Considerations for Advanced Building Thermal Simulation Programs

George N. Walton, Mechanical Engineer
Center for Building Technology
National Institute of Standards and Technology
Gaithersburg, MD 20899

ABSTRACT

In order to assess the applicability of a more modular approach to the development of building thermal analysis programs, this paper begins with a review of some of the basic numerical methods used in simulation. These are discussed with some observations from other fields of study besides building simulation. Two major examples of advanced simulation methods are presented: the use of sparse matrix methods for heat transfer simulation and a modular calculation of building airflows. Their implications on the development of the next generation of building thermal analysis programs

INTRODUCTION

The thermal modeling of buildings has traditionally been divided into two areas: modeling of the structure and modeling of the air conditioning system. The air conditioning system is sometimes considered in two parts: the air handling system and the central plant. The splitting of the building from the system and the hard-coding of system component linkages in most current building energy analysis programs limit the ability of the analyst to handle uncommon features, which, after all, is a primary benefit of a computer analysis.

Two of the more modern approaches to modular simulation are the ESP (Environmental Systems Performance) (Clarke, 1985) and SPANK (Simulation Program ANalysis Kernel) (Sowell et.al., 1986) programs. ESP uses a finite volume approach (R-C network analogy) to divide the building and its system into small elements. Transient simulation is achieved by a central difference solution of the resulting difference equations with a sparse solution technique for the simultaneous equations. SPANK treats the problem description in terms of interconnected "objects". Objects are software representations of physical systems or data manipulators. The key feature of the program is the automatic simplification of the algebraic equations through the use of graph theory.

Since it is generally believed that a more modular approach will be needed in the next generation of building thermal analysis programs, this paper reviews some of the basic numerical methods used in simulation. These are discussed with some observations from other fields of study besides building simulation. Two major examples of advanced simulation methods are presented, and their implications on the development of the next generation of building thermal analysis programs are discussed.

SYSTEMS MODELING

There is an extensive body of literature on the modeling of many types of systems. The following topics, which should be applicable to building thermal simulation, have been drawn from many areas.

Time integration

Following the discussion of Belytschko (1983, pp. 55, 419, 445), the advantages of explicit time integration are:
(1) Fewer calculations per time step.
(2) Algorithm logic and structure are simple; this implies that it is good for testing new ideas.
(3) Complex nonlinearities are easily handled.
(4) It requires little core storage compared to implicit methods using direct equation solvers.
(5) It is very reliable in terms of accuracy and completing the computation.

The only notable disadvantage is that explicit time integration is only conditionally stable so that a very large number of time steps may be required.

With regard to accuracy, since implicit methods are unconditionally stable, they can easily be used with too large a time step leading to significant time integration errors. The stability requirements for explicit time integration force the time step to be so small that the time integration error is almost always smaller than the spatial discretization error.

Certain classes of problems require small time steps to achieve suitable accuracy. Long timesteps are generally suitable for inertial problems, in which low frequencies dominate the response. Short timesteps are required for problems with high frequency transients such as wave propagation problems, e.g., airflow in ducts. Control actions also include high frequency transients.

Clausing (1969) gives a discussion of stability in terms of thermodynamic laws. This suggests a simple technique to determine the minimum stable timestep for any element in the system. In general, the smaller the thermal mass of the element, the smaller the timestep. It may be possible to greatly increase the length of the timestep with modified explicit methods (Kujawski, 1988).

Hamming (1973, pp. 422-423) discusses the problem of modeling a system which has different components with greatly different time constants. Such systems can often be broken into two sets, one with long time constants and one with short time constants. This is sometimes called "time splitting". As long as the time constants are never close under any circumstances, it should be possible to solve the two sets of equations separately.

Solution of Simultaneous Equations

One of the most important aspects of implicit methods is the solution of the linear (and non-linear) algebraic equations. There are two basic means for solving such simultaneous equations: (1) Newton's method (which become direct elimination, e.g., Gaussian elimination, for linear systems), and (2) iterative methods.

Direct elimination methods are most popular for linear systems with constant coefficients because triangularization and back substitution methods can be used. Triangularization is relatively time consuming but need be done only once; back substitution, which is much quicker, is done at every step. Non-constant systems require full solution (triangularization and back-substitution) at every step. Nonlinear problems require multiple Newton iterations to convergence at each step. Iterative methods are much more sensitive to the presence of high eigenvalues than are direct methods.

