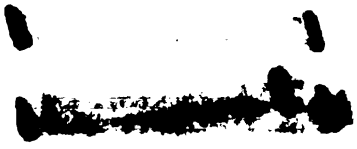
SIMULATION OF TEMPERATURE AND QUALITY PROFILES IN FROZEN FOODS SUBJECT TO STEP CHANGES IN STORAGE CONDITIONS

Dissertation for the Degree of Ph. D.

MICHIGAN STATE UNIVERSITY

ELAINE PATRICIA SCOTT

1987



SIMULATION OF TEMPERATURE AND QUALITY PROFILES IN FROZEN FOODS SUBJECT TO STEP CHANGES IN STORAGE CONDITIONS

By

Elaine Patricia Scott

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Agricultural Engineering

Department of Agricultural Engineering

1987

ABSTRACT

SIMULATION OF TEMPERATURE AND QUALITY PROFILES
IN FROZEN FOODS SUBJECT TO
STEP CHANGES IN STORAGE CONDITIONS

by

Elaine Patricia Scott

Frozen food products may be exposed to fluctuating ambient conditions during storage. An increase in storage temperature may result in an increase in the overall quality deterioration rate and/or a substantial quality differential within the food product. The overall objectives of this research were to develop a mathematical multi-dimensional model to simulate transient temperature dependent quality deterioration within a frozen food product subject to step changes in storage conditions, and to estimate surface heat transfer coefficients prevailing during step changes in storage conditions.

The temperature distribution history of the product, used in simulating the quality deterioration rate, was found numerically using finite differences. The transient surface heat transfer coefficients were estimated using experimentally determined temperature measurements in the sequential regularization method, developed for a class of problems called inverse heat conduction problems. A highly concentrated methyl-cellulose substance was used as an analog food substance in the experimental procedures. A systematic procedure was developed to select the optimal numerical parameters used in the finite difference and the

sequential regularization methods. The quality simulation model was used to determine the effects of various parameters on the quality deterioration rate. Parameters investigated included the magnitudes of the kinetic parameters and the surface heat transfer coefficient, the storage time and temperature, the magnitude of a step change in storage temperature, the food product dimensions, and the product geometry.

The sequential regularization procedure was found to provide estimates of the surface heat transfer coefficients which included the effects of the exterior packaging boundary and the accumulation and diminution of frost on the outer surface. Internal packaging boundaries were found to have a significant influence on the temperature differential within the food product. The magnitude of change in the quality deterioration rate was highly dependent on the magnitude of the kinetic parameters, and strongly influenced by product dimensions and the choice of a one or two dimensional heat transfer model.

Approved Dennis R. Heldman Feb. 21, 1987
Major Professor Date

Approved Donald D. Edwards Jr 2/26/87
Department Chairperson Date

To Mary Scott,

Thanks for your support,

Elaine

ACKNOWLEDGMENTS

I would like to give special thanks to Dr. Dennis R. Heldman, who continued to serve as my advisor after leaving Michigan State University. I greatly appreciated his guidance, support, and patience in helping me to complete this research. I would also like to thank the members of my Committee, Dr. J. V. Beck, Dr. L. J. Segerlind, Dr. J. F. Steffe, and Dr. M. A. Ubersax. Special thanks go to Dr. Beck, for his input on the Inverse Heat Conduction Problem, to Dr. Segerlind for his thoughts on determining the numerical parameters and for his humorous remarks, and especially to Dr. Steffe who handled the majority of the paper work in the absence of Dr. Heldman.

In addition, I want to thank my family, Mary, Mary Ann, and Charles, my long time friends, Cindy and Chris, and my new found Michigan friends, Sharon, Janice and Allyson, who helped me in more ways than they will ever know. Last, but not least, special thanks to Scott, who gave never ending support and encouragement through my frantic hours.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PAGE

LIST OF SYMBOLS

a - Constant

A - Surface Area (m^2)

A - Coefficient matrix for temperatures at time n+1

b - Constant

B - Coefficient matrix for temperatures at time n

C - Concentration of quality index

C - Heat Capacity Matrix

$c^{(e)}$ - Elemental Heat Capacity Matrix

CI - Confidence Interval

Cp - Specific heat (kJ/kg°C)

Cp(T) - Specific heat of frozen food as function of temperature(kJ/kg°C)

D - Vector containing known boundary conditions

D_e - Deterministic Error

Ea - Activation energy constant (kJ/mole)

g - Acceleration of gravity (m/s^2)

Gr_{Lx} - Grashof number

H - Enthalpy (kJ/kg)

H - Regularization difference matrix

hx - Surface heat transfer coefficient on boundaries along x-axis ($W/m^2 \cdot C$)

hx_{cv} - Heat transfer coefficient from forced and free convection ($W/m^2 \cdot C$)

hy - Surface heat transfer coefficients on boundaries along y-axis

$(W/m^2°C)$

J_i - i^{th} order Bessel function

k - Thermal conductivity (W/m°C)

K - Matrix containing thermal conductivity terms

$\mathbf{k}^{(e)}$ - Elemental matrix containing thermal conductivity terms

k(T) - Thermal conductivity of frozen food as a function of temperature (W/m°C)

kr - Rate constant (sec^{-1} or day^{-1})

L - Total number of nodes along x axis

Lx - Position along x axis of first boundary (m)

Lx_c - Characteristic Length (m)

Lx_o - Position along x axis of second boundary (m)

Ly - Position along y axis of second boundary (m)

M - Total number of nodes along y axis

m_j - Molecular weight of substance j (kg/mole)

M_j - Molecular mass of substance j

MC - Water content (%) of methyl-cellulose

n - Order of reaction

N - Total number of time steps

N_t - Number of thermocouples

Nu - Nussult number

Pr - Prandlt number

q - Surface heat flux (W/m^2)

$\mathbf{q}$ - Surface heat flux vector (W/m^2)

Q - Food quality (time or %)

$\mathbf{q}^*$ - Assumed heat flux vector (W/m^2)

Q_{o_r} - Initial food quality at reference temperature T_r(time)

R - Universal gas constant

R_j - Regularization terms (j = 0, 1, 2)

Ra - Raleigh number

Re_{Lx} - Reynolds number

S - Least squares function

S_i - Least squares function for i^{th} set of random numbers

S^* - Mean squared error per time step

t - Time (s)

T - Temperature (°C or °K)

$\mathbf{T}$ - Temperature Vector (°C or °K)

t_ℓ^∞ - Time for ℓ^{th} terms in summation be insignificant

t_{ss} - Time for solution to reach steady state conditions

T_{if} - Initial freezing temperature (°K)

T_r - Reference temperature (°K)

T_s - Surface temperature (°C)

$\mathbf{T}^*$ - Temperature matrix from assumed heat flux (°C)

T_o - Initial product temperature (°C)

U - Air free stream velocity (m/s)

V - Variance

W_j - Weighting factors for regularization terms (j = 0,1,2)

x - position along x axis (m)

X - Sensitivity coefficient (°C/W/m^2)

$\mathbf{X}$ - Sensitivity coefficient matrix (°C/W/m^2)

X_w - Mole fraction

x_0 - location of temperature sensor (m)

y - postion along y axis (m)

Y - Temperature measurements at x_0 (°C)

$\mathbf{Y}$ - Temperature measurements at x_0 vector (°C)

α - Regularization parameter

β - Weighting factor at time n

β_x - Weighting factor at time n in x-direction (2-D solution)

β_y - Weighting factor at time n in y-direction (2-D solution)

β_e - Expansion coefficient of air ($°K^{-1}$)

Γ - Latent heat of solvent

Δt - Time step

Δt_{ac} - Maximum time step for accuracy

Δt_m - Time increment for temperature measurements

Δt_{osc} - Limiting time step for no oscillations

Δx - Spatial step x direction

Δy - Spatial step y direction

ζ_ℓ - ℓ^{th} root of transcendental equation

η - Weighting factor at time n+1

η_x - Weighting factor at time n+1 in x-direction (2-D solution)

η_y - Weighting factor at time n+1 in y-direction (2-D solution)

ν - Kinematic viscosity (m^2/s)

κ - Thermal diffusivity (m^2/s)

λ_{max} - Maximum numerical eigenvalue ($°C^{-1}$)

$\lambda_{max}^{(e)}$ - Maximum elemental numerical eigenvalue ($°C^{-1}$)

λ_ℓ - ℓ^{th} numerical eigenvalue ($°C^{-1}$)

λ_ℓ^e - ℓ^{th} analytical eigenvalue ($°C^{-1}$)

π - Pi

ρ - Density (kg/m^3)

$\rho(T)$ - Density of frozen food as a function of temperature (kg/m^3)

σ - Standard deviation of temperature measurements (°C)

σ_S - Standard deviation of least squares function (°C)

σ_S^* - Standard deviation of least squares function per time step (°C/sec)

Φ - Arbitrary temperature function

Subscripts

a - air

c - carbohydrate

D - without input errors

F - forced convection

i - ice

lp - lipid

Lx - second boundary in x-direction

Lx_0 - first boundary in x-direction

Ly - second boundary in y-direction

m - mineral

N - free convection

p - unfrozen food product

pb - paper board

pf - plastic film

pk - packaging material

pr - protein

s - solids

V - with input errors

w - water

wi - water-ice component

wis - water-ice-solids component

wo - unfrozen water

0 - first boundary in y-direction

1 - first storage period

2 - second storage period

1_∞ - ambient conditions for first storage period

2_∞ - ambient conditions for second storage period

$-$ - at a location between m^{th} and $(m-1)^{th}$ node

$+$ - at a location between m^{th} and $(m+1)^{th}$ node

Superscripts

$\bar{}$ - Mean

$\hat{}$ - Estimated

CHAPTER 1

INTRODUCTION

Freezing is one of the most important methods of food preservation used in the United States. The freezing process cannot improve the quality of a food product; the reduction of temperature in a food product only results in the retardation of the processes which are detrimental to product quality, such as enzyme activity, microbial growth and chemical reactions. The overall quality of the food product may be affected during pre-treatment (post harvest handling and preparation), freezing, and post-freezing handling (transportation, storage and distribution). However, with proper pre-treatment and freezing, the majority of the quality reduction occurs during the post-freezing handling phase of the overall freezing process.

The rate of quality loss during this phase is primarily temperature dependent; changes in temperature during the post-freezing phase may result in a reduction in storage or shelf-life for the product (Singh and Wang, 1977). Zaritzky (1982) cited two types of temperature changes a frozen food product may be exposed to during the post-freezing phase: (1) fluctuations in the temperature of the storage chamber, and (2) sudden increases in temperature during loading and unloading of the product during transportation and distribution. In both of these cases, a rise in temperature may lead to an undesirable loss of product quality. Additionally, since frozen food products are commonly stored in large pallet loads, a sudden change in temperature may result in a

higher rate of quality loss at the surface of the pallet load than at the center.

Since product quality is largely a result of the temperature history of the product after freezing, accurate methods of predicting temperature distribution histories within the product as a result of fluctuating storage temperatures are important in estimating final product quality.

Substantial research has been devoted to developing analytical and numerical models for the simulation of freezing in foods and the estimation of freezing times (Plank, 1913, Hayakawa and Bakal, 1972, Charm et. al., 1972, and Cleland and Earle, 1977a). Only a limited amount of work has focused on extending these studies, particularly the numerical models, to simulating the food product during the post-freezing phase. Although many similarities exist in the numerical analysis of the two problems, simulation of the post-freezing phase differs from the freezing process in several ways: first, the heat transfer coefficients prevalent during the freezing phase are generally much higher than those found in the post-freezing phase. Second, the size of the body considered also differs. During freezing, the individually packaged product is considered, while during the post-freezing phase the product is often palletized, and consequently, the pallet load is the object of consideration. In addition, the duration of the freezing process is generally much shorter than the duration of the post-freezing phase. These differences in the rate of heat transfer, size of body, and duration time, suggest that care must be taken in adapting numerical methods developed for the freezing process to the post-freezing process.

The heat transfer rate depends on the temperature differential between the surrounding environment and the product, and the heat transfer coefficient. Several researchers, including Bonacina and Comini

(1972) and Chavarria and Heldman (1983) have investigated the measurement of the heat transfer coefficient during the freezing stage, but little data can be found on its value during the post-freezing stage. Values were measured by Dagerskog (1974), but only for steady-state conditions. To gain a full understanding of the post-freezing situation, it is desirable to estimate the heat transfer coefficient for both transient and steady-state heat transfer.

Estimation of internal frozen food temperatures is essential in estimating final product quality. If a first order quality deterioration mechanism is assumed, the product quality loss my be predicted from internal temperatures and known kinetic parameters.

In this study, the problem of determining temperature and quality distributions in food products subject to step changes in ambient temperatures in the post-freezing phase was considered. A numerical method was developed to simulate one or two dimensional heat transfer, and accommodate regular geometric shapes; a rectangle, cylinder or sphere. Additionally, a procedure is presented to estimate surface heat transfer coefficients from internal product temperature measurements throughout transient and steady state heat transfer. The predicted temperature distribution histories are used to estimate product quality loss, assuming a single quality deterioration process exists which limits the overall quality degradation rate.

The uniqueness of this study lies in the coupling of temperature history with quality loss estimation, in the development of a numerical model to accommodate the conditions inherent to the post-freezing process, and in the estimation of the transient and steady-state heat transfer coefficients.

1.1 Objectives

The principal objectives of this study were:

1. To develop a generalized one dimensional mathematical model to simulate transient temperature and quality distributions within frozen food products subject to step changes in storage conditions, using the implicit Crank-Nicolson finite difference scheme, and assuming a single limiting quality reaction exists, which governs the overall quality degradation rate.
2. To extend the one dimensional analysis to two dimensions using an alternating direction implicit (ADI) finite difference scheme.
3. To estimate the surface heat transfer coefficient of a frozen food product during step changes in storage conditions as a function of time, from discrete ambient and internal product temperature measurements.
4. To analyze the influence of the product boundary conditions on quality loss during storage. Parameters affecting the product boundary conditions include ambient temperature, the heat transfer coefficient, and, for step changes in storage conditions, the time interval at each storage temperature, and the magnitude of the step change in storage temperature.
5. To analyze the effects of product dimensions and kinetic parameters on the quality distribution. Specifically, it was desired to determine under what conditions the temperature and quality distributions are uniform throughout the body, and a lumped (uniform temperature) model can be used.
6. To determine the influence that geometry and the choice of a one or two dimensional model has on temperature and quality distributions. This includes determining under what conditions a one dimensional

model can be used to approximate two dimensional heat flow, that is, to determine when heat flow in an infinite cylinder can be used to approximate two dimensional heat flow through an infinite rectangular rod.

CHAPTER 2

LITERATURE REVIEW

2.1 Quality Loss in Frozen Foods during Storage

Freezing is an important method of preserving food products for later consumption. This process reduces the rate of quality loss by restricting enzyme action, reducing chemical reaction rates and inhibiting microbial growth. The process does not completely preserve food quality; detrimental changes continue at a reduced rate dependent upon storage temperature and type of product.

Physical and chemical changes are the primary factors affecting the overall quality of frozen foods. Important detrimental physical changes cited by Singh and Wang (1977) are ice crystallization with volume expansion, and desiccation at the surface of the frozen food product. Fluctuations in storage temperature increase the rate of desiccation at the surface of the product, especially in improperly packaged foods, resulting in dry, brown spots, particularly in poultry products, commonly referred to as "freezer burn".

Chemical changes occurring during frozen storage, as described by Fennema et. al. (1973), are lipid oxidation, enzymatic browning, flavor deterioration, protein insolubilization and degradation of chlorophyll and vitamins.

Most of the existing data on quality changes in frozen foods represent the allowable or tolerable time and temperature conditions for a

specified quality retention. These tests, initiated by Van Arsdel (1957), are commonly known as the time-temperature-tolerance (TTT) experiments. The Western Regional Research Center, Berkeley, California continued these tests into the early sixties on a great many fruit, vegetable, and poultry products (Jul (1984)).

Several mathematical models have been developed to predict the quality change of frozen foods during storage based on this data. Schwimmer et. al. (1955) developed series relationships for quality losses resulting from periodic storage temperature fluctuations. Van Arsdel and Guadagni (1959) presented a procedure to predict quality changes resulting from known irregular temperature fluctuations through graphical integration of temperature history curves. In the case where a limiting reaction affecting quality exits, kinetic theory may be used to describe the change in quality at a constant temperature. The loss of quality with storage time, at a given temperature may be found from:

$$-\frac{dC}{dt} = kr \cdot C^n$$

where C = concentration of quality index

t = time

kr = rate constant

n = order of the reaction

It was suggested by Charm (1971) that if a limiting quality factor exists, the empirical equation developed by Arrhenius (1889) can be used to describe the effect of temperature on the rate constant. Singh (1976) also utilized the Arrhenius equation to describe the first order reduction of a single component in a product. The kinetics of quality change in frozen foods were analyzed by Lai and Heldman (1982) in an

effort to apply kinetic models to TTT data found in the literature. In a related study, Heldman and Lai (1983) developed a model based on the Arrhenius equation where the reaction order need not be considered. Statistical methods for the computation of kinetic parameters used in the Arrhenius equation from existing TTT data were developed by Chu (1983), Haralampu et. al. (1985), and Cohen and Saguy (1985). Ross et. al. (1985) developed shelf-life prediction models based on nonlinear regression and contingency-table methods. Singh and Heldman (1976) modeled the diffusion of oxygen accompanied by a second order chemical reaction with ascorbic acid to simulate food quality loss in liquids during storage. Bhattacharya and Hanna (1986) estimated rate constants, the order of reaction and the activation energy constant for texture degradation of frozen beef during storage.

Jul (1984) warned against the use of mathematical models based on a single quality factor in particular situations where the rate of quality deterioration may be a result of several factors and no single limiting reaction exists.

The effects of temperature fluctuations during storage on product quality has been the subject of research for a number of years. Hustrulid and Winter (1943) reported fluctuating storage temperatures below -15°C had no great influence on the appearance and/or palatability of the products studied. Gortner et. al. (1948) compared quality loss in frozen food products (pork, strawberries, snap beans, and peas) subject to three different storage conditions. One storage compartment was maintained at -17.8°C, the second was held at -12.2°C, and the third fluctuated between -17.8°C and -6.7°C in a six day cycle. Quality losses in the products held at -12.2°C and in the fluctuating compartment were comparable, based on palatability, thamine, and perioxide content in pork, and ascorbic acid loss in fruits and vegetables.

A number of the TTT studies, completed during the late fifties and early sixties, investigated the effects of temperature fluctuations during storage. Dietrich et. al. (1960) found, in a study of frozen snap beans, that when storage temperatures were varied in patterns, deterioration was found to be a summation of constant temperature increments. In an investigation of ready-to-cook cut-up chicken, Klose et. al. (1959) conducted constant storage tests at -6.7°C, -12.2°C, -17.8°C, -27.8°C and -34.4°C, and periodic storage tests between -17.8°C and -6.7°C, and between -27.8°C and -12.2°C, and found only slightly greater deteriorative effects on quality in the fluctuating storage tests than found at the equivalent arithmetic mean temperature. Comparable results, using similar test conditions, were obtained by Boggs et. al. (1960) in a TTT study of frozen peas, and by Dietrich et. al. (1962) in an investigation of quality changes in cauliflower. Fennema and Powrie (1964) discussed the lack of evidence to extend the conclusions found in the TTT studies to the texture of fruits, and called for more investigations on the effects of fluctuating temperatures on fruit texture.

Ashby et. al. (1979) studied energy savings resulting from periodic fluctuating storage temperatures ranging from -23 to -15°C. For storage periods greater than six months, it was determined that the product temperature should not rise above -18°C, and should not fluctuate more than 3°C. Moleeratanond et. al. (1981) conducted similar studies on the energy consumption of a fluctuating temperature storage regime and its effects on quality changes in frozen boxed beef. Results indicated that product quality was not seriously affected in peripheral pallet locations, provided the temperature was maintained at less than -18°C and the maximum fluctuation did not exceed 3°C.

Sastry and Kilara (1983) reflected the need for analysis of heat transfer in frozen foods exposed to periodic storage conditions to determine quality variations within a given pallet load.

2.2 Simulation of Transient Heat Conduction in Frozen Foods during Storage

Considerable attention has been given to simulation of freezing in food products within the last two decades. Only recently have researchers begun to investigate the thermal behavior of frozen foods during distribution and storage. In many instances, the methods utilized in freezing studies may also be utilized in storage studies. Freezing simulation models may be categorized with regards to the results generated in two groups: (1) those producing freezing time estimations, and (2) those producing temperature distribution histories within the product. Only the latter group is of interest in storage simulation studies and is included in this review.

2.2.1 One Dimensional Analysis

Most of the models developed to simulate freezing or frozen food storage are based on one dimensional heat transfer analysis. Analytical and numerical techniques have been proposed to estimate temperature distribution histories in both the freezing and post-freezing stages.

An analytical solution to the freezing phenomenon involving a pure liquid was presented by Carslaw and Jaeger (1959). Komori and Hirai (1970) provided an analytical solution of the freezing problem in cylindrical coordinates, with the single, unique temperature and only at the solid-liquid interface. Tien and Geiger (1967) developed an

analytical solution to the solidification of a binary eutectic system, assuming three distinct regions: a solid, liquid, and a liquid-solid region in which the solid fraction is linear with position. Grange et. al. (1976) obtained an approximate analytical solution for freezing of salt solutions using an integral method, and assuming latent heat is released at a constant temperature.

A food product is a solution or mixture, however, and freezing does not occur at a single distinct temperature. Instead, the initial freezing temperature of the mixture is depressed compared to that of the pure substance, such as water (Heldman, 1982, and Chen, 1986). As the mixture freezes, the liquid portion becomes more concentrated with solute, and the freezing point is depressed further. As a result, latent heat is produced over a range of temperatures, and thermal properties, which vary according to the solid-liquid composition, are temperature dependent.

Several researchers have used analytical methods in developing solution techniques with modification to allow for temperature dependent thermal properties. Sastry and Kilara (1983) approximated constant thermal properties over small temperature ranges using an "apparent" thermal diffusivity which includes latent heat terms. An analytical solution of the linear one dimensional heat conduction problem with designated sinusoidal temperatures at the boundaries was then obtained to simulate the temperature response of frozen peas in fluctuating temperature storage conditions. Zaritzky (1982) developed both analytical and numerical models to simulate the thermal behavior of frozen meat during its storage and distribution. In the analytical model, average values for the thermal properties, including the effects of latent heat, were again used, but in this case, a boundary condition of the third kind with sinusiodal ambient temperatures was imposed. One dimensional

analytical solutions were multiplied together to generate two and three dimensional models, and results were compared with experimental data. In a related study using similar boundary conditions, Zuritz, et. al. (1986) simulated temperature fluctuations within frozen foods stored in cylindrical containers.

Zuritz and Sastry (1986) determined the effects of packaging materials on temperature fluctuations in frozen foods using an analytical model to calculate the temperature distribution histories, resulting from an imposed sinusiodal ambient temperature at the surface, and assuming constant thermal properties.

Many different approaches have been used in developing mathematical models of the freezing and post-freezing phases. De Michelis and Calvelo (1982) used a simplified model which uses three distinct precooling, freezing and tempering phases. An analytical solution with constant coefficients is obtained for the precooling and tempering phases, and the freezing phase is simulated assuming steady state heat transfer and constant coefficients. Chen et. al. (1984) used a method of lumping to incorporate diffusivity and latent heat terms into a temperature dependent 'effective' diffusivity. The resulting equation was solved using finite differences. Sanz, et. al. (1986) applied the z-transfer function method to predict temperature - time history of food stuffs during chilling and cold storage. In this procedure, the z-transfer coefficients are obtained be means of an experimental method.

A number of researchers used methods which assumed latent heat is released at a fixed freezing point. Charm et. al. (1972) assumed latent heat was released at a constant temperature over a specified region. Grange et. al. (1976) also assumed latent heat at a distinct temperature. An implicit finite difference solution was obtained assuming

variable thermal properties and compared with an approximate analytical solution.

Dix and Cizek (1971) solved the heat conduction problem replacing the usual dependent variable, temperature as a function of position and time, with the isotherm position as a function of temperature and time. This technique is termed the 'isotherm migration method' (IMM). The solution was obtained explicitly using finite differences and variable thermal properties. Chernous'ko (1970) also developed a similar methodology using isotherms for the solution of the nonlinear heat conduction problem with phase change. Talmon and Davis (1981) utilized the previously developed IMM methods in developing a new technique called the 'modified isotherm migration method' (MIMM). Unlike previous IMM methods, the MIMM uses a moving front boundary condition in the governing differential equation. Mastanaiah (1976) also incorporated a moving front boundary condition, this time by use of the transformation of coordinates. Temperature was maintained as the dependent variable, and the solution was obtained using the Crank-Nicolson finite difference method (Ozisik, 1980).

A significant number of researchers have incorporated the latent heat into an 'apparent' specific heat. Freezing is assumed to take place over a range of temperatures, consequently, all thermal properties are assumed to be temperature dependent. Lescano (1973) used the Crank-Nicolson finite difference technique to simulate freezing in codfish. Heldman (1974a) also used the Crank-Nicolson finite difference method to simulate the freezing process in spherical food products. The Kopelman (1966) equation describing the relationship of thermal conductivity with the temperature dependent product composition was implemented in this simulation model. Bonacina and Comini (1973a) used a second order accurate three level time scheme originally proposed by

Lees (1966) for the solution of the transient heat conduction equation with temperature dependent parameters. Bonacina et. al. (1973) extended this work to account for phase-change by including latent heat terms into the specific heat. Cleland and Earle (1984) noted the advantages of Lee's scheme, in that the thermal properties are evaluated at the mid-point time level instead of at the beginning time level as done in other methods, such as the Crank-Nicolson method. Tarnawksi (1976) developed finite difference equations for simultaneous heat and mass transfer in frozen food products. Zaritzky (1982) used the Douglas-Jones method of finite differences (Von Rosenberg, 1969) to simulate frozen meat in storage, and compared results with an analytical method discussed previously.

An alternate approach was presented by Joshi and Tao (1974). In this procedure, the finite difference equations were written in term of the enthalpy, and these were solved implicitly by assuming an exponential relationship between enthalpy and temperature.

2.2.2 Multi-Dimensional Analysis

Various methods have been proposed for the numerical solution of a nonlinear two dimensional heat conduction problem. Most of the solution methods may be categorized as finite element or finite difference solutions. The finite element method was used to solve transient, nonlinear heat transfer problems by De Baerdemaeker et. al. (1977) in axi-symmetric products, and by Zuritz and Singh (1985) in modeling temperature fluctuations in stored frozen foods. Comini et. al. (1974) utilized finite elements in a three dimensional analysis of a brick shaped body, including convective and radiative boundary conditions. Two dimensional finite element techniques were also utilized by De

Cindio et. al. (1985) in the analysis of ice cream brickettes and by Rebellato et. al. (1978) in the freezing of meat carcasses. Lewis et. al. (1984) applied an alternating-direction finite element scheme to the freezing problem with a substantial savings in computation time and comparable accuracy to standard schemes.

Finite difference methods were used as an alternate approach. Most of these methods may be classified as explicit, implicit or alternating-direction techniques. Dagerskog (1974) used an explicit finite difference method in three dimensions to simulate temperature distributions in foods during handling and storage. This method was severly limited in its usefulness by the stability condition on the time step. Implicit solutions to the two dimensional transient nonlinear heat conduction problem require the inversion of large matrices at each time step, requiring a substantial amount of computation time (Anderson et. al. (1984)). To overcome the difficulties of solving the two dimensional problem using explicit or implicit techniques, an alternating-direction implicit (ADI) scheme with second order accuracy was developed by Peaceman and Rachford (1955).

The ADI method involves a two step scheme, where the temperature field is determined in different directions for each time step. This results in the inversion of two tridiagonal matrices at each time step, for which efficient algorithms exist. Douglas and Gunn (1964) developed a general ADI method for two and three dimensions utilizing a Crank-Nicolson scheme which is of second order accuracy and unconditionally stable. Allada and Quon (1966) developed a stable explicit multidimensional alternation direction solution for nonhomogeneous media. Fleming (1973) utilized the Peaceman-Rachford method in simulating the freezing process with temperature dependent thermal properties. Bonacina and Comini (1973b) applied Lee's tri-level scheme in alternating directions

to simulate food freezing with two dimensional heat transfer. Evans and Gane (1978) solved the transient heat conduction problem for an annular ring, using the Peaceman-Rachford ADI method.

Alternate approaches to the finite difference solution to the transient heat conduction problem include the splitting or fractional-step methods discussed by Yanenko (1971) and developed by Soviet mathematicians about the same time ADI methods were developed in the United States. The modified box method for the heat equation and the hopscotch methods are two additional methods discussed by Anderson et. al. (1984). The modified box method is second order accurate even with variable grid spacing. The hopscotch method is a first order accurate two step alternating explicit-implicit scheme.

2.3 Estimation of the Surface Heat Transfer Coefficient

The prediction of temperature profiles within a frozen food substance during storage requires knowledge of the resistance to heat transfer between the product and the cooling medium. This resistance is characterized by a surface heat transfer coefficient (h), which may be dependent on time and/or position.

The importance of the surface heat transfer coefficient in estimating freezing times was discussed by Heldman (1974a), Hsieh et. al. (1977) and Tarnawski (1976), but little effort has been directed toward investigating the effects of the heat transfer coefficients during storage conditions.

Some simple steady state solutions for the heat transfer coefficient resulting from forced convection have been developed for regular geometries. Kays and Crawford (1980) presented solutions for constant free stream velocity flow over a constant-temperature or arbitrarily

specified temperature semi-infinite flat plate, and for flow over a semi-infinite plate with an arbitrarily specified surface heat flux.

Since the Reynolds numbers encountered in storage conditions are typically low due to low air velocities, heat transfer due to natural or free convection may also be a significant factor. Solutions for the heat transfer coefficient resulting from natural convection of a hot or cold horizontal surface facing up were given by McAdams (1954), and modified by Goldstein, et. al. (1973). In addition, solutions for the heat transfer coefficient resulting from induced flow parallel to a vertical wall were presented by Kays and Crawford, (1980).

In the situations where both free and forced convection effects are comparable, correlating equations to include both forced and free convection effects have been developed by Churchill (1977, 1983). The free convection factor will tend to either enhance or decrease the forced convection effect, depending on whether or not the bouyancy force opposes or aids the forced convective motion (Kays and Crawford, 1980, and Incropera and DeWitt, 1985).

Estimation of surface heat transfer coefficients using analytical methods is very difficult, especially for other cases, such as, transient heat transfer, odd shaped geometries, and irregular flow patterns (Lightfoot et. al., 1965). Consequently, researchers have resorted to using experimental techniques in the estimation of surface heat transfer coefficients, and many different methodologies have evolved.

Several researchers have used metal transducers to estimate heat transfer coefficients during freezing. Due to the high thermal conductivity of the metal, the transducer is assumed to have negligible internal resistance to heat transfer. Therefore, the surface heat transfer coefficient may be obtained from a logarithmic plot of dimensionless time against dimensionless temperature. Lescano (1973)

utilized geometric and kinematic similarity in using aluminum transducers to simulate heat transfer through codfish fillets during freezing. Creed and James (1985) used copper transducers in predicting heat transfer coefficients associated with plate freezers. In their study, estimations of the influence of packaging materials were made by placing a layer of the material between the transducer and the cooling medium.

Bonacina and Comini (1972) used nonlinear regression between calculated and measured temperatures to predict surface heat transfer coefficients. In this method the surface heat transfer coefficient was assumed to be constant, that is, not a function of time and/or position. Comini (1972) extended this study to investigate the design of optimum transient experiments for the determination of the surface heat transfer coefficient. These studies were based on work by Beck (1967, 1969). Beck utilized sensitivity coefficients in estimating thermal contact conductance, and in determining optimum, transient experiments for estimating conductance coefficients. Chavarria and Heldman (1983) also used nonlinear regression in estimating a convective heat transfer coefficient for ground beef during freezing. The coefficient was assumed to be constant during the freezing process, and heat transfer was assumed to be one dimensional. Succar and Hayakawa, (1986) used a surface response method for the estimation of convective and radiative heat transfer coefficients as a function of time during freezing and thawing of frozen foods. In this method, experimental and predicted temperatures were minimized using the method of least squares.

Cleland and Earle (1976) presented a new method of estimating heat transfer coefficients from surface temperature measurements of a transducer with a thermal conductivity closely resembling a food product. Some of the restrictions of this method were that the

pre-cooling part of the freezing curve be sufficiently long, and that the center temperature of the body be unaltered for several minutes after the onset of cooling. Different numbers of cardboard sheets between Tylose samples and the cooling medium were used by Cleland and Earle (1977b) to estimate the relationship between the heat transfer coefficient and the number of sheets (thickness of packaging material).

The previous methods assume the heat transfer coefficient is constant in the solution (with the exception of Succar and Hayakawa, 1986, who investigated transient heat transfer coefficients during freezing and thawing). Beck et. al. (1985) presented method of estimating heat transfer coefficients as a function of time using ambient temperatures and temperature measurements from a sensor located inside the body. In this solution, the problem is treated as part of a class of problems called inverse heat conduction problems (IHCP). In the solution of the IHCP, the boundary conditions are determined instead of the internal temperature distribution which is found in the direct solution. Various methods have been proposed to solve the inverse heat conduction problem of determining a boundary condition at the surface of a body from discrete temperature measurements. Exact analytical solutions were developed by Burgraff (1964) and Langford (1976). These methods require continuously differentiable data. Stolz (1960) provided one of the earliest solutions to the IHCP, which was found to be unstable with small time steps. A similar method involving the numerical inversion of a convolution integral and utilizing future time steps was developed by Beck (1968). This method provides a solution at each time step, and is called the sequential function specification method. Osman and Beck (1987) used the sequential function specification method in estimating heat transfer coefficients as a function of position, using a spherical coordinate system. Other integral methods using Laplace transforms have

been demonstrated in one dimensional form by Imber and Khan (1972), and in two dimensional form by Imber (1974). Weber (1981) replaced the traditional heat conduction equation by a hyperbolic one to obtain a well-posed problem with established solution techniques.

Regularization methods were proposed by Miller (1970), and Tikhonov and Arsenin (1977). These methods provide stability by the addition of smoothing factors and reduce the influence of measurement errors in the data. The influence of the regularization component is determined by the magnitude of a regularization parameter. Different criteria are found in the literature for the selection of this parameter. Tikhonov and Arsenin (1977) and Reinsch (1967) base their criteria on the errors in the measurements, while Murio (1985) considers in addition a bound based on the square of the L_2 norm of the heat flux vector. Hills and Mulholland (1979) applied the method of Backus and Gilbert (1970) to a transient heat conduction problem. This method, adapted from geophysics, also utilizes smoothing function to stabilize the solution.

Beck and Murio (1986) presented a new method which combines the sequential function specification procedure with the regularization method. This method differs from the global regularizaton methods in that the solution is found sequentially, greatly improving computational efficiency. This method was shown to be very competitive with the global regularization methods in terms of the heat flux estimates.

Difference methods have been used to solve the nonlinear IHCP, which cannot be solved using integral methods. Methods utilizing finite differences were demonstrated by Blackwell (1981), Beck (1970), Beck et. al., (1982), and Williams and Curry (1984). In Beck's methods the same concepts are used to develop the algorithms as were used for the convolution based methods. A stabilizing matrix was utilized by Hensel and Hills (1984) in developing a space marching finite difference algorithm.

Finite elements were incorporated in the solution by Krutz, et al. (1978), and Bass (1980). It is important to note that the solution of the linear IHCP with the function specification and regularization methods are independent of the method of solution of the heat conduction equation because whether numerical convolution, finite differences or finite elements are used, nearly identical solutions are obtained (provided accurate approximations are used in each case).

CHAPTER 3

THEORETICAL CONSIDERATIONS

The three major problems analyzed in this study are: (1) the determination of temperature profiles of food products in storage, (2) the estimation of the surface heat transfer coefficients encountered during storage conditions, and (3) the prediction of quality profiles within food products during storage. The determination of temperature profiles from known boundary conditions is called a direct problem, and it is considered to be mathematically well-posed. The surface heat transfer coefficient was estimated from internal product temperature measurements. This is called an indirect problem, and it is ill-posed. The three problems are interrelated in that the surface heat transfer coefficients are required as input for the direct problem; the temperature profiles resulting from the solution of the direct problem are required for the prediction of the quality profiles; and, the numerical solution of the direct problem is inherent in the solution of the indirect problem. In both the one and two dimensional solutions for the temperature profile, and the estimation of the surface heat transfer coefficient, determination of thermal properties as a function of temperature is required. Also, both the two dimensional direct problem and the one dimensional indirect problem use the solution of the one dimensional direct problem as a fundamental building block in the numerical analysis.

In the following sections, the evaluation procedure for the thermal properties is presented first, followed by an analysis of the one dimensional direct problem. The results in both of these sections are important in the solution of the two dimensional direct problem and the one dimensional indirect problem. The analyses and numerical procedures for these two problems are presented in the succeeding sections. Finally, the methods used to evaluate quality deterioration from calculated temperature distribution histories within the food product are presented.

3.1 Thermal Properties

Food products are primarily composed of water which contains various solutes. Due to the presence of solutes, the initial freezing point is depressed, compared with that of pure water. Consequently freezing occurs over a range of temperatures, and unbound liquid water can be present at temperatures associated with storage and distribution. The changing water fraction over a range of temperature results in temperature dependent thermal properties in frozen food products. Accurate prediction of thermal properties is very important in estimating temperature distributions within the product.

3.1.1 Unfrozen Water Fraction

The relationship between the unfrozen water fraction and temperature is based on the equality of the chemical potentials in different phases within a system (Heldman, 1974b). The underlying assumptions of this

derivation are (1) the solution is dilute, and (2) the conditions approach that of an ideal binary system. The derivation of this relationship results in the following equation (Moore, 1962)

$$\ln X_w = \frac{\Gamma}{R}\left(\frac{1}{T_{if}} - \frac{1}{T}\right) \tag{3.1}$$

where, X_w is the mole fraction at absolute temperature (T), which is found from an experimentally determined initial freezing point (T_{if}), the latent heat of the solvent, (Γ), and the universal gas constant (R). Substituting this value into the definition of mole fraction, shown below in Eq. (3.2), the effective molecular weight of the product solute may be found (Heldman, 1974b)

$$X_w = \frac{M_{wo}/m_w}{M_{wo}/m_w + M_s/m_s} \tag{3.2}$$

where m_w and M_{wo} refer to the molecular weight and mass of the unfrozen water, respectively. In a food product, the product solute and solids are assumed to be indistinguishable; therefore, the mass and effective molecular weight, (M_s and m_s), are of the combined solute and solids, and are hereby referred to as the mass and molecular weight of the solids. Furthermore, due to the binary solution assumption, the individual effects of the carbohydrate, lipid, protein and mineral components of the food product are lumped together in the effective molecular weight of the solids (m_s).

By equating Eqs. (3.1) and (3.2) and substituting different values for temperature, T, the mole fraction of unfrozen water may be found for temperatures below the initial freezing temperature. Since thermal properties are dependent on the relative amount of each component in the

food product, knowledge of the frozen and unfrozen water fractions as a function of temperature allows for the estimation of thermal properties during freezing and storage.

3.1.2 Density in Frozen Foods

The temperature dependence of density in frozen foods can be predicted from the relative amounts of solids, liquid water, ice, and in some cases, air present in the product. The following relationship, including the air fraction contribution, is based on the density model without air utilized by Heldman and Gorby (1975a), Hsieh et. al. (1977), and Perez (1984)

$$\frac{1}{\rho(T)} = \frac{(M_w(T)/\rho_w) + (M_i(T)/\rho_i) + (M_s/\rho_s) + (M_a/\rho_a)}{M_p} \qquad (3.3)$$

where the subscripts w, i, s and a refer to the water, ice, solids and air components, and M_p is the total mass of the food product.

Given the solids, unfrozen water and air mass fractions, and the product density above freezing, the solids density may be found from Eq. (3.3) as

$$\frac{1}{\rho_s} = \frac{(M_p/\rho_p) + (M_{wo}/\rho_w) + (M_a/\rho_a)}{M_s} \qquad T > T_{if} \qquad (3.4)$$

The relationship shown in Eq. (3.3) does not distinguish between the various components of the solid fraction. The influence of the carbohydrate, lipid, protein and mineral can be included as

$$\frac{1}{\rho(T)} = \frac{(M_w(T)/\rho_w) + (M_i(T)/\rho_i) + (M_c/\rho_c) + (M_{lp}/\rho_{lp})}{M_p}$$

$$+ \frac{(M_{pr}/\rho_{pr}) + (M_m/\rho_m) + (M_a/\rho_a)}{M_p} \qquad (3.5)$$

Eq. (3.5) is difficult to utilize practically since the densities and mass fractions of the solid components must be known. Furthermore, Hsieh et. al. (1977) showed little variation in density as a function of temperature, consequently, any variability due to the individual solids components in Eq. (3.5) would generally be insignificant.

3.1.3 Thermal Conductivity in Frozen Foods

Due to the large difference between the thermal conductivity of ice and water, the thermal conductivity of the unfrozen food product increases suddenly during freezing. Consequently, thermal conductivity is difficult to predict (Heldman, 1982).

Kopelman (1966) developed relationships for thermal conductivity in two-component-homogeneous dispersed, fibrous and layered systems. These models assume that two phases are present; a continuous phase and a discontinuous phase. More than two phases are present in frozen food products (water, ice, solids and, in some cases, air); therefore, modifications of this model are required to include the additional phases.

Heldman and Gorby (1975a) modified the Kopelman model to simulate a three phase (water, ice and solids) frozen food product. In this model, two steps are required for the estimation of thermal conductivity. Additional modification has resulted in a three step model to include

air, if present. In the first step, the water fraction is considered to be the continuous phase and the ice fraction to be the discontinuous phase. For the second step, the combination of water and ice is assumed to the continuous phase, and the solids fraction is assumed to be discontinuous. In the final step, the water-ice-solids combination is considered continuous, and the air fraction is discontinuous. The three step process is shown mathematically below.

Step 1. Continuous phase: water

Discontinuous phase: ice

$$V_i = \frac{M_i(T)/\rho_i(T)}{M_w(T)/\rho_w(T) + M_i(T)/\rho_i(T)}$$

$$Q_1 = V_i^{2/3}(1 - k_i/k_w)$$

$$k_{wi} = k_w \left[\frac{1 - Q_1}{1 - Q_1(1 - V_i^{1/3})} \right] \tag{3.6a}$$

Step 2. Continuous phase: water-ice

Discontinuous phase: solids

$$V_s = \frac{M_s/\rho_s}{M_w(T)/\rho_w(T) + M_i(T)/\rho_i(T) + M_s/\rho_s}$$

$$Q_2 = V_s^{2/3}(1 - k_s/k_{wi})$$

$$k_{wis} = k_{wi} \left[\frac{1 - Q_2}{1 - Q_2(1 - V_s^{1/3})} \right] \tag{3.6b}$$

Step 3. Continuous phase: air/water/ice

Discontinuous phase: solids

$$V_a = \frac{M_a/\rho_a}{M_w(T)/\rho_w(T) + M_i(T)/\rho_i(T) + M_s/\rho_s + M_a/\rho_a}$$

$$Q_s = V_a^{2/3}(1 - k_a/k_{wis})$$

$$k(T) = k_{wisa} = k_{wis}\left[\frac{1 - Q_s}{1 - Q_s(1 - V_a^{1/3})}\right] \qquad (3.6c)$$

The value for the thermal conductivity of the solids may be found from the Kopelman model and the experimentally determined thermal conductivity of the unfrozen food product.

In summary, estimation of thermal conductivity in frozen foods is a multi-step procedure which first requires the determination of the thermal conductivity of the solids, and the prediction of the unfrozen water fraction and product density as functions of temperature.

3.1.4 Apparent Specific Heat

A food product releases both sensible and latent heat as it freezes. Many researchers (Heldman and Gorby, 1975b, Lescano, 1973, and Bonacina and Comini, 1973a), have incorporated the sensible and latent heat effects into an apparent specific heat. This apparent specific heat can be estimated from the temperature differential of the enthalpy (H)

$$Cp(T) = \frac{dH(T)}{dT} \qquad (3.7)$$

The enthalpy can be expressed in terms of the sensible heat removed from the solid, unfrozen water, ice and, in some cases, air fractions, and from the latent heat (Γ) as follows (Heldman and Singh, 1981)

$$H(T) = \int^{T} \left[(M_w(T) \cdot Cp_w) + (M_i(T) \cdot Cp_i) + (M_s \cdot Cp_s) + (M_a \cdot Cp_a) + \Gamma \cdot \frac{dM_w(T)}{dT} \right] dT \qquad (3.8)$$

Note that the mass fractions of the solids and air fractions are not assumed to be functions of temperature. Substituting Eq. (3.8) into Eq. (3.7), results in the following expression for the apparent specific heat

$$Cp(T) = (M_w(T) \cdot Cp_w) + (M_i(T) \cdot Cp_i) + (M_s \cdot Cp_s) + (M_a \cdot Cp_a) + \Gamma \cdot \frac{dM_w(T)}{dT} \qquad (3.9)$$

The specific heat of solids can be found from an experimental value for the specific heat of the product above freezing

$$Cp_s = \frac{(M_p/Cp_p) + (M_{wo}/Cp_w) + (M_a/Cp_a)}{M_s} \qquad T > T_{if} \qquad (3.10)$$

3.2 Practical Evaluation of Thermal Properties

The temperature dependence of thermal properties must be considered during the simulation of frozen food storage. Thermal properties may be determined explicitly using Eqs. (3.3), (3.6a-c), and (3.9) for density, thermal conductivity, and specific heat, respectively. However, use of

explicit functions in a numerical solution is computationally inefficient. Alternately, a scheme was developed to determine constant thermal property values over specified temperature intervals which were selected to minimize errors in the solution.

The temperature intervals, over which thermal property values were considered constant, were selected by limiting (1) the change in the magnitude (that is, the first derivative) and (2) the change in the slope (that is, the second derivative) of the thermal property function over a given temperature interval. The derivative values were estimated numerically from Eqs. (3.3), (3.6a-c), and (3.9). The constant property values over each temperature interval were determined numerically using a five point Gauss quadrature integration method (Hornbeck, 1975). Constant thermal conductivity values and associated temperature ranges are compared in Figure 3.1 with thermal conductivity values found using Eq. (3.6a-c). (Note, in this case, k_p = 0.94 W/m°C, M_{wo} = 77%, T_{if} = -0.7°C.) These values were determined by limiting the increase in both magnitude and slope of the density curve over each constant property interval to 5% and 25%, respectively. Similar constant property temperature intervals were determined for thermal conductivity and specific heat.

3.3 Transient Heat Conduction during Frozen Food Storage

To predict quality loss in frozen food storage, the temperature distribution within the product as a function of time must be estimated accurately. This involves the solution of the transient heat conduction problem. The complexity of the solution arises from the temperature dependence of the thermal properties.

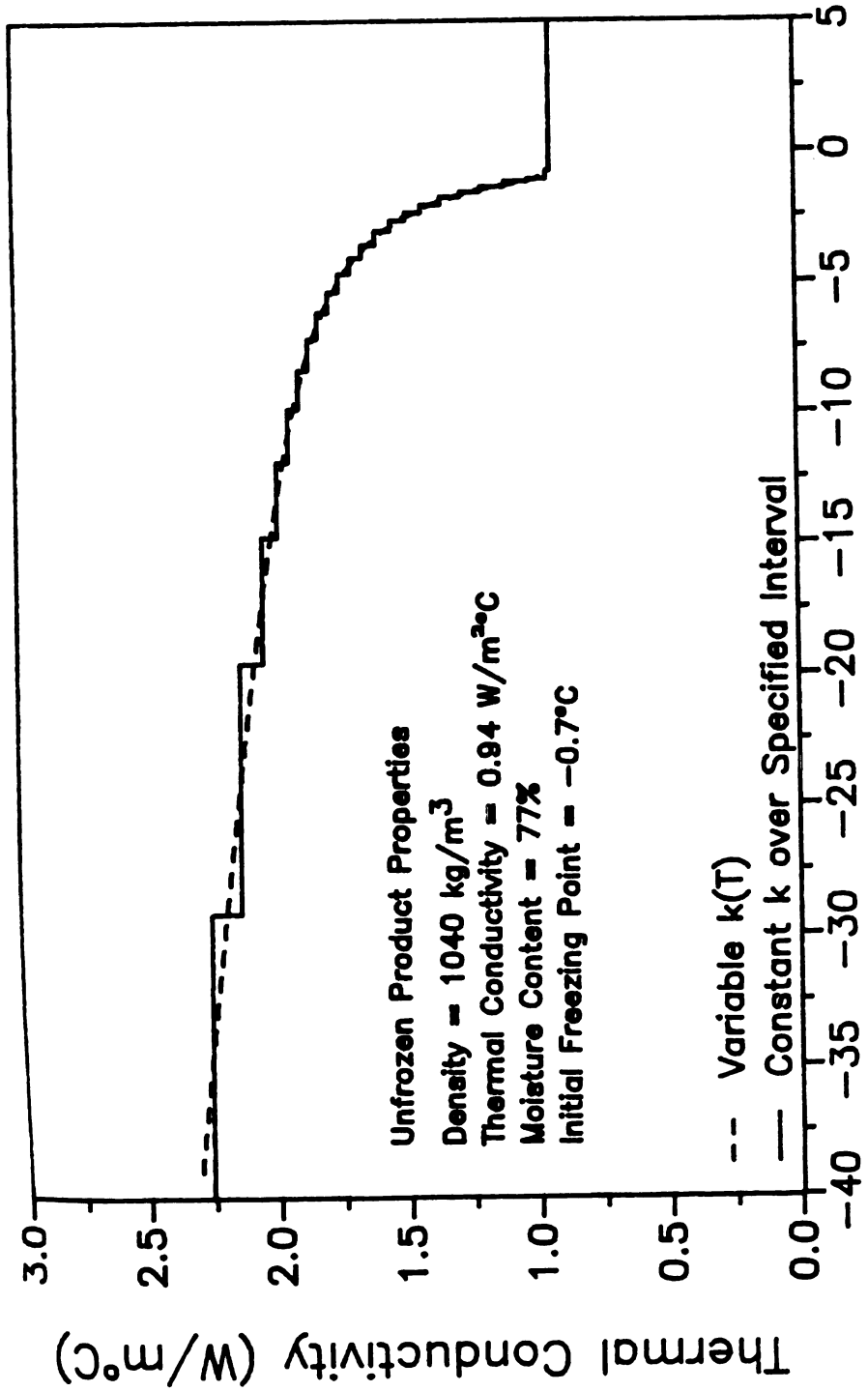


Figure 3.1 Comparison of Constant Thermal Conductivity Values with Thermal Conductivity Values as a Function of Temperature.

The assumptions used in this study were:

1. Heat transfer between the surroundings and the food product occurs by convection only, and heat transfer within the product proceeds by conduction.
2. The frozen food product is isentropic and homogeneous.
3. The surface heat transfer coefficient associated with convective heat transfer is a constant or a function of time, but not a function of position.
4. Moisture loss from the product is negligible, and total product mass remains constant.
5. Internal packaging boundaries within a large mass (such as a pallet load) of product have negligible affect on the heat transfer rate.

3.3.1 One Dimensional Heat Transfer Analysis

In this analysis, it is assumed that heat transfer occurs in one dimension only. The Fourier one dimensional transient heat conduction equation describes one dimensional conductive heat transfer through an isentropic medium (Carslaw and Jaeger, 1959). The governing partial differential equation for regular geometries is given by

$$\frac{1}{x^j}\cdot\frac{\partial}{\partial x}\left[x^j\cdot k(T)\frac{\partial T}{\partial x}\right] = \rho(T)\cdot Cp(T)\frac{\partial T}{\partial t} \qquad (3.11)$$

j = 0: infinite slab

j = 1: infinite cylinder

j = 2: sphere

An infinite slab is finite in the direction of heat transfer, and it is infinite in the other two dimensions. An infinite cylinder is finite is

the radial direction (direction of heat transfer) and has infinite length. Heat transfer is assumed to occur only in the radial direction in the sphere. The various geometries and indicated directions of heat transfer are shown in Figure 3.2. Since the product properties are functions of temperature, the problem is nonlinear.

An initial condition is required for the solution; the product is assumed to be at a uniform temperature, or at a temperature distribution that is a known function of position.

$$T = T_0 \qquad Lx_0 \le x \le Lx$$

or

$$T = T_0(x) \qquad t = 0 \qquad (3.12)$$

Note, $Lx_0 = 0$ for a slab, solid infinite cylinder and sphere, and $Lx_0 \neq 0$ for a hollow infinite cylinder.

Two boundary conditions at $x = Lx_0$ and at $x = Lx$ are required because of the second order differential with position in the governing partial differential equation (Eq. (3.11)). Convective heat transfer is assumed to be occurring at the surface of the product. For the case of symmetrical boundary conditions on both sides of an infinite slab, or a solid infinite cylinder, or a sphere, the boundary conditions are

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0 \qquad \begin{matrix} x = 0 \\ t > 0 \end{matrix} \qquad (3.13a)$$

$$k(T)\left.\frac{\partial T}{\partial x}\right|_{x=Lx} = hx_{Lx}(t)\cdot[T - T_{\infty,Lx}(t)] \qquad \begin{matrix} x = Lx \\ t > 0 \end{matrix} \qquad (3.13b)$$

If the boundary conditions are asymmetrical, or if the geometric shape is a hollow cylinder, the convective boundary condition at $x = 0$ is

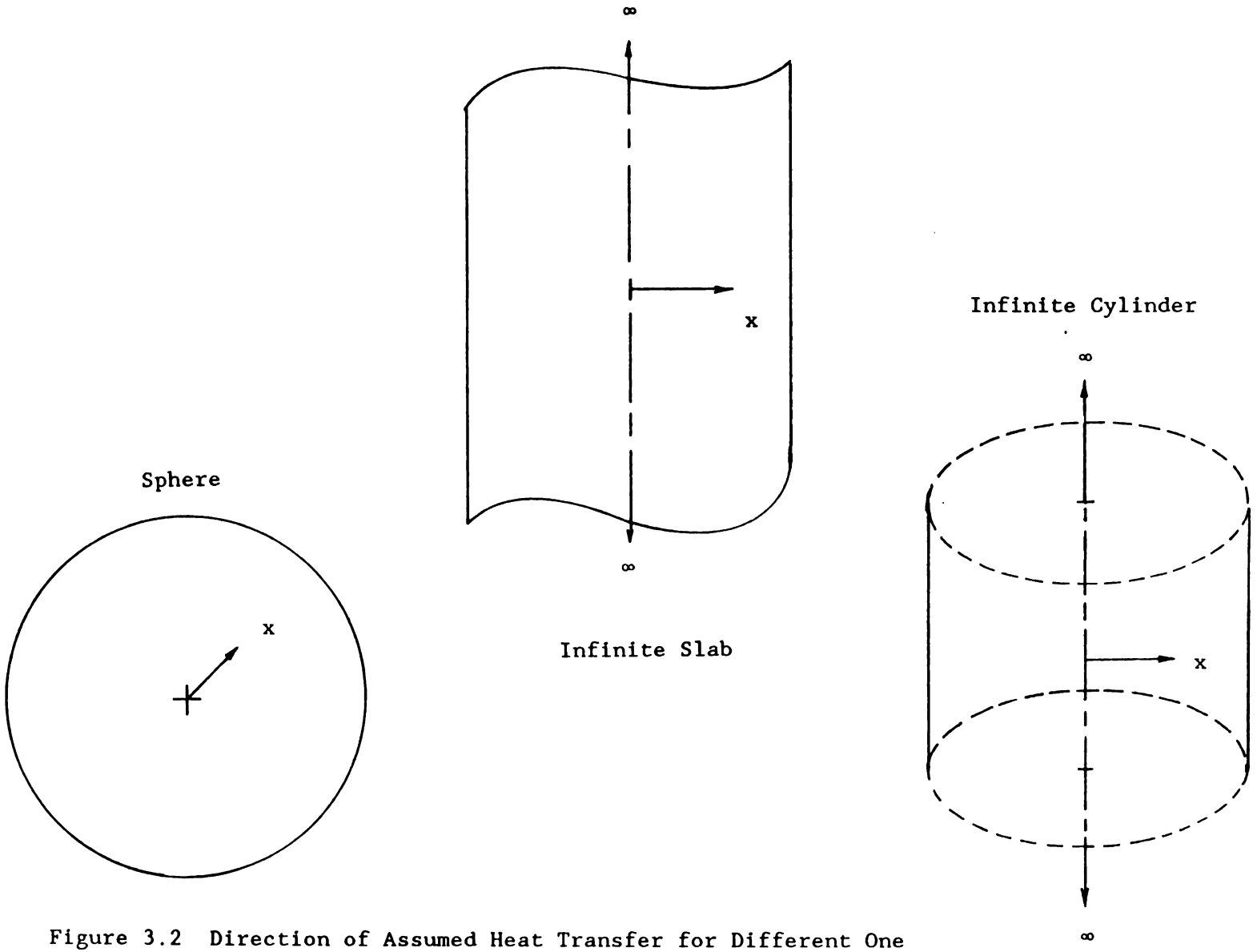


Figure 3.2 Direction of Assumed Heat Transfer for Different One Dimensional Geometries.

$$k(T)\frac{\partial T}{\partial x}\bigg|_{x=Lx_0} = hx_{Lx_0}(t)\cdot[T - T_{\infty,Lx_0}(t)] \qquad \begin{matrix} x = Lx_0 \\ t > 0 \end{matrix} \qquad (3.13c)$$

3.3.2 Two Dimensional Analysis

Heat transfer in two dimensions is considered for rectangular and cylindrical geometries. The governing partial differential equation is

$$\frac{1}{x^j}\cdot\frac{\partial}{\partial x}\left[x^j\cdot k(T)\frac{\partial T}{\partial x}\right] + \frac{\partial}{\partial y}\left[k(T)\frac{\partial T}{\partial y}\right] = \rho(T)\cdot Cp(T)\frac{\partial T}{\partial t} \qquad (3.14)$$

j = 0: infinite rectangle

j = 1: finite cylinder

An infinite rectangle is finite in two dimensions (directions of heat transfer) and infinite in the other dimension; in this case, a finite cylinder has a finite radius and length (directions of heat transfer), with no angular heat flux. The geometries and assumed directions of heat transfer are shown in Figure 3.3.

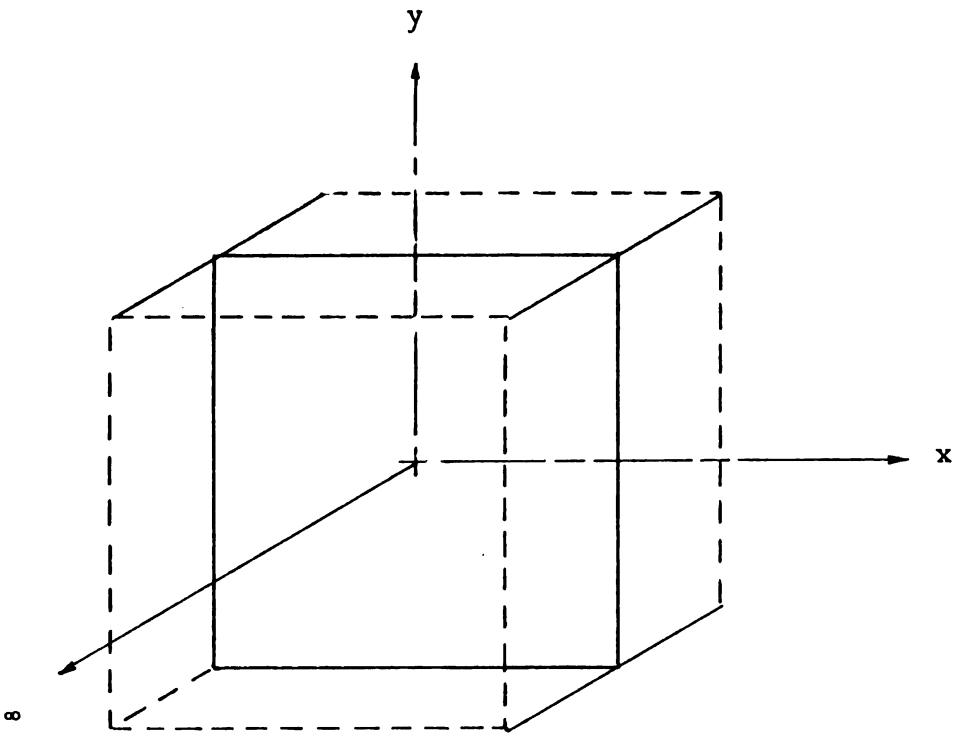
The initial condition is assumed to be constant, or a known function of position

$$T = T_0 \qquad Lx_0 \le x \le Lx,\ 0 \le y \le Ly$$

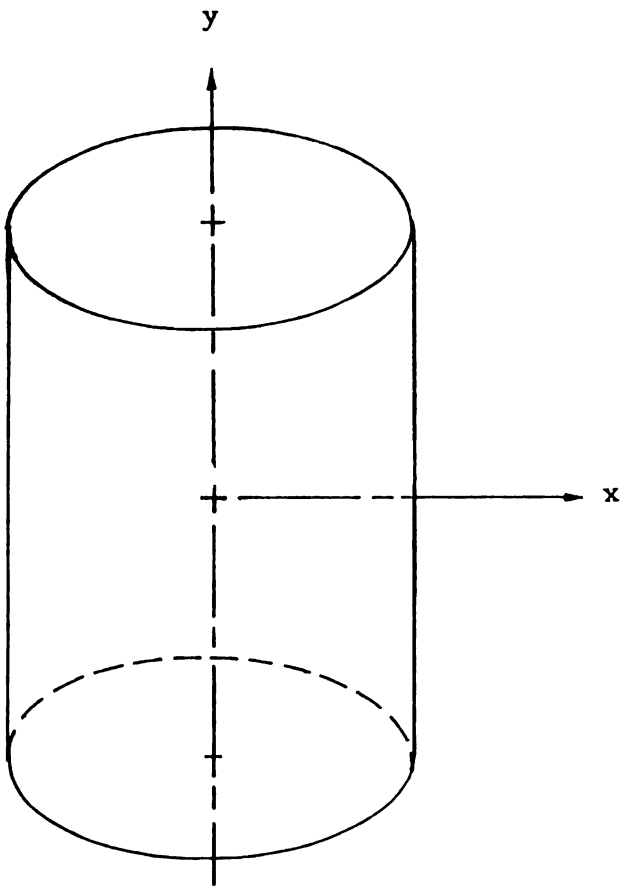
or

$$T = T_0(x) \qquad t = 0 \qquad (3.15)$$

where $Lx_0 = 0$ for an infinite rectangle or solid cylinder and $Lx_0 \neq 0$ for a hollow cylinder.



Infinite Rectangle



Finite Cylinder

Figure 3.3 Direction of Assumed Heat Transfer for Different Two Dimensional Geometries.

Four boundary conditions are needed in this case because of the second order differential in both dimensions. The boundary conditions for symmetrical heat transfer along the x-axis of an infinite rectangle, or along the radial direction of a solid cylinder are given by Eqs. (3.13a,b) for $0 \leq y \leq Ly$. Equations (3.13a,c) describe the boundary conditions for unsymmetrical heat transfer along the x-axis, or along the radial axis of a hollow cylinder for $0 \leq y \leq Ly$. The boundary conditions along the y-axis are

$$k(T)\left.\frac{\partial T}{\partial y}\right|_{y=0} = hy_0(t)\cdot[T - T_{\infty,0}(t)] \qquad \begin{array}{l} y = 0 \\ Lx_0 \leq x \leq Lx \\ t > 0 \end{array} \qquad (3.16a)$$

$$k(T)\left.\frac{\partial T}{\partial y}\right|_{y=Ly} = hy_{Ly}(t)\cdot[T - T_{\infty,Ly}(t)] \qquad \begin{array}{l} y = Ly \\ Lx_0 \leq x \leq Lx \\ t > 0 \end{array} \qquad (3.16b)$$

3.4 Numerical Time-Temperature Simulation Models

The one dimensional simulation model for frozen foods during storage was based on the predictive models developed by Lescano (1973) and Heldman and Gorby (1975b) for estimating freezing times. The Crank-Nicolson implicit finite difference scheme was used to numerically solve the governing partial differential equation given in Eq. (3.11). The two dimensional prediction model was based on the Douglas and Gunn (1964) Alternating Direction Implicit (ADI) scheme. The scheme was modified to allow temperature dependent thermal properties. In both models, thermal properties were assumed to be constant over specified temperature ranges, using the procedure described in Section 3.2, while

allowing for variation over the total temperature range in consideration. In addition, both models permit a number of different storage periods with different storage temperatures and convective heat transfer coefficients during each period.

Input parameters required for the solution of both the one and two dimensional problems are:

1. Initial freezing temperature.
2. Unfrozen water and air fractions.
3. Thermal properties of the unfrozen food product.
4. Temperature of the frozen product prior to storage.
5. Product thickness or radius, or length (for the two dimensional model).

and for each storage period:

6. Length of storage period.
7. Ambient storage temperature.
8. Heat transfer coefficients (on all sides of product).

In addition, for the two dimensional model:

9. Product length.

3.4.1 One Dimensional Heat Transfer Finite Difference Scheme

The one dimensional heat transfer given by Eq. (3.11) was solved numerically for an infinite slab, infinite cylinder and sphere using the Crank-Nicolson finite difference scheme, and temperature dependent thermal properties. The resulting implicit finite difference equation for an interior node is

$$\eta\frac{k_- A_-}{\Delta x} T_{\ell-1}^{n+1} - \left[\eta\frac{(k_- A_- + k_+ A_+)}{\Delta x} + ((\rho Cp)_- + (\rho Cp)_+)\frac{\Delta x \cdot A}{2 \cdot \Delta t} \right] T_{\ell}^{n+1} + \eta\frac{k_+ A_+}{\Delta x} T_{\ell+1}^{n+1}$$

$$= -\beta\frac{k_- A_-}{\Delta x} T_{\ell-1}^{n} + \left[\beta\frac{(k_- A_- + k_+ A_+)}{\Delta x} - ((\rho Cp)_- + (\rho Cp)_+)\frac{\Delta x \cdot A}{2 \cdot \Delta t} \right] T_{\ell}^{n} - \beta\frac{k_+ A_+}{\Delta x} T_{\ell+1}^{n}$$

(3.17)

where $\beta = \eta = 0.5$ for the Crank-Nicolson method. The thermal properties (ρ, k, and Cp) are evaluated at the n^{th} time step, and at the locations indicated in Figure 3.4. Equation (3.17) may be used for various geometries by using the appropriate cross sectional areas as shown below.

$$\left.\begin{aligned} A_- &= A = A_+ = 1.0 \end{aligned}\right\} \quad \text{for infinite slabs}$$

$$\left.\begin{aligned} A_- &= 2\pi \cdot [Lx_0 + \Delta x \cdot (\ell - 3/2)] \\ A &= 2\pi \cdot [Lx_0 + \Delta x \cdot (\ell - 1)] \\ A_+ &= 2\pi \cdot [Lx_0 + \Delta x \cdot (\ell - 1/2)] \end{aligned}\right\} \quad \text{for infinite cylinders}$$

$$\left.\begin{aligned} A_- &= 4\pi \cdot [Lx_0 + \Delta x \cdot (\ell - 3/2)]^2 \\ A &= 4\pi \cdot [Lx_0 + \Delta x \cdot (\ell - 1)]^2 \\ A_+ &= 4\pi \cdot [Lx_0 + \Delta x \cdot (\ell - 1/2)]^2 \end{aligned}\right\} \quad \text{for spheres}$$

where $\ell = 1$ at the location Lx_0. The finite difference representations of the boundary conditions given in Eqs. (3.13a,b,c) are

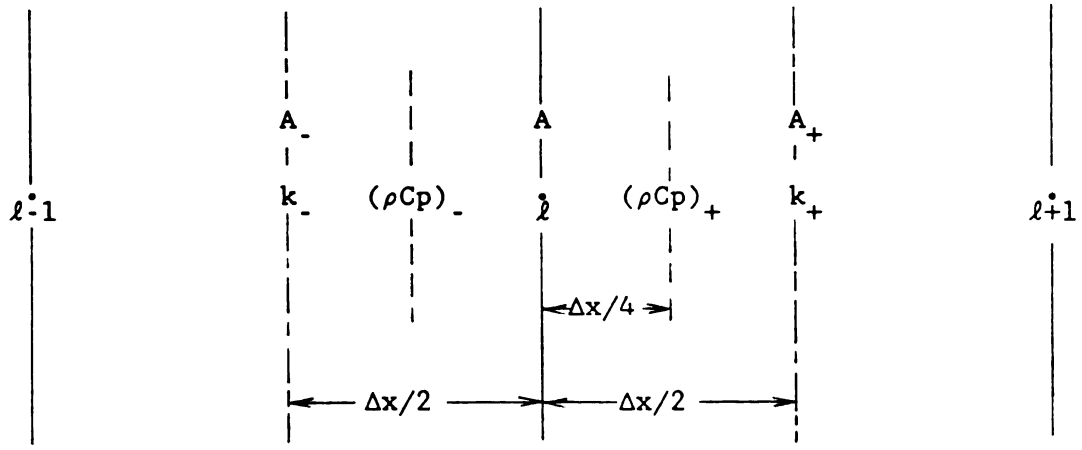


Figure 3.4 Evaluation of Thermal Properties in One Dimensional Numerical Solution.

At $x = 0$, $(\ell = 1)$

$$- \left[\eta \frac{k_+ A_{1+}}{\Delta x} + ((\rho Cp)_+) \cdot \frac{\Delta x \cdot A_1}{2 \cdot \Delta t} \right] \cdot T_1^{n+1} + \eta \frac{k_+ A_{1+}}{\Delta x} T_2^{n+1}$$

$$- \left[\beta \frac{k_+ A_{1+}}{\Delta x} - ((\rho Cp)_+) \cdot \frac{\Delta x \cdot A_1}{2 \cdot \Delta t} \right] \cdot T_1^{n} + \beta \frac{k_+ A_{1+}}{\Delta x} T_2^{n}$$

$$- (\beta \cdot T_{\infty, Lx_0}^{n} \cdot hx_{Lx_0}^{n} + \eta \cdot T_{\infty, Lx_0}^{n+} \cdot hx_{Lx_0}^{n+1}) A_1 \qquad (3.18a)$$

At $x = Lx$, $(\ell = L)$

$$\eta \frac{k_- A_{L-}}{\Delta x} T_{L-1}^{n+1} - \left[\eta \frac{k_- A_{L-}}{\Delta x} + ((\rho Cp)_-) \cdot \frac{\Delta x \cdot A_L}{2 \cdot \Delta t} \right] \cdot T_L^{n+1} = - \beta \frac{k_- A_{L-}}{\Delta x} T_{L-1}^{n}$$

$$+ \left[\beta \frac{k_- A_{L-}}{\Delta x} - ((\rho Cp)_-) \cdot \frac{\Delta x \cdot A_L}{2 \cdot \Delta t} \right] \cdot T_L^{n} - (\beta \cdot T_{\infty, Lx}^{n} \cdot hx_{Lx}^{n} + \eta \cdot T_{\infty, Lx}^{n+} \cdot hx_{Lx}^{n+1}) A_L \qquad (3.18b)$$

Note: in Eq. (3.18a), hx_1 is equal to zero for the insulated slab, solid cylinder, and solid sphere, and in Eq. (3.18b), L is the total number of nodes.

The set of finite difference equations may be expressed in matrix form as

$$\mathbf{A} \cdot \mathbf{T}^{n+1} = \mathbf{B} \cdot \mathbf{T}^{n} + \mathbf{D} \qquad (3.19)$$

The coefficient matrix, **A**, is tridiagonal. The set of equations is solved by using the Thomas Algorithm (Thomas, 1949), which makes use of the large numbers of zeros in the coefficient matrix to solve the equations efficiently.

A flow chart for the one dimensional numerical temperature solution, including the prediction of the quality profile, and the code for the computer program, written in Fortran 77 on a Vax 11/750 computer, are presented in Appendix B.

3.4.2 Two Dimensional Heat Transfer Analysis

The two dimensional Alternating Direction Implicit (ADI) method proposed by Douglas and Gunn (1964) was modified to include temperature dependent thermal properties. In this method, the Crank-Nicolson scheme is utilized, and a two step procedure was employed. The modified difference equation for an interior node is shown below.

Step 1. Use Crank-Nicolson approximation in the x-direction

$$\eta_x \frac{k_{-x} A_{-x}}{\Delta x} T^{n+\xi}_{\ell-1,m} - \left[\eta_x \frac{k_{-x} A_{-x} + k_{+x} A_{+x}}{\Delta x} + ((\rho Cp)_{-x} + (\rho Cp)_{+x} + (\rho Cp)_{-y} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{4\Delta t} \right] \cdot T^{n+\xi}_{\ell,m} + \eta_x \frac{k_{+x} A_{+x}}{\Delta x} T^{n+\xi}_{\ell+1,m}$$

$$= -\beta_x \frac{k_{-x} A_{-x}}{\Delta x} T^{n}_{\ell-1,m} - \frac{k_{-y} A_{-y}}{\Delta y} T^{n}_{\ell,m-1} + \left[\beta_x \frac{k_{-x} A_{-x} + k_{+x} A_{+x}}{\Delta x} + \frac{k_{-y} A_{-y} + k_{+y} A_{+y}}{\Delta y} - ((\rho Cp)_{-x} + (\rho Cp)_{+x} + (\rho Cp)_{-y} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{4\Delta t} \right] \cdot T^{n}_{\ell,m} - \frac{k_{+y} A_{+y}}{\Delta y} T^{n}_{\ell,m+1} - \beta_x \frac{k_{+x} A_{+x}}{\Delta x} T^{n}_{\ell+1,m} \qquad (3.20a)$$

Step 2. Use results from above for x-direction in 2D Crank-Nicolson finite difference equation.

$$\eta_y \frac{k_{-y}A_{-y}}{\Delta y} T^{n+1}_{\ell,m-1} - \left[\eta_y \frac{k_{-y}A_{-y} + k_{+y}A_{+y}}{\Delta y} + ((\rho Cp)_{-x} + (\rho Cp)_{+x} + (\rho Cp)_{-y} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{4\Delta t} \right] \cdot T^{n+1}_{\ell,m} + \eta_y \frac{k_{+y}A_{+y}}{\Delta y} T^{n+1}_{\ell,m+1}$$

$$= -\eta_x \frac{k_{-x}A_{-x}}{\Delta x} T^{n+\xi}_{\ell-1,m} - \eta_x \frac{k_{-x}A_{-x} + k_{+x}A_{+x}}{\Delta x} T^{n+\xi}_{\ell,m} - \eta_x \frac{k_{+x}A_{+x}}{\Delta x} T^{n+\xi}_{\ell+1,m}$$

$$- \beta_x \frac{k_{-x}A_{-x}}{\Delta x} T^{n}_{\ell-1,m} - \beta_y \frac{k_{-y}A_{-y}}{\Delta y} T^{n}_{\ell,m-1} + \left[\beta_x \frac{k_{-x}A_{-x} + k_{+x}A_{+x}}{\Delta x} + \beta_y \frac{k_{-y}A_{-y} + k_{+y}A_{+y}}{\Delta y} - ((\rho Cp)_{-x} + (\rho Cp)_{+x} + (\rho Cp)_{-y} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{4\Delta t} \right] \cdot T^{n}_{\ell,m} - \beta_y \frac{k_{+y}A_{+y}}{\Delta y} T^{n}_{\ell,m+1} - \beta_x \frac{k_{+x}A_{+x}}{\Delta x} T^{n}_{\ell+1,m} \qquad (3.20b)$$

where $\beta_x = \beta_y = \eta_x = \eta_y = 0.5$ for the Crank-Nicolson approximation. The thermal properties were evaluated at the n^{th} time step, unless specified otherwise, and all cross sectional areas, A, are defined at the locations about the node (ℓ,m), as shown in Figure 3.5. The cross sectional areas for the rectangular rod and solid cylinder are

$$\left.\begin{aligned} A_{-x} = A_x = A_{+x} = \Delta y \\ A_{-y} = A_y = A_{+y} = \Delta x \end{aligned}\right\} \quad \text{for rectangular rod}$$

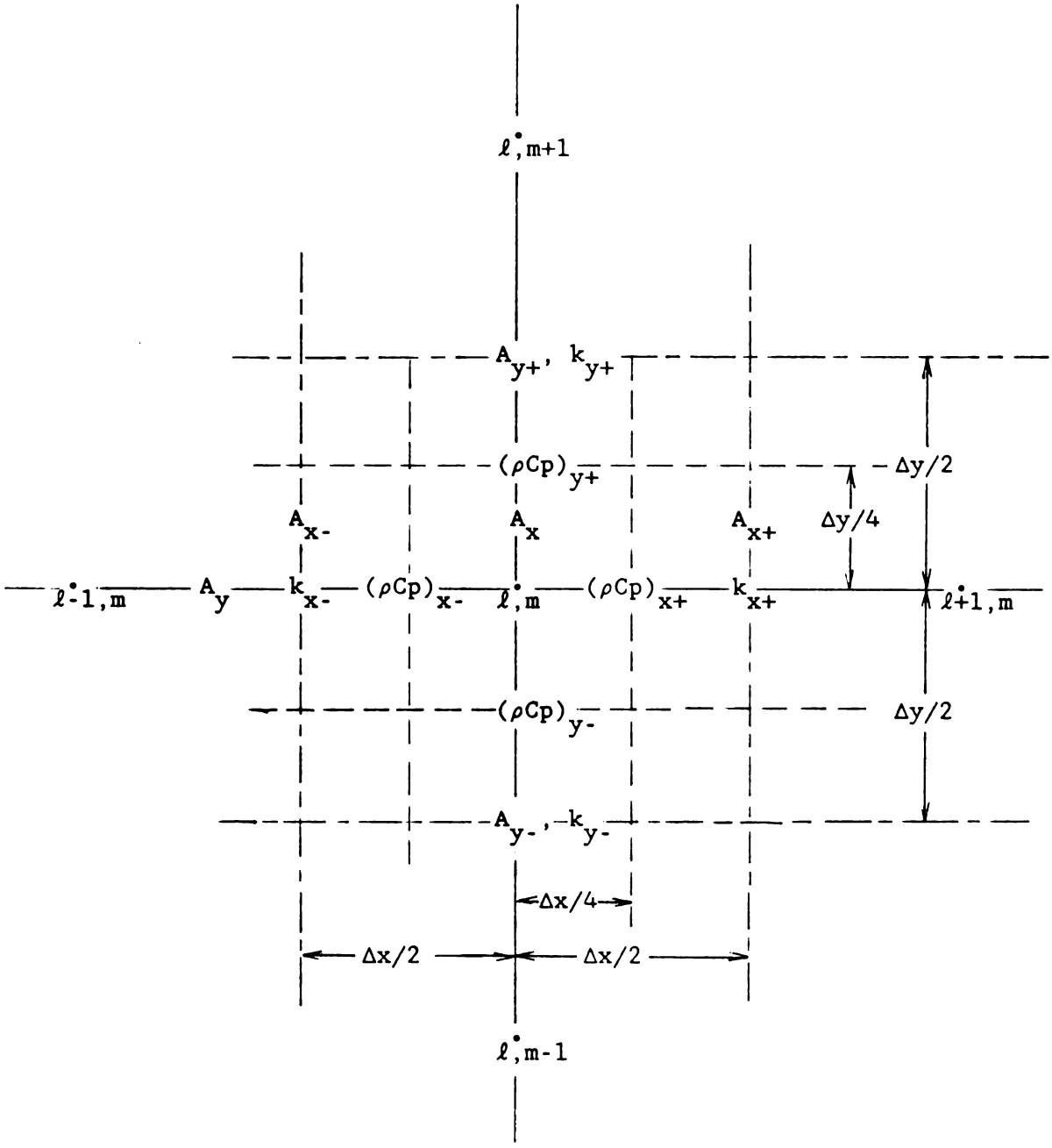


Figure 3.5 Evaluation of Thermal Properties in Two Dimensional Numerical Solution.

$$\left.\begin{aligned} A_{-x} &= 2\pi\cdot[Lx_0 + \Delta x\cdot(\ell-3/2)]\Delta y \\ A_{x} &= 2\pi\cdot[Lx_0 + \Delta x\cdot(\ell-1)]\Delta y \\ A_{+x} &= 2\pi\cdot[Lx_0 + \Delta x\cdot(\ell-1/2)]\Delta y \\ A_{-y} = A_{y} &= A_{+y} = \pi\cdot\Delta x\cdot[Lx_0 + \Delta x\cdot(\ell-1)] \end{aligned}\right\} \quad \text{for solid cylinder}$$

where $\ell = 1$ at the location Lx_0.

The Douglas-Gunn ADI method differs from the methods proposed by Peaceman and Rachford (1955) and Douglas (1955), because, in the first sweep, a full time step is used to provide an estimate of the temperature values at n+1; these values are used in the second equation and are denoted as $n+\xi$. The traditional methods evaluate the first sweep at n+1/2, the one-half time step.

Eight different expressions for the boundary nodes (four surfaces and four edges) were derived in a similar manner as for the one dimensional case. These difference equations are found in Appendix A.

A flow chart, describing the numerical solution of the two dimensional heat conduction problem, and the corresponding computer code (Fortran 77) are given in Appendix C. The two subroutines, 'PROPER', and 'CONSP', are identical to the same subroutines in the one dimensional model, and are omitted in the code listing.

3.5 Estimation of the Surface Heat Transfer Coefficient

Analytical, numerical and experimental methods have been proposed to determine the surface heat transfer coefficient. Traditional analytical methods of determining the surface heat transfer coefficient are presented first. This is followed by a investigation of a numerical method which uses experimentally determined internal temperatures to estimate the surface heat transfer coefficient. In this case, the

problem of estimating the surface heat transfer coefficient is treated as part of a group of problems called inverse heat conduction problems.

3.5.1 Analytical Methods

The surface heat transfer coefficient is a function of the air stream velocity, the temperature difference between the surface being heated or cooled and the surroundings, and the packaging layer between the free air stream and the product. The Reynolds number (Re_{Lx}) is used to characterize the free stream velocity, and the Grashof number (Gr_{Lx}) is used to characterize the temperature gradient as

$$Re_{Lx} = \frac{U_\infty \cdot Lx_c}{\nu} \tag{3.21a}$$

$$Gr_{Lx} = \frac{g \cdot \beta_e \cdot (T_s - T_\infty) \cdot (Lx_c)^3}{\nu^2} \tag{3.21b}$$

where U_∞ is the air free stream velocity, ν is the kinematic viscosity of the air, g is the acceleration due to gravity, β_e is the expansion coefficient of air, T_s is the temperature of the surface, and Lx_c is the characteristic length. The characteristic length is defined as the ratio of the cross sectional area and the perimeter of the surface (Goldstein, et. al., 1973).

Forced convection, resulting from forced air flow, and free convection, resulting from temperature induced density gradients, are both important when the square of the Reynolds number and the Grashof number are approximately equal. Forced convection effects dominate when $(Re_{Lx})^2$ is much greater than Gr_{Lx}, and free convection effects dominate when the opposite is true. These relationships are summarized below.

$$\left[\frac{Gr_{Lx}}{Re_{Lx}^2}\right] \gg 1 \qquad \text{Free Convection only} \qquad (3.22a)$$

$$\left[\frac{Gr_{Lx}}{Re_{Lx}^2}\right] \approx 1 \qquad \text{Both Free and Forced Convection} \qquad (3.22b)$$

$$\left[\frac{Gr_{Lx}}{Re_{Lx}^2}\right] \ll 1 \qquad \text{Forced Convection only} \qquad (3.22c)$$

3.5.1.1 Forced Convection over a Flat Plate

The steady state one dimensional for the determination of the mean Nussult number ($\overline{Nu}_F$) resulting from a constant free stream velocity flow along a constant temperature semi-infinite plate, and along a semi-infinite plate with an arbitrarily specified surface heat flux are (Kays and Crawford, 1980)

$$\overline{Nu}_F = 0.664 \cdot Pr^{1/3} \cdot Re_{Lx}^{1/2} \qquad \text{Constant Temperature Laminar Flow, } Pr \geq 0.6 \qquad (3.23a)$$

$$\overline{Nu}_F = 0.906 \cdot Pr^{1/3} \cdot Re_{Lx}^{1/2} \qquad \text{Constant Heat Flux Laminar Flow} \qquad (3.23b)$$

where

$$\overline{Nu}_F = \frac{\overline{hx}_F \cdot Lx}{k_a}$$

$\overline{hx}_F = 2 \cdot hx_F(x)$

k_a = thermal conductivity of fluid (air)

3.5.1.2 Free Convection

Free or natural convection is a result of temperature induced density gradients in the fluid. These temperature gradients cause free convection currents as the denser fluid (cooler fluid) falls, and the less dense fluid (warmer fluid) rises. The Nussult numbers resulting from free convection (Nu_N) for the upper surface of a horizontal heated or cooled plate, are given below (McAdams, 1954).

1. Upper surface of a heated plate ($T_s > T_\infty$).

$$\overline{Nu}_N = 0.54 \cdot Ra_{Lx}^{1/4} \qquad 10^4 \le Ra_L \le 10^7 \qquad (3.24a)$$

$$\overline{Nu}_N = 0.15 \cdot Ra_{Lx}^{1/3} \qquad 10^7 \le Ra_L \le 10^{11} \qquad (3.24b)$$

2. Upper surface of a cooled plate ($T_\infty > T_s$).

$$\overline{Nu}_N = 0.27 \cdot Ra_{Lx}^{1/4} \qquad 10^5 \le Ra_L \le 10^{10} \qquad (3.25)$$

where the Rayleigh number, Ra_{Lx} is defined as

$$Ra_{Lx} = Gr_{Lx} \cdot Pr = \frac{g \cdot \beta_e \cdot (T_s - T_\infty) \cdot (Lx_c)^3}{\nu \cdot \kappa}$$

3.5.1.3 Combined Forced and Free Convection

For the case where $(Gr_{Lx}/Re_{Lx}^2) \approx 1$, a correlating equation has been recommended as (Churchill, 1983)

$$\overline{Nu}^n = \overline{Nu}_F^n \pm \overline{Nu}_N^n \qquad (3.26)$$

where n = 3 or 7/2 for parallel or perpendicular flows respectively. The natural convection component may increase or reduce the influence of forced convection depending on whether or not the free convection induces motions in the same or opposite direction as the forced air flow.

The heat transfer coefficient resulting from free and forced convection is

$$hx_{cv} = \frac{\overline{Nu} \cdot k_a}{Lx_c} \tag{3.27}$$

3.5.1.4 Packaging Layer

The thin packaging layer between the product surface and the surrounding air may result in additional resistance to the flow of heat to the product. This resistance may be characterized by an effective packaging resistance, defined as

$$h_{pk} = \frac{k_{pk}}{L_{pk}} \tag{3.28}$$

where k_{pk} and L_{pk} are the overall effective thermal conductivity and length of the packaging layer, including the packaging material, air interfaces, and frost build-up, if any.

3.5.1.5 Overall Surface Heat Transfer Coefficient

The overall surface heat transfer coefficient is a result of the combined thermal resistance from forced and free conduction, and the

packaging layer. The overall surface heat transfer coefficient is defined as

$$\hat{hx} = \left(\frac{1}{hx_c} + \frac{1}{h_{pk}} \right) \tag{3.29}$$

3.5.2 Inverse Heat Conduction Methods.

The estimation of the surface heat transfer coefficient as a function of time is based on the solution of the inverse heat conduction problem (IHCP). In contrast to the direct problem of determining internal temperature distributions from known boundary conditions, the IHCP involves the determination of a boundary condition from internal temperature measurements.

In this study, one dimensional heat transfer in considered through a regular geometrically shaped food product with variable thermal properties. A temperature sensor is located at $x = x_0$, where measurements are taken at a discrete time interval, Δt_m. The boundary at $x = 0$ is insulated, and an unknown heat flux is imposed at the opposite boundary. The temperature distribution is assumed to be known up to the n^{th} time step, t^n, ($t^n = n \cdot \Delta t_m$). The problem is mathematically expressed as

$$\frac{1}{x^j} \cdot \frac{\partial}{\partial x} \left[x^j \cdot k(T) \frac{\partial T}{\partial x} \right] = \rho(T) \cdot Cp(T) \frac{\partial T}{\partial t}$$

$0 \le x \le Lx$

$$t \ge t^n \tag{3.30a}$$

$j = 0$: infinite slab

$j = 1$: infinite cylinder

$j = 2$: sphere

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0 \qquad x = 0 \qquad t \geq t^n \qquad (3.30b)$$

$$k(T)\left.\frac{\partial T}{\partial x}\right|_{x=Lx} = q(t) \quad \text{(unknown)} \qquad x = Lx \qquad t \geq t^n \qquad (3.30c)$$

$$T = T_0 \qquad 0 \leq x \leq Lx \qquad t < t^n \qquad (3.30d)$$

where the unknown heat flux is assumed to be the result of a convective boundary condition at x = Lx

$$q(t) = hx_{Lx}(t) \cdot (T(Lx,t) - T_{\infty,Lx}(t)) \qquad (3.30e)$$

Also, the temperature at x_0, $Y(t^n)$, is known for all time steps, t^n

$$T(x_0,t^n) = Y(t^n) \qquad x = x_0 \qquad t > t^n \qquad (3.31)$$

Note this problem can be divided into two parts, the direct problem from x = 0 to $x = x_0$, and the inverse problem from $x = x_0$ to x = L, as shown in Figure 3.6.

The surface heat transfer coefficient was determined by first estimating the surface heat flux at x = Lx. In this analysis, both the heat flux and the surface heat transfer coefficient, were assumed to be a function of time, but not position. The heat flux was first estimated as a function of time using the regularization method and finite differences in the solution. The temperature at the boundary, T(Lx,t), was calculated using the finite difference solution described for the direct

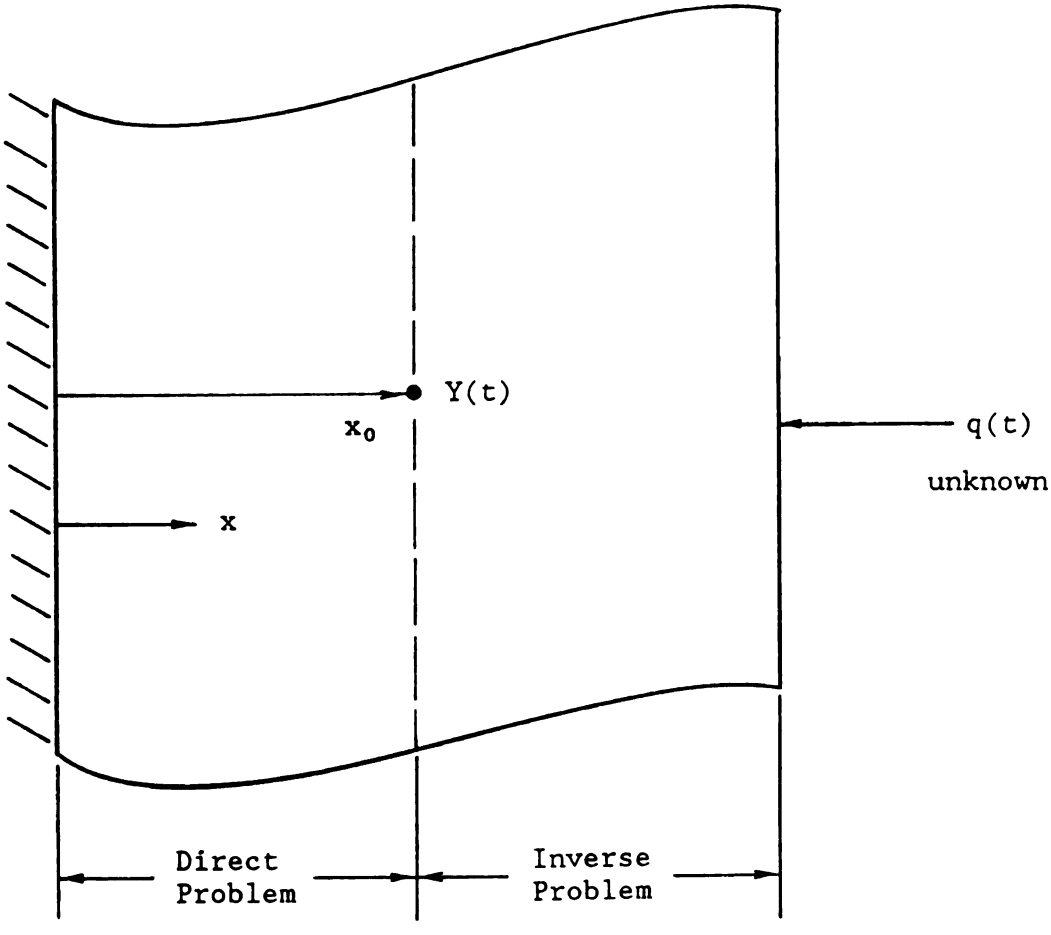


Figure 3.6 Inverse Heat Conduction Problem.

problem in Section 3.4.1, and substituting the calculated surface heat flux values for the convective boundary condition in Eq. (3.18b). Finally, the surface heat transfer coefficient was calculated from Eq. (3.30e) given the predicted surface heat flux and surface temperature, and the known ambient temperature.

Two variations of the regularization method used to estimate the surface heat flux are found: (1) the whole domain technique, and (2) the sequential technique, proposed by Beck and Murio (1986), in which the heat flux components are estimated sequentially. The sequential procedure is more computationally efficient than the whole domain method because, this procedure estimates only a few heat flux components at a time instead of simultaneously estimating all of the components, as done in the whole domain procedure. Due to its computational efficiency, the sequential regularization method was used in this study.

3.5.2.1 The Sequential Regularization Method

The modified least squared function of the regularization method in matrix form is

$$S = (\mathbf{Y}-\mathbf{T})\cdot(\mathbf{Y}-\mathbf{T})+\alpha\cdot\sum_{j=0}^{2} R_j \qquad (3.32)$$

where **Y** and **T** are the measured and estimated temperature vectors, and the terms in the summation (R_0,R_1,R_2) represent the zeroth, first and second order regularization components. The scalar term, α, is the regularization parameter which is adjusted to determine the degree of the influence the regularization terms have on the least squares function.

Tikhonov and Arsenin (1977) suggested the following expression for the various regularization terms

$$R_j = W_j \cdot \int^t \left(\frac{d^j q}{d\tau^j} \right) d\tau \qquad j = 0,\ 1,\ \text{or}\ 2 \qquad (3.33)$$

Each regularization term acts to minimize its corresponding derivative of the estimated heat flux when the least squared function, S, is minimized. Therefore, the zeroth order regularization term tends to bias the heat flux towards zero, the first order pushes the heat flux towards a constant value, and the second order term forces the heat flux towards a constant slope, Beck et. al. (1985).

Forward differences were used to approximate the regularization terms in Eq. (3.33). The expressions for j = 0, 1, 2 are shown below in Eqs. (3.34a,b,c), respectively.

$$R_0 = W_0 \cdot \sum_{i=1}^{r} \left(q^{n+i-1}\right)^2 \qquad (3.34a)$$

$$R_1 = W_1 \cdot \sum_{i=1}^{r} \left(q^{n+i} - q^{n+i-1}\right)^2 \qquad r \geq 1 \qquad (3.34b)$$

$$R_2 = W_2 \cdot \sum_{i=1}^{r} \left(q^{n+i+1} - q^{n+i} + q^{n+i-1}\right)^2 \qquad r \geq 2 \qquad (3.34c)$$

where the W_j values are weighting factors. In the sequential procedure, the summation is carried out over r-future time steps, and q^{n+i+1}, q^{n+i} and q^{n+i-1} are evaluated at the $(n+i+1)^{th}$, $(n+i)^{th}$ and $(n+i-1)^{th}$ time

steps, respectively. The heat flux at q^{n-1} is assumed to be known. Eqs. (3.34a,b,c) can be written in matrix form as:

$$R_j = W_j \cdot (H_j q)^T (H_j q) \qquad j = 0,\ 1,\ \text{or}\ 2 \qquad (3.35)$$

where the estimated heat flux components are contained in the vector q, the forward difference approximations for dq/dt are contained in the H_j matrices, and the weighting factors, W_j, are scalar. Beck et. al. (1985) proposed the following expressions for the H_j matrices

$$H_0 = I \qquad \text{(Identity matrix)} \qquad (3.36a)$$

$$H_1 = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & \cdots & 0 & -1 & 1 \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \qquad (3.36b)$$

$$H_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \qquad (3.36b)$$

The H_j matrices are $r \times r$ with $j + 1$ non-zero diagonals, and with all zeros in the last j row(s).

The regularization orders were analyzed individually by setting the j^{th} weighting factor (W_j) equal to one, and the remaining weighting

factors (W_i, $i \neq j$), equal to zero, and substituting Eq. (3.35) into Eq. (3.32) to obtain

$$S = (Y-T)^2(Y-T) + \alpha \cdot W_j \cdot (H_j q)^T (H_j q) \qquad j = 0, 1, \text{ or } 2 \qquad (3.37)$$

The unknown heat flux **q** is estimated by minimizing the least squares function in Eq. (3.37). The expression is differentiated with respect to **q** and then set equal to zero.

$$\nabla_q S = 2 \cdot \left[\nabla_q (Y-T)^T \right] (Y-T) + 2\alpha \cdot \nabla_q (H_j \hat{q})^T (H_j \hat{q})$$

$$= 0 \qquad j = 0, 1, \text{ or } 2 \qquad (3.38)$$

The temperature matrix **T** may be expanded in a Taylor's series as

$$T = T^* + \nabla_q T (\hat{q} - q^*) \qquad (3.39)$$

where T^* is the resulting temperature vector from an assumed imposed heat flux q^*. A sensitivity coefficient matrix, **X**, is defined as

$$X = \nabla_q T \qquad (3.40)$$

From Eq. (3.40),

$$\nabla_q (Y-T)^T = -X^T \qquad (3.41)$$

and,

$$\nabla_q (H_j \hat{q})^T = H_j^T \qquad j = 0, 1, \text{ or } 2 \qquad (3.42)$$

Beck et. al. (1985). Substituting Eqs. (3.39), (3.40), (3.41) and (3.42) into Eq. (3.38) produces

$$X^T X(\hat{q} - q^*) + \alpha \cdot H_j{}^T H_j \hat{q} - X^T(Y - T^*) = 0 \qquad j = 0, 1, \text{ or } 2 \qquad (3.43)$$

Noting that

$$\alpha \cdot H_j{}^T H_j q^* = 0 \qquad (3.44)$$

allows Eq. (3.43) to be rearranged and solved for $\hat{q}$:

$$\hat{q} = q^* + (X^T X + \alpha \cdot H_j{}^T H_j)^{-1} \cdot X^T(Y - T^*) \qquad j = 0, 1, \text{ or } 2 \qquad (3.45)$$

The temperature vector T^* was found by solving the direct problem using the Crank-Nicolson finite difference method as discussed in Section 3.4.2, and substituting an assumed imposed heat flux q^* at $x = Lx$, in place of the convective boundary condition shown in Eq. (3.18b). The sensitivity coefficient matrix X, is found by differentiating Eqs. (3.30a-d) with respect to q(t), the unknown heat flux at $x = Lx$. The resulting equations are shown below.

$$\frac{1}{x^j} \cdot \frac{\partial}{\partial x} \left[x^j \cdot k(T) \frac{\partial X}{\partial x} \right] = \rho(T) \cdot Cp(T) \frac{\partial X}{\partial t} \qquad (3.46a)$$

j = 0: infinite slab

j = 1: infinite cylinder

j = 2: sphere

$$\left. \frac{\partial X}{\partial x} \right|_{x=0} = 0 \qquad x = 0$$

$$t \geq t^n \qquad (3.46b)$$

$$k(T)\frac{\partial X}{\partial x}\bigg|_{x=Lx} = 1 \qquad x = Lx,\quad t \geq t^n \qquad (3.46c)$$

$$X = 0 \qquad 0 \leq x \leq Lx,\quad t < t^n \qquad (3.46d)$$

The sensitivity coefficients were found using the same finite difference algorithm used to calculate the T^* values, only substituting the boundary condition shown in Eq. (3.46c) in place of the assumed heat flux (q^*) at the boundary at x = Lx, and noting that the sensitivity coefficients are equal to zero for $t < t^n$, that is, X = 0 was used as the initial condition.

