MODELING OF HEAT TRANSFER IN BUILDINGS

by

JOHN ERVIN SEEM

A thesis submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
(Mechanical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

1987

ABSTRACT

Year long simulations of the hourly (or shorter time period) heating and cooling loads for buildings are important for sizing heating, ventilating, and air conditioning equipment, determining the effect of a design change or retrofit on energy usage, and developing optimal control strategies. Transfer functions, which are more efficient than Euler or Crank-Nicolson solutions of finite-difference or finite-element models and response factor methods, were developed to reduce the amount of computational effort. Transfer functions relate the output of a linear, time-invariant system to a time series of current and past inputs, and past outputs. Inputs are modeled by a continuous, piecewise linear curve.

Finite-difference or finite-element methods reduce transient multi-dimensional heat transfer problems into a set of first order differential equations when thermal physical properties are time-invariant and the heat transfer processes are linear. This thesis presents a method for determining transfer functions from the exact solution of a set of first order differential equations.

This thesis describes a method in which the transfer functions describing heat flows in building elements are combined into a single transfer function for an enclosure, referred to as a comprehensive room transfer function (CRTF). The method accurately models long-wave radiation and convection in an enclosure through an approximate network, referred to as the "star" network. Resistances in the star network are determined from a network which uses view factors to model long-wave radiation exchange.

This thesis presents model reduction methods for reducing the number of coefficients in transfer functions. These methods can be used to significantly reduce the computational effort of CRTF simulations.

ACKNOWLEDGMENTS

I have been very fortunate to work and study under the supervision of three great advisors: Sanford A. Klein, William A. Beckman, and John W. Mitchell. They gave me the guidance to get started and the freedom to explore new areas. I will be forever indebted to them for their help and encouragement.

Thank you Sandy for always being there to check over my work and giving me constructive suggestions. You constantly amaze me with the amount of work you get done.

Thank you Bill for teaching me the underlying principles of heat transfer. You have more courage and drive than any man I have ever met.

Thank you John for your positive outlook on life. Your enthusiasm was uplifting.

I am thankful that Jack Duffie, the director of the Solar Energy Laboratory, found funding to support me during the past four years. Financial support for this work has been provided by Lawrence Berkeley Laboratory, University of California and the Solar Heating & Cooling Research Development Branch, Office of Conservation and Solar Applications, U. S. Department of Energy.

Thanks Mom and Dad for all you have given me.

TABLE OF CONTENTS

CHAPTER 1
Introduction

1.1 OBJECTIVE

Accurate calculation of the heating and cooling loads of buildings involves the long-time solution of transient conduction, convection, and radiation heat transfer processes. Calculation of these loads requires significant computational effort. The goal of this thesis is to develop fast and accurate methods for computing heating and cooling loads in buildings.

Section 1.2 discusses uses of building simulation programs, i.e., programs which are used to compute heating and cooling loads in buildings. The advantages and disadvantages of frequency domain and time domain methods for modeling heat transfer in buildings are discussed in section 1.3. A review of time domain transfer function methods is contained in section 1.4. Section 1.5 describes the organization of this thesis. In section 1.6, the different definitions of transfer function are presented.

1.2 BUILDING SIMULATION PROGRAMS

Building simulation programs such as BLAST [BLAST (1979)], TARP [Walton (1983)], TRNSYS [Klein (1983)], and [DOE-2 (1980)] are used to calculate dynamic heating and cooling loads for rooms, zones, or buildings. A description of thermal physical properties and dimensions of building components such as walls, floors, ceilings, and windows are required inputs for these programs. In some programs, the location of various interior room surfaces must be supplied for calculation of radiation exchange factors. In addition to a detailed building description, weather data for the building location must also be available. Hall et al. (1978) have generated hourly weather data for a typical meteorological year at 26 locations in the United States. Also, a schedule of energy gains from equipment, lights, and people must be known.

Building simulation programs are used to determine the most economical design or retrofit based upon heating, cooling, equipment, and material costs. An architect has a number of variables to consider when designing a new building or retrofiting an old building. Some of the variables are the following: amount of insulation, use of additional thermal mass to reduce temperature swings, area and type of glass, and use of passive solar components.

Building simulation programs are used to size heating, ventilating, and air conditioning (HVAC) equipment by estimating peak heating or cooling loads for buildings. Oversizing HVAC equipment results in excessive capital cost for equipment and extra energy costs due to inefficient part-load operation of equipment. Undersizing HVAC equipment results in uncomfortable building occupants. Fanger (1972) discusses the variables which influence thermal comfort for human beings.

Building simulation programs are used to develop optimal control strategies which minimize HVAC operating costs while maintaining a comfortable environment for the occupants. Forrester and Wepfer (1984) discuss a number of different methods for reducing operating costs of HVAC equipment.

Passive solar components such as a collector-storage wall, direct gain window, or a sun space are used to reduce building heating requirements. There are design methods such as the solar load ratio method [Balcomb et al.(1983)] and the Un-utilizability design method [Monsen et al. (1981,1982)] available for estimating the long-term thermal performance of buildings which have passive solar components. Results from building simulation programs are used to provide data for the curve fitting of parameters in these design methods.

1.3 FREQUENCY DOMAIN VS TIME DOMAIN

Frequency domain and time domain methods are both used for modeling heat transfer processes in buildings. Frequency domain methods [Goldstein and Lokmanhekim (1979), Hittle (1981), Subbarao (1982), Subbarao and Anderson (1982,1983), Athientis (1985), Athientis and Sullivan (1983,1985)] are based upon the response to sinusoidal inputs. Time domain methods are based upon the response to time dependent inputs. The next section of this chapter describes time domain methods for modeling heat transfer processes in buildings. Both frequency and time domain methods were studied in order to answer the following question: Should improvements be made in frequency domain methods, time domain methods, or a combination of frequency and time domain methods?

Currently, there is no frequency domain approach available which can be used to accurately calculate the energy requirements of a building with thermostatic controls. In the future, a frequency domain method may be developed which will be able to accurately model a thermostat, but there is still another problem with the frequency domain approach. Fourier synthetic weather data approximates real weather data. Anderson and Subbarao (1981), and Smart and Ballinger (1984), used time domain simulation techniques to compare the auxiliary energy use of buildings that were driven with hourly weather data and Fourier synthesized weather data which was obtained from the hourly weather data. Anderson and Subbarao state that "... a few Fourier coefficients can be useful for 'quick and dirty' analysis of buildings which are not highly solar driven."

Smart and Ballinger state that the use of Fourier synthesized weather data leads to systematic underestimating of both heating and air conditioning loads due to the underestimating of the severity of climates.

ASHRAE (1985) has described time domain methods, which use transfer functions, for calculating the energy requirements of a building with thermostatic controls. A disadvantage of time domain methods is the large amount of computational effort required to perform yearly simulations.

Time domain methods are contained in this thesis because accurate calculation of heating and cooling loads are important for sizing heating, ventilating, and air conditioning equipment; determining the effect of a design change or retrofit on energy usage; developing optimal control strategies; and developing design methods. The methods presented in this thesis can be used to significantly reduce the computational effort of time domain simulations.

1.4 TRANSFER FUNCTION AND RESPONSE FACTOR METHODS

Transfer function and response factor methods are used in building simulation programs to compute long-time solutions of transient heat transfer problems in which the system properties are time-invariant. Response factors relate the output of a linear system to a time series of current and past inputs. Transfer functions additionally relate the current output to past outputs, significantly reducing computational effort. In both cases, the inputs or driving functions are modeled by a continuous, piecewise linear curve or equivalently, a series of triangular pulses. See figure 1.1 for a representation of an input by a continuous piece-wise linear curve.

Transfer function and response factor methods are more efficient for solving long-time heat transfer problems than Euler or Crank-Nicolson solutions of finite-difference or finite-element formulations because of two reasons. First, the time step used in a Euler or Crank-Nicolson simulation may be less than the time step in a transfer function simulation. In a transfer function simulation, the time step is selected based upon how well a continuous, piece-wise linear curve represents the inputs. In a Euler or Crank-Nicolson solution, a small time step may be required to avoid numerical oscillation. Second, in a Euler of Crank-Nicolson solution all nodal temperatures are computed at every time step even though they all may not be outputs of interest. In a transfer function simulation, only outputs of interest are computed.

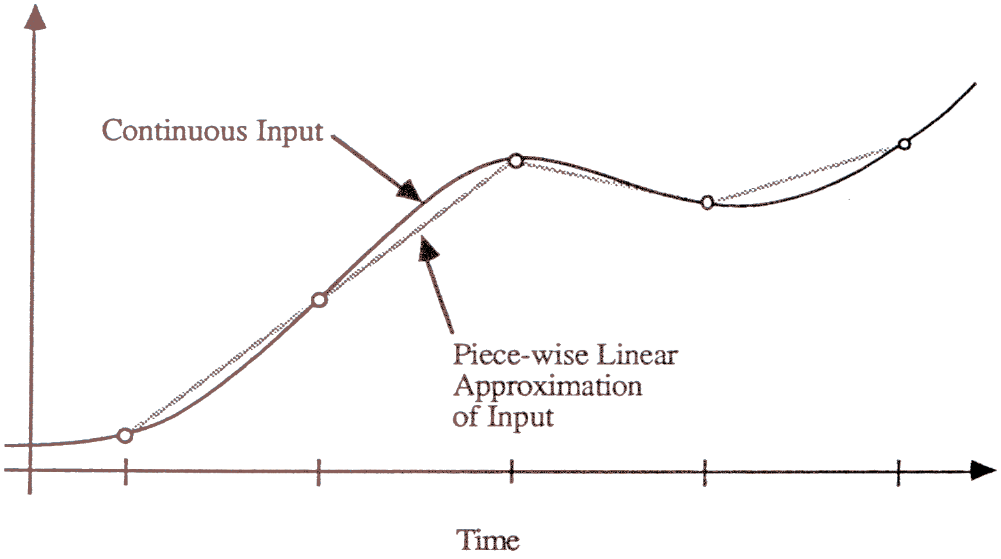


Figure 1.1 Representation of an input by a continuous, piece-wise linear curve.

Stephenson and Mitalas (1967) developed the response factor approach to calculate the transient one-dimensional heat transfer through multi-layered walls, floors, and roofs. The form of the response factor equation to calculate the heat transfer through multi-layered slabs is

$$q''_t = \sum_{j=0} \left[\left(Y_j \, T_{t-j\delta,o} \right) + \left(Z_j \, T_{t-j\delta,i} \right) \right] \qquad (1.4.1)$$

where

q''_t = heat flux across interior surface

Y_j = response factor coefficient for j time steps prior to time t

$T_{t-j\delta,o}$ = outside temperature j time steps prior to time t

Z_j = response factor coefficient for n time steps prior to time t

$T_{t-j\delta,i}$ = inside temperature j time steps prior to time t

Theoretically the number of response factor coefficients approaches infinity, but for computational purposes a finite number of coefficients are used. Kusuda (1969) extended the response factor approach to cylindrical and spherical coordinate systems. Response factors with cylindrical or spherical coordinate systems are important when analyzing the heat transfer from underground pipes, tunnels, and storage tanks.

Stephenson and Mitalas (1971) presented a method for determining transfer functions for one-dimensional heat transfer through multi-layered slabs by solving the conduction equation with Laplace and z-transform theory. Transfer functions for computing heat flow through walls are of the form

$$q''_t = \sum_{j=0} \left[\left(a_j \, T_{t-j\delta,o} \right) + \left(b_j \, T_{t-j\delta,i} \right) \right] - \sum_{j=1} \left(c_j \, q''_{t-j\delta} \right) \qquad (1.4.2)$$

where

a_j = transfer function coefficient for j time steps prior to time t

b_j = transfer function coefficient for j time steps prior to time t

c_j = transfer function coefficient for j time steps prior to time t

$q''_{t-j\delta}$ = heat flux j time steps prior to time t

Mitalas and Arsenault (1971) wrote a program for computing transfer function coefficients based upon the method of Stephenson and Mitalas. Transfer function coefficients computed from Mitalas and Arsenault's program for 40 roof, 103 wall, and 47 interior partition constructions are listed in the ASHRAE Handbook of Fundamentals (1977,1981,1985).

Peavy (1978) presented an alternative approach for calculating transfer functions. He used the fact that successive response factor coefficients are approximately related by a common ratio to convert response factor coefficients into transfer function coefficients. Hittle (1981) has presented a very detailed derivation of transfer functions for multi-layered slabs which used the method of Peavy.

In the calculation of response factor or transfer function coefficients, a numerical search for the roots of a nonlinear equation must be performed. Hittle and Bishop (1983) developed an improved numerical technique for calculating the roots of the nonlinear equation. The improved procedure eliminates the need to take very small steps when numerically searching for the roots. This procedure should decrease the computation time while improving the reliability of finding roots which are very close. Walton (1983) used this procedure in TARP.

To develop transfer functions for multi-dimensional heat transfer, it is necessary to spatially discretize the problem by use of finite-difference or finite-element techniques. Spatial discretization results in a set of first order differential equations. Pawelski (1976) used linear regression on results from a finite-difference simulation to determine transfer function coefficients. Ceylan and Myers (1980) have presented a method for determining the exact set of transfer functions from a set of first order differential equations.

ASHRAE (1985) has discussed the energy balance method (ASHRAE referred to this as the heat balance method) for calculating sensible heating or cooling loads for buildings. An energy balance equation is written for every surface in a room and the room air. For a room with N surfaces, these energy balance equations can be formulated in the matrix equation

$$\mathbf{Z}\,\mathbf{T} = \mathbf{X} \tag{1.4.3}$$

where **Z** is a (N+1) by (N+1) matrix which contains transfer function coefficients, convection coefficients, and linearized long-wave radiation resistances; **T** is a vector of N+1 temperatures with all rows equal to an interior surface temperature, except the last row which is the room air temperature; and **X** is a vector of current inputs, past inputs, past interior surface temperatures, and transfer function coefficients. After solving the matrix equation for the **T** vector, the load due to convective heat transfer between surfaces and the room air can be calculated. The computational effort is reduced when the **Z** matrix is time invariant, i.e., convection coefficients and linearized long-wave radiation resistances are constant. Mitalas (1965) has shown the cooling loads for a room are quite insensitive to changes in the interior convection coefficient. Walton (1980) has shown that long-wave radiation exchange between surfaces in a room can be linearized without introducing significant errors. Assuming long-wave radiation resistances are constant does not introduce significant errors because the average temperature of surfaces in a room is fairly constant. Estimating exterior convection coefficients which vary with time is difficult due to the large number of factors (e.g., building size, building shape, building surrounding, wind direction, local wind velocity) which affect convective heat transfer from exterior surfaces.

Madsen (1982) developed a comprehensive room transfer function (CRTF) by using linear regression on results from an energy balance simulation. A CRTF is a single transfer function equation for computing loads or floating indoor air temperatures in a room or zone. CRTF simulations require less computational effort than energy balance simulations because only outputs of interest are computed.

1.5 ORGANIZATION

Finite-difference or finite-element methods reduce transient multi-dimensional heat transfer problems into a set of first order differential equations when thermal physical properties are time-invariant and the heat transfer processes are linear. Chapter 2 presents a method for determining transfer functions from the exact solution of a set of first order differential equations when the inputs are modeled by a continuous, piecewise linear curve.

Chapter 3 describes a method in which the transfer functions describing heat flows in building elements are combined into a single transfer function for an enclosure, referred to as a comprehensive room transfer function (CRTF). The method accurately models long-wave radiation exchange and convection in an enclosure through an approximate network, referred to as the "star" network. Resistances in the star network are determined from a network which uses view factors to model long-wave radiation exchange.

Chapter 4 contains model reduction methods for reducing the number of significant coefficients in transfer functions. These methods can be used to significantly reduce the computational effort of simulations.

A discussion of the conclusions and recommendations for future work is contained in chapter 5.

1.6 TERMINOLOGY

The definition of transfer function used in the field of heat transfer in buildings is different from that used in the field of automatic controls. In automatic controls a transfer function is the Laplace or z-transform of the output divided by the Laplace or z-transform of the input. In heat transfer, a transfer function is a recursive difference equation which relates the outputs of a linear, time-invariant system to a time series of current and past inputs, and a time series of past outputs. In this thesis, the latter definition will be used. Also, this thesis uses Laplace transfer function as the definition for the Laplace transform of the output divided by the Laplace transform of the input and z-transfer function as the definition for the z-transform of the output divided by the z-transform of the input.

CHAPTER 2

Transfer Functions for Multi-Dimensional Heat Transfer

Ceylan and Myers (1980) present a method for calculating transfer functions for multi-dimensional heat transfer from a set of first order differential equations. Their method involves first calculating response factor coefficients and then converting the response factor coefficients into transfer function coefficients. Also, their method requires the calculation of the eigenvalues and eigenvectors of a matrix.

This chapter presents a method for calculating transfer functions for multi-dimensional heat transfer which results in fewer coefficients than the method of Ceylan and Myers. In addition, the intermediate step of calculating response factor coefficients is eliminated and it is not necessary to calculate the eigenvalues and eigenvectors of a matrix.

Section 2.1 presents the development of equations for computing transfer functions from a system of first order differential equations. In section 2.2, a transfer function is developed for a two node example. Section 2.3 presents a procedure for efficiently computing transfer functions. In section 2.4, transfer functions from finite-difference models are compared with the exact transfer function for a 12" homogeneous concrete wall. Finally, section 2.5 demonstrates the importance of modeling multi-dimensional heat transfer in a roof deck.

2.1 FORMULATION

A state space formulation has traditionally been used to analyze linear systems which may have many inputs and outputs. A heat transfer problem may be formulated in a state space representation by using finite-difference or finite-element methods [Myers (1971)] to spatially discretize the problem. A state space representation for a continuous, linear, time-invariant system with n states, p inputs, and m outputs is

$$\frac{d\mathbf{x}}{d\tau} = \mathbf{Ax} + \mathbf{Bu} \qquad (2.1.1)$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \qquad (2.1.2)$$

where

$\mathbf{x}$ = vector of n state variables

τ = time

$\mathbf{A}$ = (n x n) constant coefficient matrix

$\mathbf{B}$ = (n x p) constant coefficient matrix

$\mathbf{u}$ = vector of p inputs

$\mathbf{y}$ = vector of m outputs

$\mathbf{C}$ = (m x n) constant coefficient matrix

$\mathbf{D}$ = (m x p) constant coefficient matrix

Equation (2.1.1) is called the state equation and equation (2.1.2) is called the output equation in a state space formulation.

In a number of textbooks [Brogan (1985), Bronson (1973), and Chen (1984)] the solution to a system of first order differential equations with constant coefficients is given by

$$\mathbf{x}_{t+\delta} = e^{\mathbf{A}\delta}\,\mathbf{x}_t + \int_{t}^{t+\delta} e^{\mathbf{A}(t+\delta-\tau)}\,\mathbf{B}\,\mathbf{u}(\tau)\,d\tau \tag{2.1.3}$$

where

$\mathbf{x}_{t+\delta}$ = vector of n state variables at time t+δ

t = discrete point in time

δ = time step

$e^{\mathbf{A}\delta}$ = exponential matrix

$\mathbf{x}_t$ = vector of n state variables at time t

$\mathbf{u}(\tau)$ = vector of p inputs between times t and t+δ

The exponential matrix, $e^{\mathbf{A}\delta}$, is defined by the power series

$$e^{\mathbf{A}\delta} = \mathbf{I} + \mathbf{A}\,\delta + \frac{\mathbf{A}^2\,\delta^2}{2!} + \frac{\mathbf{A}^3\,\delta^3}{3!} + \ldots + \frac{\mathbf{A}^j\,\delta^j}{j!} + \ldots \tag{2.1.4}$$

Appendix A describes a numerically efficient method for computing the exponential matrix.

The first term on the right hand side of equation (2.1.3) is called the complementary function, force-free response or zero-input response. The second term on the right hand side of equation (2.1.3) is called the particular integral, forced response or zero-state response. The zero-input response of a system involves the response of the state variables to the conditions at time t and the zero-state response is the convolution integral which integrates the response of the state variables to the inputs between times t and t+δ. Inputs between times t and t+δ are modeled by a continuous, piecewise linear function and are calculated by

$$\mathbf{u}(\tau) = \mathbf{u}_t + \frac{(\tau - t)}{\delta}\left(\mathbf{u}_{t+\delta} - \mathbf{u}_t\right) \tag{2.1.5}$$

where

$\mathbf{u}_t$ = vector of p inputs at time t

$\mathbf{u}_{t+\delta}$ = vector of p inputs at time t+δ

At this point, the solution of the state equation for heat transfer applications differs from the solution of the state equations in digital control systems, because inputs for digital control systems are not continuous, piecewise linear functions. Åström and Wittenmark (1984) discuss input construction for digital control systems. Substituting equation (2.1.5) into equation (2.1.3) results in

$$\mathbf{x}_{t+\delta} = e^{\mathbf{A}\delta}\,\mathbf{x}_t + \int_t^{t+\delta} e^{\mathbf{A}\,(t+\delta-\tau)}\,\mathbf{B}\left[\,\mathbf{u}_t + \frac{(\tau - t)}{\delta}\left(\mathbf{u}_{t+\delta} - \mathbf{u}_t\right)\right] d\tau \tag{2.1.6}$$

By making the change of variables $\alpha = \tau - t$, Equation (2.1.6) can be rewritten as

$$\mathbf{x}_{t+\delta} = e^{\mathbf{A}\delta}\,\mathbf{x}_t + \left[\int_0^{\delta} e^{\mathbf{A}\,(\delta-\alpha)}\,d\alpha\right]\mathbf{B}\,\mathbf{u}_t + \left[\int_0^{\delta} \alpha\, e^{\mathbf{A}\,(\delta-\alpha)}\,d\alpha\right]\left[\frac{\mathbf{B}}{\delta}\left(\mathbf{u}_{t+\delta} - \mathbf{u}_t\right)\right]$$

(2.1.7)

Appendix B describes the steps for integrating the two integrals in equation (2.1.7). The solution to the first integral is

$$\int_0^{\delta} e^{\mathbf{A}\,(\delta - \alpha)}\, d\alpha = \mathbf{A}^{-1}\left(e^{\mathbf{A}\delta} - \mathbf{I}\right) \tag{2.1.8}$$

and the solution to the second integral is

$$\int_0^{\delta} \alpha\, e^{\mathbf{A}\,(\delta - \alpha)}\, d\alpha = \mathbf{A}^{-1}\,\mathbf{A}^{-1}\left(e^{\mathbf{A}\delta} - \mathbf{I}\right) - \mathbf{A}^{-1}\,\delta \tag{2.1.9}$$

Substituting the solution of the two integrals, equations (2.1.8) and (2.1.9), into equation (2.1.7) yields

$$\mathbf{x}_{t+\delta} = \Phi\, \mathbf{x}_t + \left(\Gamma_1 - \Gamma_2\right)\mathbf{u}_t + \Gamma_2\, \mathbf{u}_{t+\delta} \tag{2.1.10}$$

where

$$\Phi = e^{\mathbf{A}\delta} \tag{2.1.11}$$

$$\Gamma_1 = \mathbf{A}^{-1}\left(e^{\mathbf{A}\delta} - \mathbf{I}\right)\mathbf{B} \tag{2.1.12}$$

$$\Gamma_2 = \left[\,\mathbf{A}^{-1}\,\mathbf{A}^{-1}\left(e^{\mathbf{A}\delta} - \mathbf{I}\right) - \mathbf{A}^{-1}\,\delta\right]\frac{\mathbf{B}}{\delta} = \mathbf{A}^{-1}\left(\frac{\Gamma_1}{\delta} - \mathbf{B}\right) \tag{2.1.13}$$

Equation (2.1.10) relates the states at time t + δ to the states at time t and the inputs at the times t and t + δ.

The forward shift operator, F, [Box and Jenkins (1976)] defined by

$$\mathrm{F}\, v_t = v_{t+\delta} \tag{2.1.14}$$

where

v_t = value of a state or signal at time t

$v_{t+\delta}$ = value of state or signal at time t+δ

will be now be used to relate the states to previous inputs. Using the forward shift operator, equation (2.1.10) can be written as

$$(\mathbf{FI} - \mathbf{\Phi})\, \mathbf{x}_t = \left(\mathrm{F}\Gamma_2 + \Gamma_1 - \Gamma_2 \right) \mathbf{u}_t \tag{2.1.15}$$

Multiplying equation (2.1.15) by the inverse of the (**FI** - **Φ**) matrix gives

$$\mathbf{x}_t = (\mathbf{FI} - \mathbf{\Phi})^{-1} \left(\mathrm{F}\Gamma_2 + \Gamma_1 - \Gamma_2 \right) \mathbf{u}_t \tag{2.1.16}$$

Substituting equation (2.1.16) into equation (2.1.2) yields

$$\mathbf{y}_t = \left[\ \mathbf{C}\, (\mathbf{FI} - \mathbf{\Phi})^{-1} \left(\mathrm{F}\Gamma_2 + \Gamma_1 - \Gamma_2 \right) + \mathbf{D} \ \right] \mathbf{u}_t \tag{2.1.17}$$

Equation (2.1.17) relates the outputs from the system to the inputs.

The inverse of the (**FI** - **Φ**) matrix can be written as

$$(\mathbf{FI} - \mathbf{\Phi})^{-1} = \frac{\mathbf{R}_0 F^{n-1} + \mathbf{R}_1 F^{n-2} + \ldots + \mathbf{R}_{n-2} F + \mathbf{R}_{n-1}}{F^{n} + e_1 F^{n-1} + \ldots + e_{n-1} F + e_n} \tag{2.1.18}$$

The **R** matrices and the e scaler constants can be determined by computing the adjoint of the (**FI** - **Φ**) matrix and dividing by the determinant of the (**FI** - **Φ**) matrix or by using Leverrier's algorithm [Wiberg (1971)] described in appendix C. Substituting equation (2.1.18) into (2.1.17) results in

$$\left(F^{n} + e_1 F^{n-1} + \ldots + e_n\right) \mathbf{y}_t =$$

$$\left[\mathbf{C}\left(\mathbf{R}_0 F^{n-1} + \mathbf{R}_1 F^{n-2} + \ldots + \mathbf{R}_{n-2} F + \mathbf{R}_{n-1}\right)\left(F\Gamma_2 + \Gamma_1 - \Gamma_2\right) + \mathbf{D}\left(F^{n} + e_1 F^{n-1} + \ldots + e_n\right)\right] \mathbf{u}_t \tag{2.1.19}$$

Multiplying the matrices on the right side of equation (2.1.19) and combining common terms of the forward shift operator gives

$$\begin{aligned}\left(F^{n} + e_1 F^{n-1} + \dots + e_n\right) \mathbf{y}_t = \Big\{ & \left(\mathbf{C}\, \mathbf{R}_0 \Gamma_2 + \mathbf{D}\right) F^{n} \\ & + \left[\mathbf{C} \left(\mathbf{R}_1 \Gamma_1 - \mathbf{R}_1 \Gamma_2 + \mathbf{R}_2 \Gamma_2 \right) + e_2 \mathbf{D} \right] F^{n-2} \\ & + \dots \\ & + \left[\mathbf{C} \left(\mathbf{R}_{n-2} \Gamma_1 - \mathbf{R}_{n-2} \Gamma_2 + \mathbf{R}_{n-1} \Gamma_2 \right) + e_{n-1} \mathbf{D} \right] F \\ & + \left[\mathbf{C} \left(\mathbf{R}_{n-1} \Gamma_1 - \mathbf{R}_{n-1} \Gamma_2 \right) + e_n \mathbf{D} \right] \Big\} \mathbf{u}_t \qquad (2.1.20)\end{aligned}$$

Using the definition of the forward shift operator, equation (2.1.20) can be rewritten as

$$\begin{aligned}\mathbf{y}_{t+n\delta} + e_1 \mathbf{y}_{t+(n-1)\delta} + \dots + e_n \mathbf{y}_t = & \left(\mathbf{C}\, \mathbf{R}_0 \Gamma_2 + \mathbf{D} \right) \mathbf{u}_{t+n\delta} \\ & + \left\{ \mathbf{C} \left[\mathbf{R}_0 \left(\Gamma_1 - \Gamma_2 \right) + \mathbf{R}_1 \Gamma_2 \right] + e_1 \mathbf{D} \right\} \mathbf{u}_{t+(n-1)\delta} \\ & + \left\{ \mathbf{C} \left[\mathbf{R}_1 \left(\Gamma_1 - \Gamma_2 \right) + \mathbf{R}_2 \Gamma_2 \right] + e_2 \mathbf{D} \right\} \mathbf{u}_{t+(n-2)\delta} \\ & + \dots \\ & + \left\{ \mathbf{C} \left[\mathbf{R}_{n-2} \left(\Gamma_1 - \Gamma_2 \right) + \mathbf{R}_{n-1} \Gamma_2 \right] + e_{n-1} \mathbf{D} \right\} \mathbf{u}_{t+\delta} \\ & + \left[\mathbf{C}\, \mathbf{R}_{n-1} \left(\Gamma_1 - \Gamma_2 \right) + e_n \mathbf{D} \right] \mathbf{u}_t \qquad (2.1.21)\end{aligned}$$

Shifting the inputs and outputs in equation (2.1.21) n time steps back gives

$$\begin{aligned} y_t + e_1\, y_{t-\delta} + \dots + e_n\, y_{t-n\delta} = & \left(\mathbf{C}\,\mathbf{R}_0\,\Gamma_2 + \mathbf{D}\right)\mathbf{u}_t \\ & + \left\{\mathbf{C}\left[\mathbf{R}_0\left(\Gamma_1 - \Gamma_2\right) + \mathbf{R}_1\,\Gamma_2\right] + e_1\,\mathbf{D}\right\}\mathbf{u}_{t-\delta} \\ & + \left\{\mathbf{C}\left[\mathbf{R}_1\left(\Gamma_1 - \Gamma_2\right) + \mathbf{R}_2\,\Gamma_2\right] + e_2\,\mathbf{D}\right\}\mathbf{u}_{t-2\delta} \\ & + \dots \\ & + \left\{\mathbf{C}\left[\mathbf{R}_{n-2}\left(\Gamma_1 - \Gamma_2\right) + \mathbf{R}_{n-1}\,\Gamma_2\right] + e_{n-1}\,\mathbf{D}\right\}\mathbf{u}_{t-(n-1)\delta} \\ & + \left[\mathbf{C}\,\mathbf{R}_{n-1}\left(\Gamma_1 - \Gamma_2\right) + e_n\,\mathbf{D}\right]\mathbf{u}_{t-n\delta} \qquad (2.1.22) \end{aligned}$$

Equation (2.1.22) can be written more compactly as

$$y_t = \sum_{j=0}^{n}\left(\mathbf{S}_j\,\mathbf{u}_{t-j\delta}\right) - \sum_{j=1}^{n}\left(e_j\,y_{t-j\delta}\right) \qquad (2.1.23)$$

where

$$\mathbf{S}_0 = \mathbf{C}\,\mathbf{R}_0\,\Gamma_2 + \mathbf{D} \qquad (2.1.24)$$

$$\mathbf{S}_j = \mathbf{C}\left[\mathbf{R}_{j-1}\left(\Gamma_1 - \Gamma_2\right) + \mathbf{R}_j\,\Gamma_2\right] + e_j\,\mathbf{D} \quad \text{for } 1 \le j \le n-1 \qquad (2.1.25)$$

$$\mathbf{S}_n = \mathbf{C}\,\mathbf{R}_{n-1}\left(\Gamma_1 - \Gamma_2\right) + e_n\,\mathbf{D} \qquad (2.1.26)$$

Equation (2.1.23) is a transfer function equation which relates current outputs to time series of current and past inputs and time series of past outputs. Ceylan and Myers' derivation results in one additional S coefficient. The transfer function coefficients in equation

equation (2.1.23) may become numerically insignificant as j increases. Thus, the effort of calculating transfer function coefficients can be reduced if only numerically significant coefficients are calculated. The number of significant coefficients can be reduced by using the model reduction methods described in chapter 4.

A large amount of computer memory would be required to store the n nxn **R** matrices if equations (2.1.24) through (2.1.26) were used to calculate transfer functions. Fortunately, the storage requirement for the **R** matrices can be reduced to two nxn matrices if Leverrier's algorithm is combined with equations (2.1.24) through (2.1.26). Section 2.3 contains the steps for computing numerically significant transfer function coefficients with a minimum amount of storage for the **R** matrices.

2.2 TWO NODE EXAMPLE

This section demonstrates the calculation of a transfer function equation for a homogeneous plane wall with constant thermal properties. Heat transfer through the wall is assumed to be one-dimensional. Inputs are inside and outside air temperatures and the heat flux at the interior surface of the wall is the output of interest. The first step in calculating a transfer function equation is to use finite-difference methods [Myers (1971)] to spatially discretize the problem. A two node finite-difference model can be seen in figure 2.1.

