# ERRATA

iv Appendices . . . . . . . . . . . . . . . . . . . . 133

viii 6.6 Schematics of the experimental furnace. . . 130

ix A. Listing of the Program . . . . . . . . . . . . 133

9 Add: Note: In a nonisotropic medium equation (2.1) holds only if the axes coincide with the axes of the conductivity ellipsoid.

32 (3.4.10) Eliminate $\cos^2\gamma$ from the last parameter.

52 (3.8.5) $0(\Delta x)^3$ instead of $0(\Delta x)^2$.

54 (3.8.12) $0(\Delta x)^3$ instead of $0(\Delta x)^4$.

109 Title of Table 5.1: Values used in the computation.

110 Remove word "Data" from title of Table 5.2.

113 Remove word "Data" from title of Table 5.3.

115 Remove word "Data" from title of Table 5.4.

116 Remove word "Data" from title of Table 5.5.

118 Remove word "Data" from title of Table 5.6.

120 Remove word "Data" from title of Table 5.7.

128 Figure 6.4 Note: The solid material is left of the interface.

Figure 6.5. Replace $\Delta$ by $-\Delta$. Note: The solid material is right of the interface.

# HEAT TRANSFER IN SOLIDIFYING BARS

by

Andrew Szántó Teller

## ABSTRACT

A method for the solution of a two-dimensional Stefan problem is developed. The computations based on this method can predict not only the temperature distribution and the location of the solid-liquid interface but also the shape of the interface. In addition, a versatile horizontal tube furnace has been constructed, which is capable of producing decanted solid-liquid interfaces during normal freezing and zone refining experiments. This feature allows the preservation of these interfaces for later examination.

The problem considered consists of a rectangle containing two different phases of the same material in two regions, corresponding to the longitudinal center plane of a bar in a normal freezing experiment. The temperature distribution around the rectangle is an arbitrary function of time and location. The problem is formulated in terms of two simultaneous equations, namely, (a) a differential equation describing the motion of the interface, and (b) a partial differential equation defining the temperature distribution in each of the regions. Several finite difference procedures are considered for the solution of these equations. A simple explicit procedure is used to solve the difference form of equation (a) while the Alternating Direction Implicit Method is used to solve equation (b). In order to minimize the amount of computations needed to test the method, several simplifying assumptions

are made. The most significant of these is the assumption of constant density throughout the rectangle, which in turn implies no movement of material within the regions.

An IBM 7090 digital computer was used to carry out the computations. A problem containing 45 x 5 grid points, required 280 seconds for 600 iterations when the temperature distribution was printed in every 20 iterations, and it required 480 seconds for 1200 iterations when the temperature distribution was printed in every 100 iterations.

The finite difference computations were carried out for several possible material constants and parameters. The effects of the heat transfer coefficient, velocity, latent heat and thermal diffusivity on the numerical solution were investigated. With the temperature distribution used for these computations the interface was slightly convex under equilibrium conditions. The calculations showed that the convexity of the interface should decrease, become plane, or even become concave as the bar starts to move in the furnace. The magnitude of the change seemed to depend primarily on the velocity of the bar and on the latent heat of the material. The computed results show at least qualitative agreement with observed results. The convergence of the numerical method is also shown in several sample computations.

The experiments were carried out using indium antimonide. The material was encapsulated in a 22 cm. long, 2 cm. x 2 cm. square quartz tube. The material filled about 60% of the tube when it was all liquid. In the first experiment 40% of the bar was normally frozen in the furnace. The velocity of the bar during the process was 0.73 inches/hr. In the second experiment the bar was placed in the furnace and allowed to come to equilibrium; the position of the bar was such that approximately 40%

of the bar was frozen while the rest melted in the warmer part of the furnace. Photographs of the interface show that the experimental furnace accomplished its objective. Although the change in the interface curvature agrees with the calculated change, direct quantitative comparison of the experimental results is not possible due to (a) the assumptions made in simplifying the problem and (b) the awkward geometry used for the experiment. Several suggestions are made to improve the accuracy, generality and applicability of the computations and to modify the experimental procedure used to verify the computed results.

# HEAT TRANSFER IN SOLIDIFYING BARS

by

Andrew Szántó Teller

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in the
University of Michigan
1966

Doctoral Committee:

Associate Professor James O. Wilkes, Chairman
Professor Wilbur C. Bigelow
Professor Stuart W. Churchill
Associate Professor Robert H. Kadlec
Doctor Donald R. Mason, Radiation, Inc.
Professor Ernst Katz

# ACKNOWLEDGMENTS

The author gratefully acknowledges the assistance of his Doctoral Thesis Committee. I am deeply indebted to Professor James O. Wilkes, Chairman of the Committee, for his encouragement and active assistance, without whose help this thesis would not have materialized. I wish to thank especially Professor Stuart W. Chruchill and Dr. Donald R. Mason, who helped formulate the problem (they were the former Co-Chairmen) and Professor Gabriel Weinreich who also served as a member of the Committee.

I am further indebted to the National Science Foundation, the Institute of Science and Technology and the various other units of the University for the graduate fellowships they provided. Acknowledgment is also made of the financial aid afforded by the National Science Foundation through its grant No. GK-100.

I would like to thank Mr. S. J. Lewis for his aid in constructing the equipment and the many persons who helped in the preparation of the manuscript.

# TABLE OF CONTENTS

Page

CHAPTER

Table of Contents (continued)

Chapter Page

## LIST OF TABLES

# LIST OF FIGURES

List of Figures (continued)

List of Figures (continued)

# LIST OF APPENDICES

Appendix Page

# ABSTRACT

A method for the solution of a two-dimensional Stefan problem is developed. The computations based on this method are sufficiently accurate to predict not only the temperature distribution and the location of the solid-liquid interface but also the shape of the interface. In addition, a versatile horizontal tube furnace has been constructed, which is capable of producing decanted solid-liquid interfaces during normal freezing and zone refining experiments. This feature allows the preservation of these interfaces for later examination.

The problem considered consists of a rectangle containing two different phases of the same material in two regions, corresponding to the longitudinal center plane of a bar in a normal freezing experiment. The temperature distribution around the rectangle is an arbitrary function of time and location. The problem is formulated in terms of two simultaneous equations, namely, (a) a differential equation describing the motion of the interface, and (b) a partial differential equation defining the temperature distribution in each of the regions. Several finite difference procedures are considered for the solution of these equations. A simple explicit procedure is used to solve the difference form of equation (a) while the Alternating Direction Implicit Method is used to solve equation (b). In order to minimize the amount of computations needed to test the method, several simplifying assumptions are made. The most significant of these is the assumption of constant density throughout the rectangle, which in turn implies no movement of material within the regions.

An IBM 7090 digital computer was used to carry out the computations. A problem containing 45 x 5 grid points, required 280 seconds for 600 iterations when the temperature distribution was printed in every 20 iterations, and it required 480 seconds for 1200 iterations when the temperature distribution was printed in every 100 iterations.

The finite difference solutions are computed for several possible operating conditions and for several possible material constants. The computed results show at least qualitative agreement with observed results. The convergence of the numerical method is shown in sample computations.

The previously-mentioned experimental furnace is described and the operating procedures used in two experiments are given. Photographs of the interfaces obtained in the experiments show that the experimental furnace accomplishes its objectives. Direct quantitative comparison of the experimental and calculated results is not possible due to (a) the assumptions made in simplifying the problem and (b) the awkward geometry used for the experiment. Several suggestions are made to improve the accuracy, generality, and applicability of the computations and to modify the experimental procedure used to verify the computed results.

# CHAPTER I

## INTRODUCTION

The object of this investigation is to develop a method for computing the position and shape of the solid-liquid interface in a solidifying bar. Thus the Stefan problem is to be solved in two or more dimensions. One has to keep in mind that the accurate location of the interface is the main goal of interest rather than the temperature distribution or the time required to freeze a portion of the bar. Although these temperatures and times will be available as part of the computation, the above consideration places limitations on the methods which can be used to solve the Stefan problem. The amount of computation required for a three-dimensional model would have been more than this investigation would have allowed. Work was, therefore, carried out on a two-dimensional model. The present computations and some of the derivations are for a specific problem which was simplified for sake of expediency. However, the method used for its solution contains many general ideas which are extendable to more general, more useful, and even more complex problems in two and three dimensions. The need for this investigation can be seen after a brief review of the developments which preceded it.

The heat transfer problem associated with solidifying liquids was not studied until 1891, when Stefan (89) wrote an article on the formation of ice. Since this seems to be the first article published on this subject, freezing problems are frequently called "Stefan" problems. Carslaw and Jaeger (18) point out that some of the mathematical results applicable to these problems were known even earlier. They also give a brief review of the early developments in the field of heat conduction with change of state.

In recent times several complicated engineering problems have been closely related to the freezing problem. These include the ablation cooling of space vehicles, the underground storage of liquified natural gas and the directional solidification of alloys. These problems were not solvable with the available methods, thus giving rise to the current interest in alternative techniques. Examples of the latter include: transformation of the problem to one which is solvable, at least approximately, by analytic methods or the use of numerical methods to solve the problem either in its original form or after transformation. These numerical methods became more popular with the development and general availability of high speed digital computers. Nevertheless, progress was slow.

The one-dimensional Stefan problem has been discussed extensively in recent years by Altman (2), Boley (7,8,9), Churchill and Seider (19), Churchill and Teller (20), Citron (21), Crank (25), Dewey (26), Donald (27), Douglas (30,31,33), Douglas and Gallie (36), Douglas and Jones (39), Ehrlich (44), Evans (45), Friedman (46,47,48), Goodman (50), Hamill and Bankoff (51), Kolodner (58), Kyner (60,61), Landau (62), Lax (65), Longwell (68), Lotkin (69), Luke (70), McMordie (71), Miranker (73,74), Murray and Landis (76), Poots (81), Price (83), Sunderland and Grosh (91), Trench (95), and by many other investigators. These articles indicate the amount of work that has been done to develop new one-dimensional methods, establish their applicability and accuracy, and indicate their advantages and disadvantages. The information is still not well organized, however, and it is scattered throughout the literature.

Many problems cannot be solved by one-dimensional methods directly even with the use of transformations. In certain others the approximation achieved by these simple methods is inadequate. Until recently these problems were considered unsolvable because of the large amount of computation required in even the simplest multi-dimensional problem. As the size and speed of computers increased, and numerical methods utilizing them became available for multi-dimensional problems, some two-dimensional problems were solved by Allen (1), Cook, Mason, and Smith (23), Hashemi (53), Wilcox and Duty (99), Wilkes (100), and other investigators. These works, along with the present investigation, show that two-dimensional problems are solvable on existing large computers and that smaller three-dimensional problems may also lie within their capability. But today's large and complex multi-dimensional problems, unsolvable on our present computers, will be solved by improved methods using larger, faster computers which are being developed.

The discovery of transistors and other semiconducting devices gave rise to new interest in the preparation of high purity materials. Among the processes used for this purpose are the various zone refining, zone leveling and repeated partial normal freezing techniques. These same processes, along with crystal pulling from the melt, can be used to prepare single crystals. Recently, directional solidification was used to cast alloys with special properties. It would be desirable in all these processes to be able to solve the heat, momentum, and material balance equations simultaneously. Although this is a large task which remains to be done in the future, the present study can be considered to solve some of the questions arising in these problems. Previously, Cook, Mason, and Smith (23) calculated the temperature distribution in a cylindrical sample. They neglected the heat of fusion, reducing the problem to one of simple conduction. The error in such a procedure was pointed out by Lightfoot (67) in 1930, but when a good solution is not available, a crude approximation is better than none at all. Wilcox and Duty (99) presented a more sophisticated procedure by assuming a pseudo steady state and using a trial and error procedure to compute the interface shape in a cylindrical bar. Hashemi (53) proposed a general method to obtain the temperature distribution in a two-dimensional Stefan problem, in which he assumed that freezing occurred over a temperature range. This assumption was excellent for certain materials, like wet soil, and his computations for these kinds of materials were good. But for pure materials with a sharp melting point the computation is of questionable value as it is not clear how the solution would behave for smaller and smaller melting temperature ranges. The method presented in this work is general enough to be usable for both kinds of materials.

The organization of this dissertation is as follows: First, in Chapter II, the mathematical models available for presenting the Stefan problem are discussed and some of the ideas available to solve them are reviewed. In Chapter III, the problem is defined in its specific form and the equations used to solve it are derived. The computer program used to solve the problem is described in Chapter IV, followed by a discussion of the computed results in Chapter V. Next, in Chapter VI, the design of a furnace capable of producing decanted interfaces in a freezing, horizontal, rectangular bar is presented along with the results obtained during the operation of such furnace. It will be pointed out that a more complex mathematical model is required to represent this experiment and predict the location and shape of the interface correctly. Finally, in Chapter VII, an improved experimental technique for the verification of this and similar computations and the solution of mathematical models representing more significant physical problems are suggested for future investigation.