3.5.2.2 Determination of the Temperature at the Surface

The temperature values at x = Lx are required to estimate the surface heat transfer coefficient (Eq. (3.30e)). These values were found by using the estimated heat flux values, $\hat{q}(t^n)$, in place of the boundary condition given in Eq. (3.18b), and solving Eq. (3.17) for the temperature at x = Lx, using the one dimensional finite difference algorithm. The boundary condition at x = Lx is shown below.

$$k(T)\frac{\partial T}{\partial x}\bigg|_{x=Lx} = \hat{q}(t) \qquad x = Lx,\quad t^n < t \leq t^{n+r} \qquad (3.47)$$

3.5.2.3 Estimation of the Surface Heat Transfer Coefficient

The surface heat transfer coefficient may be found from the estimated heat flux, calculated temperatures at x = Lx, and the measured ambient temperatures, for discrete times t^n, as, Beck et. al. (1985)

$$\hat{h}_x^n = \frac{\hat{q}^n}{T_\infty^n - 0.5\cdot(\hat{T}^n - \hat{T}^{(n-1)})\Big|_{x=0}} \tag{3.48}$$

where $\hat{T}^n$ and $\hat{T}^{(n-1)}$ are the estimated temperatures at x = 0 for times t^n and $t^{(n-1)}$.

A flow chart outlining the numerical solution of estimating the surface heat transfer coefficient is given in Appendix D, along a listing of the program written in Fortran 77, on a Vax 11/750 computer.

3.6 Quality Loss Prediction in Frozen Foods during Storage

The numerical solutions to the direct one and two dimensional heat conduction problems described in Section 3.3 were used to determine the rate of quality loss in frozen foods. The quality loss model developed by Heldman and Lai (1983), which assumes the rate of quality loss is a function of time and temperature, was used in this study. In this model, the rate of quality deterioration is an explicit function of time, temperature and the kinetic properties of the product. For the situation where product temperature is position dependent, that is, in pallet loads, quality will also be a function of position. Quality is often expressed in terms of the products shelf-life, and consequently, the units are expressed in days or months. The model, including the

spatial and temperal dependence of product quality or shelf-life is shown below.

$$Q(x,t) = Q_{0_r} - \int_0^t \exp\left\{ -\frac{Ea}{R} \left[\frac{1}{T(x,\tau)} - \frac{1}{T_r} \right] \right\} d\tau \qquad (3.49)$$

where Q(x,t) is the remaining quality or shelf-life of the food, if stored at the reference temperature T_r, given the temperature history of the product at the spatial location x, Q_{0_r} is the initial quality or shelf-life at a reference temperature T_r, Ea is the activation energy constant for the given food product, and R is the gas constant. Note, for two dimensional heat transfer, T = T(x,y,t), therefore, Q = Q(x,y,t). The integral is determined numerically by assuming the integrand is constant over each time step, and summing over the total time interval as

$$Q(x,t) = Q_{0_r} - \sum_{n=1}^{N} \exp\left\{ -\frac{Ea}{R} \left[\frac{1}{T(x,t)} - \frac{1}{T_r} \right] \right\} \Delta t \qquad (3.50)$$

where N is the total number of time steps.

The input parameters required in the solution are the product temperature distribution history, T(x,t), and the kinetic parameters, including the activation energy constant, Ea, and the initial product quality, Q_{0_r}, at a reference temperature, T_r. The product temperatures are obtained from the finite difference solution described previously. The kinetic parameters are evaluated using a statistically based method, such as, Chu (1983), from time-temperature data found in the literature, and are assumed to be known in this investigation. By using a statistically based method, the variance of the predicted quality loss may be determined from the variance of the estimated input kinetic parameters

$$V(Q) = \left[\frac{\partial Q}{\partial Q_{o_r}}\right]^2 \cdot V(Q_{o_r}) + \left[\frac{\partial Q}{\partial Ea}\right]^2 \cdot V(Ea)$$

$$+ 2\left[\left[\frac{\partial Q}{\partial Q_{o_r}}\right] \cdot \left[\frac{\partial Q}{\partial Ea}\right] \cdot cov(Q_{o_r}, Ea)\right] \qquad (3.51)$$

The initial product quality and the activation energy constant are independent (Chu, 1983), therefore, the covariance term drops out and

$$V(Q) = \left[\frac{\partial Q}{\partial Q_{o_r}}\right]^2 \cdot V(Q_{o_r}) + \left[\frac{\partial Q}{\partial Ea}\right]^2 \cdot V(Ea) \qquad (3.52)$$

Substituting Eq. (3.50) into Eq. (3.52) and performing the indicated differentiations yields

$$V(Q(x,t)) = 1 \cdot V(Q_{o_r}) + \left\{-\sum \left[\exp\left\{-\frac{Ea}{R}\left[\frac{1}{T(x,t)} - \frac{1}{T_r}\right]\right\}\right.\right.$$

$$\left.\left. \cdot \left\{-\frac{1}{R}\left[\frac{1}{T(x,t)} - \frac{1}{T_r}\right]\right\} \cdot \Delta t\right]\right\}^2 \cdot V(Ea) \qquad (3.53)$$

In summary, this procedure allows the estimation of quality loss and its associated variance in frozen foods as a function of time and position, based on predicted internal temperature measurements and estimated kinetic parameters and associated variances.

CHAPTER 4.

EXPERIMENTAL PROCEDURES

Frozen foods in fluctuating temperature storage conditions were simulated experimentally using the Karlsruhe Test Substance (Gutschmidt, 1960), as a substitute food product. The test substance was alternately placed in two adjacent storage chambers (A and B) with average ambient temperatures of -6 and -30°C, respectively. Temperature distributions within the test substance were determined from thermocouple readings recorded by a data aquisition computer. These measurements were used in confirming the one and two dimensional finite difference models, discussed in Section 3.4, and in estimating the surface heat transfer coefficients, discussed in Section 3.5.3.

4.1 Karlsruhe Test Substance

The Karlsruhe Test Substance, developed at the Federal Research Institute, West Germany, is a highly concentrated methyl-cellulose mixture, which has similar thermal properties and freezing characteristics as common food products. Methyl-cellulose mixtures have been used in studying the freezing process in foods by many researchers, such as, Bonacina, et. al. (1973), Cleland and Earle (1977b), and Succar and Hayakawa (1986).

One half gallon paperboard ice cream containers, measuring 0.170 m long, 0.125 m across, and 0.90 m high, were used as the product containers. The test substance was shaped into brickettes to fit into the boxes, wrapped twice with plastic film to prevent moisture loss, and the placed in the paperboard containers. The thickness of then paperboard boxes and the two layers of plastic film were measured with a micrometer. The paperboard box thickness averaged 1.7 mm, and the plastic film averaged 0.3 mm. The volume of the containers was 0.0019 m^3. Assuming a density of 1040 kg/m^3 for the Karlsruhe Test Substance (Specht et. al., 1981), 1.98 kg of test substance were required for each container.

The following procedure was followed to prepare the test substance (Gutschmidt, 1960):

1. The water content of the supplied methyl-cellulose (MC) was found from the manufacturers (Dow Chemical, U.S.A), and the additional water required for a test substance with 77% moisture content was determined. This amount of water was increased by approximately 4% of the weight of the methyl-cellulose, since, as Gutschmidt, (1960) has shown, this amount of water vaporizes in the warming of the water, and in the stirring, kneading and forming of the methyl-cellulose.
2. Salt (NaCl) was added to depress the freezing point of the methyl-cellulose from -0.6°C to -1.0°C. The amount of salt added was determined from Adballa and Singh (1984).

$$\text{SALT (kg)} = 0.024 \cdot (1 - 0.01 \cdot \text{MC})$$

3. In addition, 1 gm parachorometacresol was added as a preservative for every 100 grams added water.

4. The water was warmed in a large beaker to approximately 60-70°C, and then the heating source was turned off.

5. The salt and parachorometacresol were added while stirring continuously.

6. The methyl-cellulose powder, Methocel A4m Premium, supplied by Dow Chemical, U.S.A., was slowly poured into the salt water mixture, and was stirred continuously until the solution became homogeneous.

7. The test substance was allowed to cool to 35-40°C.

8. The substance was kneaded until it formed a bread-like dough.

9. Brickettes were formed by flattening the sides on a smooth planar surface. They were then wrapped in plastic film, and placed in the paperboard containers.

The following quantities were required for every 1 kg Karlsruhe Test Substance:

0.772 kg water

0.237 kg methyl-cellulose (4500 cps, 2% moisture content)

5.57 gm NaCl

0.76 gm parachlorometacresol

For better mixing, the test substance was prepared in 0.99 kg batches, and two batches were combined just prior to kneading to form each brickette.

There is some discrepancy between the various thermal properties for the Karlsruhe Test Substance given in the literature. The initial freezing point given by Gutschmidt (1960) has not been used by other researchers (Specht, et. al., 1981). Values for T_{if}, ρ_p, k_p, and Cp_p

are shown in Table 4.1. The properties given by Specht, et. al. (1981) were used in succeeding calculations for consistency, unless otherwise noted.

4.2 Containers for the Karlsruhe Brickettes

The Karlsruhe Test Substance was used in the following three ways: (1) in the determination of the surface heat transfer coefficients; (2) for comparison with the one dimensional numerical model; and (3) for comparison with the two dimensional model. This resulted in three configurations for the containers to hold the Karlsruhe samples.

The first configuration was used in estimating the surface heat transfer coefficient. A container was constructed to hold three adjacent pairs of brickettes with an exposed top surface, and insulation around the remaining three sides. An open topped box or trough, consisting of 0.0125 m plywood, formed structural support for the container. Foam insulation, 0.077 m in thickness, was glued to the interior sides of the box. This insulation thickness was based on that used by other researchers using the Karlsruhe Test Substance (Cleland and Earle, 1977b, and Succar and Hayakawa, 1986). Values for the thermal conductivity of the foam insulation found in the literature varied around 0.28-0.31 W/m°C (Baumeister, et. al., 1978) and 0.35 W/m°C (Cleland and Earle, 1977b).

The box was constructed in two parts, and clamped shut after filling, for better packing of the Karlsruhe Test Substance brickettes. The container with brickettes in place is shown in Figure 4.1.

For the second configuration, a similar container was constructed, but without a bottom, to hold six brickettes in a single layer. This layer was secured to the top of the first configuration to form a double

Table 4.1 Thermal Properties of the Karlsruhe Test Substance

Moisture Content (%)	Initial Freezing Temperature (°C)	Density (kg/m^3)	Thermal Conductivity (W/m°C)	Specific Heat (kJ/kg°C)	Reference
77%	-1.0				Gutschmidt (1960)
77%	-0.7	1040	0.944	3.8	Specht (1981)

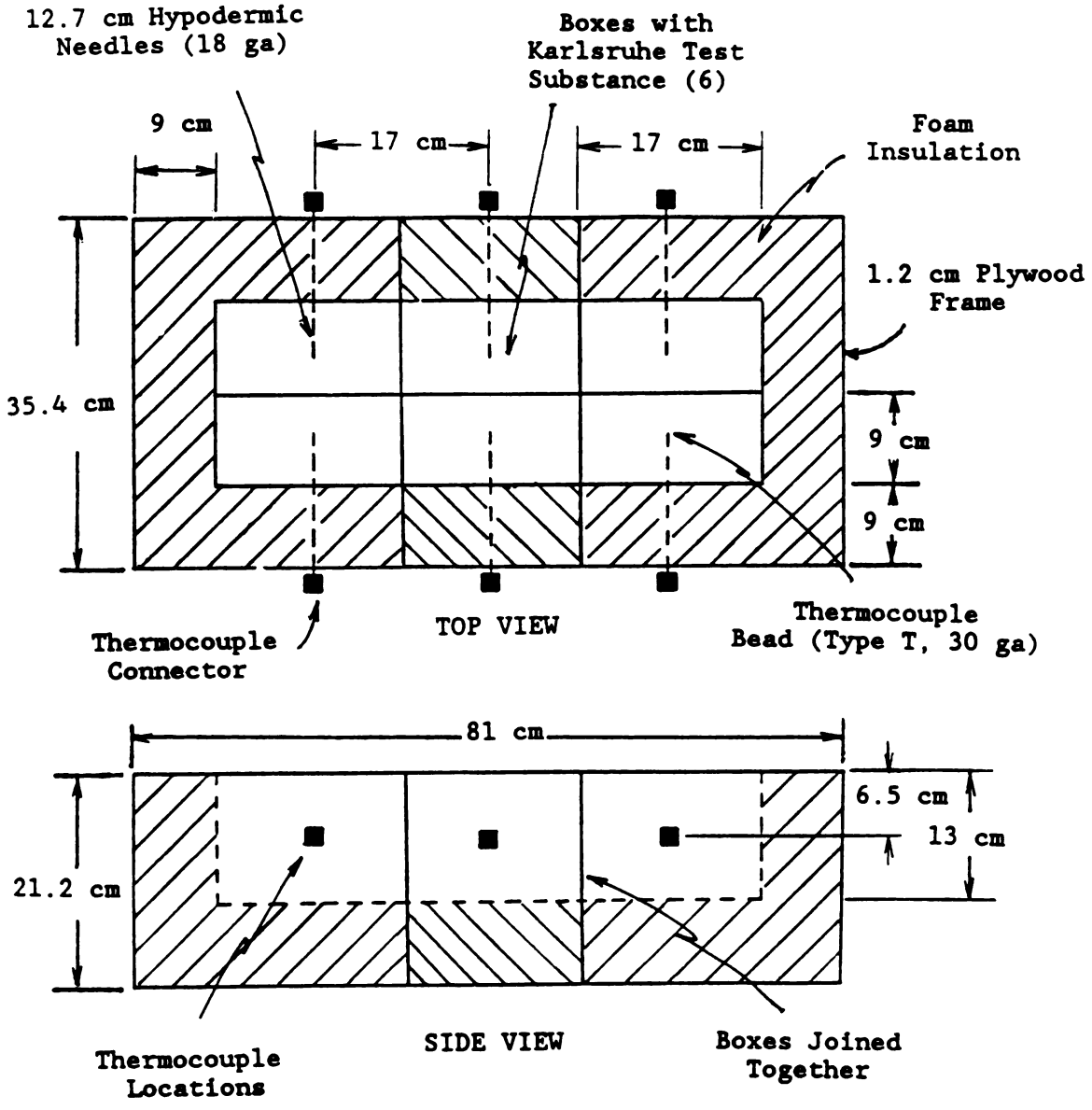


Figure 4.1 Single Layer, One Dimensional Container Configuration.

layer slab with only the top side exposed. The configuration is shown is Figure 4.2.

In the final case, two adjacent sides were exposed to allow two dimensional heat conduction. An insulated plywood box was constructed to hold twelve brickettes, in three layers. The container holding the brickettes is shown in Figure 4.3.

4.3 Temperature Measurement

Internal product temperature measurements were obtained using 30 gage, Type T, thermocouples (Omega, 1985). The thermocouples were placed at the geometric center of each brickette through 0.127 m long 18 ga hypodermic needles.

Prior to placement in the brickettes, the thermocouple wire was threaded through the needles, and soldered at the point of the needle (Figure 4.4). The end cavity of the needle was filled with epoxy to keep the thermocouple in place. The hypodermic needle was placed through a hole drilled through the plywood and the insulation at the desired thermocouple location. This served as a guide for the hypodermic needle to minimize the variability of the thermocouple location within the brickette. The ends of the thermocouples were insulated with foam to limit conduction down the thermocouple wire and hypodermic needle. The placement of the thermocouples in the three configurations are also shown in Figures 4.1, 4.2, and 4.3.

The ambient temperatures were measured using a thermocouple placed above the Karlsruhe container, midway between the container and the ceiling of the storage chamber.

All thermocouples measurements were recorded by a Hewlett Packard 3497A Data Aquisition/Control unit, coupled to a Hewlett Packard 85 desk

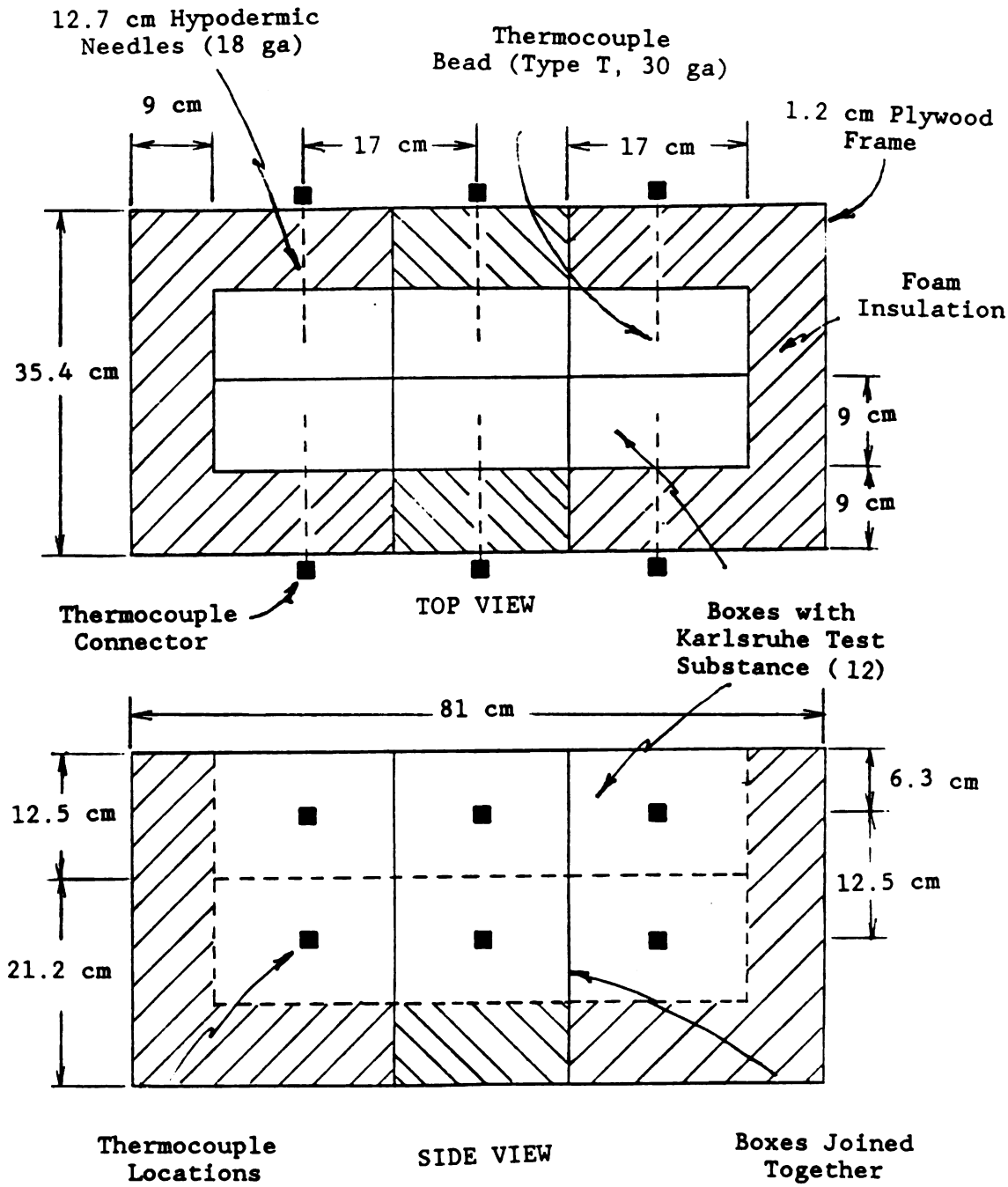


Figure 4.2 Double Layer, One Dimensional Container Configuration.

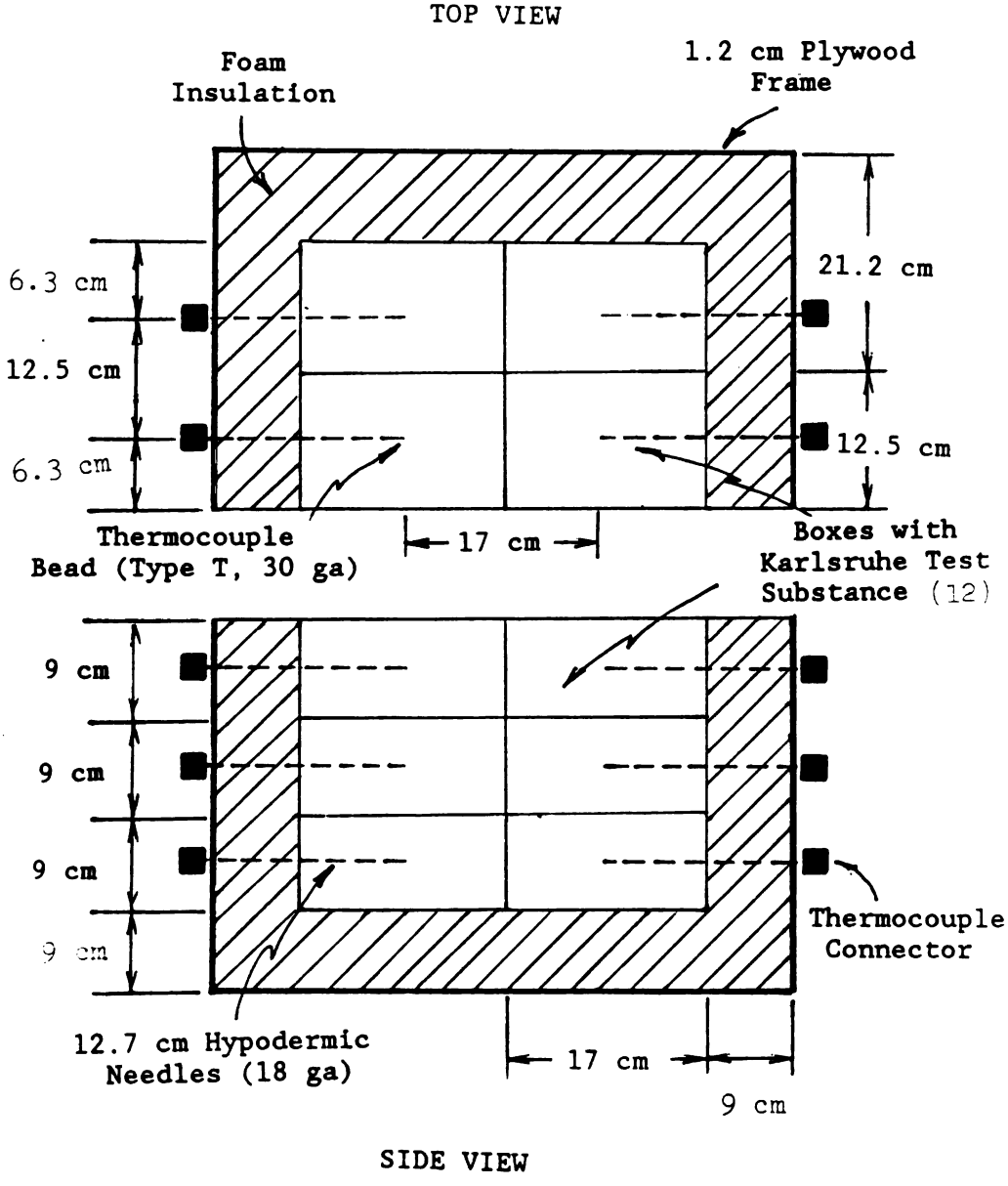


Figure 4.3 Triple Layer, Two Dimensional Container Configuration.

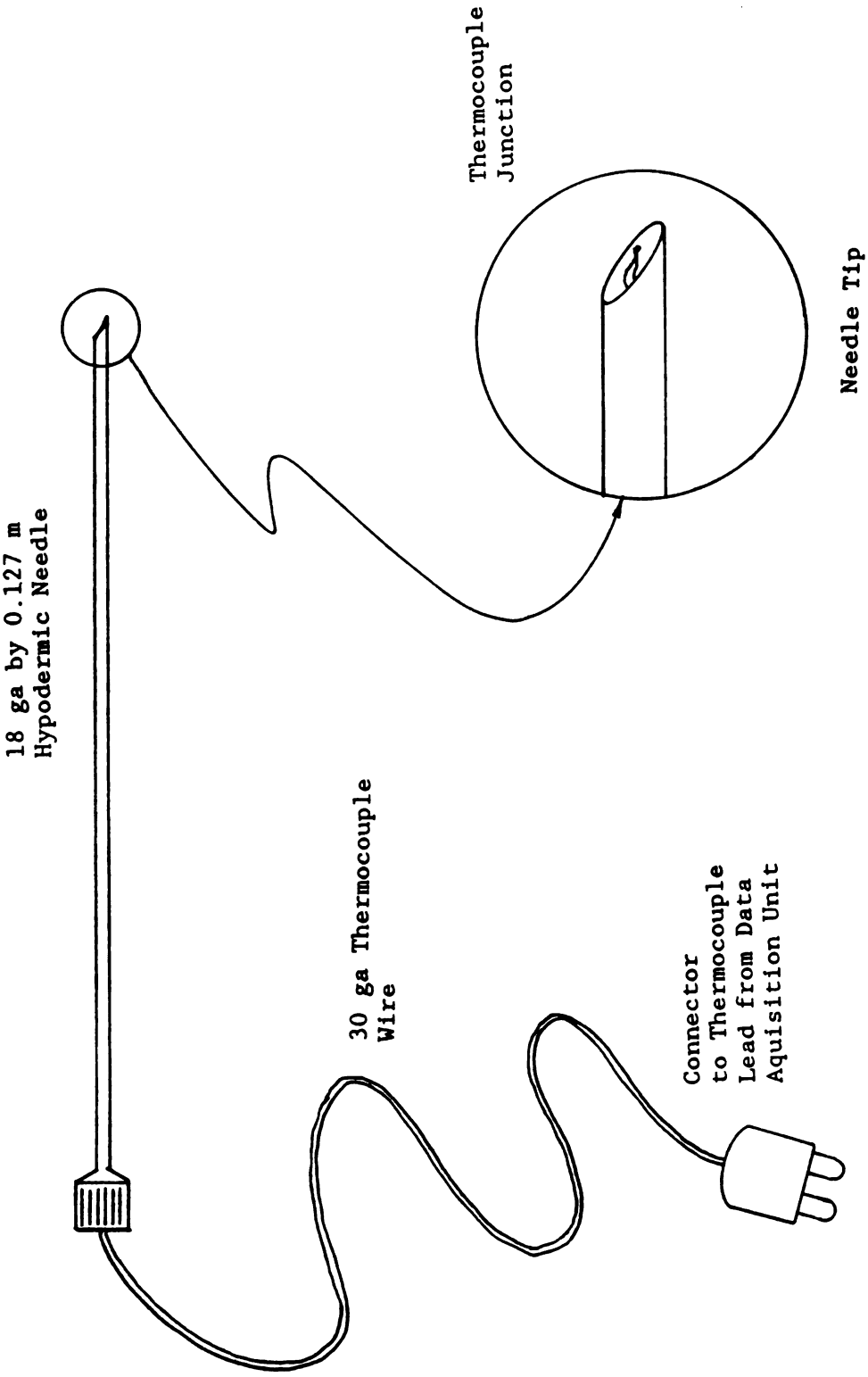


Figure 4.4 Thermocouple and Hypodermic Needle Assembly.

top computer. The capacity of the data aquisition device was limited to a maximum of 16 simultaneous thermocouple measurements over a total of 216 time steps.

4.4 Velocity Measurements

The air free stream velocity was required for the analytical determination of the surface heat transfer coefficient. The two storage chambers used in the study measured 2.20 m across by 6.90 m long by 2.53 m high for chamber A, and 1.81 m across by 6.90 m long by 2.53 m high for chamber B. The forced air flow in both chambers resulted from two 0.37 kW fans., placed in the upper end corners of both chambers. The fans were at the same end in chamber A, and they were at opposite ends in chamber B. Velocity measurements were taken at 0.045 m increments in a grid pattern about the region where the Karlsruhe containers were placed, using a hot wire anemometer (Model 2440, Weathertronics, Inc.). The measurements were taken in the middle of a defrost cycle. The horizontal velocities in chamber A ranged from 0.1 - 0.6 m/s and averaged 0.25 m/s, and ranged from 0.8 to 1.25 in chamber B, and averaged 1.0 m/s. The airflow in chamber A was lower due to a large obstruction in the storage room near the fans. Vertical velocity measurements in chamber B varied from $\pm$ 0.5 to $\pm$ 1.0 m/s, indicating mixed air flow conditions. Streamers were used as a visual conformation of the flow conditions. The streamers, made of magnetic tape, were hung from string placed at 0.45 m intervals, in a grid pattern, across each chamber. Consistent with the velocity measurements, the streamers fluttered up and down randomly in chamber B, and showed comparatively streamline characteristics in chamber A.

4.5 Experimental Storage Conditions

The Karlsruhe brickettes were prepared just prior to placement in the plywood containers, and then immediately placed in chamber B until equilibrium conditions were obtained (approximately four to five days).

One complete step change cycle in storage temperature was used for all configurations. The Karlsruhe Test Substance, initially at equilibrium in chamber B, was placed in chamber A for a given storage time period, and then placed back in chamber B for an equivalent time period, to complete the cycle. The storage period was limited by the capacity of the data aquisition unit. Since the temperature measurements obtained using the first configuration, shown in Figure 4.1, were required in the sequential regularization solution for the surface heat transfer coefficients, a smaller time interval for the temperature measurements was used, than in the other two cases. The time steps and the storage time periods for the three cases are shown in Table 4.2.

Each test was repeated three times, using the same Karlsruhe samples for each configuration. The samples were allowed to equilibrate in chamber B after the conclusion of each test. In subsequent sections, the three repetitions using the single layer, double layer, and two dimensional triple layer configurations will be referred to as Tests 1a-c, Tests 2a-c, and Tests 3a-c, respectively.

The defrost cycle period was approximately 1.5 hours in storage chamber A, and four hours in chamber B. Some frost accumulated on the surface of each container after storage in chamber B. The average storage temperature in chamber A was -6°C for all test cases. The average storage temperature varied from -33 to -34°C for all three tests using the first configuration (single layer), but due to a failing

Table 4.2 Storage Times and Measurement Intervals for the Three Configurations

Test	Storage Period in each Chamber (hours)	Measurement Interval (mins)
1 (single layer)	18	10
2 (double layer)	24	30
3 (two dimensional)	48	30

compressor, increased to -26°C by the end of the final test using the third configuration (two dimensional case).

At the conclusion of the tests, the hypodermic needles were removed from the Karlsruhe brickettes, and the brickettes were removed from the containers. Due to the starchy nature of the Karlsruhe Test Substance, the hole left by the hypodermic needle was left in tack. To determine the exact location of the thermocouples, red food dye was placed in the hole using a glass pipette, and each brickette was cut open to revel the end point of the hole. This location was then measured and recorded.

Finally, the distance between the Karlsruhe brickette and the outer surface of the paperboard carton was measured. The thickness of the paperboard carton and the plastic film were subtracted from this value to obtain the air interspace thickness between the carton and the Karlsruhe test substance. The air interface was found to vary between one and ten millimeters.

CHAPTER 5.

DETERMINATION OF NUMERICAL PARAMETERS

The numerical analysis in this investigation of frozen foods during storage focuses on three major problems: (1) the development of a numerical procedure to estimate heat transfer coefficients typical of storage conditions (indirect problem); (2) the development of a multi-dimensional model to simulate temperature changes within frozen foods during storage (direct problem); and (3) The estimation of temperature dependent quality deterioration. These models were described in detail in Chapter 3. Application of these models require input of parameters inherent to the problem being studied, that is, product properties and geometry, and user specified parameters inherent to the numerical procedure. Input parameters, determined by the user, intrinsic in the finite difference solutions of both the direct and indirect problems, include the number of nodes and the time step. In addition, the user must select the magnitude of the regularization parameter, the order of regularization, the time between temperature measurements, and the number of future temperature measurements used in the solution for the indirect problem.

An investigation of the influence of the user definable parameters on the numerical models described above was completed to provide a systematic procedure for optimal parameter selection. Since the analytical solution of the nonlinear problem with temperature dependent product properties is not readily obtainable, this investigation of user

definable parameters focused on the constant property solution, for which exact solutions exist. The resulting observations and conclusions from the study of the linear problem was used as a basis for the selection of the parameters in the nonlinear case.

5.1 Selection of Parameters Inherent in the Finite Difference Solution

The one dimensional direct problem is an intrinsic part of the solution of the inverse problem, and it provides a basis for the solution of the two dimensional problem. Because of its importance in both problems, the influences of the user adjustable parameters; node spacing and time steps, on the accuracy and numerical oscillatory tendencies of the one dimensional solution, were studied in detail. Results from this analysis were also used in the indirect problem, and expanded upon in the analysis of the two dimensional problem.

The number of nodes and the time step, along with the product properties and boundary conditions, are important factors determining the numerical oscillatory tendencies and the accuracy of the numerical solution. In both cases, the eigenvalues resulting from the set of finite difference equations, Eq. (3.19), play an important role in the analysis.

5.1.1 Numerical Oscillations

In evaluating the numerical oscillation criteria, the matrix equation shown in Eq. (3.19) is considered. Multiplying Eq. (3.19) by $\mathbf{A}^{-1}$ results in:

$$\mathbf{T}^{n+1} = \mathbf{A}^{-1}\mathbf{B}\cdot\mathbf{T}^{n} + \mathbf{A}^{-1}\mathbf{D} \tag{5.1}$$

Numerical oscillations are related to the eigenvalues of the matrix $\mathbf{A}^{-1}\mathbf{B}$; oscillations will occur in a stable solution if some of the eigenvalues are negative, but greater than -1, (Meyers, 1971). Therefore, to avoid oscillations, $\mathbf{A}^{-1}\mathbf{B}$ must be positive definite. Segerlind (1984) showed that the study of the eigenvalues of $\mathbf{A}^{-1}\mathbf{B}$ may be reduced to a study of **A** and **B** where:

$$\mathbf{A} = \frac{1}{\eta\Delta t}\mathbf{C} + \mathbf{K} \tag{5.2a}$$

$$\mathbf{B} = \frac{1}{\beta\Delta t}\mathbf{C} - \mathbf{K} \tag{5.2b}$$

The **C** and **K** matrices contain the heat capacity (ρ•Cp) terms and the thermal conductivity (k) terms, respectively. For the simplified case of a one dimensional finite slab with constant product properties, and insulated at the first boundary (x = 0), and with a convective boundary condition at the second boundary (x = Lx), the **C** and **K** matrices are given below.

$$\mathbf{C} = \begin{bmatrix} \frac{\rho Cp\Delta x}{2} & 0 & \cdots & 0 \\ 0 & \rho Cp\Delta x & & \\ & & \rho Cp\Delta x & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & 0 & \frac{\rho Cp\Delta x}{2} \end{bmatrix} \tag{5.3a}$$

$$\mathbf{K} = \begin{bmatrix} \frac{k}{\Delta x} & -\frac{k}{\Delta x} & 0 & \cdots & 0 \\ -\frac{k}{\Delta x} & \frac{2k}{\Delta x} & -\frac{k}{\Delta x} & & \vdots \\ 0 & -\frac{k}{\Delta x} & \frac{2k}{\Delta x} & -\frac{k}{\Delta x} & 0 \\ & & \ddots & \ddots & \ddots \\ 0 & \cdots & & -\frac{k}{\Delta x} & \frac{k}{\Delta x} + hx_{Lx} \end{bmatrix} \qquad (5.3b)$$

For $\mathbf{A}^{-1}\mathbf{D}$ to be positive definite, **A** and **B** must both be positive definite. The condition for **A** is satisfied since from Eqs. (5.3a,b), **C** is positive definite, and **K** is positive definite in this case because of the hx_{Lx} term (Segerlind, 1984). The second matrix in question, **B** is positive definite if (Fried, 1979)

$$\det\left[\mathbf{K} - \lambda_\ell \mathbf{C} \right] = 0 \qquad \ell = 1, 2, \ldots, L$$

(total No. of nodes)

where

$$\lambda_\ell = \frac{1}{\Delta t_\ell \cdot \beta} \qquad (5.4)$$

and

β = the weighting coefficient used in the numerical method

Therefore, considering the worst case where $\lambda_\ell = \lambda_{max}$, the maximum eigenvalue, numerical oscillations can be avoided if ($\beta = 0.5$)

$$\Delta t_{osc} < \frac{2}{\lambda_{max}} \tag{5.5}$$

The following hypothetical problem, similar to the conditions used during the experimental procedures for the single product layer described in Section 4.5, with the exception of the use of constant thermal properties, was considered in the determination of the oscillatory criteria

$$\frac{\partial^2 T}{\partial x^2} = \frac{\rho Cp}{k} \cdot \frac{\partial T}{\partial t} \qquad 0 \le x \le Lx \quad 0 < t \le t_1 \tag{5.6a}$$

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0 \qquad 0 < t \le t_1 \tag{5.6b}$$

$$\left.\frac{\partial T}{\partial x}\right|_{x=Lx} = hx_{Lx}(T_{\infty,Lx} - T(Lx,t)) \qquad 0 < t \le t_1 \tag{5.6c}$$

$$T(x,0) = T_0 \qquad 0 \le x \le Lx \tag{5.6d}$$

where

$Lx = 0.13$ m

$T_{\infty,Lx} = -5°C$

$T_0 = -33°C$

$t_1 = 18$ hours

Experimental results from similar conditions indicated that the temperature of the Karlsruhe Test Substance at the mid-section of the layer changed from ≈ -33°C to ≈ -12°C after 18 hours. Therefore, in the

numerical oscillatory analysis, the problem given by Eqs. (5.6a-d) was considered using constant thermal properties of the Karlsruhe Test Substance evaluated at the two extreme temperatures of -33°C and -12°C. The value of the surface heat transfer coefficient was set at 7.85 W/m^2C, which is consistent with Dagerskog (1974), who reported surface heat transfer coefficients ranging from 4 to 13 W/m^2C, using transducers, for free air over a frozen food pallet in storage.

The heat capacity and conductivity matrices shown in Eqs. (5.3a,b), respectively, were evaluated using 5, 9, 13, and 17 nodes, and with the thermal properties evaluated at -33 and -12°C, using Eqs. (3.3), (3.6a-c), and (3.9) for ρ, k and Cp, respectively. Table 5.1 shows the values of Δx, ρ, k, and Cp used for the eight cases considered.

Equation (5.4) was solved for λ_ℓ using the Jacobi Method (Bathe and Wilson, 1976), for each hypothetical condition. The largest eigenvalue (λ_{max}) was used to determine Δt_{osc} according to Eq. (5.5), for each case considered. The values for λ_ℓ and Δt_{osc} are shown in Table 5.2a,b.

The results indicate that as the number of nodes increase, the time step to prevent numerical oscillations decreases. In addition, the time step limitation was much more severe for the cases where the thermal properties were evaluated at -33°C, rather than -5°C; for all values of Δx investigated, the critical time steps found using thermal properties at -33°C were ≃ 60% of those values found using the thermal properties at -12°C. This coincides with the inverse ratio of the thermal diffusivities (κ) evaluated at the respective temperatures. Thus

$$\frac{\kappa(-12)}{\kappa(-33)} = \frac{\Delta t_{osc}(-33)}{\Delta t_{osc}(-12)} \qquad (5.7)$$

Table 5.1 Values for Δx, ρ, k, and Cp Used in Evaluating λ_{max} (hx_{Lx} = 7.85 W/m°C).

Case No.	No. of Nodes	Δx (m)	ρ (kg/m^3)	k (W/m°C)	Cp (kJ/kg°C)	comments
1	5	0.03250	972.2	2.24	2.47	Properties evaluated at -33°C
2	9	0.01625	972.2	2.24	2.47	
3	13	0.01083	972.2	2.24	2.47	
4	17	0.00813	972.2	2.24	2.47	
5	5	0.03250	974.6	1.98	3.66	Properties evaluated at -12°C
6	9	0.01625	974.6	1.98	3.66	
7	13	0.01083	974.6	1.98	3.66	
8	17	0.00813	974.6	1.98	3.66	

Table 5.2a Eigenvalues and Resulting Critical no Oscillations Time Step (sec) for Properties Evaluated at -33°C.

Eigenvalues (sec^{-1})	Case No. (No. of Nodes) 1 (5)	2 (9)	3 (13)	4 (17)
λ_1	$2.1858 \cdot 10^{-5}$	$2.1736 \cdot 10^{-5}$	$2.1732 \cdot 10^{-5}$	$2.2235 \cdot 10^{-5}$
λ_2	$5.6591 \cdot 10^{-4}$	$5.8627 \cdot 10^{-4}$	$5.9016 \cdot 10^{-4}$	$5.9197 \cdot 10^{-4}$
λ_3	$1.8149 \cdot 10^{-3}$	$2.1176 \cdot 10^{-3}$	$2.1781 \cdot 10^{-3}$	$2.1999 \cdot 10^{-3}$
λ_4	$3.0637 \cdot 10^{-3}$	$4.4082 \cdot 10^{-3}$	$4.7028 \cdot 10^{-3}$	$4.8096 \cdot 10^{-3}$
λ_5	$3.5591 \cdot 10^{-3}$	$7.1100 \cdot 10^{-3}$	$7.9929 \cdot 10^{-3}$	$8.3214 \cdot 10^{-3}$
λ_6		$9.8117 \cdot 10^{-3}$	$1.1824 \cdot 10^{-2}$	$1.2601 \cdot 10^{-2}$
λ_7		$1.2102 \cdot 10^{-2}$	$1.5936 \cdot 10^{-2}$	$1.7483 \cdot 10^{-2}$
λ_8		$1.3633 \cdot 10^{-2}$	$2.0047 \cdot 10^{-2}$	$2.2780 \cdot 10^{-2}$
λ_9		$1.4149 \cdot 10^{-2}$	$2.3878 \cdot 10^{-2}$	$2.8289 \cdot 10^{-2}$
λ_{10}			$2.7169 \cdot 10^{-2}$	$3.3798 \cdot 10^{-2}$
λ_{11}			$2.9693 \cdot 10^{-2}$	$3.9095 \cdot 10^{-2}$
λ_{12}			$3.1281 \cdot 10^{-2}$	$4.3977 \cdot 10^{-2}$
λ_{13}			$3.1800 \cdot 10^{-2}$	$4.8256 \cdot 10^{-2}$
λ_{14}				$5.1768 \cdot 10^{-2}$
λ_{15}				$5.4378 \cdot 10^{-2}$
λ_{16}				$5.5985 \cdot 10^{-2}$
λ_{17}				$5.6506 \cdot 10^{-2}$
Δt_{osc}	562	141	63	35

Table 5.2b Eigenvalues and Resulting Critical no Oscillations Time Step (sec) for Properties Evaluated at -12°C.

Eigenvalues (sec^{-1})	Case No. (No. of Nodes) 5 (5)	6 (9)	7 (13)	8 (17)
λ_1	$1.4405 \cdot 10^{-5}$	$1.4390 \cdot 10^{-5}$	$1.4146 \cdot 10^{-5}$	$1.4383 \cdot 10^{-5}$
λ_2	$3.4016 \cdot 10^{-4}$	$3.5241 \cdot 10^{-4}$	$3.5438 \cdot 10^{-4}$	$3.5537 \cdot 10^{-4}$
λ_3	$1.0829 \cdot 10^{-3}$	$1.2633 \cdot 10^{-3}$	$1.2986 \cdot 10^{-3}$	$1.3114 \cdot 10^{-3}$
λ_4	$1.8254 \cdot 10^{-3}$	$2.6256 \cdot 10^{-3}$	$2.7995 \cdot 10^{-3}$	$2.8629 \cdot 10^{-3}$
λ_5	$2.1186 \cdot 10^{-3}$	$4.2324 \cdot 10^{-3}$	$4.7554 \cdot 10^{-3}$	$4.9507 \cdot 10^{-3}$
λ_6		$5.8391 \cdot 10^{-3}$	$7.0331 \cdot 10^{-3}$	$7.4946 \cdot 10^{-3}$
λ_7		$7.2013 \cdot 10^{-3}$	$9.4773 \cdot 10^{-3}$	$1.0397 \cdot 10^{-2}$
λ_8		$8.1121 \cdot 10^{-3}$	$1.1922 \cdot 10^{-2}$	$1.3546 \cdot 10^{-2}$
λ_9		$8.4158 \cdot 10^{-3}$	$1.4199 \cdot 10^{-2}$	$1.6821 \cdot 10^{-2}$
λ_{10}			$1.6155 \cdot 10^{-2}$	$2.0096 \cdot 10^{-2}$
λ_{11}			$1.7656 \cdot 10^{-2}$	$2.3245 \cdot 10^{-2}$
λ_{12}			$1.8600 \cdot 10^{-2}$	$2.6148 \cdot 10^{-2}$
λ_{13}			$1.8907 \cdot 10^{-2}$	$2.8692 \cdot 10^{-2}$
λ_{14}				$3.0779 \cdot 10^{-2}$
λ_{15}				$3.2331 \cdot 10^{-2}$
λ_{16}				$3.3287 \cdot 10^{-2}$
λ_{17}				$3.3595 \cdot 10^{-2}$
Δt_{osc}	944	238	106	60

In the application to the nonlinear problem, given Δt_{osc} for a specific value of the thermal diffusivity evaluated at T_1, the critical time step using thermal properties evaluated at T_2 can be determined by multiplying Δt_{osc} evaluated at T_1 by the inverse ratio of the respective thermal diffusivities. This provides a basis for using a variable step in the solution of the nonlinear problem.

An alternate method of estimating λ_{max} was presented by Fried (1979). He found that the maximum eigenvalue for a global finite element matrix is less than or equal to the maximum eigenvalue for all of its elemental matrices, that is

$$\lambda_{max} \leq \max_{e} \left\{ \lambda_{max}^{(e)} \right\} \tag{5.8}$$

where $\lambda_{max}^{(e)}$ is the maximum eigenvalue of the e^{th} element matrix. The heat capacity and thermal conductivity matrices for a one dimensional 'element' using finite differences is found from the contributions from two adjacent nodes. These matrices are given below.

$$\mathbf{c}^{(e)} = \frac{\rho \cdot Cp \cdot \Delta x}{2} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.9a}$$

$$\mathbf{k}^{(e)} = \frac{k}{\Delta x} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{5.9b}$$

The maximum eigenvalue was found from Eq. (5.4) by replacing the global matrices by the element matrices as

$$\det \left[\mathbf{k}^{(e)} - \lambda_{\ell}^{(e)} \cdot \mathbf{c}^{(e)} \right] = 0 \tag{5.10}$$

Solving for $\lambda_{max}^{(e)}$ from the determinant yielded

$$\lambda_{max}^{(e)} = \frac{4 \cdot k}{\rho \cdot Cp \cdot (\Delta x)^2} \tag{5.11}$$

The maximum eigenvalues calculated from the elemental matrix ($\lambda_{max}^{(e)}$) for 5, 9, 13, and 17 nodes and thermal properties evaluated at -33 and -12°C. Upon comparison with λ_{max} calculated from the global matrices (Table 5.2a,b), it was found that

$$\lambda_{max} \approx \lambda_{max}^{(e)} \tag{5.12}$$

for the finite difference grid. The maximum eigenvalues calculated from the elemental matrices and the global matrices are compared in Table 5.3.

5.1.2 Accuracy

The accuracy of the numerical solution of the non-linear problem described by Eqs. (3.11), (3.12), and (3.13a,b) is difficult to determine, because the analytical solution is impractical to obtain. Insight into the accuracy of the nonlinear problem was achieved, however, by considering the linear problem with constant properties.

The analytical solution to the simplified one dimensional problem described by Eqs. (5.6a-d) is given by Carslaw and Jaeger (1959) as

$$T(x,t) = T_0 + (T_{\infty,Lx} - T_0) \cdot \left[1 - \sum_{\ell=1}^{\infty} R_\ell e^{-\lambda_\ell^e t} \right] \tag{5.13a}$$

where

Table 5.3 Comparison of $\lambda_{max} \cdot 10^2$ and $\lambda_{max}^{(e)} \cdot 10^2$.

	Temperature (No. of Nodes)							
	-33°C				-12°C			
(sec^{-1})	(5)	(9)	(13)	(17)	(5)	(9)	(13)	(17)
λ_{max}	0.356	1.415	3.180	5.651	0.212	0.842	1.891	3.359
$\lambda_{max}^{(e)}$	0.353	1.412	3.177	5.648	0.210	0.839	1.889	3.357
%	99.2	99.8	99.9	99.9	99.0	99.7	99.9	99.9

$$R_\ell = \frac{2 \cdot hk \cdot \cos(\zeta_\ell x)}{[(hk^2 + \zeta_\ell^2) \cdot Lx + hk] \cdot \cos(\zeta_\ell Lx)}$$

$$\lambda_\ell^e = (k/\rho Cp) \cdot \zeta_\ell^2$$

$$hk = hx_{Lx}/k$$

and ζ_ℓ is the ℓ^{th} root of the transcendental equation

$$\zeta_\ell \cdot \tan(\zeta_\ell \cdot Lx) = Lx \cdot hk \qquad (5.13b)$$

The eigenvalues found in the numerical solution, λ_ℓ, are approximations to the exponential term, λ_ℓ^e, of the analytical solution. Since there are a finite number of eigenvalues in the numerical solution, it is important to determine how many of the terms in the summation shown in Eq. (5.13a) are significant for any given time. Considering terms < 0.01°C to be insignificant, the time for any given term in the summation to be insignificant can be calculated for a specific location of x. Values for $R_{\ell,x}$ were calculated, with thermal properties evaluated at -33 and -12°C, for the first nine values of ζ_ℓ (Abramowitz and Stegun, 1965), at the insulated boundary (x = 0), the mid-section (x = Lx/2), and at the convective boundary (x = Lx). Due to the insulated boundary condition and resulting damping effects, the values for $R_{\ell,x}$ at x = 0, ($R_{\ell,0}$), were greater in magnitude than those at the other locations. Consequently, the values of $R_{\ell,0}$ were the most crucial values in determining the times (t_ℓ^∞) for each term in the summation to be insignificant. The values for t_ℓ^∞ were calculated for each term such that

$$\left| R_{\ell,0} \cdot e^{-(\lambda_\ell^e \cdot t_\ell^\infty)} \right| < 0.01 \qquad (5.14)$$

Values for λ_ℓ^e, $R_{\ell,0}$, $R_{\ell,Lx/2}$, $R_{\ell,Lx}$ and t_ℓ^∞ evaluated using $R_{\ell,0}$ for the first nine terms in the summation are found in Tables 5.4a,b for thermal properties evaluated at -33 and -12°C, respectively. The values for t_ℓ^∞ monotonically decreased to less than one after only four eigenvalues; therefore, the the damping effects of the exponential factors were more influential on the behavior of t_ℓ^∞ than the sinusiodal effects of the R_ℓ terms. Since only the first three eigenvalues were significant after 100 seconds, these terms were considered the most the important when comparing the eigenvalues of the numerical solution (λ_ℓ) to the exponential factors of the analytical solution (λ_ℓ^e). In addition, due to the differences in thermal properties, the t_ℓ^∞ values evaluated at -12°C were greater than those values evaluated at -33°C (Eq. (5.7)), and consequently, there were more significant terms using the thermal properties evaluated at -12°C than at -33°C.

Since the first three exponential terms were the most significant in the analytical solution, it was decided that the first three eigenvalues of the numerical solution be within 5% of the corresponding analytical terms. The eigenvalues were calculated as percentages of the associated λ_ℓ^e terms for the eight conditions shown in Tables 5.2a,b with $\ell \leq 9$ in Figure 5.1. From these results, three eigenvalues were found within 5% of λ_ℓ^e using nine or more nodes, for the thermal properties evaluated at both -33 and -12°C. Therefore, $\Delta x = 0.01625$ meters (Table 5.1) was the maximum spatial increment considered, given the 5% accuracy criteria.

Segerlind (1986) has proposed that the limiting time step with regards to accuracy be based on the time for the solution to reach steady state (t_{ss}). The first eigenvalue controls the time required to reach steady state. Considering the exponential term as the dominant factor at t_{ss}, he has suggested the following criteria for determining the maximum time step for accuracy (Δt_{ac})

Table 5.4a Summation Terms in Series Solution and Resulting Time for Terms to Vanish (Properties Evaluated at -33°C).

ℓ	λ_ℓ^e (sec^{-1})	$R_{\ell,0}$ (°C)	$R_{\ell,Lx/2}$ (°C)	$R_{\ell,Lx}$ (°C)	t_ℓ^∞ (sec)
1	$2.173 \cdot 10^{-5}$	1.065	1.01	0.86	214,818
2	$5.932 \cdot 10^{-4}$	$-8.05 \cdot 10^{-2}$	$5.55 \cdot 10^{-2}$	$7.98 \cdot 10^{-2}$	3,517
3	$2.227 \cdot 10^{-3}$	$2.22 \cdot 10^{-2}$	$-2.22 \cdot 10^{-2}$	$2.21 \cdot 10^{-2}$	358
4	$4.949 \cdot 10^{-3}$	$-1.01 \cdot 10^{-2}$	$-2.74 \cdot 10^{-4}$	$1.01 \cdot 10^{-2}$	2
5	$8.760 \cdot 10^{-3}$	$5.71 \cdot 10^{-3}$	$-5.71 \cdot 10^{-3}$	$5.71 \cdot 10^{-3}$	<1
6	$1.369 \cdot 10^{-2}$	$-3.66 \cdot 10^{-3}$	$6.00 \cdot 10^{-4}$	$3.67 \cdot 10^{-3}$	<1
7	$1.965 \cdot 10^{-2}$	$2.55 \cdot 10^{-3}$	$-2.55 \cdot 10^{-3}$	$2.55 \cdot 10^{-3}$	<1
8	$2.672 \cdot 10^{-2}$	$-1.88 \cdot 10^{-3}$	$-1.94 \cdot 10^{-5}$	$1.88 \cdot 10^{-3}$	<1
9	$3.492 \cdot 10^{-2}$	$1.44 \cdot 10^{-3}$	$1.44 \cdot 10^{-3}$	$1.44 \cdot 10^{-3}$	<1

Table 5.4b Summation Terms in Series Solution and Resulting Time for Terms to Vanish (Properties Evaluated at -12°C).

ℓ	λ_ℓ^e (sec^{-1})	$R_{\ell,0}$ (°C)	$R_{\ell,Lx/2}$ (°C)	$R_{\ell,Lx}$ (°C)	t_ℓ^∞ (sec)
1	$1.435 \cdot 10^{-5}$	1.07	1.01	0.85	325,782
2	$3.563 \cdot 10^{-4}$	$-8.97 \cdot 10^{-2}$	$6.95 \cdot 10^{-3}$	$8.86 \cdot 10^{-3}$	6,158
3	$1.328 \cdot 10^{-3}$	$2.51 \cdot 10^{-2}$	$-2.51 \cdot 10^{-2}$	$2.50 \cdot 10^{-2}$	698
4	$2.946 \cdot 10^{-3}$	$-1.14 \cdot 10^{-2}$	$-3.10 \cdot 10^{-4}$	$1.14 \cdot 10^{-2}$	45
5	$5.210 \cdot 10^{-3}$	$6.47 \cdot 10^{-3}$	$6.47 \cdot 10^{-3}$	$6.47 \cdot 10^{-3}$	<1
6	$8.124 \cdot 10^{-3}$	$-4.16 \cdot 10^{-3}$	$6.80 \cdot 10^{-5}$	$4.16 \cdot 10^{-3}$	<1
7	$1.168 \cdot 10^{-2}$	$2.89 \cdot 10^{-3}$	$-2.89 \cdot 10^{-3}$	$2.89 \cdot 10^{-3}$	<1
8	$1.589 \cdot 10^{-2}$	$-2.13 \cdot 10^{-3}$	$-2.50 \cdot 10^{-5}$	$2.13 \cdot 10^{-3}$	<1
9	$2.074 \cdot 10^{-2}$	$1.63 \cdot 10^{-3}$	$1.61 \cdot 10^{-3}$	$1.61 \cdot 10^{-3}$	<1

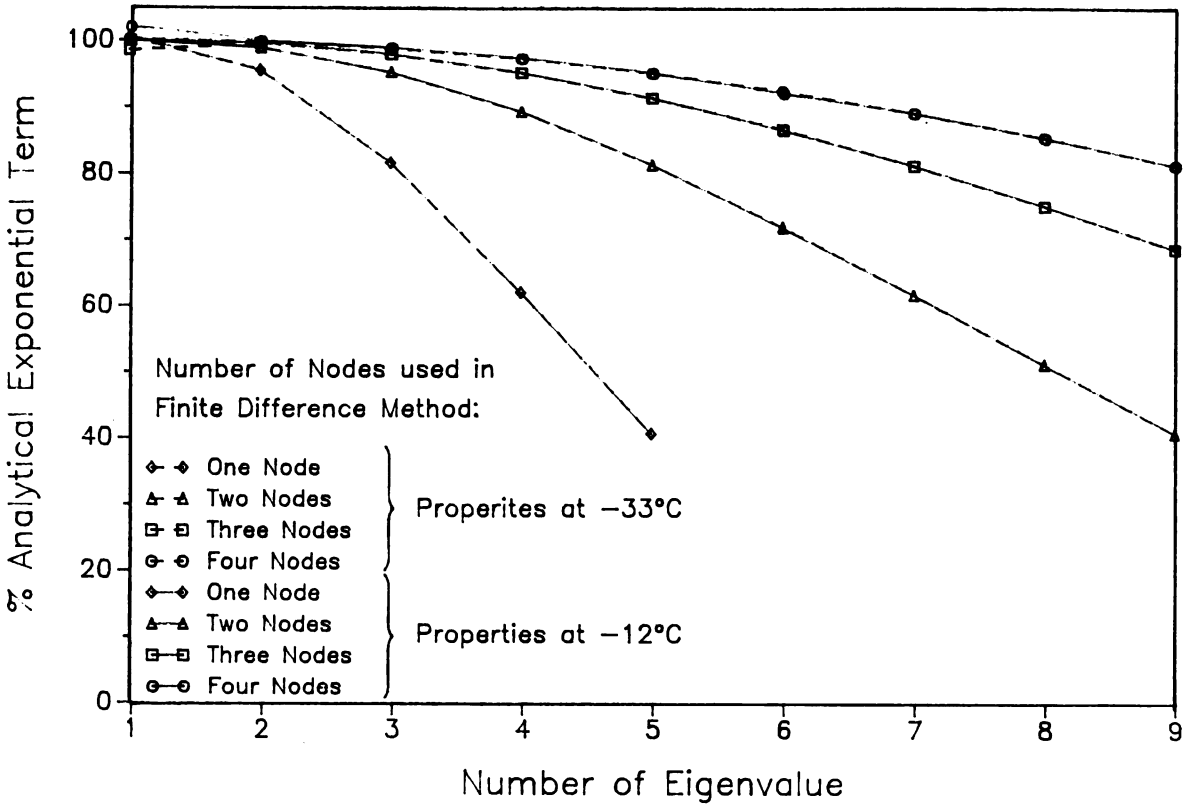


Figure 5.1 Numerical Eigenvalues as Percentages of Analytical Exponential Terms.

$$\Delta t_{ac} = \frac{t_{ss}}{100} \tag{5.15}$$

where,

$$t_{ss} = 5/\lambda_1$$

The values for the time step based on accuracy are shown in Table 5.5. From comparison of these values (t_{ss}) with the times steps resulting from the oscillation criteria (Tables 5.2a,b), the oscillation criteria was far more limiting than the accuracy criteria, in this particular problem.

5.1.3 Selection of the Optimal Time Step and Spatial Step

To ensure an efficient and accurate solution without oscillations, maximum spatial and time steps were chosen to satisfy both the oscillation and accuracy criteria discussed in the two previous sections. From the analysis of accuracy, the maximum spatial step was found using a minimum of nine nodes, and the limiting time step, based on the oscillation criteria for nine nodes, was 141 seconds, using thermal properties evaluated at -33°C, and 238 seconds, using thermal properties evaluated at -12°C. A time step of 120 seconds, which satisfies the oscillation criteria for both sets of properties was used.

Segerlind (1986) has suggested a procedure for analyzing the long time solution using the values of Δx and Δt chosen above. In this analysis, the first exponential term of the analytical solution after N ($t_{max}/\Delta t$) time steps with the numerical approximation. In this case the maximum time is 18 hours or 64,800 seconds, which is greater than t_2^{∞} using thermal properties evaluated at both -33 and -12°C; therefore,

Table 5.5 Limiting Time Step Based on Accuracy.

	Temperature (No. of Nodes)							
	-33°C				-12°C			
	(5)	(9)	(13)	(17)	(5)	(9)	(13)	(17)
Δt_{ac} (sec)	2288	2300	2301	2249	3471	3475	3535	3476

only one eigenvalue of the numerical solution is significant. The numerical approximation to $\exp\left[-\lambda_1^e \cdot t \right]$ was found using the Padre expansion of the Crank Nicolson finite difference method with N time steps.

The Padre expansion for the first exponential term was found by considering both the analytical and numerical solutions to the time dependent differential equation is the separation of variables solution of the linear heat conduction problem, Ozisik (1980)

$$\frac{\partial \Phi}{\partial t} + \lambda_1^e \cdot \Phi = 0 \tag{5.16}$$

The analytical solution is

$$\Phi = a \cdot e^{-\lambda_1^e t} \tag{5.17}$$

where,

a = constant

From Figure 5.2, the Crank-Nicolson approximations to the time derivative, and Φ are determined as

$$\frac{\partial \Phi}{\partial t} = \frac{\Phi^1 - \Phi^0}{\Delta t} \tag{5.18}$$

$$\Phi = \frac{\Phi^1 + \Phi^0}{2} \tag{5.19}$$

Substituting Eqs. (5.18), and (5.19) into Eq. (5.16) and replacing λ_1^e by the Crank-Nicolson approximation (λ_1), and solving for Φ_1 yields

$$\Phi^1 = \frac{(1 - \lambda_1 \Delta t/2)}{(1 + \lambda_1 \Delta t/2)} \cdot \Phi^0 \tag{5.20}$$

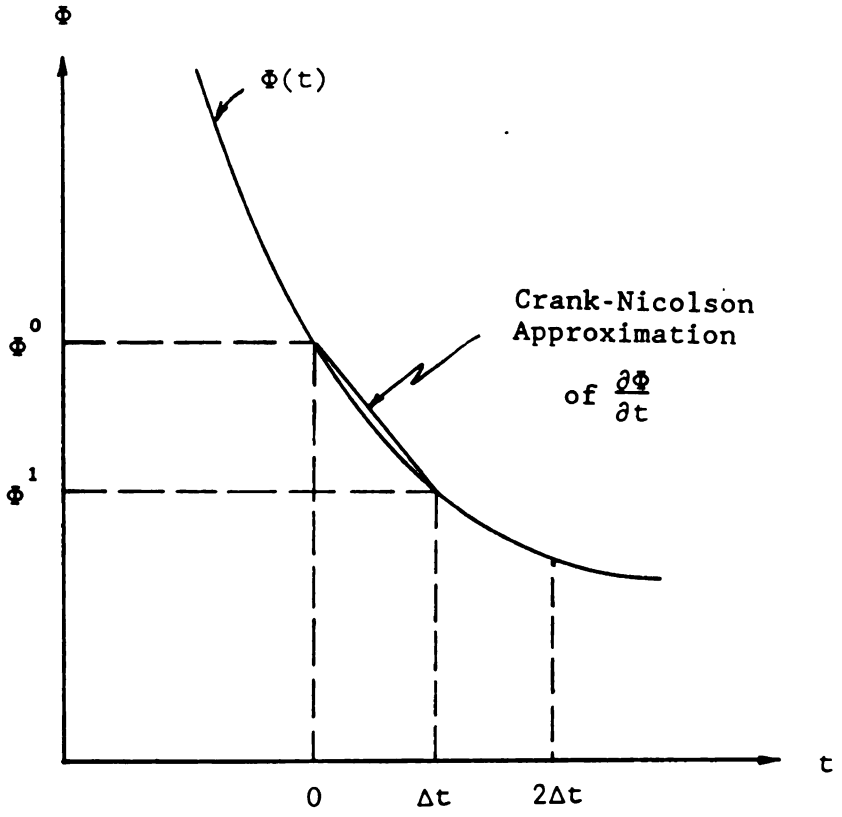


Figure 5.2 Crank-Nicolson Approximation for the Time Derivative.

After n time steps, the Crank-Nicolson approximation is

$$\Phi^n = \left[\frac{(1 - \lambda_1 \Delta t/2)}{(1 + \lambda_1 \Delta t/2)} \right]^n \cdot \Phi^0 \qquad (5.21)$$

Evaluating Eq. (5.17), at t = 0, gives $\Phi = a$, and therefore, $\Phi^0 = a$, and the Crank-Nicolson approximation to the exponential term is

$$e^{-\lambda_1^e t} \approx \left[\frac{(1 - \lambda_1 \Delta t/2)}{(1 + \lambda_1 \Delta t/2)} \right]^n \qquad (5.22)$$

where $t = n \cdot \Delta t$.

The exponential term of the analytical solution after $t_1 = 64{,}800$ seconds, and the approximate numerical term after n = 540 time steps ($\Delta t = 120$ seconds) for properties evaluated at -33 and -12°C are

1. Thermal properties at -33°C:

$$\left.\begin{aligned} e^{-\lambda_1^e \cdot t_1} &= 0.2446 \\ \frac{1 - \lambda_1 \cdot \Delta t/2}{1 + \lambda_1 \cdot \Delta t/2} &= 0.2444 \end{aligned}\right\} \text{ 99.9\% Accuracy}$$

where: $\lambda_1^e = 2.173 \cdot 10^{-5}$

$\lambda_1 = 2.174 \cdot 10^{-5}$

2. Thermal properties at -12°C:

$$\left.\begin{aligned} e^{-\lambda_1^e \cdot t_1} &= 0.3946 \\ \frac{1 - \lambda_1 \cdot \Delta t/2}{1 + \lambda_1 \cdot \Delta t/2} &= 0.3936 \end{aligned}\right\} \text{99.7\% Accuracy}$$

where: $\lambda_1^e = 2.173 \cdot 10^{-5}$

$\lambda_1 = 2.174 \cdot 10^{-5}$

5.1.4 Analysis of Time and Spatial Steps for Other Geometries

Cylindrical and spherical geometries were considered using nine nodes and compared with results from the previous sections. The eigenvalues from the numerical solution were found by modifying the conductivity and heat capacity matrices, Eqs. (5.3a,b), to account for the different geometries, and solving Eq. (5.4) as described in Section 5.1.1. The limiting time steps for the numerical oscillation criteria and for the accuracy criteria proposed by Segerlind (1986) were found using Eq. (5.5) and Eq. (5.15), respectively, for the thermal properties evaluated at -33 and -12°C.

Results are shown in Table 5.6 for both cylindrical and spherical geometries using nine nodes. Again, the oscillatory criteria was more restrictive on the time step than the accuracy criteria of Eq. (5.15). Comparing Table 5.6 with Table 5.2 a,b revealed that Δt_{osc} for the sphere was significantly less than that for the cylinder, and that the Δt_{osc} for the infinite slab was least restrictive of all. For the thermal properties evaluated at both -30 and -12°C, the critical time steps for no oscillations using spherical and cylindrical geometries were 63 and 83%, respectively, of those for an infinite slab.

Table 5.6 Limiting Time Steps for Cylindrical and Spherical Geometries.

	Cylindrical Geometry		Spherical Geometry	
	-33°C	-12°C	-33°C	-12°C
Δt_{osc} (sec)	117	197	89	150
Δt_{ac} (sec)	1072	1674	706	1089

The eigenvalues of the numerical solution were compared with the exponential terms (λ_ℓ^e) of the analytical solution; the analytical solutions for an infinite solid cylinder and a solid sphere with a convective boundary condition at r = 0.13 m are given below (Carslaw and Jaeger, 1959).

1. For an infinite solid cylinder

$$T(r,t) = T_0 + (T_{\infty,Lx} - T_0)\cdot\left[1 - \sum C_\ell \cdot \exp(-\lambda_\ell \cdot t^e)\right] \qquad (5.23a)$$

where

$$C_\ell = \frac{2(hk)\cdot J_0(\zeta_\ell r)}{Lx\cdot(\zeta^2 + hk^2)\cdot J_0(\zeta_\ell Lx)}$$

$$\lambda_\ell^e = (k/\rho Cp)\cdot\zeta_\ell^2$$

$$hk = hx_{Lx}/k$$

and, ζ_ℓ is the ℓ^{th} root of the transcendental equation

$$\zeta_\ell \cdot J_0(\zeta_\ell \cdot Lx) = (hk)\cdot J_0(\zeta_\ell Lx) \qquad (5.23b)$$

2. For a solid sphere

$$T(r,t) = T_0 + (T_{\infty,Lx} - T_0)\cdot\left[1 - \sum S_\ell \cdot \exp(-\lambda_\ell \cdot t^e)\right] \qquad (5.24a)$$

where

$$S_\ell = \frac{2 \cdot Lx \cdot hk \cdot \zeta_\ell^2 + (Lx \cdot hk - 1)^2}{r \cdot \zeta_\ell^2 \left[(Lx \cdot \zeta_\ell)^2 + Lx \cdot hk \cdot (Lx \cdot hk - 1) \right]} \cdot \sin(Lx \cdot \zeta_\ell) \cdot \sin(r \cdot \zeta_\ell)$$

$$\lambda_\ell^e = (k/\rho Cp) \cdot \zeta_\ell^2$$

$$hk = hx_{Lx}/k$$

and, ζ_ℓ is the ℓ^{th} root of the transcendental equation

$$Lx \cdot \zeta_\ell \cot(\zeta_\ell \cdot Lx) = 1 - Lx \cdot hk \qquad (5.24b)$$

The time for each term in the summation to be insignificant (t_ℓ^∞) was also determined for the cylinder and the sphere, using the criteria presented in Eq. (5.14), giving

a. Cylinder:

$$\left| C_{\ell,0} \cdot e^{-(\lambda_\ell^e t_\ell^\infty)} \right| \leq 0.01 \qquad (5.25a)$$

b. Sphere:

$$\left| S_{\ell,0} \cdot e^{-(\lambda_\ell^e t_\ell^\infty)} \right| \leq 0.01 \qquad (5.25b)$$

The times for each summation term to become insignificant were calculated and compared for the solid sphere and the infinite solid cylinder. Results for the first six eigenvalues are shown in Table 5.7a,b for thermal properties evaluated at -33 and -12°C, respectively.

Table 5.7a Comparison of t_ℓ^∞ Values for Cylindrical and Spherical Geometries (Properties Evaluated at -33°C).

ℓ	Cylindrical Geometry			Spherical Geometry		
	λ_ℓ^e (sec^{-1})	$C_{\ell,0}$ (°C)	t_ℓ^∞ (sec)	λ_ℓ^e (sec^{-1})	$S_{\ell,0}$ (°C)	t_ℓ^∞ (sec)
1	$4.48 \cdot 10^{-5}$	1.106	105,066	$6.80 \cdot 10^{-5}$	1.131	69,550
2	$8.58 \cdot 10^{-4}$	0.144	3,111	$1.16 \cdot 10^{-3}$	0.202	2,582
3	$2.76 \cdot 10^{-3}$	$6.04 \cdot 10^{-2}$	651	$3.34 \cdot 10^{-3}$	0.118	738
4	$5.76 \cdot 10^{-3}$	$3.49 \cdot 10^{-2}$	217	$6.61 \cdot 10^{-3}$	$8.35 \cdot 10^{-2}$	321
5	$9.84 \cdot 10^{-3}$	$2.34 \cdot 10^{-2}$	86	$1.10 \cdot 10^{-2}$	$6.48 \cdot 10^{-2}$	170
6	$1.50 \cdot 10^{-2}$	$1.70 \cdot 10^{-2}$	35	$1.64 \cdot 10^{-2}$	$5.29 \cdot 10^{-2}$	102

Table 5.7b Comparison of t_ℓ^∞ Values for Cylindrical and Spherical Geometries (Properties Evaluated at -12°C).

	Cylindrical Geometry			Spherical Geometry		
ℓ	λ_ℓ^e (sec^{-1})	$C_{\ell,0}$ (°C)	t_ℓ^∞ (sec)	λ_ℓ^e (sec^{-1})	$S_{\ell,0}$ (°C)	t_ℓ^∞ (sec)
1	$2.98 \cdot 10^{-5}$	1.120	158,543	$4.61 \cdot 10^{-5}$	1.149	102,945
2	$5.15 \cdot 10^{-4}$	-0.162	5,410	$7.00 \cdot 10^{-4}$	-0.228	44,670
3	$1.65 \cdot 10^{-3}$	$6.83 \cdot 10^{-2}$	1,167	$2.00 \cdot 10^{-3}$	0.133	1,294
4	$3.43 \cdot 10^{-3}$	$3.96 \cdot 10^{-2}$	401	$3.95 \cdot 10^{-3}$	$-9.46 \cdot 10^{-2}$	568
5	$5.85 \cdot 10^{-3}$	$2.65 \cdot 10^{-2}$	167	$6.57 \cdot 10^{-3}$	$7.34 \cdot 10^{-2}$	304
6	$8.93 \cdot 10^{-3}$	$1.93 \cdot 10^{-2}$	74	$9.81 \cdot 10^{-3}$	$5.99 \cdot 10^{-2}$	183

The first two terms in the summation approach zero faster for the sphere compared with the cylinder, and for the cylinder compared with the infinite slab (Table 5.4a,b), as would be expected due to the increased surface area to volume ratio of the sphere and cylinder, compared with the slab. However, the time for the remaining terms in the summation to become insignificant is longer for the sphere compared with the cylinder, and longer for the cylinder compared with the slab, indicating that more eigenvalues are significant when evaluating the accuracy criteria for the sphere and cylinder. From Table 5.7a, with thermal properties evaluated at -33°C, there are four and six significant eigenvalues after 100 seconds for the cylinder and the sphere, respectively, compared with three for the infinite slab (Table 5.4a). For properties evaluated at -12°C, the number of significant terms increases to 5 terms for the cylinder, and at least six terms for the sphere (Table 5.7b), compared with three for the infinite slab (Table 5.4b).

The numerical eigenvalues, expressed as percentages of the analytical exponential factors (λ_ℓ^e) for the infinite solid cylinder and the solid sphere are shown in Figure 5.3, using nine nodes and thermal properties evaluated at -33 and -12°C. Only the first two eigenvalues for both the cylindrical and spherical cases satisfy the 5% accuracy criteria set for the infinite slab, indicating that a smaller position step (Δr) is required for the cylinder and the sphere, than for the infinite slab to satisfy the given accuracy criteria.

In summary, using observations from the infinite solid cylinder and the solid sphere: (1) the limiting time step for the oscillation criteria (Δt_{osc}) decreased as the surface area to volume of the geometry increased; (2) the accuracy of the eigenvalues expressed as $\lambda_\ell/\lambda_\ell^e \cdot 100\%$ decreased with increased surface area to volume ratio; and (3) the time required for each exponential term to be insignificant decreased with

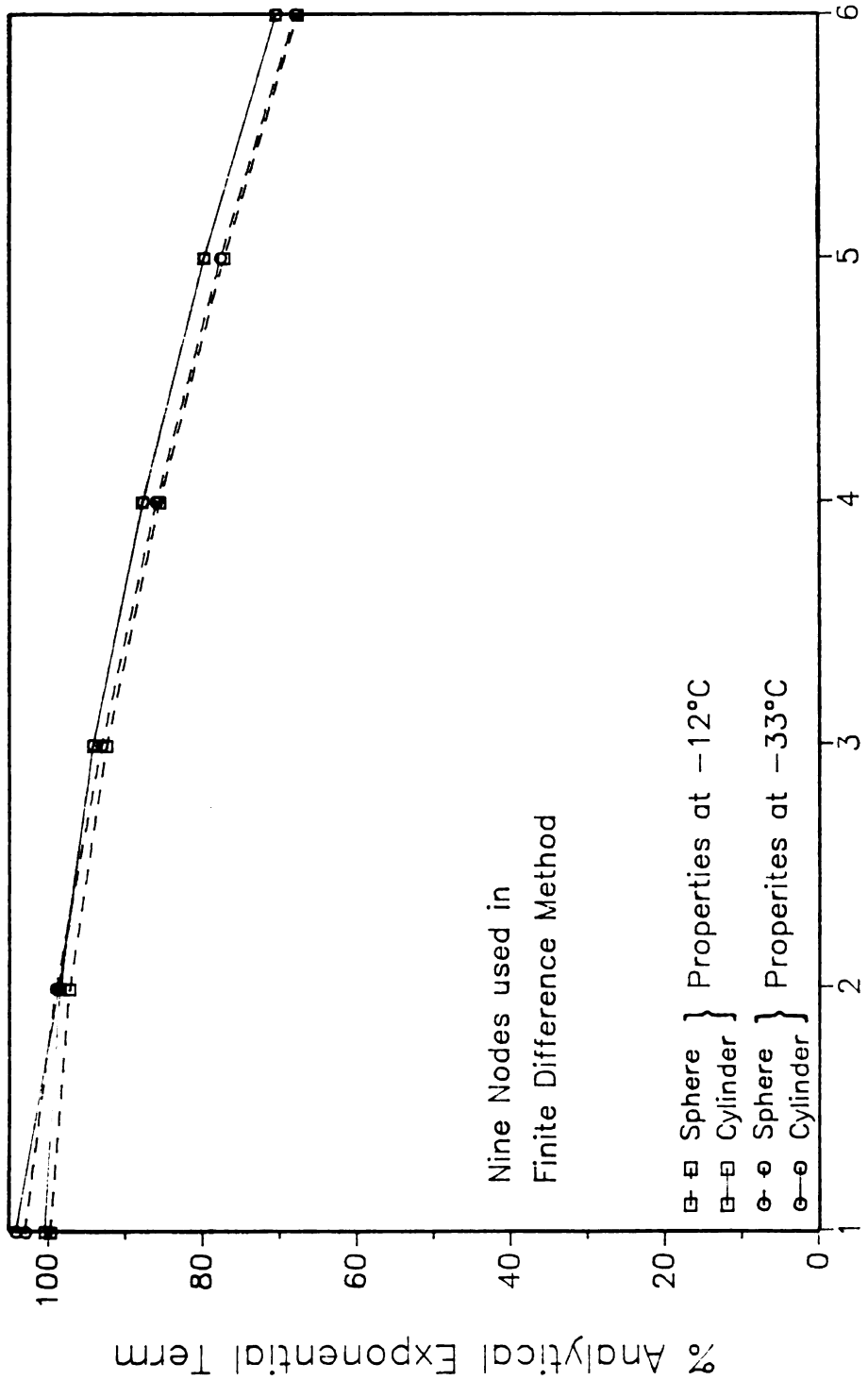


Figure 5.3 Numerical Eigenvalues as Percentages of Analytical Exponential Terms for Cylindrical and Spherical Geometries.

increased surface area to volume ratio for the first two terms, but increased for the remaining terms.

5.1.5 Analysis of Time and Spatial Steps for the Two Dimensional Model

The analysis of the oscillation criteria and accuracy of the one dimensional problem required the determination of the numerical eigenvalues as shown in Eq. (5.4) for an L by M matrix (L, M = number of nodes in x and y directions). The matrices associated with the two dimensional problem are considerably larger: for equal number of nodes in two dimensions, the coefficient matrices will be L^2 by M^2, for which it is generally impractical to solve for the eigenvalues, except for small values of L and M. Therefore, a simplified analysis was sought which would give insight into oscillation criteria and the accuracy of the solution.

In the previous discussion of the oscillation criteria, it was found that the maximum eigenvalue of the elemental matrices ($\lambda_{max}^{(e)}$) provided an excellent estimate of λ_{max} of the global matrix. For the two dimensional element, the elemental matrices are shown below (Belytschko and Hughes, 1983)

$$\mathbf{k}^{(e)} = \begin{bmatrix} a+b & -b & 0 & -a \\ -a & a+b & -b & 0 \\ 0 & -a & a+b & -b \\ -b & 0 & -a & a+b \end{bmatrix} \qquad (5.26a)$$

where $a = \frac{k\Delta y}{\Delta x}$, and $b = \frac{k\Delta x}{\Delta y}$

$$c^{(e)} = \frac{\rho \cdot Cp \cdot (\Delta x \cdot \Delta y) \cdot I}{4} \tag{5.26b}$$

where I is the identity matrix. Assuming $\Delta x = \Delta y$, and substituting these matrices in to Eq. (5.10), resulted in the following expression for $\lambda_{max}^{(e)}$.

$$\lambda_{max}^{(e)} = \frac{16 \cdot k}{\rho \cdot Cp(\Delta x)^2} \tag{5.27}$$

This expression for $\lambda_{max}^{(e)}$ is four times the value of $\lambda_{max}^{(e)}$ for the one dimensional element, Eq. (5.11); therefore, the no oscillation time step is four times as restrictive for the two dimensional case, as for the one dimensional case with equal position increments.

The values for λ_{max} calculated from the global matrix are compared with $\lambda_{max}^{(e)}$ for both a 3x3 and a 4x4 two dimensional grid with $\Delta x = \Delta y$, using thermal properties evaluated at -33°C. Again, results indicate that for a finite difference grid, $\lambda_{max}^{(e)}$ provides a good estimation of λ_{max} within 5%, as shown in Table 5.8.

5.1.6 Summary of Observations in the Determination of Finite Difference Parameters

Two criteria were used in the selection of the time step and position increment used in the finite difference solution. The position increment (Δx) was selected to satisfy the accuracy criteria, and the time step was determined according to the oscillation criteria. The oscillation and accuracy analysis is summarized below.

1. The no oscillation time step is determined using the largest

Table 5.8 Comparison of $\lambda_{max}^{(e)}$ and λ_{max} for a Two Dimensional Grid ($\Delta x = \Delta y$, Properties Evaluated at -33°C).

	Grid Size (position increment)	
	3x3 (Δx = 0.065m)	4x4 (Δx = 0.0433m)
λ_{max}	$1.822 \cdot 10^{-3}$	$3.873 \cdot 10^{-3}$
$\lambda_{max}^{(e)}$	$1.765 \cdot 10^{-3}$	$3.972 \cdot 10^{-3}$
(%)	96.9%	102.5%

eigenvalue (λ_{max}) of the set of finite difference equations.

2. The elemental eigenvalue ($\lambda_{max}^{(e)}$) is approximately the value of λ_{max}, within 99% accuracy for the one dimensional case and 95% accuracy for the two dimensional case.
3. The no oscillation time step decreases as the surface area to volume ratio of the geometric shape increases:

$$\Delta t_{osc}(\text{slab}) > \Delta t_{osc}(\text{cylinder}) > \Delta t_{osc}(\text{sphere}).$$

4. The no oscillation time step of a two dimensional grid with equal position increments in the x and y directions is half the value for the one dimensional grid, using the same position increment.
5. The oscillation time step at one temperature, given Δt_{osc} at a second temperature, is proportional to the inverse ratio of their respective thermal diffusivities.
6. The position step was determined according to the accuracy of the significant eigenvalues in the series solution.
7. The time for each term in the series solution to be insignificant increased with decreasing thermal diffusivity.
8. The number of significant eigenvalues increased with increasing surface to volume ratio.
9. The limiting time step for accuracy, proposed as 1% of the total time to steady state conditions, was not as restrictive as the time step for the no oscillation criteria.

5.2 Parameters Used in the Solution of the Inverse Heat Conduction Problem

The sequential regularization method, using finite differences, was used to estimate surface heat transfer coefficients. This method requires input values for: (1) the regularization parameter, α; (2) the

order of regularization, W_0, W_1, or W_2; (3) the number of future time steps, r, used the sequential procedure; (4) and the time increment between temperature measurements, Δt_m, in addition to the user adjustable parameters inherent in the finite difference technique discussed in Section 5.1. The values selected for the spatial and times steps (Δx and Δt) in the one dimensional direct problem were also used in the one dimensional indirect problem. The time increment between temperature measurements, Δt_m, was limited by the data aquisition unit used in the experimental procedures, to a minimum value of 600 seconds.

Two criteria, the deterministic bias and the variance, were used in determining optimal values for the parameters inherent to the inverse problem. The deterministic bias is defined as a measure of the bias or error in the estimator when input temperature measurement errors are equal to zero. The variance is a measure of the estimators sensitivity to random measurement errors (Beck et. al., 1985). The deterministic bias and variance of the estimated heat flux for the n^{th} time step, D_e^n and V^n, are defined by Scott and Beck (1985) as

$$D_e^n = E(\hat{q}_D^n) - q^n \tag{5.28a}$$

$$V^n = E\{[\hat{q}_V^n - E(\hat{q}_V^n)]^2\} \tag{5.28b}$$

where q^n is the true heat flux, $E(\hat{q}_D^n)$ is the expected value of the estimated heat flux with no errors in the temperature input values, and $E(\hat{q}_V^n)$ is the expected value of $\hat{q}^n$ with random temperature measurement errors.

It was desired to find parameter values which minimized both types of errors. The mean square error, S, was defined as the sum of the variance and the square of the deterministic bias

$$(S^N)^2 = V^N + (D_e^N)^2 \tag{5.29}$$

where N is the total number of time steps.

The problem used to evaluate the spatial and time steps (Equations 5.6a-d), assuming constant thermal properties and a surface heat transfer coefficient of 7.85 W/m°C, was also used in the determining of the sequential regularization parameters, with the exception that, in this case, two 18 hour storage periods were used. The first storage period was -30°C, and the ambient temperature for the second storage period was -5°C. The initial temperature was set at -30°C. The analytical solution of this problem was solved to obtain temperature values at x = Lx/2, corresponding to the location of the thermocouple in the experimental procedures for the single layer slab described in Section 4.3. These temperature values were used as input data for the sequential regularization solution of the IHCP.

The surface heat flux was first estimated using exact temperatures from the analytical solution as input, and the deterministic error was calculated for various values of α and r, and for the zeroth, first and second regularization order. The process was repeated using imposed random measurement errors in the temperature input values to estimate the mean squared error.

5.2.1 The Deterministic Bias

The deterministic bias was estimated by Scott and Beck (1985)

$$D_e = \left[\sum_{n=1}^{N} (\hat{hx}_{Lx}^n - hx_{Lx})^2 \right]^{0.5} \tag{5.30}$$

where $\hat{hx}_{Lx}^{n}$ is the estimated heat flux at the n^{th} time step, and hx_{Lx} is the constant heat flux value used in generating the input temperature values for the IHCP.

The regularization parameter (α) regulates the degree the regularization terms, shown in Equation (3.34a-c), influence the solution. The number of future time steps, r, influences the stability of the solution and computation time. As r becomes large, the solution approaches the whole domain solution, and computation time increases significantly (Beck, et. al., 1985). To determine the critical values for both of these parameters on the solution, the surface heat flux values were estimated using various values of α and r, for the zeroth, first and second order regularization orders, with the time step for temperature measurements, Δt_m, equal to 600 seconds. Thermal properties were evaluated at -33°C, since from Tables 5.2a,b, the time step for no oscillations was most restrictive at that temperature.

The range of values for r were conservatively chosen from two to fifteen, based on observations by Scott and Beck (1985), who noted that the sequential regularization solution of the IHCP is independent of r for $r \geq 8$. In determining the range of values considered for the regularization parameter, α, it was noted that the regularization term is added to the sensitivity coefficient matrix product, $\mathbf{X}^T\mathbf{X}$, in the regularization method. Therefore, the magnitude of the coefficients in the $\mathbf{X}^T\mathbf{X}$ matrix, with α equal zero and r equal fifteen, were used to determine the range of values for α. The magnitudes of the $\mathbf{X}^T\mathbf{X}$ matrix product ranged from $\simeq 10^{-9}$ to $\simeq 10^{-3}$. The regularization parameter, α, is multiplied by $\mathbf{H}_i^T\mathbf{H}_i$, i = 0, 1, or 2, which, from Equations (3.36a-c) are of order one; therefore, values of α ranging from 10^{-9} to 10^{-3} were used to determine its influence on the solution.

The deterministic error was calculated and compared for discrete values of α and r. Results are shown in Tables 5.9 a-g, for r equal 2, 4, 6, 8, 10, 12, and 15; for the zeroth, first and second regularization orders; and for $\alpha = 10^{-9}$, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, and 10^{-3}, respectively. The time step between temperature measurements, Δt_m, was equal to 600 seconds and thermal properties were evaluated at -33°C in all cases.

The results for exact temperature input values support the observations by Scott and Beck, (1985), in that the deterministic bias shows little dependence on r, for r greater than eight. The critical values for the regularization parameter, α, ranged from $\approx 10^{-4}$ to $\approx 10^{-7}$, for all regularization orders. For $\alpha \geq 10^{-5}$, the deterministic bias, D_e, increased with increasing order of regularization, and for $\alpha \leq 10^{-6}$, D_e decreased with increasing α.

5.2.2 Mean Squared Error

The hypothetical problem used to determine the deterministic bias was also used to estimate the mean squared error. Input ambient and internal temperatures were modified by the addition of normally distributed random numbers.

The standard deviation used for the random numbers was determined from the error limits of the thermocouples used in the experimental procedures. The error limits of the T-type thermocouples were given as the greatest value between (Omega, 1985)

$$\pm 1\ °C \text{ or } \pm 1.5\ \% \text{ of maximum } |°C|$$

Table 5.9a. Deterministic Bias for $\alpha = 10^{-9}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

	r						
	2	4	6	8	10	12	15
W_0	------	75.44	78.26	78.24	78.24	78.24	78.24
W_1	------	54.72	69.09	68.88	68.91	68.91	68.91
W_2	------	102.44	28.16	28.78	28.99	27.62	28.14

Table 5.9b. Deterministic Bias for $\alpha = 10^{-8}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

	r						
	2	4	6	8	10	12	15
W_0	22.02	55.73	54.86	54.88	54.89	54.89	54.89
W_1	------	45.13	37.48	37.52	37.55	37.55	37.55
W_2	------	22.21	23.78	26.22	26.66	26.70	26.71

Table 5.9c. Deterministic Bias for $\alpha = 10^{-7}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

	r = 2	4	6	8	10	12	15
W_0	25.02	24.98	24.96	24.94	24.93	24.93	24.93
W_1	29.40	13.57	18.06	19.16	19.29	19.29	19.29
W_2	------	18.30	15.57	15.25	15.49	15.64	15.67

Table 5.9d. Deterministic Bias for $\alpha = 10^{-6}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

	r = 2	4	6	8	10	12	15
W_0	12.33	11.77	14.78	14.81	14.84	14.84	14.84
W_1	24.12	9.92	12.36	12.63	13.13	13.13	13.14
W_2	------	7.90	9.41	11.82	11.77	11.90	11.96

Table 5.9e. Deterministic Bias for $\alpha = 10^{-5}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

	r = 2	4	6	8	10	12	15
W_0	16.88	12.37	13.17	14.06	14.09	14.42	14.43
W_1	81.51	10.39	13.42	13.49	13.88	13.90	13.95
W_2	------	31.64	14.53	15.66	15.62	15.66	15.71

Table 5.9f. Deterministic Bias for $\alpha = 10^{-4}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

	r = 2	4	6	8	10	12	15
W_0	64.51	22.52	19.05	20.05	20.05	19.93	20.02
W_1	129.51	35.43	21.46	24.05	23.01	22.67	22.83
W_2	------	102.44	28.16	28.78	28.99	27.62	28.14

Table 5.9g. Deterministic Bias for $\alpha = 10^{-3}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

	r						
	2	4	6	8	10	12	15
W_0	125.83	113.61	51.72	39.38	36.78	37.07	37.64
W_1	137.32	107.44	56.38	42.82	46.31	47.29	44.81
W_2	------	133.23	144.48	51.26	55.81	58.32	53.87

The experimental data ranged from ≃ -33°C to -5°C, therefore, the first criteria (± 1°C) was the greatest. The standard deviation was calculated assuming that ± 1 °C represented the 99.5% confidence interval of the thermocouple temperature measurements. The standard deviation was calculated from the $t_{\alpha/2,\nu}$ probability distribution, with α = 99.5%, and the number of degrees of freedom, ν, equal to five, corresponding to the six thermocouples used in the experimental procedures. Therefore, the standard deviation was estimated from the confidence interval (CI), where (Walpole and Myers, 1978)

$$CI = \pm 1°C = \pm \frac{t_{\alpha/2,\nu} \cdot \sigma}{(N_t)^{.5}} \tag{5.31}$$

where the number of thermocouples, N_t, equaled six, and $t_{\alpha/2,\nu}$ = 3.365. From Equation (5.31), σ was estimated to be 0.73°C. This was considered to be a conservative estimate, since the largest variation between thermocouples located within the Karlsruhe test substance from the experimental results was only 0.7°C.

Beck, et. al. (1985) recommended using a single temperature measurement error as an estimation of the variance. An alternate approach, used by Scott and Beck, (1985) is sometimes referred to as the Monte Carlo Method. In this case, the heat transfer coefficient is estimated from input values with added random temperature measurement values. The mean square error was estimated from the estimated heat transfer coefficient using random errors in the input temperature values as follows

$$\hat{S}_i = \left[\sum_{n=1}^{N} \left(hx_{Lx} - \hat{hx}^n_{Lx} \right)^2 \right]^{0.5} \tag{5.32}$$

Since the value of $\hat{S}_i$ depends on the set of random numbers added to the input temperatures, different sets of random numbers (i) were added to the input temperature values to calculate different values for $\hat{S}_i$. The mean squared error, $\hat{S}$, was determined from the average of the $\hat{S}_i$ values from twelve different sets of random numbers

$$\hat{S} = \sum_{n=1}^{N_r} (\hat{S}_i/i) \tag{5.33}$$

where twelve different sets of random numbers (N_r) were used.

From Tables 5.9a-g, the range of α for which the deterministic error was less 25, was selected as the range over which the mean squared error was determined. The critical values of α ranged from $\simeq 10^{-7}$ to $\simeq 10^{-4}$. Values of $\hat{S}$ were calculated for $\alpha = 10^{-4}$, 10^{-5}, and 10^{-6}, r = 4, 6, 8, 10, and 12, and for W_0, W_1, and W_2. Results for the average mean squared error $\hat{S}$, and the standard deviation of $\hat{S}$ are shown in Tables 5.10a-c, for $\alpha = 10^{-4}$, 10^{-5}, and 10^{-6}, respectively. Since the results for $\hat{S}$ using $\alpha = 10^{-6}$ were much greater the those using $\alpha = 10^{-5}$, the mean squared error using $\alpha = 10^{-7}$ was not considered.

The results indicate that $\hat{S}$ calculated using $\alpha = 10^{-4}$ provided the lowest mean squared error $\hat{S}$, and that the solution is independent of r, for $r \geq 10$. The zeroth order regularizer provided a slightly better estimator than the first or second order estimators.

5.2.3 Time Increment between Temperature Measurements

Since the time increment between temperature measurements was limited to 600 seconds by the data aquisition device used in the experimental procedures, the values of Δt_m were limited to 600 seconds or

Table 5.10a. Average Mean Squared Error ($\hat{S}$) and Standard Deviation of $\hat{S}$ (σ_S) for $\alpha = 10^{-6}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

		r = 4	6	8	10	12
W_0	$\hat{S}$	231.08	205.57	200.42	197.60	194.18
	σ_S	38.06	45.12	43.45	43.77	44.18
W_1	$\hat{S}$	167.32	158.89	131.57	125.26	123.85
	σ_S	17.53	18.54	15.44	14.94	14.35
W_2	$\hat{S}$	254.36	100.32	108.39	106.21	99.74
	σ_S	23.72	6.52	7.94	8.38	8.67

Table 5.10b. Average Mean Squared Error ($\hat{S}$) and Standard Deviation of $\hat{S}$ (σ_S) for $\alpha = 10^{-5}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

		r = 4	r = 6	r = 8	r = 10	r = 12
W_0	$\hat{S}$	92.52	72.87	72.32	64.70	62.92
	σ_S	7.51	5.96	6.94	8.85	8.70
W_1	$\hat{S}$	100.91	49.51	55.08	48.02	48.10
	σ_S	7.61	4.66	4.19	4.09	4.47
W_2	$\hat{S}$	44.20	55.86	44.94	42.15	41.41
	σ_S	2.36	5.17	3.97	4.36	4.15

Table 5.10c. Average Mean Squared Error ($\hat{S}$) and Standard Deviation of $\hat{S}$ (σ_S) for $\alpha = 10^{-4}$ (Δt_m = 600 seconds; thermal properties evaluated at -33°C).

		r = 4	6	8	10	12
W_0	$\hat{S}$	36.37	30.89	27.25	26.99	26.90
	σ_S	1.77	1.88	1.56	1.58	1.60
W_1	$\hat{S}$	40.34	32.84	28.84	29.19	28.34
	σ_S	1.09	1.70	1.22	1.33	1.31
W_2	$\hat{S}$	102.50	34.72	31.72	31.46	30.22
	σ_S	0.32	0.93	0.58	0.52	0.48

a multiple of 600 seconds. To determine the influence of Δt_m on the mean squared error, the solution was determined for the zeroth, first and second regularization orders using a Δt_m of 1200 seconds, with ten future time steps and $\alpha = 10^{-4}$. Random errors were again added to the input temperature values, and the average mean squared error was calculated from Equation (5.33) using twelve sets of random numbers ($N_r = 12$). Since the mean squared error is dependent on the total number of time steps, N, as shown in Equation (5.32), to compare these results with those found using $\Delta t_m = 600$ seconds, the resulting average mean squared values were divided by the total number of time steps used in each case to obtain an average mean squared error per time step, $\hat{S}^*$. These values for $\hat{S}^*$ are shown in Table 5.11, along with the respective standard deviations, σ_S^*, (σ_S divided by the total number of time steps), for $\Delta t_m = 600$ seconds and $\Delta t_m = 1200$ seconds, with $\alpha = 10^{-4}$ and ten future time steps. The results show that the average mean squared error per time step is significantly higher using $\Delta t_m = 1200$ seconds than that found using $\Delta t_m = 600$ seconds, for all regularization orders.

5.2.4 Selection of Optimal Parameters used in the Inverse Heat Conduction Problem of Estimating the Surface Transfer Coefficient

The values for the regularization parameter, α, the number of future time steps, r, the order of regularization, W_i, i = 0, 1, or 2, and the time increment between temperature increments, Δt_m, were selected to minimize the average mean squared error ($\hat{S}$) in the solution. From the results shown in Tables 5.9a-g, 5.10a-c, and 5.11, the following values were selected for the parameters inherent in the sequential regularization IHCP solution:

$$\alpha = 10^{-4},$$

Table 5.11. Average Mean Squared Error per Time Step ($\hat{S}^*$) and Standard Deviation (σ_S^*) for Δt_m = 600 and 1200 seconds ($\alpha = 10^{-4}$, $r = 10$).

		W_0	W_1	W_2
Δt_m = 600	$\hat{S}^*$	0.129	0.139	0.150
	σ_S^*	0.008	0.006	0.002
Δt_m = 1200	$\hat{S}^*$	0.188	0.203	0.238
	σ_S^*	0.012	0.014	0.007

$r = 10$,

$W_0 = 1$, $(W_1 = W_2 = 0)$, and

$\Delta t_m = 600$ seconds.

Surface heat transfer coefficients ($\hat{hx}_{Lx}$), estimated using both exact input temperatures and input temperatures with added random errors, and the above values for the input parameters, are shown as a function of time in Figure 5.4. The mean squared error for this particular set of random numbers ($\hat{S}_i$) was 23.7 $W/m^2°C$. (From Table 5.10c, $\hat{S}$ = 26.99 $W/m^2°C$).

Several observations may be noted from the results shown in Figure 5.4. (1) At time, t = 0, the simulated product was exposed suddenly to a change in ambient temperature from -30°C to -5°C. During this time, the damping effect of the sequential regularization procedure on $\hat{hx}_{Lx}$ was evident in the gradual increase of the predicted value of $\hat{hx}_{Lx}$, using exact data, to a constant value at t ≈ 3 hours. (2) At t = 18 hours, when the simulated ambient temperature changed suddenly from -5°C to -30°C, $\hat{hx}_{Lx}$ suddenly decreased, and then increased gradually to a constant value as before with t = 0. (3) The sinusiodal nature of $\hat{hx}_{Lx}$ using input temperatures with added random errors was a result of the estimators tendency to smooth out variations in the input temperature values. (4) The high variability of the estimated heat flux was a result of the large standard deviation used in the random errors added to the input temperature values.