In general we do not directly solve nonlinear equations on the computer. We solve linear equations in an iterative fashion to represent the nonlinear equations. A significant decision is where the nonlinearity is approximated in the solution process. This relates to the notes on explicit methods. If the timestep is small enough, there will be a relatively small change in systems conditions at each timestep. This allows the system to be adequately modeled by using a linear approximation of system performance at each timestep. Analysis with ESP (and TARP) indicate that nonlinearities in the building shell heat transfer analysis are not a significant computational problem. Conditions are changing slowly enough that successive linear solutions are adequate. Note that SPANK is working toward a solution of simultaneous nonlinear equations. This may not be important except in cases of extreme nonlinearities because linearization at each timestep may provide adequate solutions. On the other hand, extreme nonlinearities, such as discontinuous derivatives, can cause problems with Newton's method.

There appears to be a convergence between the ideas of matrix representation of equations and the SPANK graph representation. Sparse matrix techniques presently rely on graph theory to order the equations for a numerically stable solution of the equations while preserving sparsity (Pissanetzky, 1984, chs. 4 & 5). The graph theory approach automatically maintains sparsity and is beginning to consider efficient ordering of the equations for numerical stability as well as for efficient solution. The latest ideas in SPANK put all the equations "on the table" for analysis (instead of hiding them in subroutines) somewhat like the matrix approach displays all of the equations simultaneously. Matrix theory has been developed to test for convergence for iterative and numerical stability for direct solution processes. SPANK does not presently have the mathematical tools for analysis of equations that matrix theory has. Development of such tools may take years of work. During development of SPANK it has sometimes been found that extreme nonlinearities (due to control actions) caused problems requiring "clever" solutions.

Large Systems

Coupled systems are an area of current research, especially in the area of coupled field problems. The most promising of the mixed time integration schemes that have been described is an implicit-explicit partitioning of the system of equations (Liu & Lin, 1983). That is, the equations for quickly responding elements are integrated using short explicit timesteps, and the entire system is integrated implicitly at a long timestep. This is in agreement with the prior observation that explicit integration is most applicable to problems with rapidly changing conditions, while implicit integration is best for inertial problems.

Some work has been done with combined explicit and implicit modeling for building thermal performance. Sebald (1979) used explicit calculations of the massive elements in passive solar buildings together with implicit calculations for the massless elements. This method was used to reduce the number of simultaneous equations to speed the solution. Relatively long 20-minute timesteps were used. Note that this is exactly the opposite of the method used by Liu and Lin, which indicates more study on explicit/implicit methods could be appropriate.

Most energy analysis programs to date have assumed quasi-steady HVAC component models. Some are questioning this assumption, and tools are needed to resolve that question. The GEMS program was used to model the control of a residential heat pump in detail (Benton et.al., 1982). This showed that the existing control algorithm was inefficient and led to the development of a better control algorithm. A model of the heat pump and control algorithm suitable for use with a long timestep energy analysis was developed from the short timestep state-space model and Z-transform theory. This idea of using detailed models to create simplified models should be generally valid (although nonlinear models will require a different process).

Control Theory

Modern control theory has evolved from frequency domain analysis to time domain analysis because of the capabilities of the modern digital computer (Elbert, 1984). The state-space method is built around liner systems of algebraic and differential equations which take advantage of matrix theory and matrix solution techniques. It leads to a concise, systematic approach to computer modeling of systems. It is possible to unify many fields under systems theory. Optimization is possible through the calculus of variations.

It is recommended (Clarke & McLean, 1986, p. 5.4) that new plant component models for ESP be developed in state-space form. This terminology was not used in the earlier reference (Clarke, 1985) indicating the appropriateness of state-space theory to what is primarily a control problem. The previously mentioned GEMS program, which has its primary emphasis on controls simulation, is based on the state-space method.

