

J. E. Seem

S. A. Klein

W. A. Beckman

J. W. Mitchell

Solar Energy Laboratory,
University of Wisconsin—Madison,
Madison, WI 53706

Transfer Functions for Efficient Calculation of Multidimensional Transient Heat Transfer

Finite difference or finite element methods reduce transient multidimensional 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 paper presents a method for determining the exact solution to a set of first-order differential equations when the inputs are modeled by a continuous, piecewise linear curve. For long-time solutions, the method presented is more efficient than Euler, Crank-Nicolson, or other classical techniques.

Introduction

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 linear and time invariant. Response factors relate the output of a 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. Transfer function and response factor methods are more efficient for solving long-time heat transfer problems than Euler, Crank-Nicolson, or other classical techniques because there is no critical time step and the internal temperature distribution is not calculated.

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 difference equation that 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 paper, the latter definition will be used.

For one-dimensional problems, Stephenson and Mitalas (1967) determine the "exact" set of transfer functions and/or response factors by solving the conduction equation by Laplace and/or z transforms. To develop response factors or transfer functions for multidimensional heat transfer, it is necessary to discretize the problem spatially by use of finite difference or finite element techniques. Spatial discretization results in a set of first-order differential equations.

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

This paper presents a method for calculating transfer functions for multidimensional heat transfer that 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. The necessary equa-

tions and algorithms for computing transfer functions for multidimensional heat transfer are included in this paper. These transfer functions allow computationally simple and accurate two or three-dimensional heat transfer calculations. The method is useful for calculating heat transfer in buildings.

Analytical Solution

A state space formulation has traditionally been used to analyze linear systems that 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 discretize the problem spatially. A state space representation for a continuous, linear, time-invariant system is

$$\frac{dx}{d\tau} = Ax + Bu \quad (1)$$

$$y = Cx + Du \quad (2)$$

Equation (1) is called the state equation and equation (2) is called the output equation in a state space formulation.

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

$$x_{t+\delta} = e^{A\delta} x_t + \int_t^{t+\delta} e^{A(t+\delta-\tau)} Bu(\tau) d\tau \quad (3)$$

The exponential matrix is defined by the power series

$$e^{A\delta} = I + A\delta + \frac{A^2\delta^2}{2!} + \frac{A^3\delta^3}{3!} + \dots + \frac{A^n\delta^n}{n!} + \dots \quad (4)$$

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

The first term on the right-hand side of equation (3) is called the complementary function, force-free response, or zero-input response, and the second term on the right-hand side of equation (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 $t + \delta$. Inputs between times t and $t + \delta$ are modeled by a continuous, piecewise linear function and are calculated by

$$u(\tau) = u_t + \frac{(\tau-t)}{\delta} (u_{t+\delta} - u_t) \quad (5)$$

At this point, the solution of the state equation for heat transfer applications differs from the solution of the state

Contributed by the Heat Transfer Division and presented at the ASME/AIChE National Heat Transfer Conference, Pittsburgh, Pennsylvania, August 9-12, 1987. Manuscript received by the Heat Transfer Division November 20, 1987. Keywords: Building Heat Transfer, Numerical Methods.

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 (5) into equation (3) results in

$$x_{t+\delta} = e^{A\delta} x_t + \int_t^{t+\delta} e^{A(t+\delta-\tau)} B \left[u_t + \frac{(\tau-t)}{\delta} (u_{t+\delta} - u_t) \right] d\tau \quad (6)$$

By making the change of variables $\alpha = \tau - t$, equation (6) can be rewritten as

$$x_{t+\delta} = e^{A\delta} x_t + \left[\int_0^\delta e^{A(\delta-\alpha)} d\alpha \right] B u_t + \left[\int_0^\delta \alpha e^{A(\delta-\alpha)} d\alpha \right] \left[\frac{B}{\delta} (u_{t+\delta} - u_t) \right] \quad (7)$$

Appendix B describes the steps for integrating the two integrals in equation (7). The solution to the first integral is

$$\int_0^\delta e^{A(\delta-\alpha)} d\alpha = A^{-1} (e^{A\delta} - I) \quad (8)$$

and the solution to the second integral is

$$\int_0^\delta \alpha e^{A(\delta-\alpha)} d\alpha = A^{-1} A^{-1} (e^{A\delta} - I) - A^{-1} \delta \quad (9)$$

Substituting the solution of the two integrals, equations (8) and (9), into equation (7) yields

$$x_{t+\delta} = \Phi x_t + (\Gamma_1 - \Gamma_2) u_t + \Gamma_2 u_{t+\delta} \quad (10)$$

where

$$\begin{aligned} \Phi &= e^{A\delta} \\ \Gamma_1 &= A^{-1} (e^{A\delta} - I) B = A^{-1} (\Phi - I) B \\ \Gamma_2 &= \left[A^{-1} A^{-1} (e^{A\delta} - I) - A^{-1} \delta \right] \frac{B}{\delta} = A^{-1} \left[\frac{\Gamma_1}{\delta} - B \right] \end{aligned}$$

Equation (10) relates the states at time $t + \delta$ to the states at time t and the inputs at the times t and $t + \delta$.

