ,JA).LT.TROCK(I))) THEN
C
C      HANDLING REVERSED AIRFLOW
C
          FL1=DPPB(IQ,JA,1)-DPPB(IQ,JA+1,1)
          FLL=(DPPA(IQ,JA,1)-LA(I))/(DPPA(IQ,JA,1)-DPPA(IQ,JA+1,1))
          TLAST=DPPB(IQ,JA,2)+(TAUXB(IQ,JA)-DPPB(IQ,JA,2))*FLL
          SECT=LA(I)-DPPB(IQ,JA,1)+FL1*FLL
        ENDIF
        FA=XNEW(I)*SECT/CP
        FAS=0.0
        IF (FA.LT.25.0) FAS=EXP(-FA)
        VL=SECT/LA(I)
        TRD(I)=TROCK(I)+(TLAST-TROCK(I))*FAS-
     .        (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
      ELSE
C
C      AIR TEMP. AT AIRWAY ENDING IN WET CASE.
C
        EPF3=FAW(I)*SECT
        EXPN(I)=EXP(-EPF3)
        TRD(I)=2*A(I)*(CNT(I)-(CNT(I)-HK(I)*DCOAGE(I)*O(I)
     .        *TLAST/(2*A(I)))*EXPN(I))/(HK(I)*DCOAGE(I)*
     .        O(I))-DZRD(I)*SECT/(LA(I)*778.26*CP)
      ENDIF
C
C      WHEN A C.V EXCEEDS AIRWAY ENDING, THE DATA RELATED TO THAT C.V.
C     ARE DISCARDED AND DATA RECORDING ARRAYS ARE CONDENSED.
C
      JAA=JA+1
      DO 80 K=JAA,11
        IF (K.EQ.JAA) TAUXC(IQ)=TAUXA(IQ,K-1)
        IF (K.LE.10) THEN
          KA=K-JAA+1
          TAUXA(IQ,KA)=TAUXA(IQ,K)
```

```
          DO 60 L=1,5
            DPPA(IQ,KA,L)=DPPA(IQ,K,L)
60        CONTINUE
        ELSE
          TAUXA(IQ,10)=0.0
          DO 70 L=1,5
            DPPA(IQ,10,L)=0.0
70        CONTINUE
        ENDIF
80    CONTINUE
      NSAC(I,2)=NSAC(I,2)-JA
C
      RETURN
      END

C
C
C
      SUBROUTINE ARR (MARKX)
C
C     -----------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) GROUPING AIRWAYS ACCORDING TO THEIR TYPES.
C       2) ARRANGING REGULAR AIRWAYS ACCORDING TO THE PRODUCT OF
C          Q AND R.
C
C     -----------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
C       THE ORDERED AIRWAY LIST IS STORED IN ARRAY INU. FIXED
C     QUANTITY AIRWAYS ARE ON ITS "BOTTOM". THEN COME THE FAN
C     BRANCHES. THE REGULAR AIRWAYS WHICH CONTAIN NO FAN AND NO
C     ADDITIONAL RESTRICTIONS SPECIFIED BY USER ARE ON THE TOP OF
C     LIST.
C
      NBU=NB
      NBL=1
      DO 10 K=1,NB
        IF (NWTYP(K).LT.0) THEN
          INU(NBU)=K
          NBU=NBU-1
        ELSE IF (NWTYP(K).EQ.0.OR.NWTYP(K).EQ.10) THEN
          RQ(K)=ABS(R(K)*Q(K))
          NWTYP(K)=2
        ELSE
          IF (NFNUM.GT.0) THEN
            DO 5 J=1,NFNUM
              IF (NOF(J).EQ.NO(K).AND.NWTYP(K).EQ.1) GO TO 10
5           CONTINUE
          ENDIF
          INU(NBL)=K
          NBL=NBL+1
        ENDIF
10    CONTINUE
```

```
      IF (NFNUM.GT.0) THEN
        DO 25 K=1,NB
          DO 20 J=1,NFNUM
            IF (NOF(J).EQ.NO(K).AND.NWTYP(K).EQ.1) THEN
C                          *****************
              RGRAD(J)=0.0
              NFCW(J)=0
              NFREG(J)=K
              INU(NBU)=K
              NBU=NBU-1
              GO TO 25
            ENDIF
20        CONTINUE
25      CONTINUE
      ENDIF
C
C       REGULAR AIRWAYS ARE STORED IN THE ORDER OF THE PRODUCTS
C     OF Q AND R.
C
C
      IF (NBU.GT.(NBL-1)) THEN
30      NRETU=0
        DO 40 K=1,NB
          IF (NWTYP(K).GE.2) THEN
            IF (NRETU.LE.0) THEN
              MMIN=K
              NRETU=1
            ENDIF
            IF (RQ(MMIN).GT.RQ(K)) MMIN=K
          ENDIF
40      CONTINUE
        INU(NBL)=MMIN
        NBL=NBL+1
        NWTYP(MMIN)=0
        IF ((NBU+1).GT.NBL) GO TO 30
      ENDIF
C
C       AIRWAYS CONTAINING FIRE SOURCES HAVE THEIR NWTYP(I) EQUAL TO
C     10 TO DISTINGUISH THEMSELVES FROM OTHERS.
C     MARKX=1: THE PROGRAM HAS REACHED THE DYNAMIC SIMULATION PART.
C
      IF (MARKX.GT.0) THEN
        DO 60 I=1,INFLOW
          DO 50 J=1,NB
            IF (NCENT(I).EQ.NO(J)) THEN
              NWTYP(J)=10
              GO TO 60
            ENDIF
50        CONTINUE
60      CONTINUE
      ENDIF
C
      RETURN
      END
```

```
C
C
      SUBROUTINE CCDATA (MARKY,NSTOP)
C
C     ------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) COMPLETION OF INPUT DATA.
C       2) EVALUATION OF ROCK TEMP. AND AIRWAY WETNESS.
C
C     ------------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
C       NAV>0 INDICATES THE EXISTANCE OF AVERAGE VALUE CARD FOR
C     MISSING INFORMATION.
C
      IF (NAV.GT.0) THEN
        READ (9,*,ERR=130) HAAVR,HKAVR,A1,A2,AAVR,OAVR,TAVR
        KFAVR=A1+0.4
        LAAVR=A2+0.4
        IF (IOUT.LE.(-2)) THEN
          WRITE (8,330) HAAVR,HKAVR,KFAVR,LAAVR,AAVR,OAVR,TAVR
        ENDIF
      ELSE
        TAVR=75.0
        HAAVR=0.1
        HKAVR=3.0
        KFAVR=100
        LAAVR=500
        AAVR=100.0
        OAVR=40.0
        IF (MARKY.EQ.0) WRITE (8,290)
      ENDIF
      IF (HAAVR.LT.1.E-3.OR.HAAVR.GT.1.E3) HAAVR=0.1
      IF (HKAVR.LT.1.E-3.OR.HKAVR.GT.1.E3) HKAVR=3.0
      IF (KFAVR.LT.1.OR.KFAVR.GT.10000) KFAVR=100
      IF (LAAVR.LT.1.OR.LAAVR.GT.10000) LAAVR=500
      IF (AAVR.LT.1.E-3.OR.AAVR.GT.1.E5) AAVR=100.0
      IF (OAVR.LT.1.E-3.OR.OAVR.GT.1.E5) OAVR=40.0
      IF (TAVR.LT.(-50.0).OR.TAVR.GT.200.0) TAVR=75.0
      IF (MARKY.EQ.0) WRITE (8,310) TAVR,HAAVR,HKAVR,KFAVR,LAAVR,
     .                     AAVR,OAVR
C
C       MISSING OR INCONSISTENT DATA ARE MADE UP WITH THE USER-SPECIFIED
C     OR DEFAULT AVERAGE VALUES.
C
      DO 50 I=1,NB
        IF (HA(I).LE.0.0) HA(I)=HAAVR
        IF (HK(I).LE.0.0) HK(I)=HKAVR
        IF (KF(I).LE.0) KF(I)=KFAVR
        IF (LA(I).LE.0.0) LA(I)=FLOAT(LAAVR)
        IF (A(I).LE.0.0) A(I)=AAVR
        IF (O(I).LE.0.0) O(I)=OAVR
50    CONTINUE
C
```

```
C  ** CALCULATION OF ROCK TEMP. AND AIRWAY WETNESS.
C
      DO 120 J=1,NB
        IF (NO(J).LE.0) NO(J)=-NO(J)
        L=0
        M=0
C
C    NHM=0: DRY CASE ASSUMED.
C    NHM=1: WET CASE ASSUMED.
C
        IF (NHM.EQ.0) THEN
          DO 60 I=1,NJ
            IF (JS(J).EQ.JNO(I)) THEN
              ES=Z(I)
              TRS=T(I)
              L=1
            ELSE IF (JF(J).EQ.JNO(I)) THEN
              EF=Z(I)
              TRF=T(I)
              M=1
            ENDIF
            IF (Q(J).LT.0.0) THEN
              STORE=TRF
              TRF=TRS
              TRS=STORE
            ENDIF
            IF ((L+M).EQ.2) THEN
             IF (JF(J).EQ.JSTART.OR.(JF(J).GE.900.AND.JF(J).LE.950))
     .         TRF=TRS
              DZRD(J)=EF-ES
C*****************************************************
              IF (ABS(DZRD(J)).GT.LA(J)) THEN
                LA(J)=ABS(DZRD(J))
                WRITE (8,320) NO(J),LA(J)
              ENDIF
C*****************************************************
              X=0.014*O(J)/(A(J)**0.8*(ABS(Q(J)))**0.2)
              TROCKA=(TRF-TRS*EXP(-X*LA(J))+DZRD(J)*
     .        (1.+EXP(-X*LA(J)))/(2*187.))/(1.0-EXP(-X*LA(J)))
              IF (ABS(TROCK(J)).LT.1.E-3.OR.TROCK(J).LT.(-50.0).OR.
     .          TROCK(J).GT.200.0) TROCK(J)=TROCKA
              GO TO 70
            ENDIF
60        CONTINUE
70        CONTINUE
        ELSE
C
C    WET CASE:
C
          DO 110 I=1,NJ
            IF (JS(J).EQ.JNO(I)) THEN
              ES=Z(I)
              TRS=T(I)
              GB=PRH2O(I)
              L=1
            ELSE IF (JF(J).EQ.JNO(I)) THEN
```

```
          EF=Z(I)
          TRF=T(I)
          GX=PRH2O(I)
          M=1
        ENDIF
        IF (Q(J).LT.0.0) THEN
          STORE=TRF
          TRF=TRS
          TRS=STORE
        ENDIF
        IF ((L+M).EQ.2) THEN
         IF (JF(J).EQ.JSTART.OR.(JF(J).GE.900.AND.JF(J).LE.950))
            TRF=TRS
          DZRD(J)=EF-ES
          IF (ABS(DZRD(J)).GT.LA(J)) THEN
            LA(J)=ABS(DZRD(J))
C                 ************
            WRITE (8,320) NO(J),LA(J)
          ENDIF
          X=0.014*O(J)/(A(J)**0.8*(ABS(Q(J)))**0.2)
          TROCKA=(TRF-TRS*EXP(-X*LA(J))+DZRD(J)*
     .    (1.+EXP(-X*LA(J)))/(2*187.))/(1.0-EXP(-X*LA(J)))
          IF (ABS(TROCK(J)).LT.1.E-3.OR.TROCK(J).LT.(-50.0).OR.
     .      TROCK(J).GT.200.0) TROCK(J)=TROCKA
C
C      SUB. MTCOEF SERVES CALCULATION OF MASS TRANSFER COEFF. BETA.
C
          IF (MARKY.EQ.0) THEN
            TX=TROCK(J)
            CALL MTCOEF (TX,TX,J,BETA)
            DRR=DR*(TR+460)/(TX+460)
            TY=TX
            IF (TY.GT.210.0) TY=210.0
            IF (TY.LE.122.0) THEN
              PV=-1.068+0.07946*TY-0.00129*TY*TY+1.26E-5*TY**3
            ELSE
              PV=-33.786+0.97*TY-9.306E-3*TY*TY+3.65E-5*TY**3
            ENDIF
            GW=0.622*PV/(101.35-PV)
C
C     EVALUATION OF WALL WETNESS.
C
            DO 80 KI=1,INFLOW
              IF (NO(J).EQ.NCENT(KI)) THEN
                PSI(J)=0.0
                GO TO 90
              ENDIF
80           CONTINUE
            IF (GW.LT.GB.AND.GB.LE.GX) THEN
              WRITE (8,270) NO(J)
              PSI(J)=1.0
            ELSE IF (GW.GT.GB.AND.GB.GE.GX) THEN
              WRITE (8,275) NO(J)
            ELSE IF (GX.GE.GB) THEN
              AL=BETA*O(J)*LA(J)/(60.0*Q(J))
              IF (AL.LT.10.0) THEN
```

```
                EL=EXP(-AL)
              ELSE
                EL=0.0
              ENDIF
              PSI(J)=(GX-GB*EL)/(GW*(1.0-EL))
              IF (PSI(J).GT.1.0) PSI(J)=1.0
            ELSE IF (GX.LE.GB) THEN
              WRITE (8,280) NO(J),PSI(J)
              IF (PSI(J).LT.0.0) THEN
                WRITE (8,260) PSI(J),NO(J)
                PSI(J)=0.0
              ENDIF
            ENDIF
          ENDIF
90         IF (PSI(J).LT.0.0) PSI(J)=0.0
          IF (ABS(PSI(I)).LT.0.01) PSI(J)=0.6
C
C     HUMIDITY RATIO AT AIRWAY ENDING IS ASSUMED TO EQUAL THE
C    HUMIDITY RATIO IN THE JUNCTION WHICH THE AIRWAY IS
C    CONNECTED TO.
C
          DO 100 IK=1,NJ
            IF (JF(J).EQ.JNO(IK)) THEN
              RDH2O(J)=PRH2O(I)
              GO TO 120
            ENDIF
100          CONTINUE
        ENDIF
110       CONTINUE
     ENDIF
120  CONTINUE
     DO 125 I=1,NB
       IF (TROCK(J).LT.(-50.0).OR.TROCK(J).GT.200.0) TROCK(J)=TAVR
125  CONTINUE
     RETURN
130  WRITE (6,340)
     WRITE (8,340)
     NSTOP=1
     RETURN
C
220  FORMAT (I5,F10.2,4F10.5,F10.1)
260  FORMAT (/,' * ATTN * INVALID WETNESS INPUT (',F6.3,') FOR AIRWAY',
    .      I6,/,1X,'ITS WETNESS IS TAKEN AS ZERO.')
270  FORMAT (/,1X,'* ATTN * INCONSISTANT DATA FOR AIRWAY',I6,' AND ITS'
    .      ,' RELATING ENDING JUNCTIONS.',/,1X,'THE WETNESS IS ',
    .      'TAKEN AS 1.0 DUE TO THE INCREASING HUMIDITY.',/)
275  FORMAT (/,1X,'* ATTN * INCONSISTANT DATA FOR AIRWAY',I6,' AND ITS'
    .      ,' RELATING ENDING',/,' JUNCTIONS. THE INPUT VALUE IS ',
    .      'TAKEN AS ITS WETNESS.',/)
280  FORMAT (/,1X,'* ATTN * WETNESS OF AIRWAY',I6,' CAN NOT BE DEFINED'
    .      ,' BY THE KNOWN',/,1X,'CONDITIONS. INPUT VALUE',F5.1,
    .      ' IS TAKEN.')
290  FORMAT (//,' * ATTN * AVE. VALUES WERE OMITTED BY USER (NAV=0)')
300  FORMAT (//,1X,'* WRN *  DUPLICATED ADDITIONAL DATA FOR AIRWAY',I5)
310  FORMAT (///,1X,'THE FOLLOWING AVERAGE VALUES WILL BE USED TO ',
    .      'FILL ANY MISSING DATA:',//,1X,'AVE. ROCK TEMPERATURE',
```

```
     .      T30,F7.1,T43,'F',/,1X,'THERMAL DIFFUSIVITY OF',
     .      ' ROCK',T30,F9.3,T43,'FT2/HR',/,1X,'ROCK THERMAL ',
     .      'CONDUCTIVITY',T30,F9.3,T43,'BTU/(HR*FT*F)',/,1X,
     .      'FRICTION FACTOR',T30,I9,T43,'1.E-10 LBF*MIN2/FT4',/,1X,
     .      'AIRWAY LENGTH',T30,I9,T43,'FT',/,1X,'SECTIONAL AREA ',
     .      T30,F9.3,T43,'FT2',/,1X,'PERIMETER',T30,F9.3,T43,'FT',/)
320  FORMAT (/,1X,'* WRN *  THE LENGTH OF AIRWAY ',I5,' IS LESS ',
     .      'THAN ITS ELEVATION',/,' DIFFERENCE, THEY ARE NOW TAKEN ',
     .      'EQUAL TO EACH OTHER AS ',F8.1)
330  FORMAT (///,' * INPUT DATA LIST (AVERAGE VALUE DATA) *',//,
     .      T2,'HAAVR=',T8,F7.2,T18,'HKAVR=',T26,F7.2,T37,'KFAVR=',
     .      T45,F7.3,T55,'LAAVR=',T63,F7.1,/,T2,'AAVR=',T8,F7.2,
     .      T18,'OAVR=',T26,F7.2,T37,'TAVR=',T45,F7.2,/)
340  FORMAT (//,' * ERR * ERROR READING INPUT DATA.',/)
C
     END


C
C
     SUBROUTINE CDCH (NSTOP,MHH)
C
C    ----------------------------------------------------------------
C
C      SUBROUTINE PURPOSE:
C      UPDATING CONDITION CHANGES ACCORDING TO USER-SPECIFIED TIME TABL
C
C    ----------------------------------------------------------------
C
     INCLUDE 'CMMNW.DAT'
C
     DIMENSION  QFX(20),PFX(20)
     DATA QFX,PFX / 40*0.0 /
C
C
     IF (TACC.LT.DELTAT.OR.MHH.EQ.1) TS=-1.0
C
C    TS: TIME IN MIN. AT WHICH A CONDITION CHANGE IS EXPECTD TO
C        BE ACTIVATED.
C
1    IF (TS.LT.0.0) READ (9,*,ERR=106,END=107) TS,ICODE
     IF (TS*60.0.GT.TACC) RETURN

C     ICODE: CODE NUMBER. 1: CHANGE TO ORDINARY AIRWAY.   2: CHANGE
C            TO FAN BRANCH.  3: CHANGE TO FIXED PRESSURE FAN BRANCH
C            4: CHANGE TO A FIRE SOURCE.     5: FIRE CEESED.
C     NBR: AIRWAY CALLING NUMBER IN WHICH CONDITION CHANGE HAPPENS.
C
C      ICODE=1 INDICATES THAT AIRWAY NBR BECOMES AN ORDINARY AIRWAY
C     WITH RESISTANCE EQUAL TO RCH AT TS MIN. AFTER EVENT.
C
     WRITE (*,'(2A)') TS,  ICODE
4    IF (ICODE.EQ.1) THEN
       READ (9,*,ERR=105,END=107) NBR,RCH
       DO 5 I=1,NB
         IF (NO(I).EQ.NBR) THEN
```

```
            N=I
            GO TO 6
          ENDIF
5       CONTINUE
        WRITE (8,210) NBR
        GO TO 100
6       IF (RCH.GT.0.0) THEN
          RSTD(N)=RCH
          R(N)=RCH*(TMRD(N)+460.0)**2/(TR+460.0)**2
          NWTYP(N)=0
          INIFAN=1
C          ********
9         WRITE (8,110) NO(N),RSTD(N),TS
        ELSE
          WRITE (8,270) NBR,NBR,RCH
          GO TO 100
        ENDIF
C
C       ICODE=2 INDICATES THAT AIRWAY NBR BECOMES A FAN BRANCH WITH
C      CHARACTERISTICS.
C
      ELSE IF (ICODE.EQ.2) THEN
        READ (9,*,ERR=105,END=107) NBR,INDEX
        DO 10 I=1,NB
          IF (NO(I).EQ.NBR) THEN
            N=I
            GO TO 15
          ENDIF
10      CONTINUE
        WRITE (8,210) NBR
        GO TO 100
15      IF (INDEX.LE.1.OR.INDEX.GT.5) THEN
          WRITE (8,120) INDEX,NBR
          GO TO 100
        ENDIF
        READ (9,*,ERR=105,END=107) (QFX(I),PFX(I),I=1,INDEX)
        IF (NFNUM.GT.0) THEN
          DO 20 J=1,NFNUM
            IF (NOF(J).EQ.NO(N)) THEN
              MPTS(J)=INDEX
              NSKP(J)=0
              DO 18 K=1,INDEX
                QF(J,K)=QFX(K)
                PF(J,K)=PFX(K)
18            CONTINUE
              JN=J
              GO TO 35
            ENDIF
20        CONTINUE
        ENDIF
        IF (NFNUM.EQ.0) INIFAN=1
C       ***************************
        NFNUM=NFNUM+1
        IF (NFNUM.GT.IMX) THEN
          NFNUM=NFNUM-1
          NWTYP(N)=1
```

```
          R(N)=PFX(1)
          WRITE (8,280) NBR,R(N)
          GO TO 100
        ENDIF
        NWTYP(N)=1
        NOF(NFNUM)=NO(N)
C        **************
        NSKP(NFNUM)=0
        NFREG(NFNUM)=N
C        **************
        MPTS(NFNUM)=INDEX
        DO 30 K=1,INDEX
          QF(NFNUM,K)=QFX(K)
          PF(NFNUM,K)=PFX(K)
30      CONTINUE
        INIFAN=1
C        ********
        JN=NFNUM
        KK=MPTS(JN)
35      WRITE (8,140) NBR,TS,(QF(JN,K),PF(JN,K),K=1,KK)
C
C       ICODE=3 INDICATES THAT BRANCH NBR BECOMES A FIXED PRESSURE
C     FAN BRANCH WITH PRESSURE EQUAL TO RX AT TS MIN. AFTER EVENT.
C
      ELSE IF (ICODE.EQ.3) THEN
        READ (9,*,ERR=105,END=107) NBR,RX
        DO 40 I=1,NB
          IF (NO(I).EQ.NBR) THEN
            N=I
            GO TO 45
          ENDIF
40      CONTINUE
        WRITE (8,210) NBR
        GO TO 100
45      IF (NFNUM.GT.0) THEN
          DO 50 J=1,NFNUM
            IF (NOF(J).EQ.NBR) NOF(J)=30000
50        CONTINUE
        ENDIF
        NWTYP(N)=1
        R(N)=RX
        WRITE (8,150) NBR,R(N)
C
C       ICODE=4 INDICATES THAT BRANCH NBR BECOMES A FIRE SOURCE AT
C     TS MIN. AFTER EVENT.
C
      ELSE IF (ICODE.EQ.4) THEN
        IF (NHM.EQ.0) THEN
          READ (9,*,ERR=105,END=107) NBR,CONTX,CONCX,HEATX,O2MINX,
     .      SMPO2X,HTPO2X,QCENTX,TPRX
        ELSE
          READ (9,*,ERR=105,END=107) NBR,CONTX,CONCX,HEATX,O2MINX,
     .      SMPO2X,HTPO2X,QCENTX,TPRX,CNX,HNX
        ENDIF
        DO 55 I=1,NB
          IF (NO(I).EQ.NBR) THEN
```

```
          N=I
          GO TO 58
        ENDIF
55     CONTINUE
      WRITE (8,210) NBR
      GO TO 100
58    IF (INFLOW.GT.0) THEN
        DO 60 J=1,INFLOW
          IF (NCENT(J).EQ.NBR) THEN
            CONT(J)=CONTX
            CONC(J)=CONCX
            HEAT(J)=HEATX
            O2MIN(J)=O2MINX
            SMPO2(J)=SMPO2X
            HTPO2(J)=HTPO2X
            TPR(J)=TPRX
            QCENT(J)=QCENTX
            CN(J)=CNX
            HN(J)=HNX
            JN=J
            GO TO 70
          ENDIF
60        CONTINUE
      ENDIF
      INFLOW=INFLOW+1
      IF (INFLOW.LE.IMX) THEN
        NCENT(INFLOW)=NBR
        CONT(INFLOW)=CONTX
        CONC(INFLOW)=CONCX
        HEAT(INFLOW)=HEATX
        O2MIN(INFLOW)=O2MINX
        SMPO2(INFLOW)=SMPO2X
        HTPO2(INFLOW)=HTPO2X
        TPR(INFLOW)=TPRX
        QCENT(INFLOW)=QCENTX
        CN(INFLOW)=CNX
        HN(INFLOW)=HNX
        JN=INFLOW
      ELSE
        INFLOW=INFLOW-1
        WRITE (8,290) NBR
        GO TO 100
      ENDIF
70     NWTYP(N)=10
      DO 75 K=1,NB
        IF (NO(K).EQ.NBR.AND.LA(K).GT.30.0)
     .     WRITE (8,220) NO(K),LA(K)
75     CONTINUE
      WRITE (8,160) NBR,TS,NBR,CONT(JN),CONC(JN),HEAT(JN),O2MIN(JN),
     .          SMPO2(JN),HTPO2(JN),TPR(JN),QCENT(JN)
C
C     ICODE=5 INDICATES THAT BRANCH NBR BECOMES AN ORDINARY AIRWAY
C    WITHOUT A FIRE SOURCE AT TS MIN. AFTER EVENT.
C
     ELSE IF (ICODE.EQ.5) THEN
       READ (9,*,ERR=105,END=107) NBR
```

```
      DO 77 I=1,NB
        IF (NO(I).EQ.NBR) THEN
          N=I
          GO TO 78
        ENDIF
77     CONTINUE
      WRITE (8,210) NBR
      GO TO 100
78     IF (INFLOW.GT.0) THEN
        DO 90 I=1,INFLOW
          IF (NCENT(I).EQ.NBR) THEN
            NCENT(I)=30005
            NWTYP(N)=0
            WRITE (8,170) NBR,TS
            GO TO 100
          ENDIF
90       CONTINUE
      ELSE
        WRITE (8,230) NBR
      ENDIF
    ELSE IF (ICODE.EQ.6) THEN
      READ (9,*,ERR=105,END=107) DELT
      IF (DELT.GT.1.0.AND.DELT.LT.60.0) THEN
        WRITE (8,190) DELTAT,DELT,TS
      ELSE
        IF (DELT.GT.60.0) WRITE (8,200) DELT
      ENDIF
      IF (DELT.GT.1.0) DELTAT=DELT
C     ELSE IF (ICODE.EQ.7) THEN
C       JDP=JDP+1
C       IF (JDP.GT.IMX) THEN
C         WRITE (8,240)
C         JDP=JDP-1
C       ELSE
C         JDPP(JDP)=NBR
C       ENDIF
    ELSE
      WRITE (8,180) ICODE, NBR
    ENDIF
100  TS=-1.0
    GO TO 1
105  NSTOP=1
    WRITE (6,260)
    WRITE (8,260)
    RETURN
106  continue
107  RETURN
C
C
110  FORMAT (//,' * ATTN * AIRWAY',I5,' BECAME AN ORDINARY AIRWAY',
   .      ' WITH RESISTANCE EQUAL',/,' TO ',F9.3,' AT',F6.2,
   .      ' MIN. AFTER SIMULATION.',/)
120  FORMAT (//,' * ATTN * CAPACITY OF FAN CURVE ARRAYS QF AND PF',
   .      ' EXCEEDED.',/,' INPUT DATA FOR CONDITION CHANGE IGNORED.')
130  FORMAT (//,' * ATTN * NUMBER OF FAN CHARACTERISTICS BECAME',
   .      ' EXCEESIVE.',/,' BRANCH',I5,' IS TAKEN AS A FAN WITH',
```