The advantage of the state-space method is that "the discrete nature of practical solutions of the time domain problem produce the very digitally

oriented results needed for efficient use of the emerging digital hardware" (Elbert, 1984). In other words, the digital simulation can be directly related to the new digital control hardware. This has an interesting implication for the computation time required for a state-space simulation of a system. Since a digital control system is essentially a computer performing the same kind of calculations that must be done to simulate the control system, the execution time of a simulation will be of a similar order as the actual operation time. The differences lie in the greater power of computer typically used for simulation which is offset by the need to compute the performance of all the components in the system and the possibility that the digital controller may not be operating at full speed. The gains that can be made in simulation time by using a longer timestep than is used by the controller lead to questions about the accuracy of the simulation.

EXAMPLES

Solving Simultaneous Equations

Since the solution of simultaneous linear equations plays such an important part in system simulation, some direct solution processes will be studied in relation to the solution of building thermal simulation problems. The general form of the problem is

$$[A]\{T\} = \{B\} \qquad (1)$$

where

[A] is a square matrix of coefficients,

{T} is a vector of temperatures - the desired solution, and

{B} is a vector of coefficients.

In general the solution of equation (1) involves four steps. First, the coefficients of [A] and {B} must be computed from energy conservation principles. Then those values are transferred into the matrix and vector. Then [A] is factored into upper, [U], and lower, [L], triangular matrices such that [L][U]=[A]. Finally, {T} is computed by

$$\{x\} = [L]^{-1}\{B\} \qquad (2)$$

and

$$\{T\} = [U]^{-1}\{x\} \qquad (3)$$

Separating the factoring from the solution step is especially useful when multiple solutions are desired -- there are multiple {B} vectors to be solved with a single [A] matrix. In that case the factoring of [A], which is the most time consuming part of the solution, need not be repeated. Usually, the [L] and [U] matrices are stored in the same computer memory locations as [A].

Figure 1 shows the general configuration of [A] for a simple building consisting of four rooms, each room consisting of two exterior walls, two partition walls, a ceiling and a floor. Solving for all wall surface temperatures plus the air and radiant temperatures leads to a set of 40 simultaneous equations. In figure 1 those elements of [A] which are non-zero are indicated by the symbol "•". Note the relatively few non-zero entries: of the 1600 coefficients in [A] only 168 are not zeroes.

Three solution processes were developed, tested, and analyzed. The first method factors [A] by a Gauss elimination algorithm (Conte, 1973) written for maximum efficiency. Common safety features, such as partial pivoting, have been eliminated because this set of equations is diagonally dominant and pivoting will not improve accuracy or stability of the solution. Some of the zeroes will be modified during the Gauss elimination process. This is called "fill". These values are indicated in by "+" symbols. There are still only 206 non-zero values out of the total 1600 coefficients in [A].

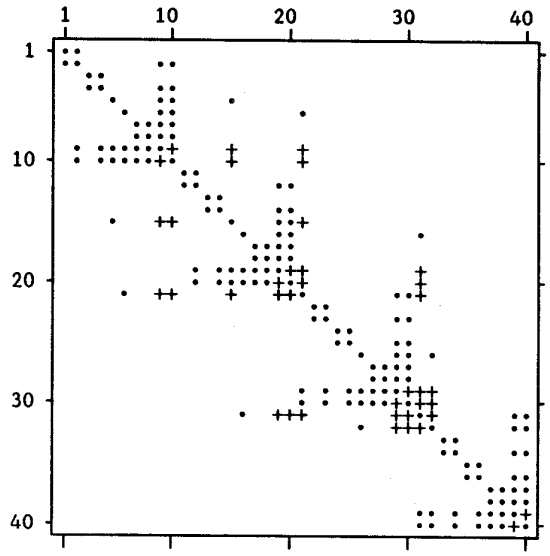

Figure 1. Configuration of the [A] matrix

The other two methods arise naturally from the structure shown in figure 1. Note that fill occurs only below the topmost element in any column above the diagonal and to the right of any element below the diagonal. The second method uses only those elements within that envelope. It is commonly called a "column solver" or "skyline" method. A FORTRAN subroutine presented by Dhatt, et.al. (1984, p. 285) was modified for this method.

However, not all the elements below the top of a column will fill. It is possible to analyze the positions of the elements in [A] to determine exactly where fill can occur. Pissanetzky (1984, p. 263) calls this a "symbolic" factorization, which determines a minimal storage requirement. Pissanetzky's algorithm for factoring a symmetric matrix served as the model for the third algorithm which factors a non-symmetric matrix.

