NISTIR 89-4072

AIRNET - A Computer Program for Building Airflow Network Modeling

George N. Walton

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
(Formerly National Bureau of Standards)
National Engineering Laboratory
Center for Building Research
Building Environment Division
Gaithersburg, MD 20899

April 1989

NISTIR 89-4072

AIRNET - A Computer Program for Building Airflow Network Modeling

George N. Walton

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
(Formerly National Bureau of Standards)
National Engineering Laboratory
Center for Building Research
Building Environment Division
Gaithersburg, MD 20899

April 1989

National Bureau of Standards became the
National Institute of Standards and Technology
on August 23, 1988, when the Omnibus Trade and
Competitiveness Act was signed. NIST retains
all NBS functions. Its new programs will encourage
improved use of technology by U.S. industry.

U.S. DEPARTMENT OF COMMERCE
Robert Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

ABSTRACT

In spite of its importance, the analysis of airflows has significantly lagged
the modeling of other building features because of limited data, computational
difficulties, and incompatible methods for analyzing different flows. Methods
have been developed to analyze airflows in HVAC ducts and to estimate infil-
tration but the interaction between building HVAC systems and infiltration
airflows has seldom been studied. This report describes a computer program
for modeling networks of airflow elements, such as openings, ducts, and fans.
It emphasizes the numerical aspects of an airflow network method which would
provide a unified approach to building airflow calculations. It also
discusses the limitations of the method and poorly understood factors that
could profit from further research.

ACKNOWLEDGEMENTS

DISCLAIMER

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Air movement models have been developed for estimating airflows in buildings. These airflows include infiltration, natural ventilation, interroom airflows through various openings including doorways, and flows through the HVAC system. The numerical estimation of average characteristics of such airflows is useful for moisture and contaminant dispersal analysis, including the design of smoke control systems, and heat transfer analysis including load and energy calculations. In spite of its importance, the analysis of airflows has significantly lagged the modeling of other building features because of limited data, computational difficulties, and incompatible methods for analyzing different flows. This is particularly true of the combined building and HVAC system simulation. Methods have been developed to analyze airflows in ducts (ASHRAE, 1985, ch 33) and to estimate infiltration (Liddament & Thompson, 1982) and ventilation (ASHRAE, 1985, ch 22), but the intimate relationship between these processes has seldom been studied. When it has, the results have sometimes been surprising (Persily, 1985).

Relatively few methods that could be applied to both processes have been developed within the building simulation community and described in detail. Several computer models developed for smoke control analysis are reviewed by Said (1988). Models for building energy analysis have been developed by Clarke (1985) and Walton (1984). All of these methods are based upon the idea that there is a simple nonlinear relationship between the flow through an opening and the relative air pressure difference across it, and that a building can be considered to be composed of a large number of rooms which are connected by openings to each other and to the outside. This is a network of rooms (nodes) and openings (connections) which is conceptually similar to the air handling system network where the connections are the ductwork and the nodes are the ductwork junctions. Conservation of mass for the flows into and out of each node leads to a set of simultaneous nonlinear equations which are solved iteratively for the airflows. This can be called an "airflow network" method. Its relationship to pipe network methods will be discussed. Such an analysis is also sometimes referred to as a multi-chamber or multi-cell method (ASHRAE, 1985, p 22.13). This report draws extensively on Axley's airflow element (1987) and contaminant element (1988) methods which are, in turn, based on numerical methods associated with finite element modeling techniques.

Modeling of airflows requires: (1) determination of the location and mathematical characterization of the airflow paths, (2) determination of the boundary conditions (primarily wind pressure), (3) calculation of the resulting airflows, and (4) a user-friendly framework in which to do the analysis. Progress has been made in such vital areas as wind pressure estimates (Swami & Chandra, 1988) and interroom airflows (Barakat, 1987). Unfortunately, it is often thought that a network model is so complex that it requires a mainframe computer for its solution (ASHRAE, 1985, p 22.13) and is, therefore, impractical. This apparent impracticality discourages gathering the data which is necessary for the use of network models.

This report will shown that a network model is practical. It will emphasize the numerical aspects of the airflow network method which allow it to provide a practical, unified approach to building airflow calculations. Details of the program AIRNET, a microcomputer implementation of this airflow network method, will be discussed. It will also discuss the limitations of the method

1

and poorly understood factors that could profit from further research.

2. AN AIRFLOW NETWORK METHOD

An airflow network consists basically of a set of nodes connected by airflow elements. The nodes may represent rooms, connection points in ductwork, or the ambient environment. The airflow elements correspond to discrete airflow passages such as doorways, construction cracks, ducts, and fans. Figure 1 is a sketch of a portion of a building consisting of two rooms, a hallway, and air distribution equipment representing a VAV system. Figure 2 shows an airflow network superimposed on the physical structure of figure 1. The large dots are nodes and the connecting lines are the various airflow elements.

2.1 Modular Approach

The network approach makes the development of element models, excitation models, and solution method somewhat independent. The computer program modules will obviously mirror the theoretical modules with input and output modules added to create a useful simulation tool. The various modules provide a toolkit for the analyst to consider a practically infinite variety of system models.

For this study an airflow network simulation computer program, AIRNET, was developed from an earlier airflow analysis program (Walton, 1984). The new program consists of:
(1) a process for establishing an initial set of values to start the iterative solution process,
(2) a solution method for nonlinear equations consisting of a traditional Newton's method combined with Steffensen iteration to accelerate convergence,
(3) airflow element subroutines which compute the flow rate and derivative of the flow with respect to pressure difference needed to form the Jacobian matrix used in Newton's method,
(4) a separate process for transferring the above data into the Jacobian matrix (called the element assembly process), and
(5) solution of the simultaneous linear equations involving the Jacobian matrix.

This discussion will begin with the solution method.

2.2 Newton's Method

Each airflow element, i, relates the mass flow rate, w_i, through the element due to the pressure drop, ΔP_i, across it. Conservation of mass at each node is equivalent to the mathematical statement that the sum of the mass flows equal zero (or the mass generated, as in the case of a fire) at each node. The flows are related nonlinearly to the pressures at the nodes thus requiring the iterative solution of a set of nonlinear equations. In Newton's method (Conte & de Boor, 1972, p 86), a new estimate of the vector of all node pressures, $\{P\}^*$, is computed from the current estimate of pressures, $\{P\}$, by

$$\{P\}^* = \{P\} - \{C\} \tag{1}$$

2

where the correction vector, {C}, is computed by the matrix relationship

$$[J] \; \{C\} = \{B\} \tag{2}$$

{B} is a column vector with each element given by

$$B_n = \sum_i w_i \tag{3}$$

where n is the node number and i indicates all flow paths connecting node n to other nodes, and [J] is the square (i.e. N by N for a network of N nodes) Jacobian matrix whose elements are given by

$$J_{n,m} = \sum_i \frac{\partial w_i}{\partial P_m} \tag{4}$$

In equations (3) and (4) w_i and $\partial w_i / \partial P_m$ are evaluated using the current estimate of pressure {P}. The AIRNET program contains subroutines for each airflow element which return the mass flow rates and the partial derivative values for a given pressure difference input.

2.3 Solution of the Equations

Equation (2) represents a set of linear equations which must be set up and solved for each iteration until a convergent solution of the set of nonlinear equations is achieved. In its full form [J] requires computer memory for N^2 values, and a standard Gauss elimination solution has execution time proportional to N^3. Sparse matrix methods can be used to reduce both the storage and execution time requirements.

A skyline solution process following the method of Dhatt (1984, pp. 282-192) was chosen. This method can be used to solve equations with symmetric or nonsymmetric matrices. It stores no zero values above the highest nonzero element in the columns above the diagonal and no zero values to the left of the first nonzero value in each row below the diagonal. Analysis of the element models will show that

$$\left| J_{n,n} \right| = \sum_{m \neq n} \left| J_{n,m} \right| \tag{5}$$

This condition allows a solution without pivoting, although scaling may be useful. Modularizing the equation solution process and the matrix assembly process will make it easy to substitute other solution processes.

Note that the degree of sparsity of Jacobian matrix is dependent on the ordering of the nodes. Ordering can be improved by various algorithms or rules-of-thumb. Also note that it is easy to define an airflow network which has no unique solution. One requirement for solution is that at least one of the node pressures be known. This is usually the ambient node. All nodes must be linked, through some path, to a known pressure. There may be several known pressure nodes. The airflow network method allows two types of nodes: those with known or unknown pressures. In AIRNET the constant pressure nodes are included in the system of equations and equation (2) processed so as to not change those node pressures. This gives an added flexibility in defining the airflow network with special processing maintaining the symmetric set of

3

equations. The form of the equations for known pressure nodes combined with the condition in equation (5) for unknown pressures nodes is a sufficient condition for the Jacobian to be nonsingular (Axley, 1987). AIRNET is presently set up so that the ambient node pressure is zero causing the computed node pressures to be values relative to the true ambient pressure. This helps maintain numerical significance in calculating ΔP.

2.4 Convergence Criteria

Conservation of mass at each node provides the convergence criterion. That is, if $\Sigma\, w_i = 0$ for all nodes for the current system pressure estimate, the solution has converged. Many iterations can be saved and sufficient accuracy attained by testing for relative convergence at each node

$$ |\Sigma\, w_i| \; / \; \Sigma|w_i| < \epsilon \tag{6} $$

with a test to prevent division by zero. The size of ϵ can be established by considering the use of the calculated airflows, such as in an energy balance.

2.5 Convergence Acceleration

Numerical tests of Newton's method solution indicated occasional instances of very slow convergence, always with oscillating corrections on successive iterations. This is depicted graphically for the successive values of pressure at a single node in figure 3. In the case shown each successive pressure correction is a constant ratio of the previous correction. The observed corrections come close to this pattern. By assuming a constant ratio, it is simple to extrapolate the corrections to an assumed solution:

$$ P_n^* = P_n - C_n/(1-r) \tag{7} $$

where r is the ratio of C_n for the current iteration to its value for the previous iteration. This extrapolated value of node pressure is used in the next Newton iteration. At every other iteration, there are two pressure correction values which may be used for an extrapolation. This method is similar to a Steffensen iteration (Conte & de Boor, 1972, p 54) which is used with a fixed point iteration method for individual nonlinear equations.

The oscillating corrections have been observed by other investigators (Wood, 1981; Demuren, 1986). Demuren uses a constant relaxation factor of 0.5 to prevent the oscillations. The iteration correction method presented in equation (7) gives a variable factor. When the solution is close to convergence, Newton's method iterations converge quadratically. By limiting the application of equation (7) to cases where r is less than some value such as -0.5, it will not interfere with the rapid convergence. Tests by the author confirm that this is faster than the constant relaxation factor. It has not been proven that equation (7) will always lead to convergence, but it can be shown that it will not prevent convergence. Newton's method converges when the estimated solution values are within some distance, called the radius of convergence, of the correct solution. Applying equation (7) when $-1 < r < 0$, will cause a smaller correction than Newton's method, which, therefore, cannot force the iterations outside the radius of convergence.

4

The meanings of other values of r are also interesting. When $r < -1$, the solution diverges in an oscillatory fashion. When $r > 1$, the solution also diverges, but in a nonoscillatory manner. For $0 < r < 1$, the solution is approached from one direction. In all three cases, equation (7) applies as long as r is truly constant over several iterations. However, for the last case, this involves a true extrapolation of correction factor which is very sensitive to the accuracy of r. This is most extreme for the case of $r = 1$, which would cause an infinite correction.

2.6 Linear Initialization

Newton's method requires an initial set of values for the node pressures. These may be obtained by including in each airflow element model a linear approximation relating the flow to the pressure drop:

$$w_i = c_i + b_i \cdot \Delta P \tag{8}$$

Conservation of mass at each node leads to a set of linear equations of the form

$$[A] \{P\} = \{B\} \tag{9}$$

The coefficient matrix $[A]$ in equation (9) has the same sparsity pattern as $[J]$ in equation (2) allowing use of the same sparse matrix solution process for both equations. This initialization handles stack effects very well and tends to establish the proper directions for the flows. The linear approximation is conveniently provided by the laminar regime of the element models described below, but it also may be provided by a secant approximation to the actual nonlinear behavior.

The initialization has been made optional in AIRNET. When solving a set of similar problems, such as when the node temperatures or wind pressures are changed by small amounts, it may be preferable to use the previous solution for the node pressures as the initial values for the new problem.

3. ELEMENT MODELS

Flow within each airflow element is assumed to be governed by Bernoulli's equation:

$$\Delta P = (P_1 + \rho V_1^2/2) - (P_2 + \rho V_2^2/2) + \rho g(z_1 - z_2) \tag{10}$$

where
ΔP = total pressure drop between points 1 and 2
P_1, P_2 = entry and exit static pressures
V_1, V_2 = entry and exit velocities
ρ = fluid density
g = acceleration of gravity (9.81 m/s^2)
z_1, z_2 = entry and exit elevations.

The following parameters apply to the nodes: pressure, temperature (to compute density and viscosity), and height. The node height values are used to determine stack effect pressures. When the node represents a room, the

5

airflow elements may connect with the room at other than its reference height. Appendix C.3 shows how to use the hydrostatic equation to relate the pressure difference across a flow element to the heights of the element ends and the node heights, assuming the air in the room is at constant temperature. Pressure terms can be rearranged and a possible wind pressure for building envelope openings added to give

$$\Delta P = P_n - P_m + PS + PW \tag{11}$$

where

P_n, P_m = total pressures at nodes n and m
PS = pressure difference due to density and height differences, and
PW = pressure difference due to wind.

Equation (11) establishes a sign convention for direction of flow: positive is from node n to node m. Since the airflow elements will be described by a relationship of the form $w = f(\Delta P)$, the partial derivative needed for [J] in equation (4) are related by $\partial w/\partial P_m = -\partial w/\partial P_n$ which establishes the relation in equation (5).

3.1 Powerlaw Flow Elements

Most infiltration models are based on the following empirical (powerlaw) relationship between the flow and the pressure difference across a crack or opening in the building shell:

$$w_i = C \sqrt{\rho_n} \ (\Delta P)^x \qquad\qquad (\Delta P > 0) \tag{12a}$$

or

$$w_i = -C \sqrt{\rho_m} \ (-\Delta P)^x \qquad\qquad (\Delta P < 0) \tag{12b}$$

where

w_i = mass flow rate of air through element i from node n to node m,
C = flow coefficient,
ρ = air density of node n or m,
ΔP = total pressure loss across the element ($P_n - P_m$), and
x = flow exponent.

Theoretically, the value of the flow exponent should lie between 0.5 and 1.0. Large openings are characterized by values very close to 0.5, while values near 0.65 have been found for small crack-like openings. The form of equations (12) is suggested by the orifice equation:

$$Q = C_d \ A \ \sqrt{2\Delta P/\rho} \tag{13}$$

where

Q = volumetric flow ($Q = w/\rho$),
C_d = discharge coefficient, and
A = orifice opening area.

Equation (12) should be considered a correlation rather than a physical law. It can be used with the element leakage area formulation which has been used to characterize openings for infiltrations calculations (ASHRAE, 1985, p 22.16). The author has used it to describe flows through ducts to an accuracy of about 2% over a range of flow rates that vary by a factor of four. Such a variation would be found in a VAV system.

6

The primary advantage of equations (12) for describing airflow components is the simple calculation of the partial derivatives for the Newton's method solution of the simultaneous equations:

$$\frac{\partial w_i}{\partial P_n} = x \, w_i \, / \, \Delta P \tag{14a}$$

and

$$\frac{\partial w_i}{\partial P_m} = -x \, w_i \, / \, \Delta P \tag{14b}$$

However, there is also a problem with equations (14): the derivatives become unbounded as the pressure drop (and the flow) go to zero. A simple way to avoid this problem is suggested by what physically happens at low flow rates: the physical character of the flow (and the form of the equation) changes. It goes from turbulent to laminar. Equation (12) can be replaced by

$$w = K \, \rho \, \Delta P \, / \, \mu \tag{15}$$

where
   K = flow coefficient,
   μ = viscosity.

The partial derivatives are simple constants.

$$\frac{\partial w_i}{\partial P_n} = K \, \rho/\mu \tag{16a}$$

and

$$\frac{\partial w_i}{\partial P_m} = -K \, \rho/\mu \tag{16b}$$

The origin of this laminar relationship is shown by the duct equations in the next section. This technique has been independently discovered and used by several researchers (Axley, 1987; Isaacs, 1980). Although there is physical reason for using equation (15) at low pressure drops, its real purpose is to assure convergence of the equations when ΔP approaches zero for one of the many flow paths in a complex network, instead of accurately representing airflows which are too small to be of interest.

The AIRNET function for powerlaw elements calculates flows using both the laminar and the turbulent models, equations (12) and (15), and selects the method giving the smaller magnitude flow. There is a discontinuity in the derivative of the w(ΔP) curve where the two equations intersect. This discontinuity is a violation of one of the sufficient conditions for convergence of Newton's method (Conte & de Boor, 1972, p 86). However, numerical tests conducted by the author for flows at that point using a small airflow network have shown no convergence problem.

3.2 Ducts

The theory of flows in ducts (and pipes) is well established and summarized in the ASHRAE Handbook of Fundamentals (ASHRAE, 1985, ch 33). More extensive treatment is given by Blevins (1984) in a long chapter on pipe and duct flow. Analysis is based on Bernoulli's equation and its assumptions. The friction losses in a section of duct or pipe are given by

7

$$\Delta P_f = f \cdot L/D \cdot \rho V^2/2 \tag{17}$$

where
   f = friction factor,
   L = duct length, and
   D = hydraulic diameter.

The dynamic losses due to fittings and so forth are given by

$$\Delta P_d = C_o \; \rho V^2/2 \tag{18}$$

where
   C_o = dynamic loss coefficient.

Total pressure losses are given by

$$\Delta P = \Delta P_f + \Sigma \; \Delta P_d \tag{19}$$

Since $w = \rho VA$, where A is the cross section area,

$$w = [2\rho A^2/(fL/D + \Sigma \; C_o)]^{\frac{1}{2}} \cdot \Delta P^{\frac{1}{2}} \tag{20}$$

The friction factor can be computed using the nonlinear Colebrook equation (ASHRAE, 1985, p 2.9)

$$1/\sqrt{f} = 1.44 + 2 \cdot \log(D/\epsilon) - 2 \cdot \log\left[1 + \frac{9.3}{Re \cdot \epsilon/D \cdot \sqrt{f}}\right] \tag{21}$$

where
   ϵ = roughness dimension, and
   Re = Reynolds number = $\rho VD/\mu = wD/\mu A$.

This nonlinear equation may be readily solved using the following iterative expression derived from equation (21) by Newton's method:

$$g^* = g - [g - \alpha + \gamma \; \ln(1+g\beta)] \; / \; [1 + \gamma\beta/(1+g\beta)] \tag{22}$$

where
   $g = 1 \; / \; \sqrt{\bar{f}}$,
   $\alpha = 1.14 - \gamma \; \ln(\epsilon/D)$,
   $\beta = 9.3/(Re \cdot \epsilon/D)$, and
   $\gamma = 2 \cdot \log(e) = 0.868589$.

The convergent solution is achieved in 2 or 3 iterations of equation (22) using $g = \alpha$ as a starting value. If the value of g has been saved from the previous time it was computed for a particular duct element, and the flow rate has not changed greatly, only one iteration of equation (22) will be needed to compute the friction factor.

The exact derivatives of equation (20) are difficult to compute. However, reasonable convergence is achieved by assuming the the coefficients in equation (20) are constant giving

$$\frac{\partial w_i}{\partial P_n} = \tfrac{1}{2} \; w_i \; / \; \Delta P \tag{23a}$$

and

$$\frac{\partial w_i}{\partial P_m} = -\tfrac{1}{2} \; w_i \; / \; \Delta P \tag{23b}$$

The above description of duct flow applies only in the fully turbulent flow regime above a Reynolds number of about 4000. When the Reynolds number is below about 2000, the flow is laminar with the laminar friction loss described by

$$\Delta P_f = k/Re \cdot L/D \cdot \rho V^2/2$$

$$= \mu/\rho \cdot kL/(2AD^2) \cdot w \tag{24}$$

where
$\quad k$ = laminar friction factor.

Laminar dynamic losses are given by

$$\Delta P_d = K_o \ \rho V^2/2$$

$$= K_o/(2\rho A^2) \cdot w^2 \tag{25}$$

where
$\quad K_o$ = laminar dynamic loss coefficient.

Expressions (24) and (25) lead to a quadratic equation for mass flow in terms of total pressure drop:

$$a \ w^2 + b \ w + c = 0 \tag{26a}$$

where
$\quad a = K_o/(2\rho A^2)$
$\quad b = \mu kL/(2\rho AD^2)$
$\quad c = |\Delta P|$
giving
$\quad w = \text{sign}(\Delta P)[(b^2+4ac)^{\frac{1}{2}} - b] \ / \ 2a \tag{26b}$

The partial derivatives are given by

$$\frac{\partial w}{\partial P_n} = \frac{1}{(b^2+4ac)^{\frac{1}{2}}} \tag{27a}$$

$$\frac{\partial w}{\partial P_m} = \frac{-1}{(b^2+4ac)^{\frac{1}{2}}} \tag{27b}$$

The derivatives at $\Delta P = 0$ are finite (i.e. $\pm 1/b$).

3.3 Doorways

Flows through large openings (e.g. doorways) tend to be more complex with the possibility of flows in opposite directions in different parts of the opening. The temperature and resulting density differences between two rooms may mean that the stack effect causes a positive pressure difference at the top of the doorway and a negative pressure difference at the bottom (or vice versa). A summary of research on heat transfer through doorways is presented by Barakat (1987). Most research has attempted to develop dimensionless correlations (using Nusselt, Prandlt, and Grashoff numbers) of the form

$$Nu_D/Pr = C \cdot Gr_D{}^b \tag{28}$$

where
$\quad b$ is approximately 0.5 and

C lies between 0.22 and 0.33 depending on the temperature difference used for the correlation. It has been shown that such a heat transfer is equivalent to an airflow which can be modeled by powerlaw elements (Walton, 1982) by dividing the total opening into several smaller openings having the same total area but configured to properly account for the magnitude and direction of airflows at different heights in the opening.

An alternative approach is to create a single airflow element which accounts for the flow over the entire opening. A simple theory which estimates the stack induced air flow through a large opening in a vertical partition is given by Brown and Solvason (1962). The derivation of the doorway element is based on the model shown in Figure 4.

By assuming that the air density in each room is constant, the hydrostatic equation is used to relate pressures at various heights in each room:

$$P_{0n} = P_n + \rho_n g(h_n - h_0) \quad \text{and} \quad P_{0m} = P_m + \rho_m g(h_m - h_0) \tag{29}$$

$$P_n(y) = P_{0n} - \rho_n gy \quad \text{and} \quad P_m(y) = P_{0m} - \rho_m gy \tag{30}$$

Following Brown & Solvason (1962) it is assumed that the velocity of the airflow as a function of height is given by the orifice equation:

$$V(y) = C_d [2(P_n(y) - P_m(y))/\rho]^{1/2} \tag{31}$$

where
    C_d = discharge coefficient, and
    ρ = density of the air going through the area.