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

$$F z_t = z_{t+\delta} \quad (11)$$

will be now be used to relate the states to previous inputs. Using the forward shift operator, equation (10) can be written as

$$(F I - \Phi) x_t = (F \Gamma_2 + \Gamma_1 - \Gamma_2) u_t \quad (12)$$

Multiplying equation (12) by the inverse of the $(F I - \Phi)$ matrix gives

$$x_t = (F I - \Phi)^{-1} (F \Gamma_2 + \Gamma_1 - \Gamma_2) u_t \quad (13)$$

Substituting equation (13) into equation (2) yields

$$y_t = \left[C(F I - \Phi)^{-1} (F \Gamma_2 + \Gamma_1 - \Gamma_2) + D \right] u_t \quad (14)$$

Equation (14) relates the outputs from the system to the inputs.

The inverse of the $(F I - \Phi)$ matrix is equal to the adjoint of $(F I - \Phi)$ divided by the determinant of $(F I - \Phi)$ (Wiberg, 1971). The degree of F for the determinant of the $(F I - \Phi)$ matrix is at most n and the degree of F for the adjoint of $(F I - \Phi)$ is at most $n - 1$. Thus, the $(F I - \Phi)$ matrix can be written as

$$(F I - \Phi)^{-1} = \frac{R_0 F^{n-1} + R_1 F^{n-2} + \dots + R_{n-2} F + R_{n-1}}{F^n + e_1 F^{n-1} + \dots + e_{n-1} F + e_n} \quad (15)$$

The R matrices and the e scalar constants can be determined by computing the adjoint of the $(F I - \Phi)$ matrix and dividing by the determinant of the $(F I - \Phi)$ matrix or by using Leverrier's algorithm (Wiberg, 1971) described in Appendix C. Substituting equation (15) into (14) results in

$$\begin{aligned} (F^n + e_1 F^{n-1} + \dots + e_n) y_t &= \left[C(R_0 F^{n-1} + R_1 F^{n-2} \right. \\ &+ \dots + R_{n-2} F + R_{n-1}) (F \Gamma_2 + \Gamma_1 - \Gamma_2) \\ &+ D (F^n + e_1 F^{n-1} + \dots + e_n) \left. \right] u_t \quad (16) \end{aligned}$$

Multiplying the matrices on the right side of equation (16) and combining common terms of the forward shift operator gives

$$\begin{aligned} (F^n + e_1 F^{n-1} + \dots + e_n) y_t &= \left\{ (C R_0 \Gamma_2 + D) F^n \right. \\ &+ \left[C(R_1 \Gamma_1 - R_1 \Gamma_2 + R_2 \Gamma_2) + e_2 D \right] F^{n-2} + \dots \\ &+ \left[C(R_{n-2} \Gamma_1 - R_{n-2} \Gamma_2 + R_{n-1} \Gamma_2) + e_{n-1} D \right] F \\ &+ \left. \left[C(R_{n-1} \Gamma_1 - R_{n-1} \Gamma_2) + e_n D \right] \right\} u_t \quad (17) \end{aligned}$$

Nomenclature

a_{ij} = entry in row i and column j of the A matrix

A = area

A = $(n \times n)$ constant coefficient matrix

B = $(n \times p)$ constant coefficient matrix

C = $(m \times n)$ constant coefficient matrix

C = thermal capacitance

c = specific heat

D = $(m \times p)$ constant coefficient matrix

e = transfer function coefficients for previous outputs

F = forward shift operator

h = convection coefficient

I = identity matrix

k = integer used in algorithm for calculating exponential matrix

L = number of terms in truncated power series of exponential matrix

m = number of outputs

n = number of state variables

p = number of inputs

q'' = heat flux

R = thermal resistance

R = $(n \times n)$ constant coefficient matrix

S = $(m \times p)$ matrix of transfer function coefficients for inputs

t = discrete point in time

T = temperature

u = vector of p inputs

x = vector of n state variables

y = vector of m outputs

z = value of a state or signal

α = dummy variable

Γ_1 = $(n \times p)$ matrix

Γ_2 = $(n \times p)$ matrix

δ = time step

τ = time

Φ = $(n \times n)$ exponential matrix

Subscripts

in = inside temperature

out = outside temperature

t = state or signal at time t

$t + n\delta$ = state or signal n time steps ahead of time t

Using the definition of the forward shift operator, equation (17) can be rewritten as

$$\begin{aligned}
 y_{t+n\delta} + e_1 y_{t+(n-1)\delta} + \dots + e_n y_t &= (\mathbf{C}\mathbf{R}_0\mathbf{\Gamma}_2 + \mathbf{D})\mathbf{u}_{t+n\delta} \\
 &+ \left\{ \mathbf{C} \left[\mathbf{R}_0(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + \mathbf{R}_1\mathbf{\Gamma}_2 \right] + e_1 \mathbf{D} \right\} \mathbf{u}_{t+(n-1)\delta} \\
 &+ \left\{ \mathbf{C} \left[\mathbf{R}_1(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + \mathbf{R}_2\mathbf{\Gamma}_2 \right] + e_2 \mathbf{D} \right\} \mathbf{u}_{t+(n-2)\delta} + \dots \\
 &+ \left\{ \mathbf{C} \left[\mathbf{R}_{n-2}(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + \mathbf{R}_{n-1}\mathbf{\Gamma}_2 \right] + e_{n-1} \mathbf{D} \right\} \mathbf{u}_{t+\delta} \\
 &+ \left[\mathbf{C}\mathbf{R}_{n-1}(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + e_n \mathbf{D} \right] \mathbf{u}_t
 \end{aligned} \quad (18)$$