Energy balances performed at the two nodes result in the set of first order differential equations

$$C\frac{dT_1}{d\tau} = h\,A\left(T_o - T_1\right) + \frac{T_2 - T_1}{R} \qquad (2.2.1)$$

$$C\frac{dT_2}{d\tau} = h\,A\left(T_i - T_2\right) + \frac{T_1 - T_2}{R} \tag{2.2.2}$$

where

C = thermal capacitance

h = convection coefficient

A = area

R = thermal resistance

T_o = outside temperature

T_i = inside temperature

T_1 = temperature of node 1

T_2 = temperature of node 2

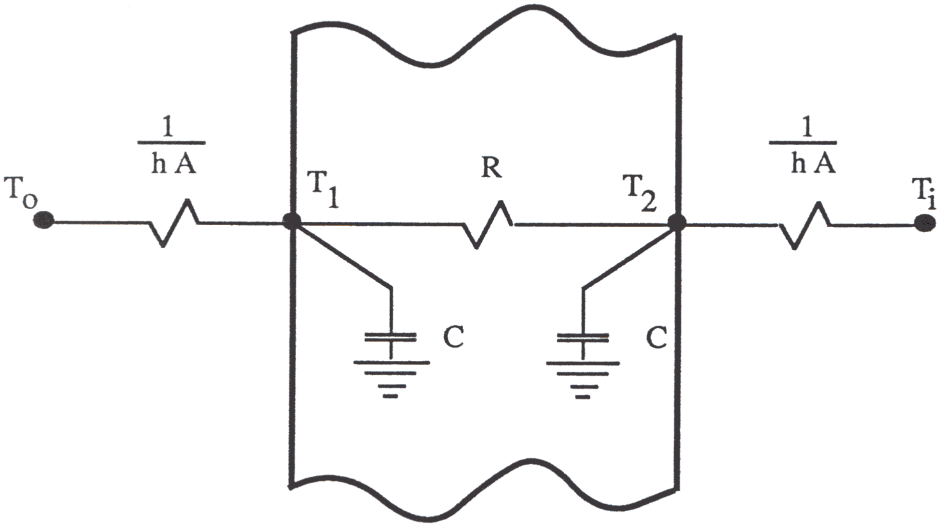


Figure 2.1 Two node finite-difference model of plane wall.

The thermal resistance between the nodes can be calculated from the following equation:

$$R = \frac{L}{k\,A} \tag{2.2.3}$$

where

L = length of wall

k = thermal conductivity of wall

The thermal capacitance of each node can be calculated by

$$C = \frac{\rho\, c\, L\, A}{2} \tag{2.2.4}$$

where

ρ = density

c = specific heat

The heat flux across the interior surface of the wall can be calculated by

$$q'' = h\left(T_2 - T_i\right) \tag{2.2.5}$$

Equations (2.2.1) through (2.2.5) can be formulated in the following state space representation by letting the temperatures of the nodes be the two states:

$$\begin{bmatrix} \dfrac{dT_1}{d\tau} \\[2ex] \dfrac{dT_2}{d\tau} \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{RC} - \dfrac{hA}{C} & \dfrac{1}{RC} \\[2ex] \dfrac{1}{RC} & -\dfrac{1}{RC} - \dfrac{hA}{C} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} + \begin{bmatrix} \dfrac{hA}{C} & 0 \\[2ex] 0 & \dfrac{hA}{C} \end{bmatrix} \begin{bmatrix} T_o \\ T_i \end{bmatrix}$$

$$\left[q''\right] = \begin{bmatrix} 0 & h \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} + \begin{bmatrix} 0 & -h \end{bmatrix} \begin{bmatrix} T_o \\ T_i \end{bmatrix}$$

For a 12" concrete wall with a density of 140 lb_m/ft^3, specific heat of 0.20 Btu/lb_m-°F, thermal conductivity of 1.0 Btu/hr-ft-°F and convection coefficients of 1.46 $Btu/hr\text{-}ft^2$-°F, the matrices in the state space formulation become

$$\mathbf{A} = \begin{bmatrix} -0.1757 & 0.0714 \\ 0.0714 & -0.1757 \end{bmatrix} hr^{-1} \qquad \mathbf{B} = \begin{bmatrix} 0.1043 & 0.0 \\ 0.0 & 0.1043 \end{bmatrix} hr^{-1}$$

$$\mathbf{C} = \begin{bmatrix} 0.0 & 1.46 \end{bmatrix} Btu/hr\text{-}ft^2\text{-}°F \qquad \mathbf{D} = \begin{bmatrix} 0.0 & -1.46 \end{bmatrix} Btu/hr\text{-}ft^2\text{-}°F$$

The exponential matrix Φ can be calculated by the algorithm described in appendix A or by the techniques demonstrated by Brogan (1985) or Chen (1984). The Brogan and Chen methods are easier to use when making hand calculations of an exponential matrix. (When the number of nodes in the finite-difference or finite-element model is small, an interactive matrix package such as Matlab (1982) can be used to compute the exponential matrix and the transfer function coefficients.) For this example, the exponential matrix with a 1 hour time step is

$$\Phi = e^{\mathbf{A}\delta} = \begin{bmatrix} 0.8410 & 0.0600 \\ 0.0600 & 0.8410 \end{bmatrix}$$

Carrying out the matrix manipulations described by equations (2.1.12) and (2.1.13) results in

$$\Gamma_1 = \mathbf{A}^{-1}(\Phi - \mathbf{I})\,\mathbf{B} = \begin{bmatrix} 0.0957 & 0.0033 \\ 0.0033 & 0.0957 \end{bmatrix}$$

$$\Gamma_2 = \mathbf{A}^{-1}\left(\frac{\Gamma_1}{\delta} - \mathbf{B}\right) = \begin{bmatrix} 0.0492 & 0.0011 \\ 0.0011 & 0.0492 \end{bmatrix}$$

The inverse of the (FI - Φ) matrix is computed by dividing the adjoint of the (FI - Φ) matrix by the determinant of the (FI - Φ) matrix.

$$(\mathbf{FI} - \Phi)^{-1} = \begin{bmatrix} F - 0.8410 & -0.0600 \\ -0.0600 & F - 0.8410 \end{bmatrix}^{-1}$$

$$= \frac{\begin{bmatrix} F - 0.8410 & 0.0600 \\ 0.0600 & F - 0.8410 \end{bmatrix}}{(F - 0.8410)\,(F - 0.8410) - 0.0600^2}$$

$$= \frac{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} F + \begin{bmatrix} -0.8410 & 0.0600 \\ 0.0600 & -0.8410 \end{bmatrix}}{F^2 - 1.682F + 0.0704}$$

The constant coefficient matrices and scalar constants in equation (2.1.18) are

$e_1 = -1.682$

$e_2 = 0.0704$

$$\mathbf{R}_0 = \mathbf{I}$$

$$\mathbf{R}_1 = \begin{bmatrix} -0.8410 & 0.0600 \\ 0.0600 & -0.8410 \end{bmatrix}$$

Equations (2.1.24) through (2.1.26) can be used to compute the **S** matrices.

$$\mathbf{S}_0 = \mathbf{C}\,\mathbf{R}_0\,\Gamma_2 + \mathbf{D} = \begin{bmatrix} 0.0017 & -1.3881 \end{bmatrix} \text{ Btu/hr-ft}^2\text{-°F}$$

$$\mathbf{S}_1 = \mathbf{C}\left[\mathbf{R}_0\left(\Gamma_1 - \Gamma_2\right) + \mathbf{R}_1\,\Gamma_2\right] + e_1\,\mathbf{D} = \begin{bmatrix} 0.0061 & 2.4632 \end{bmatrix} \text{ Btu/hr-ft}^2\text{-°F}$$

$$\mathbf{S}_2 = \mathbf{C}\,\mathbf{R}_1\left(\Gamma_1 - \Gamma_2\right) + e_2\,\mathbf{D} = \begin{bmatrix} 0.0014 & -1.0843 \end{bmatrix} \text{ Btu/hr-ft}^2\text{-°F}$$

The transfer function equation for this two node example is

$$\begin{aligned} q''_t &= \mathbf{S}_0\,\mathbf{u}_t + \mathbf{S}_1\,\mathbf{u}_{t-\delta} + \mathbf{S}_2\,\mathbf{u}_{t-2\delta} - e_1\,q''_{t-\delta} - e_2\,q''_{t-2\delta} \\ &= 0.0017\,T_{t,o} + 0.0061\,T_{t-\delta,o} + 0.0014\,T_{t-2\delta,o} \\ &\quad - 1.3881\,T_{t,i} + 2.4632\,T_{t-\delta,i} - 1.0843\,T_{t-2\delta,i} \\ &\quad + 1.682\,q''_{t-\delta} - 0.0704\,q''_{t-2\delta} \end{aligned}$$

2.3 EFFICIENT CALCULATION OF TRANSFER FUNCTIONS

A large amount of computer memory would be required to store the n nxn **R** matrices if equations (2.1.24), (2.1.25), and (2.1.26) were used to compute transfer function coefficients. Fortunately, the storage requirement can be reduced to two nxn matrices if Leverrier's algorithm is combined with the analytical solution. Also, it may not be necessary to calculate all transfer function coefficients because the coefficients may become numerically insignificant as j increases. At some point, the absolute values of the e_j coefficients decrease as j increases. This fact can be used as a criteria to stop calculating transfer function coefficients. The following steps can be used to compute numerically significant coefficients with a minimum amount of computer storage for the **R** matrices (only two nxn **R** matrices need to be stored):

```
1) Compute the exponential matrix $\mathbf{\Phi} = e^{\mathbf{A}\delta}$
2) Use equations (2.1.12) and (2.1.13) to compute $\Gamma_1$ and $\Gamma_2$
3) Use equation (2.1.24) to compute $\mathbf{S}_0$
4) $\mathbf{R}_{new} = \mathbf{I}$
5) For j = 1 to n with a step size of 1
       $\mathbf{R}_{old} = \mathbf{R}_{new}$
       $e_j = -\frac{\mathrm{Trace}\left(\mathbf{\Phi}\,\mathbf{R}_{old}\right)}{j}$
       $\mathbf{R}_{new} = \mathbf{\Phi}\,\mathbf{R}_{old} + e_j\,\mathbf{I}$
       $\mathbf{S}_j = \mathbf{C}\left[\mathbf{R}_{old}\left(\Gamma_1 - \Gamma_2\right) + \mathbf{R}_{new}\,\Gamma_2\right] + e_j\,\mathbf{D}$
        Stop if the absolute value of $e_j$ is less than a tolerance limit
6) $e_n = -\frac{\mathrm{Trace}\left(\mathbf{\Phi}\,\mathbf{R}_{new}\right)}{n}$
```

7) $\mathbf{S}_n = \mathbf{C}\,\mathbf{R}_{new}\left(\Gamma_1 - \Gamma_2\right) + e_n\,\mathbf{D}$

Chapter 4 shows that the number of numerically significant coefficients in a transfer function can be reduced by using model reduction methods.

2.4 RESULTS

The equations and algorithms presented in section 2.2 and appendix A were used to write a 150 line FORTRAN program for calculating transfer functions from a state space formulation. (The program used a library routine in LINPACK (1979) for calculating the inverse of a matrix.) The program was used to compute sets of transfer function coefficients for 2 through 50 node finite-difference models of a 12" homogeneous concrete wall with a density of 140 lb_m/ft^3, specific heat of 0.20 Btu/lb_m-°F, thermal conductivity of 1.0 Btu/hr-ft-°F, and convection coefficients at both sides of the wall of 1.46 Btu/hr-ft^2-°F. Inside and outside air temperatures were inputs to the transfer function equation and the heat flux at the interior surface of the wall was the output. (Section 2.2 contained the steps required to calculate transfer function coefficients for a two node finite-difference model.) Mitalas and Arsenalt's program (1971), which is based upon the solution of a system of partial differential equations, was also used to compute transfer function coefficients. Transfer function coefficients for a 2, 5, and 20 node finite-difference models and the Mitalas and Arsenalt program are listed in table 2.1. As the number of nodes in the finite-difference model increases, the transfer function coefficients from the state space formulation approach those of Mitalas and Arsenalt which are based upon the solution of the partial differential equation (i.e., the continuous model).

Table 2.1 Transfer function coefficients for a 12" concrete wall and 2, 5, and 20 node finite-difference models of the concrete wall.

Transfer Function Coefficient	Partial Differential Equation	2 node	5 node	20 node
S_0 Out. Temp. Btu/hr-ft^2-°F	0.0000	0.0017	0.0001	0.0000
S_1 Out. Temp. Btu/hr-ft^2-°F	0.0010	0.0061	0.0019	0.0010
S_2 Out. Temp. Btu/hr-ft^2-°F	0.0053	0.0014	0.0044	0.0053
S_3 Out. Temp. Btu/hr-ft^2-°F	0.0040		0.0017	0.0039
S_4 Out. Temp. Btu/hr-ft^2-°F	0.0005		0.0001	0.0005
S_0 Ins. Temp. Btu/hr-ft^2-°F	-1.2045	-1.3881	-1.2597	-1.2073
S_1 Ins. Temp. Btu/hr-ft^2-°F	2.2677	2.4632	2.6960	2.2820
S_2 Ins. Temp. Btu/hr-ft^2-°F	-1.2929	-1.0843	-1.9575	-1.3117
S_3 Ins. Temp. Btu/hr-ft^2-°F	0.2278		0.5808	0.2361
S_4 Ins. Temp. Btu/hr-ft^2-°F	-0.0088		-0.0707	-0.0097
S_5 Ins. Temp. Btu/hr-ft^2-°F	0.0001		0.0029	0.0001
e_1	-1.7442	-1.6820	-1.9782	-1.7502
e_2	0.9050	0.7034	1.3116	0.9147
e_3	-0.1395		-0.3509	-0.1437
e_4	0.0041		0.0383	0.0045
e_5	0.0000		-0.0014	0.0000

To compare the transfer functions, heat fluxes were computed when the air temperature on one side of the wall varied with the periodic temperature profile

$$T = 5^\circ\text{F} + 5^\circ\text{F} \sin[(\tau\ \pi)/24 \text{ hours}]$$

and the air temperature on the other side of the wall was 0°F. Figure 2.2 (outside air temperature varying) and figure 2.3 (inside air temperature varying) show the heat flux at the interior surface of the wall for transfer functions based upon two and five node finite-difference models and the continuous model. Table 2.2 contains the sum of squares of the residuals (SSQ) between the calculated heat flux for the finite-difference models and the continuous model for a 24 hour period. As the number of nodes in the finite-difference model increases the SSQ decreases.

Table 2.3 contains the central processing unit (CPU) time of a MicroVax computer to compute all transfer function coefficients, numerically significant coefficients, and the exponential matrix for different numbers of nodes. (The tolerance limit in section 2.5 for the calculation of numerically significant coefficient was 0.000001.) Table 2.3 demonstrates the importance of computing only numerically significant coefficients. Also, a majority of the effort in calculating numerical significant coefficients involves the calculation of the exponential matrix.

Table 2.2 Sum of squares of the residuals between continuous model and finite-difference model for a 24 hour period.

	Sum of Square of Residuals Btu^2/hr^2-ft^4	
Number of Nodes	Outside Temp. Varying	Inside Temp. Varying
2	0.10	20.0
5	0.0017	0.16
10	0.000070	0.0060
15	0.000012	0.0010
20	0.0000035	0.00030

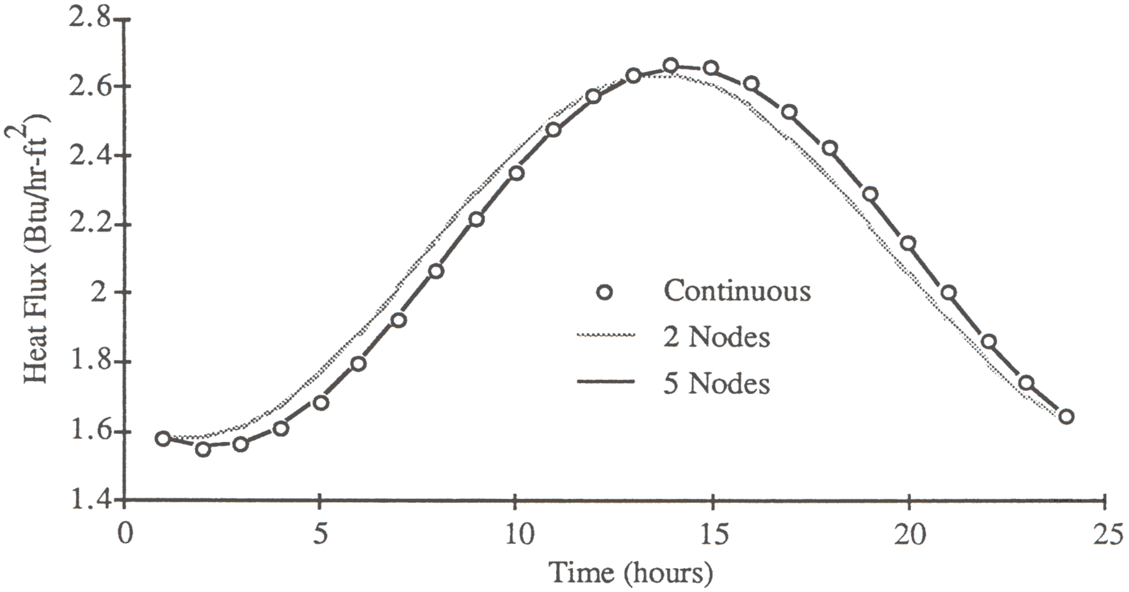


Figure 2.2 Heat flux at interior surface of concrete wall with outside air temperature varying for two and five node finite-difference models and the continuous model.

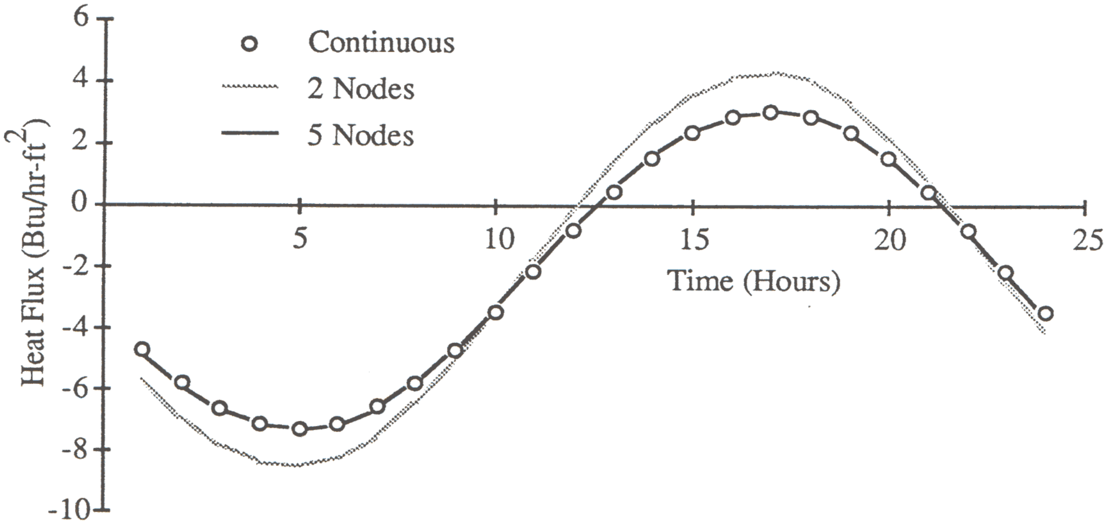


Figure 2.3 Heat flux at interior surface of concrete wall with inside air temperature varying for two and five node finite-difference models and the continuous model.

Table 2.3 CPU time to compute the exponential matrix, all transfer function coefficients, and all numerically significant transfer function coefficients.

	Central Processing Unit Time in Seconds		
Number of Nodes	Exponential Matrix	All Coefficients	Numerically Significant Coefficients
10	0.25	0.54	0.32
20	1.8	4.7	2.1
30	6.1	19.1	6.8
40	15.0	54.5	16.3
50	30.6	123.2	32.9

2.5 APPLICATION

Building simulation programs such as DOE-2 (1980), TRNSYS (1983), and TARP (1983) assume one-dimensional heat transfer through walls, roofs, and floors. Many walls of common construction cannot be accurately modeled with one-dimensional heat transfer. The equations and algorithms described in this paper can be used to calculate transfer functions for walls which require two-dimensional models, e.g., walls which contain metal tee-bars or tie rods. The ASHRAE handbook of fundamentals (1977, 1981, and 1985) lists tables of transfer functions for one-dimensional heat transfer through multilayered slabs. These tables could be updated to include walls which have multi-dimensional heat transfer. An example demonstrating the importance of properly modeling a roof with a steel bulb tees follows.

Transfer function coefficients were generated for one-dimensional and two-dimensional models of a roof deck section taken from an example in ASHRAE (1985).

Figures 2.4 and 2.5 show details of the geometry, inputs, and outputs for a section of the roof. Transfer function coefficients for the one-dimensional model were determined for a multi-layered roof with the same area weighted thermal physical properties as the roof deck. Figure 2.6 shows the nodal spacing for the two-dimensional finite-difference model. Table 2.4 contains the thermal physical properties of the materials in the roof. The outside convection coefficient is 6.0 Btu/hr-ft^2-°F and the inside convection coefficient is 1.63 Btu/hr-ft^2-°F. Table 2.5 contains the transfer function coefficients for the one-dimensional model and the two-dimensional model of the roof deck.

Figure 2.7 shows the response to a 1°F step change in outdoor temperature with an indoor temperature equal to 0°F for one and two-dimensional models. The steady-state and transient response for the two models is significantly different. This demonstrates the importance of properly modeling the steel bulb tees in the roof deck.

Table 2.4 Thermal properties of the materials in roof deck.

Material	Thermal Conductivity Btu/hr-ft-°F	Density lbm/ft^3	Specific Heat Btu/lbm-°F
Roofing	0.094	70	0.35
Gypsum Concrete	0.14	51	0.21
Steel	26.0	490	0.12
Glass Fiber	0.021	5	0.23

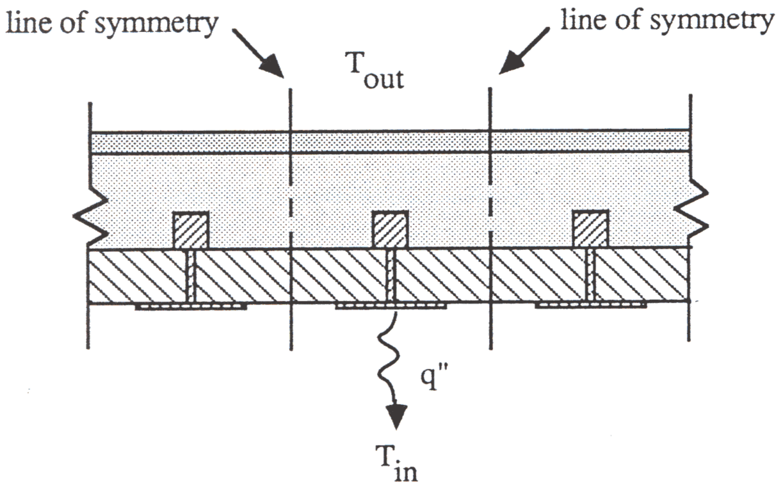


Figure 2.4 Roof deck.

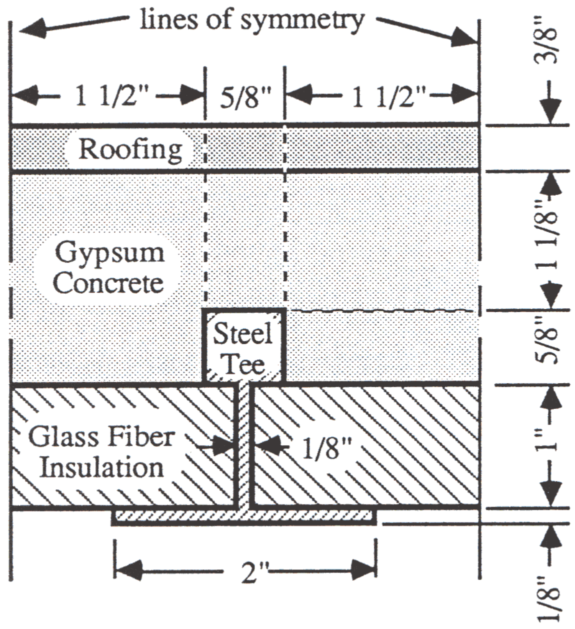


Figure 2.5 Enlarged section of roof deck.

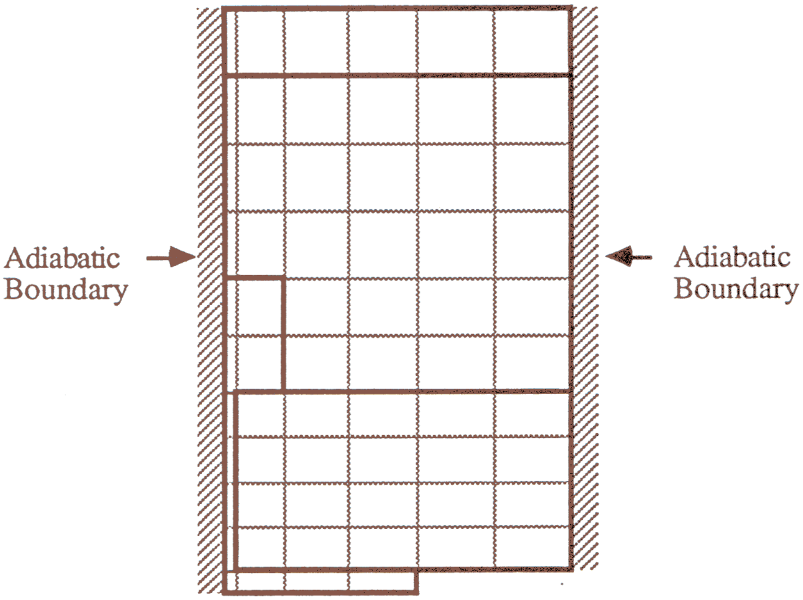


Figure 2.6 Nodal spacing of two-dimensional finite-difference model.

Table 2.5 Transfer function coefficients for one-dimensional and two-dimensional models of roof deck.

	One-dimensional			Two-dimensional		
	Outside Temp. S_j	Inside Temp. S_j		Outside Temp. S_j	Inside Temp. S_j	
j	Btu/hr-ft^2-°F	Btu/hr-ft^2-°F	e_j	Btu/hr-ft^2-°F	Btu/hr-ft^2-°F	e_j
0	0.00895	-1.10605		0.05312	-0.80438	
1	0.20642	0.80496	-0.40554	0.15092	0.62278	-0.46860
2	0.02019	-0.15043	0.00432	0.02194	-0.04532	0.01943
3	0.00002	0.00002	0.00000	0.00002	0.00037	-0.00001

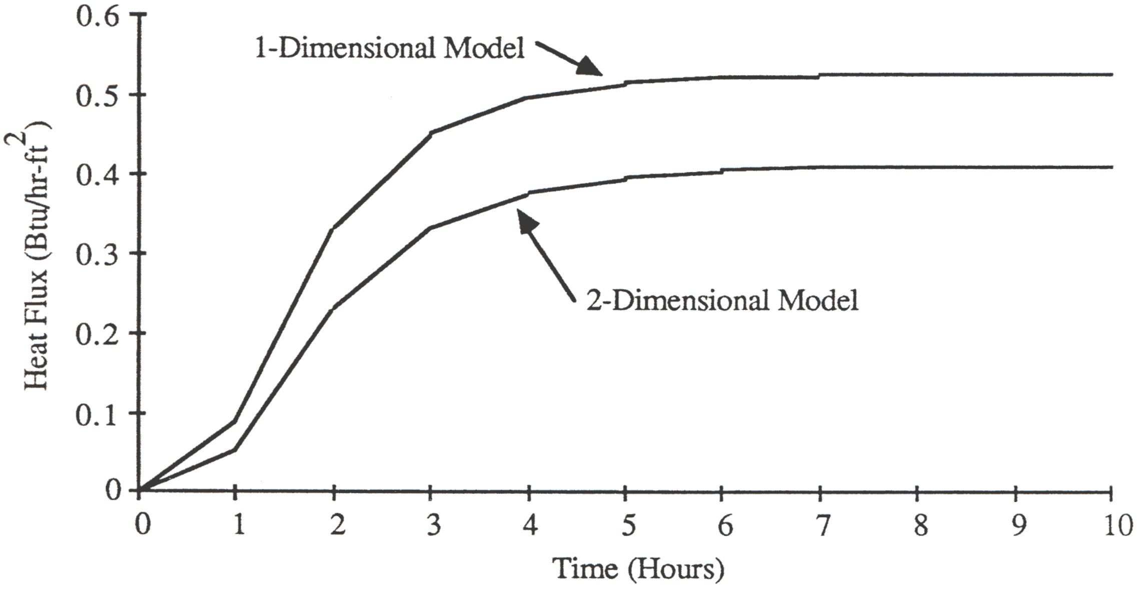


Figure 2.7 Response to 1 °F step change in outdoor temperature for the roof deck.

2.6 SUMMARY

A method for determining transfer functions from the exact solution of a system of first-order differential equations has been presented in this chapter. This method can be used to determine transfer functions for multi-dimensional heat transfer. The number of significant coefficients in transfer functions can be reduced by using the model reduction methods presented in chapter 4.

This chapter has shown that multi-dimensional effects are important when modeling the heat transfer processes in a roof deck which contains steel bulb tees. There are a number of areas of application, besides roofs, where transfer functions for multi-dimensional heat transfer could be used. For example, transfer functions could be developed to model multi-dimensional heat transfer processes in an attic or a basement.

CHAPTER 3
Comprehensive Room Transfer Functions

This chapter describes a method in which the transfer functions describing heat flows in building elements can be combined into a Comprehensive Room Transfer Function (CRTF) for an enclosure. Section 3.1 presents the derivation of the equations for combining transfer functions with parallel heat transfer paths. Section 3.2 is devoted to a method for accurately modeling the convection and radiation heat transfer processes in an enclosure by a star network. A star network allows individual transfer functions for building elements to be easily combined for rooms. Section 3.3 presents equations for combining transfer functions for a room model based on a star network. Section 3.4 compares the computational effort of energy balance simulations with CRTF simulations for two rooms.

3.1 TRANSFER FUNCTION COMBINATION FOR PARALLEL PATHS

Many walls of common construction have parallel heat flow paths. Wilkes (1983) compared heat transfer through a stud-frame wall and an uninsulated concrete wall for a two-dimensional model and a parallel path one-dimensional model. There was less than a three percent difference between the two models. Equations for combining transfer functions for walls with parallel heat transfer paths, as shown in figure 3.1, are derived in this section.

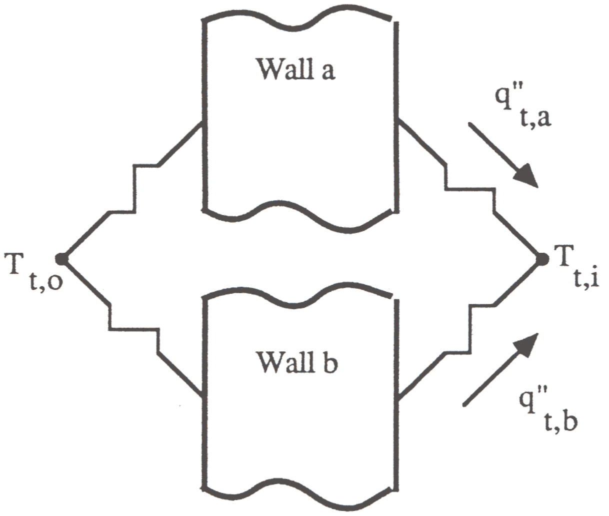


Figure 3.1 Walls with parallel heat transfer paths.

The transfer function equation for the heat flux at the interior surface of wall a is

$$q''_{t,a} = \sum_{j=0} \left(a_{j,a}\, T_{t-j\delta,o} + b_{j,a}\, T_{t-j\delta,i} \right) - \sum_{j=1} \left(c_{j,a}\, q''_{t-j\delta,a} \right) \qquad (3.1.1)$$

where

$a_{j,a}$ = outdoor temperature transfer function coefficient for wall a

$b_{j,a}$ = indoor temperature transfer function coefficient for wall a

$c_{j,a}$ = transfer function coefficient for past heat fluxes for wall a

The upper limits on the summations in equation (3.1.1) are dependent upon the method used to obtain the transfer function coefficients, the wall material properties, and the time step.