```
     .     'FIXED PRESSURE ',F6.2,' IN. W.G.',/)
140  FORMAT (//,' * ATTN * THE FOLLOWING DATA FOR FAN',
     .     I5,' ACTIVATED AT',F6.2,' MIN. AFTER EVENT:',//,
     .     5(F8.0,F6.2),/,5(F8.0,F6.2),/,5(F8.0,F6.2),/)
150  FORMAT (//,' * ATTN * AIRWAY',I5,' BECAME A FAN BRANCH WITH ',
     .     'FIXED P ',F6.2,' IN. W.G.',/,' AT',F6.2,' MIN.',
     .     ' AFTER SIMULATION.',/)
160  FORMAT (//,' * ATTN * BRANCH',I5,' BECAME A FIRE SOURCE AT',
     .     F6.2,' MIN. AFTER EVENT',//,
     .     12X,'FIXED INFLUX',T31,'O2-RICH',T41,'FUEL-RICH',T55,
     .     'LEADING T',T67,'REF. Q',/,' AIRWAY  FLUX  %CONC',
     .     3X,'HEAT',5X,'%O2',4X,'FUMES',2X,'HEAT',//,
     .     I5,F8.1,F6.2,1X,E9.3,F7.2,F7.2,F7.1,F10.2,F12.0)
170  FORMAT (//,' * ATTN * FIRE',I5,' CEESED AT',F6.2,' MIN. AFTER'
     .     ,' EVENT.',/)
180  FORMAT (//,' * ATTN * INVALID CONDITION CODE',I5,' FOR AIRWAY',I5,
     .     ' IGNORED.',/)
190  FORMAT (//,' * ATTN * SIMULATION STEP WAS CHANGED FROM',1X,
     .     F6.2,' SEC.',/,' TO ',F6.2,' SEC. AT',F6.2,' MIN. AFTER',
     .     ' EVENT.'/)
200  FORMAT (//,' * ATTN * LARGE SIMULATION STEP (',F6.2,' SEC.',
     .     ' ) WILL AFFECT',/,' SIMULATION ACCURACY.',/)
210  FORMAT (//,' * ATTN * AIRWAY',I5,' NOT FOUND, DATA IGNORED%%[ PrinterError: Out Of Paper ]%%¼¼.')
220  FORMAT (//,' * ATTN * AIRWAY',I5,' OF LENGTH',F7.0,' FT CONTAIN',
     .     ' A HEAT SOURCE.',/,' IT IS ADVISED TO DIVIDE THE ',
     .     'AIRWAY INTO TWO AND LOCATE THE',/,' FIRE SOURCE IN THE',
     .     ' ONE SHORTER THAN 30 FT.',/)
230  FORMAT (//,' * ATTN * FIRE BRANCH ',I5,' NOT FOUND, CHANGE ',
     .     'IGNORED.',/)
240  FORMAT (//,' * ATTN * VALUE OF JDP EXCEEDED CAPACITY, INPUT ',
     .     'IGNORED.',/)
250  FORMAT (//,' * ATTN * JUNCTION',I6,' NOT FOUND, CHANGE IGNORED.')
260  FORMAT (//,' * ERR * ERROR READING INPUT DATA.',/)
270  FORMAT (/,' * ATTN * NEGATIVE RESISTANCE INPUT FOR AIRWAY ',I5,
     .     ' IGNORED.',/,' CURRENT NBR= ',I5,'  RCH= ',F9.3)
280  FORMAT (/,' * ATTN * CAPACITY OF FAN ARRAYS EXCEEDED. BRANCH ',I4,
     .     ' IS ASSUMED',/,' AS A FIXED-P FAN WITH PRESSURE EQUAL TO',
     .     F9.3,' IN.W.G.',/)
290  FORMAT (/,' * ATTN * CAPACITY OF HEAT SOURCE ARRAYS EXCEEDED.',/,
     .     ' DATA FOR HEAT',/,' SOURCE ',I4,' IN THE TIME TABLE ',
     .     'IGNORED.',/)
C
      END


C
C
      SUBROUTINE  CDJN (ISQC,MARKS,NSTOP)
C
C  ----------------------------------------------------------------
C
C     SUBROUTINE PURPOSES:
C     1) EVALUATION OF THERMAL STATES IN JUNCTIONS.
C     2) DEVELOPING A NEW DATA RECORD INTO AIRWAYS (A) WHOSE BEGINNING
C        JUNCTION HAS DRASTIC CONDITION CHANGES; OR (B) WHERE AIRFLOW
C        REVERSAL HAPPENED; OR (C) WHERE A FIRE SOURCE EXISTS; OR (D)
```

```
C        WHICH GOT DETAILED RECORDS IN PREVIOUS INTERVALS ALREADY.
C
C    ---------------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
C  ** UPDATING JUNCTION CONDITIONS.
C
      CP=0.24
      DO 60 I=1,NJ
        K=JLR(I)
C
C  ** DEVELOPING DATA POINTS INTO THE DOWN-STREAM AIRWAYS.
C
        JY=1
        IF (I.GT.1) JY=LOUT(I-1)+1
        JZ=LOUT(I)
        DO 50 J=JY,JZ
          NW=NGOUT(J)
          CO2=0.0
          H2O=0.0
          WR=0.0
C
C     NWTYP(I)=10 INDICATES THAT AIRWAY NO(I) CONTAINS A FIRE SOURCE.
C     NSAC(I,2): NUMBER OF CONTROL VOLUMES IN AIRWAY NO(I).
C     IBTN(I) STORES THE ADDRESS OF DATA RECORDS FOR AIRWAY NO(I) IN
C            ARRAY DPPA, DPPB ETC.
C
          IF (A(NW).GE.100000.0) GO TO 50
C                  ********
          IF (MARKS.NE.0.AND.NWTYP(NW).NE.10) THEN
            IF (JCH(K).EQ.0) GO TO 50
C
C       WHEN THE NEWLY OBTAINED DATA NEED TO BE UPDATED AGAIN IN THE
C     PRESENT TIME INTERVAL, THE DATA OBTAINED IN THE PREVIOUS SEARCH
C     IN THE INTERVAL ARE ABANDONED TO AVOID DUPLICATED DATA
C     RECORDS.
C
            IF (IBTN(NW).GT.0) THEN
              IZ=IBTN(NW)
              NSAC(NW,2)=NSACB(NW,2)
              DO 5 IX=1,10
                DO 5 IY=1,5
                  DPPA(IZ,IX,IY)=DPPB(IZ,IX,IY)
5             CONTINUE
            ENDIF
          ELSE IF (NWTYP(NW).NE.10) THEN
C
C     NON-QUALIFIED AIRWAYS GET NO DETAILED DATA RECORDS AND ARE IGNORED
C
            IF (JDP.GT.0) THEN
              DO 7 IX=1,JDP
                IF (NO(NW).EQ.JDPP(IX)) GO TO 8
7             CONTINUE
            ENDIF
            IF (JCH(K).EQ.0.AND.IBTN(NW).EQ.0) GO TO 50
```

```
      ENDIF
C
C ** CHECK TO SEE IF A FIRE SOURCE EXISTS IN THE AIRWAY.
C
8       DO 10 L=1,INFLOW
          IF (NCENT(L).EQ.NO(NW)) GO TO 20
10      CONTINUE
C
C   NO FIRE SOURCE CONTAINED IN THE AIRWAY.
C
      CONTAM=0.0
      CONTQ=0.0
      HEATAD=0.0
      TFS=0.0
      GO TO 30
C
C   HAVING A FIRE SOURCE WITH USER-SPECIFIED HEAT GENERATION RATE.
C
20      NM=1
      FACT=1.0
      IF (QCENT(L).GT.10.0) THEN
        Q1=QQ2(NW)
        IF (Q1.LT.QCENT(L)) Q1=Q1+QCENT(L)
        FACT=1.0+(QQ2(NW)-QCENT(L))/Q1
      ENDIF
      IF (ABS(CONT(L)).GT.1.E-3) THEN
        CONTAM=CONT(L)*CONC(L)/100.0
        CONTQ=CONT(L)
      ELSE
        CONTAM=0.0
        CONTQ=0.0
      ENDIF
      O2BEH(L)=(0.21-PROP(K)-CONTAM/QQ2(NW))*100.0
      HEATAD=HEAT(L)*FACT
C
C   OXYGIN RICH FIRE
C
      IF (O2MIN(L).GT.1.0.AND.HEAT(L).GT.(-1.E-5)) THEN
        O2SCL=O2MIN(L)+(21.0-O2MIN(L))*(FACT-1.0)
        TACCA=TACC+DELTAT
        IF (TACCA.GT.TPR(L)*60.0) TACCA=TPR(L)*60.0
        PROPA=PROP(K)
        IF (NHM.GT.0) PROPA=PROP(K)*(CN(L)*32.0+HN(L)*8.0)/
     .                 (CN(L)*32.0)
        O2CONS=((0.21-PROPA-O2SCL/100.0)*TACCA/
     .         (TPR(L)*60.0))*QQ2(NW)
        IF (O2CONS.LT.0.0) O2CONS=0.0
        O2BEH(L)=(0.21-PROPA-O2CONS/QQ2(NW))*100.0
        IF (NHM.GT.0) THEN
          CO2=O2CONS*CN(L)*32.0/(CN(L)*32.0+HN(L)*8.0)
          H2O=O2CONS*HN(L)*8.0/(CN(L)*32.0+HN(L)*8.0)
          WR=2.0*H2O*18.0/(28.9*QQ2(NW))
          HEATAD=O2CONS*437.0
          CONTQ=0.0
          CONTAM=CO2
        ELSE
```

```
        WR=0.0
        CONTQ=0.0
        HEATAD=O2CONS*437.0
      ENDIF
      CONTAM=PROPA*QQ2(NW)+O2CONS
    ENDIF
C
C   FUEL RICH FIRE, THE MODELS OF THE ABOVE THREE TYPES OVERRIDE
C   THE ONES IN FRONT.
C
    IF (SMPO2(L).GT.1.E-3.AND.HEAT(L).GT.(-1.E-5)) THEN
      O2LT=0.21-0.21*(TACC+DELTAT)/(TPR(L)*60.0)
      IF (O2LT.LT.0.0) O2LT=0.0
      PROPA=PROP(K)
      IF (NHM.GT.0) PROPA=PROP(K)*(CN(L)*32.0+HN(L)*8.0)/
   .                  (CN(L)*32.0)
      O2CONS=(0.21-O2LT-PROPA)*QQ2(NW)
      IF (O2CONS.LT.0.0) O2CONS=0.0
      CONTAM=(0.21-O2LT)*QQ2(NW)
      IF ((0.21-O2LT).LT.PROPA) CONTAM=PROPA*QQ2(NW)
      O2BEH(L)=O2LT*100.0
      IF (FACT.LE.1.0) THEN
        HTSCL=HTPO2(L)
      ELSE
        HTSCL=HTPO2(L)/FACT
      ENDIF
      IF (NHM.GT.0) THEN
        CO2=CONTAM*CN(L)*32.0/(CN(L)*32.0+HN(L)*8.0)
        H2O=CONTAM*HN(L)*8.0/(CN(L)*32.0+HN(L)*8.0)
        WR=2.0*H2O*18.0/(28.98*QQ2(NW))
        HEATAD=O2CONS*HTSCL
        CONTQ=0.0
        CONTAM=CO2
      ELSE
        WR=0.0
        CONTQ=0.0
        HEATAD=O2CONS*HTSCL
      ENDIF
    ENDIF
    GFS(L)=PRH2O(K)+WR
C
C ** EVALUATION OF AIR TEMP. TJS(I) AT THE BEGINNING OF AIRWAY NO(I).
C
30   TK=T(K)
    IF (NWTYP(NW).EQ.10) TK=TAS(K)
    VART=(9900.0+TK)**2+2.0*HEATAD/(QQ2(NW)*2.4E-5*DR)
    IF (VART.LT.9.4E7) THEN
      WRITE (8,110) NO(I),Q(I)
      NSTOP=1
      RETURN
    ENDIF
    TJS(NW)=-9900.0+SQRT(VART)
    IF (TJS(I).LT.(-200.0).OR.TJS(I).GT.5000.0) THEN
      WRITE (8,100) NO(I),Q(I),TJS(I)
      NSTOP=1
      RETURN
```

```
        ENDIF
C         HTAD(L)=(TJS(NW)-T(K))*0.24*QQ2(NW)*DR
        IF (NWTYP(NW).EQ.10) THEN
          TRD(NW)=TJS(NW)
          RDPROP(NW)=PROP(K)*(QQ2(NW)-CONTQ)/QQ2(NW)+CONTAM/QQ2(NW)
          RDCH4(NW)=PRCH4(K)
          HTAD(L)=(TJS(NW)-T(K))*0.24*QQ2(NW)*DR
C          ****************************************
          GO TO 50
        ENDIF
C
C ** CHECK AND PREPARATION OF DATA STORAGE ARRAYS.
C
        NTAUXC=0
        IF (NSAC(NW,1).EQ.0) THEN
C
C   NSAC(I,1)=1 INDICATES THAT THE FUME FRONT IS WITHIN AIRWAY NO(I).
C
          NTAUXC=1
          NSAC(NW,1)=1
        ENDIF
        NSAC(NW,2)=NSAC(NW,2)+1
        NS=NSAC(NW,2)
        IF (IBTN(NW).EQ.0) THEN
          ISQC=ISQC+1
          IF (ISQC.LT.NMZ) THEN
            IBTN(NW)=ISQC
C
C     NSEQE(I): A BRIDGE ARRAY INDICATING THE LOCATION OF DATA FOR
C    WET CASE TEMP. CORR. CALCULATION FOR AIRWAY NO(I).
C
            IF (ISQC.EQ.1) NSEQE(ISQC)=0
            NSEQE(ISQC+1)=NSEQE(ISQC)+NSP(NW)
          ELSE
C
C     WHEN THE CAPACITIES OF DATA RECORDING ARRAYS ARE SATURATED, LATE
C    REQUEST WILL BE TURNED DOWN AND A MESSAGE WILL BE GIVEN.
C
            NSAC(NW,1)=0
            NSAC(NW,2)=0
            IF (IAC.GT.0) THEN
              DO 35 I1=1,IAC
                IF (IOMIT(I1).EQ.NW) GO TO 36
35            CONTINUE
            ENDIF
            IF (IAC.GE.IMX) GO TO 50
            IAC=IAC+1
    %%[ PrinterError: Out Of Paper ]%%¼¼          I1=IAC
36           IOMIT(I1)=NW
            ROMIT(I1,1)=T(K)
            ROMIT(I1,2)=PROP(K)
            ROMIT(I1,3)=PRCH4(K)
            GO TO 50
          ENDIF
        ENDIF
        NQ=IBTN(NW)
```

```
      NN=0
      IF (DCOAGE(NW).LT.0.0) THEN
        TMST=TACC+DELTAT
        CALL KALPHA (TMST,TM,FX,CP,NW,1)
        COAGE=BI(NW)-FX*BI(NW)*BI(NW)/(0.375+BI(NW))
        DCOAGE(NW)=COAGE
        XNEW(NW)=HK(NW)*O(NW)*O(NW)*COAGE/(120.*DR*QQ2(NW)*A(NW))
      ENDIF
C
C ** ADVANCING DATA POINTS.
C
      IF (NHM.EQ.0) THEN
C
C   PURELY DRY CASE IS ASSUMED WHEN NHM=0.
C
        TS=DELTAT/60.0
        IF (TLEFT(K).GE.0.0) TS=TLEFT(K)+0.00001
        DPPA(NQ,NS,5)=DR*QQ2(NW)*TS
        IF (NS.GT.1.AND.DELTAT/60.0.GT.TS) THEN
          DPPA(NQ,NS-1,5)=DPPB(NQ,NS-1,5)+DR*QQ2(NW)*
     .               (DELTAT/60.0-TS)
        ENDIF
        TAUXA(NQ,NS)=TJS(NW)
        CALL COLUMN (TJS(NW),SEG,NW,NQ,NS)
        FE=SEG/LA(NW)
        FA=XNEW(NW)*SEG/CP
        IF (FA.LT.25.0) THEN
          FAS=EXP(-FA)
        ELSE
          FAS=0.0
        ENDIF
C
C   AIR TEMP. OF THE C.V. AT ITS NEWLY ARRIVED POSITION IS ESTIMATED.
C
        DPPA(NQ,NS,2)=TROCK(NW)+(TJS(NW)-TROCK(NW))*FAS-
     .             (DZRD(NW)*FE/(2.0*778.26*CP))*(1.0+FAS)
      ELSE
C
C ** WET CASE ASSUMED.
C
        TS=DELTAT/60.0
        IF (TLEFT(K).GE.0.0) TS=TLEFT(K)+0.00001
        DPPA(NQ,NS,5)=DR*QQ2(NW)*TS
        IF ((NS-1).GT.0.AND.(DELTAT/60.0-TS).GT.0.0) THEN
          DPPA(NQ,NS-1,5)=DPPB(NQ,NS-1,5)+DR*QQ2(NW)*
     .               (DELTAT/60.0-TS)
        ENDIF
        HC=BI(NW)*HK(NW)*O(NW)/(2.0*A(NW))
        HLW=1050.0
        DRR=DR
C
C   TB: AIR TEMP. BEFORE A DATA POINT ADVANCED. F.
C   TQ: AIR TEMP. AFTER THE DATA POINT ADVANCED. F.
C   TTSS: MEAN AIR TEMP. IN THE SEGMENT COVERED BY THE DATA POINT
C       WITHIN THE PRESENT TIME INTERVAL.
C   TXM: MEAN WALL TEMP. IN THE SEGMENT. F.
```

```
C
      JLE=NSP(NW)
      TAUXA(NQ,NS)=TJS(NW)
      TB=TJS(NW)
      TQ=TJS(NW)-5.0
      TTSS=0.7*TB
      IF (TB.LT.200.0) TTSS=0.9*TB
      SEG=QQ2(NW)*TS*(TTSS+460)/(A(NW)*(TR+460))
25     TXM=DCOAGE(NW)*(TROCK(NW)-TTSS)/BI(NW)+TTSS
      CALL MTCOEF (TXM,TTSS,NW,BETA)
      FE=SEG/LA(NW)
C
C   WMTR: MASS TRANSFER BETWEEN WALL FILM AND BULK FLOW AT
C       STATIONS IN THE PRESENT TIME INTERVAL, LBM/FT2.
C   GFF: VARIATION IN HUMIDITY RATIO CAUSED BY VAPOR CONDENSATION
C      AS MIST IN THE SEGMENT IN THE PRESENT INTERVAL.
C
      WC=0.0
      WD=0.0
      NC=1
      DO 45 JL=JLE,1,-1
        SDL=LA(NW)*(JL-1)/JLE
        IF (SEG.GE.SDL) THEN
          WC=WC+WMTR(NQ,JL)
          WD=WD+GFF(NQ,JL)
          NC=NC+1
        ENDIF
45     CONTINUE
      GADD=WC/NC
      GF=WD/NC
      DRR=(DR+DR*GADD*18/28.98)*(TR+460)/((TTSS+460)*(1+GADD))
      GDP=GADD*DRR
      IF (GDP.LT.0.0) GDP=0.0
      CNT(NW)=HC*WTCORR(NQ,1)+HK(NW)*DCOAGE(NW)*TROCK(NW)
     .     *O(NW)/(2.0*A(NW))+BETA*GDP*CP*(TXM-TQ)-
     .     HLW*QQ2(NW)*DR*GF/(O(NW)*SEG)
      FAW(NW)=HK(NW)*DCOAGE(NW)*O(NW)*O(NW)/(2*QQ2(NW)*A(NW)*
     .     60.0*CP*DR)
      SEG1=SEG
      CALL COLUMN (TB,SEG,NW,NQ,NS)
      EL=FAW(NW)*SEG
      IF (EL.LT.25.0) THEN
        EXPN(NW)=EXP(-EL)
      ELSE
        EXPN(NW)=0.0
      ENDIF
C
C   AIR TEMP. OF THE DATA POINT AT ITS NEWLY ARRIVED POSITION.
C
      DPPA(NQ,NS,2)=2*A(NW)*(CNT(NW)-(CNT(NW)-HK(NW)*DCOAGE(NW)
     .     *O(NW)*TB/(2*A(NW)))*EXPN(NW))/(HK(NW)*DCOAGE(NW)
     .     *O(NW))-DZRD(NW)*SEG/(LA(NW)*778.26*CP)

C
C   ITERATION PROCESS IS CALLED FOR (REFER TO NOTES FOR DRY CASE).
C
```

```
          NN=NN+1
          IF (ABS(SEG1-SEG).GT.0.1) THEN
            TSS=TTSS
            B=2*A(NW)/(HK(NW)*DCOAGE(NW)*O(NW)*EL)
            E=B*(CNT(NW)-HK(NW)*DCOAGE(NW)*TB*O(NW)/(2*A(NW)))
     .        *(1.0-EXPN(NW))
            TTSS=2*A(NW)*CNT(NW)/(HK(NW)*DCOAGE(NW)*O(NW))-E-
     .          SEG*DZRD(NW)/(2*778.26*LA(NW)*CP)
            GADD=WMTR(NQ,1)*SEG*O(NW)*60.0/(QQ2(NW)*DR*TS)
            TQ=DPPA(NQ,NS,2)
C
C   GADD: CHANGE IN HUMIDITY RATIO DUE TO MASS TRANSFER BETWEEN THE
C         WALL FILM AND BULK FLOW.
C   GF: CHANGE OF HUMIDITY RATIO OF C.V. BY CONDENSATION AS FOG.
C   ** ALL IN LB OF VAPOR/LB OF DRY AIR.
C
            IF (NN.LE.20) GO TO 25
            WRITE (8,*) 'SUB. CDJN NN EXCEEDED.',SEG,SEG1,NW,NS
          ENDIF
        ENDIF
C
C   DATA RECORDED IN DPPA: (1) LOCATION, (2) TEMP. (3) CH4,
C                   (4) FUME, (5) AIR MASS IN C.V..
C
        FA=XNEW(NW)*SEG/CP
        FAS=0.0
        IF (FA.LT.25.0) FAS=EXP(-FA)
        FE=SEG/LA(NW)
        IF (NTAUXC.EQ.1) TAUXC(NQ)=TROCK(NW)+(TTJS(NW)-TROCK(NW))*
     .             FAS-(DZRD(NW)*FE/(2.0*778.26*CP))*(1.+FAS)
        DPPA(NQ,NS,1)=SEG
        DPPA(NQ,NS,3)=PROP(K)*(QQ2(NW)-CONTQ)/QQ2(NW)+CONTAM/QQ2(NW)
        DPPA(NQ,NS,4)=(PRCH4(K)*QQ2(NW)+CH4V(NW)*FE)/
     .            (QQ2(NW)+CH4V(NW)*FE)
50    CONTINUE
60  CONTINUE
C
    RETURN
100 FORMAT (/,' * ERR * ABNORMAL TEMPERATURE IN HEAT SOURCE ',I5,/,
     .    ' CURRENT AIRFLOW:',F8.0,'  TEMPERATURE: ',F8.1,/,
     .    ' "HEAT" AND/OR "QCENT" MUST BE ADJUSTED.')
110 FORMAT (/,' * ERR * ABNORMAL TEMPERATURE IN HEAT SOURCE ',I5,/,1X,
     .    'CURRENT AIRFLOW:',F8.0,'. ADJUST "HEAT" AND/OR "QCENT".')
    END


C
C
    SUBROUTINE CHECK1 (NSTOP,MAXNO)
C
C   --------------------------------------------------------
C
C     SUBROUTINE PURPOSE:
C     1) MAKE-UP OF THE MISSING DATA.
C     2) CHECK FOR CONSISTENCY OF INPUT DATA.
C
```

```
C     -------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
C  ** CALCULATION OF VENTILATION RESISTANCE.
C
      MAXNO=0
      DO 10 I=1,NB
        IF (NO(I).GT.MAXNO) MAXNO=NO(I)
        IF (NWTYP(I).EQ.0) THEN
          IF (R(I).LE.0) THEN
            IF (A(I).LE.0) THEN
              WRITE (8,130) NO(I)
              NSTOP=1
            ELSE
              R(I)=KF(I)*LA(I)*O(I)/(5.2*A(I)**3)*DR/0.075
            ENDIF
          ENDIF
          RSTD(I)=R(I)
        ELSE
          RSTD(I)=R(I)
        ENDIF
10    CONTINUE
C************************************************************************
      LF1=0
      DO 15 I=1,NFNUM
        DO 15 J=1,NB
          IF (NWTYP(J).EQ.1.AND.NO(J).EQ.NOF(I)) LF1=1
15    CONTINUE
      IF (LF1.EQ.0) THEN
        DO 16 J=1,NB
          IF(NWTYP(J).EQ.1.AND.R(J).GT.0.0) LF1=1
16      CONTINUE
      ENDIF
      IF (LF1.EQ.0)  WRITE (8,*) "* WARNING * NO FAN IN THE NETWORK. "
C************************************************************************
C
C  ** CHECK FOR DATA CONSISTENCY.
C
      MAXJ=0
      DO 30 I=1,NB
        IF (IABS(NWTYP(I)).GT.1) THEN
          WRITE (8,160) NWTYP(I),NO(I)
          NSTOP=1
        ENDIF
        IF (JS(I).EQ.JF(I)) THEN
          WRITE (8,100) NO(I)
          NSTOP=1
        ENDIF
        NREV(I)=0
        N1=0
        N2=0
        DO 20 J=1,NJ
          IF (JNO(J).GT.MAXJ.AND.I.EQ.1) MAXJ=JNO(J)
          IF (JS(I).EQ.JNO(J)) N1=N1+1
          IF (JF(I).EQ.JNO(J)) N2=N2+1
```

```
20      CONTINUE
       IF (N1.EQ.0) THEN
         WRITE (8,105) NO(I),JS(I)
         NSTOP=1
       ELSE IF (N1.GE.2) THEN
         WRITE (8,110) NO(I),JF(I)
         NSTOP=1
       ENDIF
       IF (N2.EQ.0) THEN
         WRITE (8,105) NO(I),JS(I)
         NSTOP=1
       ELSE IF (N2.GE.2) THEN
         WRITE (8,110) NO(I),JF(I)
         NSTOP=1
       ENDIF
30    CONTINUE
     DO 50 J=1,NJ
       IF (T(J).LT.(-100.0).OR.T(J).GT.1500.0) T(J)=TR
       IF (CH4C(J).LT.0.0.OR.CH4C(J).GT.75.0) CH4C(J)=0.0
      J1=0
      DO 40 I=1,NB
        IF (JS(I).EQ.JNO(J).OR.JF(I).EQ.JNO(J)) THEN
          J1=J1+1
        ENDIF
40     CONTINUE
      IF (J1.EQ.0) THEN
         WRITE (8,115) JNO(J)
         NSTOP=1
       ENDIF
50    CONTINUE
     DO 70 I=1,NB
       J1=0
       J2=0
       J3=0
       DO 60 J=1,NB
         IF (NO(I).EQ.NO(J)) J1=J1+1
         IF (JS(I).EQ.JS(J)) J2=J2+1
         IF (JS(I).EQ.JF(J)) J2=J2+1
         IF (JF(I).EQ.JS(J)) J3=J3+1
         IF (JF(I).EQ.JF(J)) J3=J3+1
60      CONTINUE
       IF (J1.GT.1) THEN
         WRITE (8,140) NO(I)
         NSTOP=1
       ENDIF
       IF (J2.LE.1.OR.J3.LE.1) THEN
         IF (J2.LE.1) WRITE (8,170) NO(I),JS(I)
         IF (J3.LE.1) WRITE (8,170) NO(I),JF(I)
         NSTOP=1
       ENDIF
70    CONTINUE
     DO 90 I=1,NJ
       J2=0
       DO 80 J=1,NJ
         IF (JNO(I).EQ.JNO(J)) J2=J2+1
80      CONTINUE
```

```
      IF (J2.GT.1) THEN
        WRITE (8,150) JNO(I)
        NSTOP=1
      ENDIF
90    CONTINUE
C
100   FORMAT (/,1X,'* ERR *  AIRWAY',I5,' IS ISOLATED FROM ',
     .       'THE NETWORK.')
105   FORMAT (/,1X,'* ERR *  AIRWAY',I5,'  IS ISOLATED FROM ',
     .       'THE NETWORK',/,' FOR THE UNCONNECTED END',I5)
110   FORMAT (/,1X,'* ERR *  DUBLICATED JUNCTION NUMBER',I5)
115   FORMAT (/,1X,'* ERR *  JUNCTION',I5,'  IS ISOLATED FROM ',
     .       'THE NETWORK.')
130   FORMAT (/,1X,'* ERR *  NO RESISTANCE AND DIMENSION WERE',
     .       ' STATED FOR AIRWAY',I5)
140   FORMAT (/,1X,'* ERR *  DUPLICATED AIRWAY ID NUMBER',I5)
150   FORMAT (/,1X,'* ERR *  DUPLICATED JUNCTION ID NUMBER',I5)
160   FORMAT (/,1X,'* ERR *  INVALID AIRWAY TYPE',I4,' FOR AIRWAY',
     .       I6,' DETECTED.')
170   FORMAT (/,1X,'* ERR *  AIRWAY',I6,' IS ISOLATED FROM THE ',
     .       'NETWORK DUE TO DEADEND',I6)
C
      RETURN
      END


C
C
      SUBROUTINE CHSFIT (NSTOP1)
C
C    ----------------------------------------------------------------
C
C      SUBROUTINE PURPOSE:
C      1) SELECT SUITABLE FAN CURVE FITTING METHODS FOR EACH FAN.
C
C    ----------------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
      DIMENSION HFN(IMX), QFN(IMX)
C
      DO 75 L=1,NFNUM
      IF (NSWT(L).LT.3) GO TO 75
        DFQ1=ABS(QF(L,2)-QF(L,1))
        DFP1=ABS(PF(L,2)-PF(L,1))
        DO 55 I=2,MPTS(L)-1
          DFQ2=ABS(QF(L,I)-QF(L,I-1))
          DFP2=ABS(PF(L,I)-PF(L,I-1))
          IF (DFQ1.LT.DFQ2) THEN
            DFQ1=DFQ2
          ELSEIF (DFP1.LT.DFP2) THEN
            DFP1=DFP2
          ENDIF
55      CONTINUE
        DQMAX=DFQ1
```