The input temperatures (ambient temperatures and internal temperatures) using exact data and using added random errors are shown in Figure 5.5. Since the maximum variation in the input temperatures with random errors was higher than that observed from the actual temperature measurements, it is expected that the resulting variability in the estimated surface heat transfer coefficient using temperature

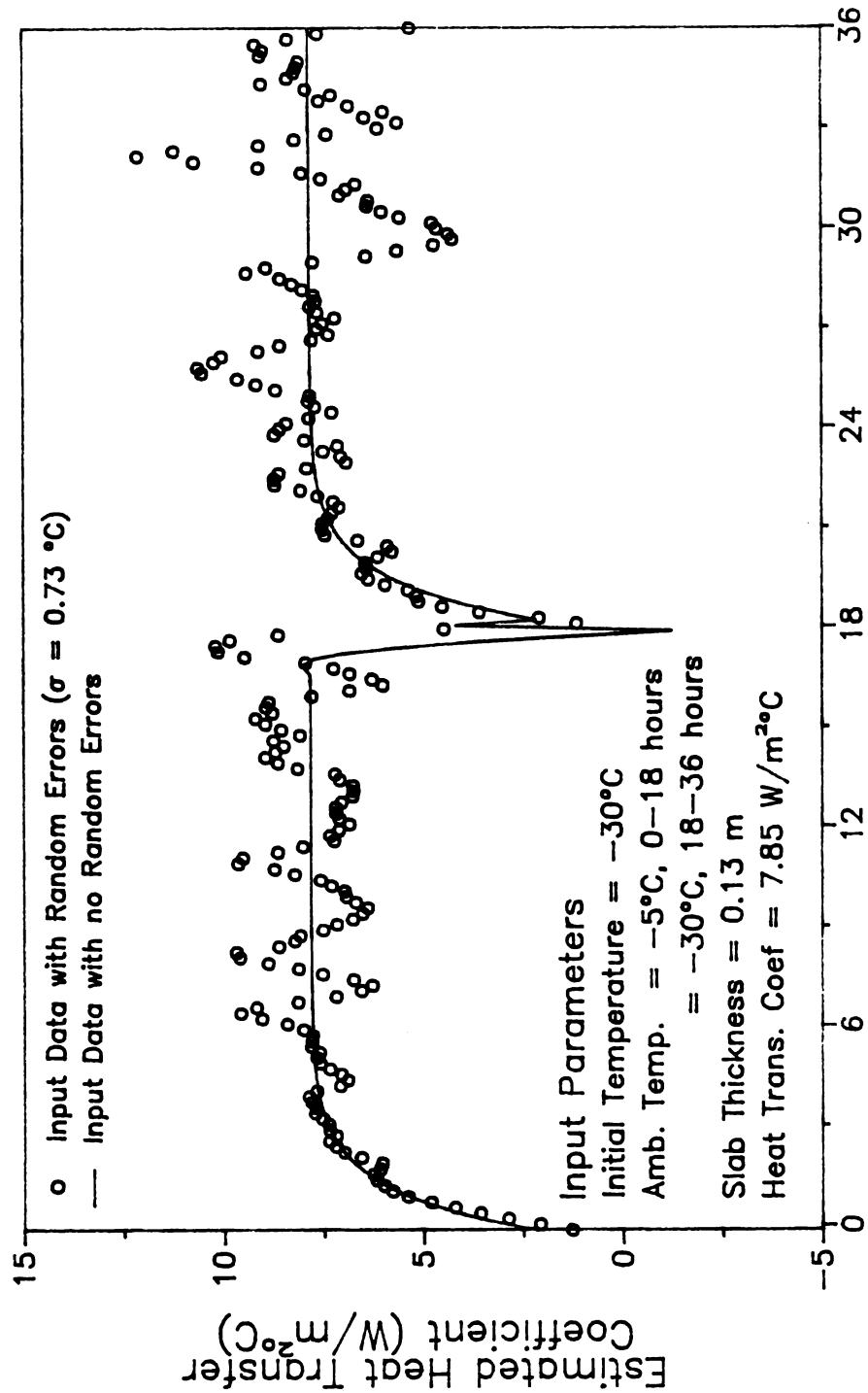


Figure 5.4 Estimated Heat Transfer Coefficients with and without Random Errors in Input Data with Standard Deviation, σ, of 0.73°C.

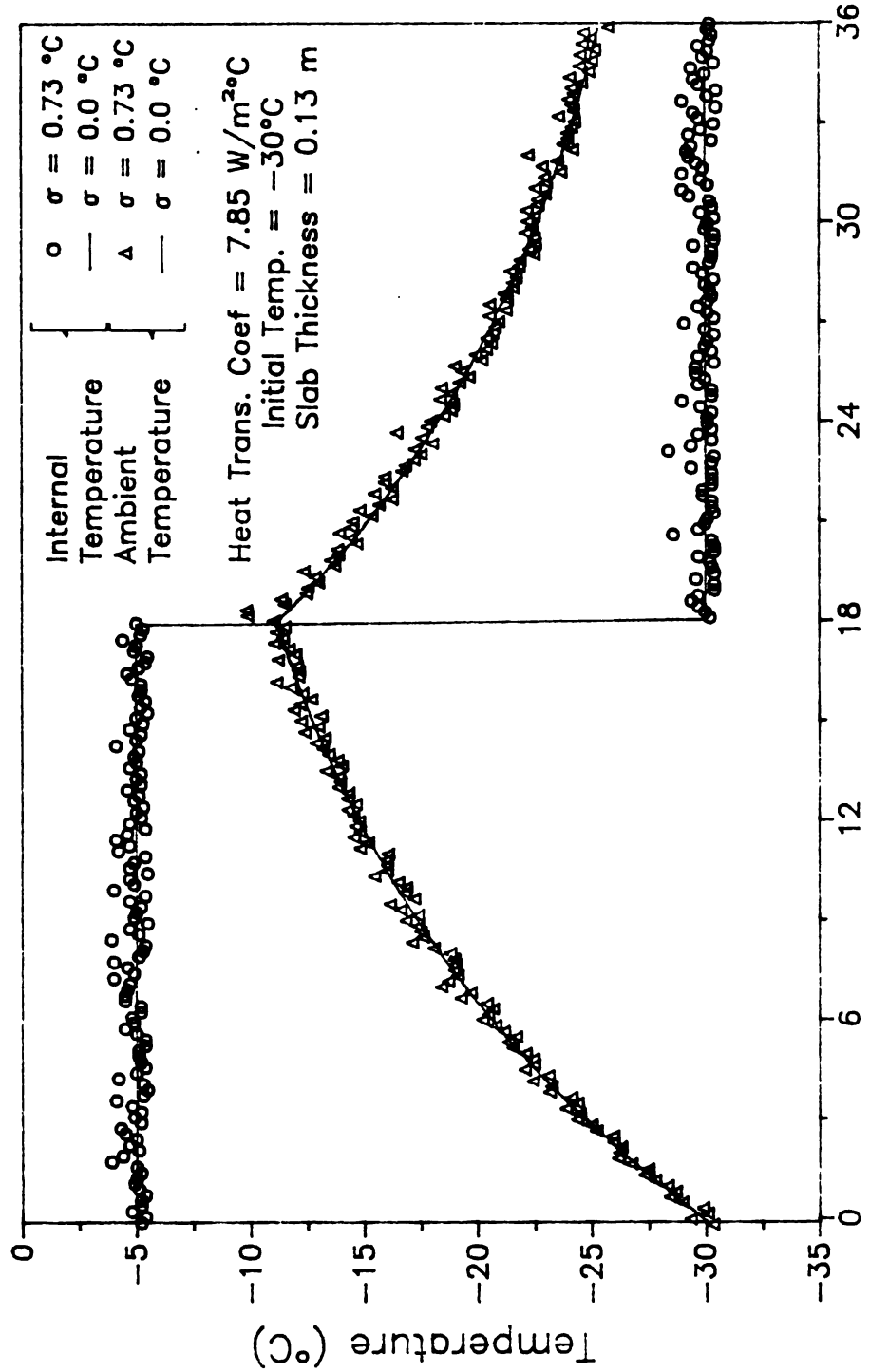


Figure 5.5 Ambient and Internal Temperature Input Data with and without Random Errors.

measurements as input will be less than that shown in Figure 5.4. The surface heat transfer coefficient was estimated using a standard deviation one half of that used previously (σ(new) = 0.5•σ(old) = 0.5•0.73 = 0.365°C). Results are shown in Figure 5.6; the maximum variation in $\hat{hx}_{Lx}$ for this case was ≈ 2 W/m^2°C, corresponding to a maximum variation in input temperatures of ≈ 0.5°C.

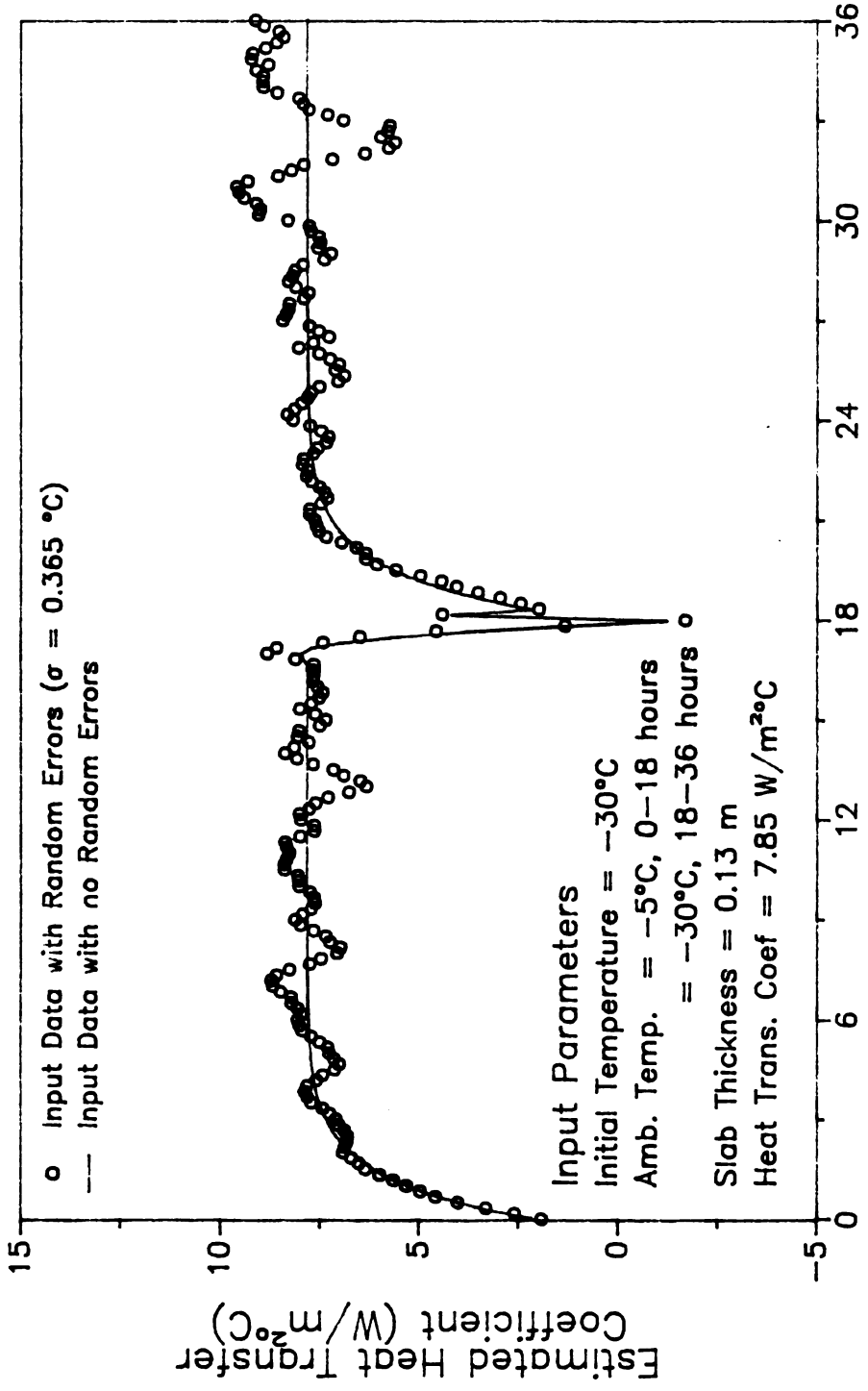


Figure 5.6 Estimated Heat Transfer Coefficients with and without Random Errors in Input Data with Standard Deviation, σ, of 0.36°C.

CHAPTER 6.

RESULTS AND DISCUSSION

Both analytical and inverse heat conduction methods were used in estimating the surface heat transfer coefficient; results are discussed in Section 6.1. The one and two dimensional direct numerical solutions were verified by comparison with analytical solutions, assuming constant product properties, and experimental results, assuming variable thermal properties. These comparisons are described in Sections 6.2 and 6.3.

Some of the parameters affecting the temperature and quality distribution histories of a simulated food product were also investigated. This study concentrated on two areas: (1) the effects of boundary conditions (Section 6.4), and (2) the effects of size and geometry (Section 6.5). The primary objectives in this analysis were to determine how the parameters associated with these areas affected the overall rate of quality deterioration, and the variation of quality deterioration within the product.

6.1 Estimation of the Surface Heat Transfer Coefficient

The surface heat transfer coefficient was estimated using both analytical and inverse heat conduction methods. Both methods required ambient and product temperatures, and in addition, the analytical solution required knowledge of the velocity profile over the surface of the

product. The ambient and average product temperature measurements, obtained in the first repetition for the single layer slab with one exposed surface (Test 1a, Section 4.5), are shown in Figure 6.1. (Measurements obtained in the second two repetitions are shown in Figures E.1a,b.) These values were used in both analytical and inverse solutions, and the velocity measurements for the analytical case were determined experimentally, as discussed in Section 4.4.

6.1.1 Analytical Estimation of the Surface Heat Transfer Coefficient

The surface heat transfer coefficient was estimated using the analytical methods described in Section 3.5.1. For both forced and free convection to be significant, Eq. (3.22b) must be satisfied. The Reynolds number in Eq. (3.22b) was determined using average velocity measurements. The average air velocities over the product during the first and second storage intervals, $\bar{U}_{1\infty}$ and $\bar{U}_{2\infty}$, from 0 to 18 hours and from 18 to 36 hours, respectively, were estimated from the velocity measurements, found using a hot wire anemometer, in each storage chamber as described in Section 4.4. The kinematic viscosity was calculated using the average ambient temperature values shown in Figure 6.1 (Incropera and Dewitt, 1985). For the first storage interval, the average ambient temperature, $\bar{T}_{1\infty}$ equaled -6°C, and for the second storage interval, the average ambient temperature, $\bar{T}_{2\infty}$ equaled -33°C. (Similar values were found for the second two test cases, shown in Figures E.1a,b.) The Reynolds number was calculated for both 18 hour storage intervals from Eq. (3.21a). Results are shown below.

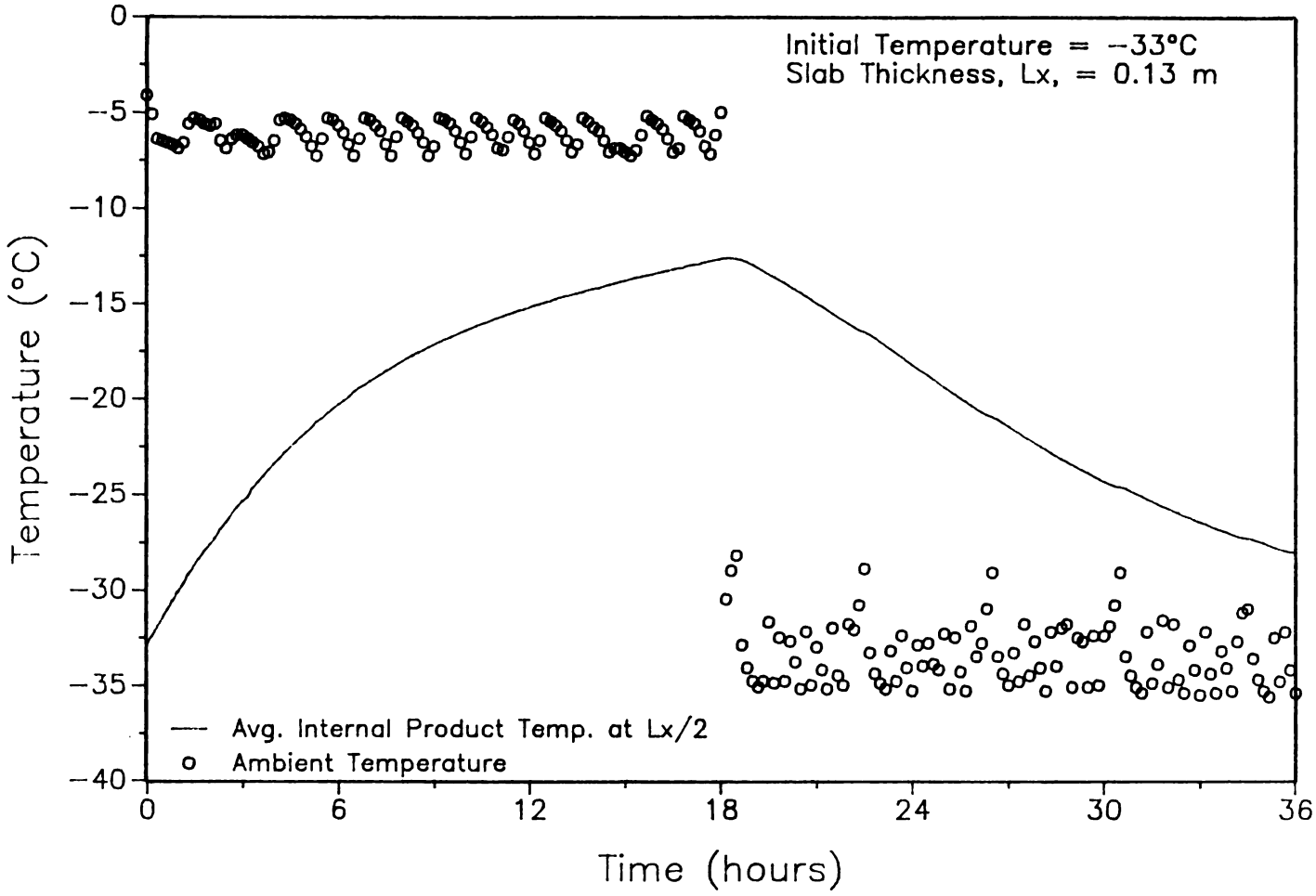


Figure 6.1 Ambient and Average Internal Temperature of Karlsruhe Test Substance Measurements using Single Layer Slab with One Exposed Surface (Test 1a).

$$Re_{Lx,1} = \frac{\bar{U}_{2\infty} \cdot Lx_c}{\nu_1} = 2.74 \cdot 10^3 \qquad 0 \geq t > 18 \text{ hours}$$

$$Re_{Lx,2} = \frac{\bar{U}_{2\infty} \cdot Lx_c}{\nu_2} = 1.10 \cdot 10^4 \qquad 18 \geq t \geq 36 \text{ hours}$$

where: $Lx_c = 0.118$ m

$\bar{U}_{1\infty} = 0.25$ m/s $\qquad \bar{U}_{2\infty} = 1.0$ m/s

$\nu_1 = 1.296 \cdot 10^{-5}$ m^2/s $\qquad \nu_2 = 1.067 \cdot 10^{-5}$ m^2/s

The Grashof number was calculated from Eq. (3.21b) by approximating average internal product temperature measurements for the surface temperature measurements. The expansion coefficient, β_e, was determined assuming an ideal gas, and using the average ambient storage temperature values. Using the extreme average product temperature values at the beginning and ending of both storage intervals, ($\bar{Y}$ at 0, 18$^-$, 18$^+$, and 36 hours) the maximum and minimum Grashof numbers were found at these times, as

First storage interval:

$$Gr_{Lx} = \frac{g \cdot \beta_{e,1} \cdot (\bar{Y}(0) - \bar{T}_{1\infty}) \cdot Lx_c^3}{\nu_1^2} = 9.35 \cdot 10^6 \qquad t = 0 \text{ hrs}$$

$$= \frac{g \cdot \beta_{e,1} \cdot (\bar{Y}(18) - \bar{T}_{1\infty}) \cdot Lx_c^3}{\nu_1^2} = 2.16 \cdot 10^6 \qquad t = 18^- \text{ hrs}$$

Second storage interval:

$$Gr_{Lx} = \frac{g \cdot \beta_{e,2} \cdot (\bar{Y}(18) - \bar{T}_{2\infty}) \cdot Lx_c^3}{\nu_2^2} = 1.24 \cdot 10^7 \qquad t = 18^+ \text{ hrs}$$

$$= \frac{g \cdot \beta_{e,2} \cdot (\bar{Y}(36) - \bar{T}_{2\infty}) \cdot Lx_c^3}{\nu_2^2} = 2.95 \cdot 10^6 \qquad t = 36 \text{ hrs}$$

The criterion for both forced and free convection was calculated from Eq. (3.22b) at 0, 18^-,18^+, and 36 hours. Results are shown below.

$$\left.\begin{array}{lll} \left[\frac{Gr_{Lx}}{Re_{Lx}^2}\right] = 1.25 & & t = 0 \text{ hrs} \\ = 0.29 & & t = 18^- \text{hrs} \end{array}\right\} \bar{T}_{1\infty} = -6°C$$

$$\left.\begin{array}{lll} \left[\frac{Gr_{Lx}}{Re_{Lx}^2}\right] = 0.10 & & t = 18^+ \text{hrs} \\ = 0.02 & & t = 36 \text{ hrs} \end{array}\right\} \bar{T}_{2\infty} = -33°C$$

The upper and lower limits of the criterion shown in Eqs. (3.22a-c) were arbitrarily set from 0.1 to 10.0. Both forced and free convection were found to be significant in all but one case; therefore, both forced and free convection were considered in the analysis.

6.1.1.1 Forced Convection

Steady state analytical solutions are available for simple boundary conditions such as a constant temperature or a constant heat flux at the surface of a specified geometry. The actual conditions prevailing

during the experimental procedures were not bound to steady state conditions or either of the two boundary criteria; however, the solutions to these simplified conditions were used to provide order of magnitude estimates of the surface heat transfer coefficients.

The Nussult numbers resulting from forced convection were calculated from Eqs. (3.23a,b), for both the constant temperature and constant heat flux boundary conditions, and for both 18 hour storage interval. Prandlt numbers were found from the properties of air at $\bar{T}_{1\infty}$, and $\bar{T}_{2\infty}$ (Incropera and Dewitt, 1985). Results are shown in Table 6.1.

6.1.1.2 Free Convection

Nussult numbers resulting from free convection were calculated using Eq. (3.25) for the first storage interval ($\bar{T}_{1\infty} > \bar{Y}$), and Eq. (3.24a or b) for the second storage interval ($\bar{T}_{2\infty} < \bar{Y}$), depending on the magnitude of the Rayleigh number. Since the maximum Rayleigh number for the second storage interval was found to be $< 10^7$, Eq. (3.24a) was used. Nussult numbers were calculated using average ambient and internal product temperature measurements from the results shown in Figure 6.1. Both Rayleigh and Nussult numbers are shown in Table 6.2.

6.1.1.3 Combined Free and Forced Convection.

Equation (3.26) with n equal 7/2 was used to calculate Nussult numbers from combined free and forced convection for both aiding and abating free convection. Heat transfer coefficients were determined from the Nussult numbers using Eq. (3.27). Nussult numbers and resulting heat transfer coefficients for both aiding and abating flows are

Table 6.1 Average Nussult Numbers Resulting from Forced Convection over a Flat Plate.

Boundary Conditions	Average Ambient Temperature $\bar{T}_\infty$	Reynolds Number Re_{Lx}	Prandlt Number Pr	Average Nussult $\overline{Nu}_F$
Constant Temperature	-6°C	$2.74 \cdot 10^3$	0.716	31.09
Constant Heat Flux				42.42
Constant Temperature	-33°C	$1.10 \cdot 10^4$	0.723	62.52
Constant Heat Flux				85.30

Table 6.2 Average Nussult Numbers Resulting from Free Convection.

	Time (hours)	Rayleigh Number $Ra_{Lx}(\cdot 10^6)$	Average Nussult Number $\overline{Nu}_N$
	0	6.67	13.72
Storage	3	5.13	12.85
Interval	6	3.85	11.96
No. 1	9	2.89	11.13
(-6°C)	12	2.44	10.67
	15	2.05	10.22
	18^-	1.80	9.88
	18^+	8.34	29.02
Storage	21	7.51	28.27
Interval	24	6.26	27.01
No. 2	27	4.59	24.99
(-33°C)	30	3.75	23.77
	33	2.50	21.48
	36	2.09	20.52

shown in Table 6.3a for the constant temperature assumption, and in Table 6.3b for the constant heat flux assumption.

6.1.1.4 Packaging Layer.

The effective packaging resistance was found using Eq. (3.28), from the thickness and thermal properties of the packaging material. Since the packaging layer was actually composed of three substances; the paperboard box, the plastic film wrapping, and air trapped between the two materials, (Section 4.5) Eq. (3.28) was modified as follows to account for all three substances

$$h_{pk} = \frac{k_{pb}}{L_{pb}} + \frac{k_a}{L_a} + \frac{k_{pf}}{L_{pf}} \qquad (6.1)$$

where L_{pb}, L_a and L_{pf} are the thicknesses of the paperboard, air interface and plastic film, respectively. The thickness of the paperboard, L_{pb}, was found to be $\approx$ 1.7 mm, and the plastic film was $\approx$ 0.3 mm. The air interface varied from 1 to 10 mm. Thermal conductivities were found to be 0.18 W/m°C for the paperboard, and 0.2256 W/m°C for air at the average temperature between -6 and -33°C (Incropera and Dewitt, 1985). The thermal conductivity of the plastic film was 0.20 W/m°C (Modern Plastics Encyclopedia 1984-85). This resulted in an effective packaging coefficient, h_{pk}, ranging from 2.20 W/m°C to 18.10 W/m°C, and averaging 4.75 W/m°C.

Table 6.3a Combined Free and Forced Nussult Numbers, $\overline{Nu}$, and Convective Heat Transfer Coefficients, hx_{cv}, assuming a Constant Temperature Boundary Condition.

Time	Aiding Flow[1]		Abating Flow[2]	
(hours)	$\overline{Nu}$	hx_{cv}	$\overline{Nu}$	hx_{cv}
0	31.59	6.33	30.57	6.13
3	31.49	6.31	30.68	6.15
6	31.40	6.30	30.77	6.17
9	31.33	6.28	30.84	6.18
12	31.30	6.28	30.88	6.19
15	31.27	6.27	30.91	6.20
18^-	31.25	6.27	30.93	6.20
18^+	63.71	11.61	61.27	11.16
21	63.61	11.59	61.38	11.18
24	63.45	11.56	61.55	11.22
27	63.23	11.52	61.79	11.26
30	63.12	11.50	61.91	11.28
33	62.94	11.47	62.09	11.31
36	62.88	11.46	62.16	11.32

1. Free convection aiding forced convection.
2. Free convection opposing forced convection.

Table 6.3b Combined Free and Forced Nussult Numbers, $\overline{Nu}$, and Convective Heat Transfer Coefficients, hx_{cv}, assuming a Constant Heat Flux Boundary Condition.

Time	Aiding Flow[1]		Abating Flow[2]	
(hours)	$\overline{Nu}$	hx_{cv}	$\overline{Nu}$	hx_{cv}
0	42.65	8.55	42.19	8.46
3	42.60	8.54	42.23	8.47
6	42.56	8.53	42.28	8.48
9	42.53	8.53	42.31	8.48
12	42.52	8.52	42.32	8.49
15	42.50	8.52	42.34	8.49
18^-	42.49	8.52	42.35	8.49
18^+	85.86	15.64	84.74	15.44
21	85.81	15.63	84.79	15.45
24	85.73	15.62	84.86	15.46
27	85.63	15.60	84.97	15.48
30	85.58	15.59	85.02	15.49
33	85.49	15.58	85.10	15.51
36	85.47	15.57	85.13	15.51

1. Free convection aiding forced convection.
2. Free convection opposing forced convection.

6.1.1.5 Overall Surface Heat transfer Coefficient.

Overall surface heat transfer coefficients, including the effects of free and forced convection and the packaging layer, were found from Eq. (3.29), using the results shown in Tables 6.3a,b, and in Section 6.1.1.4. Results for both aiding and abating flow conditions are shown in Figures 6.2a,b, for constant temperature and constant heat flux boundary conditions, respectively. There were insignificant differences in both the constant temperature and constant heat flux solutions for aiding and abating flow conditions, indicating that the free convection term had very little influence on the solution.

6.1.2 Estimation of Surface Heat Transfer Coefficients using Inverse Heat Transfer Estimation Techniques.

Surface heat transfer coefficients were estimated using the inverse heat conduction (IHCP) techniques described in Section 3.5.2 from the three test results using the single layer slab of Karlsruhe Test Substance with one exposed surface (Tests 1a-c), described in Section 4.5. The experimental ambient and internal temperature values, shown in Figure 6.1, for the first repetition of Test 1, and in Figures E.1a,b for the second two repetitions of Test 1, and the optimum parameters determined in Section 5.2.4 were used as input to the solution algorithm outlined in Appendix D. Results for the surface heat flux and surface heat transfer coefficients are shown in Figure 6.3, for the first repetition, and in Figures E.2a,b for the second two repetitions.

Estimation of the surface heat flux, $\hat{q}$, and the surface heat transfer coefficient, $\hat{hx}_{Lx}$, produced similar results in all three repetitions. In all cases, there is a sudden increase in both $\hat{q}$ and

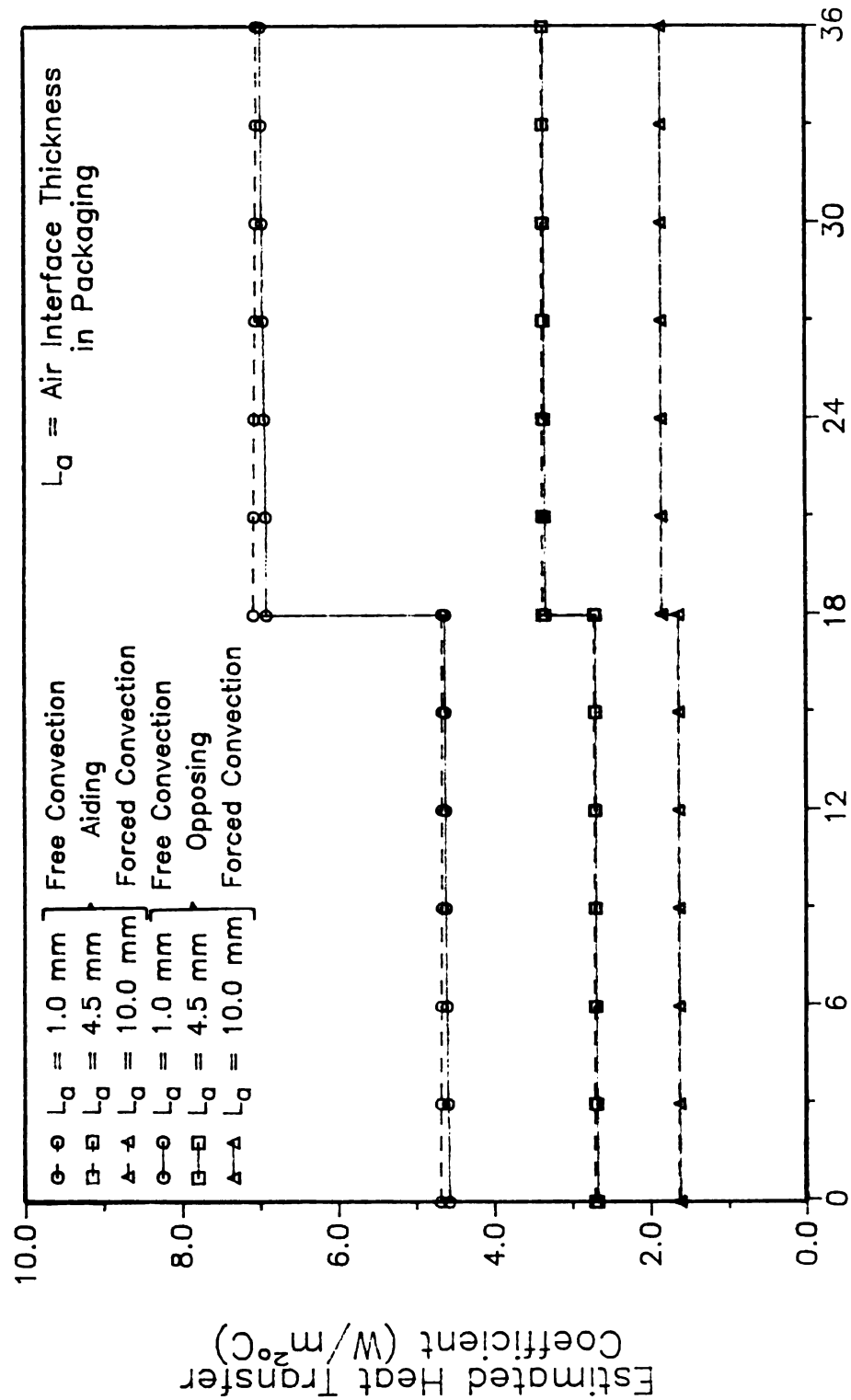


Figure 6.2a Analytical Determination of the Overall Heat Transfer Coefficient using Constant Temperature Boundary Condition.

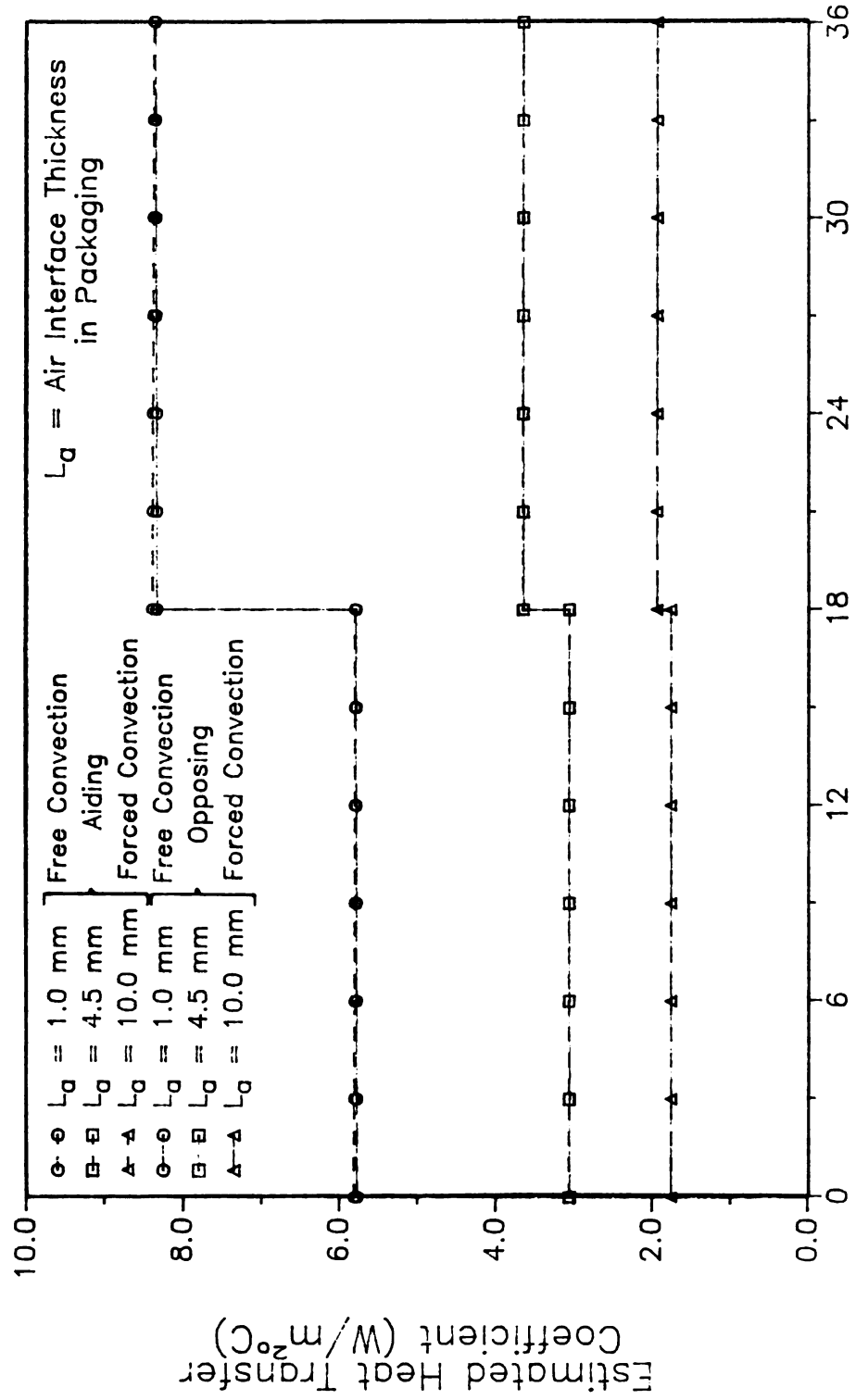


Figure 6.2b Analytical Determination of the Overall Heat Transfer Coefficient using Constant Heat Flux Boundary Condition.

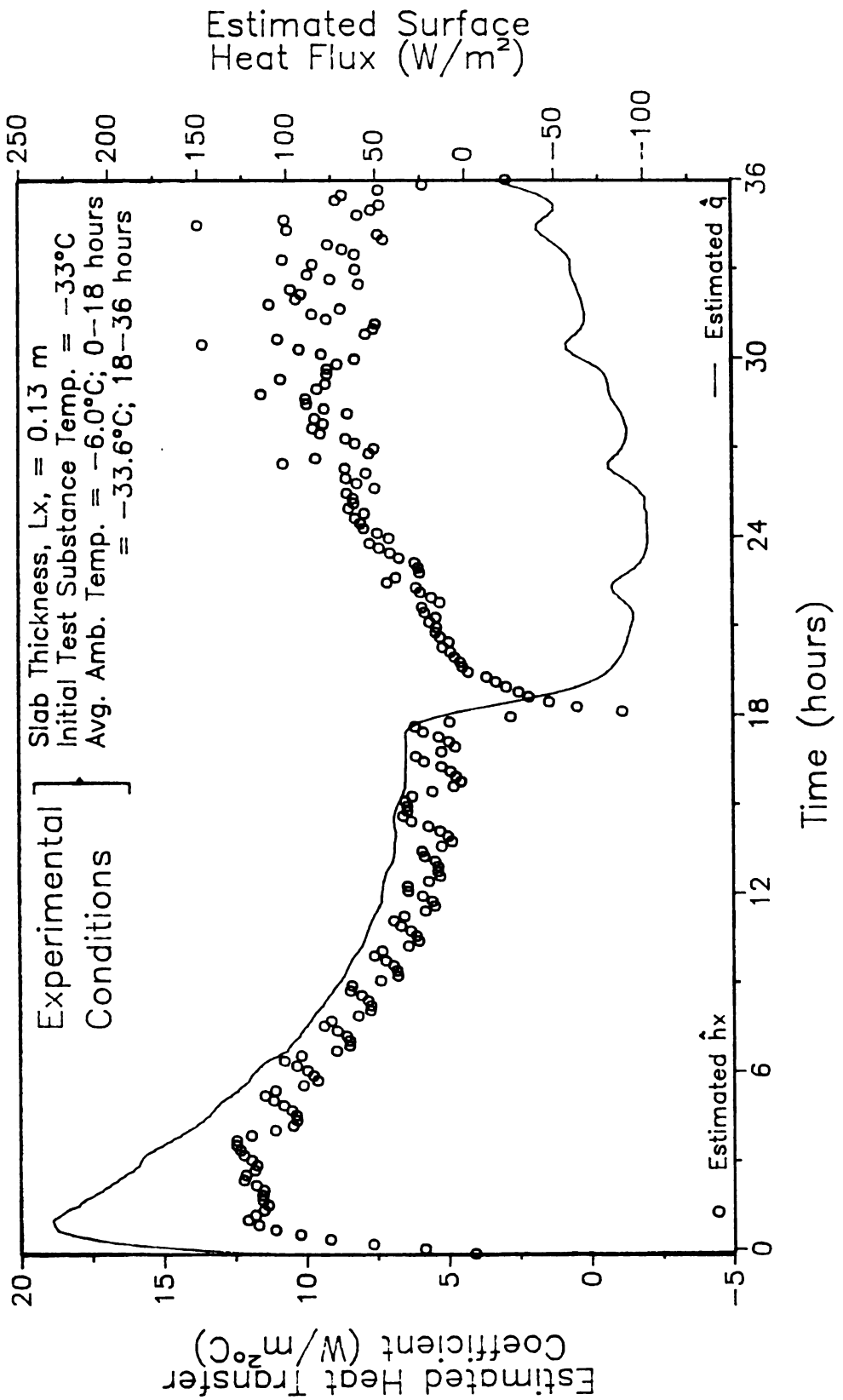


Figure 6.3 Estimated Heat Transfer Coefficients, $\hat{hx}$, and Surface Heat Flux, $\hat{q}$, using Experimental Results with Karlsruhe Test Substance from Single Layer Slab with One Exposed Surface (Test 1a).

$\hat{hx}_{Lx}$ at t = 0 hours, and a sudden decrease followed by a sharp increase in $\hat{hx}_{Lx}$ at t = 18 hours. These observations corresponded to the sudden increase in the estimated surface heat transfer coefficient at t = 0 hours, and sharp drop followed by an increase in $\hat{hx}_{Lx}$ at t = 18 hours, in the solution using exact data from a constant heat transfer coefficient as input (Figure 5.4). Therefore, the sudden changes in the estimated surface heat flux, as shown in Figure 6.3, are assumed to be a direct consequence of the solution method, and not characteristic of the actual boundary conditions.

The estimation procedure provided relatively smooth curves for the surface heat flux compared with those for the estimated surface heat transfer coefficient. The algorithm to estimate $\hat{q}$ was designed to dampen the effects of irregularities in the solution, by including regularization terms and by incorporating several future time steps in the procedure. The surface heat transfer coefficient, however, is a direct function of $\hat{q}$, which is a smoothed function, and the ambient temperature, which fluctuates with the defrost cycle, as shown in Figure 6.1. To dampen the irregularities in the solution resulting from the defrost cycles, the ambient temperatures for each storage interval were averaged in the estimation of $\hat{hx}_{Lx}$. Results for the three repetitions, using averaged ambient temperatures from Figures 6.1 and E.1a,b, are shown in Figures 6.4a-c. The estimation of the surface heat flux was unchanged; however, the irregularities in the estimation of $\hat{hx}_{Lx}$ were damped using the average ambient temperatures.

Comparing $\hat{hx}_{Lx}$ for the two storage intervals indicated that different boundary conditions prevailed in each storage room. The two fans in the first room were at the same end, and operated continuously during each defrost cycle. This resulted in estimated surface heat transfer coefficients which decreased smoothly, indicating that free convection

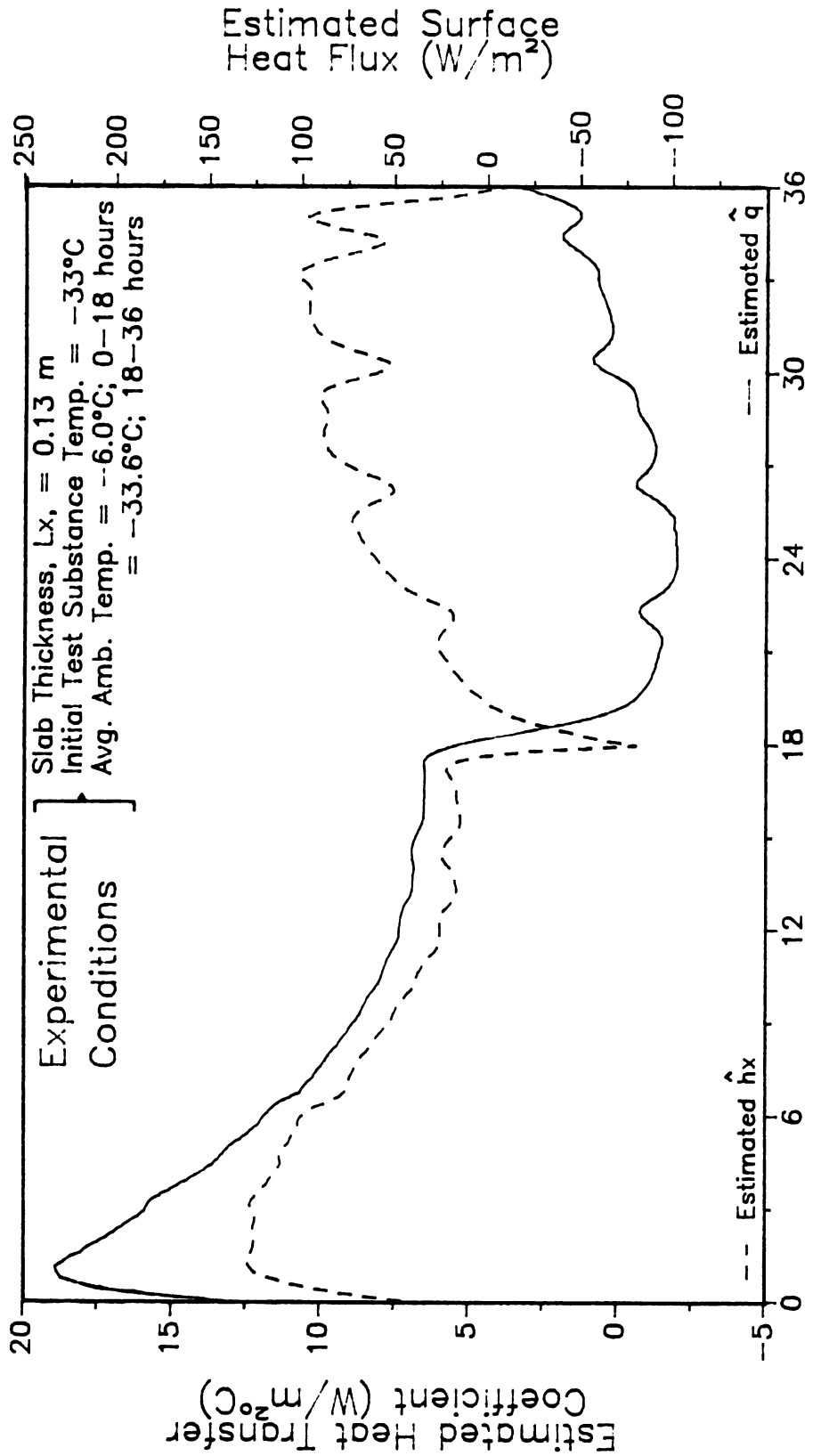


Figure 6.4a ^Estimated Heat Transfer Coefficients, hx, and Surface Heat Flux, q, using Experimental Results with Karlsruhe Test Substance from Single Layer Slab with One Exposed Surface (Test 1a), and Average Ambient Temperatures for Each Storage Interval.

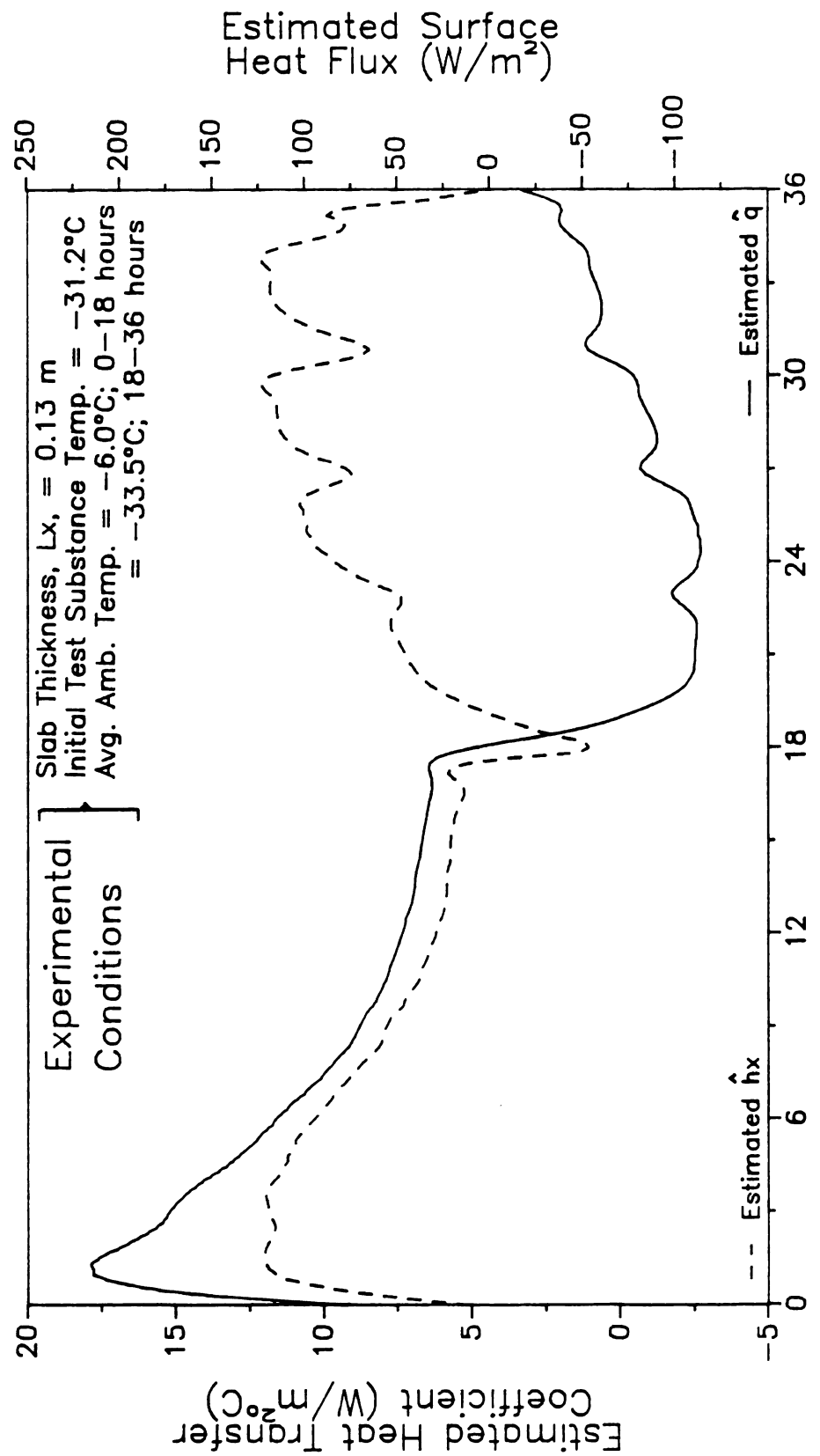


Figure 6.4b ^Estimated Heat Transfer Coefficients, $\hat{h}x$, and Surface Heat Flux, $\hat{q}$, using Experimental Results with Karlsruhe Test Substance from Single Layer Slab with One Exposed Surface (Test 1b), and Average Ambient Temperatures for Each Storage Interval.

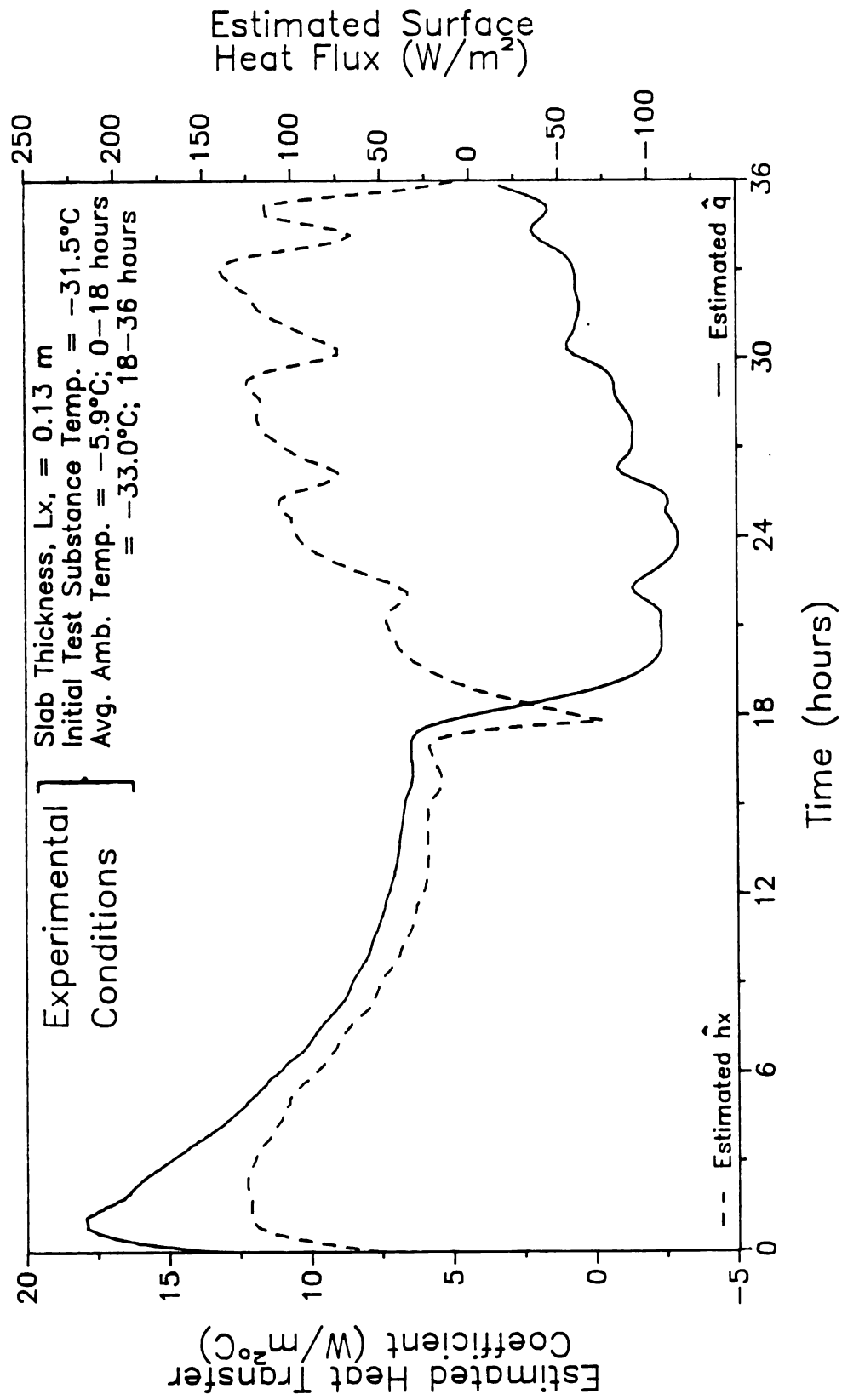


Figure 6.4c ^Estimated Heat Transfer Coefficients, hx, and Surface Heat Flux, q, using Experimental Results with Karlsruhe Test Substance from Single Layer Slab with One Exposed Surface (Test 1c), and Average Ambient Temperatures for Each Storage Interval.

was a significant factor, and that it abated the influence of forced convection. The two fans in the second storage room were at opposite ends, providing a mixed flow, and they cycled on and off in synchronization with the defrost cycle. This resulted in a cyclic curve for $\hat{hx}_{Lx}$ which followed the defrost cycle. Free convection appeared to have less influence during this storage interval as shown by the small overall change in magnitude in $\hat{hx}_{Lx}$ with time. This was assume to be a result of the mixed flow conditions prevailing in the storage room.

6.1.3 Comparison of Results using Analytical and Inverse Heat Conduction Methods

In comparing the analytical and inverse heat conduction solutions for the surface heat transfer coefficient, the assumptions used in generating the solutions were first examined.

The assumptions made in generating the analytical solution are given below.

1. Steady state conditions,
2. Constant temperature or constant heat flux boundary conditions,
3. Laminar, unidirectional air flow over surface of product,
4. Known temperature at surface of product,
5. Negligible changes at packaging interface, such as the build up of frost,
6. One dimensional heat transfer,
7. Negligible moisture loss, and
8. Constant ambient temperature.

The assumptions used in the inverse heat conduction solution are as follows.

1. One dimensional heat transfer,

2. The surface heat transfer coefficient is a constant, or a function of time only,
3. Homogeneous, isentropic thermal properties,
4. Temperature measurements are made at a known position within product, and
5. Negligible moisture loss.

The assumptions used in the IHCP solution are consistent with the conditions prevailing during the experimental procedures; however, many of the assumptions used in the analytical solution conflict with these conditions. Examples of conditions which conflict the assumptions used in the analytical case are transient heat transfer, mixed flow conditions, and known internal product temperatures (not at surface). The assumptions used in the analytical solution were very restrictive, compared to those used in the IHCP solution, which was designed to accommodate a wide variety of boundary conditions and variable thermal properties.

Both methods yielded surface heat transfer coefficients ranging from 2 to 18 W/m°C, as shown in Figures 6.2a,b and 6.4a-c. In the first storage interval, the IHCP solution indicated that either free convection or the diminution of frost was the limiting factor influencing the magnitude of the surface heat transfer coefficient, while in the analytical solution, forced convection dominated. The four primary factors influencing the magnitude and influence of free convection in the analytical solution were: (1) the magnitude of the coefficients in Eqs. (3.24a,b) and (3.25); (2) the magnitude of the difference between the ambient and product temperatures, used in determining the Rayleigh number; (3) the value of n used in Eq. (3.26); and, (4) the magnitude of the air free stream velocity used in calculating the Reynolds number for forced convection. The magnitudes of the coefficients in Eqs. (3.24a,b)

and (3.25) were given for very specified flow conditions, and the optimum coefficients for the flow conditions used in this study may have actually been greater or less than those presented in these equations. Futhermore, since internal product temperatures were used instead of surface temperatures, these estimated values of $\hat{h}x_{Lx}$ are assumed to higher than those that might have been obtained using surface temperatures. The value of n in Eq. (3.26) was also very influential: as n increases, the influence of the higher Nussult number (free or forced) increases exponentially; therefore, if n = 3 had been used, free convection would have been more influential. Finally, the variability of the velocity measurements was high, and the velocity was assumed to be constant over each storage interval, disregarding the defrost cycles. In summary, small changes in the analytical determination of the surface heat transfer coefficients may have resulted in more or less influence from free convection.

Both solutions yielded little overall change in the surface heat transfer coefficient with time during the second storage interval; however, the IHCP solution responded to the defrost cycle, with decreases and increases in the estimated surface heat transfer coefficient as the fans turned off and on. To detect similar variations in air velocities using the analytical method, velocity measurements would have been required throughout the storage interval.

Since the conductivity of air is very low, especially at low temperatures, the air interface thickness, L_a, provided the highest resistance to heat transfer, and it was most influential in the determination of the overall surface coefficient. Therefore, using different values for L_a changed the solution significantly, as seen in Figures 6.2a,b. The air interface thickness was very difficult to measure accurately, and the measurements varied significantly, within a given

package. The inverse heat conduction solution does not require knowledge of the surface conditions in the estimation procedure; therefore it is not subject to the high variation in the analytical solution resulting from the interface measurements. Furthermore, the analytical solution does not account for the accumulation and diminution of frost as the product is cooled and heated, both in the package interface and on the surface of the product. The heat transfer coefficients of snow and ice are 10 to 100 times greater than that of air; therefore, as the frost layer accumulates or diminishes, the resistance to heat transfer decreases or increases accordingly. This results in an increasing or decreasing heat transfer coefficient.

The analytical solution provided a continuous, smooth estimation of the surface heat transfer coefficients during each storage interval, with a step change in the estimation of $\hat{hx}_{Lx}$ between storage intervals. The IHCP method predicted, however, a sharp increase in $\hat{hx}_{Lx}$ at the beginning of the first storage interval, a sudden dip in $\hat{hx}_{Lx}$ between storage intervals, and a sharp drop in $\hat{hx}_{Lx}$ at the end of the second storage interval. As discussed previously, these sudden changes in $\hat{hx}_{Lx}$ are all assumed to be characteristic of the estimation method, and not the actual boundary conditions. This indicates that this method does not respond well to step changes in surface conditions.

In summary, although use of the analytical and IHCP methods yielded surface heat transfer coefficients within similar ranges, (from 2 to 8 W/m°C for the analytical solution, and from ≈ 5 to 17 W/m°C for the inverse solution), several differences in the solution methods must be noted. Unlike the analytical method, the IHCP method provided a solution valid for time dependent boundary conditions, without any restrictions on the airflow pattern, or changes at the product surface, such as frost accumulation. In addition, the analytical solution was

highly dependent on experimental measurements of air velocity and air interface thickness, which were both extremely variable and difficult to measure accurately. In comparing the results for the three test cases, shown in Figures 6.4a-c, almost identical estimations for $\hat{hx}_{Lx}$ were obtained, indicating that the inverse method produces repeatable results. The primary drawback of the inverse heat conduction method was its poor response to step changes in surface conditions. Poor response to step changes at the beginning and end of the overall experimental test time can be avoided, however, by taking temperature measurements before, and continuing measurements after the designated testing time (Beck et. al., 1985).

6.2 Simulation of One Dimensional Heat Conduction Through a Food Product

The one dimensional heat conduction program (Appendix B) was verified by comparison with analytical and experimental results. Analytical solutions obtained using constant thermal properties, and experimental data (Section 4.5) from the single layer slab with one exposed surface (Tests 1a-c), and the double layer slab with one exposed surface (Tests 2a-c) were used in the evaluations.

6.2.1 Comparison with Analytical Solutions

The analytical solution, assuming constant thermal properties, was determined for a step change in ambient temperature. The temperature distribution within a body, initially at a uniform temperature, T_0, then exposed to constant ambient conditions, $T_{1\infty}$ and hx_{Lx}, at $x = Lx$, and then subject to a step change in ambient conditions ($T_{1\infty}$ and hx_{Lx}) at

time t_1, is given below (Carslaw and Jaeger, 1959). (The boundary at x = 0 is insulated throughout.)

$$T(x,t) = T_o + (T_{1\infty} - T_o)\cdot\left[\, 1 - R_\ell \cdot e^{-\lambda_\ell t} \,\right] \qquad 0 < t \le t_1 \qquad (6.2a)$$

$$T(x,t) = T_{2\infty} - R_\ell \cdot T_o' \cdot e^{-\lambda_\ell t} \qquad t_1 < t \le (t_1+t_2) \qquad (6.2b)$$

where

$$\lambda_\ell = (k/\rho Cp)\cdot\zeta_\ell^2$$

$$R_\ell = \frac{2\cdot hk\cdot\cos(\zeta_\ell x)}{[((hk)^2 + \zeta_\ell^2)\cdot Lx + hk]\cdot\cos(\zeta_\ell Lx)}$$

$$T_o' = T_{1\infty} - T_o + (T_{2\infty} - T_{1\infty})\cdot e^{-\lambda_\ell t_1}$$

$$hk = hx_{Lx}/k$$

The times t_1 and t_2 were both set at 18 hours, and the thickness, Lx, was set equal to 0.13 meters; all of these values were consistent with the values used in the experimental procedures for the single layer slab with one exposed surface (Tests 1a-c). The ambient temperatures, $T_{1\infty}$ and $T_{2\infty}$, and the heat transfer coefficient, hx_{Lx}, were based on the overall average values found from Tests 1a-c. The initial temperature was set at -33°C, and the ambient temperatures, $T_{1\infty}$ and $T_{2\infty}$, were set at -6°C and -33°C, respectively. The average overall heat transfer coefficient, was found from Figures 6.4a-c to be 8.5 W/m^2°C, and was used for hx_{Lx} in the analytical solution. The thermal properties were calculated

using the average values between -33 and -12°C. This range was determined from the minimum and maximum product temperature values shown in Figures 6.1 and E.1a,b.

The numerical one dimensional solution was found using identical initial and boundary conditions, and using both constant and temperature dependent thermal properties. A time step of 120 seconds and a position increment of 0.016 m were used in the numerical solution. These are the same values proposed for use in Section 5.1.3.

The temperature histories at the mid-section of the slab (x = 0.065 m), from the analytical solution, using constant thermal properties, and from the numerical solution, using both constant and temperature dependent thermal properties, are shown in Figure 6.5. The numerical solution with constant thermal properties provided an excellent approximation to the analytical solution (using constant properties). The effect of the temperature dependent thermal properties was found to be very significant: at the end of the first storage interval, the solution using variable thermal properties was 27% higher than both analytical and numerical solutions using constant thermal properties. This indicates the importance of accurate estimation of thermal properties of foods during the freezing and post-freezing processes.

Similar solutions were obtained using the same conditions, described in the experimental procedures, for the double layer slab, with a thickness (Lx) of 0.25 m, and only one exposed surface (Tests 2a-c). In these tests, the methyl-cellulose boxes were layered two high, and both storage intervals were increased to 24 hours. The same initial and boundary conditions, average thermal property values, time step, and spatial increment used for the comparison with the single layer slab (Figure 6.5), were also used in this comparison. Solutions for the

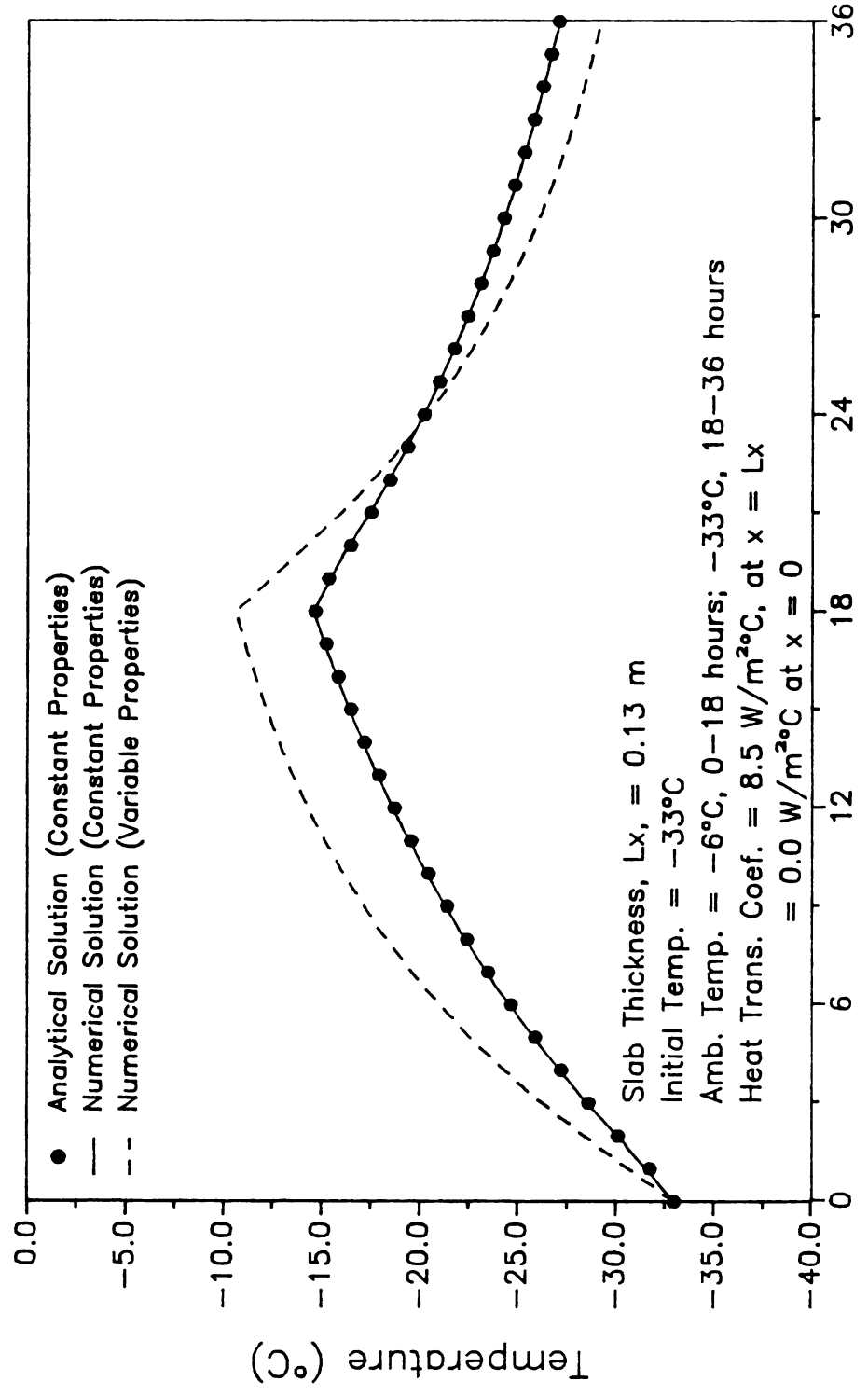


Figure 6.5 One Dimensional Numerical Solution Compared with Analytical Solution with Constant Thermal Properties of Karlsruhe Test Substance at x = Lx/2.

temperature histories at Lx/4 (x = 0.0625 m), and 3•Lx/4 (x = 0.1875 m) are shown in Figure 6.6. Again, the analytical and numerical solutions using constant thermal properties were almost identical, while the numerical solution using variables thermal properties yielded temperature values significantly higher.

6.2.2 Comparison with Experimental Results.

Results obtained using the numerical one dimensional heat conduction program were compared to the experimental results obtained for the single layer slab, with one exposed surface, (Tests 1a-c), and for the double layer slab, with one exposed surface, (Tests 2a-c). In simulating the conditions in the experimental procedures, the predicted heat transfer coefficients, shown in Figures 6.4a-c, were averaged over specified time increments for each storage interval. These values were used as input to the numerical model and compared with the experimental results.

In using the numerical model to compare with the experimental results, the surface heat transfer coefficient was determined in two ways. First, the surface heat transfer coefficients were averaged over the total storage interval, and second, the surface heat transfer coefficients were averaged over two to four hourly increments over the total storage interval. In each case, variable thermal properties, and identical product thicknesses (Lx = 0.13 m) were used. The thermal properties were calculated using the initial freezing temperature, T_{if}, given by Gutschmidt (1960), and the thermal properties of the unfrozen methyl-cellulose, ρ, k, and Cp, given by Specht et. al. (1981), shown in Table 4.1. The initial and ambient temperatures were obtained from average initial product temperature and overall ambient temperature

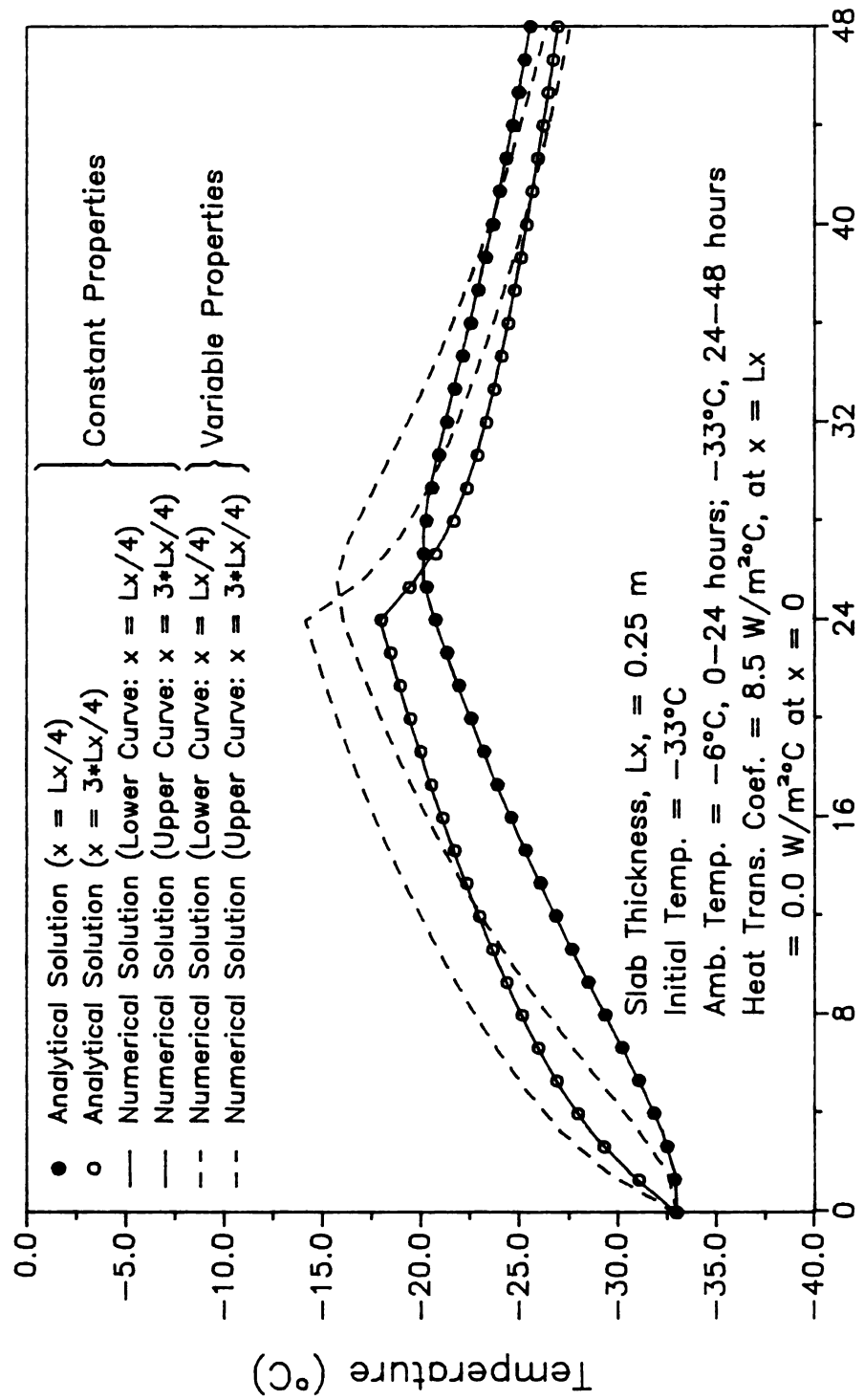


Figure 6.6 One Dimensional Numerical Solution Compared with Analytical Solution with Constant Thermal Properties of Karlsruhe Test Substance at x = Lx/4 and x = 3·Lx/4.

measurements over each storage interval. Again, an insulated boundary condition was imposed at x = 0. The initial and boundary conditions used in the numerical solution are shown in Table 6.4.

Results are shown in Figure 6.7 for the first repetition using the single layer slab with one exposed surface, (Tests 1a), and in Figure E.3a,b for the second two repetitions (Tests 1b,c). In all cases the solution obtained using incrementally averaged surface heat transfer coefficients yielded very similar values as the experimental results. This was expected, since the one dimensional solution was used directly in estimating the surface heat transfer coefficients. The temperature solutions obtained using overall averaged surface heat transfer coefficients yielded slightly lower values than the experimental data.

To compare the numerical solution with the results of the two layer slab with one exposed surface (Tests 2a-c), the surface heat transfer coefficients shown in Table 6.4, calculated using the experimental results for the single layer slab, were used. The time increments for the surface heat transfer coefficients were adjusted to account for the longer storage intervals (24 hours) used in the experimental procedures for the double layer slab, compared with the intervals (18 hours) used for the single layer slab. The initial and ambient temperatures used in the numerical solution were based on the average initial and ambient temperatures for the experimental results of the double layer slab. These values are shown in Table 6.5.

Numerical solutions were found at Lx/4 (x = 0.63 m) and at 3•Lx/4 (x = 0.188 m), and compared with temperatures measurement at approximately the same locations in the double layer slab configuration, described in Section 4.2. These comparisons are shown in Figure 6.8, for the first repetition of the experimental test using the double layer slab, and in Figures E.4a,b, for the second two repetitions of the test.

Table 6.4. Surface Heat Transfer Coefficients used in the Numerical Solution in the Comparison with Experimental Results of the Single Layer Slab with One Exposed Surface (Tests 1a-c).

Time	Surface Heat Transfer Coefficient ($W/m^2°C$) Test 1a	Test 1b	Test 1c	Average
Storage Interval 1:				
0 - 2 hr	12.31	11.90	12.00	12.07
2 - 4 hr	12.16	11.81	11.99	11.99
4 - 6 hr	11.18	11.09	10.78	11.01
6 - 8 hr	9.11	9.48	9.10	9.23
8 -10 hr	7.81	7.85	7.55	7.74
10 -12 hr	6.44	6.63	6.47	6.51
12 -14 hr	5.65	5.93	5.88	5.82
14 -16 hr	5.56	5.66	5.71	5.65
16 -18 hr	5.49	5.45	5.58	5.51
Storage Interval 2:				
18 -20 hr	5.37	6.62	6.65	6.21
10 -12 hr	7.34	9.20	9.46	7.44
12 -14 hr	9.24	10.86	11.10	10.40
14 -16 hr	9.56	10.86	10.91	10.38
Average for Storage Interval 1:	8.41	8.42	8.34	8.39
Average for Storage Interval 2:	8.37	9.73	9.93	9.34
	Initial and Storage Temperatures (°C)			
T_0	-33.0	-31.2	-31.5	-31.9
$T_{\infty,1}$	-6.0	-6.0	-5.9	-6.0
$T_{\infty,2}$	-33.6	-33.5	-33.0	-33.4

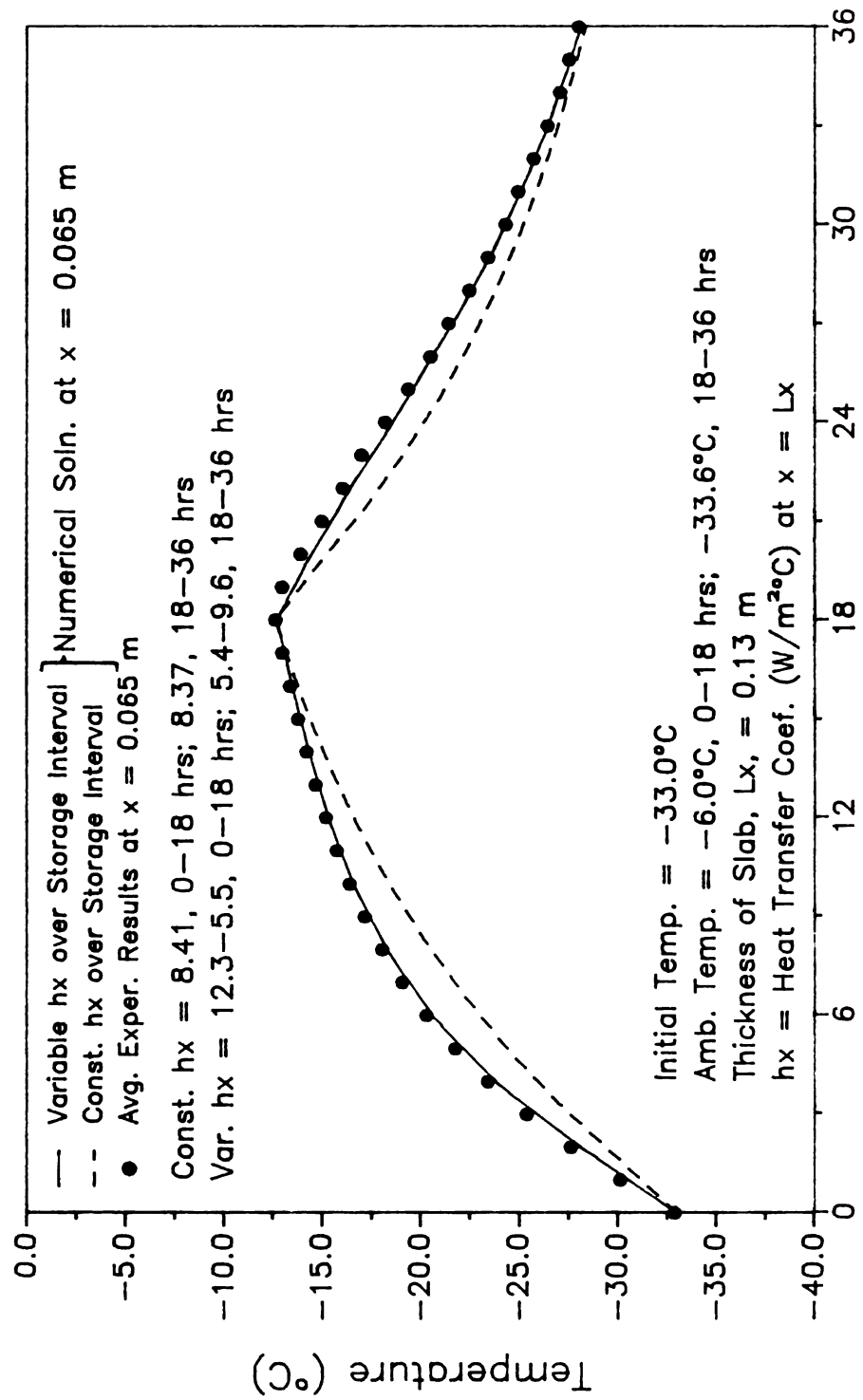


Figure 6.7 One Dimensional Numerical Solution Compared to Experimental Results from Single Layer Slab with One Exposed Surface (Test 1a).

Table 6.5. Initial and Ambient Temperatures used in Numerical Solution for Comparison with Experimental Results using the Double Layer Slab with One Exposed Surface (Test 2a-c).

	Initial and Storage Temperatures (°C)		
	Test 2a	Test 2b	Test 2c
T_0	-25.9	-27.7	-24.0
$T_{\infty,1}$	- 6.0	- 6.1	- 6.1
$T_{\infty,2}$	-29.2	-26.9	-26.0

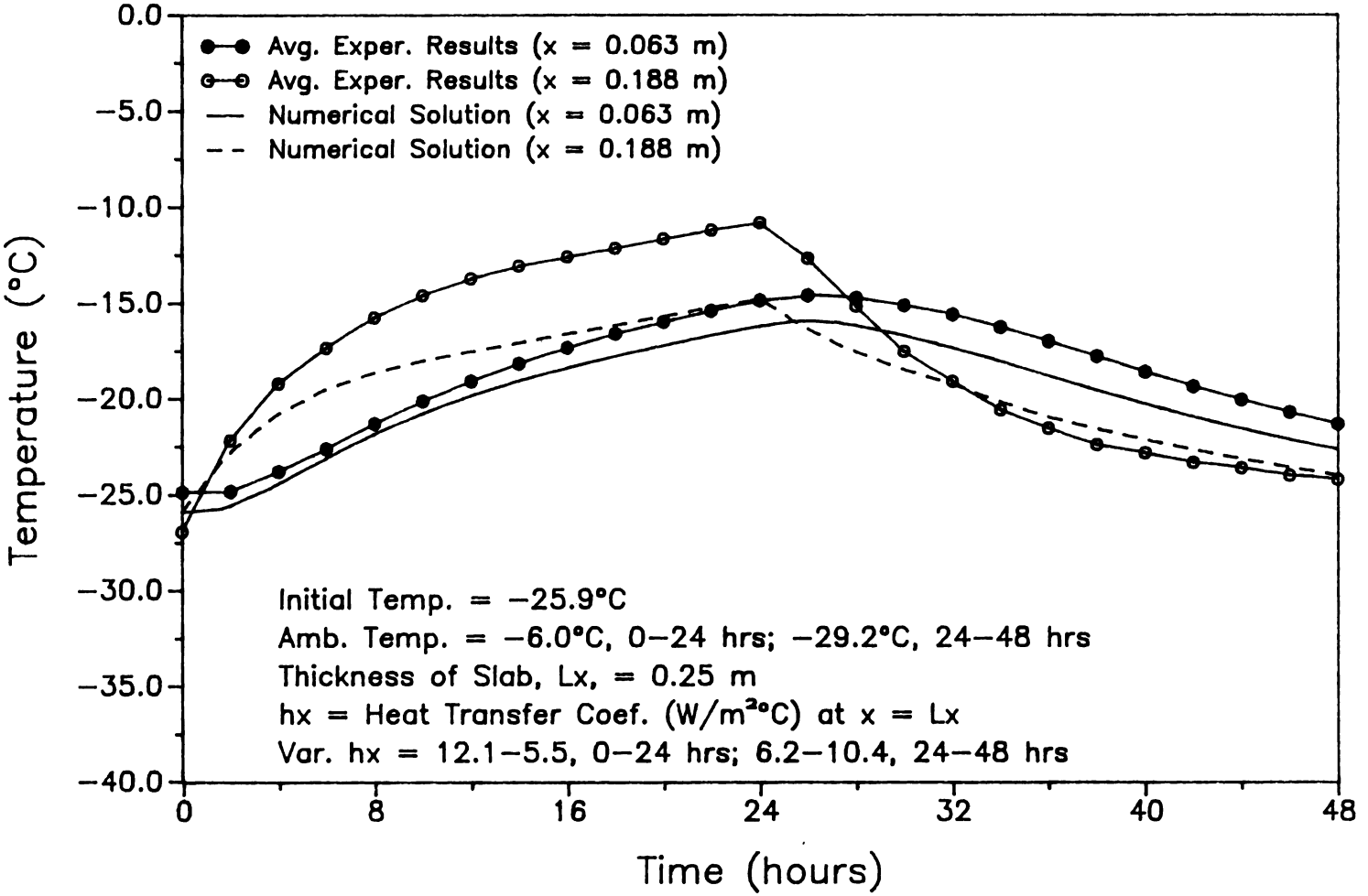


Figure 6.8 One Dimensional Numerical Solution Compared to Experimental Results from Double Layer Slab with One Exposed Surface (Test 2a).

In all cases, the predicted values from the numerical solution using variable thermal properties were significantly lower than those obtained experimentally. In addition, the predicted values nearer the surface at x = 0.188 were less accurate than those obtained near the insulated surface at x = 0.063 m. (After 24 hours, results yielded a 37% error at x = 0.188 m, while the error at x = 0.063 m was 9%.)

The following explanations for these results were proposed: (1) the initial freezing point of the methyl-cellulose was actually -0.7°C (Specht et. al., 1981), and not -1.0°C, as given by the original source, Gutschmidt (1960); (2) the assumption of negligible resistance to heat transfer due to the packaging interface within the total product mass was invalid; and, (3) the assumption of perfect insulation along the first boundary (at x = 0), used in the experimental procedures, was invalid.

To test these hypothesis, the numerical simulation was first repeated using T_{if} = -0.7°C, instead of T_{if} = -1.0°C, as given by Gutschmidt (1960), to determine the temperature dependent thermal properties. All other input values for the numerical model remained unchanged from those used for the solution shown in Figure 6.8. The numerical solutions at x = Lx/4 and x = 3•Lx/4 are compared with the experimental results in Figure 6.9. Comparing the numerical solutions in Figure 6.8, using T_{if} = -1.0°C, and in Figure 6.9, using T_{if} = -0.7°C, there was very little difference in the numerical solutions, indicating that the difference in the numerical and experimental values was not primarily a result of the differences in the magnitude of T_{if} found in the literature.

The second explanation, proposed to explain the differences between the numerical and experimental results, was that the assumption of

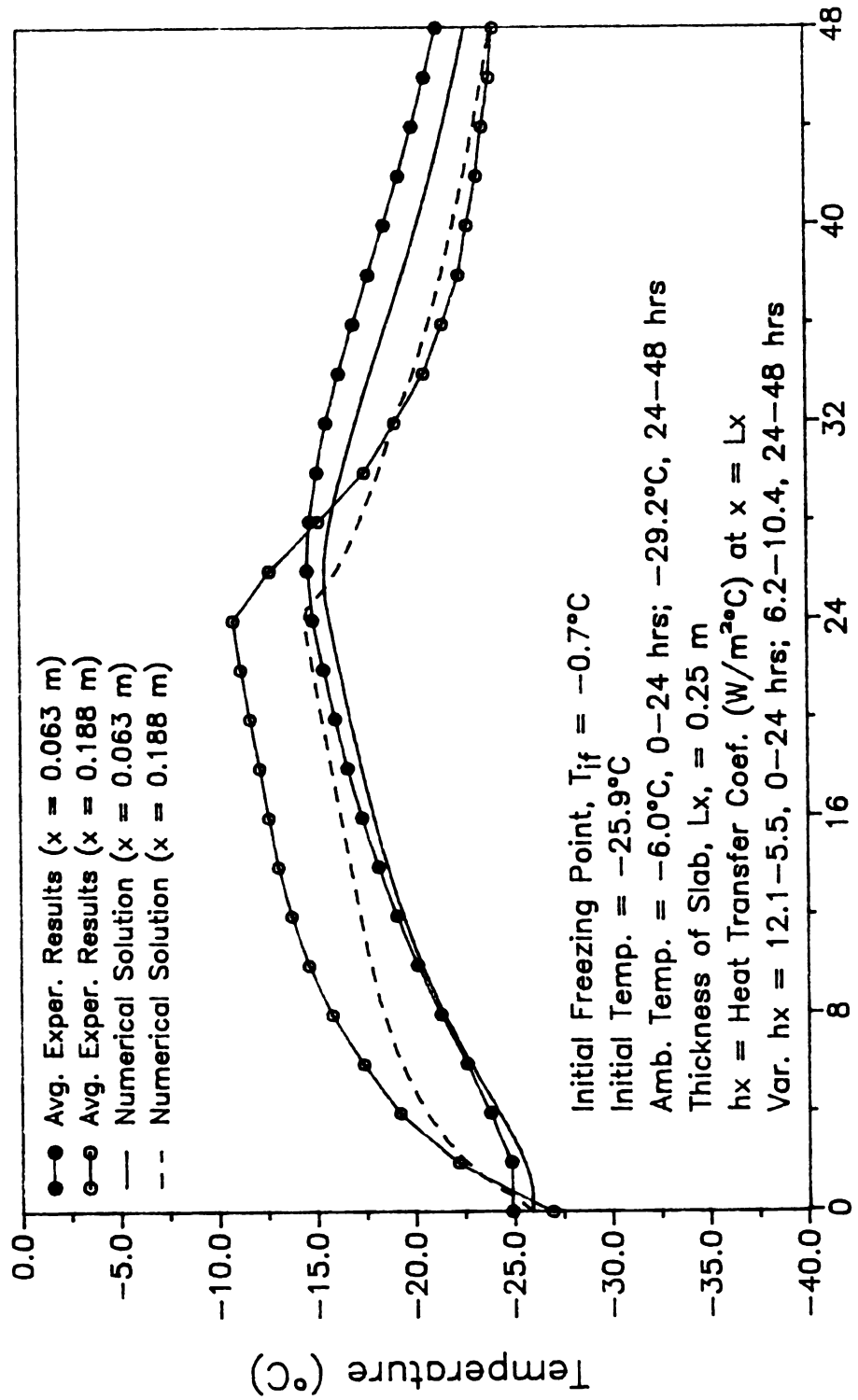


Figure 6.9 One Dimensional Numerical Solution, using Initial Freezing Point, T_{if}, Given by Specht et. al. (1981), Compared to Experimental Results from Double Layer Slab with One Exposed Surface (Test 2a).

negligible resistance to heat transfer due the internal packaging between the containers was invalid. To test this hypothesis, the extreme case of infinite resistance to heat transfer at the packaging boundary was considered. The numerical simulation shown in Figure 6.8 was repeated using an insulated boundary condition at the location of the internal packaging boundary (Lx/2). This problem was similar to the case of the single layer box with one exposed surface (Test 1a-c). The numerical solution at 3•Lx/4 (x = 0.188 m) for this hypothesis was compared to the experimental and previous numerical results, given in Figure 6.8 at the same location. Results are shown in Figure 6.10. The experimental results fall between the two extreme cases of no resistance to heat transfer and infinite resistance to heat transfer at the packaging interface, supporting the hypothesis that the assumption of negligible resistance to heat transfer at the packaging interface was invalid.

The third hypothesis was that the assumption used in the experimental procedures, of a perfectly insulated boundary condition at the inner container surface (x = 0), was invalid. The container which held the methyl-cellulose paperboard boxes was designed to limit the heat flow at the unexposed surfaces to less than 1% of the expected heat transfer rate at the exposed surface, assuming equal surface heat transfer coefficients on all sides of the container. However, the bottom of the container rested on a stainless steel cart (high conductivity), so that in the extreme case of perfect conductance between the container and the cart, the bottom of the container may have been at the ambient temperature. Based on the thicknesses of the insulation board and plywood in the container (Section 4.2), a conservative estimate of the effective heat transfer coefficient, defined as (insulation thickness)/(thermal conductivity), through the insulation materials was 1 $W/m^2 \cdot C$. The

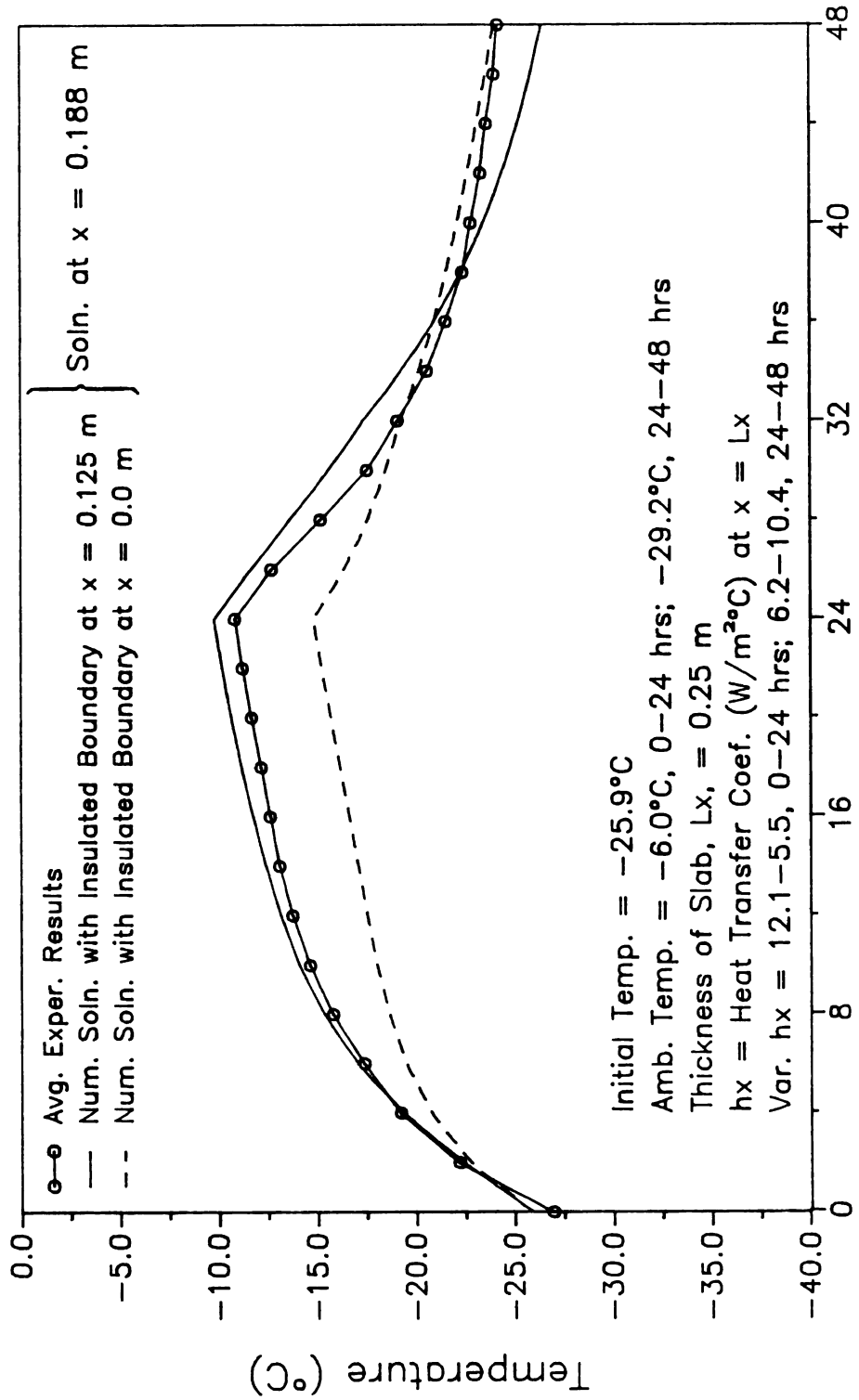


Figure 6.10 One Dimensional Numerical Solution, with and without Imposed Insulated Boundary Condition at Packaging Interface (x = Lx/2), Compared to Experimental results from Double Layer Slab with One Exposed Surface (Test 2a).

simulation shown in Figure 6.8 was repeated using a heat transfer coefficient (hx_{Lx_0}) of 1.0 $W/m^2°C$ at x = 0. Results are shown in Figure 6.11. The temperatures at Lx/4 from 0 to 24 hours were over-estimated, while the temperature values from 24 to 48 hours were under-estimated, supporting the third hypothesis.

Comparing Figure 6.10 and Figure 6.11, suggests that both assumptions of negligible internal packaging resistance to heat transfer, and a perfectly insulated boundary at x = 0 were invalid.

These results indicate that there is a need for further study in estimating the resistance to heat transfer due to the packaging interface. The packaging interface can be thought of as a contact resistance, with a contact resistance coefficient associated with it. This coefficient may be estimated using the same methods used for estimating the surface heat transfer coefficients. In this case, there are two unknowns, the surface heat transfer coefficient and the contact resistance, requiring additional internal temperature measurements. At least one thermocouple would be required on each side of the packaging interface. The existing computer program developed to estimate the surface heat transfer coefficient (Appendix D) could be used with minor modification to also estimate the contact resistance coefficient.

6.3 Simulation of Two Dimensional Heat Conduction Through a Food Product.

The two dimensional heat conduction program (Appendix C) was verified by comparison with the one dimensional model and experimental results. The one dimensional model, discussed in Section 6.2, was used to demonstrate that the two dimensional solution can be reduced to the one dimensional solution. Consistant with the comparisons described in

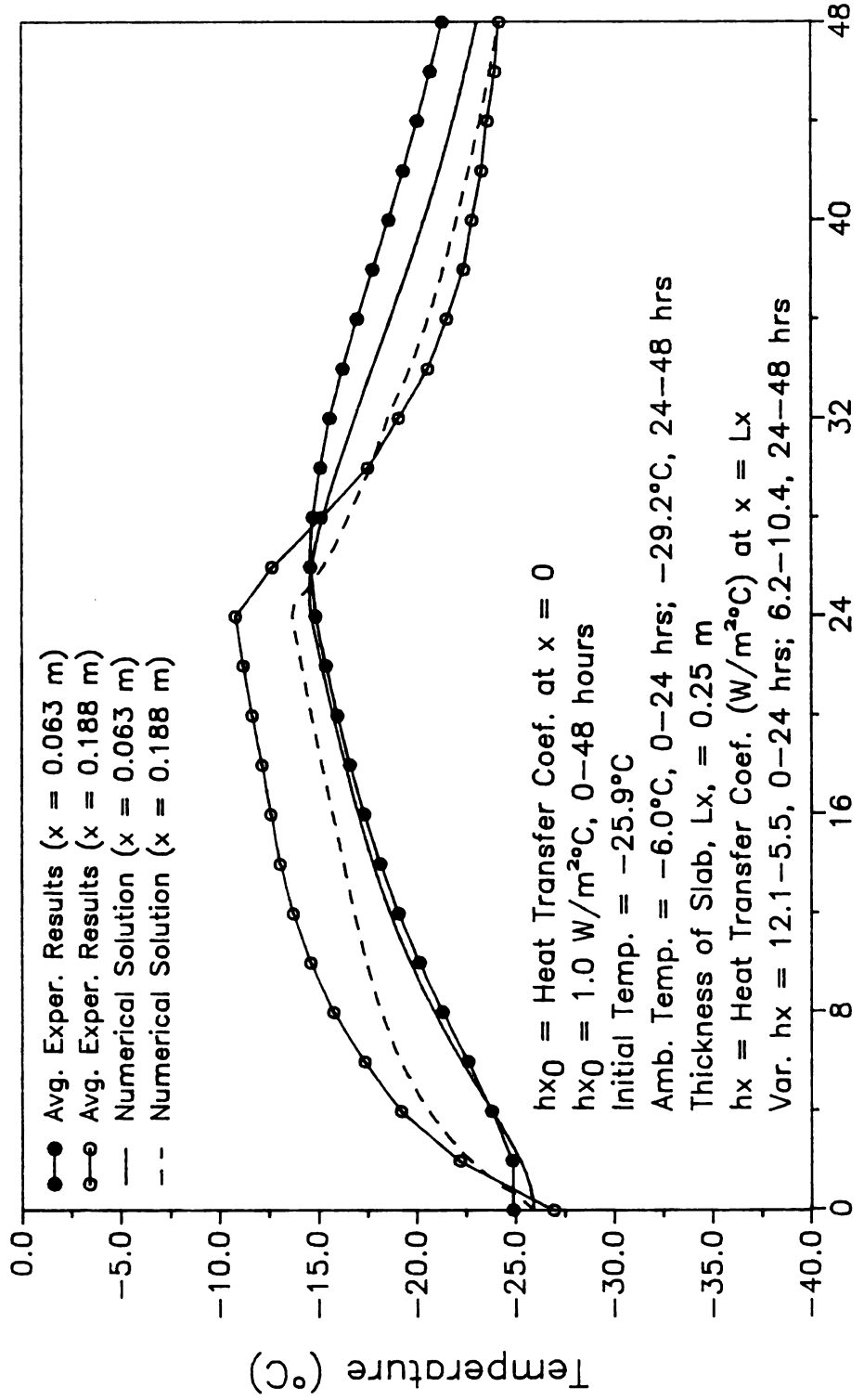


Figure 6.11 One Dimensional Numerical Solution, with hx_o = 1 $W/m^2°C$ at x = 0, Compared to Experimental results from Double Layer Slab with One Exposed Surface (Test 2a).

Section 6.2.1, the two dimensional solution was compared with the one dimensional solution using constant thermal properties. The experimental data (Section 4.5, Tests 3a-c) was compared with two dimensional numerical solution using variable thermal properties.

6.3.1 Verification of Two Dimensional Model.

The one dimenisonal numerical model, used in verifying the two dimensional numerical model, was shown to approximate the analytical solution excellently, using constant thermal properties, in Section 6.2.1. The two dimensional model was compared with the one dimensional numerical model for two reasons: (1) to verify the accuracy of the two dimensional model; and (2) to show that the two dimensional model may be reduced to the one dimensional model.

