T-3891

HEAT CONDUCTION THROUGH A POLYMER FILM DURING FLASH FUSION

by

Trina J. Igelsrud

ARTHUR LAKES LIBRARY
COLORADO SCHOOL of MINES
GOLDEN, COLORADO 80401

ProQuest Number: 10783605

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 10783605

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

A thesis submitted to the faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Chemical and Petroleum-Refining Engineering).

Golden, Colorado

Date April 9, 1990

Signed: Trina J. Igelsrud
Trina J. Igelsrud

Approved: Sami Selim
Dr. Sami Selim
Thesis Advisor

Approved: Victor Yesavage
Dr. Victor Yesavage
Thesis Advisor

Golden, Colorado

Date 4/9/90

A. J. Kidnay
Dr. A. J. Kidnay
Department Head, Chemical and Petroleum Refining Engineering

ABSTRACT

Conventional office copiers have high maintenance associated with paper jams and high energy costs. These problems can be reduced by using a non-contact fusing technique such as flash fusion where toner particles are radiated for several milliseconds to accomplish the fusion process.

The purpose of this work is to study the temperature distributions and temperature gradients which develop in the polymer film during flash fusion. The heat conduction equation is solved analytically with the appropriate boundary conditions. The Crank-Nicholson finite difference method is used to numerically solve the heat equation and results were compared with the value of the analytical solution for the case of constant physical and thermodynamic properties.

The numerical solution when compared with the analytical solution was shown to be extremely accurate. The actual error analysis showed typical errors of 5.0X10-5.

An extension of the numerical technique to include first variable thermal conductivity and then variable heat capacity was implemented. Having solved these cases the paper was added to the scope of the problem.

When the extension to variable thermal conductivity was analyzed, the results were sufficiently close to the constant property case, and it was concluded that thermal conductivity has minimal effect on the temperature gradient for the temperature values of the polymer during flash fusion.

In the variable c_p case, there was a significant difference between constant property and variable property results. When the values of C_p for the variable case were consistantly below that of the constant C_p case the resulting temperature profiles throughout the polymer and in particular the surface temperature of the polymer were much higher. Conversely, when the values of C_p for the variable case were consistantly above that of the constant C_p case the resulting temperature was lower throughout the polymer.

Addition of paper to the model caused the average temperature of both variable C_p cases to decrease between $45^{o}C$ and $47^{o}C$. This is due to the paper absorbing some of the heat generated by the flash fusion process.

TABLE OF CONTENTS

List of Figures

List of Tables

ACKNOWLEDGEMENTS

This work is dedicated in memory of my grandfather, Dr. Iver Igelsrud, whose influence helped me decide the direction of my career.

I would like to thank Dr. Selim and Dr. Yesavage for their direction, inspiration, and knowledge which guided me throughout this work. Their support will be always remembered.

I would also like to thank everyone on my committee, especially Dr. Baldwin, for their support and encouragement during the most intense and emotional period of this project.

My parents helped in several ways which caused me to have both the time and resources to do most everything at home. A special thanks goes to my mother for always being there when I needed her and her willingness to help with my son while I was busy at school.

I would also like to thank the people at JSD Simulation Service Company (with special thanks to Jay Dweck) for the flexibility of hours and the unlimited computer resources.

My husband John, was always a pillar of support when it came to switching places where my son was concerned. I thank both he and my son Erik whose mere presense light my life and gave me the much needed strength to continue.

This part of my life was one of the most intense and emotional periods I have ever been through. There are a few people which I must give special thanks to for all the words of encouragement and inspiration I received. When I started this program I did not realize I would finish almost 10 years later. The following people remain in my thoughts as true friends whose advice helped me to continue: Denise Goddard, Kristin Thurlow, Karen Taylor, Steve Lythgoe, Randy Pfeiffer, Patsy Wegner Oppenheim, Tim Barbari, Linda Marchetti, and Doug Rosenoff.

Chapter I

Introduction

In a conventional office copier, polymeric toner particles pigmented with carbon black are fused onto a copy paper by pressing them using a hot roll fuser. Disadvantages of this technique include high maintenance associated with paper jams, high energy costs, and potential sticking problems resulting from the pressure process.

These difficulties can be mitigated in an electrophotographic process by using a non-contact fusing technique such as flash fusion, where the toner particles are intensely radiated for several milliseconds to accomplish the fusion process.

Figure I.1 shows a schematic diagram of the flash fusion process. A polymer film of thickness ℓ is radiated by a uniform heat flux q for a specified time. The polymer is supported on a paper substrate of thickness ℓ_p. The purpose of the present work is to study the temperature distributions and the temperature gradients which develop in the polymer film during the flash fusion process. To this end, the heat conduction equation is solved analytically

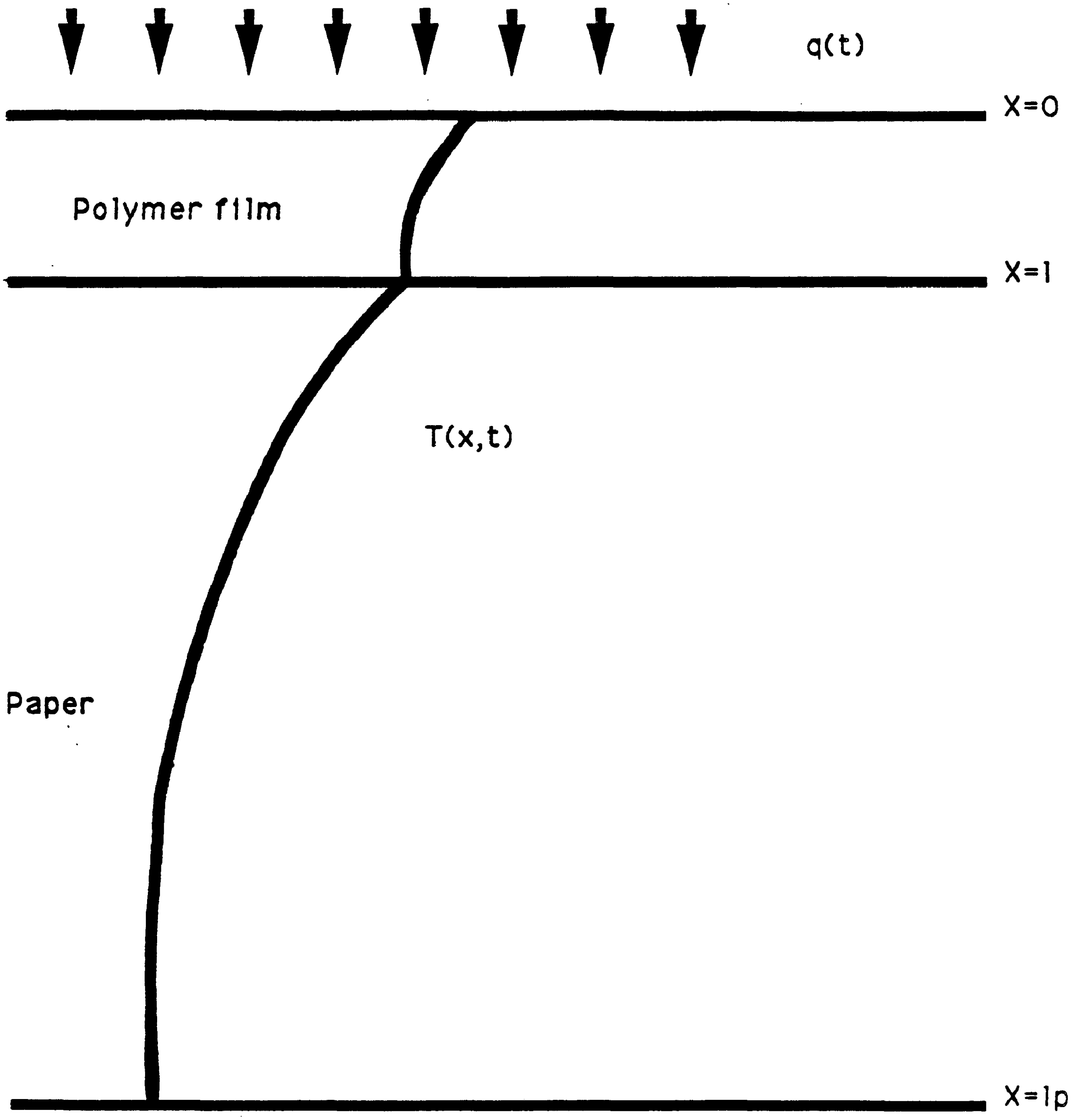

Figure I.1 **Schematic of Flash Fusion Process** under appropriate heat flux boundary conditions. The Crank-Nicholson finite difference method is used to solve the heat equation numerically subject to a boundary condition of the second kind. Results from the numerical solution are compared with the analytical solution for the case of constant physical properties.

The scheme is then extended to include variable thermal conductivity of the system. Comparisons between the constant property solution and the solution with variable thermal conductivity are presented.

With the effects of variable thermal conductivity evaluated, the next property to be varied is the heat capacity. The finite-difference scheme is therefore extended to the case with variable heat capacity. Again, comparisons with the constant property solution and the variable heat capacity solution are performed.

The final phase of the study includes extension of the constant property solution to allow for the presence of the supporting substrate; namely the paper region. This is carried out by using the Crank-Nicholson finite difference scheme to solve the heat conduction equation in the polymer and the paper regions. At the interface between the two regions, a Taylor Series expansion is used to provide continuity of temperature and heat fluxes.

Chapter II
Heat Conduction in a Finite Polymer Film with a Boundary Condition of the Second Kind

Mathematical Formulation

Consider the heating of a polymer film which occupies the finite region $0 \le x \le \ell$ initially at a uniform temperature T_i. At time $t=0$, the polymer is heated by applying a constant heat flux q_0 at the surface $x=0$. The surface at $x=\ell$ is kept insulated. The appropriate differential equations and associated boundary and initial conditions are given by:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \qquad 0 < x < \ell,\ t > 0 \qquad \text{(II.1)}$$

$$-k \frac{\partial T}{\partial x} = q_o \qquad x = 0,\ t > 0 \qquad \text{(II.2)}$$

$$\frac{\partial T}{\partial x} = 0 \qquad x = \ell,\ t > 0 \qquad \text{(II.3)}$$

$$T = T_i \qquad 0 < x < \ell,\ t = 0 \qquad \text{(II.4)}$$

This problem is cast into a more convenient form by defining the following dimensionless variables:

$$T^* = \frac{T - T_i}{q_o \ell / k} \quad , \quad x^* = \frac{x}{\ell} \quad , \quad t^* = \frac{\alpha t}{\ell^2} \qquad \text{(II.5)}$$

The problem then runs as follows:

$$\frac{\partial T^*}{\partial t^*} = \frac{\partial^2 T^*}{\partial x^{*2}} \qquad 0 < x^* < 1,\ t^* > 0 \qquad \text{(II.6)}$$

$$\frac{\partial T^*}{\partial x^*} = -1 \qquad x^* = 0,\ t^* > 0 \qquad \text{(II.7)}$$

$$\frac{\partial T^*}{\partial x^*} = 0 \qquad x^* = 1,\ t^* > 0 \qquad \text{(II.8)}$$

$$T^* = 0 \qquad 0 < x^* < 1,\ t^* = 0 \qquad \text{(II.9)}$$

The temperature distribution $T^*(x^*,t^*)$ is assumed to take the following form:

$$T^*(x^*,t^*) = T_H(x^*, t^*) + \Phi(x^*) + \Psi(t^*) \qquad \text{(II.10)}$$

where $T_H(x^*,t^*)$ shall be required to satisfy a homogeneous boundary value problem. As such, it represents the solution for short times. The functions $\Phi(x^*)$ and $\Psi(t^*)$ represent the solution for long times.

Substitution of Equation (II.10) into Equations (II.6) through (II.9) gives:

$$\frac{\partial T_H}{\partial t^*} + \frac{d\Psi}{dt^*} = \frac{\partial^2 T_H}{\partial x^{*2}} + \frac{d^2\Phi}{dx^{*2}} \qquad 0 < x^* < 1,\ t^* > 0 \qquad \text{(II.11)}$$

$$\frac{\partial T_H}{\partial x^*} + \frac{d\Phi}{dx^*} = -1 \qquad x^* = 0,\ t^* > 0 \qquad \text{(II.12)}$$

$$\frac{\partial T_H}{\partial x^*} + \frac{d\Phi}{dx^*} = 0 \qquad x^* = 1,\ t^* > 0 \qquad \text{(II.13)}$$

$$T_H(x^*,0) + \Phi(x^*) + \Psi(0) = 0 \qquad 0 < x^* < 1,\ t^* = 0 \qquad \text{(II.14)}$$

This system of equations to be satisfied by $T_H(x^*,t^*)$ is selected as follows:

$$\frac{\partial T_H}{\partial t^*} = \frac{\partial^2 T_H}{\partial x^{*2}} \qquad 0 < x^* < 1,\ t^* > 0 \qquad \text{(II.15)}$$

$$\frac{\partial T_H}{\partial x^*} = 0 \qquad x^* = 0,\ t^* > 0 \qquad \text{(II.16)}$$

$$\frac{\partial T_H}{\partial x^*} = 0 \qquad x^* = 1,\ t^* > 0 \qquad \text{(II.17)}$$

$$T_H = -\Phi(x^*) - \Psi(0) \qquad 0 < x^* < 1,\ t = 0 \qquad \text{(II.18)}$$

Accordingly, the functions $\Phi(x^*)$ and $\Psi(t^*)$ satisfy the following system of equations:

$$\frac{d\Psi}{dt^*} = \frac{d^2\Phi}{dx^{*2}} \qquad \text{(II.19)}$$

$$\frac{d\Phi}{dx^*} = -1 \qquad x^* = 0 \qquad \text{(II.20)}$$

$$\frac{d\Phi}{dx^*} = 0 \qquad x^* = 1 \qquad \text{(II.21)}$$

Since t^* and x^* are independent, Equation (II.19) may be integrated to give:

$$\Psi(t^*) = Ct^* + C_1 \qquad \text{(II.22)}$$

$$\Phi(x^*) = \frac{Cx^{*2}}{2} + C_2x^* + C_3 \qquad \text{(II.23)}$$

where, C, C_1, C_2, and C_3 are arbitrary constants. Application of the boundary conditions given in Equations (II.20) and (II.21) to Equation (II.23) gives

$$\Psi(t^*) = t^* + C_1 \qquad \text{(II.24)}$$

and

$$\Phi(x^*) = \frac{x^{*2}}{2} - x^* + C_3 \qquad \text{(II.25)}$$

Substitution of these equations into Equation (II.10) gives:

$$T(x^*,t^*)=T_H^*(x^*,t^*) + \frac{1}{2}x^{*2} - x^* + t^* + C_4 \qquad \text{(II.26)}$$

where $C_4 = C_1 + C_3$. The initial condition (Equation II.18) may now be written as:

$$T_H = -\frac{1}{2}x^{*2} + x^* - t^* - C_4 \quad 0< x^*<1,\ t^*= 0 \qquad \text{(II.27)}$$

We may now proceed to the solution of the homogeneous boundary value problem for $T_H(x^*,t^*)$. Using the method of separation of variables, the solution may be found as:

$$T_H = A_0 + \sum_{n=1}^{\infty} A_N \cos(\lambda_n x^*)\ e^{-\lambda_n^2 t^*} \qquad \text{(II.28)}$$

where $\lambda_n = n\pi$. Applying the initial condition (Equation II.27), we obtain

$$A_N = -2\ \frac{1}{\lambda_n^{\ 2}} \qquad \text{(II.29)}$$

and

$$A_0 = \frac{1}{3} - C_4 \qquad \text{(II.30)}$$

Substituting Equations (II.28), (II.29), and (II.30) into Equation (II.26) gives the final solution as:

$$T^*(x^*,t^*) = \frac{1}{2} x^{*2} - x^* + t^* + \frac{1}{3} +$$

$$\sum_{n=1}^{\infty} -\frac{2}{\lambda_n^{\ 2}} \cos(\lambda_n x^*)\ e^{(-\lambda_n^{\ 2} t^*)} \quad \text{where } \lambda_n = n\pi \qquad \text{(II.31)}$$

The numerical solution to be developed in the next chapter will be compared with the present analytical solution.

Chapter III

Numerical Solution of the Heat Conduction Equation in a Finite Polymer Film with a Boundary Condition of the Second Kind

Mathematical Formulation

Consider the heating of a polymer film which occupies the finite region $0 \le x \le \ell$ initially at a uniform temperature T_i. At time t=0, the film is heated by applying a constant heat flux q_0 at the surface x=0. The polymer film is heated until a time $t=t_f$. The appropriate differential equations and associated boundary and initial conditions are given by:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \qquad 0 \le x \le \ell \qquad \text{(III.1)}$$

$$-k \frac{\partial T}{\partial x} = q_0 \qquad x = 0,\ t_f > t > 0 \qquad \text{(III.2)}$$

$$-k \frac{\partial T}{\partial x} = q_\ell \qquad x = \ell,\ t_f > t > 0 \qquad \text{(III.3)}$$

$$T = T_i \qquad 0 < x < \ell,\ t = 0 \qquad \text{(III.4)}$$

This problem is cast into a more convenient form by defining the following dimensionless variables:

$$X = \frac{x}{\ell} \quad , \quad \tau = \frac{\alpha t}{\ell^2} \quad , \quad U = \frac{T - T_i}{q_0(0)\ell/k} \qquad \text{(III.5)}$$

Equations (III.1) - (III.4) expressed in terms of these dimensionless variables are given by

$$\frac{\partial U}{\partial \tau} = \frac{\partial^2 U}{\partial X^2} \qquad 0 < X < 1, \ \tau > 0 \qquad \text{(III.6)}$$

$$\frac{\partial U}{\partial X} = G_0 \qquad X = 0, \ \tau > 0 \qquad \text{(III.7)}$$

$$\frac{\partial U}{\partial X} = G_1 \qquad X = 1, \ \tau > 0 \qquad \text{(III.8)}$$

$$U = 0 \qquad 0 < X < 1, \ \tau = 0 \qquad \text{(III.9)}$$

where $G_0 = -q_0/q_0$ and $G_1 = -q_\ell/q_0$.

Finite-Difference Formulation

Finite Difference Grid: In order to obtain a finite-difference solution to the parabolic partial differential equation, a typical finite-difference grid is constructed with spacial increments of equal size in the X-direction and time steps of equal size in the τ-direction as shown in Figure (III.1). In this figure, the grid lines i=1, i=N, and j=0 represent respectively the boundaries X=0, X=1, and the

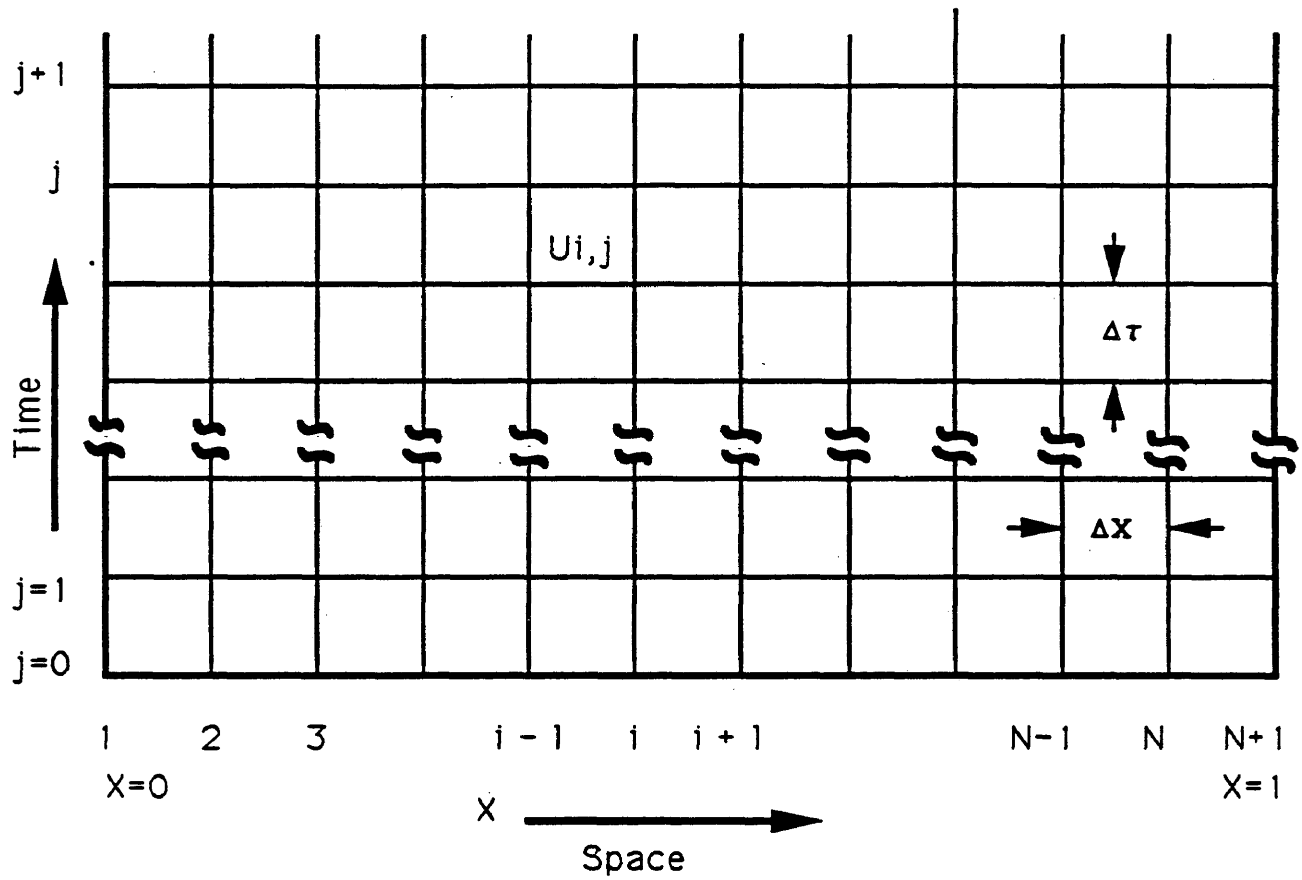

Figure III.1 Finite Difference Grid

initial time τ=0. The term $U_{i,j}$ denotes the estimate of the temperature at the grid points $i\Delta X$ and $j\Delta\tau$.

Finite-Difference Formulation of the Heat Conduction Problem: Equations (III.6)-(III.9) were solved numerically using an implicit scheme involving the Crank-Nicholson finite-difference method. The marching solution starts at time τ = 0 and can continue indefinitely.

A finite-difference formulation of equation (III.6) using the Crank-Nicholson scheme (see Figure III.2) is given by

$$\frac{U_{i,j+1} - U_{i,j}}{\Delta\tau} = \frac{1}{2}\left[\frac{U_{i+1,j+1} - 2U_{i,j+1} + U_{i-1,j+1}}{(\Delta X)^2} + \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{(\Delta X)^2}\right] \qquad i=2,3,\ldots N \qquad \text{(III.10)}$$

The central difference representation of the boundary conditions at X = 0 (i=1) and X = 1 (i=N+1) are:

$$\frac{U_{2,j} - U_{0,j}}{2\Delta X} = G_0 \qquad i = 1 \qquad \text{(III.11)}$$

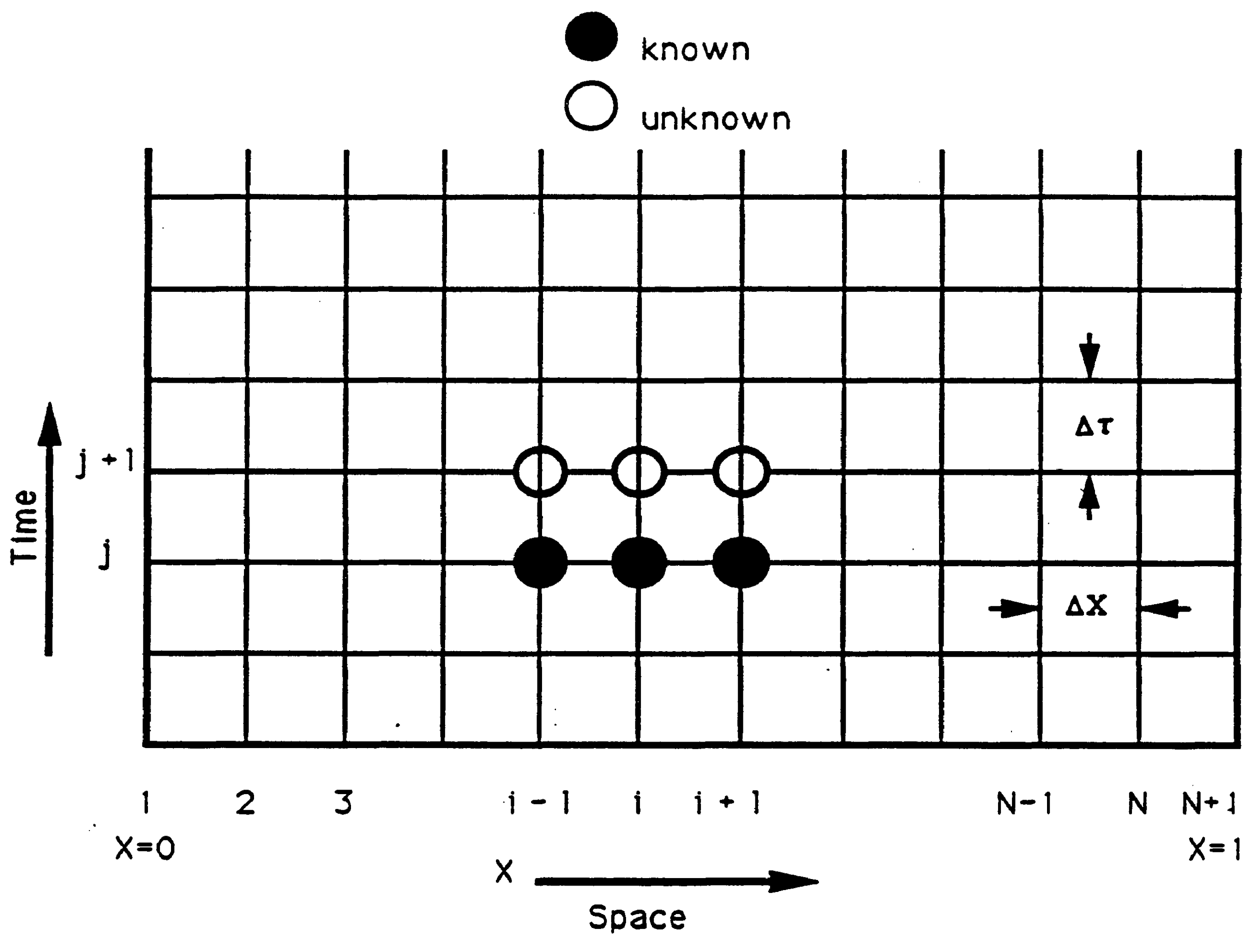

Figure III.2 **Graphical Description of the Crank-Nicholson Finite Difference Scheme**

$$\frac{U_{N+2,j} - U_{N,j}}{2\Delta X} = G_1 \qquad i = N+1 \qquad \text{(III.12)}$$

$$U_{i,0} = T_0 \qquad i = 1,\ldots N+1 \qquad \text{(III.13)}$$

Equations (III.11) and (III.12) may be rearranged to obtain expressions for the imaginary points $U_{0,j}$ and $U_{N+2,j}$.

$$U_{0,j} = U_{2,j} - 2\Delta X\ G_0 \qquad \text{(III.14)}$$

$$U_{N+2,j} = U_{N,j} + 2\Delta X\ G_1 \qquad \text{(III.15)}$$

These equations may be also written at the time level j+1 to give:

$$U_{0,j+1} = U_{2,j+1} - 2\Delta X\ G_0 \qquad \text{(III.16)}$$

$$U_{N+2,j+1} = U_{N,j+1} + 2\Delta X\ G_1 \qquad \text{(III.17)}$$

Equation (III.10) may be rearranged as follows:

$$-\lambda U_{i+1,j+1} + 2(1+\lambda)U_{i,j+1} - \lambda U_{i-1,j+1} =$$

$$\lambda U_{i+1,j} + 2(1-\lambda)U_{i,j} + \lambda U_{i-1,j} \qquad i=2,3\ldots N \qquad \text{(III.18)}$$

where $\lambda = \Delta\tau/\Delta X^2$. With $U_{i,j}$ (i=1,2...N+1) known at the time level j, Equation (III.18) represents N-1 equations for the unknown temperature $U_{i,j+1}$ (i=2,3..N). Equation (III.18) is written at the grid points i=1 and i=N+1 and then combined with Equations (III.14) through (III.17) to give:

$$2(1+\lambda)U_{1,j+1} - 2\lambda U_{2,j+1} = 2(1-\lambda)U_{1,j} + 2\lambda U_{2,j} - 4\lambda\Delta X\, G_0 \qquad \text{(III.19)}$$

$$-2\lambda U_{N,j+1} + 2(1+\lambda)U_{N+1,j+1} = 2\lambda U_{N,j} + 2(1-\lambda)U_{N+1,j} + 4\lambda\Delta X G_1 \qquad \text{(III.20)}$$

With $U_{i,j}$ (i=1,2,...N+1) known at the time level j, Equations (III.18), (III.19), and (III.20) yield N+1 equations for the unknown temperatures $U_{i,j+1}$ (i=1,2..N+1) and may be represented by the following equation:

$$A\, U_{j+1} = B \qquad \text{(III.21)}$$

where, A is the following tridiagonal matrix:

$$A = \begin{bmatrix} (1+\lambda) & -\lambda & & & & \\ -\lambda & 2(1+\lambda) & -\lambda & & & \\ & -\lambda & 2(1+\lambda) & -\lambda & & \\ & & -\lambda & 2(1+\lambda) & -\lambda & \\ & & & & \vdots & \\ & & & & -\lambda & (1+\lambda) \end{bmatrix}$$

and B is the following column vector:

$$B = \begin{bmatrix} (1-\lambda)U_{1,j}+\lambda U_{2,j}-2\lambda\Delta X\ G_0 \\ \lambda U_{2,j}+2(1-\lambda)U_{1,j}+\ \lambda U_{0,j} \\ \lambda U_{3,j}+2(1-\lambda)U_{2,j}+\ \lambda U_{1,j} \\ \lambda U_{4,j}+2(1-\lambda)U_{3,j}+\ \lambda U_{1,j} \\ \vdots \\ \lambda U_{N,j}\ +\ (1-\lambda)U_{N+1,j}\ +\ 2\lambda\Delta X\ G_1 \end{bmatrix},$$

$$U_{j+1} = \begin{bmatrix} U_{1,j+1} \\ U_{2,j+1} \\ U_{3,j+1} \\ \vdots \\ U_{N+1,j+1} \end{bmatrix}$$

<u>Compuational Algorithm</u>

A FORTRAN 77 computer program was written which implements the finite-difference equations described previously. All calculations were performed on an IBM PC/AT clone in double-precision arithmetic. The computer programs for the computations are included in Appendix A. The system consists of a main program and several subroutines which calculate the exact solution, the numerical solution and an actual error analysis. The LAHEY compiler (Version 3.1) was used for these FORTRAN programs. The computational algorithm is described as follows:

(1) Computations for the heating period are obtained by the solution of the set of simultaneous equations given by Equation (III.21). This is done by using the Thomas Algorithm. Since the Crank-Nicholson finite difference method is stable for all finite values of λ ($\lambda = \Delta\tau/(\Delta X^2)$, several values of λ were evaluated to determine which λ produced the smallest actual errors. The algorithm can continue indefinitely but a finite number of time steps were used when performing the actual error analysis. The term 'actual error' is defined in table III.1. The boundary conditions are given in function format so they can easily be modified as the problem becomes more complex. The results presented below are given for the case where $G_0 = -1$ and $G_1 = 0$. This corresponds to the situation where a constant heat flux q_0 is applied to the surface of the film at x=0 while the boundary at $x=\ell$ is kept insulated. An analytical solution for this case was presented in Chapter II. The numerical results obtained from the finite difference solution are compared next with the results from the analytical solution.

Error Analysis

Several values of λ were used to determine the appropriate combination of ΔX and $\Delta\tau$ which would yield acceptable accuracy. In order to change λ, ΔX was fixed at 1/32 or 0.03125 and $\Delta\tau$ was modified. Table (III.1) gives the results of the error analysis calculations. In addition to Table (III.1), Figures (III.3) through (III.6) show the temperature distribution in the polymer film as computed from both the numerical and analytical solutions. As can be seen by the table the absolute differences between the analytical and numerical solutions range from 6.77X10-3 to 1.51X10-7. A relative error analysis was not possible because there were several occasions when division by zero occured. The relative error is defined as:

(Actual value - Numerical Value)/ Actual value X 100%

In several cases the actual value was zero causing a division by zero which is not possible. Figures (III.7) thru (III.10) show the effects of the different λ's at a specific time (t=4.75X10-3 sec). From the figures, every value of λ looks reasonable. However, even though the Crank-Nicholson finite difference scheme is very stable for a large range of λ's, the errors increase with increasing λ. Table (III.1) shows actual errors for λ values of 2.0, 5.0,

Table III.1
Actual Error Analysis

Time(Sec)	X=0.1875	X=0.5625	X=0.9375	Lambda
5.2d-5	.236d-3	.201d-4	.151d-6	2
	.276d-3	.526d-4	.774d-6	5
	.135d-2	.156d-3	.373d-5	10
	.677d-2	.417d-3	.197d-4	20
9.896d-4	.971d-4	.777d-4	.628d-4	2
	.945d-4	.783d-4	.659d-4	5
	.857d-4	.805d-4	.767d-4	10
	.858d-4	.890d-4	.120d-3	20
1.354d-3	.866d-4	.801d-4	.752d-4	2
	.857d-4	.804d-4	.763d-4	5
	.824d-4	.811d-4	.801d-4	10
	.422d-4	.843d-4	.955d-4	20
1.875d-3	.824d-4	.811d-4	.802d-4	2
	.822d-4	.812d-4	.804d-4	5
	.815d-4	.813d-4	.812d-4	10
	.741d-4	.819d-4	.843d-4	20
2.6d-3	.815d-4	.813d-4	.813d-4	2
	.815d-4	.814d-4	.813d-4	5
	.814d-4	.814d-4	.814d-4	10
	.835d-4	.814d-4	.817d-4	20

X is the distance into the toner, where X=0 is the surface and X=1 is the paper-toner interface. Lambda = $(\Delta t/ \Delta X^{**}2)$.