Equation (3.1.1) can be expressed in terms of the backshift operator, B, as

$$\left(\sum_{j=0} c_{j,a}\, B^{j}\right) q''_{t,a} = \left(\sum_{j=0} a_{j,a}\, B^{j}\right) T_{t,o} + \left(\sum_{j=0} b_{j,a}\, B^{j}\right) T_{t,i} \tag{3.1.2}$$

where $c_{0,a} = 1.0$. The backshift operator [Box and Jenkins (1976)] is defined as

$$B\, v_t = v_{t-\delta} \tag{3.1.3}$$

where

v_t = value of a state or signal at time t

$v_{t-\delta}$ = value of a state or signal at time t-δ

Dividing by the summation on the left side of equation (3.1.2) results in

$$q''_{t,a} = \frac{\left(\sum_{j=0} a_{j,a}\, B^{j}\right) T_{t,o} + \left(\sum_{j=0} b_{j,a}\, B^{j}\right) T_{t,i}}{\left(\sum_{j=0} c_{j,a}\, B^{j}\right)} \tag{3.1.4}$$

A similar equation for wall b is

$$q''_{t,b} = \frac{\left(\sum_{j=0} a_{j,b}\, B^{j}\right) T_{t,o} + \left(\sum_{j=0} b_{j,b}\, B^{j}\right) T_{t,i}}{\left(\sum_{j=0} c_{j,b}\, B^{j}\right)} \tag{3.1.5}$$

The heat flux for walls a and b combined, $q''_{t,\chi}$, is

$$q''_{t,\chi} = \frac{A_a}{A_a + A_b} q''_{t,a} + \frac{A_b}{A_a + A_b} q''_{t,b} = f_a q''_{t,a} + f_b q''_{t,b} \tag{3.1.6}$$

where

A_a = area of wall a

A_b = area of wall b

f_a = area fraction of wall a

f_b = area fraction of wall b

Substituting equations (3.1.4) and (3.1.5) into equation (3.1.6) gives

$$\begin{aligned} q''_{t,\chi} = f_a &\left[\frac{\left(\sum\limits_{j=0} a_{j,a} B^j\right) T_{t,o} + \left(\sum\limits_{j=0} b_{j,a} B^j\right) T_{t,i}}{\left(\sum\limits_{j=0} c_{j,a} B^j\right)} \right] \\ + f_b &\left[\frac{\left(\sum\limits_{j=0} a_{j,b} B^j\right) T_{t,o} + \left(\sum\limits_{j=0} b_{j,b} B^j\right) T_{t,i}}{\left(\sum\limits_{j=0} c_{j,b} B^j\right)} \right] \end{aligned} \tag{3.1.7}$$

Multiplying by the denominators of the terms on the right hand side of equation (3.1.7) results in

$$\left(\sum_{j=0} c_{j,a} B^{j}\right)\left(\sum_{j=0} c_{j,b} B^{j}\right) q''_{t,\chi} = \left[f_a\left(\sum_{j=0} a_{j,a} B^{j}\right)\left(\sum_{j=0} c_{j,b} B^{j}\right)\right] T_{t,o}$$

$$+\left[f_b\left(\sum_{j=0} a_{j,b} B^{j}\right)\left(\sum_{j=0} c_{j,a} B^{j}\right)\right] T_{t,o}$$

$$+\left[f_a\left(\sum_{j=0} b_{j,a} B^{j}\right)\left(\sum_{j=0} c_{j,b} B^{j}\right)\right] T_{t,i}$$

$$+\left[f_b\left(\sum_{j=0} b_{j,b} B^{j}\right)\left(\sum_{j=0} c_{j,a} B^{j}\right)\right] T_{t,i} \qquad (3.1.8)$$

Carrying out the algebra and combining common powers of the backshift operator yields

$$\left(\sum_{j=0} c_{j,\chi} B^{j}\right) q''_{t,\chi} = \left(\sum_{j=0} a_{j,\chi} B^{j}\right) T_{t,o} + \left(\sum_{j=0} b_{j,\chi} B^{j}\right) T_{t,i} \qquad (3.1.9)$$

where

$$a_{j,\chi} = \sum_{k=0}^{j}\left(f_a\, a_{k,a}\, c_{j-k,b} + f_b\, a_{k,b}\, c_{j-k,a}\right) \qquad (3.1.10)$$

$$b_{j,\chi} = \sum_{k=0}^{j}\left(f_a\, b_{k,a}\, c_{j-k,b} + f_b\, b_{k,b}\, c_{j-k,a}\right) \qquad (3.1.11)$$

$$c_{j,\chi} = \sum_{k=0}^{j}\left(c_{k,a}\, c_{j-k,b}\right) \qquad (3.1.12)$$

Using the definition of the backshift operator, equation (3.1.9) can be rewritten in a form that looks exactly like equation (3.1.1).

$$q''_{t,\chi} = \sum_{j=0} \left(a_{j,\chi}\, T_{t-j\delta,o} + b_{j,\chi}\, T_{t-j\delta,i} \right) - \sum_{j=1} \left(c_{j,\chi}\, q''_{t-j\delta,\chi} \right) \tag{3.1.13}$$

Equations (3.1.10), (3.1.11), and (3.1.12) define the transfer function coefficients in equation (3.1.13). The number of previous time steps in the combined transfer function equation is equal to the summation of the number of previous time steps for the individual transfer functions, i.e., the number of previous time steps in the transfer function for wall a plus the number of previous time steps in the transfer function for wall b. The number of previous time steps in the combined transfer function can be reduced by using the model reduction methods in chapter 4 to obtain an approximate transfer function which accurately models the heat flow from both walls.

If transfer functions are combined for two identical walls, the combined transfer function will have twice as many past steps as the transfer function for the walls. Appendix D shows that the number of past time steps in the combined transfer function can be reduced back to the number of past time steps for the individual transfer function.

3.2 STAR NETWORK

Direct combination of individual wall transfer functions into a CRTF when view factors are used to model long-wave radiation in the room requires the manipulation of polynomial matrices [Chen (1984)], an area of research in systems analysis. Approximating the actual radiation and convection heat transfer processes in a room by a star network allows for easy combination (i.e., no manipulation of polynomial matrices) of transfer functions. Figure 3.2 shows a star network for an enclosure with three surfaces. This section presents a computationally easy method for determining the resistances in a star network from a network which uses wall-to-wall view factors to model long-wave radiation exchange, i.e., a view factor network.

Figure 3.3 shows a view factor network for an enclosure with three surfaces. The resistance to long-wave radiation exchange between surfaces in the view factor network is

$$R_{i\text{-}j,rad} = \frac{1}{\varepsilon_i A_i G_{i\text{-}j} \sigma 4 \overline{T}^3} \tag{3.2.1}$$

where

ε_i = emmitance of surface i

$G_{i\text{-}j}$ = absorption factor between surfaces i and j

σ = Stefan-Bolzmann constant

$\overline{T}$ = average temperature of surfaces in room

The absorption factor [Gebhart (1971)], $G_{i\text{-}j}$, is the fraction of energy emitted by surface i which is absorbed by surface j.

Section 3.2.1 presents a method for determining the resistances in the star network from the resistances in the view factor network. Section 3.2.2 shows that this method results in an exact transformation for rooms with two surfaces and for rooms which have the same resistance to long-wave radiation heat transfer and the same resistance to convective heat transfer for all surfaces. In Section 3.2.3, the star network is compared with the view factor network for rooms in which an exact transformation does not exist.

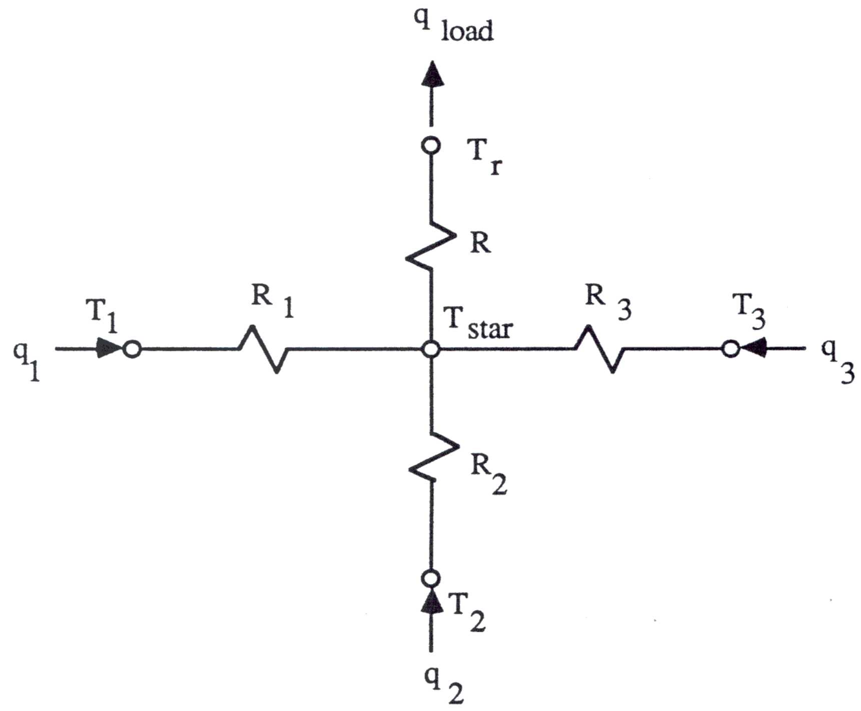
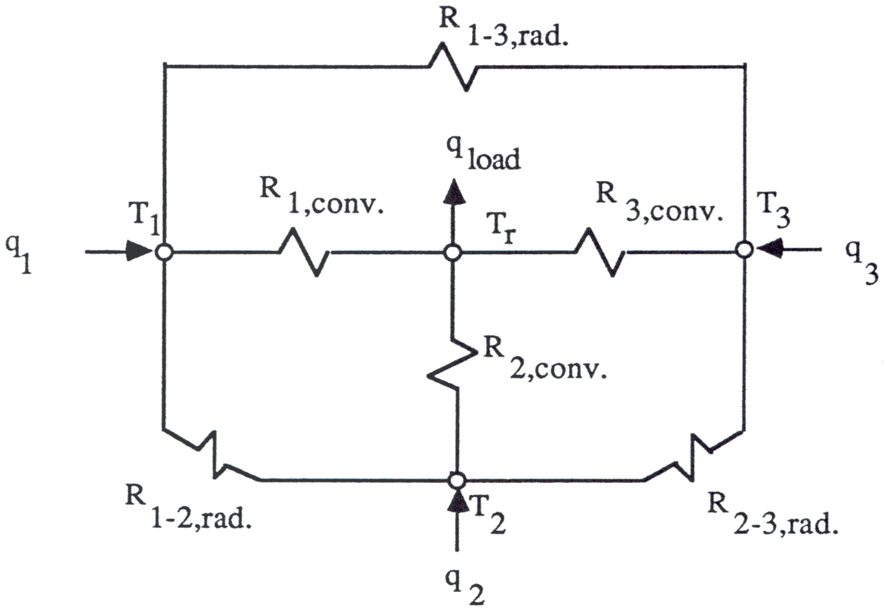


Figure 3.2 Star network for a room with three surfaces.



3.2.1 Formulation

Two main steps are involved in determining the resistances in the star network from the resistances in the view factor network. First, the resistance between each pair of nodes in the view factor network is determined when all other nodes are floating. A floating node is defined as one in which heat transfer occurs only by convection to the air or by long-wave radiation exchange with other surfaces in the enclosure. As a result, conduction through building elements, solar radiation gains, and radiation gains from people, equipment, and lights do not affect floating surface nodes, and infiltration or convection gains from people, equipment, and lights do not affect the floating room air node. (If surface node i is floating, then q_i is 0. If the room air node is floating, then q_{load} is 0.) Second, an approximation is used to determine the resistances in the star network from the resistances between nodes in the view factor network.

To compute the resistances between nodes in the view factor network, an energy balance must be performed on each surface in the enclosure and on the air in the enclosure. An energy balance for surface i in an enclosure with N surfaces is

$$\frac{\left(T_1 - T_i\right)}{R_{1\text{-}i,rad}} + \frac{\left(T_2 - T_i\right)}{R_{2\text{-}i,rad}} + \ldots + \frac{\left(T_N - T_i\right)}{R_{N\text{-}i,rad}} + \frac{\left(T_r - T_i\right)}{R_{i,conv}} + q_i = 0 \qquad (3.2.2)$$

where

$R_{i\text{-}j,rad}$ = resistance to long-wave radiation between surfaces i and j

$R_{i,conv}$ = resistance to convective heat transfer between surface i and room air

q_i = energy input to surface i other than by convection with the room air or long-wave radiation exchange with other surfaces in the room

An energy balance for the air in the room is

$$\frac{\left(T_1 - T_r\right)}{R_{1,conv}} + \frac{\left(T_2 - T_r\right)}{R_{2,conv}} + \ldots + \frac{\left(T_N - T_r\right)}{R_{N,conv}} = q_{load} \qquad (3.2.3)$$

where

q_{load} = heating or cooling load

The N energy balances for every surface in the room and the energy balance on the room air can be combined into the matrix equations

$$\mathbf{XY = Z} \qquad (3.2.4)$$

where

(For i = 1 to N and j = 1 to N)

$$x_{(i,i)} = -\left(\sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{1}{R_{i\text{-}j,rad}}\right) - \frac{1}{R_{i,conv}}$$

$$x_{(i,j)} = x_{(j,i)} = \frac{1}{R_{i\text{-}j,rad}}$$

$$x_{(i,N+1)} = 0$$

$$x_{(N+1,i)} = \frac{1}{R_{i,conv}}$$

$$x_{(N+1,N+1)} = -1$$

$$y_{(i)} = T_i - T_r$$

$$y_{(N+1)} = q_{load}$$

$$z_{(i)} = -q_i$$

$$z_{(N+1)} = 0$$

To compute the resistance, R_{i-j}, between surfaces i and j when all other nodes are floating, arbitrarily let q_i be unity so that

$$q_i = -q_j = 1.0 \text{ [W or Btu/hr]} = \frac{\left(T_i - T_j\right)}{R_{i-j}} \tag{3.2.5}$$

then

$$\begin{aligned} R_{i-j} &= \left(T_i - T_r\right) - \left(T_j - T_r\right) \\ &= y_{(i)} - y_{(j)} \\ &= x_{(i,j),inv} + x_{(j,i),inv} - x_{(i,i),inv} - x_{(j,j),inv} \end{aligned} \tag{3.2.6}$$

where

$x_{(i,j),inv}$ = element in row i and column j of the inverse of the **X** matrix

To compute the resistance, R_{i-r}, between surface i and the room air let

$$q_i = q_{load} = \frac{\left(T_i - T_r\right)}{R_{i-r}} = 1.0 \text{ [W or Btu/hr]} \tag{3.2.7}$$

then

$$\begin{aligned} R_{i-r} &= T_i - T_r \\ &= - x_{(i,i),inv} \end{aligned} \tag{3.2.8}$$

A number of approximations could be used to obtain the unknown resistances in the star network from the resistances between nodes in the view factor network. For example, nonlinear regression could be used to minimize the error in heat flow between nodes, or linear regression could be used to minimize the resistance to heat transfer between nodes. An approximation that accurately models the heat processes in a room and requires less computational effort than linear or nonlinear regression is as follows.

The net heat flow to the air (i.e., load) for steady-state temperature differences between enclosure surfaces and the air will be the same for the star network and the view factor network if the following N equations are satisfied

$$R_1 + R = R_{1\text{-}r}$$

$$R_2 + R = R_{2\text{-}r}$$

$$\cdot$$
$$\cdot$$
$$\cdot$$

$$R_N + R = R_{N\text{-}r} \tag{3.2.9}$$

One more equation is needed to provide N+1 equations with N+1 unknowns. The last equation is selected so that the heat transfer between surfaces for the star network approximates the heat transfer between surfaces in the view factor network. The difference in resistance to heat transfer between surface nodes i and j when other nodes are floating for the view factor network and the star network is

$$R_i + R_j - R_{i\text{-}j} = R_{i\text{-}r} + R_{j\text{-}r} - R_{i\text{-}j} - 2R \tag{3.2.10}$$

Squaring a dimensionless form of this error in resistance between all surface nodes gives the function

$$\psi_1 = \sum_{i=2}^{N} \sum_{j=1}^{i-1} \frac{\left(R_{i-r} + R_{j-r} - R_{i-j} - 2R\right)^2}{R_{i-j}^2} \qquad (3.2.11)$$

Surfaces with a lower resistance (R_{i-j}) between them have a larger heat transfer for the same temperature difference. The following error function will place more weight on resistances between surfaces with a smaller R_{i-j}:

$$\psi_2 = \sum_{i=2}^{N} \sum_{j=1}^{i-1} \frac{\left(R_{i-r} + R_{j-r} - R_{i-j} - 2R\right)^2}{R_{i-j}^3} \qquad (3.2.12)$$

Other weighting functions could be used to obtain ψ_2, but as will be shown, this weighting function results in accurate modeling of the heat transfer processes for rooms with a wide variety of thermal physical properties, resistances to long-wave radiation exchange, and resistances to convective heat transfer. The derivative of ψ_2 with respect to the resistance R between the star node and the air is

$$\frac{d(\psi_2)}{dR} = \sum_{i=2}^{N} \sum_{j=1}^{i-1} \frac{-4\left(R_{i-r} + R_{j-r} - R_{i-j} - 2R\right)}{R_{i-j}^3} \qquad (3.2.13)$$

Setting the derivative ψ_2 with respect to R equal to zero gives

$$R = \frac{\displaystyle\sum_{j=2}^{N}\sum_{i=1}^{j-1}\frac{R_{i-r}+R_{j-r}-R_{i-j}}{R_{i-j}^{3}}}{\displaystyle 2\sum_{j=2}^{N}\sum_{i=1}^{j-1}\frac{1}{R_{i-j}^{3}}} \qquad (3.2.14)$$

The second derivative of ψ_2 with respect to R will be positive for all positive values of R. Therefore, ψ_2 will be a minimum when R is positive. The other N unknown resistances in the star network can be obtained from equation (3.2.9) after equation (3.2.14) is used to compute the resistance between the star node and the room air node.

The following step-by-step procedure can be used to compute the resistances in the star network from the resistances in the view factor network:

1) Use equations (3.2.4), (3.2.6), and (3.2.8) to determine the resistances between nodes in the view factor network when other nodes are floating.

2) Use equation (3.2.14) to determine the resistance between the star node and the air node.

3) Use equation (3.2.9) to determine the resistances between the surface nodes and the star node.

3.2.2 Exact Transformations

This section shows that the method described in section 3.2.1 results in an exact transformation for a room with two surfaces and for rooms which have the same resistances to long-wave radiation exchange and resistances to convective heat transfer for all surfaces. Figure 3.4 is a schematic of the heat flows for a two surface room which uses view factors to model long-wave radiation exchange. The resistances between nodes for the two surface room when one node is floating are

$$R_{1\text{-}2} = \frac{R_{1\text{-}2,rad}\left(R_{1,conv} + R_{2,conv}\right)}{R_{1\text{-}2,rad} + R_{1,conv} + R_{2,conv}} \tag{3.2.15}$$

$$R_{1\text{-}r} = \frac{R_{1,conv}\left(R_{1\text{-}2,rad} + R_{2,conv}\right)}{R_{1\text{-}2,rad} + R_{1,conv} + R_{2,conv}} \tag{3.2.16}$$

$$R_{2\text{-}r} = \frac{R_{2,conv}\left(R_{1\text{-}2,rad} + R_{1,conv}\right)}{R_{1\text{-}2,rad} + R_{1,conv} + R_{2,conv}} \tag{3.2.17}$$

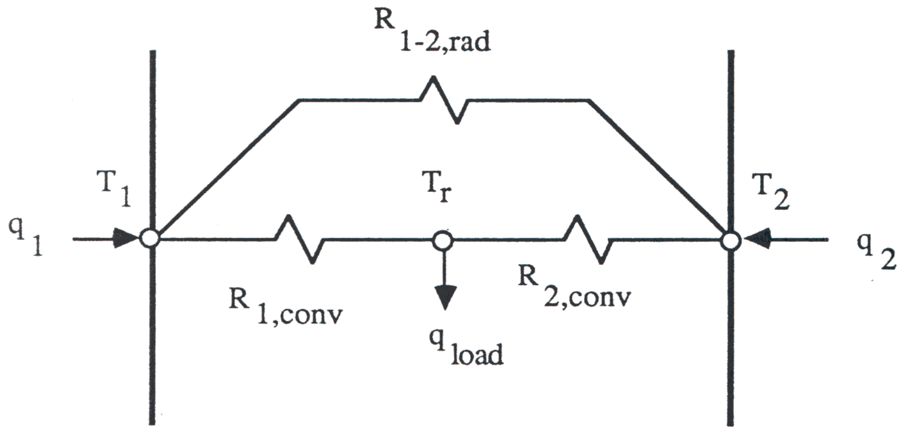


Figure 3.4 Heat flows for a two surface room which uses view factors to model long-wave radiation exchange.

Using equation (3.2.14) for a room with two surfaces, i.e., N = 2, gives the following equation for the resistance R between the star node and the air node:

$$R = \frac{R_{1\text{-}r} + R_{2\text{-}r} - R_{1\text{-}2}}{2}$$

$$= \frac{R_{1,conv}\, R_{2,conv}}{R_{1\text{-}2,rad} + R_{1,conv} + R_{2,conv}} \qquad (3.2.18)$$

Equation (3.2.9) can be used to compute the resistances between the surfaces and the star node, i.e., R_1 and R_2.

$$R_1 = R_{1\text{-}r} - R = \frac{R_{1,conv}\, R_{1\text{-}2,rad}}{R_{1\text{-}2,rad} + R_{1,conv} + R_{2,conv}} \qquad (3.2.19)$$

$$R_2 = R_{2\text{-}r} - R = \frac{R_{2,conv}\, R_{1\text{-}2,rad}}{R_{1\text{-}2,rad} + R_{1,conv} + R_{2,conv}} \qquad (3.2.20)$$

The resistances R_1, R_2, and R_3 obtained by this method are identical to those that would be found if a Y-Δ transformation [Hayt and Kemmerly (1978)] would be performed. Thus, the star network provides an exact representation of the heat transfer processes for a two surface room.

Now it will be shown that the star network provides an exact representation of the heat flows for rooms which have the same resistances to long-wave radiation exchange and resistances to convective heat transfer for all surfaces. All floating resistances, $R_{i\text{-}r}$, between each surface and the room air are equal for a room with the same resistance to convective heat transfer for all surfaces and the same resistance to long-wave radiation exchange between all surfaces. Thus, the N equations in equation (3.2.9) are

$$R_i + R = R_{i\text{-}r} \qquad (3.2.21)$$

for a room with the same resistance for convective heat transfer for all surfaces and resistance to long-wave radiation exchange between all surfaces. Using equation (3.2.14), the resistance between the star node and the room air is

$$R = R_{i-r} - \frac{R_{i-j}}{2} \tag{3.2.22}$$

The difference between resistance to heat transfer for the actual network and the star network when other nodes are floating is

$$R_i + R_j - R_{i-j} = 2R_i - R_{i-j} \tag{3.2.23}$$

Substituting equations (3.2.21) and (3.2.22) into equation (3.2.23) gives

$$R_i + R_j - R_{i-j} = 2\left(R_{i-r} - R\right) - 2R_{i-r} + 2R = 0 \tag{3.2.24}$$

Thus, the star network provides an exact result for the heat flow calculations for a room with the same resistance to convective heat transfer for all surfaces and the same resistance to radiation heat transfer between the walls.

3.2.3 Example for an Enclosure with Three Surfaces

This section contains the development of a star network for an enclosure in which the heat flows are approximate, i.e., heat flows in the star network are different than the heat flows in a view factor network. To keep the example simple, an enclosure with three surfaces is used.

Assume, the resistances in the view factor network are

$$R_{1\text{-}2,rad} = 3.44 \text{ (hr-°F)/Btu}$$

$$R_{1\text{-}3,rad} = R_{2\text{-}3,rad} = 0.0344 \text{ (hr-°F)/Btu}$$

$$R_{1,conv} = R_{2,conv} = 0.0625 \text{ (hr-°F)/Btu}$$

$$R_{3,conv} = 0.0625 \text{ (hr-°F)/Btu}$$

The elements in the **X** matrix can be computed from equation (3.2.4)

$$x_{(1,1)} = -\frac{1}{R_{1\text{-}2,rad}} - \frac{1}{R_{1\text{-}3,rad}} - \frac{1}{R_{1,conv}} = -45.36 \text{ Btu/(hr-°F)}$$

$$x_{(2,2)} = -\frac{1}{R_{1\text{-}2,rad}} - \frac{1}{R_{2\text{-}3,rad}} - \frac{1}{R_{2,conv}} = -45.36 \text{ Btu/(hr-°F)}$$

$$x_{(3,3)} = -\frac{1}{R_{1\text{-}3,rad}} - \frac{1}{R_{2\text{-}3,rad}} - \frac{1}{R_{3,conv}} = -218.14 \text{ Btu/(hr-°F)}$$

$$x_{(1,2)} = x_{(2,1)} = \frac{1}{R_{1\text{-}2,rad}} = 0.2907 \text{ Btu/(hr-°F)}$$

$$x_{(1,3)} = x_{(3,1)} = \frac{1}{R_{1\text{-}3,rad}} = 29.07 \text{ Btu/(hr-°F)}$$

$$x_{(2,3)} = x_{(3,2)} = \frac{1}{R_{2\text{-}3,rad}} = 29.07 \text{ Btu/(hr-°F)}$$

$$x_{(1,4)} = x_{(2,4)} = x_{(3,4)} = 0$$

$$x_{(4,1)} = \frac{1}{R_{1,conv}} = 16 \text{ Btu/(hr-°F)}$$

$$x_{(4,2)} = \frac{1}{R_{2,conv}} = 16 \text{ Btu/(hr-°F)}$$

$$x_{(4,3)} = \frac{1}{R_{3,conv}} = 160 \text{ Btu/(hr-°F)}$$

$$x_{(4,4)} = -1$$

Thus, the **X** matrix is

$$\mathbf{X} = \begin{bmatrix} -45.36 & 0.29 & 29.07 & 0 \\ 0.29 & -45.36 & 29.07 & 0 \\ 29.07 & 29.07 & -218.14 & 0 \\ 16 & 16 & 160 & -1 \end{bmatrix}$$

Next, the inverse of the **X** matrix is computed.

$$\mathbf{X}^{-1} = \begin{bmatrix} -0.0244 & -0.0024 & -0.0036 & 0 \\ -0.0024 & -0.0244 & -0.0036 & 0 \\ -0.0036 & -0.0036 & -0.0055 & 0 \\ -1 & -1 & -1 & -1 \end{bmatrix}$$

The resistances between surfaces when other nodes are floating can be computed from equation (3.2.6).

$$R_{1-2} = x_{(1,2),inv} + x_{(2,1),inv} - x_{(1,1),inv} - x_{(2,2),inv} = 0.0348 \text{ (hr-°F)/Btu}$$

$$R_{1-3} = x_{(1,3),inv} + x_{(3,1),inv} - x_{(1,1),inv} - x_{(3,3),inv} = 0.0227 \text{ (hr-°F)/Btu}$$

$$R_{2\text{-}3} = x_{(2,3),inv} + x_{(3,2),inv} - x_{(2,2),inv} - x_{(3,3),inv} = 0.0227 \text{ (hr-°F)/Btu}$$

The resistances between the surfaces and the air node when other surface nodes are floating can be computed from equation (3.2.8).

$$R_{1\text{-}r} = -x_{(1,1),inv} = 0.0244 \text{ (hr-°F)/Btu}$$

$$R_{2\text{-}r} = -x_{(2,2),inv} = 0.0244 \text{ (hr-°F)/Btu}$$

$$R_{3\text{-}r} = -x_{(3,3),inv} = 0.0055 \text{ (hr-°F)/Btu}$$

The resistance between the star node and the air node is computed from equation (3.2.14).

$$R = \frac{\dfrac{R_{1\text{-}r} + R_{2\text{-}r} - R_{1\text{-}2}}{R^3_{1\text{-}2}} + \dfrac{R_{1\text{-}r} + R_{3\text{-}r} - R_{1\text{-}3}}{R^3_{1\text{-}3}} + \dfrac{R_{2\text{-}r} + R_{3\text{-}r} - R_{2\text{-}3}}{R^3_{2\text{-}3}}}{2\left(\dfrac{1}{R^3_{1\text{-}2}} + \dfrac{1}{R^3_{1\text{-}3}} + \dfrac{1}{R^3_{2\text{-}3}}\right)} = 0.0035 \, \frac{\text{hr-°F}}{\text{Btu}}$$

The resistances between the surfaces and the star node can be computed from equation (3.2.9).

$$R_1 = R_{1\text{-}r} - R = 0.0209 \text{ (hr-°F)/Btu}$$

$$R_2 = R_{2\text{-}r} - R = 0.0209 \text{ (hr-°F)/Btu}$$

$$R_3 = R_{3\text{-}r} - R = 0.0020 \text{ (hr-°F)/Btu}$$

The difference in resistance between surface nodes when other nodes are floating for the view factor and the star network can be computed from equation (3.2.10).

$R_1 + R_2 - R_{1-2} = -0.0021$ (hr-°F)/Btu

$R_1 + R_3 - R_{1-3} = 0.0001$ (hr-°F)/Btu

$R_2 + R_3 - R_{2-3} = 0.0001$ (hr-°F)/Btu

Thus, the star network is an approximation to the view factor network for this example. The next section demonstrates that the error which results from using the star network is not significant.

3.2.4 Accuracy of Star Network

To test the star network for rooms in which an exact transformation does not exist, loads with the star and view factor networks were compared for rooms which contained building elements with a wide range of thermal physical properties, resistances to long-wave radiation exchange, and resistances to convective heat transfer. A three surface room and an eight surface room were used in the comparison. The three surface room contained the following building elements:

1) 32 ft^2 of exterior glazing
2) 32 ft^2 of an exterior frame wall with 3 inches of insulation
3) 320 ft^2 of 12" heavy concrete interior partition

The resistances derived in the previous section for the star network were used to model the three surface room. The eight surface room contained the following building elements:

1) 64 ft^2 of exterior glazing
2) 9.6 ft^2 of the stud path of an exterior frame wall
3) 54.4 ft^2 of the insulation path of an exterior frame wall
4) 128 ft^2 of an 8 inch low weight concrete floor deck
5) 64 ft^2 of a frame partition with 1 inch wood
6) 128 ft^2 of interior partition consisting of an 8 inch clay tile with 0.75 inch plaster
7) 64 ft^2 of interior partition consisting of 4 inch clay tile with 0.75 inch plaster
8) 128 ft^2 of a 4 inch wood deck with false ceiling

Loads resulting from step changes in outdoor temperature, indoor temperature, solar radiation gains, and radiation gains from people, equipment, and lights were computed for the star and view factor networks for both rooms. A one hour time step was used. The time step with the largest percent difference in loads between the networks is shown figure 4. Table 1 contains the percent difference in steady-state loads between the star and view factor networks for the following inputs: temperature difference between the ambient and indoor air, solar radiation gains, and radiation gains from people, equipment, and lights. Figure 4 and table 1 demonstrate that the star network accurately models the heat transfer processes for both rooms and all inputs.

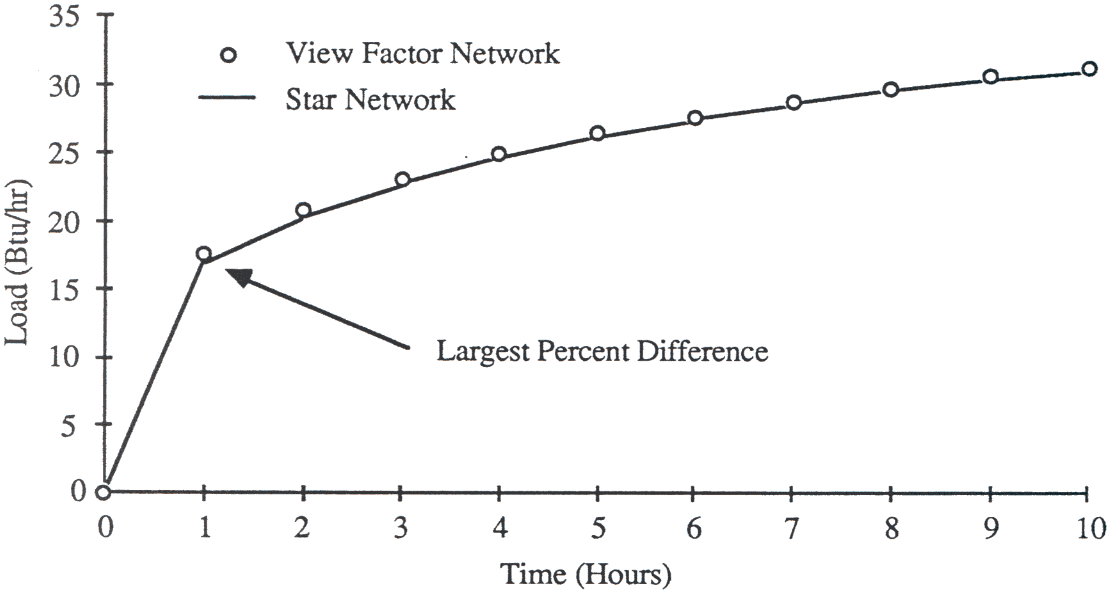


Figure 3.5 Response to 1 °F step change in outdoor temperature for the eight surface room.