```
      DPMAX=DFP1
      DO 60 I=2,MPTS(L)-1
        DFQ2=ABS(QF(L,I+1)-QF(L,I))
        DFP2=ABS(PF(L,I+1)-PF(L,I))
        IF (DFQ1.GT.DFQ2) THEN
          DFQ1=DFQ2
        ENDIF
        IF (DFP1.GT.DFP2) THEN
          DFP1=DFP2
        ENDIF
60     CONTINUE
      DQMIN=DFQ1
      DPMIN=DFP1
      DIVQ=DQMAX/DQMIN
      DIVP=DPMAX/DPMIN
      IF (DIVQ.GT.4.AND.DIVP.GT.4) THEN
        NSWT(L)=1
        WRITE (8,975) L
        GO TO 75
      ENDIF
      INDEX=MPTS(L)
      IF (INDEX.LT.3) THEN
        WRITE (*,980) INDEX
         NSTOP1=1
        RETURN
      ELSE
        THSUM1=0.
        THSUM2=0.
        DO 444 K3=1,INDEX-1
          DQZZ=(QF(L,K3+1)-QF(L,K3))
          QFM=QF(L,K3)+DQZZ/2
          CALL SPLINE (L,QFM,HFM,TR,0)
          DIFH1=HFM-PF(L,K3)
          DIFH2=HFM-PF(L,K3+1)
          IF (PF(L,K3).GT.PF(L,K3+1)) THEN
            IF (DIFH1.GT.0) THEN
              THSUM1=THSUM1+DIFH1
            ELSEIF (DIFH2.LT.0) THEN
              THSUM1=THSUM1+ABS(DIFH2)
            ENDIF
          ELSE
            IF (DIFH2.GT.0) THEN
              THSUM1=THSUM1+DIFH2
            ELSEIF (DIFH1.LT.0) THEN
              THSUM1=THSUM1+ABS(DIFH1)
            ENDIF
          ENDIF
          CALL LSFAN (L,QFM,HFM,TR,0,0)
          DIFH3=HFM-PF(L,K3)
          DIFH4=HFM-PF(L,K3+1)
          IF (PF(L,K3).GT.PF(L,K3+1)) THEN
            IF (DIFH3.GT.0) THEN
              THSUM2=THSUM2+DIFH3
            ELSEIF (DIFH4.LT.0) THEN
              THSUM2=THSUM2+ABS(DIFH4)
            ENDIF
```

```
          ELSE
            IF (DIFH4.GT.0) THEN
              THSUM2=THSUM2+DIFH4
            ELSEIF (DIFH3.LT.0) THEN
              THSUM2=THSUM2+ABS(DIFH3)
            ENDIF
          ENDIF
444     CONTINUE
      ENDIF
      IF (ABS(THSUM1).GT.1.2*ABS(THSUM2)) THEN
        NSWT(L)=1
        WRITE (*,1000) L
      ELSE
        NSWT(L)=2
        WRITE (*,1005) L
      ENDIF
      IF (NSWT(L).EQ.3) GO TO 75
      WRITE (*,1010) NOF(L)
      WRITE (*,1020)
      LL=MPTS(L)
      WRITE (*,1030) (QF(L,I)/1000.0,PF(L,I),I=1,LL)
      WRITE (*,1045) NOF(L)
        WRITE (*,1086)
        WRITE (*,1085)
        DO 441 K1=1,INDEX-1
          QZZ=(QF(L,K1+1)-QF(L,K1))/5.0
          DO 42 K2=1,5
            QZZ1=QF(L,K1)+(K2-1)*QZZ
            NFCWZ=NFCW(L)
            CALL SPLINE (L,QZZ1,HZZ,TR,1)
            NFCW(L)=NFCWZ
            HFN(K2)=HZZ
            QFN(K2)=QZZ1
42        CONTINUE
          WRITE (*,1090) (QFN(K4)/1000.,HFN(K4),K4=1,5)
441     CONTINUE
        WRITE (*,1087)
        WRITE (*,1085)
        DO 445 K1=1,INDEX-1
          QZZ=(QF(L,K1+1)-QF(L,K1))/5.0
          DO 443 K2=1,5
            QZZ1=QF(L,K1)+(K2-1)*QZZ
            NFCWZ=NFCW(L)
            CALL LSFAN (L,QZZ1,HZZ,TR,1,0)
            NFCW(L)=NFCWZ
            HFN(K2)=HZZ
            QFN(K2)=QZZ1
443       CONTINUE
          WRITE (*,1090) (QFN(K4)/1000.,HFN(K4),K4=1,5)
445     CONTINUE
      WRITE (*,1100)
      READ (*,*) NSWT(L)
75    CONTINUE
C
     RETURN
C
```

```
975  FORMAT (//,2X,'SOME DATA POINTS OF FAN',I5,2X,'CURVE ARE TOO CLOSE
    .',/,'SO THE LEAST SQUARE METHOD IS DESIRED')
980  FORMAT (//,2X,'THE DATA POINTS OF FAN CHARACTERISTIC ARE',I5,/,
    .'AT LEAST 3 DATA POINTS ARE DESIRED',/)
1000 FORMAT (//,2X,'RECOMMEND LEAST SQUARE METHOD FOR FAN',I5,/,
    .'SURVE FITTING',/)
1005 FORMAT (//,2X,'RECOMMEND SPLINE METHOD FOR FAN',I5,/,
    .'SURVE FITTING')
1010 FORMAT (//,'THESE CHARACTERISTICS WERE STORED FOR FAN IN AIRWAY'
    .,I5,/)
1020 FORMAT (//,5('  Q*1000   PF '),/)
1030 FORMAT (5(F8.1,F6.2)/,5(F8.1,F6.2),/)
1045 FORMAT (/,T5,'FAN',I5,/)
1085 FORMAT (/,2X,'INTERPOLATION:',/,5('  Q*1000   PF '),/)
1086 FORMAT (//,2X,'APPLYING SPLINE METHOD')
1087 FORMAT (//,2X,'APPLYING LEAST SQUARE METHOD')
1090 FORMAT (5(F8.1,F6.2))
1100 FORMAT (//,2X,'INPUT SELECT FAN CURVE FITTING METHOD',/,
    .'1-- FOR LEAST SQUARE METHOD, 2--FOR SPLINE METHOD')
C
     END
C

C
C
     SUBROUTINE  DISP (MARKX,ISQC,IBB)
C
C    --------------------------------------------------------------
C
C      SUBROUTINE PURPOSES:
C      1) DISCARDING THE TENTATIVELY OBTAINED DATA IN THE PREVIOUS
C         ITERATION TO AVOID UNNECESSARY RECORDS.
C      2) RESTORATION OF INITIAL CONDITIONS EXCEPT FOR AIRFLOW RATES.
C
C    --------------------------------------------------------------
C
     INCLUDE 'CMMNW.DAT'
C
C ** ABANDON THE DATA FOR AIRWAYS WHICH GOT THEIR DETAILED DATA RECORDS
C    IN THE PREVIOUS ITERATION.
C
     DO 40 LRW=1,NB
       IF (IBTN(LRW).GT.0.AND.NSACB(LRW,1).EQ.0) THEN
         ISQCX=IBTN(LRW)
         DO 5 IX=ISQCX,ISQC
           DO 5 IY=1,NB
             IF (IBTN(IY).EQ.IX) NSEQE(IX)=NSEQE(IX+1)-NSP(IY)
             IF (NSEQE(IX).LT.0) NSEQE(IX)=0
5          CONTINUE
         DO 10 IV=1,NB
           IF (IBTN(IV).GE.ISQCX) THEN
             IBTN(IV)=IBTN(IV)-1
           ENDIF
10         CONTINUE
         DO 20 IV=ISQCX,ISQC
           TAUXC(IV)=TAUXC(IV+1)
```

```
            DO 20 IU=1,10
              IF (NHM.GT.0) THEN
                STV(IV,IU)=STV(IV+1,IU)
                HABS(IV,IU)=HABS(IV+1,IU)
                TDA(IV,IU)=TDA(IV+1,IU)
                WMTR(IV,IU)=WMTR(IV+1,IU)
              ENDIF
              TAUXA(IV,IU)=TAUXA(IV+1,IU)
              DO 20 IW=1,5
                DPPA(IV,IU,IW)=DPPA(IV+1,IU,IW)
20          CONTINUE
          IBTN(LRW)=0
          NSAC(LRW,1)=0
          NSAC(LRW,2)=0
          ISQC=ISQC-1
        ENDIF
40    CONTINUE
C
C  ** RESTORATION OF INITIAL CONDITIONS FOR THE PRESENT TIME INTERVAL
C     EXCEPT FOR AIRFLOW DISTRIBUTION.
C
      DO 60 I=1,NB
        TJS(I)=TTJS(I)
        TRD(I)=TTRD(I)
        RDPROP(I)=RDOP(I)
        RDCH4(I)=RCH4(I)
        RDH2O(I)=RH2O(I)
C
C     Q(I): AIRFLOW RATE IN AIRWAY NO(I) AT THE END OF THE PRESENT
C           INTERVAL.
C     QTP(I): INITIAL AIRFLOW RATE IN AIRWAY NO(I), CONSTANT WITHIN
C             EACH TIME INCREMENT.
C     QQ(I): TIME AVERAGED AIRFLOW RATE IN AIRWAY NO(I).
C
        QQA=0.55*Q(I)+0.45*QTP(I)
        QQ(I)=0.85*QQA+0.15*QQ(I)
        QQA2=0.55*ABS(Q(I))+0.45*ABS(QTP(I))
C       ***************************************
        QQ2(I)=0.85*ABS(QQA2)+0.15*ABS(QQ(I))
C       ***************************************
        NSAC(I,1)=NSACB(I,1)
        NSAC(I,2)=NSACB(I,2)
        IQ=IBTN(I)
        IF (IQ.GT.0) THEN
          TAUXC(IQ)=TAUXD(IQ)
          DO 50 J=1,10
            TAUXA(IQ,J)=TAUXB(IQ,J)
            DO 50 JJ=1,5
              DPPA(IQ,J,JJ)=DPPB(IQ,J,JJ)
50        CONTINUE
        ENDIF
60    CONTINUE
      DO 65 I=1,NJ
        T(I)=TAS(I)
        PROP(I)=PROPS(I)
        PRCH4(I)=PRCH4S(I)
```

```
      PRH2O(I)=PRH2OS(I)
65   CONTINUE
     DO 70 I=1,IMX
       HTAD(I)=0.0
70   CONTINUE
     IBB=0
     CALL BASE (NSTOP)
     CALL MSLIST
C
     RETURN
     END

C
C
     SUBROUTINE  DTTR (MARKL,MARKP)
C
C    ----------------------------------------------------------------
C
C      SUBROUTINE PURPOSE:
C      REARRANGING DETAILED DATA RECORDS FOR THE AIRWAY WHOSE AIRFLOW
C    REVERSED.
C
C    ----------------------------------------------------------------
C
     INCLUDE 'CMMNW.DAT'
C
     DO 90 I=1,NB
       DO 1 J=1,NJ
         IF (JNO(J).EQ.JS(I)) JU=J
         IF (JNO(J).EQ.JF(I)) JW=J
1      CONTINUE
C
C     MARKP=0: A THERMALLY BALANCED SOLUTION HAS NOT BEEN REACHED YET
C             WITHIN A TIME INTERVAL.
C     MARKP=1: THE CALCULATION OF THE PRESENT TIME INTERVAL IS DONE.
C     Q(I): AIRFLOW RATE IN AIRWAY NO(I) AT THE END OF THE PRESENT
C           INTERVAL.
C     QQ(I): TIME AVERAGED AIRFLOW RATE.
C
       IF (MARKP.EQ.0) QRS=QQ(I)
       IF (MARKP.EQ.1) QRS=Q(I)
       IF (QRS.LT.0.0) THEN
         MARKL=1
         IQ=IBTN(I)
         IF (IQ.GT.0) THEN
           IF (MARKP.EQ.1) THEN
             IS=NSAC(I,2)
             IF (IS.GT.0) THEN
               DO 2 J=1,IS
                 DO 2 K=1,5
                   GBTN(J,K)=DPPA(IQ,J,K)
2              CONTINUE
               DO 6 J=1,IS
                 JP=IS-J+1
                 DO 4 K=1,5
                   DPPA(IQ,JP,K)=GBTN(J,K)
```

```
4           CONTINUE
            DPPA(IQ,JP,1)=LA(I)-DPPA(IQ,JP,1)
            IF (J.EQ.1) THEN
              DPPA(IQ,JP,2)=TAUXC(IQ)
            ELSE
              DPPA(IQ,JP,2)=TAUXA(IQ,J-1)
            ENDIF
6         CONTINUE
          DO 7 J=1,IS
            JP=IS-J+1
            IF (J.EQ.1) THEN
              TAUXA(IQ,JP)=TRD(I)
            ELSE
              TAUXA(IQ,JP)=GBTN(J-1,2)
              IF (J.EQ.IS) TAUXC(IQ)=GBTN(J,2)
            ENDIF
7         CONTINUE
          FMASS1=DPPA(IQ,1,5)
          IF (IS.GE.2) THEN
            DO 8 J=1,IS-1
              DPPA(IQ,J,5)=DPPA(IQ,J+1,5)
8           CONTINUE
          ENDIF
          IF (FMASS(IQ).LT.0.0) THEN
            FKKX=535.0/(460.0+(DPPA(IQ,IS,2)+TRD(I))/2.0)
            FMASS(IQ)=A(I)*DPPA(IQ,IS,1)*0.075*FKKX
          ENDIF
          DPPA(IQ,IS,5)=FMASS(IQ)
          FMASS(IQ)=FMASS1
          RDPROP(I)=DPPA(IQ,1,3)
          RDCH4(I)=DPPA(IQ,1,4)
          RDOP(I)=DPPA(IQ,1,3)
          RCH4(I)=DPPA(IQ,1,4)
        ELSE
          IF (NWTYP(I).NE.10) THEN
            RDPROP(I)=PROP(JU)
            RDCH4(I)=PRCH4(JU)
            RDH2O(I)=PRH2O(JU)
            RDOP(I)=PROP(JU)
            RCH4(I)=PRCH4(JU)
            RH2O(I)=PRH2O(JU)
          ENDIF
        ENDIF
      ELSE
        IS=NSACB(I,2)
        IF (IS.GT.0) THEN
          DO 10 J=1,IS
            DO 10 K=1,5
              GBTN(J,K)=DPPB(IQ,J,K)
10        CONTINUE
          DO 30 J=1,IS
            JP=IS-J+1
            DO 20 K=1,5
              DPPB(IQ,JP,K)=GBTN(J,K)
20          CONTINUE
            DPPB(IQ,JP,1)=LA(I)-DPPB(IQ,JP,1)
```

```
          IF (J.EQ.1) THEN
            DPPB(IQ,JP,2)=TAUXD(IQ)
          ELSE
            DPPB(IQ,JP,2)=TAUXB(IQ,J-1)
          ENDIF
30       CONTINUE
        DO 32 J=1,IS
          JP=IS-J+1
          IF (J.EQ.1) THEN
            TAUXB(IQ,JP)=TTRD(I)
          ELSE
            TAUXB(IQ,JP)=GBTN(J-1,2)
            IF (J.EQ.IS) TAUXD(IQ)=GBTN(J,2)
          ENDIF
32       CONTINUE
        FMASS1=DPPB(IQ,1,5)
        IF (IS.GE.2) THEN
          DO 35 J=1,IS-1
            DPPB(IQ,J,5)=DPPB(IQ,J+1,5)
35         CONTINUE
        ENDIF
        IF (FMASS(IQ).LT.0.0) THEN
          FKKX=535.0/(460.0+(DPPB(IQ,IS,2)+TTRD(I))/2.0)
          FMASS(IQ)=A(I)*DPPB(IQ,IS,1)*0.075*FKKX
        ENDIF
        DPPB(IQ,IS,5)=FMASS(IQ)
        FMASS(IQ)=FMASS1
        DO 40 J=1,IS
          DPPA(IQ,J,5)=DPPB(IQ,J,5)
40       CONTINUE
        RDPROP(I)=DPPB(IQ,1,3)
        RDCH4(I)=DPPB(IQ,1,4)
        RDOP(I)=DPPB(IQ,1,3)
        RCH4(I)=DPPB(IQ,1,4)
      ELSE
        IF (NWTYP(I).NE.10) THEN
          RDPROP(I)=PROPS(JU)
          RDCH4(I)=PRCH4S(JU)
          RDH2O(I)=PRH2OS(JU)
          RDOP(I)=PROPS(JU)
          RCH4(I)=PRCH4S(JU)
          RH2O(I)=PRH2OS(JU)
        ENDIF
      ENDIF
    ENDIF
    IF (NHM.GT.0) THEN
      JLE=NSP(I)
      DO 50 J=1,JLE
        GBTN(J,1)=TDA(IQ,J)
        GBTN(J,2)=STV(IQ,J)
        GBTN(J,3)=WMTR(IQ,J)
        GBTN(J,4)=WTCORR(IQ,J)
        GBTN(J,5)=HABS(IQ,J)
50     CONTINUE
      DO 60 J=1,JLE
        JP=JLE-J+1
```

```
              TDA(IQ,J)=GBTN(JP,1)
              STV(IQ,J)=GBTN(JP,2)
              WMTR(IQ,J)=GBTN(JP,3)
              WTCORR(IQ,J)=GBTN(JP,4)
C       write (8,*) 'wtcorrdttr',iq,j,wtcorr(iq,j)
              HABS(IQ,J)=GBTN(JP,5)
60           CONTINUE
          ENDIF
        ENDIF
        TRDA=TRD(I)
        TTRDA=TTRD(I)
        TRD(I)=TJS(I)
        TTRD(I)=TTJS(I)
        TJS(I)=TRDA
        TTJS(I)=TTRDA
        JE=JS(I)
        JS(I)=JF(I)
        JF(I)=JE
        DZRD(I)=-DZRD(I)
        FRNVP(I)=-FRNVP(I)
        QTP(I)=-QTP(I)
        QQ(I)=-QQ(I)
        Q(I)=-Q(I)
C*****************************************************
        NUMCT=0
        DO 65 J=1,INFLOW
          IF (NO(I).EQ.NCENT(J)) THEN
            DO 64 JJ=1,NB
              IF (JF(JJ).EQ.JNO(JU)) THEN
                NUMCT=NUMCT+1
                SFUME=SFUME+RDPROP(JJ)
              ENDIF
64           CONTINUE
            PROP(JU)=SFUME/NUMCT
          ENDIF
65       CONTINUE
C*****************************************************
C
C    NREV(I): NUMBER OF CHANGES IN AIRFLOW DIRECTION (QQ) IN AIRWAY
C          NO(I) IN THE PRESENT TIME INTERVAL.
C    NNREV(I): NUMBER OF CHANGES IN AIRFLOW DIRECTION (Q) IN AIRWAY
C           NO(I) SINCE TIME ZERO.
C
        NREV(I)=NREV(I)+1
      ENDIF
      IF (MARKP.EQ.1.AND.Q(I)*QTP(I).LT.0.0) NNREV(I)=NNREV(I)+1
      IF (ABS(QQ(I)).LT.1.E-5) QQ(I)=1.E-5
      IF (ABS(Q(I)).LT.1.E-5) Q(I)=1.E-5
90   CONTINUE
C
C     WHEN THERE IS AIRFLOW REVERSAL HAPPENED (MARKL>0), NETWORK
C    SCHEME ARRAYS WILL BE RE-STRUCTURED.
C
     IF (MARKL.NE.0) THEN
       L=0
       M=0
```

```
      N=1
      DO 120 I=1,MAXJ
        K=L
        DO 100 J=1,NB
          IF (JS(J).EQ.I) THEN
            L=L+1
            NGOUT(L)=J
          ENDIF
100     CONTINUE
        LOUT(N)=L
        MM=M
        DO 110 J=1,NB
          IF (JF(J).EQ.I) THEN
            M=M+1
            NGIN(M)=J
          ENDIF
110     CONTINUE
        MIN(N)=M
        IF (MM.NE.M.OR.K.NE.L) THEN
          JNOL(N)=I
          N=N+1
        ENDIF
120   CONTINUE
      DO 140 I=1,NJ
        DO 130 J=1,NJ
          IF (JNOL(I).EQ.JNO(J)) THEN
            JLR(I)=J
            GO TO 140
          ENDIF
130     CONTINUE
140   CONTINUE
    ENDIF
C
C
    RETURN
    END


C
C
    SUBROUTINE  INPUT (JUMP1,NSTOP,MARKY,MAXNO)
C
C   ----------------------------------------------------------------
C
C     SUBROUTINE PURPOSE:
C     INPUT OF DATA.
C
C   ----------------------------------------------------------------
C
    INCLUDE 'CMMNW.DAT'
C
C
    IF (JUMP1.EQ.1) THEN
      IF (LMX.LT.1000) THEN
        WRITE (8,1120) LMX
        NSTOP=1
```

```
      ENDIF
C
C  ** INPUT OF DATA OF NETWORK STRUCTURE.
C
      READ (9,*,ERR=99) A1,A2,A3,A4,A5,TR,A6,A7,A8,TINC,SPAN,A9,DR,
     .           A11
      NB=A1+0.4
      NFNUM=A2+0.4
      INFLOW=A3+0.4
      NVPN=A4+0.4
      NETW=A5+0.4
      MADJ=A6+0.4
      ITN=A7+0.4
      NTEMP=A8+0.4
      IOUT=A9+0.4
      IF (A9.LT.0.0) IOUT=A9-0.4
      NHM=A11+0.4
      IF (IOUT.LE.(-2)) WRITE (8,1000) NB,NFNUM,INFLOW,NVPN,NETW,TR,
     .             MADJ,ITN,NTEMP,TINC,SPAN,IOUT,DR,NHM
      IF (NB.LT.0.OR.NFNUM.LT.0.OR.INFLOW.LT.0) THEN
        WRITE (8,1220)
        NSTOP=1
      ENDIF
      IF (TR.LT.(-100.0).OR.TR.GT.150) TR=75.0
      IF (ABS(TR).LT.0.01) TR=75.0
C      *****************************
      DR1=2116/(53.352*(TR+460.0))
      IF (DR.LE.0.03.OR.DR.GE.0.11) THEN
        DR=DR1
      ELSE IF (ABS(DR-DR1)/DR1.GT.0.5) THEN
        WRITE (8,1200)
      ENDIF
      IF (MADJ.LT.10.OR.MADJ.GT.50) MADJ=15
      IF (ITN.LT.10.OR.ITN.GT.50) ITN=15
      IF ((NB+INFLOW).GT.NMX.OR.NB.LE.1) THEN
        WRITE (8,210) NMX,NB,INFLOW
        NSTOP=1
      ENDIF
      IF (NVPN.GE.1) THEN
        NETW=1
        WRITE (8,270)
      ENDIF
      IF (SPAN.LT.0.1) SPAN=5.0
      IF (IOUT.LE.(-2).AND.NHM.LE.0) WRITE (8,1005)
      IF (IOUT.LE.(-2).AND.NHM.GE.1) WRITE (8,1010)
      DO 1 K=1,NB
        IF (NHM.GE.1) THEN
          READ (9,*,ERR=99) A1,A2,A3,A4,R(K),Q(K),A5,LA(K),A(K),
     .         O(K),HA(K),HK(K),CH4V(K),CH4PA(K),TROCK(K),PSI(K)
          NO(K)=A1+0.4
          JS(K)=A2+0.4
          JF(K)=A3+0.4
          NWTYP(K)=A4+0.4
          IF (A4.LT.0.0) NWTYP(K)=A4-0.0
          KF(K)=A5+0.4
          IF (IOUT.LE.(-2)) WRITE (8,1020) NO(K),JS(K),JF(K),R(K),
```

```
     .    Q(K),KF(K),LA(K),A(K),O(K),HA(K),HK(K),CH4V(K)
        ELSE
          READ (9,*,ERR=99) A1,A2,A3,A4,R(K),Q(K),A5,LA(K),A(K),
     .           O(K),HA(K),HK(K),CH4V(K),CH4PA(K),TROCK(K)
          NO(K)=A1+0.4
          JS(K)=A2+0.4
          JF(K)=A3+0.4
          NWTYP(K)=A4+0.4
          IF (A4.LT.0.0) NWTYP(K)=A4-0.4
          KF(K)=A5+0.4
          IF (IOUT.LE.(-2)) WRITE (8,1020) NO(K),JS(K),JF(K),R(K),
     .    Q(K),KF(K),LA(K),A(K),O(K),HA(K),HK(K),CH4V(K)
        ENDIF
        K1=JS(K)
        K2=JF(K)
        IF (NO(K).LE.0.OR.JS(K).LE.0.OR.JF(K).LE.0.OR.KF(K).LT.0.
     .    OR.LA(K).LT.0.0.OR.A(K).LT.0.0.OR.O(K).LT.0.0.OR.HA(K).
     .    LT.0.0.OR.HK(K).LT.0.0.OR.CH4V(K).LT.0.0.OR.TROCK(K).
     .    LT.0.0) THEN
          WRITE (8,1240) K
          NSTOP=1
        ENDIF
        IF (K1.GE.951.OR.K2.GE.951.OR.K1.LE.0.OR.K2.LE.0) THEN
          WRITE (8,1110) NO(K),JS(K),JF(K)
          NSTOP=1
        ENDIF
        MSL(K1)=1
        MSL(K2)=1
        IF (Q(K).LT.1000.0) Q(K)=1000.0
1     CONTINUE
      NJ=0
      DO 2 I=1,1000
        IF (MSL(I).EQ.1) THEN
          NJ=NJ+1
          MSL(I)=0
        ENDIF
2     CONTINUE
      IF (NJ.GT.NMY) WRITE (8,220) NJ,NMY
      DO 3 I=1,NB
        JSB(I)=JS(I)
3     CONTINUE
C
C    NVPN: MARK FOR PRESENCE OF JUNCTION CARDS.
C    NHM: MARK FOR DESIRED CALCULATION WITH WATER EVAPORATION
C        AND CONDENSATION.
C
      IF (NVPN.LE.0) THEN
        IF (NHM.EQ.0) THEN
          WRITE (8,1030)
          DO 4 K =1,NJ
            READ (9,*,ERR=99) A1,T(K),Z(K),CH4C(K)
            JNO(K)=A1+0.4
            IF (IOUT.LE.(-2)) THEN
              WRITE (8,1040) JNO(K),T(K),Z(K),CH4C(K)
            ENDIF
4           CONTINUE
```

```
         ELSE
           WRITE (8,1035)
           DO 5 K=1,NJ
             READ (9,*,ERR=99) A1,T(K),Z(K),CH4C(K),HJN(K)
             JNO(K)=A1+0.4
             IF (IOUT.LE.(-2)) THEN
               WRITE (8,1040) JNO(K),T(K),Z(K),CH4C(K),HJN(K)
             ENDIF
             IF (T(K).LT.(-100.0).OR.T(K).GT.1500.0) T(K)=75.0
             IF (ABS(T(K)).LT.0.01) T(K)=75.0
C             ********************************
             IF (HJN(K).LT.0.0.OR.HJN(K).GT.100.0) HJN(K)=80.0
             IF (ABS(HJN(K)).LT.0.01) HJN(K)=80.0
C             **********************************
             TS=T(K)
             IF (TS.GT.210.0) TS=210.0
             IF (TS.LE.122.0) THEN
              PV=-1.0682+0.07946*TS-0.00129*TS*TS+1.26E-5*TS*TS*TS
             ELSE
               PV=-33.786+0.97*TS-9.305E-3*TS*TS+3.65E-5*TS*TS*TS
             ENDIF
             PRH2O(K)=6.22E-3*PV*HJN(K)/(101.35-PV*HJN(K)/100.)
5          CONTINUE
         ENDIF
       ENDIF
C
C  ** INPUT OF FAN CHARACTERISTICS.
C     NFNUM: NUMBER OF FANS WITH THEIR CHARACTERISTICS.
C
       IF (NFNUM.GT.0) THEN
         IF (NFNUM.GT.IMX) THEN
           WRITE (8,230) NFNUM,IMX
           NSTOP=1
         ENDIF
         DO 10 K=1,NFNUM
           READ (9,*,ERR=99) A1,A2,A3
                                ***
           NOF(K)=A1+0.4
           MPTS(K)=A2+0.4
           NSWT(K)=A3+0.4
C          ***************
           IF (IOUT.LE.(-2)) WRITE (8,1050) K,NOF(K),MPTS(K)
           DO 6 KP=1,NB
             IF (NO(KP).EQ.NOF(K)) GO TO 7
6          CONTINUE
           WRITE (8,1230) NOF(K)
7          INDEX=MPTS(K)
           IF (INDEX.LE.1.OR.INDEX.GT.IMY) THEN
             WRITE (8,240) INDEX,IMY
             NSTOP=1
           ENDIF
           READ (9,*,ERR=99) (QF(K,I),PF(K,I),I=1,INDEX)
           IF (IOUT.LE.(-2)) THEN
             WRITE (8,1060)
             WRITE (8,1070) (QF(K,I)/1000.0,PF(K,I),I=1,INDEX)
           ENDIF
```

```
          DO 8 I=1,INDEX-1
            IF (QF(K,I+1).LE.QF(K,I)) THEN
              WRITE (8,360) NOF(K)
              NSTOP=1
            ENDIF
8          CONTINUE
10       CONTINUE
      READ (9,*,ERR=99) A1
      NPLOT=A1+0.4
      ENDIF
C
C ** INPUT OF DATA FOR TEMPERATURE PART OF PROGRAM.
C
    ELSE IF (JUMP1.EQ.2) THEN
      READ (9,*,ERR=99) A1,A2,TSTART,TIME,CRITSM,CRITGS,CRITHT,WRNPR,
     .        WRNSM,WRNGS,WRNHT,HSTART,CRITW
      NAV=A1+0.4
      JSTART=A2+0.4
      IF (IOUT.LE.(-2)) THEN
        WRITE (8,1100) NAV,JSTART,TSTART,TIME,CRITSM,CRITGS,CRITHT,
     .             WRNPR,WRNSM,WRNGS,WRNHT,HSTART,CRITW
      ENDIF
      IF (TIME.LT.0.5) TIME=10.0
      J3=0
      DO 12 I=1,NJ
        IF (JNO(I).EQ.JSTART) J3=1
12     CONTINUE
      IF (J3.EQ.0) THEN
        WRITE (8,330) JSTART
        NSTOP=1
      ENDIF
      IF (TSTART.LT.(-100.0).OR.TSTART.GT.180.0) TSTART=TR
      IF (CRITW.LT.1.0) CRITW=122.0
      IF (NHM.GT.0.AND.MARKY.EQ.0) WRITE (8,260) CRITW
      TS=TSTART
      IF (TS.LE.122.0) THEN
        PV=-1.0682+0.07946*TS-0.00129*TS*TS+1.26E-5*TS*TS*TS
      ELSE
        PV=-33.786+0.97*TS-9.305E-3*TS*TS+3.65E-5*TS*TS*TS
      ENDIF
C
C   HSTART: HUMIDITY RATIO IN THE USER-SPECIFIED STARTING JUNCTION.
C
      HSTART=6.22E-3*PV*HSTART/(101.35-PV*HSTART/100.0)
    ELSE IF (JUMP1.EQ.3) THEN
C
C ** INPUT OF DATA FOR FIRE SOURCES.
C   INFLOW: NUMBER OF CONTAMINATION CARDS.
C
      IF (INFLOW.GT.0) THEN
        IF (INFLOW.GT.IMX) THEN
          WRITE (8,250) INFLOW,IMX
          NSTOP=1
        ENDIF
        IF ((NB+INFLOW).GT.NMX) THEN
          WRITE (8,340) NB,INFLOW,NMX
```