At the neutral height, Y, the velocity of the air is zero. From equation (31) this must occur when $P_n(y) = P_m(y)$. From equations (30)

$$Y = \frac{P_{0n} - P_{0m}}{g(\rho_n - \rho_m)} \quad \left(= \frac{P_{0m} - P_{0n}}{g(\rho_m - \rho_n)} \right) \tag{32}$$

If $0 < Y < H$, there is a two-way airflow through the doorway. If $\rho_n = \rho_m$, the neutral height cannot be computed, but, since there is no possibility of two-way flow, the doorway can be considered a simple orifice opening.

Define $\Delta\rho \equiv \rho_n - \rho_m$ and a transformed height coordinate $z \equiv Y - y$. Then the pressure difference across the doorway is given by

$$\Delta P(z) = -gz\Delta\rho \tag{33}$$

The mass flow through the doorway above the neutral height is given by

$$w_a = \int_{z=0}^{z=H-Y} (\rho V)_i \, W \, dz \qquad i = n \text{ or } m \tag{34}$$

and the mass flow below the neutral height by

10

$$w_b = \int_{z=-Y}^{z=0} (\rho V)_i \, W \, dz \qquad\qquad i = n \text{ or } m \qquad\qquad (35)$$

Whether the subscript i should be n or m depends on the direction of flow. Integration of equations (34) and (35) gives several different solutions for the airflow depending on the value of Y and the sign of $\Delta\rho$. Defining:

$$G \equiv \frac{2}{3} W \, C_d \, [2g|\Delta\rho|]^{1/2} \qquad f_a \equiv |H-Y|^{3/2} \qquad f_b \equiv |Y|^{3/2} \qquad (36a)$$

and

$$G' \equiv W \, C_d \, [2/(g|\Delta\rho|)]^{1/2} \qquad f_a' \equiv |H-Y|^{1/2} \qquad f_b' \equiv |Y|^{1/2} \qquad (36b)$$

gives the following equations for flows and derivatives.

Case 1: $Y \le 0$

$$\Delta\rho > 0: \qquad w = -G \sqrt{\rho_m} \, |f_a - f_b| \qquad \partial w/\partial P_n = G' \sqrt{\rho_m} \, |f_a' - f_b'| \qquad (37a)$$

$$\Delta\rho < 0: \qquad w = G \sqrt{\rho_n} \, |f_a - f_b| \qquad \partial w/\partial P_n = G' \sqrt{\rho_n} \, |f_a' - f_b'| \qquad (37b)$$

Case 2: $Y \ge H$

$$\Delta\rho > 0: \qquad w = G \sqrt{\rho_n} \, |f_a - f_b| \qquad \partial w/\partial P_n = G' \sqrt{\rho_n} \, |f_a' - f_b'| \qquad (37c)$$

$$\Delta\rho < 0: \qquad w = -G \sqrt{\rho_m} \, |f_a - f_b| \qquad \partial w/\partial P_n = G' \sqrt{\rho_m} \, |f_a' - f_b'| \qquad (37d)$$

Case 3: $0 < Y < H$

$$\Delta\rho > 0: \qquad w_a = -G \sqrt{\rho_m} \, f_a \qquad \partial w/\partial P_n = G' \sqrt{\rho_m} \, f_a' \qquad (37e)$$

$$w_b = G \sqrt{\rho_n} \, f_a \qquad \partial w/\partial P_n = G' \sqrt{\rho_n} \, f_b' \qquad (37f)$$

$$\Delta\rho < 0: \qquad w_a = G \sqrt{\rho_n} \, f_a \qquad \partial w/\partial P_n = G' \sqrt{\rho_m} \, f_a' \qquad (37g)$$

$$w_b = -G \sqrt{\rho_m} \, f_a \qquad \partial w/\partial P_n = G' \sqrt{\rho_m} \, f_b' \qquad (37h)$$

with $\partial w/\partial P_m = -\partial w/\partial P_n$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (37i)$

This model of a doorway tends to be faster that the multiple opening approach. However, it also complicates the assembly process for the Jacobian matrix because one or two flows may exist (i.e., case 3 above). More importantly, development of the doorway element model requires knowledge of the vertical temperature profile used in the node model (here assumed to be constant) in order to compute the pressure difference as a function of height across the opening. This requirement compromises the independence of the modularity of airflow network program.

3.4 Fans

The theory of flows induced by fans is summarized in the ASHRAE Equipment Handbook (ASHRAE, 1983, ch 3). More extensive treatment is given by Osborne (1977). Fan performance is normally characterized by a performance curve such

11

as shown in figure 5. This curve relates the total pressure <u>rise</u> to the flow rate for a given fan speed and air density. The fan performance curve is well represented by one or more cubic polynomials:

$$P = a_0 + a_1 w + a_2 w^2 + a_3 w^3 \tag{38}$$

Multiple polynomials might conveniently be obtained by a cubic spline fit to the performance data. There are two important factors to note on the shape of the fan performance curve. First, it is described by a relationship of the form $P(w)$ instead of $w(P)$ which would be more appropriate for the calculation of flow and partial derivatives. The basic shape of the performance curve cannot be well represented by a simple polynomial with P as the independent variable. Equation (38) requires an iterative solution to determine the flow rate. A modified false-position method (Conte & de Boor, 1972, p 31) works quickly and reliably. Fortunately, the derivative dw/dP is simply the inverse of dP/dw, which is a simple expression for a polynomial.

Second, it is common for the performance curve to contain points of contraflecture, with up to three different flow rates possible at certain values of fan pressure. This causes difficulty in solving for the flow rate and, more importantly, has points where the derivative goes to infinity. However, it is usually not recommended that the fan operate in the region of the contraflecture points. Therefore, the fan can be modeled with a performance curve that does not include the contraflecture so long as the user checks that the air distribution system does not permit operation in that region.

It is easy to identify the points of contraflecture from the coefficients of the polynomial by the evaluation of simple derivatives:

$$P = a_0 + a_1 w + a_2 w^2 + a_3 w^3 \tag{39a}$$

$$P' = a_1 + 2a_2 w + 3a_3 w^2 \tag{39b}$$

$$P'' = 2a_2 + 6a_3 w \tag{39c}$$

$P' = 0$ at the points of contraflecture. Solving equation (39b) for w gives:

$$w = (-2a_2 - \sqrt{4a_2{}^2 - 12a_1 a_3} \;) \; / \; 6a_3 \tag{40}$$

If $(4a_2{}^2 - 12a_1 a_3) > 0$, there are two real points of contraflecture, with equation (39c) defining the highest root (a_3 must be negative to give the typical fan curve). If $P''(w) > 0$, the point is a maximum; if $P'' = 0$, it is a point of inflection.

The performance of a given fan at various speeds and air densities can be related to a single fan performance curve through the "fan laws".

$$w_1 \; / \; w_2 = (N_1 \rho_1) \; / \; (N_2 \rho_2) \tag{41}$$

and

$$P_1 \; / \; P_2 = (N_1{}^2 \rho_1) \; / \; (N_2{}^2 \rho_2) \tag{42}$$

where
    w = volume flow rate,
    P = total pressure rise,
    N = rotational speed,

ρ = density.

These laws are valid if all flow conditions at the two speeds are similar. In particular, they will not apply at very low flows where fully turbulent conditions have not been developed.

Numerical tests with AIRNET for flows at the laminar-turbulent transition indicate some convergence difficulty: about twice as many iterations as usual are needed for convergence. In one case the iterations showed potential divergence with $r < -1$, but the convergence acceleration algorithm saved the cases tested and produced a solution.

3.5 Quadratic Flow Elements

Baker, Sharples, and Ward (1987) indicate that infiltration openings can be more accurately modeled by a quadratic relationship of the form

$$\Delta P = A\,Q + B\,Q^2 \qquad\qquad (Q, \Delta P \geq 0) \qquad\qquad (43a)$$
and
$$\Delta P = A\,Q - B\,Q^2 \qquad\qquad (Q, \Delta P \leq 0) \qquad\qquad (43b)$$

This form can be used as an airflow element by solving the quadratic equation for w (= ρQ). Letting $a = A/\rho$ and $b = B/\rho^2$ allows equations (43a,b) to be rewritten as

$$\Delta P = a\,w + b\,w^2 \qquad\qquad (w, \Delta P \geq 0) \qquad\qquad (44a)$$
and
$$\Delta P = a\,w - b\,w^2 \qquad\qquad (w, \Delta P \leq 0) \qquad\qquad (44b)$$

These quadratic equations solve as

$$w = (\sqrt{a^2 + 4b\Delta P} - a)\,/\,2b \qquad\qquad (\Delta P \geq 0) \qquad\qquad (45a)$$
and
$$w = (a - \sqrt{a^2 - 4b\Delta P})\,/\,2b \qquad\qquad (\Delta P \leq 0) \qquad\qquad (45b)$$

The partial derivatives are given by

$$\partial w/\partial P_n = +\,1\,/\,(a + 2b|w|) \qquad\qquad\qquad (46a)$$

$$\partial w/\partial P_m = -\,1\,/\,(a + 2b|w|) \qquad\qquad\qquad (46b)$$

Equations (45) and (46) require that a and b both be nonzero to prevent a division by zero. There is no problem as ΔP and w go to zero -- the derivatives are finite.

4. SAMPLE CALCULATIONS

Several simple airflow networks have been analyzed to demonstrate the procedure described in the previous sections.

13

4.1 Simple Test Cases

Figure 6 shows two cases involving powerlaw elements in series. The first case, consisting of three nodes and two flow elements, can be considered as modeling a room with small 0.01 m^2 openings on opposite sides with wind pressure driving flow through the room. The second case divides the single room into two room with a partition containing a large 2.00 m^2 opening. This case with a very low resistance (large opening) mixed with large resistances (small openings) is difficult to solve with some methods (Walton, 1982; Clarke, 1985, p 206). In both cases AIRNET required only two iterations and computed the expected nearly identical flows.

Figure 7 shows three cases involving doorway elements. In the first case a 0.8 m by 2.0 m doorway connects two rooms with a 4°C temperature difference. The computed two-way airflow is 0.259 kg/s. In the second case ten 0.16 m^2 airflow openings at different heights are used to represent a doorway. The computed two-way airflow is 0.261 kg/s. The third case represents six rooms in series connected by doorway elements. The computed flows are identical to the first case. All three cases were solved in two iterations.

Figure 8 shows a test involving two fans in parallel. This is a problem in the textbook by Osborne (1977, p 75). The computed pressures agree with the text to within 2 Pa (about 0.5%) and the flows to within 1.5%. These differences are probably due to the inaccuracies in the polynomial fit for the fan performance curve and in the graphical solution used in the text.

Figure 9 shows a test with two fans in series (Osborne, p 76). The computed room pressure differ from the value reported by Osborne by 2.5 Pa and the airflows differ by less than 1%.

Figure 10 shows one floor of a 36-room airflow network created to test execution time for a larger network. This test case describes a four story building with six rooms, a hallway, an elevator shaft, a stairwell, and a node representing ambient on each floor. The nodes representing the elevator shaft and stairwell on each floor are connected by very large (2.0m^2) openings. Similar openings connect each room to the hallways. Very small (0.01m^2) openings connect the building nodes to the outside. Intermediate size openings (0.1m^2) connect the large vertical shafts to the hallways. This case was solved, with a 0.01% convergence criterion, in 5 iterations and 2.89 seconds on a PC compatible computer (4.77 MHz 8088 CPU with 8087 math coprocessor). This is about 16 times faster than the predecessor to AIRNET, the AIRMOV program (Walton, 1984), could solve this sample problem.

Appendix B discusses some of these and other test cases in greater detail.

4.2 A Comparison of Methods

The ESP building thermal simulation program (ABACUS, 1986) includes a separate program, ESPAIR, for calculating airflows. ESPAIR was compared to the AIRNET program. Both programs were recompiled and run on a workstation computer using the 36-room test case. ESPAIR solved this case using default 5% convergence in about 8800 iterations requiring a total of 150 seconds. AIRNET solved it using the default 0.01% convergence in 5 iterations requiring 0.16 seconds, or about 1000 times faster.

This extreme difference in calculation times occurs partially because of the difficulty which the ESPAIR algorithm has with large openings (Clarke, 1985, p 206). Limiting all the openings to an area of $0.01m^2$ allowed ESPAIR to reach a solution in only 137 iteration and 2.10 seconds. AIRNET was also somewhat faster for this case: 2 iterations and 0.06 seconds, or about 35 times faster than ESPAIR. Greater accuracy in the ESPAIR solution (0.5% convergence) required more iterations (22,000) and more time (400 seconds). This is a particularly extreme comparison, since it was concerned with a relatively large problem and large interroom openings where ESPAIR is weakest and does not consider overhead calculations such as I/O, but it does demonstrate the potential of the new method. The ESPAIR results may explain why airflow network calculations have a reputation for being slow.

5. DIRECTIONS FOR FUTURE RESEARCH

5.1 Alternate Solution Methods

Although the simple tests of the AIRNET program and its comparison to ESPAIR look very promising, some important questions remain. The most important question concerns the reliability of the method for solving the airflow network equations. Solution of the nonlinear equations has been demonstrated in several tests but has not been mathematically proven. The literature for the solution of similar equations may be helpful. The airflow network is very similar to a pipe network with the flow resistance of openings and ducts corresponding to the resistance of pipes and fans corresponding to pumps.

Much of the theory for computing fluid flows in pipe networks is described by Jeppson (1976). The basic flow phenomena are nonlinear and must be described by a set of nonlinear algebraic equations. These equations may be expressed in terms of the unknown flows in the pipes (referred to as loop equations) or the unknown heads at the junctions (node equations). The equations are derived from a form of Kirchoff's circuit laws: (1) the sum of flows into a junction equals the sum of outward flows, and (2) the total headloss around any loop in the system must be zero. Wood and Rayes (1981) give an excellent comparison of several algorithms. Five methods are described and tested; three are based on the loop equations and two on the node equations. The least reliable methods (those least likely to converge to the correct solution) are the method that adjusts each loop flow individually, the method that adjusts each node head individually, and the method that adjusts the node heads simultaneously.

It is interesting to note that ESPAIR solves the airflow problem with a version of the algorithm which adjusts node heads individually. Among the airflow algorithms used in smoke control algorithms, Klote & Fothergill (1983) use individual node head adjustments while Sander (1974) uses the simultaneous node head adjustment algorithm, both of which are among the least reliable methods, according to Wood and Rayes. The method used in AIRNET also does simultaneous node head adjustment, but it is so different that it should be evaluated separately. It addresses the two problems observed by Wood and Rayes: (1) Large openings (low resistances) give inexact flows because small differences in the computed pressures lead to large differences in the flows. This is solved by stringent requirements on mass balance convergence at each

15

node. Such accuracy is not costly, because Newton's method is quadratically convergent -- near the solution each iteration greatly improves the accuracy. (2) Failure of the node adjustment method to converge because of oscillating corrections is handled by the Steffensen iteration applied to the Newton's method correction factors.

The two simultaneous loop methods have a good history of convergence for pipe network problems. On the other hand, they are more difficult to set up than the node methods since independent loops must be defined; they tend to require the solution of more simultaneous equations; the equations do not have the very desirable feature of diagonal dominance; they tend to be less sparse than the node equations; and some airflow elements may be difficult to implement. The doorway model may be difficult because it can have either one or two flows which may make it especially difficult to define the loops.

Of particular interest to the idea of establishing a general modular program is that the loop methods require the airflow elements to compute pressure drop as a function of flow rate, which is opposite the requirement for the node method. For some of the airflow elements, such as powerlaw elements, the transformation is simple. Others are described more naturally in one form than the other. For example, the duct and fan models are described more naturally for the loop method. This indicates the need to consider the solution technique in the development of element models.

The work of Wood and Rayes indicates that several apparently reasonable solution methods for the simultaneous nonlinear equations are not very reliable. The ideal solution to the question of reliability would be mathematical proofs of the convergent nature of the solution algorithm and the limitations on the element models. Such proofs may be difficult to achieve for nonlinear systems. Alternatively, extensive tests of different methods would give some confidence as to their reliability.

5.2 Other Element Models

The modular structure of AIRNET would allow many more airflow elements to be developed. These elements could provide either new capabilities or more accurate simulation. The necessary requirements for element models are $w(\Delta P)$ uniquely defined for all ΔP, and bounded derivatives for all ΔP.

It appears possible to represent dampers as variable flow resistance elements in an airflow network. The relationship between resistance and actuator position could be represented by a polynomial. The flow characteristics of some airflow elements may depend significantly on the direction of flow. In pipe networks check valves act in such a manner. These could be represented by elements with separate performance curves applied to different pressure drop or flow regimes.

Much more work could be done on the development of the doorway models. Complex flow patterns involving boundary layer flows can exist. These patterns are related to the geometries and surface and air temperature distributions in the adjoining rooms. For example, Hill (1986) uses a model which incorporates nonuniform temperatures in the rooms which leads to multiple neutral pressure levels in the doorway and compares the computed

16

airflows to measured flows. Here the intimate relation between the doorway element model and the node models is important. The constant temperature node model could be expanded to three more complex models: (1) temperature varies uniformly with height, (2) two uniform temperature layers in the room, and (3) two layers each having uniformly varying temperature. It may be necessary to develop several doorway models to account for different type of airflow. Detailed doorway calculations would then involve methods to identify which model to use.

The experimental data base for two-way flows between nodes at different heights (through stairs and elevator shafts) appears insufficient to develop element models. It should be possible to extend the airflow network method to include 2- and 3-dimensional fluid elements for the detailed modeling of airflows within rooms. Of course, this would greatly increase computation time.

5.3 General Limitations

The simple airflow network method outlined above has some inherent limita-tions. These include inability to quickly model airflow patterns within a room or to model the transient airflows caused by short-term transients in wind pressure distributions. These effects can possibly be approximated by dividing rooms into several nodes and adding transient flows to the average flows, but the direct modeling of such effects would greatly increase calculation time and would probably be impractical for most engineering analysis. The existence of such known limitations, not to mention unknown factors, makes experimental validation of airflow network calculations essential.

It must be expected that uncertainty in the input parameters will always limit the absolute accuracy of airflow calculations. However, a network model based on physical laws will be useful for evaluating design alternatives because relative changes in flow values should be fairly accurate. Modularity can be used further in the design of an airflow analysis program. Figure 11 shows a structure for such a program similar to that used in ESP (ABACUS, 1986). The program separates the evaluation of wind pressures from the airflow calculation to allow alternate inputs: manual entries, measured values, or simulated values. Input of airflow element data would involve a data base of element data. The computed airflows go to an output file which could be used in either indoor air quality or loads calculations. The entire procedure could be incorporated into an energy analysis program.

6. SUMMARY AND CONCLUSIONS

This report has discussed how an airflow network method can be used to provide a unified model of major building airflows. Of particular importance is the idea of modularity. ESP's modularity made the comparison test possible. It is often very difficult to isolate a single computational feature of a monolithic program. AIRNET includes modularity of the airflow elements, allowing elements with greatly different flow characteristics to be connected to the core algorithm by a common interface. More airflow elements could be added. The sparse matrix solution of the simultaneous equations

17

involving the Jacobian matrix allows larger systems of equations to be handled without the full execution time penalty of using the complete matrix. By separating the solution and matrix assembly processes, faster solution processes could be easily substituted.

The performance of AIRNET relative to ESPAIR indicates that it is practical to solve the flow network in detail. Solution of complex airflow networks for the steady-state case is practical on current small computers. Solution of the dynamic case for many timesteps is now possible. The use of small computers will make advanced user input features available which could significantly aid in the airflow analysis process.

Research is still needed in several areas. These include determination of the most reliable airflow network solution method, a mathematical analysis of the network flow equations and the solution method, development of additional airflow elements (especially improved large opening models), experimental validation of the simplifying assumptions in the element models and network method, expansion of the wind pressure and airflow element performance database, and modeling of intraroom effects by simplified methods and by integration with microscopic modeling methods.

7. REFERENCES

ABACUS, 1986. "ESP, A Building and Plant Energy Simulation System, Version 5, Release 3", University of Strathclyde, Scotland.

ASHRAE, 1983. ASHRAE Handbook - 1983 Equipment, ASHRAE, Atlanta GA.

ASHRAE, 1985. ASHRAE Handbook - 1985 Fundamentals, ASHRAE, Atlanta GA.

Axley, J., 1987. Indoor Air Quality Modeling Phase II Report, US National Bureau of Standards report NBSIR 87-3661.

Axley, J., 1988. "Multi-Zone Dispersal Analysis by Element Assembly", submitted to Building and Environment.

Baker, P.H., Sharples, S., & Ward, I.C., 1987. "Air FLow Through Cracks", Building and Environment, Vol. 22, No 4, Pergamon, pp. 293-304.

Barakat, S.A., 1987. "Inter-Zone Convective Heat Transfer in Buildings: A Review", ASME Journal of Solar Engineering, Vol. 109, May.

Blevins, R.D., 1984. Applied Fluid Dynamics Handbook, Van Nostrand Reinhold.

Brown, W.G. & Solvason, K.R., 1962. "Natural Convection Through Rectangular Openings in Partitions - 1: Vertical Partitions", International Journal of Heat and Mass Transfer, Vol. 5, pp. 859-867.

Clarke, J.A., 1985. Energy Simulation in Building Design, Adam Hilger Ltd., Bristol and Boston.

Conte, S.D. & de Boor, C., 1972. Elementary Numerical Analysis: an Algorithmic Approach, McGraw-Hill.

Dhatt, G., Touzot, G., & Catin, G., 1984. The Finite Element Method Displayed, John Wiley & Sons, New York.

Demuren, A.O. & Ideriah, F.J.K., 1986. "Pipe Network Analysis by Partial Pivoting Method", Journal of Hydraulic Engineering, ASCE, Vol. 112, pp. 327-333.

Hill, D., Kirkpatrick, A. & Burns, P., 1986. "Analysis and Measurements of Interzone Natural Convection Heat Transfer in Buildings", Journal of Solar Energy Engineering, ASME, Vol. 108, August.

Isaacs, L.T. & Mills, K.G., 1980. "Linear Theory Methods for Pipe Network Analysis", Journal of the Hydraulics Division, Proceedings ASCE, July, pp. 1191-1201. (See also author's closure, Jan 1982, p. 153)

Jeppson, R.W., 1976. Analysis of Flow in Pipe Networks, Ann Arbor Science, Ann Arbor, MI, 1976.

Klote, J.H. & Fothergill, J.W.Jr., 1983. "Design of Smoke Control Systems for Buildings", ASHRAE, Atlanta GA.

Liddament, M. & Thompson, C., 1982. "Mathematical models of air infiltration -a brief review and bibliography", Technical Note AIC 9, The Air Infiltration Centre, Bracknell, England.

Persily, A.K. & Grot, R.A., 1985. "The Airtightness of Office Building Envelopes", Proceedings of the ASHRAE-DOE-BTECC Conference on the Thermal Performance of the Exterior Envelopes of Buildings III, Clearwater Beach FL.

Osborne, W.C., 1977. Fans, 2nd edition, Pergamon Press.

Said, M.N.A., 1988. "A Review of Smoke Control Models", ASHRAE Journal, Vol. 30, No 4, April.

Sander, D.M., 1974. "Fortran IV Program to Compute Air Infiltration in Buildings", NRC Canada, Div. Bldg. Res., Computer Program No. 37.