# CHAPTER II

## DISCUSSION OF PREVIOUS WORK

In order to solve a multi-dimensional freezing problem using a numerical technique, one has to be familiar with a large number of subjects related to heat transfer and phase changes in general, their mathematical representation, the various numerical methods and their properties as they apply to these problems, the programming of computers, and so on. No attempt will be made to review comprehensively the work done in all or any of these fields; only those works and ideas will be mentioned which have some bearing on the following developments.

In a review of this kind it is difficult to decide how far one should go in giving credit to the original or first investigators. For example, in treating the curved boundary, an interpolation of degree two is needed. Even though a good method was proposed by Shortly and Weller (88) in 1938 and Mikeladze (72) in 1941, many modern papers either derive their results using Taylor series and do not give any reference, or refer to a general text, such as Forsythe and Wasow (49), which discusses these articles while treating curved boundaries in general. Similarly, the various Alternating Direction Implicit (ADI hereafter) methods are treated with increasing frequency without referring to the articles of Peaceman and Rachford (77) and Douglas and

Rachford (40). This is due to two factors: first, the information is becoming widely known and used, being often presented in lectures and discussions without mentioning the original source, and this carries over to the written works; secondly, some ideas are first presented in a form which has narrow applicability, and considerable work is done by many investigators before it reaches its final, most useful form, making it practically impossible to give credit to all those investigators who were responsible for its development. In this work no attempt will be made to reference all the ideas presented even though some of the important fundamental articles are included in the bibliography.

Let us consider a body in which the temperature distribution is such that in various parts of the body different phases are present. This body occupies region R in space and bonded by surface S. Within this region there exist a number of homogenous subregions, $R_i$, each

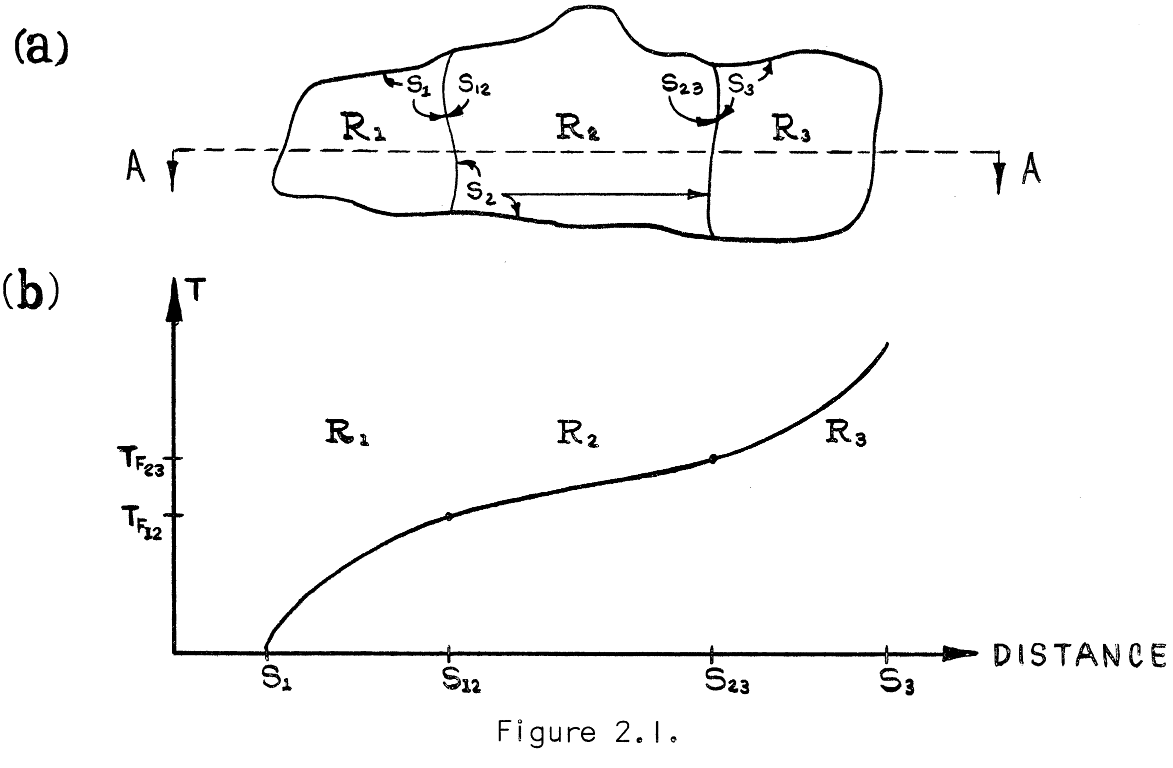

Figure 2.1.

(a) A two-dimensional body containing three subregions.

(b) Graph of a possible temperature distribution along the cut A-A.

containing one phase of the material, and each is bounded by the surface $S_i$. Those portions of the subregion surfaces which belong to two different subregions, $S_i$ and $S_j$ where $i \neq j$, we shall call interfaces and designate as $S_{ij}$, as illustrated in Figure 2.1. It would be desirable to know the temperature distribution in this body and the location of the interfaces as a function of time.

The temperature distribution in each subregion satisfies the heat equation as long as there is no movement of material. In three-dimensional rectangular coordinates, we have:

$$\frac{\partial}{\partial x_1} k_{i1} \frac{\partial T_i}{\partial x_1} + \frac{\partial}{\partial x_2} k_{i2} \frac{\partial T_i}{\partial x_2} + \frac{\partial}{\partial x_3} k_{i3} \frac{\partial T_i}{\partial x_3} = \rho_i c_i \frac{\partial T_i}{\partial t} \tag{2.1}$$

$$\text{when } \bar{x} \in R_i \text{ and } t > 0$$

subject to the boundary conditions:

$$\alpha_i T_i + \beta_i \frac{\partial T_i}{\partial n} = \gamma_i \quad \text{when } \bar{x} \in S_i \text{ and } t > 0 \tag{2.2}$$

and to the initial condition:

$$T_i = \phi_i, \qquad \text{when } \bar{x} \in R_i \text{ and } t = 0, \tag{2.3}$$

where

$\bar{x} = (x_1, x_2, x_3)$, the location of a point in space,

$t$ = time,

$T_i$ = temperature in the subregion $R_i$,

$k_{i1}, k_{i2}, k_{i3}$ = thermal conductivity. In an isotropic medium $k_{i1}=k_{i2}=k_{i3}=k_i$. In the following discussion an isotropic medium will be considered, and it will be assumed that k is a function only of the temperature, position and the phase present; $k_i=k(\bar{x},T_i)$.

$\rho_i$ = density. Although the density is a function of temperature and position in general, one has to assume that it is a function of position only if no material movement is to occur. Crank (25) nevertheless solves a one-dimensional problem containing two phases of different densities. Some simple temperature dependence of similar nature can be accommodated in numerical solutions also, but tend to complicate the computations. We will return to ways to handle this question in the section on recommendations for future investigations. Until then, it will be assumed that $\rho_i = \rho(\bar{x})$.

$c_i$ = $(\frac{\partial H}{\partial T})$, is the heat capacity, where H is the enthalpy of the material. It is in general a function of both the location and the temperature; $c_i=c(\bar{x},T_i)$.

$\phi_i$ = the initial temperature distribution in the subregion $R_i$.

The coefficients $\alpha_i$, $\beta_i$ and $\gamma_i$ arise from the boundary conditions and they may depend in general on both the temperature and position. On an interface $S_{ij}$ equation (2.2) becomes:

$$T_i = T_j = T_{F_{ij}} \tag{2.4}$$

and

$$k_i \frac{\partial T_i}{\partial n_{ij}} - k_j \frac{\partial T_j}{\partial n_{ij}} = L \rho \frac{dN_s}{dt} \quad , \tag{2.5}$$

where

$n_{ij}$ is the direction normal to the interface,

$L$ is the latent heat of fusion,

$T_{F_{ij}}$ is the transformation temperature,

$\frac{dN_s}{dt}$ is the velocity of the moving interface.

Equation (2.4) merely states that the boundary separating the two subregions is at the transformation temperature; if the phase in subregion $R_i$ is stable above the transformation temperature, then $T_i \geqq T_{F_{ij}} \geqq T_j$ holds in the two subregions. Equation (2.5) is an energy balance around the interface stating that the net heat input by conduction equals the heat absorbed during the change of phase as latent heat.

The equations (2.1 - 2.5) form a nonlinear system of partial differential equations as is pointed out by Landau (62) and by Carslaw and Jaeger (18). The nonlinearity arises mainly from the nonlinear boundary condition, equation (2.5). Nonetheless, the equations can be solved analytically in special cases. Most of these involve problems which can be reduced to one dimension. Carslaw and Jaeger (18) is a standard reference book on the existing one-dimensional problems, while a person interested more in theory would turn to Friedman (48). But not all one-dimensional problems can be solved analytically, and nothing shows this better than the large number of methods proposed for their solution.

Many authors use the following model to demonstrate their methods:

$$T_{xx} = T_t \quad \text{when} \quad 0 < x < S(t) \quad \text{and} \quad t > 0, \tag{2.6}$$

$$T = 0 \quad \text{when} \quad x \geqq S(t), \tag{2.7}$$

$$T_x = -1 \quad \text{when} \quad x = 0 \quad \text{and} \quad t > 0, \tag{2.8}$$

$$\frac{dS(t)}{dt} = -T_x \quad \text{when} \quad x = S(t) \quad \text{and} \quad t > 0, \tag{2.9}$$

$$S(0) = 0 \quad \text{when} \quad t = 0. \tag{2.10}$$

Equation (2.6) is the heat equation for the region under consideration, and on one boundary a constant heat flux is imposed on this region, while the other boundary is a moving interface. These are illustrated in Figure 2.2. The problem is reduced to considering a single region, however, by assuming that the temperature distribution in the other region is already at the transformation temperature. This may be a

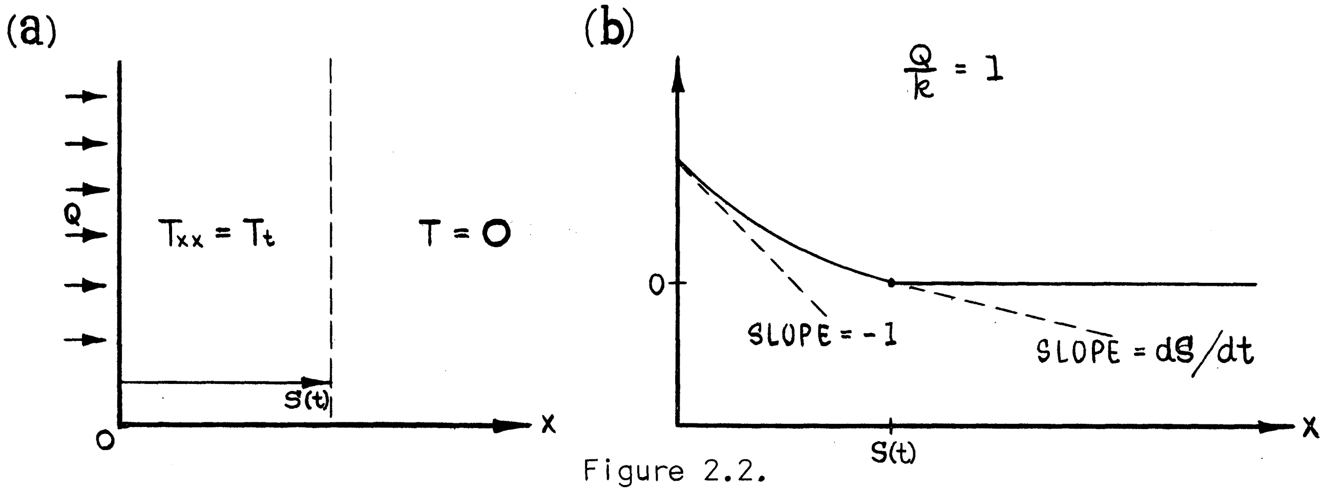

Figure 2.2.

(a) The semi-infinite region and

(b) temperature distribution defined by equations (2.6) - (2.10).

significant simplification because equation (2.9) contains only one derivative and not the difference of two derivatives as in equation (2.5). Hence, equation (2.9) can be approximated more accurately.