```
      NSTOP=1
    ELSE
      IF (IOUT.LE.(-2).AND.NHM.GE.1) WRITE (8,1080)
      IF (IOUT.LE.(-2).AND.NHM.LE.0) WRITE (8,1085)
      DO 60 I=1,INFLOW
        IF (NHM.GE.1) THEN
          READ (9,*,ERR=99) A1,CONT(I),CONC(I),HEAT(I),
     .        O2MIN(I),SMPO2(I),HTPO2(I),QCENT(I),TPR(I),
     .        CN(I),HN(I)
          NCENT(I)=A1+0.4
          IF (IOUT.LE.(-2)) THEN
            WRITE (8,1090) NCENT(I),CONT(I),CONC(I),HEAT(I),
     .        O2MIN(I),SMPO2(I),HTPO2(I),TPR(I),QCENT(I),
     .        CN(I),HN(I)
          ENDIF
          IF (CN(I).LE.1.E-10.OR.HN(I).LT.1.E-10) THEN
            CN(I)=2.0
            HN(I)=4.0
          ENDIF
        ELSE
          READ (9,*,ERR=99) A1,CONT(I),CONC(I),HEAT(I),
     .        O2MIN(I),SMPO2(I),HTPO2(I),QCENT(I),TPR(I)
          NCENT(I)=A1+0.4
          IF (IOUT.LE.(-2)) THEN
            WRITE (8,1090) NCENT(I),CONT(I),CONC(I),HEAT(I),
     .        O2MIN(I),SMPO2(I),HTPO2(I),TPR(I),QCENT(I)
          ENDIF
        ENDIF
        IF (MAXJ.LT.951) MAXJ=951
        DO 50 J=1,NB
          IF (NO(J).EQ.NCENT(I)) THEN
            IF (NWTYP(J).NE.0) THEN
              WRITE (8,1250) NCENT(I)
              NSTOP=1
            ENDIF
            IF (TPR(I).LT.1.E-5) TPR(I)=1.E-5
            IF (QCENT(I).LT.0.0) QCENT(I)=0.0
            IF (QCENT(I).GT.10.0) THEN
              VART=(9900.0+TR)**2+2.0*HEAT(I)/
     .            (QCENT(I)*DR*2.4E-5)
              IF (VART.LT.9.4E7) THEN
                WRITE (8,930) NCENT(I)
                NSTOP=1
              ENDIF
              TESTI=-9900.0+SQRT(VART)
            IF (O2MIN(I).LE.0.1.AND.SMPO2(I).LE.1.E-3.AND.
     .        HTPO2(I).LT.1.0.AND.MARKY.LE.0) THEN
              IF (TESTI.LT.(-70.0).OR.TESTI.GT.3000.0)
     .          WRITE (8,870) NCENT(I)
            ENDIF
            ENDIF
            DO 145 K=1,NJ
              IF (JS(J).EQ.JNO(K)) GO TO 146
145         CONTINUE
            WRITE (8,900) NCENT
            NCENT(I)=0
```

```
                GO TO 60
146              DO 155 I2=NB+1,J+1,-1
                  NO(I2)=NO(I2-1)
                  JS(I2)=JS(I2-1)
                  JF(I2)=JF(I2-1)
                  NWTYP(I2)=NWTYP(I2-1)
                  R(I2)=R(I2-1)
                  RSTD(I2)=RSTD(I2-1)
                  Q(I2)=Q(I2-1)
                  KF(I2)=KF(I2-1)
                  LA(I2)=LA(I2-1)
                  A(I2)=A(I2-1)
                  O(I2)=O(I2-1)
                  HA(I2)=HA(I2-1)
                  HK(I2)=HK(I2-1)
                  CH4V(I2)=CH4V(I2-1)
                  CH4PA(I2)=CH4PA(I2-1)
                  TROCK(I2)=TROCK(I2-1)
155              CONTINUE
                NO(J+1)=1000+MAXNO+1
                IF (NO(J+1).GT.9995) NO(J+1)=MAXNO+1
                JS(J+1)=MAXJ+1
                JF(J)=MAXJ+1
                NWTYP(J)=10
                RSTD(J+1)=R(J+1)
                R(J)=R(J)*0.01
                IF (R(J).GT.0.001) R(J)=0.001
                RSTD(J)=R(J)
                LA(J+1)=LA(J)
                LA(J)=0.1
                CH4V(J)=0.0
                CH4PA(J)=0.0
                JNO(NJ+1)=MAXJ+1
                T(NJ+1)=T(K)
                Z(NJ+1)=Z(K)
                CH4C(NJ+1)=CH4C(K)
                NB=NB+1
                NJ=NJ+1
                MAXNO=MAXNO+1
                MAXJ=MAXJ+1
                GO TO 60
              ENDIF
50           CONTINUE
60          CONTINUE
           DO 65 I=1,NB
             JSB(I)=JS(I)
65          CONTINUE
         ENDIF
       ENDIF
     ENDIF
     RETURN
99    NSTOP=1
     WRITE (6,1210)
     WRITE (8,1210)
     RETURN
C
```

```
210  FORMAT (//,1X,'* ERR * NB+INFLOW MUST BE LESS THAN ',I5,' AND ',
     .     'LARGER THAN 1',/,' CURRENT NB= ',I4,'   INFLOW= ',I3)
220  FORMAT (//,1X,'* ERR * NETWORK CONTAINS',I4,' JUNCTIONS',/,
     .     1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I5)
230  FORMAT (//,1X,'* ERR * NETWORK CONTAINS',I4,' FANS',/,
     .     1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I4)
240  FORMAT (//,1X,'* ERR * FAN CURVE CONTAINS',I4,' POINTS',/,
     .     1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I4)
250  FORMAT (//,1X,'*  ERR * NETWORK CONTAINS',I4,' FIRE SOURCES'
     .     ,/,1X,'WHICH IS BEYOND THE CAPACITY OF ARRAYS:',I4)
260  FORMAT (/,1X,'* ATTN * DRY CONDITION ASSUMED WHEN WALL',
     .     ' TEMP. EXCEEDS',F6.0,' DEG. F.',/)
270  FORMAT (//,1X,'NO JUNCTION CARDS ARE EXPECTED TO READ IN',
     .     ' DUE TO NVPN > 0.',/,' ONLY NETWORK PART WILL BE',
     .     ' PERFORMED.')
290  FORMAT (4I5,F10.3,F10.0,5X,I5,F10.0,2F10.1)
310  FORMAT (I5,F10.2,4F10.5)
330  FORMAT (//,1X,'* ERR * START JUNCTION ',I5,' NOT IN JNO LIST.')
340  FORMAT (//,' * ERR * CURRENT INPUT DATA:  NB:',I5,5X,'INFLOW:',
     .     I5,/,' THE SUM OF NB AND INFLOW MUST BE LESS THEN ',I5)
350  FORMAT (//,' * ATTN * A NEW BRANCH HAS BEEN SET UP FOR FIRE ',
     .     'SOURCE',I6,/,' WHICH IS NOW CALLED BRANCH',I6,/)
360  FORMAT (//,' * ERR * FAN CURVE DATA INPUT MUST BE IN THE ORDER',
     .     ' FROM SMALL-QF',/,' TO LARGE-QF, INVALID ORDER DETECTED',
     .     ' FOR FAN ',I5)
870  FORMAT (/,1X,'* WRN *  ABNORMAL TEMPERATURE MAY RESULT IN ',
     .     'BRANCH',I5,/,' ADJUST HEAT AND/OR QCENT.')
880  FORMAT (/,' * ERR *  FUME CONCENTRATION (CONC) HIGHER THAN ',
     .     '100% (CONC=',F5.0,')',/,' FOR FIRE SOURCE',I5)
890  FORMAT (/,1X,'* WRN *  (HTPO2=',F5.0,') FOR FIRE SOURCE',I5,
     .     ' IS UNLIKELY HIGH,',/,' ANOMALOUS CONSEQUENCE MAY ',
     .     'RESULT.')
900  FORMAT (/,1X,' * WRN *  AIRWAY OR STARTING JUNCTION NUMBER ',
     .     'OF HEAT',/,1X,'SOURCE ',I5,' IS NOT ON THE LIST. DATA ',
     .     'IGNORED.')
930  FORMAT (/,1X,'* ERR *  ABNORMAL TEMPERATURE WAS RESULTED IN ',
     .     'AIRWAY ',I5,/,' ADJUST HEAT INPUT AND/OR QCENT.')
1000 FORMAT (//,' * CONTROL DATA LIST *',//,T2,'NB=',T7,I7,T18,'NFNUM='
     .    ,T25,I7,T35,'INFLOW=',T45,I7,T55,'NVPN=',T61,I7,/,T2,'NETW='
     .     ,T7,I7,T18,'TR=',T25,F7.1,T35,'MADJ=',T45,I7,T55,'ITN=',
     .     T61,I7,/,T2,'NTEMP=',T7,I7,T18,'TINC=',T25,F7.1,T35,'SPAN='
     .     ,T45,F7.1,T55,'IOUT=',T61,I7,/,T2,'DR=',T7,F7.4,T18,
     .     'NHM=',T25,I7)
1005 FORMAT (///,' * INPUT DATA LIST (AIRWAY: NWTYP, CH4PA, TROCK NOT',
     .     ' SHOWN) *',//,T3,'NO',T7,'JS',T12,'JF',T18,'R',T26,'Q',
     .     T34,'KF',T40,'LA',T46,'A',T52,'O',T58,'HA',T64,'HK',T69,
     .     'CH4V',/)
1010 FORMAT (///,' * INPUT DATA LIST (AIRWAY: NWTYP, CH4PA, TROCK AND',
     . ' PSI NOT SHOWN) *',//,T3,'NO',T7,'JS',T12,'JF',T18,'R',T26,'Q',
     .     T34,'KF',T40,'LA',T46,'A',T52,'O',T58,'HA',T64,'HK',T69,
     .     'CH4V',/)
1020 FORMAT (T2,I4,T6,I4,T10,I4,T15,F7.2,T23,F7.0,T32,I4,T37,F6.0,
     .     T44,F5.1,T50,F5.1,T56,F5.1,T62,F5.1,T68,F5.1)
1030 FORMAT (///,' * INPUT DATA LIST (JUNCTION DATA) *',//,
     .        T3,'JNO',T16,'T',T28,'Z',T40,'CH4C',/)
1040 FORMAT (T2,I5,T12,F7.1,T23,F8.1,T38,F6.2,T51,F6.2)
```

```
1035 FORMAT (///,' * INPUT DATA LIST (JUNCTION DATA) *',//,
    .           T3,'JNO',T16,'T',T28,'Z',T40,'CH4C',T53,'HJN',/)
1050 FORMAT (///,' * INPUT DATA LIST (FAN CURVE DATA) *',//,
    .          T5,'FAN ',I3,':   NOF=',I5,'    MPTS=',I5,/)
1060 FORMAT (/,5('  Q*1000   PF '),/)
1070 FORMAT (5(F8.2,F6.2)/,5(F8.2,F6.2),/)
1080 FORMAT (///,' * INPUT DATA LIST (HEAT SOURCE DATA) *',//,
    .          T2,'NCENT',T8,'CONT',T14,'CONC',T21,'HEAT',T28,'O2MIN',
    .      T35,'SMPO2',T41,'HTPO2',T49,'TPR',T56,'QCENT',T64,'CN',T70,
    .      'HN',/)
1085 FORMAT (///,' * INPUT DATA LIST (HEAT SOURCE DATA) *',//,
    .          T2,'NCENT',T8,'CONT',T14,'CONC',T21,'HEAT',T28,'O2MIN',
    .      T35,'SMPO2',T41,'HTPO2',T49,'TPR',T56,'QCENT',/)
1090 FORMAT (T2,I4,T6,F6.0,T12,F6.2,T18,F8.0,T28,F5.2,T34,F6.1,T40,F6.1
    .      ,T46,F6.2,T53,F8.0,T62,F5.2,T68,F5.2)
1100 FORMAT (///,' * INPUT DATA LIST (CONTROL DATA II) *',//,
    .          T2,'NAV=',T8,I7,T18,'JSTART=',T26,I7,T37,'TSTART=',T45,
    .      F7.2,T56,'TIME=',T63,F7.2,/,T2,'CRITSM=',T8,F7.2,T18,
    .      'CRITGS=',T26,F7.2,T37,'CRITHT=',T45,F7.2,T56,'WRNPR=',T63,
    .      F7.2,/,T2,'WRNSM=',T8,F7.2,T18,'WRNGS=',T26,F7.2,T37,
    .      'WRNHT=',T45,F7.2,T56,'HSTART=',T63,F7.2,/,T2,'CRITW=',
    .      T8,F7.2,/)
1110 FORMAT (/,' * ERR * JUNCTION ID MUST BE WITHIN 1 AND 950.',/,
    .      ' CURRENT INPUT:  NO=',I5,'  JS=',I5,'  JF=',I5)
1120 FORMAT (/,' * ERR * PARAMETER LMX MUST BE 1000 OR LARGER.')
1200 FORMAT (/,' * ATTN * REF. DENSITY MUST MATCH REF. TEMPERATURE ',
    .       'AND ATM. PRESSURE.',/)
1210 FORMAT (//,' * ERR * ERROR READING INPUT DATA.',/)
1220 FORMAT (/,' * ERR * INVALID NEGATIVE INPUT DETECTED IN CONTROL ',
    .       'DATA LINE I.')
1230 FORMAT (/,' * ATTN * FAN ',I5,' IS ISOLATED FROM THE NETWORK.')
1240 FORMAT (/,' * ERR * INVALID NEGATIVE INPUT DETECTED FOR AIRWAY ',
    .       'DATA GROUP LINE ',I4)
1250 FORMAT (/,' * ERR * A HEAT SOURCE MUST IN AN ORDINARY AIRWAY ',
    .       '(NWTYP=0).')
C
      END


C
C
      SUBROUTINE ITR (MARKX,MARKN,NSFLOW,NSNVP,MADJC,ITCT)
C
C     ------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       APPLICATION OF HARDY CROSS METHOD TO NETWORK BALANCING.
C
C     ------------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
C  ** APPLICATION OF HARDY CROSS METHOD.
C
5     IT=0
10    DQSUM=0.
```

```
      MBEGW=1
      DO 50 K=1,MNO
        MENDW=MEND(K)
        DPSUM=0.
        RQSUM=0.
        N=MSL(MBEGW)
        IF (N.LT.0) N=-N
        IF (NWTYP(N).EQ.(-1)) THEN
          MBEGW=MENDW+1
          GO TO 50
        ENDIF
        DO 30 J=MBEGW,MENDW
          N=MSL(J)
          FACT=1.
          IF (N.LT.0) THEN
            N=-N
            FACT=-1.
          ENDIF
          IF(NWTYP(N).EQ.1) THEN
            IF (NFNUM.GT.0) THEN
              DO 20 L=1,NFNUM
                IF (NFREG(L).EQ.N) THEN
                  RQSUM=RQSUM-(RGRAD(L)*100000)
C                                 ******
                  GO TO 25
                ENDIF
20            CONTINUE
            ENDIF
25          IF (IABS(JS(N)).NE.JSB(N)) FACT=-FACT
            DPSUM=DPSUM-FACT*R(N)
          ELSE
            RQ2=R(N)*ABS(Q(N))*2.E-5
            RQSUM=RQSUM+RQ2
            DP=R(N)*Q(N)*ABS(Q(N))*1.E-10
            DPSUM=DPSUM+FACT*DP
          ENDIF
30      CONTINUE
        IF (ABS(RQSUM).GT.1.E-10) THEN
          DQ=(DPSUM-FNVP(K))*1.E5/RQSUM
        ELSE
          DQ=0.0
        ENDIF
        DO 40 J=MBEGW,MENDW
          N=MSL(J)
          FACT=1.
          IF (N.LT.0) THEN
            N=-N
            FACT=-1.
          ENDIF
          Q(N)=Q(N)-(DQ*FACT)
40      CONTINUE
        DQSUM=DQSUM+ABS(DQ)
        MBEGW=MENDW+1
50    CONTINUE
C
      DO 100 KI=1,NB
```

```
      IF (NWTYP(KI).EQ.1.AND.NFNUM.GT.0) THEN
C        ***************
        DO 90 J=1,NFNUM
          IF (NOF(J).EQ.NO(KI)) THEN
            NFCW(J)=0
            NEGQ(J)=0
            IF (Q(KI).LT.0.0.AND.IABS(JS(KI)).EQ.JSB(KI))
     .        NEGQ(J)=1
            IF (Q(KI).GT.0.0.AND.IABS(JS(KI)).NE.JSB(KI))
     .        NEGQ(J)=1
            NPTS=MPTS(J)
            IF (NVPN.GE.1.AND.MADJC.LE.0) THEN
              TABF=TR
            ELSE
              NABF=JS(KI)
              DO 70 L=1,NJ
                IF (NABF.EQ.JNO(L)) THEN
                  TABF=T(L)
                  GO TO 80
                ENDIF
70            CONTINUE
            ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
80          IF (NEGQ(J).EQ.0) THEN
              IF (NSWT(J).EQ.0) NSWT(J)=2
              IF (NSKP(J).EQ.0) THEN
                IF (NSWT(J).EQ.2) THEN
                  CALL SPLINE (J,Q(KI),R(KI),TABF,0)
                ELSEIF (NSWT(J).EQ.1) THEN
                  CALL LSFAN (J,Q(KI),R(KI),TABF,0,0)
                ENDIF
                NSKP(J)=1
              ELSE
                IF (NSWT(J).EQ.2) THEN
                  CALL SPLINE (J,Q(KI),R(KI),TABF,1)
                ELSEIF (NSWT(J).EQ.1) THEN
                  CALL LSFAN (J,Q(KI),R(KI),TABF,1,0)
                ENDIF
              ENDIF
              GO TO 100
            ELSE
              R(KI)=PF(J,1)
              R(KI)=R(KI)*(TR+460.)/(TABF+460.)
              RGRAD(J)=0.0
            ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          ENDIF
90      CONTINUE
      ENDIF
100   CONTINUE
C
C ** CONTINUE COMPUTING IN PROPER PATH.
C   MARKX: 0: THE SUB. IS CALLED FROM NETWORK OR TEMP. PART OF
C          THE MAIN PROG..  1: CALLED FROM DYNAMIC SIMULATION PART.
C   MARKN: 0: NORMAL CONDITION.  1: SUB. IS TERMINATED DUE TO USER-
C
```

```
      IT=IT+1
        ITCT=ITCT+1
      IF (MARKX.LE.0) THEN
        IF (IT.LE.1) THEN
          GO TO 10
        ELSE IF ((DQSUM/FLOAT(MNO)).LT.10.) THEN
          RETURN
C         ELSE IF (ITCT.GT.80.AND.ITCT.LE.500) THEN
C           WRITE(8,*) 'DQSUM: ',DQSUM
        ELSE IF (ITCT.GT.500) THEN
C                   ***
           MARKN=1
           RETURN
        ELSE IF (IT.GT.50) THEN
C                  ****
          CALL ARR (MARKX)
          CALL BASE (NSTOP)
          CALL MSLIST (NSTOP)
          CALL MBLNC
C
C     NSNVP: 0: SUB. IS CALLED FROM NETWORK PART.  1: SUB. IS
C            CALLED FROM TEMP. OR DYNAMIC SIMULATION PART.
C
          IF (NSNVP.LE.0) THEN
            IF (NVPN.LE.0) CALL NVP1
          ELSE
            CALL NVP2
          ENDIF
          GO TO 5
        ENDIF
      ELSE
        IF (DQSUM/FLOAT(NB-NJ+1).LT.10.) RETURN
        IF (IT.GT.200) THEN
          WRITE (8,200)
          RETURN
        ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        IF (INIFAN.EQ.1.AND.IT.GT.50) THEN
          CALL ARR (MARKX)
          CALL BASE (NSTOP)
          CALL MSLIST (NSTOP)
          CALL MBLNC
          CALL NVP1
C          *********
          CALL NVP2
          GO TO 10
        ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      ENDIF
      GO TO 10
C
200   FORMAT (//,1X,'* ATTN * THE ACCURACY IN NETWORK BALANCING ',
     .        'WAS NOT SATISFIED',/,' DUE TO ITERATIONS BECAME ',
     .        'EXCESIVE.')
C
      END
```

```
C
C
      SUBROUTINE LSFAN (L,QKI,RKI,TABF,NTW,NEXP)
C     ******************************************
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       USING LEAST SQUARE METHOD FOR FAN CURVE FITTING.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
      DIMENSION ZZ(IMX),S(IMX),ALPHA(IMX),BETA(IMX),SIGMA2(IMX),
     .PPP(IMX,IMX)
      DOUBLE PRECISION W(IMX)
C
      NFCW(L)=0
      NP=MPTS(L)
      IF (NTW.EQ.1) GO TO 145
      MARKXX=0
      DO 1 I=1,10
1      ALPHA(I)=0.0
      IF (NP.GT.2) GO TO 35
      ND(L)=1
      N1A=2
      NCOF(L)=N1A
      IF (NP.LT.1) THEN
        STOP 10
      ELSEIF (NP.EQ.1) THEN
        COF(L,1)=PF(L,1)
        COF(L,2)=0.
        GO TO 145
      ENDIF
      IF (QF(L,1).EQ.QF(L,2)) THEN
        STOP 25
      ELSE
        COF(L,1)=(PF(L,1)*QF(L,2)-PF(L,2)*QF(L,1))/(QF(L,2)-QF(L,1))
        COF(L,2)=(PF(L,2)-PF(L,1))/(QF(L,2)-QF(L,1))
        GO TO 145
      ENDIF
35     DSQ=0.0
      DO 40 J=1,NP
        DSQ=DSQ+PF(L,J)*PF(L,J)
        PPP(1,J)=0.
40       PPP(2,J)=1.
      W(1)=NP
      BETA(1)=0.
      N1A=4
      IF (NP.LT.5) N1A=NP-1
      IF (NEXP.GT.0) N1A=NEXP
      DO 85 I=1,N1A
        K=I+1
        ZZ(I)=0.
        DO 60 J=1,NP
60        ZZ(I)=ZZ(I)+PF(L,J)*PPP(K,J)
```

```
      S(I)=ZZ(I)/W(I)
      DSQ=DSQ-S(I)*S(I)*W(I)
      SIGMA2(I)=DSQ/FLOAT(NP-I)
      IF(SIGMA2(I).LE.1.D-6) GO TO 105
      IF (I.GE.N1A) GO TO 90
        ALPHA(I)=0.
        DO 75 J=1,NP
75        ALPHA(I)=ALPHA(I)+QF(L,J)*PPP(K,J)*PPP(K,J)
        ALPHA(I)=ALPHA(I)/W(I)
        W(K)=0.
        DO 80 J=1,NP
          PPP(K+1,J)=(QF(L,J)-ALPHA(I))*PPP(K,J)-BETA(I)*PPP(I,J)
80         W(K)=W(K)+PPP(K+1,J)*PPP(K+1,J)
85   BETA(K)=W(K)/W(I)
90   SMALL=SIGMA2(1)
     I=1
     DO 100 J=2,N1A
       IF (SMALL.LE.SIGMA2(J)) GO TO 100
       I=J
       SMALL=SIGMA2(J)
100  CONTINUE
105  WRITE (8,195) SIGMA2(I)
     N1A=I
     NCOF(L)=N1A
     N=N1A-1
     IF (N.LE.0) THEN
       ND(L)=1
       COF(L,1)=S(1)
       COF(L,2)=0.
       GO TO 145
     ENDIF
     ND(L)=N
     DO 125 I=1,N1A
       DO 120 J=1,I
120        PPP(I,J)=0.0
125    PPP(I,I+1)=1.
     DO 130 J=1,N
130    PPP(1,J+2)=-PPP(1,J+1)*ALPHA(J)-PPP(1,J)*BETA(J)
     DO 135 I=2,N
       DO 135 J=I,N
135        PPP(I,J+2)=PPP(I-1,J+1)-PPP(I,J+1)*ALPHA(J)-PPP(I,J)*BETA(J)
     DO 140 I=1,N1A
       COF(L,I)=0.
       DO 140 J=1,N1A
140  COF(L,I)=COF(L,I)+PPP(I,J+1)*S(J)
     NSKP(L)=1
     WRITE (8,*) 'ENTER FAN CURVE FITTING'
145  IF (MARKXX.EQ.0) WRITE (8,200) (COF(L,I),I=1,N1A)
     IF (ABS(QKI).LT.1.D-6) GO TO 160
     MARKXX=1
     IF (NPLOT.EQ.1) THEN
       IF (QKI.LT.QF(L,1).OR.QKI.GT.QF(L,NP)) NFCW(L)=1
       FANQ=QKI
       RGRAD(L)=0.0
       II=1
       GO TO 165
```

```
      ELSE IF (NPLOT.EQ.2) THEN
        IF (QKI.LT.QF(L,NP)) GO TO 146
        RKI=PF(L,NP)
        RGRAD(L)=0.0
        NFCW(L)=1
        RETURN
146     IF (QKI.GT.QF(L,1)) GO TO 147
        NFCW(L)=1
147      FANQ=QKI
        RGRAD(L)=0.0
        II=1
        GO TO 165
      ELSE IF (NPLOT.EQ.3) THEN
        IF (QKI.LT.QF(L,NP)) GO TO 150
        RKI=PF(L,NP)
        RGRAD(L)=0.0
        NFCW(L)=1
        RETURN
150     IF (QKI.GT.QF(L,1)) GO TO 155
        RKI=PF(L,1)
        RGRAD(L)=0.0
        NFCW(L)=1
        RETURN
155      FANQ=QKI
        RGRAD(L)=0.0
        II=1
        GO TO 165
      ENDIF
160   QCS=QF(L,NP)-QF(L,1)
      QCSA=QCS/120.
      RGRAD(L)=0.0
      Q0=QF(L,1)-5.*QCSA
      Q2(1)=Q0
      FANQ=Q0
      II=130
165   NE=ND(L)
      DO 180 I=1,II
        H2(I)=COF(L,NE+1)*FANQ
        IF (NE.LE.1) GO TO 175
        KT=NE
        DO 170 J=2,NE
          H2(I)=(H2(I)+COF(L,KT))*FANQ
          IF (MARKXX.EQ.1) RGRAD(L)=(RGRAD(L)+KT*COF(L,KT+1))*FANQ
170         KT=KT-1
175      H2(I)=H2(I)+COF(L,1)
        RGRAD(L)=RGRAD(L)+COF(L,2)
        IF (MARKXX.EQ.1) RKI=H2(1)
        Q2(I+1)=FANQ+QCSA
        FANQ=Q2(I+1)
180   CONTINUE
      IF (MARKXX.EQ.1) RETURN
      WRITE (6,190) (Q2(I),H2(I),I=1,130)
190   FORMAT (1X,5(F8.5,F6.2))
195   FORMAT (8HSIGMA2=,1PE14.7)
200   FORMAT (1H0,17X,12HCOEFFICIENTS,//8X,1P6E18.7)
      RETURN
```

```
      END
C
C
C                  MFIRE 2.10
C     TRANSIENT-STATE SIMULATION PROGRAM FOR MINE VENTILATION
C             UNDER THE INFLUENCE OF FIRES
C
C               VERSION OF DECEMBER 1994
C
C           WRITTEN BY    XINTAN CHANG, PH.D
C           SUPERVISED BY   DR. R. E. GREUER
C
C     MODIFIED BY X. ZHOU AND L. LAAGE
C
      PROGRAM MFIRE
      INCLUDE 'CMMNW.DAT'
C
C     PARAMETER
C     . ( NMX=500, NMY=350, NMZ=500, IMX=10, IMY=10, IMZ=10, LMX=15000 )
C
C     REAL LA
C     INTEGER CONCT
C
C     COMMON  /CONTRL/ NB,NJ,NFNUM,NVPN,NETW,NTEMP,MADJ,ITN,DR,TR,TINC,
C     .            SPAN,NHM,CRITW,IOUT,TOUT,CONCT
C
C     NB: NUMBER OF AIRWAYS IN NETWORK.
C     NJ: NUMBER OF JUNCTIONS IN NETWORK.
C     NFNUM: NUMBER OF FAN CHARACTERISTICS.
C     NVPN: MARK FOR PRESENCE OF JUNCTION CARDS, 1: PRESENT.
C     NETW: MARK, 1: ONLY THE NETWORK PART WILL BE PERFORMED.
C     NTEMP: MARK, 1: ONLY UP TO THE TEMP. PART WILL BE PERFORMED.
C     MADJ: MAX. NO. OF ITERATION IN TEMP. PART.
C     ITN: MAX. NO. OF ITERATION IN NETWORK BALANCE.
C     DR: REFERENCE DENSITY OF AIR, LBM/FT3.
C     TR: REFERENCE TEMPERATURE OF AIR, DEG. F.
C     TINC: TIME INCREMENT IN SIMULATION, SEC..
C     SPAN: TIME SPAN OF SIMULATION, MIN..
C     NHM: MARK FOR WET CASE CALCULATION, 0 DRY; 1 WET.
C     CRITW: CRITERION IN WALL TEMP. ASSUMING DRY, DEG. F.
C     IOUT: OUTPUT, 0: BRIEF; 1: NORMAL; -1: DETAIL; -2: MORE DETAILED.
C
C     COMMON  /CTRL/ NAV,MAXJ,INFLOW,CRITSM,CRITGS,CRITHT
C
C     NAV: MARK FOR PRESENCE OF AVE. VALUE CARDS.
C     MAXJ: HIGHEST JUNCTION NUMBER.
C     INFLOW: NUMBER OF CONTAMINATION CARDS.
C     CRITSM: ACCURACY OF FUME CALCULATION.
C     CRITGS: ACCURACY OF METHAME CALCULATION.
C     CRITHT: ACCURACY OF TEMPERATURE CALCULATION.
C
```