Swami, M.V. & Chandra, S., 1988. "Correlations for Pressure Distribution on Buildings and Calculation of Natural-Ventilation Airflow", ASHRAE Transactions, Vol. 94, Part 1.

Walton, G.N., 1982. "Airflow and Multiroom Thermal Analysis", ASHRAE Transactions, Vol. 88, Part 2.

Walton, G.N., 1984. "A Computer Algorithm for Predicting Infiltration and Interroom Airflows", ASHRAE Transactions, Vol. 90, Part 1.

Wood, D.J. & Rayes, A.G., 1981. "Reliability of Algorithms for Pipe Network Analysis", Journal of the Hydraulics Division, ASCE, Vol. 107, pp. 1145-1161.
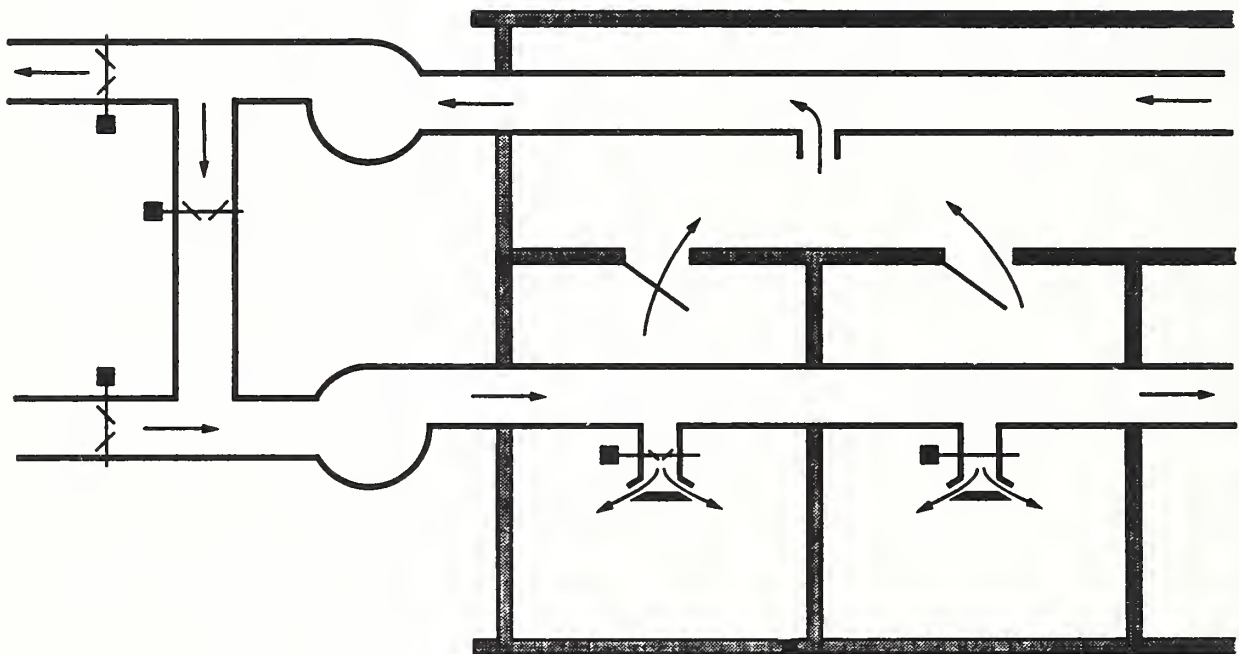
Figure 1.   Plan of a Portion of a Building Served by a VAV System

20

F - Fan elements
V - Damper elements
D - Doorway elements
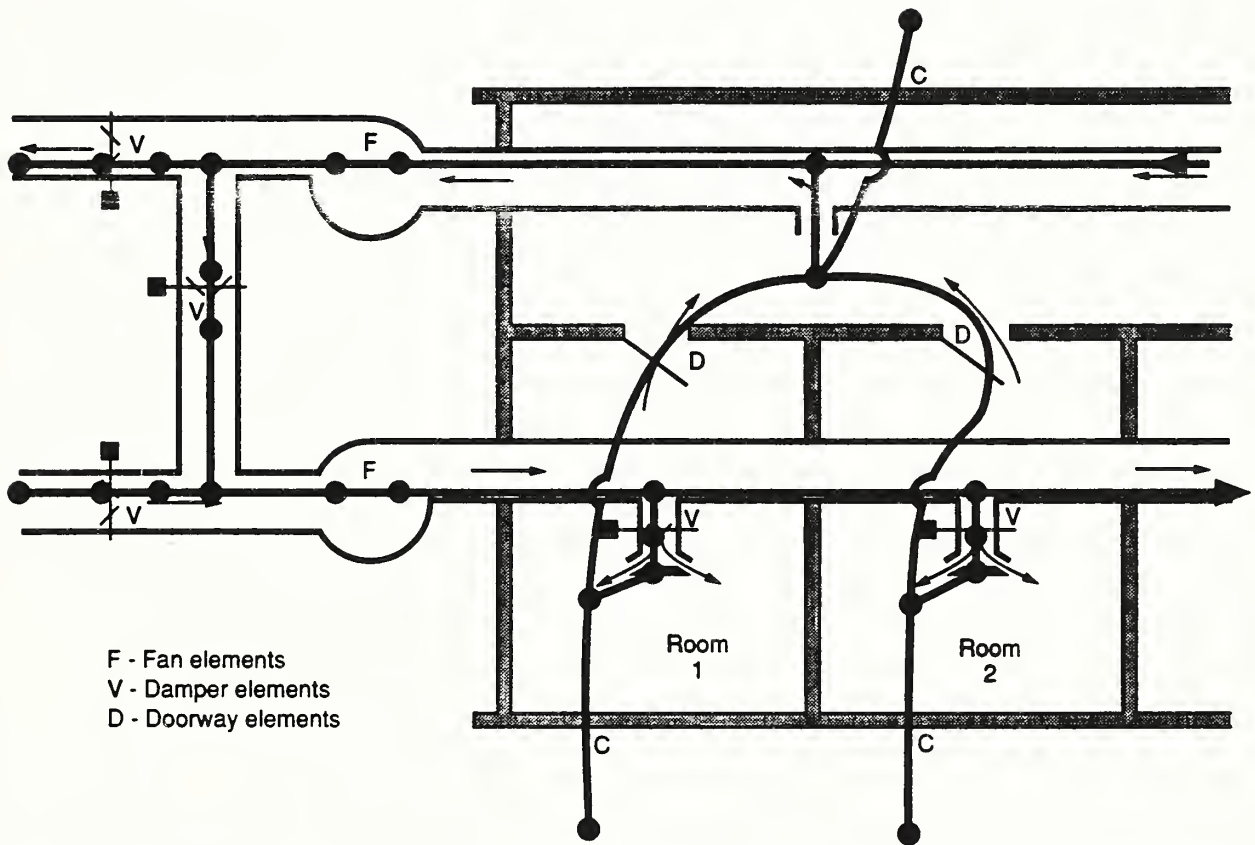
Figure 2. Airflow Network for VAV System of Figure 1

21
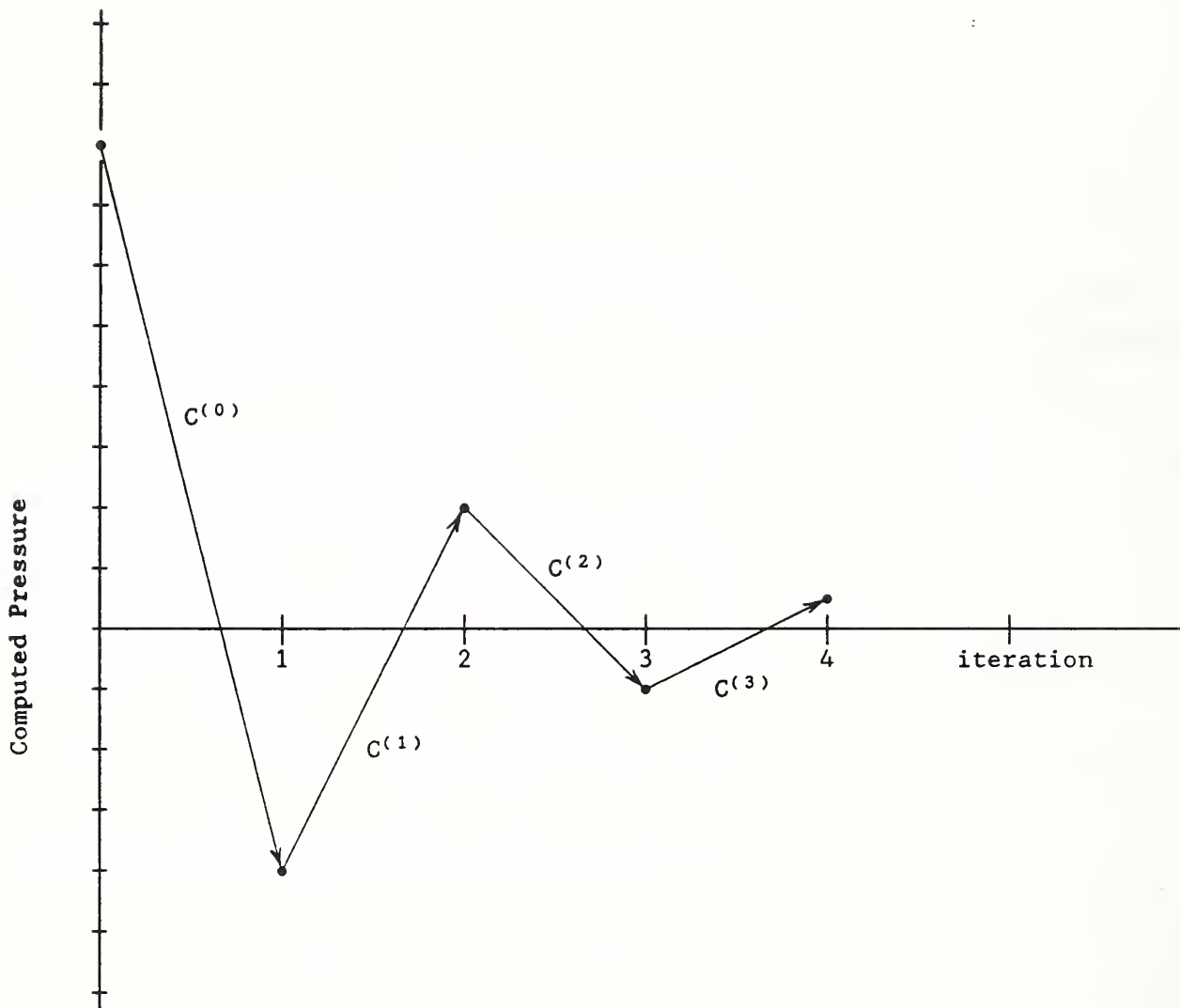
Figure 3. Oscillating Pressure Corrections
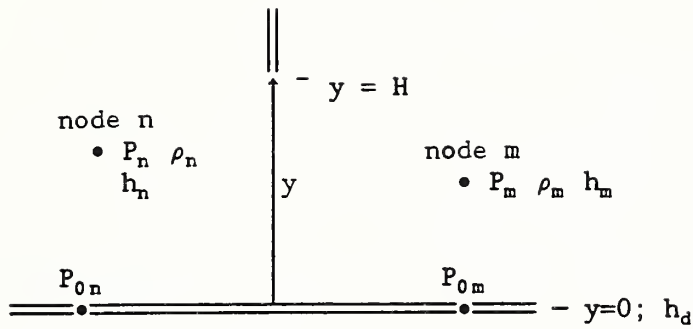
with

h_d = height of the base of the doorway,

h_n, h_m = heights of nodes n and m,

P_n, P_m = reference pressures of rooms (nodes) n and m,

ρ_n, ρ_m = air densities of rooms (nodes) n and m,

P_{0n}, P_{0m} = reference pressures of rooms n and m at base of doorway,

H = height of doorway, and W = width of doorway.
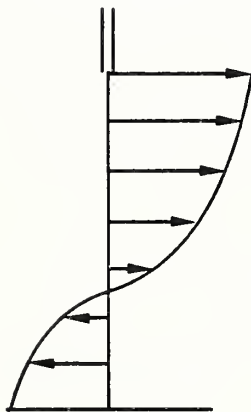
neutral plane
within doorway
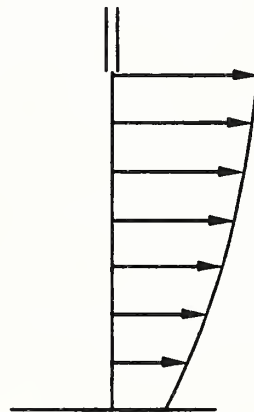
neutral plane
below doorway
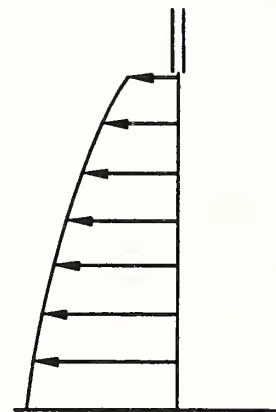
neutral plane
above doorway

Figure 4.   Doorway Coordinate System and Three Flow Patterns

23
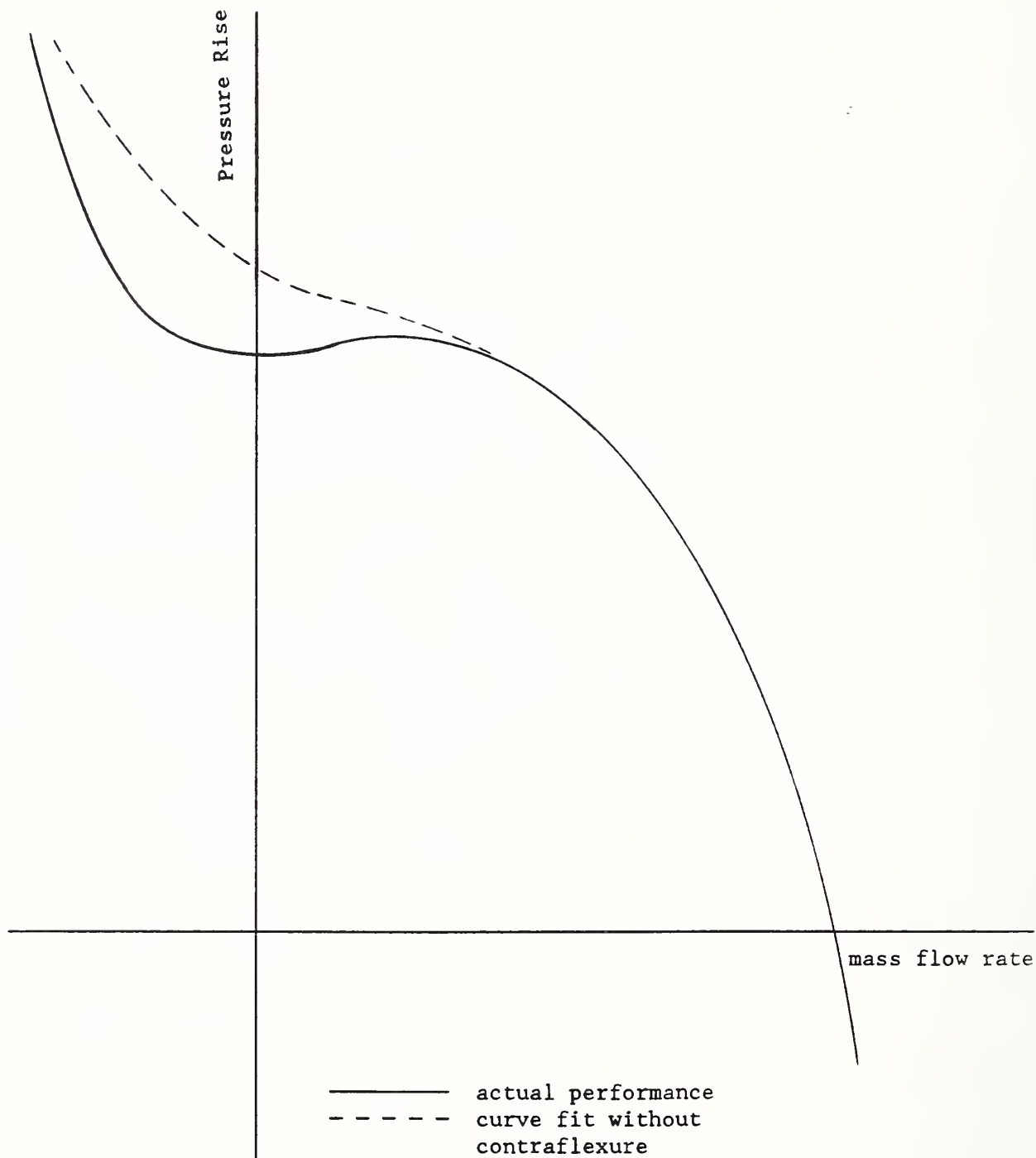
Figure 5. Typical Fan Performance Curve

24

Figure 6. Powerlaw Airflow Elements in Series

25
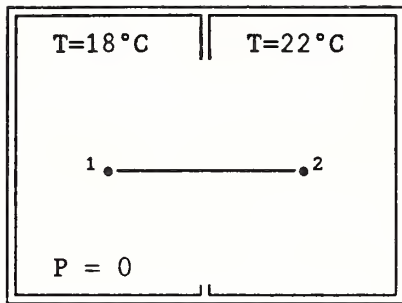
Doorway simulated
by doorway element
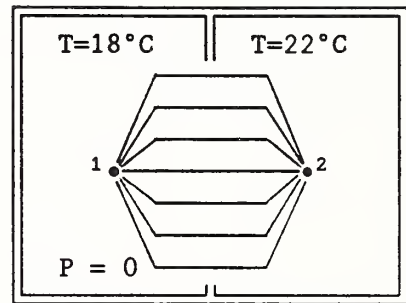
Doorway simulated
by multiple
powerlaw elements
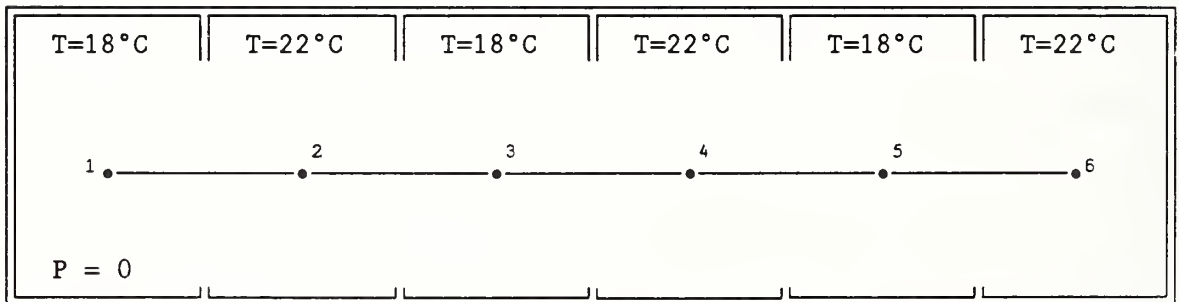
Five doorway elements in series

Figure 7.   Three Airflow Networks for Doorways
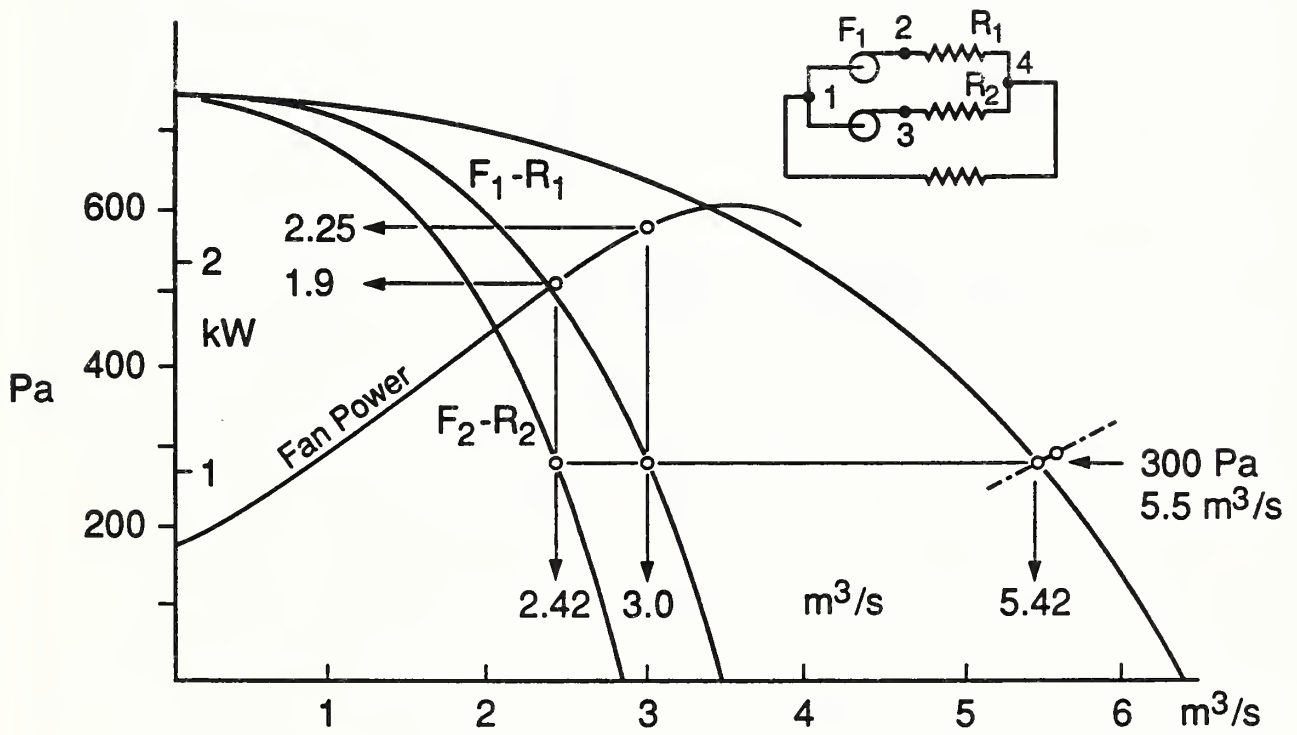
Figure 8. Fans in Parallel (from Osborne)
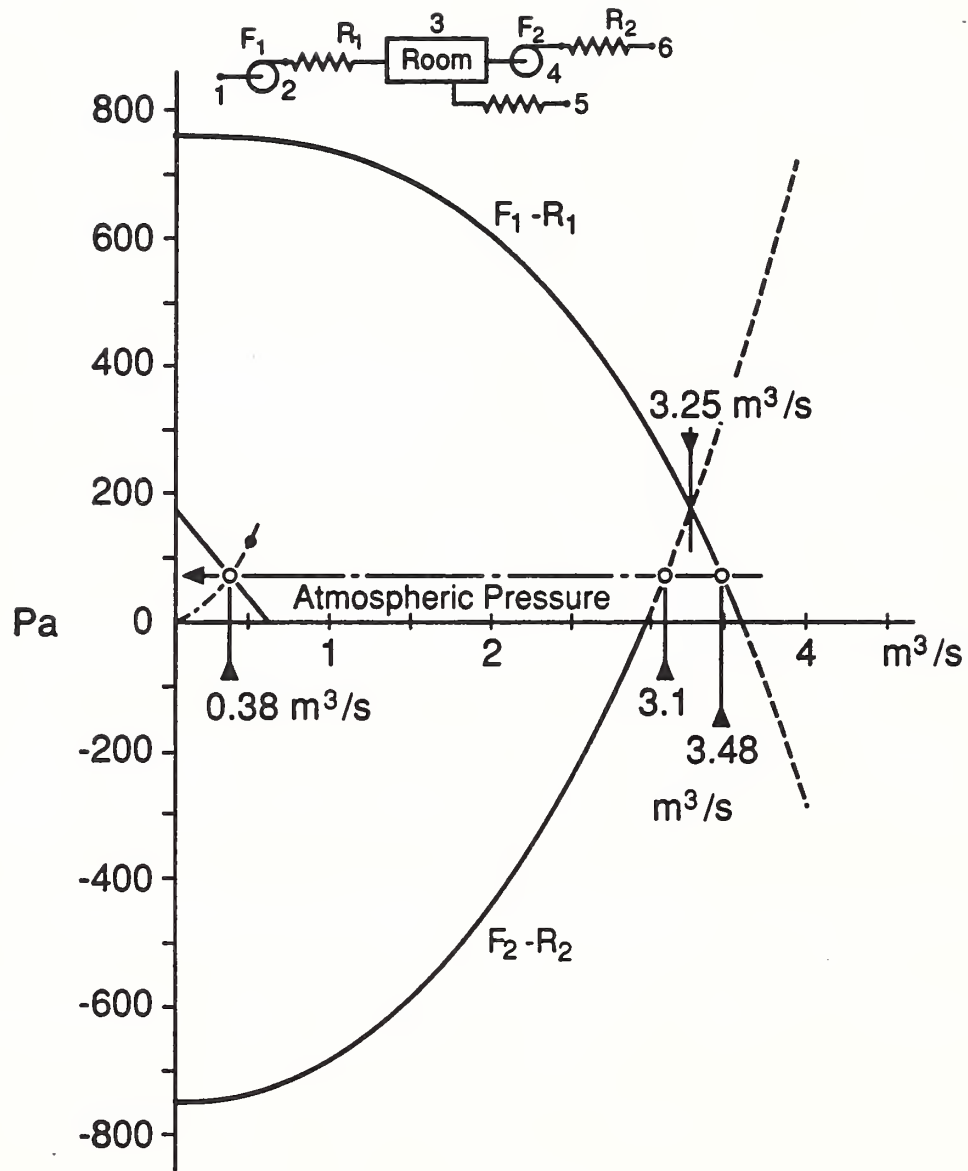
Figure 9. Fans in Series (from Osborne)