These algorithms were coded in FORTRAN and C and executed on various computers. The results of the C algorithms run on an engineering workstation are summarized in table 1. The number of elements required to represent [A] is reported for each method. "Fill fraction" is the ratio of the number of elements required by the sparse techniques to the number in the full array. Sparse matrix methods can save considerable storage space. Note that fill fraction may not decrease uniformly with increasing array size, and that the fill fraction is strongly influenced by the ordering of the equations.

Number of equations	12	40	80	160
Gauss elimination				
elements in [A]	144	1600	6400	25600
fill time	1.7s	7.1	17.3	49.3
factor time (full)	3.6	109.5	847.6	6650.
factor time (quick)	1.5	16.7	88.2	405.0
solve time	0.8	6.6	24.6	95.0
Skyline method				
elements in [A]	60	284	1192	3140
fill fraction	41.7%	17.8%	18.6%	12.3%
fill time	1.8s	6.2	13.1	26.7
factor time	1.5	7.4	40.6	120.7
solve time	0.6	2.2	7.2	18.2
Symbolic method				
elements in [A]	48	206	668	1624
fill fraction	33.3%	12.9%	10.4%	6.3%
fill time	0.7s	2.7	6.7	14.8
factor time	1.0	4.6	22.6	72.0
solve time	0.5	1.8	5.1	11.8

Table 1. Comparison of direct solution methods

Times are seconds required to perform each operation 1000 times. The four problems sizes correspond to 1 room, 4 rooms, 8 rooms (4 rooms x 2 floors), and 16 rooms (4 rooms x 4 floors).

No time was used to compute the values going into [A] and {B}; they were constant. However, the time required to transfer those values from the arrays in which they were stored to [A] and {B} is a nonnegligible fraction of the total solution time for the sparse techniques. This indicates that calculating the values could be a very significant portion of the total solution time.

Times are reported for two slightly different Gauss elimination factoring methods. The first processes all of [A] and reaches very large execution times. This behavior follows the classic analysis showing that matrix factoring is an order N^3 process, where N is the number of equations. The second process checks for zeroes below the diagonal and greatly improves performance. Solution time is order N^2.

The symbolic method is generally about one third faster than the skyline method in factoring and solution time. Note that the symbolic factoring time is generally less than the Gauss solution time -- for this case factoring is approximately an order N^2 process. The skyline process requires less time to set up than the symbolic method, and so may be preferred for a single solution of a given [A] structure. It is also more easily vectorized for supercomputer simulation.

Some comparisons to SPANK may be helpful. SPANK reduces the number of equations to be solved. Anderson (1986, p. 24) reports a typical factor of N/4. When using the order N^3 Gauss method, SPANK should be 64 times faster. However, the symbolic sparse method described above would be faster than SPANK as long as $N^2 < (N/4)^3$, or $N > 64$, and sparsity is maintained while SPANK must use a full matrix. The time required to compute all the Jacobian coefficients can also be high. It could be useful to retain a sparse Jacobian.

Airflow Calculations

One of the weakest areas in building thermal simulation has been airflow analysis, in spite of its importance for energy, moisture, and contaminant migration calculations. Modeling of airflows requires: (1) determination of the location and mathematical characterization of the airflow openings, (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. The analysis of airflows has significantly lagged other features because of limited data and computational difficulties. This is particularly true of the combined building and plant simulation. Methods have been developed to analyze airflows in ducts and to estimate infiltration, but the intimate relationship between these two processes has seldom been studied.

Relatively few methods that could be applied to both processes have been developed within the building simulation community and described in detail. These include methods by Sander (1974), Klote (1983), 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 connections (openings) which is conceptually similar to the airflow network comprising the air handling system. Conservation of mass for the flows into and out of each room leads to a set of simultaneous nonlinear equations which are solved iteratively to a satisfactory accuracy.