Actual error is defined as:

$$|\text{Actual value} - \text{Numerical value}| = \text{Table value}$$

Actual and Numerical values above are Temperature values.

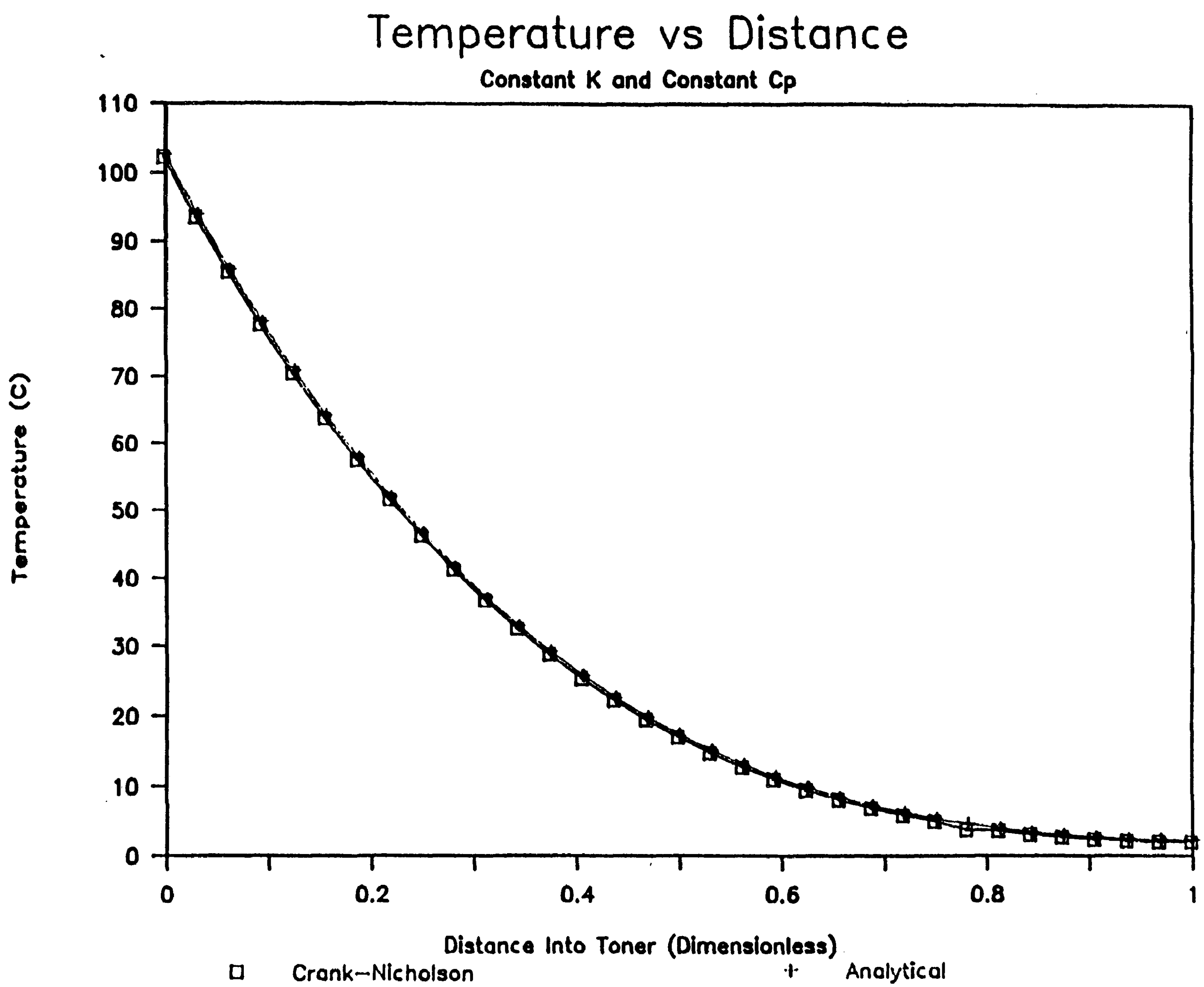

Figure III.3: Analytical vs Numerical Results, $\lambda=2$, t=3.84X10-4 sec

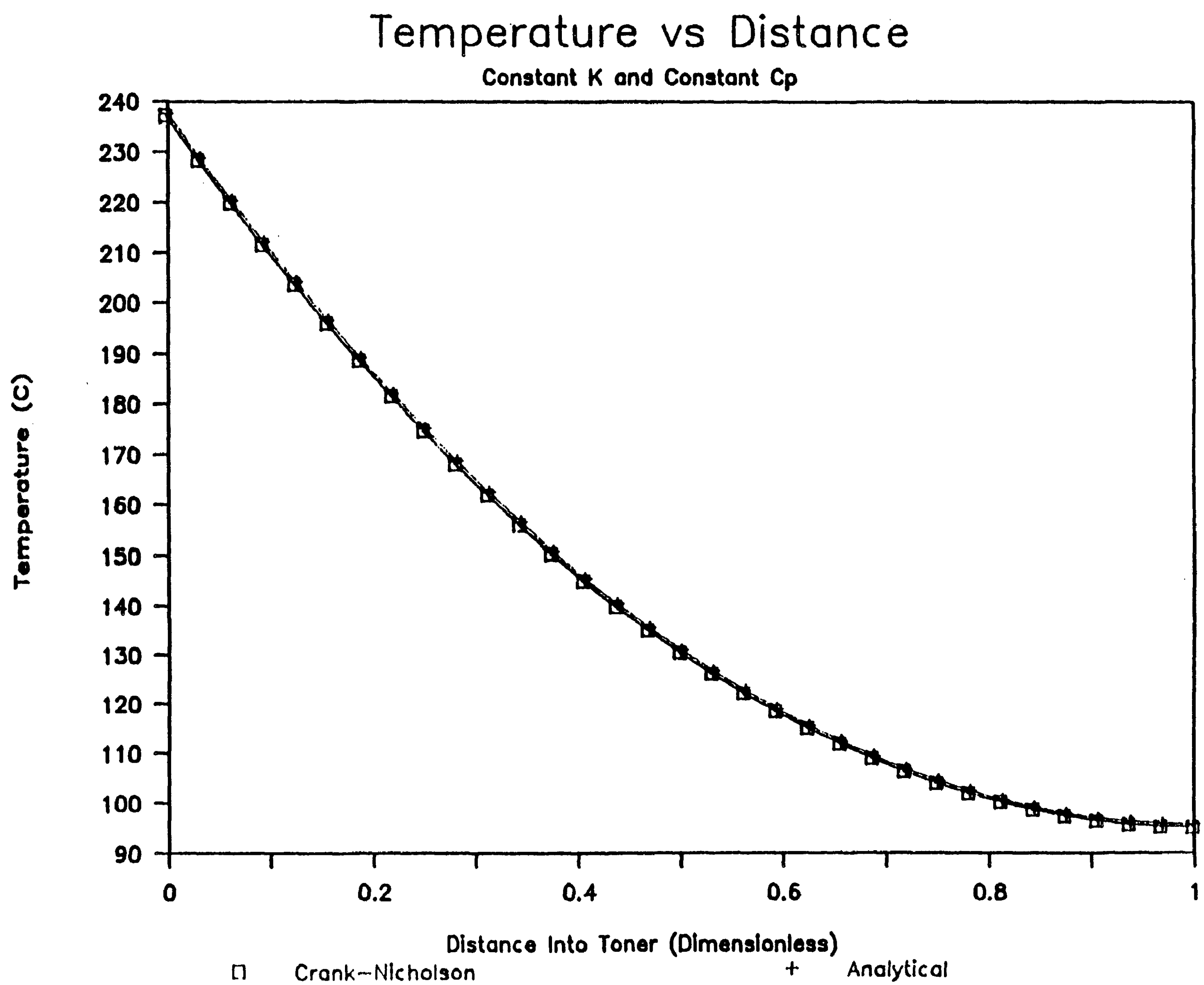

Figure III.4: Analytical vs Numerical Results, λ=2, t=1.9X10-3 sec

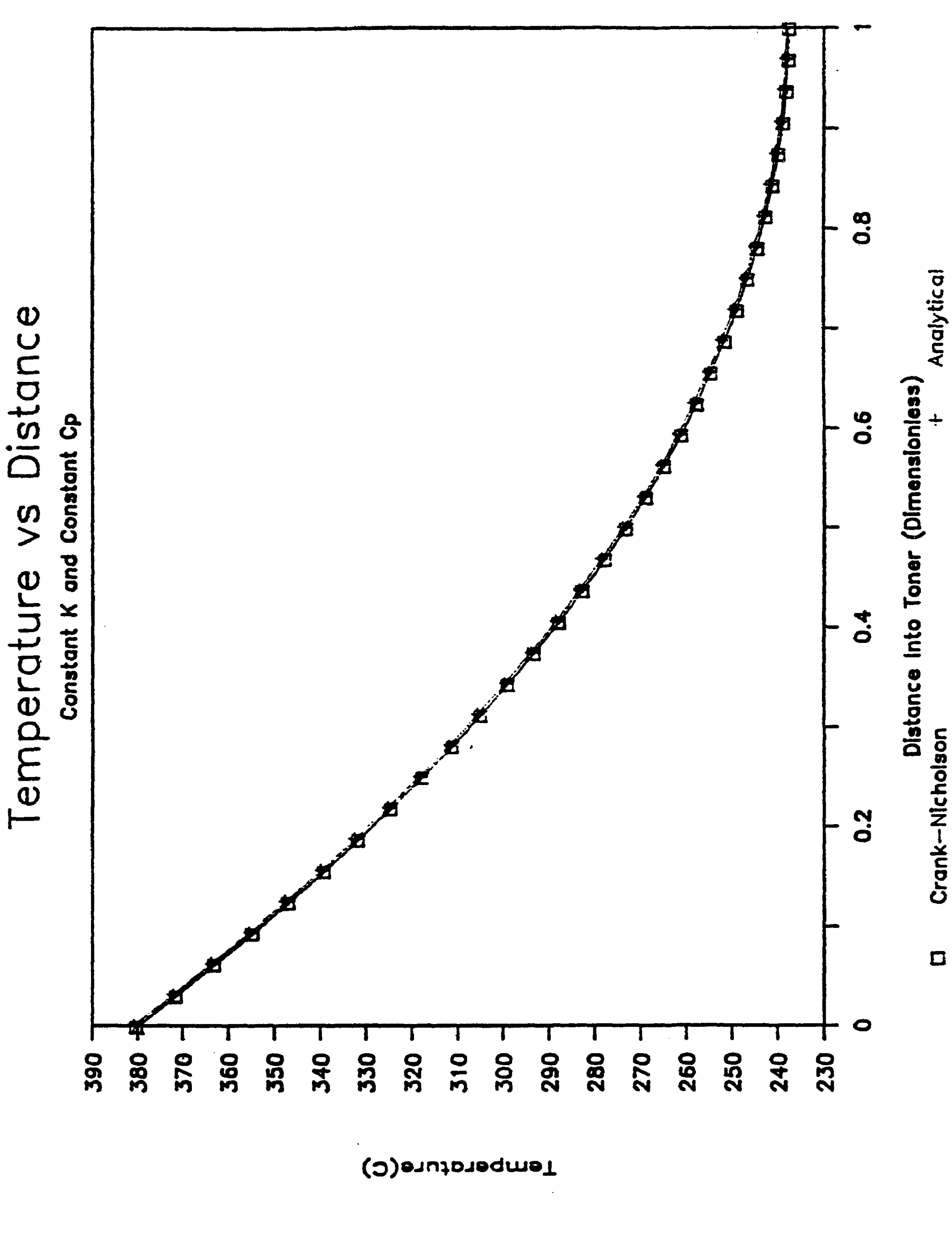

Figure III.5: Analytical vs Numerical Results, $\lambda=2$, t=3.8X10-3 sec

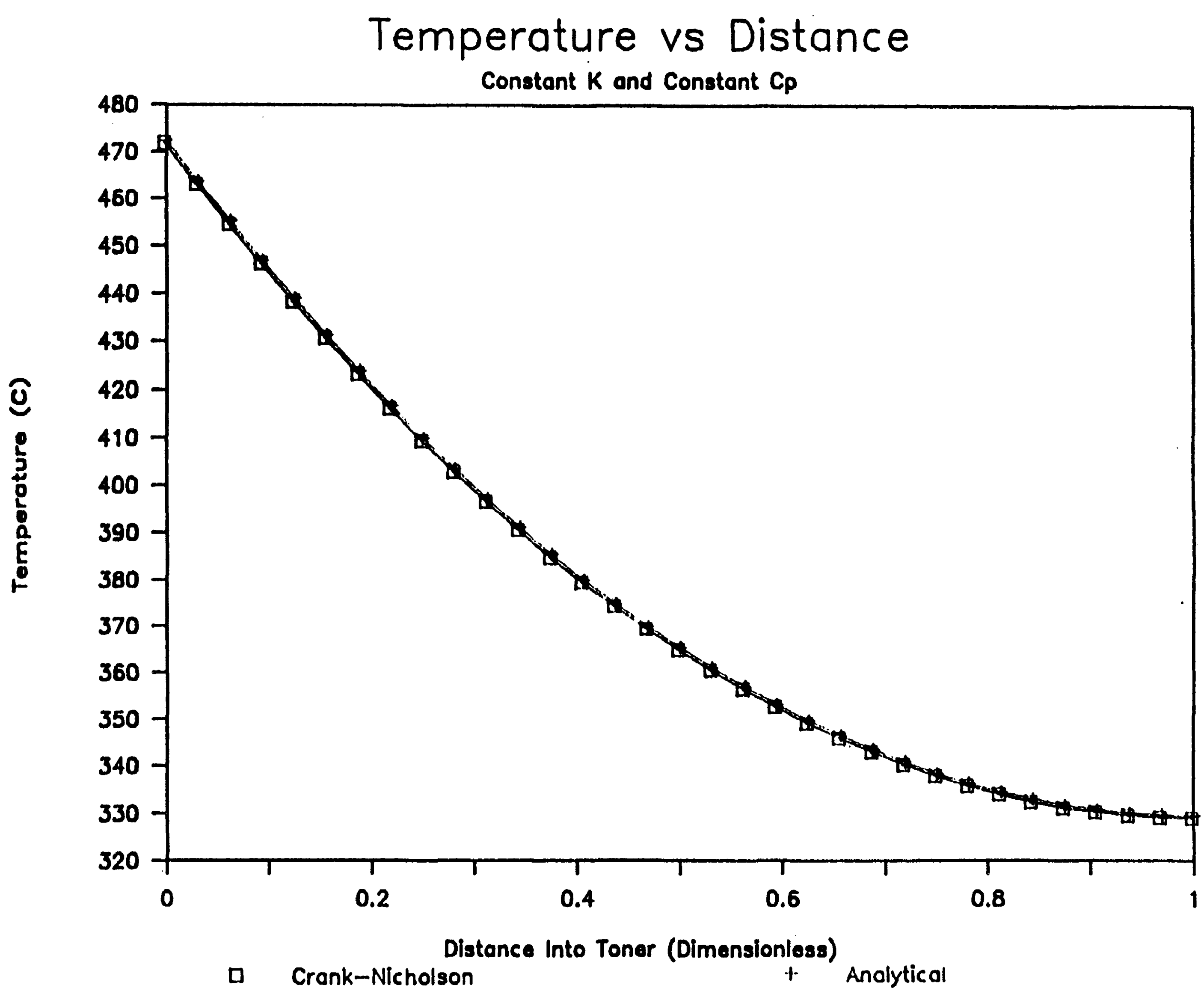

Figure III.6: Analytical vs Numerical Results, λ=2, t=5.0X10-3 sec

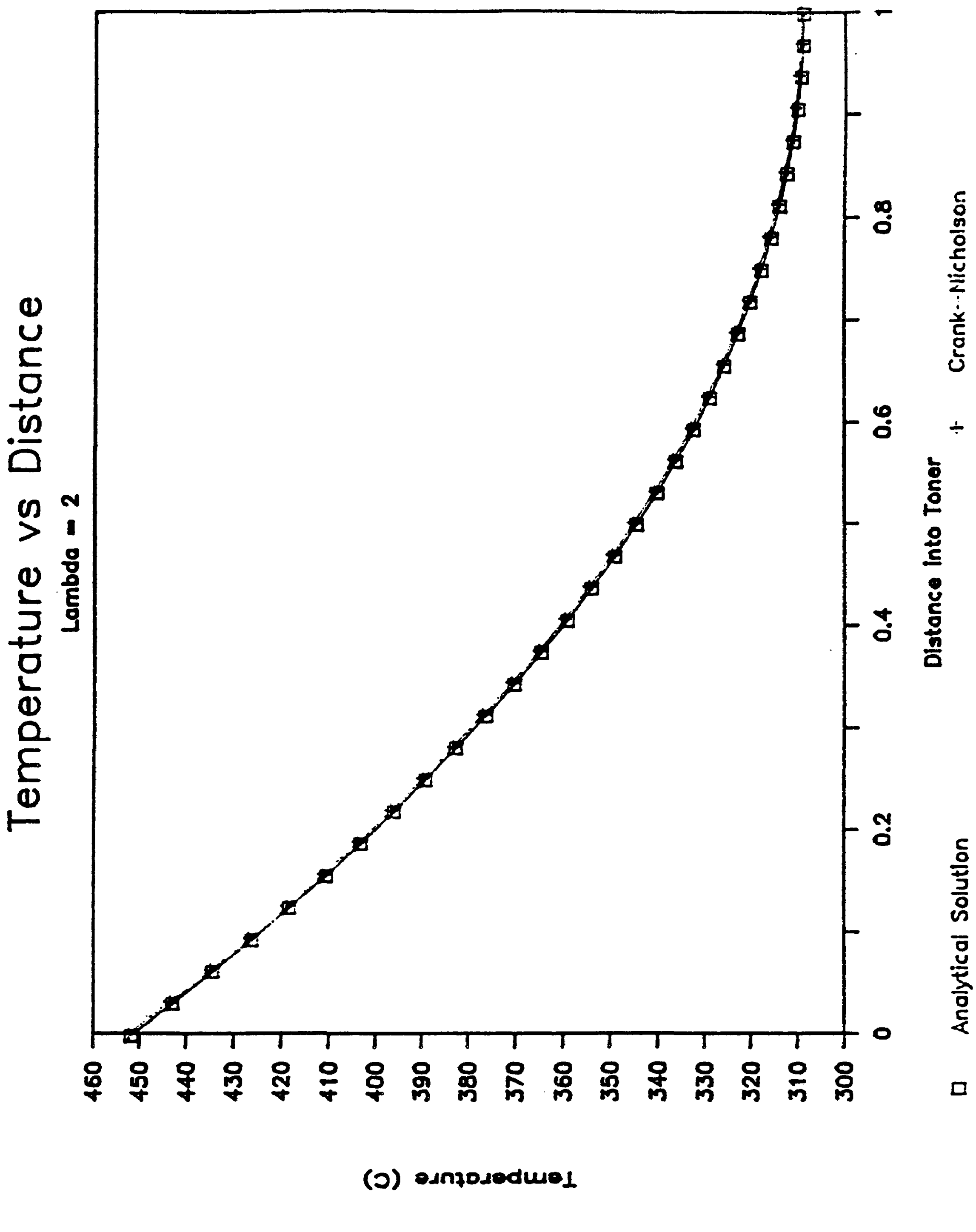

Figure III.7: Analytical vs Numerical Results, λ=2, t=4.75X10-3 sec

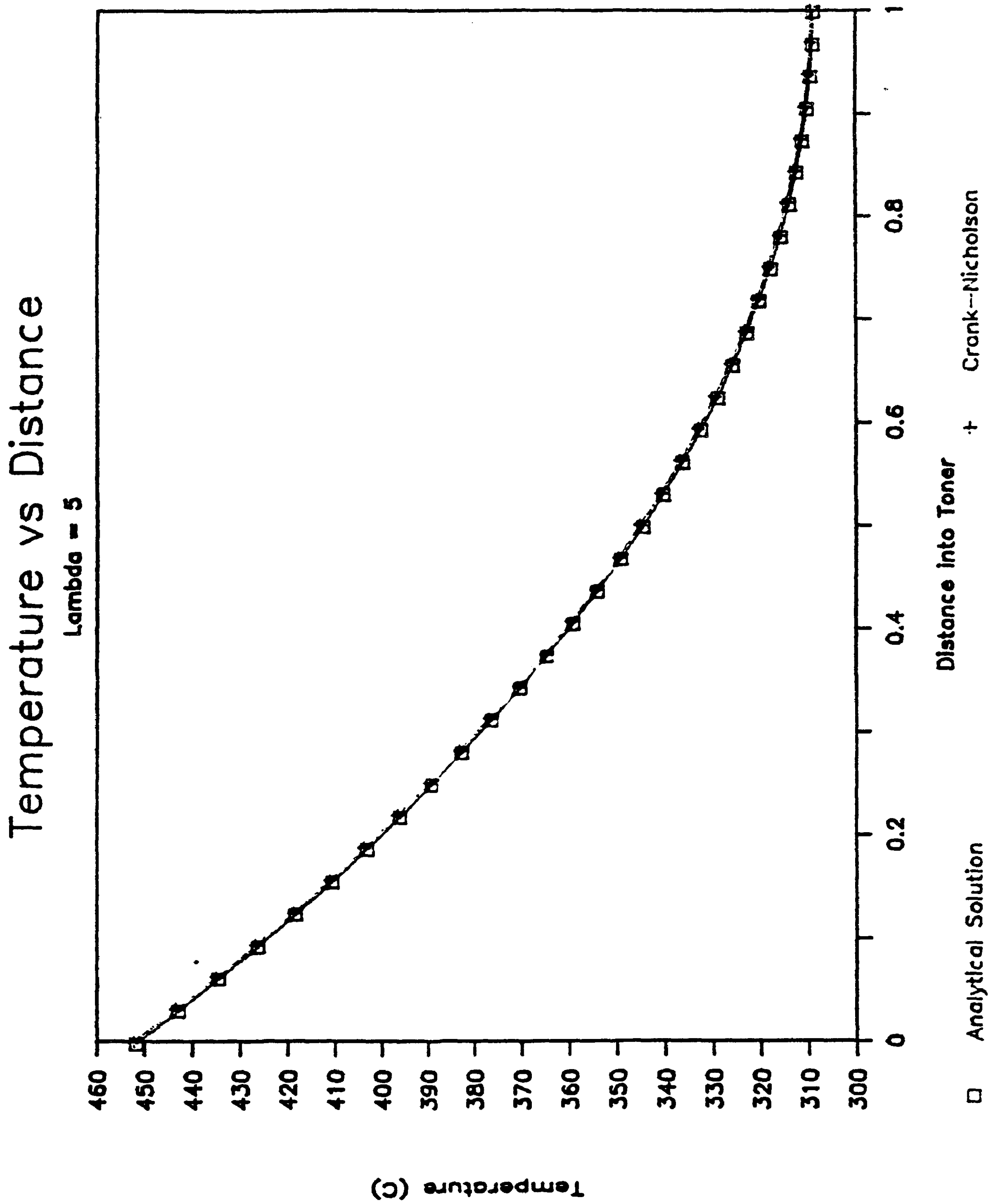

Figure III.8: Analytical vs Numerical Results, λ=5, t=4.75X10-3 sec

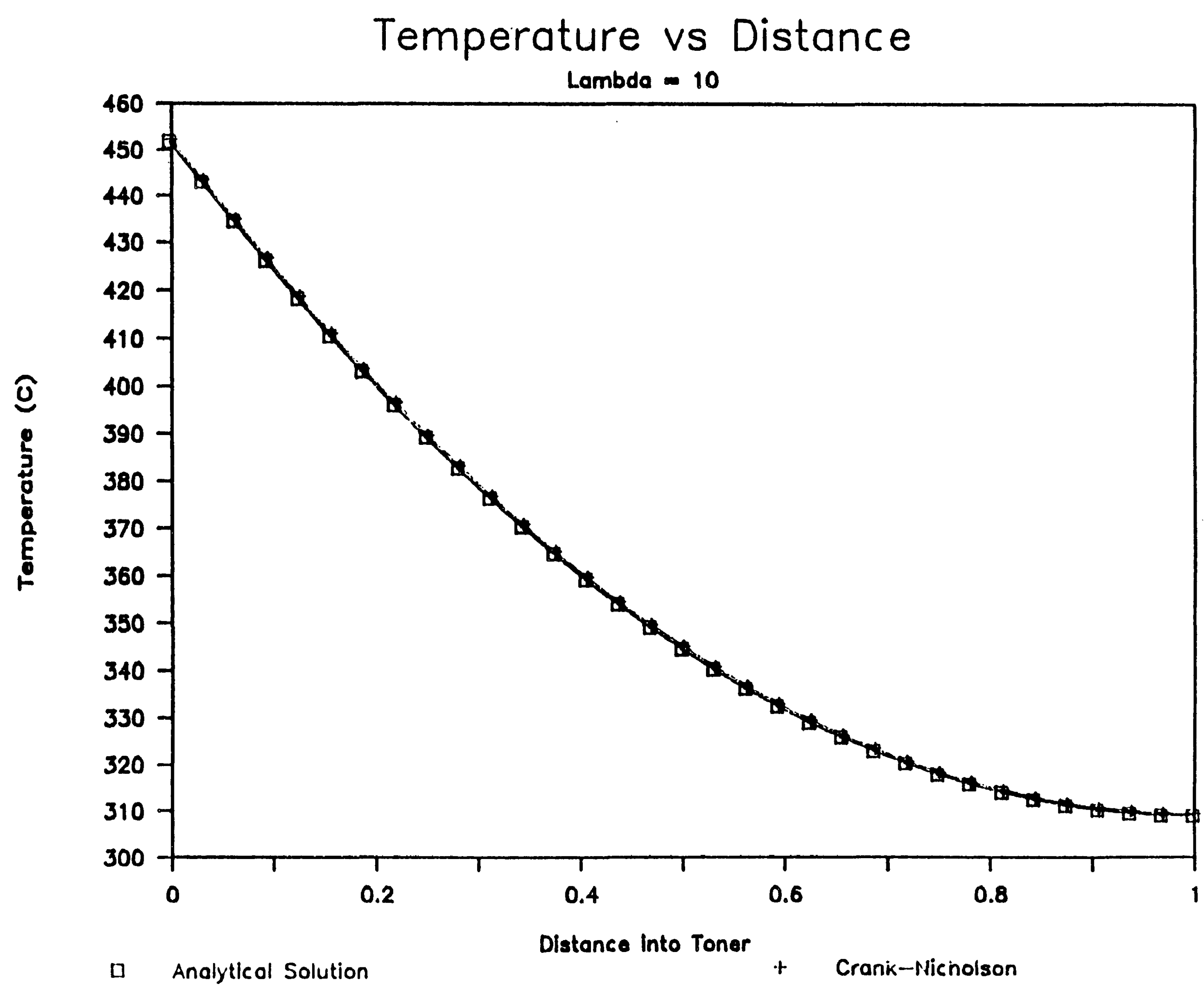

Figure III.9: Analytical vs Numerical Results, λ=10, =4.75X10-3 sec

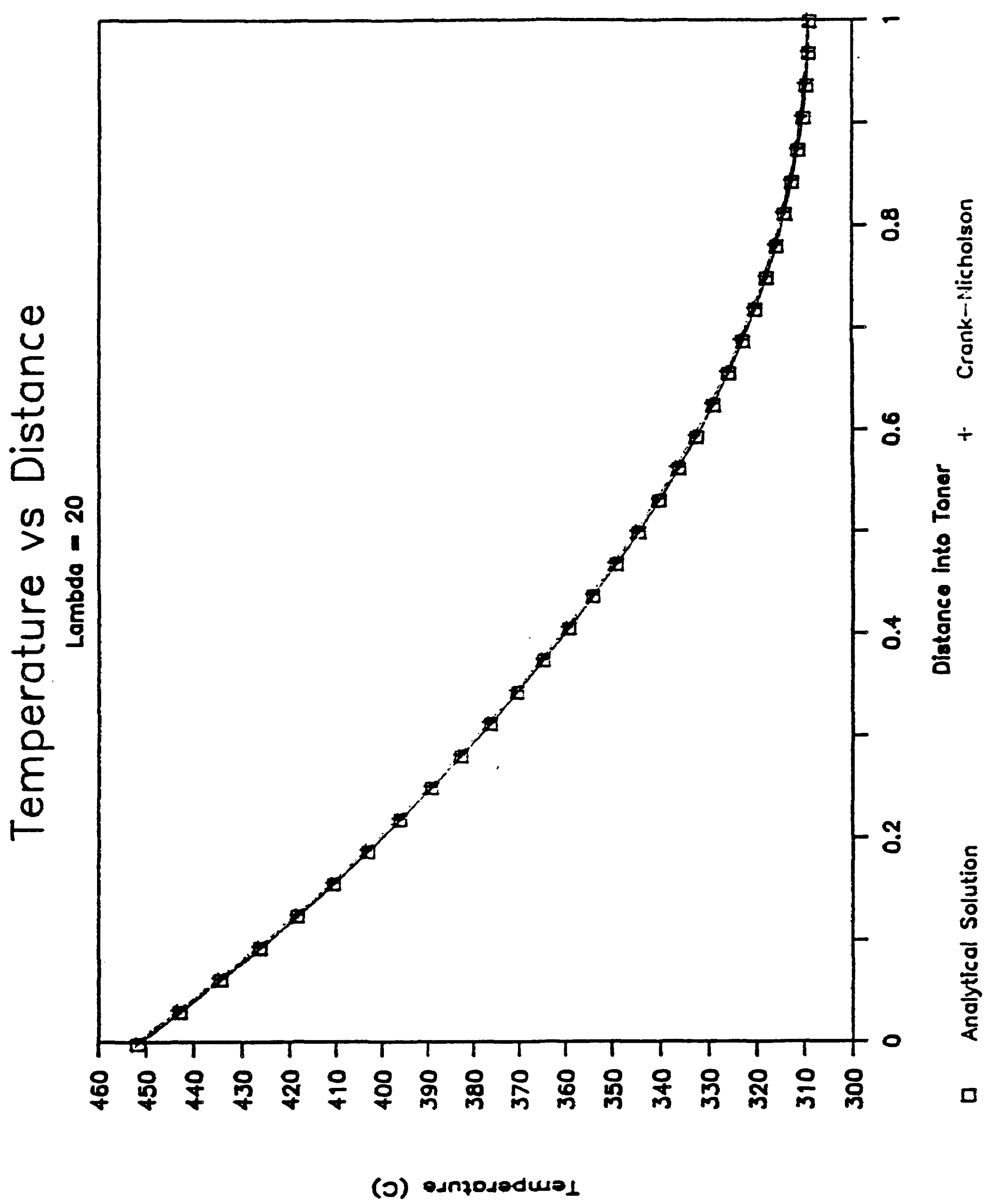

Figure III.10: Analytical vs Numerical Results, λ=20, t=4.75X10-3 sec

ARTHUR LAKES LIBRARY
COLORADO SCHOOL of MINES
GOLDEN, COLORADO 80401

10.0 and 20.0 for different positions and times throughout the polymer film. The smallest errors are with a λ of 2.0.For the purpose of this work, λ was chosen as 2.0, which sets $\Delta\tau$ at 1.953e-3. The time range under consideration is from 0-5 milliseconds. Although small, this $\Delta\tau$ is acceptable due to the time factor. The number of spatial steps, N, was chosen as 32 in order to minimize round-off error. When using a step size of 1/64, the accuracy did not change significantly when compared with the results using a step size of 1/32.