28
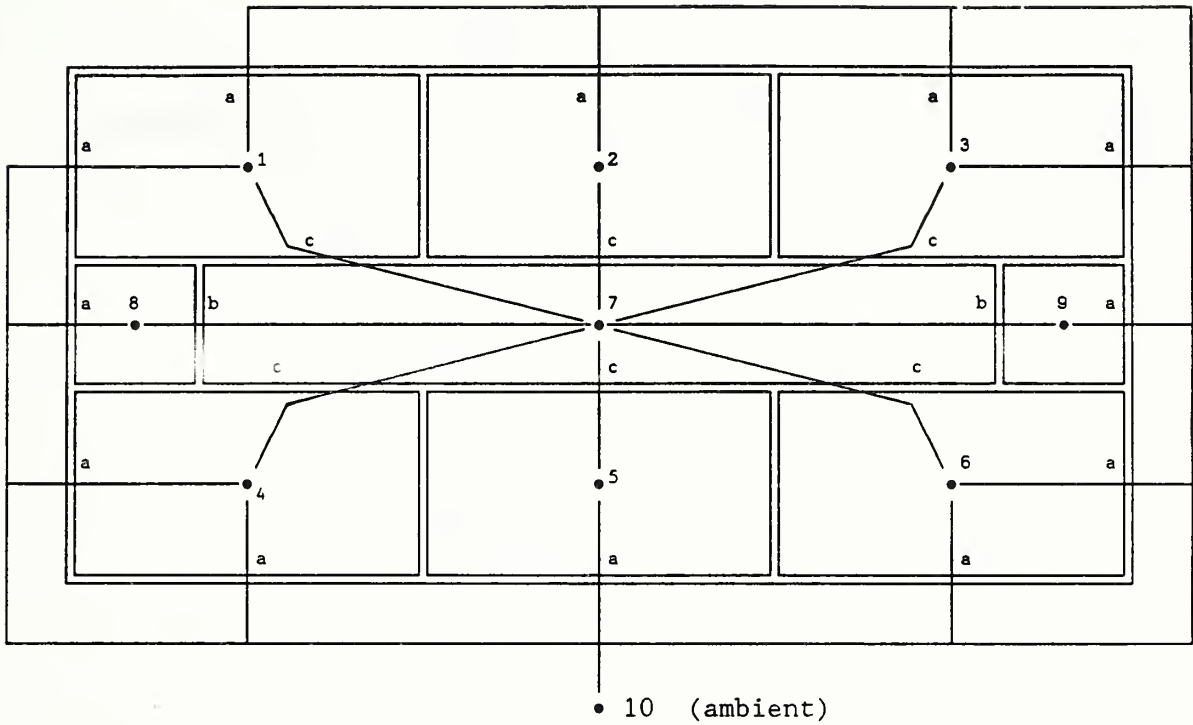
Elements: a = 0.01 m² orifice
              b = 0.10 m² orifice
              c = 2.00 m² orifice

Nodes 8 and 9 connected to similar nodes
on adjacent floors by type c openings

Figure 10.   One Floor of the 4 Floor / 36 Room Timing Test Case
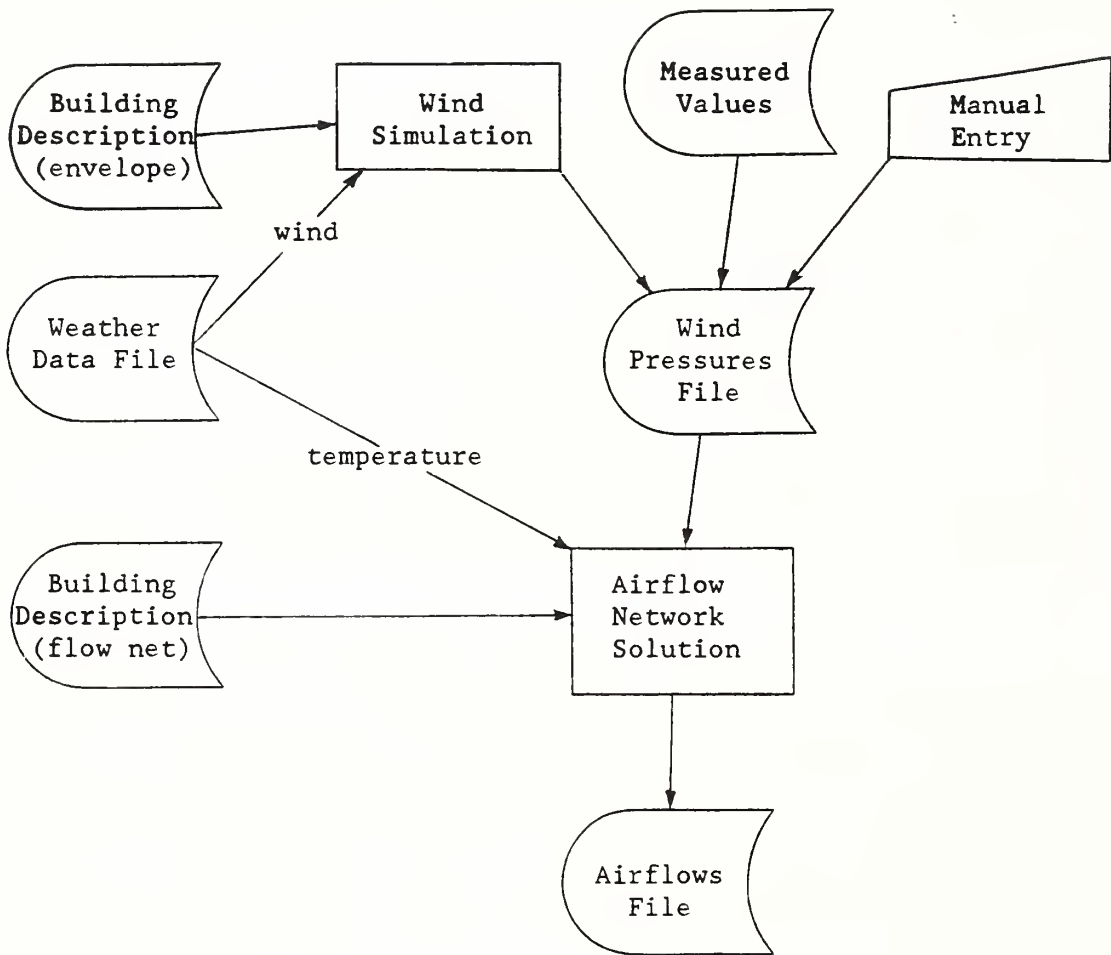
Figure 11. Proposed Structure of Airflow Network Program

APPENDIX A: AIRNET User's Manual

A.1 Introduction

     AIRNET is a program for testing airflow network calculations which might
be used for modeling of infiltration, natural ventilation, interroom, and
mechanical system airflows. The program has been written for IBM PC* and
compatible computers using the Turbo C v 2.0** compiler. The C source code is
included on the distribution disk to allow modification of the program for
research. Execution of the program is controlled interactively, but most of
the problem data is stored in files which are created by the user.

A.2 Interactive input

     There are two primary forms of interactive input. The first consists of
a question of the form

> Question? (y/n)

to which the user can press the Y and ENTER keys for a "yes" response or N
and ENTER for a "no". The second form allows the entry of numeric data:

> enter parameter [min = P1, max = P2, default = P3]

where the information in the brackets in the minimum permitted value, the
maximum and the default. Pressing only enter is equivalent to entering the
default value. If a default is not given, a value must be entered.

     Execution of AIRNET begins with several questions about the input and
output files. The user must first provide the name of the network data file,
which contains the description of the airflow network and elements. Second,
the name of the wind pressure coefficients file is entered. Third is an
option to name an output file which will hold the results of the calculations.
The names of the input files are written to the output file to help document
the simulation.

     The user may then enter several run control parameters. The following
parameters are requested:
(1) "output control flag". Higher values cause more writing to the output
file: 0 gives node pressures; 1 adds element flow rates; 2 adds data on the
iterations; 3 adds an echo of the input data which can be useful in locating
errors in the input files; higher numbers can be used to dump intermediate
calculations if the program is recompiled with appropriate debugging
parameters defined.
(2) "skip initialization flag". 0 means use the linear initialization
method; 1 means use the current values for node pressures to begin the
Newton's method.
(3) "maximum iterations". This is the number of Newton's method iterations
tried until it is assumed the solution is not converging.

_____

* Internation Business Machines Personal Computer - copyright/trademark
** Turbo C version 2.0 by Borland International - trademark

(4) "maximum pressure change (Pa)". Limit the maximum change in node pressure during a single Newton iteration.
(5) "relative airflow convergence". The solution at a node is assumed convergent if | Σ flows | / Σ |flows| < this factor.
(6) "absolute airflow convergence (kg/s)". The solution at a node is assumed convergent if | Σ flows | < this factor.
(7) "convergence acceleration limit". If the ratio of successive pressure corrections is less than this limit, use Steffensen acceleration algorithm. Processing of the run control parameters ceases when the user responds that all values are correct.

The user may then enter several weather values related to wind pressure.
(1) "ambient temperature (C)".
(2) "barometric pressure (Pa)". These two values are used to determine the air density. The barometric pressure should be the absolute pressure for the site, i.e. not corrected to sea level.
(3) "wind speed (m/s)".
(4) "wind direction". This is the direction, in degrees clockwise from north, that the wind is blowing from.
Again, processing of the weather data ceases when the user responds that all values are correct. See section A.4 for details on the wind pressure calculation.

Both data files are then processed including the allocation of memory to store the simulation data. This allows the size of the problem to be limited only by the amount of available memory. Warnings and error messages may be given. The user can then print information relative to the structure of the Jacobian matrix. This will indicate how efficiently the matrix is stored for a sparse solution. See section A.5 for more details on this subject.

The user can interactively change some of the network description data from the input file. The user can change the values of all control values from their default settings of 1. See the information with the airflow elements for uses of control values. The revised network description parameters may be printed. The user must then explicitly allow the airflow network calculations to proceed. This allows a chance to revise any of the above data.

After computing the airflows, the user can continue to execute AIRNET. The program returns to the point where the run control parameters are reviewed. The user can change these parameters, the weather data, the control values, or some of the airflow network parameters to create a modified problem, which can then be solved. The following network parameters can be altered: the height, temperature, or pressure of each node; the link heights, element, wind profile, or wind pressure modifier of each link.

A.3 The AIRNET data files

Most of the data required to simulate an airflow network is contained in two data files. Files are used, rather than interactive input, because of the potential size and complexity of airflow network problems. ASCII files are used for portability.

A.3.1 Wind coefficients file

The wind coefficients file contains data relating the wind pressure to the wind direction. The first line of the file is a title which is echoed to the output. The remaining lines each consist of a profile name (of up to 15 characters) followed by 16 wind pressure modifiers corresponding to wind directions of 0°, 22.5°, 45°, 67.5°, etc. See section A.4 for details on the wind pressure calculation. Each wind profile must have a different name.

A.3.2 Network definition file

The AIRNET network data file contains a complete description of the airflow network. The file contains node data, element data, and linkage data.

It is intended that the network data file be self documenting. Node, element, and linkage names may each be up to 15 character long (no imbedded blanks). Names should be chosen for physical significance. In addition, an optional comment can be placed on any line after the last data entry. Empty lines may be inserted to improve readability. An asterisk in the first column of a line may be used to signify end-of-data; any information after that is not processed. The first line of data is a title which is echoed on the output listings.

In the following description keywords are shown underlined.

A.3.2.1 Node data

The nodes in the airflow network can represent either rooms or ductwork linkage points. Each node description has the following form:

node name type ht temp pres

node   - This word identifies the following data as node data.
name   - The name identifies the node for later reference and in the output listings. Each node must have a different name.
type   - node type (single character) (v = variable or unknown pressure, c = constant or known pressure, a = ambient node) The ambient node is a constant pressure node whose temperature is set to ambient temperature by the program.
ht   - reference height (m) at which the node pressure is computed.
temp   - node air temperature (C).
pres   - node pressure (Pa) (this value is not required for type v nodes).

Notes:
There must be at least one constant pressure (or ambient) node in the network in order to have a solution to the simultaneous equations. In addition, every node must be connected by some path to a constant pressure node. This condition is tested by the program.

The sequence of nodes determines the sparsity of the Jacobian matrix for the Newton's method solution and can, therefore, significantly effect execution time. The nodes can be easily reordered with a text processor. See section A.5 for more details.

A.3.2.2 Element data

Each airflow element begins with an identifier (element), a name, and an element type followed by performance parameters on one or more lines. Each element must have a different name.

Powerlaw element:

element name plr init lam turb expt

init - coefficient for linear initialization (this may be the same as lam);
lam   - coefficient for laminar flow;
turb - coefficient for turbulent flow;
expt - pressure difference exponent;

Detailed duct element (Darcy-Weisback model / Colebrook friction correlation):

element name dwc len dh area rgh
              tdlc lflc ldlc init

len   - length (m);
dh   - hydraulic diameter (m);
area - cross section area (m^2);
rgh   - roughness dimension (m);
tdlc - turbulent dynamic loss coefficient;
lflc - laminar friction loss coefficient;
ldlc - laminar dynamic loss coefficient;
init - laminar initialization coefficient.

Doorway element:

element name dor init lam turb expt
              dtmin ht wd cd

init - expt: powerlaw coefficients for low temperature difference;
dtmin - minimum temperature difference for two-way flow (C);
ht   - height of doorway (m);
wd   - width of doorway (m);
cd   - doorway discharge coefficient.

Constant flow element:

element name cfr flow

flow   - rated mass flow (kg/s);
Control parameter = actual mass flow / rated mass flow.

Note that since flow is not a function of pressure, constant flow elements cannot be counted as part of the linkage of variable pressure nodes to constant pressure nodes (A.3.2.1).

34

Detailed fan element:

```
element   name   fan   init   lam   turb   expt
          rdens  fdf   sop    ltt   nr    mf1
          a11    a12   a13    a14   mf2
          a21    a22   a23    a24   mf3
          ...                       mfn
```

init - expt: powerlaw coefficients for fan speed = 0;
rdens   - reference fluid (air) density (kg/m^3);
fdf   - free delivery flow (kg/s);
sop   - shutoff pressure (Pa);
ltt   - laminar/turbulent transition (RPM/rated RPM);
nr   - number of ranges for performance curve;
mf1   - minimum mass flow of range 1 (kg/s);
a11-a14: range 1 polynomial coefficients for f(p) at rated RPM;
mf2   - maximum mass flow rate of range 1 (kg/s) =
          minimum mass flow of range 2;
a21-a24: range 2 polynomial coefficients for f(p) at rated RPM;
mf2   - maximum mass flow rate of range 2 (kg/s);
...
mfn   - maximum mass flow rate of range nr (kg/s);
Control parameter = actual RPM / rated RPM

Constant power fan element:

```
element   name   cpf   upo
```

upo   - useful power output of fan (W);
Control parameter = actual power / rated power.

Quadratic flow element:

```
element   name   qfr   a   b
```

a   - flow coefficient;
b   - flow^2 coefficient;

A.3.2.3 Linkage data

     The linkage data defines the airflow network. Each linkage description
consists of the identifier (link) followed by six or seven parameters.

link name node-1 ht-1 node-2 ht-2 element wind wpmod

name   - This user assigned name identifies the link in the output listings.
node-1 - name of the first node. This refers to the names which were
          assigned with the node data.
ht-1   - height (m) of the linkage point relative to the first node height.
node-2 - name of the second node. Node-1 and node-2 define the direction of
          airflow: flow from node-1 to node-2 is positive, the reverse flow

35

```
                is negative.
ht-2      - height (m) of the linkage point relative to the second node height.
element   - airflow element name.  This refers to the names which were assigned
            with the element data.
wind      - wind pressure coefficients profile.  This is the name of the profile
            given in the wind pressure coefficients file.  The name "null" is
            reserved to mean that there is no wind pressure on a given link.
wpmod     - This wind pressure modifier is applied for the particular link.
            See section A.4 for details.
```

Note that the wind coefficient profile name must exist in the wind
coefficients file and that the node and element names must be defined in the
network file before they are referenced by a link.

A.4 Wind Pressure Calculations

The wind pressure, PW_i, acting on link i is proportional to the kinetic
energy of the airstream: $\frac{1}{2}\rho_a V_a^2$. The ambient air density, ρ_a, is determined
from the weather data parameters: temperature and barometric pressure. The
wind velocity, V_a, is also a weather data parameter. The wind pressure is
also a function of the wind direction, D. D is a weather data parameter; the
function is given in the wind coefficients file in terms of 16 wind pressure
modifiers for evenly spaced wind directions. The wind direction pressure
modifier, f(D), is determined from the wind coefficients by linear interpo-
lation. Wind speed, and therefore pressure, also varies with height and local
shielding of the surface. These variations can be included in the link wind
pressure modifier, wpmod. Therefore, the wind pressure acting at a given link
is

$$PW_i = \text{wpmod} \cdot f(D) \cdot \tfrac{1}{2}\rho_a V_a^2 \qquad\qquad\qquad (A.1)$$

This procedure should give considerable flexibility in defining the wind
pressure, but it does not say what coefficients should be used. That is left
to the user.

One particular aspect of the input should also be noted. A positive wind
pressure tends to increase flow in the positive direction and vice versa for a
negative value of wind pressure. Flow direction is determined by the ordering
of the two nodes in the link command. Therefore, in order for positive wind
pressures to push air into the building and negative wind pressures to pull
air out, each link which represents an opening in the envelope of the building
should be defined with the ambient node first and the interior node second.

A.5 Equation Ordering in the Jacobian

The sequence in which the nodes are listed in the network data file
determines the ordering of the simultaneous equations and pattern in which the
Jacobian is filled. Different equation ordering can significantly effect the
sparsity of the Jacobian and, therefore, the memory requirements and execution
time of the program. Memory requirements increase linearly with the number of
nonzeros in the Jacobian; computation time increases more than linearly. See
chapter 2 of Sparse Matrix Technology (Academic Press, 1984) by S. Pissanetzky
for a detailed discussion of this topic.

Consider the following example. Nodes B, C, D, and E are linked to node A but not to each other. If only node A is constant pressure and are ordered A → E, the diagram of the upper part of the symmetric Jacobian will look like (using • to indicate initial nonzero values, o to indicate zeros, and + for new nonzero values after factoring):

```
• • • • •              • • • • •
  • o o o                • + + +
    • o o                  • + +
      • o                    • +
        •                      •
```

before and after factoring, respectively, showing that this numbering scheme results in non-zero elements, or "fill", below the diagonal.

If the nodes are ordered B → E, A, the diagram of the Jacobian will look like

```
• o o o •
  • o o •
    • o •
      • •
        •
```

both before and after factoring, thus completely avoiding fill. See section B.10 for another example of equation reordering.

To reduce the computations associated with fill, a rule of thumb for the ordering of nodes is that the node which is linked to the greatest number of other nodes should appear later in the node list. In buildings consisting of many similar levels this is partially accomplished by grouping all nodes on each level together.