A second problem, illustrated in Figure 2.3 can be stated as follows:

(A) During the time period $0 < t < t^*$

$$\alpha \ T_{xx} = T_t \qquad \text{for} \quad 0 < x < w, \qquad (2.11)$$

subject to the boundary conditions:

$$k \ T_x = -Q(t) \qquad \text{at } x = 0 \quad \text{and} \qquad (2.12)$$

$$T_x = 0 \qquad \text{at } x = w, \qquad (2.13)$$

and to the initial condition

$$T = 0 \qquad \text{at } t = 0, \text{ for } 0 \leqq x \leqq w. \qquad (2.14)$$

(B) For times $t > t^*$

$$\alpha T_{xx} = T_t \qquad \text{for } s(t) < x < w, \qquad (2.15)$$

subject to the boundary conditions

$$k\, T_x = -Q(t) + \rho L \frac{ds}{dt} \qquad \text{at } x = s(t), \qquad (2.16)$$

$$T = T^* \qquad \text{at } x = s(t), \qquad (2.17)$$

$$T_x = 0 \qquad \text{at } x = w. \qquad (2.18)$$

The time $t^*$ is the time when the surface $x = 0$ reaches the transformation temperature $T^*$, hence it is also true that $s(t^*) = 0$. In the above equations L stands for the latent heat of fusion, $Q(t)$ is an arbitrary heat flux and the other, previously not defined variables are given in Figure 2.3. This problem is very important in predicting the effectiveness of an ablative cooling shield. Figure 2.3a and equations (2.11) - (2.14) represent the initial heating of a material which originally was at a uniform temperature. When the surface reaches the transformation temperature of the material some of the heat is absorbed as latent heat by the material which melts or sublimates or decomposes,

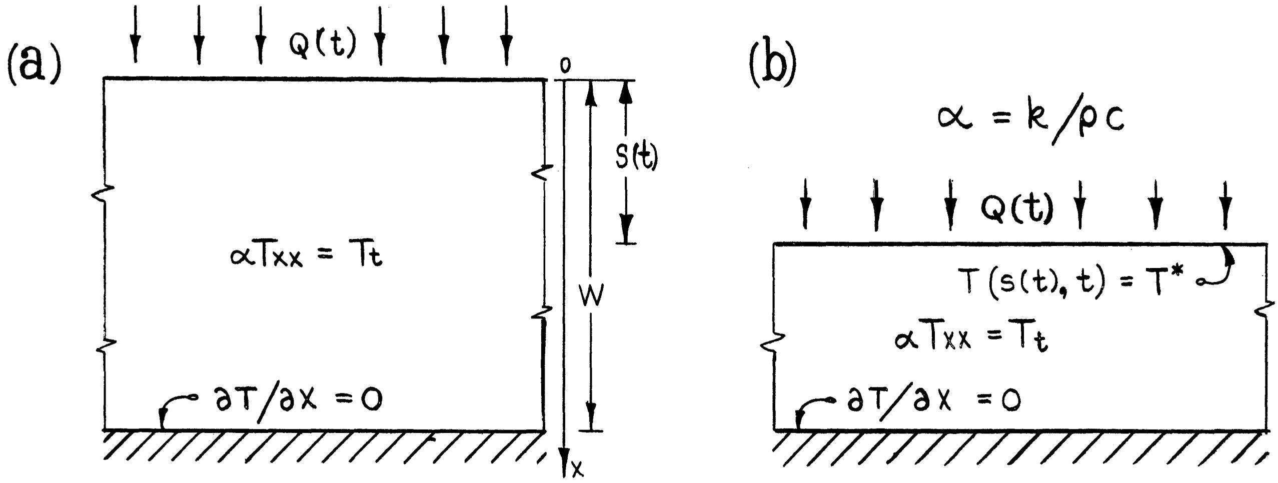

Figure 2.3.

(a) The plate before time $t^*$.

(b) The plate after time $t^*$.

Time = $t^*$ when the surface at x=0 reaches the melting temperature $T^*$.

but is immediately removed by some mechanism. The thickness of the material is reduced at a rate determined by equation (2.16), but the temperature at the surface remains constant. This problem was discussed by Evans, Isaacson and MacDonald (45), Landau (62), Citron (21), Dewey, Schlesinger and Sadhkin (26), Lotkin (69), Boley (79), Altman (2), Sunderland and Grosh (91) among others. Most of these solutions are designed for this particular problem and are not extendable, although the transformation suggested by Landau (62) is used for various solutions of all the one-dimensional moving boundary problems.

The two previously mentioned problems consider the temperature distribution in one region only. The more general treatment using two regions is usually stated in the following form:

$$\alpha T_{1_{xx}} = T_{1_t} \qquad \text{in } R_1, \tag{2.19}$$

$$\alpha T_{2_{xx}} = T_{2_t} \qquad \text{in } R_2, \tag{2.20}$$

$$\rho L \, \frac{ds_{12}}{dt} = k_1 \frac{\partial T_i}{\partial x} - k_2 \frac{\partial T_2}{\partial x} \qquad \text{at } x = S_{12}, \tag{2.21}$$

$$T_1 = T_2 \qquad \text{at } x = S_{12}, \tag{2.22}$$

subject to an initial temperature distribution as in equation (2.3) and to outside boundary conditions as in equation (2.2) usually at $x = 0$ and at $x = w$, but the latter is sometimes replaced by

$$T_2 = A \qquad \text{as } x \to \infty \tag{2.23}$$

The methods developed for solving the problem defined by equations (2.19) - (2.23) are usually adaptable to the problem defined in equations (2.6) - (2.10).

Before going into the numerical solutions available to solve these equations as they stand, let us review two ways to eliminate interfacial boundary conditions and thereby simplify the solution. The interfacial boundary condition (equations (2.5), (2.9), and (2.21)) was introduced to account for the latent heat liberated at these boundaries. If one knows the position of the interface as a function of time, one can consider a continuous conduction problem in which a heat source (or sink) is passing through the region. If the intensity of the heat source at point x and time t is A (x,t), then the heat equation becomes:

$$T_t = \alpha T_{xx} + \frac{1}{\rho c} A(x,t) \qquad \text{in R for } t > 0, \qquad (2.24)$$

subject to boundary conditions on S but not on the interfaces.

Several numerical methods were suggested utilizing this idea in a finite-difference approximation. Dusinberre (43) refered to earlier work, where the second term of the equation was calculated for a spatial interval before the computation of the heat equation started, and the term was called the temperature suppression. During the computation of the temperature distribution this term was added (or subtracted) over one or more time steps whenever the calculated temperature of an interval tended to pass through the phase change temperature. He stated that the method easily applied to hand computation and the heat balance correctly observed, but it introduces unrealistic oscillations in the calculated temperatures. Price and Slack (83) applied a similar method to the problem defined by equations (2.6) - (2.10) and illustrated in Figure 2.2. In the finite-difference computations they calculated the temperature distribution in the region under consideration, but any one interval does not become part of the region until the heat conducted into (or away from) the interval equals the latent heat. Later, it will be pointed out that this is equivalent to a zero-order approximation of the moving boundary problem. The significance of this method is that it is directly adaptable to analog computers as long as the number of spatial increments in the region is not large.

Dusinberre (43) calls attention to the fact that one can consider the source of all the difficulty to be the discontinuity in the temperature-enthalpy relation. This is illustrated in Figure 2.4a. This discontinuity gives rise to an infinite heat capacity at the melting point. Dusinberre (43) recommends the use of a heat capacity function:

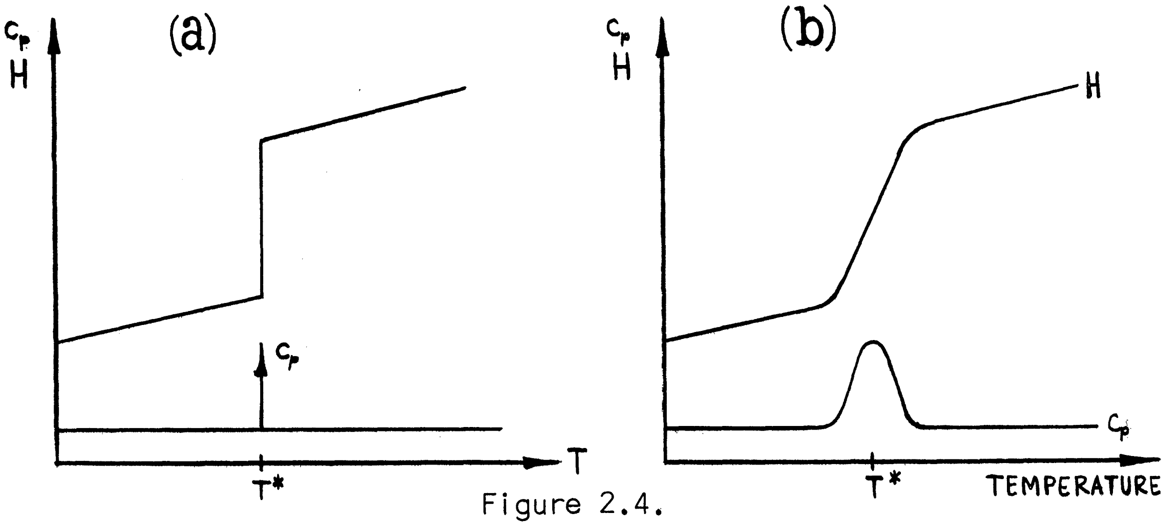

Figure 2.4.

(a) True temperature-enthalpy relation.

(b) Approximate temperature-enthalpy relation. $c_p = \frac{dH}{dT}$ is also plotted.

$$c_p = c_o + \frac{1}{a + b(T - T^*)^2} , \tag{2.25}$$

where a and b are chosen to satisfy the condition:

$$L = \pi \sqrt{ab} \tag{2.26}$$

and the resulting continuous enthalpy-temperature relation is plotted in Figure 2.4b. The larger the b/a ratio in the above relation, the better it approximates the true enthalpy-temperature relationship.

Using this heat capacity function, the problem is reduced to a simple heat conduction problem with variable coefficients. This problem can be approximated numerically, and as long as the amount of heat transferred in any time interval between any two neighboring spatial intervals is small, the approximation remains a good one. Unfortunately, the time and/or spatial increment to be used for any given b/a ratio has not been worked out, nor is the reverse relationship known.

Hashemi (53) recommends the use of an "effective heat capacity,"

$$c_{p_{eff}} = c_p + \lambda \cdot \exp \left[ - \beta^2 (T - T^*)^2 \right] , \tag{2.27}$$

with appropriately chosen values of $\lambda$ and $\beta$. The enthalpy-temperature and heat capacity-temperature relationships remain similar to Figure 2.4b. He does not claim that this is in any way superior to the heat capacity function suggested by Dusinberre; in fact, he does not mention the work of Dusinberre. On the other hand, he points out that the method is superior to others when the material does not freeze at a given fixed temperature but over a temperature range. This is probably so, but he does not substantiate this statement nor does he demonstrate convergence as this approximate enthalpy-temperature relation approaches the true enthalpy-temperature relation. Thus, the usability of these methods rests on intuitive reasoning and not on mathematical analysis or on agreement with experimental evidence. These arguments do not diminish the value of these investigations but only point out the need for more work in these areas.

Returning to the problems posed by equations (2.6) - (2.23), let us review some of the methods suggested to solve the mathematical problem as stated. The difficulty seems to arise from the interface boundary conditions: how to calculate the position of the boundary, how to approximate the slope at the boundary and how to compute the temperatures at the nodal points immediately next to the boundary. Douglas and Gallie (36) proposed a method for solving equations (2.6) - (2.10) in which the region is divided by a uniform mesh spacing $\Delta x$. They propose choosing $\Delta t$, the time step to be used at any one time, in such a way that the boundary will travel the distance $\Delta x$ in the time step $\Delta t$. This means that the time after n time steps is given by:

$$t_n = \sum_{j=0}^{n-1} \Delta t_j \qquad (2.28)$$

and the interfacial boundary is at

$$S(t_n) = n\,\Delta x. \qquad (2.29)$$

Since the boundary is always at a nodal point, the equation can be approximated by normal numerical methods. Douglas and Gallie (36) used the backward difference method which is generally stable for any value of $\Delta t/(\Delta x)^2$. Even though $\Delta t$ keeps increasing as time increases, Douglas (31) has proved the uniqueness and rate of convergence of the solution.