Shifting the inputs and outputs in equation (18) n time steps back gives

$$\begin{aligned}
 y_t + e_1 y_{t-\delta} + \dots + e_n y_{t-n\delta} &= (\mathbf{C}\mathbf{R}_0\mathbf{\Gamma}_2 + \mathbf{D})\mathbf{u}_t \\
 &+ \left\{ \mathbf{C} \left[\mathbf{R}_0(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + \mathbf{R}_1\mathbf{\Gamma}_2 \right] + e_1 \mathbf{D} \right\} \mathbf{u}_{t-\delta} \\
 &+ \left\{ \mathbf{C} \left[\mathbf{R}_1(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + \mathbf{R}_2\mathbf{\Gamma}_2 \right] + e_2 \mathbf{D} \right\} \mathbf{u}_{t-2\delta} + \dots \\
 &+ \left\{ \mathbf{C} \left[\mathbf{R}_{n-2}(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + \mathbf{R}_{n-1}\mathbf{\Gamma}_2 \right] + e_{n-1} \mathbf{D} \right\} \mathbf{u}_{t-(n-1)\delta} \\
 &+ \left[\mathbf{C}\mathbf{R}_{n-1}(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + e_n \mathbf{D} \right] \mathbf{u}_{t-n\delta}
 \end{aligned} \quad (19)$$

Equation (19) can be written more compactly as

$$y_t = \sum_{j=0}^n (\mathbf{S}_j \mathbf{u}_{t-j\delta}) - \sum_{j=1}^n (e_j y_{t-j\delta}) \quad (20)$$

where

$$\mathbf{S}_0 = \mathbf{C}\mathbf{R}_0\mathbf{\Gamma}_2 + \mathbf{D}$$

$$\mathbf{S}_j = \mathbf{C} \left[\mathbf{R}_{j-1}(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + \mathbf{R}_j\mathbf{\Gamma}_2 \right] + e_j \mathbf{D} \quad \text{for } 1 \leq j \leq n-1$$

$$\mathbf{S}_n = \mathbf{C}\mathbf{R}_{n-1}(\mathbf{\Gamma}_1 - \mathbf{\Gamma}_2) + e_n \mathbf{D}$$

Equation (20) is a transfer function equation that 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 \mathbf{S} coefficient. The additional \mathbf{S} coefficient is usually not significant.

The transfer function coefficients in equation (20) 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. A large amount of computer memory would be required to store the $n \times n \times n$ \mathbf{R} matrices if equation (20) was used to calculate transfer functions. Fortunately, the storage requirement for the \mathbf{R} matrices can be reduced to $2 \times n \times n$ matrices if Leverrier's algorithm is combined with equation (20). Appendix D contains the steps for computing numerically significant transfer function coefficients with a minimum amount of storage for the \mathbf{R} matrices.

Results

The equations and algorithms presented in this paper were used to write a 150 line FORTRAN program for calculating transfer functions from a state space formulation. (The pro-

Table 1 Transfer function coefficients for a 0.3 m concrete wall and two, five, and twenty-node finite-difference models of the concrete wall

Transfer Function Coefficient	Partial Differential Equation	2 node			5 node			20 node		
S_0 Out. Temp. $W/m^2 \cdot ^\circ C$	0	0.0094	0.0006	0						
S_1 Out. Temp. $W/m^2 \cdot ^\circ C$	0.0057	0.0346	0.0108	0.0057						
S_2 Out. Temp. $W/m^2 \cdot ^\circ C$	0.0301	0.0079	0.02450	0.0301						
S_3 Out. Temp. $W/m^2 \cdot ^\circ C$	0.0227	0	0.0097	0.0221						
S_4 Out. Temp. $W/m^2 \cdot ^\circ C$	0.0028	0	0.0006	0.0028						
S_0 Ins. Temp. $W/m^2 \cdot ^\circ C$	-6.8395	-7.8820	-7.1530	-6.8554						
S_1 Ins. Temp. $W/m^2 \cdot ^\circ C$	12.8767	13.9866	15.3087	12.9579						
S_2 Ins. Temp. $W/m^2 \cdot ^\circ C$	-7.3415	-6.1566	-11.1153	-7.4482						
S_3 Ins. Temp. $W/m^2 \cdot ^\circ C$	1.2935	0	3.2980	1.3406						
S_4 Ins. Temp. $W/m^2 \cdot ^\circ C$	-0.0500	0	-0.4015	-0.0551						
S_5 Ins. Temp. $W/m^2 \cdot ^\circ C$	0.0006	0	0.0165	0.0006						
e_1	-1.7442	-1.6820	-1.9782	-1.7502						
e_2	0.9050	0.7037	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						

gram 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 0.3 m homogeneous concrete wall with a density of 2200 kg/m^3 , specific heat of $0.84 \text{ kJ/kg} \cdot ^\circ C$, thermal conductivity of $1.7 \text{ W/m} \cdot ^\circ C$, and convection coefficients at both sides of the wall of $8.3 \text{ W/m}^2 \cdot ^\circ C$. Inside and outside air temperatures were the inputs to the transfer function equation and the heat flux at the interior surface of the wall was the output. Appendix E contains the steps required to calculate transfer function coefficients for a two-node finite-difference model. (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 transfer function coefficients.) 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 compared in Table 1. As the number of nodes in the finite-difference model increase, the transfer function coefficients from the state space formulations approach those of Mitalas and Arsenalt, which are based upon the solution of the partial differential equation (i.e., the continuous model).