A.6 Sample Output File

The following output file for a simple test network illustrates its contents by notes which are indicated in brackets: [..] .

powerlaw test input file   [1]
wind coefficients file - no profiles

Run control data:
                    output control flag: 3   [2]
               skip initialization flag: 0
                      maximum iterations: 20
                maximum pressure change: 500 Pa
            relative airflow convergence: 0.0001
            absolute airflow convergence: 1e-06 kg/s
       convergence acceleration limit: -0.5

Weather data:
            ambient temperature: 20 C
            barometric pressure: 101325 Pa
                     wind speed: 0 m/s
                 wind direction: 0 (N=0, E=90, ...)

```
node    node-1    c    0.0    20.0    50.0                                    [3]
node    node-2    v    0.0    20.0
node    node-3    v    0.0    20.0
node    node-4    c    0.0    20.0   -50.0

element   orf-0.0001 plr 7.2e-9    7.2e-9     8.48528e-5   0.5  orf-0.0001 m^2
element   orf-0.0100 plr 7.2e-6    7.2e-6     0.00848528   0.5  orf-0.01 m^2

link    link-1    node-1    0.0    node-2    0.0    orf-0.0001    null
link    link-2    node-2    0.0    node-3    0.0    orf-0.0100    null
link    link-3    node-3    0.0    node-4    0.0    orf-0.0001    null

********

Time to read data files:   0.44                                              [4]

     1234
  1: *                                                                        [5]
  2:  **
  3:   *
  4:    *
     1234

Initial fill fraction:  0.500000

Final fill fraction:    0.500000

Number of nonzero elements in AU: 2
Number of nonzero elements in AL: 2

Unallocated memory:  54212 bytes                                             [6]
```

node	temp	dens	visc	
node-1	20.000000	1.204742	0.000018	[7]
node-2	20.000000	1.204148	0.000018	
node-3	20.000000	1.204148	0.000018	
node-4	20.000000	1.203554	0.000018	
link	sp	wp	dp	
link-1	0.000000	0.000000	0.000000	
link-2	0.000000	0.000000	0.000000	
link-3	0.000000	0.000000	0.000000	

```
Initialization
Pn:     5.0000e+01     3.7316e-02     -1.2647e-02    -5.0000e+01            [8]
Begin iteration 1
Rev:   2  1.422954e-03   3.414537e-02  1.000000e+00  3.170580e-03          [9]
Rev:   3 -1.422954e-03  -3.416223e-02  1.000000e+00  2.151550e-02
Begin iteration 2
Rev:   2 -1.919686e-03  -2.790587e-02  5.502771e-01  1.852654e-02          [10]
Rev:   3  1.919686e-03   2.793733e-02  5.501204e-01  6.146604e-03
Begin iteration 3
Rev:   2  3.775709e-04   4.514769e-03  1.000000e+00  1.401177e-02
Rev:   3 -3.775709e-04  -4.508156e-03  1.000000e+00  1.065476e-02
```

```
Begin iteration 4
Rev:    2 -1.189833e-04 -7.423793e-04  1.000000e+00  1.475415e-02
Rev:    3  1.189833e-04  7.383248e-04  1.000000e+00  9.916436e-03
Begin iteration 5
Rev:    2 -1.083831e-05 -7.782953e-05  1.000000e+00  1.483198e-02
Rev:    3  1.083835e-05  8.408471e-05  1.000000e+00  9.832351e-03
Begin iteration 6
Fin:    1  0.000000e+00  0.000000e+00  1.204742e+00  5.000000e+01          [11]
Fin:    2 -8.919034e-08  1.316844e-03  1.204148e+00  1.483198e-02
Fin:    3  8.914789e-08  1.316844e-03  1.204148e+00  9.832351e-03
Fin:    4  0.000000e+00  0.000000e+00  1.203554e+00 -5.000000e+01
```

```
Time to compute node pressures:   0.66                                     [12]
Number of iterations:  6
```

```
        Link:  i   n   m   e     pdrop        flow-1        flow-2   [13]
        link-1: 1   1   2   1  4.998517e+01  6.584668e-04  0.000000e+00
        link-2: 2   2   3   2  4.999629e-03  6.583776e-04  0.000000e+00
        link-3: 3   3   4   1  5.000983e+01  6.584668e-04  0.000000e+00
```

```
        Node:  n     pres          sumf                              [14]
        node-1: 1  5.000000e+01  -6.584668e-04
        node-2: 2  1.483198e-02   8.917414e-08
        node-3: 3  9.832351e-03  -8.917414e-08
        node-4: 4 -5.000000e+01   6.584668e-04
```

Notes:
[1] This is the echo of the input file titles.
[2] These are the values of the run control parameters and weather data.
[3] This is the echo of the input files as set by the output control flag.
[4] The input processing time is accurate to ±0.02 second.
[5] This sketch shows the structure of the Jacobian. *'s indicate array
positions filled by the airflow element functions; +'s indicate positions
filled when the matrix is factored (See Appendix C.1). The two fill fractions
compare the positions in the sparse matrix to a full square matrix. AU and AL
refer to the upper and lower triangular matrices.
[6] Memory is allocated to store the problem parameters and variables and the
various arrays. The unallocated memory indicates that larger problems can be
accommodated.
   Notes 7 through 11 are for output generated by an output control flag >= 2.
[7] The air temperature, density, and viscosity are reported for each node.
Nodes are identified by name and number, which is the sequence in which they
occur in the input file. The stack pressure, wind pressure, and total
pressure drop are reported for each link.
[8] These are the pressures computed by linear initialization and which are
used to start the Newton's method iterations.
[9] At each iteration 4 values are printed for each variable pressure node:
the sum of the flows (kg/s) into the node, the computed pressure correction
factor, a correction acceleration factor (1.0 = no correction), and the new
estimate of node pressure.
[10] On the second iteration oscillating pressure corrections have been
observed for both nodes and correction acceleration factors computed.

[11] When the convergence criteria have been met, 4 values are printed for all nodes: Σ flows, Σ |flows|, air density, and node pressure. (Flows are not computed for the constant pressure nodes.)
[12] The time to compute air properties and node pressures is accurate to ±0.02 second.
[13] The airflows through each link are reported if the output control flag >= 1. The following values are printed for each link: the link name, link number, 1st node number, 2nd node number, element number, pressure drop across the link, mass flow rate (kg/s), 2nd mass flow rate (possible for doorways).
[14] The final node values are reported if the output control flag >= 0. The following values are printed for each node: the node name, node number, the node pressure (Pa) relative to ambient, and the sum of the flows into the node and the sum of the flows ignoring direction of flow.

A.7 Compiling AIRNET

A.7.1 Distribution diskette

    The distribution diskette is a standard 360K byte floppy disk containing the following files:

file	size (k bytes)	description
AIRNET.EXE	58	program executable on a PC compatible microcomputer
COMSEP.C	10	compilable code for the preprocessor program
AIRNET.ALL	126	all header files and functions for the AIRNET program
ELEMENT.EXE	56	program to help generate element models
ELEMENT.STR	6	file used by ELEMENT.EXE to limit program length
ELEMENT.ALL	59	all header files and functions for the ELEMENT program
NOWIND		wind coefficients file containing no wind profiles
WIND		wind coefficients file with some wind profiles
AFDATA.PL1		input data for powerlaw element test #1
AFDATA.PL2		input data for powerlaw element test #2
AFDATA.PL3		input data for powerlaw element test #3
AFDATA.ST1		input data for stack effect test #1
AFDATA.ST2		input data for stack effect test #2
AFDATA.WP1		input data for wind pressure test #1
AFDATA.WP2		input data for wind pressure test #2
AFDATA.DW1		input data for duct element test #1
AFDATA.DW2		input data for duct element test #2
AFDATA.DR1		input data for doorway element test #1
AFDATA.CF1		input data for constant flow element test #1
AFDATA.CF2		input data for constant flow element test #2
AFDATA.FN1		input data for fan element test #1
AFDATA.FN2		input data for fan element test #2
AFDATA.FN3		input data for fan element test #3
AFDATA.QF1		input data for quadratic flow element test #1
AFDATA.QF2		input data for quadratic flow element test #2
AFDATA.08F		input data for 8-floor execution time test

A.7.2 The COMSEP program

The COMSEP program is used to assist in the compilation of the AIRNET and ELEMENT programs. COMSEP combines and separates files involved in a project and creates several useful project files. First, compile COMSEP. COMSEP automatically creates files names based on the project name entered by the user. COMSEP presents the user several possible actions to be performed.

(1) "combine subfiles": This option takes a list of filenames and combines them into a single file. Each of the individual files must be headed by a single line of the form:

/*subfile filename ******/

Note that this line is in the form of a C comment and does not effect compilation of the file.

(2) "separate into subfiles": This option takes a single file and separates it into individual files based on header lines indicated above. AIRNET.ALL must be separated for compilation. This process creates the list of subfiles which is used with the other processes. Subfile names may not be duplicated.

(3) "create compile batch file": This option creates a batch file for use with the command-line version of Turbo C, TCC.EXE. Be sure to include the ".bat" extension in the name of the batch file. The compile options used to create AIRNET.EXE were "-ms -c -d -f87 -G -O -Z". Other options may be appropriate for a particular computer or problem. In particular, the -ml large memory option will allow the program to use the entire memory of the computer for solving large problems. Large problems almost require the use of a math coprocessor.

(4) "create link response file": This option creates a response file for use with TLINK.EXE to link the object files and libraries. A particular directory structure is assumed (see below), but the link response file can be edited to suit the user's individual needs. AIRNET.EXE was linked with the small memory (s) and math emulation options. Emulation uses the coprocessor if one is present. The coprocessor greatly improves execution time. Compiling with the math coprocessor option reduces the program size by about 10k bytes. It does not significantly improve execution speed, but will cause the program to fail if a math coprosessor is not present. Therefore, the emulation option was chosen for the distribution version of AIRNET.EXE.

(5) "create library response file": This option creates a response file for use with TLIB.EXE for the creation of an object library.

(0) "exit": Exit the preprocessor program.

A.7.3 Compilation

The following directory structure is assumed:
```
...\TURBOC          contains the Turbo C programs;
...\TURBOC\INC      contains the include files;
...\TURBOC\LIB      contains the libraries and C0x.OBJ files;
...\TURBOC\AIRNET   contains the AIRNET program files.
```
The file TURBOC.CFG must be appropriately defined and \TURBOC must be set in the PATH command.

Use COMSEP to split AIRNET.ALL into subfiles, create a compile file, and create a link response file. AIRNET.ALL contains three header files. Copy these files to the \INC directory. Execute the compile batch program to create object files. Use the linker and link response file to create an executable version of AIRNET.

The header file AIRNET.H includes several definable parameters (TESTI, TESTC, and TESTS) which were used to test the program during development. When these parameters are not defined, the dump functions (dumps.c, dumpv.c, dumpa.c, dumpf.c, and dumpn.c) are not called and can be removed from the link response file to reduce the size of AIRNET.EXE. These functions increase the size of the program by about 5k bytes. They are not included in AIRNET.EXE on the distribution diskette.

A.7.4 The ELEMENT Program

The ELEMENT program is an aid for the creation of airflow elements for the AIRNET program. All input is interactive; an output file will include element descriptions that can be transferred to the AIRNET network data file by a word processor. Both ELEMENT.EXE and ELEMENT.STR must be in the default directory when ELEMENT is executed. The compilation process for ELEMENT.ALL is the same as for AIRNET.ALL.

There are presently ten capabilities:
1: convert the physical characteristics of an orifice to a plr element;
2: convert the physical characteristics of a crack to a plr element;
3: convert one or two pressure/flow test points to a plr element;
4: convert the physical characteristics of a duct to a plr element;
5: convert the physical characteristics of a duct to a dwc element;
6: convert the physical characteristics of a doorway to a dor element;
7: convert the performance data of a fan to a fan element;
8: convert the physical characteristics of a crack to a qfr element;
9: convert two pressure/flow test points to a qfr element;
10: convert the physical characteristics of a duct to a qfr element.

APPENDIX B: AIRNET Validation Tests

B.1 Introduction

Validation is an essential part of the development of a computer simulation. Inaccurate results are not always due to program errors as shown in the SERI report on validation of building energy analysis programs (1983) which identified seven error sources classified into two groups. External sources are those which are not under the control of the developer of the computer code. These errors include:
1. differences between the actual weather around the building and the weather used in the simulation;
2. differences between the actual effect of occupant behavior and those effects assumed by the user;
3. user error, including inappropriate simplifying assumptions, in deriving the input files; and
4. differences between the actual thermal and physical properties of the building and those input by the user.
    Internal error sources are those contained within the coding of the program. They include:
1. differences between the actual heat/mass transfer mechanisms and the algorithmic representations of those mechanisms;
2. differences between the actual interactions of heat/mass transfer mechanisms and those interactions between the algorithms; and
3. coding errors.
    Three types of tests have been used to validate building energy analysis programs. One is comparison to other simulation programs. Another is comparison to analytically calculated results. The third is comparison to experimental data. The following table from the SERI validation report summarizes the advantages and disadvantages of each method.

VALIDATION TECHNIQUES

Technique	Advantages	Disadvantages

Comparative
  Relative test
  of different
  programs
	No input uncertainty
	Any level of complexity
	Inexpensive
	Many comparisons possible
		No truth standard

Analytical
  Test of
  numerical
  solution
	No input uncertainty
	Exact truth standard
	  given the simplicity
	  of the model
	Inexpensive
		Does not test the model
		Limited to cases for
		  which analytical
		  solutions can be
		  derived

Empirical
  Comparison to
  measured building
  performance
	Approximate truth
	  standard within accuracy
	  of data acquisition
	Any level of complexity
		Measurement involves some
		  input uncertainty
		High quality, detailed
		  measurements are time
		  consuming & expensive

This report will concentrate on analytic validation tests which are best for revealing coding errors.

B.1.1 Simplified Analysis of Airflow Elements in Parallel and in Series

It will be necessary to develop analytic solutions for simple airflow networks to test the AIRNET program. The analysis of DC electric networks is simple because the elements are linear. The following discussion will show a similar means of network analysis for nonlinear airflow elements.

(a) Laminar flow: $w = (\rho/\mu) \ K \ \Delta P$

Parallel:
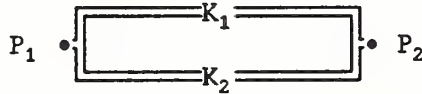
Flows through elements 1 and 2 are given by:

$$w_1 = (\rho/\mu) \ K_1 \ (P_1-P_2) \qquad\qquad w_2 = (\rho/\mu) \ K_2 \ (P_1-P_2)$$

$$w_1 + w_2 = (\rho/\mu) \ (K_1+K_2) \ (P_1-P_2)$$

$$w_1 + w_2 = (\rho/\mu) \ K_e \ (P_1-P_2) \qquad \text{where} \quad K_e = K_1 + K_2$$

Therefore, for elements in parallel: $\boxed{K_e = \sum K_i}$ (B.1.1)
    as long as ρ and μ are constant
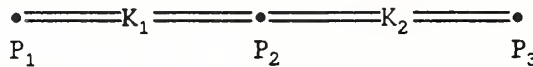
Series:

Pressure drops through elements 1 and 2 are given by:

$$P_1-P_2 = (\mu/\rho) \ w_1 \ / \ K_1 \qquad\qquad P_2-P_3 = (\mu/\rho) \ w_2 \ / \ K_2$$

$$P_1-P_3 = (P_1-P_2)+(P_2-P_3) = (\mu/\rho) \ (w_1/K_1 + w_2/K_2)$$

Since $w_1 = w_2 \equiv w$

$$P_1-P_3 = (\mu/\rho) \ w \ (1/K_1 + 1/K_2)$$
or
$$w = (\rho/\mu) \ K_e \ (P_1-P_3) \qquad \text{where} \ 1/K_e = 1/K_1 + 1/K_2$$

Therefore, for elements in series: $\boxed{1/K_e = \sum 1/K_i}$ (B.1.2)
    as long as ρ and μ are constant

These relationships are, of course, identical to the relationships for electric conductances.

(b) Turbulent flow: $w = \sqrt{\rho} \ C \ \sqrt{\Delta P}$

Parallel:
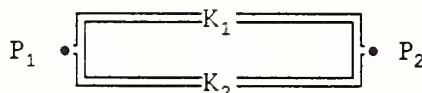
Flows through elements 1 and 2 are given by:

$$w_1 = \sqrt{\rho}\ C_1\ \sqrt{P_1 - P_2} \qquad\qquad w_2 = \sqrt{\rho}\ C_2\ \sqrt{P_1 - P_2}$$

$$w_1 + w_2 = \sqrt{\rho}\ (C_1 + C_2)\ \sqrt{P_1 - P_2}$$

$$w_1 + w_2 = \sqrt{\rho}\ C_e\ \sqrt{P_1 - P_2} \qquad \text{where}\quad C_e = C_1 + C_2$$

Therefore, for elements in parallel: $\boxed{C_e = \sum C_i}$ (B.1.3)
  as long as ρ is constant

Series:

$$\bullet\!=\!=\!=\!=\!C_1\!=\!=\!=\!=\!\bullet\!=\!=\!=\!=\!C_2\!=\!=\!=\!=\!\bullet$$
$$P_1 \qquad\qquad\qquad P_2 \qquad\qquad\qquad P_3$$
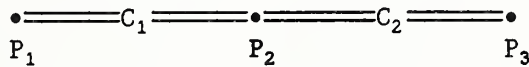
Pressure drops through elements 1 and 2 are given by:

$$P_1 - P_2 = (1/\rho)(w_1/C_1)^2 \qquad\qquad P_2 - P_3 = (1/\rho)(w_2/C_2)^2$$

$$P_1 - P_3 = (P_1 - P_2) + (P_2 - P_3) = (1/\rho)[(w_1/C_1)^2 + (w_2/C_2)^2]$$

Since $w_1 = w_2 \equiv w$

$$P_1 - P_3 = (1/\rho)\ w^2(1/C_1^2 + 1/C_2^2)$$
or
$$w = \sqrt{\rho}\ C_e\ \sqrt{P_1 - P_3} \qquad \text{where}\quad 1/C_e^2 = 1/C_1^2 + 1/C_2^2$$

Therefore, for elements in series: $\boxed{1/C_e^2 = \sum 1/C_i^2}$ (B.1.4)
  as long as ρ is constant

These relationships will allow the creation of relatively complex airflow
networks whose analytic solutions can be used to test AIRNET. The test cases
will be limited to constant ρ and μ, that is, no temperature or height
differences in the networks. Slightly different versions of the turbulent
relationships (equations B.1.3 and B.1.4) appear in Klote (1983, pp 33-36).

B.1.2 Thermal Properties of Air

    Two primary values are known for each node n: the temperature, T_n, and
the pressure, P_n, relative to the absolute ambient pressure, PB. From these
values the nodal density and viscosity are computed:

$$\rho_n = 0.0034838 \cdot (PB + P_n)/(T_n + 273.15) \qquad\qquad (B.1.5)$$
and
$$\mu_n = 0.0000171432 + 0.00000004828 \cdot T_n \qquad\qquad (B.1.6)$$
where
    T_n is in °C and PB and P_n are in Pa.

It would be relatively easy to substitute the proper psychrometric relation-
ships to compute these values more accurately and include humidity as a factor
in air density.

B.2 Powerlaw Element Tests

B.2.1 Powerlaw Element Test #1

This is a test of the powerlaw element calculations in both the turbulent and laminar regimes and of solution convergence at the laminar-turbulent transition.

The air in all nodes is at 20°C; the ambient pressure is 101325 Pa.

$$\rho_n = 0.0034838 \cdot (PB + P_n)/(T_n + 273.15) = 1.20415 \text{ kg/m}^3 \quad (P_n = 0) \qquad \text{(B.2.1)}$$
$$\mu_n = 0.0000171432 + 0.00000004828 \cdot T_n = 0.0000181088 \text{ kg/ms} \qquad \text{(B.2.2)}$$

The coefficients for the powerlaw elements used in this test are computed from an orifice model. The critical properties of an orifice are its area, A, hydraulic diameter, D, and discharge coefficient, c. The orifice flow equation is

$$Q = c \, A \, \sqrt{2\Delta P/\rho} \qquad \text{(B.2.3)}$$

which can be rearranged to the powerlaw element equation

$$w \; (= \rho Q) = C \, \sqrt{\rho \Delta P} \qquad \text{(B.2.4)}$$

by letting

$$\boxed{C = c \, A \, \sqrt{2}} \qquad \text{(B.2.5)}$$

Assuming that the flow is laminar below a certain Reynolds number (Re = $\rho VD/\mu$ = $wD/\mu A$) and that the laminar and turbulent flow equations give the same flow at that point, gives

$$w = \mu A Re/D = \rho K \Delta P/\mu \qquad \text{and} \qquad w = \mu A Re/D = cA\sqrt{2\rho \Delta P}$$

$$K = \mu^2 A Re/\rho D \Delta P \qquad \text{and} \qquad \Delta P = \mu^2 Re^2/2\rho c^2 D^2$$

$$\boxed{K = 2ADc^2/Re} \qquad \text{(B.2.6)}$$

The first powerlaw element assumes D = 0.1 m, A = 0.01 m^2, c = 0.6, and transition at a Reynolds number of 100. These values give C_1 = 0.00848528, and K_1 = 0.0000072 . Laminar flow occurs if w_1 (= $\mu A Re/D$) $\leq$ 0.000181088 kg/s. The second powerlaw element assumes D = 0.2 m, A = 0.04 m^2, and the same c and Re. This gives C_2 = 0.03394113, and K_2 = 0.0000576 . Laminar flow occurs if w_2 $\leq$ 0.000362176 kg/s.

In this test the two powerlaw airflow elements are arranged in series:

$$P_1 \qquad\qquad P_2 \qquad\qquad P_3$$

46
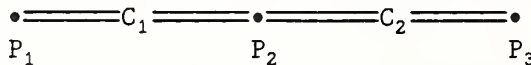
For turbulent flow in both elements, the equivalent powerlaw coefficient is

$$C_e = [\ 1/C_1{}^2 + 1/C_2{}^2\ ]^{-\frac{1}{2}} = 0.00823193 \qquad\qquad (B.2.7)$$

For laminar flow in both elements, the equivalent coefficient is

$$K_e = [\ 1/K_1 + 1/K_2\ ]^{-1} = 0.0000064 \qquad\qquad (B.2.8)$$

Substituting these two coefficients and thermal properties into the appropriate expressions give

$$w_T = 0.00903321\ \sqrt{P_1 - P_3} \qquad\text{and}\qquad w_L = 0.425570\ (P_1 - P_3) \qquad (B.2.9)$$

Flow is turbulent in both elements if $P_1 - P_3 \geq 0.00160752$ Pa. Flow is laminar in both elements if $P_1 - P_3 \leq 0.000042552$ Pa. AIRNET should give the following flow values as a function of P_1 (assuming $P_3 = 0$):

P_1 (Pa)	w (kg/s)
1.000	0.00903321
0.100	0.00285655
0.010	0.000903321
0.00161	0.000362456
0.0016076	0.000362541
0.0000426	0.000018108
0.000042	0.000017874
0.00004	0.000017023

Notes:
(1) AIRNET computed the expected flows in 4 or fewer iterations.
(2) If two equal elements were used in the airflow network, the AIRNET initialization algorithm would estimate $P_2 = (P_1+P_3)/2$, which is exactly correct. The iterative solution process would not be tested.
(3) When flow is laminar in both elements, the initialization gives the exact solution.

The followinng input data is include as file AFDATA.PL1 on the AIRNET distribution diskette.

powerlaw test #1 input file

```
node   node-1   c   0.0   20.0   1.0
node   node-2   v   0.0   20.0
node   node-3   c   0.0   20.0   0.0

element  orf-0.01  plr  7.2e-6   7.2e-6   0.00848528  0.5   orifice - 0.01 m^2
element  orf-0.04  plr  5.76e-5  5.76e-5  0.03394113  0.5   orifice - 0.04 m^2

link   link-1   node-1   0.0   node-2   0.0   orf-0.01   null
link   link-2   node-2   0.0   node-3   0.0   orf-0.04   null
```

****

B.2.2 Powerlaw Element Test #2

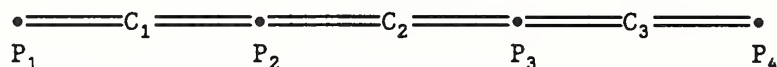This is a test of the AIRNET convergence acceleration algorithm. It involves a simple case which can be very difficult to solve. In this test three powerlaw airflow elements are arranged in series:

$$\bullet\!\!=\!\!=\!\!=\!\!C_1\!\!=\!\!=\!\!=\!\!\bullet\!\!=\!\!=\!\!=\!\!C_2\!\!=\!\!=\!\!=\!\!\bullet\!\!=\!\!=\!\!=\!\!C_3\!\!=\!\!=\!\!=\!\!\bullet$$
$$P_1 \qquad\qquad P_2 \qquad\qquad P_3 \qquad\qquad P_4$$

When C_2 is very large compared to C_1 and C_3, as in the case of a doorway connecting two rooms which have only small cracks connecting to ambient, convergence can be very slow with some algorithms.