For this study Walton's method (1984) was rewritten into a more modular structure in a program called AIRNET (Walton, 1989) which includes the following features:

(1) a solution method for nonlinear equations consisting of a traditional Newton's method (Conte & de Boor, p. 86) combined with Steffensen iteration (Conte & de Boor, p. 54) to accelerate convergence;
(2) airflow element subroutines which compute the flow rate and derivative of the flow with respect to pressure for a given pressure difference;
(3) a process for transferring the above data into the Jacobian matrix;
(4) solution of the simultaneous linear equations involving the Jacobian matrix (skyline method); and
(5) a process for establishing an initial set of values to start the iterative solution process.

This modularization of the airflow solution process has allowed several problems to be attacked individually. The Richardson iteration, which is normally applied to fixed point iteration, has solved a problem of oscillation in Newton's method for simultaneous nonlinear equations. Airflow element subroutines allowed different features to be modeled. These include infiltration cracks, ductwork, fans, dampers, and doorways. The separate process to transfer data into the Jacobian matrix made it easy to convert from the use of a complete matrix to a sparse matrix.

The ESP program includes a separate program for calculating airflows, ESPAIR. This module was compared to AIRNET. Both programs used the same compiler and workstation computer. A test case was developed which described a four story building with six rooms, a hallway, elevator shaft, and stairwell on each floor. The nodes representing the elevator shaft and stairwell on each floor were connected by very large (2.0 m^2) openings. Similar openings connected each room to the hallways. Very small (0.01 m^2) openings connected the building nodes to the outside. Intermediate size openings (0.1 m^2) connected the large vertical shafts to the hallways. ESPAIR solved this case to 5% accuracy in about 8800 iterations requiring a total of 150 seconds. AIRNET solved it to 0.01% accuracy 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 ESPAIR has with large openings (Clarke, 1985, p. 206). Limiting all the openings to an area of 0.01 m^2 allowed ESPAIR to reach a solution in only 137 iteration and 2.10 seconds. AIRNET was also somewhat faster: 2 iterations and 0.06 seconds, or about 35 times faster. Greater accuracy in the ESPAIR solution required more iterations and more time. The performance of the test method indicates that it is practical to solve the flow network in detail for such problems. Solution of complex airflow networks for the steady-state case is practical on current PC's. Solution of the dynamic case for many timesteps is now feasible. Research is still needed on the dynamic case, where the flows at one timestep may be good initial values for the next timestep.

The modularity of the ESP program and AIRNET had several advantages. ESP's modularity made the test possible. It is often very difficult to isolate a single computational feature of a monolithic program. The AIRNET program included 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 skyline solution of the simultaneous equations 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 filling processes, faster solution processes could be easily substituted.

Some factors relate directly to the current version of the SPANK program. SPANK's use of pointers to functions, a feature of the C language, have significantly improved the interface between the core algorithm and the airflow element models. SPANK's Newton's method solution could be inadequate if its equation simplifying process leaves the same equations that are presently being solved.

An important question remains about the reliability of the AIRNET algorithm. Solution of the nonlinear equations has been demonstrated in several tests but has not been mathematically proven. Literature for the solution of similar equations could 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. Note that all the airflow programs mentioned previously use node methods.

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; they tend to require the solution of more simultaneous equations; the equations do not have strong diagonals; they tend to be less sparse than the node equations; and some airflow elements, such as doorways, may be difficult to implement. 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. Some of the airflow elements are described more naturally in one form than the other. For others, the transformation is simple. This indicates the need to consider the solution technique in the development of component models. The work of Wood and Rayes indicates that several apparently reasonable solution methods for the simultaneous nonlinear equations are not very reliable.

This question of reliable solution techniques carries over into the full building performance simulation. The ESP and TARP programs, which consider simultaneous airflow and heat transfer, operate by iterating between separate solutions of the airflow and the thermal equations. This has been observed to function well except in cases where control actions on fans can cause significant changes in the airflows during the iterations. A completely simultaneous solution remains to be done.

SUMMARY AND CONCLUSIONS

These system modeling trends suggest several ideas for an advanced building thermal modeling system:
(1) Generality and modifiability are almost always achieved at a cost of execution time performance. In order to compare different programs, we must quantitatively know the importance of each factor. Since these are not known, we are led to a qualitative discussion of program performance.
(2) Time to calculate system dynamics overwhelms the building heat transfer calculations because of

short timesteps (about 1/1000th as long) and increased number of equations. This change in the scale of the problem applies to any solution process. At this time scale explicit solution of the equations may be advantageous, if numerical stability can be attained.