Table 3.1 Percent difference in steady-state loads between the star and view factor networks.

Input	3 Surface Room	8 Surface Room
Temperature Difference	0.45%	0.69%
Solar	0.12%	0.36%
Radiation	0.006%	0.17%

3.3 TRANSFER FUNCTION COMBINATION FOR A STAR NETWORK

This section presents a method for combining individual building component transfer functions into a single transfer function for an enclosure modeled by a star network. Section 3.3.1 contains the development of equations and section 3.3.2

considers an example.

3.3.1 Formulation

Three main steps are required to determine a CRTF for a room which uses a star network to model the radiation and convection heat transfer processes. First, transfer functions are developed for each building component (e.g., wall, floor, or ceiling) which relate heat flow to the star node with the inputs. Second, transfer functions for each building component are combined in order to relate the total heat flow to the star node with the inputs. Third, the star temperature is removed from the combined transfer function equation by relating the building load to the temperature difference between the star node and the room air temperature.

A transfer function equation for an exterior wall will be developed which will relate the heat flow to the star node with solar radiation gains (from one direction), outdoor temperature, star temperature, and radiation gains from people, equipment and lights as indicated in figure 3.6. The methods previously discussed can be used to calculate the coefficients for a transfer function equation of the form

$$q_{t,k,int} = \sum_{j=0} \left(a_{j,k} \, A_k \, T_{t-j\delta,k,sa} + b_{j,k} \, A_k \, T_{t-j\delta,k,int} \right) - \sum_{j=1} c_{j,k} \, q_{t-j\delta,k,int} \qquad (3.3.1)$$

where

$T_{t-j\delta,k,sa}$ = sol-air temperature for wall k

$T_{t-j\delta,k,int}$ = temperature of interior surface of wall k

The sol-air temperature [Mitchell (1983)] for wall k is

$$T_{t-j\delta,k,sa} = T_{t-j\delta,amb} + I_{t-j\delta}\,\alpha_k\,R_{k,out}\,A_k \qquad (3.3.2)$$

where

$T_{t-j\delta,amb}$ = ambient temperature at j time steps prior to time t

$I_{t-j\delta}$ = incident solar radiation at j time steps prior to time t

α_k = solar absorptance of the exterior surface of wall k

$R_{k,out}$ = resistance to convective heat transfer between exterior surface of wall k and the outdoor air

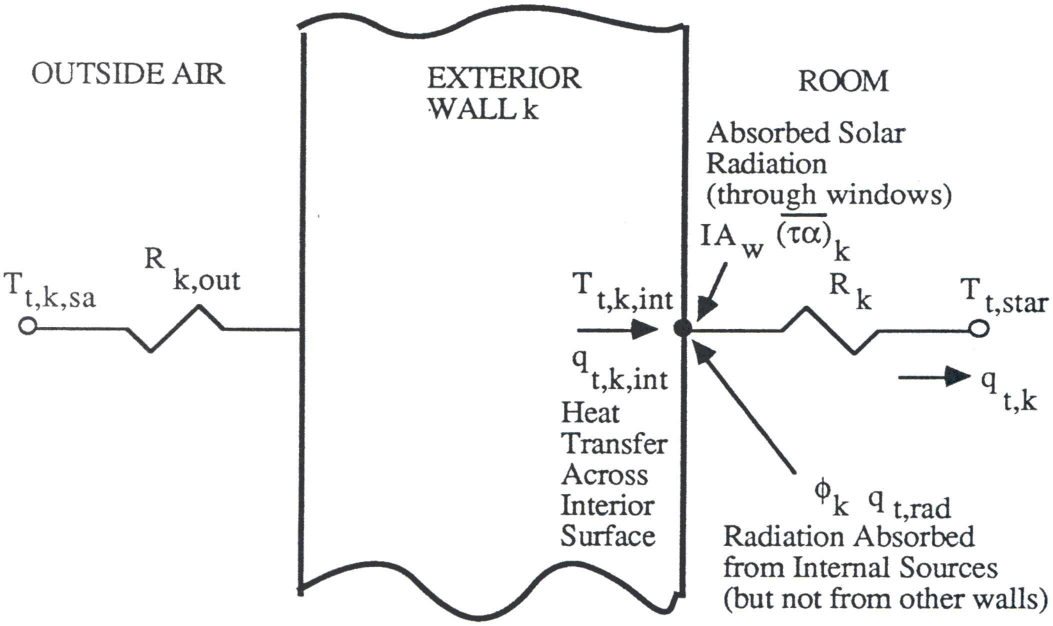

OUTSIDE AIR
EXTERIOR WALL k
ROOM
Absorbed Solar Radiation (through windows)
$IA_w\,\overline{(\tau\alpha)}_k$
$T_{t,k,sa}$
$R_{k,out}$
$T_{t,k,int}$
R_k
$T_{t,star}$
$q_{t,k,int}$
Heat Transfer Across Interior Surface
$q_{t,k}$
$\phi_k\,q_{t,rad}$
Radiation Absorbed from Internal Sources (but not from other walls)

Figure 3.6 Energy flows for an exterior wall.

An energy balance on the interior surface results in the following equation for heat flow to the star node from wall k:

$$q_{t-j\delta,k,int} = q_{t-j\delta,k} - \phi_k \, q_{t-j\delta,rad} - I_{t-j\delta} \, A_w \, \overline{(\tau\alpha)}_k \tag{3.3.3}$$

where

ϕ_k = fraction of radiation gains from people, equipment and lights which is absorbed at the interior surface of building element k

$q_{t-j\delta,rad}$ = radiation gains from people, equipment, and lights at j time steps prior to time t

The interior surface temperature is related to the star temperature and heat flow to the star node by

$$T_{t-j\delta,k,int} = R_k \, q_{t-j\delta,k} + T_{t-j\delta,star} \tag{3.3.4}$$

Substituting equations (3.3.2), (3.3.3), and (3.3.4) into equation (3.3.1) results in

$$q_{t,k} = \sum_{j=0} \left(d_{j,k} \, T_{t-j\delta,amb} + e_{j,k} \, T_{t-j\delta,star} + f_{j,k} \, I_{t-j\delta} + g_{j,k} \, q_{t-j\delta,rad} \right) - \sum_{j=1} h_{j,k} \, q_{t-j\delta,k} \tag{3.3.5}$$

where

$$d_{j,k} = \frac{a_{j,k} \, A_k}{1 - b_{0,k} \, A_k \, R_k} \tag{3.3.6}$$

$$e_{j,k} = \frac{b_{j,k} \, A_k}{1 - b_{0,k} \, A_k \, R_k} \tag{3.3.7}$$

$$f_{j,k} = \frac{c_{j,k} \, A_w \, \overline{(\tau\alpha)}_k + a_{j,k} \, \alpha_k \, R_{k,out} \, A_k^2}{1 - b_{0,k} \, A_k \, R_k} \tag{3.3.8}$$

$$g_{j,k} = \frac{c_{j,k}\,\phi_k}{1 - b_{0,k}\,R_k\,A_k} \tag{3.3.9}$$

$$h_{j,k} = \frac{c_{j,k} - b_{j,k}\,R_k\,A_k}{1 - b_{0,k}\,R_k\,A_k} \tag{3.3.10}$$

Equation (3.3.5) relates the heat flow to the star node for exterior wall k with the inputs. Next, transfer functions for heat flow to the star node from an interior partition and a window will be presented.

As shown in figure 3.7, the surface temperatures on both sides of a interior partition are assumed to be identical. Thus, the transfer function for the heat flow from an interior partition is of the form

$$q_{t,k,int} = \sum_{j=0} b_{j,k}\,A_k\,T_{t-j\delta,k,int} - \sum_{j=1} c_{j,k}\,q_{t-j\delta,k,int} \tag{3.3.11}$$

Equations (3.3.3) and (3.3.4) for the exterior wall are identical to the equations for the interior partition. Substituting equations (3.3.3) and (3.3.4) into equation (3.3.11) results in

$$q_{t,k} = \sum_{j=0}\left(e_{j,k}\,T_{t-j\delta,star} + f_{j,k}\,I_{t-j\delta} + g_{j,k}\,q_{t-j\delta,rad}\right) - \sum_{j=1} h_{j,k}\,q_{t-j\delta,k} \tag{3.3.12}$$

where

$$e_{j,k} = \frac{b_{j,k}\,A_k}{1 - b_{0,k}\,A_k\,R_k} \tag{3.3.13}$$

$$f_{j,k} = \frac{c_{j,k} A_w \overline{(\tau\alpha)}_k}{1 - b_{0,k} A_k R_k} \tag{3.3.14}$$

$$g_{j,k} = \frac{c_{j,k} \phi_k}{1 - b_{0,k} R_k A_k} \tag{3.3.15}$$

$$h_{j,k} = \frac{c_{j,k} - b_{j,k} R_k A_k}{1 - b_{0,k} R_k A_k} \tag{3.3.16}$$

Equation (3.3.12) relates the heat flow to the star node with the inputs for an interior partition. Equations (3.3.12) through (3.3.16) could also be used for room furnishings.

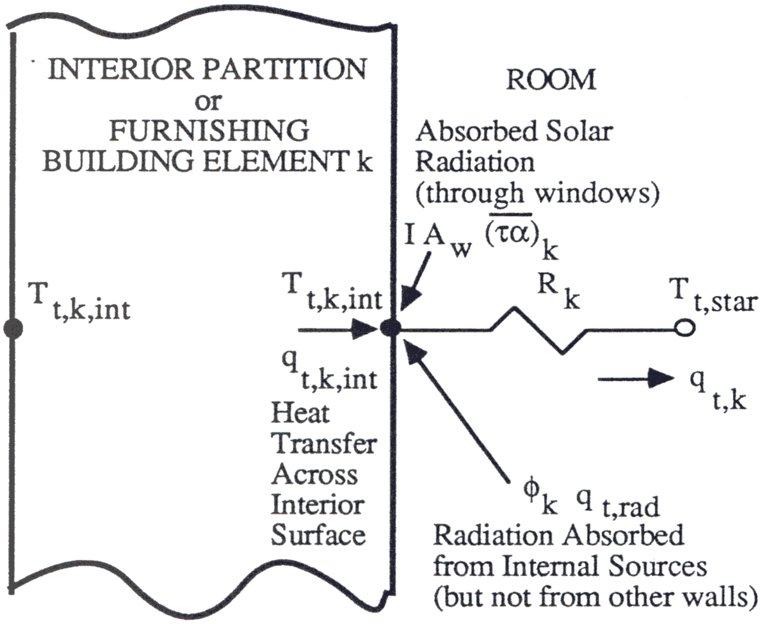


Figure 3.7 Energy flows for an interior partition.

Figure 3.8 shows energy flows for a window with a single pane of glass. The thermal capacitance of glass is small when compared to the thermal capacitance of other building elements, e.g., walls, roofs, or floors. Thus, assuming no thermal capacitance for glass will not significantly affect the load calculations for a room.

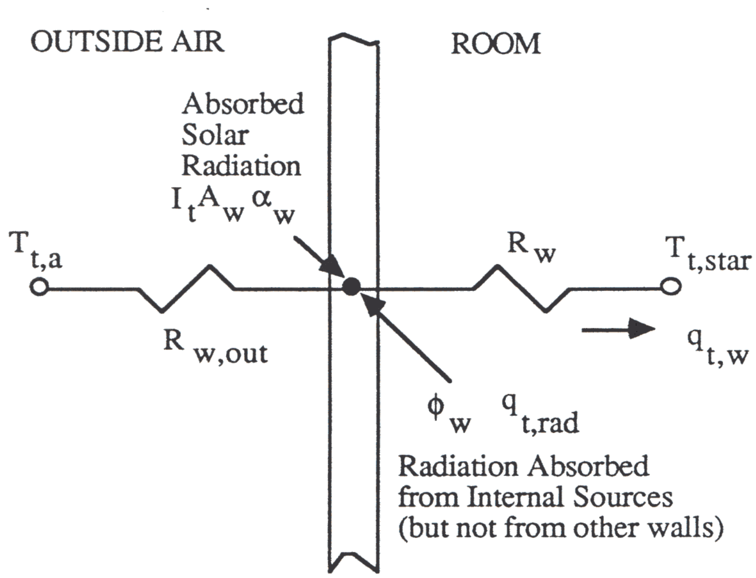


Figure 3.8 Energy flows for a window.

The heat flow to the star node for the window is

$$q_{t,w} = \frac{T_{t,amb} - T_t}{R_w + R_{w,out}} + \left(I_t A_w \alpha_w + q_{t,rad} \phi_w\right)\left(\frac{R_{w,out}}{R_w + R_{w,out}}\right) \tag{3.3.17}$$

Equation (3.3.17) can be rewritten into a transfer function form as

$$q_{t,w} = d_{0,w} T_{t,amb} + e_{0,w} T_{t,star} + f_{0,w} I_t + g_0 q_{t,rad} \tag{3.3.18}$$

where

$$d_{0,w} = \frac{1}{R_w + R_{w,out}} \tag{3.3.19}$$

$$e_{0,w} = - \frac{1}{R_w + R_{w,out}} \tag{3.3.20}$$

$$f_{0,w} = \frac{A_w \; \alpha_w \; R_{w,out}}{R_w + R_{w,out}} \tag{3.3.21}$$

$$g_{0,w} = \frac{\phi_w \; R_{w,out}}{R_w + R_{w,out}} \tag{3.3.22}$$

The transfer functions for heat flow from each building element to the star node can be combined in a nested fashion (i.e., combine the transfer functions for building elements 1 and 2, then combine the transfer function for building element 3 with the combined transfer function for building elements 1 and 2, and continue). The backshift operator can be used to combine any two transfer functions to give the combined heat flow, $q_{t,\chi}$:

$$q_{t,\chi} = \sum_{j=0} \left(d_{j,\chi} \, T_{t-j\delta,amb} + e_{j,\chi} \, T_{t-j\delta,star} + f_{j,\chi} \, I_{t-j\delta} + g_{j,\chi} \, q_{t-j\delta,rad} \right)$$

$$- \sum_{j=1} h_{j,\chi} \, q_{t-j\delta,\chi} \tag{3.3.23}$$

where

$$q_{t-j\delta,\chi} = q_{t-j\delta,1} + q_{t-j\delta,2} \tag{3.3.24}$$

$$d_{j,\chi} = \sum_{k=0}^{j} \left(d_{j-k,1} \, h_{k,2} + d_{k,2} \, h_{j-k,1} \right) \tag{3.3.25}$$

$$e_{j,\chi} = \sum_{k=0}^{j} (e_{j-k,1}\, h_{k,2} + e_{k,2}\, h_{j-k,1}) \tag{3.3.26}$$

$$f_{j,\chi} = \sum_{k=0}^{j} (f_{j-k,1}\, h_{k,2} + f_{k,2}\, h_{j-k,1}) \tag{3.3.27}$$

$$g_{j,\chi} = \sum_{k=0}^{j} (g_{j-k,1}\, h_{k,2} + g_{k,2}\, h_{j-k,1}) \tag{3.3.28}$$

$$h_{j,\chi} = \sum_{k=0}^{j} h_{k,1}\, h_{j-k,2} \tag{3.3.29}$$

Combining heat flows from every surface to the star node gives

$$\begin{aligned} q_{t,\chi} &= \left(\left(\cdots \left(\left(\left(q_{t,1} + q_{t,2}\right) + q_{t,3}\right) + q_{t,4}\right) + \ldots + q_{t,N-1}\right) + q_{t,N}\right) \\ &= q_{t,load} \end{aligned} \tag{3.3.30}$$

The load is related to the temperature difference between the star node and the air node simply by

$$q_{t,load} = \frac{T_{t,star} - T_{t,r}}{R} \tag{3.3.31}$$

Substituting equations (3.3.31) and (3.3.30) into equation (3.3.23) gives

$$q_{t,load} = \sum_{j=0} \left(d_j T_{t-j\delta,amb} + e_j T_{t-j\delta,r} + f_j I_{t-j\delta} + g_j q_{t-j\delta,rad} \right) - \sum_{j=1} h_j q_{t-j\delta,load} \quad (3.3.32)$$

where

$$d_j = \frac{d_{j,\chi}}{1 - R\, e_{0,\chi}} \quad (3.3.33)$$

$$e_j = \frac{e_{j,\chi}}{1 - R\, e_{0,\chi}} \quad (3.3.34)$$

$$f_j = \frac{f_{j,\chi}}{1 - R\, e_{0,\chi}} \quad (3.3.35)$$

$$g_j = \frac{g_{j,\chi}}{1 - R\, e_{0,\chi}} \quad (3.3.36)$$

$$h_j = \frac{h_{j,\chi} - R\, e_{j,\chi}}{1 - R\, e_{0,\chi}} \quad (3.3.37)$$

Equation (3.3.32) is a CRTF which relates the load for an enclosure to past and current inputs and past loads.

The following step-by-step procedure can be used to determine a comprehensive room transfer function from the transfer functions for individual building elements:

1) Determine a transfer function which relates the heat flow to the star node with the inputs for every building element. Use equations (3.3.6) through (3.3.10) for an exterior wall, roof, or floor. Use equations (3.3.13) through (3.3.16) for an interior partition. Use equations (3.3.19) through (3.3.22) for a window with a single pane of glass.

2) Use equations (3.3.25) through (3.3.29) to combine the building element transfer functions for a star network.

3) Use equations (3.3.33) through (3.3.37) to remove the star temperature from the combined transfer function, i.e., the transfer function which was obtained in the previous step.

3.3.2 Example for a Room with Three Surfaces

This section discusses the development of a CRTF for the three surface room described in sections 3.2.3 and 3.2.4. Properties of the building elements can be seen in table 3.2. Mitalas and Arsenault's program (1971) was used to generate the transfer function coefficients for the individual building elements, although the method described in chapter 2 could have been used. Table 3.3 contains the transfer function coefficients for the individual building elements. It required 5.52 seconds of central processing unit (CPU) time on a Digital Equipment Company MicroVax computer to compute the transfer functions for the individual building elements.

The step-by-step procedure described in the previous section was used to compute the CRTF listed in table 3.4. It required 0.37 seconds of CPU time on a MicroVax computer to compute the CRTF coefficients from the transfer functions for the individual building elements and a building description. Thus, the amount of computational effort required to compute the CRTF coefficients is small when compared to the effort required to compute transfer function for individual building elements.

Tables 3.3 and 3.4 show that the number of past time steps in the CRTF is greater than the number of past time steps for the individual building element transfer functions. Not all of the coefficients in the CRTF are needed and the model reduction methods described in the next chapter can be used to determine a smaller set of CRTF coefficients which provides nearly the same results as the full set of coefficients.

Table 3.2 Properties of the room.

	Window	Exterior Wall	Interior Partition
α	0.1	0.5	–
$(\tau\alpha)$	0.0	0.0	0.8
ϕ	0.0833	0.0833	0.8333
R_{out}	0.0381 hr-°F/Btu	0.0104 hr-°F/Btu	-

Table 3.3 Transfer functions coefficients for the individual building elements.

	Exterior Wall			Interior Partition	
j	a_j Btu/hr-°F	b_j Btu/hr-°F	c_j	b_j Btu/hr-°F	c_j
0	0.01873	-1.35837	1.00000	-4.14658	1.00000
1	0.04125	1.63904	-0.27106	7.64987	-1.37632
2	0.00287	-0.34363	0.00010	-4.35962	0.52523
3		0.00011		0.90665	-0.05488
4				-0.05111	0.00106
5				0.00078	

Table 3.4 CRTF coefficients for the three surface room.

j	d_j Btu/hr-°F	e_j Btu/hr-°F	f_j ft^2	g_j Btu/hr-°F	h_j
0	7.22392	-166.37440	3.77590	0.13485	1.00000
1	-17.20155	428.61942	-8.13945	-0.28796	-2.51688
2	15.67038	-420.61942	6.60672	0.22942	2.41420
3	-7.11566	202.29428	-2.58204	-0.08705	-1.13912
4	1.69724	-50.26068	0.50493	0.01627	0.27766
5	-0.19811	6.02287	-0.04554	-0.00137	-0.03270
6	0.00882	-0.27649	0.00146	0.00004	0.00147
7	-0.00012	0.00397	-0.00001	0.00000	-0.00002

3.4 COMPUTATIONAL EFFORT

Table 3.5 contains the number of multiplications required per time step for energy balance simulations of view factor networks with time independent **A** matrices and CRTF simulations for the three and eight surface rooms previously described. Table 2 shows that for the three surface room there is no computational savings in performing a CRTF simulation, and for the eight surface room there is not a large computational savings. For a CRTF simulation, the number of multiplications required per time step is equal to the number of coefficients in the CRTF. The number of multiplications required per time step for an energy balance simulation was determined from an equation described in appendix E.

Table 3.5 Number of multiplications required per time step for an energy balance simulation and a CRTF simulation.

	Multiplications per Time Step	
	Energy Balance	CRTF
3 surface room	35	38
8 surface room	152	109

3.5 SUMMARY

This chapter described a method in which transfer functions for individual building elements are combined into a comprehensive room transfer function (CRTF), i.e., a single transfer function which relates the heating or cooling loads for a room to the inputs. Long-wave radiation exchange and convection in the room are accurately modeled with the star network. A procedure for determining the resistances in the star network from a network which uses view factors to model long-wave radiation exchange was presented.

The computational effort of a CRTF simulation was compared with the computational effort of an energy balance simulation for two rooms. For the rooms tested, the computational effort of a CRTF simulation was not significantly different than the computational effort of an energy balance simulation. Computational effort of CRTF simulations can be significantly reduced by using model reduction methods to find a smaller set of coefficients. The next chapter is devoted to model reduction methods.

CHAPTER 4
Model Reduction

The previous two chapters have described methods for obtaining transfer functions for building elements and for rooms. This chapter describes model reduction methods for reducing the number of coefficients in transfer functions. These methods are essential for decreasing the computational effort of simulations.

A number of different methods for reducing the order of transfer functions have been developed by researchers in the fields of automatic controls and system analysis. The motivation behind the development of these methods is to reduce computer time for system simulation and to make control system design and analysis easier.

Shamash (1980) said that the Padé approximation is a popular method for reducing single-input Laplace transfer functions because it requires little computational effort, cancels common factors if they exist, and matches the steady-state response of the original and reduced Laplace transfer functions for polynomial inputs. There are two disadvantages of the Padé approximation. First, if the original transfer function is stable, the reduced transfer function is not guaranteed to be stable. Second, reduced multiple-input transfer functions cannot be obtained with the Padé approximation. A number of model reduction methods [Jamshidi (1983)] have been developed which do not have the stability preservation problem and can be used to reduce multiple-input transfer functions. After comparing four different methods which do not have the stability preservation problem with the Padé approximation, Shamash (1982) said:

> "...if there is a common pole and zero close to the origin, then all these four methods will lead to erroneous results. Further, all these methods are based upon approximating the dominant (closest to the origin) poles and zeros of the system. Thus, if a system has dominant poles that are furthest away from the origin, then again these methods will produce erroneous approximations. Finally, when applied to multi-variable systems these methods may lead to models that are higher in order than the original system.
>
> These problems do not occur in Padé approximations."

Shih and Wu (1973) have used the bilinear transformation [Kuo (1980)] and continued fraction expansion, which is a reduction method for Laplace transfer functions, to reduce single-input (discrete) transfer functions. Jamshidi (1983) said that the continued fraction method has several limitations. One of the limitations is that it cannot be used to reduce multiple-input transfer functions when the number of inputs is equal to the number of outputs. Thus, continued fraction expansion cannot be used to obtain a reduced multiple-input CRTF.

Section 4.1 reviews the equations for reducing single-input Laplace transfer functions by means of the Padé approximation. Section 4.2 extends the Padé approximation to single-input transfer functions by using the bilinear transformation. Then, section 4.3 presents a new model reduction method which does not have the stability preservation problem and can be used to reduce both single-input and multiple-input transfer functions with any number of inputs or outputs. This new model reduction method uses equations presented in sections 4.1 and 4.2 to determine transfer function coefficients for the inputs. Section 4.4 describes a procedure for determining the minimum number of past time steps required to accurately model heat transfer processes with reduced transfer functions. In section 4.5, the computational effort of energy balance simulations is compared with the computational effort of reduced CRTF simulations for two rooms.

4.1 PADE APPROXIMATION

Jamshidi (1983) presented equations for reducing Laplace (i.e., continuous) transfer functions when the order of the numerator is equal to or less than the order of the denominator. In the next section, equations for reducing Laplace transfer functions when the order of the numerator is equal to the order of the denominator are needed. Therefore, this section contains equations for reducing single-input Laplace transfer functions with the Padé approximation when the order of the numerator is equal to the order of the denominator.

The reduced Laplace transfer function

$$G_r(s) = \frac{\bar{d}_0 + \bar{d}_1 s + \bar{d}_2 s^2 + \ldots + \bar{d}_m s^m}{1 + \bar{e}_1 s + \bar{e}_2 s^2 + \ldots + \bar{e}_m s^m} \qquad (4.1.1)$$

is the Padé approximation of

$$G(s) = \frac{\bar{a}_0 + \bar{a}_1 s + \bar{a}_2 s^2 + \ldots + \bar{a}_n s^n}{1 + \bar{b}_1 s + \bar{b}_2 s^2 + \ldots + \bar{b}_n s^n} \qquad (4.1.2)$$

(where m is less than n) if the power series expansion for $G_r(s)$ is equal to the power series expansion of G(s) for terms of order s^0 to s^{2m}. Next, the equations for calculating the power series of

$$G(s) = \bar{c}_0 + \bar{c}_1\, s + \bar{c}_2\, s^2 + \bar{c}_3\, s^3 + \ldots \tag{4.1.3}$$

will be formulated. The following equation results from equating equation (4.1.3) with equation (4.1.2):

$$\bar{a}_0 + \bar{a}_1\, s + \bar{a}_2\, s^2 + \ldots + \bar{a}_n\, s^n = \left(1 + \bar{b}_1\, s + \bar{b}_2\, s^2 + \ldots + \bar{b}_n\, s^n\right)\Big(\bar{c}_0 + \bar{c}_1\, s + \bar{c}_2\, s^2 + \ldots \tag{4.1.4}$$

Multiplying the terms on the right hand side of equation (4.1.4) together and combining common powers of the Laplace transform variable s results in

$$\bar{a}_0 + \bar{a}_1\, s + \bar{a}_2\, s^2 + \ldots + \bar{a}_n\, s^n = \bar{c}_0 + \left(\bar{c}_0\, \bar{b}_1 + \bar{c}_1\right) s + \left(\bar{c}_0\, \bar{b}_2 + \bar{c}_1\, \bar{b}_1 + \bar{c}_2\right) s^2 + \ldots \tag{4.1.5}$$

The $\bar{c}_i$ coefficients for the power series expansion of G(s) are determined by equating the coefficients of equal powers of s in equation (4.1.5).

$$\bar{c}_0 = \bar{a}_0$$

$$\bar{c}_1 = \bar{a}_1 - \bar{c}_0\, \bar{b}_1$$

$$\vdots$$

$$\bar{c}_2 = \bar{a}_2 - \bar{c}_0\, \bar{b}_2 - \bar{c}_1\, \bar{b}_1$$

$$\bar{c}_i = \bar{a}_i - \sum_{j=0}^{i-1} \bar{c}_j\, \bar{b}_{i-j} \tag{4.1.6}$$

To calculate the Padé approximation for G(s), the power series for G(s) is set equal to the Laplace transfer function $G_r(s)$. The following equation results from combining equal powers of s when the numerator of $G_r(s)$ is set equal to the denominator of $G_r(s)$ times G(s):

$$
\begin{aligned}
\bar{d}_0 + \bar{d}_1 s + \bar{d}_2 s^2 + \ldots + \bar{d}_m s^m &= \left(1 + \bar{e}_1 s + \bar{e}_2 s^2 + \ldots + \bar{e}_m s^m\right)\left(\bar{c}_0 + \bar{c}_1 s + \bar{c}_2 s^2 + \ldots\right. \\
&= \bar{c}_0 + \left(\bar{c}_1 + \bar{e}_1 \bar{c}_0\right) s + \left(\bar{c}_2 + \bar{e}_1 \bar{c}_1 + \bar{e}_2 \bar{c}_0\right) s^2 \\
&\quad + \ldots \\
&\quad + \left(\bar{c}_m + \bar{e}_1 \bar{c}_{m-1} + \bar{e}_2 \bar{c}_{m-2} + \ldots + \bar{e}_m \bar{c}_0\right) s^m \\
&\quad + \left(\bar{c}_{m+1} + \bar{e}_1 \bar{c}_m + \bar{e}_2 \bar{c}_{m-1} + \ldots + \bar{e}_m \bar{c}_1\right) s^{m+1} \\
&\quad + \left(\bar{c}_{m+2} + \bar{e}_1 \bar{c}_{m+1} + \bar{e}_2 \bar{c}_m + \ldots + \bar{e}_m \bar{c}_2\right) s^{m+2} \\
&\quad + \ldots \\
&\quad + \left(\bar{c}_{2m} + \bar{e}_1 \bar{c}_{2m-1} + \bar{e}_2 \bar{c}_{2m-2} + \ldots + \bar{e}_m \bar{c}_m\right) s^{2m} \\
&\quad + \ldots \qquad (4.1.7)
\end{aligned}
$$

A set of m linear equations with m unknown denominator coefficients ($\bar{e}_i$) of the Padé approximation can be formulated by equating powers of s from (m+1) to (2m) in equation (4.1.7).

$$\begin{bmatrix} \bar{c}_m & \bar{c}_{m-1} & \cdots & \bar{c}_2 & \bar{c}_1 \\ \bar{c}_{m+1} & \bar{c}_m & \cdots & \bar{c}_3 & \bar{c}_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \bar{c}_{2m-2} & \bar{c}_{2m-3} & \cdots & \bar{c}_m & \bar{c}_{m-1} \\ \bar{c}_{2m-1} & \bar{c}_{2m-2} & \cdots & \bar{c}_{m+1} & \bar{c}_m \end{bmatrix} \begin{bmatrix} \bar{e}_1 \\ \bar{e}_2 \\ \cdot \\ \cdot \\ \cdot \\ \bar{e}_{m-1} \\ \bar{e}_m \end{bmatrix} = \begin{bmatrix} -\bar{c}_{m+1} \\ -\bar{c}_{m+2} \\ \cdot \\ \cdot \\ \cdot \\ -\bar{c}_{2m-1} \\ -\bar{c}_{2m} \end{bmatrix} \quad (4.1.8)$$

The denominator of the Padé approximation is determined by solving equation (4.1.8) for the $\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_m$ coefficients. After determining the denominator of the Padé approximation, numerator coefficients of the Padé approximation are determined by equating powers of s from 0 to m in equation (4.1.7).

$$\bar{d}_0 = \bar{c}_0$$

$$\bar{d}_1 = \bar{c}_1 + \bar{e}_1 \bar{c}_0$$

$$\bar{d}_2 = \bar{c}_2 + \bar{e}_1 \bar{c}_1 + \bar{e}_2 \bar{c}_0$$

$$\cdot$$

$$\cdot$$

$$\cdot$$

$$\bar{d}_m = \bar{c}_m + \sum_{i=1}^{m} \bar{e}_i \bar{c}_{m-i} \quad (4.1.9)$$

In summary, the following step-by-step procedure can be used to obtain a reduced Laplace transfer function by the Padé approximation:

1) Use equation (4.1.6) to determine the power series expansion of a Laplace transfer function for terms of order s^0 to s^{2m}.

2) Use equation (4.1.8) to determine the denominator of the reduced Laplace transfer function.

3) Use equation (4.1.9) to determine the numerator of the reduced Laplace transfer function.

4.2 PADE APPROXIMATION AND BILINEAR TRANSFORMATION

This section extends the Padé approximation to model reduction for single-input (discrete) transfer functions. Section 4.2.1 contains equations for reducing single-input transfer functions with the Padé approximation and the bilinear transformation. An algorithm which is useful for efficiently computing reduced transfer functions is described in section 4.2.2. In section 4.2.3, an example of model reduction for single-input transfer functions is presented. Section 4.2.4 contains a discussion of the conversion of multiple-input transfer functions, e.g., the CRTF developed in Chapter 3, into single-input transfer functions. Section 4.2.5 contains examples of model reduction for a building element and a CRTF.

4.2.1 Formulation

This section contains equations for reducing single-input transfer functions with the bilinear transformation and Padé approximation.