This test uses a series of different size orifices, each with c = 0.6 and $D^2 = A$, varying in size from 0.0001 m^2 to 100.0 m^2. The computed flows can be checked by the equivalent powerlaw coefficient (turbulent flow) which is

$$C_e = [\ 1/C_1{}^2 + 1/C_2{}^2 + 1/C_3{}^2\ ]^{-\frac{1}{2}} \qquad\qquad (B.2.10)$$

AIRNET should give the following flow values ($P_1 - P_4$ = 100 Pa) for each combination of three elements listed. The numbers of iterations are the results of the test.

Elements	w (kg/s)	Iterations		Elements	w (kg/s)	Iterations	
1 - 1 - 1	0.00053758	2	(2)	2 - 1 - 2	0.00092195	5	(5)
1 - 2 - 1	0.00065676	5	(5)	2 - 2 - 2	0.00537585	2	(2)
1 - 3 - 1	0.00065839	8	(10)	2 - 3 - 2	0.00656764	5	(5)
1 - 4 - 1	0.00065840	8	(29)	2 - 4 - 2	0.00658388	8	(10)
1 - 5 - 1	0.00065840	7	(83)	2 - 5 - 2	0.00658404	9	(29)
1 - 6 - 1	0.00065840*	5	(>100)	2 - 6 - 2	0.00658404	11	(93)
1 - 7 - 1	0.00065840*	5	(>100)	2 - 7 - 2	0.00658404*	9	(>100)

Notes:
(1) AIRNET computed the expected flows.
(2) * indicates cases with laminar flow in the center element.
(3) The iterations in parnetheses are without convergence acceleration.

powerlaw test #2 input file

```
node      node-1    c    0.0    20.0     50.0
node      node-2    v    0.0    20.0
node      node-3    v    0.0    20.0
node      node-4    c    0.0    20.0    -50.0
element   orf-0.0001 plr 7.2e-9      7.2e-9      8.48528e-5    0.5    orf - 0.0001m^2
element   orf-0.001  plr 2.2769e-7  2.2769e-7  0.000848528   0.5    orf - 0.001 m^2
element   orf-0.01   plr 7.2e-6      7.2e-6      0.00848528    0.5    orf - 0.01 m^2
element   orf-0.1    plr 2.2769e-4  2.2769e-4  0.0848528     0.5    orf - 0.1 m^2
element   orf-1.0    plr 0.0072      0.0072      0.848528      0.5    orf - 1.0 m^2
element   orf-10.    plr 0.22769    0.22769    8.48528       0.5    orf - 10.0 m^2
element   orf-100.   plr 7.2         7.2         84.8528       0.5    orf - 100.0 m^2
link      link-1    node-1   0.0    node-2   0.0    orf-0.0001    null
link      link-2    node-2   0.0    node-3   0.0    orf-0.0001    null
link      link-3    node-3   0.0    node-4   0.0    orf-0.0001    null
```

****

B.2.3   Powerlaw Element Test #3

This is a test of the convergence of the AIRNET iterative solution method for a relatively complex network of airflow elements.   It involves 12 nodes and 20 powerlaw airflow elements arranged in series and parallel:
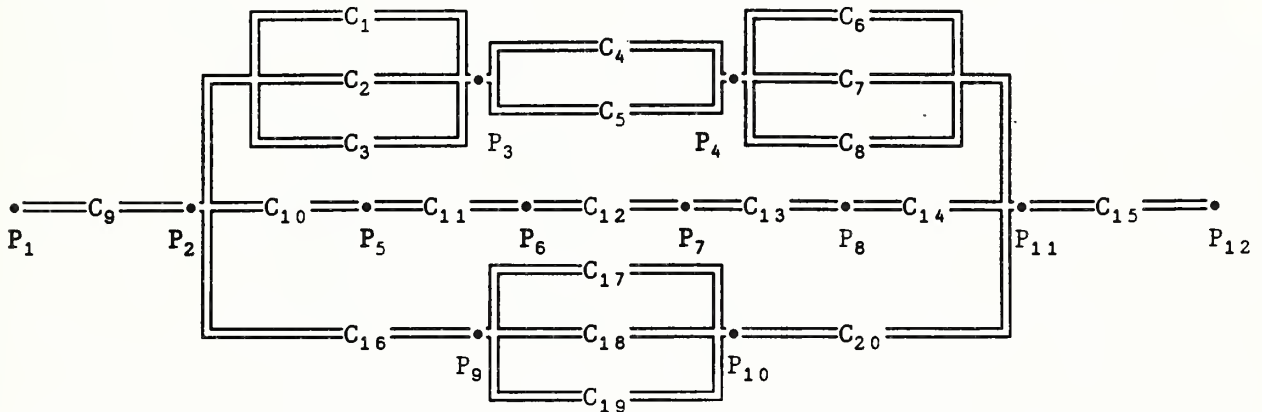
Using the conversions for parallel airflow elements gives:
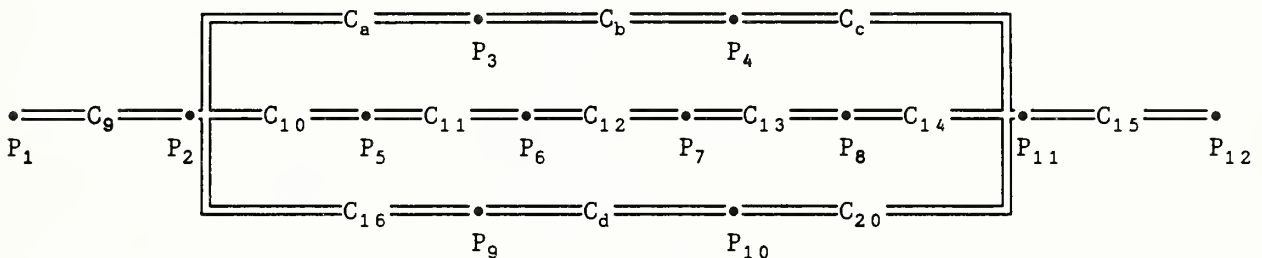
where
$$C_a = C_1 + C_2 + C_3, \qquad C_b = C_4 + C_5,$$
$$C_c = C_6 + C_7 + C_8, \text{ and } \quad C_d = C_{17} + C_{18} + C_{19}.$$

Using the conversions for series airflow elements gives:
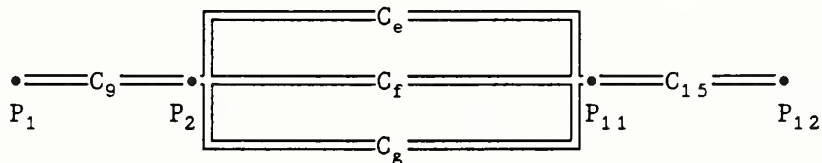
where
$$C_e = [\ 1/C_a{}^2 + 1/C_b{}^2 + 1/C_c{}^2\ ]^{-\frac{1}{2}},$$
$$C_f = [\ 1/C_{10}{}^2 + 1/C_{11}{}^2 + 1/C_{12}{}^2 + 1/C_{13}{}^2 + 1/C_{14}{}^2\ ]^{-\frac{1}{2}}, \text{ and}$$
$$C_g = [\ 1/C_{16}{}^2 + 1/C_d{}^2 + 1/C_{20}{}^2\ ]^{-\frac{1}{2}}.$$

Again using the conversion for parallel airflow elements gives:
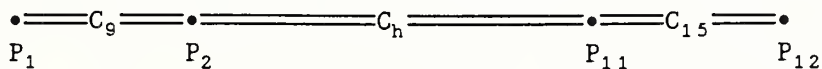
where
$$C_h = C_e + C_f + C_g.$$

Again using the conversion for series airflow elements gives:

$$P_1 \; \bullet\!=\!=\!=\!=\!=\!=\!=\!C_i\!=\!=\!=\!=\!=\!=\!=\!\bullet \; P_{12}$$

where
$$C_i = [\; 1/C_9{}^2 + 1/C_h{}^2 + 1/C_{15}{}^2 \;]^{-\frac{1}{2}},$$

In order to test AIRNET convergence, powerlaw coefficient values are created for orifices of greatly differning areas:

number	c	D(m)	A(m^2)	C	K
1	.6	0.1	0.01	0.00848528	7.2e-6
2	.6	1.0	1.0	0.848528	0.0072
3	.6	2.0	4.0	3.394113	0.0576
4	.6	0.05	0.0025	0.00212132	9.0e-7
5	.6	0.06	0.0036	0.00305470	1.555e-6
6	.6	1.0	1.0	0.848528	0.0072
7	.6	1.0	1.0	0.848528	0.0072
8	.6	1.0	1.0	0.848528	0.0072
9	.6	1.0	1.0	0.848528	0.0072
10	.6	1.0	1.0	0.848528	0.0072
11	.6	0.01	0.0001	0.000084853	7.2e-9
12	.6	1.0	1.0	0.848528	0.0072
13	.6	0.02	0.0004	0.000339411	5.76e-8
14	.6	2.0	4.0	3.394113	0.0576
15	.6	1.0	1.0	0.848528	0.0072
16	.6	0.02	0.0004	0.000339411	5.76e-8
17	.6	2.0	4.0	3.394113	0.0576
18	.6	0.01	0.0001	0.000084853	7.2e-9
19	.6	1.0	1.0	0.848528	0.0072
20	.6	0.03	0.0009	0.000763675	1.944e-7

The coefficients for the simplified networks are:
C_a = 4.251126
C_b = 0.00517602
C_c = 2.545584
C_d = 4.242726
C_e = 0.00517601
C_f = 0.00008232
C_g = 0.00031016
C_h = 0.00556848
C_i = 0.00556824
which gives w = 0.0611024 kg/s for P_1-P_{12} = 100 Pa and ρ = 1.20415 kg/m^3.

AIRNET computed the expected flow in 12 iterations. This indicates that the number of iterations increases with the complexity of the network. However, this large number of iterations is for a combination of large and small airflow resistances chosen to be difficult to solve. The solution required 157 iterations without convergence acceleration. AIRNET solved the same network containing 20 identical airflow coefficients (C = .00848528) in only 6 iterations. Tests on different computers and with different compilers indicate that the number of iterations for this network configuration is very sensitive to round-off errors.
See the input file AFDATA.PL3 on the diskette.

B.3   Stack Effect Tests

Once the performance of the powerlaw element model has been verified, it can be used to test the stack pressure and wind pressure calculations.

B.3.1   Stack Effect Test #1

The following figure shows two rooms represented by airflow nodes 1 and 2 connected by two simple openings at different heights.
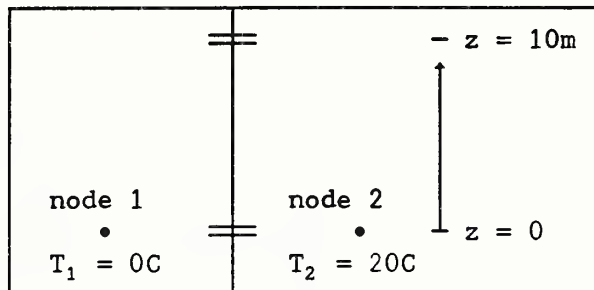
The standard equation for stack effect (Klote, 1983, p12) is

$$\Delta P = \frac{gP}{R} \left(\frac{1}{T_1} - \frac{1}{T_2} \right) h \qquad\qquad (1)$$

where

g = acceleration of gravity (9.80 m/s^2),
P = absolute pressure (101325. Pa),
R = gas constant ($1/R$ = 0.0034838),
T_i = absolute temperature of node i (°K), and
h = distance from the neutral plane (m).

The values in parentheses are the numbers used in AIRNET. They give gP/R = 3459.36 (Klote gives 3460 for this value). Since the openings are identical, the height of the neutral plane should be 5 m. Substituting into equation (1) gives ΔP = 4.3202 Pa.

AIRNET gives slightly different results for ΔP at the two openings because the relationship between flow rate and pressure drop depends on the direction of flow. The computed flow rate is 0.019692 kg/s at pressure drops of -4.472866 and 4.167554 Pa for the upper and lower openings, respectively. The average pressure drop (neglecting sign) through the two openings is 4.3202 Pa. An extra iteration produces a very small change in these results.

```
stack test #1 input file

node    node-1    c   0.0     0.0     0.0
node    node-2    v   0.0    20.0

element  orf-0.01  plr  7.2e-6     7.2e-6      0.00848528   0.5  orf - 0.01 m^2

link    link-1    node-1   0.0   node-2   0.0   orf-0.01   null
link    link-2    node-1  10.0   node-2  10.0   orf-0.01   null

********
```

B.3.2 Stack Effect Test #2

The following figure shows two rooms represented by airflow nodes 1 and 2 connected by three airflow elements which connect to node 2 at different heights.
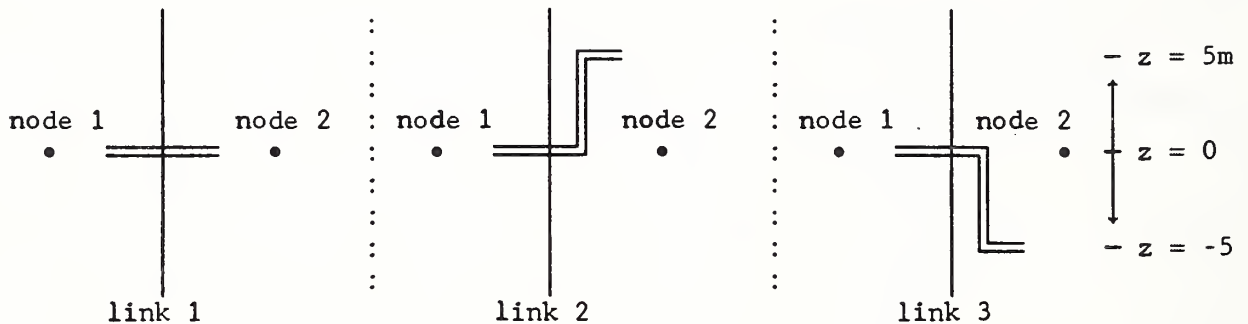
The pressure of node 1 is 5 Pa greater than node 2. The temperature of node 1 is 0 C; node 2 is 20 C. Because of this temperature difference, the flow through element 2 will ge less than the flow through element 1, and the flow through element 3 will be greater ($w_2 < w_1 < w_3$). The standard stack effect equation indicates a pressure difference of about ±4.3202 Pa due to the height changes. That is, $\Delta P_1 = 5.0$ Pa, $\Delta P_2 \simeq 0.68$ Pa, and $\Delta P_3 \simeq 9.32$ Pa.

The first AIRNET solution is computed with unknown airflow directions. This gives: $\Delta P_1 = 5.0$ Pa, $\Delta P_2 \simeq 2.84$ Pa, and $\Delta P_3 \simeq 7.16$ Pa. Recalculation with known flow directions gives the expected results. See the discussion about stack effect calculation in Appendix C.

Also note that AIRNET "solved" a case where all the node pressures are specified. Computer algorithms often run into numerical problems on such trivial cases -- they provide good tests for algorithmic errors.

stack test #2 input file

```
node    node-1    c    0.0     0.0     5.0
node    node-2    c    0.0    20.0     0.0

element  orf-0.01  plr  7.2e-6    7.2e-6    0.00848528   0.5  orf - 0.01 m^2

link    link-1   node-1   0.0   node-2    0.0   orf-0.01   null
link    link-2   node-1   0.0   node-2    5.0   orf-0.01   null
link    link-3   node-1   0.0   node-2   -5.0   orf-0.01   null
```

****

B.4 Wind Pressure Tests

AIRNET does not include a model for converting environmental variables such as wind speed and direction to wind pressure. It does include a flexible way for the user to transfer data from a wind pressure model (or measurements) to the appropriate airflow element. The following tests are designed to insure that the AIRNET portion of that data transfer occurs correctly.

The wind pressure tests require use of the wind coefficients file named WIND. The contents of this file are:

wind coefficients file

```
north   1.00 0.924 0.707 0.383 0.00 -.383 -.707 -.924 -1.0 -.924 -.707 ...
east    0.00 0.383 0.707 0.924 1.00 0.924 0.707 0.383 0.00 -.383 -.707 ...
south   -1.0 -.924 -.707 -.383 0.00 0.383 0.707 0.924 1.00 0.924 0.707 ...
west    0.00 -.383 -.707 -.924 -1.0 -.924 -.707 -.383 0.00 0.383 0.707 ...

plus-one    1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
minus-one  -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.
```

The four lines of data shown with "..." are actually complete on the file; they are truncated here because of margin requirements. The four profiles given direction names have coefficients equal to the cosine of angle between the wind direction and the direction indicated.

B.4.1 Wind Pressure Test #1

The wind pressure tests are based on a simple three-element airflow network:
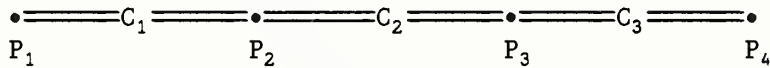
Powerlaw coefficients of 0.000848528, 0.00848528, and 0.000848528 are assumed for the three elements. These combine to give an effective powerlaw coefficient of 0.000598506 .

Assuming standard barometric pressure, an ambient temperature of 20°C, and a wind speed of 5.0 m/s gives a potential wind pressure of ($\frac{1}{2}\rho V^2$) 15.0519 Pa. Test #1 uses wind coefficient profiles that are independent of wind direction. Both links use a wind pressure modifier of 2. These combine to give a total wind pressure across the network of 60.2075 Pa. Therefore, the expected airflow through the network is 0.00509605 kg/s.

```
wind pressure test #1 input file

node    node-1    c    0.0    20.0    0.0
node    node-2    v    0.0    20.0
node    node-3    v    0.0    20.0
node    node-4    c    0.0    20.0    0.0

element  orf-0.001  plr 2.2769e-7 2.2769e-7  0.000848528  0.5  orf - 0.001 m^2
element  orf-0.010  plr 7.2e-6    7.2e-6     0.00848528   0.5  orf - 0.01 m^2

link    link-1    node-1    0.0    node-2    0.0    orf-0.001    plus-one   2.0
link    link-2    node-2    0.0    node-3    0.0    orf-0.010    null
link    link-3    node-4    0.0    node-3    0.0    orf-0.001    minus-one 2.0
```

****

Note how the order of the nodes in link-1 and link-3 is from the "outside" to
the "inside" which requires a reversal in the numbering.

B.4.2   Wind Pressure Test #2

This test is almost the same as the first test except the "north" and
"south" wind coefficient profiles are used, instead of "plus-one" and "minus-
one". Since these profiles vary as cosD, where D is the wind direction, the
total wind pressure across the network of 60.2075·cosD Pa, and the expected
airflow through the network is $\pm 0.00509605 \cdot \sqrt{|\cos D|}$ kg/s.

wind directions	expected flow rate	computed flow rate
0, 360	0.00509605 kg/s	0.00509604 kg/s
30, 330	0.00474241	0.00470292
60, 300	0.00360345	0.00357087
90, 270	0.0	0.0
120, 240	-0.00360345	-0.00357087
150, 210	-0.00474241	-0.00470292
180	-0.00509605	-0.00509604

The difference between the computed and the expected flow rates is apparently
caused by the linear interpolation of the wind coefficient profiles.

B.5 Duct Element Tests

This is a test of the AIRNET Darcy-Weisbach-Colebrook duct model. The expected flow rates are based on the ASHRAE friction chart (ASHRAE, 1985, pp 33.5 & 33.26). It involves a simple network in which three serial duct elements are in parallel with a duct element of equivalent length. The flows in both sides of the network should be equal.
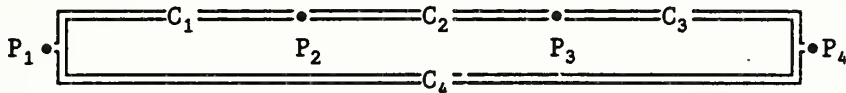
Two design points were selected from the chart. The first point is for a duct 250 mm in diameter (area = 0.04909 m^2) with a pressure drop of 0.9 Pa/m of duct, giving a flow velocity of 4.0 m/s and a flow rate of 195 l/s. The second point is for a duct 630 mm in diameter (area = 0.31172 m^2) with a pressure drop of 4.0 Pa/m of duct, giving a flow velocity of 16.0 m/s and a flow rate of 5000 l/s. The absolute roughness dimension of both ducts is 0.15 mm. Duct lengths are 2 m, 3 m, 5 m, and 10 m for elements 1 through 4, respectively.
In the first case the expected flow rate is 0.24 kg/s while in the second case it is about 6.0 kg/s. The values computed by AIRNET are 0.245 and 6.19 kg/s, respectively. These values are within 3% of those on the ASHRAE chart.

duct (Darcy-Weisbach-Colebrook model) test #1 input file

```
node    node-1    c    0.0    20.0      9.0
node    node-2    v    0.0    20.0
node    node-3    v    0.0    20.0
node    node-4    c    0.0    20.0      0.0

element   duct-2x25   dwc   2.0    0.25   0.04909   0.00015    duct 2m x 25cm
                            0.0   64.0   0.0       128.
element   duct-3x25   dwc   3.0    0.25   0.04909   0.00015    duct 3m x 25cm
                            0.0   64.0   0.0       128.
element   duct-5x25   dwc   5.0    0.25   0.04909   0.00015    duct 5m x 25cm
                            0.0   64.0   0.0       128.
element   duct-10x25 dwc  10.0    0.25   0.04909   0.00015    duct 10m x 25cm
                            0.0   64.0   0.0       128.

link    link-1    node-1    0.0    node-2    0.0    duct-2x25    null
link    link-2    node-2    0.0    node-3    0.0    duct-3x25    null
link    link-3    node-3    0.0    node-4    0.0    duct-5x25    null
link    link-4    node-1    0.0    node-4    0.0    duct-10x25   null
```

****

B.6 Doorway Element Test

This test compares two different ways to simulate the two-way airflows through doorways. The test represents a 0.8 m by 2.0 m doorway connecting two rooms with a 4°C temperature difference as shown below. The computed results can be compared to the correlation presented by Weber and Kearney (1980) for heat transfer through a doorway:

$$Nu = \alpha \cdot Pr \cdot \sqrt{Gr} \qquad (B.6.1)$$

where

    Nu = Nusselt number = $h \cdot H/k$,
    Pr = Prandtl number = $c_p \cdot \mu/k$,
    Gr = Grashof number = $\rho^2 \cdot g \cdot \beta \cdot H^3 \cdot \Delta T/\mu^2$, and
    α = experimentally determined coefficient.

In addition,

    ρ = density,
    μ = viscosity,
    c_p = specific heat,
    k = thermal conductivity,
    g = acceleration of gravity,
    β = coefficient of thermal expansion = $-\Delta\rho/(\rho \cdot \Delta T)$,
    H = doorway height,
    h = convection coefficient = $q/(W \cdot H \cdot \Delta T)$,
    q = heat flux rate = $w \cdot c_p \cdot \Delta T$,
    W = doorway width, and
    w = mass flow rate.

Substituting into equation (B.6.1) and simplifying gives:

$$w = \alpha \cdot W \cdot \sqrt{\rho \cdot g \cdot \Delta\rho \cdot H^3} \qquad (B.6.2)$$

Assuming the doorway behaves as an orifice leads to the relationship $\alpha = c/3$, where c is the orifice coefficient. Substituting the values $\alpha = 0.26$, $\rho = 1.20415$ kg/m^3, $\Delta\rho = 0.0164312$ kg/m^3, $g = 9.80$ m/s^2, $H = 2.0$ m, and $W = 0.8$ m gives $w = 0.25906$ kg/s.