The other investigators used fixed $\Delta t$ for their methods. Murray and Landis (76) give two distinct methods for the solution of the problem. The first, restricted to regions of finite width, uses a fixed number of grids in each subregion. The length of these intervals is equal within each subregion, but not in the whole region, and it changes with time. In this method the boundary is again at a grid or nodal point at all times. In the second method they use fixed spatial increments along with fixed time steps. Since in this method the fusion front does not necessarily coincide with a grid point, they have to use a special variable to record the location of the interface $S_{12}$. Figure 2.5 illustrates their treatment of the boundary conditions near the interface. When the interface is closest to the q-th interval, they interpolate (and/or extrapolate) two temperature values for the interval: $T_{q1}$ and $T_{q2}$, using the known values of T at q-2, q-1, $S_{12}$, q+1 and q+2 by a second order extrapolation-interpolation formula. They use

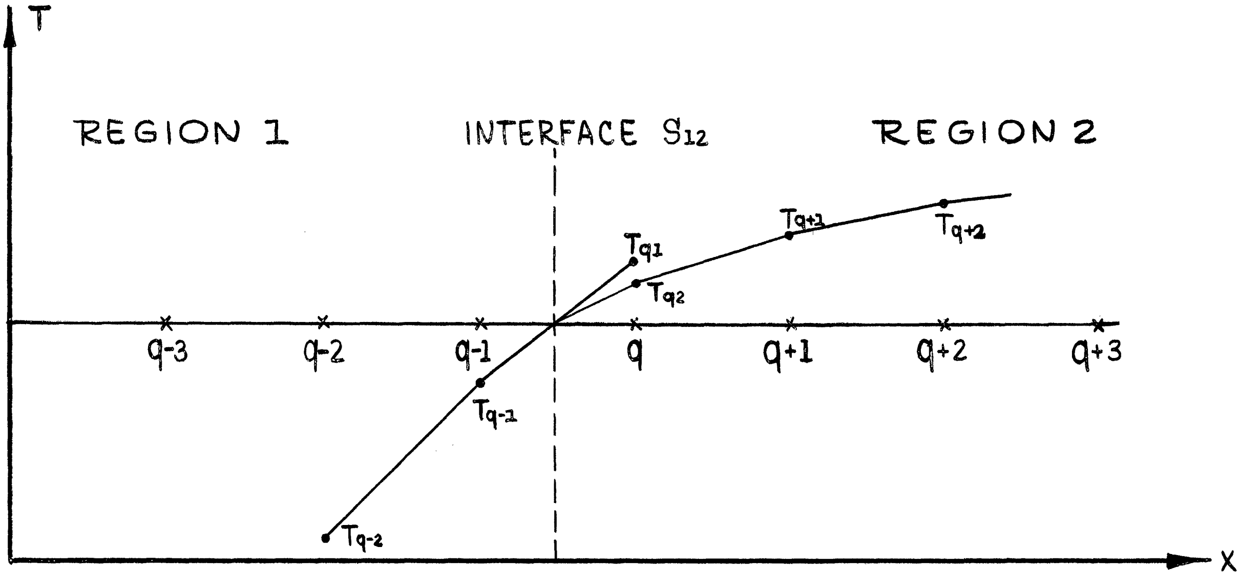

Figure 2.5.

Illustration of Murray and Landis' (76) method.

$T_{q1}$ in computing the temperature distribution by an explicit method in the region $R_1$, and they use this same temperature in computing $\frac{dT}{dx}$ near the interface. Similarly, they use $T_{q2}$ to compute the same things in $R_2$ and then they can explicitly calculate the movement of the interface. Depending on the position of the interface $S_{12}$ the temperature at the q-th grid point is $T_{q1}$ or $T_{q2}$.

This second-order approximation of the temperature at the interface can be considered an improvement of the method proposed by Price and Slack (83). They use $T_q = T^*$ during the time interval the interface is in the q-th space interval, which is a zero-order approximation. Crank (24) and Ehrlich (44) also use an uneven grid spacing directly adjacent to the interface to achieve better approximation in connection with Crank-Nicolson type calculations. Douglas and Jones (39) point out that a predictor-corrector type computation sometimes can be more efficient than the Crank-Nicolson type without loss of accuracy. Rose (85) also proposes a method similar to the one used by Lax (64) for solving non-linear hyperbolic equations. He notes that the method is independent of dimensionality, but does not illustrate this point.

Before turning to the two-dimensional investigations, let us mention some of the investigators who proved some of the mathematical theorems which provide the foundation for these methods. The existence of the solution is proved with various restrictions by Bellman (6), Douglas (31), Friedman (46,47,48), Kolodner (58), Kyner (60,61), Miranker (73) and Trench (95). The proof of uniqueness for the solution appears in the works of Boley and Weiner (9), Douglas (31), Friedman (46,47,48), Kolodner (58) and Kyner (60). Bellman (6) shows that the solution is bounded, while Boley (8) and Hamill and Bankoff (51) show that the bounds can be computed. Some of the results are applicable only under restricted circumstances; Douglas (33) shows the applicability of the numerical methods to quasi-linear problems. Thompson (92) also proves that the conditions to be placed on the non-linear problem are sometimes the same as the conditions on the linear problem.

Du Fort and Frankel (42), beside proposing an unconditionally stable numerical method, point out the need for analysis of a formal approximation, as certain approximations may lead to inconsistent results if the limiting process is carried out thoughtlessly. Setting up the initial conditions and starting the numerical computation may require special consideration as Murray and Landis (76) state. Sometimes these special considerations are simplified by the use of an approximate initial temperature distribution. This introduces a starting error which usually diminishes rapidly. Such a temperature distribution was previously used by Churchill and Teller (20) and it is used in the present investigation.

Turning to the previous investigations of the two-dimensional Stefan problem, the most recent work seems to be that of Hashemi (53). He solves the two-dimensional heat conduction problem with variable coefficients. As previously mentioned, the latent heat of fusion is taken into account by the use of an "effective heat capacity" over a finite temperature interval. Wilcox and Duty (99) propose a method in which they assume a shape for the solidifying interface, compute the

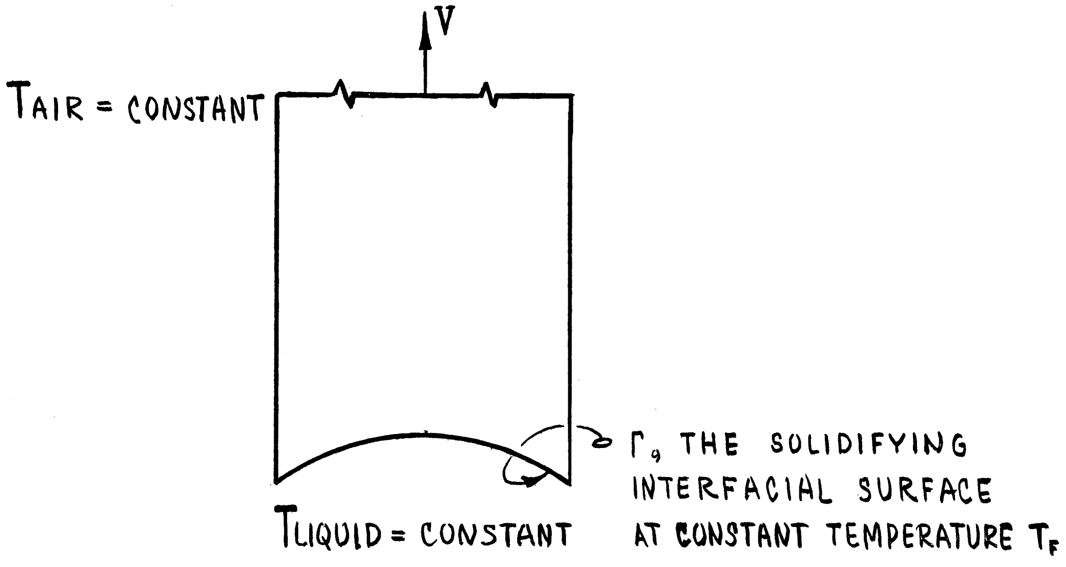
V
TAIR = CONSTANT
$\Gamma_9$ THE SOLIDIFYING INTERFACIAL SURFACE AT CONSTANT TEMPERATURE $T_F$
TLIQUID = CONSTANT

Figure 2.6.

The problem solved by Wilcox and Duty (99).

temperature distribution in the bar and then check the heat balance around the solid portion of the bar. The problem is illustrated in Figure 2.6. If the heat balance does not check, they modify their assumptions and start the computation again. Their method is not usable for transient problems, but can give the pseudo-steady state temperature distribution and interface shape. They only published a few results for the steady state case (the velocity of the freezing bar is zero) without giving the computer program or the algorithm used to solve the problem. The efficiency of their method is also not known. Allen and Severn (1) propose a method to solve equation (2.24) in two dimensions using a relaxation method. They claim that the error of their numerical computation is less than one percent in their sample problem.

# CHAPTER III

# THE HEAT TRANSFER PROBLEM WITH PHASE CHANGE

## 3.1 Introduction

The heat transfer problem with change of phase is quite complex. As it was pointed out earlier, the problem was attacked in several different ways. These methods fall into two major groups:

1. Solve a simplified problem rigorously.
2. Obtain an approximate solution to a complex problem.

Both approaches have their advantages and disadvantages. The first has the advantage that the method which was used for the solution can often be extended to more complex problems by eliminating some of the simplifying assumptions, but until then it is applicable only to the specific simplified problem or its equivalents. The second approach gives an immediate solution, but the approximation is sometimes not sufficiently accurate. In this work the first route is followed. Using the information gathered in the rigorous solution of one-dimensional problems, a method is proposed to solve a simple two-dimensional Stefan problem with sufficient accuracy to predict the shape of the interface. Many of the ideas presented in the following chapter are general enough to be used in more complex problems. Some of the possible extensions will be recommended for future investigation in Chapter VII.

In this chapter a heat transfer problem with phase change, the physical problem to be solved is defined and the equations used for its solution are derived. The problem is stated in Section 2, and its mathematical description is given in Section 3. This description is transformed into a dimensionless form in Section 4. The numerical technique used to solve the heat conduction equations is presented in Section 5. In Section 6 some characteristics of this technique are discussed. The system of grid points necessary for the numerical approximation of the mathematical problem is introduced in Section 7. In Section 8 the derivatives are approximated by finite differences and in Sections 9 and 10 the system of equations used in the numerical technique is derived. The method for solving this system of equations will be described in Section 11. The motion of the interface will be discussed in Section 12, while the question of obtaining the initial conditions are taken up in Section 13.

## 3.2 The Problem Statement

The problem may be stated as follows: a rectangular bar, made of a material which undergoes a phase transformation, is pushed through a furnace in which the temperature distribution is symmetric around the axis of this furnace, but is not uniform along this axis. Indeed, the temperature variation in the furnace is such that the temperature at one end of the bar will be below the transformation temperature, while at the other end it will be above the transformation temperature. One end of the bar, therefore, will contain the phase which is stable at low temperatures, while the other will contain the phase which is stable at high temperatures. The problem is to find the location of the phase boundary and calculate the temperature distribution. We will restrict our attention to the longitudinal center plane of the bar. This center plane is a rectangle shown in Figure 3.1.

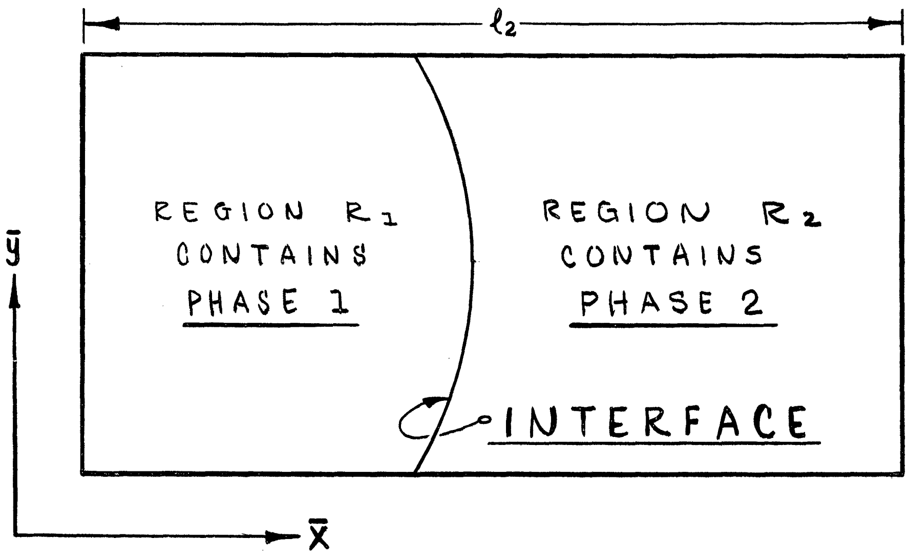

Figure 3.1.