The following single-input transfer function relates the inputs of a system to the outputs:

$$y_t = \sum_{j=0}^{n}\left(a_j\, u_{t-j\delta}\right) - \sum_{j=1}^{n}\left(b_j\, y_{t-j\delta}\right) \tag{4.2.1}$$

where

$y_{t-j\delta}$ = output j time steps prior to time t

n = number of past time steps in transfer function

a_j = transfer function coefficient for input

$u_{t-j\delta}$ = input at j time steps prior to time t

b_j = transfer function coefficient for output

Taking the z-transformation [Jury (1964)] of equation (4.2.1) results in

$$\left[\sum_{j=0}^{n}\left(b_j\, z^{-j}\right)\right] Y(z) = \left[\sum_{j=0}^{n}\left(a_j\, z^{-j}\right)\right] U(z) \tag{4.2.2}$$

where

$b_0 = 1$

Y(z) = z-transform of the output

U(z) = z-transform of the input

From equation (4.2.2), the z-transfer function is

$$G(z) = \frac{Y(z)}{U(z)} = \frac{\sum_{j=0}^{n} a_i z^{-j}}{\sum_{j=0}^{n} b_i z^{-j}} \qquad (4.2.3)$$

Equation (4.2.3) is a z-transfer function which relates the z-transform of the input to the z-transform of the output and it is unstable if there are poles outside the unit circle. (Poles are roots of the denominator of a transfer function.) To reduce z-transfer functions, the bilinear transformation [Kuo (1980)]

$$z = \frac{1 + w}{1 - w} \qquad (4.2.4)$$

is used to transform a z-transfer function into a w-transfer function. (A w-transfer function is a ratio of polynomials of the complex variable w.) The bilinear transformation maps the unit circle on the z-plane into the left half of the w-plane. A w-transfer function behaves like a Laplace transfer function because both transfer functions are unstable if they have poles in the right half of their complex planes. Substituting equation (4.2.4) into equation (4.2.3) gives the following w-transfer function:

$$G(w) = \frac{\sum_{j=0}^{n} a_j \left(\frac{1 + w}{1 - w}\right)^{-j}}{\sum_{j=0}^{n} b_j \left(\frac{1 + w}{1 - w}\right)^{-j}} = \frac{\sum_{j=0}^{n} a_j \left(\frac{1 - w}{1 + w}\right)^{j}}{\sum_{j=0}^{n} b_j \left(\frac{1 - w}{1 + w}\right)^{j}} = \frac{\sum_{j=0}^{n} a_j \,(1 - w)^{j}\,(1 + w)^{n-j}}{\sum_{j=0}^{n} b_j \,(1 - w)^{j}\,(1 + w)^{n-j}} \qquad (4.2.5)$$

Section 4.2.2 describes an algorithm for determining the $v_{i(j,n)}$ coefficients in the following equation:

$$(1 - w)^{j} (1 + w)^{n-j} = \sum_{i=0}^{n} v_{i(j,n)} w^{i} \tag{4.2.6}$$

Substituting equation (4.2.6) into equation (4.2.5) gives

$$G(w) = \frac{\sum_{j=0}^{n} a_j \left(\sum_{i=0}^{n} v_{i(j,n)} w^{i} \right)}{\sum_{j=0}^{n} b_j \left(\sum_{i=0}^{n} v_{i(j,n)} w^{i} \right)} = \frac{\sum_{i=0}^{n} \left(\sum_{j=0}^{n} a_j v_{i(j,n)} \right) w^{i}}{\sum_{i=0}^{n} \left(\sum_{j=0}^{n} b_j v_{i(j,n)} \right) w^{i}} \tag{4.2.7}$$

Equation (4.2.7) is a ratio of polynomials of the complex variable w. The Padé approximation for reducing Laplace transfer functions can also be used to reduce w-transfer functions. To use the equations in section 4.1.1 to reduce w-transfer functions, the coefficient for w^0 in the denominator of equation (4.2.7) must be set equal to 1.

$$G(w) = \frac{\sum_{i=0}^{n} \bar{a}_i w^{i}}{\sum_{i=0}^{n} \bar{b}_i w^{i}} \tag{4.2.8}$$

where

$$\bar{a}_i = \frac{\sum_{j=0}^{n} a_j \, v_{i(j,n)}}{\sum_{j=0}^{n} b_j \, v_{0(j,n)}} \tag{4.2.9}$$

$$\bar{b}_i = \frac{\sum_{j=0}^{n} b_j \, v_{i(j,n)}}{\sum_{j=0}^{n} b_j \, v_{0(j,n)}} \tag{4.2.10}$$

The Padé approximation described in section 4.1 can be used to obtain a reduced w-transfer function of the following form:

$$G_r(w) = \frac{\sum_{i=0}^{m} \bar{d}_i \, w^i}{\sum_{i=0}^{m} \bar{e}_i \, w^i} \tag{4.2.11}$$

where

$\bar{e}_0 = 1$

m = number of past time steps in reduced transfer function

Next, the reduced w-transfer is transformed into a reduced z-transfer function by using the bilinear transformation

$$w = \frac{z - 1}{z + 1} = \frac{1 - z^{-1}}{1 + z^{-1}} \tag{4.2.12}$$

Substituting equation (4.2.12) into equation (4.2.11) results in

$$G_r(z) = \frac{\sum_{i=0}^{m} \bar{d}_i \left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)^i}{\sum_{i=0}^{m} \bar{e}_i \left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)^i} = \frac{\sum_{i=0}^{m} \bar{d}_i \left(1 - z^{-1}\right)^i \left(1 + z^{-1}\right)^{m-i}}{\sum_{i=0}^{m} \bar{e}_i \left(1 - z^{-1}\right)^i \left(1 + z^{-1}\right)^{m-i}} \tag{4.2.13}$$

The algorithm described in section 4.2.2 can be used to compute the $v_{j(i,m)}$ coefficients in the following equation:

$$\left(1 - z^{-1}\right)^i \left(1 + z^{-1}\right)^{m-i} = \sum_{j=0}^{m} v_{j(i,m)} z^{-j} \tag{4.2.14}$$

Substituting equation (4.2.14) into equation (4.2.13) gives

$$G_r(z) = \frac{\sum_{i=0}^{m} \bar{d}_i \left(\sum_{j=0}^{m} v_{j(i,m)} z^{-j} \right)}{\sum_{i=0}^{m} \bar{e}_i \left(\sum_{j=0}^{m} v_{j(i,m)} z^{-j} \right)} = \frac{\sum_{j=0}^{m} \left(\sum_{i=0}^{m} \bar{d}_i v_{j(i,m)} \right) z^{-j}}{\sum_{j=0}^{m} \left(\sum_{i=0}^{m} \bar{e}_i v_{j(i,m)} \right) z^{-j}} \tag{4.2.15}$$

Equation (4.2.15) can be rewritten as

$$G_r(z) = \frac{\sum_{j=0}^{m} d_j z^{-j}}{\sum_{j=0}^{m} e_j z^{-j}} \tag{4.2.16}$$

where

$$d_j = \frac{\sum_{i=0}^{m} \bar{d}_i v_{j(i,m)}}{\sum_{i=0}^{m} \bar{e}_i v_{0(i,m)}} \tag{4.2.17}$$

$$e_j = \frac{\sum_{i=0}^{m} \bar{e}_i v_{j(i,m)}}{\sum_{i=0}^{m} \bar{e}_i v_{0(i,m)}} \tag{4.2.18}$$

Transforming equation (4.2.16) back into the time domain gives the following reduced transfer function.

$$y_t = \sum_{j=0}^{m} d_j u_{t-j\delta} - \sum_{j=1}^{m} e_j y_{t-j\delta} \tag{4.2.19}$$

Next, a step-by-step procedure for determining reduced transfer functions by using the Padé approximation and bilinear transformation will be presented. The step-by-step procedure also identifies the equations needed for writing a computer program to do the reduction.

1) Use z-transform theory and the bilinear transformation to determine the w-transfer function from the transfer function. [Use equations (4.2.9) and (4.2.10) along with the algorithm described in section 4.2.2 to obtain the coefficients in equation (4.2.8).]

2) Use equation (4.1.6) to determine the power series expansion of the w-transfer function for terms of order w^0 to w^{2m}.

3) Use equation (4.1.8) to determine the denominator of the reduced w-transfer function.

4) Use equation (4.1.9) to determine the numerator of the reduced w-transfer function.

5) Use the bilinear transformation and z-transform theory to obtain a reduced transfer function from a reduced w-transfer function. [Use equations (4.2.17) and (4.2.18) along with the algorithm presented in section 4.2.2 to obtain the coefficients in equation (4.2.19).]

4.2.2 Extension of Pascal's Triangle

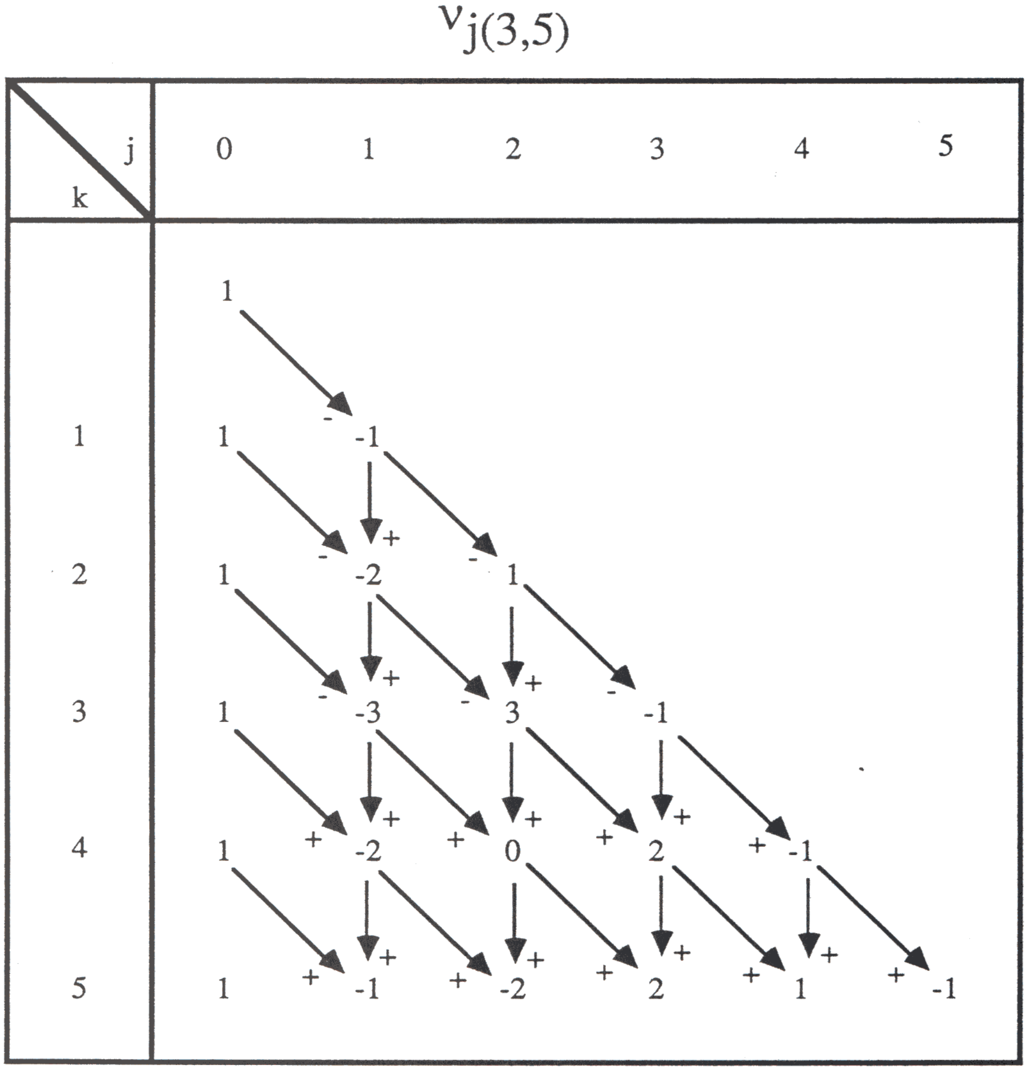
Numerical analysis textbooks [Conte and de Boor (1980) and Secgewick (1983)] contain algorithms for multiplying polynomials. These algorithms could be used to obtain the $v_{j(i,n)}$ coefficients in the following equation:

$$(1 - x)^{i} (1 + x)^{n-i} = \sum_{j=0}^{n} v_{j(i,n)} x^{j} \tag{4.2.20}$$

This section contains a numerically efficient algorithm for determining the $v_{j(i,n)}$ coefficients in equation (4.2.20). The algorithm is numerically efficient because no multiplications or divisions are required. The following algorithm for computing the $v_{j(i,n)}$ coefficients in equation (4.2.20) is based upon an extension of Pascal's triangle [Spiegel (1968)]:

```
$v_{0(i,n)} = 1$
For k = 1 to i with a step size of 1
    $v_{k(i,n)} = -v_{k-1(i,n)}$
    For j = (k-1) to 1 with a step size of -1
        $v_{j(i,n)} = v_{j(i,n)} - v_{j-1(i,n)}$
    Next j
Next k
For k = (i+1) to n with a step size of 1
    $v_{k(i,n)} = v_{k-1(i,n)}$
    For j = (k-1) to 1 with a step size of -1
        $v_{j(i,n)} = v_{j(i,n)} + v_{j-1(i,n)}$
    Next j
Next k
```

Figure 4.1 shows the steps required to compute the coefficients when i equals 3 and n equals 5.



$v_{j(3,5)}$

k \ j	0	1	2	3	4	5
	1					
1	1	-1				
2	1	-2	1			
3	1	-3	3	-1		
4	1	-2	0	2	-1	
5	1	-1	-2	2	1	-1

Figure 4.1 Steps to compute the coefficients when i = 3 and n = 5.

4.2.3 Example

The following transfer function will be used to demonstrate model reduction for discrete systems by using the bilinear transformation with the Padé approximation:

$$y_t = a_0\, u_t + a_1\, u_{t-\delta} + a_2\, u_{t-2\delta} - b_1\, y_{t-\delta} - b_2\, y_{t-2\delta}$$
$$= u_t + 0.5\, u_{t-\delta} + 0.1\, u_{t-2\delta} + 1.1\, y_{t-\delta} - 0.3\, y_{t-2\delta}$$

From equation (4.2.5), the w-transfer function is

$$G(w) = \frac{a_0\,(1+w)^2 + a_1\,(1-w)\,(1+w) + a_2\,(1-w)^2}{b_0\,(1+w)^2 + b_1\,(1-w)\,(1+w) + b_2\,(1-w)^2}$$
$$= \frac{(1+w)^2 + 0.5\,(1-w)\,(1+w) + 0.1\,(1-w)^2}{(1+w)^2 - 1.1\,(1-w)\,(1+w) + 0.3\,(1-w)^2}$$
$$= \frac{1.6 + 1.8\,w + 0.6\,w^2}{0.2 + 1.4\,w + 2.4\,w^2}$$

The coefficients in equation (4.2.8) are

$$G(w) = \frac{\bar{a}_0 + \bar{a}_1\,w + \bar{a}_2\,w^2}{\bar{b}_0 + \bar{b}_1\,w + \bar{b}_2\,w^2} = \frac{8 + 9\,w + 3\,w^2}{1 + 7\,w + 12\,w^2}$$

Equation (4.1.6) can be used to calculate the power series expansion for G(w).

$$G(w) = \bar{c}_0 + \bar{c}_1\,w + \bar{c}_2\,w^2 + \ldots = 8 - 47\,w + 236\,w^2 + \ldots$$

Assume, the reduced w-transfer function is of the form

$$G_r(w) = \frac{\bar{d}_0 + \bar{d}_1 w}{1 + \bar{e}_1 w}$$

Using equation (4.1.8), the unknown coefficient in the denominator of reduced w-transfer function is

$$\bar{e}_1 = -\frac{\bar{c}_2}{\bar{c}_1} = -\frac{236}{-47} = 5.02128$$

Numerator coefficients of the reduced w-transfer function are determined from equation (4.1.9).

$$\bar{d}_0 = \bar{c}_0 = 8$$

$$\bar{d}_1 = \bar{c}_1 + \bar{e}_1 \bar{c}_0 = -47 + 5.02128\ (8) = -6.82979$$

Thus, the reduced w-transfer function is

$$G_r(w) = \frac{8 - 6.82979\ w}{1 + 5.02128\ w}$$

The reduced z-transfer function can be calculated from equation (4.2.13).

$$G_r(z) = \frac{\bar{d}_0\left(1 + z^{-1}\right) + \bar{d}_1\left(1 - z^{-1}\right)}{\bar{e}_0\left(1 + z^{-1}\right) + \bar{e}_1\left(1 - z^{-1}\right)}$$

$$= \frac{8\left(1 + z^{-1}\right) - 6.82979\left(1 - z^{-1}\right)}{1 + z^{-1} + 5.02128\left(1 - z^{-1}\right)}$$

$$= \frac{0.19435 + 2.46290\ z^{-1}}{1 - 0.66784\ z^{-1}}$$

Finally, the reduced z-transfer function is transformed back into the time domain.

$$y_t = 0.19435\ u_t + 2.46290\ u_{t-\delta} + 0.66784\ y_{t-\delta}$$

4.2.4 Single-Input Transfer Functions from Multiple-Input Transfer Functions

To use single-input model reduction methods for multiple-input transfer functions, superposition is used to decompose multiple-input transfer functions into single-input transfer functions. One single-input transfer function is required for each input. Performing a simulation with single-input transfer functions would require more computational effort than would be required with the multiple-input transfer functions; however, the model reduction methods discussed in sections 4.2.1 and 4.3.1 can be used to obtain reduced sets of single-input transfer functions. Section 4.4 shows that reduced sets of single-input transfer functions greatly decrease the necessary computational effort of performing simulations.

For linear systems, the net response of several inputs acting simultaneously is equal to the summation of the individual inputs acting one at a time. Thus, when the heat transfer processes in a room are linear, the net heating or cooling load for a room is

$$q_{t,load} = q_{t,load,amb} + q_{t,load,r} + q_{t,load,I} + q_{t,load,rad} \tag{4.2.21}$$

where

$q_{t,load,amb}$ = load due to changes in the ambient temperature

$q_{t,load,r}$ = load due to changes in the room temperature

$q_{t,load,I}$ = load due to changes in solar radiation gains

$q_{t,load,rad}$ = load due to changes in radiation gains from people, equipment, and lights

The load due to changes in the individual inputs can be computed from the following four single-input CRTF's:

$$q_{t,load,amb} = \sum_{j=0} \left(d_j \, T_{t-j\delta,amb} \right) - \sum_{j=1} \left(h_j \, q_{t-j\delta,load,amb} \right) \tag{4.2.22}$$

$$q_{t,load,r} = \sum_{j=0} \left(e_j \, T_{t-j\delta,r} \right) - \sum_{j=1} \left(h_j \, q_{t-j\delta,load,r} \right) \tag{4.2.23}$$

$$q_{t,load,I} = \sum_{j=0} \left(f_j \, I_{t-j\delta} \right) - \sum_{j=1} \left(h_j \, q_{t-j\delta,load,I} \right) \tag{4.2.24}$$

$$q_{t,load,rad} = \sum_{j=0} \left(g_j \, q_{t-j\delta,rad} \right) - \sum_{j=1} \left(h_j \, q_{t-j\delta,load,rad} \right) \tag{4.2.25}$$

The coefficients in the four single-input CRTF's, equations (4.2.22) through (4.2.25), are equal to the coefficients in the multiple-input CRTF, equation (3.3.32).

4.2.5 Applications

The ASHRAE Handbook of Fundamentals (1977, 1981, 1985) contains tables of single-input transfer functions for walls roofs, and interior partitions. The Padé approximation and bilinear transformation can be used to obtain a reduced set of coefficients which closely model the response of the full set of coefficients listed in ASHRAE. As an example, table 4.1 contains transfer function coefficients listed in ASHRAE for exterior wall 17 (4-in. face brick, 8-in. common brick with air space). Table 4.1 also contains reduced transfer function coefficients for 3 time steps back rather than 6 as given by ASHRAE. Figure 4.2 is a plot of the response to a 1 °F step input for the full set of coefficients, the reduced set of coefficients, and a dropped set of coefficients, i.e., the ASHRAE coefficients for 3 time steps back. The reduced coefficients closely reproduce the response of the full set of coefficients while the dropped set results in errors.

To test the Padé approximation and bilinear transformation for building elements with a wide range of properties, reduced transfer functions were determined for the following ASHRAE (1977) building elements:

1) Exterior Wall 4 (4" brick, air space, and 8" high weight concrete).
2) Exterior Wall 25 (frame wall with 4 inch brick veneer).
3) Exterior wall 32 (4" concrete wall with 2" insulation on the outside).
4) Exterior Wall 36 (frame wall with 3" insulation).
5) Interior Partition 2 (4" low weight concrete block with 0.75" plaster).
6) Interior Partition 23 (frame partition with 0.75" gypsum board).
7) Interior Partition 29 (2" furniture).
8) Roof 18 (steel sheet with 1 inch insulation).

Responses to step changes in indoor and outdoor temperatures were used to compare the reduced transfer functions with the original transfer functions. For all of the building elements tested, there was a reduced transfer function which closely modeled the response of the original transfer function. Also, all of the reduced transfer functions were stable.

When combining transfer function coefficients for building elements, the number of past time steps in the resulting transfer function increases. Fortunately, the number of past time steps required to perform a simulation can be significantly reduced by using the Padé approximation and bilinear transformation. Reduced single-input CRTF's were developed for both the three and eight surface rooms considered in section 3.2.4. Appendix F contains the original and reduced single-input CRTF's for the eight surface room. Figure 4.3 shows the response to a 1°F step change in outdoor temperature for original and reduced single-input CRTF's for the eight surface room. Similar figures for step changes in indoor temperature, solar radiation gains, and radiation gains from people, equipment, and lights are shown in appendix F.

Table 4.1 Transfer function coefficients for ASHRAE wall 17.

	ASHRAE Coefficients		Reduced Coefficients	
j	a_j Btu/(hr-°F-ft^2)	c_j	a_j Btu/(hr-°F-ft^2)	c_j
0	0.00000	1.00000	-0.00089	1.00000
1	0.00001	-2.35214	0.00412	-2.13547
2	0.00022	1.98104	-0.00686	1.47238
3	0.00090	-0.73353	0.00614	-0.32658
4	0.00080	0.12178		
5	0.00019	-0.00859		
6	0.00001	0.00021		

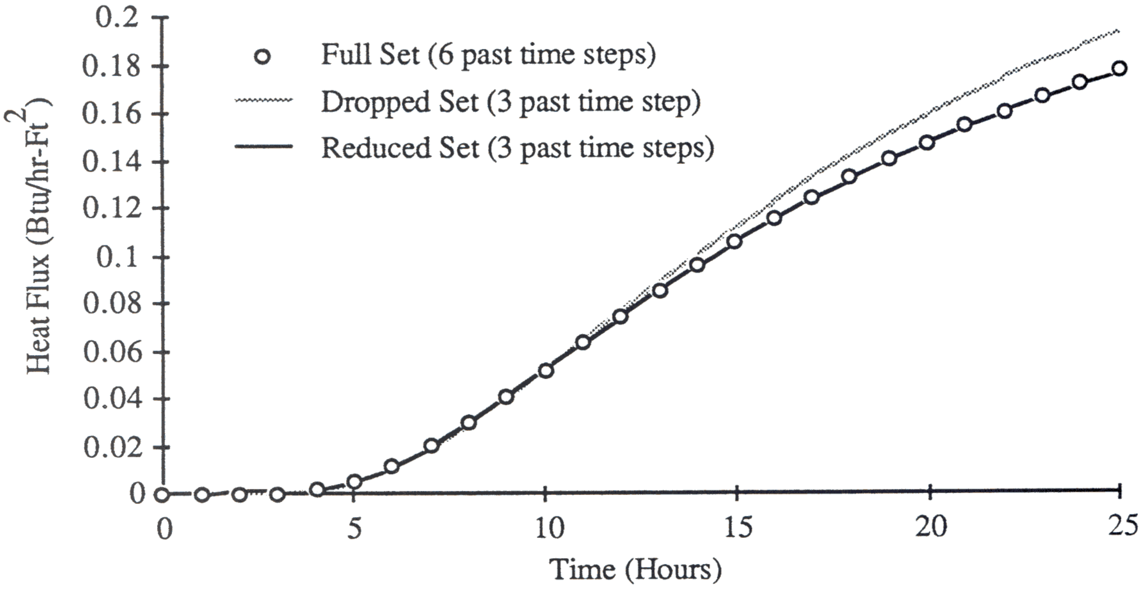


Figure 4.2 Response to 1°F step change in outdoor temperature for ASHRAE exterior wall 17.

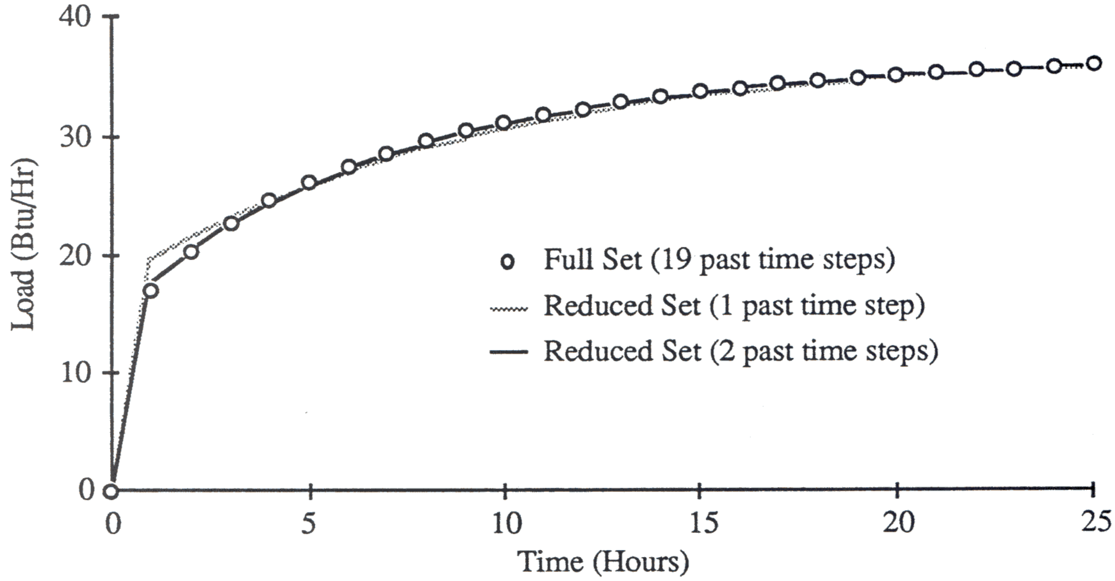


Figure 4.3 Response to 1°F step change in temperature for the eight surface room.

4.3 DOMINANT ROOT MODEL REDUCTION

This section describes a new model reduction method for reducing the number of coefficients in transfer functions which are used to solve heat transfer problems. There are two advantages of this method over the Padé approximation and bilinear transformation. First, if the original transfer function is stable, the reduced transfer function will also be stable. Second, reduced multiple-input transfer functions can be determined by this method.

Section 4.3.1 contains the development for reducing single-input transfer functions. In section 4.3.2, an example of model reduction for a single-input transfer function is considered. Section 4.3.3 presents a step-by-step procedure for reducing multiple-input transfer functions. Finally, section 4.3.4 contains an application of dominant root model reduction for a multiple-input transfer function.

4.3.1 Formulation for Single-Input Transfer Functions

This section describes a method for reducing single-input transfer functions. First, a derivation for determining the dominant roots of a transfer function based upon the infinite response to a step input is presented. The dominant roots of the original transfer function are used to determine the output coefficients of the reduced transfer function. Next, the input coefficients of the reduced transfer function are determined by equating the power series expansion of the original w-transfer function with the reduced w-transfer function for terms of order w^0 to w^m. (m is equal to the number of past time steps in the reduced transfer function.)

The following single-input transfer function relates the inputs of a system to the outputs:

$$y_t = \sum_{j=0}^{n} \left(a_j\, u_{t-j\delta} \right) - \sum_{j=1}^{n} \left(b_j\, y_{t-j\delta} \right) \tag{4.3.1}$$

The z-transform of the output, Y(z), is related to the z-transform of the input, U(z), by the following relationship:

$$Y(z) = \frac{\sum_{j=0}^{n} \left(a_j\, z^{-j} \right)}{\sum_{j=0}^{n} \left(b_j\, z^{-j} \right)}\, U(z) = \frac{\sum_{j=0}^{n} \left(a_j\, z^{n-j} \right)}{\sum_{j=0}^{n} \left(b_j\, z^{n-j} \right)}\, U(z) \tag{4.3.2}$$

The z-transform of a unit step is

$$U(z) = \frac{z}{z - 1} \tag{4.3.3}$$

Substituting equation (4.3.3) into equation (4.3.2) gives

$$Y(z) = \frac{z \sum_{j=0}^{n} a_j\, z^{n-j}}{(z - 1) \sum_{j=0}^{n} b_j\, z^{n-j}} \tag{4.3.4}$$

Equation (4.3.4) is the z-transform of the output when the input is a unit step. Equation

(4.3.4) can be rewritten in the following form:

$$Y(z) = \frac{\sum_{j=0}^{n}\left(a_j z^{n+1-j}\right)}{\prod_{j=0}^{n}\left(z - \lambda_j\right)} \tag{4.3.5}$$

where $\lambda_0 = 1$ and $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n$ are the roots of

$$z^n + b_1 z^{n-1} + b_2 z^{n-2} + \ldots + b_{n-1} z + b_n = 0 \tag{4.3.6}$$

For heat transfer problems, the roots of equation (4.3.6) are real and between zero and one. Hittle (1981) shows that all the roots are distinct for partial differential equations describing heat transfer. When all roots are distinct, the response (e.g., temperature or heat flux) to a step change in an input (e.g., temperature or heat flux) is a summation of exponentials. Transfer functions generated from finite-difference/element models or from combination of transfer functions for building elements may have multiple real roots. Appendix D describes a method for eliminating multiple roots in transfer functions.

When the roots are real and distinct, partial fraction expansion can be used to write the z-transform of the output as

$$Y(z) = \frac{\beta_0 z}{z - \lambda_0} + \frac{\beta_1 z}{z - \lambda_1} + \frac{\beta_2 z}{z - \lambda_2} + \ldots + \frac{\beta_n z}{z - \lambda_n} \tag{4.3.7}$$

where

$$\beta_j = \frac{\sum_{i=0}^{n} \left(a_i \lambda_j^{n-i} \right)}{\prod_{\substack{i=0 \\ i \neq j}}^{n} \left(\lambda_j - \lambda_i \right)} \tag{4.3.8}$$

Transforming equation (4.3.7) back to the time domain gives

$$y_{t+k\delta} = \beta_0 + \beta_1 \lambda_1^k + \beta_2 \lambda_2^k + \dots + \beta_n \lambda_n^k \tag{4.3.9}$$

where

$k \geq 0$

Equation (4.3.9) is the response at time $k\delta$ to a step input at 0. The response to a step input can be split into two parts: the steady-state response and the transient response. The steady-state response is β_0 and the transient response is

$$y_{t+k\delta} - \beta_0 = \beta_1 \lambda_1^k + \beta_2 \lambda_2^k + \dots + \beta_n \lambda_n^k \tag{4.3.10}$$

The summation of the transient response from time zero to infinity is

$$\sum_{k=0}^{\infty} \left(y_{t+k\delta} - \beta_0 \right) = \sum_{k=0}^{\infty} \left(\beta_1 \lambda_1^k + \beta_2 \lambda_2^k + \dots + \beta_n \lambda_n^k \right)$$

$$= \frac{\beta_1}{1 - \lambda_1} + \frac{\beta_2}{1 - \lambda_2} + \dots + \frac{\beta_n}{1 - \lambda_n} \tag{4.3.11}$$

The roots with the largest effect on the transient response are the roots with the largest value of the following quantity:

$$\omega_j = \left| \frac{\beta_j}{1 - \lambda_j} \right| \tag{4.3.12}$$

The dominant roots are defined as the roots with the largest effect on the transient response, i.e., the roots with the largest ω_j computed from equation (4.3.12). (Section 4.3.5 shows that the largest root is not always the dominant root.) The dominant roots are used to determine the denominator of the reduced transfer function.

$$G_r(z) = \frac{z^m \sum_{j=0}^{m} d_j z^{-j}}{\prod_{j=1}^{m} \left(z - \bar{\lambda}_j\right)} = \frac{\sum_{j=0}^{m} d_j z^{-j}}{\prod_{j=1}^{m} \left(1 - \bar{\lambda}_j z^{-1}\right)} \tag{4.3.13}$$

where

$\bar{\lambda}_j$ = one of the m dominant roots

m = number of past time steps in reduced transfer function

Multiplying the m terms together in equation (4.3.13) results in

$$G_r(z) = \frac{\sum_{j=0}^{m} d_j z^{-j}}{\sum_{j=0}^{m} e_j z^{-j}} \tag{4.3.14}$$

where

$$e_0 = 1$$

$$e_1 = - \sum_{i=1}^{m} \bar{\lambda}_i$$

$$e_2 = \sum_{i=1}^{m-1} \left(\bar{\lambda}_i \sum_{j=i+1}^{m} \bar{\lambda}_j \right)$$

$$e_3 = - \sum_{i=1}^{m-2} \left[\bar{\lambda}_i \sum_{j=i+1}^{m-1} \left(\bar{\lambda}_j \sum_{k=j+1}^{m} \bar{\lambda}_k \right) \right]$$

$$\cdot$$

$$\cdot$$

$$\cdot$$

$$e_m = (-1)^m \prod_{i=1}^{m} \bar{\lambda}_i \tag{4.3.15}$$

As an alternative to equation (4.3.15), section 4.3.2 contains an algorithm for determining the e_j coefficients in equation (4.3.14).