The doorway element test compares the doorway element model with an approximation composed of multiple orifice elements.

Doorway simulated
by doorway element

Doorway simulated
by multiple
powerlaw elements
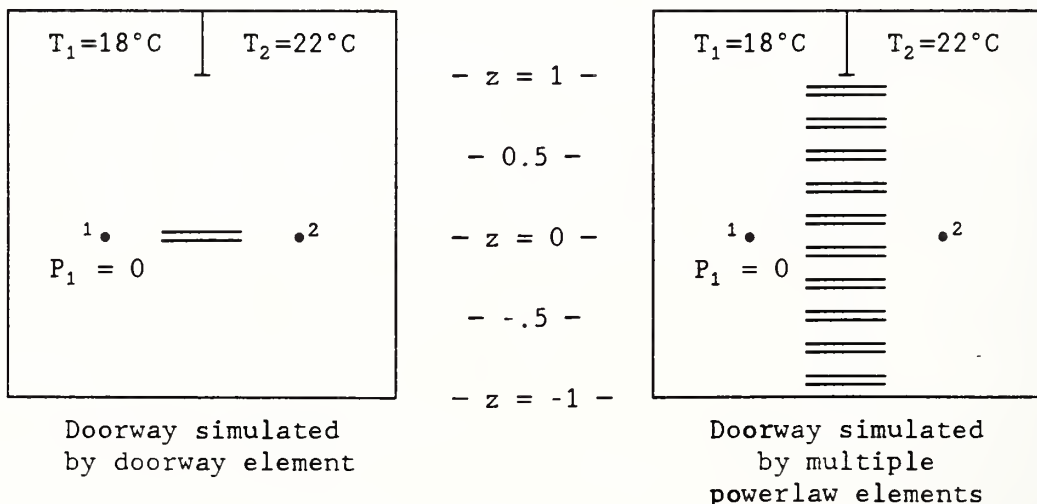
In the first case a doorway element with a discharge coefficient of 0.78 is used giving computed two-way airflows of 0.25913 and 0.25899 kg/s. These are essentially exact. In the second case ten 0.16 m² powerlaw openings at equally spaced heights are used to represent a doorway. The computed two-way airflow is 0.261 kg/s which is within 1% of the exact value. Both cases were solved in two iterations.


```
doorway test #1 input file

node    node-1    c    0.0    18.0      0.0
node    node-2    v    0.0    22.0

element  orf-0.16  plr  0.0015575 0.0015575  0.176494   .500      opening-.16m^2
element  dor-1.60  dor  0.015575  0.015575   1.76494    .500      doorway-1.6m^2
                        0.0001     2.0          0.8          0.78

link    link-1    node-1    0.9    node-2    0.9    orf-0.16    null
link    link-2    node-1    0.7    node-2    0.7    orf-0.16    null
link    link-3    node-1    0.5    node-2    0.5    orf-0.16    null
link    link-4    node-1    0.3    node-2    0.3    orf-0.16    null
link    link-5    node-1    0.1    node-2    0.1    orf-0.16    null
link    link-6    node-1   -0.1    node-2   -0.1    orf-0.16    null
link    link-7    node-1   -0.3    node-2   -0.3    orf-0.16    null
link    link-8    node-1   -0.5    node-2   -0.5    orf-0.16    null
link    link-9    node-1   -0.7    node-2   -0.7    orf-0.16    null
link    link-10   node-1   -0.9    node-2   -0.9    orf-0.16    null
link    link-11   node-1   -1.0    node-2   -1.0    dor-1.60    null
```

****

Note the heights used with link-11 for the doorway element. In this case the reference heights of the two nodes is one meter above the floor, so the doorway element must be located at -1.0 relative to each node.

B.7 Constant Flow Element Tests

B.7.1 Constant Flow Test #1

In this test a powerlaw airflow elements and a constant flow element are arranged in series:
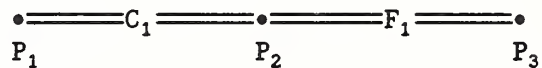
The flow rate is set to 1.0 kg/s. Since the flow through both elements must be the same and the powerlaw coefficient is 0.0848528, the expected pressure drop across the powerlaw element is 115.342 Pa.

constant flow test #1 input file

```
node    node-1    c    0.0    20.0    0.0
node    node-2    v    0.0    20.0
node    node-3    c    0.0    20.0    0.0

element   orf-0.1000 plr 2.2769e-4 2.2769e-4   0.0848528      0.5  orf - 0.1 m^2
element   flow-1.000 cfr 1.0                              constant flow of 1.000 kg/s

link    link-1    node-1    0.0    node-2    0.0    orf-0.1000    null
link    link-2    node-2    0.0    node-3    0.0    flow-1.000    null
```

****

B.7.2 Constant Flow Test #2

This test uses the same airflow network as powerlaw test #3 except that a constant flow element is added at node 13. Node 12 becomes an unknown pressure and new node 13 must be a known pressure. If the constant flow is set to 0.0611025 kg/s and the pressure of node 1 set to 50 Pa, this test should give the same pressures and flows as powerlaw test #3.

It does give the same answers (to within the convergence limits) and does this in only 6 iterations. See the input file AFDATA.CF2 on the diskette.

B.8 Fan Element Tests

The fan element tests are based on problems in the textbook by Osborne (1977, pp 71-75). All tests use a fan whose performance curve was developed from the following characteristics:

volume flow	Total pressure
0.0 m^3/s	750 Pa
1.0	755
2.0	730
3.0	590
4.0	275

The performance curve (assuming a density of 1.204 kg/m^3) is:

$$w = 764.429 - 18.2922 \cdot P + 19.4633 \cdot P^2 - 7.63940 \cdot P^3$$

This curve does not include the contraflexure evident in the table of characteristics. This was done by increasing the pressure rise at zero flow until the contraflexure was eliminated (765 Pa).

B.8.1 Fan Element Test #1

This corresponds to problem 1 in Osborne. It consists of a fan in series with a powerlaw element. The expected answer is 3.167 kg/s (2.63 m^3/s) at a total fan pressure of 660 Pa. The computed results are 3.158 kg/s at 660.3 Pa.

B.8.2 Fan Element Test #2

This corresponds to problem 5 in Osborne. It consists of two fans in parallel and three powerlaw elements forming a complete circuit.
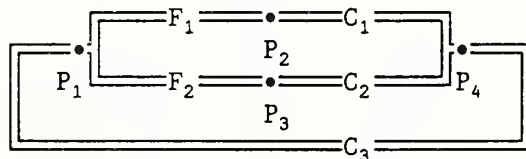
values	expected	computed
$P_4 - P_1$	293. Pa	291.1 Pa
w_1	2.91 kg/s (2.42 m^3/s)	2.909 kg/s
w_2	3.61 kg/s (3.00 m^3/s)	3.616 kg/s
w_3	6.53 kg/s (5.42 m^3/s)	6.524 kg/s

The differences can be attributed to the inaccuracies in the fit of the fan performance coefficients and the inaccuracies in the graphic procedures used to compute the textbook answers.

59

B.8.1 Fan Element Test #3

This corresponds to problem 6 in Osborne. It consists of two fans in series and three powerlaw elements forming an open network.
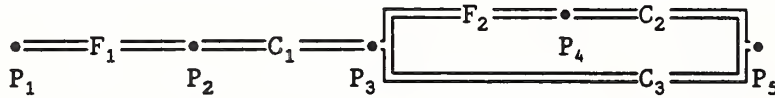
values	expected		computed
P_3 - P_5	70. Pa		72.5 Pa
w_1	4.19 kg/s	(3.48 m^3/s)	4.171 kg/s
w_2	3.73 kg/s	(3.10 m^3/s)	3.713 kg/s
w_3	0.46 kg/s	(0.38 m^3/s)	0.459 kg/s

fan test #3 input file: comparison to Osborne problem #6

```
node    node-1   c    0.0    20.0    0.0
node    node-2   v    0.0    20.0
node    node-3   v    0.0    20.0
node    node-4   v    0.0    20.0
node    node-5   c    0.0    20.0    0.0

element  fan-A  fan  3.0e-5    7.2e-6  0.084853  .500      Osborne fan A
         1.204  750.0  5.00   0.10     3    -100.0
         764.429  -18.2922  19.4633  -7.63940  1.0         no contraflexure
         764.429  -18.2922  19.4633  -7.63940  5.0
         764.429  -18.2922  19.4633  -7.63940  100.
element  rest1  plr  1.0e-5  1.585e-05  0.19005   0.5      resistance 1
element  rest2  plr  1.0e-5  1.110e-05  0.13306   0.5      resistance 2
element  rest3  plr  1.0e-5  4.092e-06  0.04907   0.5      resistance 3

link    link-1   node-1   0.0   node-2   0.0   fan-A   null
link    link-2   node-2   0.0   node-3   0.0   rest1   null
link    link-3   node-3   0.0   node-4   0.0   fan-A   null
link    link-4   node-3   0.0   node-5   0.0   rest3   null
link    link-5   node-4   0.0   node-5   0.0   rest2   null
```

****

B.9 Quadratic Flow Element Test

The test of the quadratic flow element is an extension of the duct element test. Is also shows the possibilities of modeling ducts components by powerlaw or quadratic flow elements. The airflow network is:

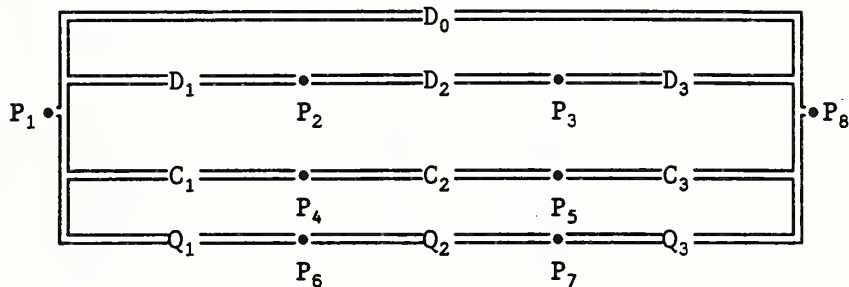Element D_0 is a 10 meter long duct element; elements D_1, D_2, and D_3 are 2, 3, and 5 meter long duct elements. Elements C_1, C_2, and C_3 are powerlaw elements whose coefficients were generated by the ELEMENT program to correspond to the duct elements. The elements were fitted at Reynolds numbers of 10000 and 40000. Elements Q_1, Q_2, and Q_3 are quadratic flow elements with coefficients generated to correspond to the duct elements. There should be equal flows through each branch of the network. The results of the test are summarized below:

ΔP	F_0	F_d	F_p	diff	F_q	diff	Re
0.01	0.0049276	0.0049276	0.0045302	-8.06%	0.0040626	-17.55%	1371.
0.02	0.0073755	0.0073755	0.0080704	+9.39%	0.0066006	-10.53%	2053.
0.04	0.0109604	0.0109604	0.0116082	+5.91%	0.0103411	-5.65%	3050.
0.08	0.0161748	0.0161748	0.0166968	+3.23%	0.0157490	-2.63%	4502.
0.16	0.0237102	0.0237108	0.0240167	+1.29%	0.0234849	-0.95%	6597.
0.32	0.0345369	0.0345369	0.0345440	+0.02%	0.0344891	-0.14%	9612.
0.64	0.0500175	0.0500174	0.0496870	-0.66%	0.0500971	+0.16%	13920.
1.28	0.0720693	0.0720691	0.0714681	-0.83%	0.0722028	+0.18%	20057.
2.56	0.1033972	0.1033964	0.1027976	-0.58%	0.1034881	+0.09%	28776.
5.12	0.1478221	0.1478200	0.1478615	+0.03%	0.1477486	-0.05%	41139.
10.24	0.2107488	0.2107428	0.2126813	+0.92%	0.2103540	-0.18%	58652.
20.48	0.2998266	0.2998018	0.3059127	+2.03%	0.2988997	-0.31%	83443.
40.96	0.4258877	0.4258392	0.4400455	+3.32%	0.4241279	-0.41%	118526.

where
    $\Delta P = P_1 - P_8$,
    F_0 = mass flow (kg/s) through element D_0 ,
    F_d = mass flow through the 3 duct elements branch,
    F_p = mass flow through the powerlaw elements branch,
    F_q = mass flow through the quadratic flow elements branch,
    diff = percent difference compared to F_0 , and
    Re = Reynolds number computed from F_0 .

The agreement for both fitted elements is quite good in the turbulent region, Re > 4000, with the quadratic flow element giving the best agreement. Good agreement is not expected in the transition and laminar region, Re < 4000.

```
quadratic flow relationship test #1 input file

node    source    c    0.0    20.0    0.32
node    duct-1    v    0.0    20.0
node    duct-2    v    0.0    20.0
node    dplr-1    v    0.0    20.0
node    dplr-2    v    0.0    20.0
node    dqfr-1    v    0.0    20.0
node    dqfr-2    v    0.0    20.0
node    sink      c    0.0    20.0    0.0

element duct-2x25 dwc 2 0.25 0.0490874 0.0015 duct - 2m long by 25cm dia
                     0 64   0 64
element duct-3x25 dwc 3 0.25 0.0490874 0.0015 duct - 3m long by 25cd dia
                     0 64   0 64
element duct-5x25 dwc 5 0.25 0.0490874 0.0015 duct - 5m long by 25cm dia
                     0 64   0 64
element duct-10x25 dwc 10 0.25 0.0490874 0.0015 duct - 10m long by 25cm dia
                     0 64   0 64
element dplr-2x25 plr 3.40642e-05 3.40642e-05 0.133079 0.524431 plr-duct ...
element dplr-3x25 plr 2.27095e-05 2.27095e-05 0.107587 0.524431 plr-duct ...
element dplr-5x25 plr 1.36257e-05 1.36257e-05 0.0823032 0.524431 plr-duct ...
element dqfr-2x25 qfr 0.310242 44.8089 qfr-duct - 2m long by 25cm dia
element dqfr-3x25 qfr 0.465364 67.2134 qfr-duct - 3m long by 25cm dia
element dqfr-5x25 qfr 0.775606 112.022 qfr-duct - 5m long by 25cm dia

link    duct-0    source    0.0    sink      0.0    duct-10x25    null
link    duct-1    source    0.0    duct-1    0.0    duct-2x25     null
link    duct-2    duct-1    0.0    duct-2    0.0    duct-3x25     null
link    duct-3    duct-2    0.0    sink      0.0    duct-5x25     null
link    dplr-1    source    0.0    dplr-1    0.0    dplr-2x25     null
link    dplr-2    dplr-1    0.0    dplr-2    0.0    dplr-3x25     null
link    dplr-3    dplr-2    0.0    sink      0.0    dplr-5x25     null
link    dqfr-1    source    0.0    dqfr-1    0.0    dqfr-2x25     null
link    dqfr-2    dqfr-1    0.0    dqfr-2    0.0    dqfr-3x25     null
link    dqfr-3    dqfr-2    0.0    sink      0.0    dqfr-5x25     null

********
```

Note:
    The a and b values for the quadratic flow elements are proportional to the
length of each element. This could be convenient for modeling duct networks.

B.10 Execution Time Tests

These tests are designed to provide simple benchmarks for relatively large airflow networks. Analytic solutions have not been developed for these cases. The runs were made on a PC compatible computer (4.77 MHz 8088 CPU with 8087 math coprocessor). This is a minimal microcomputer by today's standards.

The tests are based on the 9-node floor module mentioned in the main report and shown in figure 10. Test faces were developed for buildings of up to 32 floors. Versions of AIRNET using both the small and large memory modules are compared in the following table.

			calculation time		
Small memory model					
floors	nodes	iterations	input	airflows	unallocated memory
4	37	5	2.8 s	1.6 s	≈ 44000 bytes
8	73	5	5.3	3.3	31000
16	145	5	9.8	6.7	7000
Large memory model					
16	145	5	12.4	7.6	333000
32	289	5	24.7	15.3	265000

Four points to note are that (1) calculation time increases almost linearly with the number of nodes because of the sparse equation solver, (2) the input files process slowly because of their length, (3) the large memory model requires slightly more time than the small model, and (4) this C version of AIRNET is about 40% faster than the FORTRAN version used in the comparisons in the main report.

These network data files can be easily modified to show the effect of improper node order on sparsity and performance by altering the sequence of nodes on each floor. In the 37 node case the number of nonzero elements in [A] increases from 127 to 253 (an increase of 1008 bytes) and execution time increases from 1.6 to 2.6 seconds. The change in the fill pattern of the Jacobian is shown in the two figures below.


```
1234567890123456789012345678901234567    1234567890123456789012345678901234567
```

63

B.11 <u>References</u>

ASHRAE, 1985. <u>ASHRAE Handbook - 1985 Fundamentals</u>, ASHRAE, Atlanta GA.

Judkoff, R., Wortman, D., O'Doherty, R. and Burch, J., 1983. "A Methodology for Validating Building Energy Analysis Simulations", Solar Energy Research Institute report SERI/TR-254-1508, Golden, CO.

Klote, J.H. & Fothergill, J.W.Jr., 1983. "Design of Smoke Control Systems for Buildings", ASHRAE, Atlanta GA.

Osborne, W.C., 1977. <u>Fans</u>, 2nd edition, Pergamon Press.

Weber, D.D., and Kearney, R.J., 1980. "Natural Convective Heat Transfer Through an Aperture in Passive Solar Heated Buildings", <u>5th National Passive Solar Conference</u>, American Solar Energy Society, pp. 1037-1041.

APPENDIX C: Calculational Details

C.1 Newton's Method

This section reviews the theory of Newton's method for simultaneous nonlinear equations (Stoecker, 1971) as applied to the airflow equations.

Assume a case consisting of three nodes which are to be solved for three unknown pressures P_1, P_2, and P_3. Conservation of mass at each node is used to set up three nonlinear equations in terms of the unknown pressures:

$$f_1(P_1, P_2, P_3) = 0 \qquad\qquad (C.1.1a)$$

$$f_2(P_1, P_2, P_3) = 0 \qquad\qquad (C.1.1b)$$

$$f_3(P_1, P_2, P_3) = 0 \qquad\qquad (C.1.1c)$$

If the correct solution is P_{1c}, P_{2c}, and P_{3c}, a Taylor series expansion retaining only the low-order terms gives:

$$f_1(P_1, P_2, P_3) \simeq \qquad\qquad (C.1.2)$$

$$f_1(P_{1c}, P_{2c}, P_{3c}) + \left[\frac{\partial f_1(P_{1c}, P_{2c}, P_{3c})}{\partial P_1} \right] (P_1 - P_{1c})$$

$$+ \left[\frac{\partial f_1(P_{1c}, P_{2c}, P_{3c})}{\partial P_2} \right] (P_2 - P_{2c}) + \left[\frac{\partial f_1(P_{1c}, P_{2c}, P_{3c})}{\partial P_3} \right] (P_3 - P_{3c})$$

and similarly for f_2 and f_3, where the variables P_1, P_2, and P_3 are only an approximation to the correct solution.

$$b_1 \simeq a_{11}(P_1 - P_{1c}) + a_{12}(P_2 - P_{2c}) + a_{13}(P_3 - P_{3c}) \qquad (C.1.3a)$$

$$b_2 \simeq a_{21}(P_1 - P_{1c}) + a_{22}(P_2 - P_{2c}) + a_{23}(P_3 - P_{3c}) \qquad (C.1.3b)$$

$$b_3 \simeq a_{31}(P_1 - P_{1c}) + a_{32}(P_2 - P_{2c}) + a_{33}(P_3 - P_{3c}) \qquad (C.1.3c)$$

where

$$b_i = f_i(P_1, P_2, P_3) \qquad\qquad (C.1.3d)$$

and

$$a_{ij} = \frac{\partial f_i(P_{1c}, P_{2c}, P_{3c})}{\partial P_j} \qquad\qquad (C.1.3e)$$

However, since the exact pressure values are not yet known, compute the a_{ij} terms at the approximate pressure values:

$$a_{ij} \simeq \frac{\partial f_i(P_1, P_2, P_3)}{\partial P_j} \qquad\qquad (C.1.3f)$$

Equations (C.1.3) are a set of simultaneous linear algebraic equations in terms of the unknowns: $(P_1 - P_{1c})$, $(P_2 - P_{2c})$, and $(P_3 - P_{3c})$. Solution of

the equations for these values is used to improve the approximate pressure values:

$$P_1{}^* = P_1 - (P_1 - P_{1c})$$ (C.1.4a)

$$P_2{}^* = P_2 - (P_2 - P_{2c})$$ (C.1.4b)

$$P_3{}^* = P_3 - (P_3 - P_{3c})$$ (C.1.4c)

The new pressure values replace the old pressure values in equations (C.1.3) and solution continues iteratively until a sufficiently accurate set of pressure values has been computed.

These relationships are expressed in vector/matrix form in the main paper: Each airflow element, i, relates the mass flow rate, w_i, through the element to the pressure drop, ΔP_i, across it. A new estimate of the vector of all node pressures, $\{P\}^*$, is computed from the current estimate of pressures, $\{P\}$, by

$$\{P\}^* = \{P\} - \{C\}$$ (C.1.4')

where the correction vector, $\{C\}$, is computed by the matrix relationship

$$[J]\ \{C\} = \{B\}$$ (C.1.3')

$\{B\}$ is a column vector with each element given by

$$B_n = \sum_i w_i$$ (C.1.3d')

where n is the node number and i indicates all flow paths connecting node n to other nodes, and $[J]$ is the square (i.e. N by N for a network of N nodes) Jacobian matrix whose elements are given by

$$J_{n,m} = \sum_i \frac{\partial w_i}{\partial P_m}$$ (C.1.3f')

Since the airflows for the various airflow elements are computed from relationships of the form $w = f(\Delta P)$, where
$\Delta P = P_n - P_m + PS + PW$,
P_n, P_m = total pressures at nodes n and m,
PS = pressure difference due to density and height differences, and
PW = pressure difference due to wind,
the partial derivative needed for $[J]$ in equation (C.1.3f') are related by

$$\partial w/\partial P_m = -\partial w/\partial P_n .$$ (C.1.5)

These last two equations indicate that the Jacobian has a strong diagonal and is symmetric, two properties which can be used to speed the factoring of the Jacobian.

C.2 Solving Simultaneous Linear Equations

The following discussion is designed to provide a brief introduction to the method used to solve simultaneous equations in AIRNET. See Pissanetzky (1984) and Dhatt & Touzot (1984) for details.

The general form of the problem is

$$[A] \{X\} = \{B\} \tag{C.2.1}$$

where
   $[A]$ is a square matrix of coefficients,
   $\{X\}$ is a vector of solution values,
   $\{B\}$ is a matrix of coefficients.
It is often computationally preferable to factor the $[A]$matrix into triangular matrices which are simple to solve numerically.