(3) Analysis with ESP (and TARP) indicate that nonlinearities in the building shell heat transfer analysis are not a significant computational problem. Conditions are changing slowly enough that successive linear solutions are adequate. Note that SPANK is working toward a solution of simultaneous nonlinear equations. This may not be important except in cases of extreme nonlinearities with discontinuous derivatives which can cause problems for Newton's method as used in SPANK.

(4) The simultaneous equations resulting from application of conservation laws are nearly always sparse. There will be significant gains in execution time if the sparseness is retained during the solution process. This is a primary problem with SPANK: it reduces the number of algebraic equations but produces a set of near-full simultaneous equations.

(5) GEMS tried to use the short timestep component models to develop long timestep component models. This seems promising.

(6) Modern control theory has evolved from frequency domain analysis to time domain analysis because of the capabilities of the modern digital computer. This has been based on the state-space method which takes advantage of matrix theory and matrix solution techniques. When the control actions are dominating the simulation, the simulation should use the technique most suited to modeling controls.

(7) It appears to be impossible for one program to satisfy all needs. The development of two programs is likely: (a) an energy analysis program operating at a time scale of several minutes to one hour, and (b) a system/control dynamics program operating at a time scale of about one second. Program (b) would make use of the building heat transfer algorithms from program (a) to establish boundary conditions, and it would be used to provide program (a) with simplified component models to be used with the longer timesteps.

ACKNOWLEDGEMENTS

This work was sponsored by the US Department of Energy, Office of Building and Community Systems, contract DEAI0136CE21013 modification no. 003. The author also wishes to thank Jim Axley for the long discussions on simulation methods, Joe Clarke for use of the ESP program, and Ed Sowell for access to the SPANK program.

REFERENCES

Anderson, J.L., 1986. "A Network Definition and Solution of Simulation Problems", LBL Report 21522, Lawrence Berkeley Laboratory, Berkeley CA.

Belytschko, T., 1983. "An Overview of Semidiscretization and Time Integration Procedures", Chapter 1 in Computational Methods for Transient Analysis, edited by T. Belytschko and T.J.R. Hughes, North-Holland.

Benton, R., MacArthur, J.W., Mahesh, J.K. and Cockroft, J.P., 1982. "Generalized Modeling and Simulation Software Tools for Building Systems", ASHRAE Transactions, Vol 88, Part 2.

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

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

Clausing, A.M., 1969. "Numerical Methods in Heat Transfer" in Advanced Heat Transfer, Ed. B.T. Chao, University of Illinois Press, Urbana IL.

Conte, S.D. & de Boor, C., 1972. Elementary Numerical Analysis: an Algorithmic Approach, 2nd Ed., McGraw-Hill.

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

Elbert, T.F., 1984. Estimation and Control of Systems, Van Nostrand Reinhold.

Hamming, R.W., 1973. Numerical Methods for Scientists and Engineers, McGraw-Hill.

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.

Kujawski, J., 1988. "Stability Improvement of the Euler FOrward Method for Initial Value Problems", Numerical Methods for Partial Differential Equations, Vol.4, pp. 319-327.

Liu, W.K. & Lin, J.I., 1983. "Mixed Time Integration Schemes for Transient Conduction Forced-Convection Analysis" in Numerical Properties and Methodologies in Heat transfer: Proceedings of the Second National Symposium, Edited by T.M. Shih, Hemisphere Publishing Corp.

Pissanetzky, S., 1984. Sparse Matrix Technology, Academic Press.

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

Sebald, A.V., 1979. "Efficient Solution of Large, Controlled Passive Solar Energy Systems: Forward Differencing in Thermal Networks", UCSD Energy Center Report, University of California, San Diego, La Jolla CA.

Sowell, E.F., Buhl, W.F., Erdem, A.E., & Winkelmann, F.C., 1986, "A Prototype Object-Based System for HVAC Simulation", LBL report 22106, Lawrence Berkeley Laboratory, Berkeley CA.

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

Walton, G.N., 1984. "AIRNET - A Computer Program for Building Airflow Network Modeling", NISTIR 89-4072, Natl Inst of Standards & Technology.

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.