Chapter IV
Heat Conduction in a Polymer Film with Variable Thermal Conductivity

Consider the heating of a polymer film occupying the region $0 \le x \le \ell$ and initially at a uniform temperature T_i. At time t=0, the film is heated by applying a constant heat flux q_0 at the surface. The film is heated until a time $t=t_f$. Previously, both the thermal conductivity, k and heat capacity c_p were assumed constant. In the present case the heat capacity is kept constant while thermal conductivity is allowed to vary with temperature. The appropriate differential equations and associated boundary and initial conditions are given by:

$$\rho_0 c_{p0} \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k(T) \frac{\partial T}{\partial x} \right) \qquad 0<x<\ell,\ t_f>t>0 \qquad \text{(IV.1)}$$

$$-k(T) \frac{\partial T}{\partial x} = q_0 \qquad x=0,\ t_f>t>0 \qquad \text{(IV.2)}$$

$$-\ k(T) \frac{\partial T}{\partial x} = q_\ell \qquad x=1,\ t_f>t>0 \qquad \text{(IV.3)}$$

$$T = T_i \qquad 0<x<\ell,\ t=0 \qquad \text{(IV.4)}$$

This problem is cast into a more convenient form by defining the following dimensionless variables:

$$U = \frac{T - T_i}{q_0\ (\ell/k_0)}\ , \quad X = \frac{x}{\ell}\ , \quad \tau = \frac{\alpha_0 t}{\ell^2}\ ,$$

and

$$K(T) = \frac{k(T)}{k_0}\ , \quad \text{where} \quad \alpha_0 = \frac{k_0}{\rho_0 c_{p0}} \qquad \text{(IV.5)}$$

where k_0, ρ_0, c_{p0}, and α_0 are the thermophysical properties at T_i.

The problem then runs as follows:

$$\frac{\partial U}{\partial \tau} = \frac{\partial}{\partial X}\left(K(U)\,\frac{\partial U}{\partial X} \right) \qquad 0< X < 1,\ \tau >0 \qquad \text{(IV.6)}$$

$$\frac{\partial U}{\partial X} = -\frac{G_0}{K(U)} \qquad X = 0,\ \tau>0 \qquad \text{(IV.7)}$$

$$\frac{\partial U}{\partial X} = \frac{G_1}{K(U)} \qquad X = 1,\ \tau>0 \qquad \text{(IV.8)}$$

$$U = 0 \qquad 0<X<1,\ \tau=0 \qquad \text{(IV.9)}$$

where $G_0 = q_0/ q_0$ and $G_1 = q_\ell / q_0$.

As discussed in Chapter 2 the Crank-Nicholson finite difference scheme is used to solve equations (IV.6) - (IV.9). In using this scheme, the analogs to the space derivatives are centered about the time level $t_{j+1/2}$. In addition, the first space derivative is approximated at $X_{i+1/2}$ and $X_{i-1/2}$ by the analogs over one space increment. The conductivity, K(U), must also be evaluated at these same points. The heat capacity and the density are held constant throughout this derivation. The details of this derivation are outlined in Rosenberger (6). The resulting equations are

$$\frac{U_{i,j+1} - U_{i,j}}{\Delta\tau} = \frac{K_{+1/2}}{2(\Delta X)^2}\left[U_{i+1,j+1} - U_{i,j+1} + U_{i+1,j} - U_{i,j}\right]$$

$$+ \quad \frac{K_{-1/2}}{2(\Delta X)^2}\left[U_{i,j+1} - U_{i-1,j+1} + U_{i,j} - U_{i-1,j}\right] \qquad \text{(IV.10)}$$

where,

$$K_{-1/2} = \frac{\left[\frac{K_{i,j} + K_{i-1,j}}{2}\right] + \left[\frac{K_{i,j+1} + K_{i-1,j+1}}{2}\right]}{2} \qquad \text{(IV.11)}$$

$$K_{+1/2} = \frac{\left[\frac{K_{i+1,j} + K_{i,j}}{2}\right] + \left[\frac{K_{i+1,j+1} + K_{i,j+1}}{2}\right]}{2} \qquad \text{(IV.12)}$$

We may note that the quantities $K_{-1/2}$ and $K_{+1/2}$ represent $K_{i-1/2,j+1/2}$ and $K_{i+1/2,j+1/2}$, respectively.

The central difference representation of the boundary conditions at X = 0 (i=1) and X = 1 (i=N+1), and the initial condition are given by:

$$\frac{K(U)_{2,j}U_{2,j} - K(U)_{0,j}U_{0,j}}{\Delta X} = G_0 \qquad i = 1 \qquad \text{(IV.13)}$$

$$\frac{K(U)_{N+2,j}U_{N+2,j} - k(U)_{N,j}U_{N,j}}{\Delta X} = G_1 \qquad i = N+1 \qquad \text{(IV.14)}$$

$$U_{i,0} = T_0 \qquad i = 1,\ldots N+1 \qquad \text{(IV.15)}$$

Equations (IV.14) and (IV.15) may be rearranged to obtain expressions for both imaginary points $U_{0,j}$ and $U_{N+2,j}$.

$$U_{0,j} = \frac{K(U)_{2,j}U_{2,j} + G_0\ \Delta X}{K(U)_{0,j}} \qquad \text{(IV.16)}$$

$$U_{N+2,j} = \frac{K(U)_{N,j}U_{N,j} - G_1\ \Delta X}{K(U)_{N+2,j}} \qquad \text{(IV.17)}$$

These equations may be also written at the time level j+1 to give:

$$U_{0,j+1} = \frac{K(U)_{2,j+1}U_{2,j+1} + G_0\Delta X}{K(U)_{0,j+1}} \qquad \text{(IV.18)}$$

$$U_{N+2,j+1} = \frac{K(U)_{N,j+1}U_{N,j+1} - G_1\ \Delta X}{K(U)_{N+1,j+1}} \qquad \text{(IV.19)}$$

Equation (IV.10) may be rearranged as follows:

$$-K_{-1/2}U_{i-1,j+1} + \left[\frac{2}{\lambda} + K_{+1/2} + K_{-1/2}\right]U_{i,j+1} - K_{+1/2}\ U_{i+1,j+1}$$

$$= K_{-1/2}\ U_{i-1,j} + \left[\frac{2}{\lambda} - K_{+1/2} - K_{-1/2}\right]U_{i,j} + K_{+1/2}\ U_{i+1,j} \qquad \text{(IV.20)}$$

where $\lambda = \Delta\tau/\Delta X^2$. Equation (IV.20) is written at the grid points i=1 and i=N+1 and then combined with Equations (IV.16) through (IV.19) to give:

$$\left[\frac{2}{\lambda} + K_{+1/2} + K_{-1/2}\right]U_{1,j+1} - \left[\frac{K_{-1/2}\,K(U)_{2,j+1}}{K_{0,j+1}} + K_{+1/2}\right]U_{2,j+1}$$

$$= \left[\frac{2}{\lambda} - K_{+1/2} - K_{-1/2}\right]U_{1,j} + \left[\frac{K_{-1/2}\,K(U)_{2,j}}{K_{0,j}} + K_{+1/2}\right]U_{2,j}$$

$$- 2K_{-1/2}\,G_0\,\Delta X\left[\frac{1}{K_{0,j}} + \frac{1}{K_{0,j+1}}\right] \qquad i=1 \qquad \text{(IV.21)}$$

$$- \left[\frac{K_{+1/2}\,K_{N-1,j+1}}{K_{N+1,j+1}} + K_{-1/2}\right]U_{N-1,j+1} +$$

$$\left[\frac{2}{\lambda} + K_{+1/2} + K_{-1/2}\right]U_{N,j+1} = \left[\frac{K_{+1/2}\,K_{N-1,j}}{K_{N+1,j}} + K_{-1/2}\right]U_{N-1,j} +$$

$$\left[\frac{2}{\lambda} - K_{+1/2} - K_{-1/2}\right]U_{N,j} - 2K_{+1/2}\,G_1\,\Delta X\left[\frac{1}{K_{N+1,j+1}} + \frac{1}{K_{N+1,j}}\right]$$

$$i=N+1 \qquad \text{(IV.22)}$$

With $U_{i,j}$ (i=1,2,...N+1) known at the time level j, Equations (IV.20), (IV.21), and (IV.22) represent N+1 equations for the unknown temperatures $U_{i,j+1}$ (i=1,2,...N+1). The computer program developed in Chapter 2 was modified for the solution of these equations. We may note that in computing the unknown temperatures $U_{i,j+1}$ from these equations, knowledge of the K values at the j+1 level is required. Accordingly, the following iterative scheme was adopted:

(i) With $U_{i,j}$ known at the j time step, the K values were first computed from

$$K_{-1/2} = \frac{K_{i,j} + K_{i-1,j}}{2}, \qquad K_{+1/2} = \frac{K_{i+1,j} + K_{i,j}}{2}$$

(ii) With the $U_{i,j}$'s known and the K's known from step (i), the $U_{i,j+1}$'s were calculated from Equations (IV.20) - (IV.22).

(iii) With $U_{i,j+1}$'s known, the K's were recalculated from Equations (IV.11) - (IV.12).

(iv) With the $U_{i,j}$'s known and the K's calculated in step (iii), the $U_{i,j+1}$'s were recomputed.

Steps (iii) and (iv) were repeated until the values of the $U_{i,j+1}$'s from two successive iteration differ by a small tolerance. The latter was set equal to 10^{-7}. This required on the average three to four iterations to reach convergence at each time step.

In order to access the effect of variable thermal conductivity on the temperature distribution in the polymer film, the following values for the physical properties were taken:

ρ_0 = 1000 kg/m^3

c_{p0} = 2000 J/kg C

k_0 = 0.21 J/m-sec C (Constant K case)

$k(T) = k_0[\ 1 + b(T - T_i)]$ where T is in degrees C

The value for k_0 was from IBM data(7). The values for ρ_0 and C_{p0} are representative values from literature(8).

So,

$k(T) = 0.21[\ 1 + b(T - 0)]$

In addition, the following values were taken for T_0, q_0, and q_ℓ:

T_0 = 0

q_0 = 3.0X10+06 J/m^2-sec

q_ℓ = 0.0

ℓ = 20X10-06 meters

Again, the values for q_0 and ℓ were from IBM(7). Note that the heat flux at the surface X=0 was taken as a constant. The flux at the surface X=ℓ, was set to zero due to the boundary condition at that point. The initial temperature was set at 0^{o}C for the calculations. Realizing the initial temperature for most copiers would be set at 25^{o}C the results shown in this chapter and subsequent chapters must have 25^{o}C added to the temperatures.

A representative polymer was used to show the behavior of a constant value of K vs a variable value of K. Polyethylene with a crystalline content as shown by a density of 0.948 is used in the comparison. The physical properties used for polyethylene were as follows:

ρ_0 = 1000 kg/m^3

C_{p0} = 2000 J/kg c

K(T) = 1 - 2.46914X10-3(T - 0) where K(T) = k(T)/k_0(T) and T is in degrees C.

Also, the following values were taken for T_0, q_0, q_ℓ, and ℓ:

T_i = 0.0

q_0 = q_0 = 3.0X10+6 J/m^2-sec

$q\ell$ = $q\ell$ = 0.0

ℓ = 20X10-06 meters

When the variable K case was calculated, there were very small differences between the constant K case and the polymer. Figures (IV.1) and (IV.4) show differences of 1-2^{o}C between the constant K case and the variable K case.. This is to be expected as the values of K of most common polymers

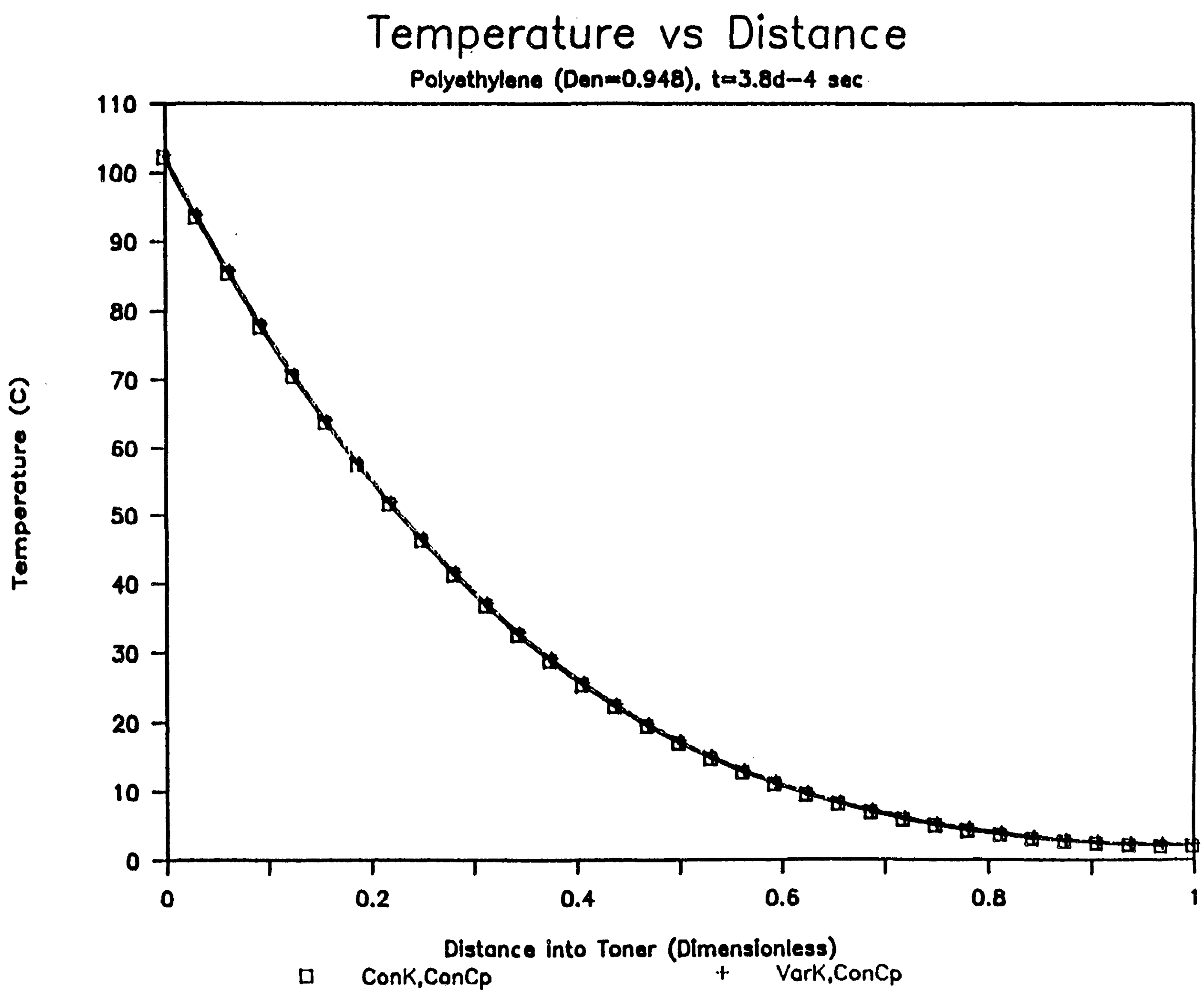

Figure IV.1: Constant vs Variable K for Polyethylene, t=3.8X10-4 sec

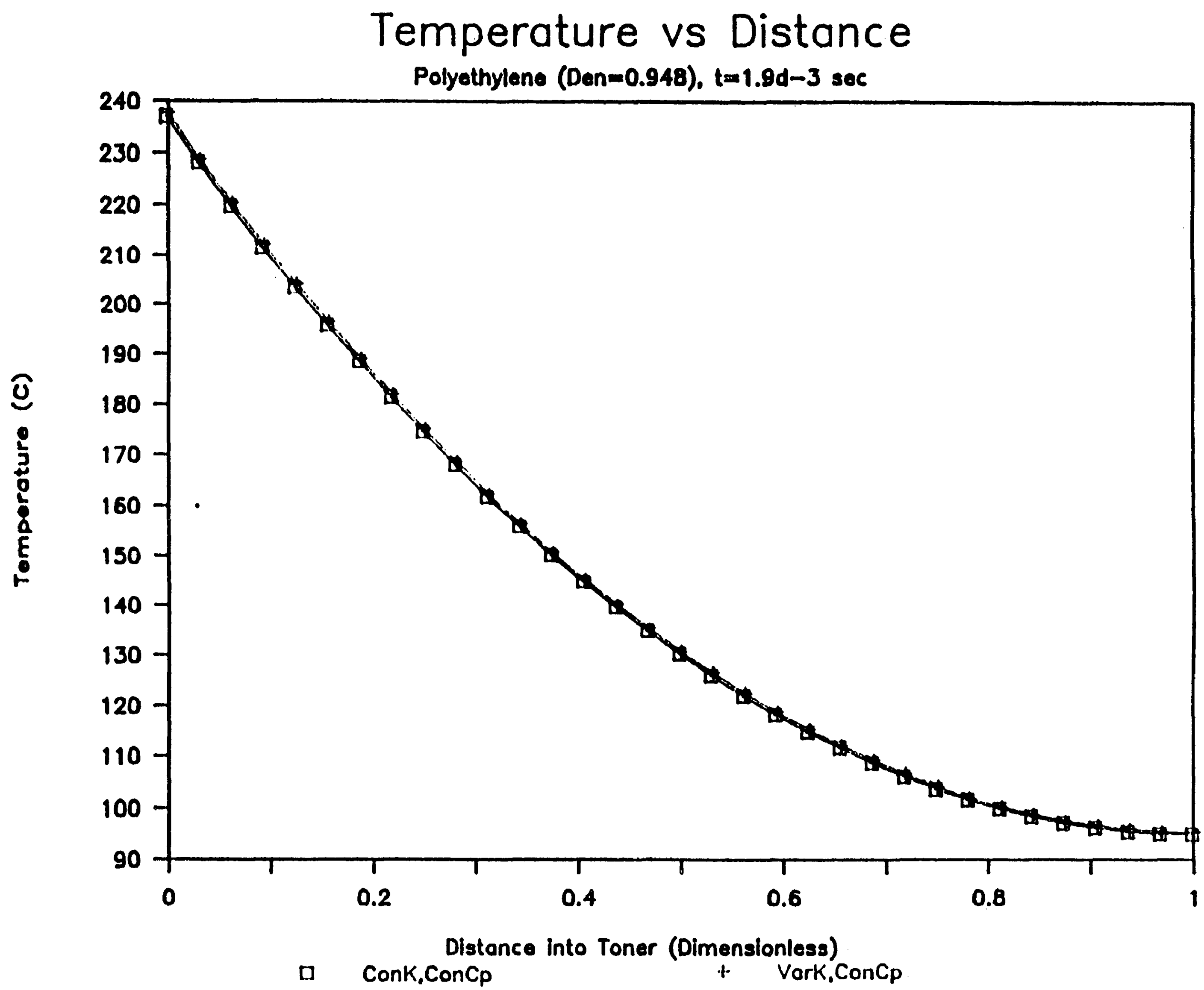

Figure IV.2: Constant vs Variable K for Polyethylene, t=1.9x10-3 sec

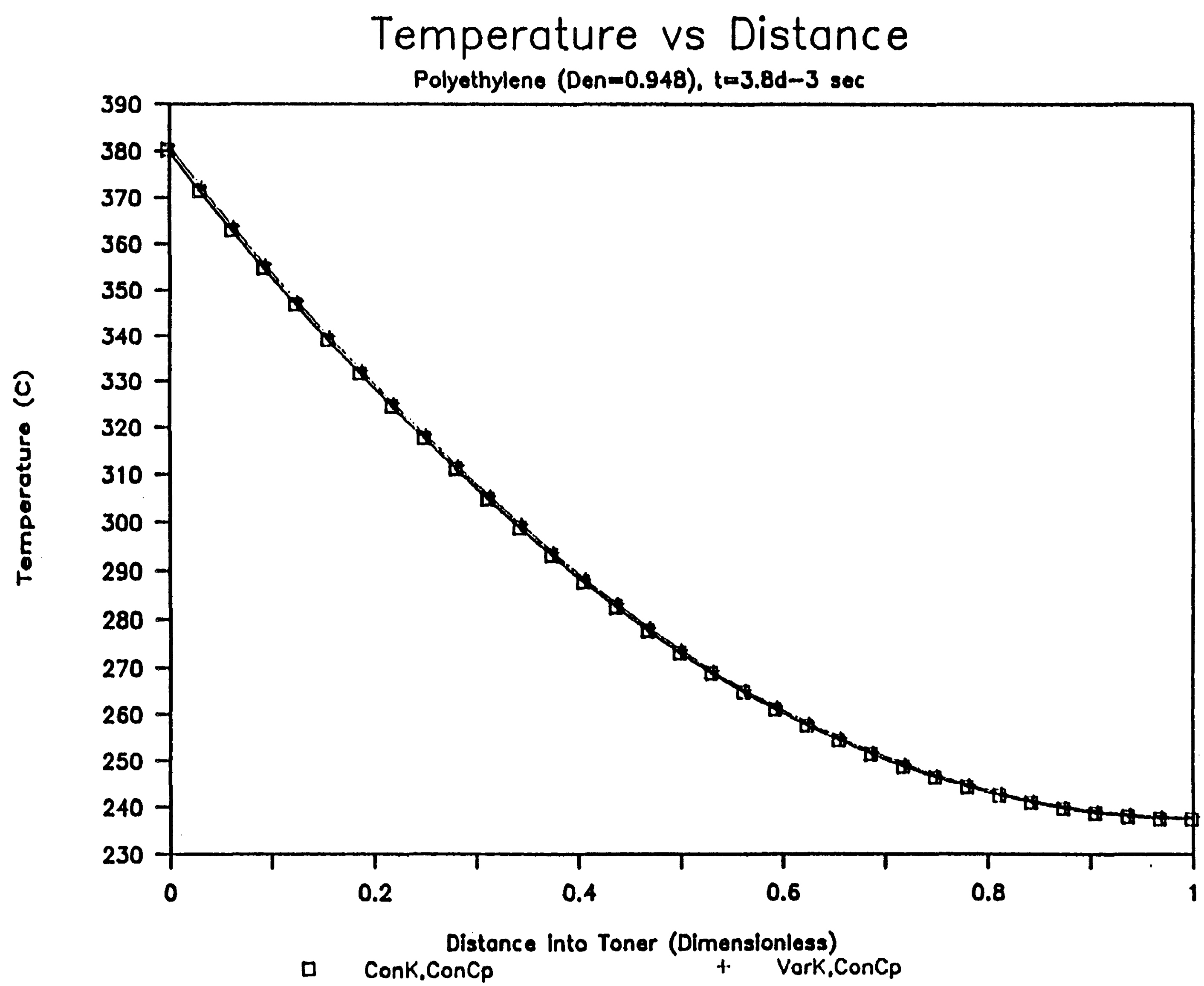

Figure IV.3: Constant vs Variable K for Polyethylene, t=3.8X10-3 sec

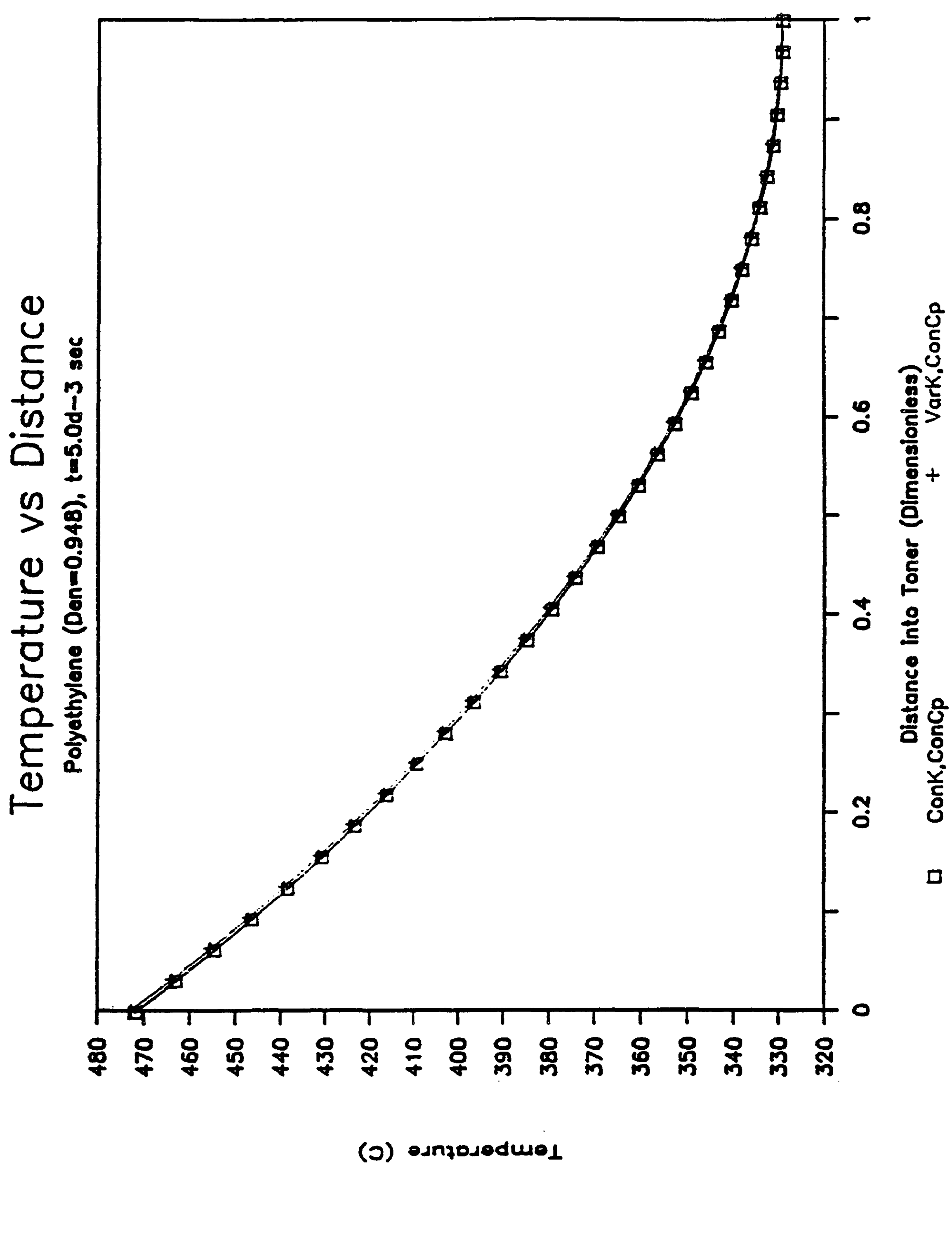

Figure IV.4: Constant vs Variable K for Polyethylene, t=5.0X10-3 sec

do not vary with temperature in excess of 30-40% (2). The common range for thermal conductivity in polymers is 3-12 X 10^{-4} cal/cm-sec-C. When converted to these units the constant K value is 5.0d-4 cal/cm-sec-C.

The next phase of the work was to make the C_p a function of temperature also. This is discussed in the next chapter.

Chapter V

Heat Conduction in a Polymer Film with Variable Thermal Conductivity and Variable Heat Capacity

Mathematical Formulation

Consider the heating of a polymer film occupying the finite region $0 \le x \le \ell$ and initially at a uniform temperature T_i. At time t=0, the film is heated by applying a constant heat flux q_0 at the surface. The region is heated until a time $t=t_f$. In the previous chapter, the thermal conductivity was taken as a function of temperature while the heat capacity and density were held constant. In the present chapter, we shall investigate the effect of a variable heat capacity as well as a variable thermal conductivity. The appropriate differential equation and associated boundary conditions and initial conditions are given by:

$$\rho_0 \, c_p(T) \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left[k(T) \frac{\partial T}{\partial x} \right] \qquad 0<x<\ell, \; t>0 \qquad (V.1)$$

$$- k(T) \frac{\partial T}{\partial x} = q_0 \qquad x=0, \; t>0 \qquad (V.2)$$

$$- K(T) \frac{\partial T}{\partial x} = q_\ell \qquad x=1, \; t>0 \qquad (V.3)$$

$$T = T_i \qquad 0<x<\ell, \; t=0 \qquad (V.4)$$

As in the previous chapters, the dimensionless variable used for these equations are:

$$U = \frac{T - T_i}{q_0(\ell/k_0)} \; , \quad X = \frac{x}{\ell} \; , \quad \tau = \frac{\alpha_0 t}{\ell^2} \; ,$$

and

$K(T) = \dfrac{k(T)}{k_0}$, $C_v(T) = \dfrac{c_v(T)}{c_{v0}}$, where c_v is the volumetric heat capacity defined as $c_v = c_p(T)\rho$ and

$c_{v0} = c_{p0}\rho_0$. Also, $\alpha_0 = \dfrac{k_0}{\rho_0 c_{p0}}$ (V.5)

The problem then runs as follows:

$$C_v(U) \frac{\partial U}{\partial \tau} = \frac{\partial}{\partial X} \left[K(U) \frac{\partial U}{\partial X} \right] \qquad 0 < X < 1, \ \tau > 0 \qquad (V.6)$$

$$\frac{\partial U}{\partial X} = - \frac{G_0}{K(U)} , \qquad x = 0, \ \tau>0 \qquad (V.7)$$

$$\frac{\partial U}{\partial X} = - \frac{G_1}{K(U)} , \qquad X = 1, \ \tau>0 \qquad (V.8)$$

$$U = 0 , \qquad 0 < X < 1, \ \tau=0 \qquad (V.9)$$

where $G_0 = q_0 / q_0$ and $G_1 = q_\ell / q_0$.

The Crank-Nicholson finite difference scheme was used to solve equations (V.6 - V.9) as in chapters III and IV. For these equations however, the thermal conductivity, K(U), is treated as described in Chapter IV and C_v is evaluated at each grid point. The resulting equations become:

$$\frac{C_{v_{i,n+1}} U_{i,n+1} - C_{v_{i,n}} U_{i,n}}{\Delta\tau} =$$

$$\frac{K_{+1/2}}{2(\Delta X)^2}\left[U_{i+1,n+1} - U_{i,n+1} + U_{i+1,n} - U_{i,n}\right] +$$

$$\frac{K_{-1/2}}{2(\Delta X)^2}\left[U_{i,n+1} - U_{i-1,n+1} + U_{i,n} - U_{i-1,n}\right] \quad (V.10)$$

where $K_{+1/2}$ and $K_{-1/2}$ are defined by equations (IV.11) and (IV.12) in Chapter IV. Since the central difference representations of the boundary conditions are the same as those which are in Chapter IV, the resulting equations for the imaginary points are the same. Equations (IV.16) through (IV.19) were used to determine the boundary condition equations for the present problem. Equation (V.10) may be rearranged as follows:

$$-K_{-1/2}U_{i-1,n+1} + \left[\frac{2C_{v\ i,n+1}}{\lambda} + K_{+1/2} + K_{-1/2}\right] U_{i,n+1}$$

$$-K_{+1/2}U_{i+1,n+1} = K_{-1/2}\ U_{i-1,n} + \left[\frac{2C_{v\ i,n}}{\lambda} - K_{+1/2} - K_{-1/2}\right]U_{i,n}$$

$$+ K_{+1/2}U_{i+1,n} \quad , \quad i=2,3,4,\ldots N \quad (V.11)$$

where $\lambda = \Delta\tau/\Delta X^2$. Equation (V.11) is written at the grid points i=1 and i=N+1 and then combined with Equations (IV.16 - IV.19) to give:

$$\left[\frac{2C_{v\ 1,n+1}}{\lambda} + K_{+1/2} + K_{-1/2}\right]U_{1,n+1} - \left[\frac{K_{-1/2}K(U)_{2,n+1}}{K_{0,n+1}} + K_{+1/2}\right]*$$

$$U_{2,n+1} = \left[\frac{2C_{v\ 1,n}}{\lambda} - K_{+1/2} - K_{-1/2}\right]U_{1,n} + \left[\frac{K_{-1/2}K(U)_{2,n}}{K_{0,n}} + K_{+1/2}\right]$$

$$2K_{-1/2}\ G_0\ \Delta X\left[\frac{1}{K_{0,n}} + \frac{1}{K_{0,n+1}}\right], \qquad i=1 \qquad (V.12)$$

and

$$-\left[\frac{K_{+1/2}K(U)_{N-1,n+1}}{K(U)_{N+1,n+1}} + K_{-1/2}\right]U_{N-1,n+1} + \left[\frac{2C_{v\ N,n+1}}{\lambda} + K_{+1/2}\right.$$

$$\left. + K_{-1/2}\right]U_{N,n+1} = \left[\frac{K_{+1/2}K(U)_{N-1,n}}{K(U)_{N+1,n}} + K_{-1/2}\right]U_{N-1,n} +$$

$$\left[\frac{2C_{v\ N,n+1}}{\lambda} - K_{+1/2} - K_{-1/2}\right]U_{N,n} - 2K_{+1/2}\ G_1\Delta X$$

$$*\left[\frac{1}{K(U)_{N+1,n+1}} + \frac{1}{K(U)_{N+1,n}}\right] \qquad i=N+1 \qquad (V.13)$$

With $U_{i,j}$ (i=1,2,...n+1) known at the time level j, Equations (V.11), (V.12) and (V.13) represent N+1 equations for the unknown temperatures $U_{i-1,j+1}$, $U_{i,j+1}$, and $U_{i+1,j+1}$.

Both density and thermal conductivity were treated as constants to investigate the variation of heat capacity with temperature. Two polymers (PvF2 and Dalvor 8200) were used to evaluate the effect of heat capacity on the temperature distribution. Tables (V.1) and (V.2) represent the values of C_p vs temperature used in this analysis. Graphical representation for both polymers is shown in Figure (V.1). For Dalvor 8200, the value used for the constant Cp case was an average Cp calculated over the temperature region in Figure V.1. When Commerical PVF2 was analyzed, the initial value at the beginning of the temperature range in Figure V.1 was used for the constant Cp case. The constant Cp values were chosen in this manner because varying relationships with the variable Cp case was desired. In the case of Dalvor 8200, the constant Cp value is above the variable Cp values for most of the temperature range in question. Conversely, the constant Cp value for Commercial PVF2 is below the variable Cp values for most of the temperature region. Their values are listed below with the rest of the physical properties for each polymer. In each case the thermal conductivity was taken from data given to

Table V.1
Commercial PVF2
Temperature vs Specific Heat

Temperature (^oC)	Specific Heat (cal/g ^oC)
< 76	0.325
76	0.325
86	0.360
100	0.375
110	0.382
120	0.395
130	0.415
140	0.450
150	0.500
160	0.800
162.5	0.975
165	0.770
170	0.425
172.5	0.375
180	0.379
185	0.382
190	0.382
> 190	0.382

It may be noted that the specific heat was taken as constant both below and above the range of these data.