Original problem: $\quad [A]\{X\} = \{B\}$

Factored problem: $\quad [L][U]\{X\} = \{B\}$ $\tag{C.2.2}$

Forward substitution: $[L]\{B'\} = \{B\}$ $\tag{C.2.3}$

Back substitution: $\quad [U]\{X\} = \{B'\}$ $\tag{C.2.4}$

The processing of the matrices is dependent on the way the matrices are represented. In this case we are interested in the skyline storage method.

$$[L] \ [U] = [A]$$

$$
\begin{bmatrix}
1 & 0 & 0 & \cdots \\
L_{21} & 1 & 0 & \cdots \\
L_{31} & L_{32} & 1 & \cdots \\
\cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\begin{bmatrix}
D_1 & U_{21} & U_{31} & \cdots \\
0 & D_2 & U_{32} & \cdots \\
0 & 0 & D_3 & \cdots \\
\cdots & \cdots & \cdots & \cdots
\end{bmatrix}
=
\begin{bmatrix}
E_1 & C_{21} & C_{31} & \cdots \\
R_{21} & E_2 & C_{32} & \cdots \\
R_{31} & R_{32} & E_3 & \cdots \\
\cdots & \cdots & \cdots & \cdots
\end{bmatrix}
$$

$n = 1$: $\quad 1 \cdot D_1 = E_1 \qquad$ implies: $\quad D_1 = E_1$

$n = 2$: $\quad L_{21}E_1 = R_{21} \qquad\qquad\qquad\quad L_{21} = R_{21}/D_1$
$\qquad\quad 1 \cdot U_{21} = C_{21} \qquad\qquad\qquad\quad U_{21} = C_{21}$
$\qquad\quad L_{21}U_{21} + D_2 = E_2 \qquad\qquad\quad D_2 = E_2 - L_{21}U_{21}$

$n = 3$: $\quad L_{31}E_1 = R_{31} \qquad\qquad\qquad\quad L_{31} = R_{31}/D_1$
$\qquad\quad L_{31}U_{21} + L_{32}E_2 = R_{32} \qquad\quad L_{32} = (R_{32} - L_{31}U_{21})/D_2$
$\qquad\quad 1 \cdot U_{31} = C_{31} \qquad\qquad\qquad\quad U_{31} = C_{31}$
$\qquad\quad L_{21}U_{31} + U_{32} = C_{32} \qquad\qquad U_{32} = C_{32} - L_{21}U_{31}$
$\qquad\quad L_{31}U_{31} + L_{32}U_{32} + D_3 = E_3 \quad D_3 = E_3 - L_{31}U_{31} - L_{32}U_{32}$

$n = 4$: $\quad$ etc.

Factoring algorithm:

for all values of n from 1 to N:

$$L_{ni} = (R_{ni} - \sum_{m=1}^{i-1} L_{nm}U_{im}) \ / \ D_i \qquad i = 1, 2, \ldots, n-1$$

$$U_{ni} = C_{ni} - \sum_{m=1}^{i-1} L_{im}U_{nm} \qquad i = 1, 2, \ldots, n-1$$

$$D_n = E_n - \sum_{m=1}^{n-1} L_{nm}U_{nm}$$

Forward substitution algorithm:

for all values of n from 1 to N:

$$B_n = B_n - \sum_{j=1}^{n-1} L_{nj}B_j$$

Back substitution algorithm:

for all values of n from N to 1:

$$B_n = B_n/D_n$$

$$B_i = B_i - U_{ni}B_n \qquad i = 1, 2, \ldots, n-1$$

$(X_n = B_n$ when finished)

The development of the skyline method is based on the simple observation that array locations above the topmost nonzero element in any column above the diagonal and to the left of the first nonzero element in any row below the diagonal must remain zero during factoring. It is unnecessary to allocate array space for those locations that are always zero. The [A] matrix is reorganized into the main diagonal vector of N elements, the lower triangular matrix of N variable length rows, and the upper triangular matrix of N variable length columns. The factoring and solution algorithms are then reorganized to accommodate this new data structure with its new numbering of array locations. All operations can be performed in place, i.e., without creating separate arrays for setting up the problem and for the factoring.

C.3 Arrays in C

C.3.1 2-dimensional arrays

2-D arrays can be handled directly or by arrays of pointers. Reviewing the C declarations (Kernighan & Ritchie, pp 103-110) for a 2-D array:

double aa[3][4];   is a declaration for an array of 3 rows and 4 columns.

```
  aa and aa[0] -->   aa[0][0]      where aa[i][j] is both a location in
                     aa[0][1]      memory and the C access syntax for it.
                     aa[0][2]      "-->" means "points to".
                     aa[0][3]
          aa[1] -->  aa[1][0]
                     aa[1][1]
                     aa[1][2]
                     aa[1][3]
          aa[2] -->  aa[2][0]
                     aa[2][1]
                     aa[2][2]
                     aa[2][3]
```

Note that the array indices start at zero for both columns and rows.

When aa is passed to a function the number of columns must be given in the declaration in the function:
double aa[][4];
It is not necessary to state the number of rows (the first dimension in a multidimensional array).

Now consider the use of an array of pointers:

double *pa[4];   is a declaration for an array of 4 pointers.

```
          pa[0] -->  aa(0,0)      where aa(i,j) is a location in memory
                     aa(0,1)      for the appropriate value.
                     aa(0,2)
                     aa(0,3)
          pa[1] -->  aa(1,0)
                     aa(1,1)
                     aa(1,2)
                     aa(1,3)
          pa[2] -->  aa(2,0)
                     aa(2,1)
                     aa(2,2)
                     aa(2,3)
          pa[3] -->    ---
```

Element aa(0,0) is accessed by *(pa+0+0) (= *pa) or pa[0][0]. Element aa(1,3) is accessed by *(pa[1]+3) or pa[1][3]. The pointer past the end of the array is used to determine the end of the last row during dumps. The pointer vector must be dimensioned one more than the number of rows in the array.

Since there is a problem in writing functions which are passed 2-D arrays and since it is desirable to always use the same method of representing such arrays, we will always represent a 2-D array by means of an array of pointers.

For arrays with very few columns, the addition of an array of pointers represents a significant increase in storage. In many such cases it may be more desirable to use a 1-D array of structures, where the structure element names can help document the program.

We often want to use arrays with elements 1 through N instead of 0 through N-1. In this case, point to the location just before each row:

```
double  *pa[5];        is a declaration for an array of 5 pointers.

        pa[1] -->      ---
        pa[0] -->      aa(1,1)
                       aa(1,2)
                       aa(1,3)
        pa[2] -->      aa(1,4)
                       aa(2,1)
                       aa(2,2)
                       aa(2,3)
        pa[3] -->      aa(2,4)
                       aa(3,1)
                       aa(3,2)
                       aa(3,3)
        pa[4] -->      aa(3,4)
```

The zero pointer is not used, and the final pointer is needed to determine the end of the last row during dumps. This means that the pointer array will have to be dimensioned two more than the number of rows (if we want row N accessed by pa[N] instead of pa[N-1]). Element aa(1,3) is accessed by *(pa[1]+3) or pa[1][3].

The user must know whether the array begins at zero or at one.

C.3.2 Variable Length Rows

We may also want to use arrays with variable length rows. In this case, the use of an array of pointers is essential:

    double *pa[5];   is a declaration for an array of 5 pointers.

            pa[0] -->   aa(0,0)
                        aa(0,1)
                        aa(0,2)
            pa[1] -->   aa(1,0)
                        aa(1,1)
            pa[2] -->   aa(2,0)
                        aa(2,1)
            pa[3] -->   aa(3,0)
                        aa(3,1)
                        aa(3,2)
                        aa(3,3)
            pa[4] -->   ---

Element aa(0,0) is accessed by *(pa+0+0) (= *pa) or pa[0][0]. Element aa(2,1) is accessed by *(pa[2]+1) or pa[2][1]. The pointer to the end of the array is used to determine the end of the last row during dumps. The pointer vector must be dimensioned one more than the number of rows in the array.

In the case of arrays with elements 1 through N instead of 0 through N-1, point to the location just before each row:

    double *pa[6];   is a declaration for an array of 6 pointers.

            pa[1] -->   ---
            pa[0] -->   aa(1,1)
                        aa(1,2)
            pa[2] -->   aa(1,3)
                        aa(2,1)
            pa[3] -->   aa(2,2)
                        aa(3,1)
            pa[4] -->   aa(3,2)
                        aa(4,1)
                        aa(4,2)
                        aa(4,3)
            pa[5] -->   aa(4,4)

The zero pointer is not used, and the final pointer is needed to determine the end of the last row during dumps. This means that the pointer array will have to be dimensioned two more than the number of rows. Element aa(1,3) is accessed by *(pa[1]+3) or pa[1][3].

The user must know whether the array begins at zero or at one.

C.4 Stack Effect

The following figure shows two rooms represented by airflow nodes n and m, respectively. It is assumed that each node can be characterized by a single temperature and a single static pressure at some height relative to a common data plane. The two rooms are shown with an airflow element connecting them. The inlet and outlet to the element are at different heights from each other and from the nodes representing the rooms to show the entire calculation for height differences.
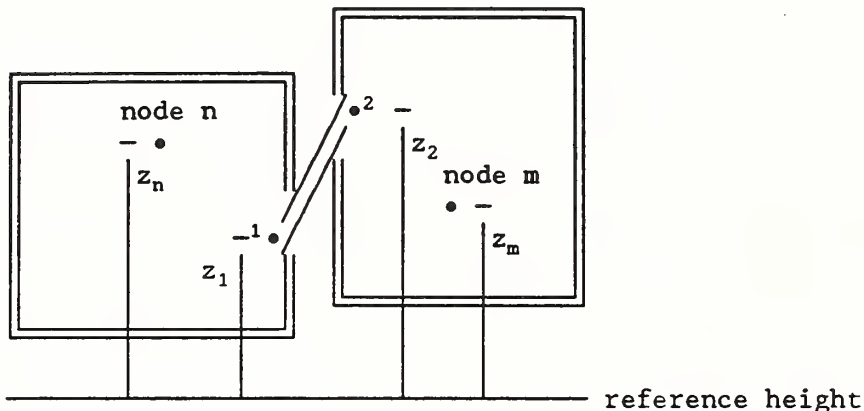
Analysis of airflow through the element i is based on Bernoulli's equation and its assumptions.

$$\Delta P_i = (p_1 + \rho V_1{}^2/2) - (p_2 + \rho V_2{}^2/2) + \rho g(z_1 - z_2) \tag{C.4.1}$$

where

ΔP_i = sum of all friction and dynamics losses (Pa),

p_1, p_2 = entry and exit static pressures (Pa),

V_1, V_2 = entry and exit velocities (m/s),

ρ = density of fluid flowing through the element (kg/m^3),

g = acceleration of gravity (9.81 m/s^2), and

z_1, z_2 = entry and exit elevations (m).

ρ may be either ρ_n or ρ_m depending on the direction of flow. One possibility is to let ρ equal the average of the two densities and accept the inaccuracy in the computed airflow. The other possibility is to select the pressure based on the most recently computed flow direction and let the iterative solution of the airflow equation operate.

Equation (C.4.1) defines a sign convention for the direction of airflow: positive flow is from point 1 to point 2 -- node n to node m. Equation (C.4.1) can be simplified for use in the airflow computer algorithm by defining several related terms. Dynamic pressures are the $\rho V^2/2$ terms in equation (C.4.1) and total pressure is defined to be $P \equiv p + \rho V^2/2$. If nodes n and m represent rooms, the dynamic pressures are effectively zero. If the nodes are part of a duct network, there will be a positive dynamic pressure.

The pressures at the inlet and outlet of the airflow element can be related to the node pressures by the hydrostatic law:

$$P_1 = P_n + \rho_n g(z_n - z_1) = P_n - \rho_n g h_1 \quad \text{where } h_1 \equiv z_1 - z_n. \tag{C.4.2a}$$

and

$$P_2 = P_m + \rho_m g(z_m - z_2) = P_n - \rho_m g h_2 \quad \text{where } h_2 \equiv z_2 - z_m. \tag{C.4.2b}$$

72

The relative heights, h_1 and h_2, are a convenient way of expressing the element inlet and outlet heights. Similar flow elements in similar rooms tend to have the same relative heights even though the rooms are at different heights. If the element is part of a duct network, the relative heights will be zero. Equation (C.4.1) is thus reduced to

$$\Delta P_i = P_n - P_m + \rho g(z_n + h_1 - z_m - h_2) - \rho_n g h_1 + \rho_m g h_2 \qquad (C.4.3)$$

The terms $[\rho g(z_n + h_1 - z_m - h_2) - \rho_n g h_1 + \rho_m g h_2]$ can be collectively called the stack pressure, PS_i, acting on element i. For airflow in the positive direction:

$$PS_i = \rho_n g(z_n - z_m) + h_2 g(\rho_m - \rho_n) \qquad (C.4.4a)$$

For flow in the negative direction:

$$PS_i = \rho_m g(z_n - z_m) + h_1 g(\rho_m - \rho_n) \qquad (C.4.4b)$$

If the direction of flow is unknown, use

$$PS_i = (g/2)[(\rho_n + \rho_m)(z_n - z_m) + (\rho_m - \rho_n)(h_1 + h_2)] \qquad (C.4.4c)$$

Equation (C.4.4c) could also be used if it is assumed that the air temperature within the element is the average of the two nodal temperatures.

Elements in building envelope surfaces connect the rooms to ambient air which may be moving at significant speed. This speed is reduced to zero as the air meets the building surface; the dynamic pressure of the ambient air is converted to an additional pressure on the element, PW_i, which will be called the wind pressure. Wind pressure may act in either a positive or negative fashion on the element airflow. Equation (C.4.3) is thus reduced to

$$\Delta P_i = P_n - P_m + PS_i + PW_i \qquad (C.4.5)$$

Calculation of the stack pressure interacts with the iterative solution for the nodal pressures. It is possible to recompute PS_i at every iteration, but this can mean that as the estimated pressures change with each iteration flow directions may also change leading to different values for PS_i during the iterations. These changing estimates of the stack pressure interfere with the convergence of the nodal pressure solution. It appears to be better to compute PS_i for each component based on the most recent node pressures and to leave these values unchanged during the solution process. The new pressure values can be used to recompute PS_i and then again solve for pressures in a very few iterations because the pressures will probably change by only very small amounts. Because of the relatively small effect of flow direction on the overall solution, it is probably not necessary to recompute PS_i during a given timestep in a transient building performance simulation.

More complicated stack effects can be created by assuming the nodal temperature varies with height. This could be used to model the effects of various temperature stratifications. Three fairly simple temperature variations come to mind. First, the temperature may vary linearly with height. Second, each node could consist of two regions of constant temperature, perhaps with the height of the interface varying. Third, the first two models could be combined.

C.5 Fitting powerlaw coefficients

Data is seldom available for direct substitution into the powerlaw equation:

$$w = C \sqrt{\rho} \, (\Delta P)^x \qquad\qquad (C.5.1)$$

where
    w = mass flow rate of air,
    C = flow coefficient,
    ΔP = pressure difference
    ρ = air density, and
    x = flow exponent.

It is usually necessary to convert the available data to that form.

If x is known or can be assumed (to be ½ for example), C can be computed from the inverse of equation (C.5.1)

$$C = w \, / \, [\sqrt{\rho} \, (\Delta P)^x] \qquad\qquad (C.5.2)$$

For example, this can be used with the component leakage area formulation which has been used to characterize openings for infiltrations calculations (ASHRAE, 1985, p 22.16). The effective leakage area is given by

$$L = 10000 \, Q \, / \, \sqrt{2\Delta P/\rho} \qquad\qquad (C.5.3)$$

where
    L = the effective leakage area (cm^2),
    Q = the volumetric flow (m^3/s),
    ΔP = standard pressure difference (4 Pa), and
    ρ = density of air (1.2 kg/m^3).
This corresponds to a value of $C = 0.0001414 \, L$ for use with equation (C.5.1).

When two values, $w_1(\Delta P_1)$ and $w_2(\Delta P_2)$, are known, both C and x can be computed:
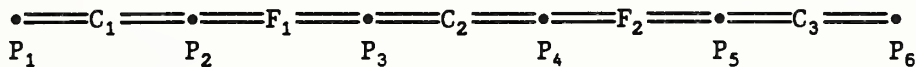
$$x = [\, \ln(w_1) - \ln(w_2) \,] \, / \, [\, \ln(\Delta P_1) - \ln(\Delta P_2) \,] \qquad\qquad (C.5.4)$$

with C then computed from equation (C.5.2). This can be used to convert detailed duct calculations to powerlaw formulation. Tests indicate that the characteristics of ducts with friction modeled by the Colebrook equation can be replaced by the powerlaw representation with an accuracy of better than 2% for flows that vary by up to a factor of four (a typical variation in a VAV system). Equation (C.5.4) would also be useful for reducing a collection of flow elements which are in series to a single flow element.

C.6 Constant Flow Element

One particularly simple but useful airflow element sets a constant flow between two nodes. Since the flow is constant, the partial derivatives of flow with respect to the node pressures must be zero. The constant flow element element does not contribute to the Jacobian, [A], but it does add to the right side vector, {B}.

Constant flow elements do not mathematically link the pressures of the adjacent nodes. It is necessary that all node in the network be linked to constant pressure nodes in order to have a unique solution. Violation of this restriction will produce a division by zero somewhere in the solution of the equations. Consider the following simple network:

$$\bullet\!=\!=\!=\!C_1\!=\!=\!=\!\bullet\!=\!=\!=\!F_1\!=\!=\!=\!\bullet\!=\!=\!=\!C_2\!=\!=\!=\!\bullet\!=\!=\!=\!F_2\!=\!=\!=\!\bullet\!=\!=\!=\!C_3\!=\!=\!=\!\bullet$$
$$P_1 \qquad P_2 \qquad P_3 \qquad P_4 \qquad P_5 \qquad P_6$$

where P_1 and P_6 are known pressures, and F_1 and F_2 are known flows. Since the flow through C_2 is determined by $P_3 - P_4$, and no other flow is related to those pressures, there are not enough equations to determine P_3 and P_4 uniquely. In addition, it is possible for F_1 and F_2 to be assigned different flows, which produces a physically impossible condition.

C.7 Constant Power Fan Element

One simple fan model is applicable for an assumption of near constant operating conditions. This is the constant power fan model, which assumes that a constant power is delivered by the fan to the airstream. The basic equation is:

$$H = Q \cdot \Delta P = w \cdot \Delta p / \rho \tag{C.7.1}$$

where
$$H = \text{power (W)},$$
$$Q = \text{volume flow rate (m}^3\text{/s)},$$
$$\Delta P = \text{pressure rise (Pa)},$$
$$w = \text{mass flow rate (kg/s), and}$$
$$\rho = \text{air density (kg/m}^3\text{)}.$$

Equation (C.7.1) rearranges to

$$w = H \cdot \rho / (P_m - P_n) \tag{C.7.2}$$

for which the partial derivatives are:

$$\frac{\partial w_i}{\partial P_n} = -H\rho/\Delta P^2 \tag{C.7.3}$$

and

$$\frac{\partial w_i}{\partial P_m} = H\rho/\Delta P^2 \tag{C.7.4}$$

C.8 Relationships for Cracks

Baker, Sharples, and Ward (1987) indicate that infiltration openings can be more accurately modeled by a quadratic relationship of the form

$$\Delta P = A\,Q + B\,Q^2 \qquad\qquad\qquad\qquad (C.8.1)$$

They give theoretical relationships between A and B and the physical characteristics of the openings. These are

$$A = 12\mu z / Ld^3 \qquad\qquad\qquad\qquad (C.8.2a)$$
and
$$B = \rho C / 2d^2 L^2 \qquad\qquad\qquad\qquad (C.8.2b)$$
where
μ = viscosity,
ρ = density,
z = distance along the direction of flow,
d = crack width,
L = crack length, and
C = 1.5 + number of bends in the flow path.

Equations (C.8.2a,b) are easily converted to mass flow form for the quadratic flow element:

$$a = 12\mu z / \rho Ld^3 \qquad\qquad\qquad\qquad (C.8.3a)$$
and
$$b = C / 2\rho d^2 L^2 \qquad\qquad\qquad\qquad (C.8.3b)$$

Clarke (1985) also indicates a relationship for flow through cracks that can be converted directly into a powerlaw airflow element. This relationship is:

$$Q = k\,a\,(\Delta P)^x \qquad\qquad\qquad\qquad (C.8.4a)$$

$$x = 0.5 + 0.5\,\exp(-W/2) \qquad\qquad\qquad\qquad (C.8.4b)$$

$$k = 0.0097(0.0092)^x \qquad\qquad\qquad\qquad (C.8.4c)$$
where
W = crack width (mm), and
a = crack length (m).

The powerlaw equation is:

$$w_i = C\,\sqrt{\rho_n}\,(\Delta P)^x \qquad\qquad\qquad\qquad (C.8.5)$$

Therefore, x is given by (C.8.4b) and

$$C = \sqrt{\rho_n}\,a\,0.0097(0.0092)^x \qquad\qquad\qquad\qquad (C.8.6)$$

C.9 <u>References</u>

ASHRAE (1985) <u>ASHRAE Handbook - 1985 Fundamentals</u>, ASHRAE, Atlanta GA.

Baker, P.H., Sharples, S., & Ward, I.C. (1987) "Air Flow Through Cracks", <u>Building and Environment</u>, Vol 22, No 4, Pergamon, pp 293-304.

Clarke, J.A., 1985. <u>Energy Simulation in Building Design,</u> Adam Hilger Ltd., Bristol and Boston.

Conte, S.D. & de Boor, C., 1972. <u>Elementary Numerical Analysis: an Algorithmic Approach</u>, McGraw-Hill, pp 84-87.

Dhatt, G., & Touzot, G. (translated by G. Cantin), 1984. <u>The Finite Element Method Displayed</u>, John Wiley & Sons.

Kernighan, B.W. & Ritchie, D.M., 1978. <u>The C Programming Language</u>, Prentice-Hall, Inc.

Pissanetzky, S., 1984. <u>Sparse Matrix Technology</u>, Academic Press.

Stoecker, W.F., 1971. <u>Design of Thermal Systems</u>, McGraw-Hill, pp 92,93.

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA** **SHEET** (See instructions)	1. PUBLICATION OR REPORT NO. NISTIR 89-4072	2. Performing Organ. Report No.	3. Publication Date APRIL 1989

4. TITLE AND SUBTITLE

AIRNET - A Computer Program for Building Airflow Network Modeling

5. AUTHOR(S)

George N. Walton

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions) NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY ~~NATIONAL BUREAU OF STANDARDS~~ **U.S. DEPARTMENT OF COMMERCE** **GAITHERSBURG, MD 20899**	7. Contract/Grant No. 8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

U.S. Department of Energy
Washington, DC 20585

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

In spite of its importance, the analysis of airflows has significantly lagged the modeling of other building features because of limited data, computational difficulties, and incompatible methods for analyzing different flows. Methods have been developed to analyze airflows in HVAC ducts and to estimate infiltration but the interaction between building HVAC systems and infiltration airflows has seldom been studied. This report describes a computer program for modeling networks of airflow elements, such as openings, ducts, and fans. It emphasizes the numerical aspects of an airflow network method which would provide a unified approach to building airflow calculations. It also discusses the limitations of the method and poorly understood factors that could profit from further research.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

airflow; computer; duct; fan; infiltration; modeling; systems; ventilation.

13. AVAILABILITY ☒ Unlimited ☐ For Official Distribution. Do Not Release to NTIS ☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. ☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161	14. NO. OF PRINTED PAGES 84 15. Price $14.95