```
C   COMMON  /NTWK/ NO(NMX),JS(NMX),JF(NMX),NWTYP(NMX),R(NMX),Q(NMX),
C  .         P(NMX),KF(NMX),LA(NMX),A(NMX),O(NMX),RSTD(NMX),
C  .         DZRD(NMX),FRNVP(NMX),NREV(NMX),RDCH4(NMX),
C  .         RDPROP(NMX),TRD(NMX),TJS(NMX),RDOP(NMX),RCH4(NMX),
C  .         FFRNVP(NMX),RA(NMX),NNREV(NMX),RDH2O(NMX),RH2O(NMX)
C
C   NO(NMX): AIRWAY NUMBER.
C   JS(NMX): JUNCTION NUMBER OF AIRWAY BEGINNING.
C   JF(NMX): JUNCTION NUMBER OF AIRWAY END.
C   NWTYP(NMX): AIRWAY TYPE.
C   R(NMX): RESISTANCE OF AIRWAY 10**10 IN. W.G./(CFM)**2.
C   Q(NMX): BALANCED AIRFLOW RATE, ASSUMING EQUILIBRIA, CFM
C   P(NMX): CALCULATED PRESSURE LOSS IN IN. W.G..
C   KF(NMX): FRICTION FACTOR.
C   LA(NMX): AIRWAY LENGTH IN FT.
C   A(NMX): AIRWAY CROSS SECTIONAL AREA IN FT**2.
C   O(NMX): AIRWAY PERIMETER IN FT.
C   RSTD(NMX): RESISTANCE FACTOR BASED ON TR.
C   DZRD(NMX): ELEVATION CHANGE IN AIRWAY IN FT.
C   FRNVP(NMX): PRODUCT TMRS*DZRD.
C   NREV(NMX): NUMBER OF AIRFLOW REVERSALS IN AN INTERVAL.
C   RDCH4(NMX): METHANE CONCENTRATION AT ROADWAY ENDS.
C   RDPROP(NMX): CONTAMINANT CONCENTRATION AT ROADWAY ENDS.
C   TRD(NMX): TEMPERATURE AT ROADWAY ENDS.
C   TJS(NMX): TEMPERATURE AT ROADWAY BEGINNING.
C   RDOP(NMX): BACKUP ARRAY FOR RDPROP.
C   RCH4(NMX): BACKUP ARRAY FOR RDCH4.
C   FFRNVP(NMX): BACKUP ARRAY FOR FRNVP.
C   RA(NMX): BACKUP ARRAY FOR R.
C   NNREV(NMX): NUMBER OF AIRFLOW REVERSALS SINCE TIME ZERO..
C   RDH2O(NMX): HUM. RATIO AT ENDS OF AIRWAYS.
C   RH2O(NMX): BACKUP ARRAY FOR RDH2O.
C
C   COMMON  /FAN/ NOF(IMX),NFREG(IMX),RGRAD(IMX),NFCW(IMX),MPTS(IMX),
C  .        QF(IMX,IMY),PF(IMX,IMY),NSKP(IMX),NEGQ(IMX),JSB(NMX)
C
C   NOF(IMX): AIRWAY NUMBER OF FAN.
C   NFREG(IMX): LIST OF FANS WITH CHARACTERISTICS.
C   RGRAD(IMX): SLOPE OF FAN CHARACTERISTIC.
C   NFCW(IMX): LIST OF FANS WHOSE CHAR. EXCEEDED.
C   MPTS(IMX): NUMBER OF POINTS DEFINING FAN CHAR.
C   QF(IMX,IMY): AIRFLOW AT GIVEN POINT OF FAN CHAR.
C   PF(IMX,IMY): PRESSURE AT GIVEN POINT OF FAN CHAR.
C   NSKP(IMX): MARK INDICATING PERFORMANCE OF SPLINE.
C   NEGQ(IMX): MARK INDICATING AIRFLOW REVERSAL IN FAN BRANCH.
C   JSB(NMX): BACKUP ARRAY FOR JS.
C
C   COMMON  /MESH/ MNO,MEND(NMY),MSL(LMX),FNVP(NMY),RQ(NMX),INU(NMX),
C  .         KNO(NMX),KJS(NMX),KJF(NMX)
C
C   MNO: NUMBER OF MESHES IN NETWORK.
C   MEND(NMY): LIST OF MESH ENDS IN MSL-LIST.
C   MSL(LMX): LIST OF ALL INDEPENDENT MESHES.
C   FNVP(NMY): LIST OF NVP IN MESHES.
C   RQ(NMX): AUXILIARY LIST OF R*Q.
C   INU(NMX): AUXILIARY LIST FOR FORMING BASE SYST.
```

```
C    KNO(NMX): LIST OF AIRWAYS IN BASE SYSTEM.
C    KJS(NMX): LIST OF JUNCTIONS IN BASE SYSTEM.
C    KJF(NMX): LIST OF JUNCTIONS IN BASE SYSTEM.
C
C    COMMON  /JUNCT/ JNO(NMY),T(NMY),Z(NMY),CH4C(NMY),JNOL(NMY),
C  .             PROP(NMY),PRCH4(NMY),JLR(NMY),HJN(NMY),PRH2O(NMY)
C
C    JNO(NMY): JUNCTION NUMBER.
C    T(NMY): TEMPERATURE OF AIR IN JUNCTIONS.
C    Z(NMY): ELEVATION OF JUNCTIONS.
C    CH4C(NMY): METHANE CONCENTRATION IN JUNCTION.
C    JNOL(NMY): LIST OF JUNCTION NUMBERS IN INCREASING ORDER.
C    PROP(NMY): CONTAMINANT CONCENTRATION IN JUNCTION.
C    PRCH4(NMY): METHANE CONCENTRATION IN JUNCTION.
C    JLR(NMY): LIST RELATING JNO- AND JNOL-LISTS.
C    HJN(NMY): RELATIVE HUMIDITY IN JUNCTIONS, PERCENTAGE.
C    PRH2O(NMY): ABSOLUTE HUMIDITY IN JUNCTIONS.
C
C    COMMON  /TEMP/ TROCK(NMX),TMRD(NMX),HA(NMX),HK(NMX)
C
C    TROCK(NMX): AVERAGE ROCK TEMPERATURE IN AIRWAY.
C    TMRD(NMX): MEAN TEMPERATURE OF AIR IN AIRWAY.
C    HA(NMX): THERMAL DIFFUSIVITY OF ROCK.
C    HK(NMX): THERMAL CONDUCTIVITY OF ROCK.
C
C    COMMON  /FUME/ NCENT(NMX),CH4V(NMX),CH4PA(NMX),CONT(IMX),CONC(IMX)
C  .            ,HEAT(IMX),O2MIN(IMX),SMPO2(IMX),HTPO2(IMX),
C  .            TFSI(IMX),CN(IMX),HN(IMX),O2BEH(IMX),TPR(IMX),
C  .            HTAD(IMX),QCENT(IMX)
C
C    NCENT(NMX): NUMBER OF AIRWAYS INTO WHICH FUME ENTERS.
C    CH4V(NMX): METHANE EMISION RATE IN AIRWAYS.
C    CH4PA(NMX): METHANE EMISION RATE PER UNIT SURFACE AREA.
C    CONT(IMX): VOLUME FLOW RATE OF CONTAMINATED GAS INFLOW.
C    CONC(IMX): CONCENTRATION OF CONTAMINANT IN GAS INFLOW.
C    HEAT(IMX): HEAT ENTERING AIRWAY.
C    O2MIN(IMX): OXYGEN CONCENTRATION OF FUMES LEAVING FIRE ZONE
C    SMPO2(IMX): CONTAMINANT PRODUCTION PER CUBIC FT OF OXYGEN.
C    HTPO2(IMX): HEAT PRODUCTION PER CUBIC FT OF OXYGEN DELIVERY
C    TFSI(IMX): AIR TEMPERATURE BEHIND HEAT SOURCE.
C    CN(IMX): EVERAGE CARBON CONTENT IN FUEL.
C    HN(IMX): EVERAGE HYDROGEN CONTENT IN FUEL.
C    O2BEH(IMX): OXYGEN CONTENT BEHIND FIRE SOURCES.
C    TPR(IMX): TRANSITION TIME OF FIRE IN MIN.
C    HTAD(IMX): REDUCED HEAT INPUT DUE TO HIGH AIR TEMPERATURE.
C    QCENT(IMX): "STD. Q' AT WHCIH FIRE TAKES THE INPUT PARAMETERS.
C
C    COMMON  /SCHEME/ NGIN(NMX),MIN(NMX),NGOUT(NMX),LOUT(NMX)
C
C    NGIN(NMX): LIST OF AIRWAYS ENTERING JUNCTION.
C    MIN(NMX): LIST OF LAST AIRWAY PER JUNCTION IN NGIN-LIST.
C    NGOUT(NMX): LIST OF AIRWAYS LEAVING JUNCTION.
C    LOUT(NMX): LIST OF LAST AIRWAY PER JUNCTION IN NGOUT-LIST.
C
C    COMMON  /EST/ MEMREC(NMX),NOREC(NMX),ESTPR(NMX),ESTCH4(NMX),
C  .           ESTTR(NMX)
```

```
C
C   MEMREC(NMX): TEMPORARY LIST OF AIRWAYS CARRYING RECIRC. AIR.
C   NOREC(NMX): PERMANENT LIST OF AIRWAYS.
C   ESTPR(NMX): ESTIMATED CONCENTRATION FOR RECIRCULATED AIR.
C   ESTCH4(NMX): ESTIMATED METHANE CONCENTRATION.
C   ESTTR(NMX): ESTIMATED TEMPERATURE.
C
C   COMMON  /FACTOR/ XNEW(NMX),DCOAGE(NMX),WTCORR(NMZ,10),NSEQS(NMZ)
C
C   XNEW(NMX): EXPONENT FOR TEMPERATURE CALCULATION.
C   DCOAGE(NMX): COEFF. OF AGE FOR "DRY CASE".
C   WTCORR(NMZ,10): TEMPERATURE CORRECTION IN WET CASE.
C   NSEQS(NMZ): COUNTER OF TIME INTERVALS.
C
C   COMMON  /TTJ/ TIME,TSTART,JSTART,TLEFT(NMX)
C
C   TIME: TIME ASSUMING QUASI-EQUILIBRIA.
C   TSTART: TEMPERATURE AT THE START JUNCTION.
C   JSTART: START JUNCTION (ATMOSPHERE).
C   TLEFT(NMX): TIME INCREMENT FOR MAKING UP A NORMAL INTERVAL.
C
C   COMMON  /SUM/ SUMQ(NMY),SUMC(NMY),SUMM(NMY),SMHEAT(NMY),SUMW(NMY)
C
C   SUMQ(NMY): TOTAL AIRFLOW RATES ENTERING JUNCTION.
C   SUMC(NMY): TOTAL CONTAMINANT FLOW RATES ENTERING JUNCTION.
C   SUMM(NMY): TOTAL METHANE FLOW RATES ENTERING JUNCTION.
C   SMHEAT(NMY): TOTAL ENTHALPY/REFRENCE DENSITY ENTERING JUNCTI
C   SUMW(NMY): TOTAL WATER VAPOR ENTERING JUNCTION.
C
C   COMMON  /AUX/ PROPS(NMX),PRCH4S(NMX),QQ(NMX),QTP(NMX),TTJS(NMX),
C  .             TTRD(NMX),TAS(NMX),TMM(NMX),BI(NMX),CNT(NMX),
C  .             EXPN(NMX),FAW(NMX)
C
C   PROPS(NMX): BACKUP ARRAY FOR PROP.
C   PRCH4S(NMX): BACKUP ARRAY FOR PRCH4.
C   QQ(NMX): DAMPED AND TIME-AVERAGED AIRFLOW RATE.
C   QTP(NMX): BACKUP ARRAY FOR Q, FIXED IN EACH INTERVAL.
C   TTJS(NMX): BACKUP ARRAY FOR TJS.
C   TTRD(NMX): BACKUP ARRAY FOR TRD.
C   TAS(NMX): BACKUP ARRAY OF T.
C   TMM(NMX): (NOT IN EFFECTIVE USE.)
C   BI(NMX): BIOT NUMBER.
C   CNT(NMX): INTERMEDIATE STORAGE ARRAY.
C   EXPN(NMX): INTERMEDIATE STORAGE ARRAY.
C   FAW(NMX): EXPONENTIAL FACTOR IN TEMP. EVALUATION.
C*** FBN(NMX): WEIGHING FACTOR IN CALCULATING JUNCTION TEMPERATURE.
C
C   COMMON  /HUMID/ PRH2OS(NMY),HSTART,PSI(NMX),NSP(NMX),STV(NMZ,10),
C  .             HABS(NMZ,10),NSEQE(NMZ),TDA(NMZ,10),WMTR(NMZ,10),
C  .             GFF(NMZ,10)
C
C   PRH2OS(NMY): BACKUP ARRAY FOR PRH2O.
C   HSTART: REL. HUM. IN THE START JUNCTION.
C   PSI(NMX): WETNESS RATIO OF AIRWAYS.
C   NSP(NMX): NUMBER OF SECTIONS FOR T. CORR. EVALUATION.
C   STV(NMZ,10): INTERMEDIATE DATA FOR TEMP. CORR.
```

```
C    HABS(NMZ,10): ABS. HUMIDITY AT SECTION BOUNDARY.
C    NSEQE(NMZ): ADDRESS OF RECORD FOR WET TEMP. CORR.
C    TDA(NMZ,10): AIR TEMP. AT STATIONS.
C    WMTR(NMZ,10): MASS TRANSFER OF WATER VAPOR, LBM/FT**2.
C    GFF(NMZ,10): VAPOR CONDENSED AS MIST, LB/LB.
C
C    COMMON  /WRN/ WRNPR,WRNGS,WRNSM,WRNHT,IOMIT(IMX),ROMIT(IMX,3),IAC
C
C    WRNPR: PRESSURE DROP WARNING CRITERIA.
C    WRNGS: METHANE CONCENTRATION WARNING CRITERIA.
C    WRNSM: FUME CONCENTRATION WARNING CRITERIA.
C    WRNHT: HIGH TEMPERATURE WARNING CRITERIA.
C    IOMIT(IMX): AIRWAYS FOR WHICH DETAILED DATA RECORDS WERE OMITTED.
C    ROMIT(IMX,3): DATA AT STARTING J OF DATA-OMITTED AIRWAY.
C    IAC: COUNTER OF IOMIT AND ROMIT.
C
C    COMMON  /TRANS/ DELTAT,SUMFNV
C
C    DELTAT: SPAN OF TIME INCREMENT.
C    SUMFNV: SUM OF NVP CORRECTION IN SYSTEM.
C
C    COMMON  /RCD/ IBTN(NMX),DPPA(NMZ,10,5),NSAC(NMX,2),TMRDA(NMX),
C  .           DELTAQ(NMX),DPPB(NMZ,10,5),NSACB(NMX,2),GBTN(10,5),
C  .           JCH(NMY),TAUXA(NMZ,10),TAUXB(NMZ,10),TAUXC(NMZ),
C  .           TAUXD(NMZ),FMASS(NMZ)
C
C    IBTN(NMX): ARRAY HOLDING ADDRESS OF RECORD FOR AIRWAY (N
C    DPPA(NMZ,10,5): MASTER DATA STORAGE ARRAY IN DYNAMIC SIMULATI
C    NSAC(NMX,2): DATA RECORD STATUS OF AIRWAY (NMX).
C    TMRDA(NMX): BACKUP ARRAY FOR TMRD.
C    DELTAQ(NMX): VARIATION OF AIRFLOW IN AIRWAYS.
C    DPPB(NMZ,10,5): BACKUP ARRAY FOR DPPA.
C    NSACB(NMX,2): BACKUP ARRAY FOR NSAC.
C    GBTN(10,5): AUXILIARY ARRAY FOR DATA TRANSFER.
C    JCH(NMY): MARK INDICATING JUNCTION COND. CHANGE.
C    TAUXA(NMZ,10): AIR TEMP. AT REAR ENDING OF SEGMENTS.
C    TAUXB(NMZ,10): BACKUP ARRAY FOR TAUXA.
C    TAUXC(NMZ): AIR TEMP. BEYOND THE FIRST SEGMENT.
C    TAUXD(NMZ): BACKUP ARRAY OF TAUXC.
C    FMASS(NMZ): MASS OF AIR BEYOND THE FIRST SEGMENT.
C
C    COMMON  /CDEV/ GFS(IMX),TACC
C
C    GFS(IMX): HUMIDITY RATIO BEHIND FIRE SOURCE.
C    TACC: CUMULATED TIME AFTER EVENT IN SEC.
C
C    COMMON  /CCJ/ JDP, JDPP(IMX)
C
C    JDP: NUMBER OF JUNCTIONS INITIATING DATA RECORDS.
C    JDPP(IMX): JUNCTIONS WHICH INITIATE DATA RECORDS.
C
C    COMMON  /TREND/ QRCD(NMX,10),TMRCD(NMX,10)
C
C    QRCD(NMX,10): AIRFLOW RATES DURING ITERATIONS.
C    TMRCD(NMX,10): MEAN AIR TEMP. IN AIRWAYS DURING ITERATION.
C
```

```
C
      COMMON /FAN1/COF(IMX,IMX),ND(IMX),N1A,NSWT(IMX),NCOF(IMX),
C   .         INIFAN,NPLOT
C     COF: THE COEFFIENT OF FAN EQUATION FITTED BY LEAST SQUARES.
C     ND: THE DEGREE OF FAN EQUATION FITTED BY LEAST SQUARES.
C     N1A: THE NUMBER OF COEFFICIENTS OF FAN EQUATION.
C     NSWT: TO SELECT FAN CURVE FITTING METHOD
C              1 LEAST SQUARE 2.SPLINE 3.AUTOMATIC 4.AUTO+MANUAL
C     NCOF: ARRAY FOR THE NUMBER OF COEFFICIENTS OF FAN EQUATION
C     INIFAN: THE FLAG SHOWS THE LARGE STATE CHANGE IN SYSTEM AND THE
C            MESHES REFORM DESIRED.
C     NPLOT: TREATMENT OF THE BOUNDARY RANGE OF FAN CURVE
C           1: EXTEND FAN CURVE BY FOLLOWING GRADIENTS OF TWO ENDS.
C           2: EXTEND FAN CURVE AS ABOVE AT LEFT BROUNDARY RANGE AND
C             SEND GRADIENT OF RIGHT BOUNDARY RANGE TO ZERO.
C           3. SEND GRADIENT OF BOTH SIDES OF BOUNDARY RANGE TO ZERO.
C     COMMON /PLOT/QQ1(NMX),H1(NMX),Q2(150),H2(NMX),Q3(NMX),H3(NMX),
C   .      QQ2(NMX)
C
      LOGICAL EXISTS
      CHARACTER*50 MD1
      CHARACTER*1 CONKEYC    COMMON C
C     MDATA: NAME OF THE INPUT DATA FILE.
C
C  ** NETWORK PART
C
C     MARKX: 0: THE PROG. HAS NOT REACHED THE DYNAMIC SIMULATION
C           PART YET.  1: THE DYNAMIC PART HAS BEEN REACHED.
C     MARKY: 0: BEFORE FINAL QUASI-EQUILIBRIUM SIMULATION.
C           1: IN THE FINAL QUASI-EQUIL. STAGE, READY FOR TERMINATION.
C     MARKN: 0: NORMAL CONDITION.  1: NETWORK BALANCING WAS TERMINATED
C           DUE TO USER-SPECIFIED MAX. MUNBER OF ITERATIONS EXCEEDED.
C     MARKC: 0: NORMAL CONDITION.  1: CALCULATION IN TEMP. PART WAS
C           TERMINATED DUE TO MAX. NUMBER OF ITERATIONS EXCEEDED.
C     NSTOP: MARK FOR INPUT DATA ERROR.
C
      NSTOP=0
      MARKX=0
      MARKY=0
      ITT=0
      MD1='                                         '
      WRITE (6,910)
      WRITE (6,900)
5     WRITE (6,680)
      READ (5,'(A)') MD1
C
      INQUIRE (FILE = MD1, EXIST = EXISTS)
      IF (.NOT. EXISTS) THEN
        WRITE (*, '(2A/)') '  CAN NOT FIND FILE ', MD1
10       WRITE (*, '(2A/)') '  DO YOU WANT TO CONTINUE?'
        WRITE (*, '(2A/)') '   Y(es)  or  Q(uit)'
        READ (*,'(A)') CONKEY
          IF (CONKEY .EQ. 'Q') THEN
            STOP
          ELSEIF (CONKEY .EQ. 'q') THEN
            STOP
```

```
         END IF
         IF (CONKEY .EQ. 'Y') THEN
           GOTO 5
         ELSEIF (CONKEY .EQ.'y') THEN
           GOTO 5
         END IF
         GOTO 10
      END IF
      CONTINUE
C
      OPEN (9,FILE=MD1,STATUS='UNKNOWN')
      OPEN (8,FILE='WFOUT',STATUS='UNKNOWN')
1     REWIND (9)
      CALL INPUT (1,NSTOP,MARKY,MAXNO)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF (NPLOT.EQ.0) NPLOT=2
      DO 2 I=1,NFNUM
        IF (NSWT(I).EQ.0) NSWT(I)=1
2     CONTINUE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF (NSTOP.GE.1) GO TO 300
      CALL CHECK1 (NSTOP,MAXNO)
      IF (NETW.NE.1) CALL INPUT (3,NSTOP,MARKY,MAXNO)
      IF (NSTOP.GE.1) GO TO 300
      CALL OUTPUT (1,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
      MADJC=0
      MARKN=0
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF (MARKY.EQ.1) GO TO 100
      NSTOP1=0
      CALL CHSFIT (NSTOP1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
100   ITCT=0
      IF (MARKY.EQ.0) WRITE (6,920)
      CALL ARR (MARKX)
      CALL BASE (NSTOP)
      IF (NSTOP.GE.1) GO TO 300
      CALL MSLIST
      CALL MBLNC
      CALL NVP1
      CALL ITR (MARKX,MARKN,NSFLOW,0,MADJC,ITCT)
      CALL RGLT
      CALL OUTPUT (2,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
      IF (MARKY.EQ.0) WRITE (6,930)
      IF (NETW.EQ.1) THEN
        WRITE (8,510)
        WRITE (6,1000)
        GO TO 300
      ENDIF
C
C  ** TEMPERATURE PART
C
      TAVR=0.0
      MARKZ=0
      ITCT=0
      CALL INPUT (2,NSTOP,MARKY,MAXNO)
```

```
      IF (NSTOP.GE.1) GO TO 300
      CALL CCDATA (MARKY,NSTOP)
      IF (NSTOP.GE.1) GO TO 300
      IF (MARKY.EQ.1) THEN
        TACC=SPAN*60.0
        CALL CDCH (NSTOP,1)
        IF (NSTOP.GE.1) GO TO 300
      ENDIF
      CALL OUTPUT (3,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
      IF (MARKY.EQ.0) THEN
        WRITE (6,940)
      ELSE
        WRITE (6,950)
      ENDIF
40    NSFLOW=1
      CALL FWCT (NSFLOW,0)
C       IF (NSFLOW.LE.0) THEN
C         CALL ARR (MARKX)
C         CALL BASE (NSTOP)
C         CALL MSLIST
C       ENDIF
      IF (MARKZ.EQ.0) THEN
        CALL CH4EVA
        MARKZ=1
      ENDIF
C
C     DETERMINATION OF CONDITIONS IN THE STARTING JUNCTION.
C
      DO 50 I=1,NJ
        PROP(I)=0.0
        PRCH4(I)=0.0
        IF (JNOL(I).EQ.JSTART) THEN
          MSTART=I
          ISTART=JLR(I)
          T(ISTART)=TSTART
          PRH2OS(ISTART)=HSTART
        ENDIF
50    CONTINUE
70    MRC=0
      MARKC=0
      L=ISTART
      NSTART=MSTART
90    CALL CDENDS (MARKY,NSTART,L,NSTOP)
      IF (NSTOP.GE.1) GO TO 300
      JUMPBK=0
      CALL CDJUNC (NSTART,JUMPBK,L,MOMIT,0)
      IF (JUMPBK.EQ.1) GO TO 90
      CALL RECIRC (MRC,JUMPBK,NSTART,L)
      IF (JUMPBK.EQ.1) GO TO 90
      CALL PREP (MRC,LST)
      IF (MRC.NE.0) THEN
C
C     ITN: USER-SPECIFIED MAX. NUMBER 0F ITERATION IN NETWORK BALANCING
C          (SUB. ITR) AND DATA PREPARATION (TEMP. PART).
C     ITCT: COUNTER OF ITERATIONS.
C
```

```
      ITCT=ITCT+1
      IF (LST.NE.0) THEN
        IF (ITCT.LE.ITN) THEN
          L=ISTART
          NSTART=MSTART
          GO TO 90
        ENDIF
        MARKC=1
      ENDIF
    ENDIF
    CALL TEVAL
C
C   NSFLOW: 0: DATA PREPARATION FOR NETWORK BALANCING IN TEMP. PART
C          HAS NOT BEEN COMPLETED. AIRFLOW REVERSAL EXISTS IN THE
C          SYSTEM.  1: DATA PREPARATION COMLETED.
C   SUMFNV: SUM OF ABSOLUTE DIFFERENCE OF NVP IN SUCCESSIVE INTERVALS.
C
C
C   MADJ: USER-SPECIFIED MAX. NUMBER OF ITERATIONS IN ONE PROG. RUN.
C   MADJC: COUNTER OF ITERATIONS IN ONE PROG. RUN.
C
    IF (MADJC.LT.MADJ) THEN
      MADJC=MADJC+1
      ITCT=0
      MARKN=0
      DO 120 I=1,NB
        IF (JF(I).LT.0) JF(I)=-JF(I)
120   CONTINUE
      IF (NSFLOW.LE.0) THEN
        CALL ARR (MARKX)
        CALL BASE (NSTOP)
        CALL MSLIST
        CALL MBLNC
      ENDIF
      CALL NVP2
      CALL ITR (MARKX,MARKN,NSFLOW,1,MADJC,ITCT)
      WRITE (6,960) MADJC,SUMFNV/MNO
      IF (MADJC.LE.2.AND.MARKY.EQ.1) GO TO 124
      IF (MADJC.EQ.1) GO TO 124
C     ****************************************
      IF ((SUMFNV/MNO).LE.2.E-4) GO TO 130
124   IF (MADJC.GE.(MADJ-9)) THEN
        J=MADJC-MADJ+10
        DO 125 I=1,NB
          QRCD(I,J)=ABS(Q(I))
          IF (JSB(I).NE.IABS(JS(I)).AND.Q(I).GT.0.0)
     .      QRCD(I,J)=-ABS(Q(I))
          IF (JSB(I).EQ.IABS(JS(I)).AND.Q(I).LT.0.0)
     .      QRCD(I,J)=-ABS(Q(I))
          TMRCD(I,J)=TMRD(I)
125     CONTINUE
      ENDIF
      ITCT=0
      DO 30 I=1,NB
        IF (Q(I).LT.0.0) GO TO 40
30    CONTINUE
```

```
      NSFLOW=1
      GO TO 70
    ENDIF
130  CALL HUMI
    IF (MARKY.EQ.0) WRITE (8,620)
    MARKL=0
    CALL FWCT (NSFLOW,0)
    CALL OUTPUT (4,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
    CALL FWCT (NSFLOW,1)
C    DO 140 I=1,NB
C       DCOAGE(I)=-1.0
C140  CONTINUE
    IF (NTEMP.EQ.1) THEN
      WRITE (8,520)
      WRITE (6,1000)
      GO TO 300
    ENDIF
    IF (MARKY.EQ.1) THEN
      WRITE (6,1000)
      GO TO 300
    ENDIF
    MADJC=0
    IF (NSFLOW.EQ.0) THEN
      CALL BASE (NSTOP)
      CALL MSLIST
    ENDIF
C
C ** DYNAMIC SIMULATION PART
C
C   MARKR: 0: OUTPUT WAS GIVEN IN THE LAST INTERVAL.
C          1: OUTPUT WILL BE OMITTED IN THE PRESENT INTERVAL DUE TO
C          INSIGNIFICANCY OF CHANGE. CHANGES IN Q WILL BE CUMULATED.
C   MH: CONTROLLING THE WRITING OF THE HEADLINE FOR THE CONVERGENCE
C       INFORMATION OF OMITTED OUTPUT.
C   ISQC: COUNTER OF AIRWAYS WHICH HAVE THEIR DETAILED DATA RECORDS.
C
    MARKR=0
    MH=0
    MARKX=1
    JDP=0
    INIFAN=0
C   ********
    TACC=0.0
    ISQC=0
    QS=100.0
    WRITE (8,560)
    WRITE (8,550)
    CALL INIT (1)
    WRITE (6,970)
C
C   TINC: USER-SPECIFIED SPAN OF TIME INTERVAL IN SEC..
C   QSTD: A CERTAIN PERCENTAGE OF THE SUM OF ABSOLUTE AIRFLOW CHANGES
C         IN AIRWAYS IN THE FIRST INTERVAL.
C   QS: VARIATION IN SUM OF ABS. AIRFLOW CHANGES IN SUCCESSIVE
C       INTERVALS, CRITERION FOR OUTPUT DURING DYNAMIC SIMULATION.
C   DELTAT: TIME INCREMENT THAT IS ACTUALY ADOPTED IN THE CURRENT
```