The geometry of the problem considered here was based on the geometry used in the experimental procedures for Tests 3a-c. A rectangular rod, measuring 0.25 m by 0.27 m, with perfectly insulated surfaces along the third dimension was considered in the two dimensional model. A surface heat transfer coefficient of 8.5 $W/m^2°C$ was alternately imposed along two of the remaining parallel surfaces, and an insulated boundary condition was imposed along the other two surfaces. The initial temperature was -33°C, and the simulated ambient conditions were taken to be -6°C from 0 - 48 hours, and -33°C from 48 to 96 hours. These were the same storage time intervals used in Tests 3a-c. First, the surface heat transfer coefficient was imposed at $x = 0.25$ m, and the surface at $y = 0.27$ m was taken to be insulated. This solution was compared with the one dimensional solution using a slab, 0.25 m in thickness, with a surface heat transfer coefficient of 8.5 $W/m^2°C$ at $x = 0.25$ m, and an insulated condition at $x = 0.0$ m. Results are shown in

Figure 6.12a. A similar comparison was made using a surface heat transfer coefficient of 8.51 $W/m^2°C$ along the surface at y = 0.27 m, and an insulated condition on all other surfaces, as the boundary conditions for the two dimensional model, and similar boundary conditions (hy = 0, at y = 0; hy = 8.5 $W/m^2°C$, at y = 0.27 m), with y = 0.27 m, for the one dimensional model, as shown in Figure 6.12b. in both instances, the solutions proved to be identical, which demonstrates that the two dimensional numerical solution may by reduced to the one dimensional solution, and that, since the one dimensional solution accuracy was previously verified (Section 6.2.1), the two dimensional numerical solution is accurate.

6.3.2 Comparison with Experimental Results.

The two dimensional numerical heat conduction program was compared with the experimental results using the triple layer, two dimensional configuration (Tests 3a-c), shown in Figure 4.3. Numerical solutions were found using the same geometry for the configuration used in Tests 3a-c (Lx = 0.25 m, Ly = 0.22 m), with insulated boundary conditions imposed along the surfaces at x = 0, and y = 0. The heat transfer coefficients shown in Table 6.4 were used as input for hx_{Lx} and hy_{Ly} in the numerical model, (hx_{Lx} and hy_{Ly} were considered to be equal). Since the storage interval used for the experimental tests was longer than either of the one dimensional tests (Test 1a-c and Test 2a-c), the time increment for the last heat transfer coefficient of each storage interval, shown in Table 6.4, was adjusted to account for the longer storage time of the two dimension tests. This was done on the assumption that the surface heat transfer coefficients, shown in Figures 6.4a-c, approach a constant value at the end of each storage period. (The sudden

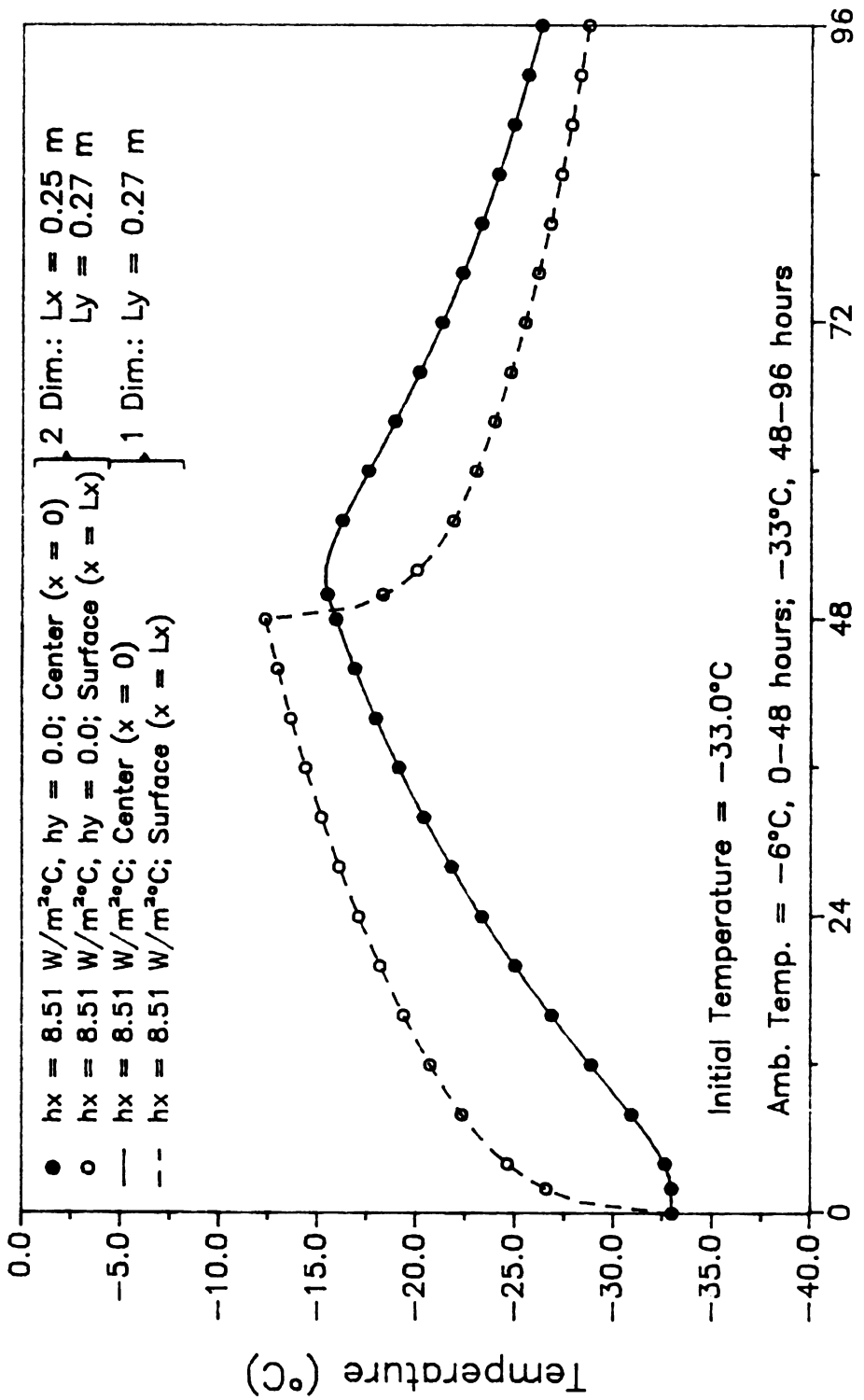


Figure 6.12a Verification of Two Dimensional Numerical Solution with hx = 8.51 W/m²°C, hy = 0.0, with One Dimensional Numerical Solution with hx = 8.51 W/m²°C.

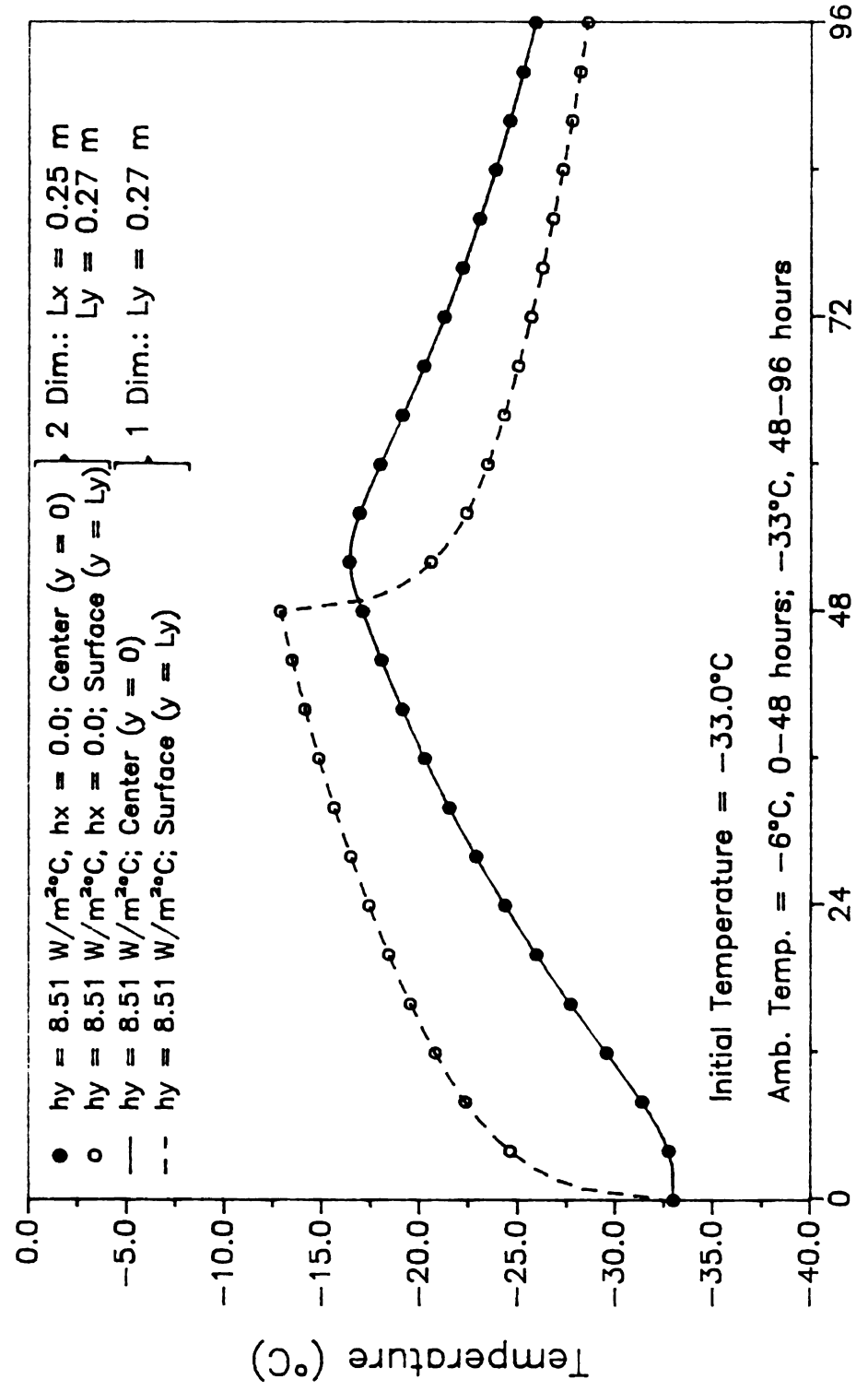


Figure 6.12b Verification of Two Dimensional Numerical Solution with hx = 0.0, hy = 8.51 $W/m^2°C$, with One Dimensional Numerical Solution with hy = 8.51 $W/m^2°C$.

changes in the surface heat transfer coefficients at 18 and 36 hours, as in Figures 6.4a-c, discussed previously as being characteristic of the solution method, were not included in the determination of the incrementally averaged values shown in Table 6.4.)

Numerical solutions were determined at the locations x = 0.063 m, y = 0.035 m, and x = 0.225 m, y = 0.19 m, which were the average thermocouple locations of the experimental measurements in Tests 3a-c. A time step of 60 seconds was used, with a spatial increment of 0.016 m, for Δx, and 0.017 m for Δy, based on the analysis given in Section 5.1.5. Thermal properties, given by Gutschmidt (1960) and Specht et. al. (1981) were again used to determine the thermal properties below the initial freezing point.

Results are shown in Figure 6.13 for the first repetition using the triple layer configuration with two exposed surfaces (Test 3a), and in Figures E.5a,b, for the second two repetitions of the test. Similar results were found in all repetitions. These results have the same characteristics as the results found for the double layer slab configuration with one exposed surface (Test 2a-c), shown in Figures 6.8, and E.4a,b. The temperatures at the location nearest the surface (x = 0.19 m, y = 0.225 m) were under-estimated during the first storage interval, and over-estimated during the second storage interval. At the end of 48 simulated storage hours, the numerical solution gave temperatures 1.5 C° lower than the experimental results at the same time and location. The numerical solution varied less than 1 C° from the experimental values at the location x = 0.063 m, y = 0.035 m. As was found in the comparisons shown in Figures 6.8 and E.4a,b, the numerical solution predicted a smaller variation in temperature between the locations shown, than the experimental results. The latter two hypothesis

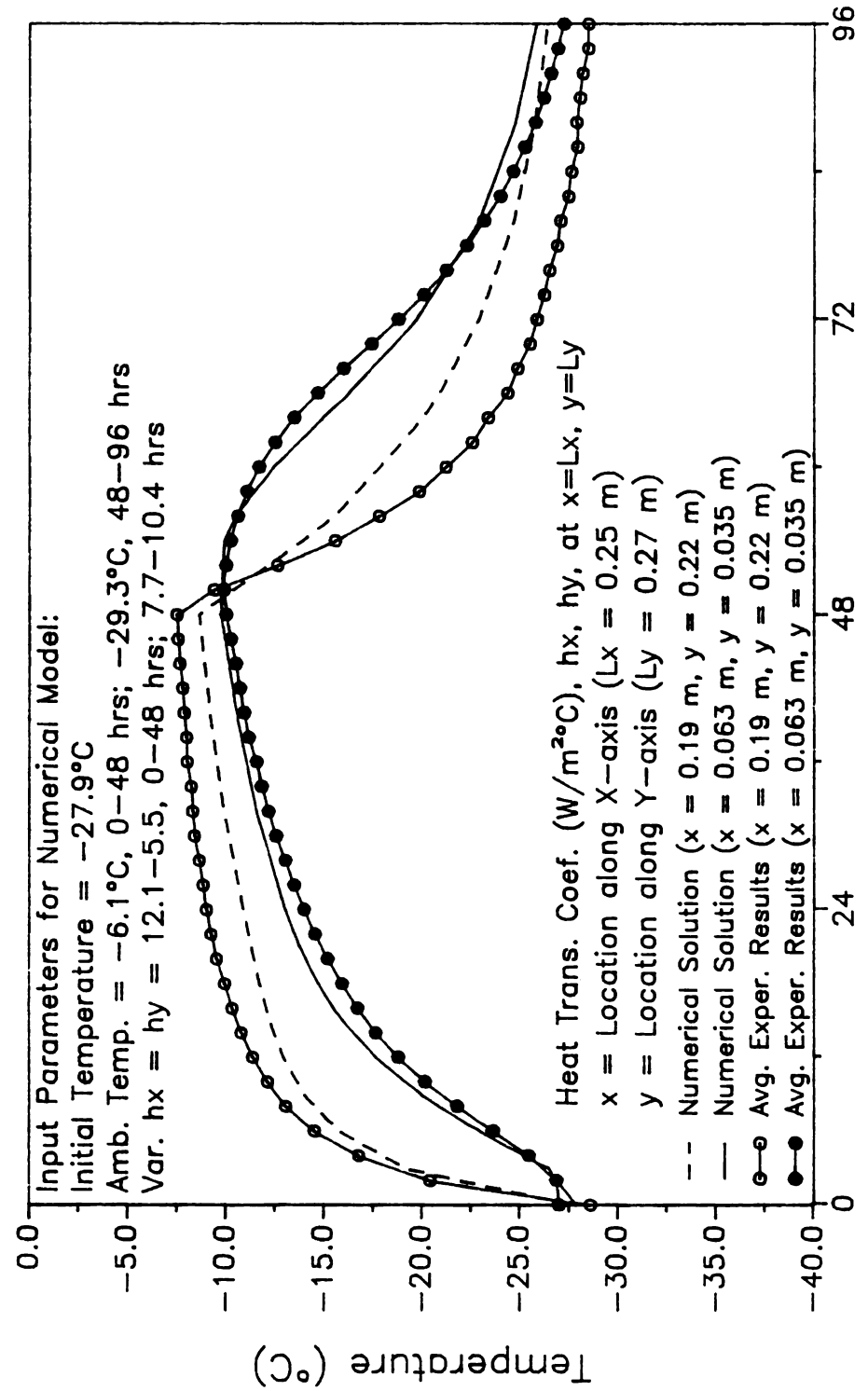


Figure 6.13 Two Dimensional Numerical Solution Compared to Experimental Results from Triple Layer Slab with Two Exposed Surfaces (Test 3a).

presented in Section 6.2.2 were proposed as explanations for the results.

In the first case, the assumption of negligible influence due to the packaging interface was hypothesized to be invalid. This hypothesis was tested in the same manner as was done in Section 6.2.2. An insulated boundary condition was imposed at the packaging interface around the corner methyl-cellulose box (x = 0.13, y = 0.18 m), and the numerical simulation was repeated using all other input values as before. Results are shown in Figure 6.14. As was found in the one dimensional comparison (Figure 6.10), the experimental results fell between the two numerical solutions with and without the insulated condition at the packaging interface. This indicated that the resistance due to the packaging interface was greater than zero, but less than infinity, again supporting the hypothesis that the assumption of no resistance to heat flow by the packaging material was invalid.

The third hypothesis, presented in Section 6.2.2, that the assumption of perfect insulation at the interior boundaries (x = 0, y = 0) was invalid, was tested by imposing a surface heat transfer coefficient of 1 $W/m^2 \cdot C$ along the these boundaries. Again, similar results were obtained as was found in Section 6.2.2. The estimated temperatures are compared with the experimental results in Figure 6.15. The solution nearest the interior boundary (x = 0.63 m, y = 0.35 m) over-estimated the experimental results during the first storage period and over estimated the results during the second storage period. Comparing Figures 6.13 and 6.15, the experimental values at x = 0.63 m, y = 0.35 m, were bounded by the numerical solutions using an insulated condition and a surface heat transfer coefficient of 1 $W/m^2 \cdot C$ at the boundary at x = 0, y = 0. This again supports the hypothesis that the heat transfer coefficient at the interior boundaries was somewhat greater than zero.

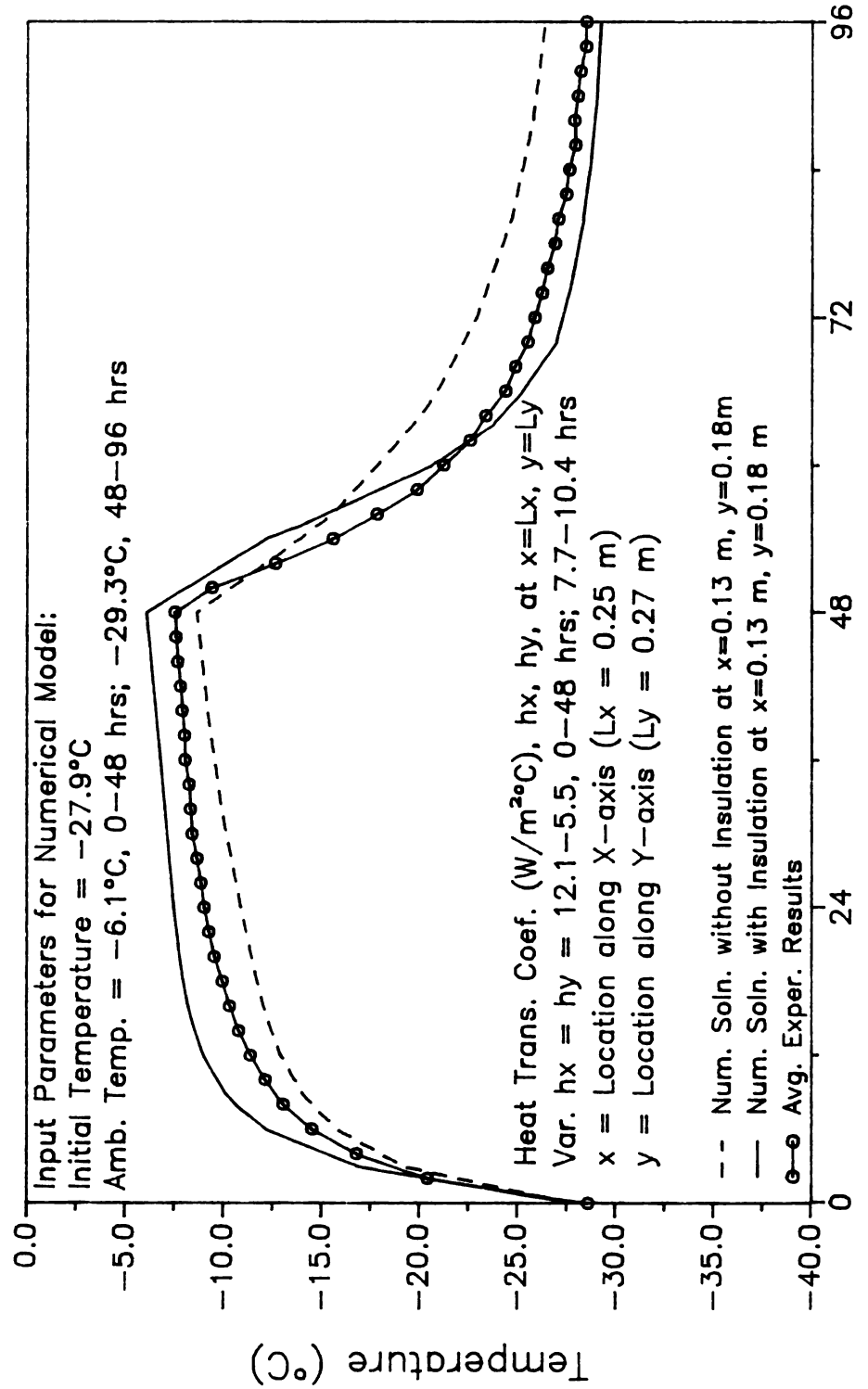


Figure 6.14 Two Dimensional Numerical Solution, with and without Imposed Insulated Boundary Condition at Packaging Interface Compared to Experimental results from Triple Layer Slab with Two Exposed Surfaces (Test 3a).

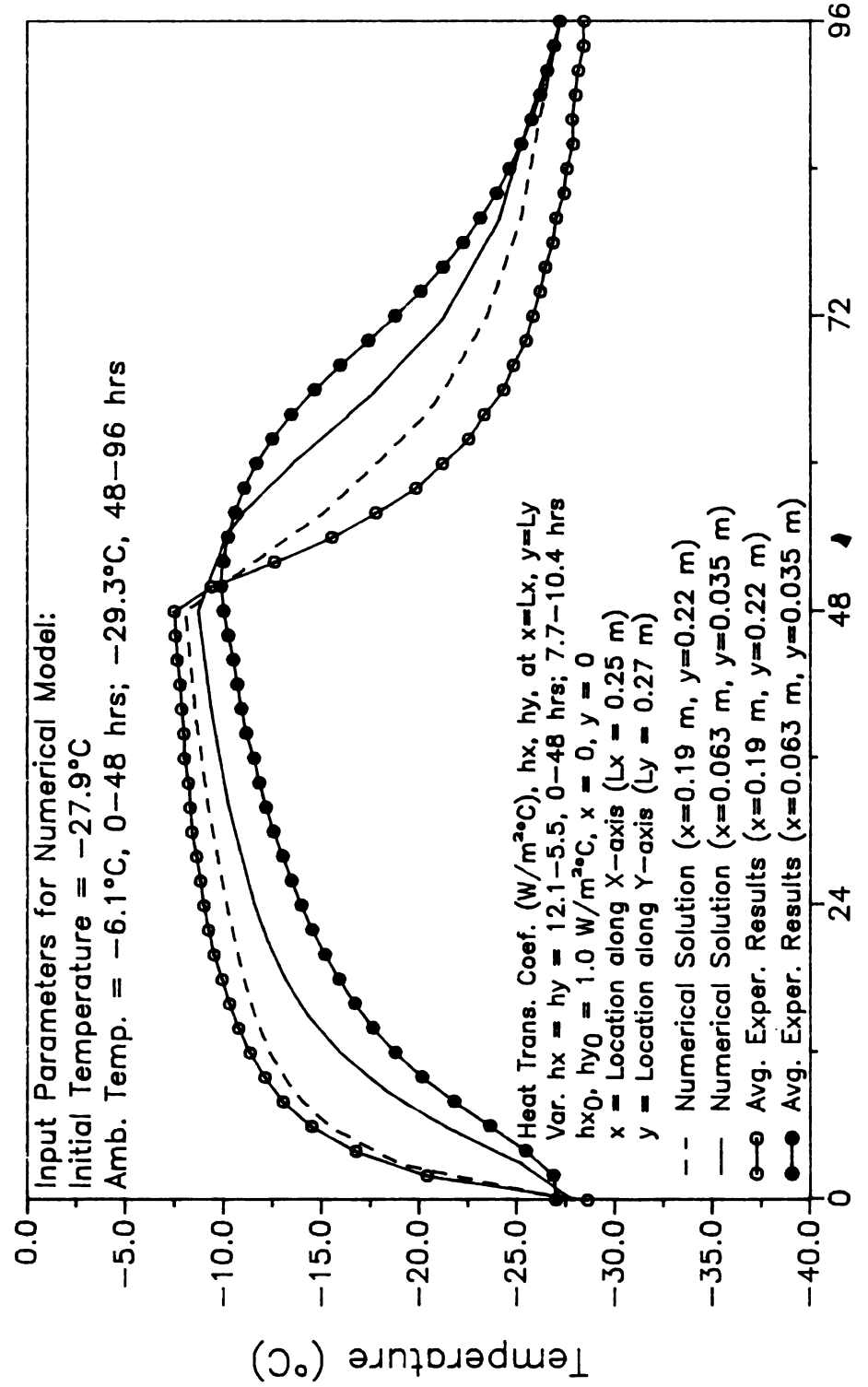


Figure 6.15 Two Dimensional Numerical Solution, with $hx_o = hy_o = 1\ W/m^2°C$, Compared to Experimental results from Triple Layer Slab with Two Exposed Surfaces (Test 3a).

Therefore, the results shown in Figures 6.14 and 6.15 suggest that both assumptions of negligible internal packaging resistance to heat transfer, and perfectly insulated interior boundaries were invalid, supporting the results found in Section 6.2.2. This again indicates that there is an need for further study in estimating the resistance to heat transfer due to the packaging interface.

6.4 Effects of the Magnitude of the Activation Energy Constant on the Mass Average Quality History

The degree of temperature dependence of quality degradation is dependent on the magnitude of the activation energy constant (Ea); typical values for food products range from 40 to 200 kJ/mole (Bonner, et. al., 1984). Two quality criteria are commonly used to determine the activation energy constant: (1) High Quality Life (HQL), and (2) Practical Storage Life (PSL). High Quality Life is defined as the length of product storage time until a detectable change in frozen food quality exists when compared to the same product stored at conditions where quality change is limited. Practical Storage Life is defined as the length of product storage time until the frozen food has unacceptable quality (Scott, et. al., 1984). Activation energy constants reported for HQL are generally higher than those reported for PSL (Bonner, et. al.,1984).

The effects of the magnitude of the activation energy constant (Ea) on the mass average rate of quality deterioration was demonstrated by comparing the predicted mass average quality histories for a range of values of Ea. One dimensional heat transfer was considered through a rectangularly shaped product mass, measuring two meters in the direction of heat transfer, and initially at a uniform temperature of -30°C. A surface heat transfer coefficient of 8.5 $W/m^2°C$ was imposed on the boundaries perpendicular to the direction of heat flow, and all other boundaries were considered to be perfectly insulated. The hypothetical food product was subject to storage conditions of 100 days at -5°C. The thermal properties of strawberries were used in generating the temperature distribution histories. These values are shown in Table 6.6. Note the thermal properties given for strawberries are very similar to those

Table 6.6 Thermal Properties of Unfrozen Strawberries.

Density (kg/m^3)	Thermal Conductivity (W/m°C)	Specific Heat (kJ/kg°C)
1040	0.54	3.93

(Heldman, 1982)

given for the Karlsruhe Test Substance. Quality distribution histories were determined using a hypothetical reference shelf-life of 500 days, and activation energy constants ranging from 20 to 200 kJ/mole.

The resulting quality was expressed as a percent of the initial shelf-life at the reference temperature, in this example, 50% quality would mean the product would have 50% of 500 days, or 250 days remaining storage life at -18°C (reference temperature). Results are shown in Figure 6.16 for activation energy constants of 0, 20, 40, 60, 100, 120, 140, 160, 180, and 200 kJ/mole. An activation energy constant of zero indicates no temperature dependence for the rate constant, and was used as the base line for comparison. From Figure 6.16, products with activation energy constants of sixty or less have little temperature dependence, products with activation energy constants from 60-100 kJ/mole have moderate temperature dependence, and products with Ea greater than 100 kJ/mole have high temperature dependence.

Strawberries were chosen as the food product considered in all of the subsequent quality analysis, since the activation energy constants found in the literature for the HQL and PSL criteria cover both extremities of the range for typical food products. Guadagui (1969) reported an activation energy constant (Ea) of 182.37 kJ/mole, with a reference shelf life of 630 days at -18°C for HQL of bulk frozen strawberries. Tressler, et. al. (1957), reported an activation energy constant of 49.13 kJ/mole, and a reference shelf life of 540 days at -18°C for the PSL of sliced frozen strawberries. The kinetic properties for strawberries are shown in Table 6.7.

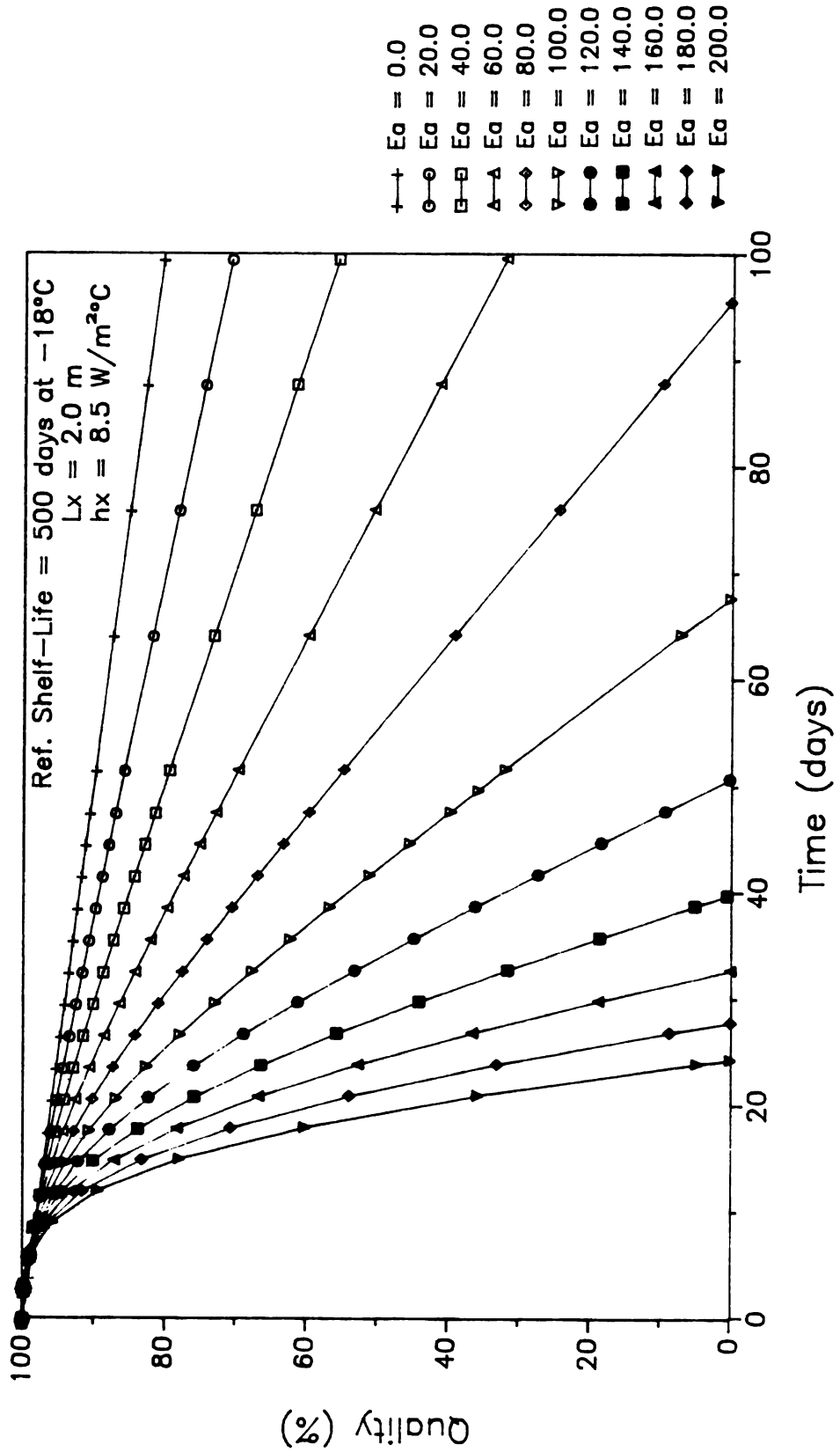


Figure 6.16 Effect of Magnitude of Activation Energy Constant, Ea (kJ/mole), on Food Quality Deterioration Rate, for a Product Initially at -30°C, Exposed to 100 days in Storage at -5°C.

Table 6.7 Kinetic Properties of Frozen Strawberries.

	Activation Energy Constant (kJ/mole)	Reference Shelf Life (days)	Reference Shelf Life Temperature (°C)
HQL Criterion[1]	182.37	630.0	-18
PSL Criterion[2]	49.13	540.0	-18

1. Guadagui, 1969.
2. Tressler, 1957.

6.5 Effects of Boundary Conditions on Temperature and Quality Histories of Frozen Foods during Storage.

Variations in boundary conditions, such as step changes in storage temperatures, or a change in the surface heat transfer coefficient, can change the rate of heat transfer, which consequently, change the temperature history of the food product. Since quality degradation was assumed to be temperature dependent (Eq. 3.49), variations in boundary conditions will also affect the quality profile within the product. It was desired to determine under what conditions the quality distribution history was affected the most. Several different parameters affecting the boundary conditions were investigated: (1) the magnitude of the surface heat transfer coefficient; (2) the storage time interval for step changes in the ambient storage temperature; and, (3) the magnitude of the storage temperature, and the amplitude of step changes in the storage temperature.

One dimensional heat transfer through a rectangularly shaped food product of thickness, Lx, equal to 2.0 m, was considered. The initial temperature was uniform and equal to -30°C in all cases. This might correspond to a pallet load of frozen foods with two non-adjacent exposed outer surfaces, and perfectly insulated on all other sides.

The total simulated storage time was set at 100 days. From Tables 5.4a,b, only one eigenvalue was found to be significant after approximately two hours, and the numerical approximation for the first eigenvalue was very accurate, even for large Δx (Figure 5.1). Therefore, only 25 nodes ($\Delta x = 0.042$ m) were used for the long storage time solutions. A time step of 3600 seconds was used; this value is near the upper end of the accuracy criterion, shown in Table (5.5), for thermal properties evaluated at -12°C, and it is much higher than the

time step criterion requirement for no oscillations (Tables 5.2a,b). Use of this time step and spatial increment was justified by comparison with results using 32 nodes and a time step of 600 seconds. No difference in the temperature distribution history was found after 50 hours of simulated storage time at -5°C, with the surface heat transfer coefficient, hx, equal to 8.5 $W/m^2°C$.

The influence of the magnitude of the surface heat transfer coefficient was demonstrated using a single storage interval at -5°C for 100 days. The effects of step changes in storage temperatures were investigated using nine different combinations of storage temperatures and storage time intervals. These nine cases are defined in Table 6.8, and will be referred to by their respective case number, for example, Case 1 refers to fluctuating storage conditions between one day storage periods at -5°C, and ten day storage periods at -30°C, for a total of 14 storage periods over 100 days.

6.5.1 Influence of the Surface Heat Transfer Coefficient on Temperature and Quality Distribution Histories.

The effects of the surface heat transfer coefficient (hx) on the temperature and quality distributions within a frozen food product were demonstrated by simulating storage conditions by using the extreme range of values for hx found in the literature (Dagerskog, 1974, and Zaritzky, 1982). For the upper limit of the range, hx = 20 $W/m^2°C$ was selected, and for the lower limit, hx = 1 $W/m^2°C$ was used. One dimensional heat transfer was simulated in a product, initially at -30°C, and subject to storage for 100 days at -5°C.

Results obtained using hx = 20 $W/m^2°C$ and hx = 1 $W/m^2°C$ were compared with the results found with hx = 8.5 $W/m^2°C$. (This is the overall

Table 6.8 Definition of Boundary Condition Cases, with Step Changes in Storage Temperatures over Given Storage Interval.

	High Temperature (°C)	Storage Period (days)	Low Temperature (°C)	Storage Interval (days)
Case 1	-5	1	-30	10
Case 2	-5	5	-30	10
Case 3	-5	10	-30	10
Case 4	-5	1	-18	10
Case 5	-5	5	-18	10
Case 6	-5	10	-18	10
Case 7	-13	1	-18	10
Case 8	-13	5	-18	10
Case 9	-13	10	-18	10

average surface heat transfer coefficient determined from the experimental results, described in Section 6.1.2.) The effects of the surface heat transfer coefficient on the temperature history at the exposed surface and at the geometric center are shown in Figure 6.17.

The temperature differential between the two surfaces was very small for hx = 1.0 $W/m^2°C$, compared with the results shown for hx = 8.5 $W/m^2°C$, and hx = 20.0 $W/m^2°C$, during the first 20 days of the simulated storage period. However, as the magnitude of the surface heat transfer coefficients increased, the solution approached lumped steady state conditions (constant temperature with time and position) faster.

In addition, the shape of the curves changed dramatically by increasing the surface heat transfer coefficient from hx = 1.0 $W/m^2°C$, to hx = 8.5 $W/m^2°C$, compared with the change in the shape of the curves shown by increasing hx from 8.5 to 20.0 $W/m^2°C$. This indicates that a change in magnitude of the surface heat transfer coefficient has a greater influence on the solution at lower values of hx than at higher values of hx.

The effects of the magnitude of the surface heat transfer coefficient on the quality distribution histories, determined for strawberries from Eq. (3.49), are shown in Figure 6.18 for the criterion with Ea = 182 kJ/mole, and reference shelf-life of 630 days at -18°C, and in Figure 6.19, for the criterion with Ea = 49 kJ/mole, and reference shelf-life of 540 days at -18°C. The high activation energy constant resulted in a high temperature dependence of the quality deterioration rate. All of the simulations using the high activation energy constant resulted in zero shelf life by the end of the 100 day storage period. The shelf life was completely diminished by the end of 79 days storage time for the case where the surface heat transfer coefficient equaled

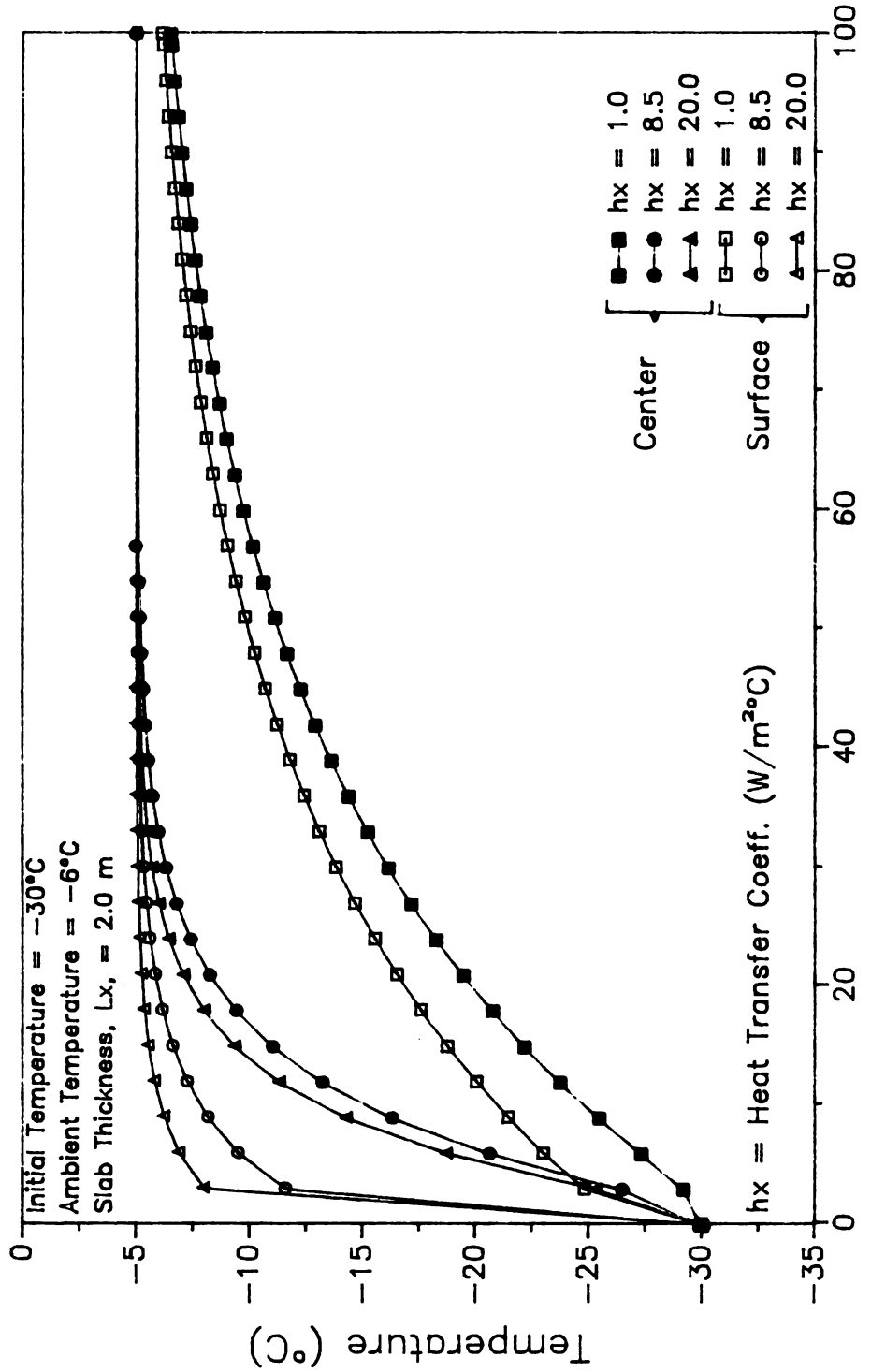


Figure 6.17 Effect of Surface Heat Transfer Coefficient, hx, on Product Temperature History at Geometric Center and Exposed Surface.

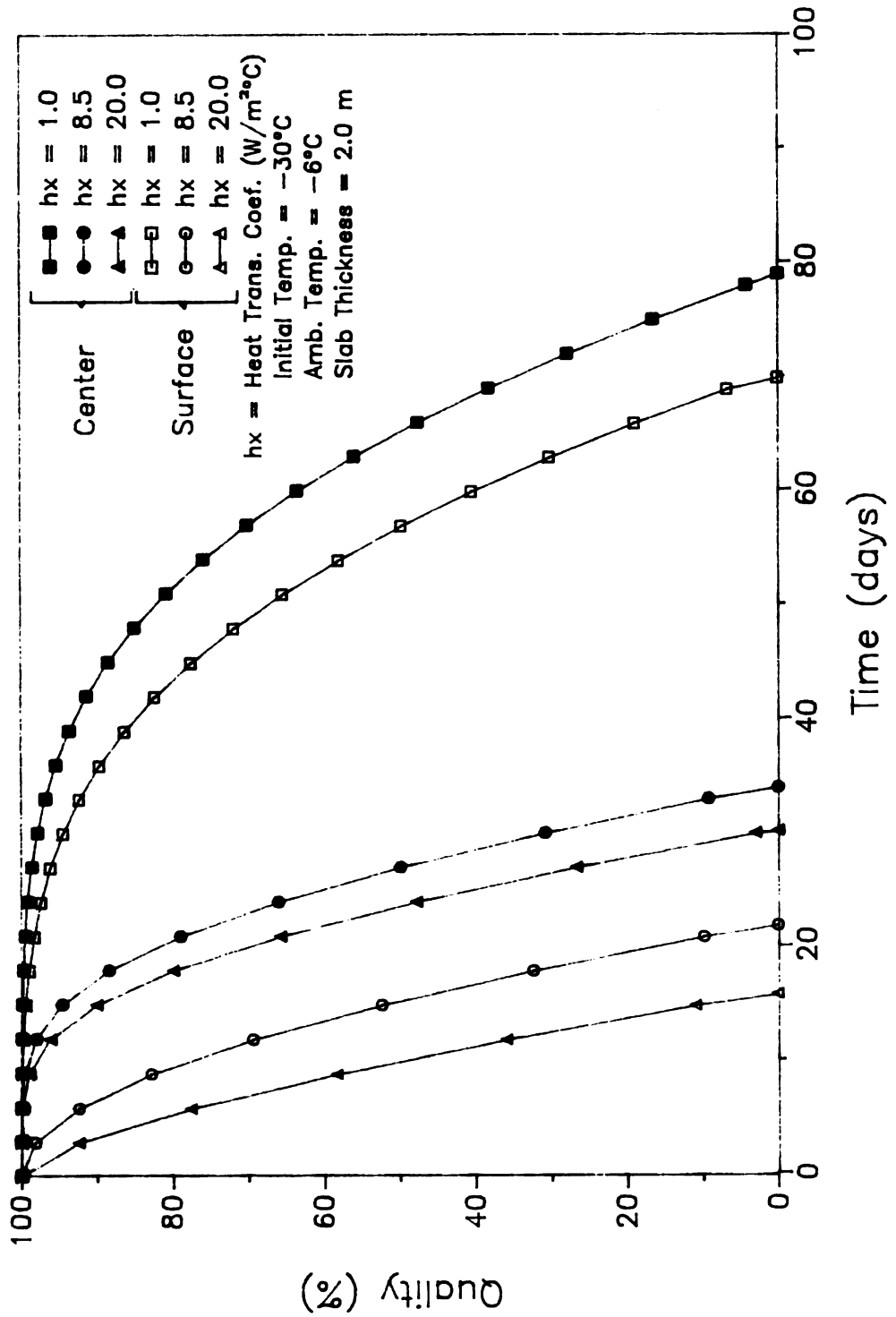


Figure 6.18 Effect of Surface Heat Transfer Coefficient, hx, on Product Quality Deterioration Rate at Geometric Center and Exposed Surface (Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

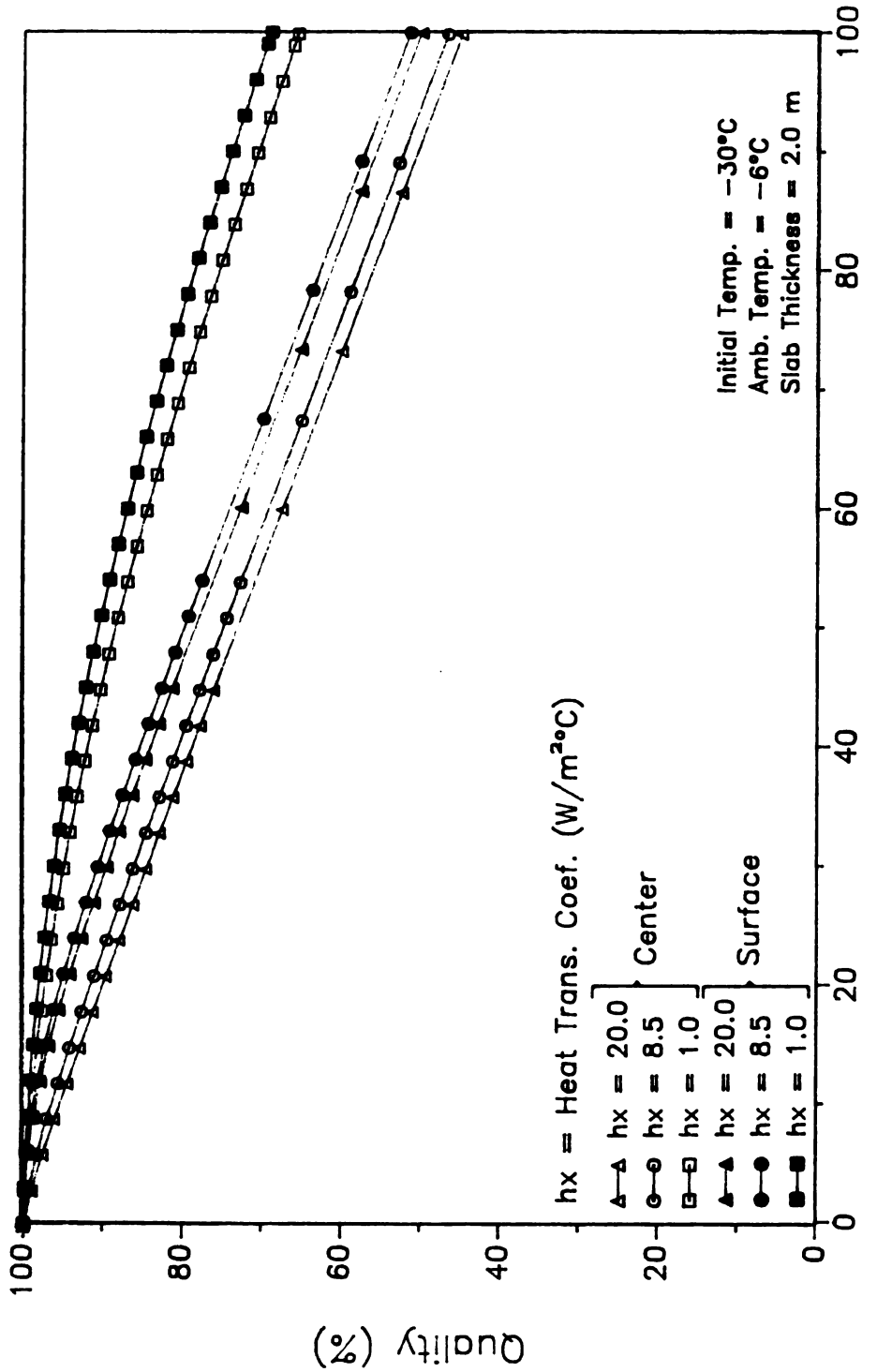


Figure 6.19 Effect of Surface Heat Transfer Coefficient, hx, on Product Quality Deterioration Rate at Geometric Center and Exposed Surface (Ea = 49 kJ/mole, Shelf-life at -18°C = 540 days).

1.0 $W/m^2°C$, while for the cases where the surface heat transfer coefficient equaled 8.5 $W/m^2°C$ and 20.0 $W/m^2°C$, shelf-life was diminished by the end of 35 and 21 days, respectively.

The difference in quality of the product at the outer surface compared to the quality at the geometric center increased as the surface heat transfer coefficient was increased. When the high activation energy constant was considered, the differences in quality between the two surfaces were substantial: ninety percent of the quality at the center of the product mass was retained at the time when the quality at the surface was diminished, using hx = 20.0 $W/m^2°C$, while 77% of the initial quality was retained using hx = 8.5 $W/m^2°C$, and 33% of the quality remained at the center using hx = 8.5 $W/m^2°C$.

Due to the lower temperature dependence on the quality deterioration rate for the criterion with the low activation energy constant, there was very little difference between each of the solutions using hx = 1, 8.5, and 20 W/m°C, and even smaller variations in the product quality between the center and the exposed outer surfaces, as shown in Figure 6.19. In none of the cases was the product shelf life exceeded, and the difference in quality between the center and the outer boundary at the end of 100 days storage was less than 5% of the initial quality, for all cases.

In summary, changes in the surface heat transfer coefficient at low values ($\leq$ 8.5 $W/m^2°C$) had more influence on the temperature distribution than equivalent changes at higher magnitudes ($\geq$ 8.5 $W/m^2°C$). High values for the heat transfer coefficient yielded solutions which had a greater temperature differential between the two boundaries initially, but approached the steady state solution rapidly. The effects of the magnitude of the surface heat transfer coefficient were strongly dependent on the magnitude of the activation energy constant. For low values

of Ea ($\leq$ 50 kJ/mole), increasing the surface heat transfer coefficient increased the quality degradation rate moderately, but with only small variations in quality within the product (Lx $\leq$ 2.0 m). On the other hand, for high activation energy constants ($\geq$ 180 kJ/mole), increasing the surface heat transfer coefficient decreased product shelf life substantially, and resulted in high quality variations within the product.

6.5.2 Effects of the Frequency of Step Changes in Storage Temperatures on Temperature and Quality Distribution Histories.

The effects of the fluctuation frequencies of step changes in the storage conditions were investigated using three different intervals for the step changes: (1) one day at -5°C, and ten days at -30°C (Case 1); (2) five days at -5°C, and ten days at -30°C (Case 2); and, (3) ten days at -5°C, and ten days at -30°C (Case 3). Each of these cycles were repeated for a total storage time of 100 days. Again, one dimensional heat transfer was considered through a 2.0 m product, with an initial temperature of -30°C, and with a constant heat transfer coefficient of 8.5 $W/m^2°C$ on the exposed boundaries.

The resulting temperature histories are shown in Figure 6.20a, at the geometric center, and in Figure 6.20b, at the outer surface. For the boundary conditions described in Case 1, the temperature at the geometric center changed only slightly, while the temperature at the outer surface changed significantly (14°C), but only for a very short period of time, compared to the overall product shelf life. Using the boundary conditions described in Case 2 resulted in temperature fluctuations at the center of 6°C, and fluctuations of 19°C at the outer

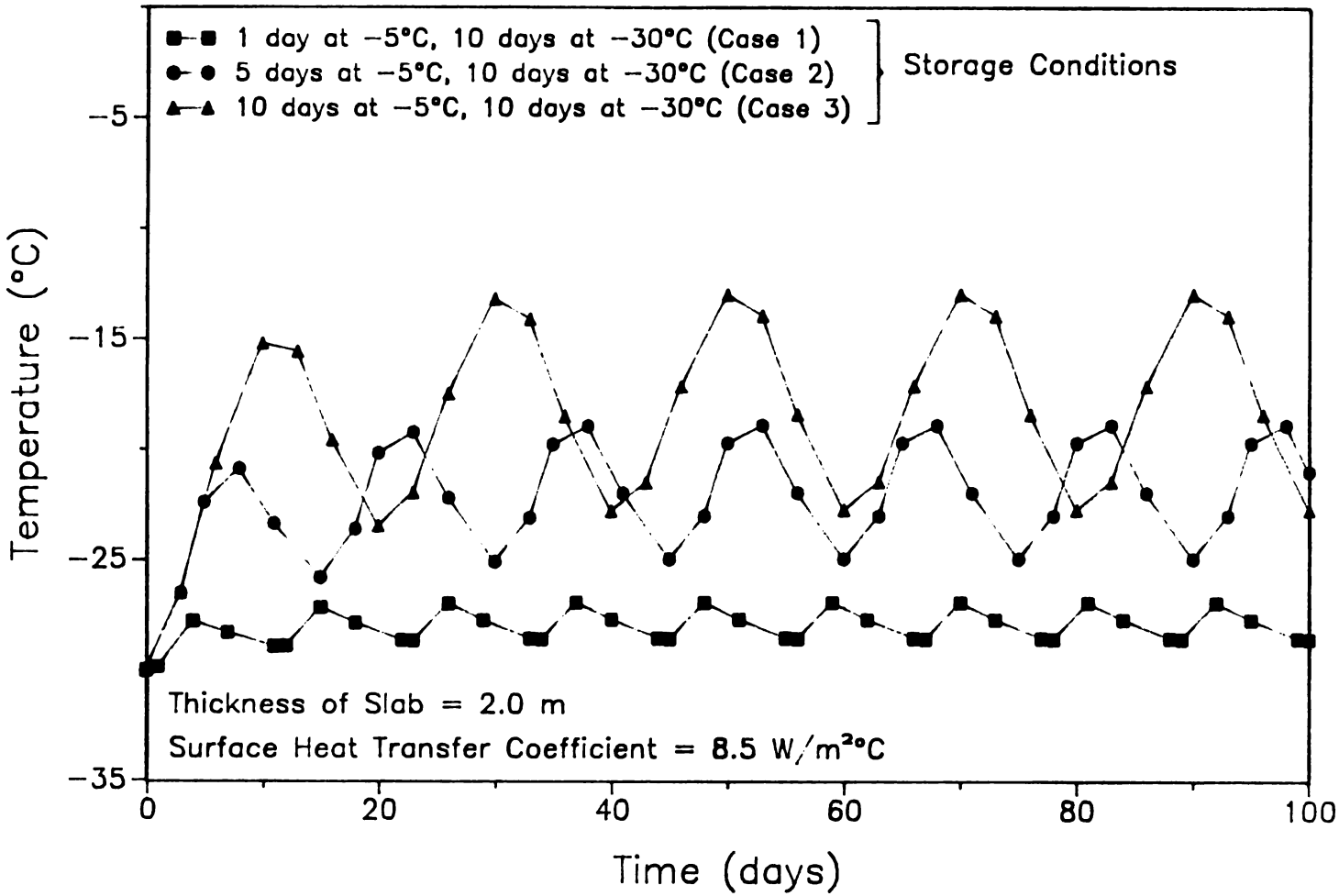


Figure 6.20a Effect of Duration of Step Changes in Storage Temperatures on Product Temperature History at Geometric Center, for a Product with Initial Temperature of -30°C (Cases 1, 2 and 3).

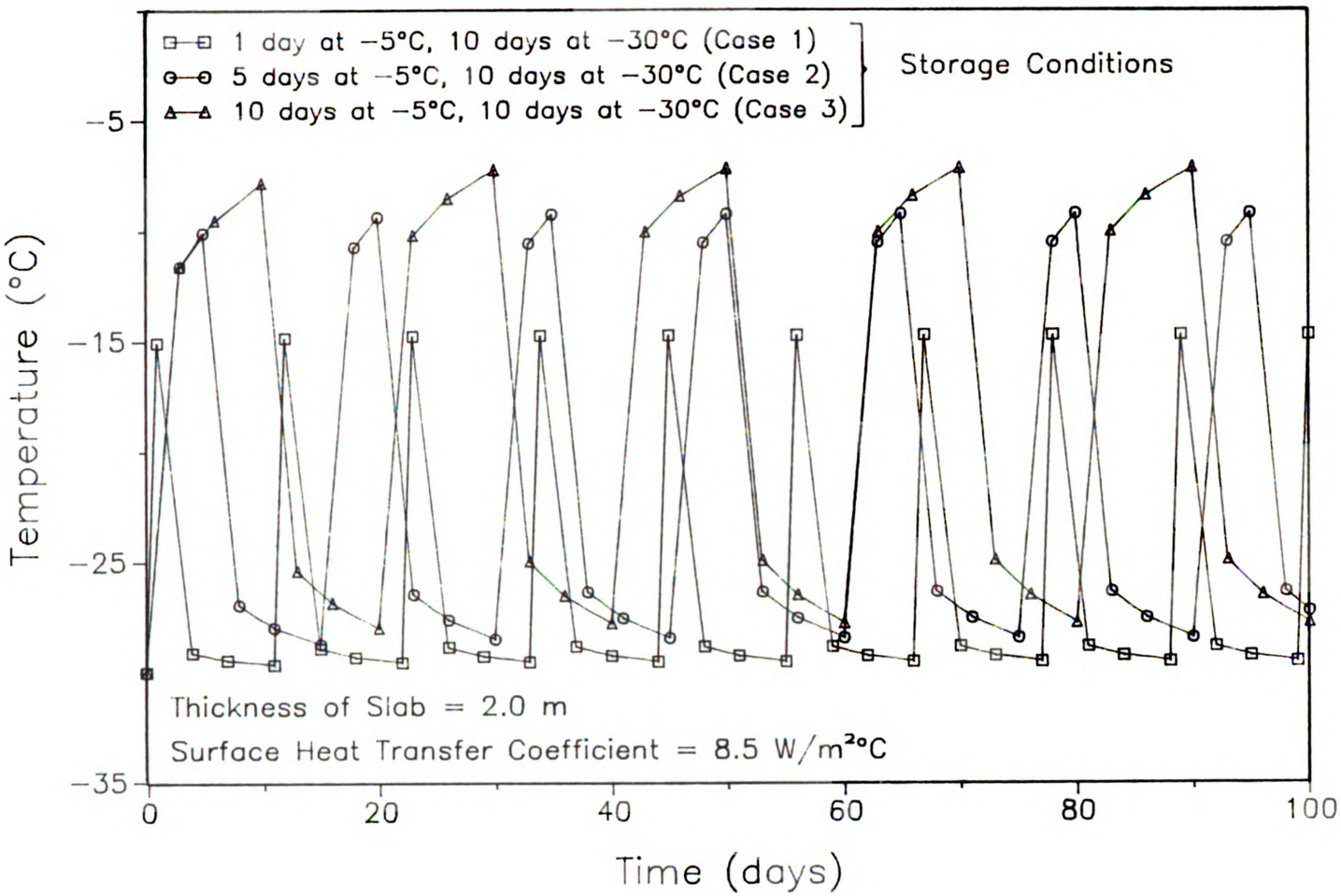


Figure 6.20b Effect of Duration of Step Changes in Storage Temperatures on Product Temperature History at Exposed Surface, for a Product with Initial Temperature of -30°C (Cases 1, 2 and 3).

surface, for a slightly longer period of time. When the storage intervals were equal for each storage temperature (Case 3), fluctuations of 10°C were found at the center, and fluctuations of 20°C were found at the exposed surface. The mass average product temperature increased from 29°C for Case 1, and 23°C for Case 2, to 18°C for Case 3.

The effects of these step changes in storage conditions on the quality histories at the geometric center and at the outer surface are shown for the criterion with an activation energy constant of 182°C in Figure 6.21, and for the criterion with an activation energy constant of 49°C in Figure 6.22. In the cases where the storage intervals were less than or equal to five days at -5°C (Cases 1 and 2) the step changes in the storage conditions had very little effect on the quality history at the center for both high and low activation energy constants. The effect of the short high temperature interval (Case 1) on the quality at the exposed surface was also very small, with very little variation in quality within the product mass. The steeper slope of the curves resulting from storage conditions of one day at -5°C (Case 1) in Figure 6.22, compared to the corresponding curves in Figure 6.21, were a result of the shorter reference shelf life for the PSL criterion (540 days), compared with the HQL criterion (630 days).

Increasing the storage time interval at the higher storage temperature to five days (Case 2) resulted in a substantial loss of quality at the outer surface (50%), with very little change in quality at the geometric center, using the quality criterion with the higher activation energy constant. For the quality criterion with an activation energy constant of 49 kJ/mole, there was a slight drop (8%) in overall quality, compared to the results using one day at -5°C (Case 1), and only a 4% change in quality between the geometric center and the outer surface.

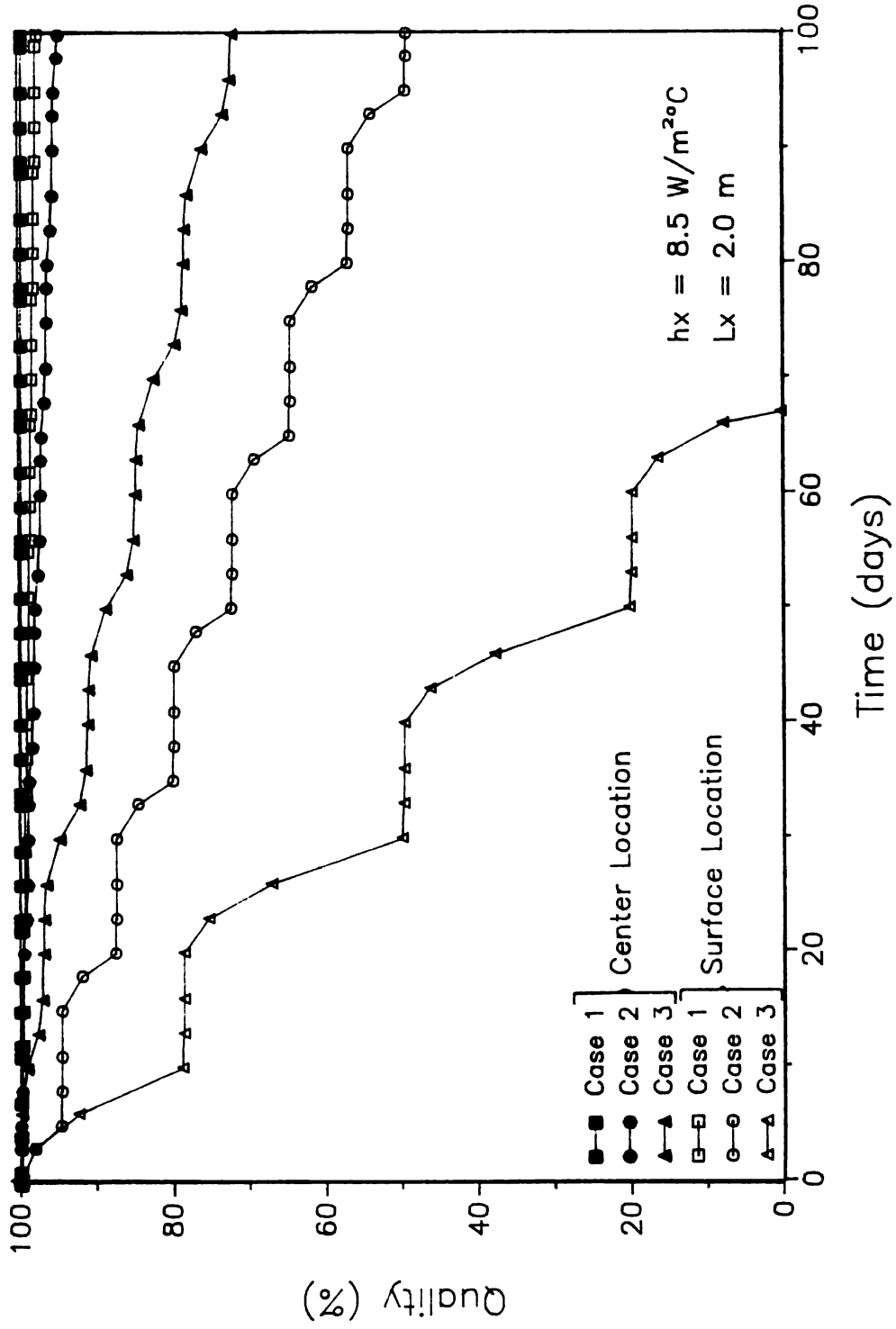


Figure 6.21 Effect of Duration of Step Changes in Storage Temperatures on Product Quality Deterioration Rate, for a Product with Initial Temperature of -30°C (Cases 1, 2 and 3; Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

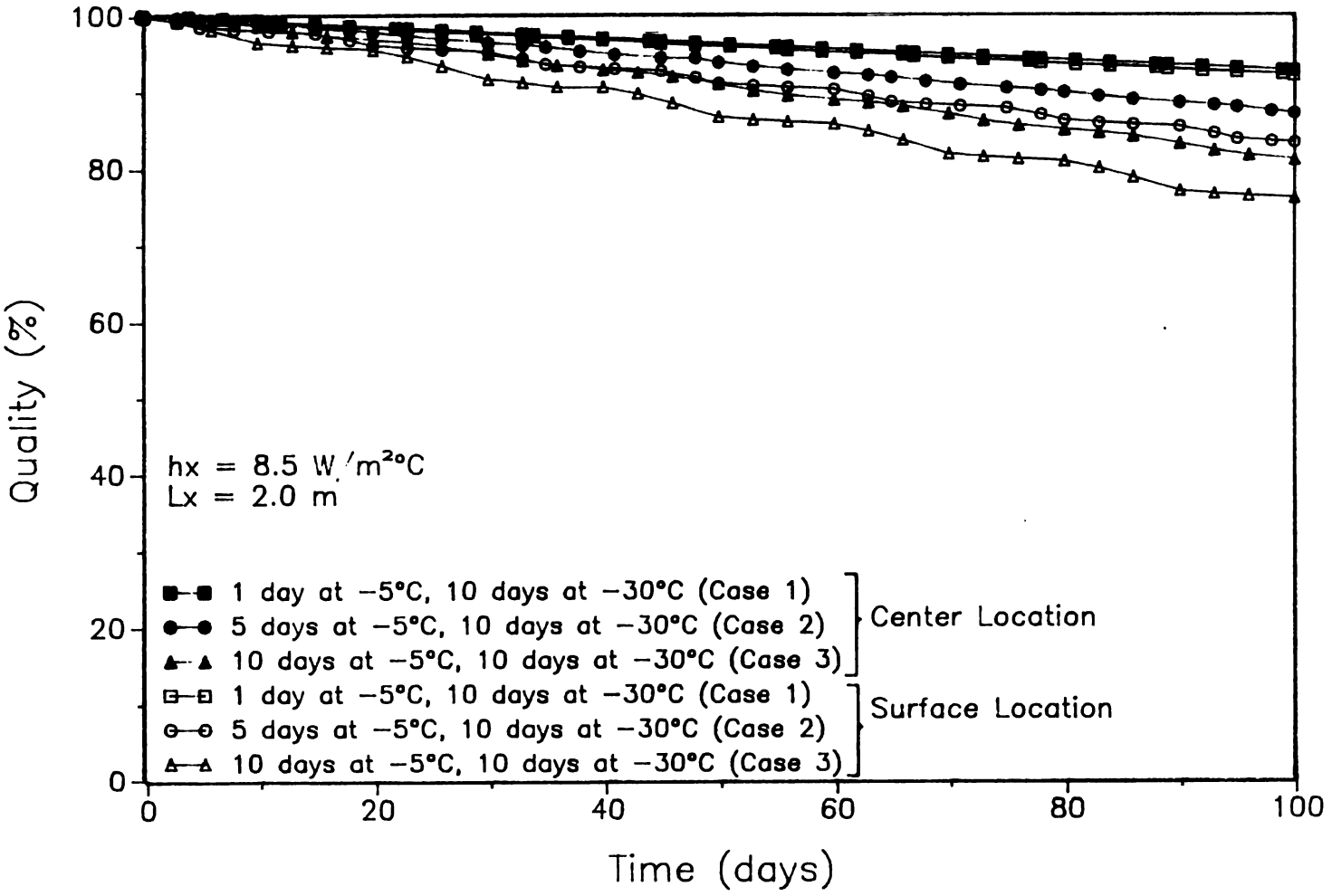


Figure 6.22 Effect of Duration of Step Changes in Storage Temperatures on Product Quality Deterioration Rate, for a Product with Initial Temperature of -30°C (Cases 1, 2 and 3; Ea = 49 kJ/mole, Shelf-life at -18°C = 540 days).

In Case 3, where the storage interval at -5°C was equal to the interval at -30°C, there was significant quality loss for the higher activation energy constant at both the geometric center and the outer surface, and a substantial difference in quality between the two locations. The product quality was diminished at the surface after 67 days of storage, while the product quality at the center was only reduced to 84% of its initial value.

When the lower activation energy constant was used (Figure 6.22), there was also a decrease in overall product quality and an increase in the quality differential within the product, but the degree of change was much smaller in both cases. Product quality at the surface dropped to 75% of its initial value, compared with 93% remaining quality after 100 days in storage, using step change intervals of only one day at -5°C (Case 1). Again, the quality differential within the product was less than 5% of its initial quality.

To summarize, increases in ambient temperature for short time intervals, compared with the overall storage time at the lower temperature ($\leq$ 10% overall storage time), had negligible effects on overall product quality. Step changes in storage conditions with equivalent storage time intervals at high and low storage temperatures had much greater influence on quality reduction, especially when using high activation energy constants. Step changes in the storage temperatures with the storage interval at -5°C equal to one half the interval at -30°C, resulted in substantial reductions in quality at the surface, with high quality variations within the product, when using high activation energy constants. However, when using low activation energy constants under the same conditions, only small changes in both the overall quality and the internal quality variation were observed.

6.5.3 Effects of Ambient Temperature during Step Changes in Storage Conditions on Temperature and Quality Distribution Histories.

The effect of the magnitude of the ambient storage temperature in fluctuating storage conditions was demonstrated using the same three repeating step change intervals described by Cases 1, 2 and 3 (Table 6.8), but with different magnitudes for the ambient temperatures. Three different sets of high and low ambient temperature were considered for these three step change cycles (for a total of nine cases): (1) low storage temperature = -30°C, high storage temperature = -5°C, (same temperatures used in Cases 1, 2, and 3); (2) low storage temperature = -18°C, high storage temperature = -5°C, (Cases 4, 5, and 6); and, (3) low storage temperature = -18°C, high storage temperature = -13°C, (Cases 7, 8, and 9). These cases are also defined in Table 6.8. The same product geometry, initial temperature, and surface heat transfer coefficients described in Section 6.4.2 were also used in this analysis.

The temperature histories at the geometric center and at the outer surface for the step change intervals of one day at -5°C and ten days at -30°C (Case 1), one day at -5°C and ten days at -18°C (Case 4), and one day at -13°C and ten days at -18°C (Case 7), are shown in Figures 6.23a,b. The mean temperatures were higher for both Cases 4 and 7 due to the increase in the minimum storage temperature. The temperature at the geometric center fluctuated slightly (1°C) using Case 4 boundary conditions, and negligibly using Case 7. The outer surface temperature fluctuated only 2°C.

The effects of the changes in ambient temperature on the quality deterioration rate for the HQL and PSL criteria are shown in Figures 6.24 and 6.25, respectively. Note that the scale for % Quality ranges form 60 to 100%, instead of 0 to 100%. The solid boxes represent the

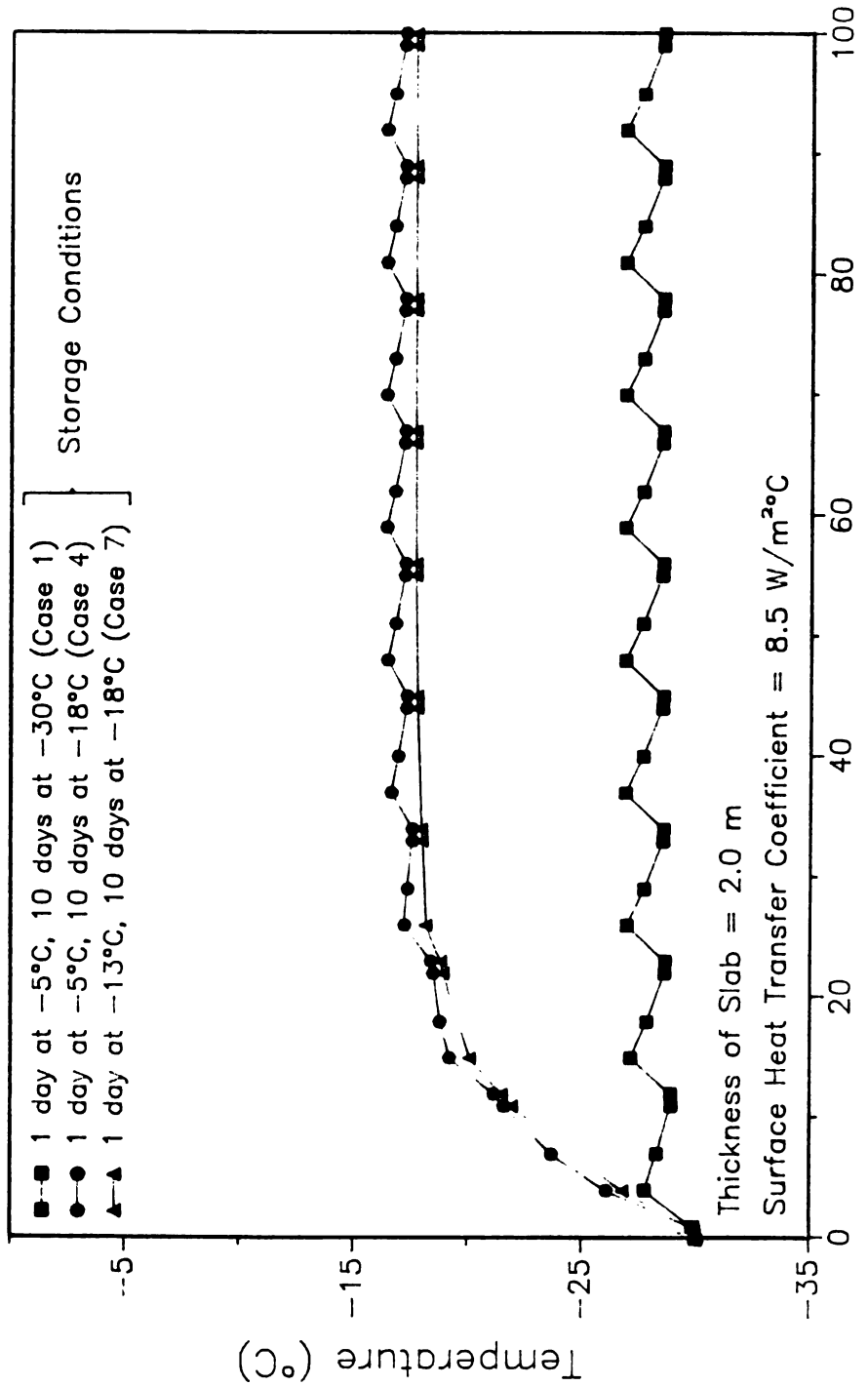


Figure 6.23a Effect of Magnitude of Step Change in Storage Temperatures on Product Temperature History at Geometric Center, for a Product with Initial Temperature of -30°C (Cases 1, 4 and 7).

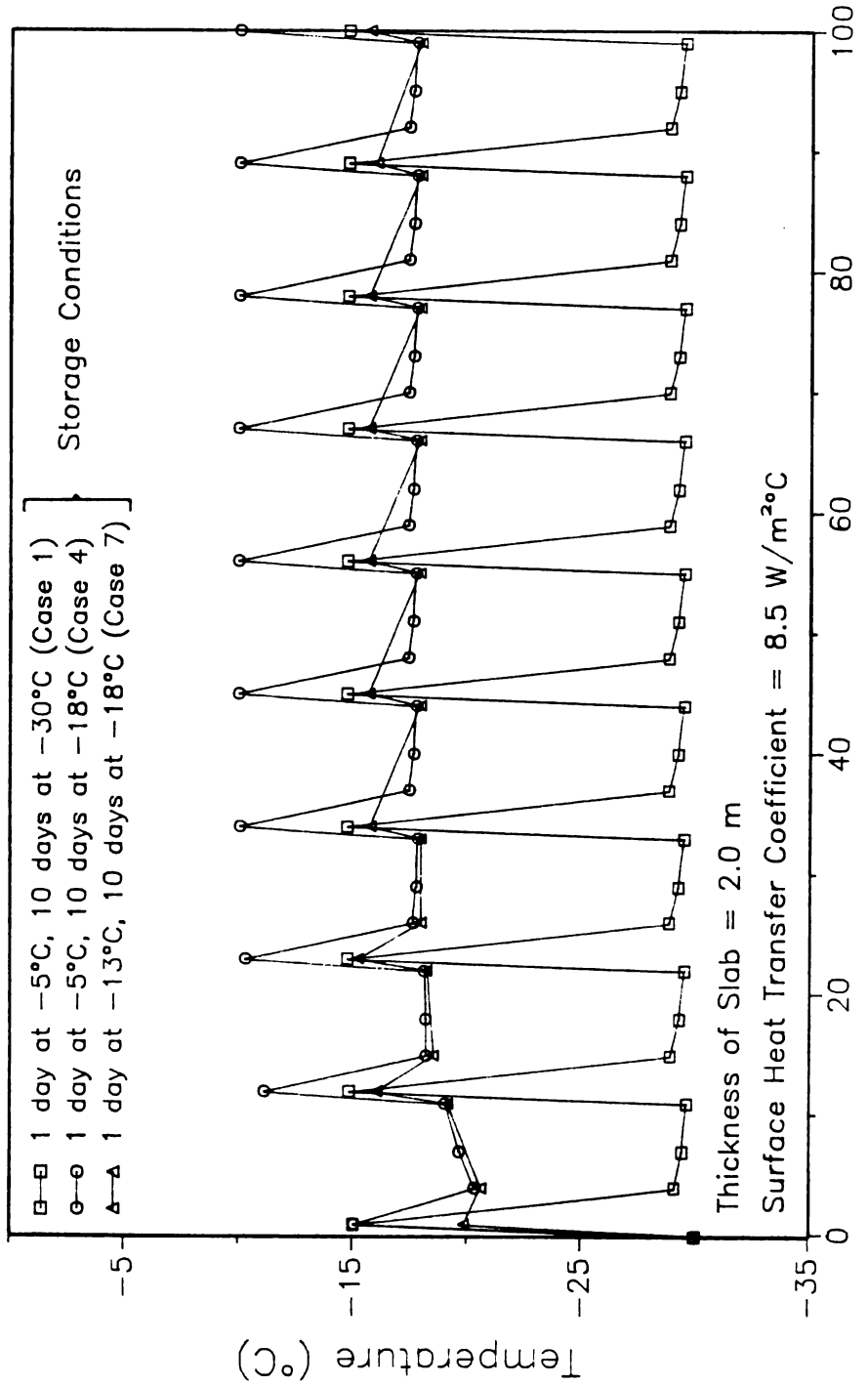


Figure 6.23b Effect of Magnitude of Step Change in Storage Temperatures on Product Temperature History at Exposed Surface, for a Product with Initial Temperature of -30°C (Cases 1, 4 and 7).

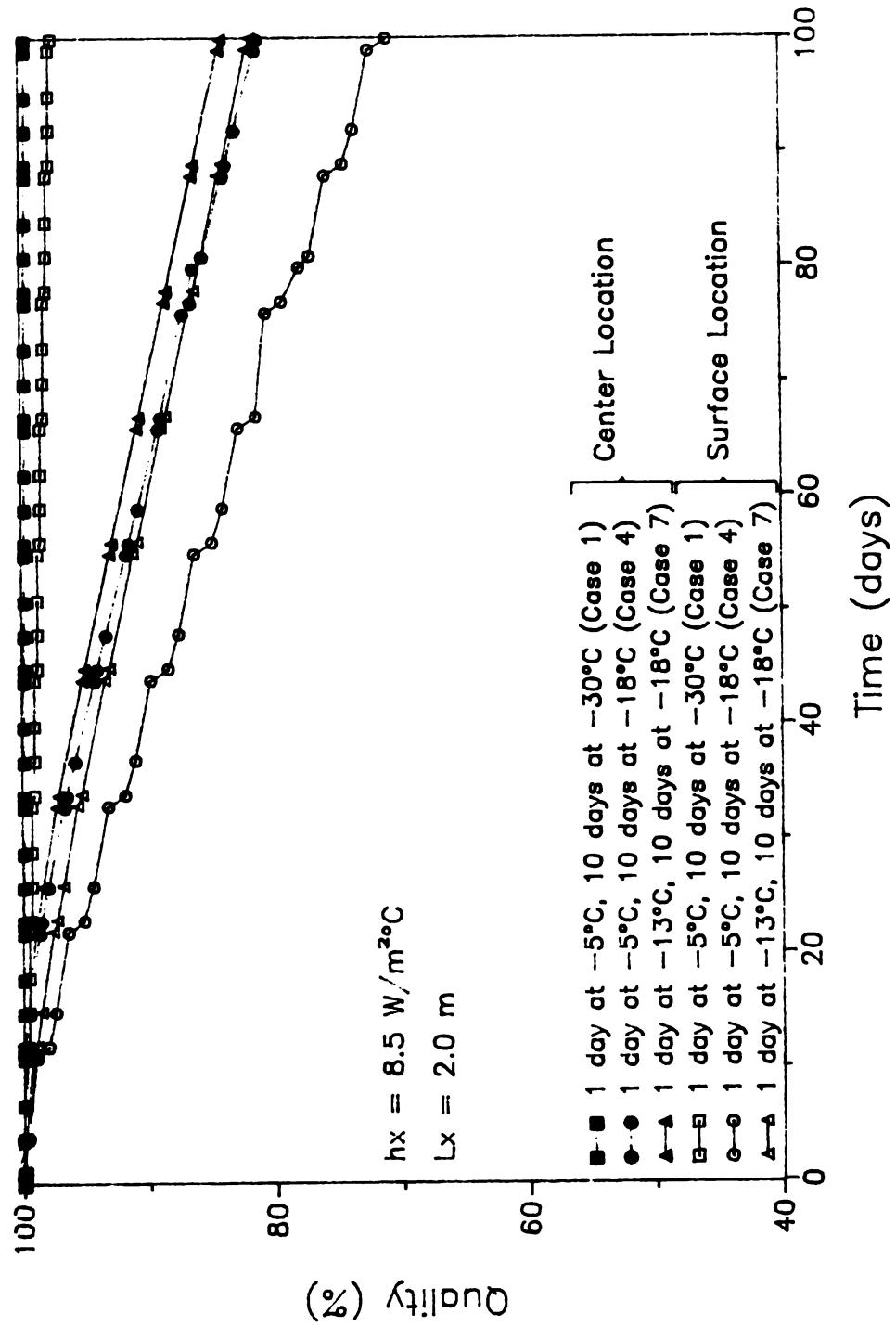


Figure 6.24 Effect of Magnitude of Step Change in Storage Temperatures on Product Quality Deterioration Rate, for a Product with Initial Temperature of -30°C (Cases 1, 4 and 7; Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

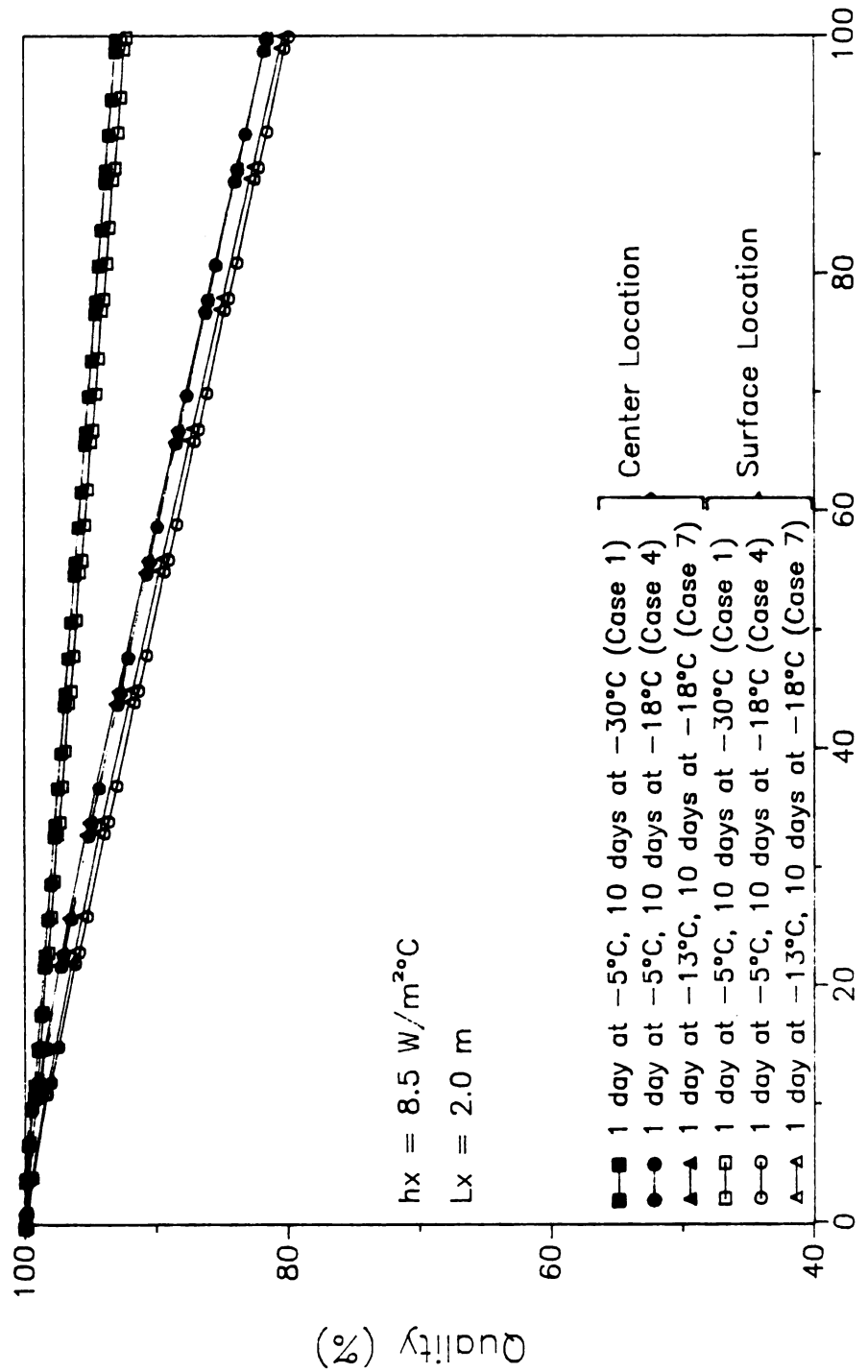


Figure 6.25 Effect of Magnitude of Step Change in Storage Temperatures on Product Quality Deterioration Rate, for a Product with Initial Temperature of -30°C (Cases 1, 4 and 7; Ea = 49 kJ/mole, Shelf-life at -18°C = 540 days).

results of using step changes in the boundary conditions of one day at -5°C and ten days at -30°C (Case 1) in both figures, and are identical to the corresponding curves shown in Figures 6.21 and 6.22. Both conditions of step changes of one day at -5°C and ten days at -18°C (Case 4), and one day at -13°C and ten days at -18°C (Case 7) resulted in a lower quality retention at the end of 100 days, compared to the results using Case 1 (one day at -5°C and ten days at -30°C) results, because of the overall higher mean storage temperature.

Step changes in storage temperatures between -5°C and -18°C (Case 4) resulted in a 10% quality differential across the two surfaces using the activation energy constant of 182 kJ/mole, and 1.5% using the lower activation energy constant. The results from using temperatures fluctuating between -18°C and -13°C (Case 7) indicated very little influence on the quality distribution history in all cases. The quality differential between the center and the surface was 2% using the high activation energy constant, and 1% using the low activation energy constant.

The same sets of ambient temperatures (-5°C and -30°C, -5°C and -18°C, and -13°C and -18°C) were used in Cases 2, 5, and 8, but with storage periods alternating between five days at the higher storage temperature, and ten days at the lower storage temperature. The solutions for the temperature histories at the geometric center and at the outer surface are shown in Figures 6.26a,b. The solutions for Case 2 (five days at -5°C and ten days at -30°C) in both figures are the same as the corresponding curves in Figures 6.20a,b. The temperatures at the geometric center fluctuated approximately 3°C for the conditions where the high storage temperature equaled -5°C, and the low storage temperature equaled -18°C, (Case 5) and only 1°C using -13°C for the high storage temperature, and -18°C for the low storage temperature. The temperature fluctuation at the outer surface was substantial in all

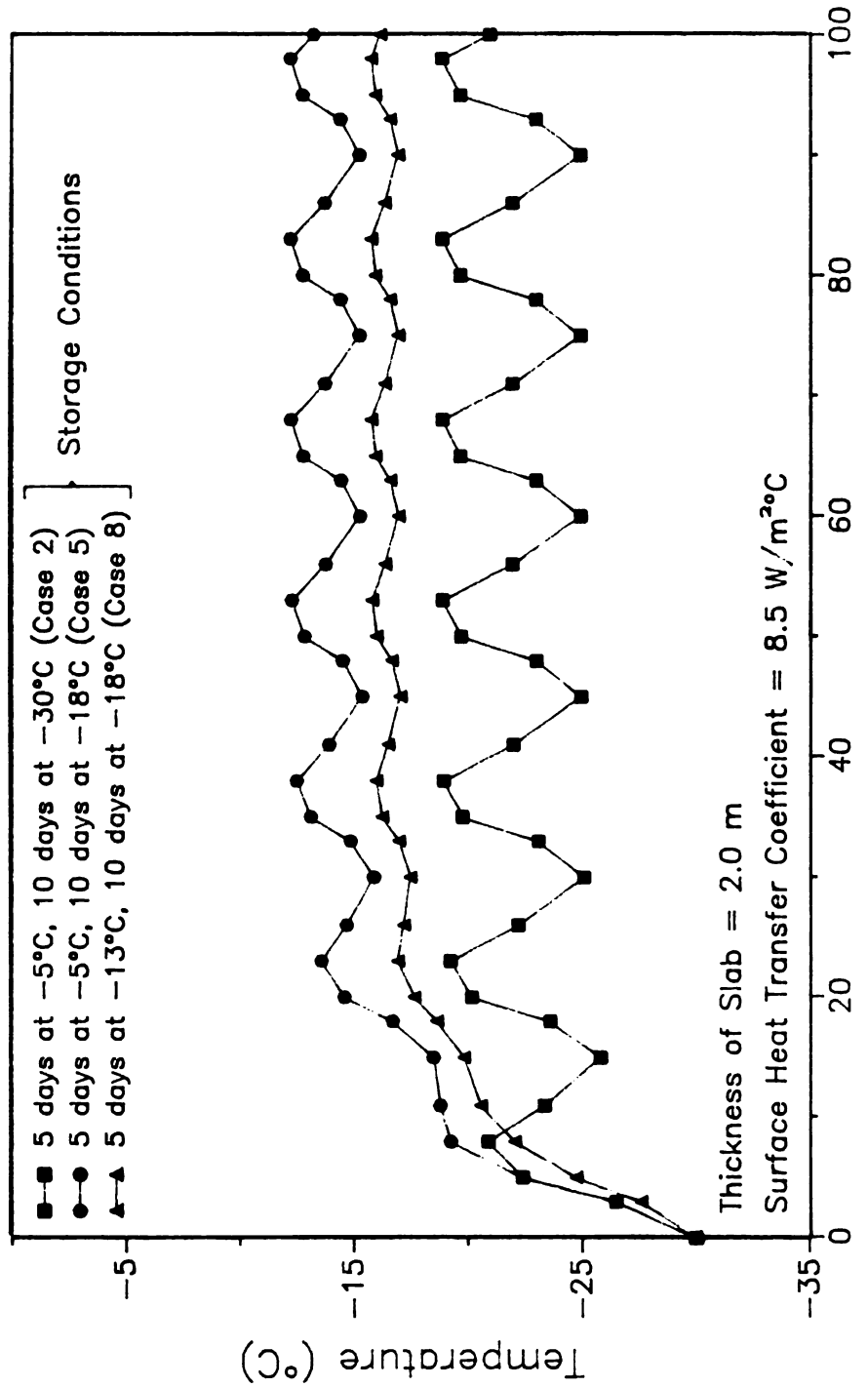


Figure 6.26a Effect of Magnitude of Step Change in Storage Temperatures on Product Temperature History at Geometric Center, for a Product with Initial Temperature of -30°C (Cases 2, 5 and 8).

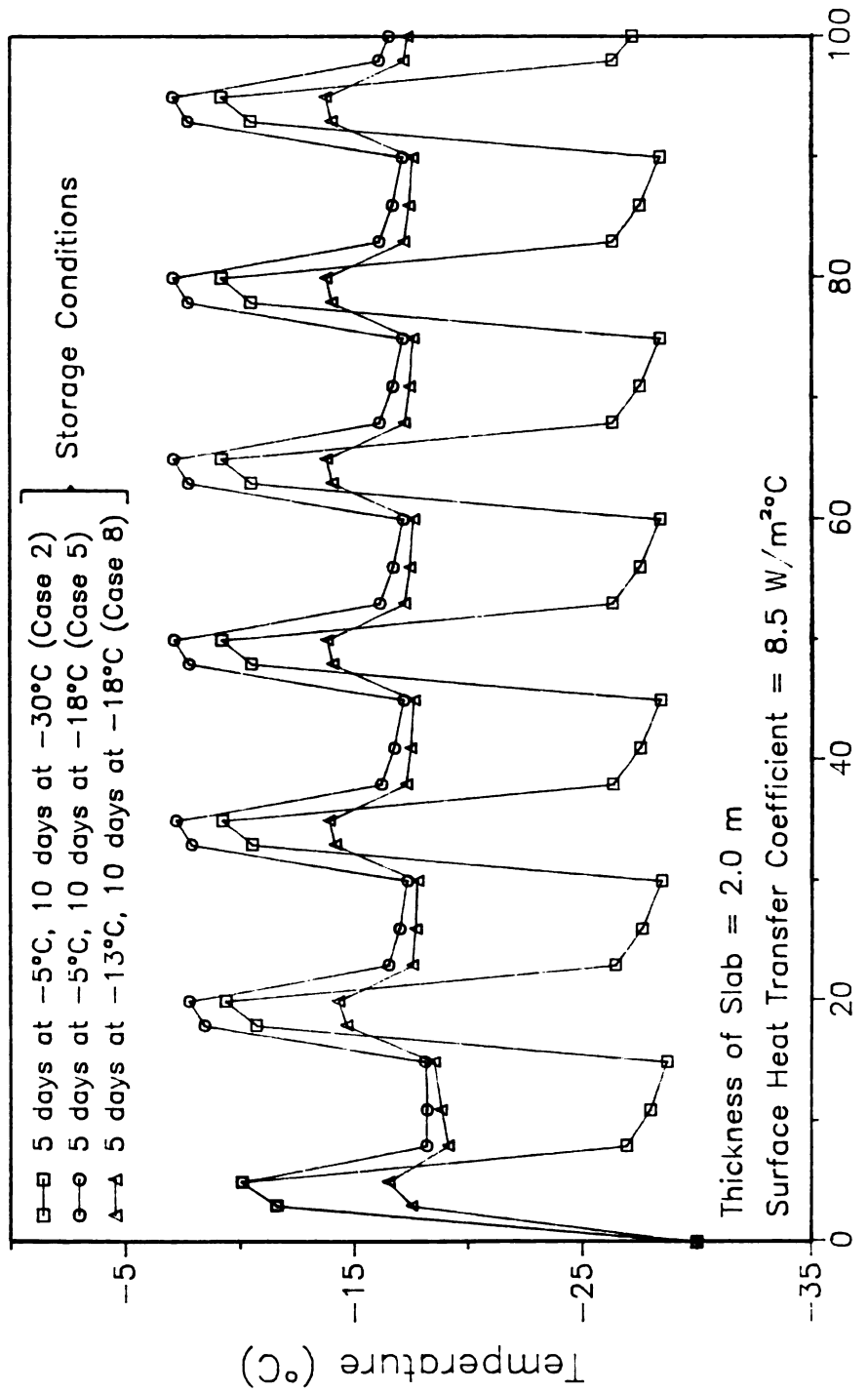


Figure 6.26b Effect of Magnitude of Step Change in Storage Temperatures on Product Temperature History at Exposed Surface, for a Product with Initial Temperature of -30°C (Cases 2, 5 and 8).

cases, as shown in Figure 6.26b. The surface temperature fluctuated 10°C for Case 5, and 4°C for Case 8, compared to 19°C for Case 2.

The quality distribution histories for the step changes in boundary condition alternating five days at the higher storage temperature ten days at the lower storage temperature, with storage temperatures of -5°C and -30°C (Case 2), -5°C and -18°C (Case 5), and -13°C and -18°C (Case 8), are shown in Figure 6.27, using the high activation energy constant, and in Figure 6.28, using the low activation energy constant. The higher minimum storage temperature used in Case 5, compared with that used in Case 2, had a very significant effect on the quality deterioration rate using the high activation energy constant. The shelf life at the outer surface was diminished after 78 storage days, while at the same time, 60% of the initial quality was retained at the geometric center. Moderate differences (6%) in quality were determined between the center and the surface using step changes in the storage temperature between -13°C and -18°C (Case 8), and the high activation energy constant.

For the quality criterion with the lower activation energy constant, little variation in quality within the product mass was found for both step changes between -5°C and -18°C (Case 5), and between -13°C and -18°C (Case 8).