The heat transfer problem with phase change.

The dimensions of the rectangle in which we want to compute the temperature distribution and the location of the phase boundary are $\ell_2$ and $\ell_1$ in the $\bar{x}$ and $\bar{y}$ direction respectively. Denoting the phase transformation temperature $\theta_m$, the phase which is stable below this temperature is phase 1, the other, phase 2 and identifying the regions occupied by these phases with the same numbers, we know that the temperature in region 1 ($R_1$) is always less than or equal to $\theta_m$ while in region 2 ($R_2$)

it is greater than or equal to $\theta_m$. The interface temperature is $\theta_m$. The rectangle is heated or cooled by its surroundings depending on the relative temperatures of the rectangle's side and the surroundings. Assume that the amount of heat transferred to the material or away from it depends linearly on the temperature difference between the material's surface and the surroundings, the constant of proportionality being h, the heat transfer coefficient.

The material is taken to be of constant density and therefore no movement of material occurs. Crank (25), Wilkes (100) and others solved problems without making this assumption. This assumption greatly simplifies the present problem and allows the attention to be concentrated to the moving boundary. Possible future work without this assumption is discussed in Chapter VII. The other physical constants, the heat capacity, c, the thermal conductivity, k, and the heat transfer coefficient, h, are assumed to be constants for each of the phases. This again is not necessary as Hashemi (53) previously solved a two-dimensional heat conduction problem with temperature-dependent coefficients. This assumption is used to further simplify the computation. The same kind of calculation can surely be carried out even if these constants are functions of both location and temperature, but the resulting computation will take more time and the necessary computer program will be considerably more complex.

The temperature distribution in the furnace is not necessarily constant, but may be a function of time. The bar and the furnace can be in motion relative to each other. It is important, however, that the temperature of the bar's surroundings are a known function of time. The actual temperature distributions used in the sample computation is discussed in the next chapter.

## 3.3 Mathematical Description

In this section the mathematical equations describing the problem are given. The heat balance at each point of the regions are given by Laplace's equation:

$$\frac{\partial \theta_1}{\partial \bar{t}} = \alpha_1 \nabla^2 \theta_1 \qquad \text{in } R_1, \tag{3.3.1}$$

$$\frac{\partial \theta_2}{\partial \bar{t}} = \alpha_2 \nabla^2 \theta_2 \qquad \text{in } R_2, \tag{3.3.2}$$

where $\theta_i$ is the temperature in $R_i$,

$\alpha_i$ is the thermal diffusivity, $k_i/(\rho \cdot c_i)$, $\text{cm}^2/\text{sec}$,

$\bar{t}$ is time, sec.

These partial differential equations (P.D.E. hereafter) hold inside each of the regions. At the boundaries of the regions the heat balance is given by the boundary conditions. There are two boundary conditions for each spatial variable in each subregion as the P.D.E.'s are second order in the spatial variables. Let us denote the temperature of the surroundings in the immediate neighborhood of a boundary point by $\theta_s$. The heat balance equations at the boundary points are

$$\text{when } x = 0: \qquad k_1 \frac{\partial \theta_1}{\partial \bar{t}} = h_1(\theta_1 - \theta_s), \qquad (3.3.3)$$

$$\text{when } x = \ell_2: \qquad k_2 \frac{\partial \theta_2}{\partial \bar{t}} = h_2(\theta_s - \theta_2), \qquad (3.3.4)$$

$$\text{when } y = 0: \qquad k_i \frac{\partial \theta_i}{\partial \bar{t}} = h_i(\theta_i - \theta_s), \qquad (3.3.5)$$

$$\text{when } y = \ell_1: \qquad k_i \frac{\partial \theta_i}{\partial \bar{t}} = h_i(\theta_s - \theta_i), \qquad (3.3.6)$$

where $k_i$ is the thermal conductivity in the subregion $R_i$ and $h_i$ is the heat transfer coefficient at the boundaries of the subregion $R_i$. The remaining two boundary conditions are given at the interface of the two subregions:

$$\theta_1 = \theta_2 = \theta_m \qquad (3.3.7)$$

and

$$k_1 \frac{\partial \theta_1}{\partial \bar{n}} - k_2 \frac{\partial \theta_2}{\partial \bar{n}} = \rho L \frac{d\bar{N}}{d\bar{t}}, \qquad (3.3.8)$$

where $\bar{n}$ designates the direction normal to the interface,

$\frac{d\bar{N}}{d\bar{t}}$ is the rate of motion of the interface, cm/sec,

$L$ is the latent heat of fusion, cal/gm,

$\rho$ is the density of the material, gm/cc.

Carslaw and Jaeger (18) show that equation (3.3.8) is non-linear, and Landau (62) even proposed a transformation which would move the non-linearity from the boundary condition to the P.D.E. Since that transformation does not seem to simplify the present problem, only one more simplifying transformation is made before obtaining the dimensionless form of the equations and boundary conditions. It is known that the component of the normal derivatives in a particular direction equals the directional derivative. In particular

$$\frac{\partial\theta}{\partial\bar{n}} \cos \gamma = \frac{\partial\theta}{\partial\bar{x}} , \qquad (3.3.9)$$

where $\gamma$ is the angle between the $\bar{x}$ and the normal direction. This is the same as the angle between the interface and the $\bar{y}$ direction as can be seen on Figure 3.2. Using this identity, equation (3.3.8) can be put into a more functional form:

$$\frac{\partial\theta_1}{\partial\bar{x}} - \frac{k_2}{k_1} \frac{\partial\theta_2}{\partial\bar{x}} = \frac{\rho\lambda\cos^2\gamma}{k_1} \frac{d\bar{X}}{d\bar{t}} . \qquad (3.3.10)$$

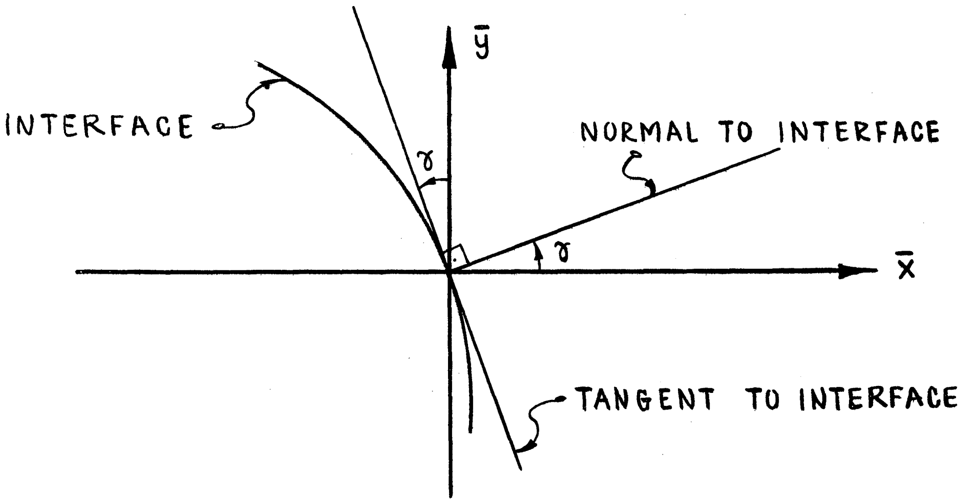

Figure 3.2.
Definition of the angle $\gamma$.

## 3.4 Transformation to Dimensionless Form

The dimensionless forms of the equations given in the previous section can be obtained by defining a suitable set of dimensionless variables. If $\theta_h$ is a reference temperature, the following dimensionless variables are introduced:

$$x = \frac{\bar{x}}{\ell_1}, \quad y = \frac{\bar{y}}{\ell_1}, \quad t = \frac{\alpha_1 \bar{t}}{\ell_1^2}, \quad T = \frac{\theta - \theta_m}{\theta_h - \theta_m}, \tag{3.4.1}$$

where T is the dimensionless temperature and x, y and t (the letters without the diacritical mark wherever such is used) are also dimensionless. It can be seen that T is so chosen that the interface temperature is always zero, and therefore the temperature in $R_1$ is always negative while it is always positive in $R_2$. Substituting these new variables into equations (3.3.1) - (3.3.7) and (3.3.10), the following equations are obtained:

$$\frac{\partial T_1}{\partial t} = \frac{\partial^2 T_1}{\partial x^2} + \frac{\partial^2 T_1}{\partial y^2} \quad \text{in } R_1, \tag{3.4.2}$$

$$\frac{\partial T_2}{\partial t} = \frac{\alpha_2}{\alpha_1} \left[ \frac{\partial^2 T_2}{\partial x^2} + \frac{\partial^2 T_2}{\partial y^2} \right] \quad \text{in } R_2 \tag{3.4.3}$$

and the boundary conditions on the sides of the rectangle become

$$\frac{\partial T_1}{\partial x} = \frac{h_1 \ell_1}{k_1}(T_1 - T_s) \qquad \text{when } x = 0, \tag{3.4.4}$$

$$\frac{\partial T_2}{\partial x} = \frac{h_2 \ell_1}{k_2}(T_s - T_2) \qquad \text{when } x = \ell_2/\ell_1\ , \tag{3.4.5}$$

$$\frac{\partial T_i}{\partial x} = \frac{h_i \ell_1}{k_i}(T_i - T_s) \qquad \text{when } y = 0, \tag{3.4.6}$$

$$\frac{\partial T_i}{\partial x} = \frac{h_i \ell_1}{k_i}(T_s - T_i) \qquad \text{when } y = 1, \tag{3.4.7}$$

while at the interface the following two conditions hold:

$$T_1 = T_2 = 0 \qquad \text{on the interface and} \tag{3.4.8}$$

$$\frac{\partial T_1}{\partial x} - \frac{k_2}{k_1}\ \frac{\partial T_2}{\partial x} = \frac{\alpha_1 \rho\lambda\cos^2\gamma}{k_1(\theta_h - \theta_m)}\ \frac{dX}{dt} \quad \text{on the interface.} \tag{3.4.9}$$

It can be seen that the dependent variables $T_1$, $T_2$ and the location of the interface depend on the following independent variables and parameters:

$$x,\ y,\ t,\ \alpha_2/\alpha_1,\ \ell_2/\ell_1,\ h_1\ell_1/k_1,\ h_2\ell_1/k_2,\ T_s,\ \frac{\alpha_1\rho\lambda\cos^2\gamma}{k_1(\theta_h-\theta_m)}, \tag{3.4.10}$$

the last one of which can be eliminated by a suitably chosen $\theta_h$, the arbitrary reference temperature. The assumption of identical properties in the two phases would further reduce the number of parameters. Neither the elimination of the last dimensionless group, nor other ways of reducing the number of groups would lead to more useful results in this case.

Dimensionless groups and variables find their greatest usefulness when a large class of problems have identical dimensionless solutions or when a correlation exists between the dimensionless groups. Neither of these conditions are applicable to our problem in the strict sense. In the general case the properties, and therefore many of the dimensionless groups, are temperature-dependent. Even more important, the most frequently changed parameter is $T_s$. As a matter of fact, $T_s$ is not even a constant parameter, but a function of both time and location. It is defined separately for each problem. Hence we do not deal with one problem, but with many problems which have the same form. Every one of these problems has a different solution, and they are as dissimilar as the $T_s$'s, the boundary temperature distributions, used.

The dimensionless groups were introduced because they are useful in a different way. Their use somewhat simplified the notation. Also, in the sample computations not the boundary conditions but rather, the material properties were varied to study how they affect the solution. This is not the usual case when one wants to use this type of program. Usually the material is given and the problem is solved using various functions for $T_s$. In either case, the use of the dimensionless temperature makes the relative temperature distribution easy to visualize.

## 3.5 The Numerical Method

We are now in position to outline a numerical procedure for solving the problem defined by equations (3.4.2) - (3.4.9). Seemingly, equation (3.4.9) can be used to find the location of the interface if one knew the temperature distribution. Also, if the location of the interface is known, equations (3.4.2) - (3.4.8) can be used to solve the temperature distribution. But since the required knowledge is not readily available, one has to solve the equations simultaneously. This is done by using a so-called marching technique. In this technique one has to know the solution to the problem at some time t. Using some method one computes the solution at a later time $t+\Delta t$. By repeating the method over and over again, the solution can be obtained at all later times. At this point one has to answer two questions: 1. Is there such a method? 2. Do we know the solution at any one time?