Table V.2
Davlor 8200
Temperature vs Specific Heat

Temperature (°C)	Specific Heat (cal/g °C)
< 82	0.340
82	0.340
90	0.350
100	0.375
110	0.395
120	0.340
130	0.440
140	0.475
150	0.500
152	0.550
155	0.700
160	0.900
165	1.210
170	0.975
175	0.590
180	0.400
> 180	0.400

It may be noted that the specific heat was taken as constant both below and above the range of these data.

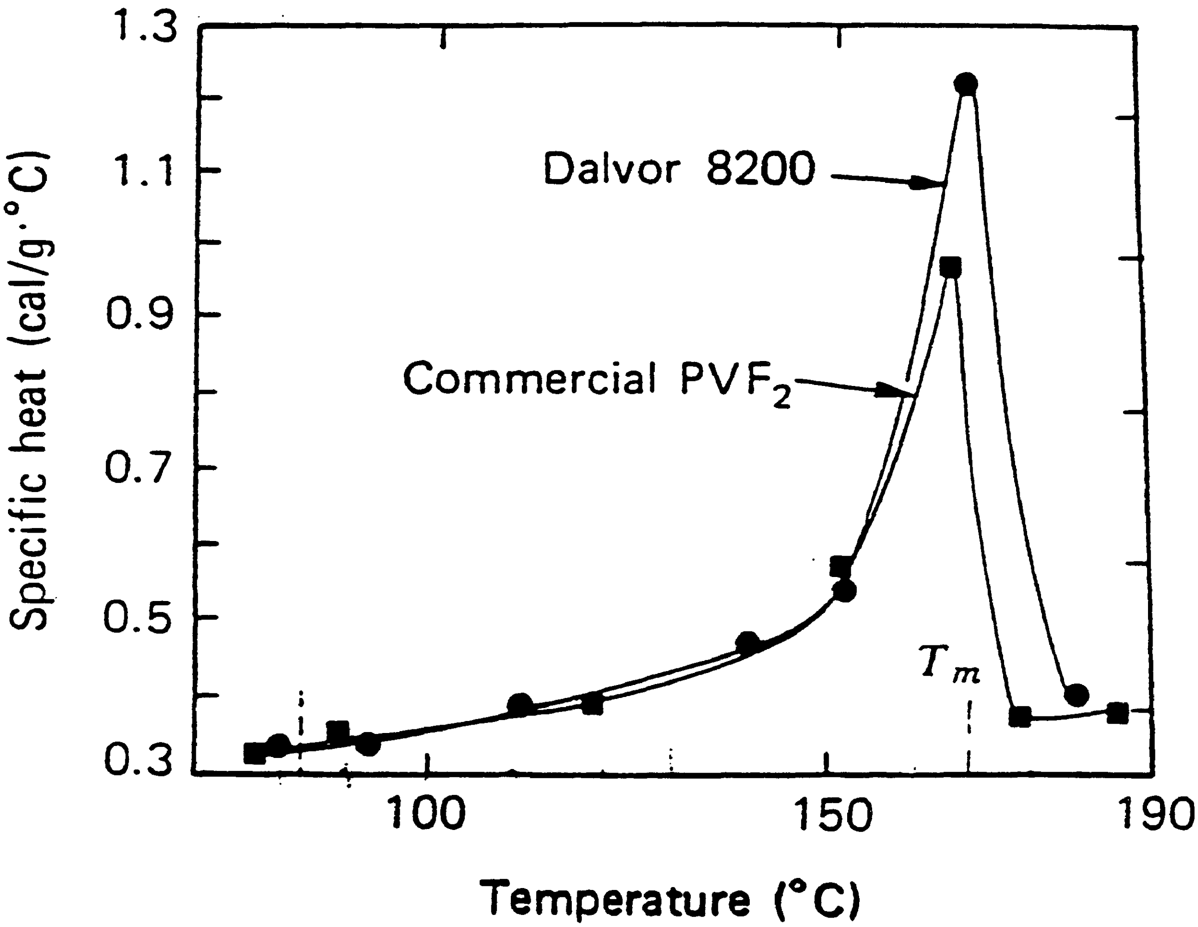

Figure V.1: C_p vs Temperature for Commerical PVF_2 and Dalvor 8200 this author by IBM(7), and the density was obtained from representative literature values(8). The C_p values used for the constant property base cases were obtained from Tadmore and Gogos(2). The physical properties for PvF2 were taken as:

ρ_0 = 1000 kg/m^3

k_0 = 0.21 J/m-sec C

C_p = 1359.15 J/kg-C

The physical properties for Dalvor 8200 were taken as:

ρ_0 = 1000 kg/m^3

k_0 = 0.21 J/m-sec C

C_p = 2296.0 J/kg-C

Also, the following values were taken for T_0, q_0, q_ℓ and ℓ for both polymers:

T_0 = 0

q_0 = 3.0X10+6 J/m^2-sec

q_ℓ = 0.0

ℓ = 20X10-6 meters

Figures (V.2) through (V.9) show the temperature distributions throughout the median of both cases. The first four figures show the variable C_v curve above the constant C_v curve. This is a result of the C_v values being below that used for the constant Cv curve. The temperature range for the impulse curve shown for the Dalvor 8200 polymer spanned temperatures from 70°C to 190°C. The temperature range for the polymer during the flash fusion process was from 0°C to approximately 600°C. When the temperature was either below or above the range of the curve, a constant value from the curve was used. Figure (V.2) shows a temperature range between 10°C and 120°C. Since the impulse was not seen until 150°C the resulting profile for a time of 3.84X10-4 sec had a temperature difference between the constant Cv case and the variable Cv case of 25°C at the polymer surface. This was due to the variable C_v values being lower (an average value being 1672 J/kg C) than the constant C_v value of 2296 J/kg C. As time increases, the impulse becomes more important because the temperature range now includes the values in Figure (V.1). Figure (V.3) shows the variable Cv temperature profile also above the constant property case. In this figure the slopes of the two curves are very close until the variable C_v curve reaches the impulse temperature range. When this occurs, the slope is reduced causing the curve to be more flat. This is directly a result of the impulse variation in the C_v curve. Before the curve reaches the impulse temperature region the C_v value is constant at 1672 J/kg C which is again lower than that used for the constant property case (2296 J/kg C). At a time of 1.95X10-3 sec the temperature difference between the two cases is 43°C. As time increases again to 3.84X10-3 sec (Figure

ARTHUR LAKES LIBRARY
COLORADO SCHOOL of MINES
GOLDEN, COLORADO 80401

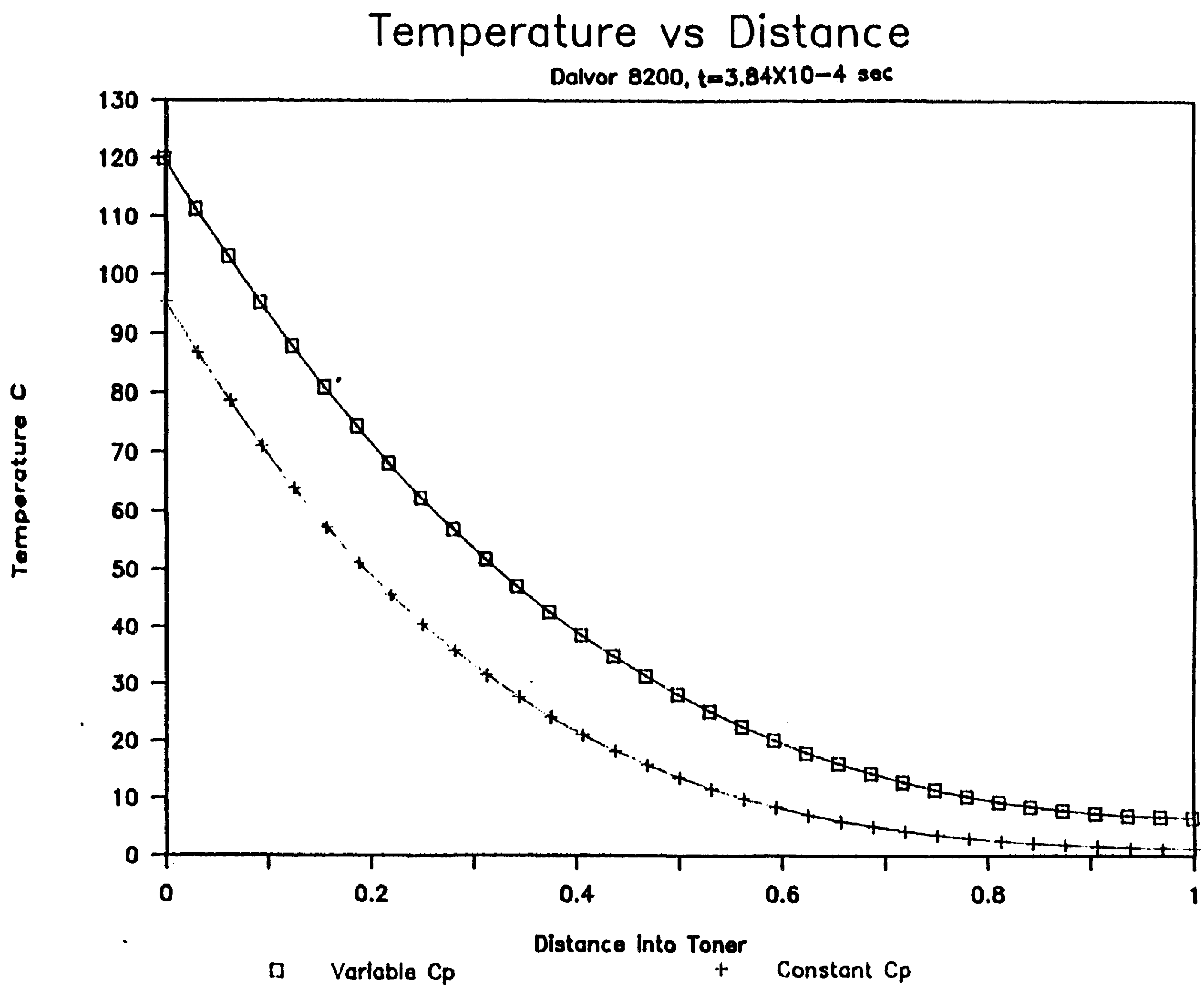

Figure V.2: Constant vs Variable C_p, Dalvor 8200, t=3.8X10-4 sec

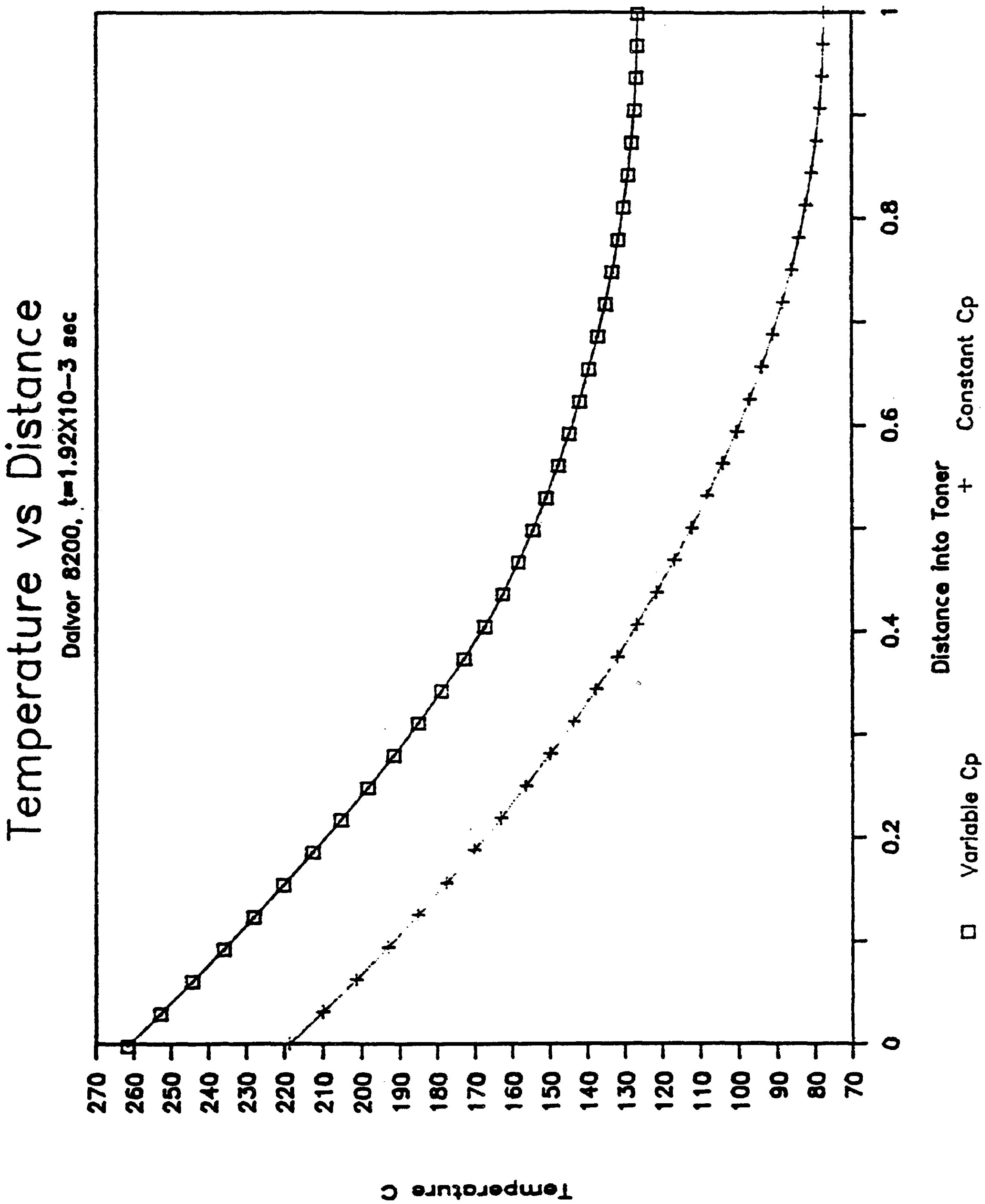

Figure V.3: Constant vs Variable C_p, Dalvor 8200, t=1.9X10-3 sec

(V.4)) and finally 5.0X10-3 sec (Figure (V.5)) the temperature at the surface of the polmer increases from 342^{o}C to 418^{o}C and 420^{o}C to 522^{o}C respectively. Realizing the final temperature to be very high, it was noted that paper singeing was observed during the flash fusion process. Addition of paper to the model will reduce the temperature at the polymer surface because it will absorb some heat from the flash fusion process. When analyzing the resulting temperature profiles obtained for Commercial PVF2 the same trends in temperature were observed but were reversed due to the fact the constant C_v value was always below those values used in the variable Cv case. In figure (V.5), both curves are identical. The value used for the constant C_v case was the lowest value on the curve or 1359.15 J/kg C. Since the impulse does not show up in the calculations until much later the value of 1359.15 J/kg C was used for C_v when temperatures were below figure (V.1)'s range. The resulting profiles for very early times are represented by Figure (V.6). As time increases, the resulting C_v curve is shown in Figure (V.7) for a time of 1.95X10-3 sec. Again, when the temperature is in the range of the impulse the slope is slightly less causing a flatting out of the profile curve during this region. As can be seen by Figure (V.7) the variable C_v curve starts to relax at a temperature of 170^{o}C. When using a larger C_v the resulting surface temperature of

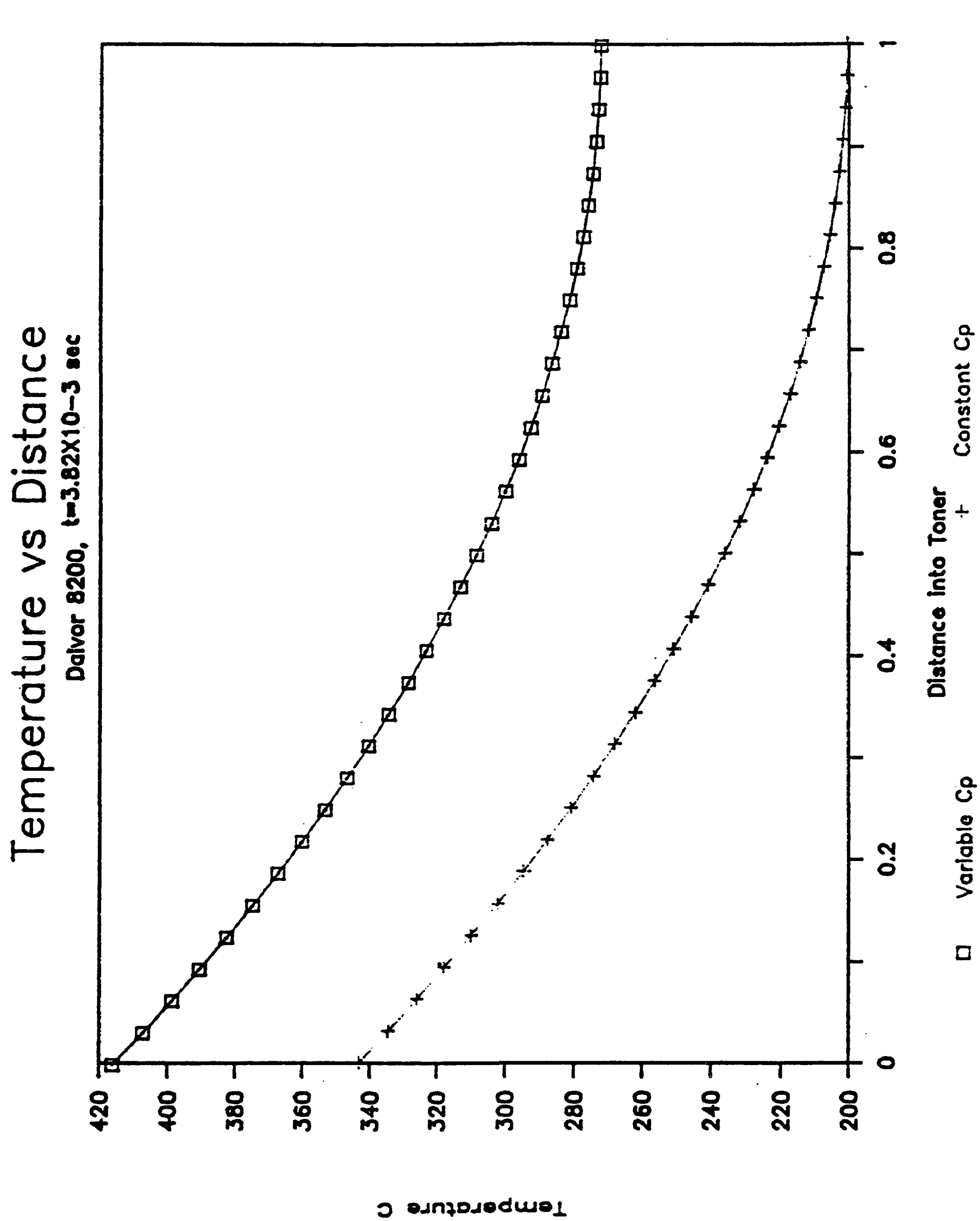

Figure V.4: Constant vs Variable C_p, Dalvor 8200, t=3.8X10-3 sec

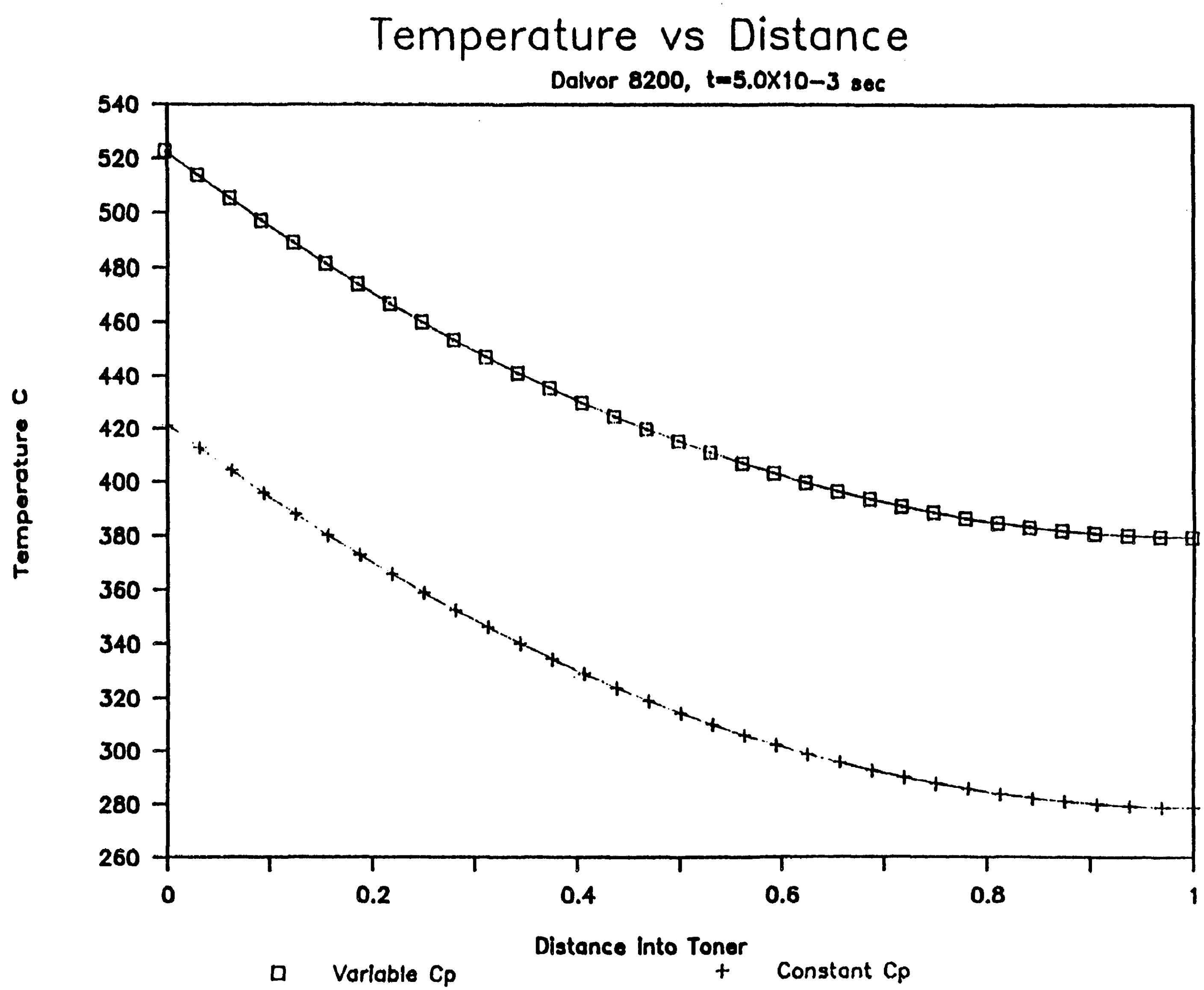

Figure V.5: Constant vs Variable C_p, Dalvor 8200, t=5.0X10-3 sec

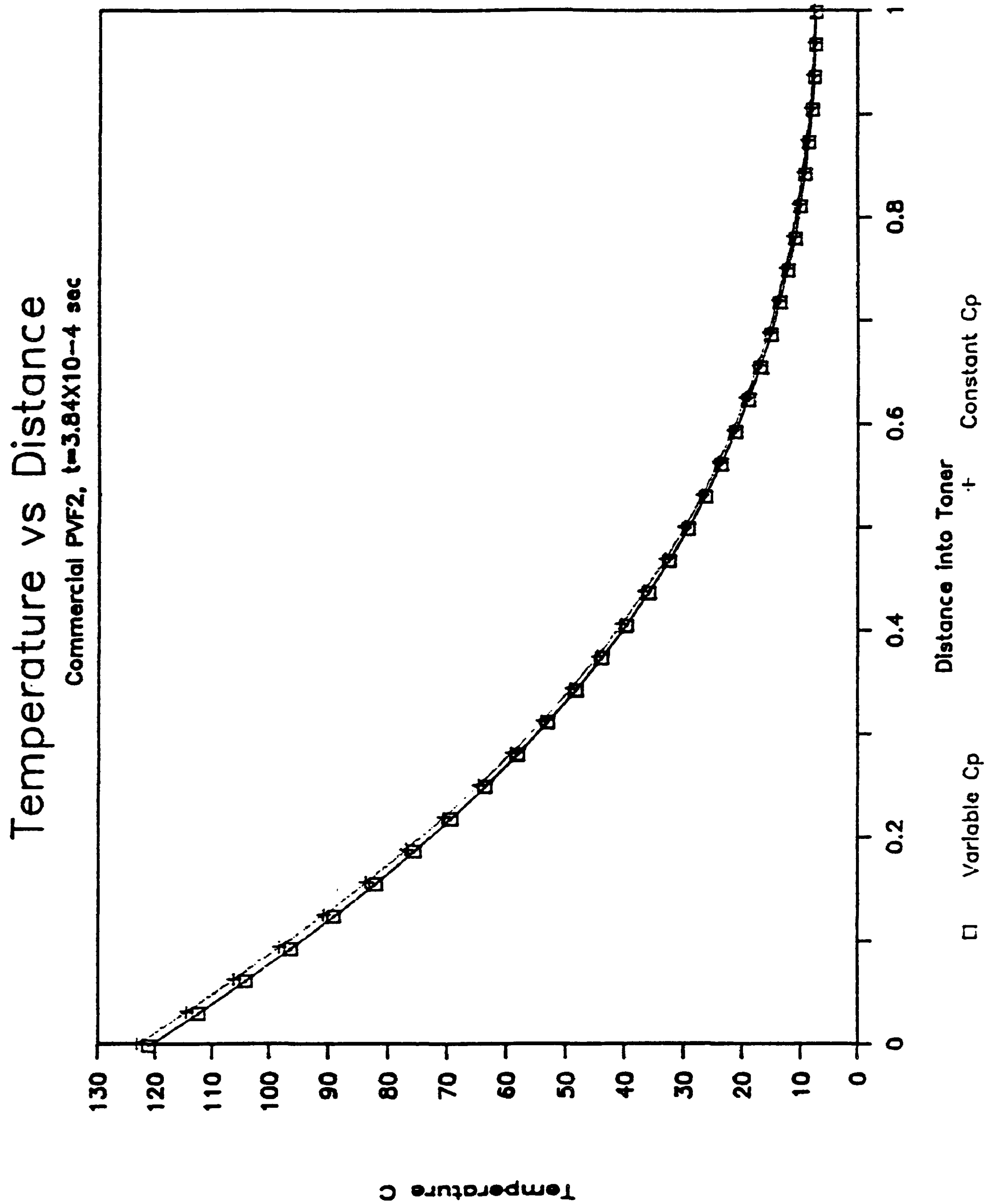

Figure V.6: Constant vs Variable C_p, PVF_2, t=3.8X10-4 sec

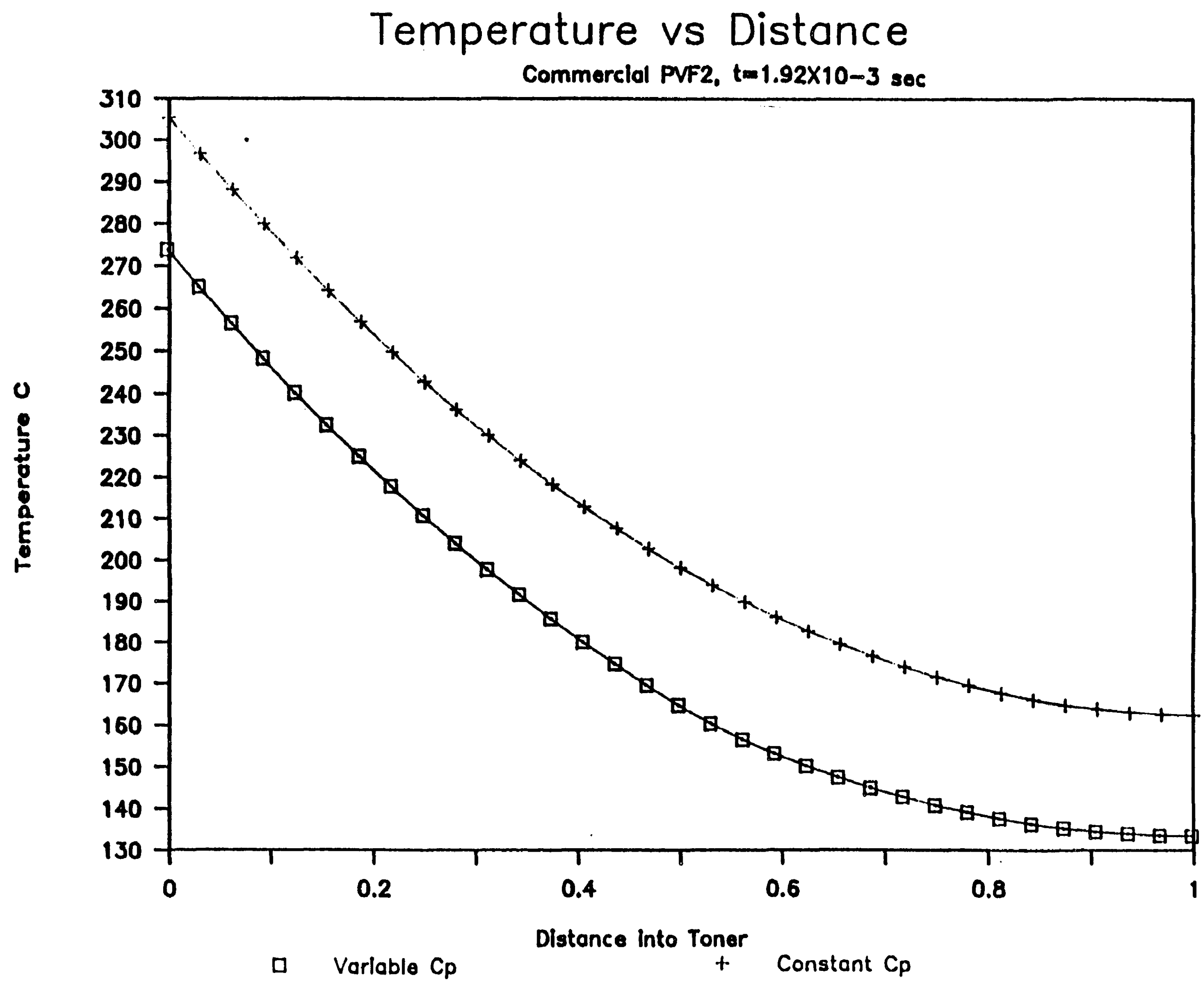

Figure V.7: Constant vs Variable C_p, PVF_2, t=1.9X10-3 sec

the polymer is reduced. At this time, the temperature reduces by approximately 31^{o}C. In Figures (V.8) and (V.9) the temperature differences increase as time increases. For the time of 3.84X10-3 sec the resulting surface temperature is 75^{o}C cooler and for the final time of 5.0X10-3 sec the temperature reduces from 649^{o}C to 559^{o}C. When choosing a polymer for the flash fusion process the C_p values should be evaluated to determine the maximum surface temperature the polymer can reach. The results show C_p to be an important factor when choosing a polymer for this process.