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

$$T = -15^\circ C + 2.8^\circ C \sin[(\tau\pi)/24 \text{ h}]$$

and the air temperature on the other side of the wall was $-17.7^\circ C$. Figure 1 (outside air temperature varying) and Fig. 2 (inside air temperature varying) contain a graph of 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 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 3 contains the central processing unit (CPU) time of a Micro Vax computer to compute all transfer function coefficients, numerically significant coefficients, and the exponential matrix for different numbers of nodes. (The tolerance limit in Appendix D for the calculation of numerically signifi-

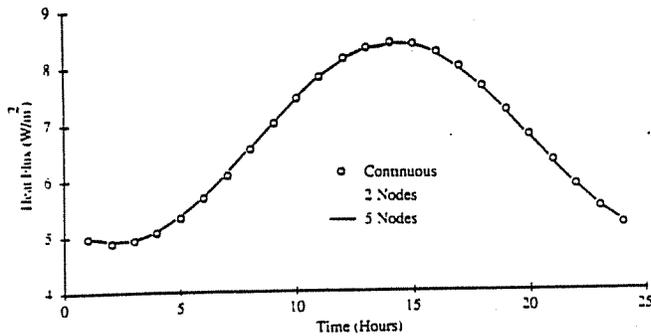


Fig. 1 Heat flux at interior surface of concrete wall with outside air temperature varying

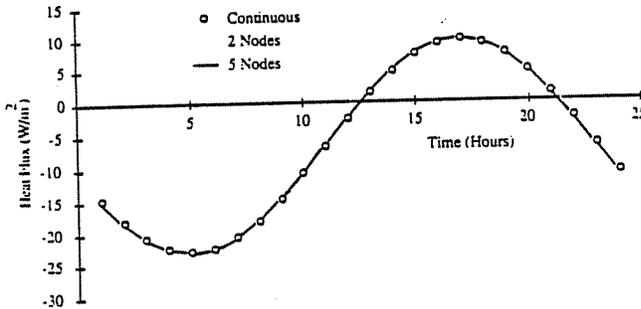


Fig. 2 Heat flux at interior surface of concrete wall with inside air temperature varying

cant coefficient was 0.000001.) Table 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.

Application

Building simulation programs such as DOE-2, TRNSYS, and TARP (1980, 1983, 1983) currently use transfer functions and response factors to model 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 that require two-dimensional models, e.g., walls that contain metal tee-bars or tie rods. Simulation programs could use these transfer functions to model multidimensional heat transfer in building elements. The ASHRAE handbook of fundamentals (1977, 1981, 1985) lists tables of transfer functions for one-dimensional heat transfer through multilayered slabs. These tables could be updated to include walls that have multidimensional heat transfer. An example demonstrating the importance of properly modeling a roof deck section with 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 (1977, 1981, 1985). Figure 3 shows the inputs (T_{in} , T_{out}) and output (q'') for the transfer functions. The transfer functions were determined for the section of the roof deck shown in Fig. 4. Transfer function coefficients for the one-dimensional model were determined for a multilayered roof with the same area-weighted thermal physical properties as the roof deck section. The steady-state response for the one-dimensional model is equal to the steady-state response in the ASHRAE example. Figure 5 shows the nodal spacing for the two-dimensional finite difference model. Table 4 contains the thermal physical properties of the materials in the roof. The outside convection coefficient is 34 $W/m^2\cdot^{\circ}C$ and the inside convection coefficient is 9.3

Table 2 Sum of squares of the residuals between continuous model and finite-difference model for a 24 hour period

Number of Nodes	Sum of Square of Residuals W^2/m^4	
	Outside Temp. Varying	Inside Temp. Varying
2	1.0	200.0
5	0.017	1.6
10	0.00070	0.060
15	0.00012	0.010
20	0.000035	0.0030

Table 3 CPU time to compute the exponential matrix, all transfer function coefficients, and all numerically significant transfer function coefficients

Number of Nodes	Central Processing Unit Time in Seconds		
	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	6.3
40	15.0	54.5	16.3
50	30.6	123.2	32.9

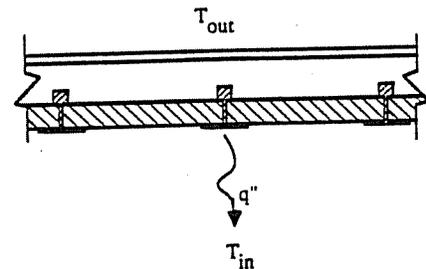


Fig. 3 Roof deck with bulb tees 0.6 m on center

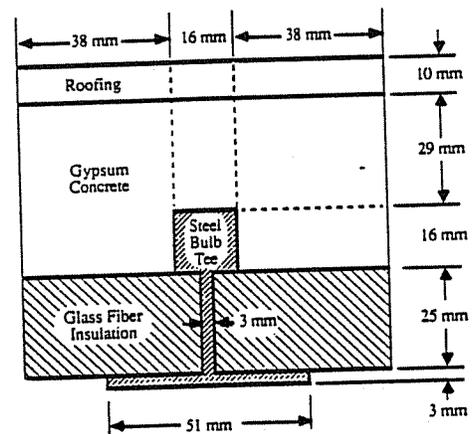


Fig. 4 Section of the roof deck

$W/m^2\cdot^{\circ}C$. Table 5 contains the transfer function coefficients for the one-dimensional and two-dimensional models of the roof deck section.