The answer to the first question is that the methods available do not give an exact solution, the error involved depends on the size of $\Delta t$. The larger $\Delta t$ is, the larger the error. This forces us to use relatively small $\Delta t$'s repeatedly. This means repeating the computations more often, possibly compounding the errors. The method chosen has to give a more accurate solution with the repeated use of these shorter time intervals, to be useful. In this chapter only those methods will be discussed which have this property, that is, they are convergent. This question will be further discussed in the next section.

The answer to the second question is no in most cases. However, there are ways to overcome this difficulty and these will be discussed in Section 3.13.

Returning to the first question, when a numerical method is used one usually breaks the problem into subproblems and solves these. Once a solution to these problems is available, one can go ahead with the whole problem. Sometimes the sequential solution of the subproblems is a solution of the whole. At other times a more complex way is necessary. Let us first discuss how one would go about solving equations (3.4.2) and (3.4.3), and return to the solution of the whole problem at a later time.

Common to all the numerical methods to be discussed is the approximation of the time derivative:

$$\frac{dT}{dt} \cong \frac{T^{(t+\Delta t)} - T^{(t)}}{\Delta t} , \qquad (3.5.1)$$

where the superscript identifies the time. Let us denote with $\Delta_x^2 \, T^{(t)}$ the approximation of the second partial derivative of T with respect to x taken at time t. Let us use similar notation for the derivative taken with respect to y. Following the idea developed for one-dimensional techniques, one can write the equations of three possible numerical methods:

$$\frac{T^{(t+\Delta t)} - T^{(t)}}{\Delta t} = \Delta_x^2 \; T^{(t)} + \Delta_y^2 \; T^{(t)} , \qquad (3.5.2)$$

$$\frac{T^{(t+\Delta t)} - T^{(t)}}{\Delta t} = \Delta_x^2 \; T^{(t+\Delta t)} + \Delta_y^2 \; T^{(t+\Delta t)} \qquad (3.5.3)$$

$$\frac{T^{(t+\Delta t)} - T^{(t)}}{\Delta t} = \frac{1}{2} \; \Delta_x^2 \; [T^{(t+\Delta t)} + T^{(t)} ]$$

$$+ \frac{1}{2} \; \Delta_y^2 \; [T^{(t+\Delta t)} + T^{(t)} ] \, . \qquad (3.5.4)$$

It turns out that these methods are rarely used. Equation (3.5.2) defines the explicit method because the equation contains only one unknown, and can be solved explicitly. It is seldom used in this form because stability considerations limit the size of the largest usable Δt to a very small value. Equation (3.5.3) defines the implicit method. The equation generally contains five unknowns, and the temperature at any point is obtained by solving a system of equations. This requires a considerable amount of computation and is therefore infrequently used. The Crank-Nicolson method, which is defined by equation (3.5.4) is also rarely used, for the same reason. This method is better than the implicit method as its convergence is second order in time while the implicit method is only first order. Methods requiring less computation with similar convergence characteristics have been developed. These methods are perturbations of the Crank-Nicolson method and are the ones usually used in multi-dimensional work. One, developed by Peaceman and Rachford (77) and often referred to as The Alternating Direction Implicit Method can be defined by two equations:

$$\frac{T^{(t+\Delta t/2)} - T^{(t)}}{\Delta t/2} = \Delta_x^2 \ T^{(t)} + \Delta_y^2 \ T^{(t+\Delta t/2)} , \tag{3.5.5}$$

$$\frac{T^{(t+\Delta t)} - T^{(t+\Delta t/2)}}{\Delta t/2} = \Delta_x^2 \ T^{(t+\Delta t)} + \Delta_y^2 \ T^{(t+\Delta t/2)} . \tag{3.5.6}$$

The method derived its name because it is implicit for a Δt/2 time period in the y and explicit in the x direction while for the following Δt/2 time interval it is implicit in the x and explicit in the y direction. An equivalent method was proposed by Douglas and Rachford (40) which is given by the following equations:

$$\frac{T^{*} - T^{(t)}}{\Delta t} = \Delta_x^2 \ T^{(t)} + \Delta_y^2 \ T^{*}, \tag{3.5.7}$$

$$\frac{T^{(t+\Delta t)} - T^{(t)}}{\Delta t} = \Delta_x^2 \ T^{(t+\Delta t)} + \Delta_y^2 \ T^{*}. \tag{3.5.8}$$

This is also an alternating direction implicit method. $T^{*}$ and in later equations $T^{**}$ stand for intermediate approximations of the temperature distribution. The difference between these two methods may be encountered when one tries to use them in three dimensions. If one attempts to extend the first method, it seems natural to try to introduce three time levels. One would use an implicit method for a time interval of Δt/3 in each direction. Unfortunately, the method so conceived would not have the stability characteristics of the other Crank-Nicolson type methods. On the other hand, equations (3.5.7) - (3.5.8) can be readily extended to the following convergent three-dimensional method:

$$\frac{T^{*} - T^{(t)}}{\Delta t} = \Delta_x^2 \, T^{(t)} + \Delta_y^2 \, T^{(t)} + \Delta_z^2 \, T^{*}, \qquad (3.5.9)$$

$$\frac{T^{**} - T^{(t)}}{\Delta t} = \Delta_x^2 \, T^{(t)} = \Delta_y^2 \, T^{**} + \Delta_z^2 \, T^{*}, \qquad (3.5.10)$$

$$\frac{T^{(t+\Delta t)} - T^{(t)}}{\Delta t} = \Delta_x^2 \, T^{(t+\Delta t)} + \Delta_y^2 \, T^{**} + \Delta_z^2 \, T^{*}. \qquad (3.5.11)$$

Brian (15) recently proposed a third variation of the Crank-Nicolson method which in two dimensions can be given most visually by the following three equations:

$$\frac{T^{*} - T^{(t)}}{\Delta t/2} = \Delta_x^2 \, T^{(t)} + \Delta_y^2 \, T^{*}, \qquad (3.5.12)$$

$$\frac{T^{**} - T^{(t)}}{\Delta t/2} = \Delta_x^2 \, T^{**} + \Delta_y^2 \, T^{*}, \qquad (3.5.13)$$

$$\frac{T^{(t+\Delta t)} - T^{(t)}}{\Delta t} = \Delta_x^2 \, T^{**} + \Delta_y^2 \, T^{*}. \qquad (3.5.14)$$

$T^{**}$ can be eliminated between equations (3.5.13) and (3.5.14). The resulting two equations are equivalent to those previously mentioned. This method also has a convergent three-dimensional equivalent, and the four equations defining it most visually are:

$$\frac{T^{*} - T^{(t)}}{\Delta t/2} = \Delta_x^2\ T^{(t)} = \Delta_y^2\ T^{(t)} + \Delta_z^2\ T^{*}, \tag{3.5.15}$$

$$\frac{T^{**} - T^{(t)}}{\Delta t/2} = \Delta_x^2\ T^{(t)} + \Delta_y^2\ T^{**} + \Delta_z^2\ T^{*}, \tag{3.5.16}$$

$$\frac{T^{***} - T^{(t)}}{\Delta t/2} = \Delta_x^2\ T^{***} + \Delta_y^2\ T^{**} + \Delta_z^2\ T^{*}, \tag{3.5.17}$$

$$\frac{T^{(t+\Delta t)} - T^{(t)}}{\Delta t} = \Delta_x^2\ T^{***} + \Delta_y^2\ T^{**} + \Delta_z^2\ T^{*}. \tag{3.5.18}$$

In this case again $T^{***}$ can be eliminated and further simplification of the equations result in a method which is probably better than the one proposed by Douglas and Rachford (40). A method equivalent to this was also proposed by Douglas (35). These are still being improved and generalized (see Mitchell (75) for example), but for our purposes this should give a sufficient insight into the basic methods.

There is another technique available which should be mentioned. In the previously-discussed methods only the solution at time t was used to compute the results at time t+Δt, although the implicit methods implicitly took into account the temperature distribution at times other than t. The explicit method, and on certain problems all methods, can be improved by assuming that the results calculated by them are only "predicted results." Repeating the computation using both the solution at time t and the predicted solution at time t+Δt one can obtain improved results. This is especially true with problems containing variable coefficients, and it is probably true in the problem under consideration. The disadvantage of this method is that one has to do the computations twice. Since the increased accuracy allows the use of larger $\Delta t$'s than the original methods themselves would, the amount of computation for the same degree of accuracy is not doubled. Whether there is any saving in overall computing time depends on the individual problem. Certain problems, however, cannot be solved easily without the use of this method.

It can be seen that any one of the above mentioned methods can be used to solve equations (3.4.2) and (3.4.3) with constant coefficients and with linear boundary conditions. Hashemi (53) used the predictor-corrector technique with the A.D.I. method to solve the problem with variable coefficients and non-linear boundary conditions. In this work the A.D.I. method is used for the calculations with some boundary conditions defined on a moving boundary. The location of this boundary will be determined using the explicit approximation available for equation (3.4.9), as will be stated in Section 12. When the problem is used with variable coefficients, a predictor-corrector type of computation is required and would probably also reduce the error introduced by the explicit approximation used for equation (3.4.9).

## 3.6 Characteristics of the Numerical Method

In the previous sections several numerical methods were reviewed. In these the partial derivatives appearing in the P.D.E. were approximated by finite differences. It would be desirable to know that solution of the finite difference equations (abbreviated as F.D.E. hereafter) is an approximate solution of the P.D.E. Before this can be ascertained, several characteristics of the F.D.E. have to be examined.

First, one has to make sure the F.D.E. is consistent with the P.D.E. Let us call the difference between the F.D.E. and the P.D.E. the truncation error. This error has to approach zero as a limit as the increment size of the independent variables is reduced to zero. This happens in most cases, but du Fort and Frankel (42) did discover a finite difference approximation to the $u_t = u_{xx}$ equation in which one of the terms in the truncation error was proportional to $(\Delta t/\Delta x)^2$. This meant that the consistency of this approximation depended on how $\Delta t$ and $\Delta x$ approached zero. Their method is mentioned in the literature only to point out that consistency is not automatic in every case. Most authors tend to prove consistency over and over again, although it is sufficient to refer to the work of Peaceman and Rachford (77) who showed that their method is a consistent approximation of the two-dimensional heat equation in a rectangular coordinate system. Let us now turn our attention to the other characteristics of the F.D.E.

Once consistency has been proven it is certain that the F.D.E. approximates the P.D.E. This, however, is not sufficient to insure that the solution of the F.D.E. is a solution of the problem. Other errors may be committed in approximating the boundary conditions. Still others may be committed during computation. For example, during the computation all intermediate results are represented as finite decimal numbers having a given number of significant figures. No matter what that number may be, some of the results may require a larger number, and these will be rounded off, introducing an error. During the computation these errors will occur repeatedly and may change the results significantly.

Before one can say that the numerical solution of the F.D.E. can approximate the solution of the P.D.E., one has to show that the method is convergent, that is the difference between the two solutions vanishes as the time and spatial increments become smaller and smaller. Lax and Richtmyer (65) have presented a theorem which shows that a finite difference approximation of a linear P.D.E. is convergent if the method is consistent and stable. Douglas (30), Friedman (47,48) and others have extended this proof to certain non-linear problems. It is not the object of this study to show that the present problem can be transformed to a problem for which this proof is or is not applicable, even if such a procedure were available. It was stated however that at present the problem remains somewhat undefined because the function used for the boundary condition is undefined. Therefore, only the linear portion of the problem can be investigated. It was stated above that convergence requires that the method be stable. Let us take a look at this property of the F.D.E.