In Cases 3, 6, and 9, the step change intervals were ten days for both high and low storage temperatures, and again, the magnitudes of storage temperatures were -30 and -5°C, -18 and -5°C, and -18 and -13°C, for Cases 3, 6, and 9, respectively. The solutions for the temperature histories at the geometric center and the outer surface are shown in Figures 6.29a,b. The longer storage period at the high storage temperature resulted in a greater fluctuation in temperatures at both locations. Fluctuations of 9°C, 5°C, and 3°C were observed at the

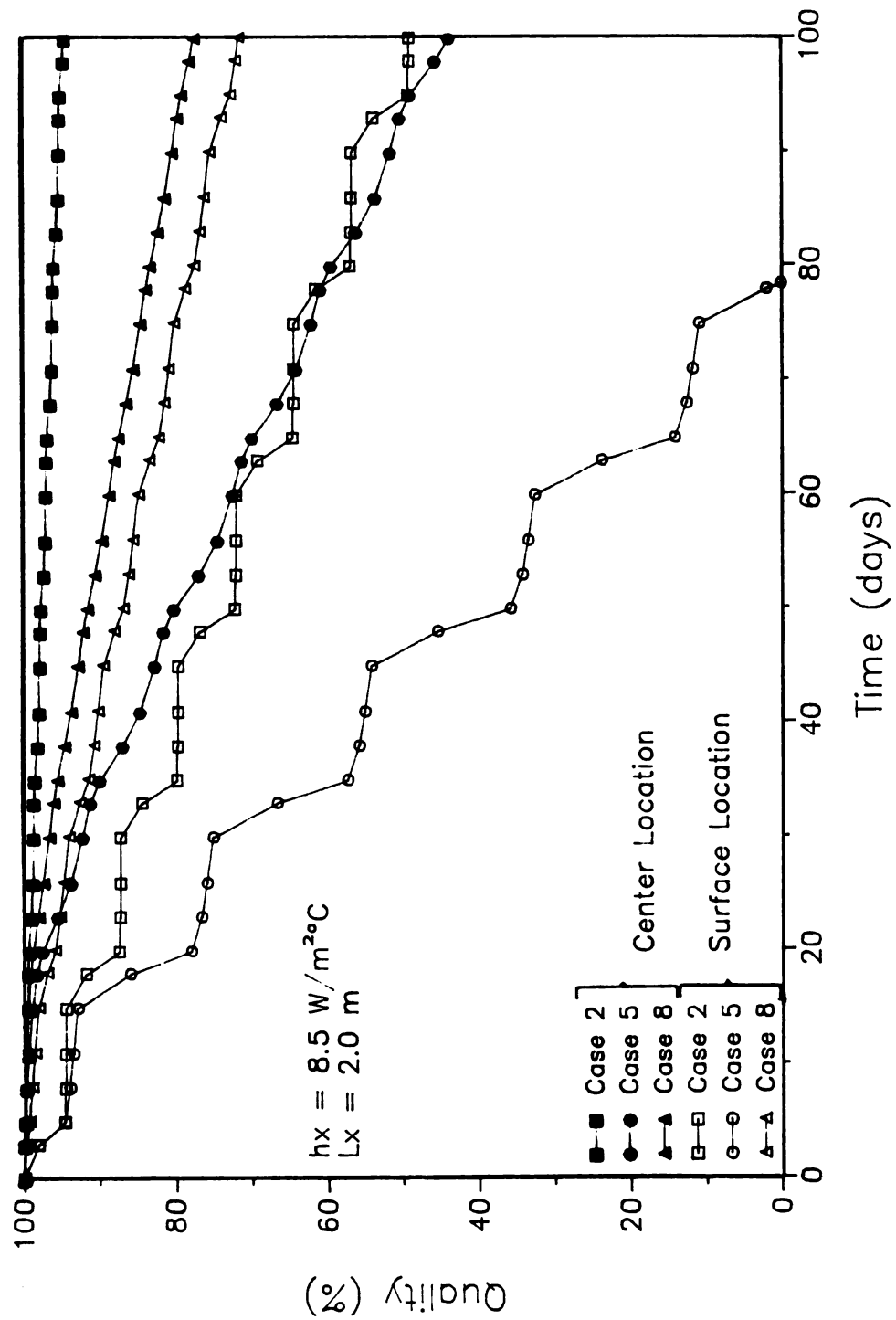


Figure 6.27 Effect of Magnitude of Step Change in Storage Temperatures on Product Quality Deterioration Rate, for a Product with Initial Temperature of -30°C (Cases 2, 5 and 8; Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

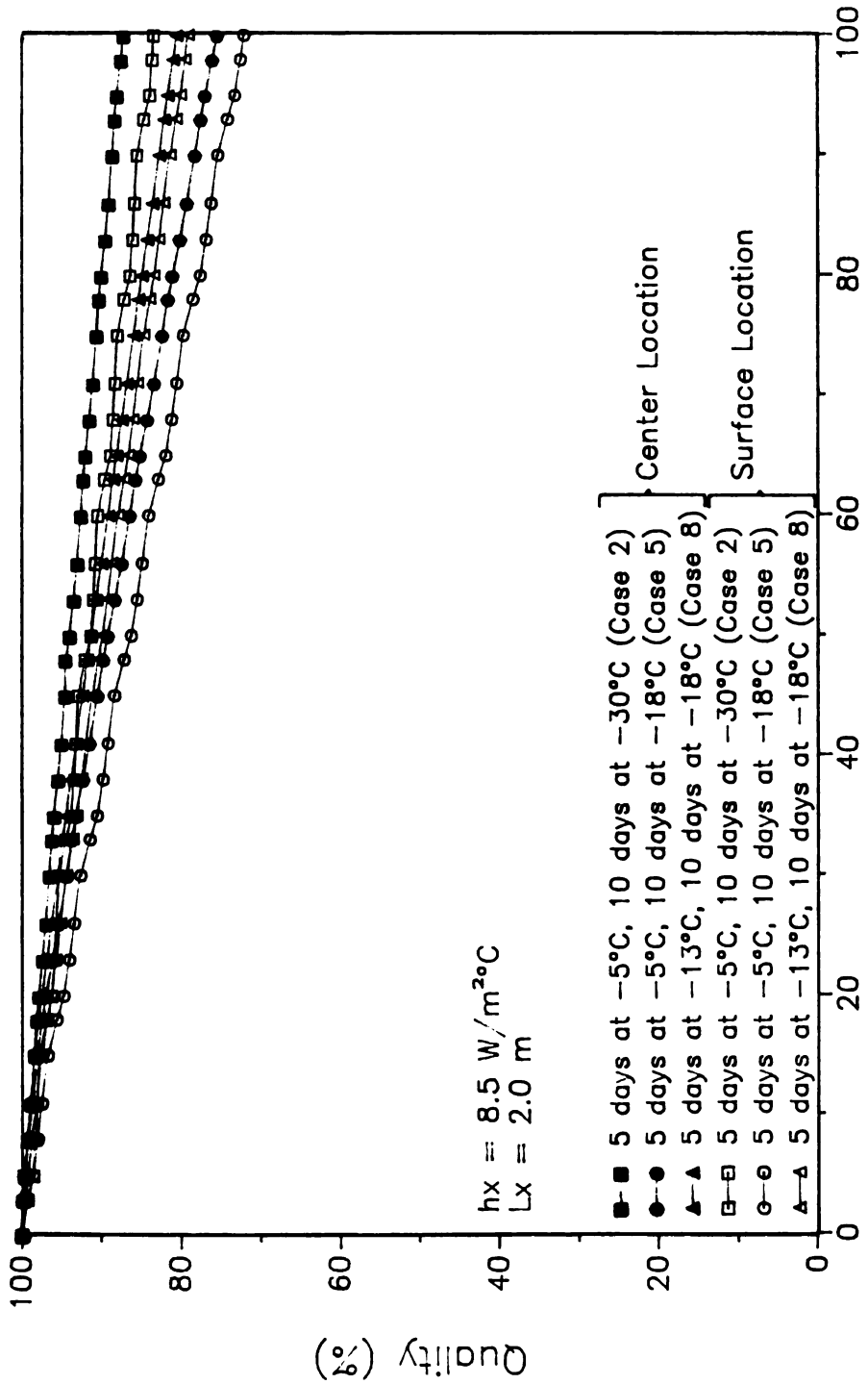


Figure 6.28 Effect of Magnitude of Step Change in Storage Temperatures on Product Quality Deterioration Rate, for a Product with Initial Temperature of -30°C (Cases 2, 5 and 8; Ea = 49 kJ/mole, Shelf-life at -18°C = 540 days).

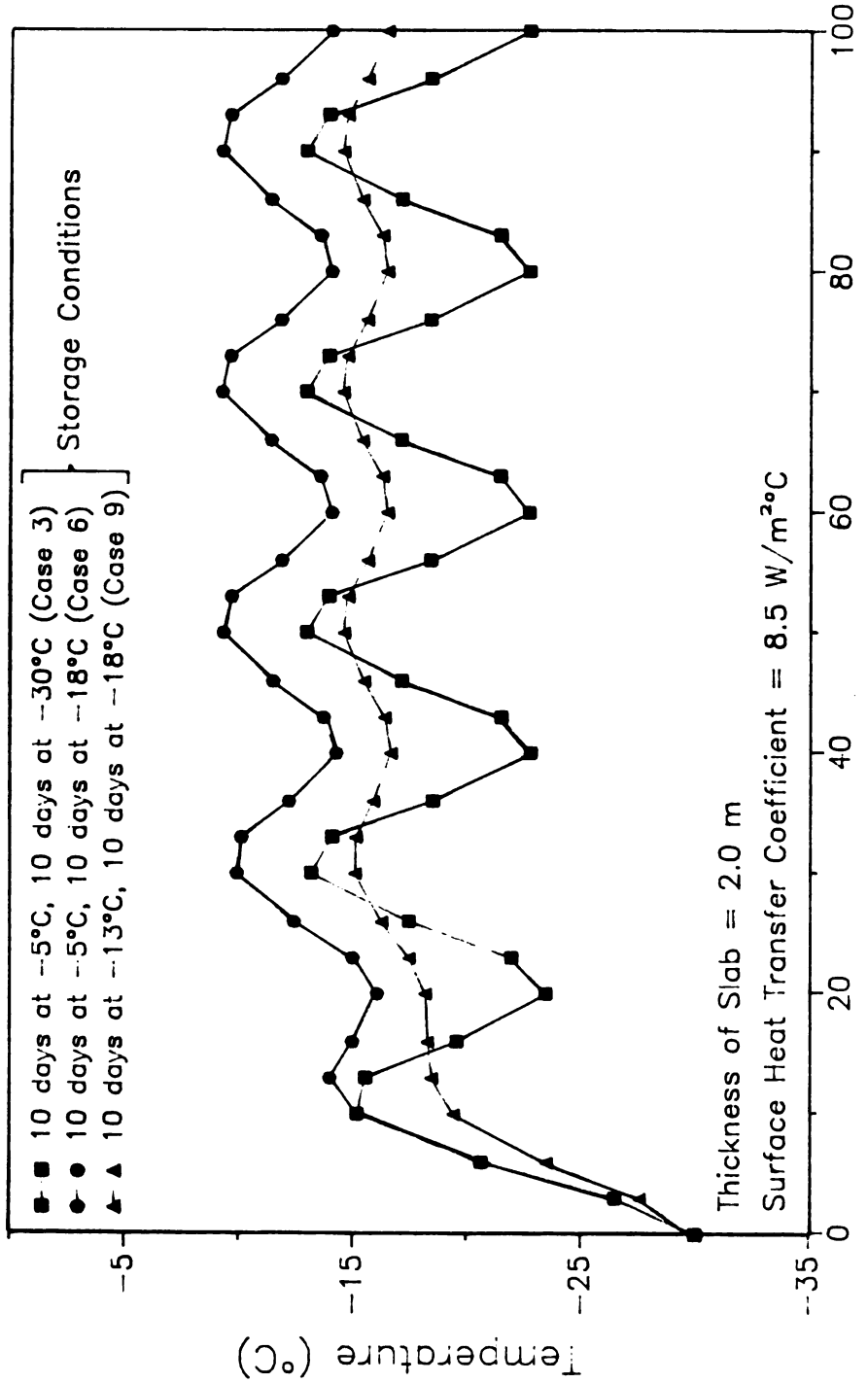


Figure 6.29a Effect of Magnitude of Step Change in Storage Temperatures on Product Temperature History at Geometric Center, for a Product with Initial Temperature of -30°C (Cases 3, 6 and 9).

geometric center, for storage temperatures alternating between -5°C and -30°C (Case 3), -5°C and -18°C (Case 6), and -13°C and -18°C (Case 9), respectively. At the outer surface, temperatures varied 15°C for Case 3, 10°C for Case 6, and 3.5°C for Case 9.

In all cases, the retention time near the higher storage temperature was longer than that found for the previous step change cycles. This resulted in a substantial increase in the rate of quality deterioration using the high activation energy constant, as shown in Figure 6.30. The shelf-life at the surface, using storage temperatures between -5°C and -30°C, was predicted to be exhausted after 67 storage days. However, using storage temperatures between -18 and -5°C resulted in the total lost of quality throughout the entire product after 88 days. The outer surface exceeded it recommended shelf life in 45 days, while the product at the geometric center still retained 64% of its initial shelf life.

When using the low activation energy constant, as shown in Figure 6.31, the slope of the curves was slightly greater than those shown in Figure 6.28 for the five day step change interval time at the high storage temperature. Again, the step changes in storage conditions had very little effect on the quality distribution history compared to that shown in Figure 6.30 for the activation energy constant of 182 kJ/mole. In all cases, the variation in quality within the product was less than 5%.

In comparing the results from Figures 6.24, 6.25, 6.27, 6.28, 6.30, and 6.31, several observations were noted. Step changes with short storage intervals (one day or less) at high storage temperatures had very little effect on the quality distribution for most cases using both high and low activation energy constants. The exception to this observation occurred when temperatures were allowed to fluctuate between -18

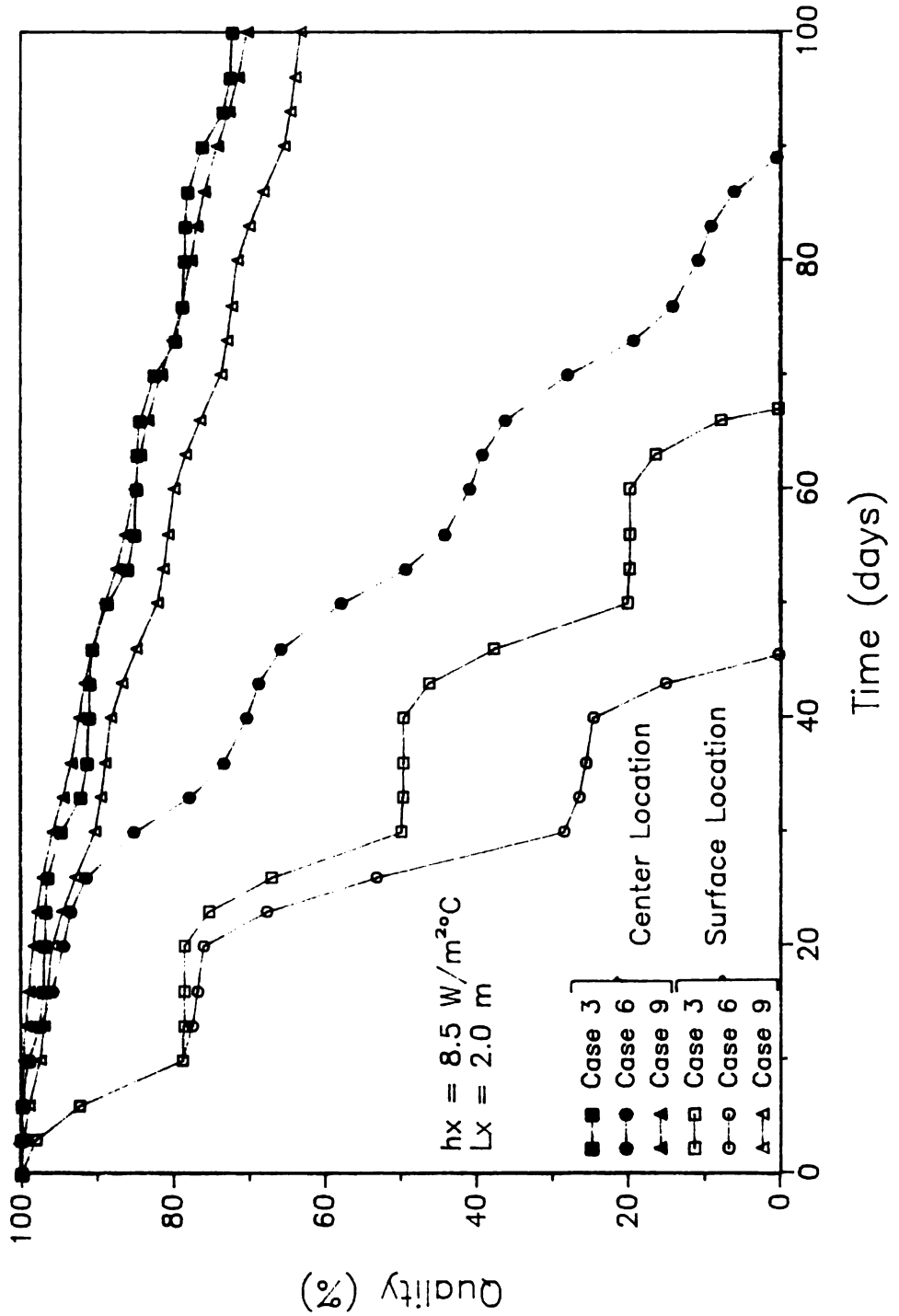


Figure 6.30 Effect of Magnitude of Step Change in Storage Temperatures on Product Quality Deterioration Rate, for a Product with Initial Temperature of -30°C (Cases 3, 6 and 9; Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

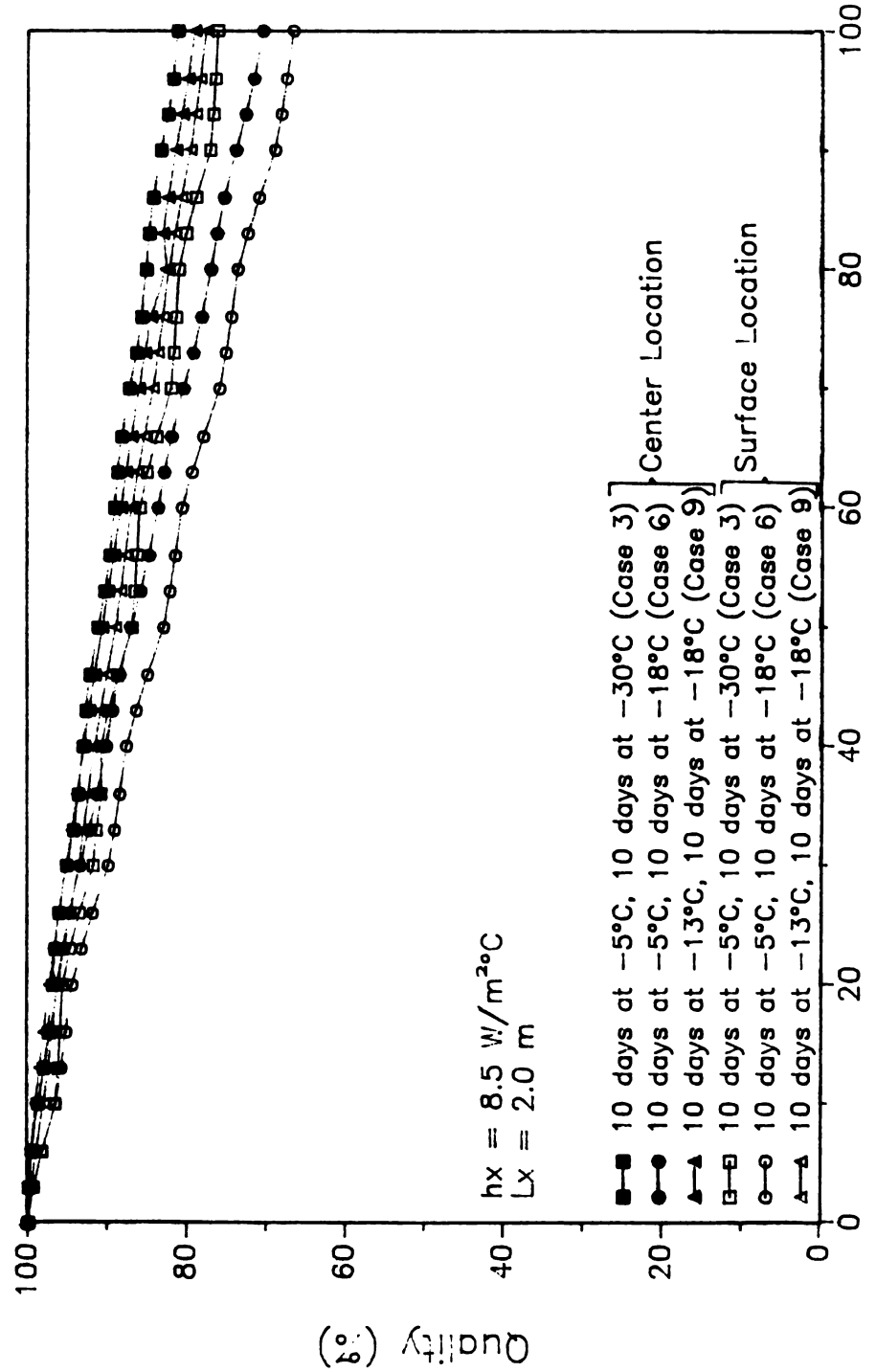


Figure 6.31 Effect of Magnitude of Step Change in Storage Temperatures on Product Quality Deterioration Rate, for a Product with Initial Temperature of -30°C (Cases 3, 6 and 9; Ea = 49 kJ/mole, Shelf-life at -18°C = 540 days).

and -5°C, and the high activation energy constant was used; in this case, a 10% variation in quality within the product was found. There was very little change in the quality deterioration rate using the low activation energy constant, in all cases. In contrast, use of the high activation energy constant resulted in substantial quality changes, expecially for the situations where the storage interval at the higher temperature was greater than or equal to five days, and the storage temperature fluctuation was greater than or equal to 13°C. Step changes in storage temperatures between -18°C and -13°C (Cases 4, 5, and 6) had very little affect on the overall quality deterioration rate of the product. This observation supports the work by Moleerantanond, et. al. (1982), who found very little change in the quality of frozen beef resulting from 3°C fluctuations in storage temperatures, and who proposed use of cyclic storage temperatures as a means of energy conservation.

6.6 Effects of Size, Two Dimensional Geometry, and Geometric Shape on Temperature and Quality Histories of Frozen Foods During Storage

Size and geometry are important factors in determining the rate of heat transfer though a mass of food product. Three aspects were considered here: (1) the product size or thickness; (2) the ratio of thickness versus length; and, (3) the geometric shape. In the first case one dimensional heat transfer was assumed. The objectives for this case were to determine under what conditions a product with a low activation energy constant will have a significant quality variation between the two sides perpendicular to heat flow, and to determine when a high activation energy product will have insignificant quality variation between these two boundaries. In the second case, two dimensional

heat transfer was assumed, and the influence of the length versus thickness ratio was investigated.

Finally, the influence of geometric shape was studied. In this case, it was desired to determine if and under what circumstances a one dimensional geometry could be used to approximate a two dimensional shape. For example, one dimensional heat transfer through a cylindrical mass, with insulated ends, might be used to simulate two dimensional heat transfer through cube shaped mass, insulated on two opposite sides.

6.6.1 Influence of Product Thickness

The influence of product thickness was first considered using the criterion with the low activation energy constant (49 kJ/mole). In Sections 6.5.2, and 6.5.3, a one dimensional slab of thickness 2.0 m was considered. The quality differential between the center and outer surfaces never exceeded 6% in all of the nine cases (Table 6.8) considered in these sections. A very large product mass was simulated to determine if the quality distribution within the product increased significantly. One dimensional heat transfer through a product 6.0 m in thickness, using step changes in boundary conditions of one day at -5°C and ten days at -18°C (Case 4), five days at -5°C and ten days at -18°C (Case 6), and ten days at -5°C and ten days at -18°C (Case 8), was considered.

The resulting temperature histories at the geometric center and at the outer surface are shown in Figures 6.32a,b. The temperature at the inner surface gradually increased with negligible fluctuations (Figure 6.32a), while the temperature at the outer surface approached the storage temperature in less than three days, and fluctuated continuously in synchronization with the step changes in storage temperatures.

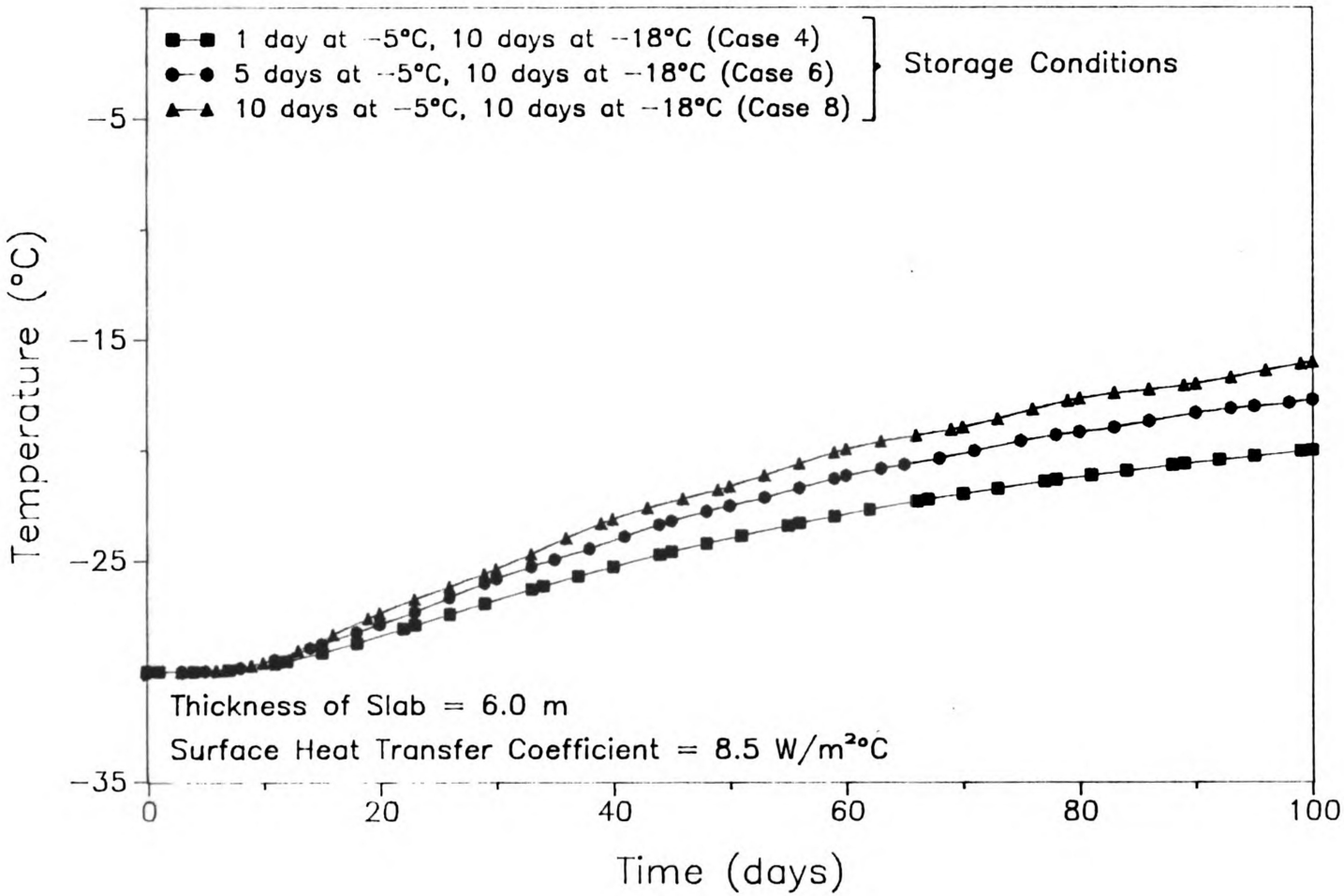


Figure 6.32a Effect of Large Product Thickness on Product Temperature History at the Geometric Center for Different Step Change Intervals, for a Product with Initial Temperature of -30°C (Cases 4, 6 and 8).

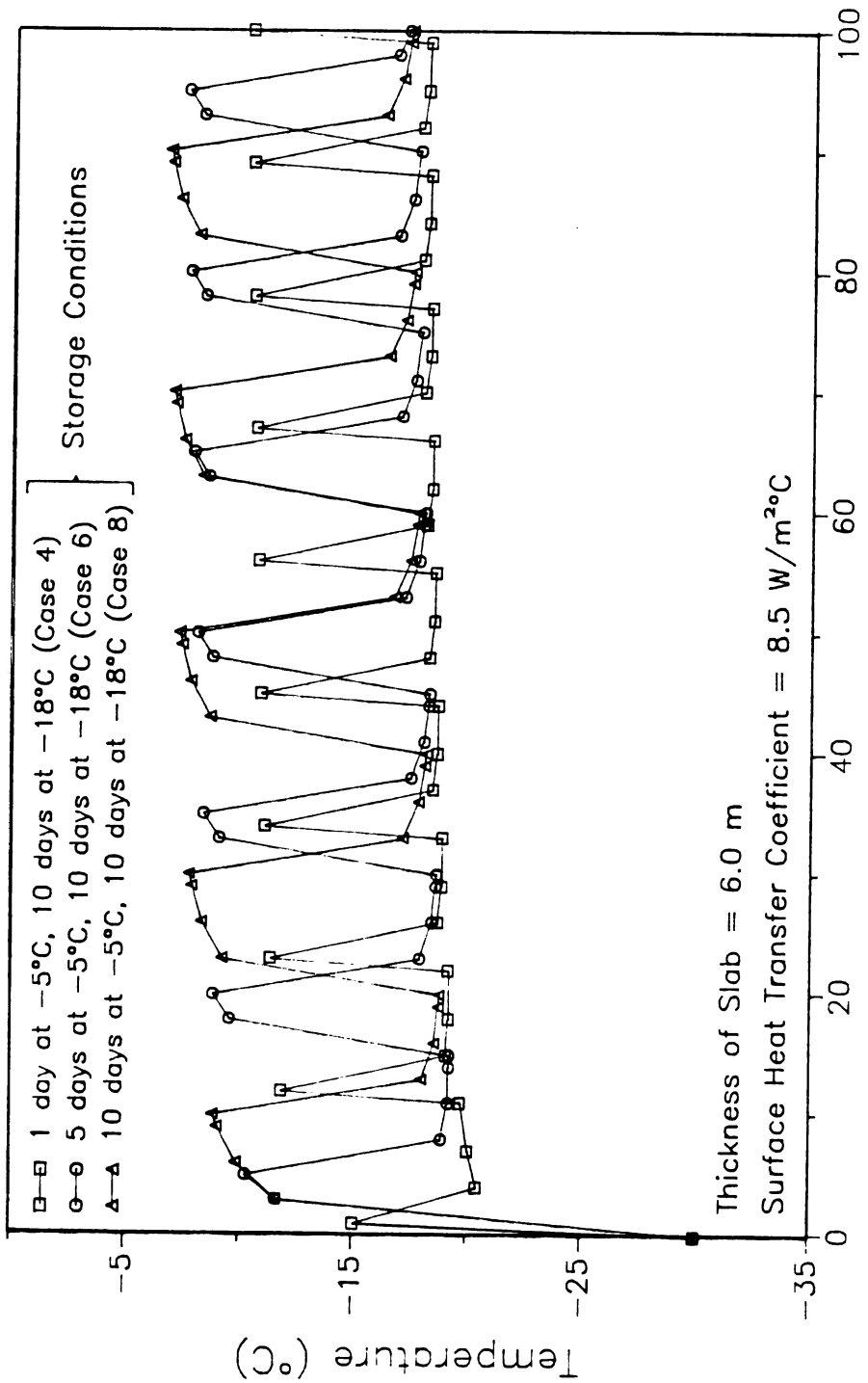


Figure 6.32b Effect of Large Product Thickness on Product Temperature History at the Exposed Surface for Different Step Change Intervals, for a Product with Initial Temperature of -30°C (Cases 4, 6 and 8).

During the initial phase of the total storage time, the temperature differential within the product was as much as 28°C; at the end of the total storage time, the temperature differential reduced to about 8°C.

The effects of the temperature differential on the quality histories at the two locations are shown in Figure 6.33. Even with high temperature variations, the quality variations within the product were very moderate, especially when compared to those shown for the 2.0 m product mass, using the high activation energy constant (Figure 6.27, and 6.30). In the situation where the step change interval for the storage temperature at -5°C was one day, ten days for the storage temperature at -18°C (Case 4), a 9% variation in quality was found after 100 storage days. Increasing the storage period at -5°C to five days increased the quality differential to 14%, and for equal storage periods at -5 and -18°C, the quality differential increased to almost 18%. Therefore, for low activation energy constants, significant (< 10%) quality variations within the product mass were only found using very large product thicknesses (> 6.0 m).

Next, the product using the high activation energy constant was considered, with the objective of determining if and under what conditions, the quality variability within the product was negligible. Again, Cases 4, 6, and 8 (step changes of: one day at -5°C and ten days at -18°C, five days at -5°C and ten days at -18°C, and ten days at -5°C and ten days at -18°C), were used for the simulated boundary conditions. The quality variation between the geometric center and the outer surface was significant (> 10%) using these cases, and a product mass of 2.0 m, as shown in Figures 6.24, 6.27, and 6.30; therefore, the product thickness was reduced to 1.0 m, and the simulation processes were repeated.

The temperature histories at the geometric center and the outer surface are shown in Figures 6.34a,b. With the exception of the curve

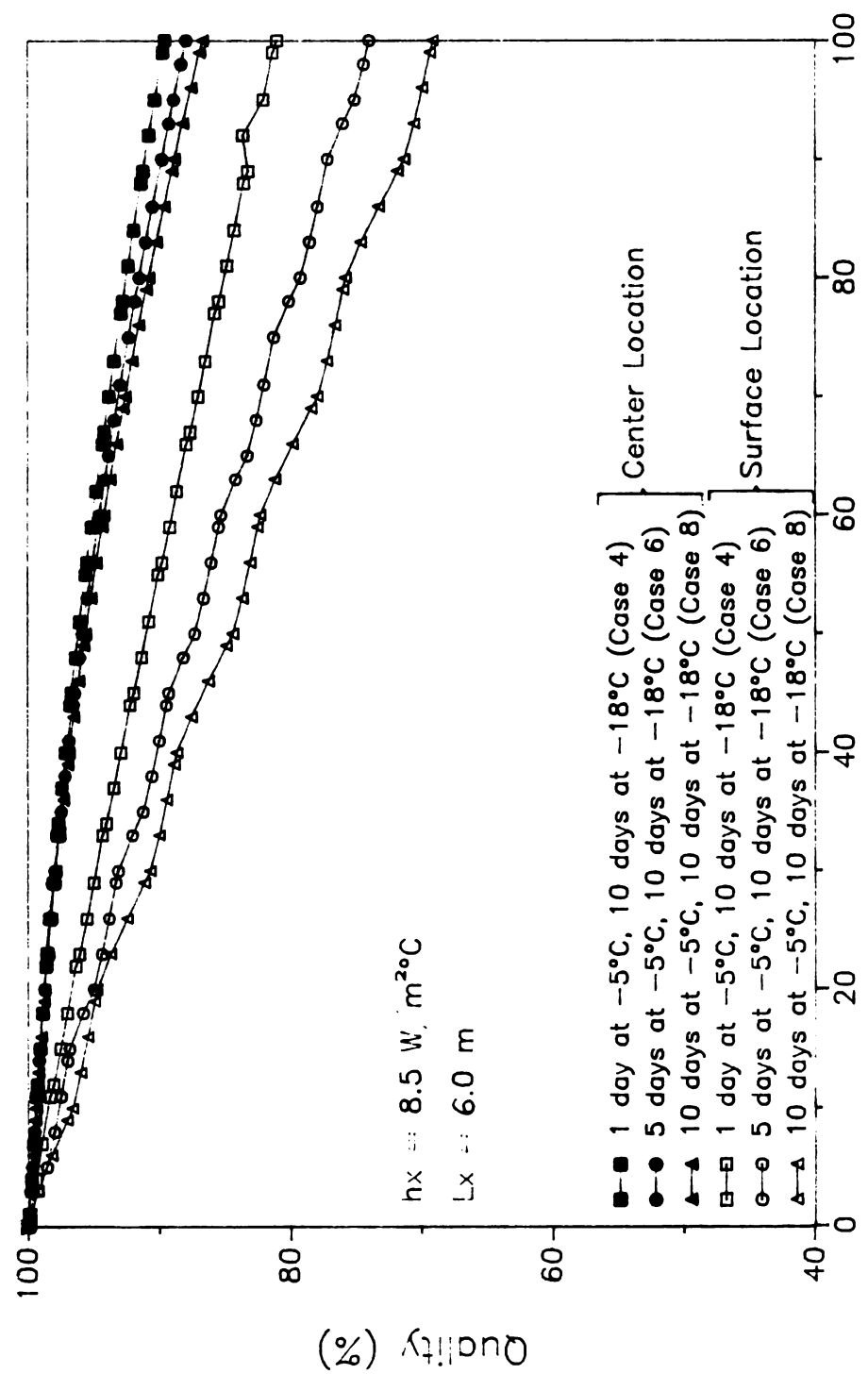


Figure 6.33 Effect of Large Product Thickness on Product Quality Deterioration Rate at the Exposed Surface for Different Step Change Intervals, for a Product with Initial Temperature of -30°C (Cases 4, 6 and 8; Ea = 49 kJ/mole, Shelf-life at -18°C = 540 days).

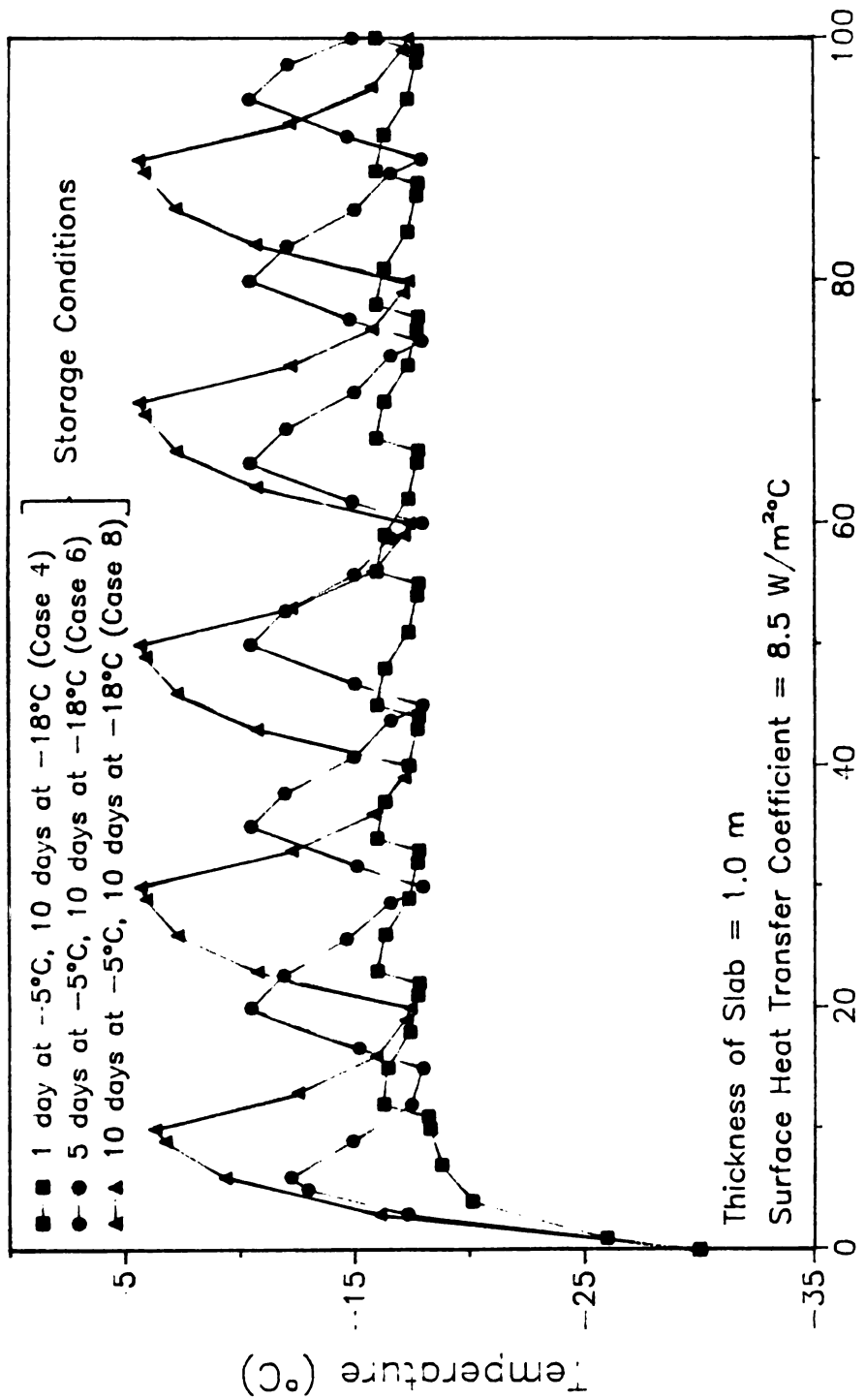


Figure 6.34a Effect of Small Product Thickness on Product Temperature History at the Geometric Center for Different Step Change Intervals, for a Product with Initial Temperature of -30°C (Cases 4, 6 and 8).

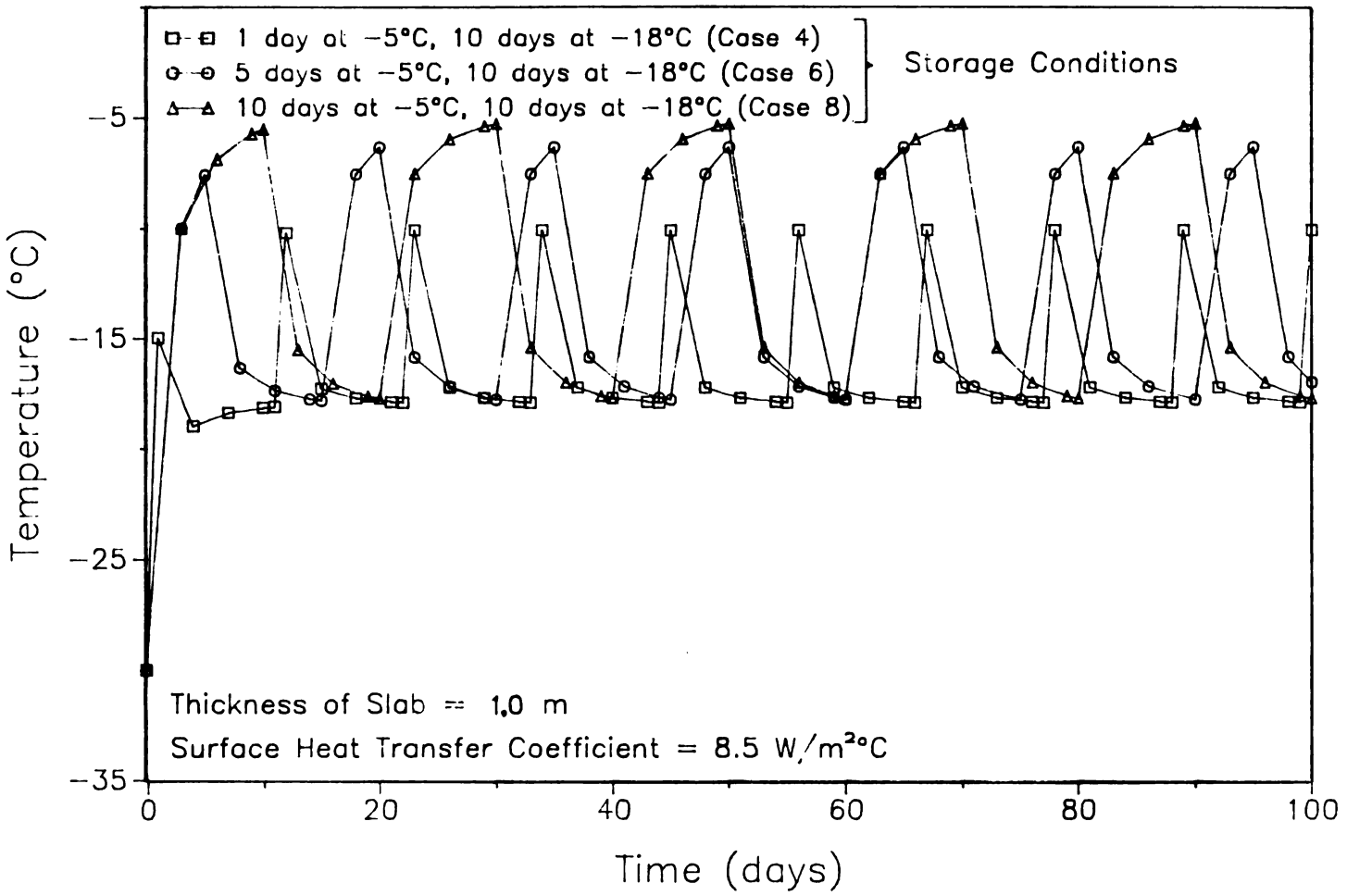


Figure 6.34b Effect of Small Product Thickness on Product Temperature History at the Exposed Surface for Different Step Change Intervals, for a Product with Initial Temperature of -30°C (Cases 4, 6 and 8).

for resulting from one day at -5°C and ten days at -18°C (Case 4), the temperature fluctuations increased significantly, compared to those found using a thickness of 2.0 m. Comparing Figures 6.34a and 6.34b, the resonance time near the high storage temperature at the outer surface was about twice that at the geometric center.

The resulting quality histories, using the criterion with an activation energy constant of 182 kJ/mole, at the geometric center and at the outer surface, are shown in Figure 6.35 for the three cases considered. The quality variation within the product was small (8%) using Case 4 boundary conditions (one day at -5°C, ten days at -18°C), while the quality variations using the other two cases were high. For step changes of five days at -5°C and ten days at -18°C in the storage conditions (Case 6), the shelf-life at the geometric center was exhausted after 65 days, at which time 35% of the initial quality was retained at the outer surface. When the boundary conditions described by Case 8 (10 days at both -5°C and -18°C) were used, the quality at the surface was diminished after 29 days, while 32% of the initial quality remained in the product at the geometric center.

Although the quality variation between the two locations was less than that found using Lx = 2.0 m, there was still a high variation in quality within the product using Lx = 1.0 m using the boundary conditions defined by Cases 6 and 8. The same simulations were repeated using Lx = 0.2 m. The quality variation between the center and the outer surfaces was about 6.5% of the initial quality, for the Case 4 boundary condition; this value changed only slightly, compared to the results using Lx = 2.0 m, due to the short storage interval at -5°C. The temperature variations between the surfaces, using Cases 6 and 8, were found to be very small (<0.5°C), resulting in very little variation in quality between the surfaces for these cases, as shown in Figure 6.36.

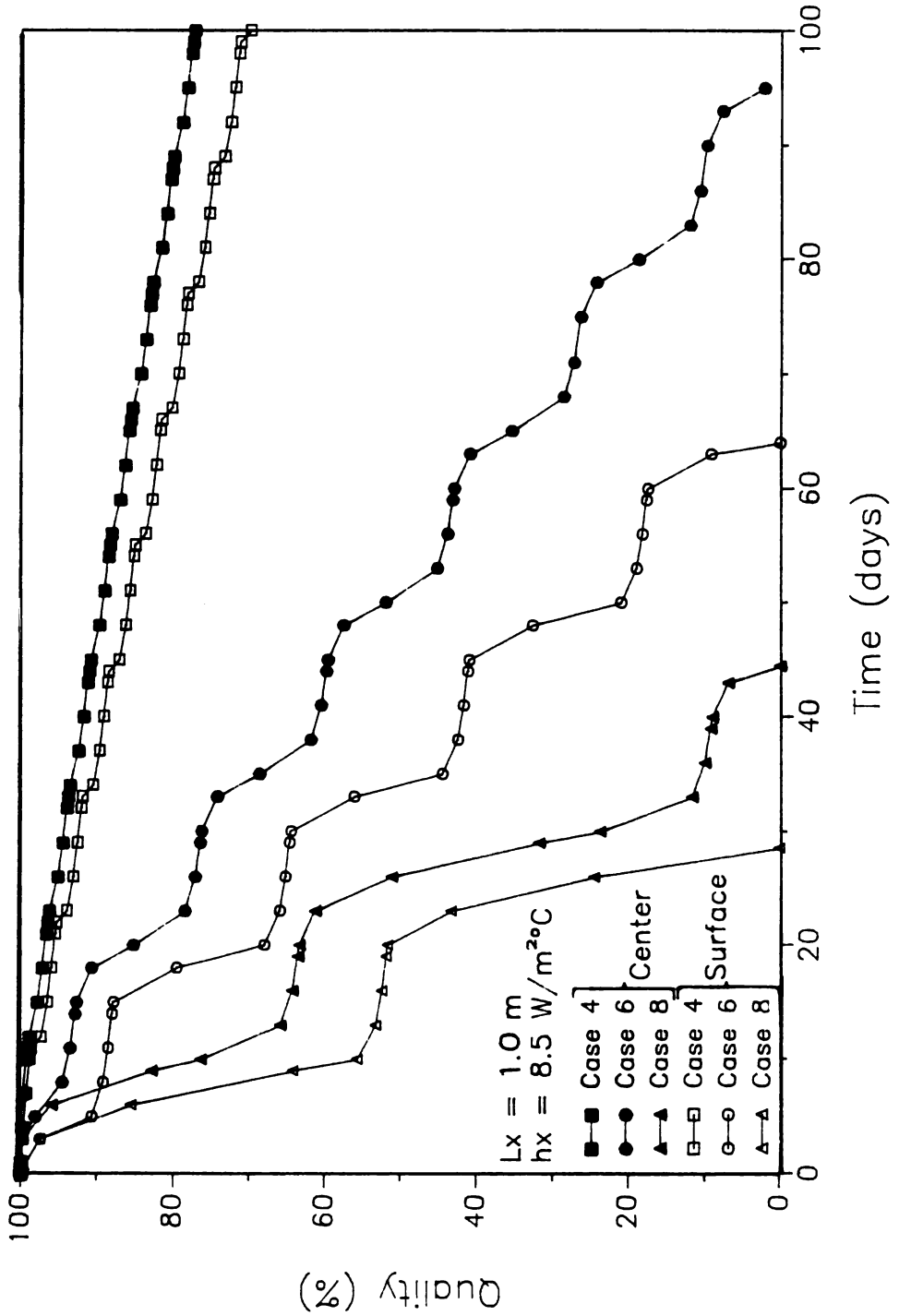


Figure 6.35 Effect of Small Product Thickness, Lx = 1.0 m, on Product Quality Deterioration Rate at the Exposed Surface for Different Step Change Intervals, for a Product with Initial Temperature of -30°C (Cases 4, 6 and 8; Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

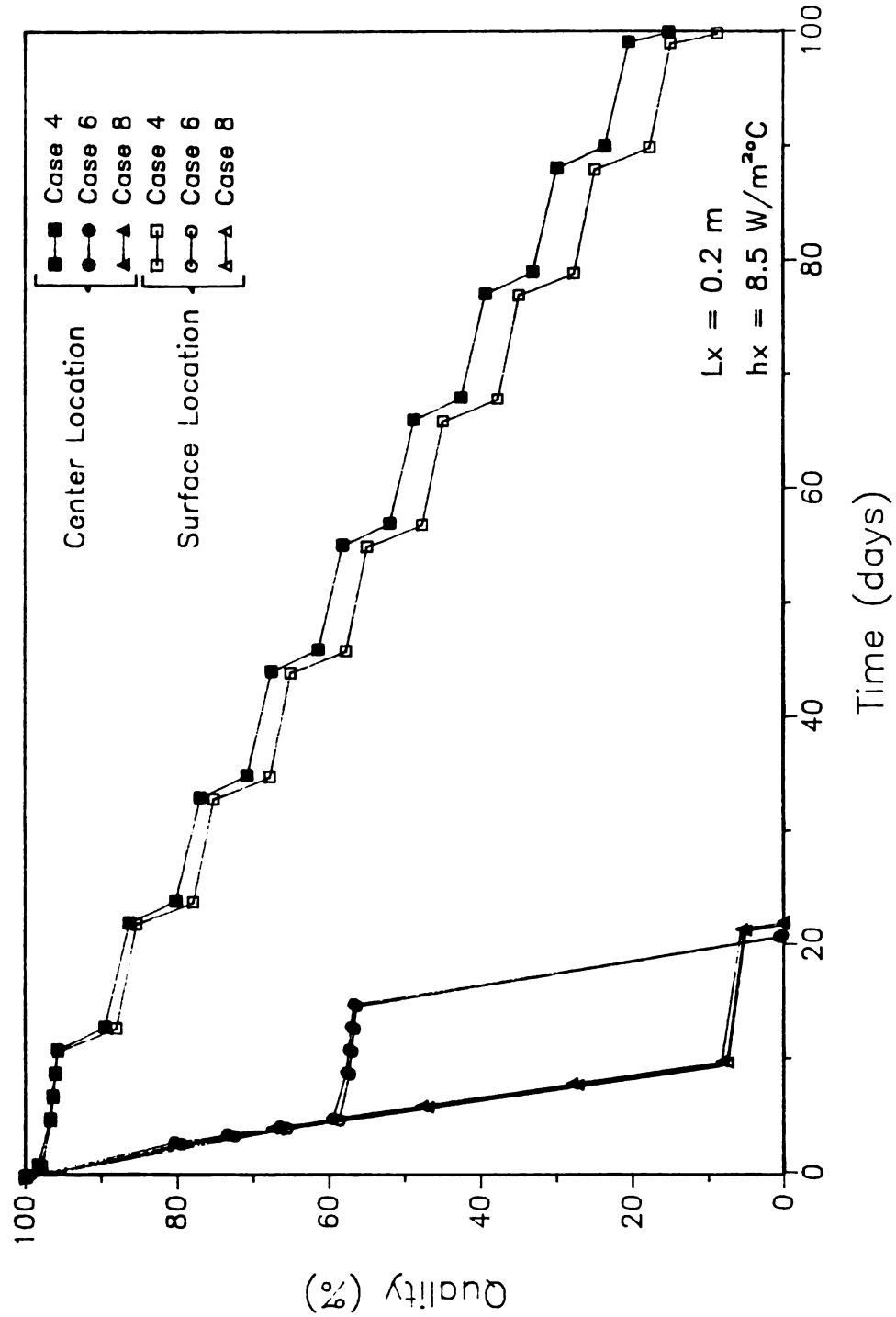


Figure 6.36 Effect of Small Product Thickness, Lx = 0.2 m, on Product Quality Deterioration Rate at the Exposed Surface for Different Step Changes Intervals, for a Product with Initial Temperature of -30°C (Cases 4, 6 and 8; Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

To summarize, for low activation energy products, quality variations within the product were found to be significant only for very large product masses (≥ 6.0 m), considering one dimensional heat transfer, and a surface heat transfer coefficient of 8.5 $W/m^2°C$ on two parallel surfaces, with insulated conditions on all other boundaries. For high activation energy products, the quality variation was insignificant only for very small product masses (≤ 0.2 m), using the step changes in boundary conditions of at least five days at -5°C, and ten days at -18°C. In addition, the overall rate of quality deterioration increased substantially as the product size considered decreased.

6.6.2 Effects of Two Dimensional Heat Transfer on Temperature and Quality Histories.

Two dimensional heat transfer was simulated in a rectangularly shaped product mass, using step changes in the boundary conditions of five days at -5°C and ten days at -18°C (Case 6). A constant heat transfer coefficient of 8.5 $W/m^2°C$ was considered on the surfaces perpendicular to the directions of heat transfer, and the surfaces parallel to the heat flow were considered to be insulated. Various width versus height ratios were considered for the exposed boundaries.

Temperature histories, assuming two dimensional heat transfer in the x and y directions, were first determined using equal lengths of 2.0 m for both the width, Lx, and height, Ly. A constant heat transfer coefficient of 8.5 $W/m^2°C$ was imposed along all boundaries perpendicular to the direction of heat flow. The two dimensional temperature history was determined at the geometric center ($x = 0$, $y = 0$), the midpoint of the exposed sides ($x = 0$, $y = Ly/2$; and $x = Lx/2$, $y = 0$), and at the exposed

corner, (x = Lx/2, y = Ly/2) for all cases considered in this section. These locations are shown in Figure 6.37.

The resulting temperature histories were compared to the one dimensional solution of a 2.0 m product mass subject to the same initial and boundary conditions, shown in Figure 6.26a,b. The temperature histories are shown in Figure 6.38, for the two dimensional simulation at the locations shown in Figure 6.37, and for the one dimensional simulation at the geometric center and exposed surface. Comparing the one and two dimensional solutions, the temperature at the geometric center using the one dimensional solution was slightly higher than that found with the two dimensional solution. The temperature at the surface, using the one dimensional solution was bounded by the temperatures at the midpoint of the sides and the corners. Note that the temperatures around the perimeter of the two dimensional case were bounded by the temperatures at the corners and the midpoint of the sides, and due to symmetry, the temperature histories at (0,Ly/2) and (Lx/2,0) are identical.

The quality histories, using the activation energy constant of 182 kJ/mole, at these locations are shown in Figure 6.39. In all cases, the rate of quality deterioration increased, compared with the one dimensional simulation. The one dimensional solution predicted quality 8% higher than the two dimensional solution after 100 storage days. The predicted quality at the exposed surface, using the one dimensional solution, was similar to that predicted at the side midpoints in the two dimensional solution. The quality at the most extreme point (Lx/2,Ly/2) deteriorated after only 47 days. Although the difference in temperature around the perimeter of the product mass was small (two dimensional case), the difference in predicted quality around the outer surface was high, due to the high temperature dependence of the rate constant.

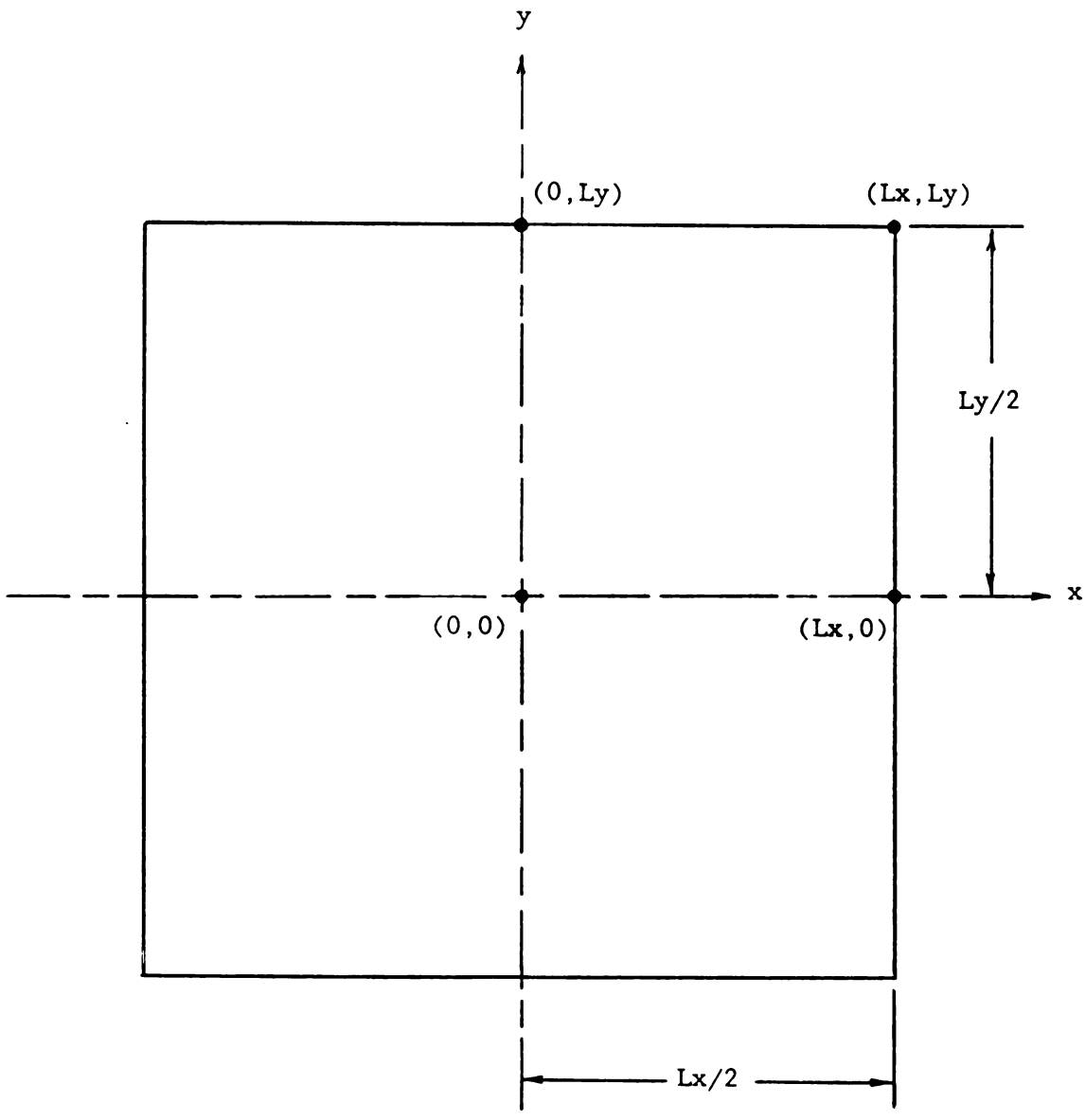


Figure 6.37 Locations of Solutions for Two Dimensional Geometry.

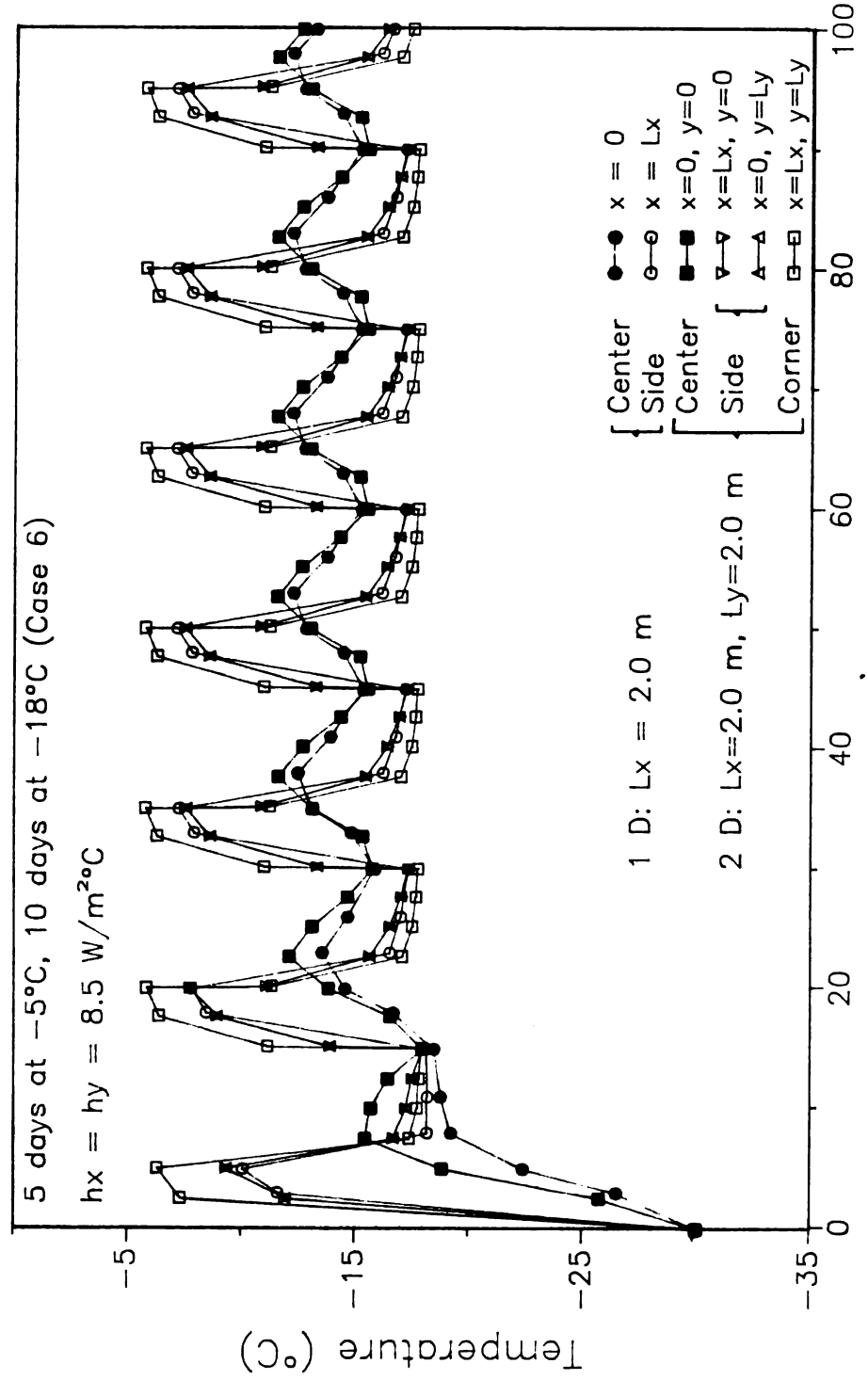


Figure 6.38 Two Dimensional Numerical Solution for Square Rod (Lx = Ly = 2.0 m) Compared to One Dimensional Solution for a Slab with Equal Dimension, Lx, for Product with Initial Temperature of -30°C (Case 6).

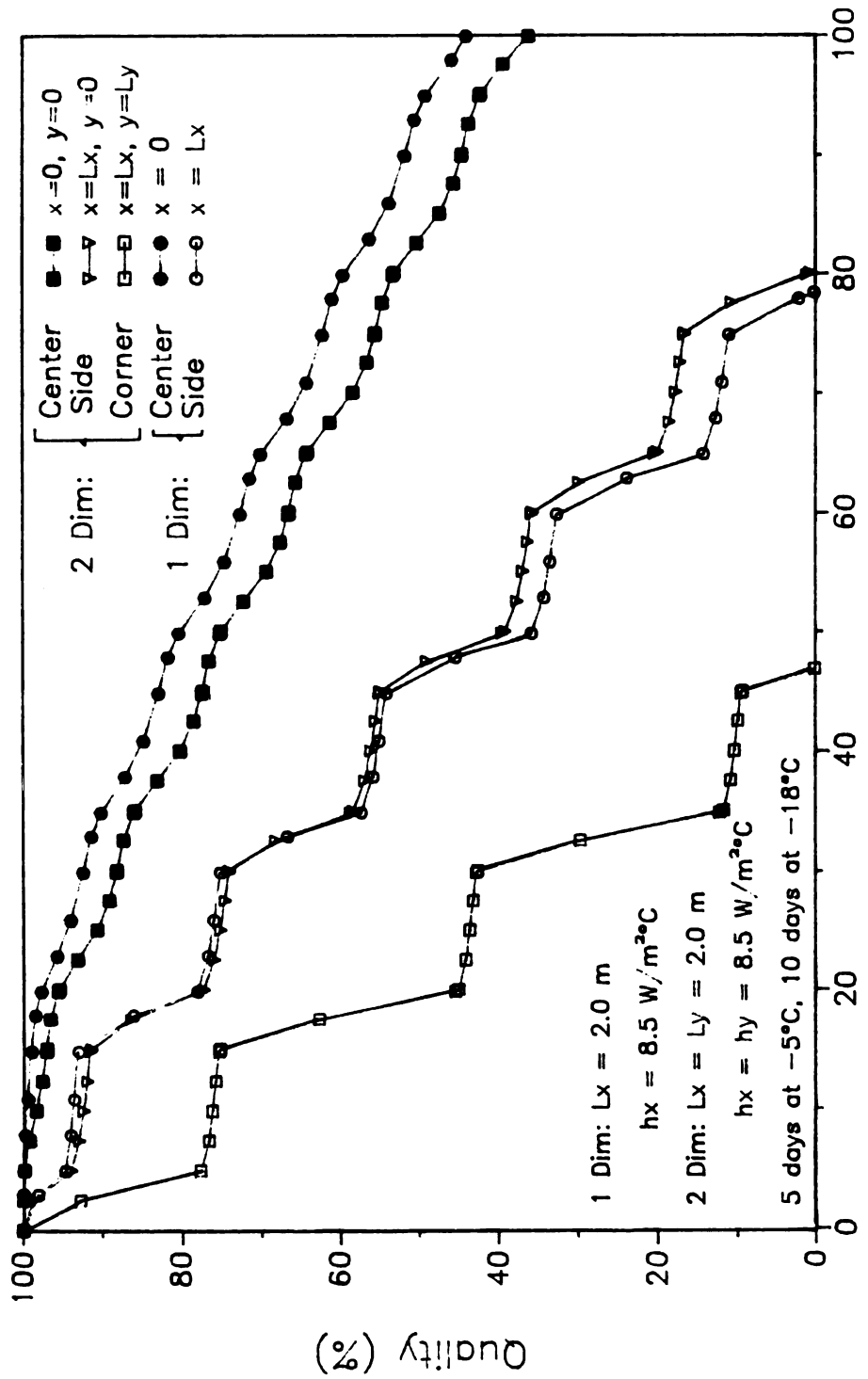


Figure 6.39 Quality Deterioration Rate Resulting from Two Dimensional Solution for Square Rod (Lx = Ly = 2.0 m) Compared to One Dimensional Solution for a Slab with Equal Dimension, Lx, for Product with Initial Temperature of -30°C (Case 6; Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

The two dimensional simulations were repeated using Lx = 2.0 m and Ly = 1.0 m; results are shown in Figure 6.40. The temperature variations at the geometric center, using the two dimensional solution, were higher than that shown for the previous case. Due to lack of symmetry, the temperature histories at (0,Ly/2) and (Lx/2,0) were not the same. The two dimensional solution was compared to the one dimensional solution for an infinite slab 1.0 m in thickness. Again, the one dimensional temperature solution was bounded by the two dimensional solution. The predicted quality histories resulting from these temperature histories are shown in Figure 6.41, using the activation energy constant of 182 kJ/mole. The quality deterioration at the geometric center of the two dimensional case was greater compared to that found for the 2.0 m by 2.0 m rod; however, little change was found in the quality profile at the exposed corner, compared to the 2.0 m by 2.0 m solution. The one dimensional solution for a 1.0 m slab over-estimated the quality at the geometric center, and closely approximated the quality along the sides, while greatly over-estimating the quality at the corners.

Finally a 2 m by 0.2 m rod was considered. Due to the small height versus width ratio, the temperature distribution was very similar at all points in the rod, with high temperature fluctuations in all cases, as shown in Figure 6.42. The resulting one dimensional solution (not shown here) for an infinite slab, 0.2 m thick, was bounded by the solution for the two dimensional case. The resulting quality histories were very similar for all locations (Figure 6.43). Due to the high temperature fluctuations, the quality deterioration rate was very rapid, and little variation in quality within the product (< 10%) was found. The one dimensional temperature solution produced very similar quality results,

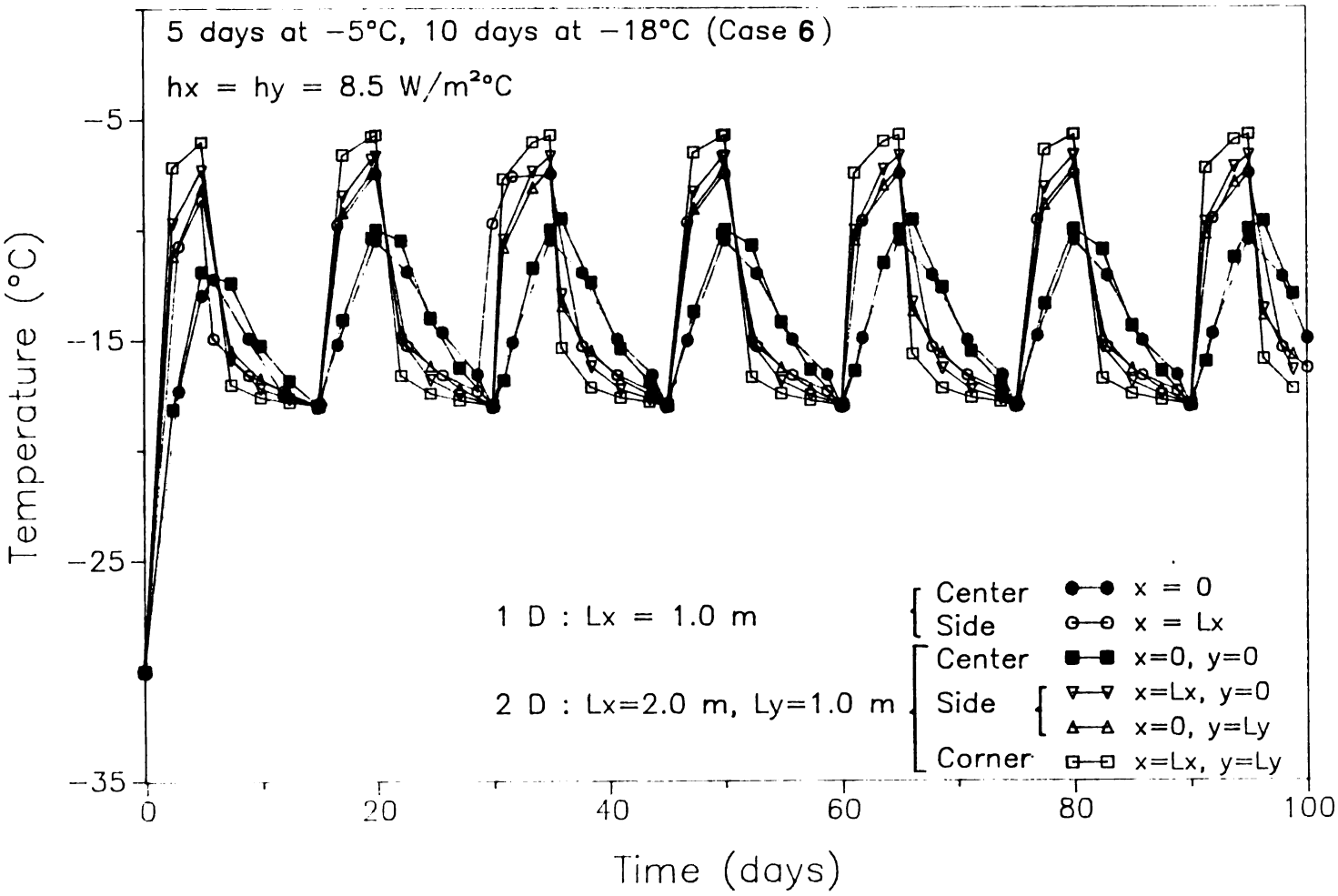


Figure 6.40 Two Dimensional Numerical Solution for Rectangular Rod (Lx = 2.0 m, Ly = 1.0 m) Compared to One Dimensional Solution for a Slab with Dimension, Lx = 1.0 m, for Product with Initial Temperature of -30°C (Case 6).

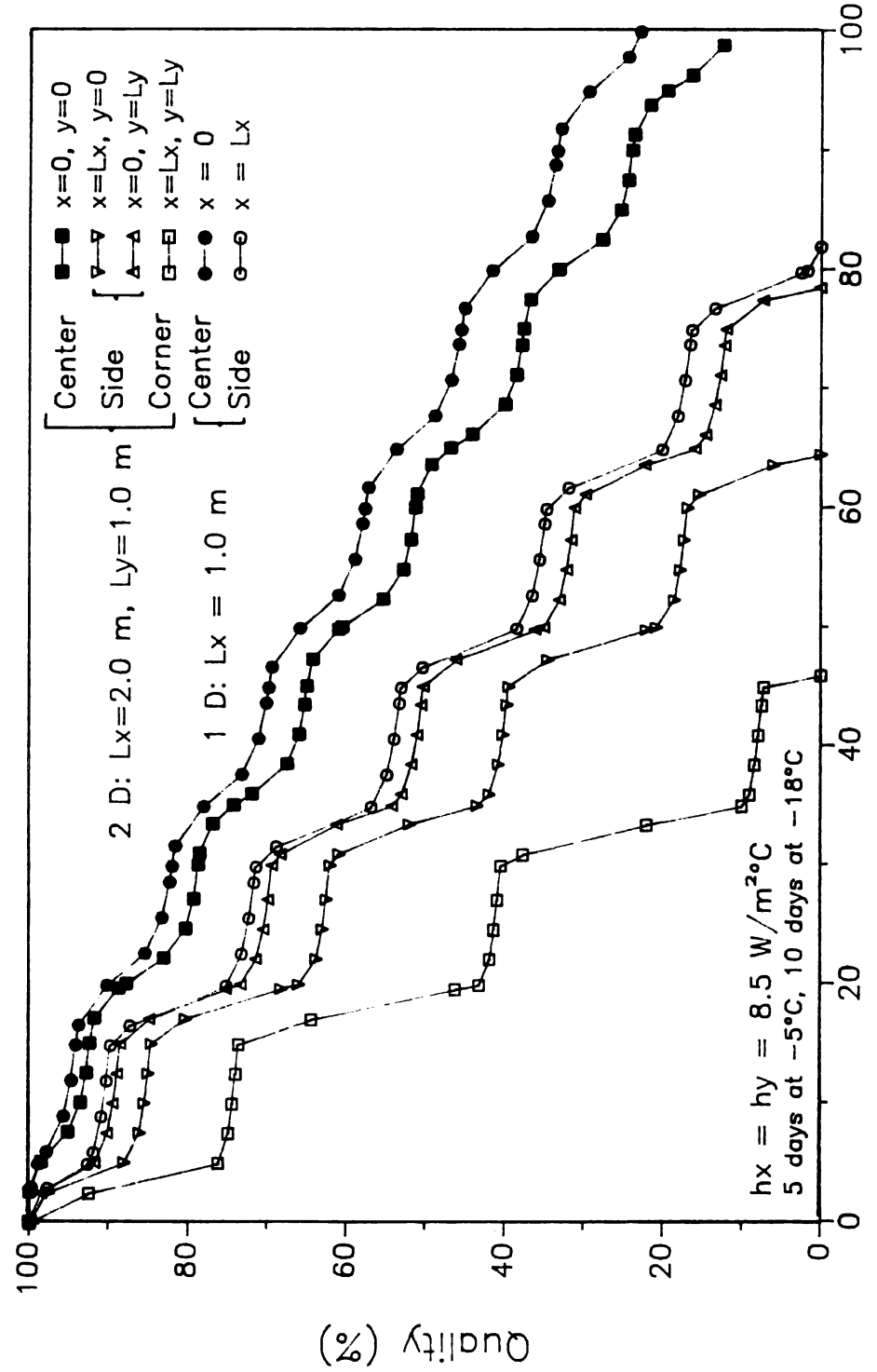


Figure 6.41 Quality Deterioration Rate Resulting from Two Dimensional Solution for Rectangular Rod (Lx = 2.0 m, Ly = 1.0 m) Compared to One Dimensional Solution for Slab with Dimension, Lx = 1.0 m, for Product with Initial Temperature of -30°C (Case 6; Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

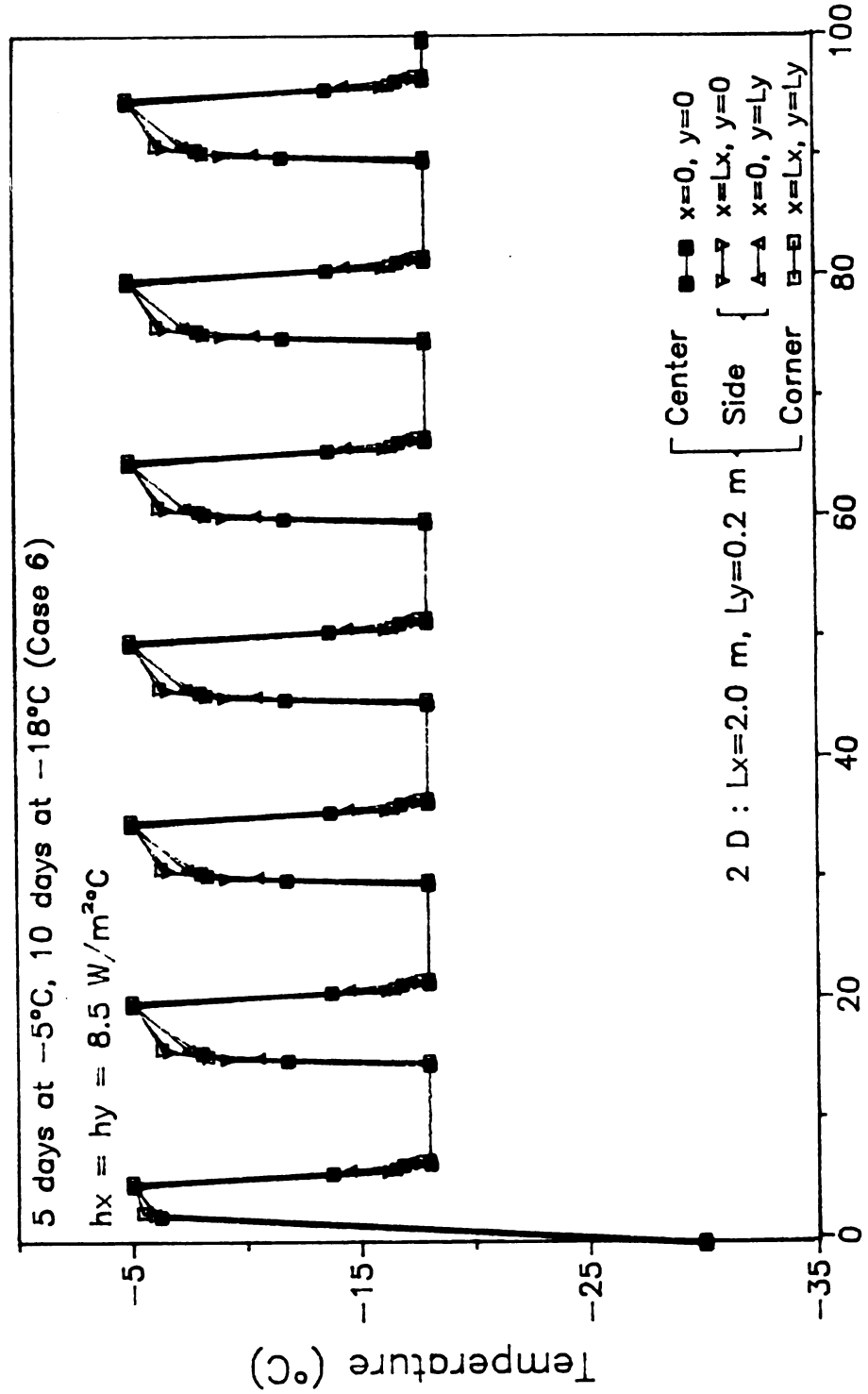


Figure 6.42 Two Dimensional Numerical Solution for Rectangular Rod (Lx = 2.0 m, Ly = 0.2 m) Compared to One Dimensional Solution for a Slab with Dimension, Lx = 0.2 m, for Product with Initial Temperature of -30°C (Case 6).

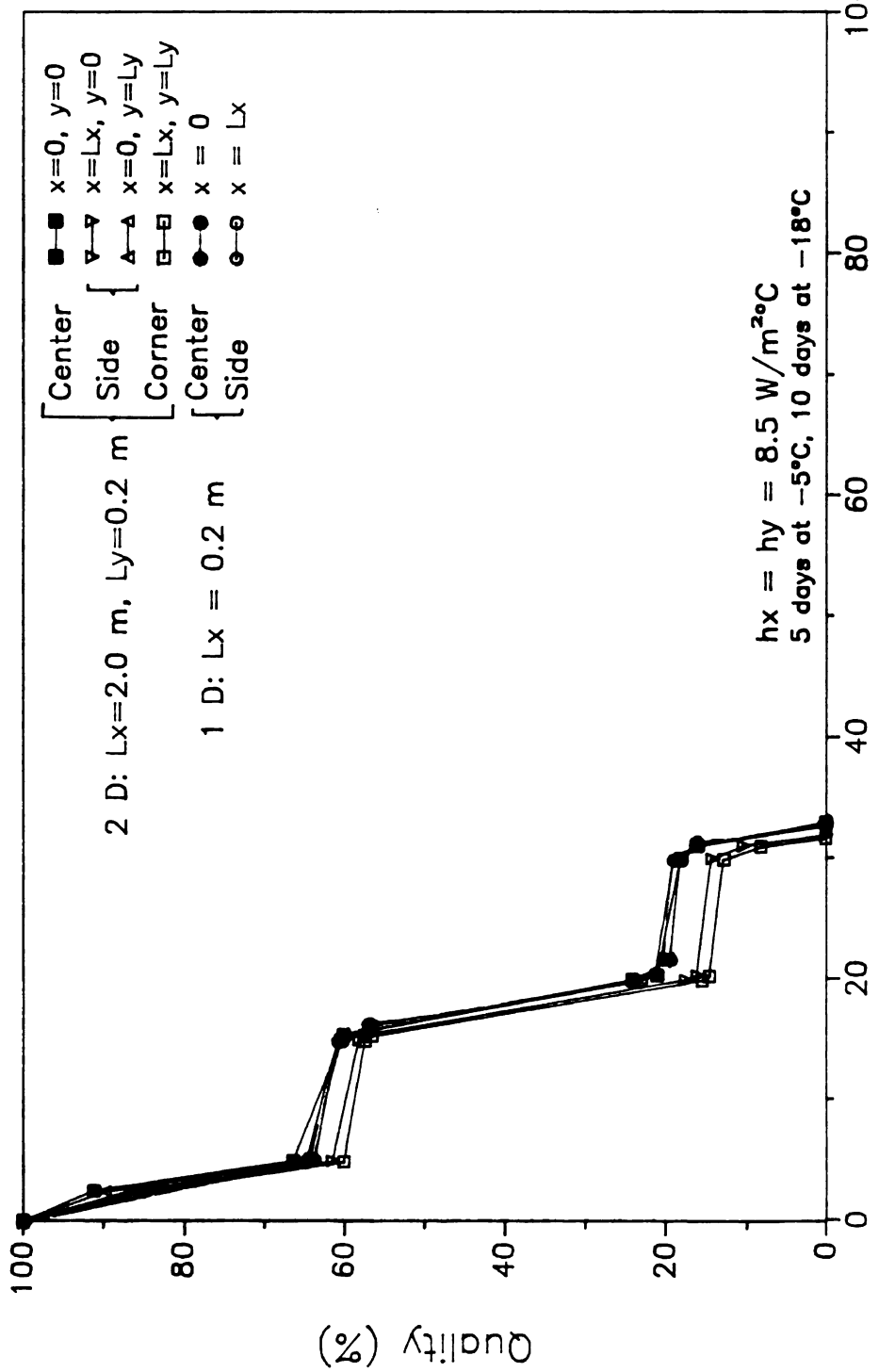


Figure 6.43 Quality Deterioration Rate Resulting from Two Dimensional Solution for Rectangular Rod (Lx = 2.0 m, Ly = 0.2 m) Compared to One Dimensional Solution for Slab with Dimension, Lx = 0.2 m, for Product with Initial Temperature of -30°C (Case 6; Ea = 182 kJ/mole, Shelf-life at -18°C = 630 days).

indicating that for the limiting case where Lx >> Ly, the one dimensional solution for an infinite slab may be used to approximate two dimensional heat transfer through a rectangular rod.

The same three cases (Lx = Ly = 2.0 m; Lx = 2.0 m, Ly = 1.0 m; and, Lx = 2.0 m, Ly = 0.2 m) were repeated using the activation energy constant of 49 kJ/mole. Results are shown in Figures 6.44-6.46. In all cases, the slope of the curves were very similar, and very little variation in quality was found within the product. Since the temperature solution for the one dimensional case was bounded by the solution for the two dimensional case, the resulting one dimensional quality profiles would be bounded by the two dimensional solutions shown in Figures 6.44-6.46. The implications from these results are that for products with low activation energy constants (> 60 kJ/mole), the one dimensional solution may be used to approximate quality deterioration rates resulting from two dimensional heat transfer with little error.

In conclusion, quality variations within the product, using the high activation energy constant, were significant for width versus height ratios greater than or equal to 50%, and insignificant for ratios less than or equal to 10%. However, for low activation energy products, the quality distribution was relatively independent of the width vs height ratio, indicating that a one dimensional model, or a model using the mass average temperature (lumped capacitance model) might provide an excellent estimation of the product quality. When comparing the two dimensional solution, using equal width and height, to the one dimensional solution of equivalent geometry, the quality at the geometric center and the corners of the two dimensional case were over estimated by the one dimensional solution. The one dimensional solution closely approximated at the outer surface closely approximated the two dimensional solution at the midpoint of the sides.

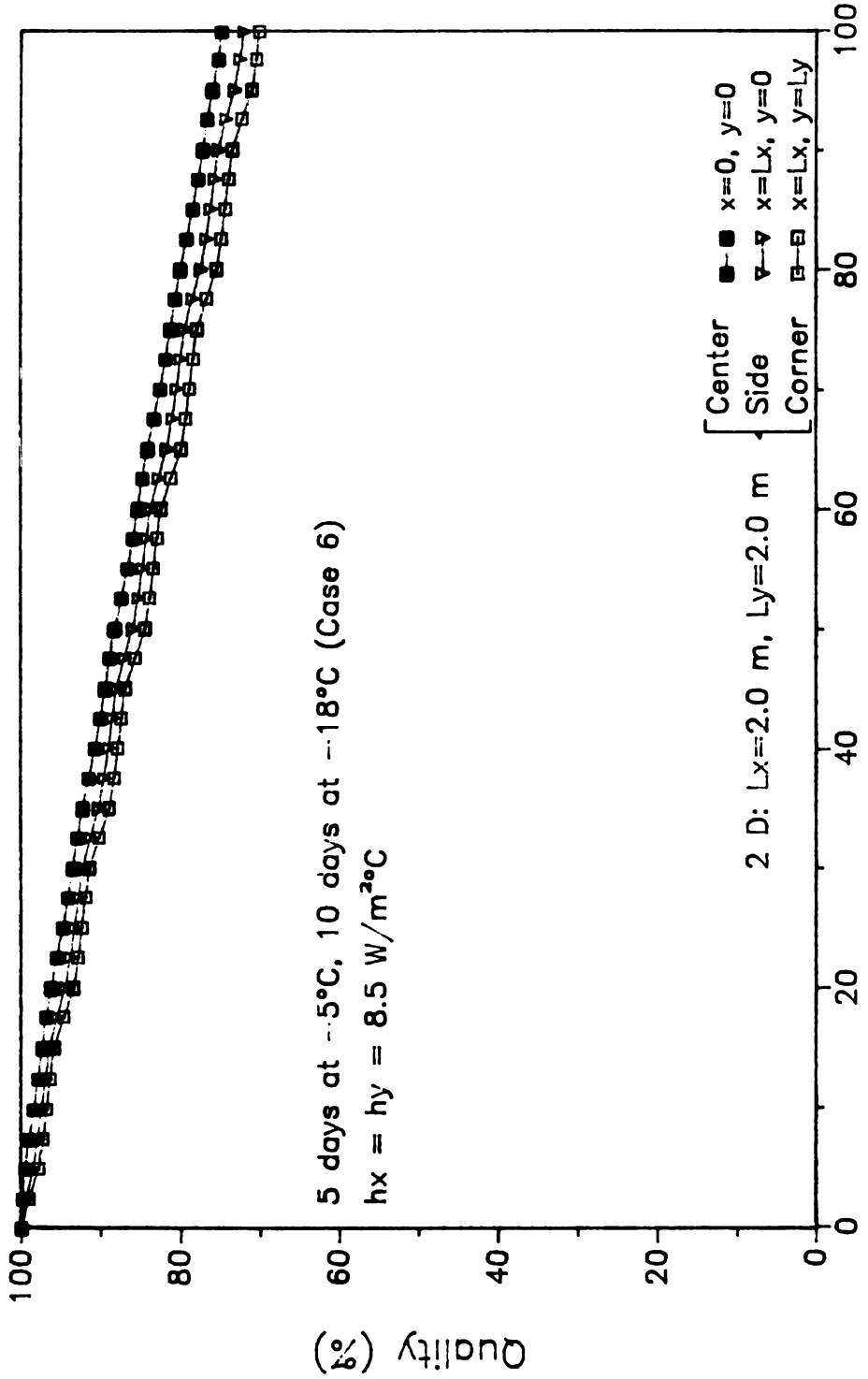


Figure 6.44 Quality Deterioration Rate Resulting from Two Dimensional Solution for Square Rod (Lx = Ly = 2.0 m) Compared to One Dimensional Solution for a Slab with Equal Dimension, Lx, for Product with Initial Temperature of -30°C (Case 6; Ea = 49 kJ/mole, Shelf-life at -18°C = 540 days).

6.6.3 Effects of Geometrical Shape

Different geometrical shapes were compared to determine if a two dimensional model might be better approximated by a one dimensional model of a different geometric shape, than that found for the one dimensional model using the same geometry in Section 6.6.2. Heat transfer through a two dimensional square rod and a one dimensional cylinder were used in the comparison. Two criteria were used to determine the radius (R) of the cylinder: (1) shapes of equal surface area; and, (2) shapes of equal volume. In both cases, no heat transfer was assumed along the axis of the rod and the cylinder, and the surface area and volume were calculated using unit length along this axis. Two dimensional heat transfer through a square rod with dimensions 2 m by 2 m (Figure 6.37). Uniform boundary conditions, of five days at -5°C, and ten days at -18°C (Case 6), were imposed on all surfaces perpendicular to the direction of heat transfer, and an insulated boundary was imposed in the axial direction. This was the same problem solved for first in Section 6.5.2, and shown in Figure 6.38. The radius of a cylinder with equal surface area was found to be 1.273 m. The solution for one dimensional heat transfer through the cylinder is compared to the solution of the two dimensional rod in Figure 6.47. The solutions at the geometric centers for both geometries were very similar; and the solution at the outer surface of the cylinder was approximately the average of the solutions at the midpoint of the sides and the corners for the square rod.

The quality histories were determined using the activation energy constant of 182 kJ/mole, and compared at the same locations. Results are shown in Figure 6.48. The difference between the quality histories predicted for the rod and the cylinder at the geometric center, was less than 2% at the end of 100 days in simulated storage. The solution at

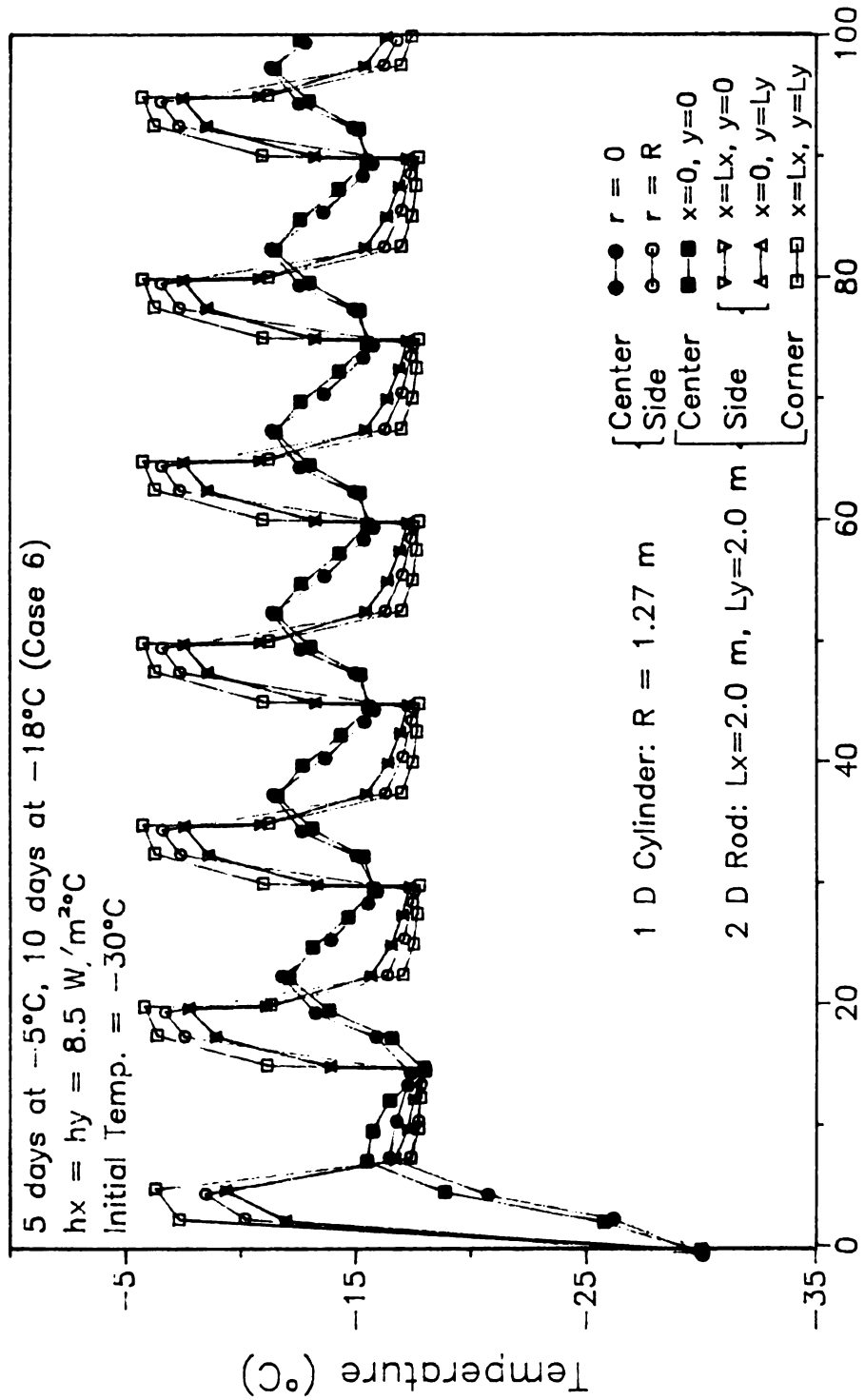


Figure 6.47 Comparison of Solutions for Two Dimensional Heat Transfer Through a Square Rod and One Dimemsional Heat Transfer Through a Cylinder of Equal Surface Area (Case 6).

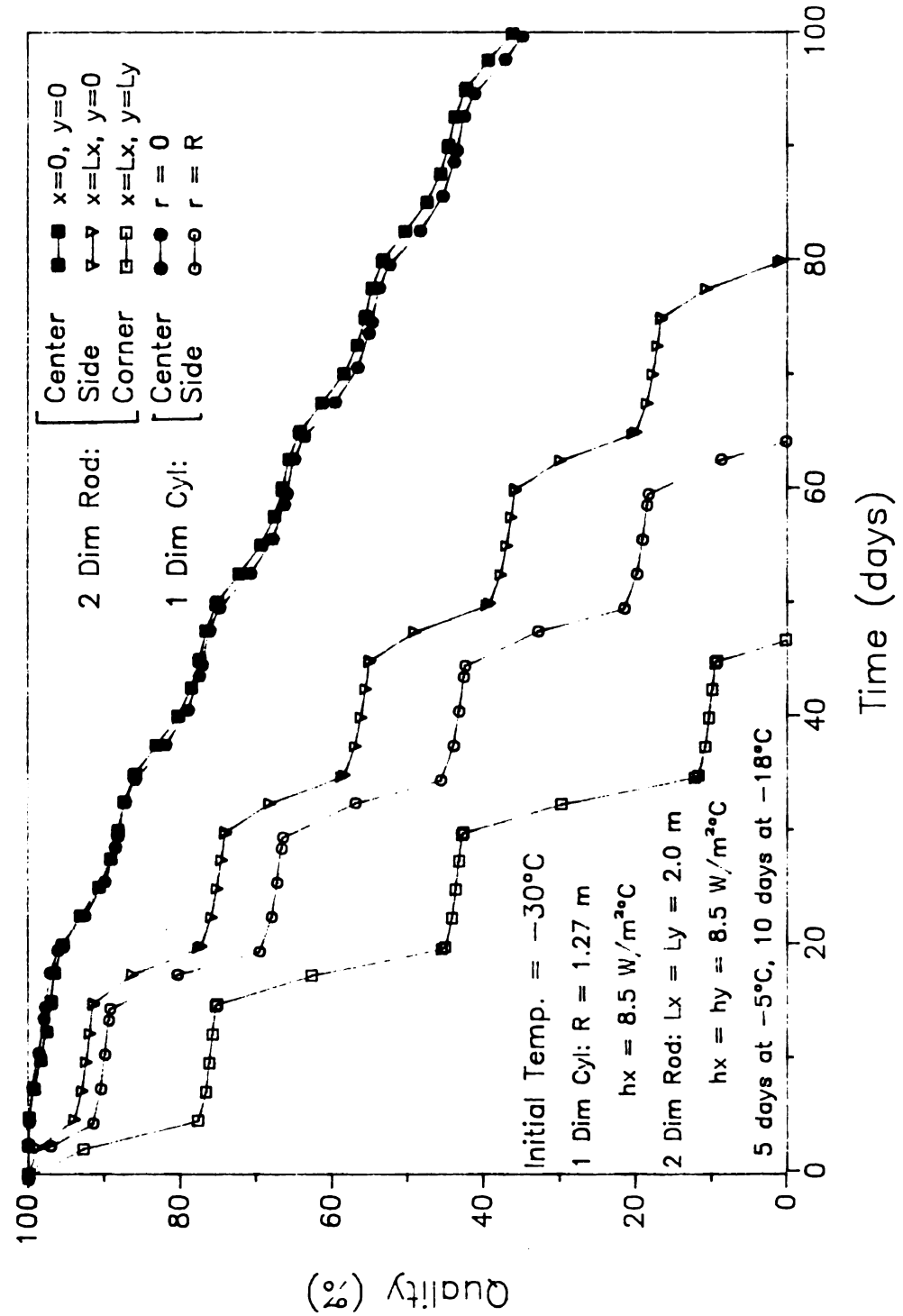


Figure 6.48 Quality Deterioration Rate Resulting from Two Dimensional Heat Transfer Through a Square Rod and One Dimensional Heat Transfer Through a Cylinder of Equal Surface Area (Case 6).

the outer surface of the cylinder, was between the two solutions at the side midpoints and the corners of the rod. This solution provided an estimate of the average quality at the surface, by under estimating the quality along the sides, and over estimating the quality at the corners.