A graph of a $0.56^{\circ}C$ ($1^{\circ}F$) step change in outdoor temperature with an indoor temperature equal to zero for one and two-dimensional models can be seen in Fig. 6. The steady-state and transient response for the two models is significantly different. This graph demonstrates the importance of properly modeling the steel bulb tees in the roof deck section.

The equations presented in this paper could be used to compute a comprehensive room transfer function (CRTF) for a room. A CRTF is a single transfer function that relates the loads for a room, zone, or building to the inputs, e.g., solar

Table 4 Thermal properties of the materials in the roof deck

Material	Thermal Conductivity (W/m·°C)	Density (kg/m ³)	Specific Heat (kJ/kg·°C)
Roofing	0.16	110.0	1.5
Gypsum Concrete	0.24	82.0	0.88
Steel	45.0	7800.0	0.50
Glass Fiber	0.036	8.0	0.96

Table 5 Transfer function coefficients for one-dimensional and two-dimensional models of roof deck section

i	ONE-DIMENSIONAL			TWO-DIMENSIONAL		
	Outside Temp. S _i (W/m ² ·°C)	Inside Temp. S _i (W/m ² ·°C)	ε _i	Outside Temp. S _i (W/m ² ·°C)	Inside Temp. S _i (W/m ² ·°C)	ε _i
0	0.508	-6.280		0.304	-4.567	
1	1.172	4.571	-0.406	0.857	3.536	-0.469
2	0.115	-0.085	0.004	0.125	-0.257	0.019
3	0.000	0.000	0.000	0.000	0.002	-0.000

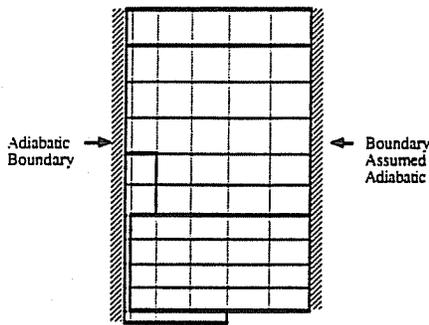


Fig. 5 Nodal spacing of two-dimensional finite-difference model

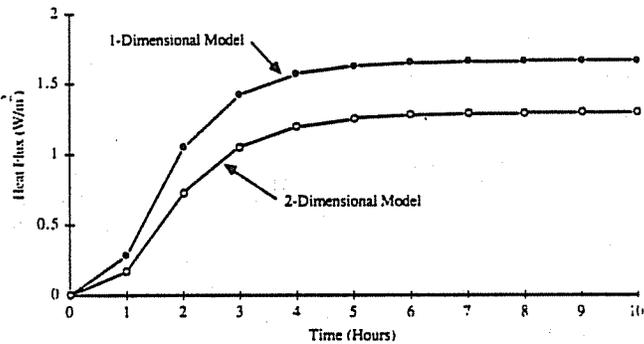


Fig. 6 Response to 0.58°C (1°F) step change in outdoor temperature for a section of the roof deck

gains outdoor temperature, indoor temperature. Seem et al. (1987) have reported on the computational savings of using a CRTF over the heat balance method discussed in ASHRAE (1977, 1981, 1985).

Conclusion

There are a number of areas of application, besides walls, where transfer functions for multidimensional heat transfer could be used. Transfer functions could be developed to model multidimensional heat transfer processes in an attic, a basement, or earth-contact structure, or a room or building. These transfer functions could be used to improve the speed and accuracy of building simulation programs. Transfer functions for all these applications can be efficiently calculated by using the algorithms presented in this paper.

Acknowledgments

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.

References

ASHRAE Handbook of Fundamentals, 1977, 1981, 1985. American Society of Heating, Refrigeration, and Air Conditioning Engineers, Atlanta, GA.

Åström, K. A., and Wittenmark, B., 1984, *Computer-Controlled Systems Theory and Design*, Prentice-Hall, Inc., Englewood Cliffs, NJ, pp. 20-32.

Atkinson, K. E., 1978, *An Introduction to Numerical Analysis*, Wiley, New York, pp. 415-421.

Box, G. E. P., and Jenkins, G. M., 1976, *Time Series Analysis Forecasting and Control*, Holden Day, San Francisco, CA, p. 8.

Brogan, W. L., 1985, *Modern Control Theory*, 2nd ed., Prentice Hall, Inc., Englewood Cliffs, NJ.

Bronson, R., 1973, *Modern Introductory Differential Equations*, McGraw-Hill, New York.

Cadzow, J. A., and Martens, H. R., 1970, *Discrete-Time and Computer Control Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, pp. 389-390.