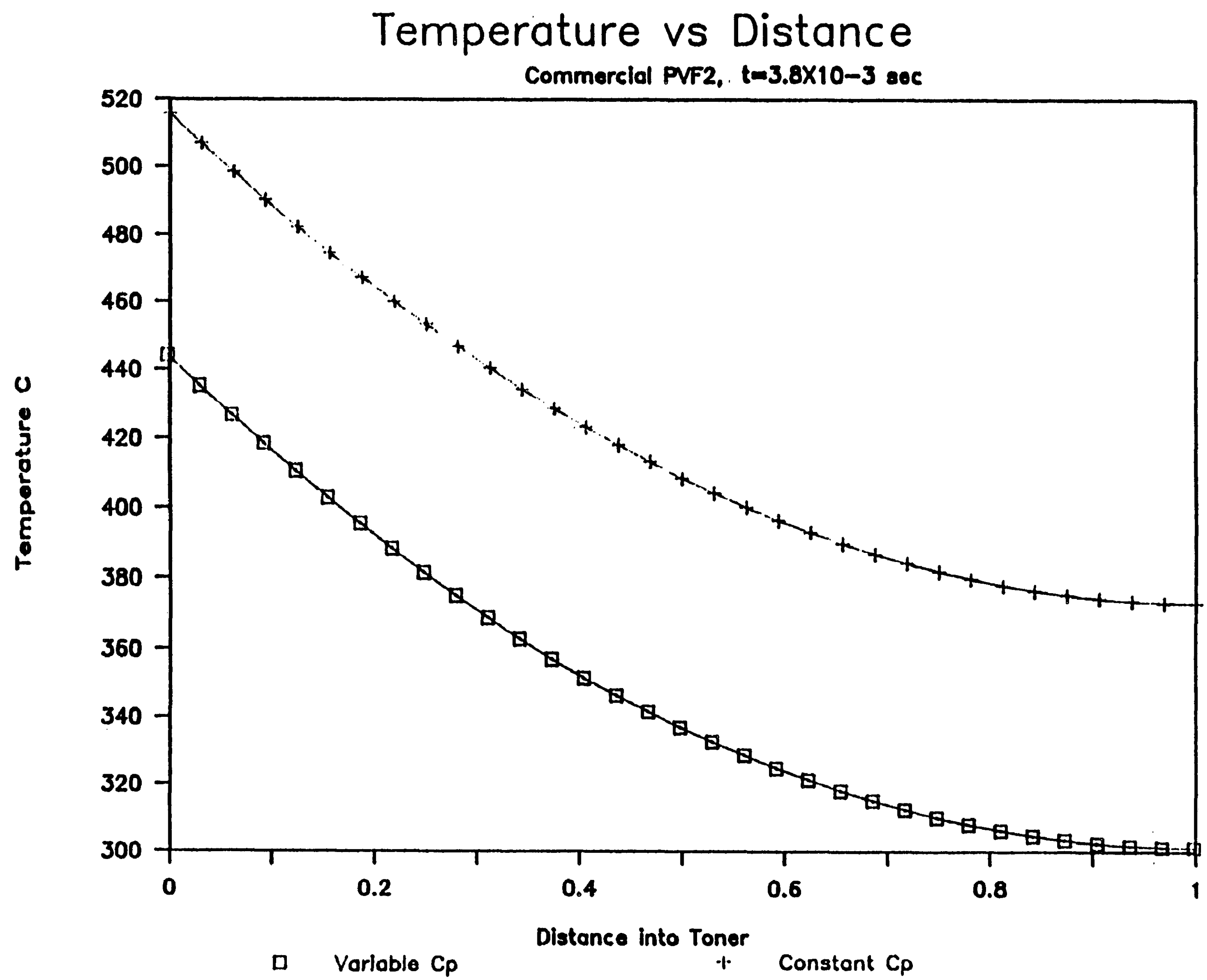

Figure V.8: Constant vs Variable C_p, PVF_2, t=3.8X10-3 sec

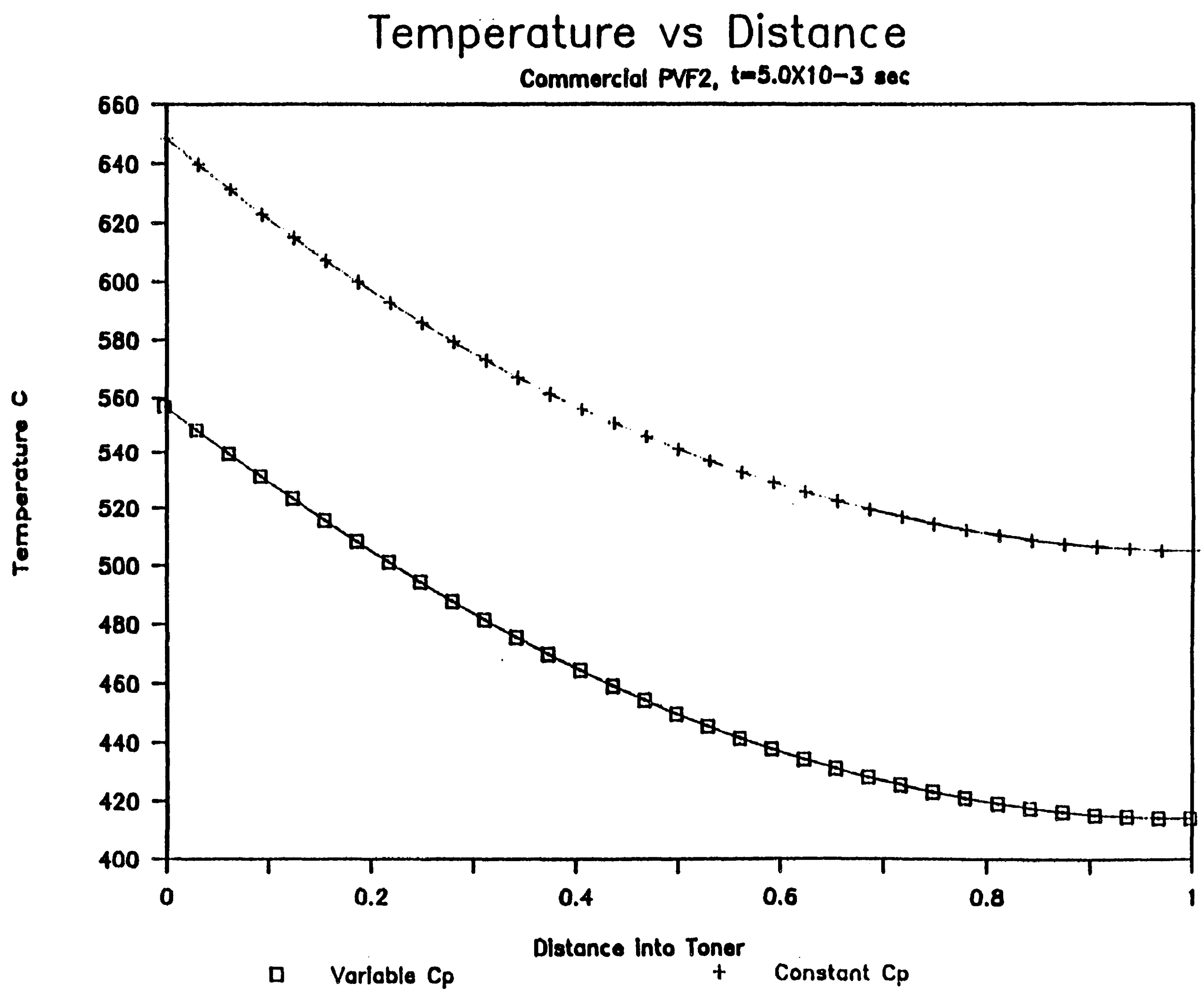

Figure V.9: Constant vs Variable C_p, PVF_2, t=5.0X10-3 sec

Chapter VI

Effect of Supporting Paper on Heat Conduction through a Polymer Film during Flash Fusion

Figure (VI.1) shows a schematic diagram of the flash fusion of a polymer film supported on paper. In the present chapter, the effect of heat conduction through the paper is investigated. The mathematical problem may be stated as follows:

Consider the heating of a finite region divided into two sections; the first $0 \le x \le \ell$ and the second $\ell \le x \le 3\ell$ initially at a uniform temperature T_i. At time t=0, the region is heated by applying a constant heat flux q_o at the surface x=0. The surface at $x=3\ell$ is kept insulated. All physical properties are taken as constants for this application. The appropriate differential equations and associated boundary and initial condition for each region are given by:

$$\frac{\partial T}{\partial t} = \alpha_1 \frac{\partial^2 T}{\partial x^2} \qquad 0 \le x \le \ell, \quad t>0 \qquad \text{(VI.1)}$$

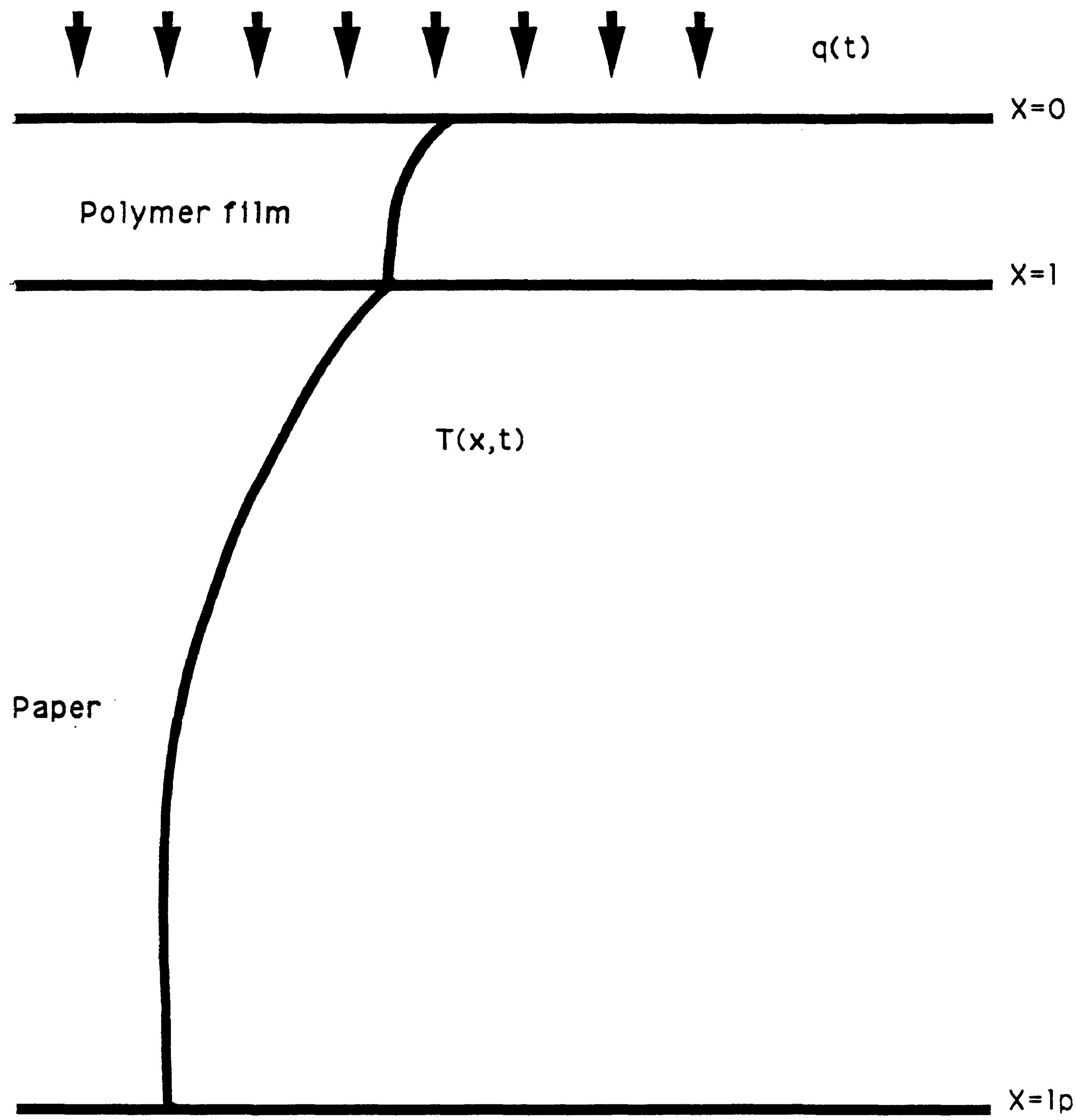

Figure VI.1 Schematic of Flash Fusion Process

$$\frac{\partial T}{\partial t} = \alpha_2 \frac{\partial^2 T}{\partial x^2} \qquad \ell \le x \le 3\ell,\ t>0 \qquad \text{(VI.2)}$$

$$- k_1 \frac{\partial T}{\partial x} = q_0 \qquad x=0,\ t>0 \qquad \text{(VI.3)}$$

$$T_{-\ell} = T_{+\ell} \qquad x=\ell,\ t>0 \qquad \text{(VI.4)}$$

$$-k_1 \frac{\partial T}{\partial x} = -k_2 \frac{\partial T}{\partial x} \qquad x=\ell,\ t>0 \qquad \text{(VI.5)}$$

$$- k_2 \frac{\partial T}{\partial x} = 0 \qquad x=3\ell,\ t>0 \qquad \text{(VI.6)}$$

$$T = T_i \qquad 0 \le x \le 3\ell,\ t=0 \qquad \text{(VI.7)}$$

where,

$$\alpha_1 = \frac{k_1}{\rho_1 C_{p1}} \quad \text{and} \quad \alpha_2 = \frac{k_2}{\rho_2 C_{p2}} \qquad \text{(VI.8)}$$

This problem can be cast into a more convenient form by defining the following dimensionless variables:

$$X = \frac{x}{\ell}\ , \quad \tau_1 = \frac{\alpha_1 t}{\ell^2}\ , \quad \tau_2 = \frac{\alpha_2 t}{\ell^2}\ , \quad U = \frac{T - T_i}{q_0\ \ell\ /\ k_1}\ , \qquad \text{(VI.9)}$$

Equations (VI.1) - (VI.6) expressed in terms of these dimensionless variables are given by:

$$\frac{\partial U}{\partial \tau_1} = \alpha_1 \frac{\partial^2 U}{\partial X^2} \qquad 0<X<1,\ \tau_1>0 \qquad (VI.10)$$

$$\frac{\partial U}{\partial \tau_2} = \alpha_2 \frac{\partial^2 U}{\partial X^2} \qquad 1<X<3,\ \tau_2>0 \qquad (VI.11)$$

$$\frac{\partial U}{\partial X} = -1 \qquad X=0,\ \tau_1>0 \qquad (VI.12)$$

$$U_{-1} = U_{+1} \qquad X=1,\ \tau_1>0,\ \tau_2>0 \qquad (VI.13)$$

$$k_1 \frac{\partial U}{\partial X} = k_2 \frac{\partial U}{\partial X} \qquad X=1,\ \tau_1>0,\ \tau_2>0 \qquad (VI.14)$$

$$\frac{\partial U}{\partial X} = 0 \qquad X=3,\ \tau_2>0 \qquad (VI.15)$$

ARTHUR LAKES LIBRARY
COLORADO SCHOOL of MINES
GOLDEN, COLORADO 80401

$$U = 0 \qquad 0<x<3,\ \tau_1 = \tau_2 = 0 \qquad \text{(VI.16)}$$

In the present analysis both the polymer and paper have constant property values. The finite-difference formulation for both regions, 0<X<1, and 1<X<3 is similar to that developed in Chapter III (see equation III.18). The boundary condition for X=0 (i=1) remains the same and is represented by equation (III.19). Equation (III.20) is valid but for X=3 (i=3N+1) instead of X=1. At X=1, equations (VI.15) and (VI.16) must be satisfied. The temperature just inside each section at X=1 must be equal, $T_{-1} = T_{+1}$. The temperature at X=1 can be represented by a Taylor Series expansion as follows:

$$U_{-1} = U_0 + \left.\frac{\partial U}{\partial X}\right)_{0-}(-\Delta X_1) + \left.\frac{\partial^2 U}{\partial X^2}\right)_{0-}(\Delta X_1)^2 + \ldots \qquad \text{(VI.17)}$$

$$U_{+1} = U_0 + \left.\frac{\partial U}{\partial X}\right)_{0+}(\Delta X_2) + \left.\frac{\partial^2 U}{\partial X^2}\right)_{0+}(\Delta X_2)^2 + \ldots \qquad \text{(VI.18)}$$

and,

$$\Delta\tau_1 = \frac{\alpha_1 \Delta t}{\ell^2}\ ,\ \Delta\tau_2 = \frac{\alpha_2 \Delta t}{\ell^2}\ ,\ \Delta X_1 = \frac{\Delta x_1}{\ell}\ ,\ \Delta X_2 = \frac{\Delta x_2}{\ell} \qquad \text{(VI.19)}$$

We now define the following mesh ratios:

$$\lambda_1 = \frac{\Delta\tau_1}{\Delta X^2}\ , \qquad \lambda_2 = \frac{\Delta\tau_2}{\Delta X^2}$$

We shall also take $\lambda_1=\lambda_2=\lambda$ so that:

$$\frac{\Delta\tau_1}{(\Delta X_1)^2} = \frac{\Delta\tau_2}{(\Delta X_2)^2} \quad ===> \quad (\Delta X_2)^2 = \frac{\alpha_2}{\alpha_1}(\Delta X_1)^2 \qquad \text{(VI.20)}$$

Rearranging Equations (VI.17) and (VI.18) yields:

$$\left.\frac{\partial U}{\partial X}\right)_{0+} = \frac{k_1}{k_2}\left(\frac{\partial U}{\partial X}\right)_{0-} \quad \text{and} \quad \left.\frac{\partial^2 U}{\partial X^2}\right)_{0+} = \frac{\alpha_1}{\alpha_2}\left(\frac{\partial^2 U}{\partial X^2}\right)_{0-}$$

The equations above coupled with Equation (VI.20) are substituted into Equation (VI.18) to generate:

$$U_1 = U_0 + \frac{k_1}{k_2}\left(\frac{\partial U}{\partial X}\right)_{0-}\left(\frac{\alpha_2}{\alpha_1}\right)^{1/2}\Delta X_1 + \left.\frac{\partial^2 U}{\partial X^2}\right)_{0-}(\Delta X_1)^2 \qquad \text{(VI.21)}$$

Multiplying both sides by:

$\frac{k_2}{k_1}\left(\frac{\alpha_1}{\alpha_2}\right)^{1/2}$, which will be called C1 for simplicity, we have

$$(C1)\ T_1 = (C1)\ T_0 + \left.\frac{\partial U}{\partial X}\right)_{0-} \Delta X_1 + (C1)\ \left.\frac{\partial^2 U}{\partial X^2}\right)_{0-} (\Delta X_1)^2 \qquad \text{(VI.22)}$$

Adding Equations (VI.17) and (VI.22) together to eliminate the first partial derivative gives:

$$U_{-1} + (C1)\ T_1 = (1 + C1)\ T_0 + (1 + C1)(\Delta X_1)^2\ \frac{\partial^2 U}{\partial X^2} \qquad \text{(VI.23)}$$

Rearranging to obtain an expression for the second partial derivative yields:

$$\frac{\dfrac{U_{-1} + (C1)\ U_1}{C1 + 1} - U_0}{(\Delta X\ )^2} = \left.\frac{\partial^2 U}{\partial X^2}\right)_{0-} \qquad \text{(VI.24)}$$

Using this expression for the second partial derivative in Equation (VI.10) and performing the Crank-Nicholson finite difference scheme on the resulting equation will give an expression for the dimensionless temperatures to be used at the interface point. Combining Equations (VI.10) and (VI.24) gives:

$$\frac{\partial U}{\partial \tau_1} = \frac{\dfrac{U_{-1} + (C1)\ U_1}{C1 + 1} - U_0}{(\Delta X_1)^2} \qquad \text{(VI.25)}$$

Applying the Crank-Nicholson finite difference scheme to Equation (VI.25) results in:

$$\frac{2}{\lambda}\left[U_{i,j+1} - U_{i,j}\right] = \left[\frac{U_{i-1,j} + (C1)\ U_{i+1,j} - (C1 + 1)U_{i,j}}{C1 + 1}\right]$$

$$+ \left[\frac{U_{i-1,j+1} + (C1)\ U_{i+1,j+1} - (C1 + 1)\ U_{i,j+1}}{C1 + 1}\right] \qquad \text{(VI.26)}$$

Rearranging Equation (VI.26) into a more convenient form yields:

$$-U_{i-1,j+1} + ((C1+1) + 2(C1+1)/\lambda)\ U_{i,j+1} - (C1)U_{i+1,j+1} =$$

$$U_{i-1,j} + (2(C1+1)/\lambda - (C1+1))\ U_{i,j} + (C1)\ U_{i+1,j} \qquad \text{(VI.27)}$$

Equation (III.19) is still valid at X=0 (i=1) and Equation (III.20) is also valid but for X=3 (i=3N+1).

The results from the above mentioned numerical scheme are shown in Figures (V.1) through (V.4). As can be seen

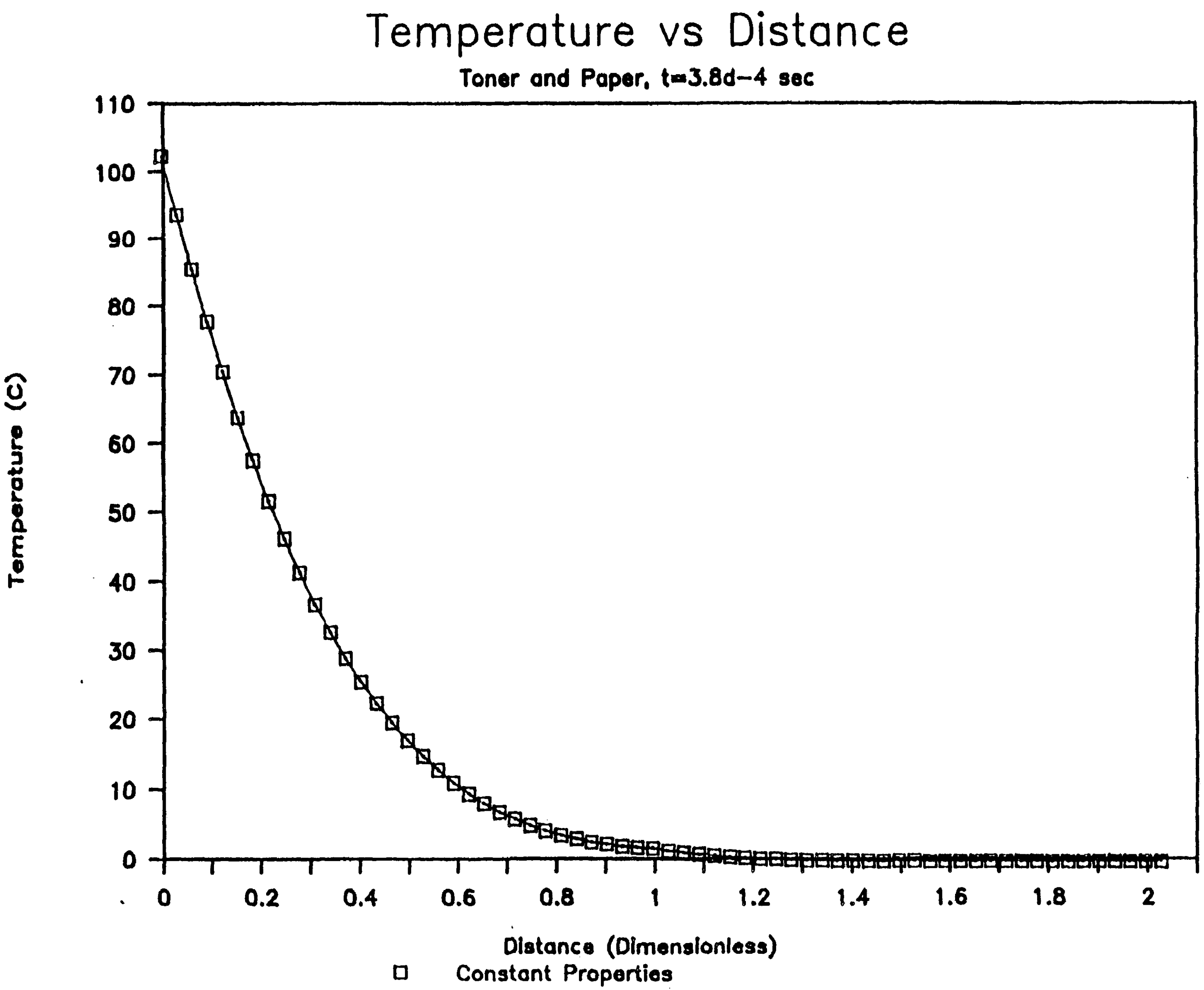

Figure VI.2: Constant Properties for Polymer and Paper, t=3.8X10-4 sec

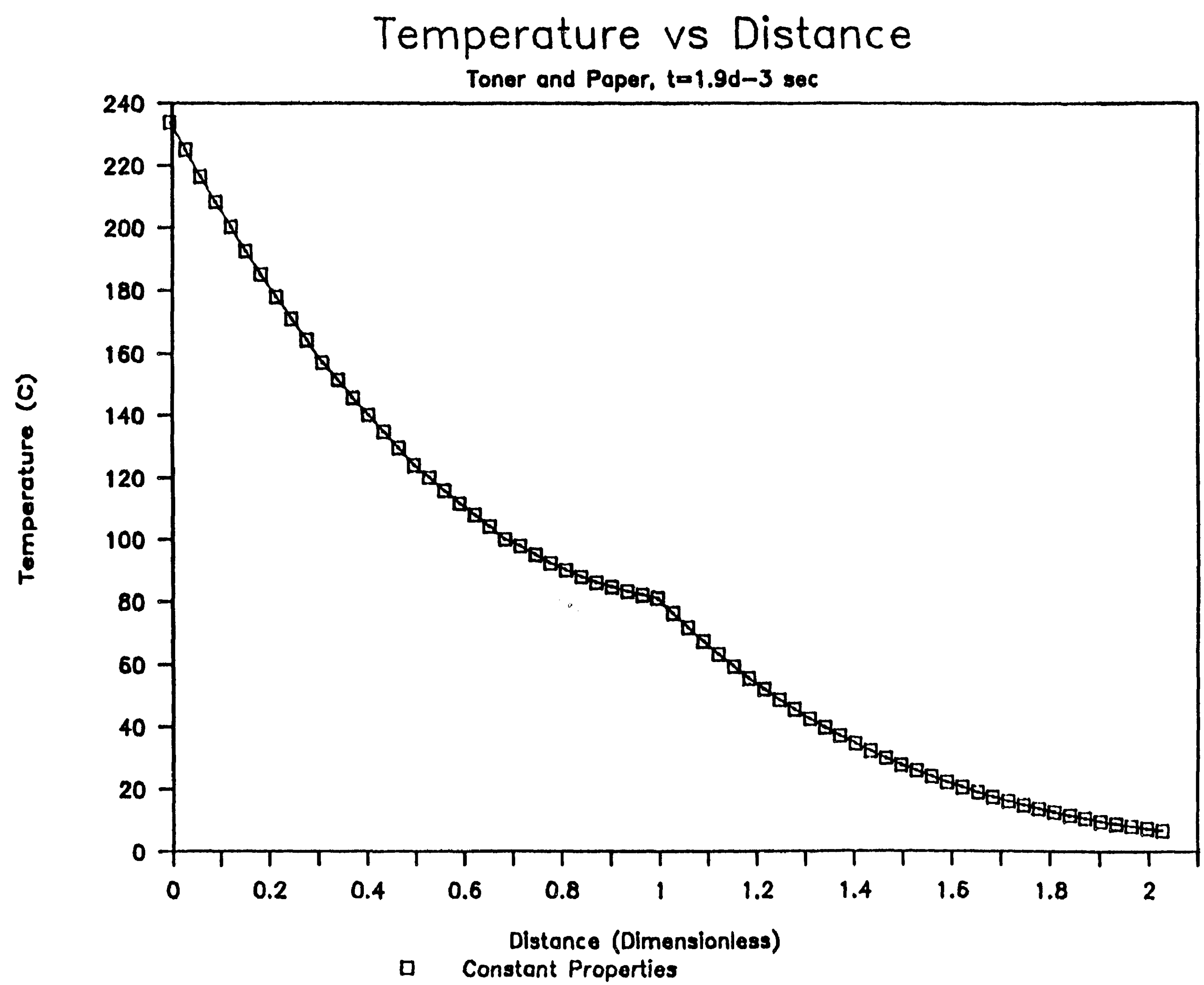

Figure VI.3: Constant Properties for Polymer and Paper, t=1.9X10-3 sec

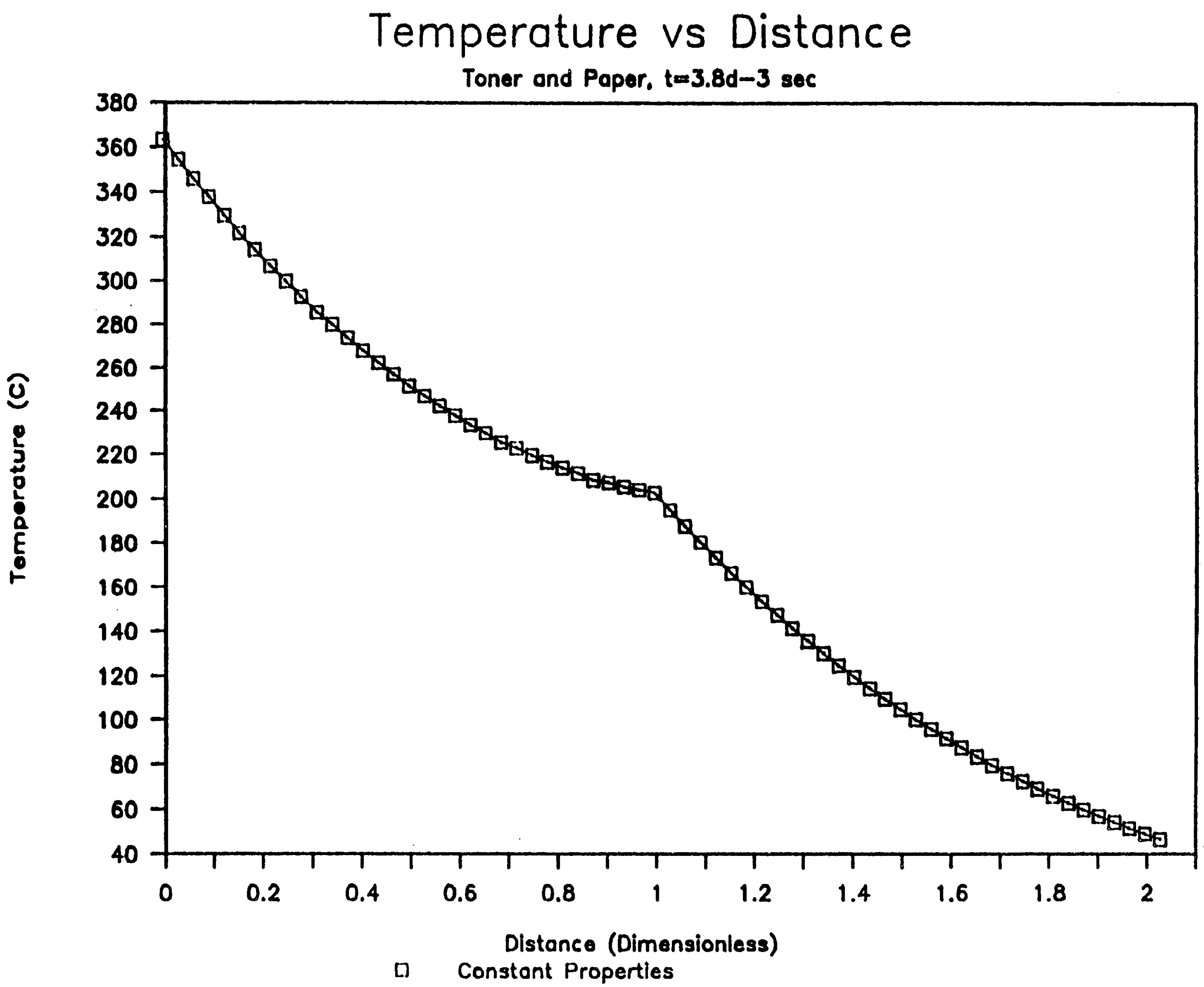

Figure VI.4: Constant Properties for Polymer and Paper, t=3.8X10-3 sec

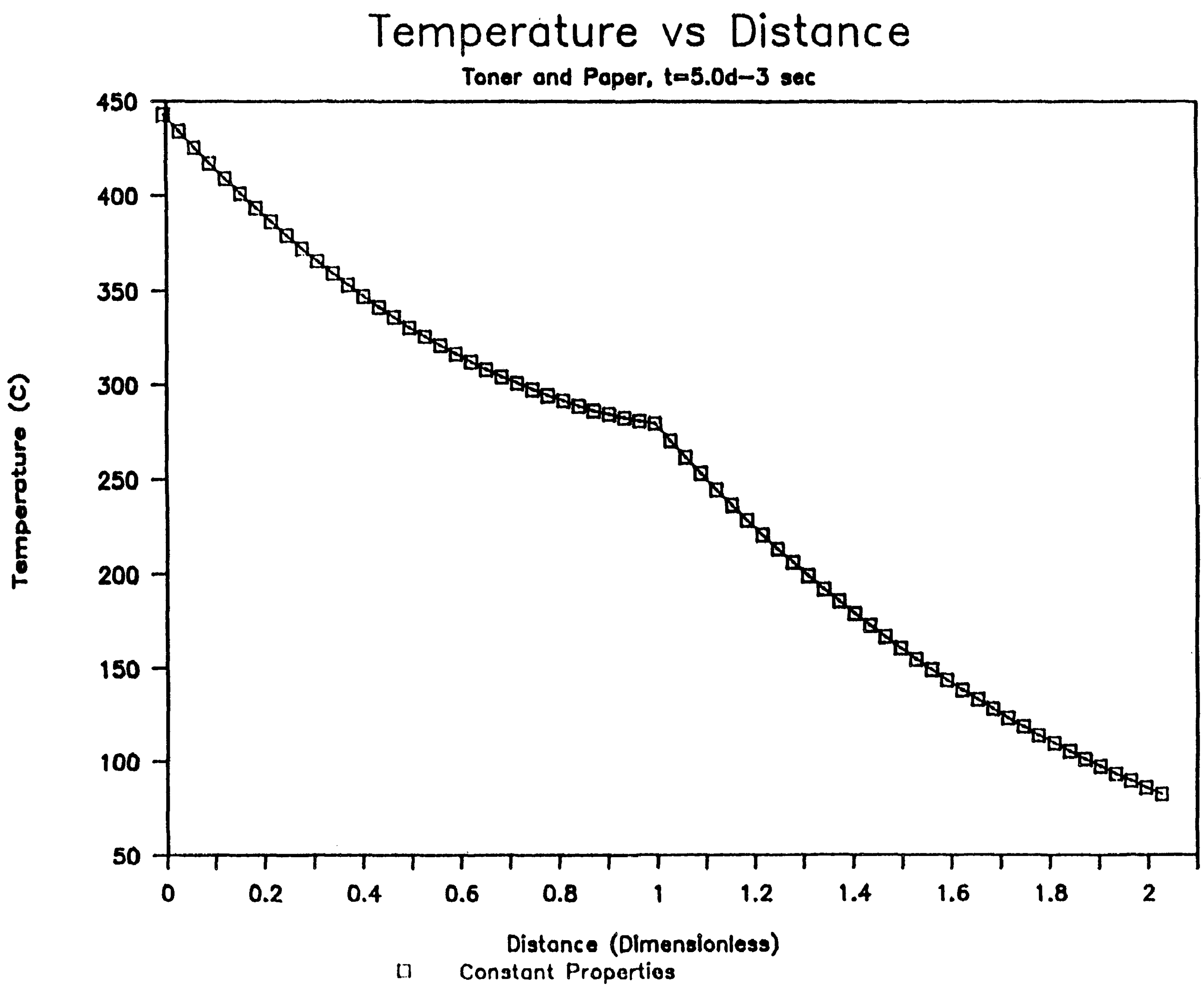

Figure VI.5: Constant Properties for Polymer and Paper, t=5.0X10-3 sec

from the graphs the derivative changes at the point X=1 where the paper and polymer meet. This is a result from the ratio of k_1/k_2 differing from 1. If both thermal conductivities are set equal then there would be no change in the slope. At this point, X=1, both equations have the same value but the derivatives are different. These results were obtained with constant physical properties for both the paper and the polymer. As stated in both Chapters (IV) and (V) the k_0 is from data obtained from IBM(7) and both ρ_0 and C_{p0} are representative values from literature(8). The physical properties used for the polymer were:

ρ_0 = 1000 kg/m^3

k_0 = 0.21 J/m-sec C

C_{p0} = 2000 J/kg C

For the paper both k_0 and C_{p0} were obtained from Perrys Handbook(9). The density was calculated by measuring the weight, and calculating the volume of photocopying paper in the lab.