The numerator of the reduced z-transfer function is determined by equating the power series expansion of the original w-transfer function, G(w), with the power series expansion of the reduced w-transfer function, $G_r(w)$, for terms of order of w^0 to w^m.

Next, equations for determining the denominator of the reduced w-transfer function will be presented. This will be followed by a description of equations for determining the numerator of the reduced w-transfer function.

Substituting the the bilinear transformation, equation (4.2.7), into equation (4.3.14) results in the following reduced w-transfer function:

$$G_r(w) = \frac{N_r(w)}{\sum_{j=0}^{m} e_j \left(\frac{1 - w}{1 + w}\right)^j} = \frac{\left(N_r(w)\right)(1 + w)^m}{\sum_{j=0}^{m} e_j (1 - w)^j (1 + w)^{m-j}} \tag{4.3.16}$$

where

$N_r(w)$ = function of complex variable w

Section 4.2.2 contains an algorithm for determining the $v_{i(j,m)}$ coefficients in the following equation:

$$(1 - w)^j (1 + w)^{m-j} = \sum_{i=0}^{m} v_{i(j,m)} w^i \tag{4.3.17}$$

Substituting equation (4.3.17) into equation (4.3.16) gives

$$G_r(w) = \frac{\left(N_r(w)\right)(1 + w)^m}{\sum_{j=0}^{m} e_j \left(\sum_{i=0}^{m} v_{i(j,m)} w^i\right)} = \frac{\left(N_r(w)\right)(1 + w)^m}{\sum_{i=0}^{m} \left(\sum_{j=0}^{m} e_j v_{i(j,m)}\right) w^i} \tag{4.3.18}$$

Setting the coefficient for w^0 in the denominator of the reduced w-transfer function, equation (4.3.18), equal to one results in.

$$G_r(w) = \frac{\sum_{i=0}^{m} \bar{d}_i w^i}{\sum_{i=0}^{m} \bar{e}_i w^i} \tag{4.3.19}$$

where

$$\bar{e}_i = \frac{\sum_{j=0}^{m} e_j v_{i(j,m)}}{\sum_{j=0}^{m} e_j v_{0(j,m)}} \tag{4.3.20}$$

There are three steps involved in determining the numerator of the reduced w-transfer function from the original transfer function coefficients and the denominator of the reduced w-transfer function. First, the bilinear transformation and z-transform theory are used to determine a w-transfer function from the original transfer function. Second, the power series expansion for the original w-transfer function is computed for terms of order w^0 to w^m. Third, the numerator of the reduced transfer function is determined by equating the power series expansion for the original w-transfer with the reduced transfer function for terms from w^0 to w^m. Sections (4.1.1) and (4.2.1) contain equations which can be used to determine the numerator of the reduced w-transfer function from the denominator of the reduced w-transfer function and the original transfer function.

After the numerator of the reduced w-transfer function is determined, equation (4.2.17), which is based upon the bilinear transformation and z-transform theory, can be used to determine the d_j coefficients for the reduced z-transfer function, equation (4.3.14), from the reduced w-transfer function. Finally, the reduced z-transfer function, i.e., equation (4.3.14), is transformed back into the time domain.

$$y_t = \sum_{j=0}^{m} d_j u_{t-j\delta} - \sum_{j=1}^{m} e_j y_{t-j\delta} \qquad (4.3.21)$$

Equation (4.3.21) is a reduced transfer function.

In summary, the following step-by-step procedure can be used to determine a reduced transfer function:

1) Use a root finding procedure to determine the roots of equation (4.3.6).

2) Use equations (4.3.8) and (4.3.12) to determine the m dominant roots.

3) Use equation (4.3.15) or the algorithm described in section 4.3.2 to determine the output transfer function coefficients from the m dominant roots.

4) Use equation (4.3.20) and the algorithm described in section 4.2.2 to determine the denominator of the reduced w-transfer function.

5) Use z-transform theory and the bilinear transformation to determine the w-transfer function from the original transfer function. [Use equations (4.2.9) and (4.2.10) along with the algorithm described in section 4.2.2 to obtain the coefficients in equation (4.2.8).]

6) Use equation (4.1.6) to determine the power series expansion of the w-transfer function for terms of order w^0 to w^m.

7) Use equation (4.1.9) to determine the numerator of the reduced w-transfer function from the denominator of the reduced w-transfer function and the power series expansion of the original w-transfer function.

8) Use equation (4.2.17) to determine the coefficients for current and past inputs in the reduced transfer function from the reduced w-transfer function.

4.3.2 Algorithm for Determining Output Coefficients

This section contains an algorithm which can be used to determine the output coefficients of the reduced transfer functions from m dominant roots of the original transfer function. The following equation relates the m dominant roots of the original transfer function with the output coefficients of the reduced transfer function:

$$\prod_{j=1}^{m} \left(1 - \lambda_j z^{-1} \right) = \sum_{j=0}^{m} e_j z^{-j} \qquad (4.3.22)$$

where

$\bar{\lambda}_j$ = one of the m dominant roots

The following algorithm can be used to determine the e_j coefficients in equation (4.3.14) or equation (4.3.22):

```
$e_0 = 1$
$e_1 = -\bar{\lambda}_1$
For i = 2 to m with a step size of 1
    $e_i = 0$
    For j = i to 1 with a step size of -1
        $e_j = e_j - e_{j-1} \bar{\lambda}_i$
    Next j
Next i
```

4.3.3 Example

To demonstrate dominant root model reduction, a reduced transfer function with one past time step (i.e., m=1) will be determined from a transfer function with two past time steps (i.e., n=2). For this example, the following transfer function will be reduced:

$$y_t = a_0\, u_t + a_1\, u_{t-\delta} + a_2\, u_{t-2\delta} - b_1\, y_{t-\delta} - b_2\, y_{t-2\delta}$$

$$= u_t + 0.5\, u_{t-\delta} + 0.1\, u_{t-2\delta} + 1.1\, y_{t-\delta} - 0.3\, y_{t-2\delta}$$

From equation (4.3.4), the z-transform of the output when the input is a unit step is

$$Y(z) = \frac{z\left(a_0\, z^2 + a_1\, z + a_2\right)}{(z - 1)\left(b_0\, z^2 + b_1\, z + b_2\right)} = \frac{z\left(z^2 + 0.5\, z + 0.1\right)}{(z - 1)\left(z^2 - 1.1\, z + 0.3\right)}$$

The quadratic equation can be used to determine the roots of equation (4.3.6).

$$\lambda_1, \lambda_2 = \frac{-b_1 \pm \sqrt{b_1^2 - 4\,b_2}}{2} = \frac{1.1 \pm \sqrt{(-1.1)^2 - 4\,(0.3)}}{2} = 0.6,\ 0.5$$

Using the roots determined from the quadratic equation, the z-transform of the output can be written as

$$Y(z) = \frac{a_0\,z^3 + a_1\,z^2 + a_2\,z}{\left(z - \lambda_0\right)\left(z - \lambda_1\right)\left(z - \lambda_2\right)} = \frac{z^3 + 0.5\,z^2 + 0.1\,z}{(z - 1)\,(z - 0.6)\,(z - 0.5)}$$

From equation (4.3.8), the β_j coefficients in equation (4.3.7) are

$$\beta_1 = \frac{a_0\,\lambda_1^2 + a_1\,\lambda_1 + a_2}{\left(\lambda_1 - \lambda_0\right)\left(\lambda_1 - \lambda_2\right)} = \frac{1\,(0.6)^2 + 0.5\,(0.6) + 0.1}{(0.6 - 1)\,(0.6 - 0.5)} = -19$$

$$\beta_2 = \frac{a_0\,\lambda_2^2 + a_1\,\lambda_2 + a_2}{\left(\lambda_2 - \lambda_0\right)\left(\lambda_2 - \lambda_1\right)} = \frac{1\,(0.5)^2 + 0.5\,(0.5) + 0.1}{(0.5 - 1)\,(0.5 - 0.6)} = 12$$

Using equation (4.3.12), the following ω_j quantities can be computed for the roots:

$$\omega_1 = \left|\frac{\beta_1}{1 - \lambda_1}\right| = \left|\frac{-19}{1 - 0.6}\right| = 47.5$$

$$\omega_2 = \left| \frac{\beta_2}{1 - \lambda_2} \right| = \left| \frac{12}{1 - 0.5} \right| = 24$$

ω_1 is larger than ω_2, therefore the dominant root is

$$\bar{\lambda}_1 = \lambda_1 = 0.6$$

Next, the coefficient in the denominator of the reduced z-transfer function is computed from equation (4.3.15).

$$e_1 = - \bar{\lambda}_1 = - 0.6$$

From equation (4.3.16), the reduced w-transfer function is

$$G_r(w) = \frac{N_r(w)\ (1 + w)}{e_0\ (1 + w) + e_1\ (1 - w)} = \frac{N_r(w)\ (1 + w)}{1 + w + (-0.6)\ (1 - w)} = \frac{N_r(w)\ (1 + w)}{0.4 + 1.6\ w}$$

Rearranging the reduced w-transfer function results in

$$G_r(w) = \frac{\bar{d}_0 + \bar{d}_1\ w}{1 + \bar{e}_1\ w} = \frac{\bar{d}_0 + \bar{d}_1\ w}{1 + 4w}$$

From section 4.2.3, the power series expansion for the original w-transfer is

$$G(w) = \bar{c}_0 + \bar{c}_1 w + \bar{c}_2 w^2 + \dots = 8 - 47 w + 236 w^2 + \dots$$

Numerator coefficients of the reduced w-transfer function are determined from equation (4.1.9).

$$\bar{d}_0 = \bar{c}_0 = 8$$

$$\bar{d}_1 = \bar{c}_1 + \bar{e}_1 \bar{c}_0 = -47 + 4\ (8) = -15$$

Equation (4.2.13) can be used to compute the reduced z-transfer function from the reduced w-transfer function.

$$\begin{aligned} G_r(z) &= \frac{\bar{d}_0\left(1 + z^{-1}\right) + \bar{d}_1\left(1 - z^{-1}\right)}{\bar{e}_0\left(1 + z^{-1}\right) + \bar{e}_1\left(1 - z^{-1}\right)} \\ &= \frac{8\left(1 + z^{-1}\right) - 15\left(1 - z^{-1}\right)}{1 + z^{-1} + 4\left(1 - z^{-1}\right)} \\ &= \frac{-1.4 + 4.6\ z^{-1}}{1 - 0.6\ z^{-1}} \end{aligned}$$

Transforming the reduced z-transfer function into the time domain gives

$$y_t = -1.4\ u_t + 4.6\ u_{t-\delta} + 0.6\ y_{t-\delta}$$

4.3.4 Multiple-Input Transfer Functions

This section extends the single-input model reduction method described in section 4.3.1 to multiple-input transfer functions. First, a step-by-step procedure for reducing multiple-input transfer functions is presented. Then, a discussion of dominant root model reduction for building element transfer functions and CRTF's will be presented.

The following procedure can be used to compute reduced multiple-input transfer functions:

1) Use a root finding procedure to determine the roots of equation (4.3.6). The roots in equation (4.3.6) are the same for all inputs because equation (4.3.6) is based upon the transfer function coefficients for outputs.

2) Use equation (4.3.8) to determine the n β_j terms in equation (4.3.8) for every input. (n is equal to the number of past time steps in the original transfer function.)

3) Use equation (4.3.12) to determine the n ω_j values for every input.

4) Select the dominant roots for every input. Let m equal the total number of dominant roots for all inputs.

5) Use equation (4.3.15) or the algorithm described in section 4.3.2 to determine the transfer function coefficients for past outputs from the m dominant roots.

6) Use equation (4.3.20) and the algorithm described in section 4.2.2 to determine the denominator of the reduced w-transfer function.

7) Use z-transform theory and the bilinear transformation to determine single-input w-transfer functions for every input from the original transfer function, i.e., use equations (4.2.9) and (4.2.10) and the algorithm described in section 4.2.2 to determine the coefficients in equation (4.2.8) for every input.

8) Use equation (4.1.6) to determine the power series expansion of the w-transfer functions for terms of order w^0 to w^m. (A power series expansion must be computed for every input.)

9) Use equation (4.1.9) to determine the numerator of the reduced w-transfer functions from the denominator of the reduced w-transfer functions and the power series expansion of the original w-transfer functions.

10) Use equation (4.2.17) to determine the coefficients for current and past inputs in the reduced multiple-input transfer function from the reduced single-input w-transfer functions.

When using the methods of Stephenson and Mitalas (1971), Ceylan and Myers (1980), and Hittle (1981) to determine transfer functions for building elements, the roots are determined before the output transfer function coefficients are computed. Thus, a root finding procedure is not needed when computing reduced transfer functions for building elements, i.e., step one of the step-by-step procedure could be eliminated.

Step one of the step-by-step procedure involved determining the roots of a transfer function. Determining the roots of a CRTF may be a numerically difficult problem. It is possible to avoid this numerical problem. First, determine the roots of the building element transfer functions which relate the heat flow to the star node with the inputs. The roots of the combined transfer function (i.e., the transfer function which relates the heat flow to the star node with the inputs) are equal to the roots for the individual building element transfer functions. Then, dominant root model reduction can be used to determine a reduced transfer function which relates heat flow to the star node with the inputs. Finally, the transfer function which relates heat flow to the star node with the inputs can be converted into a CRTF.

4.3.4 Applications

To test dominant root model reduction for building elements with a wide range of properties, reduced transfer functions for the following ASHRAE (1977) building elements were computed:

1) Exterior Wall 4 (4" face brick, air space, and 8" high).
2) Exterior Wall 25 (frame wall with 4" brick veneer).
3) Exterior Wall 28 (metal curtain wall with 2" of insulation).
4) Exterior Wall 36 (frame wall with 3" insulation).
5) Exterior Wall 54 (4"face brick, air space, and 12" high weight concrete).

For all of the building elements tested, dominant root model reduction was used to obtain a reduced set of coefficients which closely modeled the response of the full set of coefficients.

Table 4.2 contains transfer function coefficients which were generated from Mitalas and Arsenault's program for ASHRAE wall 25. Table 4.3 contains the roots and values of ω_j for ASHRAE wall 25. Table 4.3 shows that the first root is the dominant root for a step change in outdoor temperature and the second root is the dominant root for a step change in indoor temperature. These two dominant roots were used to obtain the reduced transfer function coefficients in table 4.4. Figure 4.4 is a graph of the response to a 1°F step change in outdoor temperature for the full set of coefficients, the reduced set of coefficients, and a dropped set of coefficients. Figure 4.5 is a similar graph for a 1°F step change in indoor temperature. Both these graphs demonstrate that the reduced set of coefficients closely match the response of the full set of coefficients and the dropped set of coefficients produces a response different than the full set of coefficients.

To demonstrate that the second root is dominant for the indoor temperature, reduced transfer functions were computed with both the largest root and the dominant root. Table 4.5 contains these reduced transfer functions. Figure 4.6 is a graph of the response to a 1°F step change in temperature for the full set of coefficients and reduced sets of coefficients which were obtained with both the dominant root and the largest root. The response for the reduced transfer function with the dominant root is much closer to the response of the full set of coefficients than the response for the reduced transfer function with the largest root.

Dominant root model reduction can also be used to determine reduced multi-input CRTF's. Appendix F contains a reduced multiple-input CRTF for the eight surface room considered in section 3.3.2. Figure 4.7 shows the response to a 1°F step change in outdoor temperature for the original and reduced CRTF. This figure demonstrates that dominant root model reduction can be used to significantly reduce the number of coefficients in CRTF's. Similar figures for step changes in indoor temperature, solar radiation gains, and radiation gains from people equipment and lights are shown in appendix F.

Table 4.2 Transfer function coefficients for ASHRAE wall 25.

j	a_j (Btu/hr-ft^2-°F)	b_j (Btu/hr-ft^2-°F)	c_j
0	0.00037379	-0.72085727	1.00000000
1	0.00823296	1.21752729	-1.03054437
2	0.00983592	-0.53986228	0.20122048
3	0.02352481	0.00235428	-0.00726122
4	0.00001128	-0.00003896	0.00000263

Table 4.3 Roots and ω_j values for ASHRAE wall 25.

		ω_j	
j	λ_j	Outdoor Temperature	Indoor Temperature
1	0.78640705	**0.718**	0.066
2	0.19749089	0.062	**0.722**
3	0.04628098	0.019	0.002
4	0.00036547	0.002	0.004

Table 4.4 Reduced transfer function coefficients for ASHRAE wall 25.

j	a_j (Btu/hr-ft^2-°F)	b_j (Btu/hr-ft^2-°F)	c_j
0	0.00240575	-0.71992176	1.00000000
1	0.00226031	1.18110817	-0.98389794
2	0.01600402	-0.48185666	0.15530823

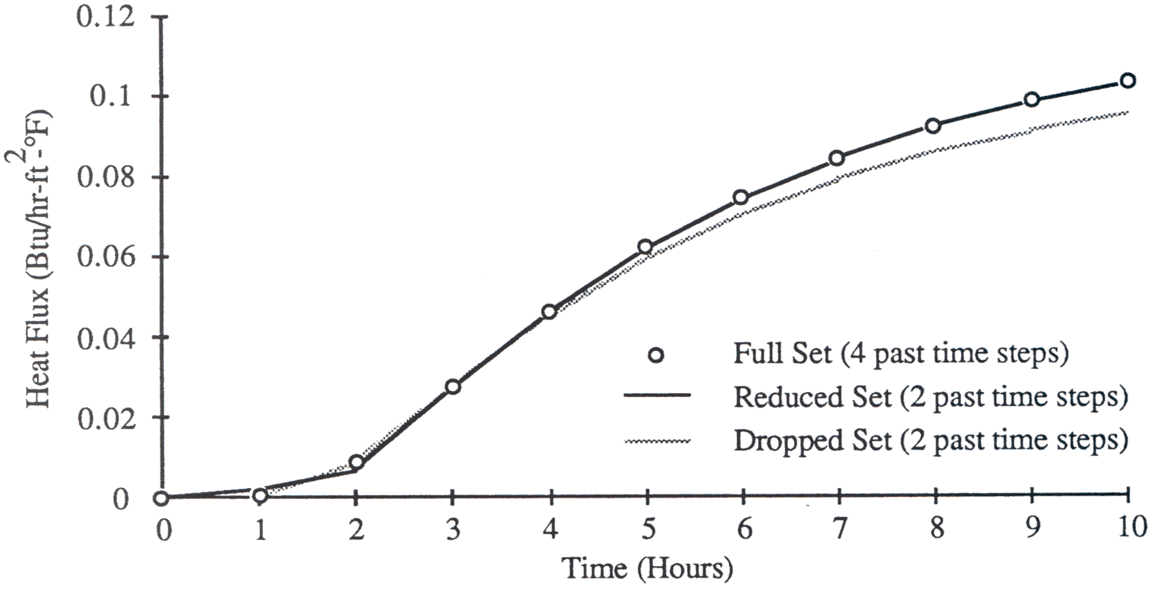


Figure 4.4 Response to 1°F step change in outdoor temperature for ASHRAE wall 25.

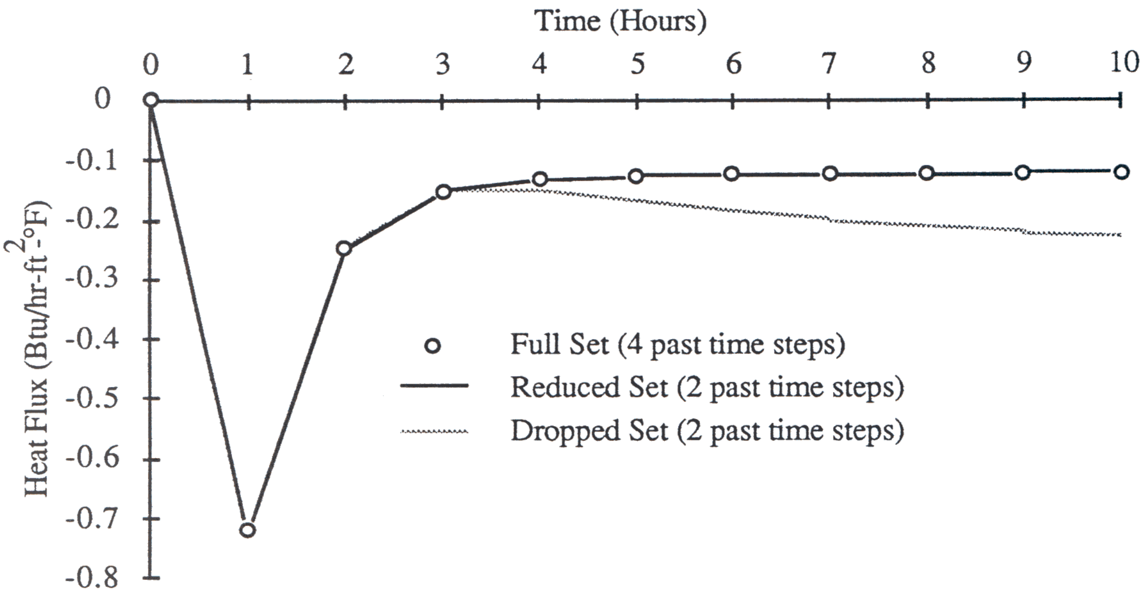


Figure 4.5 Response to 1°F step change in indoor temperature for ASHRAE wall 25.

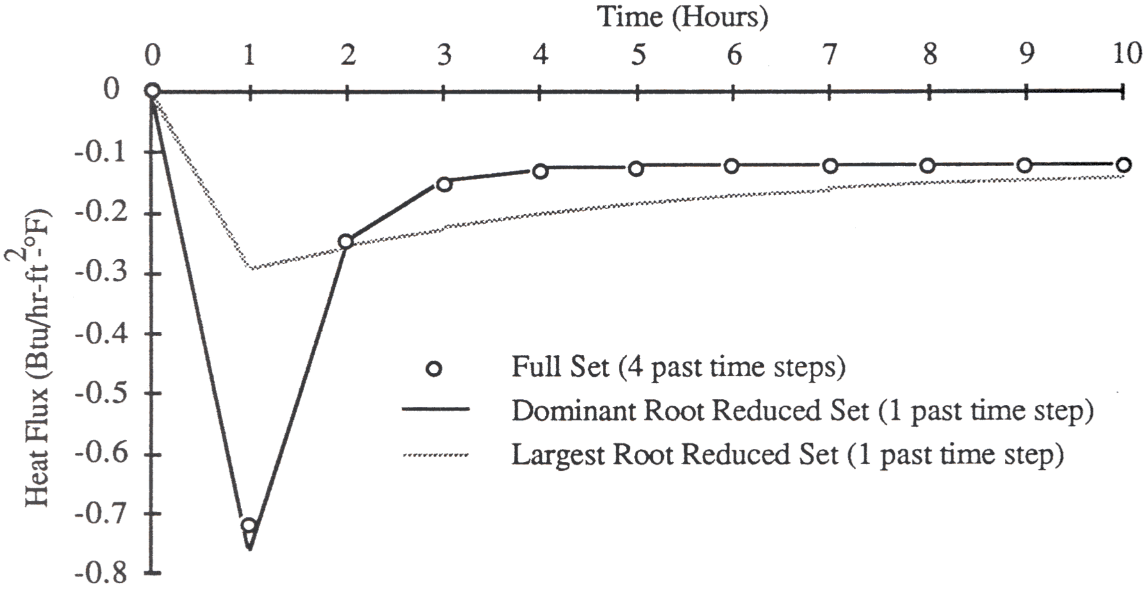


Figure 4.6 Response to 1°F step change in outdoor temperature for ASHRAE wall 25.

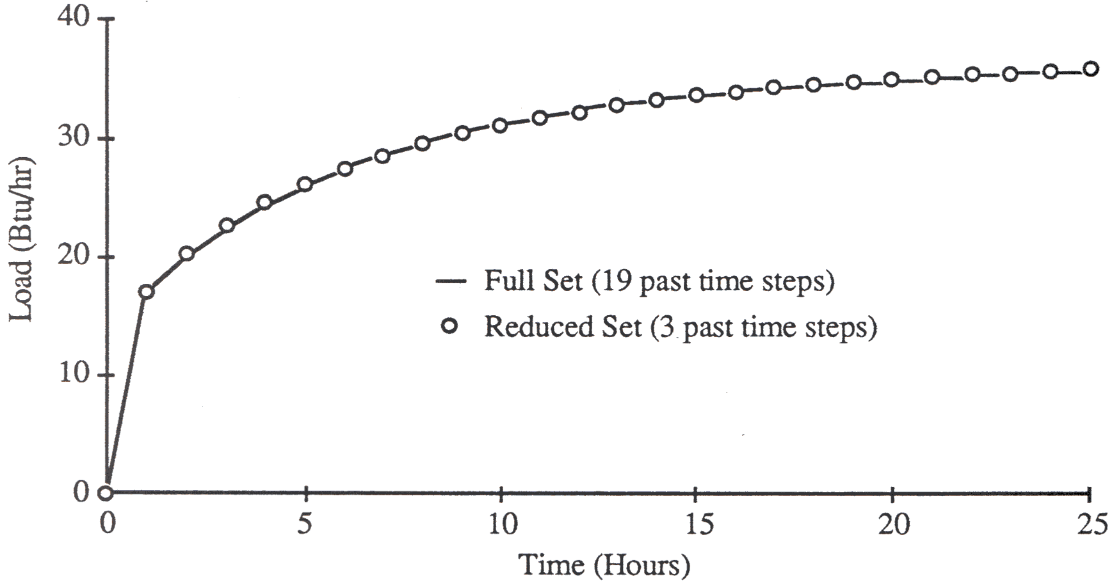


Figure 4.7 Response to 1°F step change in outdoor temperature for the eight surface room.

Table 4.5 Reduced transfer function coefficients for ASHRAE wall 25.

	Dominant Root		Largest Root	
j	b_j (Btu/hr-ft^2-°F)	c_j	b_j (Btu/hr-ft^2-°F)	c_j
0	-0.75827	1.00000	-0.29031	1.00000
1	0.66150	-0.19749	0.26456	-0.78641

4.4 TESTS FOR DETERMINING MODEL ORDER

This section is concerned with determining the "correct" model order of reduced transfer functions, i.e., the minimum number of past time steps required to accurately model the heat transfer processes in a building with a reduced transfer function. If the number of past time steps in the reduced transfer function is too small, then the reduced transfer function will not accurately model the the heat transfer processes in a building. If the number of past time steps in a reduced transfer function is too large, then performing a simulation with the reduced transfer function will require unnecessary computational effort.

The following procedure can be used to determine reduced transfer functions of the "correct" model order:

1) Set m = 1. (m is the number of past time steps in the reduced transfer function.)

2) Determine a reduced transfer function with m past time steps.

3) Use a test to compare the reduced transfer function with m past time steps with the original transfer function. If the reduced transfer function accurately models the heat transfer processes, then the reduced transfer function with m past time steps is of the "correct" order. If the reduced transfer function does not accurately model the heat transfer processes, then set m = m + 1 and return to step 2.

Next, three different methods for comparing reduced transfer functions with the original transfer function will be presented. First, a graphical method for determining model order will be presented. Then, two analytical methods for determining model order will be presented and compared.

A graph of the response to a step input can be used to compare reduced transfer functions with the original transfer function. Based on this graph, a person can select the model order for the lowest ordered reduced transfer function which closely reproduces the response of the full set of coefficients. There are two disadvantages to using this graphical method for determining model order in a building simulation program. First, the model order would be dependent upon the user and two different users may select different model orders. Second, most users of building simulation programs would not want to spend the time to determine the "correct" model order.

The following dimensionless parameter can be computed from the step responses for the original and reduced transfer functions:

$$E_{max} = \frac{\text{Maximum}\,|\text{error}|}{\text{Maximum}\,|\text{output}|} \tag{4.4.1}$$

The error in equation (4.4.1) is equal to the difference between the output for the original transfer function and the reduced transfer function. A tolerance limit can be chosen to compare with the dimensionless parameter E_{max}. If E_{max} is less than the tolerance limit, then the reduced transfer function accurately models the heat transfer processes.

For heat transfer problems, the absolute value of the maximum output to a step input is the maximum of the absolute value of the steady-state output or the absolute value of the output at the initial time step, i.e., the time at which the step is applied. The output at the initial time step is equal to the first input coefficient in the original transfer function. For the model reduction methods presented in this chapter, the steady-state output can be determined directly from the reduced transfer function coefficients. Thus, the absolute value of the maximum response to a unit step input can be computed directly from the transfer function coefficients.

$$E_{max} = \frac{\text{Maximum} \left| \text{error} \right|}{\text{Maximum}\left(\left| a_0 \right| , \left| \frac{\sum_{j=0}^{m} d_j}{\sum_{j=0}^{m} e_j} \right| \right)} \qquad (4.4.2)$$

where

a_j = input coefficient for original transfer function

d_j = input coefficient for the reduced transfer function

e_j = output coefficient for the reduced transfer function

The model reduction methods presented in this chapter ensure the correct steady-state response to a step input for reduced transfer functions. Therefore, errors do not have to be computed for a large number of time steps because the responses for the original and reduced transfer functions approach each other. A disadvantage of this test is that a simulation is required to compare the original and reduced transfer functions.

The following dimensionless parameter can be computed directly from the transfer function coefficients:

$$E_0 = \frac{\left| a_0 - d_0 \right|}{\text{Maximum}\left| \text{output} \right|} = \frac{\left| a_0 - d_0 \right|}{\text{Maximum}\left(\left| a_0 \right| , \left| \frac{\sum_{j=0}^{m} d_j}{\sum_{j=0}^{m} e_j} \right| \right)} \tag{4.4.3}$$

Graphs of E_{max} versus E_0 are shown in figure 4.8 for the Padé approximation and bilinear transformation and figure 4.9 for dominant root model reduction. These graphs show that there is a linear relationship between the logarithm of E_{max} and the logarithm of E_0. Figure 4.7 contains 97 points which were computed from transfer functions for eight standard ASHRAE walls and the three and eight surface rooms. Figure 4.8 contains 34 points which were computed from transfer functions for five standard ASHRAE walls.

Linear regression was used to determine the following relationship between E_{max} and E_0 for the Padé approximation and bilinear transformation:

$$\log_{10} E_0 = 1.15 \log_{10} E_{max} + 0.155 \tag{4.4.4}$$

With dominant root model reduction the relationship between E_0 and E_{max} is

$$\log_{10} E_0 = 1.10 \log_{10} E_{max} + 0.099 \qquad (4.4.5)$$

The sample standard deviation of the residuals was 0.188 for equation (4.4.4) and 0.103 for equation (4.4.5).

Taking the antilogarithm of equation Equation (4.4.4) gives

$$E_0 = 1.43\, E_{max}^{1.15} \qquad (4.4.6)$$

and taking the antilogarithm of equation (4.4.5) gives

$$E_0 = 1.26\, E_{max}^{1.10} \qquad (4.4.7)$$

Equations (4.4.6) and (4.4.7) can be used to compute a tolerance limit for E_0 from a tolerance limit for E_{max}. The tolerance limit for E_0 can be compared with a E_0 value computed from the reduced transfer functions and the original transfer function. If the computed E_0 parameter is less than the tolerance limit for E_0, then the reduced transfer function will accurately model the heat transfer processes.