This comparison was repeated using the equivalent volume criterion. The radius satisfying this criterion was found to be 1.128 m. The resulting temperature histories are shown in Figure 6.49. The smaller radius resulted in a poorer estimate of the temperature at the geometric center, than that found using the equal surface area criterion. On the other hand, the temperature at the surface of the cylinder was almost the same as that found using R = 1.273 m. The predicted quality profiles at the geometric center and the outer surface are compared with the solution for the rod in Figure 6.50. (The solution for the rod is the same as that shown in Figure 6.48.) Comparing Figures 6.48 and 6.49, the equal surface area criterion resulted in a better estimation of the quality history of the two dimensional rod, than either the equal volume criterion or the one dimensional solution using the same geometry (Section 6.6.2), due to the better estimation of the temperature at the geometric center, and the outer surface for the solution using the same geometry. A 10% difference in quality at the geometric center was estimated using the volume criterion, compared to only a 2% error using the equal surface area criterion.

Therefore, using one dimensional heat transfer through a cylinder provided a good model for the estimation of the quality at the geometric center and the average quality around the perimeter of a square rod, with uniform boundary conditions on four adjacent sides, and insulated on the other sides. The equivalent surface area criterion provided a better estimate of the quality at the geometric center than either the equivalent volume criterion or using the same geometry. Extending this

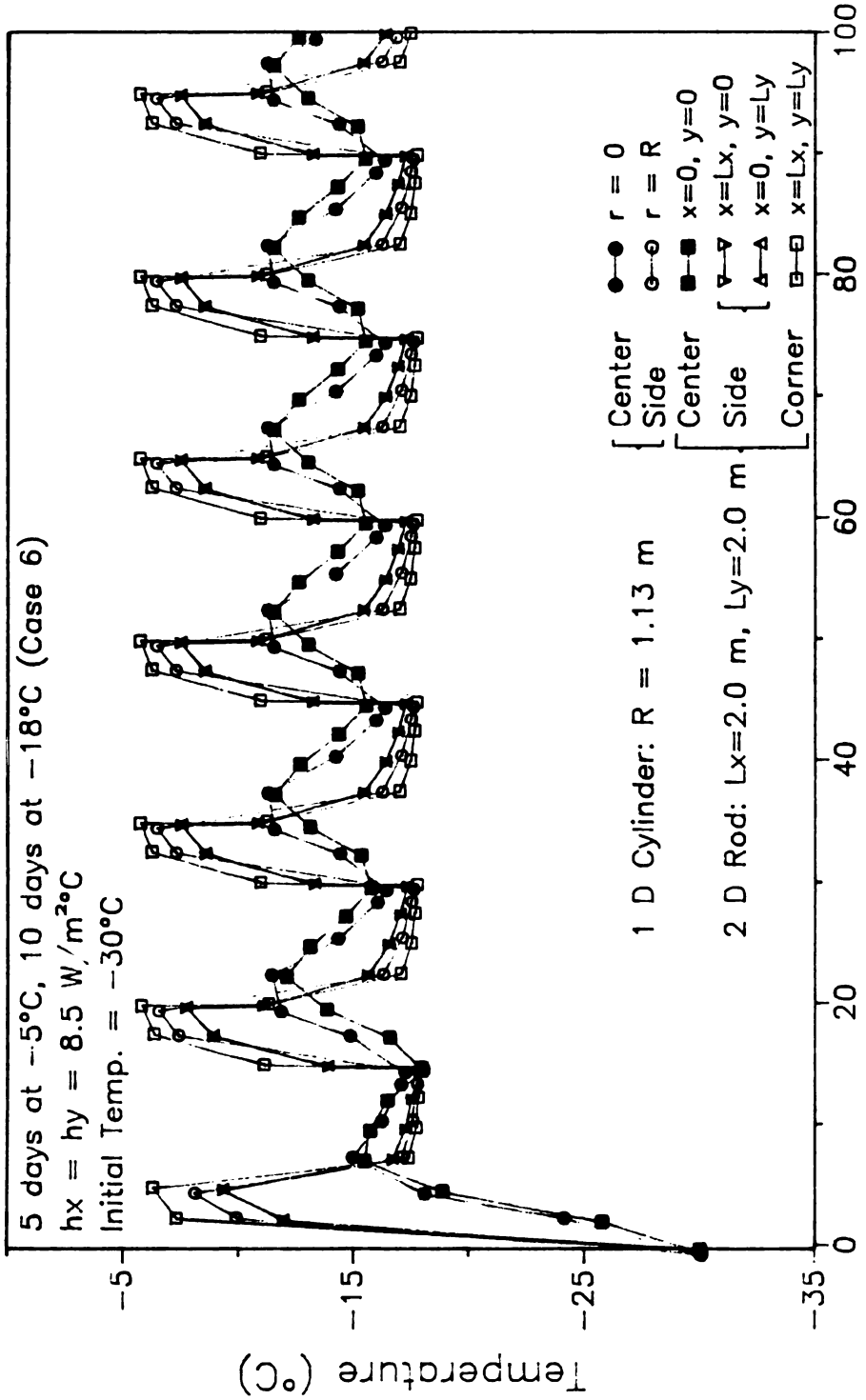


Figure 6.49 Comparison of Solutions for Two Dimensional Heat Transfer Through a Square Rod and One Dimensional Heat Transfer Through a Cylinder of Equal Volume (Case 6).

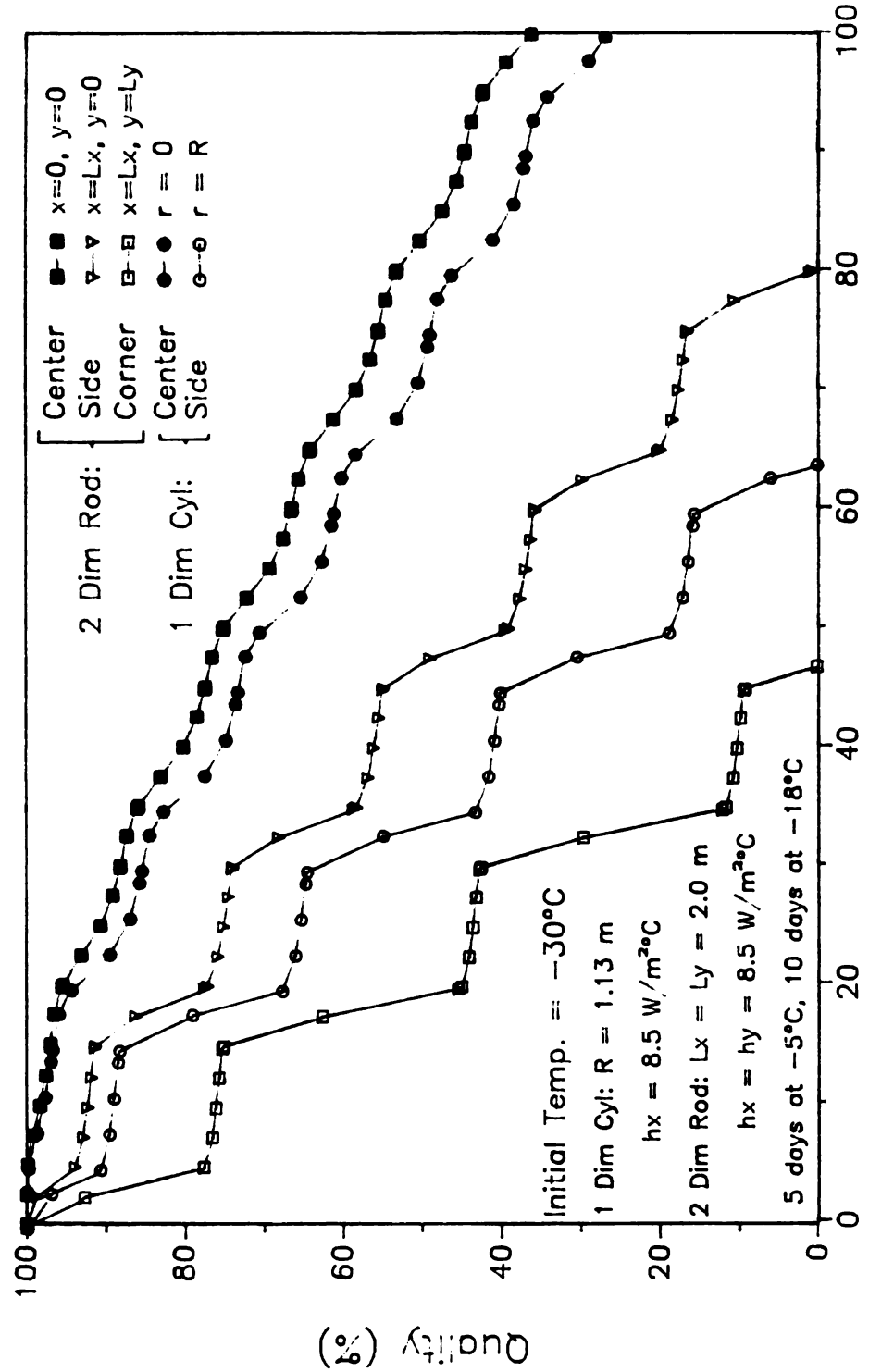


Figure 6.50 Quality Deterioration Rate Resulting from Two Dimensional Heat Transfer Through a Square Rod and One Dimensional Heat Transfer Through a Cylinder of Equal Volume (Case 6).

application to a low activation energy product would result in a better estimate of the surface quality, since, as shown in Figure 6.44, there was very little difference in the quality around the surface of the rod using the PSL criterion. This leads to a means of quickly estimating the quality profile of a large mass of product. Running the one dimensional heat transfer program for the cylindrical geometry, required approximately eight minutes of CPU time on a VAX 11/750 computer, while running the two dimensional program on the same computer took over six hours of CPU time. This concept could be extended to approximate three dimensional heat transfer through a rectangular cube by using a two dimensional finite cylinder. This method, however, should only used for uniform boundary conditions around the outer surfaces of a square rod, and may not provide accurate results if extended to other situations, such as, non-uniform boundary conditions and unequal sides.

CHAPTER 7.

SUMMARY AND CONCLUSIONS

A finite difference model, including temperature dependent thermal properties, was developed to simulate one dimensional heat transfer through frozen foods exposed to step changes in temperature storage conditions. The resulting temperature distribution histories were used to predict quality retention at different locations within the product, based on a temperature dependent quality deterioration rate constant, and a reference shelf-life. The one dimensional finite difference model was modified to simulate two dimensional heat flow, by utilizing the Crank-Nicolson approximation in an Alternating Direction Implicit finite difference model. Two dimensional quality profiles were estimated using this model.

The temperature simulation models were verified by comparison with analytical solutions using constant thermal properties, and with experimental temperature measurements, obtained in controlled storage conditions, using the Karlsruhe Test Substance, a highly concentrated methyl-cellulose mixture (Gutschmidt, 1960), as an analog for the food product. The interface between two interior product packages was found to increase the resistance to heat transfer within the product mass, resulting in a higher temperature differential between interior and exterior parts of the product, than found with the solution with no interface. Therefore, the packaging interface reduces the quality

deterioration rate at the interior of the product, and increases in quality deterioration rate in the exterior portions of the product.

The surface heat transfer coefficients prevailing during step changes in temperature storage conditions were estimated as a function of time using the sequential regularization method of estimating the surface heat flux from internal product temperature measurements, again using the Karlsruhe Test Substance. The one dimensional direct finite difference program was utilized in the solution.

The effects of the boundary conditions, size, geometrical shape, and the activation energy constant on the temperature and quality distribution histories were studied. The following conclusions were drawn from this investigation.

1. The sequential regularization method provided estimates of the transient surface heat transfer coefficients which included the effects of the exterior packaging layer, and the accumulation and diminution of frost on the outer surface.

2. The interior product packaging interface increased the resistance to heat flow within the product. This resulted in a higher temperature differential, and a potentially higher quality differential within the total product mass, than predicted for by using the assumption of negligible internal packaging resistance to heat transfer.

3. Variations in storage conditions affected the retention of quality within the product. Higher surface heat transfer coefficients resulted in a lower quality retention; changes at lower magnitudes of the surface heat transfer coefficient (1.0 - 8.5 $W/m^2°C$) had greater influence on the results than changes at higher magnitudes (8.5 - 20 $W/m^2°C$). Step

changes in storage temperature from -18°C to -5°C for one day or less, and temperature fluctuations between -18°C and -13°C had very little effect on the retention of product quality.

4. The magnitude of change in quality retention was highly dependent on the magnitude of of the activation energy constant. In most situations of step changes in ambient conditions, small internal variations in quality can be expected for products with low activation energy constants (< 60 kJ/mole), suggesting that average product temperatures (lumped capacitance solution) could be used to estimate quality histories. Products with activation energies above 60 kJ/mole are considered to be sensitive to temperature variations within the product, such that the lumped capacitance solution should not be used. Products with high activation energy constants (> 120 kJ/mole) are very sensitive to variations in storage conditions, such that variations in storage conditions would result in large quality differentials within the product mass.

5. Quality differences within the product mass for low activation energy constant products (< 60 kJ/mole), assuming one dimensional heat transfer and step changes in storage conditions, were significant (> 10% variation) for very large product masses (≥ 6.0 m), using a surface heat transfer coefficient of 8.5 $W/m^2°C$, and ambient temperature < -5°C. Products with high activation energy constants had insignificant quality differences within the product only when the product thickness was very small (≤ 0.2 m), using identical boundary conditions. For two dimensional heat transfer in low activation energy constant products, the quality differences within the product have limited dependence on the width versus height ratio, assuming identical boundary conditions in

both directions of heat flow. For high activation energy products under the same conditions, the quality differences are insignificant only for small width versus height ratios ($\leq$ 10%).

6. The quality distribution history resulting from two dimensional heat transfer through a rectangular rod can be approximated within 5% accuracy by the one dimensional model heat transfer and quality retention, for an infinite slab of equal width, only if the height versus width ratio for the rectangular slab is greater than or equal to 10:1.

7. The quality retention resulting from two dimensional heat transfer through a square rod can be approximated by the one dimensional heat transfer and quality retention model, for an infinite cylinder of equal surface area, with over 90% reduction in computation time. The equal surface area criterion provided a better estimate of the two dimensional model than either a cylindrical geometry with equal volume, or the equivalent one dimensional rectangular geometry. The cylindrical model provided an excellent estimation of the quality at the geometric center, and the average quality around the perimeter of the square rod, however, the model greatly over-estimated the quality at the corners of the rod, given a product with a high activation energy constant (<120 kJ/mole).

APPENDIX A

APPENDIX A

EQUATIONS FOR BOUNDARY NODES IN TWO DIMENSIONAL FINITE DIFFERENCE SOLUTION

The following equations were derived for the corresponding boundary nodes shown in Figure A.1 for an infinite rod and a solid cylinder, using the Douglas-Gunn alternating direction implicit finite difference scheme (Douglas and Gunn, 1964). Eight unique equations were required to determine all of the temperature values at the boundaries.

1. $\ell = 1, m = 1$

Step 1. Sweep in x-direction.

$$- \left[\frac{\eta_x}{2} \left[hx^{n+\xi}_{Lx_0} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] + ((\rho Cp)_{+x} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T^{n+\xi}_{1,1}$$

$$+ \eta_x \frac{k_{+x} A_{+x}}{2\Delta x} T^{n+\xi}_{2,1}$$

$$= \left[\frac{\beta_x}{2} \left[hx^{n}_{Lx_0} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] + hy^{n}_{0} \cdot \frac{A_y}{2} + \frac{k_{+y} A_{+y}}{2\Delta y} - ((\rho Cp)_{+x} \right.$$

$$\left. + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T^{n}_{1,1} - \frac{k_{+y} A_{+y}}{2\Delta y} T^{n}_{1,2} - \beta_x \frac{k_{+x} A_{+x}}{2\Delta x} T^{n}_{2,1}$$

$$- (\beta_x \cdot hx^{n}_{Lx_0} \cdot T^{n}_{\infty} + \eta_x \cdot hx^{n+\xi}_{Lx_0} \cdot T^{n+\xi}_{\infty}) \frac{A_x}{2} - \frac{hy^{n}_{0} A_x}{2} \cdot T^{n}_{\infty} \qquad \text{(A.1a)}$$

Step 2. Sweep in the y-direction.

$$- \left[\frac{\eta_y}{2} \left[hy_0^n \cdot A_y + \frac{k_{+y} A_{+y}}{\Delta y} \right] + ((\rho Cp)_{+x} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T_{1,1}^{n+1}$$

$$+ \eta_y \frac{k_{+y} A_{+y}}{2\Delta y} T_{1,2}^{n+1}$$

$$- \frac{\eta_x}{2} \left[hx_{Lx_0}^{n+\xi} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] T_{1,1}^{n+\xi} - \eta_x \frac{k_{+x} A_{+x}}{2\Delta x} T_{2,1}^{n+\xi}$$

$$+ \left[\frac{\beta_x}{2} \left[hx_{Lx_0}^{n} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] + \frac{\beta_y}{2} \left[hy_0^n \cdot A_y + \frac{k_{+y} A_{+y}}{\Delta y} \right] \right.$$

$$\left. - ((\rho Cp)_{+x} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T_{1,1}^{n} - \beta_y \frac{k_{+y} A_{+y}}{2\Delta y} T_{1,2}^{n} - \beta_x \frac{k_{+x} A_{+x}}{2\Delta x} T_{2,1}^{n}$$

$$- (\beta_x \cdot hx_{Lx_0}^{n} \cdot T_\infty^n + \eta_x \cdot hx_{Lx_0}^{n+\xi} \cdot T_\infty^{n+\xi}) \frac{A_x}{2}$$

$$- (\beta_y \cdot hy_0^n \cdot T_\infty^n + \eta_y \cdot hy_0^{n+1} \cdot T_\infty^{n+1}) \frac{A_y}{2} \qquad \text{(A.1b)}$$

2. $\ell = 1,\ 1 < m < M$

Step 1. Sweep in x-direction.

$$- \left[\eta_x \left[hx_{Lx_0}^{n+\xi} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] + ((\rho Cp)_{+x} + ((\rho Cp)_{-y} + (\rho Cp)_{+y}))/2) \right.$$

$$\left. \cdot \frac{A_x A_y}{4\Delta t} \right] T_{1,m}^{n+\xi} + \eta_x \frac{k_{+x} A_{+x}}{\Delta x} T_{2,m}^{n+\xi}$$

Step 2. Sweep in the y-direction.

$$
\begin{aligned}
&- \left[\frac{\eta_y}{2} \left[hy_0^n \cdot A_y + \frac{k_{+y} A_{+y}}{\Delta y} \right] + ((\rho Cp)_{+x} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T_{1,1}^{n+1} \\
&\quad + \eta_y \frac{k_{+y} A_{+y}}{2\Delta y} T_{1,2}^{n+1} \\
&- \frac{\eta_x}{2} \left[hx_{Lx_0}^{n+\xi} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] T_{1,1}^{n+\xi} - \eta_x \frac{k_{+x} A_{+x}}{2\Delta x} T_{2,1}^{n+\xi} \\
&\quad + \left[\frac{\beta_x}{2} \left[hx_{Lx_0}^{n} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] + \frac{\beta_y}{2} \left[hy_0^n \cdot A_y + \frac{k_{+y} A_{+y}}{\Delta y} \right] \right. \\
&\quad \left. - ((\rho Cp)_{+x} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T_{1,1}^{n} - \beta_y \frac{k_{+y} A_{+y}}{2\Delta y} T_{1,2}^{n} - \beta_x \frac{k_{+x} A_{+x}}{2\Delta x} T_{2,1}^{n} \\
&\quad - (\beta_x \cdot hx_{Lx_0}^{n} \cdot T_\infty^n + \eta_x \cdot hx_{Lx_0}^{n+\xi} \cdot T_\infty^{n+\xi}) \frac{A_x}{2} \\
&\quad - (\beta_y \cdot hy_0^{n} \cdot T_\infty^n + \eta_y \cdot hy_0^{n+1} \cdot T_\infty^{n+1}) \frac{A_y}{2} \qquad \text{(A.1b)}
\end{aligned}
$$

2. $\ell = 1,\ 1 < m < M$

Step 1. Sweep in x-direction.

$$
\begin{aligned}
&- \left[\eta_x \left[hx_{Lx_0}^{n+\xi} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] + ((\rho Cp)_{+x} + ((\rho Cp)_{-y} + (\rho Cp)_{+y}))/2) \right. \\
&\quad \left. \cdot \frac{A_x A_y}{4\Delta t} \right] T_{1,m}^{n+\xi} + \eta_x \frac{k_{+x} A_{+x}}{\Delta x} T_{2,m}^{n+\xi}
\end{aligned}
$$

$$= -\frac{k_{-y}A_{-y}}{2\Delta y}T^{n}_{1,m-1} + \left[\beta_x\left[hx^{n}_{Lx_0}\cdot A_x + \frac{k_{+x}A_{+x}}{\Delta x}\right] + \frac{k_{-y}A_{-y} + k_{+y}A_{+y}}{2\Delta y}\right.$$

$$\left. - ((\rho Cp)_{+x} + ((\rho Cp)_{-y} + (\rho Cp)_{+y}))/2)\cdot\frac{A_xA_y}{4\Delta t}\right]T^{n}_{1,m} - \frac{k_{+y}A_{+y}}{2\Delta y}T^{n}_{1,m+1}$$

$$- \beta_x\frac{k_{+x}A_{+x}}{\Delta x}T^{n}_{2,m} - (\beta_x\cdot hx^{n}_{Lx_0}\cdot T^{n}_{\infty} + \eta_x\cdot hx^{n+\xi}_{Lx_0}\cdot T^{n+\xi}_{\infty})A_x$$

(A.2a)

Step 2. Sweep in the y-direction.

$$\eta_y\frac{k_{-y}A_{-y}}{2\Delta y}T^{n+1}_{1,m-1} - \left[\eta_y\frac{k_{-y}A_{-y} + k_{+y}A_{+y}}{2\Delta y} + ((\rho Cp)_{+x} + ((\rho Cp)_{-y}\right.$$

$$\left. + (\rho Cp)_{+y})/2)\cdot\frac{A_xA_y}{4\Delta t}\right]T^{n+1}_{1,m} + \eta_y\frac{k_{+y}A_{+y}}{2\Delta y}T^{n+1}_{1,m+1}$$

$$= -\eta_x\left[hx^{n+\xi}_{Lx_0}\cdot A_x + \frac{k_{+x}A_{+x}}{\Delta x}\right]T^{n+\xi}_{1,m} - \eta_x\frac{k_{+x}A_{+x}}{\Delta x}T^{n+\xi}_{2,m} - \beta_y\frac{k_{-y}A_{-y}}{2\Delta y}T^{n}_{1,m-1}$$

$$+ \left[\beta_x\left[hx^{n}_{Lx_0}\cdot A_x + \frac{k_{+x}A_{+x}}{\Delta x}\right] + \beta_y\frac{k_{-y}A_{-y} + k_{+y}A_{+y}}{2\Delta y}\right.$$

$$\left. - ((\rho Cp)_{+x} + ((\rho Cp)_{-y} + (\rho Cp)_{+y})/2)\cdot\frac{A_xA_y}{4\Delta t}\right]T^{n}_{1,m} - \beta_y\frac{k_{+y}A_{+y}}{2\Delta y}T^{n}_{1,m+1}$$

$$- \beta_x\frac{k_{+x}A_{+x}}{\Delta x}T^{n}_{2,m} - (\beta_x\cdot hx^{n}_{Lx_0}\cdot T^{n}_{\infty} + \eta_x\cdot hx^{n+\xi}_{Lx_0}\cdot T^{n+\xi}_{\infty})A_x$$

(A.2b)

3. $\ell = 1$, $m = M$

Step 1. Sweep in x-direction.

$$- \left[\frac{\eta_x}{2} \left[hx^{n+\xi}_{Lx_0} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] + ((\rho Cp)_{+x} + (\rho Cp)_{-y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T^{n+\xi}_{1,M}$$

$$+ \eta_x \frac{k_{+x} A_{+x}}{2\Delta x} T^{n+\xi}_{2,M}$$

$$= - \frac{k_{-y} A_{-y}}{2\Delta y} T^{n}_{1,M-1} + \left[\frac{\beta_x}{2} \left[hx^{n}_{Lx_0} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] + \frac{k_{-y} A_{-y}}{2\Delta y} + \frac{hy_{Ly} \cdot A_x}{2} \right.$$

$$\left. - ((\rho Cp)_{+x} + (\rho Cp)_{-y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T^{n}_{1,M} - \beta_x \frac{k_{+x} A_{+x}}{2\Delta x} T^{n}_{2,M}$$

$$- (\beta_x \cdot hx^{n}_{Lx_0} \cdot T^{n}_{\infty} + \eta_x \cdot hx^{n+\xi}_{Lx_0} \cdot T^{n+\xi}_{\infty}) A_x - \frac{hy_{Ly} \cdot A_y}{2} \cdot T^{n}_{\infty} \qquad \text{(A.3a)}$$

Step 2. Sweep in the y-direction.

$$\eta_y \frac{k_{-y} A_{-y}}{2\Delta y} T^{n+1}_{1,M-1} - \left[\frac{\eta_y}{2} \left[\frac{k_{-y} A_{-y}}{\Delta y} + hy^{n+1}_{Ly} \cdot A_y \right] + ((\rho Cp)_{+x} + (\rho Cp)_{-y}) \right.$$

$$\left. \cdot \frac{A_x A_y}{8\Delta t} \right] T^{n+1}_{1,M}$$

$$= \frac{\eta_x}{2} \left[hx^{n+\xi}_{Lx_0} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] T^{n+\xi}_{1,M} - \eta_x \frac{k_{+x} A_{+x}}{2\Delta x} \cdot T^{n+\xi}_{2,M} - \beta_y \frac{k_{-y} A_{-y}}{2\Delta y} T^{n}_{1,M-1}$$

$$+ \left[\frac{\beta_x}{2} \left[hx^{n}_{Lx_0} \cdot A_x + \frac{k_{+x} A_{+x}}{\Delta x} \right] + \frac{\beta_y}{2} \left[\frac{k_{-y} A_{-y}}{\Delta y} + hy^{n}_{Ly} \cdot A_y \right] \right.$$

$$\left. - ((\rho Cp)_{+x} + (\rho Cp)_{-y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T^{n}_{1,M} - \beta_x \frac{k_{+x} A_{+x}}{2\Delta x} T^{n}_{2,M}$$

$$- (\beta_x \cdot hx^n_{Lx_0} \cdot T^n_\infty + \eta_x \cdot hx^{n+\xi}_{Lx_0} \cdot T^{n+\xi}_\infty) \frac{A_x}{2}$$

$$- (\beta_y \cdot hy^n_{Ly} \cdot T^n_\infty + \eta_y \cdot hy^{n+1}_{Ly} \cdot T^{n+1}_\infty) \frac{A_y}{2} \qquad \text{(A.3b)}$$

4. $1 < \ell < L$, $m = 1$

Step 1. Sweep in x-direction.

$$\eta_x \frac{k_{-x} A_{-x}}{2\Delta x} T^{n+\xi}_{\ell-1,1} - \left[\eta_x \frac{k_{-x} A_{-x} + k_{+x} A_{+x}}{2\Delta x} + (((\rho Cp)_{-x} + (\rho Cp)_{+x})/2 \right.$$

$$\left. + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{4\Delta t} \right] T^{n+\xi}_{\ell,1} + \eta_x \frac{k_{+x} A_{+x}}{2\Delta x} T^{n+\xi}_{\ell+1,1}$$

$$= -\beta_x \frac{k_{-x} A_{-x}}{2\Delta x} T^n_{\ell-1,1} + \left[\beta_x \frac{k_{-x} A_{-x} + k_{+x} A_{+x}}{2\Delta x} + hy_1 \cdot A_y + \frac{k_{+y} A_{+y}}{\Delta y} \right.$$

$$\left. - (((\rho Cp)_{-x} + (\rho Cp)_{+x})/2 + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{4\Delta t} \right] T^n_{\ell,1} - \frac{k_{+y} A_{+y}}{\Delta y} T^n_{\ell,2}$$

$$- \beta_x \frac{k_{+x} A_{+x}}{2\Delta x} T^n_{\ell+1,1} - hy^n_0 A_y \cdot T^n_\infty \qquad \text{(A.4a)}$$

Step 2. Sweep in the y-direction.

$$- \left[\eta_y \left[hy^{n+1}_0 \cdot A_y + \frac{k_{+y} A_{+y}}{\Delta y} \right] + (((\rho Cp)_{-x} + (\rho Cp)_{+x})/2 + (\rho Cp)_{+y} \right.$$

$$\left. \cdot \frac{A_x A_y}{4\Delta t} \right] T^{n+1}_{\ell,1} + \eta_y \frac{k_{+y} A_{+y}}{\Delta y} T^{n+1}_{\ell,2}$$

$$= - \eta_x \frac{k_{-x} A_{-x}}{2\Delta x} T^{n+\xi}_{\ell-1,1} + \eta_x \frac{k_{-x} A_{-x} + k_{+x} A_{+x}}{2\Delta x} \cdot T^{n+\xi}_{\ell,1} - \eta_x \frac{k_{+x} A_{+x}}{2\Delta x} T^{n+\xi}_{\ell+1,1}$$

$$-\beta_x \frac{k_{-x}A_{-x}}{2\Delta x} T^n_{\ell-1,1} + \left[\beta_x \frac{k_{-x}A_{-x} + k_{+x}A_{+x}}{2\Delta x} + \beta_y \left[hy^n_0 \cdot A_y + \frac{k_{+y}A_{+y}}{\Delta y} \right] \right.$$

$$\left. - (((\rho Cp)_{-x} + (\rho Cp)_{+x})/2 + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{4\Delta t} \right] T^n_{\ell,1} - \beta_y \frac{k_{+y}A_{+y}}{\Delta y} T^n_{\ell,2}$$

$$- \beta_x \frac{k_{+x}A_{+x}}{2\Delta x} T^n_{\ell+1,1} - (\beta_y hy^n_0 \cdot T^n_\infty + \eta_y hy^{n+1}_0 \cdot T^{n+1}_\infty) A_y \qquad \text{(A.4b)}$$

5. $1 < \ell < L$, $m = M$

Step 1. Sweep in x-direction.

$$\eta_x \frac{k_{-x}A_{-x}}{2\Delta x} T^{n+\xi}_{\ell-1,M} - \left[\eta_x \frac{k_{-x}A_{-x} + k_{+x}A_{+x}}{2\Delta x} + (((\rho Cp)_{-x} + (\rho Cp)_{+x})/2 \right.$$

$$\left. + (\rho Cp)_{-y}) \cdot \frac{A_x A_y}{\Delta t} \right] T^{n+\xi}_{\ell,M} + \eta_x \frac{k_{+x}A_{+x}}{2\Delta x} T^{n+\xi}_{\ell+1,M}$$

$$= -\beta_x \frac{k_{-x}A_{-x}}{2\Delta x} T^n_{\ell-1,M} - \frac{k_{-y}A_{-y}}{\Delta y} T^n_{\ell,M-1} + \left[\beta_x \frac{k_{-x}A_{-x} + k_{+x}A_{+x}}{2\Delta x} + \frac{k_{-y}A_{-y}}{\Delta y} \right.$$

$$\left. + hy^n_{Ly}A_y - (((\rho Cp)_{-x} + (\rho Cp)_{+x})/2 + (\rho Cp)_{-y}) \cdot \frac{A_x A_y}{4\Delta t} \right] \cdot T^n_{\ell,M}$$

$$- \beta_x \frac{k_{+x}A_{+x}}{2\Delta x} T^n_{\ell+1,M} - hy^n_{Ly}A_y \cdot T^n_\infty \qquad \text{(A.5a)}$$

Step 2. Sweep in the y-direction.

$$n_y \frac{k_{-y}A_{-y}}{\Delta y} T^{n+1}_{\ell,M-1} - \left[\eta_y \left[\frac{k_{-y}A_{-y}}{\Delta y} + hy^{n+1}_{Ly} \cdot A_y \right] + (((\rho Cp)_{-x} + (\rho Cp)_{+x})/2 \right.$$

$$\left. + (\rho Cp)_{-y}) \cdot \frac{A_x A_y}{4\Delta t} \right] T^{n+1}_{\ell,M}$$

$$= -\eta_x \frac{k_{-x}A_{-x}}{2\Delta x} T^{n+\xi}_{\ell-1,M} + \beta_x \frac{k_{-x}A_{-x} + k_{+x}A_{+x}}{2\Delta x} T^{n+\xi}_{\ell,M} - \eta_x \frac{k_{+x}A_{+x}}{2\Delta x} T^{n+\xi}_{\ell-1,M}$$

$$- \beta_x \frac{k_{-x}A_{-x}}{2\Delta x} T^{n}_{\ell-1,M} - \beta_y \frac{k_{-y}A_{-y}}{\Delta y} T^{n}_{\ell,M-1} + \left[\beta_x \frac{k_{-x}A_{-x} + k_{+x}A_{+x}}{2\Delta x} \right.$$

$$+ \beta_y \cdot \left[\frac{k_{-y}A_{-y}}{\Delta y} + hy^n_{Ly} \cdot A_y \right] - (((\rho Cp)_{-x} + (\rho Cp)_{+x})/2 + (\rho Cp)_{-y})$$

$$\left. \cdot \frac{A_x A_y}{4\Delta t} \right] T^{n}_{\ell,M} - \beta_x \frac{k_{+x}A_{+x}}{2\Delta x} T^{n}_{\ell-1,M} - (\beta_y \cdot hy^n_{Ly} \cdot T^n_\infty$$

$$+ \eta_y \cdot hy^{n+1}_{Ly} \cdot T^{n+1}_\infty) A_y \qquad \text{(A.5b)}$$

6. $\ell = L$, $m = 1$

Step 1. Sweep in x-direction.

$$\eta_x \frac{k_{-x}A_{-x}}{2\Delta x} T^{n+\xi}_{L-1,1} - \left[\frac{\eta_x}{2} \left[\frac{k_{-x}A_{-x}}{\Delta x} + hx^{n+\xi}_{Lx} \cdot A_x \right] + ((\rho Cp)_{-x} + (\rho Cp)_{+y}) \right.$$

$$\left. \cdot \frac{A_x A_y}{8\Delta t} \right] T^{n+\xi}_{L,1}$$

$$= -\beta_x \frac{k_{-x}A_{-x}}{2\Delta x} T^{n}_{L-1,1} + \left[\frac{\beta_x}{2} \left[\frac{k_{-x}A_{-x}}{\Delta x} + hx^n_{Lx} \cdot A_x \right] + \frac{hy_0 \cdot A_y}{2} + \frac{k_{+y}A_{+y}}{2\Delta y} \right.$$

$$\left. - ((\rho Cp)_{-x} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T^{n}_{L,1} - \frac{k_{+y}A_{+y}}{2\Delta y} T^{n}_{L,2}$$

$$- (\beta_x \cdot hx^n_{Lx} \cdot T^{n+\xi}_\infty + \eta_x \cdot hx^{n+\xi}_2 \cdot T_\infty) \frac{A_x}{2} - \frac{hy^n_0 A_x}{2} \cdot T^n_\infty$$

(A.6a)

Step 2. Sweep in the y-direction.

$$- \left[\frac{\eta_y}{2} \left[hy_0^{n+1} \cdot A_y + \frac{k_{+y} A_{+y}}{\Delta y} \right] + ((\rho Cp)_{-x} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T_{L,1}^{n+1}$$

$$+ \eta_y \frac{k_{+y} A_{+y}}{2\Delta y} T_{L,2}^{n+1}$$

$$= - \eta_x \frac{k_{-x} A_{-x}}{2\Delta x} T_{L-1,1}^{n+\xi} + \frac{\eta_x}{2} \left[\frac{k_{-x} A_{-x}}{\Delta x} + hx_{Lx}^{n+\xi} \cdot A_x \right] T_{L,1}^{n+\xi}$$

$$- \beta_x \frac{k_{-x} A_{-x}}{2\Delta x} T_{L-1,1}^{n} + \left[\frac{\beta_x}{2} \left[\frac{k_{-x} A_{-x}}{\Delta x} + hx_{Lx}^{n} \cdot A_x \right] \right.$$

$$\left. + \frac{\beta_y}{2} \left[hy_0^{n} \cdot A_y + \frac{k_{+y} A_{+y}}{2\Delta y} \right] - ((\rho Cp)_{-x} + (\rho Cp)_{+y}) \cdot \frac{A_x A_y}{8\Delta t} \right] T_{L,1}^{n}$$

$$- \beta_y \frac{k_{+y} A_{+y}}{2\Delta y} T_{L,2}^{n} - (\beta_x \cdot hx_{Lx}^{n} \cdot T_\infty^{n} + \eta_x \cdot hx_{Lx}^{n+\xi} \cdot T_\infty^{n+\xi}) \frac{A_x}{2}$$

$$- (\beta_y hy_0^{n} \cdot T_\infty^{n} + \eta_y hy_0^{n+1} \cdot T_\infty^{n+1}) \frac{A_y}{2} \qquad \text{(A.6b)}$$

7. $\ell = L,\ 1 < m < M$

Step 1. Sweep in x-direction.

$$\eta_x \frac{k_{-x} A_{-x}}{\Delta x} T_{L-1,m}^{n+\xi} - \left[\eta_x \left[\frac{k_{-x} A_{-x}}{\Delta x} + hx_{Lx}^{n+\xi} \cdot A_x \right] + ((\rho Cp)_{-x} \right.$$

$$\left. + ((\rho Cp)_{-y} + (\rho Cp)_{+y}))/2) \cdot \frac{A_x A_y}{4\Delta t} \right] T_{L,m}^{n+\xi} +$$

$$= - \beta_x \frac{k_{-x} A_{-x}}{\Delta x} T_{L+1,m}^{n} - \frac{k_{-y} A_{-y}}{2\Delta y} T_{L,m-1}^{n} + \left[\beta_x \left[\frac{k_{-x} A_{-x}}{\Delta x} + hx_{Lx}^{n} \cdot A_x \right] \right.$$

$$+ \frac{k_{-y}A_{-y} + k_{+y}A_{+y}}{2\Delta y} - ((\rho Cp)_{-x} + ((\rho Cp)_{-y} + (\rho Cp)_{+y}))/2)$$

$$\cdot \frac{A_x A_y}{4\Delta t} \Big] T^n_{L,m} - \frac{k_{+y}A_{+y}}{2\Delta y} T^n_{L,m+1} - (\beta_x \cdot hx^n_{Lx} \cdot T^n_\infty + \eta_x \cdot hx^{n+\xi}_{Lx} T^{n+\xi}_\infty) A_x$$

(A.7a)

Step 2. Sweep in the y-direction.

$$\eta_y \frac{k_{-y}A_{-y}}{2\Delta y} T^{n+1}_{L,m-1} - \Big[\eta_y \frac{k_{-y}A_{-y} + k_{+y}A_{+y}}{2\Delta y} + ((\rho Cp)_{-x} + ((\rho Cp)_{-y}$$

$$+ (\rho Cp)_{+y})/2) \cdot \frac{A_x A_y}{4\Delta t} \Big] T^{n+1}_{L,m} + \eta_y \frac{k_{+y}A_{+y}}{2\Delta y} T^{n+1}_{L,m+1}$$

$$= - \eta_x \frac{k_{-x}A_{-x}}{\Delta x} T^{n+\xi}_{L-1,m} - \eta_x \Big[\frac{k_{-x}A_{-x}}{\Delta x} + hx^{n+\xi}_{Lx} \cdot A_x \Big] T^{n+\xi}_{L,m}$$

$$- \beta_x \frac{k_{-x}A_{-x}}{\Delta x} T^n_{L-1,m} - \beta_y \frac{k_{-y}A_{-y}}{2\Delta y} T^n_{L,m-1} \Big[\beta_x \Big[\frac{k_{-x}A_{-x}}{\Delta x} + hx^n_{Lx} \cdot A_x \Big]$$

$$+ \beta_y \frac{k_{-y}A_{-y} + k_{+y}A_{+y}}{2\Delta y} - ((\rho Cp)_{-x} + ((\rho Cp)_{-y} + (\rho Cp)_{+y})/2) \cdot \frac{A_x A_y}{4\Delta t} \Big]$$

$$\cdot T^n_{L,m} - \beta_y \frac{k_{+y}A_{+y}}{2\Delta y} T^n_{L,m+1} - (\beta_x \cdot hx^n_{Lx} \cdot T^n_\infty + \eta_x \cdot hx^{n+\xi}_{Lx} \cdot T^{n+\xi}_\infty) A_x$$

(A.7b)

8. $\ell = L$, $m = M$

Step 1. Sweep in x-direction.

$$\eta_x \frac{k_{-x}A_{-x}}{2\Delta x} T^{n+\xi}_{L-1,M} - \Big[\frac{\eta_x}{2} \Big[\frac{k_{-x}A_{-x}}{\Delta x} + hx^{n+\xi}_{Lx} \cdot A_x \Big] + ((\rho Cp)_{-x} + (\rho Cp)_{-y})$$

$$\cdot\frac{A_x A_y}{8\Delta t}\Bigg]\, T^{n+\xi}_{L,M}$$

$$= -\beta_x \frac{k_{-x} A_{-x}}{2\Delta x}\, T^{n}_{L-1,M} - \frac{k_{-y} A_{-y}}{2\Delta y}\, T^{n}_{L,M-1} + \left[\frac{\beta_x}{2}\left[\frac{k_{-x} A_{-x}}{\Delta x} + hx^{n}_{Lx}\cdot A_x\right]\right.$$

$$\left. + \frac{k_{-y} A_{-y}}{2\Delta y} + \frac{hy_{Ly}\cdot A_y}{2} - ((\rho Cp)_{-x} + (\rho Cp)_{-y})\cdot\frac{A_x A_y}{8\Delta t}\right] T^{n}_{L,M}$$

$$- (\beta_x \cdot hx^{n}_{Lx}\cdot T^{n}_{\infty} + \eta_x \cdot hx^{n+\xi}_{Lx}\cdot T^{n+\xi}_{\infty})\frac{A_x}{2} - \frac{hy_2 \cdot A_y}{2}\cdot T^{n}_{\infty} \qquad \text{(A.8a)}$$

Step 2. Sweep in the y-direction.

$$\eta_y \frac{k_{-y} A_{-y}}{2\Delta y}\, T^{n+1}_{L,M-1} - \left[\frac{\eta_y}{2}\left[\frac{k_{-y} A_{-y}}{\Delta y} + hy^{n+1}_{Ly}\cdot A_y\right] + ((\rho Cp)_{-x} + (\rho Cp)_{-y})\right.$$

$$\left.\cdot\frac{A_x A_y}{8\Delta t}\right] T^{n+1}_{L,M}$$

$$= -\eta_x \frac{k_{-x} A_{-x}}{2\Delta x}\, T^{n+\xi}_{L-1,M} - \frac{\eta_x}{2}\left[\frac{k_{-x} A_{-x}}{\Delta x} + hx^{n+\xi}_{Lx}\cdot A_x\right] T^{n+\xi}_{L,M} - \beta_x \frac{k_{-x} A_{-x}}{2\Delta x}$$

$$T^{n}_{L-1,M} - \beta_y \frac{k_{-y} A_{-y}}{2\Delta y}\, T^{n}_{L,M-1} + \left[\frac{\beta_x}{2}\left[\frac{k_{-x} A_{-x}}{\Delta x} + hx^{n}_{Lx}\cdot A_x\right]\right.$$

$$\left. + \frac{\beta_y}{2}\left[\frac{k_{-y} A_{-y}}{\Delta y} + hy^{n}_{Ly}\cdot A_y\right] - ((\rho Cp)_{-x} + (\rho Cp)_{-y})\cdot\frac{A_x A_y}{8\Delta t}\right] T^{n}_{L,M}$$

$$- (\beta_x \cdot hx^{n}_{Lx}\cdot T^{n}_{\infty} + \eta_x \cdot hx^{n+\xi}_{Lx}\cdot T^{n+\xi}_{\infty})\frac{A_x}{2}$$

$$- (\beta_y \cdot hy^{n}_{Ly}\cdot T^{n}_{\infty} + \eta_y \cdot hy^{n+1}_{Ly}\cdot T^{n+1}_{\infty})\frac{A_y}{2} \qquad \text{(A.8b)}$$

APPENDIX B

APPENDIX B

ONE DIMENSIONAL TRANSIENT HEAT CONDUCTION AND QUALITY RETENTION PROGRAM

The one dimensional transient heat conduction program, including estimation of quality retention, discussed in Chapter 3, is presented here. An outline of the program is given in Table B.1, and the listing for the program, written in Fortran 77 for a Vax 11/750 is given in Table B.2.

Table B.1 Description of One Dimensional Transient Heat Conduction and Quality Retention Program.

Subroutine Title	Description
PROGRAM FREEZE	Main program; contains program menu.
SUBROUTINE PROPER	Allows interactive for thermal properties, prints out product properties as a function of temperature.
DOUBLE PRECISION FUNCTION MOIST(X)	Determines percentage of unfrozen water fraction of food product as a function of temperature below 0°C.
DOUBLE PRECISION FUNCTION DENS(X)	Determines density of food product as a function of temperature below 0°C.
DOUBLE PRECISION FUNCTION KI(X)	Determines thermal conductivity of ice as a function of temperature below 0°C.
DOUBLE PRECISION FUNCTION CONDUC(X)	Determines thermal conductivity of food product as a function of temperature below 0°C.
DOUBLE PRECISION FUNCTION SPHEAT(X)	Determines specific heat of food product as a function of temperature below 0°C.
SUBROUTINE CONSPR	Calculates constant thermal property approximations over specified temperature intervals below 0°C. Writes output to data file.
SUBROUTINE INTEGR	Determines mean property value over specified temperature interval using Gauss quadrature integration.
BLOCK DATA CONST	Defines constant thermal properties (water, ice)
SUBROUTINE INPUT1	Allows interactive input of ambient conditions and product geometry. Writes output to data file.
SUBROUTINE INPUT2	Allows interactive input of kinetic properties. Writes output to data file.
SUBROUTINE SOLN	Computes temperature distribution and quality retention as a function of temperature. Calls output subroutine.
SUBROUTINE COEFF	Determines matrix coefficients used in finite difference algorithm.

Table B.1 (cont'd).

SUBROUTINE PFIND	Finds values for thermal properties required for finite difference calculations from the property values determined in CONSPR.
SUBROUTINE OUTPUT	Writes input data and resulting temperature and quality retention values to output file.

Table B.2 Computer Code for One Dimensional Transient Heat Conduction and Quality Retention Program.

```
      PROGRAM FREEZE
c*****************************************************************
c*****************************************************************

c                    Residual Shelf-life Program

c                              by

c                          Elaine Scott

c                              1985

c*****************************************************************

c   This program calculates the temperature and quality distri-
c bution histories of a one dimensional frozen food product
c subject to fluctuating ambient temperatures during storage
c below 0C.

c   Input parameters include unfrozen product density, thermal
c conductivity and specific heat.  The initial freezing
c temperature or molecular weight of solids is required to
c predict these values for the frozen food product.

c   Boundary conditions are assumed to be convective, requiring an
c input of the ambient temperature as a function of time, and the
c convective heat transfer coefficient.  The initial condition must
c be a known function of position.

c*****************************************************************



      parameter(maxp=20,maxm=101,maxd=101,maxc=51,maxs=201)

      integer model

      double precision wf0,ms,dp,kp,cp,t0,ds,th,tl,avgd,avgk,avgc,
     &  ynavg,tdt

      character title*20,ttlfil*4,filyn1*1,filyn2*1,filyn*1,fildat*12,
     &inpdat*12

      logical itmode

      common/mod/model,/itm/itmode,/ttl/title,ttlfil,/profil/prpfil,
```

Table B.2 (cont'd).

```
     &/datfil/fildat,inpdat,kindat,/prop/wf0,ms,dp,kp,cp,t0,/d/ds,
     &/pavg/th,tl,avgd,avgk,avgc,ynavg,/toldt/tdt


C  Set ITMODE = .FALSE. if running batch.
      ITMODE = .TRUE.


      IF(ITMODE)THEN

      write(5,1000)
 1000 format('1',72('*'),/,'0',t23,'Residual Shelf-life Program',/,'0',
     &t35,'by',/,'0',t30,'Elaine Scott',/,'0',t24,'Michigan State',
     &' University',/,'0',t30,'January 1986',/,'0',72('*'))
      WRITE(5,100)
  100 FORMAT('0','Program Menu:',//,' ',' 1. Product properties (<0C)'
     &,/,' ',' 2. Temperature distribution history: known Ta and h',
     &/,' ',' 3. Temp. & qual. dist. histories: exact kinetic prop.',
     &/,' ',' 4. Temp. & qual. dist. hist.: random kinetic prop.',
     &/,'0',' Ta = Ambient temp.;h = Surface heat trans. coef.',
     &//,' ','Selection? ',$)
      ENDIF
      READ(5,10)model
   10 FORMAT(I1)
      IF(ITMODE)  write(5,200)
  200 format(' ',/,' ','Product: ',$)
      READ(5,20)TITLE
      IF(ITMODE)  write(5,300)
  300 format(' ',/,' ','Key word for data files; 4 Characters: ',$)
      READ(5,20)TTLFIL
   20 FORMAT(A)
      if(model.eq.1)then
        filyn1 = 'n'
      else
        if(itmode) write(5,400)
  400 format(' ',/,' ','Are product properties approximations',/,' ',2x,
     &'with temperature stored on file? (y/n) ',$)
        read(5,20)filyn1
        if(itmode) write(5,500)
  500 format(' ',/,' ','Are input initial and boundary conditions',/,' '
     &,2x,'and geometrical dimensions stored on file? (y/n) ',$)
        read(5,20)filyn2
        if(model.ge.3) then
          if(itmode) write(5,600)
  600 format(' ',/,' ','Are the kinetic properties stored on file? ',
     &'(y/n) ',$)
          read(5,20)filyn3
        endif
      endif
      if(filyn1.eq.'n'.or.filyn1.eq.'N')then
        call proper
        CALL CONSPR
      endif
      if(model.ne.1)then
        if(filyn2.eq.'n'.or.filyn2.eq.'N')then
          call input1
```

Table B.2 (cont'd).

```
        endif
        if(model.ge.3)then
          if(filyn3.eq.'n'.or.filyn3.eq.'N')then
            call input2
          endif
        endif
        call soln
      endif
      end
```

```
      SUBROUTINE PROPER

c      This subroutine provides the input for the property functions.
c Input values include unfrozen product moisture content, initial
c freezing point or molecular weight of solids, and unfrozen
c product density, thermal conductivity and specific heat.

c   Output includes a printout of unfrozen water (percent), density
c thermal conductivity and specific heat as functions of temper-
c ature.

c   The variables used in this subroutine are:

c     Constants-
c          C1 = 18.015 kg/kmole (molecular weight of water)
c          C2 = 1./273.15 1/K
c          C3 = 6003./8.314 K (latent heat of ice/R)
c          C6 = 273.15 K
c     Input Variables-
c         Wf0 = Unfrozen product moisture content
c          T0 = Initial freezing temperature (C)
c          Ms = Molecular weight of solids
c          Dp = Product density (kg/m^3)              } >0C
c          Kp = Product thermal conductivity (W/mK)   } >0C
c          Cp = Product specific heat (kJ/kgK)        } >0C

c     Misc. Variables-
c          Xx = Intermittent value in determining Ms or T0.
c          Tc = Temperature for printout.
c          Yn = Character- Y or N


      integer type,prpscr,prpfil

      double precision c1,c2,c3,c5,c6,wf0,ms,dp,kp,cp,t0,ds,
     &di,dw,kw,cpi,cpw,moist,dens,conduc,ki,spheat,tc,xx,t0inv,
     &tl,th,eta(20),w(20),tavg,tdif,x(20),avgd,avgk,avgc,ynavg

      character yn*1,prpfl1*10,prpfl2*12,title*20,ttlfil*4
```

Table B.2 (cont'd).

```
      logical itmode

      external moist,dens,conduc,spheat

      common/prop/wf0,ms,dp,kp,cp,t0,/cons/c1,c2,c3,/d/ds,/densi/di,dw,
     &/COND/KW,/SPH/CPI,CPW,/ITM/ITMODE,/ttl/title,ttlfil,/profil/prpfil
     &/mod/model,/pavg/th,tl,avgd,avgk,avgc,ynavg

      save

      c1=18.015d0
      c2=1.0d0/273.15d0
      c3=6003.0d0/8.314d0
      c6=273.15d0
      IF(.NOT.ITMODE)THEN
          READ(10,*)WF0,T0,DP,KP,CP,TYPE
          IF(TYPE.NE.1)MS=T0
          GO TO 20
      ELSE
        write(5,2000)
2000    format('1',72('-'),/,'0',t27,'Product Properties',/,'0',72('-'))
   5    write(5,100)
 100    format(' ',/,' ','Enter initial moisture content(%): ',$)
        READ*,WF0
        if(wf0.LT.1..OR.WF0.GT.100.)THEN
          print*,'Try again!!'
          goto 5
        ENDIF
  10    WRITE(5,300)
 300 FORMAT('0','Choose:',/,' ','   1. Initial freezing temperature',/,
     &' ','   2. Molecular weight of solids')
        READ(5,*)type
        if(type.NE.1)THEN
          WRITE(5,400)
 400      FORMAT(' ',/,' ','Molecular weight of solids: ',$)
          READ(5,*)MS
        ELSE
          WRITE(5,500)
 500      FORMAT(' ',/,' ','Initial freezing temperature (C): ',$)
          READ(5,*)T0
        ENDIF
        write(5,600)
 600    format(' ',/,' ','Enter unfrozen product property values:',/,
     &' ',2x,'density (kg/m^3): ',$)
        READ(5,*)DP
        WRITE(5,700)
 700    FORMAT(' ',2x,'thermal conductivity (W/mK): ',$)
        READ(5,*)KP
        WRITE(5,800)
 800    FORMAT(' ',2x,'specific heat (kJ/kgK): ',$)
        READ(5,*)CP

c  Enter temperature range for determination of mean property values.

        write(5,850)
 850    format(' ',/,' ','Enter temperature range for mean property ',
```

Table B.2 (cont'd).

```
     &  'values:'/,2x,'low temperature (C): ')
        read(5,*)tl
        write(5,860)
  860   format(' ',2x,'high temperature (C): ')
        read(5,*)th
        tl = tl + 273.150d0
        th = th + 273.150d0
        write(5,880)
  880   format(' ',/,' ',2x,'Use average temperatures in FD solution ?
',
     &    '(0=no; 1=y) ')
        read(5,*)ynavg
        WRITE(5,900)
  900   FORMAT(' ',/,' ','Are these values correct? (y/n) ',$)
        read(5,200)yn
  200   FORMAT(A)
        if(yn.ne.'y'.and.YN.NE.'Y')goto 5
      ENDIF
   20 wf0=wf0/100.0d0
      if(type.NE.1)THEN

C  Initial freezing point

        t0=1.0d0/(c2-log(wf0/c1/(wf0/c1+(1.0d0-wf0)/ms))*1.0d0/c3)
      ELSE
        t0=t0+273.15d0

C  Molecular weight of solids

        t0inv = 1.0d0/t0
        xx=exp(c3*(c2-t0inv))
        Ms=(1.0d0-wf0)*xx*c1/(wf0*(1.0d0-xx))
      ENDIF

c  Determine mean property value over specified range using Gauss
c    Quadrature integration: 20 pt. quad.

c  Eta values:

      eta(1) = -0.99312859918509
      eta(2) = -0.96397192727791
      eta(3) = -0.91223442825133
      eta(4) = -0.83911697182222
      eta(5) = -0.74633190646015
      eta(6) = -0.63605368072652
      eta(7) = -0.51086700195083
      eta(8) = -0.37370608871542
      eta(9) = -0.22778585114165
      eta(10) = -0.07652652113350
      do i = 1,10
        eta(21-i) = -eta(i)
      enddo

c  Weighting factors:

      w(1) = 0.01761400713915
```

Table B.2 (cont'd).

```
      w(2) = 0.04060142980039
      w(3) = 0.06267204833411
      w(4) = 0.08327674157670
      w(5) = 0.10193011981724
      w(6) = 0.11819453196152
      w(7) = 0.13168863844918
      w(8) = 0.14209610931838
      w(9) = 0.14917298647260
      w(10) = 0.15275338713073
      do i = 1,10
         w(21-i) = w(i)
      enddo

c  Transform eta onto (th - tl) interval

      tavg = (th+tl)/2.0d0
      tdif = (th-tl)/2.0d0
      avgd = 0
      avgk = 0
      avgc = 0
      do i = 1,20
        x(i) = tavg + tdif*eta(i)

c  Sum integral approximation

        avgd = avgd + w(i)*dens(x(i))
        avgk = avgk + w(i)*conduc(x(i))
        avgc = avgc + w(i)*spheat(x(i))
      enddo
      avgd = 0.50*avgd
      avgk = 0.50*avgk
      avgc = 0.50*avgc

      IF(ITMODE)THEN
        prpscr = 0
        prpfil = 0
        write(5,905)
  905 format(' ',/,' ','Display product properties on screen? (y/n) ',$)
        read(5,200)yn
        if(yn.eq.'y'.OR.YN.EQ.'Y')prpscr = 1
        write(5,906)
  906   format(' ','Do you want product properties - f(T)',
     &  ' saved in a file? (y/n) ',$)
        read(5,200)yn
        if(yn.eq.'y'.OR.YN.EQ.'Y')prpfil = 1
        IF(PRPSCR.EQ.0.AND.PRPFIL.EQ.0)GO TO 90
      ELSE
        READ(5,*)PRPFIL
      ENDIF
      IF(PRPFIL.EQ.1)THEN
        WRITE(PRPFL2,910)TTLFIL,'PRP.VAR'
  910   FORMAT(' ',A,A)
        OPEN(UNIT=12,NAME=PRPFL2(1:12),TYPE='NEW',CARRIAGECONTROL='LIST'
     &)
        write(12,1000)title
        write(12,1050)
```

Table B.2 (cont'd).

```
        write(12,1100)t0-273.150d0
        write(12,1200)ms
        write(12,1300)
        write(12,1400)
        WRITE(12,1500)
        TC=AINT(t0+2.0d0)+.150d0
40      IF(TC.GE.t0)THEN
          WRITE(12,1600)tc-c6,wf0*100.,DP,kp,cp
        ELSE
          WRITE(12,1600)tc-c6,MOIST(TC)*100.,DENS(TC),conduc(tc),
     &    SPHEAT(TC)
        ENDIF
        IF(TC.GE.t0+1.0d0)THEN
          TC = TC-1.0d0
          GO TO 40
        ENDIF
        IF(TC.GT.AINT(t0-1.0d0)+.15)THEN
          TC = TC-.250d0
          GOTO 40
        ENDIF
        IF(TC.GT.AINT(t0-4.0d0)+.15)THEN
          TC = TC-.50d0
          GOTO 40
          ENDIF
        IF(TC.GT.AINT(t0-10.0d0)+.15)THEN
          TC=TC-1.0d0
          GOTO 40
        ELSE
          TC=TC-2.0d0
          IF(TC.GE.233.150d0)GOTO 40
        ENDIF
        write(12,1700)tl-273.15,th-273.15,avgd,avgk,avgc
        CLOSE(UNIT=12)
      ENDIF

C  Printout on screen

      IF(ITMODE)THEN
        IF(PRPSCR.EQ.1)THEN
          write(5,1001)title
          write(5,1050)
          write(5,1100)t0-273.150d0
          write(5,1200)ms
          write(5,1300)
          write(5,1400)
          WRITE(5,1500)
 1000     format(' ',T31,A)
 1001     format('1',T31,A)
 1050     format(' ',t27,'Product Properties',/,' ',t22,'as a ',
     &    'Function of Temperature')
 1100     format(' ',/,' ','Initial freezing temperature(C)= ',f6.2)
 1200     format(' ',/,' ','Equivalent molecular weight= ',f6.2)
 1300     format(' ',/,' ',28x,'Product properties'//)
 1400     FORMAT(' ',/,' ',3X,'Temperature',5X,'Unfrozen',7X,'Density',
     &    4X,'Conductivity',2X,' Ap.Sp.Heat')
 1500     FORMAT(' ',7X,'(C)',10X,'Water',8x,'(kg/m^3)',7X,'(W/m*K)',
```

Table B.2 (cont'd).

```
     &    6X,'(kJ/kg*K)',/,3X,5(' ------------ '))
 1600     FORMAT(' ',T4,F7.2,T22,F6.2,T36,F6.1,T49,F6.3,T62,F8.3)
 1700     format(' ',//,' ','Average property values over the interval '
     &    ,f6.1,' to ',f6.1' :',/,' ',5x,'Average density = ',t40,f6.1,
     &    /,' ',5x,'Average thermal conductivity = ',t40,f5.3,/,' ',5x,
     &    'Average specific heat = ',t40,f5.3)
          TC=AINT(t0+2.0d0)+.150d0
   50     IF(TC.GE.t0)THEN
            WRITE(5,1600)tc-c6,wf0*100.,DP,kp,cp
          ELSE
            WRITE(5,1600)tc-c6,MOIST(TC)*100.,DENS(TC),conduc(tc),
     &      spheat(tc)
          ENDIF
          IF(TC.GE.t0+1.0d0)THEN
            TC = TC-1.0d0
            GO TO 50
          ENDIF
          IF(TC.GT.AINT(t0-1.0d0)+.15)THEN
            TC = TC-.250d0
            GOTO 50
          ENDIF
          IF(TC.GT.AINT(t0-4.0d0)+.15)THEN
            TC = TC-.50d0
            GOTO 50
          ENDIF
          IF(TC.GT.AINT(t0-10.0d0)+.15)THEN
            TC=TC-1.0d0
            GOTO 50
          ELSE
            TC=TC-2.0d0
            IF(TC.GE.233.150d0)GOTO 50
          ENDIF
        ENDIF
      ENDIF
      write(5,1700)tl-273.15,th-273.15,avgd,avgk,avgc
90    return
      END
```

```
      DOUBLE PRECISION FUNCTION MOIST(X)

c   This function determines the unfrozen water
c fraction of a food procduct below the initial freezing temp-
c perature.

c   the variables used in this function are:

c          c1= 18.015 kg/kmole; molecular weight of water
c          c2= 1./273.15 K^-1
c          c3= 6003./8.314 K; latent heat of ice/mole/R
c          wf0= moisture content of unfrozen product
c          ms= molecular weight of solids
```

Table B.2 (cont'd).

```
      double precision c1,c2,c3,xx,ms,tinv,t0,wf0,dp,kp,cp

      common/prop/wf0,ms,dp,kp,cp,t0,/cons/c1,c2,c3

      c1=18.015d0
      c2=1.0d0/273.15d0
      c3=6003.0d0/8.314d0
      tinv = 1.0d0/x
      xx=exp(c3*(c2-tinv))
      moist=xx*(1.0d0-wf0)*c1/(ms*(1.0d0-xx))
      return
      end

      DOUBLE PRECISION FUNCTION DENS(X)

c   this function determines the density of a food product
c below the initial freezing temperature,as a fuction of un-
c frozen water fraction.

c   the variables used in this function are:

c          di= density of ice (kg/m^3)
c          dp= density of unfrozen product (kg/m^3)
c          ds= density of solids (kg/m^3)
c          dw= density of water (kg/m^3)
c          wf0= moisture content of unfrozen product

      double precision moist,wf0,ms,dp,kp,cp,t0,di,dw,ds

      external moist

      common/prop/wf0,ms,dp,kp,cp,t0,/densi/di,dw,/d/ds

      save

c  Solids density
      ds=(1.0d0-wf0)/(1.0d0/dp-wf0/dw)

      dens=1.0d0/(moist(x)/dw+(1.0d0-wf0)/ds+(wf0-moist(x))/di)

      return
      end



      DOUBLE PRECISION FUNCTION KI(X)

c Thermal conductivity of ice as a function of temperature (k)

      ki=7.3640d0-0.02850d0*x+3.525d-5*x**2

      return
      end
```

Table B.2 (cont'd).

```
      DOUBLE PRECISION FUNCTION CONDUC(X)

c   This function subroutine determines the thermal conductivity
c of a food product below the initial freezing temperature.
c Thermal conductivity is a function of moisture content and
c solids content, therefore, 'conduc' is a function of temper-
c ature.

c   The variables used in this function subroutine are:

c          di= density of ice
c          dw= density of water
c       ki(x)= thermal conductivity of ice as a function of
c              temperature
c          kp= thermal conductivity of product
c          ks= thermal conductivity of solids
c          kw= thermal conductivity of water
c         wf0= unfrozen product moisture content
c    moist(x)= moisture content as a function of temperature
c          k1= intermittent value
c          k2= intermittent value
c          k3= thermal conductivity of water-ice phase
c          k4= intermittent value
c          k5= intermittent value
c          va= intermittent value
c         val= intermittent value
c          c4= 2./3.


      double precision moist,dens,ki,wf0,ms,dp,kp,cp,t0,di,dw,ds,
     &kw,c4,va,val,ks,k1,k2,k3,k4,k5

      common /prop/wf0,ms,dp,kp,cp,t0,/densi/di,dw,/d/ds,/cond/kw

      external moist,dens,ki

      c4=2.0d0/3.0d0
c  Solids density
      ds=(1.0d0-wf0)/(1.0d0/dp-wf0/dw)


c Thermal conductivity of solids

      va=(1.0d0-wf0)/ds
      val=(va/(va+wf0/dw))**c4
      ks=kw*(val-((kw-kp)/kw-kp*(1.0d0-val**.5)/val))

c Phase I: ice--water

      k1=(wf0-moist(x))/di/(moist(x)/dw+(wf0-moist(x))/di)
      k2=k1**c4*(1.0d0-ki(x)/kw)
      k3=kw*(1.0d0-k2)/(1.0d0-k2*(1.0d0-k1**(1.0d0/3.0d0)))

c Phase II:  solids--water/ice

      k4=va/(va+(wf0-moist(x))/di+moist(x)/dw)
```

Table B.2 (cont'd).

```
      k5=k4**c4*(1.0d0-ks/k3)
      conduc=k3*(1.0d0-k5)/(1.0d0-k5*(1.0d0-k4**(1.0d0/3.0d0)))
      return
      end



      DOUBLE PRECISION FUNCTION SPHEAT(X)

c   This function determines the apparent specific
c heat of a frozen food product as a function of unfrozen water
c below the initial freezing temperature.

c   The external function moist (unfrozen water fraction) used
c to determine the apparant specific heat is a function of
c temperature; therefore, 'spheat' is also a function of temper-
c ature.

c   the variables used in this function are:

c          cp = specific heat product >0C (kJ/kgC)
c          cpi= specific heat of ice (kJ/kgC)
c          cps= specific heat of solids (kJ/kgC)
c          cpw= specific heat of water (kJ/kgC)
c          wf0 = moisture content of unfrozen product

      double precision moist,wf0,ms,dp,kp,cp,t0,cpi,cpw,dh,cps,dcp,c5

      external moist

      common/prop/wf0,ms,dp,kp,cp,t0,/sph/cpi,cpw

      c5=6003.0d0/18.015d0
      dh = 0.0010d0

c solids specific heat

      cps=(1.0d0-wf0)/(1.0d0/cp-wf0/cpw)

      if(x+dh.ge.t0)then
        dcp = (moist(x)-moist(x-dh))*c5/dh
      else
        dcp = (moist(x+dh)-moist(x-dh))*c5/(2.0d0*dh)
      endif
      spheat=(1.0d0-wf0)*cps+moist(x)*cpw+(wf0-moist(x))*cpi+dcp

      return
      end



      SUBROUTINE CONSPR
```

Table B.2 (cont'd).

```
c       This subroutine provides constant property approximations to the
c     properties as a function of temperature.

      parameter(maxd = 101, maxc = 51, maxs = 201)

      integer prpfil,nsd,nsc,nss,ynavg

      double precision tc,c5,aa,a,b,t0m1,apb,
     &erdens,ercond,erpsh,
     &avgdi,avgki,avgci,
     &bstep,erldd,erldc,erlds,er2dd,er2dc,er2ds
     &ddera,dderab,rho2d,rhold,cdera,cderab,k2d,kld,
     &sdera,sderab,sph2d,sphld

c  Declare variables in common blocks

      double precision wf0,ms,dp,kp,cp,t0,
     &c1,c2,c3,
     &denst(maxd),densc(maxd),condt(maxc),condc(maxc),
     &spht(maxs),sphc(maxs),
     &tl,th,avgd,avgk,avgc,
     &MOIST,DENS,CONDUC,SPHEAT

      character title*40,ttlfil*10,prpfl3*12,fildat*16

      EXTERNAL MOIST,DENS,CONDUC,SPHEAT

      common/prop/wf0,ms,dp,kp,cp,t0,
     &/cons/c1,c2,c3,
     &/CONSTP/DENST,DENSC,CONDT,CONDC,SPHT,SPHC,
     &/NCONSTP/NSD,NSC,NSS,
     &/ttl/title,ttlfil,
     &/profil/prpfil,
     &/pavg/th,tl,avgd,avgk,avgc,ynavg

      AA = 233.150d0
      NTSD = 100
      NTSC = 50
      NTSS = 200
      ER1DD = .050d0
      ER1DC = .050d0
      ER1DS = .050d0
      ER2DD = .250d0
      ER2DC = .250d0
      ER2DS = .250d0
      BSTEP = 0.10d0
      T0M1 = T0-0.050d0

C     Find values for density

  5   I = 1
      A = AA
      DENST(1) = A
      B = BSTEP
      IP1 = I+1
      APB = A+B
```

Table B.2 (cont'd).

```
C   Determine % second derivative for density.

 10   DDERA = (DENS(A+0.010d0)-DENS(A))/DENS(A)
      DDERAB = (DENS(APB+0.010d0)-DENS(APB))/DENS(APB)
      RHO2D = (DDERAB-DDERA)/DDERA

C   Determine % first derivative for density.

      RHO1D = (DENS(A)-DENS(APB))/(DENS(A))

C   Check if second derivative is greater than allowable error; if so,
store
C temperature endpoint value and determine the average property value
for that
C section, assuming property is linear with temperature over each
segment.

      IF(ABS(RHO2D).GT.ER2DD.OR.ABS(RHO1D).GT.ER1DD)THEN
        DENST(IP1) = APB
        call integr(apb,a,avgdi,avgki,avgci,1)

        densc(i) = avgdi
        A = A+B
        I = IP1
        IP1 = I+1

C     Check if number of steps is greater than array dimensions; if so,
C  double the allowable error in the second derivative and repeat calcu-
C  lations.

        IF(I.GT.NTSD)THEN
          ER2DD = ER2DD*2.0d0
          ER1DD = ER1DD*2.0d0
          GO TO 5
        ENDIF
        B = BSTEP
      ELSE
        B = B+BSTEP
      ENDIF
      APB = A+B
      IF(APB.LT.T0-bstep)GO TO 10
      DENST(IP1) = T0
      call integr(t0ml,a,avgdi,avgki,avgci,1)
      densc(i) = avgdi
      NSD = I

C     Find values for thermal conductivity

   20 I = 1
      A = AA
      CONDT(1) = A
      B = BSTEP
      IP1 = I+1
      APB = A+B

C   Determine % second derivative for thermal conductivity.
```

Table B.2 (cont'd).

```
   30 CDERA = (CONDUC(A+0.010d0)-CONDUC(A))/CONDUC(A)
      CDERAB = (CONDUC(APB+0.010d0)-CONDUC(APB))/CONDUC(APB)
      K2D = (CDERAB-CDERA)/CDERA

C   Determine % first derivative for thermal conductivity.

      K1D = (CONDUC(A)-CONDUC(APB))/(CONDUC(A))

C   Check if second derivative is greater than allowable error; if so,
store
C temperature endpoint value and determine the average property value
for that
C section, assuming property is linear with temperature over each
segment.

      IF(ABS(K2D).GT.ER2DC.OR.ABS(K1D).GT.ER1DC)THEN
        CONDT(IP1) = APB
        call integr(apb,a,avgdi,avgki,avgci,2)
        condc(i) = avgki
        A = A+B
        I = IP1
        IP1 = I+1

C     Check if number of steps is greater than array dimensions; if so,
C  double the allowable error in the second derivative and repeat calcu-
C  lations.

        IF(I.GT.NTSC)THEN
          ER2DC = ER2DC*2.0d0
          ER1DC = ER1DC*2.0d0
          GO TO 20
        ENDIF
        B = BSTEP
      ELSE
        B = B+BSTEP
      ENDIF
      APB = A+B
      IF(APB.LT.T0-bstep)GO TO 30
      CONDT(IP1) = T0
      call integr(t0ml,a,avgdi,avgki,avgci,2)
      condc(i) = avgki
      NSC = I

C   Find values for specific heat

   40 I = 1
      A = AA
      SPHT(1) = A
      B = BSTEP
      IP1 = I+1
      APB = A+B

C   Determine % second derivative for specific heat.

   50 SDERA = (SPHEAT(A+0.010d0)-SPHEAT(A))/SPHEAT(A)
      SDERAB = (SPHEAT(APB+0.010d0)-SPHEAT(APB))/SPHEAT(APB)
```

Table B.2 (cont'd).

```
      SPH2D = (SDERAB-SDERA)/SDERA

C   Determine % first derivative for specific heat.

      SPH1D = (SPHEAT(A)-SPHEAT(APB))/(SPHEAT(A))

C   Check if second derivative is greater than allowable error; if so,
store
C temperature endpoint value and determine the average property value
for that
C section, assuming property is linear with temperature over each
segment.

      IF(ABS(SPH2D).GT.ER2DS.OR.ABS(SPH1D).GT.ER1DS)THEN
        SPHT(IP1) = APB
        call integr(apb,a,avgdi,avgki,avgci,3)

        sphc(i) = avgci
        A = A+B
        I = IP1
        IP1 = I+1

C     Check if number of steps is greater than array dimensions; if so,
C  double the allowable error in the second derivative and repeat calcu-
C  lations.

        IF(I.GT.NTSS)THEN
          ER2DS = ER2DS*2.0d0
          ER1DS = ER1DS*2.0d0
          GO TO 40
        ENDIF
        B = BSTEP
      ELSE
        B = B+BSTEP
      ENDIF
      APB = A+B
      IF(APB.LT.T0-bstep)GO TO 50
      SPHT(IP1) = T0
      call integr(t0m1,a,avgdi,avgki,avgci,3)
      sphc(i) = avgci
      NSS = I

C   Print out approximated property values in file.

      IF(PRPFIL.EQ.1)THEN
      WRITE(PRPFL3,85)TTLFIL,'PRP.CON'
   85 FORMAT(' ',A,A)
      OPEN(UNIT=12,NAME=PRPFL3(1:12),TYPE='NEW',CARRIAGECONTROL='LIST')
      WRITE(12,88)TITLE
   88 FORMAT(' ',//,' ',T31,A)
      WRITE(12,90)
   90 FORMAT(' ',/,' ',T25,'CONSTANT PROPERTIES')
      WRITE(12,100)T0-273.150d0
  100 FORMAT(' ',/,' ','Initial freezing temperature(C)= ',F6.2)
      WRITE(12,110)MS
  110 FORMAT(' ',/,' ','Equivalent molecular weight= ',F6.2)
```

Table B.2 (cont'd).

```
      WRITE(12,120)
  120 FORMAT(' ',/,' ',//,28X,'Product Properties'//)
      WRITE(12,130)
  130 FORMAT(4X,'Temperature',5X,'Unfrozen',7X,'Density',4X,
     +'Conductivity',2X,' Ap.Sp.Heat')
      WRITE(12,140)
  140 FORMAT(' ',7X,'(C)',10X,'Water',8X,'(kg/m^3)',7X,'(W/m*K)',
     +6X,'(kJ/kg*C)',/,3X,5(' ------------ '))

      TC=AINT(T0+2.0d0)+.150d0
  150 IF(TC.LT.T0)GOTO 170
      WRITE(12,160)TC-273.150d0,WF0*100.0d0,DP,KP,CP
  160 FORMAT(5X,F7.2,T22,F6.2,T36,F6.1,T49,F6.3,T62,F8.3)
      GOTO 240
  170 DO I = 1,NSD
        IF(TC.GE.DENST(I).AND.TC.LT.DENST(I+1))THEN
          DE = DENSC(I)
          GOTO 180
        ENDIF
      ENDDO
  180 DO I = 1,NSC
        IF(TC.GE.CONDT(I).AND.TC.LT.CONDT(I+1))THEN
          CO = CONDC(I)
          GOTO 190
        ENDIF
      ENDDO
  190 DO I = 1,NSS
        IF(TC.GE.SPHT(I).AND.TC.LT.SPHT(I+1))THEN
          SP = SPHC(I)
          GOTO 200
        ENDIF
      ENDDO
  200 WRITE(12,160)TC-273.150d0,MOIST(TC)*100.0d0,DE,CO,SP
  240 IF(TC.GE.T0+1.0d0)GOTO 270
      IF(TC.LE.AINT(T0-1.0d0)+.15)GOTO 250
      TC=TC-.250d0
      GOTO 290
  250 IF(TC.LE.AINT(T0-4.0d0)+.15)GOTO 260
      TC=TC-.50d0
      GOTO 290
  260 IF(TC.LE.T0-10.0d0)GOTO 280
  270 TC=TC-1.0d0
      GOTO 290
  280 TC=TC-2.0d0
  290 IF(TC.GE.233.150d0)GOTO 150
      WRITE(12,300)NSD,NSC,NSS
  300 FORMAT(' ',/,' ','No. const. density values = ',I3,/,
     &X,'No. const. conductivity values = ',I3,/,
     &X,'No. const. specific heat values = ',I3)
      WRITE(12,305)ER1DD,ER1DC,ER1DS,ER2DD,ER2DC,ER2DS
  305 FORMAT(' ',/,' ','Error 1D dens. = ',F6.4,' Error 1D cond. = ',
     &F6.4,/' ','Error 1D sp. heat = ',F6.4,' Error 2D density = ',F6.4,
     &/,X,'Error 2D cond. = ',F6.4,' Error 2D sp. heat = ',F6.4)
        WRITE(12,306)BSTEP
  306 FORMAT(' ','Step increment = ',f8.5)
      CLOSE(UNIT=12)
```

Table B.2 (cont'd).

```
      ENDIF

C  Store results in data file


C  Results for density, thermal conductivity and specific heat
C     approximations are stored in 'TTLFILprp.dat'

      WRITE(FILDAT,310)ttlfil,'PRP.DAT'
  310 FORMAT(' ',a,A)
      OPEN(UNIT=12,NAME=FILDAT(1:16),TYPE='NEW',CARRIAGECONTROL='LIST')
      WRITE(12,*)WF0,T0,MS
      WRITE(12,*)DP,KP,CP
      WRITE(12,*)NSD,NSC,NSS
      write(12,*)tl,th,avgd,avgk,avgc,ynavg
      DO I=1,NSD
        WRITE(12,*)DENST(I),DENSC(I)
      ENDDO
      DO I=1,NSC
        WRITE(12,*)CONDT(I),CONDC(I)
      ENDDO
      DO I=1,NSS
        WRITE(12,*)SPHT(I),SPHC(I)
      ENDDO
      CLOSE(UNIT=12)
      RETURN
      END




      SUBROUTINE INTEGR(thi,tlow,avgdp,avgkp,avgcp,ncase)


      integer np,ncase

      double precision eta(25),w(25),thi,tlow,tavg,tdiff,avgdp,
     &avgkp,avgcp,x(25),
     &dens,conduc,spheat

      external dens,conduc,spheat

c  Ncase = 0: 20pt. quad for density, conductivity, sp.heat over tHi -
tLo
c  Ncase = 1: 5pt. quad for density
c  Ncase = 2: 5pt. quad for thermal conductivity
c  Ncase = 3: 5pt. quad for specific heat


c  Determine mean property value over specified range using Gauss
c    Quadrature integration:

      if(ncase.eq.0)then

c  20 pt. quad.
```

Table B.2 (cont'd).

```
        np = 20

c  Eta values:

        eta(1) = -0.99312859918509
        eta(2) = -0.96397192727791
        eta(3) = -0.91223442825133
        eta(4) = -0.83911697182222
        eta(5) = -0.74633190646015
        eta(6) = -0.63605368072652
        eta(7) = -0.51086700195083
        eta(8) = -0.37370608871542
        eta(9) = -0.22778585114165
        eta(10) = -0.07652652113350
        do i = 1,10
          eta(21-i) = -eta(i)
        enddo

c  Weighting factors:

        w(1) = 0.01761400713915
        w(2) = 0.04060142980039
        w(3) = 0.06267204833411
        w(4) = 0.08327674157670
        w(5) = 0.10193011981724
        w(6) = 0.11819453196152
        w(7) = 0.13168863844918
        w(8) = 0.14209610931838
        w(9) = 0.14917298647260
        w(10) = 0.15275338713073
        do i = 1,10
           w(21-i) = w(i)
        enddo

      else

c  5 pt. quad.

        np = 5

c  Eta values:

        eta(1) = -0.90617984593866
        eta(2) = -0.53846931010568
        eta(3) = 0.0
        eta(4) = -eta(2)
        eta(5) = -eta(1)

c  Weighting factors:

        w(1) = 0.23692688505619
        w(2) = 0.47862867049937
        w(3) = 0.56888888889
        w(4) = w(2)
        w(5) = w(1)
```

Table B.2 (cont'd).

```
      endif
c  Transform eta onto (th - tl) interval

      tavg = (thi+tlow)/2.0d0
      tdif = (thi-tlow)/2.0d0
      avgdp = 0
      avgkp = 0
      avgcp = 0
      do i = 1,np
        x(i) = tavg + tdif*eta(i)

c  Sum integral approximation

        if(ncase.eq.0.or.ncase.eq.1)avgdp = avgdp + w(i)*dens(x(i))
        if(ncase.eq.0.or.ncase.eq.2)avgkp = avgkp + w(i)*conduc(x(i))
        if(ncase.eq.0.or.ncase.eq.3)avgcp = avgcp + w(i)*spheat(x(i))
      enddo
      if(ncase.eq.0.or.ncase.eq.1)avgdp = 0.50*avgdp
      if(ncase.eq.0.or.ncase.eq.2)avgkp = 0.50*avgkp
      if(ncase.eq.0.or.ncase.eq.3)avgcp = 0.50*avgcp

      return
      end




      BLOCK DATA CONST

c   the following values are defined in this block data:

c          di= density of ice (917. kg/m^3)   }  block
c          dw= density of water (998. kg/m^3) } /densi/

c          kw= thermal conductivity of water  }  block
c              (0.569 w/mk)                   } /cond/

c         cpi= specific heat of ice           }
c              (2.1 kj/kgk)                   }  block
c         cpw= specific heat of water         }  /sph/
c              (4.187 kj/kgk)                 }


      double precision DI,DW,KW,CPI,CPW
      COMMON /DENSI/DI,DW,/COND/KW,/SPH/CPI,CPW
      SAVE /DENSI/,/COND/,/SPH/
      DATA DI,DW/917.0d0,998.0d0/,KW/0.5690d0/,
     &CPI,CPW/2.10d0,4.1870d0/
      END
```

Table B.2 (cont'd).

```
      SUBROUTINE INPUT1

c   This subroutine provides the input for the boundary condi-
c tions on the product for the case where the ambient temper-
c ature and surface heat tranfer coeffient are known and assumed
c to be constant over a given storage period.

c   Input varibles include, initial product temperature, sym-
c metry of boundary conditions, number of constant temperature
c storage periods, length of storage period, and surface heat
c transfer coefficient.


      parameter(maxp=20)

      integer ct(maxp),cct(maxp),per,sym,sstep,unit,shape,m,
     &htype(maxp)

      double precision ti,temp(maxp),stor(maxp),h1(maxp),h2(maxp),
     &tunit(maxp),h,l,dz,per1,per2,amp1,amp2

      character yn*1,title*20,ttlfil*4,fildat*12,inpdat*12

      logical itmode

      common/bound/per,ti,temp,stor,h1,h2,tunit,
     &/geom/shape,h,l,dz,sym,m,mpl,sstep/ttl/title,ttlfil,/mod/model,
     &/itm/itmode,/datfil/fildat,inpdat,kindat

      save



      if(itmode)go to 3
      read*,per,ti,sym
      do i = 1,per
        read*,temp(i),unit,stor(i),h1(i),h2(i)
        if(unit.eq.1)then
          tunit(i)=3600.0d0
        else
          tunit(i)=86400.0d0
        endif
      enddo
      read shape
      if(shape.lt.3)then
        read*,l
        h = 1.0d0
      else
        read*,l
        h = 0.0d0
      endif
      go to 500

3     write(5,1)
    1 format('1',72('-'),/,'0',27x,'Storage Conditions',/,'0',72('-'))
5     write(6,10)
```

Table B.2 (cont'd).

```
10    format(' ',/,' ','Enter number of constant temp. storage ',
     &'periods: ',$)
      read*,per
      write(6,20)
20    format(' ',/,' ','Initial product temperature (C): ',$)
      read*,ti
      write(6,30)
30    format(' ',/,' ','Are the boundary conditions symmetrical? ',
     &'(0=No,1=Yes) ',/,' ','(Enter ''1'' for cylinder & sphere ',
     &'geometries) ',$)
      read*,sym
      write(6,40)
40    format(' ',/,' ','Are these values correct? (y/n) ',$)
      read(5,2)yn
    2 format(a)
      if(yn.ne.'y'.and.yn.ne.'Y')goto 5

c input boundary conditions for each storage period

      do 120 i=1,per
45    write(6,50)i
50    format(' ',/,' ','Enter data for period  ',i3,':',$)
      write(6,60)
60    format(' ',5x,'Storage temperature (C): ',$)
      read*,temp(i)
70    write(6,75)
75    format(' ',5x,'Enter units for storage temp.:',/,' ',7x,'1= hours'
     &,/,' ',7x,'2= days')
      read*,unit
      if(unit.le.0.and.unit.gt.2)then
        print*,'try again!'
        goto 70
      endif
        if(unit.eq.1)then
          tunit(i)=3600.0d0
        else
          tunit(i)=86400.0d0
        endif
77    write(6,80)
80    format(' ',5x,'Length of storage period: ',$)
      read*,stor(i)
      if(stor(i).lt.0)then
        print*,'try again!'
        goto 77
      endif
      write(6,82)
 82   format(' ',5x,'Enter curve type of heat transfer coefficient:',
     &/,' ',7x,'1.  constant',/,' ',7x,'2.  sinusiodal')
      read*,htype(i)
      if(htype(i).eq.1)then
      write(6,90)
90    format(' ',5x,'Enter surface heat transfer coefficient (W/m^2C):',
     &/,' ',7x,'side 1 = ',$)
      read*,h1(i)
      if(sym.ne.1)then
        write(6,100)
```

Table B.2 (cont'd).

```
100     format(' ',7x,'side 2 = ',$)
        read*,h2(i)
      else
        h2(i)=0.0d0
      endif
      else
      write(6,102)
102   format(' ',5x,'Enter amplitude (C) and period (hrs) of
sinusiodal',
     &' curve for h (side 1) : ',$)
      read*,ampl,perl
      if(sym.ne.1)then
        write(6,104)
104     format(' ',7x,'side 2 = ',$)
        read*,amp2,per2
      else
        amp2=0.0d0
        per2=0.0d0
        h2(i) = 0.0d0
      endif
      endif
110   write(5,115)
  115 format(' ',/,' ','Are these values correct? (y/n) ',$)
      read(5,2)yn
      if(yn.ne.'y'.and.yn.ne.'Y')goto 45
120   continue

c input geometry and size

140   write(6,150)
150   format('0','Enter product geometry: ',/,' ',5x,'1 = slab',/,' ',5x
     +,'2 = cylinder',/,' ',5x,'3 = sphere')
      read*,shape
      if(shape.gt.1.and.sym.eq.0)then
      print*,'Boundary conditions must be symmetrical for cylinder and',
     &' sphere; try again!!'
      go to 3
      endif
      if(shape.eq.1)then
        write(6,160)
160   format(' ',/,' ','Enter dimensions for slab:',/,' ',5x,
     &'thickness in direction of heat transfer (m) = ',$)
        read*,l
        h = 1.0d0
      else
        if(shape.eq.2)then
          write(6,180)
180   format(' ',/,' ','Enter dimensions for cylinder',/,
     +' ',5x,'radius (m)= ',$)
          read *,l
          h = 1.0d0
        else
          write(6,200)
200   format(' ',/,' ','Enter dimensions for sphere (m)',/,
     +' ',5x,'radius (m)= ',$)
          read *,l
```

Table B.2 (cont'd).

```
          h=0.0d0
        endif
      endif
500   ti = ti+273.150d0
      if(sym.eq.1.and.shape.eq.1)L = L*0.50d0
      do i = 1,per
        temp(i) = temp(i)+273.150d0
        stor(i)=stor(i)*tunit(i)
        if(sym.eq.1.or.shape.ne.1)then
          if(shape.eq.1)then
            h2(i) = 0.0d0
          else
            h2(i) = h1(i)
            h1(i) = 0.0d0
          endif
        endif
      enddo
      if(l.ge.0.80d0)then
        m=40
        sstep=10
        go to 590
      endif
      do i=1,10
        if(l.lt.i*0.080d0)then
          m=i*4
          sstep=i
          go to 590
        endif
      enddo
  590 write(inpdat,600)ttlfil,'inp.dat'
  600 format(' ',a,a)
      open(unit=12,name=inpdat(1:12),type='new',carriagecontrol='list')
      write(12,700)per,sym,ti
  700 format(' ',i2,2x,i1,2x,f6.2)
      do i = 1,per
        write(12,710)htype(i)
  710   format(' ',i1)
        if(htype(i).eq.1)then
        write(12,800)temp(i),stor(i),tunit(i),h1(i),h2(i)
  800 format(' ',2x,f6.2,2x,f18.2,2x,f6.0,2(2x,f8.2))
        else
          write(12,810)temp(i),stor(i),tunit(i),amp1,per1,amp2,per2
  810 format(' ',2x,f6.2,2x,f18.2,2x,f6.0,4(2x,f8.2))
        endif
      enddo
      write(12,900)shape,L,h
  900 format(' ',i1,2(2x,f8.4))
      write(12,1000)m,sstep
 1000 format(' ',i2,2x,i2)
      close(unit=12)
      return
      end
```

Table B.2 (cont'd).

```
      SUBROUTINE INPUT2

c     Kinetic properties to determine quality loss in a food product
c  are entered in this subroutine.  A file titled 'TTLFILkin.dat'
c  containing the kinetic properties is created.  This file is re-
c  opened in the solution, and it may be reused again in subsequent
c  runs.

      integer model

      double precision q0,tref,ea,vea,vq0

      character title*20,ttlfil*4,fildat*12,inpdat*12,kindat*12

      logical itmode

      common /ttl/title,ttlfil,/mod/model,/itm/itmode,
     &/datfil/fildat,inpdat,kindat


c  Read batch file data (if itmode = .false.)

      if(itmode)go to 1
      read*,q0,tref,ea
      if(model.eq.4)then
        read*,vea,vq0
      endif
      go to 30

c  Read interactive input

    1  write(5,2)
    2 format('1',72('-'),/,'0',t23,'Reference Shelf-life data',/,'0',
     &72('-'))
   10 write(5,100)
  100 format(' ',/,' ','Enter reference shelf-life (days) : ',$)
      read*,q0
      write(5,200)
  200 format(' ',/,' ','Enter reference temperature for reference',
     &' shelf-life (C) : ',$)
      read*,tref
      write(5,300)
  300 format(' ',/,' ','Enter activation energy constant (kJ/mole) : '
     &,$)
      read*,ea
      if(model.eq.4)then
        write(5,400)
  400 format(' ',/,' ','Enter standard deviation of activation ',
     &'energy const. (kJ/mole) : ',$)
        read*,vea
        write(5,500)
  500 format(' ',/,' ','Enter standard deviation of ref. shelf-life',
     &' (days) : ',$)
        read*,vq0
      endif
```

Table B.2 (cont'd).

```
      write(5,550)
  550 format(' ',/,' ','Are these values correct? (y/n) ',$)
      read(5,20)yn
   20 format(a)
      if(yn.eq.'n'.or.yn.eq.'N')go to 10

   30 tref = tref+273.150d0
      ea = ea*1000.0d0
      if(model.eq.4)then
        vea = (vea*1000.0d0)**2.0d0
        vq0 = vq0*vq0
      endif

      write(kindat,600)ttlfil,'kin.dat'
  600 format(' ',a,a)
      open(unit=12,name=kindat(1:12),type='new',carriagecontrol='list')
      write(12,700)q0,tref,ea
  700 format(' ',2x,f8.1,2x,f7.2,2x,f13.0)
      if(model.eq.4)then
        write(12,800)vea,vq0
  800 format(' ',2(2x,e11.3))
      endif
      close(unit=12)
      return
      end




      SUBROUTINE SOLN


      parameter(maxd = 101, maxc = 51, maxs = 201)
      parameter(maxm=101,maxp=20,tol=0.10d0,r=8.3140d0)

      integer per,shape,m,sym,sstep,htype(maxp)

      double precision wf0,ms,dp,kp,cp,t0,ea,q0,tref,vea,vq0,
     &h,l,dz,ti,temp(maxp),stor(maxp),h1(maxp),h2(maxp),tunit(maxp),
     &jj,kjj,eabs,eeabs,ssum,HH1(2),HH2(2),ta(2),tavg,qavg,
     &qual(maxm),dsum,vsum,eex,avdl,dqdea,ct(maxp),cct(maxp),dq(2),
     &vq,DENST(maxd),DENSC(maxd),CONDT(maxc),CONDC(maxc),
     &SPHT(maxs),SPHC(maxs),cc(maxm),dd(maxm),a(maxm),b(maxm),
     &c(maxm),d(maxm),t(maxm,2),dt,ds,tl,th,avgd,avgk,avgc,ynavg,
     &tdt,ampl,perl,amp2,per2,pi

      character title*20,ttlfil*4,fildat*12,inpdat*12,kindat*12

      logical itmode

      common/bound/per,ti,temp,stor,h1,h2,tunit,/ttl/title,ttlfil
     &,/geom/shape,h,l,dz,sym,m,mpl,sstep,/mod/model,/itm/itmode,
     &/datfil/fildat,inpdat,kindat,/NCONSTP/NSD,NSC,NSS,/dff/dif,
     &/CONSTP/DENST,DENSC,CONDT,CONDC,SPHT,SPHC,/prop/wf0,ms,dp,
```

Table B.2 (cont'd).

```
     &kp,cp,t0,/shelf/ea,q0,vea,vq0,tref,/d/ds
     &/pavg/th,tl,avgd,avgk,avgc,ynavg,/toldt/tdt

      save

c  Read in boundary and initial conditions

      write(inpdat,600)ttlfil,'inp.dat'
  600 format(' ',a,a)
      open(unit=12,name=inpdat(1:12),type='old',carriagecontrol='list')

      read(12,*)per,sym,ti
      do i = 1,per
        read(12,*)htype(i)
        if(htype(i).eq.1)then
        read(12,*)temp(i),stor(i),tunit(i),h1(i),h2(i)
        else
          read(12,*)temp(i),stor(i),tunit(i),amp1,per1,amp2,per2
        endif
      enddo

C  Input geometry and dimensions

      read(12,*)shape,L,H
      read(12,*)m,sstep
      close(unit=12)

c  Read in constant property assumptions

      WRITE(FILDAT,310)TTLFIL,'PRP.DAT'
  310 FORMAT(' ',A,A)
      OPEN(UNIT=12,NAME=FILDAT(1:12),TYPE='OLD',CARRIAGECONTROL='LIST')
      READ(12,*)WF0,T0,MS
      READ(12,*)DP,KP,CP
      READ(12,*)NSD,NSC,NSS
      read(12,*)tl,th,avgd,avgk,avgc,ynavg
      DO I=1,NSD
        READ(12,*)DENST(I),DENSC(I)
      ENDDO
      DO I=1,NSC
        READ(12,*)CONDT(I),CONDC(I)
      ENDDO
      DO I=1,NSS
        READ(12,*)SPHT(I),SPHC(I)
      ENDDO
      CLOSE(UNIT=12)

c  Read in kinetic data

      if(model.ge.3)then
        write(kindat,600)ttlfil,'kin.dat'
      open(unit=12,name=kindat(1:12),type='old',carriagecontrol='list')
        read(12,*)q0,tref,ea
        if(model.eq.4) read(12,*)vea,vq0
        close(unit=12)
      endif
```

Table B.2 (cont'd).

```
      pi = dacos(-1.0d0)
      dt = 120
      dz = L/m
      mp1 = m+1
      DO k = 1,2
        TA(k)=TEMP(1)
        if(htype(1).eq.1)then
        HH1(k)=H1(1)
        HH2(k)=H2(1)
        else
         if(amp1.ne.0)then
           hh1(k) = amp1*cos(2*pi*(k-1)*dt/(per1*3600))
         else
           hh1(k) = 0
         endif
         if(amp2.ne.0)then
           hh2(k) = amp2*cos(2*pi*(k-1)*dt/(per2*3600))
         else
           hh2(k) = 0
         endif
        endif
        DO I = 1,mp1
          t(I,k)=ti
        enddo
      enddo
      tavg = ti
      time=0
      count=0
      JJ=0
      JJJ=0
      DO I = 1,mp1
        IF(MODEL.GE.3)THEN
          QUAL(I)=Q0*86400.0d0
        ELSE
          QUAL(I) = 0.0d0
        ENDIF
      ENDDO
      qavg = q0*86400.0d0
        if(model.eq.4)then
          dq(1) = 0.0d0
          dq(2) = 0.0d0
          vq0=vq0*86400.0d0
          vq=vq0
        endif
      nprint = 0
      IF(MODEL.LT.3)THEN
        HEADTQ=1
      else
        headtq=2
      endif
      call output(nprint,headtq,t,tavg,time,jj,qual,qavg,vq,ii,eend,dt)
      j=1


c finite difference solution
```

Table B.2 (cont'd).

```
      do 160 ii=1,per
      eend=0
      if(ii.ne.1)then
      time=time+stor(ii-1)
      nprint = 2
      call output(nprint,headtq,t,tavg,time,jj,qual,qavg,vq,ii,eend,dt)
      endif
      jj=0
      jjj=0

c _____________________________________________________________

c check if time is > length of storage period

65    if(jj.ge.stor(ii))goto 155

c check if product temp. is close to ambient temp.

      eabs=abs(t(1,2)-temp(II))
      do 60 i=2,mp1
      eeabs=abs(t(i,2)-temp(ii))
      if(eabs.lt.eeabs)then
      eabs=eeabs
      endif
60    continue
      if(eabs.lt.tol)goto 105
      dtmax = stor(ii)-jj

c _____________________________________________________________

      count=count+1

c set ambient temperature = storage temperature

      ta(2)=temp(ii)

c boundary conditions

      If(htype(ii).eq.1)then
        hh1(2)=h1(ii)
        hh2(2)=h2(ii)
      else
        if(amp1.ne.0)then
          hh1(2) = amp1*cos(2*pi*(j-1)*dt/(per1*3600))
        else
          hh1(2) = 0
        endif
        if(amp2.ne.0)then
          hh2(2) = amp2*cos(2*pi*(j-1)*dt/(per2*3600))
        else
          hh2(2) = 0
        endif

      endif

c _____________________________________________________________
```

Table B.2 (cont'd).

```
c thomas algorithm

c find coefficients for thomas algorithm

      call coeff(ii,hh1,hh2,ta,dtmax,t,a,b,c,d,dt)

      cc(1)=c(1)/b(1)
      dd(1)=d(1)/b(1)
      do  k=2,mp1
        kk=k-1
        cc(k)=c(k)/(b(k)-a(k)*cc(kk))
        dd(k)=(d(k)-a(k)*dd(kk))/(b(k)-a(k)*cc(kk))
      enddo
      t(mp1,2)=dd(mp1)
      tavg = t(mp1,2)
      do k=2,mp1
        kk=m-k+2
        t(kk,2)=dd(kk)-cc(kk)*t(kk+1,2)
        tavg = tavg+t(kk,2)
      enddo
      tavg = tavg/mp1
      jj = jj+dt

c_____________________________________________________________

c find quality distribution and adjust time step

      if(model.ge.3)then
      dsum=0
      vsum=0
      endif
      ssum=0
      qavg = 0
      do 85 i=1,mp1
      if(model.ge.3)then
        eex=1.0d0/t(i,2)-1.0d0/tref
        d1=exp(-(ea/r*eex))
        qual(i)=qual(i)-d1*dt
        qavg = qavg+qual(i)
        if(model.eq.4)then
          dqdea=d1*eex*dt/r
          vsum=vsum+d1*eex*dt/r
        endif
        dsum=dsum+d1
      endif
85    continue

c find mass average quality

      if(model.ge.3)then
        qavg = qavg/mp1
        if(model.eq.4)then
          dqdea=vsum/(mp1)
          dq(2)=dq(1)+dqdea
          vq=vq0+dq(2)**2*vea
          dq(1)=dq(2)
```