Ceylan, H. T., and Myers, G. E., 1980, "Long-Time Solutions to Heat Conduction Transients With Time-Dependent Inputs," *ASME JOURNAL OF HEAT TRANSFER*, Vol. 102, No. 1, pp. 115-120.

Chen, C.-T., 1984, *Linear System Theory and Design*, Holt, Rinehart and Winston, New York.

DOE-2, 1980, "DOE-2 Reference Manual, Version 2.1," Report LBL-8706, Revision 1, Lawrence Berkeley Laboratory, CA.

Klein, S. A., et al., 1983, "TRNSYS A Transient System Simulation Program," Solar Energy Laboratory, University of Wisconsin—Madison, Engineering Experiment Station Report 38-12, Dec.

Linpack User's Guide, 1979, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Matlab, 1982, Interactive Matrix Language, University of Wisconsin—Madison, MACC Academic Computing Center, Revision 1.

Mitalas, G. P., and Arseneault, J. G., 1971, "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*, Gaithersburg, MD, NBS Building Science Series 39, Oct.

Moler, C., and Van Loan, 1978, "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *Society for Industrial and Applied Mathematics Review*, Vol. 20, No. 4, pp. 801-836.

Myers, G. E., 1971, *Analytical Methods in Conduction Heat Transfer*, McGraw-Hill, New York.

Seem, J. E., Klein, S. A., Beckman W. A., and Mitchell, J. W., 1987, "Comprehensive Room Transfer Functions for Efficient Calculation of the Transient Heat Transfer Processes in Buildings," presented at the 1987 ASME/AIChE National Heat Transfer Conference, Technical Session on Heat Transfer in Buildings, Pittsburgh, PA, Aug.

Stephenson, D. G., and Mitalas, G. P., 1967, "Cooling Load Calculations by Thermal Response Factor Method," *ASHRAE Transactions*, Vol. 73, Part I, pp. III.1.1.-III.1.7.

Stephenson, D. G., and Mitalas, G. P., 1971, "Calculation of Heat Conduction Transfer Functions for Multi-layer Slabs," *ASHRAE Transactions*, Vol. 77, Part II, pp. 117-126.

Walton, G. N., 1983, "Thermal Analysis Research Program Reference Manual (TARP)," National Bureau of Standards, NBSIR 83-2655.

Wiberg, D. M., 1971, *State Space and Linear Systems*, McGraw-Hill, New York, pp. 117-118.

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

$$e^{A\delta} \approx I + A\delta + \frac{A^2\delta^2}{2!} + \frac{A^3\delta^3}{3!} + \dots + \frac{A^L\delta^L}{L!} \quad (21)$$

for the exponential matrix is obtained by truncating the power series expansion for the exponential matrix, equation (4), after L terms. A criterion is required for determining the number of

terms to keep. Cadzow and Martens (1970) give the following relation for calculating L :

$$L = \text{minimum of } \{3\|A\delta\|_\infty + 6\} \text{ or } 100 \quad (22)$$

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

$$\|A\delta\|_\infty = \text{Maximum} \sum_{j=1}^n |a_{ij}\delta_j| \quad (23)$$

Equations (21) and (22) can be used to calculate the exponential of a matrix with at least six digits of accuracy. From equation (22) 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 (23) to calculate $\|A\delta\|_\infty$
- 2 Find the smallest integer k such that $2^k \geq \|A\delta\|_\infty$
- 3 Divide all entries in the matrix $A\delta$ by 2^k
- 4 Determine L from equation (22) for $[A\delta/2^k]$
- 5 Calculate $e^{[A\delta/2^k]}$ from equation (21)
- 6 Square $e^{[A\delta/2^k]}$ k times to obtain $e^{A\delta}$

APPENDIX B

Integration of Integrals in Equation (7)

This appendix contains the steps for evaluating the two integrals in equation (7). Substituting the power series expansion for $e^{-A\alpha}$, equation (4), into the first integral in equation (7) results in

$$\int_0^\delta e^{A(\delta-\alpha)} d\alpha = e^{A\delta} \left[\int_0^\delta \left(I - A\alpha + \frac{A^2\alpha^2}{2!} + \dots \right) d\alpha \right] \quad (24)$$

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

$$\int_0^\delta e^{A(\delta-\alpha)} d\alpha = \left(I + A\delta + \frac{A^2\delta^2}{2!} + \dots \right) \left(I\delta - \frac{A\delta^2}{2} + \frac{A^2\delta^3}{3!} - \dots \right) \quad (25)$$

Multiplying the two power series in equation (25) and combining common terms results in

$$\int_0^\delta e^{A(\delta-\alpha)} d\alpha = I\delta + \frac{A\delta^2}{2!} + \frac{A^2\delta^3}{3!} + \frac{A^3\delta^4}{4!} + \dots \quad (26)$$

Equation (26) can be rewritten as

$$\int_0^\delta e^{A(\delta-\alpha)} d\alpha = A^{-1} \left(I + A\delta + \frac{A^2\delta^2}{2!} + \frac{A^3\delta^3}{3!} + \dots \right) - A^{-1} \\ = A^{-1}(e^{A\delta} - I) \quad (27)$$

Next, the second integral in equation (7) will be determined by following a similar procedure.