A finite difference solution is stable if there is an upper limit to which any error can be amplified as the increment size is decreased. The error may already be present in the initial conditions; it may be committed during the computation or may arise from the boundary conditions. Stability in itself does not imply that the error will be small as the increment size vanishes, it merely means that it will remain bounded. One simple way to get a superficial check on stability is to assume that the solution at some time t has a Fourier series expansion in which the general term has the form

$$\phi(t) \; e^{i\beta x} \; e^{i\gamma y} \, , \tag{3.6.1}$$

where β and γ are constants and i is the imaginary unit. Since the F.D.E. acts on this error term the same way as it acts on any other term, let us substitute (3.6.1) into the finite difference form of equation (3.4.3) to obtain

$$\begin{aligned} \frac{T_{x,y}^{(t+\Delta t/2)} - T_{x,y}^{(t)}}{\Delta t/2} = \frac{\alpha_2}{\alpha_1} \Bigg[ & \frac{T_{x+\Delta x,y}^{(t)} - 2\,T_{x,y}^{(t)} + T_{x-\Delta x,y}^{(t)}}{\Delta x^2} + \\ & + \frac{T_{x,y+\Delta y}^{(t+\Delta t/2)} - 2\,T_{x,y}^{(t+\Delta t/2)} + T_{x,y-\Delta y}^{(t+\Delta t/2)}}{\Delta y^2} \Bigg] \end{aligned} \tag{3.6.2}$$

$$\begin{aligned} \frac{T_{x,y}^{(t+\Delta t)} - T_{x,y}^{(t+\Delta t/2)}}{\Delta t/2} = \frac{\alpha_2}{\alpha_1} \Bigg[ & \frac{T_{x+\Delta x,y}^{(t+\Delta t)} - 2\,T_{x,y}^{(t+\Delta t)} + T_{x-\Delta x,y}^{(t+\Delta t)}}{\Delta x^2} + \\ & + \frac{T_{x,y+\Delta y}^{(t+\Delta t/2)} - 2\,T_{x,y}^{(t+\Delta t/2)} + T_{x,y-\Delta y}^{(t+\Delta t/2)}}{\Delta y^2} \Bigg] . \end{aligned} \tag{3.6.3}$$

Substitution of (3.6.1) into (3.6.2) and (3.6.3) results in the following equations:

$$[\phi(t+\Delta t/2)-\phi(t)]e^{i\beta x}e^{i\gamma y} = \frac{\alpha_2\Delta t}{2\alpha_1\Delta x^2}[\phi(t)e^{i\gamma y}(e^{i\beta(x+\Delta x)}-2e^{i\beta x}+e^{i\beta(x-\Delta x)})]+$$

$$+\frac{\alpha_2\Delta t}{2\alpha_1\Delta y^2}[\phi(t+\Delta t/2)e^{i\beta x}(e^{i\gamma(y+\Delta y)}-2e^{i\gamma y}+e^{i\gamma(y-\Delta y)})], \tag{3.6.4}$$

$$[\phi(t+\Delta t)-(t+\Delta t/2)]e^{i\beta x}e^{i\gamma y} = \frac{\alpha_2\Delta t}{2\alpha_1\Delta x^2}[\phi(t+\Delta t)e^{i\gamma y}(e^{i\beta(x+\Delta x)}-2e^{i\beta x}+e^{i\beta(x-\Delta x)})]+$$

$$+\frac{\alpha_2\Delta t}{2\alpha_1\Delta y^2}[\phi(t+\Delta t/2)e^{i\beta x}(e^{i\gamma(y+\Delta y)}-2e^{i\gamma y}+e^{i\gamma(y-\Delta y)})]. \tag{3.6.5}$$

Division of equation (3.6.4) by $\phi(t)\ e^{i\beta x}e^{i\gamma y}$ and equation (3.6.5) by $\phi(t+\Delta t/2)e^{i\beta x}e^{i\gamma y}$ gives

$$\frac{\phi(t+\Delta t/2)}{\phi(t)} - 1 = \frac{\alpha_2\Delta t}{2\alpha_1\Delta x^2}(e^{i\beta\Delta x}-2+e^{-i\beta\Delta x}) +$$

$$+\frac{\alpha_2\Delta t}{2\alpha_k\Delta t^2}\quad\frac{\phi(t+\Delta t/2)}{\phi(t)}(e^{i\gamma\Delta y}-2+e^{i\gamma\Delta y}), \tag{3.6.6}$$

$$\frac{\phi(t+\Delta t)}{\phi(t+\Delta t/2)} - 1 = \frac{\alpha_2\Delta t}{2\alpha_1\Delta x^2}\ \frac{\phi(t+\Delta t)}{\phi(t+\Delta t/2)}(e^{i\beta\Delta x}-2+e^{i\beta\Delta x}) +$$

$$+\frac{\alpha_2\Delta t}{2\alpha_1\Delta y^2}(e^{i\gamma\Delta y}-2+e^{i\gamma\Delta y}). \tag{3.6.7}$$

Note that

$$e^{iw} - 2 + e^{-iw} = 2\cos(w) - 2 = -4\sin^2\left(\frac{w}{2}\right) \qquad (3.6.8)$$

and introduce the following simplifying notation

$$A = \frac{2\,\alpha_1 dt}{\alpha_2 \Delta x^2}, \qquad B = \frac{2\,\alpha_1 dt}{\alpha_2 \Delta y^2} . \qquad (3.6.9)$$

using the trigonometric identity of (3.6.8) and the notation of (3.6.9), equations (3.6.6) and (3.6.7) can be rewritten as

$$\frac{\phi(t+\Delta t/2)}{\phi(t)} = \frac{1 - A\sin^2\left(\frac{\beta\Delta x}{2}\right)}{1 + B\sin^2\left(\frac{\gamma\Delta y}{2}\right)}, \qquad (3.6.10)$$

$$\frac{\phi(t+\Delta t)}{\phi(t+\Delta t/2)} = \frac{1 - B\sin^2\left(\frac{\gamma\Delta y}{2}\right)}{1 + A\sin^2\left(\frac{\beta\Delta x}{2}\right)} \qquad (3.6.11)$$

Multiplying these two equations together

$$\frac{\phi(t+\Delta t)}{\phi(t)} = \frac{[1-A\sin^2(\frac{\beta\Delta x}{2})][1-B\sin^2(\frac{\gamma\Delta y}{2})]}{[1+B\sin^2(\frac{\gamma\Delta y}{2})][1+A\sin^2(\frac{\beta\Delta x}{2})]} . \qquad (3.6.12)$$

Now it is possible to show that the method is stable. The modulus of the error term at time $t = a$ is $\phi(a)$. At a later time $t = b = a + n\cdot\Delta t$ the modulus can be expressed as

$$\phi(b) = \phi(a) \left\{ \frac{[1-A\sin^2(\frac{\beta\Delta x}{2})][1-B\sin^2(\frac{\gamma\Delta y}{2})]}{[1+A\sin^2(\frac{\beta\Delta x}{2})][1+B\sin^2(\frac{\gamma\Delta y}{2})]} \right\}^n , \qquad (3.6.13)$$

where $n \geqq 1$.

It can be seen that the error term, no matter what its origin, does not increase as n increases. The same can be shown for equation (3.4.2), only A and B would not contain $\alpha_1$ and $\alpha_2$. It may be worthwhile to mention at this time that the error introduced at time t = a would tend to decrease as time (b) increases. It is important, however, that for any given time t = b (>a) the error does not increase as $\Delta t$ is decreased (n increased). Thus the method would appear stable. Probably it is, but it must be pointed out again that this has not yet been proven. The boundary conditions are part of the numerical method just as the F.D.E. is. Since the boundary condition is not really specified, this part of the proof remains to be done. In practice, however, when a computer program is available, it is easier to use the computer to test stability and convergence than to go back to the theoretical problem and try some transformation which would show the method to be stable. When the boundary conditions are undetermined or complex this may be the only sensible route to follow once one is reasonably assurred of success. The question of convergence will be further discussed in Chapter V along with the computed results.

## 3.7 The System of Grid Points

In order to derive the system of F.D.E.'s representing the P.D.E.'s and their boundary conditions, it is necessary to establish a system of grid points in the region outlined in Figure 3.1. The rectangular symmetry of the problem and the rectangular coordinate system adopted for

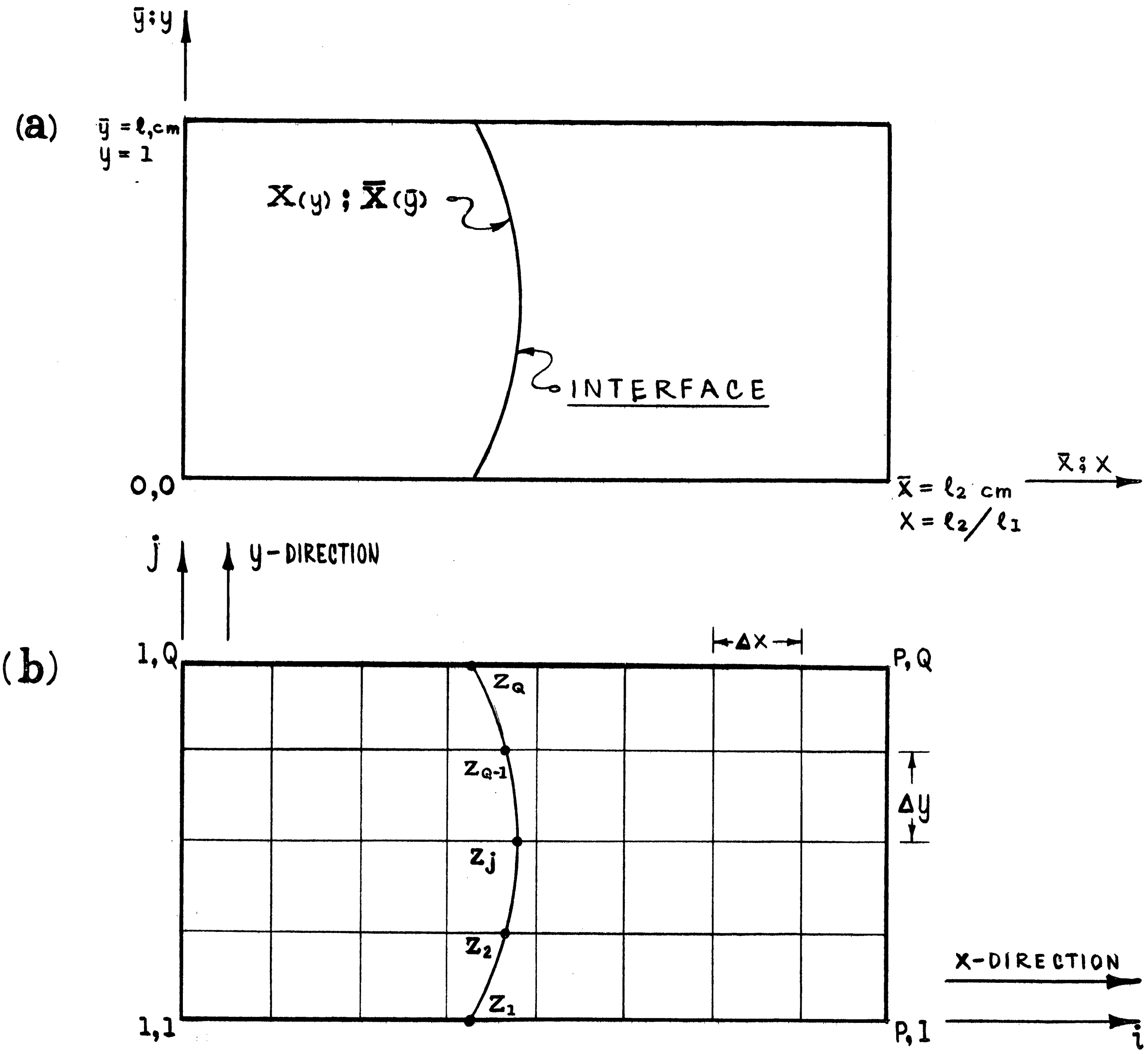

Figure 3.3.

Coordinates used: (a) for the mathematical description,
(b) for the numerical solution.

the mathematical description suggests the adoption of a rectangular grid. For easy comparison of the various coordinate systems, Figure 3.3b, which shows the proposed grid points, is placed below Figure 3.3a, which shows the coordinates used to describe the mathematical problem.

The grid points are at the intersection of horizontal and vertical lines. Let us call the horizontal lines "x-lines" as they are parallel to the x axis, and call the vertical lines "y-lines" as they are parallel to the y axis. Let us use P equally spaced y-lines and Q equally spaced x-lines. The first and last of these lines coincide with the sides of the rectangle. The interval separating the grid points is $\Delta x$ on an x-line, while it is $\Delta y$ on a y-line, where

$$\Delta x = \frac{\ell_2}{\ell_1(P-1)} \qquad \text{and} \qquad \Delta y = \frac{1}{Q-1} \quad . \tag{3.7.1}$$

Each grid point is identified by a pair of integers (i,j) where the possible values for the integers are i = 1, 2, ..., P and j = 1, 2, ..., Q. The grid point identified by (i,j) corresponds to the point having coordinates (x,y) where $x = (i-1)\,\Delta x$ and $y = (j-1)\,\Delta y$.

The interface between the two regions can be specified most readily by defining an extra set of interface points. These points are placed at the intersection of the interface with the grid lines. In the set of problems under consideration the interface always intersects all the x-grid lines and the extra Q grid points defined at these intersections will be called "x-interface points." The x-interface points will have coordinates $(z_j,j)$, where $z_j$ is a real number, usually not an integer. When the interface intersects the y-grid lines, the intersection will be called a "y-interface point." The number of y interface points varies during the problem, but the largest number which can occur in the problems under consideration is Q-1. These will be defined during the computation as needed.