Table B.2 (cont'd).

```
        endif
      endif
c _____________________________________________________________

      count=count+1
      if(count.ge.60)then

c printout
      nprint = 1
      call output(nprint,headtq,t,tavg,time,jj,qual,qavg,vq,ii,eend,dt)
      count=0
      endif

c _____________________________________________________________

c initial t for next time step

      do 100 i=1,mp1
100   t(i,1)=t(i,2)
      ta(1)=ta(2)
      hh1(1)=hh1(2)
      hh2(1)=hh2(2)
      j=j+1
      jjj=jjj+1
      goto 65

c end of finite difference calculations

c ************************************************************

c set product temp. = ambient temperature; determine quality

105   tavg = 0.0d0
      do 110 i=1,mp1
      do 110 ji=1,2
      t(i,ji)=temp(ii)
110   continue
      tavg = temp(ii)
      cct(ii)=jjj+1
      nx=4
      dt=(stor(ii)-jj)/nx
      do 130 ij=1,nx
      jj=jj+dt
      if(model.ge.3)then
        eex=1.0d0/temp(ii)-1.0d0/tref
        d1=exp(-(ea/r*eex))
        qavg = 0.0d0
        do  i=1,mp1
          qual(i)=qual(i)-d1*dt
          qavg = qavg+qual(i)
        enddo
        qavg=qavg/mp1
        if(model.eq.4)then
          dqdea=d1*eex*dt/r
          dq(2)=dq(1)+dqdea
          vq=vq0+dq(2)**2*vea
```

Table B.2 (cont'd).

```
          dq(1)=dq(2)
        endif
      endif

c printout

      nprint = 1
      call output(nprint,headtq,t,tavg,time,jj,qual,qavg,vq,ii,eend,dt)
      j=j+1
      jjj=jjj+1
      count=0
130   continue
      ct(ii)=jjj+1
140   do 150 i=1,mp1
150   t(i,1)=t(i,2)

      ta(1)=ta(2)
      hh1(1)=hh1(2)
      hh2(1)=hh2(2)
155   if(count.ne.0)then
      nprint = 1
      call output(nprint,headtq,t,tavg,time,jj,qual,qavg,vq,ii,eend,dt)
      endif

      count=0
      if(ii.eq.per)then
      eend=1
      nprint = 2
      call output(nprint,headtq,t,tavg,time,jj,qual,qavg,vq,ii,eend,dt)
      endif
160   continue
      return
      end




      SUBROUTINE COEFF(ii,hh1,hh2,ta,dtmax,t,a,b,c,d,dt)


      parameter(maxm=101,maxp=20,maxd=101,maxc=51,maxs=201)

      integer shape,m,mp1,ii

      double precision beta,nu,omega,gama,hh1(2),hh2(2),
     &aar,ar(maxm),ar1(maxm),area,avg1,avg2,
     &da,db,dc,ddd,ta(2),DENST(maxd),DENSC(maxd),
     &CONDT(maxc),CONDC(maxc),SPHT(maxs),SPHC(maxs),ck(maxm),
     &csd(maxm,2),a(maxm),b(maxm),c(maxm),d(maxm),t(maxm,2),
     &dtmax,dt,pi,dzz,dtt,wf0,ms,dp,kp,cp,t0,h,l,dz,ds,
     &th,tl,avgd,avgk,avgc,ynavg,tdt

      common/geom/shape,h,l,dz,sym,m,mp1,sstep,/prop/wf0,ms,dp,kp,
     &cp,t0,/CONSTP/DENST,DENSC,CONDT,CONDC,SPHT,SPHC,/dff/dif,
     &/NCONSTP/NSD,NSC,NSS,/d/ds,/pavg/th,tl,avgd,avgk,avgc,ynavg,
```

Table B.2 (cont'd).

```
     &/toldt/tdt


      pi = dacos(-1.0d0)

c weighting functions for finite difference method

c modified crank-nicolson method

c weight. coeff. for d2t/dz2
c      for time t:
      beta=0.50d0
c      for time t+1:
      nu=0.50d0

c weight. coeff. for dt/dt
c      for time t:
      omega=-1.0d0
c      for time t+1:
      gama=1.0d0

      q1=0.0d0
      q2=0.0d0

      dzz=1.0d0/dz
      if(shape.eq.2)then
        aar=2.0d0*pi*h
      else
        if(shape.eq.3)then
          aar=4.0d0*pi
        endif
      endif

      do 10 i=1,mp1

c slab

      if(shape.eq.1)then
        ar(i)=h
        ar1(i)=h
      else

c cylinder

        if(shape.eq.2)then
          ar(i)=aar*(i-1)*dz
          ar1(i)=ar(i)+aar*dz/2.0d0
        else

c sphere

          ar(i)=aar*((i-1)*dz)**2.0d0
          ar1(i)=aar*((i-1)*dz+dz/2.0d0)**2.0d0
        endif
      endif
```

Table B.2 (cont'd).

```
10    continue

      CALL PFIND(T,M,CK,CSD,DT,DZ,dtmax)

c ***********************************************************

c 1st boundary point

      AVG1 = (AR(1)+AR1(1))*0.50d0
      a(1)=0.0d0
      c(1)=nu*dzz*CK(1)*ar1(1)
      dc=-beta*dzz*CK(1)*ar1(1)

      b(1)=-gama*CSD(1,1)*avg1-nu*hh1(2)*ar(1)-c(1)
      db=omega*CSD(1,1)*avg1+beta*hh1(1)*ar(1)-dc

      ddd=-beta*(ta(1)*hh1(1)*ar(1)+q1)-nu*(ta(2)*hh1(2)*ar(1)+q1)
      d(1)=db*t(1,1)+dc*t(2,1)+ddd


c ***********************************************************
c
c interior points

      do 20 i=2,m
      AVG1 = (AR(I)+AR1(I))*0.50d0
      AVG2 = (AR(I)+AR1(I-1))*0.50d0
      a(i)=nu*dzz*CK(I-1)*ar1(i-1)
      da=-beta*dzz*CK(I-1)*ar1(i-1)

      c(i)=nu*dzz*CK(I)*ar1(i)
      dc=-beta*dzz*CK(I)*ar1(i)

      b(i)=-gama*(CSD(I,1)*avg2+CSD(I,2)*avg1)-a(i)-c(i)
      db=omega*(CSD(I,1)*avg2+CSD(I,2)*avg1)-da-dc

      d(i)=da*t(i-1,1)+db*t(i,1)+dc*t(i+1,1)
20    continue

c ***********************************************************

c 2nd boundary point

      AVG2 = (AR(mp1)+AR1(M))*0.50d0
      c(mp1)=0.0d0

      a(mp1)=nu*dzz*CK(M)*ar1(m)
      da=-beta*dzz*CK(M)*ar1(m)

      b(mp1)=-gama*CSD(mp1,2)*avg2-nu*hh2(2)*ar(mp1)-a(mp1)
      db=omega*CSD(mp1,2)*avg2+beta*hh2(1)*ar(mp1)-da

      ddd=-beta*(hh2(1)*ta(1)*ar(mp1)+q2)-nu*(ta(2)*hh2(2)*ar(mp1)
     &+q2)
```

Table B.2 (cont'd).

```
      d(mp1)=da*t(m,1)+db*t(mp1,1)+ddd

      return
      end




      SUBROUTINE PFIND(T,M,CK,CSPD,DT,DZ,dtmax)

      PARAMETER (MAXm=101, MAXC=51, MAXD=101, MAXS=201)

      INTEGER NC(8),NSC,NSD,NSS

      double precision TAVGK,TAVGSD(2),CK(MAXM),CSPD(MAXM,2),
     &kc,DC(2),SPC(2),t(maxm,2),dt,dtmax,dz

      double precision CONDT(MAXC),CONDC(MAXC),DENST(MAXD),
     &DENSC(MAXD),SPHT(MAXS),SPHC(MAXS),
     &wf0,ms,dp,kp,cp,t0,ds,
     &th,tl,avgd,avgk,avgc,ynavg,
     &dens,conduc,spheat

      COMMON/CONSTP/DENST,DENSC,CONDT,CONDC,SPHT,SPHC,
     &/NCONSTP/NSD,NSC,NSS,/prop/wf0,ms,dp,kp,cp,t0,/d/ds,
     &/pavg/th,tl,avgd,avgk,avgc,ynavg/toldt/tdt

      external dens,conduc,spheat


      MP1 = M+1
      eigen(1) = 0.
      emax = 1.0e10
      cc = 1000.0d0*dz/2.0d0
      c1 = 1.0d0/(cc*dz)
      DO 100 I = 1,mp1
        if(ynavg.eq.1.0)then
          ck1 = 3.0d0*avgk
          ck(i) = avgk
          do iii = 1,2
            spc(iii) = avgc
            dc(iii) = avgd
          enddo
          go to 90
        endif
        DO KK = 1,5
          NC(KK) = 0
        ENDDO
        IF(I.LE.M)THEN
          TAVGK = (T(I,1)+T(I+1,1))*0.50d0
          TAVGSD(1) = 0.750d0*T(I,1)+0.250d0*T(I+1,1)
        ENDIF
        IF(I.GT.1)THEN
          TAVGSD(2) = 0.750d0*T(I,1)+0.250d0*T(I-1,1)
```

Table B.2 (cont'd).

```
      ENDIF

      DO 10 J = 2,NSC+1
      IF(NC(1).EQ.1)go to 10
      IF(I.Eq.Mp1)go to 10
      if(tavgk.ge.t0)then
        ck(i) = kp
        nc(1) = 1
      else
        if(tavgk.ge.t0-4.0d0)then
          ck(i) = conduc(tavgk)
          nc(1) = 1
        else
          IF(TAVGK.LE.CONDT(J))THEN
            ck(i) = CONDC(J-1)
            NC(1) = 1
          endif
        endif
      endif
10    continue
      if(i.gt.1)then
        ck1 = ck(i-1)+ck(i)+(ck(i-1)*ck(i))**0.5
      endif
      DO 40 J = 2,NSD+1
        DO 30 KK = 1,2
          IF(NC(KK+2).EQ.1)go to 30
          IF(I.EQ.mp1.AND.KK.eq.1)GO TO 30
          IF(I.EQ.1.AND.KK.eq.2)GO TO 30
          if(tavgsd(kk).ge.t0)then
            dc(kk) = dp
            nc(kk+2) = 1
          else
            if(tavgsd(kk).ge.t0-4.0d0)then
              dc(kk) = dens(tavgsd(kk))
              nc(kk+2) = 1
            else
              IF(TAVGSD(KK).LE.DENST(J))THEN
                DC(KK) = DENSC(J-1)
                NC(KK+2) = 1
              ENDIF
            endif
          ENDIF
30      CONTINUE
40    CONTINUE


      DO 60 J = 2,NSS+1
        DO 50 KK = 1,2
          IF(NC(KK+5).EQ.1)go to 50
          IF(I.EQ.mp1.AND.KK.eq.1)GO TO 50
          IF(I.EQ.1.AND.KK.eq.2)GO TO 50
          if(tavgsd(kk).ge.t0)then
            spc(kk) = cp
            nc(kk+5) = 1
          else
            if(tavgsd(kk).ge.t0-4.0d0)then
```

Table B.2 (cont'd).

```
                spc(kk) = spheat(tavgsd(kk))
                nc(kk+5) = 1
              else
                IF(TAVGSD(KK).LE.SPHT(J))THEN
                  SPC(KK) = SPHC(J-1)
                  NC(KK+5) = 1
                ENDIF
              endif
            ENDIF
   50     CONTINUE
   60   CONTINUE
   90   CONTINUE

  100 CONTINUE
      do i = 1,mp1
        DO KK = 1,2
          CSPD(I,KK) = SPC(KK)*DC(KK)*cc/dt
        ENDDO
      enddo
      RETURN
      END




      SUBROUTINE OUTPUT(nprint,headtq,t,tavg,time,jj,qual,qavg,vq,
     &ii,eend,dt)



      parameter(maxp=20,maxm=101)

      integer per,shape,model,sym,sstep,m,eend,day,dead

      double precision wf0,ms,dp,kp,cp,t0,ea,q0,tref,vea,vq0,
     &h,l,dz,ti,temp(maxp),stor(maxp),h1(maxp),h2(maxp),tunit(maxp),
     &abc(5),hr,c7,time,jj,c8,t(maxm,2),qual(maxm),abcd,qavg,vq,tavg
     &,tavgl,dt

      character title*20,ttlfil*4,outfil*12,hh11*29,hh22*21

      common/ttl/title,ttlfil/mod/model/shelf/ea,q0,vea,vq0,tref
     &/prop/wf0,ms,dp,kp,cp,t0/bound/per,ti,temp,stor,h1,h2,tunit
     &/geom/shape,h,l,dz,sym,m,mp1,sstep

C  NPRINT = 0 if printing input parameters and headings
C  NPRINT = 1 if printing temperature distribution and/or
C   quality distributions
C  NPRINT = 2 if printing period no. and end line

      IF(NPRINT.EQ.0)THEN
        GO TO 1100
      ELSE
        if(nprint.eq.1)then
```

Table B.2 (cont'd).

```
          go to 1200
        else
          go to 1300
        endif
      endif
 1100 write(outfil,1000)ttlfil,'out.dat'
 1000 format(' ',a,a)
      open(unit=12,name=outfil(1:12),type='new',carriagecontrol='list')
      write(12,1)title
1     format(' ',///,3x,'Title: ',a20,/3x,'-----',//,14x,'Input Para',
     +'meters',/,14x,16('-')//)
      if(model.ge.3)then
      write(12,3)
3     format(' ','Kinetic Parameters')
      write(12,4)q0
4     format(' ',/,' ',2x,'Reference shelf-life (days).....',
     &f7.1)
      abcd=tref-273.150d0
      write(12,5)abcd
5     format(' ',2x,'Reference temperature (C).....',f6.2)
      abcd=ea/1000.0d0
      write(12,6)abcd
6     format(' ',2x,'Activation energy constant (kJ/mole)...',f8.2)
      abcd=vq0**0.50d0
      if(model.eq.4)then
      write(12,8)abcd
8     format(' ',2x,'St. dev. of ref. shelf-life (days).....',f6.2)
      abcd=vea**0.50d0/1000.0d0
      write(12,9)abcd
9     format(' ',2x,'St. dev. of ea (kj/mole).....',f6.2)
      endif
      endif
      write(12,10)
10    format(' ',/,' ','Unfrozen Product Properties',/)
      abcd=wf0*100.0d0
      write(12,11)abcd
11    format(' ',2x,'Moisture content (%).....',f6.2)
      abcd=t0-273.150d0
      write(12,12)abcd
12    format(' ',2x,'Initial freezing temperature (C).....',f6.2)
      write(12,13)ms
13    format(' ',2x,'Molecular weight of solids (kg/mole)...',f8.2)
      write(12,14)dp
14    format(' ',2x,'Unfrozen product density (kg/m^3).....',f8.2)
      write(12,15)kp
15    format(' ',2x,'Thermal conductivity (W/mK).....',f6.3)
      write(12,16)cp
16    format(' ',2x,'Specific heat (kJ/kgK).....',f7.3)
      abcd=ti-273.150d0
      write(12,17)abcd
17    format(' ',/,' ','Initial Condition:',/,' ',2x,'Product temp.'
     +,' (C) at time=0 .....',f6.2)

c product geometry
```

Table B.2 (cont'd).

```
      if(shape.eq.1)then
      if(sym.eq.1)l = l*2.0d0
      write(12,18)l
18    format( ' ',/,' ','Slab Geometry:',/,' ',2x,'thickness (m)',
     +26('.'),f10.6)
      else
      if(shape.eq.2)then
      write(12,20)l
20    format(' ',/,' ','Cyclindrical Geometry:',/,' ',2x,'radius (m)',
     +29('.'),f10.6)
      else
      write(12,22)l
22    format(' ',/,' ','Spherical Geometry:',/,' ',2x,'radius (m)',
     +29('.'),f10.6)
      endif
      endif

c boundary conditions

      do 40 i=1,per
      write(12,24)i
24    format(' ',/,' ','boundary conditions for period ',i2,':',/)
      abcd=stor(i)/tunit(i)
      if(tunit(i).eq.3600.0d0)then
      write(12,25)abcd
25    format(' ',4x,'storage time(hours).....',f7.2)
      else
      write(12,26)abcd
26    format(' ',4x,'storage time (days).....',f7.2)
      endif
      abcd=temp(i)-273.150d0
      write(12,27)abcd
27    format(' ',4x,'storage temperature (C).....',f6.1)
      write(12,28)
28    format(' ',4x,'convective heat transfer coeff. (W/m^2K):')
      if(shape.eq.1)then
        write(12,29)h1(i)
29      format(' ',6x,'side 1=',f7.2)
        if(sym.ne.1)then
          write(12,30)h2(i)
30        format(' ',6x,'side 2=',f7.2)
        endif
      else
        write(12,35)h2(i)
35      format(' ',6x,'at surface= ',f7.2)
      endif
40    continue
      write(12,45)dt
45    format(4x,'Time step = ',f6.2)



      write(12,100)title
  100 format(' ',/////,' ','Title= ',a20,/)
      if(sym.eq.1)then
      write(12,110)
```

Table B.2 (cont'd).

```
  110 format(' ','Note: Distribution is symmetrical;'/,6x,'results',
     +' are shown for half-thickness only.'/)
      endif

      hh22='DISTRIBUTION HISTORY'
      if(headtq.eq.1)then
      hh11='      TEMPERATURE (C)       '
      else
      hh11='TEMPERATURE (C) & QUALITY (%)'
      endif
      write(12,120)hh11,hh22
  120 format(' ',/,' ',19x,a,/,23x,a,/,19x,27('-'),/)
      if(model.lt.3)then
        write(12,130)
  130 format(' ',28x,'position (m)',/' ',5x,'time',5x,':',42x,
     &'Avg Temp')
      else
        write(12,135)
  135 format(' ',28x,'position (m)',/' ',5x,'time',5x,':',42x,
     &'Avg Temp Qual.')
       endif
       do i = 1,5
        abc(i)=(i-1)*sstep*dz
      enddo
      if(model.lt.3)then
        write(12,137)abc(1),abc(2),abc(3),abc(4),abc(5)
  137 format(' ',4x,'hours     :',5(f8.4))
      else
        if(model.eq.3)then
      write(12,140)abc(1),abc(2),abc(3),abc(4),abc(5)
  140 format(' ',4x,'days + hr :',5(f8.4),2x,'or Qual')
        else
      write(12,145)abc(1),abc(2),abc(3),abc(4),abc(5)
  145 format(' ',4x,'days + hr :',5(f8.4),2x,'or Qual StD(%)')
        endif
      endif
      if(model.ne.4)then
      write(12,150)
  150 format(' ',65('-'))
      else
        write(12,155)
  155   format(72('-'))
      endif
      write(12,160)
  160 format(' ','Period 1      :')


c Printout time heading

 1200 c7=86400.0d0
      tavgl = tavg-273.150d0
      do i = 1,5
      abc(i)=t((i-1)*sstep+1,2)-273.150d0
      enddo
      write(12,190)(time+jj)/3600,abc(1),abc(2),abc(3),abc(4),abc(5),
```

Table B.2 (cont'd).

```
     &tavgl
  190 format(' ',f8.2,'  hour:',6(f7.2,1x),'C')

      if(headtq.eq.2)then


C Printout quality distribution

      c8=100.0d0/(86400.0d0*q0)
      do i = 1,mpl
        if(qual(i).lt.0)then
          dead=1
        endif
      enddo
      do i = 1,5
        abc(i)=qual((i-1)*sstep+1)*c8
      enddo
      if(model.eq.3)then
      write(12,210)abc(1),abc(2),abc(3),abc(4),abc(5),qavg*c8
  210 format(' ',14x,':',6(f7.2,1x),'%')
      else
      write(12,215)abc(1),abc(2),abc(3),abc(4),abc(5),qavg*c8,
     &(vq)**0.50d0*c8
  215 format(' ',14x,':',6(f7.2,1x),1x,e7.1,'%')
      endif
      if(dead.eq.1)then
        write(12,220)
  220   format(' ',18x,'shelf-life has been exceeded')
      endif
      endif
      return



c Printout end line

 1300 if(model.ne.4)then
        write(12,300)
  300 format(' ',65('-'))
      else
        write(12,305)
  305 format(' ',72('-'))
      endif
      if(eend.eq.0)then
        write(12,310)ii
  310   format(' ','period',i3)
      else
        close(unit=12)
      endif
      return
      end
```

APPENDIX C

APPENDIX C

TWO DIMENSIONAL TRANSIENT HEAT CONDUCTION AND QUALITY RETENTION PROGRAM

The two dimensional transient heat conduction program, including estimation of quality retention, discussed in Chapter 3, is presented here. An outline of the program is given in Table C.1, and the listing for the program, written in Fortran 77 for a Vax 11/750 is given in Table C.2.

Table C.1 Description of Two Dimensional Transient Heat Conduction and Quality Retention Program.

Subroutine Title	Description
PROGRAM FREEZE	Main program; contains program menu.
SUBROUTINE PROPER	See Table B.1.
DOUBLE PRECISION FUNCTION MOIST(X)	See Table B.1.
DOUBLE PRECISION FUNCTION DENS(X)	See Table B.1.
DOUBLE PRECISION FUNCTION KI(X)	See Table B.1.
DOUBLE PRECISION FUNCTION CONDUC(X)	See Table B.1.
DOUBLE PRECISION FUNCTION SPHEAT(X)	See Table B.1.
SUBROUTINE CONSPR	See Table B.1.
SUBROUTINE INTEGR	See Table B.1.
BLOCK DATA CONST	See Table B.1.
SUBROUTINE INPUT1	Allows interactive input of ambient conditions and product geometry. Writes output to data file.
SUBROUTINE INPUT2	Allows interactive input of kinetic properties. Writes output to data file.
SUBROUTINE SOLN	Computes temperature distribution and quality retention as a function of temperature. Calls output subroutine.
SUBROUTINE COEFF1	Determines matrix coefficients used in first sweep in ADI finite difference algorithm.
SUBROUTINE COEFF2	Determines matrix coefficients used in second sweep in ADI finite difference algorithm.
SUBROUTINE THOMAL	Solves Thomas Algorithm for inversion of tri-diagonal matrix.

Table C.1 (cont'd).

SUBROUTINE PFIND	Finds values for thermal properties required for ADI finite difference calculations from the property values determined in CONSPR.
SUBROUTINE HEADING	Writes input data to output file.
SUBROUTINE OUTPUT	Writes resulting temperature and quality retention values to output file.

Table C.2 Computer Code for Two Dimensional Transient Heat Conduction and Quality Retention Program.

```
      PROGRAM FREEZE
c*****************************************************************
c*****************************************************************

c                    Residual Shelf-life Program

c                              by

c                          Elaine Scott

c                              1985

c*****************************************************************

c   This program calculates the temperature and quality distri-
c bution histories of a two dimensional frozen food product
c subject to fluctuating ambient temperatures during storage
c below 0C.

c   Input parameters include unfrozen product density, thermal
c conductivity and specific heat.  The initial freezing
c temperature or molecular weight of solids is required to
c predict these values for the frozen food product.

c   Boundary conditions are assumed to be convective, requiring an
c input of the ambient temperature as a function of time, and the
c convective heat transfer coeffient.  The initial condition must
c be a known function of position.

c*****************************************************************



      parameter(maxp=20,maxd=101,maxc=51,maxs=201)

      integer model

      double precision wf0,ms,dp,kp,cp,t0,ds

      character title*20,ttlfil*4,filyn1*1,filyn2*1,filyn*1,fildat*12,
     &inpdat*12

      logical itmode

      common/mod/model,/itm/itmode,/ttl/title,ttlfil,/profil/prpfil,
     &/datfil/fildat,inpdat,kindat,/prop/wf0,ms,dp,kp,cp,t0,/d/ds
```

Table C.2 (cont'd).

```
C  Set ITMODE = .FALSE. if running batch.
      ITMODE = .false.


      IF(ITMODE)THEN

      write(5,1000)
 1000 format('1',72('*'),/,'0',t23,'Residual Shelf-life Program',/,'0',
     &t35,'by',/,'0',t30,'Elaine Scott',/,'0',t24,'Michigan State',
     &' University',/,'0',t30,'January 1986',/,'0',72('*'))
      WRITE(5,100)
  100 FORMAT('0','Program Menu:',//,' ','   1. Product properties (<0C)'
     &,/,' ','   2. Temperature distribution history: known Ta and h',
     &/,' ','   3. Temp. & qual. dist. histories: exact kinetic prop.',
     &/,' ','   4. Temp. & qual. dist. hist.: random kinetic prop.',
     &/,'0','   Ta = Ambient temp.;h = Surface heat trans. coef.',
     &//,' ','Selection? ',$)
      ENDIF
      READ(5,10)model
   10 FORMAT(I1)
      IF(ITMODE)   write(5,200)
  200 format(' ',/,' ','Product: ',$)
      READ(5,20)TITLE
      IF(ITMODE)   write(5,300)
  300 format(' ',/,' ','Key word for data files; 4 Characters: ',$)
      READ(5,20)TTLFIL
   20 FORMAT(A)
      if(model.eq.1)then
        filyn1 = 'n'
      else
        if(itmode) write(5,400)
  400 format(' ',/,' ','Are product properties approximations',/,' ',2x,
     &'with temperature stored on file? (y/n) ',$)
        read(5,20)filyn1
        if(itmode) write(5,500)
  500 format(' ',/,' ','Are input initial and boundary conditions',/,' '
     &,2x,'and geometrical dimensions stored on file? (y/n) ',$)
        read(5,20)filyn2
        if(model.ge.3) then
          if(itmode) write(5,600)
  600 format(' ',/,' ','Are the kinetic properties stored on file? ',
     &'(y/n) ',$)
          read(5,20)filyn3
        endif
      endif
      if(filyn1.eq.'n'.or.filyn1.eq.'N')then
        call proper
        CALL CONSPR
      endif
      if(model.ne.1)then
        if(filyn2.eq.'n'.or.filyn2.eq.'N')then
          call input1
        endif
        if(model.ge.3)then
          if(filyn3.eq.'n'.or.filyn3.eq.'N')then
            call input2
```

Table C.2 (cont'd).

```
          endif
        endif
        call soln
      endif
      end
```

```
      SUBROUTINE PROPER
c See Appendix B.
```

```
      DOUBLE PRECISION FUNCTION MOIST(X)
c See Appendix B.
```

```
      DOUBLE PRECISION FUNCTION DENS(X)
c See Appendix B.
```

```
      DOUBLE PRECISION FUNCTION KI(X)
c See Appendix B.
```

```
      DOUBLE PRECISION FUNCTION CONDUC(X)
c See Appendix B.
```

```
      DOUBLE PRECISION FUNCTION SPHEAT(X)
c See Appendix B.
```

Table C.2 (cont'd).

```
      SUBROUTINE CONSPR

c See Appendix B.




      SUBROUTINE INTEGR(thi,tlow,avgdp,avgkp,avgcp,ncase)

c See Appendix B.




      BLOCK DATA CONST

c See Appendix B.




      SUBROUTINE INPUT1

c   This subroutine provides the input for the boundary condi-
c tions on the product for the case where the ambient temper-
c ature and surface heat tranfer coeffient are known and assumed
c to be constant over a given storage period.

c   Input varibles include, initial product temperature, sym-
c metry of boundary conditions, number of constant temperature
c storage periods, length of storage period, and surface heat
c transfer coefficient.

c   The variables used in this subroutine are:




      parameter(maxp=20)

      integer per,symx,symy,stepx,stepy,unit,shape,ixt,iyt,cyn

      double precision ti,temp(maxp),stor(maxp),htc(maxp,4),tunit(maxp),
     &lx0,lx,ly

      character yn*1,title*20,ttlfil*4,fildat*12,inpdat*12

      logical itmode

      common /ttl/title,ttlfil,/mod/model,/itm/itmode,/datfil/fildat,
     &inpdat,kindat
```

Table C.2 (cont'd).

```
      save



      if(itmode)go to 10
      read*,ti,shape,lx0,lx,ly,per
      do i = 1,per
        read*,temp(i),unit,stor(i)
        read*,htc(i,1),htc(i,2),htc(i,3),htc(i,4)
        if(unit.eq.1)then
          tunit(i)=3600.0d0
        else
          tunit(i)=86400.0d0
        endif
      enddo
      go to 500

   10 write(6,20)
   20 format('1',72('-'),/,'0',27x,'Storage Conditions',/,'0',72('-'))
   30 write(6,40)
   40 format(' ',/,' ','Initial product temperature (C): ',$)
      read*,ti
      write(6,50)
   50 format('0','Enter product geometry: ',/,' ',5x,'1 = slab',/,' ',5x
     &,'2 = cylinder')
      read*,shape
      if(shape.eq.1)then
        write(6,60)
   60   format(' ',/,' ','Enter dimensions for slab:',/,' ',5x,
     &  'width = ',$)
        read*,lx
        lx0 = 0.0d0
        write(6,70)
   70   format(' ',5x,'height or length = ',$)
        read*,ly
        ly0 = 0.0d0
      else
        write(6,80)
   80   format(' ',/,' ','Enter dimensions for cylinder:',//,
     &  ' ',5x,'Is the cylinder hollow? (0=No,1=Yes) ',$)
        read*,cyn
        if(cyn.eq.1)then
          write(6,90)
   90     format(' ',5x,'Inner radius (m)= ',$)
          read*,lx0
        else
          lx0 = 0.0d0
        endif
        write(6,100)
  100   format(' ',5x,'Outer radius (m)= ',$)
        read*,lx
        write(6,110)
  110   format(' ',5x,'Length of cylinder (m)= ',$)
        read*,ly
      endif
      if(shape.eq.1)then
```

Table C.2 (cont'd).

```
      write(6,120)
120   format(' ',/,' ','Indicate whether the heat transfer ',
     & 'coefficients are',/,' ',' the same on opposite ends of the ',
     & 'slab:',/,' ',5x,'in the x-direction (width)? (0=No,1=Yes) ',$)
      read*,symx
      write(6,130)
130   format(' ',5x,'in the y-direction (height or length)? ',
     & ' (0=No,1=Yes) ',$)
      read*,symy
     else
      symx = 0
      write(6,145)
145   format(' ','Are the heat transfer coefficients the same ',
     &   /,' ',5x,'on opposite ends of the cylinder? (0=No,1=Yes) ',$)
      read*,symy
     endif
     write(6,160)
160  format(' ',/,' ','Enter number of constant temp. storage ',
     &'periods: ',$)
     read*,per
     write(6,170)
170  format(' ',/,' ','Are these values correct? (y/n) ',$)
     read(5,180)yn
180  format(a)
     if(yn.ne.'y'.and.yn.ne.'Y')goto 30

c input boundary conditions for each storage period

     do  i=1,per
190   write(6,200)i
200   format(' ',/,' ','Enter data for period  ',i3,':',/,
     & ' ',5x,'Storage temperature (C): ',$)
      read*,temp(i)
210   write(6,220)
220   format(' ',5x,'Enter units for storage temp.:',/,' ',7x,
     & '1= hours',/,' ',7x,'2= days')
      read*,unit
      if(unit.le.0.and.unit.gt.2)then
        print*,'Try again!'
        goto 210
      endif
      if(unit.eq.1)then
        tunit(i)=3600.0d0
      else
        tunit(i)=86400.0d0
      endif
230   write(6,240)
240   format(' ',/,' ',5x,'Length of storage period: ',$)
      read*,stor(i)
      if(stor(i).lt.0)then
        print*,'Try again!'
        goto 230
      endif
      write(6,250)
250   format('0',5x,'Enter surface heat transfer coef. (W/m^2C):')
      if(shape.eq.1)then
```

Table C.2 (cont'd).

```
          write(6,260)
  260     format(' ',9x,'at side 1 along width of slab = ',$)
          read*,htc(i,1)
        else
          if(cyn.eq.1)then
            write(6,270)
  270       format(' ',9x,'along inner radius of cylinder = ',$)
            read*,htc(i,1)
          else
            htc(i,1) = 0.0d0
          endif
        endif
        if(symx.ne.1.)then
          if(shape.eq.1)then
            write(6,280)
  280       format(' ',9x,'at side 2 along width of slab = ',$)
          else
            write(6,290)
  290       format(' ',9x,'along outer radius of cylinder = ',$)
          endif
          read*,htc(i,3)
        else
          htc(i,3) = 0.0d0
        endif
        if(shape.eq.1)then
          write(6,300)
  300     format(' ',9x,'at side 1 along height or length of slab = ',$)
        else
          write(6,310)
  310     format(' ',9x,'at side 1 along length of cylinder = ',$)
        endif
        read*,htc(i,2)
        if(symy.ne.1.)then
          if(shape.eq.1)then
            write(6,320)
  320     format(' ',9x,'at side 2 along height or length of slab = ',$)
          else
            write(6,330)
  330       format(' ',9x,'at side 2 along length of cylinder = ',$)
          endif
          read*,htc(i,4)
        else
          htc(i,4) = 0.0d0
        endif
        write(6,170)
        read(5,180)yn
        if(yn.ne.'y'.and.yn.ne.'Y')goto 190
      enddo
c input geometry and size

500   ti = ti+273.150d0
      if(symx.eq.1.and.shape.eq.1)Lx = Lx*0.50d0
      if(symy.eq.1)Ly = Ly*0.50d0
      do i = 1,per
        temp(i) = temp(i)+273.150d0
        stor(i) = stor(i)*tunit(i)
```

Table C.2 (cont'd).

```
      enddo
      if((lx-lx0).ge.0.40d0)then
        ixt = 20
        stepx = 5
        go to 510
      endif
      if(lx.lt.0.02d0)then
        ixt = 2
        stepx = 1
        go to 510
      endif
      do i=1,5
        if((lx-lx0).lt.i*0.040d0)then
          ixt = i*4
          stepx = i
          go to 510
        endif
      enddo
  510 if(ly.ge.0.40d0)then
        iyt = 20
        stepy = 5
        go to 520
      endif
      if(ly.lt.0.02d0)then
        iyt = 2
        stepy = 1
        go to 520
      endif
     do i=1,5
        if(ly.lt.i*0.040d0)then
          iyt = i*4
          stepy = i
          go to 520
        endif
      enddo
      write(inpdat,530)ttlfil,'inp.dat'
  530 format(' ',a,a)
      open(unit=12,name=inpdat(1:12),type='new',carriagecontrol='list')
      write(12,*)ti,shape,lx0,lx,ly
      write(12,*)symx,symy,cyn,per
      do i = 1,per
        write(12,*)temp(i),stor(i),tunit(i)
        write(12,*)htc(i,1),htc(i,2),htc(i,3),htc(i,4)
      enddo
      write(12,*)ixt,stepx,iyt,stepy
      close(unit=12)
      return
      end
```

Table C.2 (cont'd).

```
      SUBROUTINE INPUT2

c     Kinetic properties to determine quality loss in a food product
c  are entered in this subroutine.  A file titled 'TTLFILkin.dat'
c  containing the kinetic properties is created.  This file is re-
c  opened in the solution, and it may be reused again in subsequent
c  runs.

      integer model

      double precision q0,tref,ea,vea,vq0

      character title*20,ttlfil*4,fildat*12,inpdat*12,kindat*12

      logical itmode

      common /ttl/title,ttlfil,/mod/model,/itm/itmode,
     &/datfil/fildat,inpdat,kindat


c  Read batch file data (if itmode = .false.)

      if(itmode)go to 1
      read*,q0,tref,ea
      if(model.eq.4)then
        read*,vea,vq0
      endif
      go to 30

c  Read interactive input

    1  write(5,2)
    2 format('1',72('-'),/,'0',t23,'Reference Shelf-life data',/,'0',
     &72('-'))
   10 write(5,100)
  100 format(' ',/,' ','Enter reference shelf-life (days) : ',$)
      read*,q0
      write(5,200)
  200 format(' ',/,' ','Enter reference temperature for reference',
     &' shelf-life (C) : ',$)
      read*,tref
      write(5,300)
  300 format(' ',/,' ','Enter activation energy constant (kJ/mole) : '
     &,$)
      read*,ea
      if(model.eq.4)then
        write(5,400)
  400 format(' ',/,' ','Enter standard deviation of activation ',
     &'energy const. (kJ/mole) : ',$)
        read*,vea
        write(5,500)
  500 format(' ',/,' ','Enter standard deviation of ref. shelf-life',
     &' (days) : ',$)
        read*,vq0
      endif
```

Table C.2 (cont'd).

```
      write(5,550)
  550 format(' ',/,' ','Are these values correct? (y/n) ',$)
      read(5,20)yn
   20 format(a)
      if(yn.eq.'n'.or.yn.eq.'N')go to 10
c
   30 tref = tref+273.150d0
      ea = ea*1000.0d0
      if(model.eq.4)then
        vea = (vea*1000.0d0)**2.0d0
        vq0 = vq0*vq0
      endif
c
      write(kindat,600)ttlfil,'kin.dat'
  600 format(' ',a,a)
      open(unit=12,name=kindat(1:12),type='new',carriagecontrol='list')
      write(12,700)q0,tref,ea
  700 format(' ',2x,f8.1,2x,f7.2,2x,f13.0)
      if(model.eq.4)then
        write(12,800)vea,vq0
  800 format(' ',2(2x,e11.3))
      endif
      close(unit=12)
      return
      end



c*******************************************************************************
c*******************************************************************************

      SUBROUTINE SOLN

      Parameter(maxd=101,maxc=51,maxs=201,maxx=31,maxp=20,
     &tol=1.0d0,r=8.3140d0)

      Integer per,shape,ixt,iyt,symx,symy,stepx,stepy,bc,xy,ix,iy,
     &cyn,ynavg,ixy,ixp1,iyp1

      Double Precision dt2,dens,conduc,spheat,ds,
     &htflx(maxp,4),a(maxx),b(maxx),c(maxx),d(maxx),
     &txy(maxx),pi,hxy(4,3),bcxy(4,3),qual(maxx,maxx),dq(2),
     &time,ptime,tavg,qavg,vq,ftime,t(maxx,maxx,3)

c  Declare all variables in common blocks.

      Double Precision wf0,ms,dp,kp,cp,t0,
     &denst(maxd),densc(maxd),condt(maxc),condc(maxc),spht(maxs),
     &  sphc(maxs),
     &ti,temp(maxp),stor(maxp),htc(maxp,4),tunit(maxp),
     &lx,lx0,ly,dx,dy,
     &ea,q0,vea,vq0,tref,
     &tl,th,avgd,avgk,avgc
```

Table C.2 (cont'd).

```
      character title*20,ttlfil*4,fildat*12,inpdat*12,kindat*12

      Common /prop/wf0,ms,dp,kp,cp,t0,
     &/conp/denst,densc,condt,condc,spht,sphc,
     &/nconp/nsd,nsc,nss,
     &/bound/per,ti,temp,stor,htc,tunit
     &/geom/shape,lx,lx0,ly,dx,dy,symx,symy,cyn,ixpl,iypl,stepy,
     &/shelf/ea,q0,vea,vq0,tref,
     &/mod/model,/itm/itmode,
     &/datfil/fildat,inpdat,kindat,
     &/ttl/title,ttlfil,
     &/pavg/tl,th,avgd,avgk,avgc,ynavg

      save

c  Read in geometry, dimensions, boundary conditions and initial
condition


      write(inpdat,100)ttlfil,'inp.dat'
  100 format(' ',a,a)
      open(unit=12,name=inpdat(1:12),type='old',carriagecontrol='list')
      read(12,*)ti,shape,lx0,lx,ly
      read(12,*)symx,symy,cyn,per
      do i = 1,per
        read(12,*)temp(i),stor(i),tunit(i)
        read(12,*)htc(i,1),htc(i,2),htc(i,3),htc(i,4)
      enddo
      read(12,*)ixt,stepx,iyt,stepy
      close(unit=12)

c  Read in constant property assumptions and associated temperature
ranges


c  Read in constant property assumptions


      write(fildat,110)'karlprp.dat'
  110 format(' ',a)
      open(unit=12,name=fildat(1:12),type='old',carriagecontrol='list')
      read(12,*)wf0,t0,ms
      read(12,*)dp,kp,cp
      read(12,*)nsd,nsc,nss
      read(12,*)tl,th,avgd,avgk,avgc,ynavg
      do i = 1,nsd
        read(12,*)denst(i),densc(i)
      enddo
      do i = 1,nsc
        read(12,*)condt(i),condc(i)
      enddo
      do i = 1,nss
        read(12,*)spht(i),sphc(i)
      enddo
      close(unit=12)
```

Table C.2 (cont'd).

```
c  Read in kinetic data

      if(model.ge.3)then
        write(kindat,100)ttlfil,'kin.dat'
        open(unit=12,name=kindat,type='old',carriagecontrol='list')
        read(12,*)q0,tref,ea
        if(model.eq.4)then
          read(12,*)vea,vq0
        else
          vea = 0.0d0
          vq0 = 0.0d0
        endif
        close(unit=12)
      endif

c  Set imposed heat flux equal to 0; user may change if desired.

      do i = 1,per
        do k = 1,4
          htflx(i,k) = 0.0d0
        enddo
      enddo
      dx = (lx-lx0)/dfloat(ixt)
      dy = ly/dfloat(iyt)
      ixpl = ixt+1
      iypl = iyt+1
      pi = dacos(-1.0d0)
      do i = 1,ixpl
        do j = 1,iypl
          do k = 1,3
            t(i,j,k) = ti
          enddo
          if(model.ge.3)qual(i,j) = q0*86400.0d0
        enddo
      enddo
      tavg = ti
      do k = 1,3
        if(shape.eq.1)then
          hxy(1,k) = htc(1,1)*dy
          hxy(2,k) = htc(1,2)*dx
          hxy(3,k) = htc(1,3)*dy
          hxy(4,k) = htc(1,4)*dx
          bcxy(1,k) = (htc(1,1)*temp(1)+htflx(1,1))*dy
          bcxy(2,k) = (htc(1,2)*temp(1)+htflx(1,2))*dx
          bcxy(3,k) = (htc(1,3)*temp(1)+htflx(1,3))*dy
          bcxy(4,k) = (htc(1,4)*temp(1)+htflx(1,4))*dx
        else
          hxy(1,k) = htc(1,1)*dy*2.0d0*pi*lx0
          hxy(2,k) = htc(1,2)*pi*dx
          hxy(3,k) = htc(1,3)*dy*2.0d0*pi*lx
          hxy(4,k) = htc(1,4)*pi*dx
          bcxy(1,k) = (htc(1,1)*temp(1)+htflx(1,1))*dy*2.0d0*pi*lx0
          bcxy(2,k) = (htc(1,2)*temp(1)+htflx(1,2))*pi*dx
          bcxy(3,k) = (htc(1,3)*temp(1)+htflx(1,3))*dy*2.0d0*pi*lx
          bcxy(4,k) = (htc(1,4)*temp(1)+htflx(1,4))*pi*dx
        endif
```

Table C.2 (cont'd).

```
      enddo
      if(model.ge.3)then
        qavg = q0*86400.0d0
        if(model.eq.4)then
          dq(1) = 0.0d0
          dq(2) = 0.0d0
          vq0 = vq0*86400.0d0
          vq = vq0
        else
          vq = 0.0d0
        endif
      else
        qavg = 0.0d0
        vq = 0.0d0
      endif
      time = 0.0d0
      ptime = 0.0d0
        ftime = 0.0d0
      count = 0
      nptime = 0
      nccc = 0
      nprint = 0
      if(model.lt.3)then
        headtq = 1
      else
        headtq = 2
      endif
      call headng
      nprint = 0
      call output(t,time,ptime,1,nprint,tavg,qual,qavg,vq)
      ntime = 1

c*********************************************************************

      do 500 ii = 1,per
      ptime = 0.0d0
      if(ii.ne.1)then
        time = time+stor(ii-1)
        if(nccc.ne.0)then
          nprint = 1
          call output(t,time,ptime,ii,nprint,tavg,qual,qavg,vq)
        endif
      endif
      dt2 = 600.0d0
      nptime = 0
      if(per.ne.1)then
      do k = 2,3
        if(shape.eq.1)then
          ta = temp(ii)
          hxy(1,k) = htc(ii,1)*dy
          hxy(2,k) = htc(ii,2)*dx
          hxy(3,k) = htc(ii,3)*dy
          hxy(4,k) = htc(ii,4)*dx
          bcxy(1,k) = (htc(ii,1)*temp(ii)+htflx(ii,1))*dy
          bcxy(2,k) = (htc(ii,2)*temp(ii)+htflx(ii,2))*dx
          bcxy(3,k) = (htc(ii,3)*temp(ii)+htflx(ii,3))*dy
```

Table C.2 (cont'd).

```
          bcxy(4,k) = (htc(ii,4)*temp(ii)+htflx(ii,4))*dx
        else
          hxy(1,k) = htc(ii,1)*dy*2.0d0*pi*lx0
          hxy(2,k) = htc(ii,2)*pi*dx
          hxy(3,k) = htc(ii,3)*dy*2.0d0*pi*lx
          hxy(4,k) = htc(ii,4)*pi*dx
          bcxy(1,k) = (htc(ii,1)*temp(ii)+htflx(ii,1))*dy*2.0d0*pi*lx0
          bcxy(2,k) = (htc(ii,2)*temp(ii)+htflx(ii,2))*pi*dx
          bcxy(3,k) = (htc(ii,3)*temp(ii)+htflx(ii,3))*dy*2.0d0*pi*lx
          bcxy(4,k) = (htc(ii,4)*temp(ii)+htflx(ii,4))*pi*dx
        endif
      enddo
      endif
c_____________________________________________________________________

c  Check if time is > length of storage period

  200 if(ptime.ge.stor(ii))go to 500

c  Check if temp. is close to ambient temperature

      eabs = abs(t(1,1,3)-temp(ii))
      do i = 1,ixpl
        do j = 1,iypl
          if(i.ne.1.and.j.ne.1)then
            eeabs = abs(t(i,j,3)-temp(ii))
            if(eabs.lt.eeabs)eabs = eeabs
          endif
        enddo
      enddo
      if(eabs.lt.tol)go to 300
      ptime = ptime+dt2

c_____________________________________________________________________
_

      if(ptime.gt.stor(ii))then
        kjtime = ptime-dt2
        ptime = stor(ii)
        dt2 = ptime-kjtime
      endif

c_____________________________________________________________________


c  First sweep

      xy = 0
      itpl = ixpl
      do j = 1,iypl
        ixy=j
        if(j.eq.1)bc = -1
        if(j.eq.iypl)bc = 1
        if(j.ne.1.and.j.ne.iypl)bc = 0
        call coef1(t,ixy,xy,hxy,bcxy,dt2,bc,a,b,c,d)
        call thomal(a,b,c,d,itpl,txy)
```

Table C.2 (cont'd).

```
        do i = 1,ixpl
          t(i,j,2) = txy(i)
        enddo
      enddo
c_____________________________________________________________________

c  Next sweep in the Y-direction

      xy = 1
      itpl = iypl
      do j = 1,ixpl
        ixy = j
        if(j.eq.1)bc = -1
        if(j.eq.ixpl)bc = 1
        if(j.ne.1.and.j.ne.ixpl)bc = 0
        call coef2(t,ixy,xy,hxy,bcxy,dt2,bc,a,b,c,d)
        call thomal(a,b,c,d,iypl,txy)
        do i = 1,iypl
          t(j,i,3) = txy(i)
        enddo
      enddo
      tavg = 0.0d0
      do i = 1,ixpl
        do j = 1,iypl
          tavg = t(i,j,3)+tavg
        enddo
      enddo
      tavg = tavg/(ixpl*iypl)

c_____________________________________________________________________

c  Find quality distribution and adjust time step

      if(model.ge.3)then
        vsum = 0
        qavg = 0
        do i = 1,ixpl
          do j = 1,iypl
            eex = 1.0d0/t(i,j,3)-1.0d0/tref
            dl = exp(-(ea/r*eex))
            qual(i,j) = qual(i,j)-dl*dt2
            qavg = qavg+qual(i,j)
            if(model.eq.4)then
              dqdea =-dl*eex*dt2/r
              vsum = vsum+dqdea
            endif
          enddo
        enddo

c  Find mass average quality and variance

        qavg = qavg/(ixpl*iypl)
        if(model.eq.4)then
          dqdea = vsum/(ixpl*iypl)
          dq(2) = dq(1)+dqdea
```

Table C.2 (cont'd).

```
          vq = vq0+dq(2)**2*vea
          dq(1) = dq(2)
        endif
      endif
c _____________________________________________________________

      ftime = ftime+dt2
c      if(ftime.ge.0.25d0*stor(ii))then
      nccc = nccc+1
      if(nccc.eq.6)then

c printout

        nprint = 2
        call output(t,time,ptime,ii,nprint,tavg,qual,qavg,vq)
        ftime = 0.0d0
        nccc = 0
      endif
c _____________________________________________________________
c _____________________________________________________________

c initial t for next time step

      do i = 1,ixp1
        do j = 1,iyp1
          t(i,j,1)=t(i,j,3)
        enddo
      enddo
      do i = 1,4
        hxy(i,1) = hxy(i,3)
        bcxy(i,1) = bcxy(i,3)
      enddo
      ntime = ntime+1
      nptime = nptime+1
      goto 200

c end of finite difference calculations

c ************************************************************

c set product temp. = ambient temperature; determine quality

300   tavg = 0.0d0
      do i = 1,ixp1
        do j = 1,iyp1
          do k = 1,3
            t(i,j,k) = temp(ii)
          enddo
        enddo
      enddo
      tavg = temp(ii)
      nx=1
      dt2 = (stor(ii)-ptime)/nx
      do ij = 1,nx
        ptime = ptime+dt
        if(model.ge.3)then
```

Table C.2 (cont'd).

```
          eex = 1.0d0/temp(ii)-1.0d0/tref
          dl = exp(-(ea/r*eex))
          qavg = 0.0d0
          do i = 1,ixpl
            do j = 1,iypl
              qual(i,j) = qual(i,j)-dl*dt2
              qavg = qavg+qual(i,j)
            enddo
          enddo
          qavg=qavg/(ixpl*iypl)
          if(model.eq.4)then
            dqdea=dl*eex*dt/r
            dq(2)=dq(1)+dqdea
            vq=vq0+dq(2)**2*vea
            dq(1)=dq(2)
          endif
        endif
        ntime = ntime+1
        nptime = nptime+1
        count=0
      enddo
      do i = 1,ixpl
        do j = 1,iypl
          t(i,j,1) = t(i,j,3)
        enddo
      enddo
      do i = 1,4
        hxy(i,1) = hxy(i,3)
        bcxy(i,1) = bcxy(i,3)
      enddo
      if(ii.eq.per.and.nccc.ne.0)then
        nprint = 3
        call output(t,time,ptime,ii,nprint,tavg,qual,qavg,vq)
      endif
  500 continue
      return
      end




c**********************************************************************
c**********************************************************************


      SUBROUTINE COEF1(t,ixy,xy,hxy,bcxy,dt,bc,a,b,c,d)


      Parameter(maxd=101,maxc=51,maxs=201,maxx=31)

      integer bc,it,itpl,ixy,shape,symx,symy,ixpl,iypl,stepx,stepy,
     &nsd,nsc,nss,cyn,xy,ynavg

      Double precision pi,beta,nu,omega,gama,hxy(4,3),bcxy(4,3),
```

Table C.2 (cont'd).

```
     &a(maxx),b(maxx),c(maxx),d(maxx),dax2,dbx2,dcx2,dax,day,dbxy,
     &dcy,dcx,dd,ck(maxx,8),csd(maxx,8),dt,t(maxx,maxx,3),hxy2,hxy4,
     &bxy2,bxy4

c  Declare all variables in common blocks.

      Double Precision wf0,ms,dp,kp,cp,t0,
     &denst(maxd),densc(maxd),condt(maxc),condc(maxc),spht(maxs),
     &  sphc(maxs),
     &lx,lx0,ly,dx,dy,
     &tl,th,avgd,avgk,avgc

      common /prop/wf0,ms,dp,kp,cp,t0,
     &/nconp/nsd,nsc,nss,
     &/conp/denst,densc,condt,condc,spht,sphc,
     &/geom/shape,lx,lx0,ly,dx,dy,symx,symy,cyn,ixpl,iypl,stepy,
     &/pavg/tl,th,avgd,avgk,avgc,ynavg

      pi = dacos(-1.0d0)
      ix = ixpl-1

c  Weighting function for ADI finite difference method.

c    Modified Crank-Nicolson Method

c    1.  Weighting coefficients for d2T/dx2;

c        a.  at time t:
      beta = 0.50d0

c        b.  at time t+1/2*dt and t+dt
      nu = 0.50d0

c     2.  Weighting coefficients for dT/dt;

c        a.  at time t:
      omega = -1.0d0
      gama = 1.0d0

      itpl = ixpl

c  Find product property values for each y-value for constant x (ix).

      call pfind(t,ixy,xy,itpl,dt,bc,ck,csd)


c_____________________________________________________________________

c  1st boundary point

      if(bc.eq.-1)then
        if(shape.eq.1)then
          hxy2 = hxy(2,1)
          bxy2 = bcxy(2,1)
        else
          hxy2 = hxy(2,1)*(lx0+0.25d0*dx)
```

Table C.2 (cont'd).

```
          bxy2 = bcxy(2,1)*(lx0+0.25d0*dx)
        endif
        a(1) = 0.0d0
        c(1) = nu*ck(1,3)
        b(1) = -nu*hxy(1,2)*0.50d0-gama*csd(1,3)-c(1)
        dcy = -ck(1,4)
        dcx = -beta*ck(1,3)
        dbxy = (beta*hxy(1,1)+hxy2)*0.50d0+omega*csd(1,3)-dcy-dcx
        dd = -(nu*bcxy(1,2)+beta*bcxy(1,1)+bxy2)*0.50d0
        d(1) = dbxy*t(1,ixy,1)+dcy*t(1,ixy+1,1)+dcx*t(2,ixy,1)+dd
      endif
      if(bc.eq.0)then
        c(1) = nu*ck(1,3)
        b(1) = -nu*hxy(1,2)-gama*(csd(1,2)+csd(1,3))-c(1)
        day = -ck(1,2)
        dcy = -ck(1,4)
        dcx = -beta*ck(1,3)
        dbxy = beta*hxy(1,1)+omega*(csd(1,2)+csd(1,3))-day-dcy-dcx
        dd = -nu*bcxy(1,2)-beta*bcxy(1,1)
        d(1) = day*t(1,ixy-1,1)+dbxy*t(1,ixy,1)+dcy*t(1,ixy+1,1)+
     &  dcx*t(2,ixy,1)+dd
      endif
      if(bc.eq.1)then
        if(shape.eq.1)then
          hxy4 = hxy(4,1)
          bxy4 = bcxy(4,1)
        else
          hxy4 = hxy(4,1)*(lx0+0.25d0*dx)
          bxy4 = bcxy(4,1)*(lx0+0.25d0*dx)
        endif
        c(1) = nu*ck(1,3)
        b(1) = -nu*hxy(1,2)*0.50d0-gama*csd(1,2)-c(1)
        day = -ck(1,2)
        dcx = -beta*ck(1,3)
        dbxy = (beta*hxy(1,1)+hxy4)*0.50d0+omega*csd(1,2)-day-dcx
        dd = -(nu*bcxy(1,2)+beta*bcxy(1,1)+bxy4)*0.50d0
        d(1) = day*t(1,ixy-1,1)+dbxy*t(1,ixy,1)+dcx*t(2,ixy,1)+dd
      endif

c_____________________________________________________________________

c  Interior Points

      do i = 2,ix
        if(bc.eq.-1)then
          if(shape.eq.1)then
            hxy2 = hxy(2,1)
            bxy2 = bcxy(2,1)
          else
            hxy2 = hxy(2,1)*2.0d0*(lx0+dfloat(i-1)*dx)
            bxy2 = bcxy(2,1)*2.0d0*(lx0+dfloat(i-1)*dx)
          endif
          a(i) = nu*ck(i,1)
          c(i) = nu*ck(i,3)
          b(i) = -gama*(csd(i,3)+csd(i,4))-a(i)-c(i)
```

Table C.2 (cont'd).

```
          dax = -beta*ck(i,1)
          dcy = -ck(i,4)
          dcx = -beta*ck(i,3)
          dbxy = hxy2+omega*(csd(i,3)+csd(i,4))-dax-dcy-dcx
          dd = -bxy2
          d(i) = dax*t(i-1,ixy,1)+dbxy*t(i,ixy,1)+dcy*t(i,ixy+1,1)
     &    +dcx*t(i+1,ixy,1)+dd
        endif
        if(bc.eq.0)then
          a(i) = nu*ck(i,1)
          c(i) = nu*ck(i,3)
          b(i) = -gama*(csd(i,1)+csd(i,2)+csd(i,3)+csd(i,4))-a(i)-c(i)
          dax = -beta*ck(i,1)
          day = -ck(i,2)
          dcy = -ck(i,4)
          dcx = -beta*ck(i,3)
          dbxy = omega*(csd(i,1)+csd(i,2)+csd(i,3)+csd(i,4))-dax-day-dcy
     &    -dcx
          d(i) = dax*t(i-1,ixy,1)+day*t(i,ixy-1,1)+dbxy*t(i,ixy,1)+
     &   dcy*t(i,ixy+1,1)+dcx*t(i+1,ixy,1)
        endif
        if(bc.eq.1)then
          if(shape.eq.1)then
            hxy4 = hxy(4,1)
            bxy4 = bcxy(4,1)
          else
            hxy4 = hxy(4,1)*2.0d0*(lx0+(i-1)*dx)
            bxy4 = bcxy(4,1)*2.0d0*(lx0+(i-1)*dx)
          endif
          a(i) = nu*ck(i,1)
          c(i) = nu*ck(i,3)
          b(i) = -gama*(csd(i,1)+csd(i,2))-a(i)-c(i)
          dax = -beta*ck(i,1)
          day = -ck(i,2)
          dcx = -beta*ck(i,3)
          dbxy = hxy4+omega*(csd(i,1)+csd(i,2))-dax-day-dcx
          dd = -bxy4
          d(i) = dax*t(i-1,ixy,1)+day*t(i,ixy-1,1)+dbxy*t(i,ixy,1)
     &    +dcx*t(i+1,ixy,1)+dd
        endif
      enddo

c_____________________________________________________________________

c  2nd Boundary

      if(bc.eq.-1)then
        if(shape.eq.1)then
          hxy2 = hxy(2,1)
          bxy2 = bcxy(2,1)
        else
          hxy2 = hxy(2,1)*(lx-0.25d0*dx)
          bxy2 = bcxy(2,1)*(lx-0.25d0*dx)
        endif
        a(ixp1) = nu*ck(ixp1,1)
        b(ixp1) = -nu*hxy(3,2)*0.50d0-gama*csd(ixp1,4)-a(ixp1)
```

Table C.2 (cont'd).

```
        dax = -beta*ck(ixpl,1)
        dcy = -ck(ixpl,4)
        dbxy = (beta*hxy(3,1)+hxy2)*0.50d0+omega*csd(ixpl,4)-dax-dcy
        dd = -(nu*bcxy(3,2)+beta*bcxy(3,1)+bxy2)*0.50d0
        d(ixpl) = dax*t(ix,ixy,1)+dbxy*t(ixpl,ixy,1)+
     &  dcy*t(ixpl,ixy+1,1)+dd
       endif
       if(bc.eq.0)then
        a(ixpl) = nu*ck(ixpl,1)
        b(ixpl) = -nu*hxy(3,2)-gama*(csd(ixpl,1)+csd(ixpl,4))-a(ixpl)
        dax = -beta*ck(ixpl,1)
        day = -ck(ixpl,2)
        dcy = -ck(ixpl,4)
        dbxy = beta*hxy(3,1)+omega*(csd(ixpl,1)+csd(ixpl,4))-dax-day-dcy
        dd = -nu*bcxy(3,2)-beta*bcxy(3,1)

        d(ixpl) = dax*t(ix,ixy,1)+day*t(ixpl,ixy-1,1)+dbxy*t(ixpl,ixy,1)
     &  +dcy*t(ixpl,ixy+1,1)+dd
       endif
       if(bc.eq.1)then
        if(shape.eq.1)then
          hxy4 = hxy(4,1)
          bxy4 = bcxy(4,1)
        else
          hxy4 = hxy(4,1)*(1x-0.25d0*dx)
          bxy4 = bcxy(4,1)*(1x-0.25d0*dx)
        endif
        a(ixpl) = nu*ck(ixpl,1)
        b(ixpl) = -nu*hxy(3,2)*0.50d0-gama*csd(ixpl,1)-a(ixpl)
        dax = -beta*ck(ixpl,1)
        day = -ck(ixpl,2)
        dbxy = (beta*hxy(3,1)+hxy4)*0.50d0+omega*csd(ixpl,1)-dax-day
        dd = -(nu*bcxy(3,2)+beta*bcxy(3,1)+bxy4)*0.50d0
        d(ixpl) = dax*t(ix,ixy,1)+day*t(ixpl,ixy-1,1)+dbxy*t(ixpl,ixy,1)
     &  +dd
       endif

       return
       end


c**********************************************************************
c**********************************************************************


       SUBROUTINE COEF2(t,ixy,xy,hxy,bcxy,dt,bc,a,b,c,d)

c  This is the preliminary version of the subroutine which finds the
coef
c  for the second sweep in the Y-direction.


       Parameter(maxd=101,maxc=51,maxs=201,maxx=31)

       integer bc,iy,itpl,xy,ixy,nsd,nsc,nss,shape,symx,symy,ixpl,
     &iypl,stepx,stepy,cyn,ynavg
```

Table C.2 (cont'd).

```
      Double precision pi,betax,betay,nuy,nux,omega,gama,hxy(4,3),
     &bcxy(4,3),a(maxx),b(maxx),c(maxx),d(maxx),dax2,dbx2,dcx2,dax,
     &day,dbxy,dcy,dcx,dd,ck(maxx,8),csd(maxx,8),t(maxx,maxx,3),
     &hxy2a,hxy2b,hxy4a,hxy4b,bxy2a,bxy2b,bxy4a,bxy4b

c  Declare all variables in common blocks.

      Double Precision wf0,ms,dp,kp,cp,t0,
     &denst(maxd),densc(maxd),condt(maxc),condc(maxc),spht(maxs),
     &  sphc(maxs),
     &lx,lx0,ly,dx,dy,
     &tl,th,avgd,avgk,avgc

      common /prop/wf0,ms,dp,kp,cp,t0,
     &/conp/denst,densc,condt,condc,spht,sphc,
     &/nconp/nsd,nsc,nss,
     &/geom/shape,lx,lx0,ly,dx,dy,symx,symy,cyn,ixp1,iyp1,stepy,
     &/pavg/tl,th,avgd,avgk,avgc,ynavg


      pi = dacos(-1.0d0)

c  Weighting function for ADI finite difference method.

c    Modified Crank-Nicolson Method

c    1.  Weighting coefficients for d2T/dx2 and d2T/dy2;

c        a.  at time t:
      betax = 0.50d0
      betay = 0.50d0

c        b.  at time t+1/2*dt and t+dt
      nux = 0.50d0
      nuy = 0.50d0

c     2.  Weighting coefficients for dT/dt;

c        a.  at time t:
      omega = -1.0d0
      gama = 1.0d0

      iy = iyp1-1

c  Find product property values for each y-value for constant x (ix).

      call pfind(t,ixy,xy,iyp1,dt,bc,ck,csd)
c_____________________________________________________________________

c  1st boundary point

      a(1) = 0.0d0
      if(bc.eq.-1)then
        if(shape.eq.1)then
          hxy2a = hxy(2,1)
          hxy2b = hxy(2,3)
```

Table C.2 (cont'd).

```
          bxy2a = bcxy(2,1)
          bxy2b = bcxy(2,3)
        else
          hxy2a = hxy(2,1)*(lx0+0.25d0*dx)
          hxy2b = hxy(2,3)*(lx0+0.25d0*dx)
          bxy2a = bcxy(2,1)*(lx0+0.25d0*dx)
          bxy2b = bcxy(2,3)*(lx0+0.25d0*dx)
        endif
        c(1) = nuy*ck(1,8)
        b(1) = -nuy*hxy2b*0.50d0-gama*csd(1,7)-c(1)
        dcx2 = -nux*ck(1,7)
        dbx2 = nux*hxy(1,2)*0.50d0-dcx2
        dcy = -betay*ck(1,4)
        dcx = -betax*ck(1,3)
        dbxy = (betax*hxy(1,1)+betay*hxy2a)*0.50d0+omega*csd(1,3)-dcy
     &  -dcx
        dd = -(nux*bcxy(1,2)+betax*bcxy(1,1)+nuy*bxy2b+
     &  betay*bxy2a)*0.50d0
        d(1) = dbx2*t(ixy,1,2)+dcx2*t(ixy+1,1,2)+dbxy*t(ixy,1,1)+
     &  dcy*t(ixy,2,1)+dcx*t(ixy+1,1,1)+dd
       endif
       if(bc.eq.0)then
         if(shape.eq.1)then
           hxy2a = hxy(2,1)
           hxy2b = hxy(2,3)
           bxy2a = bcxy(2,1)
           bxy2b = bcxy(2,3)
         else
           hxy2a = hxy(2,1)*2.0d0*(lx0+(ixy-1)*dx)
           hxy2b = hxy(2,3)*2.0d0*(lx0+(ixy-1)*dx)
           bxy2a = bcxy(2,1)*2.0d0*(lx0+(ixy-1)*dx)
           bxy2b = bcxy(2,3)*2.0d0*(lx0+(ixy-1)*dx)
         endif
         c(1) = nuy*ck(1,8)
         b(1) = -nuy*hxy2b-gama*(csd(1,7)+csd(1,8))-c(1)
         dax2 = -nux*ck(1,5)
         dcx2 = -nux*ck(1,7)
         dbx2 = -dax2-dcx2
         dax = -betax*ck(1,1)
         dcy = -betay*ck(1,4)
         dcx = -betax*ck(1,3)
         dbxy = betay*hxy2a+omega*(csd(1,3)+csd(1,4))-dax-dcy-dcx
         dd = -nuy*bxy2b-betay*bxy2a
         d(1) = dax2*t(ixy-1,1,2)+dbx2*t(ixy,1,2)+dcx2*t(ixy+1,1,2)+
     &  dax*t(ixy-1,1,1)+dbxy*t(ixy,1,1)+dcy*t(ixy,2,1)+
     &  dcx*t(ixy+1,1,1)+dd
       endif
       if(bc.eq.1)then
         if(shape.eq.1)then
           hxy2a = hxy(2,1)
           hxy2b = hxy(2,3)
           bxy2a = bcxy(2,1)
           bxy2b = bcxy(2,3)
         else
           hxy2a = hxy(2,1)*(lx-0.25d0*dx)
           hxy2b = hxy(2,3)*(lx-0.25d0*dx)
```

Table C.2 (cont'd).

```
          bxy2a = bcxy(2,1)*(1x-0.25d0*dx)
          bxy2b = bcxy(2,3)*(1x-0.25d0*dx)
        endif
        c(1) = nuy*ck(1,8)
        b(1) = -nuy*hxy2b*0.50d0-gama*csd(1,8)-c(1)
        dax2 = -nux*ck(1,5)
        dbx2 = nux*hxy(3,2)*0.50d0-dax2
        dax = -betax*ck(1,1)
        dcy = -betay*ck(1,4)
        dbxy = (betax*hxy(3,1)+betay*hxy2a)*0.50d0+omega*csd(1,4)-dax
     &  -dcy
        dd = -(nux*bcxy(3,2)+betax*bcxy(3,1)+nuy*bxy2b+
     &  betay*bxy2a)*0.5d0
        d(1) = dax2*t(ixy-1,1,2)+dbx2*t(ixy,1,2)+dax*t(ixy-1,1,1)+
     &  dbxy*t(ixy,1,1)+dcy*t(ixy,2,1)+dd
       endif

c_____________________________________________________________________

c  Interior Points

      do i = 2,iy
        if(bc.eq.-1)then
          a(i) = nuy*ck(i,6)
          c(i) = nuy*ck(i,8)
          b(i) = -gama*(csd(i,6)+csd(i,7))-a(i)-c(i)
          dcx2 = -nux*ck(i,7)
          dbx2 = nux*hxy(1,2)-dcx2
          day = -betay*ck(i,2)
          dcy = -betay*ck(i,4)
          dcx = -betax*ck(i,3)
          dbxy = betax*hxy(1,1)+omega*(csd(i,2)+csd(i,3))-day-dcy-dcx
          dd = -nux*bcxy(1,2)-betax*bcxy(1,1)
          d(i) = dbx2*t(ixy,i,2)+dcx2*t(ixy+1,i,2)+day*t(ixy,i-1,1)+
     &    dbxy*t(ixy,i,1)+dcy*t(ixy,i+1,1)+dcx*t(ixy+1,i,1)+dd
        endif
        if(bc.eq.0)then
          a(i) = nuy*ck(i,6)
          c(i) = nuy*ck(i,8)
          b(i) = -gama*(csd(i,5)+csd(i,6)+csd(i,7)+csd(i,8))-a(i)-c(i)
          dax2 = -nux*ck(i,5)
          dcx2 = -nux*ck(i,7)
          dbx2 = -dax2-dcx2
          dax = -betax*ck(i,1)
          day = -betay*ck(i,2)
          dcy = -betay*ck(i,4)
          dcx = -betax*ck(i,3)
          dbxy = omega*(csd(i,1)+csd(i,2)+csd(i,3)+csd(i,4))-dax-day
     &    -dcy-dcx
          d(i) = dax2*t(ixy-1,i,2)+dbx2*t(ixy,i,2)+dcx2*t(ixy+1,i,2)+
     &    dax*t(ixy-1,i,1)+day*t(ixy,i-1,1)+dbxy*t(ixy,i,1)+
     &    dcy*t(ixy,i+1,1)+dcx*t(ixy+1,i,1)
        endif
        if(bc.eq.1)then
          a(i) = nuy*ck(i,6)
```

Table C.2 (cont'd).

```
          c(i) = nuy*ck(i,8)
          b(i) = -gama*(csd(i,5)+csd(i,8))-a(i)-c(i)
          dax2 = -nux*ck(i,5)
          dbx2 = nux*hxy(3,2)-dax2
          dax = -betax*ck(i,1)
          day = -betay*ck(i,2)
          dcy = -betay*ck(i,4)
          dbxy = betax*hxy(3,1)+omega*(csd(i,1)+csd(i,4))-dax-day-dcy
          dd = -nux*bcxy(3,2)-betax*bcxy(3,1)
          d(i) = dax2*t(ixy-1,i,2)+dbx2*t(ixy,i,2)+dax*t(ixy-1,i,1)+
     &    day*t(ixy,i-1,1)+dbxy*t(ixy,i,1)+dcy*t(ixy,i+1,1)+dd
        endif
       enddo

c_____________________________________________________________________

c  2nd Boundary

       c(iypl) = 0.0d0
       if(bc.eq.-1)then
         if(shape.eq.1)then
           hxy4a = hxy(4,1)
           hxy4b = hxy(4,3)
           bxy4a = bcxy(4,1)
           bxy4b = bcxy(4,3)
         else
           hxy4a = hxy(4,1)*(1x+0.25d0*dx)
           hxy4b = hxy(4,3)*(1x+0.25d0*dx)
           bxy4a = bcxy(4,1)*(1x+0.25d0*dx)
           bxy4b = bcxy(4,3)*(1x+0.25d0*dx)
         endif
         a(iypl) = nuy*ck(iypl,6)
         b(iypl) = -nuy*hxy4b*0.50d0-gama*csd(iypl,6)-a(iypl)
         dcx2 = -nux*ck(iypl,7)
         dbx2 = nux*hxy(1,2)*0.50d0-dcx2
         day = -betay*ck(iypl,2)
         dcx = -betax*ck(iypl,3)
         dbxy = (betax*hxy(1,1)+betay*hxy4a)*0.50d0+omega*csd(iypl,2)-
     &   day-dcx
         dd = -(nux*bcxy(1,2)+betax*bcxy(1,1)+nuy*bxy4b+
     &   betay*bxy4a)*0.50d0
         d(iypl) = dbx2*t(ixy,iypl,2)+dcx2*t(ixy+1,iypl,2)+
     &   day*t(ixy,iypl-1,1)+dbxy*t(ixy,iypl,1)+dcx*t(ixy+1,iypl,1)+dd
       endif
       if(bc.eq.0)then
         if(shape.eq.1)then
           hxy4a = hxy(4,1)
           hxy4b = hxy(4,3)
           bxy4a = bcxy(4,1)
           bxy4b = bcxy(4,3)
         else
           hxy4a = hxy(4,1)*2.0d0*(1x0+(ixy-1)*dx)
           hxy4b = hxy(4,3)*2.0d0*(1x0+(ixy-1)*dx)
           bxy4a = bcxy(4,1)*2.0d0*(1x0+(ixy-1)*dx)
           bxy4b = bcxy(4,3)*2.0d0*(1x0+(ixy-1)*dx)
```

Table C.2 (cont'd).

```
        endif
        a(iypl) = nuy*ck(iypl,6)
        b(iypl) = -nuy*hxy4b-gama*(csd(iypl,5)+csd(iypl,6))-a(iypl)
        dax2 = -nux*ck(iypl,5)
        dcx2 = -nux*ck(iypl,7)
        dbx2 = -dax2-dcx2
        dax = -betax*ck(iypl,1)
        day = -betay*ck(iypl,2)
        dcx = -betax*ck(iypl,3)
        dbxy = betay*hxy4a+omega*(csd(iypl,1)+csd(iypl,2))-dax-day-
     &  dcx
        dd = -nuy*bxy4b-betay*bxy4a
        d(iypl) = dax2*t(ixy-1,iypl,2)+dbx2*t(ixy,iypl,2)+
     &  dcx2*t(ixy+1,iypl,2)+dax*t(ixy-1,iypl,1)+day*t(ixy,iy,1)+
     &  dbxy*t(ixy,iypl,1)+dcx*t(ixy+1,iypl,1)+dd
       endif
       if(bc.eq.1)then
         if(shape.eq.1)then
           hxy4a = hxy(4,1)
           hxy4b = hxy(4,3)
           bxy4a = bcxy(4,1)
           bxy4a = bcxy(4,3)
         else
           hxy4a = hxy(4,1)*(1x-0.25d0*dx)
           hxy4b = hxy(4,3)*(1x-0.25d0*dx)
           bxy4a = bcxy(4,1)*(1x-0.25d0*dx)
           bxy4a = bcxy(4,3)*(1x-0.25d0*dx)
         endif
         a(iypl) = nuy*ck(iypl,6)
         b(iypl) = -nuy*hxy4b*0.50d0-gama*csd(iypl,5)-a(iypl)
         dax2 = -nux*ck(iypl,5)
         dbx2 = nux*hxy(3,2)*0.50d0-dax2
         dax = -betax*ck(iypl,1)
         day = -betay*ck(iypl,2)
         dbxy = (betax*hxy(3,1)+betay*hxy4a)*0.50d0+omega*csd(iypl,1)-
     &   dax-day
         dd = -(nux*bcxy(3,2)+betax*bcxy(3,1)+nuy*bxy4b+
     &   betay*bxy4a)*0.50d0
         d(iypl) = dax2*t(ixy-1,iypl,2)+dbx2*t(ixy,iypl,2)+
     &     dax*t(ixy-1,iypl,1)+day*t(ixy,iy,1)+dbxy*t(ixy,iypl,1)+dd
        endif

       return
       end


c**********************************************************************
c**********************************************************************



       SUBROUTINE THOMAL(a,b,c,d,itpl,t)

       Parameter(maxx = 31)
```

Table C.2 (cont'd).

```
      integer itpl

      double precision a(maxx),b(maxx),c(maxx),d(maxx),cc(maxx),
     &dd(maxx),t(maxx)

      cc(1) = c(1)/b(1)
      dd(1) = d(1)/b(1)
      do i = 2,itpl
        ii = i-1
        cc(i) = c(i)/(b(i)-a(i)*cc(ii))
        dd(i) = (d(i)-a(i)*dd(ii))/(b(i)-a(i)*cc(ii))
      enddo
      t(itpl) = dd(itpl)
      do i = 2,itpl
        ii = itpl-i+1
        t(ii) = dd(ii)-cc(ii)*t(ii+1)
      enddo
      return
      end


c**********************************************************************
c**********************************************************************



      SUBROUTINE PFIND(t,ixy,xy,itpl,dt,bc,ck,csd)

      Parameter (maxd=101,maxc=51,maxs=201,maxx=31)

      Integer nsd,nsc,nss,k1,k2,k3,k4,k5,k6,k7,k8,sd2,sd4,sd6,sd8,xy,
     &nc(24),itpl,bc,ixy,shape,cyn,symx,symy,ixpl,iypl,stepy,ynavg

      Double Precision pi,dt,dti,dxi,dyi,vol1,vol2,tavgk(8),
     &tavgsd(8),ck(maxx,8),dc(8),spc(8),csd(maxx,8),
     &dens,conduc,spheat,ark(3),volsd(2),t(maxx,maxx,3)

c  Declare all variables in common blocks.

      Double Precision wf0,ms,dp,kp,cp,t0,
     &denst(maxd),densc(maxd),condt(maxc),condc(maxc),spht(maxs),
     &  sphc(maxs),
     &ds,
     &lx,lx0,ly,dx,dy,
     &tl,th,avgd,avgk,avgc

      Common /prop/wf0,ms,dp,kp,cp,t0,
     &/conp/denst,densc,condt,condc,spht,sphc,
     &/nconp/nsd,nsc,nss,
     &/d/ds
     &/geom/shape,lx,lx0,ly,dx,dy,symx,symy,cyn,ixpl,iypl,stepy
     &/pavg/tl,th,avgd,avgk,avgc,ynavg

      External dens,conduc,spheat
```

Table C.2 (cont'd).

```
      pi = dacos(-1.0d0)
      dti = 1000.0d0/dt
      dxi = 1.0d0/dx
      dyi = 1.0d0/dy
      vol1 = dx*dy*0.25d0
      vol2 = pi*dx*dy*0.5d0



      if(xy.eq.0)then
        kkxy = 4
        kk1 = 4
        kk2 = 8
        kk3 = 12
      else
        kkxy = 8
        kk1 = 8
        kk2 = 16
        kk3 = 24
      endif


      do 100 i = 1,itp1

        if(ynavg.eq.1)then
          do ii = 1,8
            ck(i,ii) = avgk
            dc(ii) = avgd
            spc(ii) = avgc
          enddo
          go to 40
        endif
        do kk = 1,kk3
          nc(kk) = 0
          if(kk.le.kk1)then
            tavgk(kk) = 0.0d0
            tavgsd(kk) = 0.0d0
          endif
        enddo
        if(xy.eq.0)then
          if(i.ne.1)tavgk(1) = 0.50*(t(i,ixy,1)+t(i-1,ixy,1))
          if(i.ne.ixp1)tavgk(3) = 0.50*(t(i,ixy,1)+t(i+1,ixy,1))
          if(ixy.ne.1)tavgk(2) = 0.50*(t(i,ixy,1)+t(i,ixy-1,1))
          if(ixy.ne.iyp1)tavgk(4) = 0.50*(t(i,ixy,1)+t(i,ixy+1,1))

          if(i.ne.1.and.ixy.ne.1)then
            tavgsd(1) = 0.0625*(9.0*t(i,ixy,1)+3.0*t(i-1,ixy,1)+
     &      3.0*t(i,ixy-1,1)+t(i-1,ixy-1,1))
          endif
          if(i.ne.ixp1.and.ixy.ne.1)then
            tavgsd(2) = 0.0625*(9.0*t(i,ixy,1)+3.0*t(i+1,ixy,1)+
     &      3.0*t(i,ixy-1,1)+t(i+1,ixy-1,1))
          endif
          if(i.ne.ixp1.and.ixy.ne.iyp1)then
            tavgsd(3) = 0.0625*(9.0*t(i,ixy,1)+3.0*t(i+1,ixy,1)+
```

Table C.2 (cont'd).

```
     &      3.0*t(i,ixy+1,1)+t(i+1,ixy+1,1))
            endif
            if(i.ne.1.and.ixy.ne.iypl)then
              tavgsd(4) = 0.0625*(9.0*t(i,ixy,1)+3.0*t(i-1,ixy,1)+
     &      3.0*t(i,ixy+1,1)+t(i-1,ixy+1,1))
            endif
          else
            if(ixy.ne.1)then
              tavgk(1) = 0.50*(t(ixy,i,1)+t(ixy-1,i,1))
              tavgk(5) = 0.50*(t(ixy,i,1)+t(ixy-1,i,1))
            endif
            if(ixy.ne.ixpl)then
              tavgk(3) = 0.50*(t(ixy,i,1)+t(ixy+1,i,1))
              tavgk(7) = 0.50*(t(ixy,i,1)+t(ixy+1,i,1))
            endif
            if(i.ne.1)then
              tavgk(2) = 0.50*(t(ixy,i,1)+t(ixy,i-1,1))
              tavgk(6) = 0.50*(t(ixy,i,1)+t(ixy,i-1,1))
            endif
            if(i.ne.iypl)then
              tavgk(4) = 0.50*(t(ixy,i,1)+t(ixy,i+1,1))
              tavgk(8) = 0.50*(t(ixy,i,1)+t(ixy,i+1,1))
            endif
            if(i.ne.1.and.ixy.ne.1)then
              tavgsd(1) = 0.0625*(9.0*t(ixy,i,1)+3.0*t(ixy-1,i,1)+
     &      3.0*t(ixy,i-1,1)+t(ixy-1,i-1,1))
              tavgsd(5) = 0.0625*(9.0*t(ixy,i,1)+3.0*t(ixy-1,i,1)+
     &      3.0*t(ixy,i-1,1)+t(ixy-1,i-1,1))
            endif
            if(i.ne.1.and.ixy.ne.ixpl)then
              tavgsd(2) = 0.0625*(9.0*t(ixy,i,1)+3.0*t(ixy+1,i,1)+
     &      3.0*t(ixy,i-1,1)+t(ixy+1,i-1,1))
              tavgsd(6) = 0.0625*(9.0*t(ixy,i,1)+3.0*t(ixy+1,i,1)+
     &      3.0*t(ixy,i-1,1)+t(ixy+1,i-1,1))
            endif
            if(i.ne.iypl.and.ixy.ne.ixpl)then
              tavgsd(3) = 0.0625*(9.0*t(ixy,i,1)+3.0*t(ixy+1,i,1)+
     &      3.0*t(ixy,i+1,1)+t(ixy+1,i+1,1))
              tavgsd(7) = 0.0625*(9.0*t(ixy,i,1)+3.0*t(ixy+1,i,1)+
     &      3.0*t(ixy,i+1,1)+t(ixy+1,i+1,1))
            endif
            if(i.ne.iypl.and.ixy.ne.1)then
              tavgsd(4) = 0.0625*(9.0*t(ixy,i,1)+3.0*t(ixy-1,i,1)+
     &      3.0*t(ixy,i+1,1)+t(ixy-1,i+1,1))
              tavgsd(8) = 0.0625*(9.0*t(ixy,i,1)+3.0*t(ixy-1,i,1)+
     &      3.0*t(ixy,i+1,1)+t(ixy-1,i+1,1))
            endif
          endif

c*************************************************************************

c  Find Product Properties Cooresponding to Averaged Temperatures

c     First find appropriate thermal conductivity value
```

Table C.2 (cont'd).

```
        do 20 kk = 1,kkxy
          if(tavgk(kk).eq.0)go to 20
          if(tavgk(kk).ge.t0)then
            ck(i,kk) = kp
          else
            if(tavgk(kk).ge.t0-4.0d0)then
              ck(i,kk) = conduc(tavgk(kk))
            else
              do j = 2,nsc
                if(nc(kk).eq.1)go to 20
                if(tavgk(kk).le.condt(j))then
                  ck(i,kk) = condc(j-1)
                  nc(kk) = 1
               endif
              enddo
            endif
          endif
   20   continue

c*************************************************************************

c     Next find appropriate density and specific heat value

        do 30 kk = 1,kkxy
          if(tavgsd(kk).eq.0)go to 30
          if(tavgsd(kk).ge.t0)then
            dc(kk) = dp
            spc(kk) = cp
            nc(kk+kk1) = 1
            nc(kk+kk2) = 1
          else
            if(tavgsd(kk).ge.t0-4.0d0)then
              dc(kk) = dens(tavgsd(kk))
              spc(kk) = spheat(tavgsd(kk))
              nc(kk+kk1) = 1
              nc(kk+kk2) = 1
            else
              do j = 2,nsd
                if(nc(kk+kk1).eq.1)go to 25
                if(tavgsd(kk).le.denst(j))then
                  dc(kk) = densc(j-1)
                  nc(kk+kk1) = 1
                endif
              enddo
   25         do j = 2,nss
                if(nc(kk+kk2).eq.1)go to 30
                if(tavgsd(kk).le.spht(j))then
                  spc(kk) = sphc(j-1)
                  nc(kk+kk2) = 1
                endif
              enddo
            endif
          endif
   30   continue
```

Table C.2 (cont'd).

```
c**********************************************************************

c  Determine surface area for conductive heat transfer

c    Areas for sweep in the X-direction


   40     if(shape.eq.1.and.xy.eq.0)then
          ark(2) = dx
          if(i.eq.1.or.i.eq.itp1)ark(2) = dx*0.50d0
          if(bc.ne.-1.and.bc.ne.1)ark(1) = dy
          if(bc.eq.-1.or.bc.eq.1)ark(1) = dy*0.50d0
          ark(3) = ark(1)
        endif
        if(shape.eq.2.and.xy.eq.0)then
          if(i.eq.1)ark(2) = pi*dx*(lx0+0.25d0*dx)
          if(i.eq.itp1)ark(2) = pi*dx*(lx-0.25d0*dx)
          if(i.ne.1.and.i.ne.itp1)ark(2) = pi*dx*(lx0+(i-1)*dx)
          ark(1) = 2.0d0*pi*dy*(lx0+((i)-1.50d0)*dx)
          ark(3) = 2.0d0*pi*dy*(lx0+((i)-0.50d0)*dx)
          if(bc.eq.-1.or.bc.eq.1)then
            ark(1) = 0.50d0*ark(1)
            ark(3) = 0.50d0*ark(3)
          endif
        endif

c    Areas for sweep in the Y-direction

        if(shape.eq.1.and.xy.eq.1)then
          ark(2) = dx
          if(bc.eq.1.or.bc.eq.-1)ark(2) = dx*0.50d0
          if(i.ne.1.and.i.ne.itp1)ark(1) = dy
          if(i.eq.1.or.i.eq.itp1)ark(1) = dy*0.50d0
          ark(3) = ark(1)
        endif
        if(shape.eq.2.and.xy.eq.1)then
          if(bc.eq.-1)ark(2) = pi*dx*(lx0+0.250d0*dx)
          if(bc.eq.1)ark(2) = pi*dx*(lx0-0.250d0*dx)
          if(bc.eq.0)ark(2) = pi*dx*(lx0+((ixy)-1.0d0)*dx)
          if(bc.ne.-1)ark(1) = 2.0d0*pi*dy*(lx0+((ixy)-1.5d0)*dx)
          if(bc.ne.1)ark(3) = 2.0d0*pi*dy*(lx0+((ixy)-0.5d0)*dx)
          if(i.eq.1.or.i.eq.itp1)then
            ark(1) = ark(1)*0.50d0
            ark(3) = ark(3)*0.50d0
          endif
        endif

c  Determine Volume Elements for use with Specific Volume

        if(shape.eq.1)then
          do jj = 1,2
            volsd(jj) = vol1*dti
          enddo
        endif
        if(shape.eq.2.and.xy.eq.0)then
```

Table C.2 (cont'd).

```
          if(i.ne.1)volsd(1) = vol2*(lx0+((i)-1.250d0)*dx)*dti
          if(i.ne.itp1)volsd(2) = vol2*(lx0+((i)-0.750d0)*dx)
     &    *dti
        endif
        if(shape.eq.2.and.xy.eq.1)then
          if(bc.ne.-1)volsd(1) = vol2*(lx0+((ixy)-1.250d0)*dx)
     &    *dti
          if(bc.ne.1)volsd(2) = vol2*(lx0+((ixy)-0.750d0)*dx)
     &    *dti
        endif

c*****************************************************************

c  Multiply thermal conductivity values by areas, and density*specific
heat
c     by volumes

c     X - sweep

        if(xy.eq.0)then
          if(i.ne.1)then
            ck(i,1) = ck(i,1)*ark(1)*dxi
            if(bc.ne.-1)then
              csd(i,1) = dc(1)*spc(1)*volsd(1)
            endif
            if(bc.ne.1)then
              csd(i,4) = dc(4)*spc(4)*volsd(1)
            endif
          endif
          if(i.ne.itp1)then
            ck(i,3) = ck(i,3)*ark(3)*dxi
            if(bc.ne.-1)then
              csd(i,2) = dc(2)*spc(2)*volsd(2)
            endif
            if(bc.ne.1)then
              csd(i,3) = dc(3)*spc(3)*volsd(2)
            endif
          endif
          if(bc.ne.-1)then
            ck(i,2) = ck(i,2)*ark(2)*dyi
          endif
          if(bc.ne.1)then
            ck(i,4) = ck(i,4)*ark(2)*dyi
          endif
        endif

c     Y - Sweep

        if(xy.eq.1)then
          if(bc.ne.-1)then
            ck(i,1) = ck(i,1)*ark(1)*dxi
            ck(i,5) = ck(i,5)*ark(1)*dxi
            if(i.ne.1)then
              csd(i,1) = dc(1)*spc(1)*volsd(1)
              csd(i,5) = dc(5)*spc(5)*volsd(1)
```

Table C.2 (cont'd).

```
            endif
            if(i.ne.itpl)then
              csd(i,4) = dc(4)*spc(4)*volsd(2)
              csd(i,8) = dc(8)*spc(8)*volsd(2)
            endif
          endif
          if(bc.ne.1)then
            ck(i,3) = ck(i,3)*ark(3)*dxi
            ck(i,7) = ck(i,7)*ark(3)*dxi
            if(i.ne.1)then
              csd(i,2) = dc(2)*spc(2)*volsd(2)
              csd(i,6) = dc(6)*spc(6)*volsd(2)
            endif
            if(i.ne.itpl)then
              csd(i,3) = dc(3)*spc(3)*volsd(1)
              csd(i,7) = dc(7)*spc(7)*volsd(1)
            endif
          endif
          if(i.ne.1)then
            ck(i,2) = ck(i,2)*ark(2)*dyi
            ck(i,6) = ck(i,6)*ark(2)*dyi
          endif
          if(i.ne.itpl)then
            ck(i,4) = ck(i,4)*ark(2)*dyi
            ck(i,8) = ck(i,8)*ark(2)*dyi
          endif
        endif
  100 continue
      return
      end



c*******************************************************************************
c*******************************************************************************
      SUBROUTINE HEADNG(nprint,headtq)



      parameter(maxp=20)

      integer per,shape,model,sym,sstep,m,eend,day,dead,cyn

c  Declare all variables in common blocks.

      Double Precision wf0,ms,dp,kp,cp,t0,
     &ti,temp(maxp),stor(maxp),htc(maxp,4),tunit(maxp),
     &lx,lx0,ly,dx,dy,
     &ea,q0,vea,vq0,tref

      character title*20,ttlfil*4,outfil*12

      Common /prop/wf0,ms,dp,kp,cp,t0,
     &/bound/per,ti,temp,stor,htc,tunit,
     &/geom/shape,lx,lx0,ly,dx,dy,symx,symy,cyn,ixpl,iypl,stepy,
```

Table C.2 (cont'd).

```
      &/shelf/ea,q0,vea,vq0,tref,
      &/mod/model,
      &/ttl/title,ttlfil

       write(outfil,1000)ttlfil,'out.dat'
 1000  format(' ',a,a)
       open(unit=12,name=outfil(1:12),type='new',carriagecontrol='list')
       write(12,1)title
1      format(' ',///,3x,'Title: ',a20,/3x,'-----',//,14x,'Input Para',
      +'meters',/,14x,16('-')//)
       if(model.ge.3)then
       write(12,3)
3      format(' ','Kinetic Parameters')
       write(12,4)q0
4      format(' ',/,' ',2x,'Reference shelf-life (days).....',
      &f7.1)
       abcd=tref-273.150d0
       write(12,5)abcd
5      format(' ',2x,'Reference temperature (C).....',f6.2)
       abcd=ea/1000.0d0
       write(12,6)abcd
6      format(' ',2x,'Activation energy constant (kJ/mole)...',f8.2)
       abcd=vq0**0.50d0
       if(model.eq.4)then
       write(12,8)abcd
8      format(' ',2x,'St. dev. of ref. shelf-life (days).....',f6.2)
       abcd=vea**0.50d0/1000.0d0
       write(12,9)abcd
9      format(' ',2x,'St. dev. of ea (kj/mole).....',f6.2)
       endif
       endif
       write(12,10)
10     format(' ',/,' ','Unfrozen Product Properties',/)
       abcd=wf0*100.0d0
       write(12,11)abcd
11     format(' ',2x,'Moisture content (%).....',f6.2)
       abcd=t0-273.150d0
       write(12,12)abcd
12     format(' ',2x,'Initial freezing temperature (C).....',f6.2)
       write(12,13)ms
13     format(' ',2x,'Molecular weight of solids (kg/mole)...',f8.2)
       write(12,14)dp
14     format(' ',2x,'Unfrozen product density (kg/m^3).....',f8.2)
       write(12,15)kp
15     format(' ',2x,'Thermal conductivity (W/mK).....',f6.3)
       write(12,16)cp
16     format(' ',2x,'Specific heat (kJ/kgK).....',f7.3)
       abcd=ti-273.150d0
       write(12,17)abcd
17     format(' ',/,' ','Initial Condition:',/,' ',2x,'Product temp.'
      +,' (C) at time=0 .....',f6.2)

c product geometry

       if(symy.eq.1)ly = ly*2.0d0
```

Table C.2 (cont'd).

```
      if(shape.eq.1)then
        if(symx.eq.1)lx = lx*2.0d0
        write(12,18)lx,ly
18    format( ' ',/,' ','Slab Geometry:',/,' ',2x,'width',
     &  ': x direction (m)',17('.'),f9.5,/,' ',2x,'height or length',
     &  ': y direction (m)',6('.'),f9.5)
      else
        if(lx0.eq.0.0)then
          write(12,20)lx,ly
20    format(' ',/,' ','Cyclindrical Geometry:',/,' ',2x,'radius (m)',
     &    29('.'),f9.5,/,' ',2x,'height or length: y direction (m)',
     &    6('.'),f9.5)
        else
          write(12,22)lx0,lx,ly
22    format(' ',/,' ','Cyclindrical Geometry:',/,' ',2x,'inner radius
',
     &    '(m)',23('.'),f9.5,/,' ',2x,'outer radius (m)',23('.'),f9.5,
     &    /,' ',2x,'height or length: y direction (m)',6('.'),f9.5)
        endif
      endif

c boundary conditions

      do 50 i=1,per
      write(12,24)i
24    format(' ',/,' ','Boundary Conditions for Period ',i2,':',/)
      abcd=stor(i)/tunit(i)
      if(tunit(i).eq.3600.0d0)then
        write(12,25)abcd
25    format(' ',2x,'Storage time(hours).....................',f7.2)
      else
        write(12,26)abcd
26    format(' ',2x,'Storage time (days).....................',f7.2)
      endif
      abcd=temp(i)-273.150d0
      write(12,27)abcd
27    format(' ',2x,'Storage temperature (C).................',f6.2,/)
      write(12,28)
28    format(' ',2x,'Convective heat transfer coeff. (W/m^2K):')
      if(shape.eq.1)then
        write(12,30)htc(i,1)
30      format(' ',4x,'in the x-direction: ',/,' ',6x,
     &         'at x = 0.0 .......................',f7.2)
        if(symx.eq.0)then
          write(12,32)lx,htc(i,3)
32        format(' ',6x,'at x = ',f9.5,' .................',f7.2)
        endif
      else
        if(cyn.eq.1)then
          write(12,34)lx0,htc(i,1)
34        format(' ',4x,'in the radial direction: ',/,' ',6x,
     &    'at x = ',f9.5,' (inner radius)','....',f7.2)
        endif
        write(12,36)lx,htc(i,3)
36        format(' ',6x,'at x = ',f9.5,' (outer radius)','....',f7.2)
      endif
```

Table C.2 (cont'd).

```
      write(12,38)htc(i,2)
38    format(' ',4x,'in the y-direction: ',/,' ',6x,
     &                'at y = 0.0 .....................',f7.2)
      if(symx.eq.0)then
        write(12,40)ly,htc(i,4)
40      format(' ',6x,'at y = ',f9.5,' .................',f7.2)
      endif

50    continue

      return
      end

c**********************************************************************

      SUBROUTINE OUTPUT(t,time,ptime,ii,nprint,tavg,qual,qavg,vq)


      parameter(maxx = 31)

      integer ixpl,iypl,ii,model,per,shape,symx,symy,stepy,cyn

      double precision t(maxx,maxx,3),time,ptime,lx,lx0,ly,dx,dy,
     &abc(10),qual(maxx,maxx),tavg,qavg,vq,ea,q0,vea,vq0,tref,hr,phr,
     &ttime,pttime,day,pday


      character title*20,ttlfil*4,outdat*15,hh11*29,hh22*21

      common/ttl/title,ttlfil,
     &/geom/shape,lx,lx0,ly,dx,dy,symx,symy,cyn,ixpl,iypl,stepy,
     &/shelf/ea,q0,vea,vq0,tref,
     &/mod/model


C  NPRINT = 0 if printing initial conditions at the beginning of first
C           storage period.
C  NPRINT = 1 if printing temperature distribution and/or quality
C           distributions at the end/beginning of itermentent storage
C           periods.
C  NPRINT = 2 If printing temperature distribution and/or quality
C           distributions at itermentent times during a storage period.
C  NPRINT = 3 if printing temperature distribution and/or quality
C           distributions at the end of last storage period.

      write(12,20)title
   20 format(///,10x,'Product = ',a,//)
      ttime = time+ptime
      day = ttime/86400.0d0
      hr = ttime/3600.d0
      pday = ptime/84600.0d0
      phr = ptime/3600.0d0
      if(nprint.eq.0)then
        write(12,30)
   30   format(10x,'These are the initial conditions at the beginning ',
     &  'of the ',/,13x,'first storage period.',/)
```

Table C.2 (cont'd).

```
      else if(nprint.eq.1)then
        write(12,40)
     &  'Total elapsed time ....................',day,'days',hr,'hrs',
     &  'Beginning of storage period ..................',ii,
     &  'End of storage period ........................',ii-1
   40   format(10x,a,f6.2,a,f8.2,a,/,2(10x,a,i3,/))
      else if(nprint.eq.2)then
        write(12,50)
     &  'Total elapsed time ..............',day,'days',hr,'hrs =',
     &  ttime,'sec',
     &  'Storage Period ...............................',ii,
     &  'Elapsed time from beginning of',
     &  '   storage period ....................',pday,'days',phr,'hrs'
   50 format(10x,a,f6.2,a,f8.2,a,f8.1,a,/,10x,a,i3,2(/,10x,a),f6.2,a,
     &f8.2,a,/)
      else if(nprint.eq.3)then
        write(12,60)
     &  'Total elapsed time ....................',day,'days',hr,'hrs',
     &  'End of last storage period ..................',ii
   60   format(10x,a,f6.2,a,f8.2,a,/,10x,a,i3,/)
      endif

      if(shape.eq.1)then
        if(symx.eq.1.and.symy.eq.0)then
          write(12,70)
   70 format(10x,'Note: Distribution is symmetrical in the x direction;'
     &    /,13x,'results are shown for half-thickness only.'/)
        else if(symx.eq.1.and.symy.eq.1)then
          write(12,80)
   80 format(10x,'Note: Distribution is symmetrical in both ',
     &    'x and y dimensions;',/,16x,'results are shown for ',
     &    'half-thicknesses only.'/)
        endif
      endif

      if(shape.eq.2)then
        if(cyn.eq.0.and.symy.eq.1)then
          write(12,90)
   90     format(10x,'Note: Distribution is symmetrical in both ',
     &    'radial and y dimensions;',/,16x,'results are shown for ',
     &    'half-diameter and half-thickness only.'/)
        else if(cyn.eq.1.and.symy.eq.1)then
          write(12,100)
  100     format(10x,'Note: Distribution is symmetrical in the y dir',
     &    'ection;'/,16x,'results are shown for half-thickness only.'/)
        endif
      endif

      hh22='DISTRIBUTION HISTORY'
      if(model.eq.2)then
        hh11='      TEMPERATURE (C)       '
      else
        hh11='TEMPERATURE (C) & QUALITY (%)'
      endif
      write(12,110)hh11,hh22
  110 format(/,29x,a,/,33x,a,/,29x,29('-'),/)
```

Table C.2 (cont'd).

```
      write(12,120)
  120 format(37x,'y-position (m)',/,11x,'x-position (m)|')
      if(iypl.eq.3)then
        nstep = 3
      else
        nstep = 5
      endif
      do i = 1,nstep
        abc(i)=(i-1)*stepy*dy
      enddo
      if(iypl.eq.3)then
        write(12,130)abc(1),abc(2),abc(3)
        write(12,150)
      else
        write(12,140)abc(1),abc(2),abc(3),abc(4),abc(5)
        write(12,160)
      endif
  130 format(25x,'|'3(f8.4))
  140 format(25x,'|'5(f8.4))
  150 format(10x,42('='))
  160 format(10x,58('='))

      if(model.ge.3)then
        c8=100.0d0/(86400.0d0*q0)
        do i = 1,ixpl
          do j = 1,iypl
            if(qual(i,j).lt.0.0d0)dead=1
          enddo
        enddo
      endif

      stepx = (ixpl-1)/4
      do i = 1,nstep
        do j = 1,nstep
          abc(j) = t((i-1)*stepx+1,(j-1)*stepy+1,3)-273.15d0
          if(model.ge.3)then
            abc(j+nstep)=qual((i-1)*stepx+1,(j-1)*stepy+1)*c8
          endif
        enddo
        if(nstep.eq.3)then
          write(12,170)(i-1)*stepx*dx,abc(1),abc(2),abc(3)
        else
          write(12,180)(i-1)*stepx*dx,abc(1),abc(2),abc(3),abc(4),
     &    abc(5)
        endif
  170   format(16x,f8.4,' |',3(f7.2,1x),'C')
  180   format(16x,f8.4,' |',5(f7.2,1x),'C')
        if(model.eq.3)then
          if(nstep.eq.3)then
            write(12,190)abc(4),abc(5),abc(6)
          else
            write(12,200)abc(6),abc(7),abc(8),abc(9),abc(10)
          endif
  190     format(25x,'|',3(f7.2,1x),'%')
  200     format(25x,'|',6(f7.2,1x),'%')
        endif
```

Table C.2 (cont'd).

```
      enddo
      if(iypl.eq.3)then
        write(12,150)
      else
        write(12,160)
      endif

      write(12,210)tavg-273.15
  210 format(///,10x,'Average temperature (C) = ',f7.2)
      if(model.ge.3)then
        write(12,220)qavg*100.0d0/(86400.0d0*q0)
  220   format(/,10x,'Average quality (% ref. quality) = ',f7.2)
        if(model.eq.4)write(12,230)(vq)**0.50d0*c8
  230   format(/,10x,'Average st. dev. of quality ',
     &  '(% of ref. quality) = ',e7.1)
      endif

      if(dead.eq.1)then
        write(12,240)
  240   format(/,10x,'Shelf-life was exceeded at some point on body.')
      endif
      if(nprint.eq.3)then
        write(12,250)'End of date file.'
  250   format(/,10x,a,i3,a)
        close(unit=12)
      endif
      return
      end
```

APPENDIX D

APPENDIX D

SURFACE HEAT TRANSFER COEFFICIENT ESTIMATION PROGRAM

The program used to estimate surface heat transfer coefficients using the sequential regularization procedure, discussed in Chapter 3, is presented here. An outline of the program is given in Table D.1, and the listing for the program, written in Fortran 77 for a Vax 11/750 is given in Table D.2.