$$\int_0^\delta \alpha e^{A(\delta-\alpha)} d\alpha = e^{A\delta} \left[\int_0^\delta \left(I\alpha - A\alpha^2 + \frac{A^2\alpha^3}{2!} - \dots \right) d\alpha \right] \\ = \left(I + A\delta + \frac{A^2\delta^2}{2!} + \dots \right) \left(\frac{I\delta^2}{2} - \frac{A\delta^3}{3} + \frac{A^2\delta^4}{8} - \dots \right)$$

$$= \frac{I\delta^2}{2!} + \frac{A\delta^3}{3!} + \frac{A^2\delta^4}{4!} + \frac{A^3\delta^5}{5!} + \dots$$

$$= A^{-1}A^{-1} \left(I + A\delta + \frac{A^2\delta^2}{2!} + \dots \right) - A^{-1}A^{-1} - A^{-1}\delta \\ = A^{-1}A^{-1} \left(e^{A\delta} - I \right) - A^{-1}\delta \quad (28)$$

APPENDIX C

Leverrier's Algorithm

Wilberg (1971) presents a proof of Leverrier's algorithm for calculating the inverse of the $(FI - \Phi)$ matrix. Leverrier's algorithm for calculating the e scalar constants and R matrices in equation (15) consists of the following sequential relationships:

$$\begin{aligned} R_0 &= I & e_1 &= \frac{\text{Tr}(\Phi R_0)}{1} \\ R_1 &= \Phi R_0 + e_1 I & e_2 &= -\frac{\text{Tr}(\Phi R_1)}{2} \\ R_2 &= \Phi R_1 + e_2 I & e_3 &= -\frac{\text{Tr}(\Phi R_2)}{3} \\ & \vdots & & \vdots \\ & \vdots & & \vdots \\ R_{n-1} &= \Phi R_{n-2} + e_{n-1} I & e_n &= -\frac{\text{Tr}(\Phi R_{n-1})}{n} \end{aligned}$$

where $\text{Tr}(G)$ is the trace of the matrix G . The trace of a matrix is equal to the sum of the diagonal elements of the matrix.

APPENDIX D

Efficient Calculation of Transfer Functions

A large amount of computer memory would be required to store the $n \times n \times n$ R matrices if Leverrier's algorithm were used to compute transfer function coefficients. Fortunately, the storage requirement can be reduced to two $n \times n$ 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. The absolute values of the e_j coefficients decrease as j increases. This fact can be used as a criterion 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 $n \times n$ R matrices need to be stored):

- 1 Compute the exponential matrix $\Phi = e^{A\delta}$
- 2 Use equation (10) to compute Γ_1 and Γ_2
- 3 Use equation (20) to compute S_0
- 4 $R_{\text{new}} = I$
- 5 For $j = 1$ to $n - 1$ with a step size of 1

$$R_{\text{old}} = R_{\text{new}} \\ e_j = -\frac{\text{Tr}(\Phi R_{\text{old}})}{j}$$

$$R_{\text{new}} = \Phi R_{\text{old}} + e_j I$$

$$S_j = C[R_{\text{old}}(\Gamma_1 - \Gamma_2) + R_{\text{new}}\Gamma_2] + e_j D$$

Stop if the absolute value of e_j is less than a tolerance limit

$$6 \quad e_n = -\frac{\text{Tr}(\Phi R_{\text{new}})}{n}$$

$$7 \quad S_n = CR_{\text{new}}(\Gamma_1 - \Gamma_2) + e_n D$$

APPENDIX E

Two-Node Example

This appendix 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. 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 to discretize the problem spatially. A two-node finite-difference model can be seen in Fig. 7.

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

$$C \frac{dT_1}{d\tau} = hA(T_{out} - T_1) + \frac{T_2 - T_1}{R} \quad (29)$$

$$C \frac{dT_2}{d\tau} = hA(T_{in} - T_2) + \frac{T_1 - T_2}{R} \quad (30)$$

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

$$R = \frac{L}{kA} \quad (31)$$

The thermal capacitance of a node can be calculated by

$$C = \frac{\rho cLA}{2} \quad (32)$$

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

$$q'' = h(T_2 - T_{in}) \quad (33)$$

Equations (29) through (33) can be formulated in the following state space representation by letting the temperatures of the nodes be the two states:

$$\begin{bmatrix} \frac{dT_1}{d\tau} \\ \frac{dT_2}{d\tau} \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC} - \frac{hA}{C} & \frac{1}{RC} \\ \frac{1}{RC} & -\frac{1}{RC} - \frac{hA}{C} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{hA}{C} & 0 \\ 0 & \frac{hA}{C} \end{bmatrix} \begin{bmatrix} T_{out} \\ T_{in} \end{bmatrix}$$

$$[q''] = [0 \quad h] \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} + [0 \quad -h] \begin{bmatrix} T_{out} \\ T_{in} \end{bmatrix}$$

For a 0.3 m concrete wall with a density of 2200 kg/m³, specific heat of 0.84 kJ/kg-°C, thermal conductivity of 1.7 W/m-°C, and convection coefficients of 8.3 W/m²-°C, the matrices in the state space formulation become