ρ_0 = 7.15d-4 kg/cm^3

k_0 = 0.075 Btu /hr-ft-F

C_{p0} = 1338.24 J/kg C

In addition, the values for T_0, q_0, $q\ell$, and ℓ were also obtained from IBM data and taken as:

T_0 = 0.0

q_0 = 3.0X10+06 J/m^2-sec

q_ℓ = 0.0

ℓ = 20X10-06 meters

When variable properties (for the polymer) are added to this problem, the equations derived in Chapter IV for the variable K case and Chapter V for the variable Cp case are still valid. Since the results in Chapter IV indicated that a variable K has a small affect on the solution, only the variable C_p case was investigated. The property values for both Dalvor 8200 and Commercial PVF2 are the same as those described in Chapter V on page 54 . The physical properties for paper remained constant throughout the calculations. The physical property values for the initial temperature, heat flux and length are taken as the same as in the constant property case. The C_p values as a function of temperature are shown in Tables (V.1) and (V.2).

The results for both polymers had the same patterns when the paper was added to the problem. The effect of adding paper on the resulting surface temperatures and temperature gradients is shown in Figures (V1.6) through (V.13). In each case the pattern of the temperature gradients was the same as described in Chapter V. That is, when the C_p value for the constant property case was above

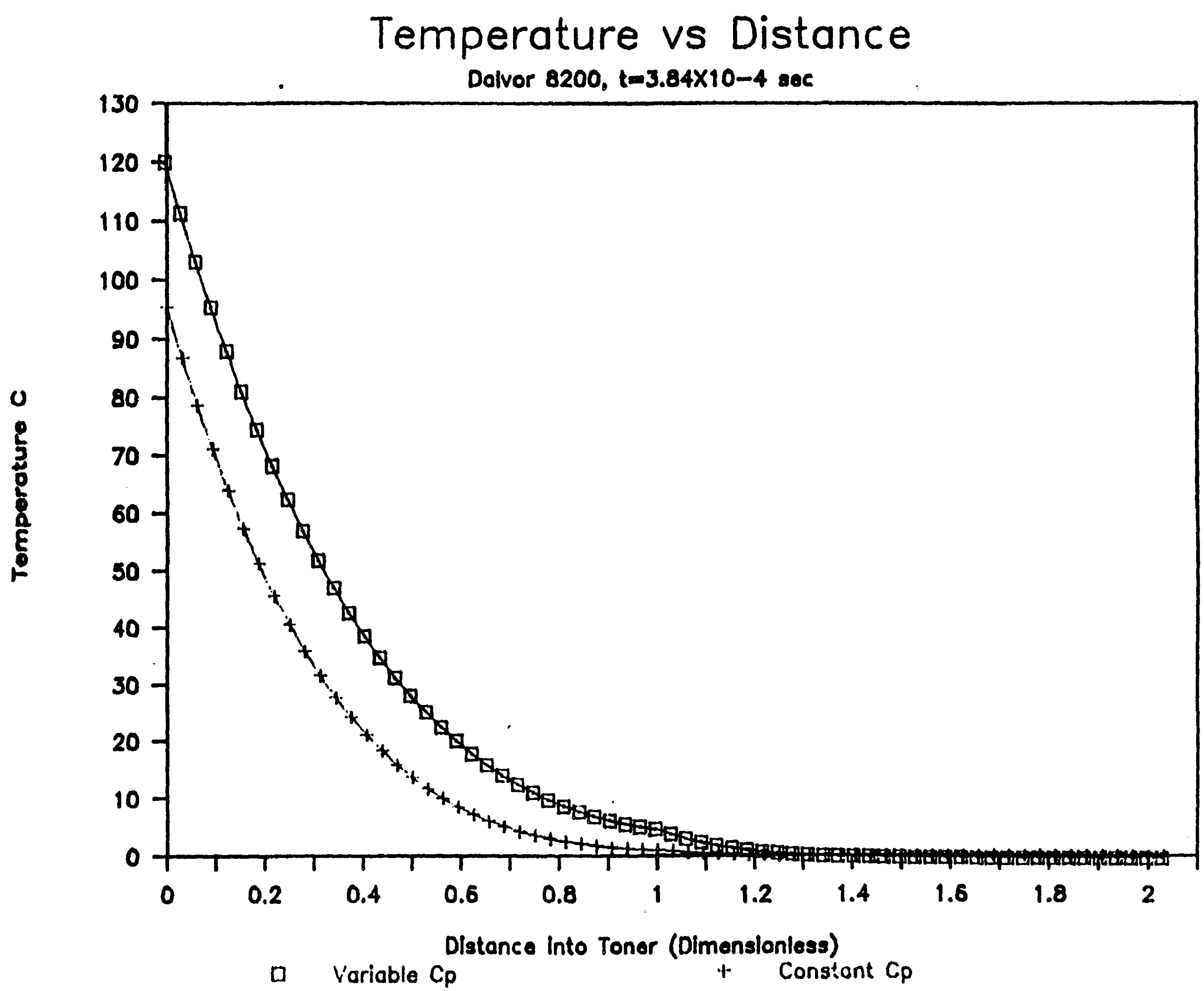

Figure VI.6: Constant vs Variable C_p, Dalvor 8200, t=3.8X10-4 sec

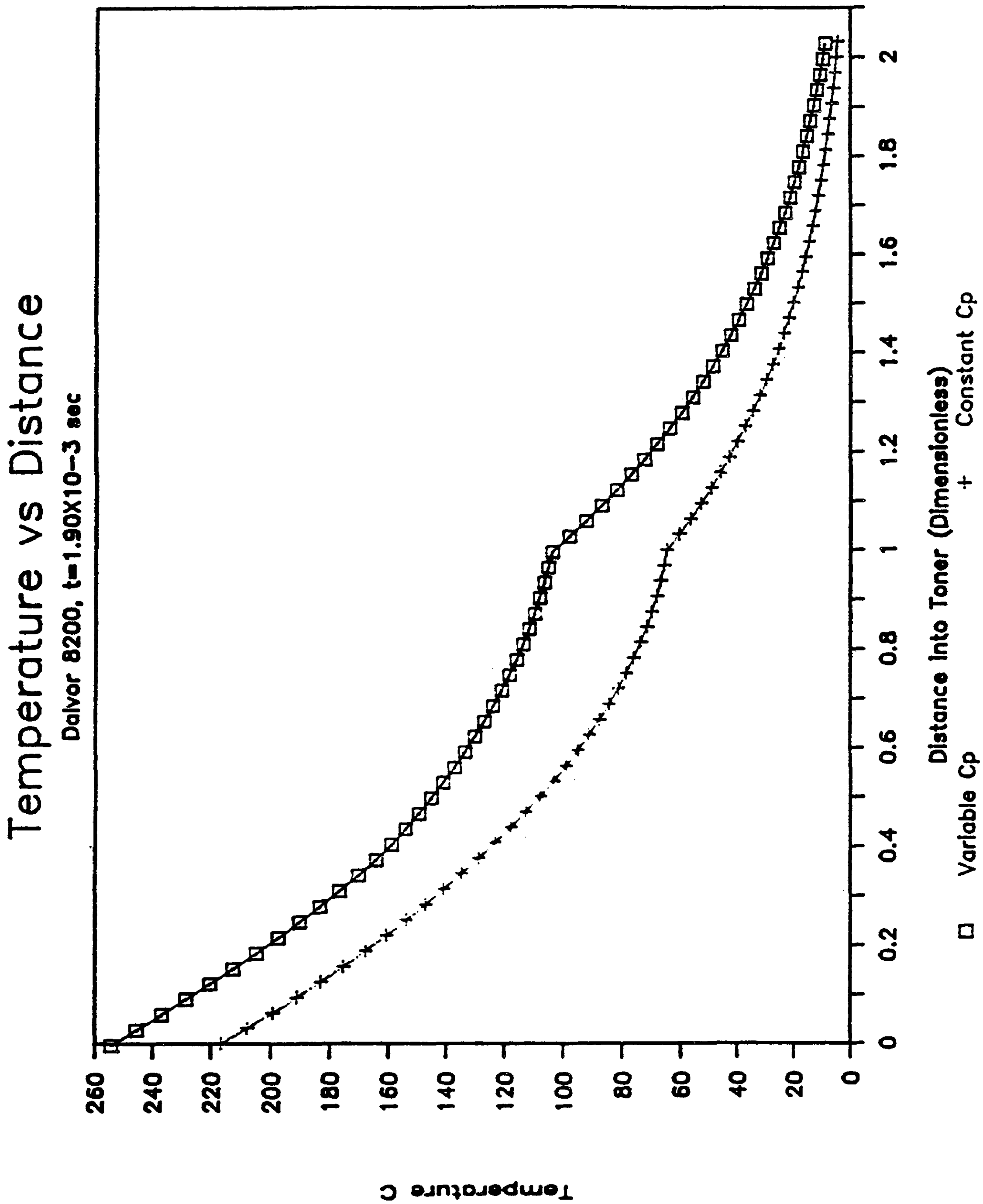

Figure VI.7: Constant vs Variable C_p, Dalvor 8200, t=1.9X10-3 sec

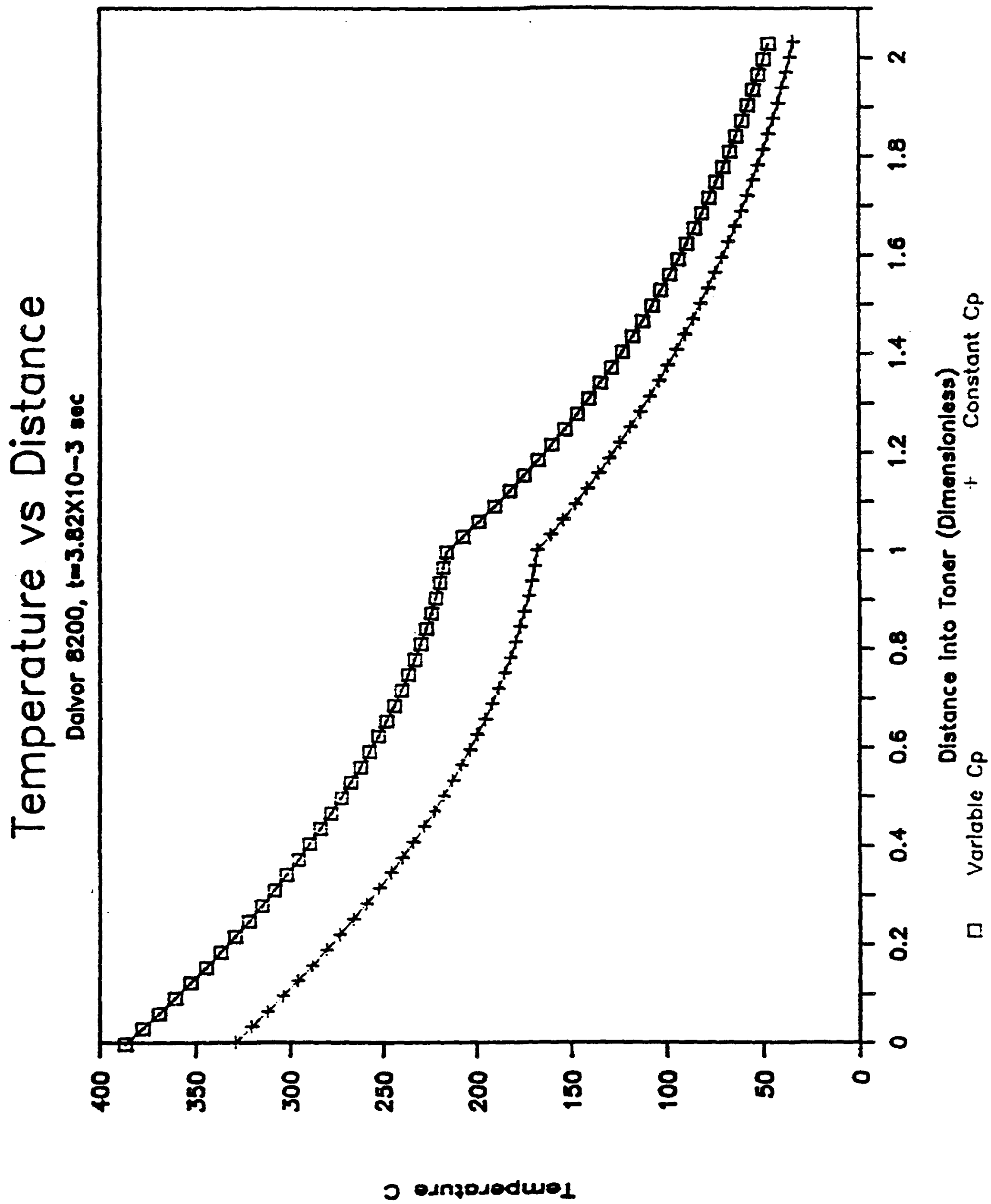

Figure VI.8: Constant vs Variable C_p, Dalvor 8200, t=3.8X10-3 sec

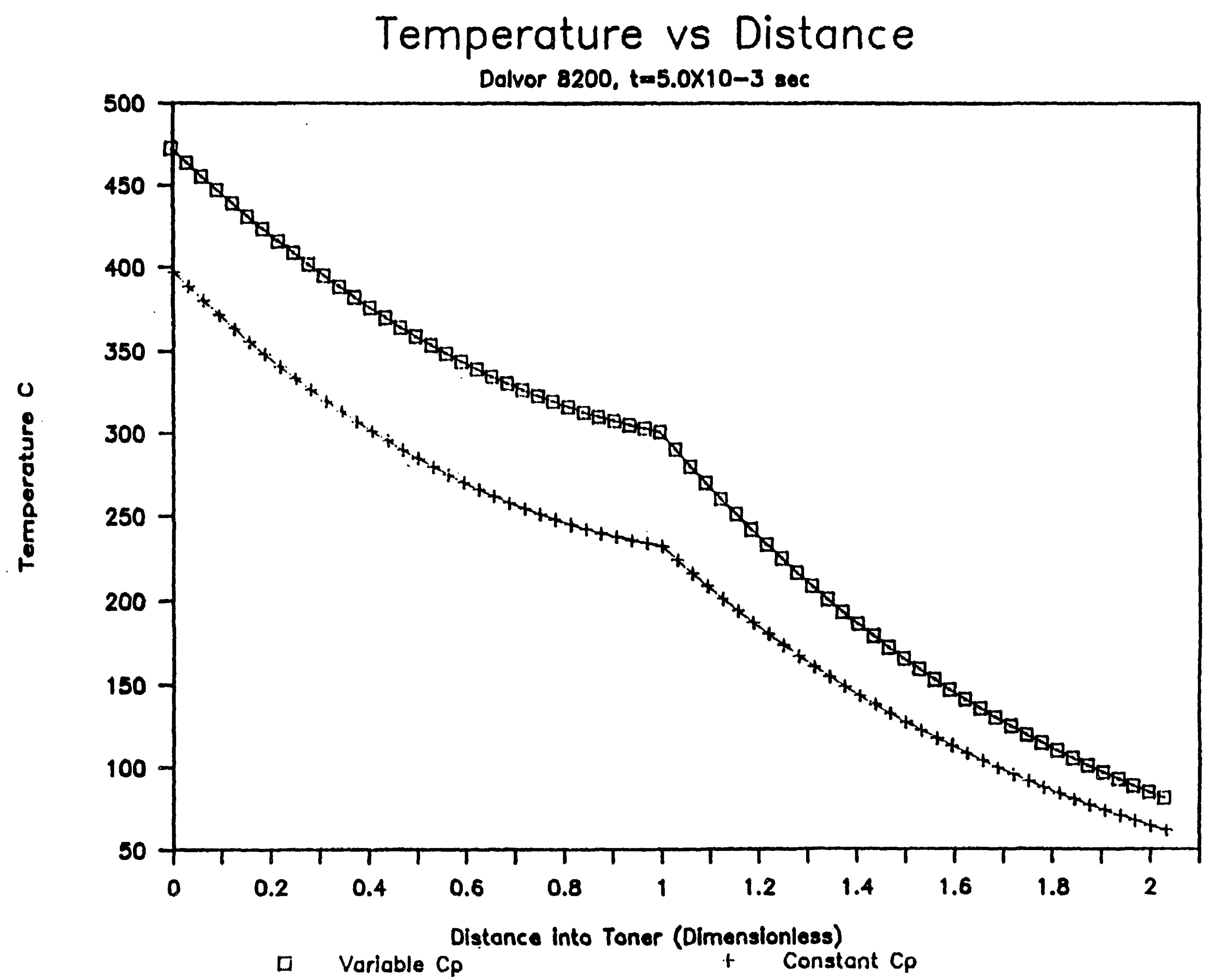

Figure VI.9: Constant vs Variable C_p, Dalvor 8200, t=5.0X10-3 sec

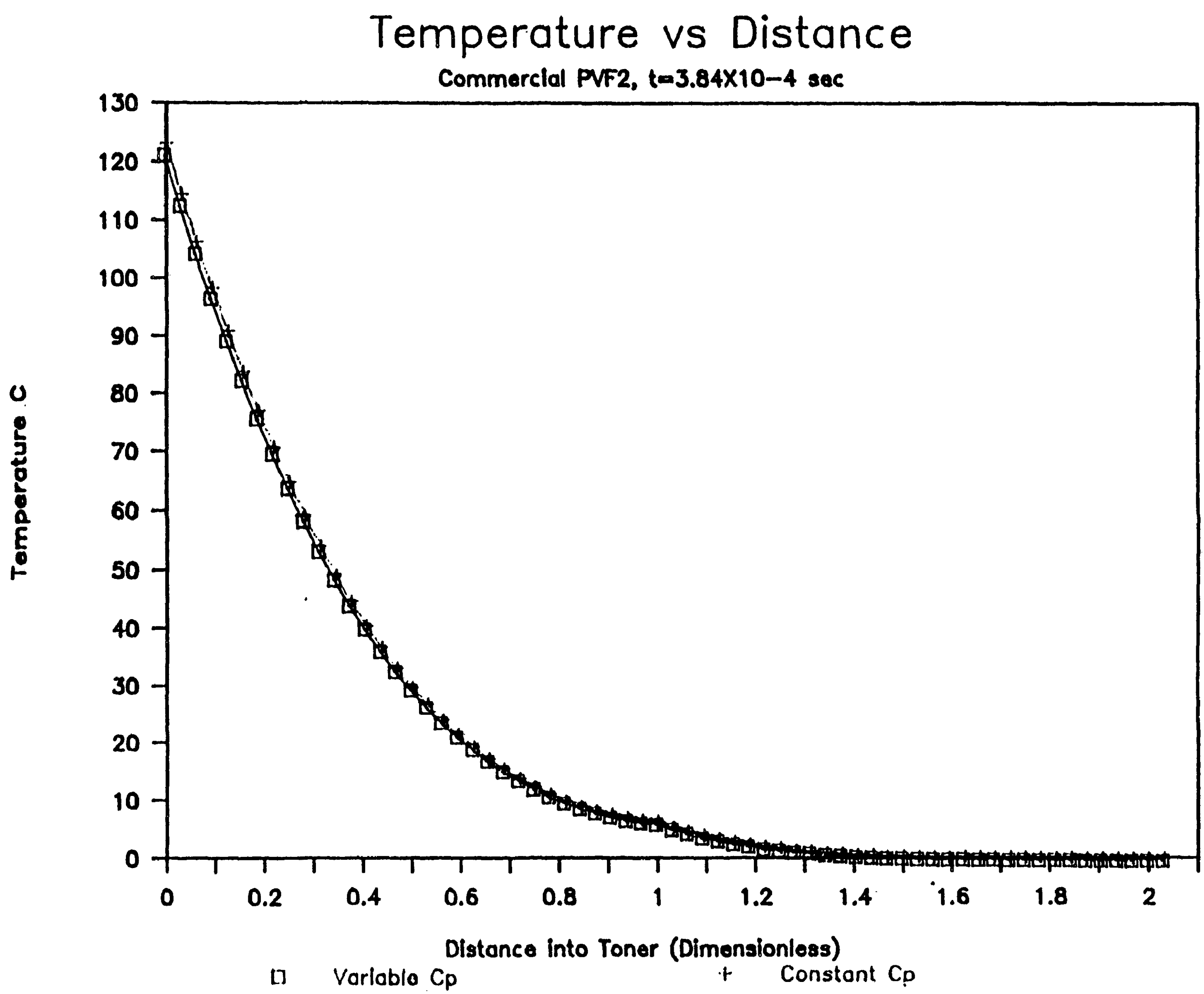

Figure VI.10: Constant vs Variable C_p, PVF_2, t=3.8X10-4 sec

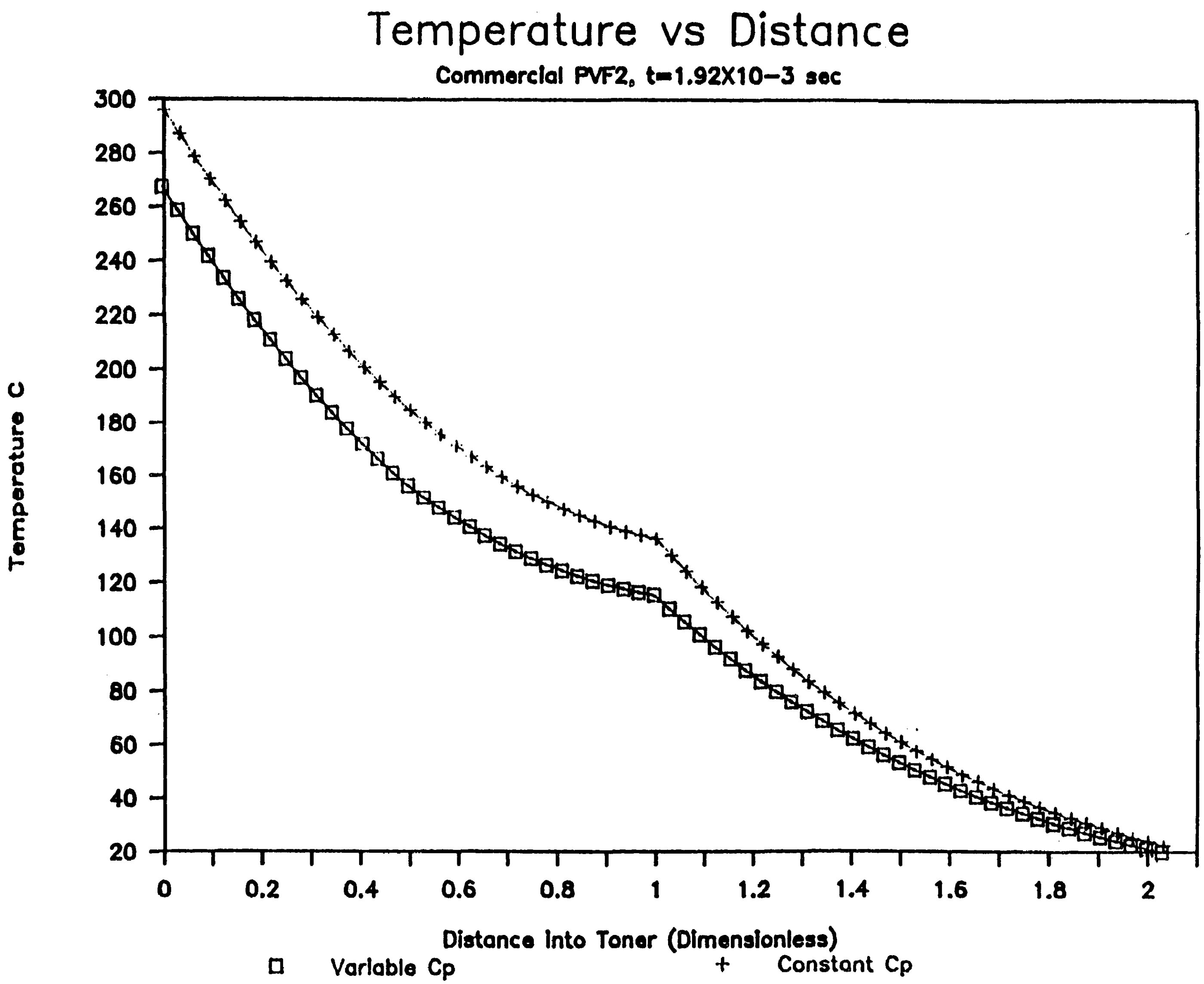

Figure VI.11: Constant vs Variable Cp, PVF_2, t=1.9X10-3 sec

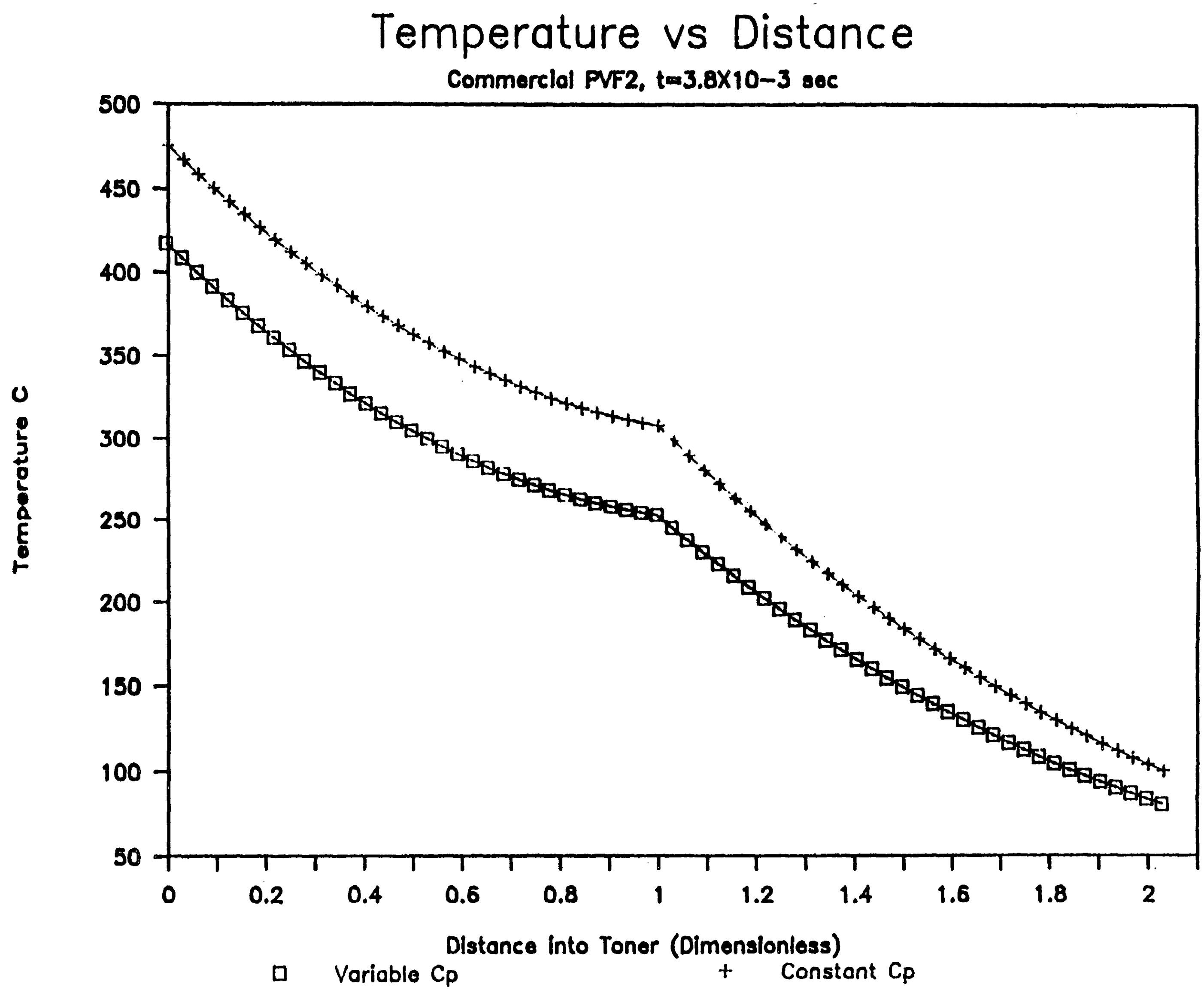

Figure VI.12: Constant vs Variable C_p, PVF_2, t=3.8X10-3 sec

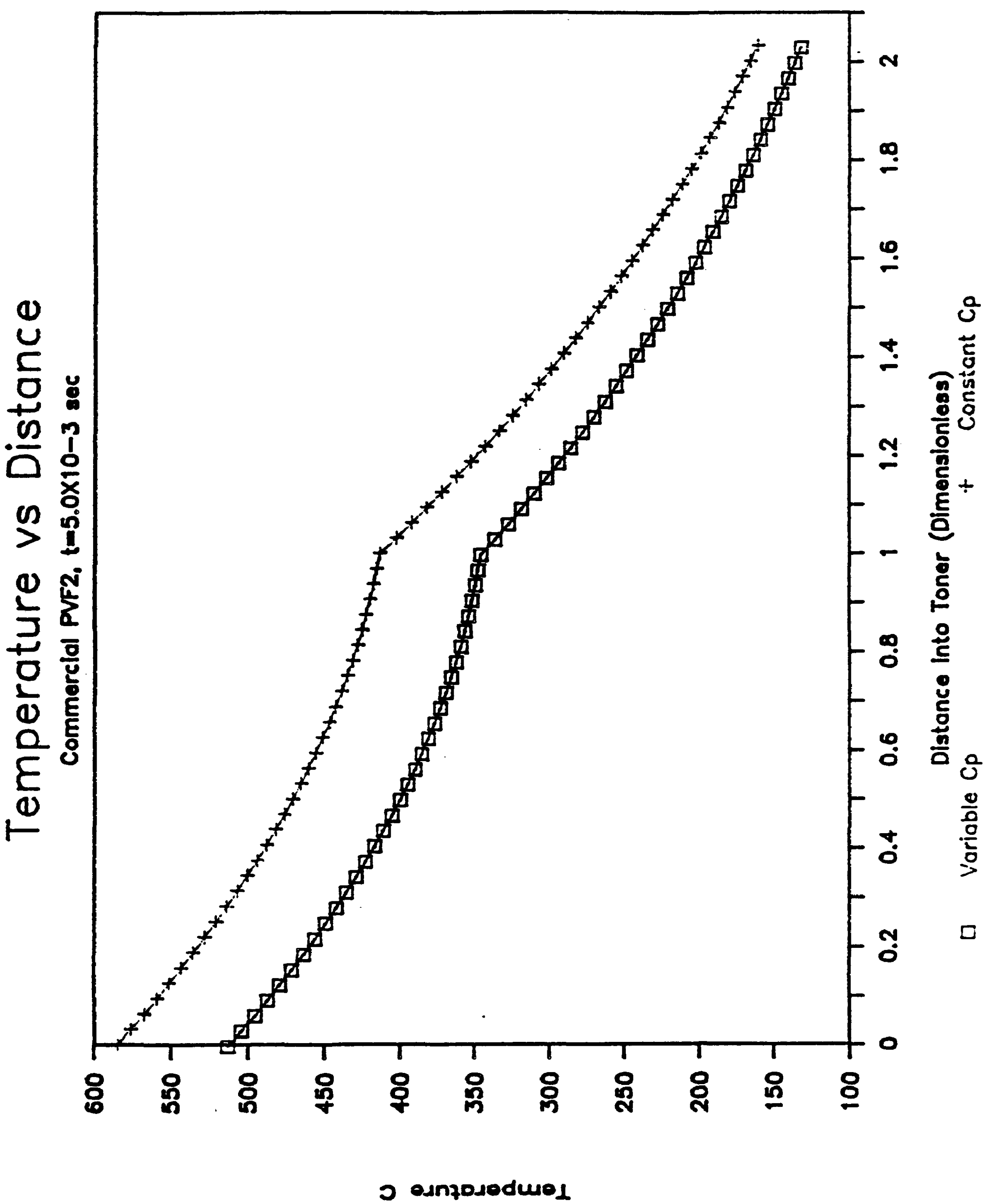

Figure VI.13: Constant vs Variable C_p, PVF_2, t=5.0X10-3 sec

the values in the variable C_p case the resulting variable C_p curve had an higher surface temperature and higher overall average temperature. And when the C_p value for the constant property case was below the values for the variable C_p case the resulting variable C_p curve had a lower overall temperature and lower surface temperature. Adding paper lowered the temperature for the surface of the polymer and overall for the temperature gradients (See Table VI.1). The difference between the temperatures resulting in including paper and excluding paper increased with time. For the Dalvor 8200, at very low times the surface of the polymer did not change: both were 120^{o}C at the time 3.84X10-4 sec. The temperature at the paper surface was 4.2^{o}C. At a time of 1.95X10-3 sec the surface of the polymer went from 261^{o}C without paper to 255^{o}C with paper. The surface temperature of the paper at this time was 99^{o}C. At very long times or 5.0X10-3 sec, the temperature difference of the polymer surface was 47^{o}C; temperature without paper being 521^{o}C and with paper at 474^{o}C. The temperature of the paper surface for a time of 5.0X10-3 sec was 292^{o}C. When evaluating the Commercial PVF2 polymer the same temperature trends occurs. At very small times there is no difference in the polymer surface temperature and at the largest time (t=5.0X10-3 sec) the temperature without the paper was 559^{o}C and with the

Table VI.1

Comparison of Surface Temperature for Polymer and Paper

Surface Temperature ^{o}C

time (sec)	Polymer Without Paper	With Paper	Paper Surface T
3.84X10-4			
D	120°C	120°C	4.2°C
P	121°C	121°C	5.4°C
1.95X10-3			
D	261°C	255°C	99°C
P	274°C	268°C	111°C
3.84X10-3			
D	418°C	388°C	209°C
P	441°C	419°C	247°C
5.0X10-3			
D	512°C	474°C	292°C
P	559°C	514°C	338°C

The D and P are used to distinguish between the resulting temperatures when the polymer used in the analysis was Dalvor 8200 or Commercial PVF2, respectively.

paper resulted in a temperature of 514°C. The temperature at the paper surface ranged from 5.4°C at very low times to 338°C at the end of the time region. Addition of paper to the problem decreased the surface temperature of the polymer in both cases by 45°C to 47°C.