Table D.1 Description of Surface Heat Transfer Coefficient Estimation Program.

Subroutine Title	Description
PROGRAM FREEZE	Main program; contains program menu.
SUBROUTINE PROPER	See Table B.1.
DOUBLE PRECISION FUNCTION MOIST(X)	See Table B.1.
DOUBLE PRECISION FUNCTION DENS(X)	See Table B.1.
DOUBLE PRECISION FUNCTION KI(X)	See Table B.1.
DOUBLE PRECISION FUNCTION CONDUC(X)	See Table B.1.
DOUBLE PRECISION FUNCTION SPHEAT(X)	See Table B.1.
SUBROUTINE CONSPR	See Table B.1.
SUBROUTINE INTEGR	See Table B.1.
BLOCK DATA CONST	See Table B.1.
SUBROUTINE INPUT1	Allows interactive input of ambient and internal product temperature measurements, and product geometry. Writes output to data file.
SUBROUTINE SOLN	Computes surface heat flux and surface heat transfer coefficients as a function of time using the sequential requlariza tion procedure. Calls output subroutine.
SUBROUTINE COEFF	Determines matrix coefficients used in first sweep in ADI finite difference algorithm.
SUBROUTINE PFIND	See Table B.1.
SUBROUTINE OUTPUT	Writes estimated surface heat flux and surface heat transfer coefficients to output data file.
SUBROUTINE SIMUL	Matrix inversion subroutine.
SUBROUTINE RAND	Generates normally distributed random numbers used in determining optimal parameters.

Table D.2 Computer Code Listing for Surface Heat Transfer Coefficient Estimation Program.

```
      PROGRAM IHCP1D
c**********************************************************************
c**********************************************************************

c        Surface Heat Transfer Coefficient Estimation Program

c                                by

c                           Elaine Scott

c                                1986

c**********************************************************************

c   This program estimates the surface heat transfer coefficient
c as a function of time during frozen food storage.  The program
c assumes one dimensional heat transfer, with a convective boun-
c dary condition at x = 0, and an insulated boundary at x = L.

c   Input property parameters include unfrozen product density,
c thermal conductivity and specific heat.  The initial freezing
c temperature or molecular weight of solids also is required to
c predict these values for the frozen food product.

c   Boundary conditions are found from internal temperature meas-
c urements and known ambient temperatures.  The input data file
c includes product geometry; number of nodes; number of time steps
c for both data points and for finite difference calulations;
c number of thermocouples and location; and ambient temperature and
c thermocouple reading at each time step.

c**********************************************************************



      parameter(maxt=250,maxm=41,maxd=101,maxc=51,maxs=201)

      integer model,ynavg

      double precision wf0,ms,dp,kp,cp,t0,
     &ds,
     &th,tl,avgd,avgk,avgc

      character title*40,ttlfil*10,filyn1*1,filyn2*1,filyn*1,fildat*16,
     &inpdat*16

      common/mod/model,
     &/ttl/title,ttlfil,
```

Table D.2 (cont'd).

```
     &/prop/wf0,ms,dp,kp,cp,t0,
     &/d/ds,
     &/pavg/th,tl,avgd,avgk,avgc,ynavg


      write(5,1000)
 1000 format('1',72('*'),/,'0',t20,'Heat Transfer Coefficient ',
     &'Estimation',/,'0',t35,'by',/,'0',t30,'Elaine Scott',/,'0',t24,
     &'Michigan State',' University',/,'0',t32,'May 1986',/,'0',72('*'))
      WRITE(5,100)
  100 FORMAT('0','Program Menu:',//,' ',' 1. Product properties (<0C)'
     &,/,' ','  2. Estimate heat transfer coefficient h',
     &/,'0','  h = Surface heat trans. coef.',
     &//,' ','Selection? ')
      READ(5,10)model
   10 FORMAT(I1)
      write(5,300)
  300 format(' ',/,' ','Key word for data files; 6 Characters: ')
      READ(5,20)TTLFIL
   20 FORMAT(A)
      if(model.eq.1)then
        filyn1 = 'n'
      else
        write(5,400)
  400 format(' ',/,' ','Are product properties approximations',/,' ',2x,
     &'with temperature stored on file? (y/n) ')
        read(5,20)filyn1
        write(5,500)
  500 format(' ',/,' ','Are input initial and boundary conditions',/,' '
     &,2x,'and geometrical dimensions stored on file? (y/n) ')
        read(5,20)filyn2
      endif
      if(filyn1.eq.'n'.or.filyn1.eq.'N')then
        call proper
        CALL CONSPR
      endif
      if(model.ne.1)then
        if(filyn2.eq.'n'.or.filyn2.eq.'N')then
          call input1
        endif
        call soln
      endif
      end




      SUBROUTINE PROPER

c See Appendix B.
```

Table D.2 (cont'd)

```
      DOUBLE PRECISION FUNCTION MOIST(X)
c See Appendix B.


      DOUBLE PRECISION FUNCTION DENS(X)
c See Appendix B.


      DOUBLE PRECISION FUNCTION KI(X)
c See Appendix B.


      DOUBLE PRECISION FUNCTION CONDUC(X)
c See Appendix B.


      DOUBLE PRECISION FUNCTION SPHEAT(X)
c See Appendix B.


      SUBROUTINE CONSPR
c See Appendix B.


      SUBROUTINE INTEGR(thi,tlow,avgdp,avgkp,avgcp,ncase)
c See Appendix B.
```

Table D.2 (cont'd)

```
      BLOCK DATA CONST

c See Appendix B.
```

```
      SUBROUTINE INPUT1

c   This subroutine provides the input for the boundary condi-
c tions on the product for the case where the ambient temper-
c ature and surface heat tranfer coeffient are known and assumed
c to be constant over a given storage period.

c   Input varibles include, initial product temperature, sym-
c metry of boundary conditions, number of constant temperature
c storage periods, length of storage period, and surface heat
c transfer coefficient.


      parameter(maxt = 250,maxtc = 10)

      integer shape,mpli,ntime,ntc,ndt

      double precision delt,tamb(maxt),tc(maxt,maxtc),xtc(maxtc),L

      character yn*1,title*40,ttlfil*10,fildat*16,inpdat*16

      common /ttl/title,ttlfil,
     &/datfil/fildat,inpdat

      save


      write(6,1)
    1 format(' ',/,' ','Product: ')
      READ(5,2)TITLE
    2 format(a)

c input geometry and size

5     write(6,10)

   10 format('1','Geometry',/,' ',8('-'),/'0','Enter product geometry: '
     &,/,' ',5x,'1 = slab',/,' ',5x,'2 = cylinder',/,' ',5x,
     &'3 = sphere')
      read*,shape
      if(shape.eq.1)then
        write(6,20)
   20 format(' ',/,' ','Enter dimensions for slab;',/,' ',5x,
     &'thickness in direction of heat transfer (m) : ')
        read*,l
      else
        if(shape.eq.2)then
          write(6,30)
```

Table D.2 (cont'd)

```
   30 format(' ',/,' ','Enter dimensions for cylinder',/,
     &' ',5x,'radius (m) : ')
          read *,l
        else
          write(6,40)
   40 format(' ',/,' ','Enter dimensions for sphere (m)',/,
     &' ',5x,'radius (m) : ')
          read *,l
        endif
      endif
      write(5,50)
   50 format(' ',/,' ','Enter total number of nodes in F.D. ',
     &'calculations : ')
      read*,mpli
      write(5,60)
   60 format('0','Storage Conditions',/,' ',18('-'))
      write(6,70)
   70 format('0',/,' ','Enter total number of temperature ',
     &'measurements ',/,' ','(include time = 0) : ')
      read*,ntime
      write(6,80)
   80 format(' ',/,' ','Enter time increment between each ',
     &'temperature measurement (sec) : ')
      read*,delt
      write(6,85)
   85 format(' ',/,' ','Enter number of time steps per temperature',
     &' measurement interval ',/' ','(for F.D. calculations) : ')
      read*,ndt
      write(6,90)
   90 format(' ',/,' ','Enter total number of thermocouples : ')
      read*,ntc
      write(6,100)
  100 format(' ',/,' ','Are these values correct? (y/n) ')
      read(5,110)yn
  110 format(a)
      if(yn.ne.'y'.and.yn.ne.'Y')goto 5
      write(6,120)
  120 format(' ',/,' ','Enter location of each thermocouple.'/)
      do itc = 1,ntc
        write(6,130)itc
  130   format(' ','Location of T.C. No. ',i2,'(m) : ')
        read*,xtc(itc)
      enddo
      write(6,140)
  140 format(' ',/,' ','Enter ambient temp. and each thermocouple ',
     &'measurement (C) ',/,' ','for every time step.',/)
      do it = 1,ntime
        write(6,150)it-1
  150 format(' ','Time step:',i3,5x,'tamb= ')
        read*,tamb(it)
        do itc = 1,ntc
          write(6,160)itc
  160     format(' ','T.C.(',i2,')= ')
          read*,tc(it,itc)
        enddo
      enddo
```

Table D.2 (cont'd)

```
      do i = 1,ntime
        if(tamb(i).ne.0.)then
          tamb(i) = tamb(i)+273.150d0
        endif
        do itc = 1,ntc
          if(tc(i,itc).ne.0.)then
            tc(i,itc) = tc(i,itc)+273.150d0
          endif
        enddo
      enddo
      write(inpdat,170)ttlfil,'ihcpinp.dat'
  170 format(' ',a,a)
      open(unit=12,name=inpdat(1:16),type='new',carriagecontrol='list')
      write(12,*)title
      write(12,*)shape,L,mpli
      write(12,*)ntime,delt,ndt
      write(12,*)ntc
      do i = 1,ntc
        write(12,*)xtc(i)
      enddo
      do i = 1,ntime
        write(12,*)tamb(i)
        do itc = 1,ntc
          write(12,*)tc(i,itc)
        enddo
      enddo
      close(unit=12)
      return
      end




      SUBROUTINE soln


      parameter(maxd = 101, maxc = 51, maxs = 201, iterat = 1)
      parameter(maxm=21,maxtc=6,maxr=15,maxrpl=16,maxt=250,r=8.3140d0)
      parameter(maxrdt = 500)

      integer shape,mpli,ntime,ntc,rfts,mx,ynavg,mistc,irfts,
     &rfti,ntci,yn

      double precision a(maxm),b(maxm),c(maxm),da(maxm),db(maxm),
     &dc(maxm),dz(maxm),dt(maxm),cc(maxm),ddt(maxm),ddx(maxm),
     &t(maxm,maxrdt),x(maxm,maxrdt),q1(maxrpl),q2(maxt),q1star,
     &xk(maxtc,maxr),ds,htc,xmat(maxr,maxr,maxtc),xtx(maxr,maxrpl),
     &h0(maxr,maxr),h1(maxr,maxr),h2(maxr,maxr),sum,sum2,
     &tt(maxr,maxtc),xx(maxr,maxtc),xmatt(maxr,maxr,maxtc),qq(maxr),
     &sumh1,sumh2,tstar(maxm,maxrdt),ratiox,ttime,sumhtc,htca,shtc

c  Declare variables in common blocks

      double precision wf0,ms,dp,kp,cp,t0,
     &l,dxi,
```

Table D.2 (cont'd)

```
      &DENST(maxd),DENSC(maxd),CONDT(maxc),CONDC(maxc),SPHT(maxs),
      &SPHC(maxs),
      &delt,
      &tamb(maxt),tc(maxt,maxtc),xtc(maxtc),
      &tl,th,avgd,avgk,avgc,
      &alpha,w0,wl,w2
      &sumtl,sumt2,ctl,ct2,tal,ta2

       character title*40,ttlfil*10,fildat*16,inpdat*16

       common/ttl/title,ttlfil,
      &/geom/shape,l,dxi,
      &/datfil/fildat,inpdat,
      &/NCONSTP/NSD,NSC,NSS,
      &/CONSTP/DENST,DENSC,CONDT,CONDC,SPHT,SPHC,
      &/prop/wf0,ms,dp,kp,cp,t0,/d/ds,
      &/meas/delt,ntime,ntc,ndt
      &/meas2/tamb,tc,xtc,
      &/pavg/tl,th,avgd,avgk,avgc,ynavg,
      &/ihcp/irfts,alpha,w0,wl,w2

       save

c  Read in boundary and initial conditions

       write(inpdat,600)ttlfil,'.dat'
  600  format(' ',a,a)
       open(unit=12,name=inpdat(1:16),type='old',carriagecontrol='list')

       read(12,*)title
       read(12,*)shape,l,mpli
       read(12,*)ntime,delt,ndt
       read(12,*)ntc
         read(12,*)xtc(1),xtc(2),xtc(3)
         read(12,*)xtc(4),xtc(5),xtc(6)
       do i = 1,ntime
       read(12,*)tamb(i),tc(i,1),tc(i,2),tc(i,3),tc(i,4),tc(i,5),tc(i,6)
         if(tamb(i).ne.0.0)then
           tamb(i) = tamb(i)+273.15d0
         endif
         sum = 0.0d0
         sumx = 0.0d0
         do j = 1,ntc
           if(tc(i,j).ne.0.0)then
             tc(i,j) = tc(i,j)+273.15d0
           endif
             sum = sum +tc(i,j)
           sumx = sumx+ xtc(j)
         enddo
       enddo

       close(unit=12)


       write(5,11)
    11 Format(' ','Average ambient temperature values? (1=y,0=n) ')
```

Table D.2 (cont'd)

```
      read(5,*)yn
      if(yn.eq.1)then
        sumt1 = 0
        sumt2 = 0
        ct1 = 0
        ct2 = 0
        do i = 1,ntime
          if(tamb(i).gt.258.15)then
            sumt1 = sumt1+tamb(i)
            ct1 = ct1+1
          else
            sumt2 = sumt2+tamb(i)
            ct2 = ct2+1
          endif
        enddo
        ta1 = sumt1/ct1
        ta2 = sumt2/ct2
        do i = 1,ntime
          if(tamb(i).gt.258.15)then
            tamb(i) = ta1
          else
            tamb(i) = ta2
          endif
        enddo
      endif

c  Read in constant property assumptions

      WRITE(FILDAT,310)'1d_t1cPRP.DAT'
  310 FORMAT(' ',a)
      OPEN(UNIT=12,NAME=FILDAT(1:16),TYPE='OLD',CARRIAGECONTROL='LIST')
      READ(12,*)WF0,T0,MS
      READ(12,*)DP,KP,CP
      READ(12,*)NSD,NSC,NSS
      READ(12,*)tl,th,avgd,avgk,avgc,ynavg
      DO I=1,NSD
        READ(12,*)DENST(I),DENSC(I)
      ENDDO
      DO I=1,NSC
        READ(12,*)CONDT(I),CONDC(I)
      ENDDO
      DO I=1,NSS
        READ(12,*)SPHT(I),SPHC(I)
      ENDDO
      CLOSE(UNIT=12)

c  Check for bad T.C.

      do i = 1,ntime
        mistc = 0
        sum = 0.0d0
        do itc = 1,ntc
          sum = sum + tc(i,itc)
          if(tc(i,itc).eq.0.0)then
            mistc = mistc + 1
          endif
```

Table D.2 (cont'd)

```
        enddo

        do itc = 1,ntc
          if(tc(i,itc).eq.0.0)then
            tc(i,itc) = sum/(ntc-mistc)
          endif
        enddo

        if(tamb(i).eq.0.0)then
          if(i.eq.1)then
            ii = 1
    4       if(tamb(ii+1).ne.0.0)then
              do jj = 1,ii
                tamb(jj) = tamb(ii+1)
              enddo
            else
              ii = ii+1
              go to 4
            endif
          else
            ii = i
    6       if(tamb(ii+1).ne.0.0)then
              tamb(i) = (tamb(i-1)+tamb(ii+1))/2.0d0
            else
              ii = ii+1
              go to 6
            endif
          endif
        endif
      enddo


c  Regularization components

      write(6,3)
    3 format(' ',/,' ','Enter alpha,w0,w1,w2: ',$)
      read*,alpha,w0,w1,w2

      write(6,2)
    2 format(' ',/,' ','Enter number of future time steps= ',$)
      read*,irfts
      rfts = irfts
      dti = delt/ndt
      mi = mpli-1
      dxi = L/mi
      ndtt = rfts*ndt
      ntci = ntc

      do i = 1,rfts
        q1(i) = 0.0d0
        do j = 1,rfts
          if(i.ne.j)then
            h0(i,j) = 0.0d0
          else
            h0(i,j) = 1.0d0
          endif
```

Table D.2 (cont'd)

```
          h1(i,j) = 0.0d0
          h2(i,j) = 0.0d0
        enddo
        if(i.lt.rfts)then
          h1(i,i) = -1.0d0
          h1(i,i+1) = 1.0d0
        endif
        if(i.lt.rfts-1)then
          h2(i,i) = 1.0d0
          h2(i,i+1) = -2.0d0
          h2(i,i+2) = 1.0d0
        endif
      enddo
      htc = 0.0d0

c  Assume value for q1*: Use q1* = 0.0

      q1star = 0.0d0

      call output(0,q1,htc,1)

c finite difference solution

      do 100 it = 1,ntime-1
        q2(it) = 0.0d0

c _____________________________________________________________

c  Assume value for q1*: Use q1* = q1(1)

      if(it.gt.1)then
        call output(1,q1,htc,it,0.)
      endif

c  Initialize Tstar and set sensitivity coefficients equal to xero.

      if(it.eq.1)then
        sum = 0.d0
        ntci = ntc
        do itc = 1,ntc
          if(tc(1,itc).eq.0.0d0.and.ntc.ne.1)then
            ntci = ntc-1
          else
            sum = sum+tc(1,itc)
          endif
        enddo
        do i = 1,mpli
          Tstar(i,1) = sum/ntci
          t(i,1) = sum/ntci
          x(i,1) = 0.0d0
        enddo
        if(tamb(1).eq.0.0)tamb(1) = tamb(2)
      endif
```

Table D.2 (cont'd)

```
c thomas algorithm

c find coefficients for thomas algorithm


      call coeff(t,dti,mpli,a,b,c,da,db,dc)

      do ir = 1,rfts
        do idt = 1,ndt
          indt = ndt*(ir-1)+idt
          if(ir.eq.1)then
            dt(1) = db(1)*tstar(1,indt)+dc(1)*tstar(2,indt)
     &       -qlstar*0.50d0
          else
            dt(1) = db(1)*tstar(1,indt)+dc(1)*tstar(2,indt)-qlstar
          endif
          if(idt.eq.1)then
            dz(1) = db(1)*x(1,indt)+dc(1)*x(2,indt)-0.5d0
          else
            dz(1) = db(1)*x(1,indt)+dc(1)*x(2,indt)-1.0d0
         endif
          do i = 2,mpli-1
            dt(i) = da(i)*tstar(i-1,indt)+db(i)*tstar(i,indt)
     &       +dc(i)*tstar(i+1,indt)
            dz(i) = da(i)*x(i-1,indt)+db(i)*x(i,indt)
     &       +dc(i)*x(i+1,indt)
          enddo
          if(ir.eq.1)then
            dt(mpli) = da(mpli)*tstar(mpli-1,indt)+db(mpli)
     &       *tstar(mpli,indt)-0.5d0*(q2(it)+q2(it+1))
          else
            dt(mpli) = da(mpli)*tstar(mpli-1,indt)+db(mpli)
     &       *tstar(mpli,indt)-q2(it+1)
          endif
          dz(mpli) = da(mpli)*x(mpli-1,indt)+db(mpli)*x(mpli,indt)


          cc(1) = c(1)/b(1)
          ddt(1) = dt(1)/b(1)
          ddx(1) = dz(1)/b(1)
          do k = 2,mpli
            kk = k-1
            cc(k) = c(k)/(b(k)-a(k)*cc(kk))
            ddt(k) = (dt(k)-a(k)*ddt(kk))/(b(k)-a(k)*cc(kk))
            ddx(k) = (dz(k)-a(k)*ddx(kk))/(b(k)-a(k)*cc(kk))
          enddo
          tstar(mpli,indt+1) = ddt(mpli)
          x(mpli,indt+1) = ddx(mpli)
          do k = 2,mpli
            kk = mpli-k+1
            tstar(kk,indt+1) = ddt(kk)-cc(kk)*tstar(kk+1,indt+1)
            x(kk,indt+1) = ddx(kk)-cc(kk)*x(kk+1,indt+1)
          enddo
        enddo
      enddo
```

Table D.2 (cont'd)

```
      do itc = 1,ntc
        do i = 2,mpli
          if(xtc(itc).lt.(i-1)*dxi)then
            mx = i
            ratiox = (xtc(itc)-(i-2)*dxi)/dxi
            go to 5
          endif
        enddo
    5   do ir = 2,rfts+1
          irr = (ir-1)*ndt+1
          tt(ir-1,itc) = tstar(mx-1,irr)+
     &    ratiox*(tstar(mx,irr)-tstar(mx-1,irr))
          xx(ir,itc) = x(mx-1,irr)+ratiox*(x(mx,irr)-x(mx-1,irr))
        enddo
      enddo

c-------------------------------------------------------------

c  Calculate q from measured temperatures

      do itc = 1,ntc
        do ii = 1,rfts
          do jj = 1,ii
            xmat(ii,jj,itc) = xx(ii-jj+2,itc)
          enddo
        enddo
      enddo
      do itc = 1,ntc
        do ii = 1,rfts
          do jj = 1,rfts
            xmatt(ii,jj,itc) = xmat(jj,ii,itc)
          enddo
        enddo
      enddo
      do ii = 1,rfts
          sum2 = 0.0d0
        do jj = 1,rfts+1
          sum = 0.0d0
          sumh1 = 0.0d0
          sumh2 = 0.0d0
          do itc = 1,ntc
            do j = 1,rfts
              if(jj.le.rfts)then
                sum = sum + xmatt(ii,j,itc)*xmat(j,jj,itc)
              else
                sum2 = sum2 + (tc(it+j,itc)-tt(j,itc))
     &                 *xmatt(ii,j,itc)
              endif
              if(itc.eq.1)then
                sumh1 = sumh1 + h1(ii,j)*h1(jj,j)
                sumh2 = sumh2 + h2(ii,j)*h2(jj,j)
              endif
            enddo
          enddo
          if(jj.le.rfts)then
            xtx(ii,jj) = sum + alpha*(w0*h0(ii,jj)+w1*sumh1+w2*sumh2)
```

Table D.2 (cont'd)

```
          else
            xtx(ii,jj) = sum2
          endif
        enddo
      enddo

c  Call Gauss elimination subroutine

      call simul(xtx,qq,rfts,1)

      do ir = 1,rfts
        ql(ir+1) = qlstar+qq(ir)
      enddo

c  Knowing ql, reevaluate d(1) and repeat back substition to find
c     t.

      do ir = 1,ndt
        if(it.eq.1.or.ir.ne.1)then
          dt(1) = db(1)*t(1,ir)+dc(1)*t(2,ir)-ql(2)
        else
          if(ir.eq.1)then
            dt(1) = db(1)*t(1,ir)+dc(1)*t(2,ir)-0.5d0*ql(1)
     &       -0.5d0*ql(2)
          endif
        endif
        do i = 2,mi
          dt(i) = da(i)*t(i-1,ir)+db(i)*t(i,ir)+dc(i)*t(i+1,ir)
        enddo
        if(ir.eq.1)then
          dt(mpli) = da(mpli)*t(mi,ir)+db(mpli)*t(mpli,ir)
     &     -0.5d0*(q2(it)+q2(it+1))
        else
          dt(mpli) = da(mpli)*t(mi,ir)+db(mpli)*t(mpli,ir)
     &     -q2(it+1)
        endif
        ddt(1)=dt(1)/b(1)
        do k=2,mpli
          kk=k-1
          ddt(k)=(dt(k)-a(k)*ddt(kk))/(b(k)-a(k)*cc(kk))
        enddo
        t(mpli,1+ir)=ddt(mpli)
        do k=2,mpli
          kk=mi-k+2
          t(kk,1+ir)=ddt(kk)-cc(kk)*t(kk+1,1+ir)
        enddo
      enddo

c_____________________________________________________________

c  Estimate the heat transfer coefficient from q and t.

      sum = 0.0d0
      do i = 1,ndt+1
        if(i.eq.1.or.i.eq.ndt+1)then
          sum = sum+0.50d0*t(1,i)
```

Table D.2 (cont'd)

```
        else
          sum = sum+t(1,i)
        endif
      enddo
      sum = sum/ndt


      htc = q1(2)/((tamb(it+1)+tamb(it))*0.50d0-sum)


c end of finite difference calculations

c ********************************************************


c  Initialize t and q1

      q1(1) = q1(2)
      if(it.eq.ntime-1)call output(1,q1,htc,ntime)
      do i = 1,mpli
        t(i,1) = t(i,ndt+1)
        tstar(i,1) = t(i,ndt+1)
      enddo

      if(it.ge.ntime-irfts)then
        rfts = rfts-1
      endif

  100 continue

c printout

      call output(2,q1,htc,it)

      return
      end




      SUBROUTINE COEFF(t,dti,mpli,a,b,c,da,db,dc)


      parameter(maxm=21,maxp=10,maxd=101,maxc=51,maxs=201,maxr=15)
      parameter(maxrdt = 500)

      integer shape,mi,mpli,ii,nsd,nsc,nss,ynavg

      double precision beta,nu,omega,gama,
     &aar,ar(maxm),ar1(maxm),area,avg1,avg2,ck(maxm),
     &csd(maxm,2),a(maxm),b(maxm),c(maxm),da(maxm),db(maxm),dc(maxm),
     &t(maxm,maxrdt),z(maxm,maxrdt),dti,pi,dxx,dtt

c Declare variables in common statements
```

Table D.2 (cont'd)

```
      double precision l,dxi,
     &wf0,ms,dp,kp,cp,t0,
     &DENST(maxd),DENSC(maxd),CONDT(maxc),CONDC(maxc),SPHT(maxs),
     &SPHC(maxs),ds,
     &th,tl,avgd,avgk,avgc

      common/geom/shape,l,dxi,
     &/prop/wf0,ms,dp,kp,cp,t0,
     &/CONSTP/DENST,DENSC,CONDT,CONDC,SPHT,SPHC,
     &/NCONSTP/NSD,NSC,NSS,
     &/d/ds,
     &/pavg/tl,th,avgd,avgk,avgc,ynavg


      pi = dacos(-1.0d0)

c weighting functions for finite difference method

c modified crank-nicolson method

c weight. coeff. for d2t/dx2
c      for time t:
      beta=0.50d0
c      for time t+1:
      nu=0.50d0

c weight. coeff. for dt/dt
c      for time t:
      omega=-1.0d0
c      for time t+1:
      gama=1.0d0



      mi = mpli-1
      dxx=1.0d0/dxi
      dtt=1.0d0/(2.0d0*dti)
      if(shape.eq.2)then
        aar=2.0d0*pi
      else
        if(shape.eq.3)then
          aar=4.0d0*pi
        endif
      endif

      do 10 i=1,mpli

c slab

      if(shape.eq.1)then
        ar(i)=1.0d0
        arl(i)=1.0d0
      else

c cylinder
```

Table D.2 (cont'd)

```
        if(shape.eq.2)then
          ar(i)=aar*(i-1)*dxi
          arl(i)=ar(i)+aar*dxi/2.0d0
        else

c sphere

          ar(i)=aar*((i-1)*dxi)**2.0d0
          arl(i)=aar*((i-1)*dxi+dxi/2.0d0)**2.0d0
        endif
      endif

10    continue

      CALL PFIND(T,Mi,CK,CSD,dti,dxi)


c ***********************************************************

c 1st boundary point

      AVG1 = (AR(1)+ARl(1))*0.50d0

      a(1) = 0.0d0
      c(1) = nu*dxx*CK(1)*arl(1)
      b(1) = -gama*CSD(1,1)*avgl-c(1)

      da(1) = 0.0d0
      dc(1) = -beta*dxx*CK(1)*arl(1)
      db(1) = omega*CSD(1,1)*avgl-dc(1)



c ***********************************************************

c interior points

      do i=2,mi
        AVG1 = (AR(I)+ARl(I))*0.50d0
        AVG2 = (AR(I)+ARl(I-1))*0.50d0

        a(i) = nu*dxx*CK(I-1)*arl(i-1)
        c(i) = nu*dxx*CK(I)*arl(i)
        b(i) = -gama*(CSD(I,1)*avg2+CSD(I,2)*avgl)-a(i)-c(i)

        da(i) = -beta*dxx*CK(I-1)*arl(i-1)
        dc(i) = -beta*dxx*CK(I)*arl(i)
        db(i) = omega*(CSD(I,1)*avg2+CSD(I,2)*avgl)-da(i)-dc(i)

      enddo

c ***********************************************************

c 2nd boundary point

      AVG2 = (AR(mpli)+ARl(Mi))*0.50d0
```

Table D.2 (cont'd)

```
      c(mpli) = 0.0d0
      a(mpli) = nu*dxx*CK(Mi)*ar1(mi)
      b(mpli) = -gama*CSD(mpli,2)*avg2-a(mpli)

      dc(mpli) = 0.0d0
      da(mpli) = -beta*dxx*CK(Mi)*ar1(mi)
      db(mpli) = omega*CSD(mpli,2)*avg2-da(mpli)


      return
      end




      SUBROUTINE PFIND(T,Mi,CK,CSPD,dti,dx)

C See Appendix A




      SUBROUTINE OUTPUT(nprint,ql,htc,it)



      parameter(maxp=10,maxm=21,maxr=15,maxt=250,maxtc=6)

      integer shape,ntime,ntc,irfts

      double precision ql(maxr),htc,ttc(6),ttamb

c Declare variables in common statements

      double precision wf0,ms,dp,kp,cp,t0,
     &l,dxi,
     &delt,
     &tamb(maxt),tc(maxt,maxtc),xtc(maxtc),
     &alpha,w0,w1,w2

      character title*40,ttlfil*10,outfil*20,hh11*30,hh22*21

      common/ttl/title,ttlfil,
     &/prop/wf0,ms,dp,kp,cp,t0,
     &/geom/shape,l,dxi,
     &/meas/delt,ntime,ntc,ndt
     &/meas2/tamb,tc,xtc,
     &/ihcp/irfts,alpha,w0,w1,w2

C  NPRINT = 0 if printing input parameters and headings.
C  NPRINT = 1 if printing estimated heat flux and heat
C   transfer coefficient.
```

Table D.2 (cont'd)

```
C  NPRINT = 2 if printing end of file.

      IF(NPRINT.EQ.0)THEN
        GO TO 1100
      ELSE
        if(nprint.eq.1)then
          go to 1200
        else
          go to 1300
        endif
      endif
 1100 write(outfil,1000)ttlfil,'out.dat'
 1000 format(' ',a,a)
      open(unit=12,name=outfil(1:20),type='new',carriagecontrol='list')
      write(12,10)title
   10 format(///,3x,'Title: ',a20,/3x,'-----',//,14x,'Input Para',
     +'meters',/,14x,16('-')//)
      write(12,20)
   20 format(/'Unfrozen Product Properties',/)
      abcd=wf0*100.0d0
      write(12,30)abcd
   30 format(2x,'Moisture content (%).....',f6.2)
      abcd=t0-273.150d0
      write(12,40)abcd
   40 format(2x,'Initial freezing temperature (C).....',f6.2)
      write(12,50)ms
   50 format(2x,'Molecular weight of solids (kg/mole)...',f8.2)
      write(12,60)dp
   60 format(2x,'Unfrozen product density (kg/m^3).....',f8.2)
      write(12,70)kp
   70 format(2x,'Thermal conductivity (W/mK).....',f6.3)
      write(12,80)cp
   80 format(2x,'Specific heat (kJ/kgK).....',f7.3)
      sum = 0
      do i = 1,ntc
        sum = sum+tc(1,i)
      enddo
      abcd = sum/ntc-273.150d0
      write(12,90)abcd
   90 format(/'Initial Condition:',
     &      /2x,'Avg. Product temp. (C) at time=0 .....',f7.2)

c product geometry


      if(shape.eq.1)then
        write(12,100)l
  100 format( /'Slab Geometry:',
     +      /2x,'thickness (m) .....',f10.6)
      else
        if(shape.eq.2)then
          write(12,110)l
  110 format(/'Cyclindrical Geometry:',
     +      /2x,'radius (m) .....',f10.6)
      else
        write(12,120)l
```

Table D.2 (cont'd)

```
  120 format(/'Spherical Geometry:',
     +      /2x,'radius (m) ..........................',f10.6)
      endif
      endif

c input conditions

      write(12,130)ntime
  130 format(/2x,'Total no. of measurements ..............',i3)
      write(12,140)delt*(ntime-1)
  140 format(2x,'Total time (sec) .......................',f10.2)
      tothrs = delt*(ntime-1)/3600.0d0
      write(12,150)tothrs
  150 format(2x,'Total time (hrs) .......................',f7.3)
      write(12,160)delt
  160 format(2x,'Time step for IHCP solution (sec) ......',f10.2)
      write(12,162)delt/ndt
  162 format(2x,'Time step for FD solution (sec) ........',f10.2)
      write(12,168)dxi
  168 format(2x,'Position step for FD solution (sec) ....',f7.4)
      write(12,170)ntc
  170 format(2x,'Total no. of thermal couples ...........',i3)
      do itc = 1,ntc
        write(12,180)itc,xtc(itc)
  180 format(4x,'Location of T.C.(',i2,') .............',f7.4)
      enddo
      write(12,182)irfts,alpha,w0,w1,w2
  182 format(2x,'IHCP regularization solution parameters:',/,
     &       4x,'No. of future time steps ..............',i2,/,
     &       4x,'Regularization parameter (alpha) .....',e7.1,/,
     &       4x,'Weighting coefficient - 0th order ....',f5.3,/,
     &       4x,'Weighting coefficient - 1st order ....',f5.3,/,
     &       4x,'Weighting coefficient - 2nd order ....',f5.3)


      write(12,190)title
  190 format(/////'Title= ',a,/)

      hh22='TRANSFER COEFFICIENTS'
      hh11='ESTIMATED HEAT FLUXES AND HEAT'
      write(12,200)hh11,hh22
  200 format(/18x,a,/,22x,a,/,18x,29('-'),/)
      write(12,210)
  210 format('|No.| Tamb | TC(1) | TC(2) | TC(3) | TC(4) | TC(5) |',
     &    ' TC(6) |  q  |  h  |'/72('='))



c Printout T.C. temperature measurements

 1200 ttamb = tamb(it) - 273.150d0
      do itc = 1,ntc
        ttc(itc) = tc(it,itc) - 273.150d0
      enddo
      write(12,220)it,ttamb,ttc(1),ttc(2),ttc(3),
     &    ttc(4),ttc(5),ttc(6),q1(1),htc
```

Table D.2 (cont'd)

```
  220 format(i3,1x,f6.1,1x,6(f7.2,1x),f6.1,1x,f6.2)
      return




c Printout end line

 1300 write(12,230)
  230 format(72('-'))
      close(unit=12)
      return
      end




      SUBROUTINE SIMUL(ASTORE,X,N,INDIC)


      parameter(maxr = 15,maxr1 = 16)
      double precision AIJCK, DETER, EPS, PIVOT, PIVOTI
      double precision A(maxr,maxr1)
      double precision IROW(maxr),JCOL(maxr),JORD(maxr)
      double precision ASTORE(maxr,maxr1),X(maxr)
C***
C***  INITIALIZE PARAMETERS
C***
      MAX = N
      IF(INDIC.GE.0)    MAX = N+1
      EPS = 1.0D-20
C***
C***  STORE THE ARRAY SINCE INVERTED IN PLACE
C***
      DO J=1,MAX
        DO I=1,N
          A(I,J) = ASTORE(I,J)
        enddo
      enddo

C***  FOR DEBUGING PURPOSES, PRINTOUT THE "A" MATRIX



C***  BEGIN THE ELIMINATION PROCEDURE

   30 DETER = 1.0D0
      DO 130 K=1,N
        KM1 = K-1
C***
C***  SEARCH FOR THE PIVOT ELEMENT
C***
        PIVOT = 0.0D0
```

Table D.2 (cont'd)

```
        DO 80 I=1,N
          DO 70 J=1,N
C***
C***  SCAN IROW AND JCOL ARRAYS FOR INVALID PIVOT SUBSCRIPTS
C***
            IF(K.EQ.1)   GO TO 60
            DO ISCAN=1,KM1
              DO JSCAN=1,KM1
                IF(I.EQ.IROW(ISCAN))    GO TO 70
                IF(J.EQ.JCOL(JSCAN))    GO TO 70
              enddo
            enddo
   60       IF(DABS(A(I,J)).LE.DABS(PIVOT))   GO TO 70
            PIVOT = A(I,J)
            IROW(K) = I
            JCOL(K) = J
   70     CONTINUE
   80   CONTINUE
C***
C***  INSURE THE SELECTED PIVOT IS LARGER THAN EPS
C***
        IF(DABS(PIVOT).GT.EPS)GO TO 90
c       WRITE(5,2000) PIVOT
c 2000 FORMAT(' ','VALUE FOR PIVOT IS TOO SMALL, SUBROUTINE
TERMINATED.')
c        WRITE(1,2000) PIVOT
        RETURN
C***
C***  UPDATE THE DETERMINANT VALUE
C***
   90   IROWK = IROW(K)
        JCOLK = JCOL(K)
        DETER = DETER*PIVOT
C***
C***  NORMALIZE PIVOT ROW ELEMENT
C***
        PIVOTI = 1.0D0/PIVOT
        DO J=1,MAX
          A(IROWK,J) = A(IROWK,J)*PIVOTI
        enddo
C***
C***  CARRY-OUT ELIMINATION AND DEVELOP INVERSE
C***
        A(IROWK,JCOLK) = PIVOTI
        DO I=1,N
          AIJCK = -A(I,JCOLK)
          IF(I.EQ.IROWK)   GO TO 120
          A(I,JCOLK) = AIJCK*PIVOTI
          DO J=1,MAX
            IF(J.NE.JCOLK)   A(I,J) = A(I,J) + AIJCK*A(IROWK,J)
          enddo
  120   enddo
  130 CONTINUE
C***
C***  ORDER THE SOLUTION VALUES (IF ANY) AND CREATE JORD ARRAY
C***
```

Table D.2 (cont'd)

```
      DO 140 I=1,N
        IROWI = IROW(I)
        JCOLI = JCOL(I)
        JORD(IROWI) = JCOLI
        IF(INDIC.GE.0)    X(JCOLI) = SNGL(A(IROWI,MAX))
  140 CONTINUE
      RETURN
      END




      SUBROUTINE RAND(tc,ntime,ntc)

      Parameter(maxt=250,maxtc=6)

      integer ntime,ntc,seedi*4

      Double Precision tc(maxt,maxtc),u(maxt),a0,a1,b1,b2,r,e,t,ak

c  Subroutine to generate normally distributed random numbers.

      write(5,1)
    1 format(' ','Enter seed no. for this run:')
      read(5,*)seedi
      a0 = 2.30753
      a1 = 0.27061
      b1 = 0.99299
      b2 = 0.04481
      do itc = 1,ntc
        do i = 1,ntime
          u(i) = ran(seedi)
        enddo
        do j = 1,ntime
          r = u(j)
          if(r-0.5d0)10,10,20
   10     ak = 1.0d0
          go to 30
   20     ak = -1.0d0
          go to 30
   30     t = (log(1.0/(r*r)))**0.5
          e = t-(a0+a1*t)/(1.0+b1*t+b2*t*t)
          tc(j,itc) = tc(j,itc) + ak*e*0.27
        enddo
      enddo

      return
      end
```

APPENDIX E

APPENDIX E

RESULTS FROM SECOND AND THIRD EXPERIMENTAL TEST REPETITIONS

This appendix contains the results from the second and third experimental test repetitions. The experimental temperature measurements from Tests 1b,c, shown in Figures E.1b,c, were used in estimating the surface heat transfer coefficients, as shown in Figures E.2b,c, and in verifying the numerical model (Figures E.3b,c). Tests 2b,c and Tests 3b,c were also used in verifying the numerical model; results are given in Figures E.4b,c and Figures E.5b,c, respectively.

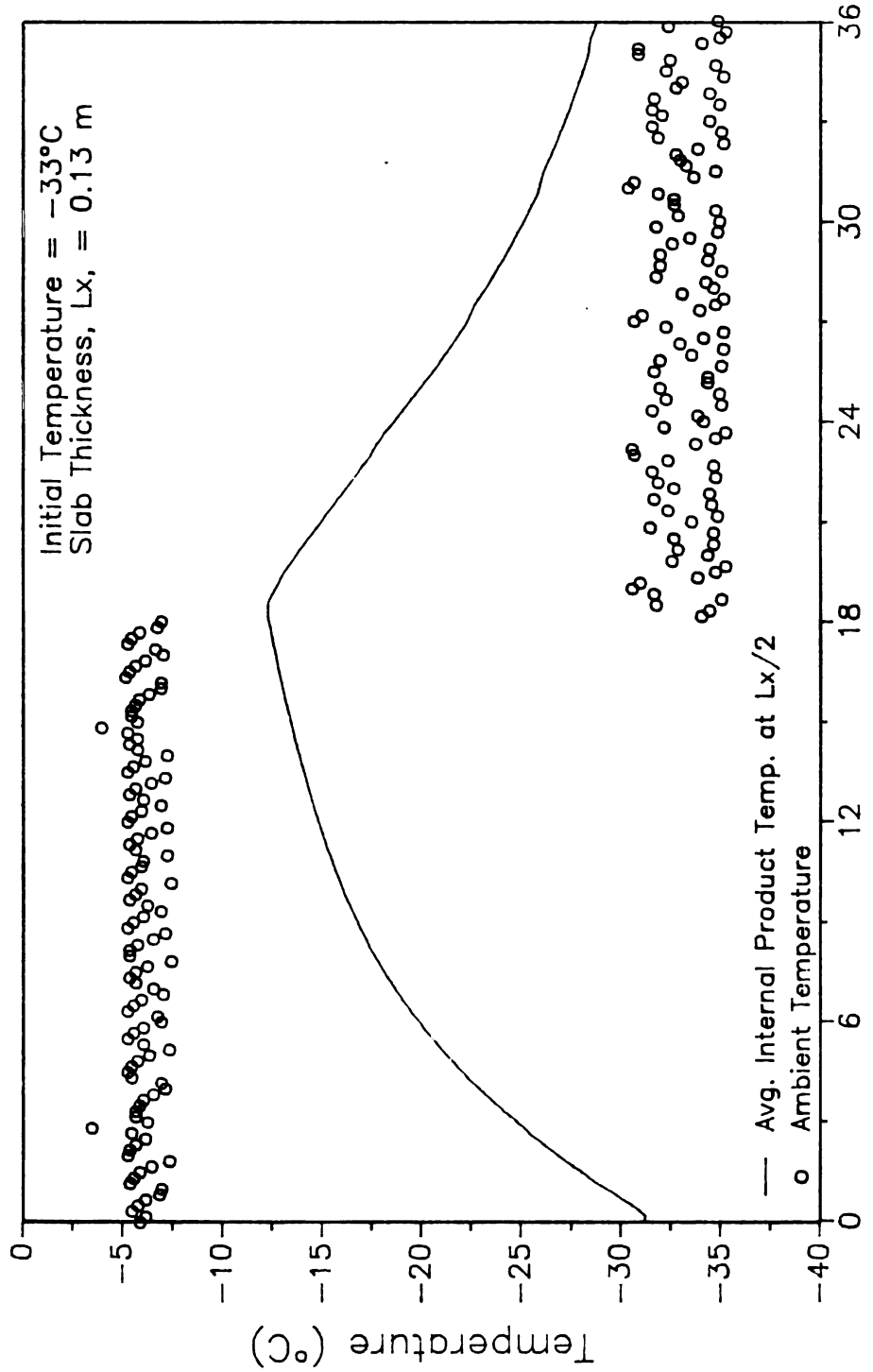


Figure E.1a Ambient and Average Internal Temperature of Karlsruhe Test Substance Measurements using Single Layer Slab with One Exposed Surface (Test 1b).

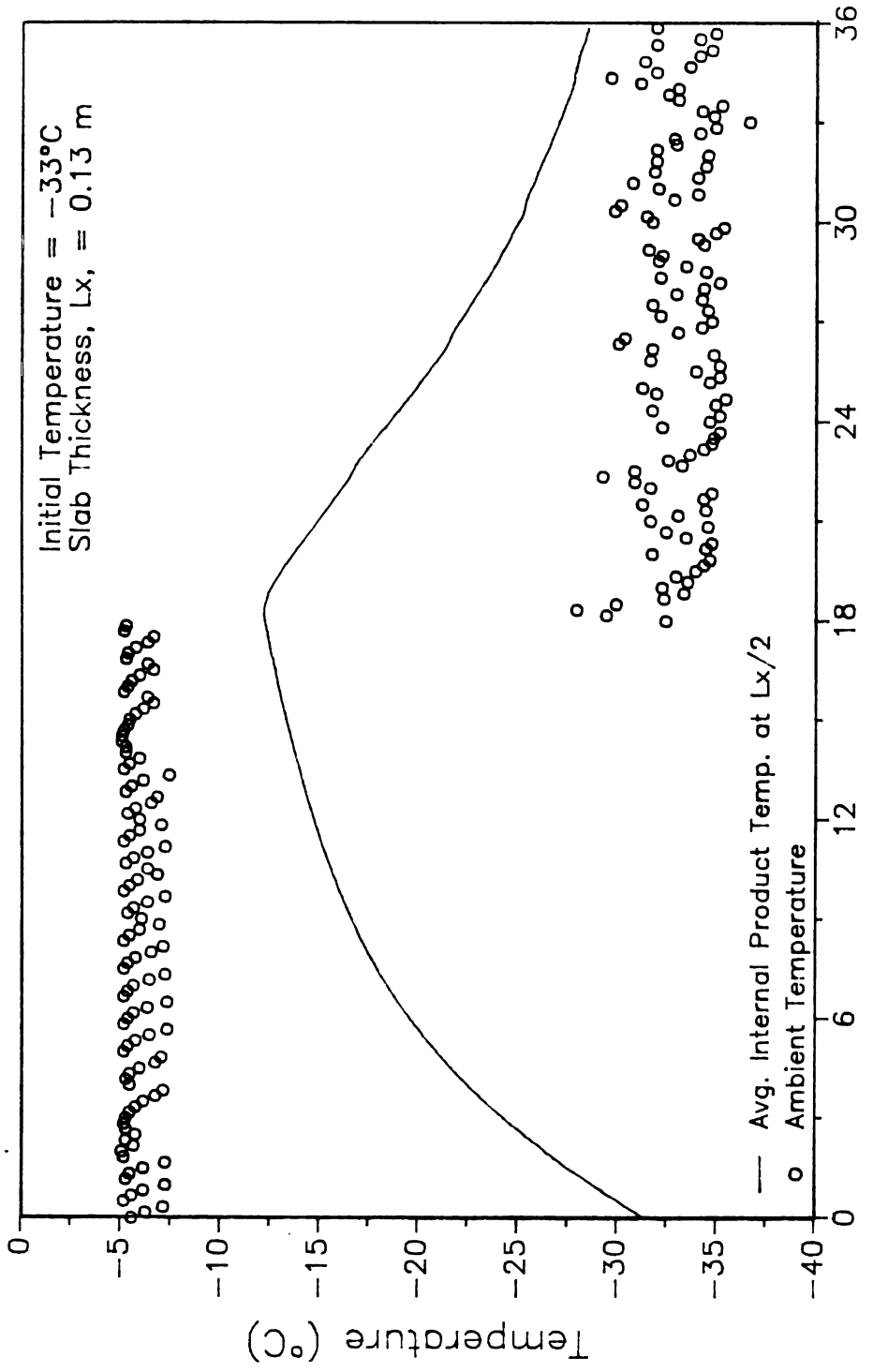


Figure E.1b Ambient and Average Internal Temperature of Karlsruhe Test Substance Measurements using Single Layer Slab with One Exposed Surface (Test 1c).

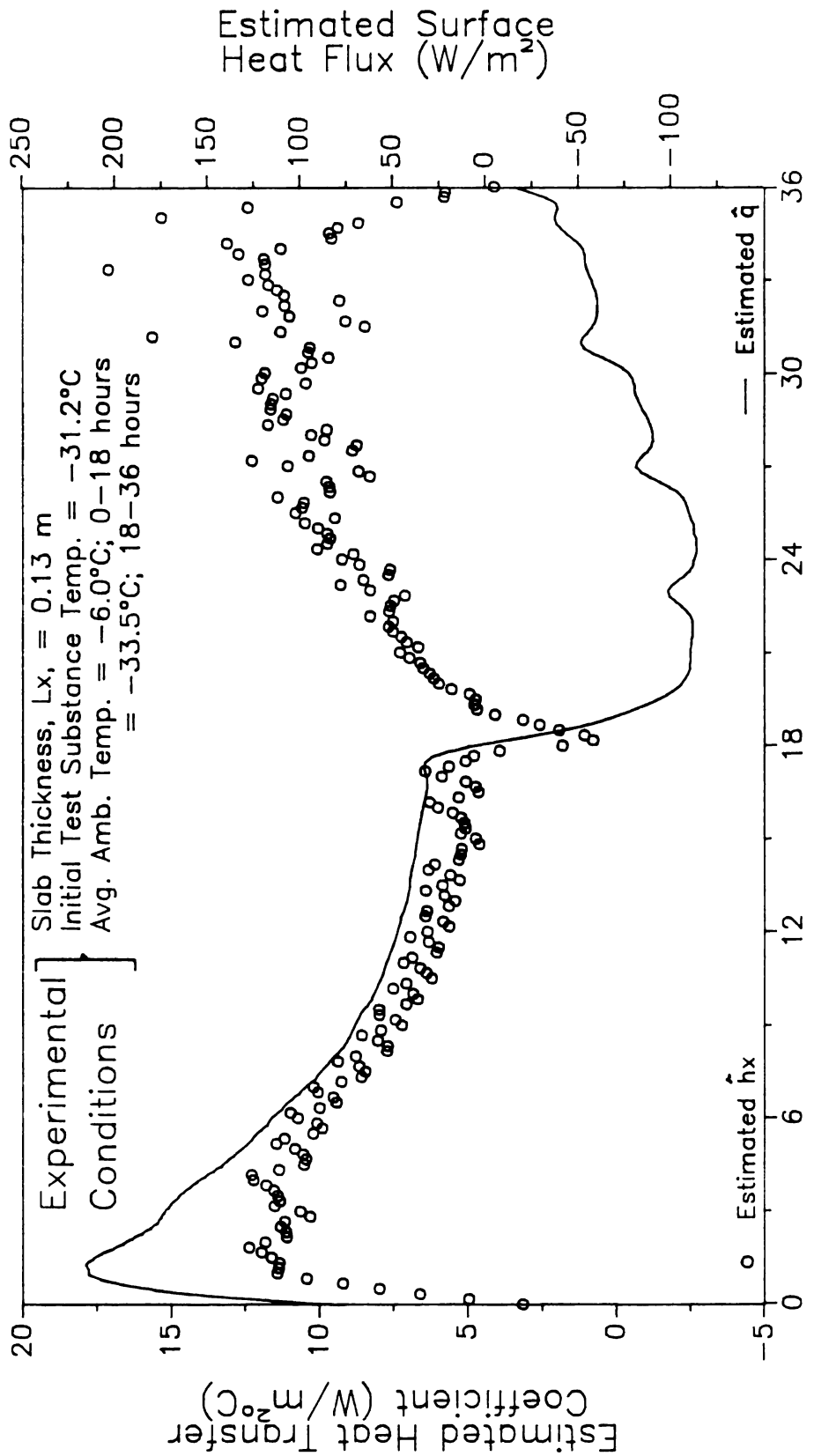


Figure E.2a Estimated Heat Transfer Coefficients, $\hat{h}x$, and Surface Heat Flux, $\hat{q}$, using Experimental Results with Karlsruhe Test Substance from Single Layer Slab with One Exposed Surface (Test 1b).

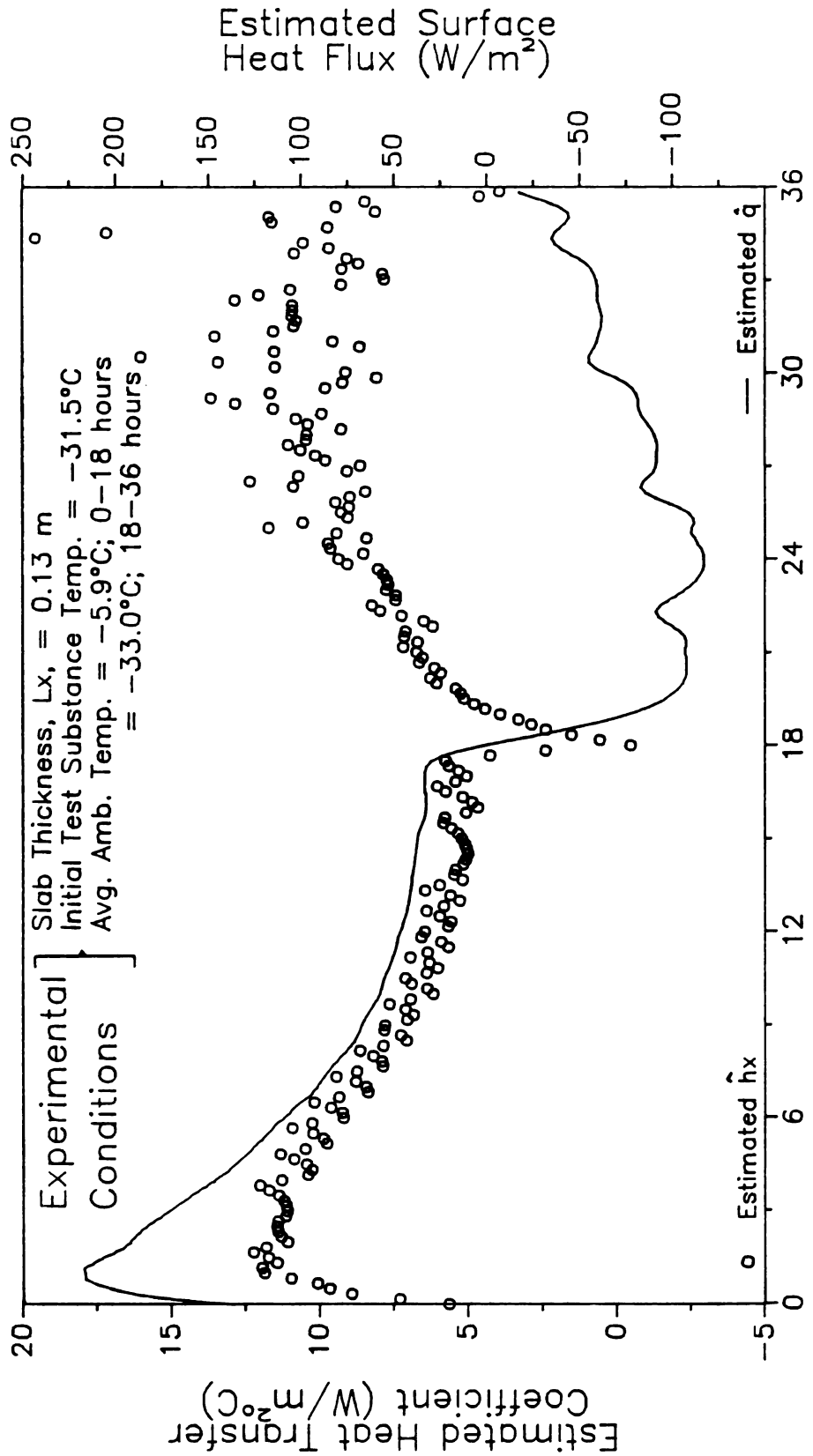


Figure E.2b Estimated Heat Transfer Coefficients, $\hat{h}x$, and Surface Heat Flux, $\hat{q}$, using Experimental Results from Single Layer Slab with One Exposed Surface (Test 1c).

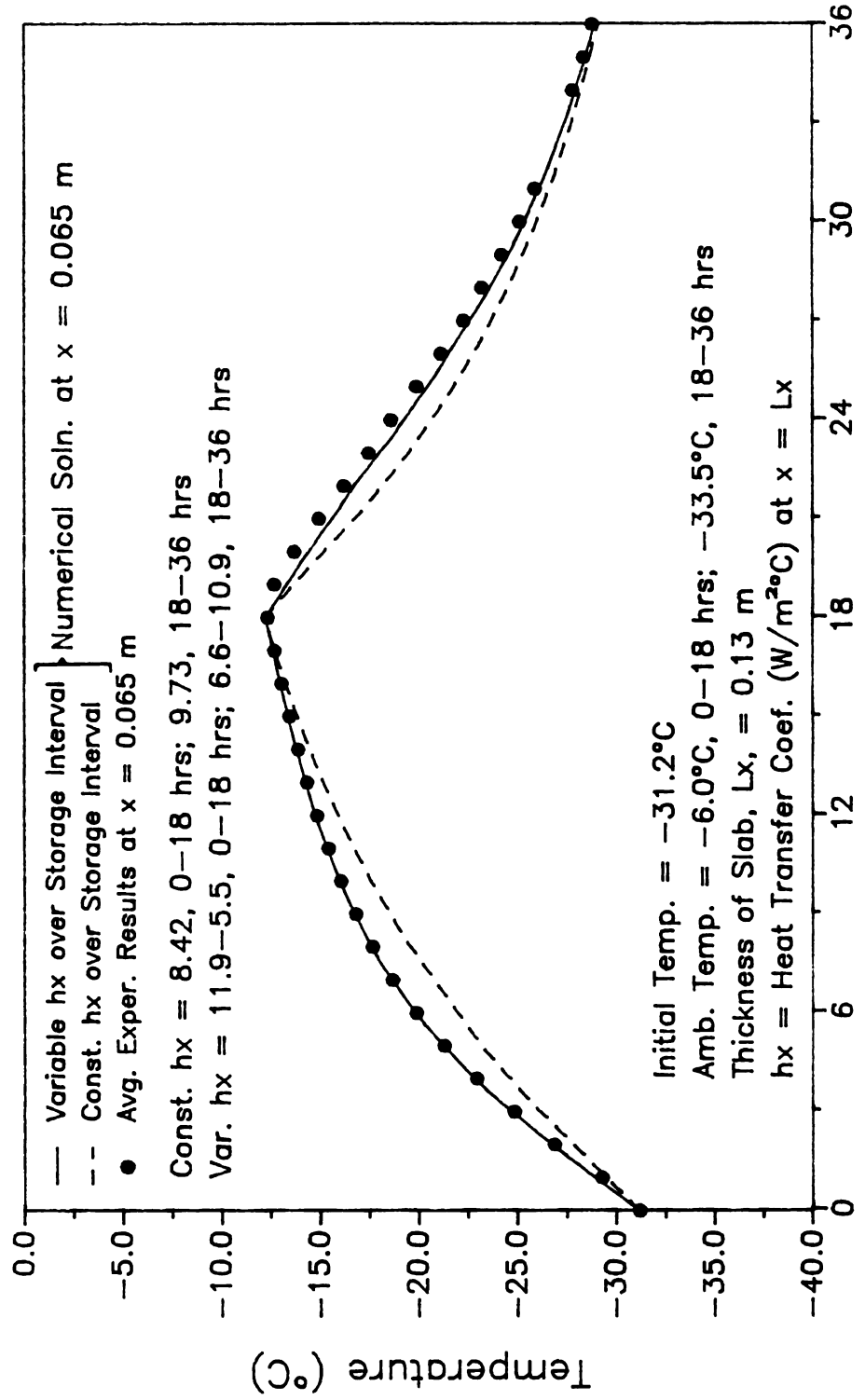


Figure E.3a One Dimensional Numerical Solution Compared to Experimental Results from Single Layer Slab with One Exposed Surface (Test 1b).

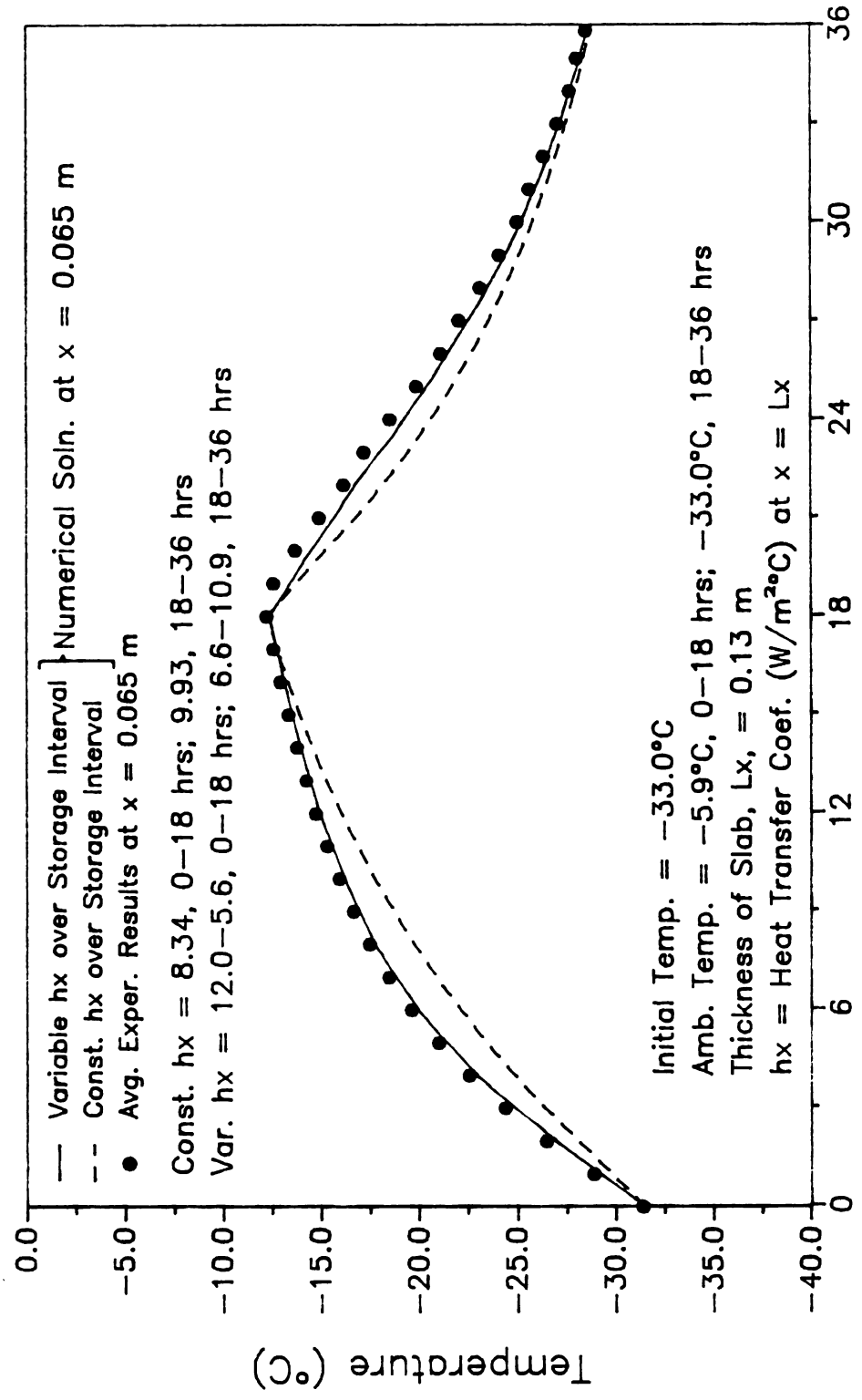


Figure E.3b One Dimensional Numerical Solution Compared to Experimental Results from Single Layer Slab with One Exposed Surface (Test 1c).

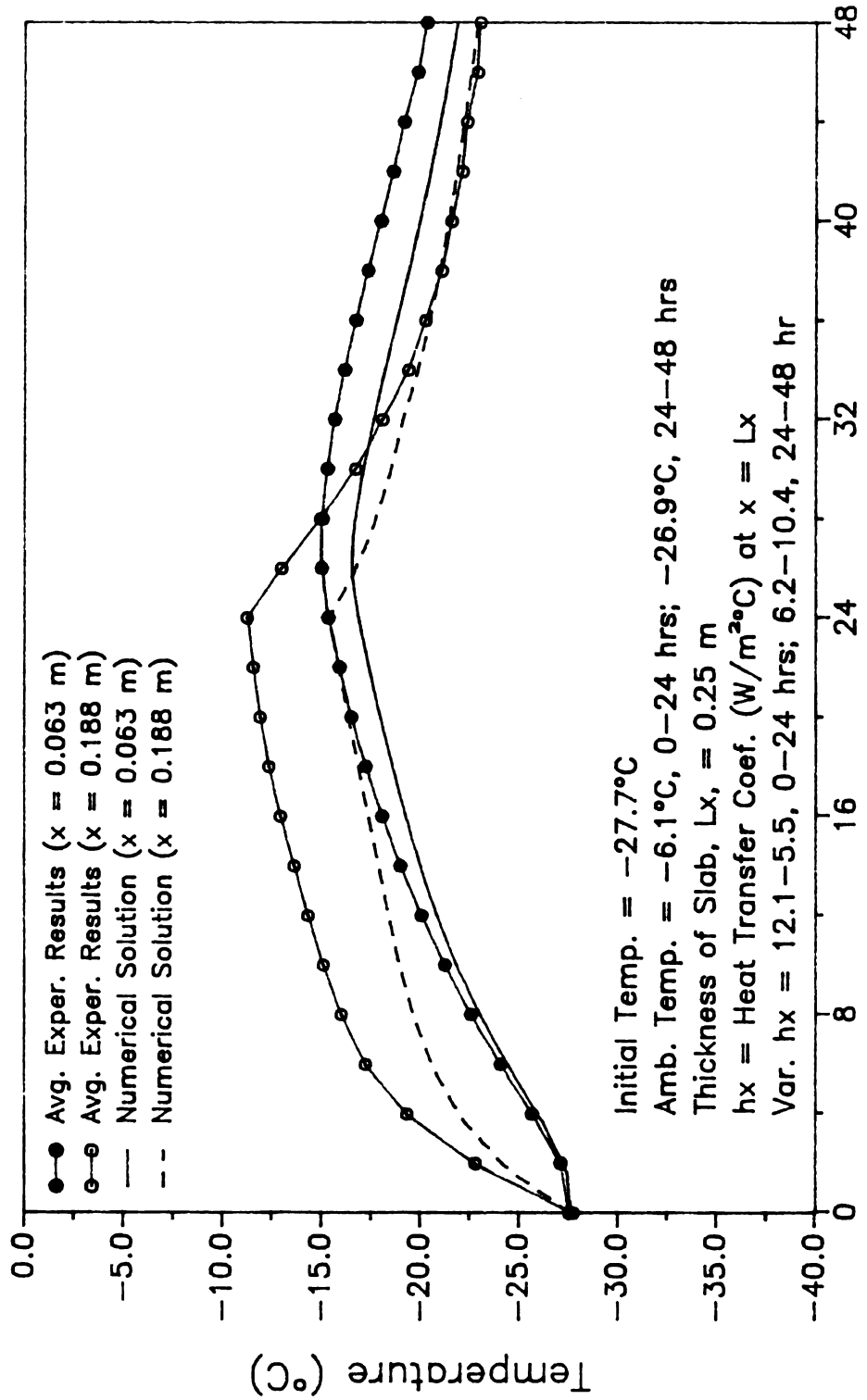


Figure E.4a One Dimensional Numerical Solution Compared to Experimental Results from Double Layer Slab with One Exposed Surface (Test 2b).

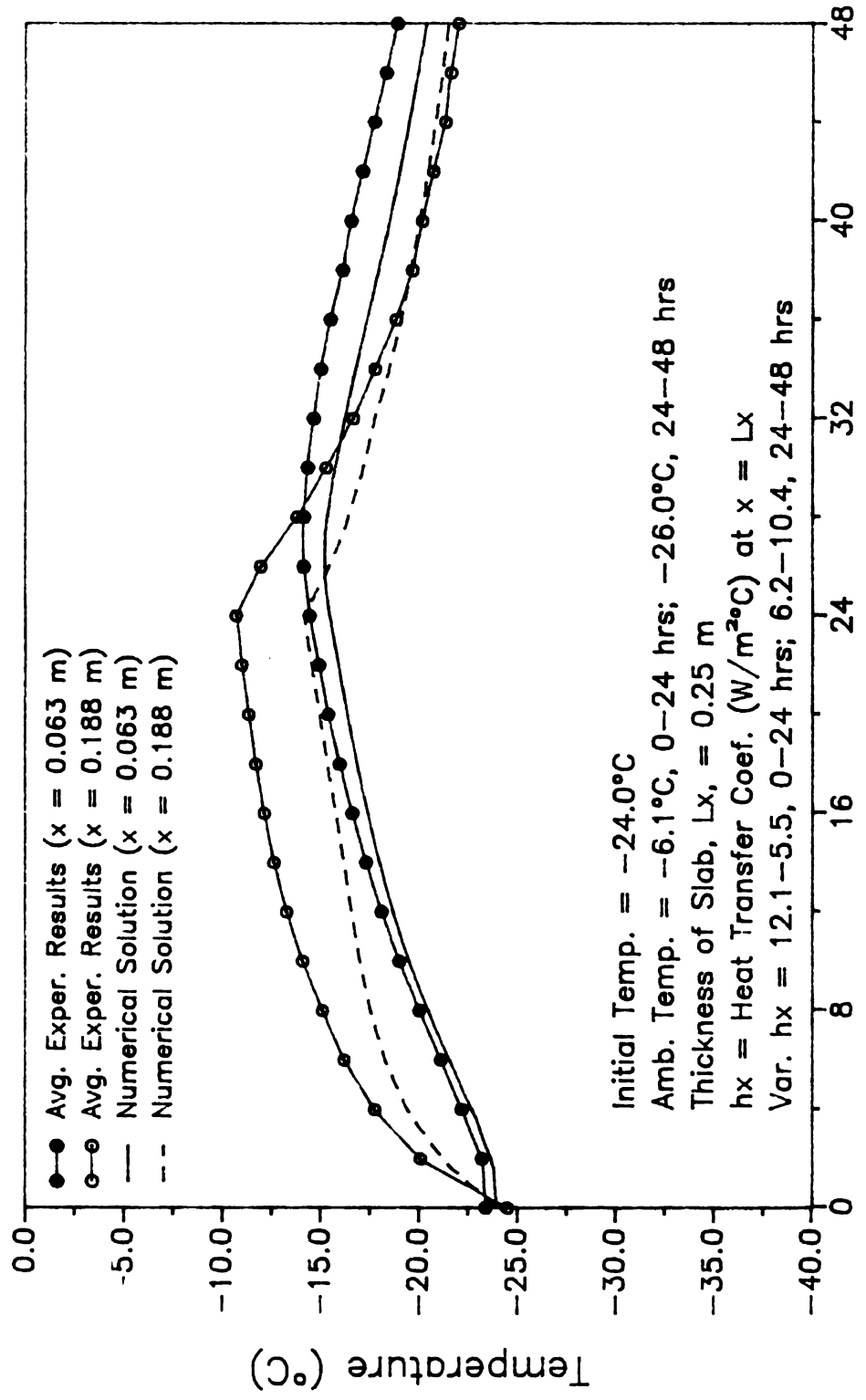


Figure E.4b One Dimensional Numerical Solution Compared to Experimental Results from Double Layer Slab with One Exposed Surface (Test 2c).

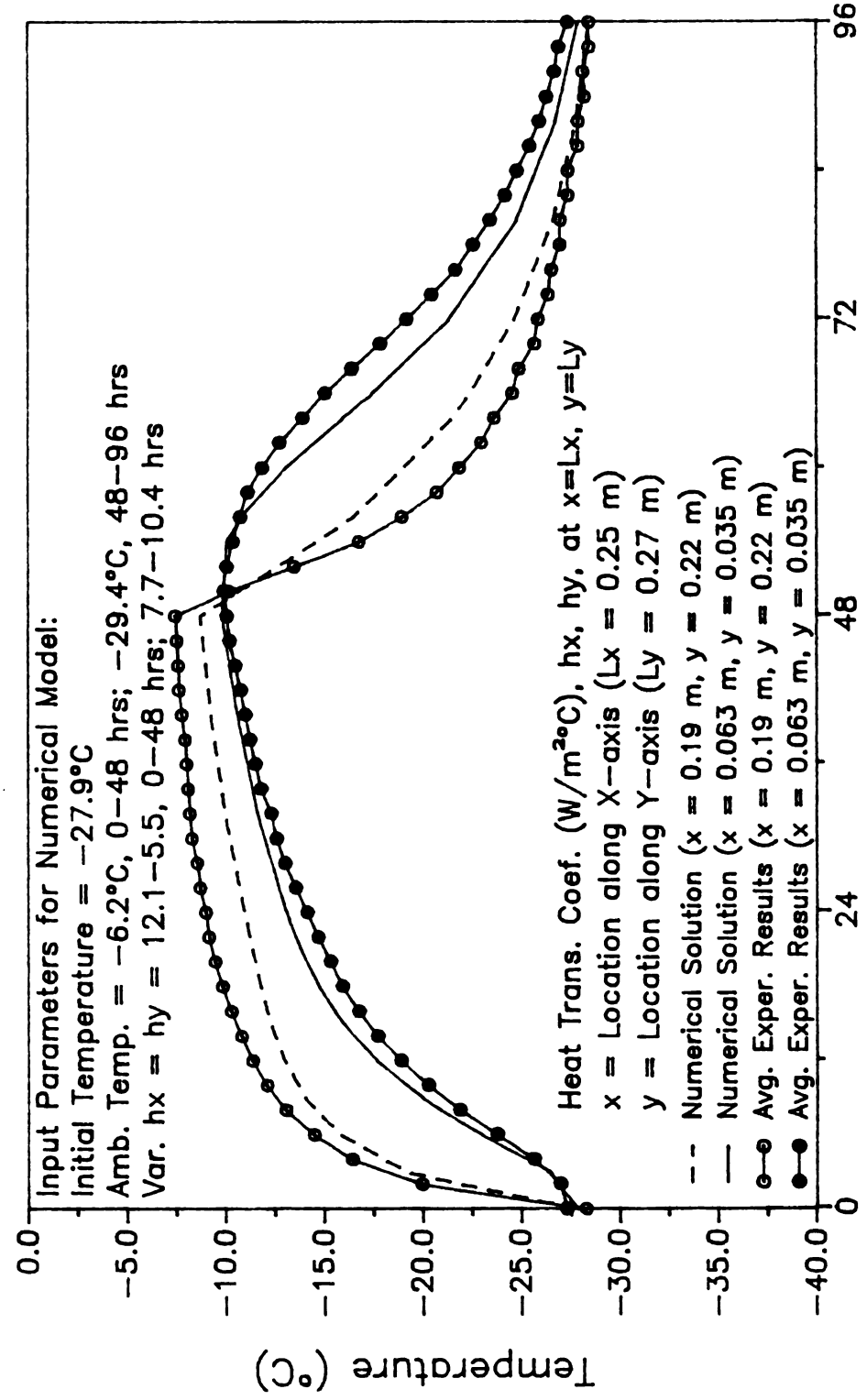


Figure E.5b Two Dimensional Numerical Solution Compared to Experimental Results from Triple Layer Slab with Two Exposed Surfaces (Test 3c).

BIBLIOGRAPHY

BIBLIOGRAPHY

Abramowitz, M. and I. A. Stegun, 1970. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Dover Publications, Inc., New York.

Abdalla, H. and R. P. Singh, 1984. Simulation of Thawing of Foods Using Finite-Element Method. Journal of Food Process Engineering.

Allada, S. R. and D. Quon, 1966. A stable, explicit numerical solution of the conduction equation for multidimensional nonhomogeneous media. Chemical Engineering Progress Symposium Series - Heat Transfer. Los Angeles, CA. 62(64):151-156.

Anderson, D. A., J. C. Tannehill, and R. H. Pletcher, 1984. Computational Fluid Mechanics and Heat Transfer. Hemisphere Publishing Corporation, McGraw-Hill Book Company, New York.

Arrhenius, 1889. Z. Physik Chem. 4:226. Cited from Charm, S. E., 1971. The Fundamentals of Food Engineering. The Avi Publishing Company, Inc. Westport, Connecticut.

Ashby, B. H., A. H. Bennett, W. A. Bailey, W. Moleeratanond, and A. Kramer, 1979. Energy savings and quality deterioration from holding frozen at two daily temperature levels. Transactions of the ASAE. 22(4):938-943.

Backus, B. and F. Gilbert, 1980. Uniqueness in the inversion in inaccurate gross earth data. Phil. Trans. R. Soc., London. 266:123-192.

Bass, B. R., 1980. Application of the finite element method to the nonlinear inverse heat conduction problem using Beck's second method. Journal of Engineering for Industry, Transactions of the ASME. 102:168-176.

Bathe, K. and E. L. Wilson, 1976. Numerical Methods in Finite Elements Analysis. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Baumeister T., E. A. Avollone, and T. Baumeister, editors, 1978. Marks Handbook for Mechanical Engineers, Eighth Edition. Mcgraw-Hill Book Company, New York.

Beck, J. V., 1967. Transient sensitivity coefficients for thermal contact conductance. International Journal of Heat and Mass Transfer. 10:1615-1617.

Beck, J. V.,1968. Surface heat flux using an integral method. Nuclear Engineering and Design, 7:170-178.

Beck, J. V., 1969. Determination of optimum transient experiments for thermal contact conductance. International Journal of Heat and Mass Transfer. 12:621-633.

Beck, J. V., 1970. Nonlinear estimation applied to the nonlinear inverse heat conduction problem. International Journal of Heat and Mass Transfer. 13:703-716.

Beck, J. V., B. Blackwell, and C. R. St. Clair, Jr., 1985. Inverse Heat Conduction Problems. John Wiley & Sons, Inc., New York.

Beck, J. V., B. Litkouhi, and C. R. St. Clair, Jr., 1982. Efficient sequential solution of the nonlinear inverse heat conduction problem. Numerical Heat Transfer. 5:275-286.

Beck, J. V. and D. A. Murio, 1986. Combined function specification-regularization procedure for solution of inverse heat conduction problem. AIAA Journal, 24:180-185.

Belytschko, T. and T. J. R. Hughes, 1983. Computational Methods for Transient Analysis. Vol. 1 in Computational Methods in Mechanics. Evsevier Science Publishering Company, Inc., New York.

Bhattacharya, M. and M. A. Hanna, 1986. Kinetics of texture and color degradation of frozen ground beef during storage. Presented at the 46th Annual Meeting and Food Expo of the Institute of Food Technologists. Dallas, Texas, June 15-18.

Blackwell, B. F., 1981. Efficient technique for the numerical solution of the one-dimensional inverse problem of the conduction. Numerical Heat Transfer. 4:229-238.

Boggs, M. M., W. C. Dietrich, M. Nutting, R. L. Olsen, F. E. Lindquist, G. S. Bohart, H. J. Neumann, and H. J. Morris, 1960. Time-temperature tolerance of frozen foods. XXI. Frozen peas. Food Technology. 14:181-185.

Bonacina, C. and G. Comini, 1972. Calculation of convective heat transfer coefficients from time-temperature curves. Proceedings of the XII International Congress of Refrigeration. pp. 157-166.

Bonacina, C. and G. Comini, 1973a. On a numerical method for the solution of the unsteady state heat conduction equation with temperature dependent parameters. Proceedings of the XII International Congress of Refrigeration. pp. 157-166.

Bonacina, C. and G. Comini, 1973b. On the solution of the nonlinear heat conduction equations by numerical methods. International Journal of Heat and Mass Transfer. 16:581-589.

Bonacina, C., G. Comini, A. Fasano, and M. Primicerio, 1973. Numerical solution of phase-change problems. International Journal of Heat and Mass Transfer. 16:1825-1832.

Burgraff, O. R., 1964. An exact solution of the inverse problem in heat conduction theory and applications. Journal of Heat Transfer, Transactions of the ASME, August, pp. 373-382.

Carslaw, H. S. and J. C. Jaeger, 1959. Conduction of Heat in Solids. Oxford University Press, Oxford.

Charm, S. E., 1971. The Fundamental of Food Engineering. The Avi Publishing Company, Inc. Westport, Connecticut.

Charm, S. E., D. H. Brand, and D. W. Baker, 1972. A simple method for estimating freezing and thawing times of cylinders and slabs. ASHREA Journal. November, 14:39-45.

Chavarria, V. M., and D. R. Heldman, 1983. Measurement of convective heat transfer coefficients during food freezing processes. Journal of Food Science. 49(3):810-814.

Chen, C. S., 1986. Effective molecular weight of aqueous solutions and liquid foods calculated from the freezing point depression. Journal of Food Science. 51(6):1537-1543.

Chen, C. S., S. V. Ting, and E. C. Hill, 1984. Predicting temperature changes in freezing of citrus sections contained in steel drums. Transactions of the ASAE. pp. 1604-1608.

Chernous'ko, F. L., 1970. Solution of non-linear heat conduction problems in media with phase changes. International Chemical Engineering. 10(1):42-48.

Chu, Y., 1983. A statistical analysis of the relationship between frozen food quality and temperature. M. S. Thesis. Department of Food Science and Human Nutrition, Michigan State University, East Lansing, MI.

Chu, Y. and D. R. Heldman, 1984. Sensitivity of frozen food shelf-life prediction to statistical variability of input parameters. Presented at the 44th Annual Meeting and Food Expo of the Institute of Food Technologists Anaheim, California, June, 1984.

Churchill, S. W., 1977. A comprehensive correlating equation for laminar, assisted, forced and free convection. AIChE Journal, 23:10.

Churchill, S. W., 1983. Combined free and forced convection around immersed bodies. In E. U. Schlunder, Ed.-in-Chief, 1983. Heat Exchanger Design Handbook, Section 2.5.9, Hemisphere Publishing Corp., New York.

Cleland, A. C. and R. L. Earle, 1976. A new method for prediction of surface heat transfer coefficients in freezing. Proceedings of the XVI International Congress of Refrigeration. pp. 361-376.

Cleland, A. C. and R. L. Earle, 1977a. Freezing time predictions for foods -- a simplified procedure. International Journal of Refrigeration. 5(3):134-140.

Cleland, A. C. and R. L. Earle, 1977b. The third kind of boundary condition in numerical freezing calculations. International Journal of Heat and Mass Transfer. 16:581-589.

Cleland, A. C. and R. L. Earle, 1984. Assessment of freezing time prediction methods. Journal of Food Science. 49(4)1034-1042.

Cohen, E. and I. Saguy, 1985. Statistical evaluation of Arrhenius model and its applicability in prediction of food quality losses. Journal of Food Processing and Preservation. 9:273-290.

Comini, G., 1972. Design of transient experiments for measurement of convective heat transfer coefficients. Proceedings of the XII International Congress of Refrigeration. pp. 157-166.

Comini, G., S. D. Guidice, R. W. Lewis, and O. C. Zienkiewicz, 1974. Finite element solution of non-linear heat conduction problems with special reference to phase change. International Journal for Numerical Methods in Engineering. 8:613-624.

Creed, P. G. and S. J. James, 1985. Heat transfer during the freezing of liver in a plate freezer. Journal of Food Science. 50:285-288,294.

Dagerskog, M., 1974. Development of a computer program to simulate temperature and time-temperature exposure of packed frozen food during handling in the distribution chain. Swedish Institute for Food Preservation Research (SIK). Report TSBN 91-7200-014-8.

De Baerdemaeker, J., R. P. Singh, and L. J. Segerlind, 1977. Modelling heat transfer in foods using the finite-element method. Journal of Food Process Engineering. 1:37-50.

De Cindio, B., G. Iorio, and V. Romano, 1985. Thermal analysis of the freezing of ice cream brickettes by the finite element method. Journal of Food Science. 50:1463-1466.

De Michelis, A. and A. Calvelo, 1982. Mathematical models for nonsymmetric freezing of beef. Journal of Food Science. 47:1211-1217.

Dietrich, C., M. F. Nutting, M. M. Boggs, and N. E. Weinstein, 1962. Time-temperature tolerance of frozen foods XIV. Quality changes in cauliflower. Food Technology. 16:123-128.

Dietrich, W. C., M. Nutting, R. L. Olsen, F. E. Lindquist, M. M. Boggs, G. S. Bohart, H. J. Neumann, and H. J. Morris, 1960. Time-temperature tolerance of frozen foods. XVI. Quality retention of frozen green snap beans in retail packages. Food Technology. 14:181-185.

Dix, R. C. and J. Cizek, 1971. The isotherm migration method for transient heat conduction analysis. Proceedings. Fourth International Heat Transfer Conference, Paris. ASME, New York. Vol. 1, Cu 1.1, pp. 1-10.

Douglas, Jr., J., 1955. On the numerical integration of $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \partial u/\partial t$ by implicit methods. SIAM Journal. 3:42-65.

Douglas, Jr., J. and J. E. Gunn, 1964. A general formulation of alternating direction methods. Part I. Parabolic and hyperbolic problems. Numerische Mathematik. 6:428-453.

Evans, D. J. and C. R. Gane, 1978. A.D.I. methods for the solution of transient heat conduction problems in r-θ geometry. International Journal for Numerical Methods in Engineering. 12:1799-1807.

Fennema, O. R. and W. D. Powrie, 1964. Fundamentals of low-temperature food preservation. Advances in Food Research. 13:219-347.

Fennema, O. R., W. D. Powrie, and E. H. Marth, 1973. Low-Temperature Preservation of Foods and Living Matter. Marcel Dekker, Inc. New York. pp. 207-227.

Fleming, A. K., 1973. The numerical calculation of freezing processes. Proceedings of the XIII International Congress of Refrigeration. AVI Publishing, Inc. Vol. II, pp. 303-311.

Fried, I., 1979. Numerical solution of Differential Equations. Academic Press, New York.

Goldstein, R. J., E. M. Sparrow, and D. C. Jones, 1973. Natural convection mass transfer adjacent to horizontal plates. International Journal of Heat and Mass Transfer. 16:1025.

Gortner, W. A., F. Fenton, F. E. Volz, and E. Gleim, 1948. Effect of fluctuating storage temperatures on quality of frozen foods. Industrial and Engineering Chemistry. 40(8):1423-1426.

Grange, B. W., R. Viskanta, and W. H. Stevenson, 1976. Diffusion of heat and solute during freezing of salt solutions. International Journal of Heat and Mass Transfer. 19:373-384.

Gutschmidt, J, 1960. Uber Das Herstellung Und Verpacken Der Karlsruher Prufmasse. Kaltetechnik. 12(8):226-229. (In German).

Haralampu, S. G., I. Saguy and M. Karel, 1985. Estimation of Arrhenius model parameters using three least squares methods. Journal of Food Processing and Preservation. 9:129-143.

Hayakawa, K. and A. Bakal, 1972. Formulas for predicting transient temperatures in food during freezing or thawing. AIChE Symposium Series, Food Preservation. 69(132):14-25.

Heldman, D. R., 1974a. Computer simulation of food freezing process. Proceedings of the IV International Congress of Food Science and Technology. Vol. IV, pp. 397-406.

Heldman, D. R., 1974b. Predicting the relationship between unfrozen water fraction and temperature during food freezing using freezing point depression. Transactions of the ASAE. 17:63-66.

Heldman, D. R., 1982. Food properties during freezing. Food Technology. 36(2):92-96.

Heldman, D. R. and D. P. Gorby, 1975a. Prediction of thermal conductivity in frozen foods. Transactions of the ASAE. 18(1):740-***.

Heldman, D. R. and D. P. Gorby, 1975b. Computer simulation of individual-quick freezing of foods. ASAE paper No. 75-6016, ASAE, St. Joseph, MI, 49085.

Heldman, D. R. and D. Lai, 1983. A model for prediction of shelf-life for frozen foods. Proceedings of 16th Congress of International Institute of Refrigeration, Paris, France, pp. 417-433.

Heldman, D. R. and R. P. Singh, 1981. Food Process Engineering. Second Edition. AVI Publishing Company, Inc., Westport, Connecticut.

Hensel, E. C. and R. G. Hills, 1984. A space marching finite difference algorithm for the one dimensional inverse conduction heat transfer problem. ASME paper No. 84-HT-48.

Hills, R. G. and Mullholland, 1979. The accuracy and resolving power of one dimensional transient inverse heat conduction theory as applied to discrete and inaccurate measurements. International Journal Heat and Mass Transfer. 22:1221-1229.

Hornbeck, R. W., 1975. Numerical Methods. Quantum Publishers, Inc., New York.

Hsieh, R., L. E. Lerew, and D. R. Heldman, 1977. Prediction of freezing times for foods as influenced by product properties. Journal of Food Process Engineering. 1:183-197.

Hustrulid, A. and J. D. Winter, 1943. The effect of temperatures on frozen fruits and vegetables. Agricultural Engineering. 24(12):416.

Imber, M., 1974. Temperature extrapolation mechanism for two dimensional heat flow. AIAA Journal. 12(8):1089-1093.

Imber, M. and J. Khan, 1972. Prediction of transient temperature distribution with embedded thermocouples. AIAA Journal. 10(6):784-788.

Incropera F. P. and D. P. Dewitt, 1985. Introduction to Heat Transfer. John Wiley & Sons, New York.

Joshi, C. and L. C. Tao, 1974. A numerical method of simulating the axisymmetrical freezing of food systems. Journal of Food Science. 39:623-626.

Jul, M., 1984. The Quality of Frozen Foods. Academic Press, New York.

Kays, W. M. and M. E. Crawford, 1980. Convective Heat and Mass Transfer, second edition. McGraw Hill Book Company, New York.

Klose, A. A., M. F. Pool, A. A. Campbell, and H. L. Hanson, 1959. Time-temperature tolerance of frozen foods. XIX. Ready-to-cook cut-up chicken. Food Technology. 13:477-484.

Komori, T. and E. Hirai, 1970. An application of Stefan's problem to the freezing of a cylindrical food-stuff. Journal of Chemical Engineering of Japan. 3(1):39-44.

Kopelman, I. J., 1966. Transient heat transfer and thermal properties in food systems. Ph. D. Thesis. Department of Agricultural Engineering, Michigan State University, East Lansing, MI.

Krutz, G. W., R. J. Schoenhals, and P. S. Hore, 1978. Application of the finite-element method to the inverse heat conduction problem. Numerical Heat Transfer. 1:489-498.

Lai, D. and D. R, Heldman, 1982. Analysis of kinetics of quality change in frozen foods. Journal of Food Process Engineering 6:179-200.

Langford, D., 1976. New analytical solutions of the one-dimensional heat equation for temperature and heat flow rate both prescribed at the same fixed boundary (with applications to the phase change problem). Quarterly of Applied Mathematics. 24(4):315-322.

Lees, M., 1966. A linear three-level difference scheme for quasilinear parabolic equations. Mathematics of Computation. 20:516.

Lescano, C. E., 1973. Predicting freezing curves in codfish fillets using the ideal binary solution assumption. M. S. Thesis. Department of Agricultural Engineering, Michigan State University, East Lansing, MI.

Lewis, R. W., K. Morgan, and P. M. Roberts, 1984. Application of an alternating-direction finite-element method to heat transfer problems involving a change of phase. Numerical Heat Transfer. 7:471-482.

Lightfoot, E. N., C. Massot, and F. Irani, 1965. Approximate estimation of heat and mass transfer coefficients. AICHE Chemical Engineering Symposium Series, Selected Topics in Transport Phenomena. 61(58):28-60.

Mastanaiah, K., 1976. On the numerical solution of phase change problems in transient non-linear heat conduction. International Journal for Numerical Methods in Engineering. 10:833-844.

McAdams, W. H., 1954. Heat Transmission, third edition. McGraw Hill Book Company, New York.

Miller, K., 1970. Least squares methods for ill-posed problems with a prescribed bound. SIAM Journal of Mathematical Analysis. 1(1):52-74.

Modern Plastics Encyclopedia 1984-85. Volume 61, Number 10a. McGraw Hill Publishing Company, New York.

Moleerantanond, W., B. H. Ashby, A. Kramer, B. W. Berry, and W. Lee, 1981. Effect of a di-thermal storage regime on quality and nutritional changes and energy consumption of frozen boxed beef. Journal of Food Science. 46:829-833,837.

Moore, W. J., 1962. Physical Chemistry, third edition. Prentice-Hall, Englewood Cliffs, New Jersey.

Murio, D. A., 1985. On the characterization of the solution of the inverse heat conduction problem. ASME Paper No. 85-WA/HT-41.

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

Omega, 1985. Complete Temperature Measurement Handbook and Encyclopedia. Omega Engineering, Inc.

Osman, A. M. and J. V. Beck, 1987. Nonlinear inverse problem for the estimation of time-and-space dependent heat transfer coefficients. AIAA-87-0150. Presented at the 25th Annual Aerospace Sciences Meeting, January, 12-15, Reno, Ne.

Ozisik, M. N., 1980. Heat Conduction. John Wiley & Sons, Inc. New York.

Peaceman, D. W. and H. H. Rachford, Jr., 1955. The numerical solution of parabolic and elliptic differential equations. Journal for the Society of Industrial and Applied Mathematics. 3(1):28-41.

Perez, M. E. C., 1984. Computer simulation of microbial quality of a food product during freezing and frozen storage. M. S. Thesis. Agricultural Engineering Department, Michigan State University, East Lansing, MI.

Plank, R. Z., 1913. Ges. Kalte-Ind. 20, 109 (In German). Cited by A. J. Ede, 1949. The calculation of the freezing and thawing of food stuffs. Modern Refrigeration. 52(3):52.

Rebellato, L., S. Del Giudice, and G. Comini, 1978. Finite element analysis of freezing processes in foodstuffs. Journal of Food Science. 43:239-242,250.

Reinsch, C. H. J., 1967. Smoothing by spline function. Numerische Mathematik. 10:177-183.

Ross, E. W., M. V. Klicka, J. Kalick, and M. E. Branagan, 1985. Acceptance of a military ration after 24-month storage. Journal of Food Science. 50:178-181,208.

Sanz, P. D., R. H. Mascheroni, M. Domenguez and S. Garcia de Vinuesa, 1986. Time-temperature prediction curves of foodstuffs by means of the z-transfer function method. International Journal of Refrigeration. 9:89-92.

Sastry, S. K. and A. Kilara, 1983. Temperature response of frozen peas to di-thermal storage regimes. Journal of Food Science. 48:77-83.

Schwimmer, S, L. L. Ingraham, and H. M. Hughes, 1955. Temperature tolerance in frozen food processing. Effective temperatures in thermally fluctuating systems. Industrial and Engineering Chemistry. 47(6):1149-1151.

Scott, E. P. and J. V. Beck, 1985. Analysis of order of sequential regularization solution of inverse heat conduction problem. ASME paper No. 85-WA/HT-43.

Segerlind, L., 1984. Applied Finite Element Analysis. John Wiley and Sons, New York.

Segerlind, L., 1986. Personal Communication.

Singh, R. P., 1976. Computer simulation of food quality during frozen food storage. International Institute of Refrigeration Commissions C2, D1, D2, D3, E1. Annex. pp. 197-204.

Singh, R. P. and D. R. Heldman, 1976. Simulation of liquid food quality during storage. Transactions of the ASAE. 19(1):178-184.

Singh, R. P. and C. Y. Wang, 1977. Quality of frozen foods - A review. Journal of Food Process Engineering. 1:97-127.

Specht, H., H. Karius, and J. Kunis, 1981. Zur Herstellung und Untersuchung von Gefrierprufmassen. Lebensmittelindustrie. 28:12-13. (In German).

Stolz, G., Jr., 1960. Numerical solutions to an inverse problem of heat conduction for simple shapes. Transactions of the ASME. Journal of Heat Transfer. Vol. 82, Series C, pp. 20-26.

Succar, J. and K-I Hayakawa, 1986. A response surface method for the estimation of convective and radiative heat transfer coefficients during freezing and thawing of foods. Journal of Food Science. 51:1314-1322.

Talmon, Y. and H. T. Davis, 1981. Analysis of propagation of freezing and thawing fronts. Journal of Food Science. 46:1478-1483,1488.

Tarnawski, W., 1976. Mathematical model of frozen consumption products. International Journal of Heat and Mass Transfer. 19:15-20.

Thomas, L. H., 1949. Elliptic problems in linear difference equation over a network. Watson Scientific Comput. Laboratory Report, Columbia University, New York. Cited from Anderson, D. A., J. C. Tannehill, and R. H. Pletcher, 1984. Computational Fluid Mechanics and Heat Transfer. Hemisphere Publishing Corporation, McGraw-Hill Book Company, New York.

Tien, R. H. and G. E. Geiger, 1967. A heat-transfer analysis of the solidification of a binary eutectic system. Transactions of the ASME. Journal of Heat Transfer. August. pp. 230-234.

Tikhonov, A. N. and V. Y. Arsenin, 1977. Solutions of Ill-posed Problems. V. H. Winston and Sons, Washington, D.C.

Van Arsdel, W. B., 1957. The time-temperature tolerance of frozen foods. I. Introduction - the problem and the attack. Food Technology. 11:28.

Van Arsdel, W. B. and D. G. Guadagni, 1959. Time-temperature tolerance of frozen foods. XV. Method of using temperature histories to estimate changes in frozen food quality. Food Technology. 13:14-19.

Von Rosenberg, D., 1969. Methods for the solution of partial differential equations. Modern Analytical and Computational Methods in Science and Mathematics. No. 16. New York. American Elsevier.

Walpole, R. E. and R. H. Myers, 1978. Probability and Statistics for Engineers and Scientists, second edition. MacMillan Publishing Company, Inc., New York.

Weber, C. 1981. Analysis and solution of the illposed inverse heat conduction problem. International Journal Heat and Mass Transfer. 24(11):1783-1792

Williams, S. D. and D. M. Curry, 1984. An analytical and experimental study for surface heat flux determination. Journal of Spacecraft. 14(10):632-637.

Yanenko, N. N., 1971. The Method of Fractional Steps: The Solution of Problems of Mathematical Physics in Several Variables. M. Holt, editor, Springer-Verlag, New York.

Zaritzky, N. E., 1982. Mathematical simulation of the thermal behavior of frozen meat during its storage and distribution. Journal of Food Process Engineering. 6:15-36.

Zurtz, C. A., S. K. Sastry, and S. C. McCoy, 1986. Modelling of temperature fluctuations within frozen foods stored in cylindrical containers. International Journal of Refrigeration. 9:215-219.

Zurtz, C. A. and S. K. Sastry, 1986. Effect of packaging materials on temperature fluctuations in frozen foods: mathematical model and experimental studies. Journal of Food Science and Technology. 51(4):1050-1056.

Zurtz, C. A. and R. P. Singh, 1985. Modeling Temperature fluctuations in stored frozen foods. International Journal of Refrigeration. 8:289-293.