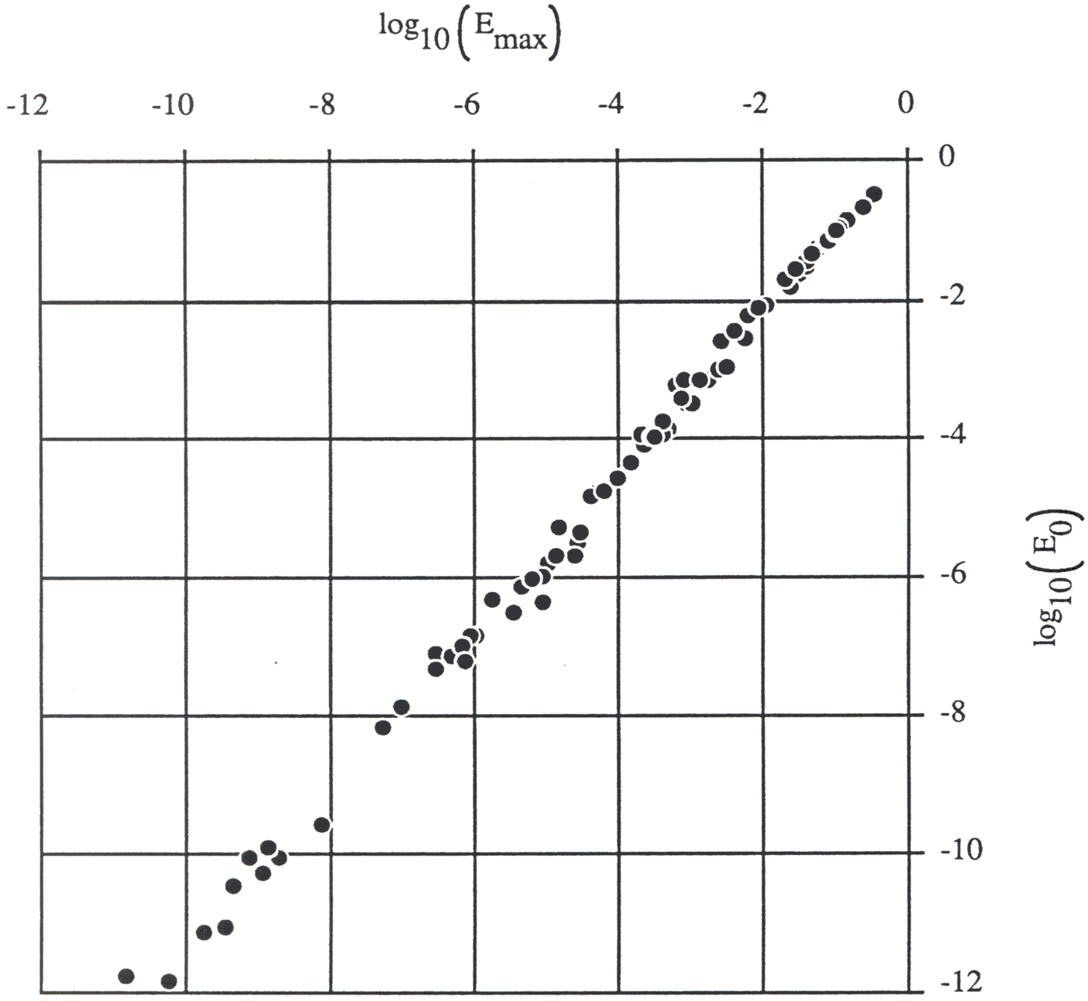


Figure 4.8 E_0 versus E_{max} for the Padé approximation and bilinear transformation.

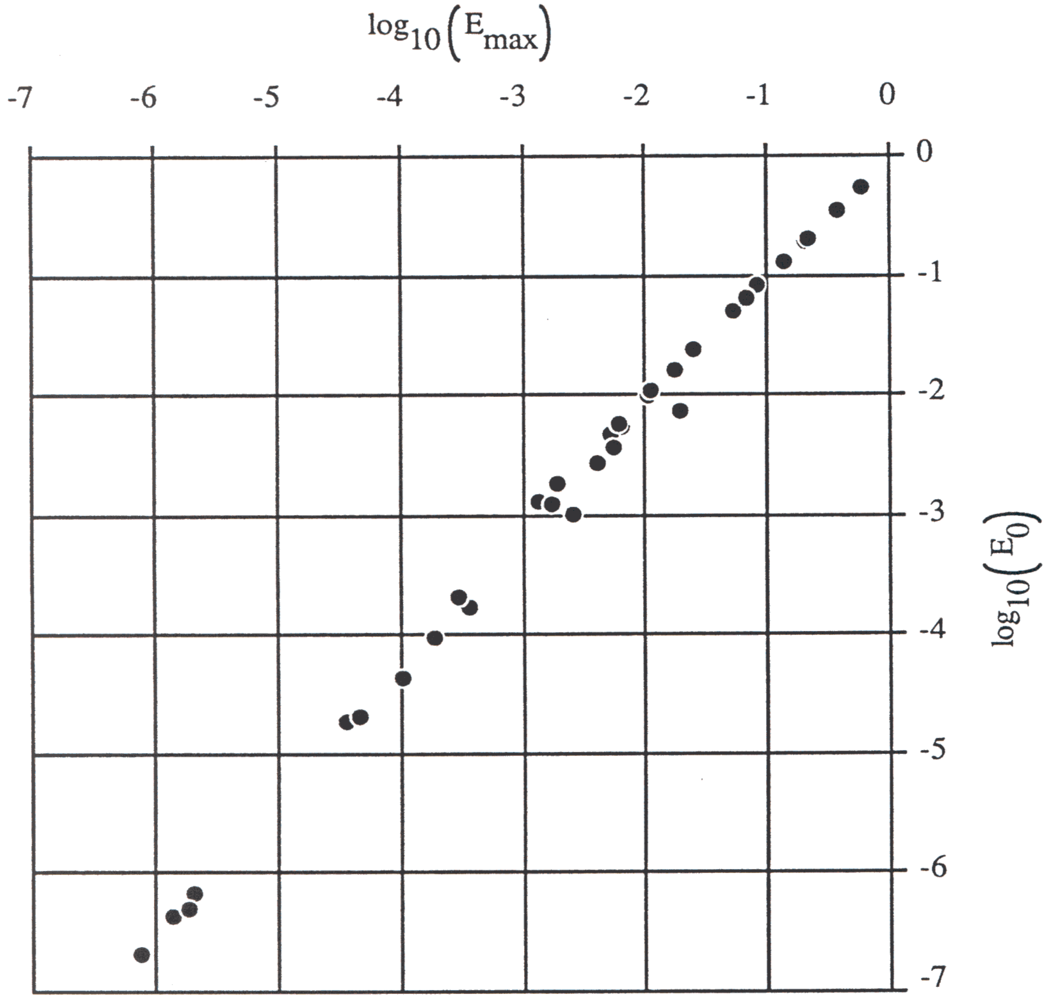


Figure 4.9 E_0 versus E_{max} for dominant root model reduction.

4.5 COMPUTATIONAL EFFORT

Section 3.4 showed that the computational effort of energy balance simulations was not significantly different than the computational effort of CRTF simulations. This section shows that model reduction methods can be used to significantly reduce the computational effort of CRTF simulations.

Table 4.6 contains the number of multiplications required per time step for energy balance simulations of view factor networks with time independent A matrices, multiple–input CRTF simulations, single-input CRTF's simulations with reduced coefficients, and multiple-input CRTF's simulations with reduced coefficients for the three and eight surface rooms previously described. The Padé approximation and bilinear transformation were used to determine the reduced single-input transfer function coefficients. Dominant root model reduction was used to determine the reduced multiple-input transfer function coefficients. Appendix F contains tables of the original and reduced transfer function coefficients and graphs of the response to step inputs for the original and reduced transfer functions for the eight surface room. For the methods compared in table 4.6, the number of additions required per time step is close to the number of multiplications required per time step. Table 4.6 demonstrates the computational savings of using reduced CRTF's increases with the complexity of the zone, i.e., the larger the number of surfaces the greater the computation savings.

Table 4.6 Number of multiplications required per time step for an energy balance simulation and CRTF simulations with full and reduced sets of coefficients.

	Multiplications per Time Step			
	Energy Balance	Multiple-Input CRTF	Reduced Single-Input CRTF	Reduced Multiple-Input CRTF
3 Surface Room	35	38	20	14
8 Surface Room	152	109	20	19

Sowell and Walton (1980) determined that the execution times are similar for energy balance simulations with time dependent A matrices (i.e., time varying convection coefficients) and an advanced weighting factor program [i.e., DOE 2.1 (1980)] which assumes heat transfer processes are linear and time-invariant. Thus, the computational effort of energy balance simulations with time independent A matrices should be less than the computational effort of DOE 2.1, and as shown in table 4.6 the computational effort of CRTF simulations with reduced coefficients is far less than the computational effort of energy balance simulations with time independent A matrices.

4.6 SUMMARY

This chapter was concerned with the development of model reduction methods for reducing the number of coefficients in transfer functions. These methods can be used to significantly reduce the computational effort of CRTF simulations. A procedure for determining the minimum number of past time steps required to accurately model the heat transfer processes with a reduced transfer function was also presented.

CHAPTER 5
Conclusion

This thesis has presented fast and accurate methods for determining the heating and cooling loads in buildings. These methods could be used to significantly reduce the effort of computing loads in building simulation programs. This would allow engineers and architects to investigate a number of building designs at a much lower cost.

For the eight surface room described in chapter 3, the computational effort of performing a simulation with a reduced comprehensive room transfer function (CRTF) is one-seventh the computational effort of an energy balance simulation. The following procedure could be used to develop reduced CRTF's:

1) **Determine reduced transfer functions for building elements.**

   For one-dimensional heat transfer through multi-layered slabs, use the method of Mitalas and Stephenson (1971) or Hittle (1981) to determine transfer functions. For building elements with parallel heat transfer paths, use the equations presented in section 3.1 to combine transfer functions. For multi-dimensional heat transfer, use the method of Ceylan and Myers (1980) or the procedure presented in chapter 2 to determine transfer functions. Use the model reduction methods described in chapter 4 to determine reduced transfer functions for the building elements.

2) **Determine the resistances in a star network.**

Section 3.2 presented a method for determining the resistances in a star network from the resistances in a network which uses view factors to model long-wave radiation exchange. Locations of room surfaces are needed to compute view factors. Carroll (1980) developed a mean radiant temperature (MRT) network which accurately models the heat flows in rooms. The MRT network is used in TARP [Walton (1983)]. There are two advantages of the MRT network over a view factor (i.e., a network which uses view factors). First, no information concerning the location of surfaces in a room is required. Second, it is easier to include furnishings in a MRT network [Walton (1984)]. The resistances in a star network could also be obtained from the resistances in a MRT network.

3) **Determine a comprehensive room transfer function.**

Section 3.3 presented equations for combining transfer functions for room models which use a star network. The equations presented in section 3.3 were developed for solar gains from one direction. These equations could easily be extended to account for solar radiation gains from multiple directions. Also, any other significant inputs (e.g., infiltration) should be included in a CRTF.

4) **Determine a reduced set of CRTF coefficients.**

Use the model reduction methods described in chapter 4 to determine a reduced set of CRTF coefficients.

APPENDIX A
Calculation of the Exponential Matrix

The calculation of the exponential matrix is an important step in the calculation of transfer functions from a system of first order differential equations with constant coefficients. There are a number of different methods available for calculating the exponential of a matrix. Moler and Van Loan (1978) have compared 19 different algorithms for calculating the exponential of a matrix and have concluded that the power series expansion with scaling and squaring is one of the most effective methods. This method is described below.

The approximation

$$e^{A\delta} \approx \mathbf{I} + A\delta + \frac{A^2\delta^2}{2!} + \frac{A^3\delta^3}{3!} + \ldots + \frac{A^L\delta^L}{L!} \qquad \text{(A.1)}$$

for the exponential matrix is obtained by truncating the power series expansion for the exponential matrix, equation (2.1.4), after L terms. A criteria is required for determining the number of terms to keep. Cadzow and Martens (1970) give the following relation for calculating L:

$$L = \text{minimum of } \{\ 3 \,\|\, A\delta \,\|_{\infty} + 6\} \text{ or } 100 \qquad \text{(A.2)}$$

$\| A\delta \|_\infty$ is a matrix row norm [Atkinson (1978)]. The matrix row norm is calculated by

$$\| A\delta \|_\infty = \underset{1 \le i \le n}{\text{Maximum}} \sum_{j=1}^{n} | a_{ij}\, \delta | \tag{A.3}$$

Equations (A.1) and (A.2) can be used to calculate the exponential of a matrix with at least six digits of accuracy. From equation (A.2) it can be seen that the number of terms in the truncated power series expansion increases and thus the computational effort increases as the matrix norm of $A\delta$ increases. Scaling and squaring can be used to reduce the computational effort when the matrix norm of $A\delta$ is large.

The following steps can be used to calculate the exponential matrix by a truncated power series expansion with scaling and squaring:

1) Use equation (A.3) to calculate $\| A\delta \|_\infty$

2) Find the smallest integer k such that $2^k \ge \| A\delta \|_\infty$

3) Divide all entries in the matrix $A\delta$ by 2^k

4) Determine L from equation (A.2) for $\left(\dfrac{A\delta}{2^k}\right)$

5) Calculate $e^{\left(\frac{A\delta}{2^k}\right)}$ from equation (A.1)

6) Square $e^{\left(\frac{A\delta}{2^k}\right)}$ k times to obtain $e^{A\delta}$

APPENDIX B

Integration of Integrals in Equation 2.1.7

This appendix contains the steps for evaluating the two integrals in equation 2.1.7. Substituting the power series expansion for $e^{-A\alpha}$, equation (2.1.3), into the first integral in equation (2.1.7) results in

$$\int_0^{\delta} e^{A(\delta - \alpha)}\, d\alpha = e^{A\delta}\left[\int_0^{\delta}\left(I - A\alpha + \frac{A^2\alpha^2}{2!} + \ldots\right) d\alpha\right] \tag{B.1}$$

Integrating the power series expansion for $e^{-A\alpha}$ term by term and substituting in the limits of integration gives

$$\int_0^{\delta} e^{A(\delta - \alpha)}\, d\alpha = \left(I + A\,\delta + \frac{A^2\delta^2}{2!} + \ldots\right)\left(I\delta - \frac{A\,\delta^2}{2} + \frac{A^2\delta^3}{3!} - \ldots\right) \tag{B.2}$$

Multiplying the two power series in equation (B.2) and combining common terms results in

$$\int_0^{\delta} e^{\mathbf{A}(\delta - \alpha)} d\alpha = \mathbf{I}\delta + \frac{\mathbf{A}\,\delta^2}{2!} + \frac{\mathbf{A}^2\,\delta^3}{3!} + \frac{\mathbf{A}^3\,\delta^4}{4!} + \ldots \tag{B.3}$$

Equation (B.3) can be rewritten as

$$\begin{aligned}\int_0^{\delta} e^{\mathbf{A}(\delta - \alpha)}\, d\alpha &= \mathbf{A}^{-1}\left(\mathbf{I} + \mathbf{A}\delta + \frac{\mathbf{A}^2\,\delta^2}{2!} + \frac{\mathbf{A}^3\,\delta^3}{3!} + \ldots\right) - \mathbf{A}^{-1} \\ &= \mathbf{A}^{-1}\left(e^{\mathbf{A}\delta} - \mathbf{I}\right) \end{aligned} \tag{B.4}$$

Next, the second integral in equation (2.1.7) will be determined by following a similar procedure.

$$\begin{aligned}\int_0^{\delta} \alpha\, e^{\mathbf{A}(\delta - \alpha)}\, d\alpha &= e^{\mathbf{A}\delta}\left[\int_0^{\delta}\left(\mathbf{I}\alpha - \mathbf{A}\,\alpha^2 + \frac{\mathbf{A}^2\,\alpha^3}{2!} - \ldots\right) d\alpha\right] \\ &= \left(\mathbf{I} + \mathbf{A}\delta + \frac{\mathbf{A}^2\,\delta^2}{2!} + \ldots\right)\left(\frac{\mathbf{I}\,\delta^2}{2} - \frac{\mathbf{A}\,\delta^3}{3} + \frac{\mathbf{A}^2\,\delta^4}{8} - \ldots\right) \\ &= \frac{\mathbf{I}\,\delta^2}{2!} + \frac{\mathbf{A}\,\delta^3}{3!} + \frac{\mathbf{A}^2\,\delta^4}{4!} + \frac{\mathbf{A}^3\,\delta^5}{5!} + \ldots \\ &= \mathbf{A}^{-1}\mathbf{A}^{-1}\left(\mathbf{I} + \mathbf{A}\,\delta + \frac{\mathbf{A}^2\,\delta^2}{2!} + \ldots\right) - \mathbf{A}^{-1}\mathbf{A}^{-1} - \mathbf{A}^{-1}\delta \\ &= \mathbf{A}^{-1}\mathbf{A}^{-1}\left(e^{\mathbf{A}\delta} - \mathbf{I}\right) - \mathbf{A}^{-1}\delta \end{aligned} \tag{B.5}$$

Appendix C
Leverrier's Algorithm

Wiberg (1971) presents a proof of Leverrier's algorithm for calculating the inverse of the (**FI** - **Φ**) matrix. Leverrier's algorithm for calculating the e scaler constants and **R** matrices in equation (2.1.18) consists of the following sequential relationships:

$$
\begin{array}{ll}
\mathbf{R}_0 = \mathbf{I} & e_1 = - \dfrac{\mathrm{Trace}\left(\mathbf{\Phi}\,\mathbf{R}_0\right)}{1} \\[2ex]
\mathbf{R}_1 = \mathbf{\Phi}\ \mathbf{R}_0 + e_1\ \mathbf{I} & e_2 = - \dfrac{\mathrm{Trace}\left(\mathbf{\Phi}\,\mathbf{R}_1\right)}{2} \\[2ex]
\mathbf{R}_2 = \mathbf{\Phi}\ \mathbf{R}_1 + e_2\ \mathbf{I} & e_3 = - \dfrac{\mathrm{Trace}\left(\mathbf{\Phi}\,\mathbf{R}_2\right)}{3} \\
\quad | & \quad | \\
\quad | & \quad | \\
\quad | & \quad | \\
\mathbf{R}_{n-1} = \mathbf{\Phi}\ \mathbf{R}_{n-2} + e_{n-1}\ \mathbf{I} & e_n = - \dfrac{\mathrm{Trace}\left(\mathbf{\Phi}\,\mathbf{R}_{n-1}\right)}{n}
\end{array}
$$

The trace of a matrix is equal to the sum of the diagonal elements of the matrix.

Appendix D
Multiple Real Roots

Section 4.3 presented a model reduction method (i.e., dominant root model reduction) which required the roots of the z-transfer function to be real, distinct, and between zero and one. When combining transfer functions or computing transfer functions from a finite-difference or finite-element model, the resulting z-transfer function may have multiple roots. Reduced transfer functions with distinct roots can be determined by following a procedure similar to dominant root model reduction, i.e., use the procedure in section 4.3 to determine reduced transfer functions, but determine the output coefficients with the distinct roots. The following step-by-step procedure can be used to determine reduced transfer functions with distinct roots:

1) Use a root finding procedure to determine the roots of equation (4.3.6). The roots in equation (4.3.6) are the same for all inputs because equation (4.3.6) is based upon the transfer function coefficients for outputs.

2) Determine the number of distinct roots. Let m equal the number of distinct roots.

3) Use equation (4.3.15) or the algorithm described in section 4.3.2 to determine the transfer function coefficients for past outputs from the m distinct roots.

4) Use equation (4.3.20) and the algorithm described in section 4.2.2 to determine the denominator of the reduced w-transfer function.

5) Use z-transform theory and the bilinear transformation to determine single-input w-transfer functions for every input from the original transfer function, i.e., use equations (4.2.9) and (4.2.10) and the algorithm described in section 4.2.2 to determine the coefficients in equation (4.2.8) for every input.

6) Use equation (4.1.6) to determine the power series expansion of the w-transfer functions for terms of order w^0 to w^m. (A power series expansion must be computed for every input.)

6) Use equation (4.1.9) to determine the numerator of the reduced w-transfer functions from the denominator of the reduced w-transfer functions and the power series expansion of the original w-transfer functions.

7) Use equation (4.2.17) to determine the coefficients for current and past inputs in the reduced multiple-input transfer function from the reduced single-input w–transfer functions.

When combining two identical transfer functions, the resulting combined transfer function will have multiple real roots. To demonstrate this, the following transfer functions will be combined by using the equations described in section 3.1:

$$q''_{t,a} = T_t + T_{t-\delta} + 0.5\, q''_{t-\delta,a} \qquad \text{(D.1)}$$

$$q''_{t,b} = T_t + T_{t-\delta} + 0.5\, q''_{t-\delta,b} \qquad \text{(D.2)}$$

The heat flux from both walls is

$$q''_t = \frac{A_a}{A_a + A_b} q''_{t,a} + \frac{A_b}{A_a + A_b} q''_{t,b} \tag{D.3}$$

The transfer function for the heat flux from both walls is

$$q''_{t,\chi} = T_t + 0.5\, T_{t-\delta} - 0.5\, T_{t-2\delta} + q''_{t-\delta,\chi} - 0.25\, q''_{t-2\delta,\chi} \tag{D.4}$$

Taking the z-transformation of equation (D.4) results in the following z-transfer function:

$$\frac{q''(z)}{T(z)} = \frac{z^2 + 0.5\, z - 0.5}{z^2 - z + 0.25} \tag{D.5}$$

Equation (D.5) can be rewritten as

$$\frac{q''(z)}{T(z)} = \frac{\left(z + 1\right)\left(z - 0.5\right)}{\left(z - 0.5\right)^2} \tag{D.6}$$

Thus, the z-transfer function, equation (D.6), has multiple real roots at 0.5. One of these roots can be canceled with the same factor in the numerator. This results in the following reduced z-transfer function:

$$\frac{q''(z)}{T(z)} = \frac{z + 1}{z - 0.5} \qquad (D.7)$$

Transforming the z-transfer function described by equation (D.7) back into the time domain gives

$$q''_{t,\chi} = T_t + T_{t-\delta} + 0.5\, q''_{t-\delta,\chi} \qquad (D.8)$$

Thus, the reduced transfer function for the combined heat flux is identical to the transfer functions for the heat flux from the individual walls. The step-by-step procedure for determining reduced transfer functions with distinct roots could have been used to determine the reduced transfer function, equation (D.8), from the combined transfer function, equation (D.4).

The response (e.g., nodal temperatures) to a step change in inputs is equal to a summation of exponentials for finite-element or finite-difference networks which have a convective boundary or specified temperature boundary [Myers (1981)]. For a transfer function with distinct roots, the response to a step change in inputs is equal to a summation of exponentials. Therfore, multiple roots can be canceled for transfer functions determined from finite-difference or finite-element models.

APPENDIX E

Computational Effort of Energy Balance Method

Chapters 3 and 4 compared the computational effort of the energy balance method [ASHRAE (1985)] with Comprehensive Room Transfer Functions. This appendix contains equations for computing the number of multiplications or divisions required per time step for an energy balance simulation.

The energy balance method is based upon performing an energy balance for interior surfaces in a room and the room air. For a room with N surfaces, these energy balances are formulated in terms of the following matrix equation:

$$\mathbf{Z}\ \mathbf{T} = \mathbf{X} \tag{E.1}$$

where **Z** is a (N+1) by (N+1) matrix; **T** is a vector of temperatures with all rows equal to an interior surface temperature, except the last row which in the temperature of the room air; and **X** is a vector of inputs. When the heat transfer processes are linear and time invariant, the **Z** matrix is inverted one time before the simulation. At each time step in the simulation the inverted **Z** matrix is multiplied by the **X** matrix to obtain the N surface temperatures and the air temperature in the room. The net heating or cooling load for the room is then computed by determining the convective heat transfer from the surfaces to the air in the room. At each time step in a simulation the **X** vector and the current heat flow across the interior surface of each building element must also be computed.

across the interior surface of each building element must also be computed.

When the indoor air temperature is an input, the number of multiplications required per time step to compute the net heating or cooling load for a room, interior surface temperatures, and interior surface heat flows in an energy balance simulation is

$$\frac{\text{\# multiplications}}{\text{time step}} = N^2 + 2N + \sum_{i=1}^{N} C_i \qquad \text{(E.2)}$$

where

C_i = number of transfer function coefficients for building element i

In equation (E.2), (N^2 + N) multiplications result from computing the N interior surface temperatures and N multiplications result from computing the heating or cooling load from the temperature differences between the N surfaces in the room and the indoor air. The summation on the right hand side of equation (E.2) results from updating the **X** matrix and computing the current heat flow across the interior surface of every building element.

When the net heating or cooling load is zero, the following number of multiplications are required per time step to compute the floating room temperature, interior surface temperatures, and interior surface heat flows:

$$\frac{\text{\# multiplications}}{\text{time step}} = (N + 1)^2 + \sum_{i=1}^{N} C_i \qquad \text{(E.3)}$$

The first term on the right hand side of equation (E.3) results from computing the **T** vector from the multiplication of the inverted **Z** matrix by the **X** vector. The summation on the right hand side of equation (E.3) results from updating the **X** matrix and computing the current heat flow across the interior surface of every building element. Next it will be shown that the computational effort of computing the heat flow across the interior surface of building element k and updating the kth row in the **X** vector is equal to the number of transfer function coefficients for building element k.

Performing an energy balance at the interior surface of building element k, shown in figure E.1, results in

$$q_{t,k,int} + q_{t,k,gains} + q_{t,k,conv} + q_{t,k,rad} = 0 \qquad \text{(E.4)}$$

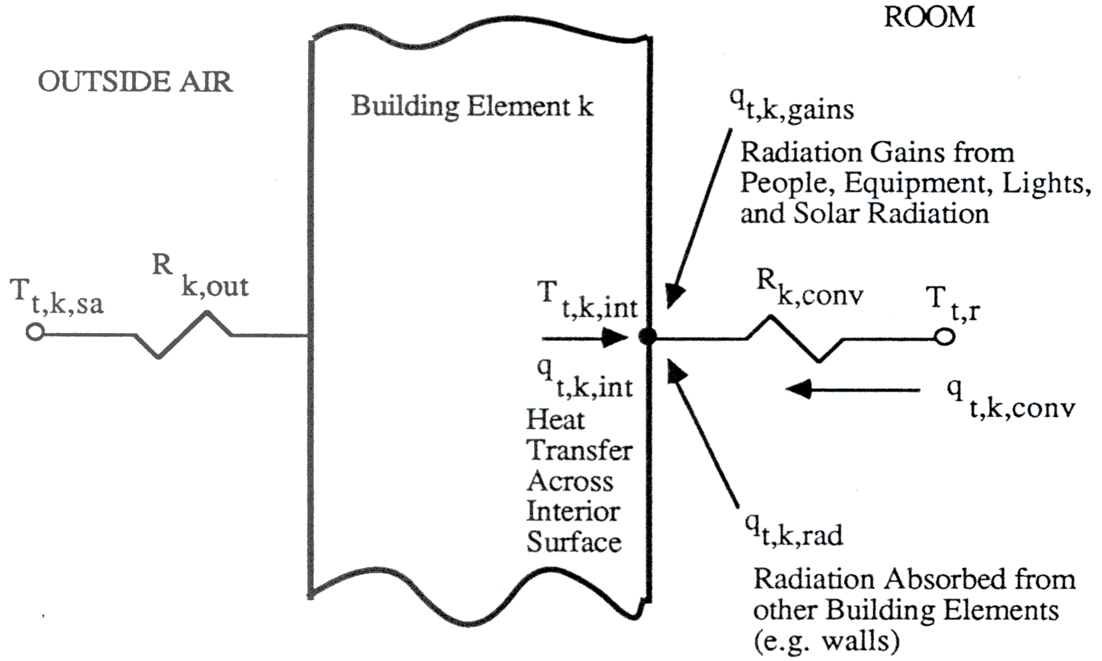

ROOM
OUTSIDE AIR
Building Element k
$q_{t,k,gains}$
Radiation Gains from People, Equipment, Lights, and Solar Radiation
$R_{k,out}$
$T_{t,k,sa}$
$T_{t,k,int}$
$R_{k,conv}$
$T_{t,r}$
$q_{t,k,int}$
Heat Transfer Across Interior Surface
$q_{t,k,conv}$
$q_{t,k,rad}$
Radiation Absorbed from other Building Elements (e.g. walls)

Figure E.1 Energy flows for a building element.

The heat flow across the interior surface of a building element k is computed from the transfer function equation

$$q_{t,k,int} = \sum_{j=0} \left(a_{j,k}\, A_k\, T_{t\text{-}j\delta,k,sa} + b_{j,k}\, A_k\, T_{t\text{-}j\delta,k,int} \right) - \sum_{j=1} c_{j,k}\, q_{t\text{-}j\delta,k,int} \qquad \text{(E.5)}$$

For an interior partition, the $a_{j,k}$ coefficients in equation (E.5) are zero. For a window, there are no past time steps in equation (E.5). The energy gain due to long-wave radiation exchange with other building elements is determined from

$$q_{t,k,rad} = \sum_{\substack{i=1 \\ i \neq k}}^{N} \frac{T_{t,i} - T_{t,k}}{R_{i\text{-}k,rad}} \qquad \text{(E.6)}$$

The heat transfer due to convection is computed from

$$q_{t,k,conv} = \frac{T_{t,r} - T_{t,k}}{R_{k,conv}} \qquad \text{(E.7)}$$

Substituting equations (E.5), (E.6), and (E.7) into equation (E.4) gives

$$\left[\sum_{j=0} \left(a_{j,k}\, A_k\, T_{t\text{-}j\delta,k,sa} + b_{j,k}\, A_k\, T_{t\text{-}j\delta,k,int} \right) - \sum_{j=0} c_{j,k}\, q_{t\text{-}j\delta,k,int} + \sum_{\substack{i=1 \\ i \neq k}}^{N} \frac{T_{t,i} - T_{t,k}}{R_{i\text{-}k,rad}} + \frac{T_{t,r} - T_{t,k}}{R_{k,conv}} + q_{t,k,gains} \right] = 0 \qquad \text{(E.8)}$$

Rearranging equation (E.8) so that the terms for the current temperatures are on the left hand side gives

$$\sum_{\substack{i=1\\i\neq k}}^{N} \frac{T_{t,i}}{R_{i\text{-}k,rad}} - \left[\sum_{\substack{i=1\\i\neq k}}^{N} \left(\frac{1}{R_{i\text{-}k,rad}} \right) - b_{0,k}\, A_k + \frac{1}{R_{k,conv}} \right] T_{t,k} + \frac{T_{t,r}}{R_{k,conv}} =$$

$$- \sum_{j=0} a_{j,k}\, A_k\, T_{t\text{-}j\delta,k,sa} - \sum_{j=1} \left(b_{j,k}\, A_k\, T_{t\text{-}j\delta,k,int} - c_{j,k}\, q_{t\text{-}j\delta,k,int} \right) - q_{t,k,gains} \tag{E.9}$$

The following equations can be used to compute the elements in the k^{th} row of the **Z** matrix for a building element which is exposed to the outdoor environment:

$$Z_{(k,i)} = \frac{1}{R_{i\text{-}k,rad}} \quad \text{for } i\neq k \tag{E.10}$$

$$Z_{(k,k)} = - \sum_{\substack{i=1\\i\neq k}}^{N} \left(\frac{1}{R_{i\text{-}k,rad}} \right) - b_{0,k}\, A_k + \frac{1}{R_{k,conv}} \tag{E.11}$$

$$Z_{(k,N+1)} = \frac{1}{R_{k,conv}} \tag{E.12}$$

The k^{th} row in the **X** vector is computed at each time step in the simulation from the following equation:

$$X_{(k)} = - \sum_{j=0} a_{j,k} A_k T_{t-j\delta,k,sa} - \sum_{j=1} \left(b_{j,k} A_k T_{t-j\delta,k,int} - c_{j,k} q_{t-j\delta,k,int} \right) - q_{t,k,gains} \tag{E.13}$$

From equation (E.13), the number of multiplications required to update a row in the **X** matrix for a building element is equal to the number of transfer functions coefficients for the building element minus one.

After the surface temperatures and indoor air temperature are computed, the current heat flow across the interior surface can be computed from:

$$q_{t,k,int} = - X_{(k)} - q_{t,k,gains} + b_{0,k} A_k T_{t,k,int} \tag{E.14}$$

Using equation (E.14) to compute the current heat flow across the interior surface of a building element requires one multiplication per time step. Thus, the computational effort for updating a row in the **X** matrix and computing the current heat flow across the interior surface of an exterior wall is equal to the number of transfer function coefficients for the exterior wall, i.e., the number of transfer function coefficients in equation (E.5).

APPENDIX F
Examples of Model Reduction for a CRTF

Building element transfer functions were generated from Mitalas and Arsenault's program (1971) for the eight surface room considered in section 3.2.4. The methods described in chapter 3 were used to determine the CRTF coefficients listed in table F.1. The steady-state response to a step change in an input is equal to the summation of the coefficients for that input divided by the summation of the coefficients for outputs, e.g., the steady-state response to a step change in outdoor temperature for a CRTF is equal to the summation of the a_j coefficients divided by the summation of the h_j coefficients. Table F.1 shows that a large number of significant digits are required to perform a CRTF simulation. Dominant root model reduction was used to determine the CRTF listed in table F2. Table F.2 shows that not as many significant digits are required to perform a reduced CRTF simulation. The Padé approximation and bilinear transformation were used to determine the reduced single-input CRTF's listed in table H.3. Figures F.1 through F.6 show the response to step changes in indoor temperatures, solar gains, and radiation gains from people, equipment, and lights for the full set of CRTF coefficents and the reduced sets of CRTF coefficients. These figures show that reduced transfer functions accurately model the heat transfer processes for the eight surface room.