```
C          SIMULATION UPDATING.
C
      IF (TINC.LE.0.0) TINC=15.0
      DELTAT=TINC
C
C       ENTERING THE PRIMARY DO-LOOP WITH TIME INCREMENT AS
C     THE ADVANCING STEP.
C
      DO 290 MASTER=1,1000
        IF (TACC.GT.SPAN*60.0) GO TO 300
        IF (MARKR.EQ.0) WRITE (8,560)
C
C     SUB. CDCH UPDATES CONDITION CHANGES ACCORDING TO A USER-SPECIFIED
C     TIME TABLE.
C
        CALL CDCH (NSTOP,0)
        IF (NSTOP.GE.1) GO TO 300
C
C     MARKP: 0: CONVERGENT RESULTS FOR THE PRESENT INTERVAL HAVE NOT
C          BEEN REACHED.  1: CONVERGENT RESULTS OBTAINED.
C     ITT: COUNTER OF ITERATION IN NETWORK BALANCING.
C     IBB: COUNTER OF ITERATION IN DATA PREPARATION.
C
        MARKP=0
        MEX=0
        ITT=0
        IBB=0
        CALL INIT (2)
C
C     MARKS: TIMER FOR DATA HANDLING.     0: ADVANCING OF EXISTING
C          C.V. IN AIRWAYS.  =>1: ADVANCING NEW C.V. INTO AIRWAYS
C          IF NECESSARY, CHECKING TO SEE IF THERE IS ANY C.V. WHICH
c          WAS SET UP IN THE PRESENT INTERVAL AND TRAVELLED MORE
C          THAN ONE AIRWAY.
C     MARKQ: 0: NORMAL CONDITION.  1: SIGNIFICANT CONDITION CHANGES
C          IN JUNCTIONS ARE DETECTED, MORE ITERATIONS ARE NEEDED.
C
160     MARKS=0
C
C     MARKL: 0: NORMAL CONDITION.  1: AIRFLOW REVERSAL FOUND IN THE
C          SYSTEM, RECONSTRUCTION OF MESH FORMATION IS NEEDED.
C
        MARKL=0
        CALL  DTTR (MARKL,MARKP)
        IF (MARKL.NE.0) THEN
          CALL BASE (NSTOP)
          CALL MSLIST
        ENDIF
        DO 165 I=1,NJ
          TLEFT(I)=-0.1
165     CONTINUE
C        DO 168 I=1,NB
C          FBN(I)=0.0
C168     CONTINUE
170     MARKQ=0
        CALL CDJN (ISQC,MARKS,NSTOP)
```

```
      Write (8,*) 'FNVP66',(FNVP(I),i=1,mno)
      IF (NSTOP.GE.1) GO TO 300
      DO 180 I=1,NJ
        SUMQ(I)=0.0
        SUMC(I)=0.0
        SUMM(I)=0.0
        SMHEAT(I)=0.0
        SUMW(I)=0.0
180   CONTINUE
      DO 240 I=1,NB
        IQ=IBTN(I)
        IS=NSACB(I,2)
        IF (IBTN(I).NE.0) THEN
C
C    ADVANCING C.V. IN AIRWAYS.
C
          TMM(I)=TMRD(I)
          IF (MARKS.GE.1) THEN
            CALL ADPT (IS,IQ,MARKS,I)
          ELSE
            TMST=TACC+DELTAT
            CALL KALPHA (TMST,TM,FX,CP,I,1)
            COAGE=BI(I)-FX*BI(I)*BI(I)/(0.375+BI(I))
            DCOAGE(I)=COAGE
            XNEW(I)=HK(I)*O(I)*O(I)*COAGE/(120.*DR*QQ(I)*A(I))
            CALL ADPT (IS,IQ,MARKS,I)
          ENDIF
C
C     SUB. WDT EVALUATES PARAMETERS AT STATIONS IN THE AIRWAYS
C    IN WHICH SIGNIFICANT CONDITION CHANGES HAPPENED FOR THE
C    EVALUATION OF TEMP. CORR. IN WET CASES.
C
          IF (NHM.GT.0) CALL WDT (I)
        ELSE
C
C    RHH2O: HUMIDITY RATIO AT AIRWAY ENDING.
C
          IF (NHM.GT.0.AND.NWTYP(I).EQ.10) THEN
            DO 190 IJ=1,INFLOW
              IF (NCENT(IJ).EQ.NO(I)) THEN
                RDH2O(I)=GFS(IJ)
                GO TO 200
              ENDIF
190         CONTINUE
          ENDIF
C
C     EVALUATION OF CH4 CONCENTRATION AT THE END OF AIRWAYS WHICH
C    HAVE NO DETAILED DATA RECORDS.
C
200       DO 215 IJ=1,NJ
            KN=JLR(IJ)
            IF(JS(I).EQ.JNO(KN)) THEN
              DST=QQ(I)*DELTAT/(A(I)*60.0)
              IF (DST.GT.LA(I)) DST=LA(I)
              CH4B=(RCH4(I)-PRCH4S(KN))*(LA(I)-DST)/LA(I)+
     .          PRCH4S(KN)
```

```
              RDCH4(I)=(CH4B*QQ(I)+CH4V(I)*DST/LA(I))/
     .                 (QQ(I)+CH4V(I)*DST/LA(I))
            ENDIF
215       CONTINUE
        ENDIF
C        TTB=(TRD(I)+TTRD(I))/2.0
C        IF (NSAC(I,2).GE.1) TTB=TRD(I)*FBN(I)+TTRD(I)*(1.0-FBN(I))
        TTB=TRD(I)
        JJ=IABS(JF(I))
        DO 220 J=1,NJ
          IF (JNO(J).EQ.JJ) THEN
            SUMQ(J)=SUMQ(J)+QQ(I)
            SUMC(J)=SUMC(J)+QQ(I)*RDPROP(I)
            SUMM(J)=SUMM(J)+QQ(I)*RDCH4(I)
            SMHEAT(J)=SMHEAT(J)+(0.2376*TTB+1.2E-5*TTB*TTB)*QQ(I)
            SUMW(J)=SUMW(J)+QQ(I)*RDH2O(I)
            GO TO 240
          ENDIF
220     CONTINUE
240   CONTINUE
      DO 245 I=1,NJ
        JCH(I)=0
245   CONTINUE
C
C
    DO 250 I=1,NJ
      IF (JNOL(I).EQ.JSTART.OR.(JNOL(I).GE.900.AND.JNOL(I).LE.950))
     .  GO TO 250
C
C     UPDATING THERMAL STATES IN JUNCTIONS EXCEPT FOR THE USER-
C    SPECIFIED STARTING JUNCTION (USUALLY THE ATMOSPHERE).
C
      K=JLR(I)
      MARKW=0
      TPP=T(K)
      PRPP=PROP(K)
      PRCPP=PRCH4(K)
      PRPH2O=PRH2O(K)
C
C    PROP(K): CONTENT OF CONTAMINANTS IN AIRFLOW.
C    PRCH4(K): CONTENT OF CH4 IN AIRFLOW.
C    PRH2O(K): CONTENT OF WATER VAPOR IN AIRFLOW.
C    T(K): TEMPERATURE OF AIRFLOW.
C    THE ABOVE PARAMETERS ARE EXPRESSED AS MASS RATIOS WITH RESPECT
C    TO THAT OF AIR CURRENT.
C
      PROP(K)=SUMC(K)/SUMQ(K)
      PRCH4(K)=SUMM(K)/SUMQ(K)
      PRH2O(K)=SUMW(K)/SUMQ(K)
      PTCONT=9.801E7+2.0*SMHEAT(K)/(SUMQ(K)*2.4E-5)
      T(K)=-9900.+SQRT(PTCONT)
C
C     MARKW=1 INDICATES THAT ALL THE DOWN-STREAM AIRWAYS OF JUNCTION
C    JNO(K) NEED TO HAVE THEIR DETAILED DATA RECORDS.
C
      IF (ABS(TPP-T(K)).GE.1.0) MARKW=1
```

```
      IF (ABS(PRPP-PROP(K)).GE.1.E-6) MARKW=1
C                           **
      IF (ABS(PRCPP-PRCH4(K)).GE.1.E-4) MARKW=1
C                           **
      IF (MARKW.EQ.1) JCH(K)=1
      IF (MARKW.EQ.1) MARKQ=1
250  CONTINUE
C
C
      IF (MARKS.LT.1.OR.MARKQ.NE.0) THEN
C
C     DATA PREPARATION HAS NOT BEEN COMPLETED, MORE ITERATIONS ARE
C    NEEDED.
C
        IBB=IBB+1
        IF (IBB.LE.7) THEN
          MARKS=MARKS+1
          GO TO 170
        ELSE
          WRITE (8,580) TACC
        ENDIF
      ENDIF
      CALL PARAM
      CALL NVP2
      ITT=ITT+1
      ITCT=0
C      *******
      CALL ITR (MARKX,MARKN,NSFLOW,1,MADJC,ITCT)
      IF (ITT.GE.(ITN-9)) THEN
        J=ITT-ITN+10
        DO 260 I=1,NB
          QRCD(I,J)=ABS(Q(I))
          IF (JSB(I).NE.IABS(JS(I)).AND.Q(I).GT.0.0)
     .      QRCD(I,J)=-ABS(Q(I))
          IF (JSB(I).EQ.IABS(JS(I)).AND.Q(I).LT.0.0)
     .      QRCD(I,J)=-ABS(Q(I))
          TMRCD(I,J)=TMRD(I)
260     CONTINUE
      ENDIF
      IF (ITT.GE.ITN.OR.(SUMFNV/MNO.LE.2.E-4.AND.ITT.GE.2)) THEN
C
C     DATA HANDLING AFTER NETWORK IS BALANCED.
C
        WRITE (6,990) ITT,SUMFNV/MNO
        CALL EVHT (MARKX,MARKP,MARKL,MARKR,QSUM,QS)
      ELSE
C
C     INTERMEDIATE DATA HANDLING BETWEEN INTERVALS.
C
        WRITE (6,990) ITT,SUMFNV/MNO
        CALL DISP (MARKX,ISQC,IBB)
        GO TO 160
      ENDIF
      IF (MASTER.LE.1) QSTD=QSUM/7.
      WRITE (6,980) TACC
      IF (TACC.LT.(SPAN*60.)) THEN
```

```
        IF (QSUM.LE.QSTD.AND.ITT.LT.ITN.AND.MARKR.LE.5.
     .      AND.IOUT.GT.(-2)) THEN
C
C      CALCULATION FOR THE PRESENT INTERVAL IS COMPLETED BUT NO
C     OUTPUT IS NEEDED.
C
          MARKR=MARKR+1
          TACC1=TACC/60.0
          IF (MH.EQ.0) THEN
            MH=1
            WRITE (8,590)
          ENDIF
          WRITE (8,600) SUMFNV/MNO,ITT,TACC1
        ELSE
          CALL FWCT (NSFLOW,0)
          MARKR=0
          MH=0
          CALL OUTPUT (4,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
          CALL FWCT (NSFLOW,1)
          IF (IOUT.LE.(-2)) THEN
            CALL OUTPUT (5,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
            IF (NHM.GT.0)
     .        CALL OUTPUT (7,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
          ENDIF
          IF (IOUT.LE.(-1))
     .      CALL OUTPUT (6,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
        ENDIF
      ELSE
C
C      TIME SPAN FOR DYNAMIC SIMULATION IS EXCEEDED, QUASI-EQUIL.
C     SIMULATION IS GOING TO PERFORM.
C
        CALL FWCT (NSFLOW,0)
        CALL OUTPUT (4,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
        CALL FWCT (NSFLOW,1)
        IF (IOUT.LE.(-2)) THEN
          CALL OUTPUT (5,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
          IF (NHM.GT.0)
     .      CALL OUTPUT (7,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
        ENDIF
        CALL OUTPUT (6,MARKX,MARKN,MARKC,MARKY,ITT,MADJC)
        MARKX=0
        MARKY=1
        DO 285 I=1,NB
          TROCK(I)=0.0
          NNREV(I)=0
          DZRD(I)=0.0
285     CONTINUE
        DO 286 I=1,IMX
          NSKP(I)=0
286     CONTINUE
        DO 288 I=1,NJ
          PROP(I)=0.0
          PRCH4(I)=0.0
288     CONTINUE
        DO 289 I=1,LMX
```

```
          MSL(I)=0
289       CONTINUE
        WRITE (8,630)
        GO TO 1
      ENDIF
290  CONTINUE
C
300  STOP
C
510  FORMAT (//,1X,'NO FURTHER CALCULATIONS BEYOND NETWORK ',
   .     'BALANCING WERE',/,' DEMANDED DUE TO NETW=1, PROG. ',
   .     'RUN COMPLETED.',/)
520  FORMAT (//,1X,'NO FURTHER CALCULATIONS BEYOND TEMP. ',
   .     'CALCULATIONS WERE',/,' DEMANDED DUE TO NTEMP=1, PROG.',
   .     ' RUN COMPLETED.',/)
550  FORMAT (/,T19,'NON-STEADY STATE SIMULATION OUTPUT',/)
560  FORMAT (/,72('*'),/)
570  FORMAT (//,1X,'AIRFLOW OSCILLATION IN A PORTION OF THE SYSTEM',
   .      1X,'MAY HAVE HAPPENED.',/,' THE FOLLOWING IS AN ',
   .      'AUXILIARY OUTPUT FOR THE',I3,'TH ITERATION,'/)
580  FORMAT (//,1X,'* ATTN *  SIMULATION STEP DELTAT NEEDS TO BE ',
   .      'SHORTENED AT AROUND',F7.0,/,' SEC. AFTER EVENT FOR A',
   .      'BETTER ACCURACY.',/)
590  FORMAT (//,1X,'OUTPUT FOR THE FOLLOWING TIME INCREMENTS',
   .     ' WERE OMITTED DUE TO',/,' INSIGNIFICANT CONDITION',
   .     ' CHANGES:',//,'  SUM OF NVP CORR.',T30,'ITERATIONS',T50,
   .     'TIME (MIN.)',/)
600  FORMAT (3X,F10.4,T31,I5,T50,F7.2)
620  FORMAT (//,72('*'),///,T18,'OUTPUT OF THE TEMPERATURE',
   .     ' PART',///,72('*'))
630  FORMAT (//,72('*'),//,T14,'QUASI-EQUILIBRIUM STATE',
   .      ' SIMULATION OUTPUT',//,72('*'))
680  FORMAT (,1X,'NAME YOUR INPUT FILE  "D:\PATH\FNAME1.EXT".',/)
900  FORMAT (,T19,10('-'),' MFIRE 2.01 APRIL 1994',11('-'),/,T20,'A PRO
   .GRAM FOR DYNAMIC SIMULATION OF MINE',/,T21,'VENTILATION AND FIRE I
   .NTERACTIONS',/,T16,'WITH WATER EVAPORATION AND CONDENSATION CONSID
   .ERED',//,T18,'WRITTEN BY:  DR. X. CHANG AND DR. R. GREUER',/,T10,'
   . U.S. BUREAU OF MINES AND MICHIGAN TECHNOLOGICAL UNIVERSITY',//)
910       FORMAT (T35,'DISCLAIMER',/,T10,'The Bureau of Mines express
   .ly declares that there are no',/,T10,'warranties express or implie
   .d which apply to the software',/,T10,'contained herein. By accepta
   .nce and use of said software',/,T10,'which is conveyed to the user
   . without consideration by the',/,T10,'Bureau of Mines, the user he
   .rof expressly waives any and',/,T10,'all claims for damage and/or
   .suits for or by reason of',/,T10,'personal injury, or property dam
   .age, including special',/,T10,'consequential or other similar dama
   .ages arising out of or',/,T10,'in any way connected with the use o
   .f the software',/,T10,'contained herein.',)
920  FORMAT (//,' DATA INPUT COMPLETED.')
930  FORMAT (//,' NETWORK CALCULATIONS COMPLETED.')
940  FORMAT (//,' TEMPARATURE PART STARTED',/,' ACCURACY CRITERIA = ',
   .      '0.0002 IN.W.G. PER MESH.',//)
950  FORMAT (//,' STEADY STATE PART STARTED',/,' ACCURACY CRITERIA =',
   .      ' 0.0002 IN.W.G. PER MESH.',//)
960  FORMAT (5X,'ITERATION ',I4,' COMPLETED, CURRENT ERROR ',F9.5)
970  FORMAT (//,' NON-STEADY STATE SIMULATION PART STARTED.',/,
```

```
     .      ' ACCURACY CRITERIA = 0.0002 IN.W.G. PER MESH.',//)
980  FORMAT (/,5X,'SIMULATION GOES ON, ',F7.1,' SEC PASSED.',/)
990  FORMAT (10X,'ITERATION ',I4,' COMPLETED, CURRENT ERROR ',F9.5)
1000 FORMAT (//,' PROGRAM RUN COMPLETED. THE OUTPUT IS SAVED IN ',
     .      'FILE "WFOUT".',/)
C
      END
C
C
      BLOCK DATA
C
C     ---------------------------------------------------------------
C
C     INITIATION OF THE VARIABLES AND ARRAYS IN COMMON BLOCKS.
C
C     ---------------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
      PARAMETER (KMX=IMX*IMY, KMY=NMZ*10, KMZ=NMZ*10*5, KNX=NMX*2,
     .          KNY=IMX*3, KNZ=NMX*10, KLX=IMX*20*4)
C
      DATA  NO,JS,JF,NWTYP,R,Q,P,KF,LA,A,O,RSTD,DZRD,FRNVP,NREV,RDCH4,
     .      RDPROP,TRD,TJS,RDOP,RCH4,FFRNVP,RA,NNREV,RDH2O,RH2O
     .      /NMX*0,NMX*0,NMX*0,NMX*0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0,
     .      NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0,
     .      NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,
     .      NMX*0.0,NMX*0,NMX*0.0,NMX*0.0/
      DATA  NOF,NFREG,RGRAD,NFCW,MPTS,QF,PF,NSKP,NEGQ,JSB
     .      /IMX*0,IMX*0,IMX*0.0,IMX*0,IMX*0,KMX*0.0,KMX*0.0,IMX*0,
     .      IMX*0,NMX*0/
      DATA  MNO,MEND,MSL,FNVP,RQ,INU,KNO,KJS,KJF
     .      /0,NMY*0,LMX*0,NMY*0.0,NMX*0.0,NMX*0,NMX*0,NMX*0,NMX*0/
      DATA  JNO,T,Z,CH4C,JNOL,PROP,PRCH4,JLR,HJN,PRH2O
     .      /NMY*0,NMY*0.0,NMY*0.0,NMY*0.0,NMY*0,NMY*0.0,NMY*0.0,NMY*0,
     .      NMY*0.0,NMY*0.0/
      DATA  TROCK,TMRD,HA,HK
     .      /NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0/
      DATA  NCENT,CH4V,CH4PA,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,TFSI,CN,
     .      HN,O2BEH,TPR,HTAD,QCENT
     .      /NMX*0,NMX*0.0,NMX*0.0,IMX*0.0,IMX*0.0,IMX*0.0,IMX*0.0,
     .      IMX*0.0,IMX*0.0,IMX*0.0,IMX*0.0,IMX*0.0,IMX*0.0,IMX*0.0,
     .      IMX*0.0,IMX*0.0/
      DATA  NGIN,MIN,NGOUT,LOUT
     .      /NMX*0,NMX*0,NMX*0,NMX*0/
      DATA  MEMREC,NOREC,ESTPR,ESTCH4,ESTTR
     .      /NMX*0,NMX*0,NMX*0.0,NMX*0.0,NMX*0.0/
      DATA  XNEW,DCOAGE,WTCORR,NSEQS
     .      /NMX*0.0,NMX*0.0,KMY*0.0,NMZ*0/
      DATA  TIME,TSTART,JSTART,TLEFT
     .      /0.0,0.0,0,NMX*0.0/
      DATA  SUMQ,SUMC,SUMM,SMHEAT,SUMW
     .      /NMY*0.0,NMY*0.0,NMY*0.0,NMY*0.0,NMY*0.0/
      DATA  PROPS,PRCH4S,QQ,QTP,TTJS,TTRD,TAS,TMM,BI,CNT,EXPN,FAW
     .      /NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,
     .      NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0,NMX*0.0/
```

```
      DATA  PRH2OS,HSTART,PSI,NSP,STV,HABS,NSEQE,TDA,WMTR,GFF
     .    /NMY*0.0,0.0,NMX*0.0,NMX*0,KMY*0.0,KMY*0.0,NMZ*0,KMY*0.0,
     .    KMY*0.0,KMY*0.0/
      DATA  DELTAT,SUMFNV
     .    /0.0,0.0/
      DATA  IBTN,DPPA,NSAC,TMRDA,DELTAQ,DPPB,NSACB,GBTN,JCH,TAUXA,
     .    TAUXB,TAUXC,TAUXD,FMASS
     .    /NMX*0,KMZ*0.0,KNX*0,NMX*0.0,NMX*0.0,KMZ*0.0,KNX*0,50*0.0,
     .     NMY*0,KMY*0.0,KMY*0.0,NMZ*0.0,NMZ*0.0,NMZ*0.0/
      DATA  GFS,TACC,NPP,HABS1
     .    /IMX*0.0,0.0,NMZ*0,KMY*0.0/
      DATA  JDP,JDPP
     .    /0,IMX*0/
      DATA  WRNPR,WRNGS,WRNSM,WRNHT,IOMIT,ROMIT,IAC
     .    /4*0.0,IMX*0,KNY*0.0,0/
      DATA  QRCD,TMRCD
     .    /KNZ*0.0,KNZ*0.0/
      DATA  FC,DK,FK,FKQ
     .    /400*0.0,20*0.0,20*0.0,KLX*0.0/
C
C
      END


C
C
      SUBROUTINE OUTPUT (IPT,MARKX,MARKC,MARKN,MARKY,ITT,MADJC)
C
C   ------------------------------------------------------------------
C
C     SUBROUTINE PURPOSE:
C     OUTPUT OF NORMAL CALCULATION RESULTS.
C
C   ------------------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
      DIMENSION HBN(NMX),IRE(NMX),LST(NMX),TQR(NMX),HFN(5),QFN(5)
C
C ** CALCULATION OF VARIATION IN AIR STATIC PRESSURES IN AIRWAYS.
C
      IF (IPT.NE.1) THEN
        PP=0.0
        DO 10 L=1,NB
          IF (NWTYP(L).LE.0.OR.NWTYP(L).EQ.10) THEN
            P(L)=R(L)*Q(L)*ABS(Q(L))*1.E-10
          ELSE
            P(L)=R(L)
          ENDIF
          PP=PP+P(L)
10      CONTINUE
        PCRIT=PP/FLOAT(NB)
        IF ((ITT.GE.ITN.OR.MADJC.GE.MADJ).AND.IPT.EQ.4) THEN
          DO 1520 I=1,NB
            TTMN=0.0
            DO 1500 J=1,10
```

```
          TTMN=TTMN+TMRCD(I,J)
1500     CONTINUE
        TTMN=TTMN/10.0
        TQR(I)=0.0
        DO 1510 J=1,10
          TQR(I)=TQR(I)+(TMRCD(I,J)-TTMN)**2
1510     CONTINUE
1520   CONTINUE
      IZ=0
      LL=0
1530   II=0
      DO 1540 I=1,NB
        IF (NO(I).GT.0) THEN
          IF (TQR(I).GT.TSR.OR.II.EQ.0) THEN
            TSR=TQR(I)
            II=I
          ENDIF
        ENDIF
1540   CONTINUE
      IF (II.EQ.0) GO TO 1590
      IF (LL.GT.0) THEN
        KK=0
        DO 1550 J=1,LL
          K=LST(J)
          IF (TQR(II).GT.TQR(K)) KK=KK+1
1550     CONTINUE
        IF (KK.GE.(LL-2).AND.TQR(II).GT.20.0) GO TO 1560
        GO TO 1590
      ENDIF
1560   LL=LL+1
      LST(LL)=II
      NO(II)=-NO(II)
      JSS=JS(II)
      JFF=JF(II)
1570   DO 1580 I=1,NB
        IF (NO(I).GT.0) THEN
          IF ((JS(I).EQ.JSS.OR.JF(I).EQ.JSS.OR.JS(I).EQ.JFF.OR.
     .      JF(I).EQ.JFF).AND.(TQR(I).GT.20.0)) THEN
            LL=LL+1
            LST(LL)=I
            NO(I)=-NO(I)
            IF (JS(I).EQ.JSS) THEN
              JSS=JF(I)
            ELSE IF (JF(I).EQ.JSS) THEN
              JSS=JS(I)
            ELSE IF (JS(I).EQ.JFF) THEN
              JFF=JF(I)
            ELSE
              JFF=JS(I)
            ENDIF
            GO TO 1570
          ENDIF
        ENDIF
1580   CONTINUE
      IZ=IZ+1
      IF (IZ.LE.3) GO TO 1530
```

```
1590    ICRT=0
        DO 1598 I=1,LL
          K=LST(I)
          DO 1596 J1=2,9
            ERR1=ABS(TMRCD(K,J1-1)-TMRCD(K,J1))
            ERR2=ABS(TMRCD(K,J1)-TMRCD(K,J1+1))
            RATIO=ERR2/ERR1
            IF (RATIO.GT.0.99) ICRT=1
1596      CONTINUE
1598    CONTINUE
        IF (ICRT.EQ.0) THEN
          ICYCLE=-1
          GO TO 1635
        ENDIF
        CRT1=0.0
        CRT2=0.0
        CRT3=0.0
        DO 1630 I=1,LL
          K=LST(I)
          NO(K)=IABS(NO(K))
          CRT=0.0
          DO 1600 J=10,3,-1
            CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-2))**2
1600      CONTINUE
          CRT1=CRT1+CRT/8.0
          CRT=0.0
          DO 1610 J=10,4,-1
            CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-3))**2
1610      CONTINUE
          CRT2=CRT2+CRT/7.0
          CRT=0.0
          DO 1620 J=10,5,-1
            CRT=CRT+(TMRCD(K,J)-TMRCD(K,J)-TMRCD(K,J-4))**2
1620      CONTINUE
          CRT3=CRT3+CRT/6.0
1630    CONTINUE
        CRT1=CRT1/LL
        CRT2=CRT2/LL
        CRT3=CRT3/LL
        LSG=LL
        IF (AMIN1(CRT1,CRT2,CRT3)*10.0.LE.MAX(CRT1,CRT2,CRT3)) THEN
          ICYCLE=2
          IF (CRT2.LT.0.95*CRT1) ICYCLE=3
          IF (CRT3.LT.0.9*CRT1.AND.CRT3.LT.CRT2) ICYCLE=4
        ELSE
          ICYCLE=0
        ENDIF
1635    CONTINUE
      ENDIF
    ENDIF
C
    IF (IPT.EQ.1) THEN
C
C  ** LIST OF INPUT DATA ON NETWORK STRUCTURE.
C
      IF (MARKY.EQ.0) WRITE (8,285) TR,DR,NHM
```

```
      DO 9 I=1,NMX
        HBN(I)=0.0
        IRE(I)=0
        LST(I)=0
        TQR(I)=0.0
9      CONTINUE
      IF ((IOUT.NE.0.AND.MARKY.EQ.0).OR.IOUT.LE.(-1)) THEN
        WRITE (8,490)
        WRITE (8,500) (NO(K),JS(K),JF(K),Q(K),NWTYP(K),LA(K),
     .            A(K),R(K),KF(K),O(K),K=1,NB)
        IF (NVPN.LE.0) THEN
          WRITE (8,980)
          WRITE (8,990) (JNO(K),T(K),Z(K),CH4C(K),K=1,NJ)
        ENDIF
      ENDIF
C
    ELSE IF (IPT.EQ.2) THEN
C
C  ** OUTPUT OF THE NETWORK PART.
C
      WRITE (8,110)
      WRITE (8,120)
      L=0
      DO 20 K=1,NB
        IF (NWTYP(K).GT.0) THEN
          L=L+1
        ELSE
          WRITE (8,130) NO(K),JS(K),JF(K),NWTYP(K),R(K),Q(K),P(K)
        ENDIF
20     CONTINUE
      IF (L.LE.0) THEN
        WRITE (8,140)
      ELSE
        WRITE (8,150)
        DO 30 K=1,NB
          IF (NWTYP(K).LE.0) GO TO 30
          WRITE (8,160) NO(K),JS(K),JF(K),Q(K),P(K)
30       CONTINUE
        IF (NFNUM.GT.0) THEN
          WRITE (8,170) (NOF(K),K=1,NFNUM)
          WRITE (8,175)
          DO 40 K=1,NFNUM
            L=MPTS(K)
            WRITE (8,180) (QF(K,I)/1000.0,PF(K,I),I=1,L)
40         CONTINUE
          IF (IOUT.LE.(-2).AND.MARKY.LE.0) THEN
            WRITE (8,1040)
            DO 45 K=1,NFNUM
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
              INDEX=MPTS(K)
              N1A=NCOF(K)
              IF (INDEX.GT.1.OR.N1A.GT.1) THEN
                IF (NSWT(K).EQ.1) THEN
                  WRITE (8,1044) NOF(K)
C                   DO 42 K1=1,N1A
                   WRITE (8,1046) (COF(K,II),II=1,N1A)
```