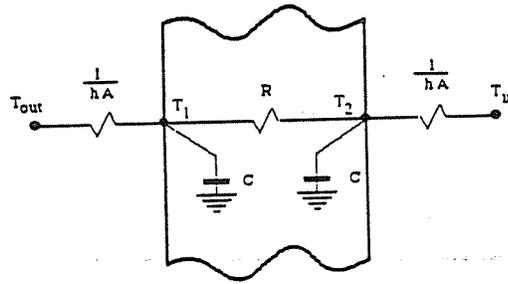


Fig. 7 Two-node model of plane wall

$$A = \begin{bmatrix} -0.1757 & 0.0714 \\ 0.0714 & -0.1757 \end{bmatrix} h^{-1} \quad B = \begin{bmatrix} 0.1043 & 0.0 \\ 0.0 & 0.1043 \end{bmatrix} h^{-1}$$

$$C = [0.0 \quad 8.3] W/m^2-^{\circ}C \quad D = [0.0 \quad -8.3] W/m^2-^{\circ}C$$

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. For this example, the exponential matrix with a 1 hour time step is

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

Carrying out the matrix manipulations described by equation (10) results in

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

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

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

$$\begin{aligned} (FI - \Phi)^{-1} &= \begin{bmatrix} F - 0.8410 & -0.0600 \\ -0.0600 & F - 0.8410 \end{bmatrix}^{-1} \\ &= \frac{\begin{bmatrix} F - 0.8410 & 0.0600 \\ 0.0600 & F - 0.8410 \end{bmatrix}}{(F - 0.8410)(F - 0.8410) - 0.0600^2} \\ &= \frac{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} F + \begin{bmatrix} -0.8410 & 0.0600 \\ 0.0600 & -0.8410 \end{bmatrix}}{F^2 - 1.6820 F + 0.7037} \end{aligned}$$

The constant coefficient matrices and scalar constants in equation (15) are

$$e_1 = -1.6820$$

$$e_2 = 0.7037$$

$$R_0 = I$$

$$R_1 = \begin{bmatrix} -0.8410 & 0.0600 \\ 0.0600 & -0.8410 \end{bmatrix}$$

Equation (20) can be used to compute the S matrices.

$$S_0 = CR_0\Gamma_2 + D = [0.0094 \quad -7.8820] \text{ W/m}^2\text{-}^\circ\text{C}$$

$$S_1 = C[R_0(\Gamma_1 - \Gamma_2) + R_1\Gamma_2] + e_1 D \\ = [0.0346 \quad 13.9866] \text{ W/m}^2\text{-}^\circ\text{C}$$

$$S_2 = CR_1(\Gamma_1 - \Gamma_2) + e_2 D = [0.0079 \quad -6.1566] \text{ W/m}^2\text{-}^\circ\text{C}$$

The transfer function equation for this two-node example is

$$q_t'' = S_0 u_t + S_1 u_{t-\delta} + S_2 u_{t-2\delta} - e_1 q_{t-\delta}'' - e_2 q_{t-2\delta}'' \\ = 0.0094 T_{t,out} + 0.0346 T_{t-\delta,out} + 0.0079 T_{t-2\delta,out} \\ - 7.8820 T_{t,in} + 13.9866 T_{t-\delta,in} \\ - 6.1566 T_{t-2\delta,in} + 1.682 q_{t-\delta}'' - 0.7037 q_{t-2\delta}''$$

ERRATA

Errata for "Natural Convection Heat Transfer From a Discrete Thermal Source on a Vertical Surface" by T. L. Ravine and D. E. Richards, published in the November 1988 issue of the ASME JOURNAL OF HEAT TRANSFER, Vol. 110, pp. 1007-1009:

Figure 2 was printed upside down. The correct orientation of the figure is shown below.

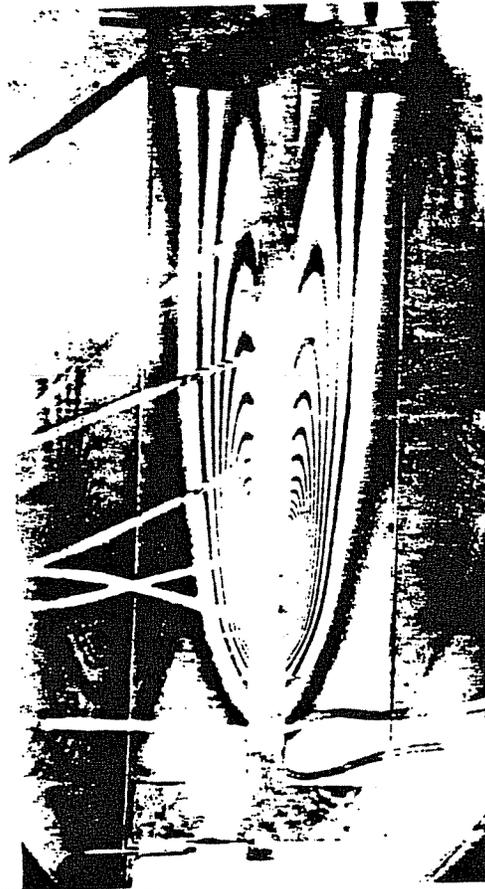


Fig. 2 Interferograms of a vertical flat plate with a discrete thermal source ($Ra_f = 5774$, $d/l = 2.00$, and $L/l = 9.00$)