Table F.1 CRTF coefficients for the eight surface room.

j	d_j Btu/hr-°F	e_j Btu/hr-°F	f_j ft^2	g_j	h_j
0	16.9558004	-236.33976539	13.24188285	0.257028349	1.00000000
1	-136.9555001	1989.20323193	-103.28529744	-2.018815091	-8.272273279
2	512.8929615	-7755.06425945	372.89439361	7.338295823	31.718166039
3	-1183.1614305	18604.54388387	-827.76744058	-16.396790592	-74.885039010
4	1884.1724460	-30778.20642564	1265.93122087	25.232314246	121.992047926
5	-2200.4081184	37297.00312342	-1416.55079126	-28.398671805	-145.651872850
6	1954.3667321	-34331.40153358	1202.43389190	24.234753019	132.163933947
7	-1351.4500856	24571.62046114	-792.35239557	-16.046275791	-93.292002479
8	738.9349735	-13886.52901097	411.48497264	8.368245441	52.022250481
9	-322.7261059	6259.83573155	-170.05463413	-3.470744509	-23.148666754
10	113.2974638	-2265.04131441	56.25428491	1.151482126	8.271391420
11	-32.0776834	660.06948958	-14.93766806	-0.306440151	-2.381192252
12	7.3306646	-155.06123864	3.18506575	0.065433854	0.552798981
13	-1.3503394	29.32756386	-0.54432608	-0.011188236	-0.103360161
14	0.1997941	-4.45100609	0.07426779	0.001525480	0.015513288
15	-0.0236092	0.53906377	-0.00804101	-0.000164786	-0.001858727
16	0.0022098	-0.05167746	0.00068503	0.000013975	0.000176352
17	-0.0001620	0.00387723	-0.00004540	-0.000000919	-0.000013101
18	0.0000092	-0.00022412	0.00000230	0.000000046	0.000000750
19	-0.0000004	0.00000976	-0.00000009	0.000000000	-0.000000032
20	0.0000000	-0.00000003	0.00000000	0.000000000	0.000000001
SUM	0.0000200	-0.0000200	0.0000281	0.000000477	0.000000541

Table F.2 Reduced multiple-input CRTF coefficients for the eight surface room.

j	d_j Btu/hr-°F	e_j Btu/hr-°F	f_j ft^2	g_j	h_j
0	17.165	-236.346	13.7983	0.25745	1.00000
1	-40.436	624.946	-29.4693	-0.55645	-2.51966
2	31.641	-550.430	20.4047	0.38956	2.11860
3	-8.220	161.680	-4.5233	-0.08703	-0.59488
SUM	0.150	-0.150	0.210	0.00358	0.00405

Table F.3 Reduced single-input CRTF coefficients for the eight surface room.

	Outdoor Temp.		Indoor Temp.		Solar		Radiation	
j	d_j (Btu/hr-°F)	h_j	e_j (Btu/hr-°F)	h_j	f_j (ft^2)	h_j	g_j	h_j
0	19.68	1.000	-208.27	1.000	18.21	1.000	0.3458	1.000
1	-15.78	-0.895	204.41	-0.896	-12.55	-0.891	-0.2535	-0.895
SUM	3.90	0.105	-3.86	0.104	5.66	0.109	0.0923	0.105

Table F.4 Reduced single-input CRTF coefficients for the eight surface room.

	Outdoor Temp.		Indoor Temp.		Solar		Radiation	
j	d_j (Btu/hr-°F)	h_j	e_j (Btu/hr-°F)	h_j	f_j (ft^2)	h_j	g_j	h_j
0	17.737	1.0000	-228.581	1.0000	14.300	1.0000	0.2815	1.0000
1	-28.060	-1.7260	419.291	-1.7220	-19.566	-1.7292	-0.4037	-1.7213
2	10.933	0.7425	-191.332	0.7388	6.111	0.7455	0.1371	0.7382
SUM	0.0610	0.0165	-0.621	0.0168	0.844	0.0163	0.0149	0.0169

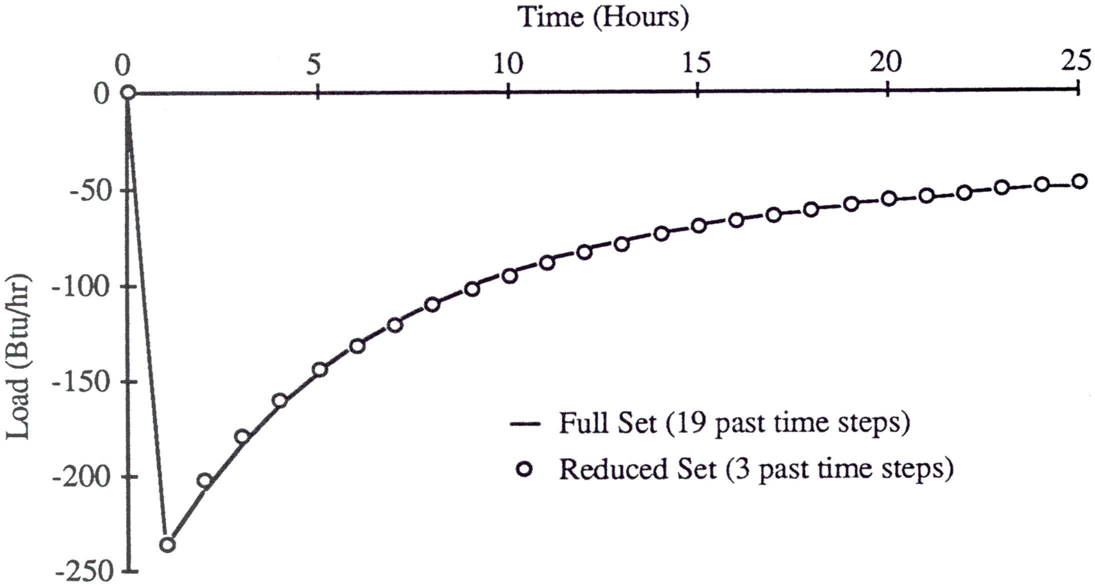


Figure F.1 Response to a 1°F step change in indoor temperature for a CRTF a reduced multiple-input CRTF.

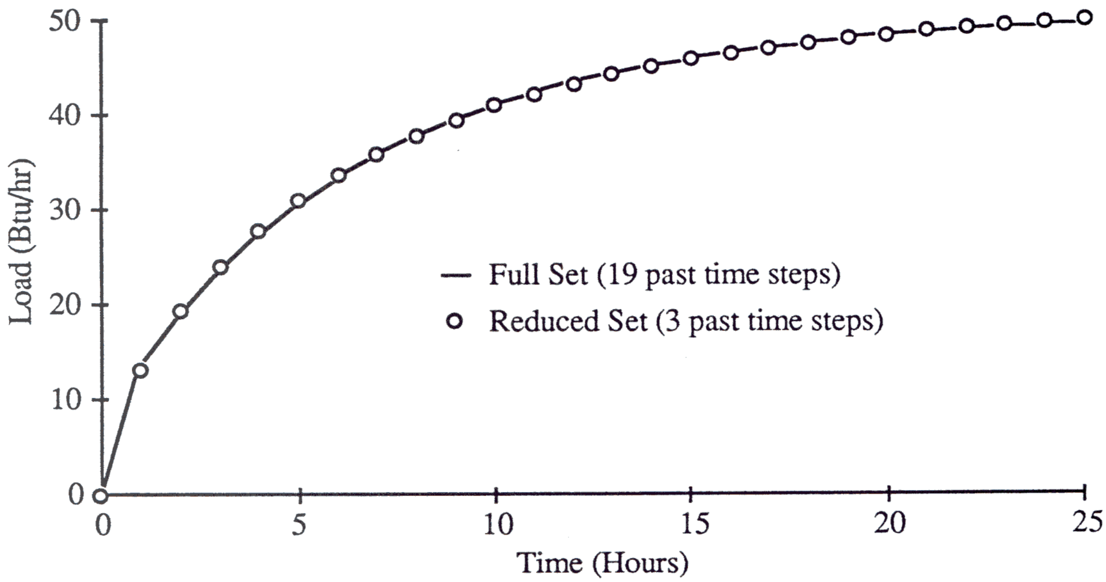


Figure F.2 Response to step change in solar radiation gains for a CRTF and a reduced multiple-input CRTF.

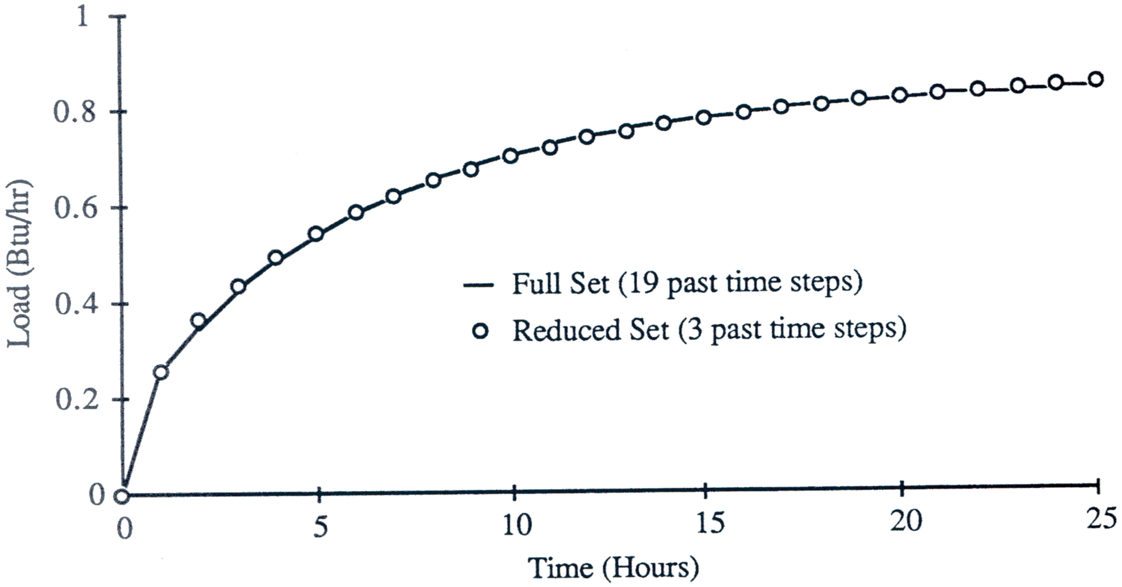


Figure F.3 Response to step change in radiation gains for a CRTF and a reduced multiple-input CRTF.

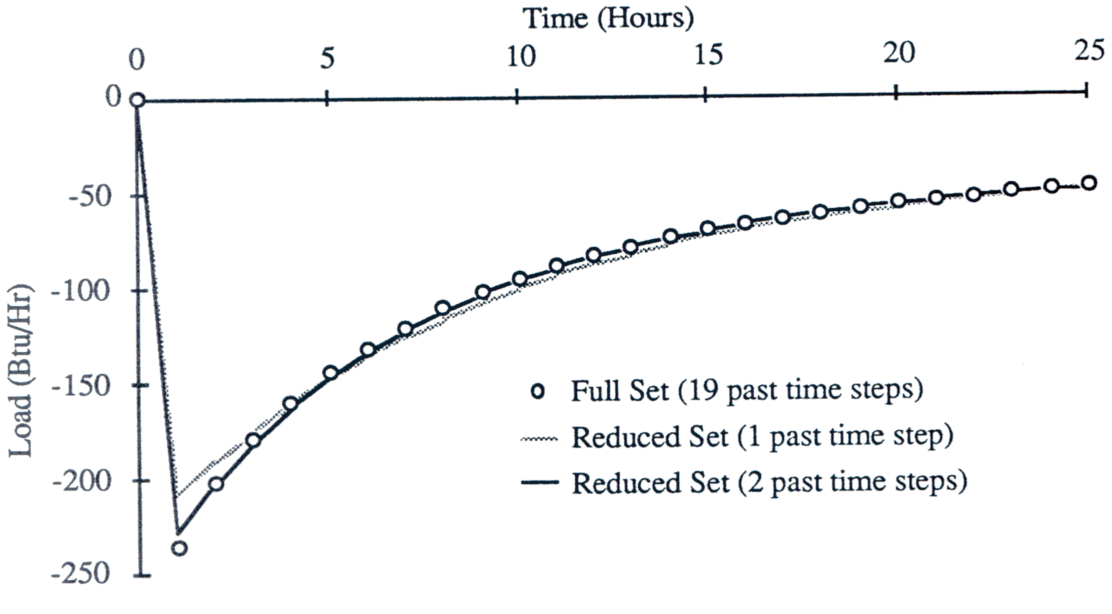


Figure F.4 Response to 1°F step change in indoor temperature for a CRTF and a reduced single-input CRTF.

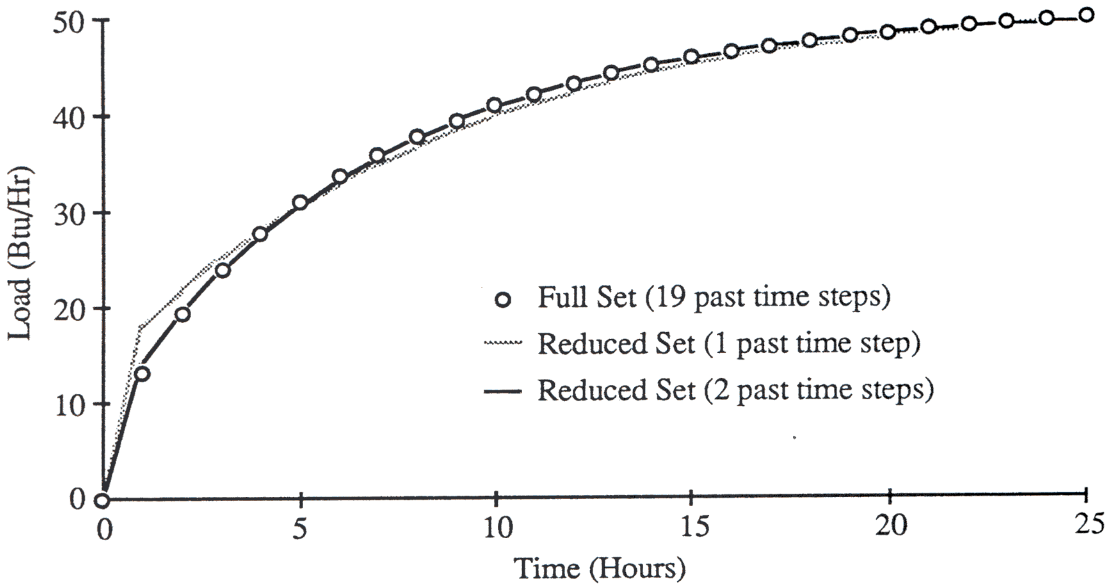


Figure F.5 Response to step change in solar radiaion gains for a CRTF and a reduced single-input CRTF.

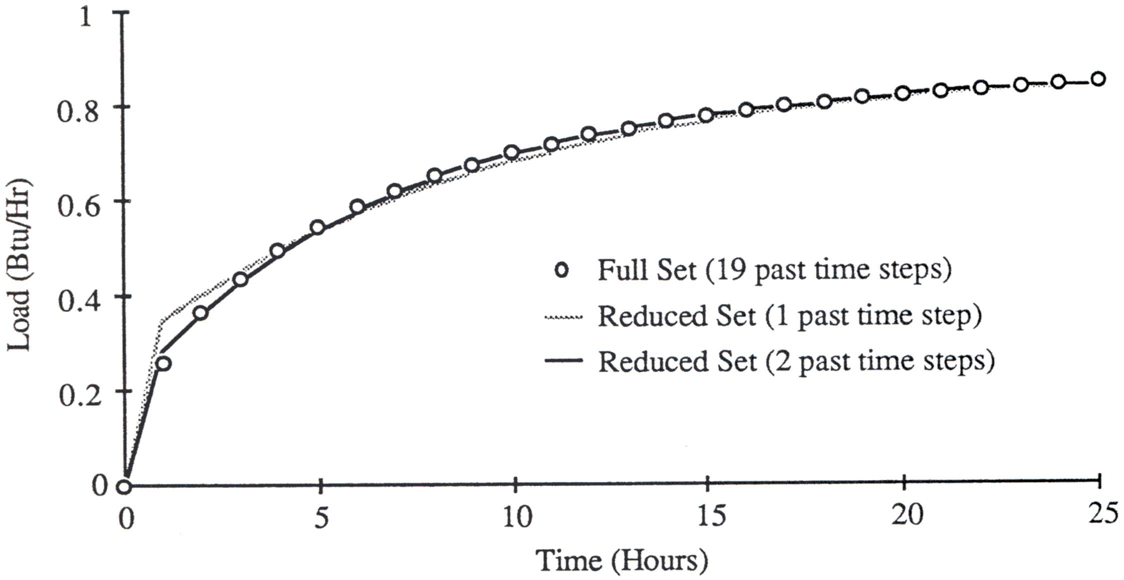


Figure F.6 Response to step change in radiaion gains for a CRTF and a reduced single-input CRTF.

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

English Letter Symbols

Symbol	Definition
a	outdoor temperature transfer function coefficient, input transfer function coefficient
$\bar{a}$	input coefficient for Laplace transfer function, input coefficient for w-transfer function
A	area
$\mathbf{A}$	(n x n) constant coefficient matrix
b	indoor temperature coefficient for a building element transfer function, output coefficient for transfer function
$\bar{b}$	output coefficient for Laplace transfer function, output coefficient for w-transfer function
B	backshift operator, equation (3.1.3)
$\mathbf{B}$	(n x p) constant coefficient matrix
c	specific heat, transfer function coefficient for past heat fluxes for a building element, input coefficient for reduced transfer function
$\bar{c}$	coefficient for power series expansion of Laplace transfer function, coefficient for power series expansion of w-transfer function
C	thermal capacitance
$\mathbf{C}$	(m x n) constant coefficient matrix
d	outdoor temperature coefficient for a building element transfer function, output coefficient for reduced transfer function

English Letter Symbols (continued)

Symbol	Definition
$\bar{d}$	input coefficient for reduced Laplace transfer function, input coefficient for reduced w-transfer function
$\mathbf{D}$	(m x p) constant coefficient matrix
e	transfer function coefficients for previous outputs, star temperature coefficient for combined transfer function, indoor temperature coefficient for CRTF
$\bar{e}$	output coefficient for reduced Laplace transfer function, output coefficient for reduced w-transfer function
E_0	dimensionless error parameter, equation (4.4.3)
E_{max}	dimensionless error parameter, equation (4.4.1)
f	solar radiation coefficient for combined transfer function and CRTF
f	area fraction
F	forward shift operator, equation (2.1.14)
g	radiation coefficient for combined transfer function and CRTF
G(s)	Laplace transfer function
$G_r(s)$	reduced Laplace transfer function
G(w)	w-transfer function
$G_r(w)$	reduced w-transfer function
G(z)	z-transfer function
$G_r(z)$	reduced z-transfer function
h	convection coefficient, transfer function coefficient for past loads
I	incident solar radiation
$\mathbf{I}$	identity matrix

English Letter Symbols (continued)

Symbol	Definition
k	integer used in algorithm for calculating exponential matrix
L	number of terms in truncated power series of exponential matrix
m	number of outputs, number of past time steps in reduced transfer function
n	number of state variables, number of past time steps in original transfer function
N	number of surfaces in enclosure or room
$N_r(w)$	function of complex variable w
p	number of inputs
q	heat flow
q''	heat flux
R	thermal resistance, resistance between star node and room air
$R_{i\text{-}j}$	resistance between surfaces i and j in the view factor network when other nodes are floating, equation (3.2.6)
$R_{i\text{-}r}$	resistance between surface i and the room air in the view factor network when other nodes are floating, equation (3.2.8)
R_k	resistance between interior surface of wall k and star node
$R_{i\text{-}j,rad}$	resistance to radiation exchange between surfaces i and j, equation (3.2.1)
$R_{i,conv}$	resistance to convective heat transfer between surface i and room air
$R_{k,out}$	resistance to convective heat transfer between exterior surface of wall k and outdoor air
$\mathbf{R}$	(n x n) constant coefficient matrix
s	complex variable which results from Laplace transformation

English Letter Symbols (continued)

Symbol	Definition
S	(m x p) matrix of transfer function coefficients for inputs
t	discrete point in time
T	temperature
$\overline{T}$	average temperature of surfaces in the room
u	vector of p inputs
U(z)	z-transform of the input
v	value of state or signal
w	complex variable which results from the bilinear transformation
$x_{(i,j)}$	element in row i and column j of the **X** matrix
$x_{(i,j),inv}$	element in row i and column j of the inverse of the **X** matrix
x	vector of n state variables
X	(N+1) by (N+1) matrix
$y_{(i)}$	element in row i of **y** vector
y	vector of m outputs
Y(z)	z-transform of the output
z	complex variable which results from z-transformation
$z_{(i)}$	element in row i of **Z** vector
Z	(N+1) vector

Greek Letter Symbols

Symbol	Definition
α	dummy variable, solar absorptance
β	constant which results from partial fraction expansion, equation (4.3.8)
δ	time step
ε	emmitance
Φ	(n x n) exponential matrix, equation (2.1.11)
ϕ	fraction of radiation gains from people, equipment, and lights which is absorbed at the interior surface
Γ_1	(n x p) matrix, equation (2.1.12)
Γ_2	(n x p) matrix, equation (2.1.13)
λ	root, equation (4.3.6)
$\bar{\lambda}$	dominant root
ψ_1	dimensionless error function, equation (3.2.11)
ψ_2	weighted error function, equation (3.2.12)
τ	time
$\overline{(\tau\alpha)}$	fraction of incident solar radiation which is absorbed at the interior surface of building element k
ω	quantity used to select dominant roots, equation (4.3.12)

Subscripts

Symbol	Definition
a	wall a
amb	ambient temperature
b	wall b
i	inside
I	solar radiation
j	transfer function coefficient for j time steps prior to time t
k	building element k
load	heating or cooling load for building
o	outside
r	room air
rad	radiation gains from people equipment and lights
sa	sol-air temperature, Eq. (3.3.2)
$t-n\delta$	input or output n time steps prior to time t
$t+n\delta$	state or signal n time steps ahead of time t
χ	combined

REFERENCES

Anderson, J. V. and Subbarao, K., "Spectral Analysis of Ambient Weather Patterns," *SOLAR ENGINEERING - 1981*, Proceedings of the ASME Solar Energy Division Third Annual Conference on Systems Simulation, Economic Analysis/Solar Heating and Cooling Results, Reno, Nevada, 1981.

ASHRAE Handbook of Fundamentals, American Society of Heating, Refrigeration, and Air Conditioning Engineers, Atlanta, GA, 1977, 1981, 1985.

Åström, K. A. and Wittenmark, B., *Computer-Controlled Systems Theory and Design*, Prentice-Hall, Inc., Englewood Cliffs, N.J., pp 20-32, 1984.

Athienitis, A. and Sullivan, H. F., "Analytical Model for Prediction of Temperature Swings," *Proceedings of Ninth SESCI Annual Conference*, Windsor, Ontario, August 1983.

Athienitis, A. and Sullivan, H. F., "Optimum Method and Algorithm for a Representative Room Temperature and its Swings," *ASME/ASES Solar Energy Conference*, Knoxville, Tennessee, March 1985.

Athienitis, A. K., "Application of Network Methods to Thermal Analysis of Passive Solar Buildings in the Frequency Domain," Ph.D. Thesis, University of Waterloo, Waterloo, Canada, 1985.

Atkinson, K. E., *An Introduction to Numerical Analysis*, John Wiley&Sons, New York, pp 415-421, 1978.

Balcomb, J. D., R. W. Jones, C. E. Kosiewicz, G. S. Lazarus, R. D. McFarland, and W. D. Wray, *Passive Solar Design Handbook, Volume 3*, American Solar Energy Society, Boulder Colorado, 1983.

BLAST, "The Building Loads Analysis and System Thermodynamics Program," Users Manual, Volume 1, Report E-153, U.S. Army Construction Engineering Research Laboratory, 1979.

Box, G. E. P. and Jenkins, G. M., *Time Series Analysis Forecasting and Control*, Holden Day, San Francisco, p 8, 1976.

Brogan, W. L., *Modern Control Theory*, Second Edition, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1985.

Bronson, R., *Modern Introductory Differential Equations*, McGraw-Hill Book Company, 1973.

Cadzow, J. A. and Martens, H. R., *Discrete-Time and Computer Control Systems*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, pp 389-390, 1970.

Carroll, J. A., "An MRT Method of Computing Radiant Energy Exchange in Rooms," *Proceedings of the Second Systems Simulations & Economic Analysis Conference*, San Diego, pp. 343-348, 1980.

Ceylan, H. T. and Myers, G. E., "Long-time Solutions to Heat Conduction Transients with Time-dependent Inputs," *ASME Journal of Heat Transfer*, Vol. 102, No.1, pp 115-120, 1980.

Chen, C. T., *Linear System Theory and Design*, Holt, Rinehart and Winston, New York, 1984.

Conte, S. D. and C. de Boor, *Elementary Numerical Analysis an Algorithmic Approach*, McGraw-Hill Book Company, New York, 1980.

DOE-2, "DOE-2 Reference Manual, Version 2.1," Report LBL-8706, Revision 1, Lawrence Berkeley Laboratory, 1980.

Fanger, P. O., *Thermal Comfort Analysis and Applications in Environmental Engineering*, McGraw-Hill, New York, 1972.

Forrester, J. R. and Wepfer, W. J., "Formulation of a Load Prediction Algorithm for a Large Commercial Building," *ASHRAE Transactions*, Vol. 90, Part 2B, 1984.

Gebhart, G., *Heat Transfer*, Second Edition, McGraw-Hill Book Company, pp. 150-158, 1971.

Goldstein, D. B. and Lokmanhekim, M., "A Simple Method for Computing the Dynamic Response of Passive Solar Buildings to Design Weather Conditions," Energy and Environment Division, Lawrence Berkely Laboratory, University of California, Berkeley, California, 1979.

Hall, I. J., Praire, R. R., Anderson, H. E., and Boes, E. C., "Generation of a Typical Meteorological Year," *Proceedings of the 1978 Meeting American Section of the International Solar Energy Society, Inc.,* Denver, Colorado, 1978.

Hayt, W. H. and J. E. Kemmerly, *Engineering Circuit Analysis*, Third Editition, McGraw-Hill Book Company, New York, 1978.

Hittle, D. C., "Calculating Building Heating and Cooling Loads Using the Frequency Response of Multilayered Slabs," Ph.D. Thesis, E-169, U.S. Army Construction Engineering Research Laboratory, Champaign Illinois, Feb. 1981.

Hittle, D. C. and R. Bishop, "An Improved Root-Finding Procedure for use in Calculating Transient Heat Flow through Multilayered Slabs," *International Journal Heat Mass Transfer*, Vol. 26, No.11, pp. 1685-1693, 1983.

Jamshidi, M., *Large-Scale Systems Modeling and Control*, North-Holland Series in System Science and Engineering, Series Volume 9, 1983.

Jury, E. I., *Theory and Application of the z-Transform Method*, John Wiley Sons, New York, 1964.

Klein, S. A., et al., "TRNSYS A Transient System Simulation Program," Solar Energy Laboratory, University of Wisconsin-Madison, Engineering Experiment Station Report 38-12, Dec. 1983.

Kusuda, T., "Thermal Response Factors for Multi-Layer Structures of Various Heat Conduction Systems," *ASHRAE Transactions*, Volume 75, Part I, 1969.

Kuo, B. C., *Digital Control Systems*, Holt, Rinehart and Winston, New York, pp. 286-287, 1980.

Linpack User's Guide, Society for Industrial and Applied Mathematics, Philadelphia 1979.

Madsen, J. M., "Modeling Heat Transfer in Rooms using Transfer Function Methods," M.S. Thesis, University of Wisconsin-Madison, 1982.

Matlab, Interactive Matrix Language, University of Wisconsin-Madison MACC Academic Computing Center, Revision 1, 1982.

Mitchell, J. W., *Energy Engineering*, John Wiley & Sons, p. 82, 1983.

Mitalas, G. P., "An Assessment of Common Assumptions in Estimating Cooling Loads and Space Temperatures," *ASHRAE Transactions*, Vol. 71, Part II,pp. 72-80, 1965.

Mitalas, G. P. and Arseneault, J. G., "Fortran IV Program to Calculate z-Transfer Functions for the Calculation of Transient Heat Transfer through Walls and Roofs," *Proceedings of the Conference on Use of Computers for Environmental Engineering Related to Buildings*, held in Gaithersburg, MD, NBS Building Science Series 39, Oct. 1971.

Moler, C. and Van Loan, C., "Nineteen Dubious Ways to Compute the Exponential of a Matrix," Society for Industrial and Applied Mathematics Review, Vol. 20, No. 4, pp 801-836, 1978.

Monsen, W. A., S. A. Klein, W. A. Beckman, "Prediction of Direct Gain Solar Heating System Performance," *Solar Energy*, Vol. 27, No. 2, pp. 143-147, 1981.

Monsen, W. A., S. A. Klein, W. A. Beckman, "The Un-Utilizability Design Method for Collector-Storage Walls," *Solar Energy*, Vol. 29, No. 5, pp. 421-429, 1981.

Myers, G. E., *Analytical Methods in Conduction Heat Transfer*, McGraw-Hill, NY, 1971.

Myers, G. E., Class Notes for ME764, Department of Mechanical Engineering, University of Wisconsin-Madison, 1981.

Pawelski, M. J., "Development of Transfer Function Load Models and Their use in Modeling the CSU Solar House I," M.S. Thesis, University of Wisconsin-Madison, 1976.

Peavy, B. A. , "A Note on Response Factors and Conduction Transfer Functions," *ASHRAE Transactions*, Vol. 84, No. 3, pp. 688-690, 1978.

Sedgewick, R., *Algorithms*, Addison Wesley Publishing Company, 1983.

Shamash, Y., "Failure of the Routh-Hurwitz Method of Reduction," *IEEE Transactions on Automatic Control*, Vol. AC-25, No. 2, pp. 313-314, 1980.

Shamash, Y., "Critical Review of Methods for Deriving Stable Reduced-Order Models," *IFAC Identification and System Parameter Estimation 1982*, Washington D. C., pp. 1519-1523, 1982.

Shih, Y. P., and Wu, W. T., "Simplification of z-Transfer Functions by Continued Fractions," *International Journal of Control*, Vol. 17, No. 5, 1973.

Smart, M. G. and Ballinger, J. A., "Fourier-synthesized Weather Data for Building Energy Use Estimation," *Building and Environment*, Vol. 19, No. 1, 1984.

Sowell, E. F., and G. N. Walton, "Efficient Calculation of Zone Loads," *ASHRAE Transactions*, Vol. 86, Part 1, pp. 49-72, 1980.

Spiegel, M. R., *Mathematical Handbook of Formulas and Tables*, *Schaum's Outline Series*, McGraw-Hill Book Company, 1968.

Stephenson, D. G. and Mitalas, G. P., "Cooling Load Calculations by Thermal Response Factor Method," *ASHRAE Transactions*, Vol. 73, Part I, pp III.1.1-III.1.7, 1967.

Stephenson, D. G. and Mitalas, G. P., "Calculation of Heat Conduction Transfer Functions for Multi-Layer Slabs," *ASHRAE Transactions*, Vol. 77, Part II, pp 117-126, 1971.

Subbarao, K., "The Dynamic Response of Thermal Masses and their Interactions using Vector Diagrams," *Thermal Mass Seminar*, Knoxville, Tennessee, SERI/TP-254-1632, Oct. 1982.

Subbarao, K. and Anderson, J., "A Frequency-Domain Approach to Passive Building Energy Analysis," Solar Energy Research Institute, SERI/TR-254-1544, July 1982.

Subbarao, K. and Anderson, J., "A Fourier Transform Approach to Building Simulations with Thermostatic Controls," *Proceedings of Sixth Annual ASME Technical Conference*, Orlando Florida, 1983.

Walton, G. N., "A New Algorithm for Radiant Interchange in Room Loads Calculations," *ASHRAE Transactions*, Vol. 86, pp. 190-208, 1980.

Walton, G. N., "Thermal Analysis Research Program Reference Manual (TARP), " National Bureau of Standards, NBSIR 83-2655, 1983.

Walton, G. N., "Developments in the Heat Balance Method for Simulating Room Thermal Response," Workshop on HVAC Controls, Modeling and Simulation, Georgia Institute Technology, Atlanta Georgia, 1984.

Wiberg, D. M., *State Space and Linear Systems*, McGraw Hill Book Company, New York, pp 117-118, 1971.

Wilkes, K. E., "Dynamic Thermal Performance of Walls and Ceilings/Attics," *Thermal Performance of the Exterior Envelopes of Buildings II*, ASHRAE SP38, 1981.