```
                  WRITE (8,*) (COF(K,II),II=1,N1A)
C42               CONTINUE
                ELSEIF (NSWT(K).EQ.2) THEN
                 WRITE (8,1045) NOF(K)
                 DO 43 K1=1,INDEX-1
                  WRITE (8,1050) (FKQ(K,K1,K2),K2=1,4),
     .                 QF(K,K1),QF(K,K1+1)
43               CONTINUE
                ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
                WRITE (8,1085)
                DO 441 K1=1,INDEX-1
                 QZZ=(QF(K,K1+1)-QF(K,K1))/5.0
                 DO 44 K2=1,5
                  QZZ1=QF(K,K1)+(K2-1)*QZZ
                  NFCWZ=NFCW(K)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
                  IF (NSWT(K).EQ.2) THEN
                   CALL SPLINE (K,QZZ1,HZZ,TR,1)
                  ELSEIF (NSWT(K).EQ.1) THEN
                   CALL LSFAN (K,QZZ1,HZZ,TR,1,0)
                  ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
                  NFCW(K)=NFCWZ
                  HFN(K2)=HZZ
                  QFN(K2)=QZZ1
44               CONTINUE
                 WRITE (8,1090) (QFN(K4)/1000.,HFN(K4),K4=1,5)
441             CONTINUE
               ENDIF
45            CONTINUE
             ENDIF
            ENDIF
           ENDIF
           WRITE (8,190) NB,NJ
           IF (NFNUM.GT.0) THEN
            DO 50 K=1,NFNUM
             DO 46 KK=1,NB
              IF (NO(KK).EQ.NOF(K)) GO TO 47
46           CONTINUE
             GO TO 50
47           IF (NFCW(K).GT.0) WRITE (8,200) NOF(K)
             IF ((Q(KK).LT.0.0.AND.IABS(JS(KK)).EQ.JSB(KK)).OR.
     .        (Q(KK).GT.0.0.AND.IABS(JS(KK)).NE.JSB(KK)))
     .        WRITE (8,840) NOF(K)
50          CONTINUE
           ENDIF
          ELSE IF (IPT.EQ.3) THEN
C
C ** LIST OF ADDITIONAL INFORMATION AND INPUT DATA FOR THE TEMP. AND
C    DYNAMIC SIMULATION PARTS OF MAIN PROGRAM.
C
           WRITE (8,1230)
           IF (IOUT.NE.0) THEN
            IF (NHM.GE.1) THEN
             WRITE (8,210)
```

```
          WRITE (8,220) (NO(I),JS(I),JF(I),DZRD(I),TROCK(I),PSI(I),
     .               CH4V(I),HK(I),HA(I),I=1,NB)
        ELSE
          WRITE (8,800)
          WRITE (8,810) (NO(I),JS(I),JF(I),DZRD(I),TROCK(I),
     .               CH4V(I),HK(I),HA(I),I=1,NB)
        ENDIF
      ENDIF
      WRITE (8,240) TSTART,JSTART
      IF (IOUT.NE.0) THEN
        IKL=0
        DO 14 I=1,NJ
          IF (JNO(I).NE.JSTART.AND.JNO(I).GE.900.AND.JNO(I).LE.950)
     .      IKL=1
14      CONTINUE
        IF (IKL.EQ.1) THEN
          WRITE (8,1210)
          DO 15 I=1,NJ
            IF (JNO(I).NE.JSTART.AND.JNO(I).GE.900.
     .      AND.JNO(I).LE.950) THEN
              WRITE (8,1220) JNO(I),T(I)
            ENDIF
15        CONTINUE
        ENDIF
      ENDIF
      IF (INFLOW.GT.0) THEN
        WRITE (8,250)
        WRITE (8,260) (NCENT(I),CONT(I),CONC(I),HEAT(I),O2MIN(I),
     .             SMPO2(I),HTPO2(I),TPR(I),QCENT(I),I=1,INFLOW)
      ELSE
        WRITE (8,270)
      ENDIF
      IAC=0
    ELSE IF (IPT.EQ.4) THEN
C
C ** OUTPUT OF THE TEMPERATURE PART.
C
      IF (MARKX.GE.1.AND.MARKY.EQ.0) THEN
        IF (IAC.GT.0) THEN
          WRITE (8,860)
          DO 51 I=1,IAC
            I1=IOMIT(I)
            WRITE (8,870) NO(I1),JS(I1),ROMIT(I,1),ROMIT(I,2),
     .                 ROMIT(I,3)
51        CONTINUE
          IAC=0
        ENDIF
      ENDIF
      IF (MARKX.GE.1) THEN
        ITL=ITT
        JTL=ITN
      ELSE
        ITL=MADJC
        JTL=MADJ
      ENDIF
      IF (ITL.GE.JTL) THEN
```

```
      WRITE (8,690) SUMFNV/MNO
      KTL=0
      DO 52 I=1,NB
        NREVA=NREV(I)
        IF (MARKX.EQ.0) NREVA=NNREV(I)
        IF (NREVA.GE.JTL/3) THEN
          IF (KTL.EQ.0) THEN
            WRITE (8,700)
            KTL=1
          ENDIF
          IF (NWTYP(I).EQ.1) THEN
            PP=R(I)
          ELSE
            PP=R(I)*Q(I)*ABS(Q(I))*1.E-10
          ENDIF
          WRITE (8,710) NO(I),Q(I),PP,NREVA,ITL
        ENDIF
52     CONTINUE
    ELSE
      IF (SUMFNV/MNO.LE.2.E-4) WRITE (8,770) SUMFNV/MNO,ITL
      IF (SUMFNV/MNO.GT.2.E-4) WRITE (8,780) SUMFNV/MNO,ITL
    ENDIF
    IF (ITT.GE.ITN.OR.MADJC.GE.MADJ) THEN
      IZ=MADJ-9
      IF (MARKX.GT.0.AND.MARKY.EQ.0) IZ=ITN-9
      IF (ICYCLE.LE.0) THEN
        WRITE (8,890) (IJ,IJ=IZ,IZ+9)
      ELSE
        WRITE (8,900) ICYCLE,(IJ,IJ=IZ,IZ+9)
      ENDIF
      DO 1640 I=1,LSG
        K=LST(I)
        IF (JSB(K).EQ.JS(K)) THEN
          JSS=JS(K)
          JFF=JF(K)
        ELSE
          JSS=JF(K)
          JFF=JS(K)
        ENDIF
        WRITE (8,910) NO(K),JSS,JFF,(TMRCD(K,J),J=1,10)
        WRITE (8,920) (QRCD(K,J)/1000.0,J=1,10)
1640   CONTINUE
    ENDIF
    IF (MARKN.GT.0) WRITE (8,280) ITL
    IF (MARKC.GT.0) WRITE (8,290) ITL
    IF (MARKX.LE.0) THEN
      TT1 = TIME*3600.0
      IF (TACC.GT.TINC) THEN
        WRITE (8,520) TIME
      ENDIF
      WRITE(8,300)
      WRITE (8,310) (NO(I),IABS(JS(I)),IABS(JF(I)),Q(I),TMRD(I),
                TRD(I),RDPROP(I),RDCH4(I),P(I),I=1,NB)
      WRITE (8,650)
      WRITE (8,660) (JNO(I),T(I),PROP(I),PRCH4(I),I=1,NJ)
    ELSE
```

```
C
C  ** OUTPUT OF THE DYNAMIC SIMULATION PART.
C
        WRITE (8,320) TACC
C
C    EVALUATION OF RELATIVE HUMIDITY AT AIRWAY ENDING.
C
        IF (NHM.GT.0) THEN
          DO 55 I=1,NB
            TY=TRD(I)
            IF (TY.GT.210.0) TY=210.0
            IF (TY.LE.122.0) THEN
              HU=-1.0682+0.07946*TY-0.00129*TY*TY+1.26E-5*TY**3
            ELSE
              HU=-33.786+0.97*TY-9.305E-3*TY*TY+3.65E-5*TY**3
            ENDIF
            PV=RDH2O(I)*101.35/(RDH2O(I)+0.622)
            HBN(I)=PV/HU
            IF (HBN(I).GE.1.0.AND.TY.GE.210.0) HBN(I)=1.E10
55         CONTINUE
          WRITE (8,330)
          WRITE (8,340) (NO(I),INT(DELTAQ(I)),INT(Q(I)),TMRD(I),
     .                TRD(I),RDPROP(I),RDCH4(I),RDH2O(I),HBN(I),
     .                P(I),I=1,NB)
        ELSE
          WRITE (8,750)
          WRITE (8,760) (NO(I),JS(I),JF(I),DELTAQ(I),Q(I),TMRD(I),
     .                TRD(I),RDPROP(I),RDCH4(I),P(I),I=1,NB)
        ENDIF
        LI=0
        DO 60 I=1,NB
          IF (NSAC(I,1).EQ.1) THEN
            LK=IBTN(I)
            LM=0
            IF (DPPA(LK,2,3).GE.1.E-4) LM=2
            IF (DPPA(LK,1,3).GE.1.E-4) LM=1
            IF (LM.GT.0) THEN
              IF (LI.NE.1) THEN
                LI=1
                WRITE (8,350)
              ENDIF
              WRITE (8,360) NO(I),DPPA(LK,LM,1),JS(I),
     .              DPPA(LK,LM,2),DPPA(LK,LM,3),DPPA(LK,LM,4)
            ENDIF
          ENDIF
60       CONTINUE
        IF (INFLOW.GT.0) THEN
          IF (NHM.GT.0) THEN
            WRITE (8,610)
          ELSE
            WRITE (8,820)
          ENDIF
          DO 135 LRW=1,INFLOW
            DO 132 JRW=1,NB
              IF (NCENT(LRW).EQ.NO(JRW)) THEN
                IF (NHM.GT.0) THEN
```

```
                    WRITE (8,620) NO(JRW),Q(JRW),TJS(JRW),
     .                  RDPROP(JRW),O2BEH(LRW),RDH2O(JRW)
     .                  ,HTAD(LRW)
                  ELSE
                    WRITE (8,830) NO(JRW),Q(JRW),TJS(JRW),
     .                       RDPROP(JRW),O2BEH(LRW),
     .                       HTAD(LRW)
                  ENDIF
                  GO TO 135
                ENDIF
132           CONTINUE
135         CONTINUE
          ENDIF
C
C     EVALUATION OF RELATIVE HUMIDITY IN JUNCTIONS.
C
          IF (NHM.GT.0) THEN
            DO 65 I=1,NJ
              TSS=T(I)
              IF (TSS.GT.210.0) TSS=210.0
              IF (TSS.LE.122.0) THEN
                HU=-1.0682+0.07946*TSS-0.00129*TSS*TSS+
     .            1.26E-5*TSS**3
              ELSE
                HU=-33.786+0.97*TSS-9.305E-3*TSS*TSS+3.65E-5*TSS**3
              ENDIF
              PV=PRH2O(I)*101.35/(PRH2O(I)+0.622)
              HJN(I)=100.0*PV/HU
              IF (HJN(I).GT.100.0) HJN(I)=100.0
65          CONTINUE
            WRITE (8,370)
            WRITE (8,380) (JNO(I),T(I),PROP(I),PRCH4(I),
     .              PRH2O(I),HJN(I),I=1,NJ)
          ELSE
            WRITE (8,650)
            WRITE (8,660) (JNO(I),T(I),PROP(I),PRCH4(I),I=1,NJ)
          ENDIF
        ENDIF
        IF (NFNUM.GT.0) THEN
          DO 68 K=1,NFNUM
            IF (NFCW(K).GT.0.OR.NEGQ(K).GT.0) WRITE (8,200) NOF(K)
            IF (NEGQ(K).GT.0) WRITE (8,840) NOF(K)
68        CONTINUE
        ENDIF
        L=0
        DO 70 I=1,NB
          IF (JF(I).LT.0.OR.JS(I).LT.0) THEN
            IF (L.EQ.0) WRITE (8,720)
            L=L+1
            IRE(L)=NO(I)
          ENDIF
70      CONTINUE
        IF (L.GT.0) WRITE (8,730) (IRE(I),I=1,L)
        KTL=0
        DO 90 I=1,NB
          IF (MARKX.EQ.0) THEN
```

```
            NNREVA=NNREV(I)
          ELSE
            NNREVA=NREV(I)
          ENDIF
          IF ((NNREVA/2)*2.NE.NNREVA) THEN
            IF (KTL.EQ.0) THEN
              WRITE (8,470)
              KTL=1
            ENDIF
            IF (Q(I).GT.0.0) THEN
C             **********************
              WRITE (8,480) NO(I),IABS(JS(I)),IABS(JF(I))
            ENDIF
C             *****
          ENDIF
90      CONTINUE
C
      ELSE IF (IPT.EQ.6) THEN
        WRNSUM=WRNPR+WRNGS+WRNSM+WRNHT
        IF (WRNSUM.LE.0.) THEN
          WRITE (8,410)
        ELSE
          J=0
          WRNP=WRNPR
          IF (PCRIT*0.05.LE.0.5*WRNPR) WRNP=PCRIT*0.05
          DO 75 I=1,NB
            K=0
            L=0
            M=0
            N=0
            IF ((100.*RDCH4(I)).GE.WRNGS) K=K+1
            IF ((100.*RDPROP(I)).GE.WRNSM) L=L+1
            IF (TRD(I).GE.WRNHT) M=M+1
            IF (P(I).LT.WRNP) N=N+1
            IF ((K+L+M+N).GT.0) THEN
              IF (J.EQ.0) THEN
                WRITE (8,420) WRNGS,WRNSM,WRNHT,WRNP
                J=J+1
              ENDIF
              WRITE (8,430) NO(I),IABS(JS(I)),IABS(JF(I)),RDCH4(I),
     .                  RDPROP(I),TRD(I),P(I)
            ENDIF
75        CONTINUE
          IF (J.LE.0) WRITE (8,440)
          J=0
          DO 80 I=1,NJ
            K=0
            L=0
            M=0
            IF ((100.*PRCH4(I)).GE.WRNGS) K=K+1
            IF ((100.*PROP(I)).GE.WRNSM) L=L+1
            IF (T(I).GE.WRNHT) M=M+1
            IF ((K+L+M).GT.0) THEN
              IF (J.EQ.0) THEN
                WRITE (8,450) WRNGS,WRNSM,WRNHT,WRNGS,WRNSM,WRNHT
                J=J+1
```

```
            ENDIF
            IF (J.EQ.1) THEN
              JJ1=JNO(I)
              PR1=PRCH4(I)
              PR2=PROP(I)
              PR3=T(I)
              J=-J
            ELSE
              WRITE (8,460) JJ1,PR1,PR2,PR3,JNO(I),PRCH4(I),
     .                 PROP(I),T(I)
              J=-J
            ENDIF
          ENDIF
80        CONTINUE
        IF (J.LT.0) WRITE (8,460) JJ1,PR1,PR2,PR3
        IF (J.EQ.0) WRITE (8,445)
      ENDIF
    ELSE IF (IPT.EQ.5) THEN
      NRW=0
      DO 95 L=1,NB
        LL=IBTN(L)
        IF (LL.GT.0) THEN
          IF (NRW.EQ.0) THEN
            WRITE (8,570)
            NRW=1
          ENDIF
          LL1=NSAC(L,2)
          IF (LL1.GT.0) THEN
            WRITE (8,580) NO(L),(DPPA(LL,JQ,1),JS(L),DPPA(LL,JQ,2)
     .                ,TAUXA(LL,JQ),(DPPA(LL,JQ,JP),JP=3,5),
     .                JQ=LL1,1,-1)
          ELSE
            WRITE (8,640) NO(L),TJS(L),TRD(L),RDPROP(L)
          ENDIF
        ENDIF
95      CONTINUE
    ELSE
      NRW=0
      DO 100 L=1,NB
        LL=IBTN(L)
        IF (LL.GT.0) THEN
          IF (NRW.EQ.0) THEN
            WRITE (8,590)
            NRW=1
          ENDIF
          JLE=NSP(L)
          MRW=0
          DO 98 J=1,JLE
            DIS=LA(L)*(J-1.0)/JLE
            IF (MRW.EQ.0) THEN
              WRITE (8,600) NO(L),DIS,JS(L),TDA(LL,J),STV(LL,J),
     .                 HABS(LL,J),WMTR(LL,J),GFF(LL,J),
     .                 WTCORR(LL,J)
              MRW=1
            ELSE
              WRITE (8,740) DIS,JS(L),TDA(LL,J),STV(LL,J),
```

```
     .                    HABS(LL,J),WMTR(LL,J),GFF(LL,J),
     .                    WTCORR(LL,J)
              ENDIF
98          CONTINUE
          ENDIF
100     CONTINUE
      ENDIF
C
      RETURN
C
110   FORMAT (///,22X,'NETWORK CALCULATION RESULTS ',/)
120   FORMAT (1X,'AIRWAY',2X,'FROM',4X,'TO',3X,'TYPE',8X,'RESISTANCE',
     .       7X,'AIRFLOW',6X,'HEADLOSS',/)
130   FORMAT (I5,I7,I6,I7,8X,E9.4,6X,F10.0,3X,F10.3)
140   FORMAT (//,' NETWORK CONTAINS NO FANS')
150   FORMAT (///,T20,'FANS'//' AIRWAY  FROM    TO      AIRFLOW    FAN',
     .       ' PRESSURE',/)
160   FORMAT (I5,I7,I7,F13.0,F13.3)
170   FORMAT (///,' THESE CHARACTERISTICS WERE STORED FOR FANS',5I5,/
     .        T45,5I5)
175   FORMAT (//,5('  Q*1000   PF '),/)
180   FORMAT (5(F8.1,F6.2)/,5(F8.1,F6.2),/)
190   FORMAT (///' THE STATED NUMBER OF AIRWAYS WAS',I9,/,' THE ',
     .       'STATED NUMBER OF JUNCTIONS WAS',I7,/)
200   FORMAT (/,1X,'FAN CHARACTERISTICS ARE EXCEEDED FOR FAN NO',I5)
210   FORMAT (///,T7,' INPUT DATA FOR CONCENTRATION AND TEMPERATURE',
     .       ' CALCULATION',//,T2,'AIRWAY  FROM  TO',T20,'ELEV DIFF',
     .       T31,'ROCK T',T40,'WETNESS',T50,'CH4 PROD.',T60,'COND.',T68,
     .       'DIFF.',/)
220   FORMAT (I5,I7,I4,T20,F8.1,T30,F7.1,T39,F7.2,T49,F8.2,T58,
     .       F7.2,T66,F7.3)
230   FORMAT (///,T10,' TIME AFTER BEGINNING OF EVENT',F7.2,'   HOURS')
240   FORMAT (///,' A TEMPERATURE OF',F7.1,'  WAS ASSIGNED TO ',
     .       'STARTING JUNCTION NO',I7)
250   FORMAT (///,T18,'THE FOLLOWING HEAT/CONTAMI. WAS ASSUMED',//,
     .       12X,'FIXED INFLUX',T31,'O2-RICH',T41,'FUEL-RICH',T55,
     .       'LEADING T',T67,'REF. Q',/,' AIRWAY  FLUX  %CONC',
     .       3X,'HEAT',5X,'%O2',4X,'FUMES',2X,'HEAT',/)
260   FORMAT (I5,F8.1,F6.2,1X,E9.3,F7.2,F7.2,F7.1,F10.2,F12.0)
270   FORMAT (///,1X,' NO HEAT/CONTAMINATION WAS SPECIFIED')
280   FORMAT (///,' THE NETWORK CALCULATION WAS NOT COMPLETED IN',
     .       I5,' ITERATIONS.')
285   FORMAT (///,' REF.TEMP.(TR): ',F6.1,' DEG.F',11X,'REF. DENSITY',
     .       ' (DR): ',F6.4,' LBM/FT3',///,' LIST OF UNITS USED IN ',
     .       'THIS OUTPUT FILE:',//,' LENGTH:  FT.;',T25,'AREA:    ',
     .       'FT2;',T42,'VOLUME:   FT3;',T61,'MASS:   LBM;',/,
     .       ' AIRFLOW: ','FT3/MIN;',T25,'TEMP.      F;',
     .       T42,'RESISTANCE:',1X,'1.E-10IN.W.G./CFM2;',/,' FRICTION',
     .       ' FACTOR:',1X,'1.E-10LBM*MIN2/FT4;',T42,'THERMAL DIFFUSI',
     .       'VITY:   FT2/HR;',/,' THERMAL CONDUCTIVITY:   BTU/HR*FT',
     .       '*F;',T42,'CONCENTRATION',7X,'PERCENTAGE;',/,' PRESSURE ',
     .       'DROP:',14X,'IN.W.G.;',T42,'HEAT INPUT:',12X,'BTU/MIN;',/,
     .       ' CH4 PRODUCTION:',13X,'FT3/MIN;',/,1X,
     .       'MARK FOR WALL CONDITIONS (NHM):',I2,' (0 DRY, 1 WET)')
290   FORMAT (///,' THE CALCULATION OF CONCENTRATIONS AND TEMPERA',
     .       'TURES WAS NOT',/,' COMPLETED IN',I5,' ITERATIONS.')
```

```
300  FORMAT (///,T8,'TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN'
   .     ,' AIRWAYS',//,' AIRWAY',T9,'FROM',T15,'TO',T20,'AIRFLOW',
   .     T31,'AVE. T',T40,'T AT END',T50,'FUMES',T57,'METHANE',T66,
   .     'P DROP',/)
310  FORMAT (I5,I6,I5,T18,0PF9.0,T30,0PF7.2,T40,0PF7.2,T49,2PF6.2,
   .     T56,2PF6.2,T66,0PF6.3)
320  FORMAT (///,T24,'AT',F7.0,' SEC. AFTER EVENT')
330  FORMAT (///,T4,'TEMPERATURE AND CONCENTRATION AT AIRWAY ENDS,',
   .     ' PRESSURES IN AIRWAYS',/,T16,'(BY THE END OF THE ',
   .     'CURRENT TIME INTERVAL)',//,2X,' NO',T7,'DELTA Q',T15,
   .     'AIRFLOW',T23,'AVE T',T30,'T AT END',T39,'FUMES',T45,
   .     'CH4',T50,'HUM RATIO',T60,'REL HUM',T70,'P',/)
340  FORMAT (1X,I4,T7,I7,T14,I7,T21,0PF7.1,T30,0PF7.1,T37,
   .     2PF6.2,T43,2PF6.2,T52,0PF7.5,T60,2PF6.2,T67,0PF6.3)
350  FORMAT (///,T21,'DATA FOR THE FUME FRONT IN AIRWAYS',
   .     //,1X,'AIRWAY',T11,'POSITION',T26,'FROM',
   .     T37,'TEMPERATURE',T51,'FUME',T61,'METHANE',/)
360  FORMAT (1X,I5,T11,0PF7.2,T25,I5,T38,0PF8.2,T49,2PF6.2,T61,2PF6.2)
370  FORMAT (///,T3,'PARAMETERS OF AIR IN JUNCTIONS AT ',
   .     'BEGINNING OF CURRENT TIME INTERVAL',//,' JUNCTION',
   .     T15,'T',T25,'FUMES',T38,'CH4',T47,'HUM. RATIO',T65,
   .     'REL. HUM',/)
380  FORMAT (I5,T11,0PF7.2,T22,2PF8.2,T35,2PF6.2,T49,0PF7.5,
   .     T65,0PF7.2)
410  FORMAT (///,1X,'NO THRESHOLD FOR CRITICAL STATES ',
   .     'WERE SPECIFIED.')
420  FORMAT (///,9X,'IN THE FOLLOWING AIRWAYS EXIST CRITICAL ',
   .     'CONDITIONS',//,1X,'AIRWAY',T10,'FROM',T16,'TO',T22,'CH4 %'
   .     ,T33,'FUMES %',T45,'TEMPERATURE',T61,'HEADLOSS',/,
   .     T22,'>',F5.3,T33,'>',F6.3,T46,'>',F5.0,' F',T59,
   .     '<',F6.3,' IN.WG.',/)
430  FORMAT (I5,I7,I5,T21,2PF6.2,T32,2PF8.3,T45,0PF8.1,T62,0PF6.3)
440  FORMAT (///,1X,'NO CRITICAL CONDITIONS AT AIRWAY ENDS ',
   .     'WERE DETECTED')
445  FORMAT (///,1X,'NO CRITICAL CONDITIONS IN JUNCTIONS WERE ',
   .     'DETECTED.')
450  FORMAT (///,10X,'IN THE FOLLOWING JUNCTIONS EXIST CRITICAL ',
   .     'CONDITIONS',//,' JUNCTION',T11,'CH4  %',T19,'FUMES %',T28
   .     ,'TEMP. F',T40,'JUNCTION',T49,'CH4  %',T57,'FUMES %',T66,
   .     'TEMP. F',/,T11,'>',F5.2,T19,'>',F6.3,T29,'>',F5.0,T49,
   .     '>',F5.2,T57,'>',F6.3,T67,'>',F5.0,/)
460  FORMAT (I6,T11,2PF5.2,T17,2PF8.2,T27,0PF7.1,T39,I6,T49,2PF5.2,
   .     T55,2PF8.2,T65,0PF7.1)
470  FORMAT (///,' REVERSAL OF AIRFLOW HAS OCCURRED IN THE ',
   .     'FOLLOWING PLACES',//)
480  FORMAT (' AIRWAY',I6,' IS NOW CARRYING AIR FROM ',I6,' TO ',I6)
490  FORMAT (///,15X,'BASIC DATA FOR AIRWAYS IN THE NETWORK',
   .     //,' AIRWAY FROM',T15,'TO',T22,'AIRFLOW',T30,'TYPE',T37,
   .     'LENGTH',T47,'AREA',T57,'R',T64,'K',T68,'PERI.',/)
500  FORMAT (I5,I6,T13,I4,T19,F10.0,T30,I3,T36,F7.1,T44,F7.1,T53,E8.3,
   .     T62,I4,F7.1)
520  FORMAT (///,T24,'AT',F5.1,' HR. AFTER EVENT',//,T12,'(ASSUMING',
   .     1X,'QUASI-EQUALIBRIUM AND DRY CONDITIONS)')
570  FORMAT (///,23X,'DATA RECORD IN AIRWAYS',//,1X,'AIRWAY',T12,
   .     'DIST.',T19,'FROM',T26,'FRONT T',T37,'REAR T',
   .     T47,'FUME',T57,'CH4',T63,'AIR M (LB)',/)
```