ARTHUR LAKES LIBRARY
COLORADO SCHOOL of MINES
GOLDEN, COLORADO 80401

Chapter VII

Conclusions and Recommendations

Conclusions

The results of the comparison between the analytical solution of the heat equation and the Crank-Nicholson method for the numerical solution produced actual errors of 5.0X10-5 or less throughout the flash fusion process. Both the short term and long term solutions were considered. Throughout the time region of 0 - 5.0X10-3 sec the actual errors were as stated above 5.0X10-4 or less.

When the expansion of the heat equation to include variable thermal conductivity was performed, the resulting curves showed that the thermal conductivity had minimal effect on the temperature distribution throughout the polymer. It furthur showed the surface temperature for the constant property case and the variable property case were the same. As stated in Chapter IV, polymers have a small range for thermal conductivities which result in virtually no effect on temperature during flash fusion. Variable heat capacity was the next property to be evaluated. This variation did have a significant effect on the temperature gradients and the surface temperatures of the polymer. When a polymer has values of heat capacity below a base case the resulting temperature gradient and particulary the surface temperature increases. As time increases, the difference between the base case or constant property case and the variable case also increase. When evaluating the Dalvor 8200 polymer the surface temperature was 120^{o}C at a time of 3.84X10-4 sec. When the flash fusion process was completed at 5.0X10-3 sec the surface temperature of the polymer was 521^{o}C. The constant property case resulted in a polymer surface temperature of 395^{o}C. Conversely, when the values of heat capacity in the variable case were consistantly above that used for the constant property case the resulting surface temperature and gradients were below the base case. This was shown when Commercial PVF2 was evaluated. At small times the temperature at the polymer surface was 121^{o}C and at the end of the flash fusion process (t=5.0X10-3 sec) the temperature was 559^{o}C. The temperature for the constant property case at t=5.0X10-3 sec was 395^{o}C. It should also be noted that the impulse curve for the polymers used in this study had minimal effect on the temperature gradients. When in the regions of the large change in heat capacities the resulting curves had a slope change causing the temperature profiles to flatten in the region of the impulse.

The final problem of this work included both the polymer and the paper modeled as two finite slabs with an interface point. The results of this expansion showed the paper to absorb enough heat causing the surface of the polymer to decrease. When using the Dalvor 8200 polymer, the surface temperature of the polymer went from 521°C to 474°C for the long time (t=5.0X10-3 sec). For smaller times the temperature difference was less. For the Commercial PVF2 the resulting surface temperature of the polymer was 514°C with the paper and 559°C without. When using Dalvor 8200 the surface temperature of the paper ranged from 42°C to 292°C in temperature. The Commercial PVF2 polymer had paper surface temperatures of 54°C to 338°C.

Recommendations

Recommendations for furthur work include varying the heat flux as a function of temperature and including variable density.

REFERENCES CITED

1. Smith, G. D. Numerical Solution of Partial Differential Equations: Finite Difference Methods. Third Edition. Oxford: Clarendon Press, 1987.

2. Tadmor, Zehev and Goges, Costas G. Principles of Polymer Processing. New York: John Wiley & Sons, 1979.

3. Carnahan, Brice and Luther, H. A., Wilkes, James O. Applied Numerical Methods. New York: John Wiley and Sons, 1969.

4. Bird, R. Byron, Stewart, Warren E. and Lightfoot, Edwin N. Transport Phenomena. New York: John Wiley & Sons, 1960.

5. Chapman, Alan J. Heat Transfer, Fourth Edition. New York: Macmillan Publishing Company, 1984.

6. Rosenberger, Dale V. Modern Analytic and Computational Methods in science and Mathematics. University of S. California: American Elsevier Publishing Co, 1969.

7. Data obtained from IBM.

8. Private conversations with Dr. S. Selim and Dr. V. Yesavage.

APPENDIX A

CRANK-NICHOLSON PROGRAM

```
C$ #10 BY: IGELSRUD DATE: 02/23/89 MODIFIED TO INCLUDE CRANK NICHOLSON
C$ #9 BY: IGELSRUD DATE: 02/08/89 ADDED RELATIVE ERROR CALCULATION
C$ #8 BY: IGELSRUD DATE: 02/07/89 ADDED CALL TO TRIDAG
C$ #7 BY: IGELSRUD DATE: 02/05/89 MODIFIED NUMBERING OF OUTPUT VARIABLES
C$ #6 BY: IGELSRUD DATE: 02/05/89 CHANGED M CALCULATION PLUS ONE
C$ #5 BY: IGELSRUD DATE: 02/05/89 MODIFIED FORMAT STATEMENTS
C$ #4 BY: IGELSRUD DATE: 12/30/88 ADDED LOOP FOR MULTIPLE RUNS
C$ #3 BY: IGELSRUD DATE: 12/28/88 ALTERATION OF INPUT INFORMATION
C$ #2 BY: IGELSRUD DATE: 12/27/88 MODIFIED TO INCLUDE CRANK NICHOLSON
METHOD
C$ #1 BY: IGELSRUD DATE: 12/27/88 PROGRAM INITIALLY TAKEN FROM CARNAHAN,
ET AL
C
C
        IMPLICIT REAL*8 (A-H,O-Z)
C       DOUBLE PRECISION
        REAL LAMBDA
C
C
        OPEN(12,STATUS='NEW',FILE='OUTPUT.DAT')
        OPEN(16,STATUS='NEW',FILE='ACTER2.DAT')
        OPEN(17,STATUS='NEW',FILE='EXACT.DAT')
        OPEN(18,STATUS='NEW',FILE='ACTER1.DAT')
        DIMENSION A(900), B(900), C(900), TOLD(900), D(900), WXSTR(10)
C
C    CHECK INPUT PARAMETERS
C
10      WRITE(*,101)
        READ(*,100) LAMBDA
        WRITE(*,102)
        READ(*,111) JMAX
        WRITE(*,103)
        READ(*,111) M
        FLOATI = M
        DX = 1.D0 / FLOATI
        DTAU = LAMBDA * (DX**2)
        JDONE = 0
        WRITE(12,200)  DTAU, DX, JMAX, M, LAMBDA
C
C  CALCULATE THE CONSTANTS FOR THIS PROBLEM
C
        MP1 = M + 1
        DO 20 I = 2, M
           A(I) = -LAMBDA
           B(I) = 2D0 * (1.D0 + LAMBDA)
           C(I) = A(I)
 20        CONTINUE
           A(1) = 0D0
           B(1) = 1.D0 + LAMBDA
           C(1) = -LAMBDA
           A(MP1) = -LAMBDA
           B(MP1) = 1.D0 + LAMBDA
```

```
          C(MP1) = 0D0
C
C  SET AND PRINT INITIAL TEMPERATURES
C
        DO 30 I = 1, MP1
          FLOATI = I - 1
          TOLD(I) = F( FLOATI * DX )
30      CONTINUE
        TAU = 0D0
        WRITE (12, 210)
            WRITE (12, 220)  TAU, (TOLD(I), I=1,MP1)
C
C  PERFORM CALCULATIONS OVER SUCCESSIVE TIME-STEPS
C
40      TAU = TAU + DTAU
        JDONE = JDONE + 1
C
C
C  SET RIGHT-HAND SIDE D VECTOR FOR BOUNDARY CONDITIONS
C
        D1C1 = 1.D0 - LAMBDA
        D1C2 = LAMBDA
        D1C3 = 2.D0 * LAMBDA * DX * G0(TAU)
        D(1) = D1C1 * TOLD(1) + D1C2 * TOLD(2) - D1C3
        DMP1C1 = LAMBDA
        DMP1C2 = 1.D0 - LAMBDA
        DMP1C3 = 2.D0 * LAMBDA * DX * G1(TAU)
        D(MP1) = DMP1C1 * TOLD(M) + DMP1C2 * TOLD(MP1) + DMP1C3
C
C
C COMPUTE RIGHT-HAND SIDE VECTOR D  FOR INSIDE GRID
C
        CST1 = LAMBDA
        CST2 = 2.D0 * (1.D0 - LAMBDA)
        CST3 = LAMBDA
        DO 50 I = 2, M
          D(I) = CST1 * TOLD(I-1) + CST2 * TOLD(I) + CST3 * TOLD(I+1)
50      CONTINUE
C
C  COMPUTE NEW VALUES OF THE FUNCTION U
C
        CALL TRIDAG ( 1, MP1, A, B, C, D, TOLD )
C
C   PRINT TEMPERATURES
C
        WRITE (12, 220)  TAU, (TOLD(I), I=1,MP1)
C
C   CALCULATE THE ACTUAL ERROR BETWEEN THE ACTUAL AND
C   NUMERICAL SOLUTION (CRANK-NICHOLSON)
C
        CALL RELCNN(TAU,DX,M,TOLD,MP1,WXSTR)
        IF(JDONE.EQ.1) THEN
```

```
C  WRITE THE HEADINGS FOR THE ACTUAL ERROR
C  TABLE
C
        WRITE(18,152) LAMBDA,(WXSTR(J),J=1,5)
        WRITE(16,156) LAMBDA,(WXSTR(J),J=6,10)
        ELSE
        CONTINUE
        ENDIF
        IF( JDONE .LE. JMAX ) GO TO 40
        GO TO 10
C
C   FORMAT STATEMENTS
C
100     FORMAT(G12.6)
111     FORMAT(I5)
101     FORMAT(1X,'ENTER THE VALUE FOR LAMBDA - R')
102     FORMAT(1X,'ENTER THE VALUE FOR JMAX - TIME GRID MAX')
103     FORMAT(1X,'ENTER THE VALUE FOR M - # INTERVALS')
155     FORMAT(1X,'CRANK-NICHOLSON METHOD, WITH PARAMETERS'/
     1  'DTAU = ',G13.7,3X,'DX = ',G13.7/3X,'RELATIVE ERROR = ',
     2  G13.7//'TCN = ',G12.5,3X,'EXAN = ',G12.5,//'MDP1 = ',I5)
152     FORMAT(1X,13X,'ACTUAL ERROR ANALYSIS FOR THE CRANK-NICH',
     1'OLSON SCHEME'/14X,'COMPARED WITH THE ANALYTICAL SOLUTION',
     2/1X,'LAMBDA = ',G12.5//1X,'X =',5(3X,F7.4))
156     FORMAT(1X,15X,'ACTUAL ERROR ANALYSIS FOR THE CRANK-NICH',
     1'OLSON SCHEME'/16X,'COMPARED WITH THE ANALYTICAL SOLUTION',
     2/1X,'LAMBDA = ',G12.5//1X,'X =',5(3X,F7.4))
200     FORMAT(1X,/
     1  'CRANK-NICHOLSON METHOD, WITH PARAMETERS'/
     2  'DTAU = ',G13.7/'DX = ',G13.7/'JMAX = ',I4/'M = ',I4/
     3  'LAMBDA = ',G13.7)
210     FORMAT(1X,'TIME',18X,'VALUES OF U AT THE GRIDPOINTS')
220     FORMAT(1X, F7.3/1X, 7X, 5(G13.7,2X)/8X,5(G13.7,2X)/8X,G13.7)
227     FORMAT(1X, F7.3/6(8X,5(G13.7,2X)/)8X,G13.7)
229     FORMAT(1X, F7.3/4(8X,5(G13.7,2X)/)8X,G13.7)
223     FORMAT(1X, F7.3/8(8X,5(G13.7,2X)/)8X,G13.7)
231     FORMAT(1X, F7.3/10(8X,5(G13.7,2X)/)8X,G13.7)
        END
        REAL*8 FUNCTION F(DIST)
        IMPLICIT REAL*8 (A-H,O-Z)
        F = 0D0
        RETURN
        END
        REAL*8 FUNCTION G0(TIME)
        IMPLICIT REAL*8 (A-H,O-Z)
        G0 = -1.D0
        RETURN
        END
        REAL*8 FUNCTION G1(TIME)
        IMPLICIT REAL*8 (A-H,O-Z)
        G1 = 0D0
        RETURN
        END
```

```
C$ BY:IGELSRUD DATE: 04/09/89 FIRST ABSORB
        SUBROUTINE RELCNN (TAU,DX,M,TOLD,MP1,WXSTR)
        IMPLICIT REAL*8 (A-H,O-Z)
C
C     PRINT THE VALUES FOR THE RELATIVE ERROR FOR X VALUES
C     FROM 0.0 TO 1.0
C
        DIMENSION  REER(10),TOLD(MP1),AERR(10),WXSTR(10)
        MD10 = M / 10
        TSTAR = TAU
        DO 20 J=1,10
           MDT = MD10 * J + 1
           MDX = MDT - 1
           XSTAR = DX * MDX
           WXSTR(J) = XSTAR
           CALL EXACT(XSTAR,TSTAR,EXAN,SUM)
             AERR(J) = (EXAN - TOLD(MDT))
             WRITE(17,157) EXAN,TOLD(MDT),MDT,XSTAR,TSTAR,SUM
20      CONTINUE
           WRITE(18,160) TSTAR,(AERR(J),J=1,5)
           WRITE(16,160) TSTAR,(AERR(J),J=6,10)
C
C   FORMAT STATEMENTS
C
154     FORMAT(1X,F7.4,5(2X,F10.7))
155     FORMAT(1X,F7.4,5(2X,F10.7))
160     FORMAT(1X,F7.4,5(1X,G12.5))
157     FORMAT(1X,15X,'EXACT SOLUTION VALUES TO CN VALUES'
     2//1X,'EXACT = ',G12.5,5X,'CRANK = ',G12.5//3X,'MDT = ',I5,
     33X,'XSTAR = ',G12.5,3X,'TSTAR = ',G12.5,3X,'SUM = ',G12.5)
         RETURN
         END
```

```
C$ BY:IGELSRUD DATE: 02/23/89 FIRST ABSORB
      SUBROUTINE EXACT(XSTAR,TSTAR,TEMEXA,SUM)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 LAMBN
C
C     CALCULATE CONSTANTS
C
      ONETHD = 1.D0/3.D0
      ONEHLF = 1.D0/2.D0
      SUM = 0D0
C
C     CALCULATE COSINE AND EXPONENTIAL SUMMATION
C
      DO 10 N = 1,500
         LAMBN = 3.14159265358979323 8D0*N
         VALUE = (-LAMBN**2)*TSTAR
         ARG1 = DEXP(VALUE)
         IF(ARG1.LT.1.D-15) GO TO 20
         ARG2 = DCOS(LAMBN*XSTAR)
         ARG3 = ARG1 * ARG2
         CST2 = -2.D0 / (LAMBN**2)
         SUM = SUM + (CST2)* ARG3
10    CONTINUE
20    CONTINUE
C
C     COMPUTE FINAL SOLUTION FOR TEMPERATURE
C
      TEMEXA = ONEHLF*(XSTAR**2) - XSTAR + TSTAR + ONETHD
     1         + SUM
      RETURN
      END
```

```
C$ BY:IGELSRUD DATE: 12/27/88 FIXED GAMMA CALCULATION
C$ BY:IGELSRUD DATE: 12/26/88 FIRST ABSORB
      SUBROUTINE TRIDAG (IF, L, A, B, C, D, U)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(L), B(L), C(L), D(L), U(L), BETA(951), GAMMA(951)
C
C    COMPUTE INTERMEDIATE ARRAYS BETA AND GAMMA
C
      BETA(IF) = B(IF)
      GAMMA(IF) = D(IF) / BETA(IF)
      IFP1 = IF + 1
      DO 10 I = IFP1, L
         BETA(I) = B(I) - A(I) * C(I-1) / BETA(I-1)
         GAMMA(I) = (D(I) - A(I) * GAMMA(I-1)) / BETA(I)
10    CONTINUE
C
C    COMPUTE FINAL SOLUTION VECTOR U
C
      U(L) = GAMMA(L)
      LAST = L - IF
      DO 20 K = 1, LAST
         I = L - K
         U(I) = GAMMA(I) - C(I) * U(I+1) / BETA(I)
20    CONTINUE
      RETURN
      END
```

APPENDIX B

VARIABLE PROPERTIES WITH NO PAPER INTERFACE

```
C$ #14 BY: IGELSRUD DATE: 10/20/89 MODIFIED FOR VARIABLE CP
C$ #13 BY: IGELSRUD DATE: 09/10/89 INCLUDED NONLINEAR THERMOCONDUCTIVITY
C$ #12 BY: IGELSRUD DATE: 02/23/89 MODIFIED TO INCLUDE CRANK NICHOLSON
C$ #11 BY: IGELSRUD DATE: 02/08/89 ADDED RELATIVE ERROR CALCULATION
C$ #10 BY: IGELSRUD DATE: 02/07/89 ADDED CALL TO TRIDAG
C$ #9 BY: IGELSRUD DATE: 02/07/89 MODIFIED FOR BACKWARD DIFFERENCEMA550
C$ #8 BY: IGELSRUD DATE: 02/05/89 MODIFIED NUMBERING OF OUTPUT VARIABLES
C$ #7 BY: IGELSRUD DATE: 02/05/89 CHANGED M CALCULATION PLUS ONE
C$ #6 BY: IGELSRUD DATE: 02/05/89 MODIFIED FORMAT STATEMENTS
C$ #5 BY: IGELSRUD DATE: 02/03/89 MODIFIED FOR FORWARD DIFFERENCE - MA550
C$ #4 BY: IGELSRUD DATE: 12/30/88 ADDED LOOP FOR MULTIPLE RUNS
C$ #3 BY: IGELSRUD DATE: 12/28/88 ALTERATION OF INPUT INFORMATION
C$ #2 BY: IGELSRUD DATE: 12/27/88 MODIFIED TO INCLUDE CRANK NICHOLSON
METHOD
C$ #1 BY: IGELSRUD DATE: 12/27/88 PROGRAM INITIALLY TAKEN FROM CARNAHAN,
ET AL
C
C
        IMPLICIT REAL*8 (A-H,O-Z)
C       DOUBLE PRECISION
        REAL LAMBDA,KMH(100),KPH(100),K1N,KMP1,K1N1,KMP1N1
        DIMENSION A(100), B(100), C(100), TOLD(100), D(100),
     1  TOLDL(100),TNEW(100),CPVN(100),CPVNP1(100),TEMP(100)
C
C
        OPEN(12,STATUS='NEW',FILE='OUTPUT.DAT')
        OPEN(16,STATUS='NEW',FILE='KVALUE.DAT')
        OPEN(17,STATUS='NEW',FILE='EXACT.DAT')
        OPEN(18,STATUS='NEW',FILE='TEMP.DAT')
C
C    CHECK INPUT PARAMETERS
C
10      WRITE(*,101)
        READ(*,100) LAMBDA
        WRITE(*,102)
        READ(*,111) JMAX
        WRITE(*,103)
        READ(*,111) M
        FLOATI = M
        DX = 1.D0 / FLOATI
        DTAU = LAMBDA * (DX**2)
        JDONE = 0
        WRITE(12,200)  DTAU, DX, JMAX, M, LAMBDA
        MP1 = M + 1
        TAU = 0D0
C
C  SET AND PRINT INITIAL TEMPERATURES
C
        DO 30 I = 1, MP1
           FLOATI = I - 1
           TOLD(I) = F( FLOATI * DX )
           TOLDL(I) = TOLD(I)
```

```
          TNEW(I) = TOLD(I)
30      CONTINUE
C
C  CALCULATE THE INITIAL VALUES FOR THE THERMOCONDUCTIVITY AT
C  I+1 AND I-1
C
        CALL KVALUE(TOLD,KPH,KMH,K1N,KMP1,M,MP1,JDONE,DX,TAU,DTAU,TOLDL,
     1             K1N1,KMP1N1)
C
C  CALCULATE THE CPV VALUES FOR THE CONSTANTS
C
        CALL CPV(CPVN,CPVNP1,MP1,JDONE,TAU,DTAU,TOLD,TOLDL,M)
C
C  CALCULATE THE CONSTANTS FOR THIS PROBLEM
C
        DO 20 I = 2, M
          A(I) = -KMH(I)
          B(I) = ( 2.D0*CPVNP1(I) / LAMBDA) + KPH(I) + KMH(I)
          C(I) = -KPH(I)
 20       CONTINUE
C
C  CALCULATE THE CONSTANTS AT THE BOUNDARIES
C
          A(1) = 0D0
          B(1) =(2.D0* CPVNP1(1)/ LAMBDA) + KPH(1) + KMH(1)
          C(1) = -1.D0 * ( KMH(1) + KPH(1) )
          A(MP1) = -1.D0 * (KMH(MP1) + KPH(MP1))
          B(MP1) = (2.D0 * CPVNP1(M) / LAMBDA) + KPH(MP1) + KMH(MP1)
          C(MP1) = 0D0
C
C INITIALIZE TAU TO START LOOP
C
        TAU = 0D0
        WRITE (12, 210)
        WRITE (12, 220)  TAU, (TOLD(I), I=1,MP1)
C
C  PERFORM CALCULATIONS OVER SUCCESSIVE TIME-STEPS
C
40      TAU = TAU + DTAU
        JDONE = JDONE + 1
C
C
C  SET RIGHT-HAND SIDE D VECTOR FOR BOUNDARY CONDITIONS
C
221     D1C1 = (2.D0 * CPVN(1) / LAMBDA) - KPH(1) - KMH(1)
        D1C2 = KMH(1) + KPH(1)
        D1C3 = 2.D0 * KMH(1) * DX * G0(TAU) *
     1   (( 1.0D0 / K1N ) + ( 1.0D0 / K1N1 ))
        D(1) = D1C1 * TOLDL(1) + D1C2 * TOLDL(2) - D1C3
        DMP1C1 = KMH(MP1) + KPH(MP1)
        DMP1C2 = (2.D0 * CPVN(M) / LAMBDA) - KPH(MP1) - KMH(MP1)
        DMP1C3 = 2.0D0 * KPH(MP1) * DX * G1(TAU) *
```

```
     1   (( 1.0D0 / KMP1 ) + ( 1.0D0 / KMP1N1 ))
       D(MP1) = DMP1C1 * TOLDL(M) + DMP1C2 * TOLDL(MP1) + DMP1C3
C
C
C COMPUTE RIGHT-HAND SIDE VECTOR D  FOR INSIDE GRID
C
        DO 50 I = 2, M
        CST1 = KMH(I)
        CST2 = (2.D0*CPVN(I) / LAMBDA) - KPH(I) - KMH(I)
        CST3 = KPH(I)
           D(I) = CST1 * TOLDL(I-1) + CST2 * TOLDL(I) +
     1     CST3 * TOLDL(I+1)
50      CONTINUE
C
C  COMPUTE NEW VALUES OF THE FUNCTION TEMPERATURE TOLD
C
        CALL TRIDAG ( 1, MP1, A, B, C, D, TOLD )
C
C   CHECK TEMPERATURES FOR CONVERGENCE
C
        DO 250 J=1,MP1
        RES = DABS(TNEW(J) - TOLD(J))
        IF(RES.GT.1.0D-6) THEN
C
C   UPDATE OLD ITERATION TEMPERATURE WITH NEW ITERATION TEMPERATURE
C
        DO 750 I=1,MP1
        TNEW(I) = TOLD(I)
750     CONTINUE
C
C   UPDATE THE VALUES OF K - THERMOCONDUCTIVITY
C
       CALL KVALUE(TOLD,KPH,KMH,K1N,KMP1,M,MP1,JDONE,DX,TAU,DTAU,TOLDL,
     1              K1N1,KMP1N1)
C
C  CALCULATE THE CPV VALUES FOR THE CONSTANTS
C
       CALL CPV(CPVN,CPVNP1,MP1,JDONE,TAU,DTAU,TOLD,TOLDL,M)
C
C   UPDATE THE CONSTANTS A, B, AND C
C
        DO 29 I = 2, M
           A(I) = -KMH(I)
           B(I) = (2.0D0*CPVNP1(I)/ LAMBDA) + KPH(I) + KMH(I)
           C(I) = -KPH(I)
 29        CONTINUE
C
C  CALCULATE THE CONSTANTS AT THE BOUNDARIES
C
           A(1) = 0D0
           B(1) =(2.D0 * CPVNP1(1) / LAMBDA) + KPH(1) + KMH(1)
           C(1) = -1.D0 * ( KMH(1) + KPH(1) )
```

ARTHUR LAKES LIBRARY
COLORADO SCHOOL of MINES
GOLDEN, COLORADO 80401

```
          A(MP1) = -1.D0 * (KMH(MP1) + KPH(MP1))
          B(MP1) = (2.D0 * CPVNP1(M)/ LAMBDA) + KPH(MP1) + KMH(MP1)
          C(MP1) = 0D0
            GO TO 221
            ELSE
            ENDIF
250       CONTINUE
C
C  UPDATE THE TEMPERATURES FOR THE NEXT PASS
C
           DO 222 I=1,MP1
               TOLDL(I) = TOLD(I)
222        CONTINUE
C
C   PRINT TEMPERATURES WHEN APPROPRIATE
C
C
C   PRINT TEMPERATURES WHEN APPROPRIATE
C
            IF(TAU.EQ.0.1015625.OR.TAU.EQ.0.5.OR.
     1      TAU.EQ.0.75.OR.TAU.EQ.1.0.OR.
     1      TAU.EQ.1.25.OR.TAU.EQ.1.3203125) THEN
            CALL CONVTT(TOLD,TEMP,MP1)
            WRITE (12, 220)  TAU, (TEMP(I), I=1,MP1)
            WRITE (18, 299)  TAU, (TEMP(I), I=1,MP1)
299         FORMAT(1X,G13.7/1X,G13.7)
            ENDIF
C
C   UPDATE THE VALUES OF K - THERMOCONDUCTIVITY
C
       CALL KVALUE(TOLD,KPH,KMH,K1N,KMP1,M,MP1,JDONE,DX,TAU,DTAU,TOLDL,
     1            K1N1,KMP1N1)
C
C  CALCULATE THE CPV VALUES FOR THE CONSTANTS
C
       CALL CPV(CPVN,CPVNP1,MP1,JDONE,TAU,DTAU,TOLD,TOLDL,M)
C
C   UPDATE THE CONSTANTS A, B, AND C
C
        DO 27 I = 2, M
           A(I) = -KMH(I)
           B(I) = (2D0*CPVNP1(I) / LAMBDA) + KPH(I) + KMH(I)
           C(I) = -KPH(I)
 27        CONTINUE
C
C  CALCULATE THE CONSTANTS AT THE BOUNDARIES
C
           A(1) = 0D0
           B(1) =(2.D0 * CPVNP1(1)/ LAMBDA) + KPH(1) + KMH(1)
           C(1) = -1.D0 * ( KMH(1) + KPH(1) )
           A(MP1) = -1.D0 * (KMH(MP1) + KPH(MP1))
           B(MP1) = (2.D0 * CPVNP1(M)/ LAMBDA) + KPH(MP1) + KMH(MP1)
```

```
          C(MP1) = 0D0
C
C CHECK TO SEE IF PROGRAM REACHED MAXIMUM  J VALUE
C
      IF( JDONE .LE. JMAX ) GO TO 40
      GO TO 10
C
C   FORMAT STATEMENTS
C
100   FORMAT(G12.6)
111   FORMAT(I5)
101   FORMAT(1X,'ENTER THE VALUE FOR LAMBDA - R')
102   FORMAT(1X,'ENTER THE VALUE FOR JMAX - TIME GRID MAX')
103   FORMAT(1X,'ENTER THE VALUE FOR M - # INTERVALS')
200   FORMAT(1X,/
     1 'CRANK-NICHOLSON METHOD, WITH PARAMETERS'/
     2 'DTAU = ',G13.7/'DX = ',G13.7/'JMAX = ',I4/'M = ',I4/
     3 'LAMBDA = ',G13.7)
210   FORMAT(1X,'TIME',18X,'VALUES OF U AT THE GRIDPOINTS')
220   FORMAT(1X, F7.3/1X, 7X, 5(G13.7,2X)/8X,5(G13.7,2X)/8X,G13.7)
227   FORMAT(1X, F7.3/6(8X,5(G13.7,2X)/)8X,G13.7)
229   FORMAT(1X, F7.3/4(8X,5(G13.7,2X)/)8X,G13.7)
223   FORMAT(1X, F7.3/8(8X,5(G13.7,2X)/)8X,G13.7)
231   FORMAT(1X, F7.3/10(8X,5(G13.7,2X)/)8X,G13.7)
      END
      REAL*8 FUNCTION F(DIST)
      IMPLICIT REAL*8 (A-H,O-Z)
      F = 0D0
      RETURN
      END
      REAL*8 FUNCTION G0(TIME)
      IMPLICIT REAL*8 (A-H,O-Z)
      G0 = -1.0D0
      RETURN
      END
      REAL*8 FUNCTION G1(TIME)
      IMPLICIT REAL*8 (A-H,O-Z)
      G1 = 0D0
      RETURN
      END
```

```
C$ BY: IGELSRUD DATE: 10/07/89 FIRST ABSORB
       SUBROUTINE CPV(CPVN,CPVNP1,MP1,JDONE,TAU,DTAU,
     1                TOLD,TOLDL,M)
       IMPLICIT REAL*8 (A-H,O-Z)
      DATA TEMPA /0.0,76.0,80.0,85.0,90.0,95.0,100.0,105.0,
     2 110.0,115.0,120.0,125.0,130.0,135.0,140.0,145.0,150.0,
     3 155.0,160.0,162.5,165.0,170.0,172.5,175.0,180.0,185.0,
     4 190.0,190.0/
      DATA TEMPB /0.0,82.0,90.0,95.0,100.,105.,110.,115.,120.0,
     1 125.0,127.0,130.0,135.0,140.0,145.0,148.0,150.0,152.0,155.0,
     2 157.5,160.,162.5,165.0,170.0,172.0,175.0,177.5,180.0,180.0/
      DATA CPVA  /0.325,0.325,0.343,0.360,0.365,0.370,0.375,
     1 0.378,0.382,0.388,0.395,0.405,0.415,0.429,0.450,0.455,
     2 0.500,0.700,0.800,0.975,0.770,0.425,0.375,0.376,0.379,
     3 0.382,0.382,0.382/
      DATA CPVB  /0.34,0.34,0.350,0.3675,0.375,0.385,0.395,0.410,
     1 0.420,0.430,0.435,0.440,0.459,0.475,0.4875,0.493,0.50,0.550,
     2 0.70,0.80,0.90,1.0,1.21,0.975,0.800,0.590,0.500,0.400,0.400/
C
C     CALCULATE THE KVALUES AT THE HALF POINTS AND AT THE END POINTS
C
       DIMENSION  TOLD(MP1),TK(200),TOLDL(MP1),TKL(200)
       DIMENSION  CPVN(MP1),CPVNP1(MP1),TEMPA(28),TEMPB(29)
       DIMENSION  CPVA(28),CPVB(29)
       CPVO = 3.25000D-1
C
C
C
C  CALCULATE TEMPERATURES AT THE Nth TIMESTEP
C
       DO 82 I=1,MP1
       TKL(I) = 285.71D0 * TOLDL(I)
82     CONTINUE
C
C   CALCULATE THE CPV'S AT THE Nth TIMESTEP
C
       DO 20 I=1,MP1
          DO 23 II=2,27
          IF(TKL(I).LE.TEMPA(2)) CPVN(I) = CPVA(1)/CPVO
          IF(TKL(I).GT.TEMPA(II).AND.TKL(I).LT.TEMPA(II+1))
     1    CPVN(I) = (CPVA(II) + CPVA(II+1)) / (2.0D0 * CPVO)
          IF(TKL(I).GE.TEMPB(28)) CPVN(I) = CPVA(28)/CPVO
23      CONTINUE
20     CONTINUE
C
C   CALCULATE THE CPV'S AT THE N+1 TIMESTEP ( KPH AND KMH )
C
       DO 22 I=1,MP1
          DO 24 II=2,27
          IF(TKL(I).LE.TEMPA(2)) CPVNP1(I) = CPVA(1)/CPVO
          IF(TKL(I).GT.TEMPA(II).AND.TKL(I).LT.TEMPA(II+1))
     1    CPVNP1(I) = (CPVA(II) + CPVA(II+1)) / (2.0D0 * CPVO)
```

```
      IF(TKL(I).GE.TEMPB(28)) CPVNP1(I) = CPVA(28)/CPV0
24    CONTINUE
22   CONTINUE
      RETURN
      END
```

```
C$ BY: IGELSRUD DATE: 09/23/89 MODIFIED THE K EQUATION
C$ BY: IGELSRUD DATE: 09/11/89 FIRST ABSORB
        SUBROUTINE KVALUE (TOLD,KPH,KMH,K1NN,KMP1N,M,MP1,JDONE,
     1   DX,TAU,DTAU,TOLDL,K1NN1,KMP1N1)
        IMPLICIT REAL*8 (A-H,O-Z)
C
C     CALCULATE THE KVALUES AT THE HALF POINTS AND AT THE END POINTS
C
        DIMENSION  TOLD(MP1),TK(100),TOLDL(MP1),TKL(100)
        REAL K1(100),K2(100),K3(100),KPHN1(100),KMHN1(100)
        REAL KP2,KE1,KE2,K01,K0MP1,K1NN1,KMP1N1
        REAL KP2N,KE1N,KE2N,K01N,K0MP1N,K1NN,KMP1N
        REAL KPHN(100),KMHN(100),K1N(100),K2N(100),K3N(100)
        REAL KPH(MP1),KMH(MP1)
        B = -2.65957D-3
C
C  CALCULATE TEMPERATURES AT THE Nth TIMESTEP
C
        DO 82 I=1,MP1
        TKL(I) = 285.71D0 * TOLDL(I)
82      CONTINUE
C
C   CALCULATE THE K'S AT THE Nth TIMESTEP ( KPH AND KMH )
C
        K1NN   = 1.0D0 +  B * TKL(1)
        KMP1N  = 1.0D0 + B * TKL(MP1)
        DO 20 I=2,M
        K1N(I) = 1.0D0 + B * TKL(I+1)
        K2N(I) = 1.0D0 + B * TKL(I)
        K3N(I) = 1.0D0 + B * TKL(I-1)
        KPHN(I) = ( K1N(I) + K2N(I)) / 2.D0
        KMHN(I) = ( K3N(I) + K2N(I)) / 2.D0
20      CONTINUE
C
C  CALCULATE THE TEMPERATURES AT THE IMAGINARY POINTS FOR THE
C  FIRST AND LAST K AT I+1/2 AND I-1/2 AT THE Nth TIMESTEP
C
        TOLD0N = TKL(2) - DX * G0(TAU) / K1NN
        TOLD1N = TKL(M) + DX * G1(TAU) / KMP1N
C
C  CALCULATE THE K VALUES AT THE IMAGINARY POINTS FOR THE
C  FIRST AND LAST K AT I+1/2 AND I-1/2 AT THE Nth TIMESTEP
C
        K01N = 1.0D0 + B * TOLD0N
        K0MP1N = 1.0D0 + B * TOLD1N
C
C  CALCULATE THE KVALUE FOR THE TWO ENDPOINTS I=1 AND I=MP1
C  AT  THE Nth TIMESTEP
C
        KP2N = 1.0D0 + B * TKL(2)
        KPHN(1) = ( K1NN + KP2N ) /2.D0
        KMHN(1) = ( K01N + K1NN ) / 2.D0
```

```
      KE1N = 1.0D0 + B * TKL(M)
      KE2N = 1.0D0 + B * TKL(M-1)
      KMHN(MP1) = ( KE1N + KE2N ) /2.D0
      KPHN(MP1) = ( KMP1N + KOMP1N ) / 2.0D0
C
C   CALCULATE THE K'S AT THE N+1 TIMESTEP ( KPH AND KMH )
C
      K1NN1 = 1.0D0 +  B * TKL(1)
      KMP1N1 = 1.0D0 + B * TKL(MP1)
      DO 22 I=2,M
      K1(I) = 1.0D0 + B * TKL(I+1)
      K2(I) = 1.0D0 + B * TKL(I)
      K3(I) = 1.0D0 + B * TKL(I-1)
      KPHN1(I) = ( K1(I) + K2(I)) / 2.D0
      KMHN1(I) = ( K3(I) + K2(I)) / 2.D0
22    CONTINUE
C
C  CALCULATE THE TEMPERATURES AT THE IMAGINARY POINTS FOR THE
C  FIRST AND LAST K AT I+1/2 AND I-1/2 AT THE N+1 TIMESTEP
C
      TOLD0 = TKL(2) -  DX * G0(TAU) / K1NN1
      TOLDN1 = TKL(M) + DX * G1(TAU) / KMP1N1
C
C  CALCULATE THE K VALUES AT THE IMAGINARY POINTS FOR THE
C  FIRST AND LAST K AT I+1/2 AND I-1/2 AT THE N+1 TIMESTEP
C
      K01 = 1.0D0 + B * TOLD0
      K0MP1 = 1.0D0 + B * TOLDN1
C
C  CALCULATE THE KVALUE FOR THE TWO ENDPOINTS I=1 AND I=MP1
C  AT  THE N+1 TIMESTEP
C
      KP2 = 1.0D0 + B * TKL(2)
      KPHN1(1) = ( K1NN1 + KP2 ) /2.D0
      KMHN1(1) = ( K01 + K1NN1 ) / 2.D0
      KE1 = 1.0D0 + B * TKL(M)
      KE2 = 1.0D0 + B * TKL(M-1)
      KMHN1(MP1) = ( KE1 + KE2 ) /2.D0
      KPHN1(MP1) = ( KMP1N1 + K0MP1 ) / 2.0D0
C
C  CALCULATE THE KVALUE'S ALONG THE N+1/2 TIMESTEP FOR THE
C  DIFFERENTIAL EQUATION
C
      DO 55 I=1,MP1
      KPH(I) = ( KPHN(I) + KPHN1(I) ) /2.D0
      KMH(I) = ( KMHN(I) + KMHN1(I) ) /2.D0
55    CONTINUE
C
       RETURN
       END
```

APPENDIX C

VARIABLE PROPERTIES WITH PAPER INTERFACE

```
C$ #16 BY: IGELSRUD DATE: 10/31/89 ADDED INTERFACE
C$ #15 BY: IGELSRUD DATE: 10/20/89 CHECKED CP WITH SIMPSONS RULE
C$ #14 BY: IGELSRUD DATE: 10/10/89 ADDED VARIABLE CP TO PROGRAM
C$ #13 BY: IGELSRUD DATE: 09/10/89 INCLUDED NONLINEAR THERMOCONDUCTIVITY
C$ #12 BY: IGELSRUD DATE: 02/23/89 MODIFIED TO INCLUDE CRANK NICHOLSON
C$ #11 BY: IGELSRUD DATE: 02/08/89 ADDED RELATIVE ERROR CALCULATION
C$ #10 BY: IGELSRUD DATE: 02/07/89 ADDED CALL TO TRIDAG
C$ #9 BY: IGELSRUD DATE: 02/07/89 MODIFIED FOR BACKWARD DIFFERENCEMA550
C$ #8 BY: IGELSRUD DATE: 02/05/89 MODIFIED NUMBERING OF OUTPUT VARIABLES
C$ #7 BY: IGELSRUD DATE: 02/05/89 CHANGED M CALCULATION PLUS ONE
C$ #6 BY: IGELSRUD DATE: 02/05/89 MODIFIED FORMAT STATEMENTS
C$ #5 BY: IGELSRUD DATE: 02/03/89 MODIFIED FOR FORWARD DIFFERENCE - MA550
C$ #4 BY: IGELSRUD DATE: 12/30/88 ADDED LOOP FOR MULTIPLE RUNS
C$ #3 BY: IGELSRUD DATE: 12/28/88 ALTERATION OF INPUT INFORMATION
C$ #2 BY: IGELSRUD DATE: 12/27/88 MODIFIED TO INCLUDE CRANK NICHOLSON
METHOD
C$ #1 BY: IGELSRUD DATE: 12/27/88 PROGRAM INITIALLY TAKEN FROM CARNAHAN,
ET AL
C
C
        IMPLICIT REAL*8 (A-H,O-Z)
C       DOUBLE PRECISION
        REAL LAMBDA,KMH(200),KPH(200),K1N,KMP1,K1N1,KMP1N1
        REAL KPOL,KPAP
        DIMENSION A(200), B(200), C(200), TOLD(200), D(200),
     1  TOLDL(200),TNEW(200),CPVN(200),CPVNP1(200),
     1  TEMP(200),TEMPS(200)
C
C
        OPEN(12,STATUS='NEW',FILE='OUTPUT.DAT')
        OPEN(16,STATUS='NEW',FILE='KVALUE.DAT')
        OPEN(17,STATUS='NEW',FILE='EXACT.DAT')
        OPEN(18,STATUS='NEW',FILE='TEMP.DAT')
C
C    CHECK INPUT PARAMETERS
C
10      WRITE(*,101)
        READ(*,100) LAMBDA
        WRITE(*,102)
        READ(*,111) JMAX
        WRITE(*,103)
        READ(*,111) M
        FLOATI = M
        DX = 1.D0 / FLOATI
        DTAU = LAMBDA * (DX**2)
        JDONE = 0
        WRITE(12,200)  DTAU, DX, JMAX, M, LAMBDA
        MP1 = M + 1
        TAU = 0D0
        TAUP = 0D0
        KPOL = 0.21D0
        KPAP =   0.03102
```

```
      RHOPOL =  1000.0D0
      RHOPAP =  715.0459D0
      CPPOL =   1359.15D0
      CPPAP =  320.0D0
      ALPHA1 = (KPOL)/(RHOPOL*CPPOL)
      ALPHA2 = (KPAP)/(RHOPAP*CPPAP)
      MPAP = 4 * M
      MPAPP1 = MPAP + 1
      DXPAP = DSQRT(ALPHA2 / ALPHA1) * DX
      DTAUP = DTAU * ((DXPAP ** 2)/(DX ** 2))
      CONMPT = (KPAP / KPOL) * DSQRT(ALPHA1 / ALPHA2)
C
C  SET AND PRINT INITIAL TEMPERATURES
C
      DO 30 I = 1, MP1
         FLOATI = I - 1
         TOLD(I) = F( FLOATI * DX )
         TOLDL(I) = TOLD(I)
         TNEW(I) = TOLD(I)
30    CONTINUE
C
      MP11 = MP1 + 1
      DO 32 I = MP11,MPAPP1
         FLOATI = I - 1
         TOLD(I) = F( FLOATI * DXPAP )
         TOLDL(I) = TOLD(I)
         TNEW(I) = TOLD(I)
32    CONTINUE
C
C  CALCULATE THE INITIAL VALUES FOR THE THERMOCONDUCTIVITY AT
C  I+1 AND I-1
C
      CALL KVALUE(TOLD,KPH,KMH,K1N,KMP1,MPAP,MPAPP1,JDONE,DX,TAU,
     1            DTAU,TOLDL,K1N1,KMP1N1,TAUP,DTAUP)
C
C  CALCULATE THE CPV VALUES FOR THE CONSTANTS
C
     CALL CPV(CPVN,CPVNP1,MPAPP1,JDONE,TAU,DTAU,TOLD,TOLDL,MPAP,
     1        TAUP,DTAUP,MP1)
C
C  CALCULATE THE CONSTANTS FOR THIS PROBLEM
C
      DO 20 I = 2, MPAP
         A(I) = -KMH(I)
         B(I) = ( 2.D0*CPVNP1(I) / LAMBDA) + KPH(I) + KMH(I)
         C(I) = -KPH(I)
 20      CONTINUE
C
C  CALCULATE THE CONSTANTS AT THE BOUNDARIES
C
         A(1) = 0D0
         B(1) =(2.D0 * CPVNP1(1) / LAMBDA) + KPH(1) + KMH(1)
```

```
            C(1) = -1.D0 * ( KMH(1) + KPH(1) )
            A(MP1) = -1.0D0
            B(MP1) = ((2.0D0 *(CONMPT + 1.D0))/LAMBDA) + (CONMPT + 1.D0)
            C(MP1) = -1.0D0 * CONMPT
            A(MPAPP1) = -1.D0 * (KMH(MPAPP1) + KPH(MPAPP1))
            B(MPAPP1) = (2.D0 / LAMBDA) + KPH(MPAPP1) + KMH(MPAPP1)
            C(MPAPP1) = 0D0
C
C INITIALIZE TAU TO START LOOP
C
        TAU = 0D0
        TAUP = 0D0
        WRITE (12, 210)
        WRITE (12, 220)  TAU, (TOLD(I), I=1,MP1)
        WRITE (12, 220)  TAUP, (TOLD(I), I=MP11,MPAPP1)
C
C  PERFORM CALCULATIONS OVER SUCCESSIVE TIME-STEPS
C
40      TAU = TAU + DTAU
        TAUP = TAUP + DTAUP
        JDONE = JDONE + 1
C
C
C  SET RIGHT-HAND SIDE D VECTOR FOR BOUNDARY CONDITIONS
C
221    D1C1 = (2.D0 * CPVN(1) / LAMBDA) - KPH(1) - KMH(1)
       D1C2 = KMH(1) + KPH(1)
       D1C3 =  2.D0 * KMH(1) * DX * G0(TAU) *
     1   (( 1.0D0 / K1N ) + ( 1.0D0 / K1N1 ))
       D(1) = D1C1 * TOLDL(1) + D1C2 * TOLDL(2) - D1C3
       DMP1C1 = KMH(MPAPP1) + KPH(MPAPP1)
       DMP1C2 = (2.D0 * CPVN(MPAP) / LAMBDA) - KPH(MPAPP1) - KMH(MPAPP1)
       DMP1C3 = 2.0D0 * KPH(MPAPP1) * DXPAP * G1(TAU) *
     1   (( 1.0D0 / KMP1 ) + ( 1.0D0 / KMP1N1 ))
       D(MPAPP1) = DMP1C1 * TOLDL(MPAP) + DMP1C2
     1   * TOLDL(MPAPP1) + DMP1C3
C
C
C COMPUTE RIGHT-HAND SIDE VECTOR D  FOR INSIDE GRID
C
        DO 50 I = 2, MPAP
        CST1 = KMH(I)
        CST2 = (2.D0*CPVN(I) / LAMBDA) - KPH(I) - KMH(I)
        CST3 = KPH(I)
           D(I) = CST1 * TOLDL(I-1) + CST2 * TOLDL(I) +
     1     CST3 * TOLDL(I+1)
50      CONTINUE
        CST331 = 1.0D0
        CST332 = ((2.0D0 * (CONMPT + 1.D0))/LAMBDA) - (CONMPT + 1.D0)
        CST333 = CONMPT
        D(MP1) = CST331 * TOLDL(MP1-1) + CST332 * TOLDL(MP1) +
     1  CST333 * TOLDL(MP1+1)
```

```
C
C  COMPUTE NEW VALUES OF THE FUNCTION TEMPERATURE TOLD
C
         CALL TRIDAG ( 1, MPAPP1, A, B, C, D, TOLD )
C
C   CHECK TEMPERATURES FOR CONVERGENCE
C
         DO 250 J=1,MPAPP1
         RES = DABS(TNEW(J) - TOLD(J))
         IF(RES.GT.1.0D-6) THEN
C
C   UPDATE OLD ITERATION TEMPERATURE WITH NEW ITERATION TEMPERATURE
C
         DO 750 I=1,MPAPP1
         TNEW(I) = TOLD(I)
750      CONTINUE
C
C   UPDATE THE VALUES OF K - THERMOCONDUCTIVITY
C
         CALL KVALUE(TOLD,KPH,KMH,K1N,KMP1,MPAP,MPAPP1,JDONE,DX,TAU,
     1              DTAU,TOLDL,K1N1,KMP1N1,TAUP,DTAUP)
C
C  CALCULATE THE CPV VALUES FOR THE CONSTANTS
C
        CALL CPV(CPVN,CPVNP1,MPAPP1,JDONE,TAU,DTAU,TOLD,TOLDL,MPAP,
     1          TAUP,DTAUP,MP1)
C
C   UPDATE THE CONSTANTS A, B, AND C
C
         DO 29 I = 2, MPAP
            A(I) = -KMH(I)
            B(I) = (2.0D0*CPVNP1(I)/ LAMBDA) + KPH(I) + KMH(I)
            C(I) = -KPH(I)
 29         CONTINUE
C
C  CALCULATE THE CONSTANTS AT THE BOUNDARIES
C
            A(1) = 0D0
            B(1) =(2.D0 * CPVNP1(1) / LAMBDA) + KPH(1) + KMH(1)
            C(1) = -1.D0 * ( KMH(1) + KPH(1) )
            A(MP1) = -1.0D0
            B(MP1) = ((2.0D0 *(CONMPT + 1.D0))/LAMBDA) + (CONMPT + 1.D0)
            C(MP1) = -1.0D0 * CONMPT
            A(MPAPP1) = -1.D0 * (KMH(MPAPP1) + KPH(MPAPP1))
            B(MPAPP1) = (2.D0 / LAMBDA) + KPH(MPAPP1) + KMH(MPAPP1)
            C(MPAPP1) = 0D0
              GO TO 221
              ELSE
              ENDIF
250        CONTINUE
C
C  UPDATE THE TEMPERATURES FOR THE NEXT PASS
```

```
C
            DO 222 I=1,MPAPP1
               TOLDL(I) = TOLD(I)
222         CONTINUE
C
C   PRINT TEMPERATURES WHEN APPROPRIATE
C
             IF(JDONE.EQ.45.OR.JDONE.EQ.223.OR.
     1       JDONE.EQ.445.OR.JDONE.EQ.585) THEN
             CALL CONVTT(TOLD,TEMP,MPAPP1)
             WRITE (12, 220)  TAU, (TEMP(I), I=1,MP1)
             WRITE (12, 220)  TAUP, (TEMP(I), I=MP11,MPAPP1)
             WRITE (18, 299)  TAU, (TEMP(I), I=1,MP1)
             WRITE (18, 299)  TAUP, (TEMP(I), I=MP11,MPAPP1)
             ENDIF
C
C   UPDATE THE VALUES OF K - THERMOCONDUCTIVITY
C
         CALL KVALUE(TOLD,KPH,KMH,K1N,KMP1,MPAP,MPAPP1,JDONE,DX,TAU,
     1               DTAU,TOLDL,K1N1,KMP1N1,TAUP,DTAUP)
C
C  CALCULATE THE CPV VALUES FOR THE CONSTANTS
C
        CALL CPV(CPVN,CPVNP1,MPAPP1,JDONE,TAU,DTAU,TOLD,TOLDL,MPAP,
     1            TAUP,DTAUP,MP1)
C
C   UPDATE THE CONSTANTS A, B, AND C
C
         DO 27 I = 2, MPAP
            A(I) = -KMH(I)
            B(I) = (2D0*CPVNP1(I) / LAMBDA) + KPH(I) + KMH(I)
            C(I) = -KPH(I)
 27         CONTINUE
C
C  CALCULATE THE CONSTANTS AT THE BOUNDARIES
C
            A(1) = 0D0
            B(1) =(2.D0 * CPVNP1(1) / LAMBDA) + KPH(1) + KMH(1)
            C(1) = -1.D0 * ( KMH(1) + KPH(1) )
            A(MP1) = -1.0D0
            B(MP1) = ((2.0D0 *(CONMPT + 1.D0))/LAMBDA) + (CONMPT + 1.D0)
            C(MP1) = -1.0D0 * CONMPT
            A(MPAPP1) = -1.D0 * (KMH(MP1) + KPH(MP1))
            B(MPAPP1) = (2.D0 / LAMBDA) + KPH(MP1) + KMH(MP1)
            C(MPAPP1) = 0D0
C
C CHECK TO SEE IF PROGRAM REACHED MAXIMUM  J VALUE
C
         IF( JDONE .LE. JMAX ) GO TO 40
         GO TO 10
C
C   FORMAT STATEMENTS
```

```
C
100     FORMAT(G12.6)
111     FORMAT(I5)
101     FORMAT(1X,'ENTER THE VALUE FOR LAMBDA - R')
102     FORMAT(1X,'ENTER THE VALUE FOR JMAX - TIME GRID MAX')
103     FORMAT(1X,'ENTER THE VALUE FOR M - # INTERVALS')
200     FORMAT(1X,/
     1  'CRANK-NICHOLSON METHOD, WITH PARAMETERS'/
     2  'DTAU = ',G13.7/'DX = ',G13.7/'JMAX = ',I4/'M = ',I4/
     3  'LAMBDA = ',G13.7)
210     FORMAT(1X,'TIME',18X,'VALUES OF U AT THE GRIDPOINTS')
220     FORMAT(1X, F7.3/1X, 7X, 5(G13.7,2X)/8X,5(G13.7,2X)/8X,G13.7)
227     FORMAT(1X, F7.3/6(8X,5(G13.7,2X)/)8X,G13.7)
229     FORMAT(1X, F7.3/4(8X,5(G13.7,2X)/)8X,G13.7)
223     FORMAT(1X, F7.3/8(8X,5(G13.7,2X)/)8X,G13.7)
231     FORMAT(1X, F7.3/10(8X,5(G13.7,2X)/)8X,G13.7)
299     FORMAT(1X,G13.7/1X,G13.7)
        END
        REAL*8 FUNCTION F(DIST)
        IMPLICIT REAL*8 (A-H,O-Z)
        F = 0D0
        RETURN
        END
        REAL*8 FUNCTION G0(TIME)
        IMPLICIT REAL*8 (A-H,O-Z)
        G0 = -1.0D0
        RETURN
        END
        REAL*8 FUNCTION G1(TIME)
        IMPLICIT REAL*8 (A-H,O-Z)
        G1 = 0D0
        RETURN
        END
```

```
C$ BY: IGELSRUD DATE: 10/07/89 FIRST ABSORB
      SUBROUTINE CPV(CPVN,CPVNP1,MPAPP1,JDONE,TAU,DTAU,
     1               TOLD,TOLDL,MPAP,TAUP,DTAUP,MP1)
      IMPLICIT REAL*8 (A-H,O-Z)
     DATA TEMPA /0.0,76.0,80.0,85.0,90.0,95.0,100.0,105.0,
     2 110.0,115.0,120.0,125.0,130.0,135.0,140.0,145.0,150.0,
     3 155.0,160.0,162.5,165.0,170.0,172.5,175.0,180.0,185.0,
     4 190.0,190.0/
     DATA TEMPB /0.0,82.0,90.0,95.0,100.,105.,110.,115.,120.0,
     1 125.0,127.0,130.0,135.0,140.0,145.0,148.0,150.0,152.0,155.0,
     2 157.5,160.,162.5,165.0,170.0,172.0,175.0,177.5,180.0,180.0/
     DATA CPVA  /0.325,0.325,0.343,0.360,0.365,0.370,0.375,
     1 0.378,0.382,0.388,0.395,0.405,0.415,0.429,0.450,0.455,
     2 0.500,0.700,0.800,0.975,0.770,0.425,0.375,0.376,0.379,
     3 0.382,0.382,0.382/
     DATA CPVB  /0.34,0.34,0.350,0.3675,0.375,0.385,0.395,0.410,
     1 0.420,0.430,0.435,0.440,0.459,0.475,0.4875,0.493,0.50,0.550,
     2 0.70,0.80,0.90,1.0,1.21,0.975,0.800,0.590,0.500,0.400,0.400/
C
C     CALCULATE THE KVALUES AT THE HALF POINTS AND AT THE END POINTS
C
      DIMENSION  TOLD(MP1),TK(200),TOLDL(MP1),TKL(200)
      DIMENSION  CPVN(MPAPP1),CPVNP1(MPAPP1),TEMPA(28),TEMPB(29)
      DIMENSION  CPVA(28),CPVB(29)
      CPV0 = 5.49278D-1
C
C
C
C  CALCULATE TEMPERATURES AT THE Nth TIMESTEP
C
      DO 82 I=1,MP1
      TKL(I) = 285.71D0 * TOLDL(I)
82    CONTINUE
C
C   CALCULATE THE CPV'S AT THE Nth TIMESTEP
C
      DO 20 I=1,MP1
         DO 23 II=2,28
         IF(TKL(I).LE.TEMPB(2)) CPVN(I) = CPVB(1)/CPV0
         IF(TKL(I).GT.TEMPB(II).AND.TKL(I).LT.TEMPB(II+1))
     1   CPVN(I) = (CPVB(II) + CPVB(II+1)) / (2.0D0 * CPV0)
         IF(TKL(I).GE.TEMPB(29)) CPVN(I) = CPVB(29)/CPV0
23     CONTINUE
20    CONTINUE
C
C   CALCULATE THE CPV'S AT THE N+1 TIMESTEP ( KPH AND KMH )
C
      DO 22 I=1,MP1
         DO 24 II=2,28
         IF(TKL(I).LE.TEMPB(2)) CPVNP1(I) = CPVB(1)/CPV0
         IF(TKL(I).GT.TEMPB(II).AND.TKL(I).LT.TEMPB(II+1))
     1   CPVNP1(I) = (CPVB(II) + CPVB(II+1)) / (2.0D0 * CPV0)
```

```
          IF(TKL(I).GE.TEMPB(29)) CPVNP1(I) = CPVB(29)/CPV0
 24      CONTINUE
 22     CONTINUE
        MPP1 = MP1 + 1
        DO 555 I=MPP1,MPAPP1
        CPVN(I) = 1.0D0
        CPVNP1(I) = 1.0D0
 555    CONTINUE
         RETURN
         END
```

```
C$ BY: IGELSRUD DATE: 09/23/89 MODIFIED THE K EQUATION
C$ BY: IGELSRUD DATE: 09/11/89 FIRST ABSORB
      SUBROUTINE KVALUE (TOLD,KPH,KMH,K1NN,KMP1N,MPAP,MPAPP1,JDONE,
     1  DX,TAU,DTAU,TOLDL,K1NN1,KMP1N1,TAUP,DTAUP)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     CALCULATE THE KVALUES AT THE HALF POINTS AND AT THE END POINTS
C
      DIMENSION  TOLD(MPAPP1),TK(200),TOLDL(MPAPP1),TKL(200)
      REAL K1(200),K2(200),K3(200),KPHN1(200),KMHN1(200)
      REAL KP2,KE1,KE2,K01,K0MP1,K1NN1,KMP1N1
      REAL KP2N,KE1N,KE2N,K01N,K0MP1N,K1NN,KMP1N
      REAL KPHN(200),KMHN(200),K1N(200),K2N(200),K3N(200)
      REAL KPH(MPAPP1),KMH(MPAPP1)
      B = -2.65957D-3
C
C  CALCULATE TEMPERATURES AT THE Nth TIMESTEP
C
      DO 82 I=1,MPAPP1
      TKL(I) = 285.71D0 * TOLDL(I)
82    CONTINUE
C
C   CALCULATE THE K'S AT THE Nth TIMESTEP ( KPH AND KMH )
C
      MREAL = MPAP / 4
      MP1 = MREAL + 1
      K1NN  = 1.0D0 +  B * TKL(1)
      KMP1N  = 1.0D0 + B * TKL(MP1)
      DO 20 I=2,MPAP
      K1N(I) = 1.0D0 + B * TKL(I+1)
      K2N(I) = 1.0D0 + B * TKL(I)
      K3N(I) = 1.0D0 + B * TKL(I-1)
      KPHN(I) = ( K1N(I) + K2N(I)) / 2.D0
      KMHN(I) = ( K3N(I) + K2N(I)) / 2.D0
20    CONTINUE
C
C  CALCULATE THE TEMPERATURES AT THE IMAGINARY POINTS FOR THE
C  FIRST AND LAST K AT I+1/2 AND I-1/2 AT THE Nth TIMESTEP
C
      TOLD0N = TKL(2) - DX * G0(TAU) / K1NN
      TOLD1N = TKL(MREAL) + DX * G1(TAU) / KMP1N
C
C  CALCULATE THE K VALUES AT THE IMAGINARY POINTS FOR THE
C  FIRST AND LAST K AT I+1/2 AND I-1/2 AT THE Nth TIMESTEP
C
      K01N = 1.0D0 + B * TOLD0N
      K0MP1N = 1.0D0 + B * TOLD1N
C
C  CALCULATE THE KVALUE FOR THE TWO ENDPOINTS I=1 AND I=MP1
C  AT  THE Nth TIMESTEP
C
      KP2N = 1.0D0 + B * TKL(2)
```

```
      KPHN(1) = ( K1NN + KP2N ) /2.D0
      KMHN(1) = ( KO1N + K1NN ) / 2.D0
      KE1N = 1.0D0 + B * TKL(MREAL)
      KE2N = 1.0D0 + B * TKL(MREAL-1)
      KMHN(MP1) = ( KE1N + KE2N ) /2.D0
      KPHN(MP1) = ( KMP1N + KOMP1N ) / 2.0D0
C
C   CALCULATE THE K'S AT THE N+1 TIMESTEP ( KPH AND KMH )
C
      K1NN1 = 1.0D0 +  B * TKL(1)
      KMP1N1 = 1.0D0 + B * TKL(MP1)
      DO 22 I=2,MREAL
      K1(I) = 1.0D0 + B * TKL(I+1)
      K2(I) = 1.0D0 + B * TKL(I)
      K3(I) = 1.0D0 + B * TKL(I-1)
      KPHN1(I) = ( K1(I) + K2(I)) / 2.D0
      KMHN1(I) = ( K3(I) + K2(I)) / 2.D0
22    CONTINUE
C
C  CALCULATE THE TEMPERATURES AT THE IMAGINARY POINTS FOR THE
C  FIRST AND LAST K AT I+1/2 AND I-1/2 AT THE N+1 TIMESTEP
C
      TOLDO = TKL(2) -  DX * G0(TAU) / K1NN1
      TOLDN1 = TKL(M) + DX * G1(TAU) / KMP1N1
C
C  CALCULATE THE K VALUES AT THE IMAGINARY POINTS FOR THE
C  FIRST AND LAST K AT I+1/2 AND I-1/2 AT THE N+1 TIMESTEP
C
      KO1 = 1.0D0 + B * TOLDO
      KOMP1 = 1.0D0 + B * TOLDN1
C
C  CALCULATE THE KVALUE FOR THE TWO ENDPOINTS I=1 AND I=MP1
C  AT  THE N+1 TIMESTEP
C
      KP2 = 1.0D0 + B * TKL(2)
      KPHN1(1) = ( K1NN1 + KP2 ) /2.D0
      KMHN1(1) = ( KO1 + K1NN1 ) / 2.D0
      KE1 = 1.0D0 + B * TKL(MREAL)
      KE2 = 1.0D0 + B * TKL(MREAL-1)
      KMHN1(MP1) = ( KE1 + KE2 ) /2.D0
      KPHN1(MP1) = ( KMP1N1 + KOMP1 ) / 2.0D0
C
C  CALCULATE THE KVALUE'S ALONG THE N+1/2 TIMESTEP FOR THE
C  DIFFERENTIAL EQUATION
C
      DO 55 I=1,MREAL
      KPH(I) = ( KPHN(I) + KPHN1(I) ) /2.D0
      KMH(I) = ( KMHN(I) + KMHN1(I) ) /2.D0
55    CONTINUE
C
C
C  CALCULATE THE KVALUE'S FOR THE PAPER
```

```
C
      MPP1 = MPAP + 1
      DO 58 I=MPP1,MPAPP1
      KPH(I) = 1.0D0
      KMH(I) = 1.0D0
58    CONTINUE
C
       RETURN
       END
```

```
C$ BY: IGELSRUD DATE: 09/11/89 FIRST ABSORB
        SUBROUTINE CONVTT(TOLD,TEMP,MPAPP1)
        IMPLICIT REAL*8 (A-H,O-Z)
C
C     INITIALIZE THE VARIABLES IN THIS ROUTINE
C
        DIMENSION  TOLD(MPAPP1),TEMP(MPAPP1)
C
C  CONVERT THE TEMPERATURES TO CENTIGRATE
C
        DO 80 I=1,MPAPP1
        TEMP(I) = 285.71D0 * TOLD(I)
80      CONTINUE
         RETURN
         END
```