```
580  FORMAT (1X,I5,11(T8,0PF9.2,T17,I5,T24,0PF9.2,T34,0PF9.2,T44,
   .     2PF7.2,T53,2PF7.2,T62,0PF9.2,/))
590  FORMAT (///,18X,'WATER EVAPO./CONDEN. DATA IN AIRWAYS',/,T16,
   .     'EVAPO. (LB/100FT2); CONDEN. (LB/100LB AIR)',//,
   .     ' AIRWAY',T10,'DIST.',T16,'FROM',T22,'AIR T',
   .     T29,'WALL T',T37,'HUM. RATIO',T49,'EVAPO.',T57,
   .     'CONDEN.',T66,'T CORR.',/)
600  FORMAT (/,T2,I5,T8,0PF7.1,T15,I5,T20,0PF7.1,T27,0PF8.1,
   .     T37,0PF8.5,T46,2PF8.4,T55,2PF8.4,T64,0PF8.2)
610  FORMAT (///,28X,'DATA FOR FIRE SOURCES',//,T2,'AIRWAY',T11,
   .     'AIRFLOW',T24,'TEMP.',T34,'FUME',T43,'O2 LEFT',T53,
   .     'H2O VAPOR',T66,'HEAT IN',/,T4,'NO.',T11,'FT3/MIN',
   .     T25,'(F)',T35,'(%)',T45,'(%)',T53,'LB/LB AIR',T66,
   .     'BTU/MIN',/)
620  FORMAT (T2,I5,T10,0PF8.0,T21,0PF8.2,T32,2PF6.3,T44,0PF6.3,
   .     T54,0PF6.4,T64,0PF9.0)
630  FORMAT (///,' ')
640  FORMAT (1X,I5,T9,'NO C.V. T(BGG)=',0PF7.1,', T(END)=',0PF7.1,',
   .     FUME=',2PF5.2,'%.',/)
650  FORMAT (///,16X,'PARAMETERS OF AIR IN JUNCTIONS',//,1X,'JUNCTION'
   .     ,T13,'TEMP.',T20,'FUMES',T27,'METHANE',T39,'JUNCTION',T50,
   .     'TEMP.',T57,'FUMES',T64,'METHANE',/)
660  FORMAT (I6,T9,0PF9.2,T18,2PF7.2,T26,2PF7.2,T38,I6,T47,0PF8.1,T55,
   .     2PF7.2,T63,2PF7.2)
690  FORMAT (///,1X,'THE CALCULATION WAS NOT COMPLETED SINCE',1X,
   .     'THE NUMBER OF ',/,' EXCHANGES BETWEEN NETWORK AND',
   .     ' TEMPERATURE PARTS BECAME EXCESSIVE.',/,
   .     1X,'CRITERIA: NVP CORR. < 0.0002 IN.W.G., CURRENT:',
   .     ' NVP CORR. ',F7.4)
700  FORMAT (//,1X,'THE FOLLOWING UNSTABLE AIRWAYS WERE',1X,
   .     'DETECTED:')
710  FORMAT (/,1X,'AIRWAY',I6,' OF AIRFLOW ',F7.0,' (CFM) & P. ',
   .     'DROP',F6.3,' IN.W.G. HAD ITS',/,' AIRFLOW REVERSED',I3,
   .     ' TIMES IN',I3,' ITERATIONS.')
720  FORMAT (///,' RECIRCULATION WAS DETECTED IN THE SYSTEM. ',
   .     1X,'THE FOLLOWING AIRWAYS',/,' WERE INVOLVED ',
   .     'IN THE RECIRCULATION PATH(ES):',/)
730  FORMAT (1X,10I7)
740  FORMAT (T8,0PF7.1,T15,I5,T20,0PF7.1,T27,0PF8.1,T37,0PF8.5,T46,
   .     2PF8.4,T55,2PF8.4,T64,0PF8.2)
750  FORMAT (///,T3,'TEMPERATURES AND CONCENTRATIONS AT AIRWAY ',
   .     'ENDS, PRESSURES IN AIRWAYS',/,T15,'(BY THE END OF THE ',
   .     'CURRENT TIME INTERVAL)',//,' AIRWAY',T9,'FROM',T15,
   .     'TO',T20,'DELTA Q',T29,'AIRFLOW',T38,'AVE. T',T46,
   .     'T AT END',T56,'FUMES',T64,'CH4',T70,'P',/)
760  FORMAT (1X,I5,I6,I5,T19,0PF8.0,T27,0PF8.0,T37,0PF7.2,T46,
   .     0PF7.2,T55,2PF6.2,T61,2PF6.2,T67,0PF6.3)
770  FORMAT (///,' THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS ',
   .     'PER MESH < 2.E-4 IN.W.G.)',/,' SATISFIED. CURRENT ',
   .     'SUMFNV PER MESH',F9.6,' IN.W.G.,    ITERATIONS ',I5)
780  FORMAT (///,1X,'THRESHOLD IN ACCURACY (SUM OF NVP',1X,
   .     'CORRECTIONS PER MESH<2.E-4 IN. W.G.)',/,' NOT ',
   .     'SATISFIED. CURRENT SUMFNV:',F9.6,' IN.W.G., ITER.',I5)
800  FORMAT (///,T2,' INPUT DATA FOR CONCENTRATION AND TEMPERATURE',
   .     ' CALCULATION',//,T2,'AIRWAY  FROM  TO',T20,'ELEV DIFF',
   .     T31,'ROCK T',T41,'CH4 PROD.',T51,'COND.',T61,'DIFF.',/)
```

```
810  FORMAT (I5,I7,I4,T20,F8.1,T30,F7.1,T41,F7.2,T49,F7.2,T59,F7.3)
820  FORMAT (///,28X,'DATA FOR FIRE SOURCES',//,T2,'AIRWAY',T11,
   .      'AIRFLOW',T24,'TEMP.',T34,'FUME',T48,'O2 LEFT',
   .                T66,'HEAT IN',/,T4,'NO.',T11,'FT3/MIN',
   .      T25,'(F)',T35,'(%)',T50,'(%)',T66,'BTU/MIN',/)
830  FORMAT (T2,I5,T10,0PF8.0,T21,0PF8.2,T32,2PF6.3,T49,0PF6.3,
   .      T64,0PF9.0)
840  FORMAT (/,1X,'* WRN * AIRFLOW REVERSED FOR FAN NO.',I5)
850  FORMAT (//,1X,'HEAT SOURCE IN AIRWAY',I6,' HAD REDUCED HEAT INPUT'
   .      ,' AS',F9.0,' BTU/MIN',/,1X,'FOR THE HIGH TEMPERATURE IN',
   .      ' THE CURRENT TIME INTERVAL.')
860  FORMAT (///,1X,'THE FOLLOWING DATA RECORDS WERE OMITTED DUE TO ',
   .      'ARRAY CAPACITIES.',//,T2,'AIRWAY',T20,'PARAMETERS AT ',
   .      'STARTING JUNCTIONS IN THE CURRENT TIME INTERVAL',/,T27,
   .      'JS',T36,'TEMPERATURE',T51,'FUME %',T66,'CH4 %',/)
870  FORMAT (T2,I5,T24,I5,T36,F8.1,T49,F7.3,T64,F7.3)
890  FORMAT (///,1X,'THE FAILURE IN CONVERGING WAS NOT CAUSED BY A ',
   .      ' CYCLIC PATTERN.',/,' RELEVANT DATA ARE SHOWN BELOW:',
   .      ' ("ITR" STANDS FOR ITERATION)',//,' UNITS: Q: AIRFLOW ',
   .      'RATE IN 1000FT3/MIN;',17X,'T: TEMP. IN F;',/,8X,'H: NVP ',
   .      'CORRECTION PER MESH IN IN.W.G.',//,' ITR',T5,I5,9I7,/)
900  FORMAT (///,1X,'THE FAILURE IN CONVERGING WAS CAUSED BY A CYCLIC',
   .      ' PATTERN IN EVERY',I3,/,' ITERATIONS AS SHOWN BELOW:',
   .      ' ("ITR" STANDS FOR ITERATION)',//,' UNITS: Q: AIRFLOW ',
   .      'RATE IN 1000FT3/MIN;',17X,'T: TEMP. IN F;',/,8X,'H: NVP ',
   .      'CORRECTION PER MESH IN IN.W.G.',//,' ITR',T5,I5,9I7,/)
905  FORMAT (1X,'H',T3,10(1X,F6.4),/)
910  FORMAT (1X,'AIRWAY: ',I4,10X,'JS:',I4,6X,'JF:',I4,/,1X,
   .      'T',T3,10F7.1)
920  FORMAT (1X,'Q',T3,10F7.1,/)
980  FORMAT (///,T2,'JUNCTION',T12,'TEMP.',T20,'ELEVATION',T32,'CH4',
   .      T40,'JUNCTION',T50,'TEMP.',T58,'ELEVATION',T70,'CH4',/)
990  FORMAT (T2,I5,T10,F7.1,T19,F8.1,T30,F5.2,T40,I5,T48,F7.1,T57,F8.1
   .      ,T68,F5.2)
1040 FORMAT (///,15X,'COEFF. OF POLYNOMIAL OF FAN CURVES',/,20X,
   .      'PF=A0+A1*Q+A2*Q**2+A3*Q**3')
1044 FORMAT (/,T5,'FAN',I5,//,T7,'A0',T19,'A1',T31,'A2',T43,'A3',/)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
1045 FORMAT (/,T5,'FAN',I5,//,T7,'A0',T19,'A1',T31,'A2',T43,'A3',T52,
   .      'FROM QF',T65,'TO QF',/)
1046 FORMAT (T3,E9.3,T15,E9.3,T27,E9.3,T39,E9.3)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
1050 FORMAT (T3,E9.3,T15,E9.3,T27,E9.3,T39,E9.3,T51,E9.3,T63,E9.3)
1085 FORMAT (/,2X,'INTERPOLATION:',/,5('  Q*1000    PF '),/)
1090 FORMAT (5(F8.1,F6.2))
1210 FORMAT (//,' THE FOLLOWING SURFACE JUNCTIONS ARE DECLARED:',/)
1220 FORMAT (' JUNCTION ',I5,' IN THE ATMOSPHERE WITH TEMPERATURE ',
   .      F8.1,' DEGREES F.')
1230 FORMAT (/,' * ATTN * A NEW BRANCH HAS BEEN SET UP FOR EACH HEAT ',
   .      'SOURCE.',//)
C
     END


     SUBROUTINE SPLINE (J,QZZ1,RKI,TABF,NTW)
C
```

```
C     ----------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       1) FAN CURVE FITTING.
C       2) FAN PRESSURE INTERPOLATION.
C
C     ----------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
C     FKQ(J,I,1-4): COEFFS. OF 3RD ORDER POLYNOMIAL FOR FAN NOF(J),
C                   SEGMENT I.
C     NSKP(J)=1: MARKER INDICATING THAT FAN CURVE FITTING FOR FAN
C                NOF(J) HAS BEEN PROCESSED.
C
      MF = MPTS(J)
      IF (NTW.EQ.0) THEN
C
        DO 5 I=1,MF-1
          DK(I)=QF(J,I+1)-QF(J,I)
5       CONTINUE
C
        DO 10 I=2,MF-1
          FC3(I)=3.0*(PF(J,I+1)*DK(I-1)-PF(J,I)*(QF(J,I+1)-QF(J,I-1))
     .           +PF(J,I-1)*DK(I))/(DK(I-1)*DK(I))
10      CONTINUE
C
        FK(1)=1.0
        FC1(1)=0.0
        FC2(1)=0.0
C
        DO 15 I=2,MF-1
          FK(I)=2.0*(QF(J,I+1)-QF(J,I-1))-DK(I-1)*FC1(I-1)
          FC1(I)=DK(I)/FK(I)
          FC2(I)=(FC3(I)-DK(I-1)*FC2(I-1))/FK(I)
15      CONTINUE
C
        FK(MF-1)=1.0
        FC3(MF-1)=0.0
        FKQ(J,MF-1,3)=0.0
C
        DO 20 I=MF-1,1,-1
          FKQ(J,I,3)=FC2(I)-FC1(I)*FKQ(J,I+1,3)
          FKQ(J,I,2)=(PF(J,I+1)-PF(J,I))/DK(I)-
     .               DK(I)*(FKQ(J,I+1,3)+2.0*FKQ(J,I,3))/3.0
          FKQ(J,I,4)=(FKQ(J,I+1,3)-FKQ(J,I,3))/(3.0*DK(I))
20      CONTINUE
C
        DO 25 I=1,MF-1
          FKQ(J,I,1)=PF(J,I)
25      CONTINUE
        NSKP(J)=1
      ENDIF
C
C
      QKI=ABS(QZZ1)
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF (NPLOT.EQ.1) THEN
        IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
          DQKI=0.0
          DQKI1=QKI-QF(J,1)
          M=1
          RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .               3*FKQ(J,M,4)*DQKI**2
          RKI=PF (J,1)+DQKI1*RGRAD(J)
          RKI=RKI*(TR+460.)/(TABF+460.)
          NFCW(J) = 1
          RETURN
        ELSE IF (QKI.GE.QF(J,MF)) THEN
          DQKI=QF(J,MF)-QF(J,MF-1)
          DQKI1=QKI-QF(J,MF)
          M=MF-1
          RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .               3*FKQ(J,M,4)*DQKI**2
          RKI=PF (J,MF)+DQKI1*RGRAD(J)
          RKI=RKI*(TR+460.)/(TABF+460.)
          NFCW(J) = 1
          RETURN
        ENDIF
      ELSEIF (NPLOT.EQ.2) THEN
        IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
          DQKI=0.0
          DQKI1=QKI-QF(J,1)
          M=1
          RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .               3*FKQ(J,M,4)*DQKI**2
          RKI=PF (J,1)+DQKI1*RGRAD(J)
          RKI=RKI*(TR+460.)/(TABF+460.)
          NFCW(J) = 1
          RETURN
        ELSE IF (QKI.GE.QF(J,MF)) THEN
          RKI = PF(J,MF)*(TR+460.)/(TABF+460.)
          RGRAD(J) = 0.0
          NFCW(J) = 1
          RETURN
        ENDIF
      ELSEIF (NPLOT.EQ.3) THEN
        IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
          RKI = PF(J,1)*(TR+460.)/(TABF+460.)
          RGRAD(J) = 0.0
          NFCW(J) = 1
          RETURN
        ELSE IF (QKI.GE.QF(J,MF)) THEN
          RKI = PF(J,MF)*(TR+460.)/(TABF+460.)
          RGRAD(J) = 0.0
          NFCW(J) = 1
          RETURN
        ENDIF
      ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF (QF(J,1).GE.QKI) THEN
        M = 1
```

```
      ELSE
        DO 40 K = 2,MF
          IF (QF(J,K).GE.QKI) THEN
            M = K-1
            GO TO 45
          ENDIF
40      CONTINUE
      ENDIF
      M = MFA
45    DQKI=QKI-QF(J,M)
      RKI = FKQ(J,M,1)+FKQ(J,M,2)*DQKI+FKQ(J,M,3)*DQKI**2
     .      +FKQ(J,M,4)*DQKI**3
      RKI=RKI*(TR+460.)/(TABF+460.)
      RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .           3*FKQ(J,M,4)*DQKI**2
C
      RETURN
      END

C
C
      SUBROUTINE TCORR (KQ,ID,I)
C
C     ---------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       EVALUATION OF WALL TEMPEARTURE CORRECTION IN WET CASES.
C
C     ---------------------------------------------------------------
C
      INCLUDE 'CMMNW.DAT'
C
      PARAMETER (NT1=45,NT2=55)
      PARAMETER (KMX=NT1*NT2*2, KNX=NMZ*10)
C
      DOUBLE PRECISION BJ00,BJ01,BJ10,BY00,BY01,BY10,RO,R1,H,SKAPPA,
     .                 SKAPA1,STEP
C
      REAL M,LW,BETA
C
      COMMON
     . /BESSEL/ BJ00,BJ01,BJ10,BY00,BY01,BY10
     . /PARAMT/ RO,R1,H,STEP
C
C       NT1 DETERMINES HOW MANY DATA POINTS CAN BE HANDLED FOR TEMP.
C     CORRECTION CALCULATION DUE TO WATER EVAPORATION. NT2 DETERMINES
C     HOW MANY TERMS IN THE INFINITE SERIES OF THE SOLUTION ARE
C     ACTUALLY SUMMED UP. THE CONVERGENCE IS DOMINANTLY CONTROLLED
C     BY THE MAGNITUDE OF R1. ENLARGED R1 GENERALLY REQUIRES LARGER
C     NT2 WHICH IN TURN INCREASES COMPUTING TIME SIGNIFICANTLY. THE
C     CURRENTLY ASSUMED R1 IS GOOD ENOUGH FOR DYNAMIC SIMULATION
C     COVERING A FEW HOURS.
C
      DIMENSION  SKAPPA(NT1,2,NT2),YKAPPA(NT1,2,NT2),H1(NMZ,10),
     .           B1(NMZ,10),NLS(NMZ,10)
      DATA SKAPPA,YKAPPA,H1,B1,NLS
```

```
     .  / KMX*0.0,KMX*0.0,KNX*0.0,KNX*0.0,KNX*0 /
C
C     SKAPPA(NT1,2,NT2): EIGENVALUES.
C     YKAPPA(NT1,2,NT2): INTERMEDIATE STORAGE ARRAY.
C     H1(NMZ,10): INTERMEDIATE STORAGE ARRAY.
C     B1(NMZ,10): INTERMEDIATE STORAGE ARRAY.
C     NLS(NMZ,10): INTERVAL RECORD ARRAY.
C
      PI=3.1415926D0
      LL=0
C
C  ** DATA BASE SETUP.
C     DELTAT: TIME INCREMENT, SEC.   TRV: ROCK TEMP. DEGREE F.
C     RO: HYDROLIC RADIUS, FT.     HA: THERMAL DIFFUSIVITY, FT/HR.
C     HC: HEAT TRANSFER COEFF. BTU/(HR*FT**2*F).
C     HK: THERMAL CONDUCTIVEITY, BTU/(HR*FT*F).
C     R1: RADIUS OF INFLUENTIAL ZONE FOR TEMP. CORR. IN WET CASE.
C     NLS: NUMBER OF INTERVALS IN WHICH NO DETAILED CALCULATIONS
C          IN TEMP. CORR. WERE GIVEN. IT IS USED TO DETERMINE WHICH
C          FORMULA TO USE IN THE LATER INTERVALS.
C
      KN=NSEQE(KQ)+ID
      IF (KN.GT.NT1) THEN
        WRITE (8,100) NO(I)
        WTCORR(KQ,ID)=0.0
        WMTR(KQ,ID)=0.0
        BETA=0.0
        RETURN
      ENDIF
      TMSTT=DELTAT/3600.0
C
C     TROCK: ROCK TEMP.   TDA: AIR T. AT STATIONS.  STV: WALL TEMP.
C     HABS: HUMIDITY RATIO OF AIR AT STATIONS.
C
      TRV=TROCK(I)
      TA=TDA(KQ,ID)
      TO=STV(KQ,ID)
      AH=HABS(KQ,ID)
      LW=1050.0
      RO=2.0*A(I)/O(I)
      R1=RO+1.0
      STEP=1.0D0
C
C     STEP: LENGTH OF STEPS IN SEARCHING FOR EIGENVALUES.
C     GENERALLY, WHEN (R1-RO)<2, STEP<1.5, WHEN 2<(R1-RO)<10,
C     STEP<0.2, WHEN 10<(R1-RO)<30, STEP<0.05. FOR SAFE SEEK, TAKE
C     SMALLER VALUE OF "STEP".
C
      ALOGR=DLOG(R1/RO)
      HC=BI(I)*HK(I)*O(I)/(2.0*A(I))
      IF (TO.GT.210.0) TO=210.0
C
C     SUB. MTCOEF EVALUATES MASS TRANSFER COEFF. BETA.
C
      CALL MTCOEF (TO,TA,I,BETA)
      TV=TO-10.0
```

```
      TU=TO+10.0
      IF (TO.LE.122.0) THEN
        PVM=-1.0682+0.07946*TO-0.00129*TO*TO+1.26E-5*TO**3
        PVB=-1.0682+0.07946*TV-0.00129*TV*TV+1.26E-5*TV**3
        PVC=-1.0682+0.07946*TU-0.00129*TU*TU+1.26E-5*TU**3
      ELSE
        PVM=-33.786+0.97*TO-9.305E-3*TO*TO+3.65E-5*TO**3
        PVB=-33.786+0.97*TV-9.305E-3*TV*TV+3.65E-5*TV**3
        PVC=-33.786+0.97*TU-9.305E-3*TU*TU+3.65E-5*TU**3
      ENDIF
      GAM=0.622*PVM/(101.35-PVM)
      GAV=0.622*PVB/(101.35-PVB)
      GAU=0.622*PVC/(101.35-PVC)
      DRR=(DR+DR*AH*18/28.98)*(TR+460)/((TO+460)*(1+AH))
      SLOPE1=(GAM-GAV)*DRR/10.0
      SLOPE2=(GAU-GAM)*DRR/10.0
      GAA=AH*DRR
      GAW=GAM*DRR
C
C    KEQUIL: 1: THE WALL FILM AND BULK FLOW IS IN A TEMPORARY EQUIL.
C            CONDITION IN MASS TRANSFER. NO WATER EVAPORATION OR
C            CONDENSATION.
C
      KEQUIL=0
      IF (GAA.LT.GAW*PSI(I).AND.TO.LE.CRITW) THEN
        PSII=PSI(I)
        SLOPE=SLOPE1
        NLS(KQ,ID)=NLS(KQ,ID)+1
      ELSE IF (GAA.GT.GAW) THEN
        PSII=1.0
        SLOPE=SLOPE2
        NLS(KQ,ID)=NLS(KQ,ID)+1
      ELSE
        PSII=GAA/GAW
        SLOPE=SLOPE1
        NLS(KQ,ID)=NLS(KQ,ID)+1
        KEQUIL=1
      ENDIF
C
C    NLS: FLAG NUMBER CONDUCTING THE USE OF SOLUTION FORMULAS
C         FOR DIFFERENT TIME INTERVAL.
C
      ITNN=NLS(KQ,ID)
      XI=HC+LW*BETA*PSII*SLOPE
      ETA=LW*BETA*(PSII*GAW-GAA)
      M = ETA/XI
      B = M/TRV
      H = XI/HK(I)
      WN = 1.0+RO*H*ALOGR
      WN1=1.0+RO*H1(KQ,ID)*ALOGR
C
C ** SOLUTION OF TEMP. CORR.
C
      TERM1 = B/(WN*(1.0+B))
      TERM2 = 0.0
      TERMM=0.0
```

```
      DO 40 L=1,NT2
C
C     SKAPA1: EIGENVALUE OF CHARACTERISTIC EQN.  SKAPPA: STORAGE
C     ARRAY OF EIGENVALUES.
C
          CALL  EIGEN (LL,SKAPA1)
          LL=1
          SKAPPA(KN,1,L) = SKAPA1
          Y1 = (SKAPPA(KN,1,L)*BJ10+H*BJ00)**2-
     .        (SKAPPA(KN,1,L)*SKAPPA(KN,1,L)+H*H)*BJ01*BJ01
          YR1 = Y1/(BJ01*BJ01)
          IF (ITNN.EQ.1) THEN
            YR = 1.0
          ELSE
            Y2 = B1(KQ,ID)*(H1(KQ,ID)-H)/(WN1*(1.0+B)*
     .          SKAPPA(KN,1,L)*SKAPPA(KN,1,L))
            Y2 = Y2+(B1(KQ,ID)-B)*H/((1.0+B)
     .          *SKAPPA(KN,1,L)*SKAPPA(KN,1,L))
            SUMYK = 0.0
            DO 30 J = 1,NT2
              SUMYK = SUMYK+YKAPPA(KN,2,J)*(H1(KQ,ID)-H)/
     .              (SKAPPA(KN,2,J)**2-SKAPPA(KN,1,L)**2)
30          CONTINUE
            IF (ITNN.EQ.2) THEN
              Y2 = Y2-SUMYK*H1(KQ,ID)*B1(KQ,ID)*2.0/
     .            (RO*(1.0+B))
            ELSE
              Y2 = Y2+SUMYK*(1.0+B1(KQ,ID))*2.0
     .            /(RO*(1.0+B))
            ENDIF
            YR = Y2*SKAPPA(KN,1,L)*SKAPPA(KN,1,L)
          ENDIF
          EXP3 = -HA(I)*TMSTT*SKAPPA(KN,1,L)*SKAPPA(KN,1,L)
          IF (ABS(YR).LT.1E-15.AND.YR.GT.0) YR=1E-15
          IF (ABS(YR).LT.1E-15.AND.YR.LT.0) YR=-1E-15
          IF (EXP3.LT.-30.) EXP3=-30.
C         ****************************
          YKAPPA(KN,1,L) = YR*EXP(EXP3)/YR1
          YX=YR/YR1
          VBJ0R = -2.0/(PI*RO)
        TMN = YKAPPA(KN,1,L)*VBJ0R
        IF (ABS(TMN).LT.1E-30.AND.TMN.GT.0) TMN=1E-30
        IF (ABS(TMN).LT.1E-30.AND.TMN.LT.0) TMN=-1E-30
C       **********************************************
        TMN1=YX*VBJ0R
        TERM2 = TERM2+TMN
        TERMM=TERMM-(TMN-TMN1)/(HA(I)*SKAPPA(KN,1,L)*SKAPPA(KN,1,L))
40    CONTINUE
      IF (ITNN.EQ.1) THEN
        VP = TERM1-PI*B*H*(1.0+RO*H*ALOGR)*TERM2
     .      /(WN*(1.0+B))
        WM=TERM1*TMSTT-PI*B*H*(1.0+RO*H*ALOGR)*TERMM/
     .    (WN*(1.0+B))
      ELSE
        VP = TERM1+PI*TERM2
        WM=TERM1*TMSTT+PI*TERMM
```

```
      ENDIF
C
C      WTCORR: TEMP. CORR. OF WALL AT STATIONS.
C      WMTR: MASS TRANSFER BETWEEN WALL FILM AND BULK FLOW AT
C           STATIONS IN THE PRESENT INTERVAL, LBM/FT2.
C
      WTCORR(KQ,ID) = VP*(TRV+M)-M
       IF (WTCORR(KQ,ID).LT.0) WTCORR(KQ,ID)=0.
C      ****************************************
      WMTR1=(WM*(TRV+M)-M*TMSTT)*PSII*SLOPE
      WMTR2=(PSII*GAW-GAA)*TMSTT
      WMTR(KQ,ID)=(WMTR1+WMTR2)*BETA*DRR
      IF (KEQUIL.EQ.1) WMTR(KQ,ID)=0.0
C
C  ** SAVE HISTORICAL DATA FOR THE NEXT INTERVAL.
C
      B1(KQ,ID) = B
      H1(KQ,ID) = H
      DO 70 J = 1,NT2
        SKAPPA(KN,2,J) = SKAPPA(KN,1,J)
        YKAPPA(KN,2,J) = YKAPPA(KN,1,J)
70    CONTINUE
C
100   FORMAT (/,' * ATTN * TEMP. CORR. DATA FOR AIRWAY ',I5,' EXCEEDED',
     .       ' THE ARRAY',/,' CAPACITY. IT IS ASSUMED DRY NOW.')
C
      RETURN
      END


      PARAMETER (NMX=500,NMY=350,NMZ=500,IMX=15,IMY=15,IMZ=10,LMX=15000,
     .           NXX=NMX*3)
C
      REAL LA
C
      COMMON
     . /CONTRL/ NB,NJ,NFNUM,NVPN,NETW,NTEMP,MADJ,ITN,DR,TR,TINC,
     .  SPAN,NHM,CRITW,IOUT
     . /CTRL/   NAV,MAXJ,INFLOW,CRITSM,CRITGS,CRITHT
     . /NTWK/   NO(NMX),JS(NMX),JF(NMX),NWTYP(NMX),R(NMX),Q(NMX),P(NMX),
     .  KF(NMX),LA(NMX),A(NMX),O(NMX),RSTD(NMX),DZRD(NMX),FRNVP(NMX),
     .  NREV(NMX),RDCH4(NMX),RDPROP(NMX),TRD(NMX),TJS(NMX),RDOP(NMX),
     .  RCH4(NMX),FFRNVP(NMX),RA(NMX),NNREV(NMX),RDH2O(NMX),RH2O(NMX)
      COMMON /FC/    FC1(IMY),FC2(IMY),FC3(IMY)
     . /FAN/    NOF(IMX),NFREG(IMX),RGRAD(IMX),NFCW(IMX),MPTS(IMX),
     .  QF(IMX,IMY),PF(IMX,IMY),NSKP(IMX),NEGQ(IMX),JSB(NMX)
     . /MESH/   MNO,MEND(NMY),MSL(LMX),FNVP(NMY),RQ(NMX),INU(NMX),
     .  KNO(NMX),KJS(NMX),KJF(NMX)
     . /JUNCT/  JNO(NMY),T(NMY),Z(NMY),CH4C(NMY),JNOL(NMY),PROP(NMY),
     .  PRCH4(NMY),JLR(NMY),HJN(NMY),PRH2O(NMY)
     . /TEMP/   TROCK(NMX),TMRD(NMX),HA(NMX),HK(NMX)
     . /FUME/   NCENT(NMX),CH4V(NMX),CH4PA(NMX),CONT(IMX),CONC(IMX),
     .  HEAT(IMX),O2MIN(IMX),SMPO2(IMX),HTPO2(IMX),TFSI(IMX),CN(IMX),
     .  HN(IMX),O2BEH(IMX),TPR(IMX),HTAD(IMX),QCENT(IMX)
      COMMON
     . /SCHEME/ NGIN(NMX),MIN(NMX),NGOUT(NMX),LOUT(NMX)
```

```
. /EST/    MEMREC(NMX),NOREC(NMX),ESTPR(NMX),ESTCH4(NMX),ESTTR(NMX)
. /FACTOR/ XNEW(NMX),DCOAGE(NMX),WTCORR(NMZ,10),NSEQS(NMZ)
. /TTJ/    TIME,TSTART,JSTART,TLEFT(NMX)
. /SUM/    SUMQ(NMY),SUMC(NMY),SUMM(NMY),SMHEAT(NMY),SUMW(NMY)
. /AUX/    PROPS(NMX),PRCH4S(NMX),QQ(NMX),QTP(NMX),TTJS(NMX),
.  TTRD(NMX),TAS(NMX),TMM(NMX),BI(NMX),CNT(NMX),EXPN(NMX),FAW(NMX)
. /HUMID/  PRH2OS(NMY),HSTART,PSI(NMX),NSP(NMX),STV(NMZ,10),
.  HABS(NMZ,10),NSEQE(NMZ),TDA(NMZ,10),WMTR(NMZ,10),GFF(NMZ,10)
. /TRANS/  DELTAT,SUMFNV
. /RCD/    IBTN(NMX),DPPA(NMZ,10,5),NSAC(NMX,2),TMRDA(NMX),
.  DELTAQ(NMX),DPPB(NMZ,10,5),NSACB(NMX,2),GBTN(10,5),JCH(NMY),
.  TAUXA(NMZ,10),TAUXB(NMZ,10),TAUXC(NMZ),TAUXD(NMZ),FMASS(NMZ)
. /CDEV/   GFS(IMX),TACC,NPP(NMZ),HABS1(NMZ,10)
. /CCJ/    JDP,JDPP(IMX)
. /WRN/    WRNPR,WRNGS,WRNSM,WRNHT,IOMIT(IMX),ROMIT(IMX,3),IAC
. /TREND/  QRCD(NMX,10),TMRCD(NMX,10)
. /CURVE/  FC(20,20),DK(20),FK(20),FKQ(IMX,20,4)
 COMMON /FAN1/COF(IMX,IMX),ND(IMX),N1A,NSWT(IMX),NCOF(IMX),
.      INIFAN,NPLOT
 COMMON /PLOT/QQ1(NMX),H1(NMX),Q2(150),H2(NMX),Q3(NMX),H3(NMX),
.      QQ2(NMX)
```