In future discussions, it will be useful to interpret the definition of the interface points in a broader sense than just as points having given coordinates. Let us choose a suitable value for $\varepsilon$ as the radius of the surroundings of the interface point, and we will assume that the interface point includes its surroundings. Using this definition point (i,j) will be considered to be in region $R_1$ if $i \leq z_j - \varepsilon$, and in region $R_2$ if $i \geq z_j + \varepsilon$. When $z_j - \varepsilon < i < z_j + \varepsilon$, the point is considered to coincide with the interface point. In some parts of the problem different values for $\varepsilon$ may be chosen, as will be pointed out in later sections.

The temperature at any given point is identified by the subscripts i,j denoting the location of the point to which it belongs. Since the two regions occupying the rectangle are not overlapping, the location of the point identifies the region to which it belongs. The interface belongs to both regions, but the temperature at the interface is identical in both regions and therefore the specific region does not have to be identified even here. $T_{i,j}$ stands for $T_1$ if point (i,j) belongs to region $R_1$ and it stands for $T_2$ when the point (i,j) is in region $R_2$. $T_{i,j}$ will have to be computed by the F.D.E. corresponding the equations (3.4.2) or (3.4.3) depending on whether point (i,j) is in region $R_1$ or $R_2$. At this time, it is also convenient to rewrite equations (3.4.2) and (3.4.3) in a common form:

$$\frac{\partial T}{\partial t} = \frac{\alpha_r}{\alpha_1} \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right), \qquad (3.7.2)$$

where the subscript r refers to the region, being 1 in region $R_1$ and 2 in region $R_2$. Equation (3.7.2) is applicable in either region. It is not a new equation, only a new way to write equations (3.4.2) and (3.4.3).

Since the computation by the numerical method is carried out by a marching technique where only the solution at time $t = a$ is required to compute the solution at time $t = a + \Delta t/2$, with some miscellaneous side information, it is often sufficient to know only the temperature at the present time, $t = a$, as the solution is computed for the next time interval, $t = a + \Delta t/2$. Sometimes, however, it is expedient to record the temperature distribution for three time levels, that is, for the complete time interval $\Delta t$. This was done in the present computations, and although it may not be necessary, it allows a certain degree of flexibility which is very useful during the development of procedures. Let us denote the temperature at time $t = a$ as $T1_{i,j}$,* the temperature at time $t = a + \Delta t/2$ as $T2_{i,j}$ and the temperature at time $t = a + \Delta t$ as $T3_{i,j}$. As soon as all the $T3_{i,j}$'s have been computed at the end of a time interval and these computed results are printed out or preserved otherwise, as needed, the $T3_{i,j}$'s become the starting $T1_{i,j}$'s for the

*The symbols T1, T2, T3, T11, T1P, TJ1, TJQ, BT, GM, BU, BUU, B1 and B11 were used in the computer program and are adopted for the present discussion. These are distinct symbols, not multiplication of several separate symbols. To eliminate any confusion, whenever such a symbol is multipiied by a constant or variable a "." will be used to indicate the multiplication.

next time step. Since the old values of the $T2_{i,j}$'s and $T3_{i,j}$'s are no longer needed, they are preserved only if they are printed out or saved otherwise. At this point the process is repeated and a new set of values are computed for T2 and T3. A separate variable n is used to calculate time. In general at the beginning of each computational cycle $T1_{i,j}$ designates the temperature at the point (i,j) at time $t = t_o +$ $+n\ \Delta t = a$, where $t_o$ stands for the time at which the computation was started. It is customary, but by no means necessary, to choose $t_o$ to be zero.

## 3.8 Approximation of the Derivatives

When the numerical method is used, the derivatives appearing in the P.D.E.'s and in the boundary conditions are approximated by finite differences. In Section 3.5 it was already stated that the time derivative is given by

$$\frac{\partial T}{\partial t} \cong \frac{T^{(t+\Delta t)} - T^{(t)}}{\Delta t} \tag{3.5.1}$$

Equations (3.6.2) and (3.6.3) already used the following approximations:

$$\frac{\partial^2 T}{\partial x^2} \cong \Delta_x^2\ T = \frac{T_{x-\Delta x,y} - 2T_{x,y} + T_{x+\Delta x,y}}{\Delta x^2}\ , \tag{3.8.1}$$

$$\frac{\partial^2 T}{\partial y^2} \cong \Delta_y^2\ T = \frac{T_{x,y-\Delta y} - 2T_{x,y} + T_{x,y+\Delta y}}{\Delta y^2}\ . \tag{3.8.2}$$

These and other finite difference approximations are obtained from the Taylor series expansion of the function. Since the derivatives are all partial derivatives with respect to one variable it is sufficient to expand the function with respect to one of the variables:

$$f(u+\Delta u) = f(u) + \Delta u \frac{\partial f(u)}{\partial u} + \frac{(\Delta u)^2}{2!} \frac{\partial^2 f(u)}{\partial u^2} + \dots + \frac{(\Delta u)^{n-1}}{(n-1)!} \frac{\partial^{n-1} f(u)}{\partial u^{n-1}} + \frac{(\Delta u)^n}{n!} \frac{\partial^n f(\eta)}{\partial u^n}, \qquad (3.8.3)$$

where $u \leqq \eta \leqq u + \Delta u$ in the remainder term. In the approximations the remainder term is assumed to be small and negligible. The truncation error so introduced is $\varepsilon = \frac{(\Delta u)^n}{n!} \frac{\partial^n f(\eta)}{\partial u^n} = 0\ (\Delta u)^n$. Equation (3.5.1) is obtained by taking $n = 2$, $f = T$, $u = t$:

$$T(t+\Delta t) = T(t) + \Delta t \frac{\partial T(t)}{\partial t} + 0(\Delta t)^2 \qquad (3.8.4)$$

and solving for the desired partial derivative. By using the same approach similar formulas for the first order space derivatives can also be derived.

In some cases this approximation is not sufficiently accurate. A better approximation can be derived from the following two expansions

$$T_{x+m\ \Delta x,y} = T_{x,y} + m\Delta x \frac{\partial T}{\partial x} + \frac{(m\Delta x)^2}{2!} \frac{\partial^2 T}{\partial x^2} + 0(\Delta x)^2, \qquad (3.8.5)$$

$$T_{x-s\,\Delta x,y} = T_{x,y} - s\Delta x\,\frac{\partial T}{\partial x} + \frac{(s\Delta x)^2}{2!}\,\frac{\partial^2 T}{\partial x^2} + O(\Delta x)^3, \qquad (3.8.6)$$

where m and s have similar orders of magnitude. Multiplying equation (3.8.5) by $s^2$ and equation (3.8.6) by $m^2$ and then subtracting the latter equation from the former

$$s^2 T_{x+m\Delta x,y} - m^2 T_{x-s\Delta x,y} = (s^2-m^2)T_{x,y} + sm(s+m)\Delta x\,\frac{\partial T}{\partial x} + O(\Delta x)^3 \qquad (3.8.7)$$

is obtained. The remainder term in this equation is third order while in equation (3.8.4) it was second order. The first order derivative can be obtained upon rearranging equation (3.8.7):

$$\frac{\partial T}{\partial x} \cong \frac{\frac{S}{m(s+m)}\,T_{x+m\Delta x,y} - \frac{s-m}{sm}\,T_{x,y} - \frac{m}{s(m+s)}\,T_{x-s\Delta x,y}}{\Delta x}\;. \qquad (3.8.8)$$

This approximation requires the use of three points, but these do not have to be spaced at equal intervals, thus the approximation is usable near the interface.

The second order derivatives are obtained from the following two expansions:

$$T_{x+\Delta x,y} = T_{x,y} + \Delta x\,\frac{\partial T}{\partial x} + \frac{(\Delta x)^2}{2!}\,\frac{\partial^2 T}{\partial x^2} + \frac{(\Delta x)^3}{3!}\,\frac{\partial^3 T}{\partial x^3} + O(\Delta x)^4, \qquad (3.8.9)$$

$$T_{x-\Delta x,y} = T_{x,y} - \Delta x\,\frac{\partial T}{\partial x} + \frac{(\Delta x)^2}{2!}\,\frac{\partial^2 T}{\partial x^2} - \frac{(\Delta x)^3}{3!}\,\frac{\partial^3 T}{\partial x^3} + O(\Delta x)^4, \qquad (3.8.10)$$

Upon addition of these two equations, the equation

$$T_{x+\Delta x,y} + T_{x-\Delta x,y} = 2T_{x,y} + (\Delta x)^2 \frac{\partial^2 T}{\partial x^2} + 0(\Delta x)^4 \qquad (3.8.11)$$

is obtained and this can be solved to give equation (3.8.1). Equation (3.8.2) is obtained by a similar derivation by using the Taylor series expansion at point (x,y) to find the temperatures at the points (x,y+Δy) and (x,y-Δy). These approximations of the second derivatives are used whenever feasible, but when the grid points are not separated by even grid spacing a different approximation is needed. This is again obtained by the use of the Taylor series expansions at point (x,y) to find the temperatures at points (x+m·Δx,y) and (x-s·Δx,y), which are given in equations (3.8.5) and (3.8.6). In this case, let us multiply equation (3.8.5) by s and (3.8.6) by m and then add the two equations:

$$s\ T_{x+m\Delta x,y} + m\ T_{x-s\Delta x,y} = (m+s)\ T_{x,y} + \frac{ms(m+s)(\Delta x)^2}{2!} \frac{\partial^2 T}{\partial x^2} + 0(\Delta x)^4 \qquad (3.8.12)$$

It can be seen that this equation is similar to (3.8.11), but the truncation error in (3.8.11) is $0(\Delta x)^4$ while in (3.8.12) it is only $0(\Delta x)^3$. Solving (3.8.12) to obtain the derivative:

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{\frac{2}{m(m+s)} T_{x+m\cdot\Delta x,y} - \frac{2}{ms} T_{x,y} + \frac{2}{s(m+s)} T_{x-s\cdot\Delta x,y}}{(\Delta x)^2}, \qquad (3.8.13)$$

This approximation is usable when the grid points are not evenly spaced, and it can be noticed that when r = s = 1 equation (3.8.13) becomes identical to (3.8.1). Again, similar derivation can be carried out to obtain the y derivative.

In some cases the second order derivative is approximated using two grid points and the first derivative. This approximation is obtained by solving the Taylor series expansion itself,

$$T_{x+\Delta x,y} = T_{x,y} + \Delta x \frac{\partial T}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 T}{\partial x^2} + O(\Delta x)^3 \tag{3.8.14}$$

or

$$T_{x-\Delta x,y} = T_{x,y} - \Delta x \frac{\partial T}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 T}{\partial x^2} + O(\Delta x)^3 \tag{3.8.15}$$

to get the desired derivatives:

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{2T_{x+\Delta x,y} - 2T_{x,y}}{\Delta x^2} - \frac{2}{\Delta x} \frac{\partial T}{\partial x} \tag{3.8.16}$$

or

$$\frac{\partial^2 T}{\partial x^2} \cong \frac{2T_{x-\Delta x,y} - 2T_{x,y}}{\Delta x^2} + \frac{2}{\Delta x} \frac{\partial T}{\partial x} \quad . \tag{3.8.17}$$

## 3.9 The Equations Defining $T^{(t+\Delta t/2)}$

Before the equations defining T2 (=$T^{(t+\Delta t/2)}$) at each grid point can be written down, it is necessary to point out that the equations are slightly different at the various points. This is due to the fact that there are two different types of points:

1. Interior points
2. Boundary points.

The boundary points can again be subdivided into groups, but let us first define the interior points. In Figure 3.4.(a) the system of grid points are reproduced, while certain special features are emphasized in the other parts of the figure. An interior point (i,j) is shown in 3.4.(b). It is characterized by the four neighboring grid points with coordinates (i+1,j), (i-1,j), (i,j+1), (i,j-1), all of which lie in the same region. Any grid point which is not an interior point is a boundary point. A boundary point may belong to one or more of the following eight classifications:

1. i = 1. There is no grid point with coordinates (0,j).
2. i = P. There is no grid point with coordinates (P+1,j).
3. Point (i+1,j) does not belong to the same region as (i,j).
4. Point (i-1,j) does not belong to the same region as (i,j).
5. j = 1. There is no grid point with coordinates (i,0).
6. j = Q. There is no grid point with coordinates (i,Q+1).

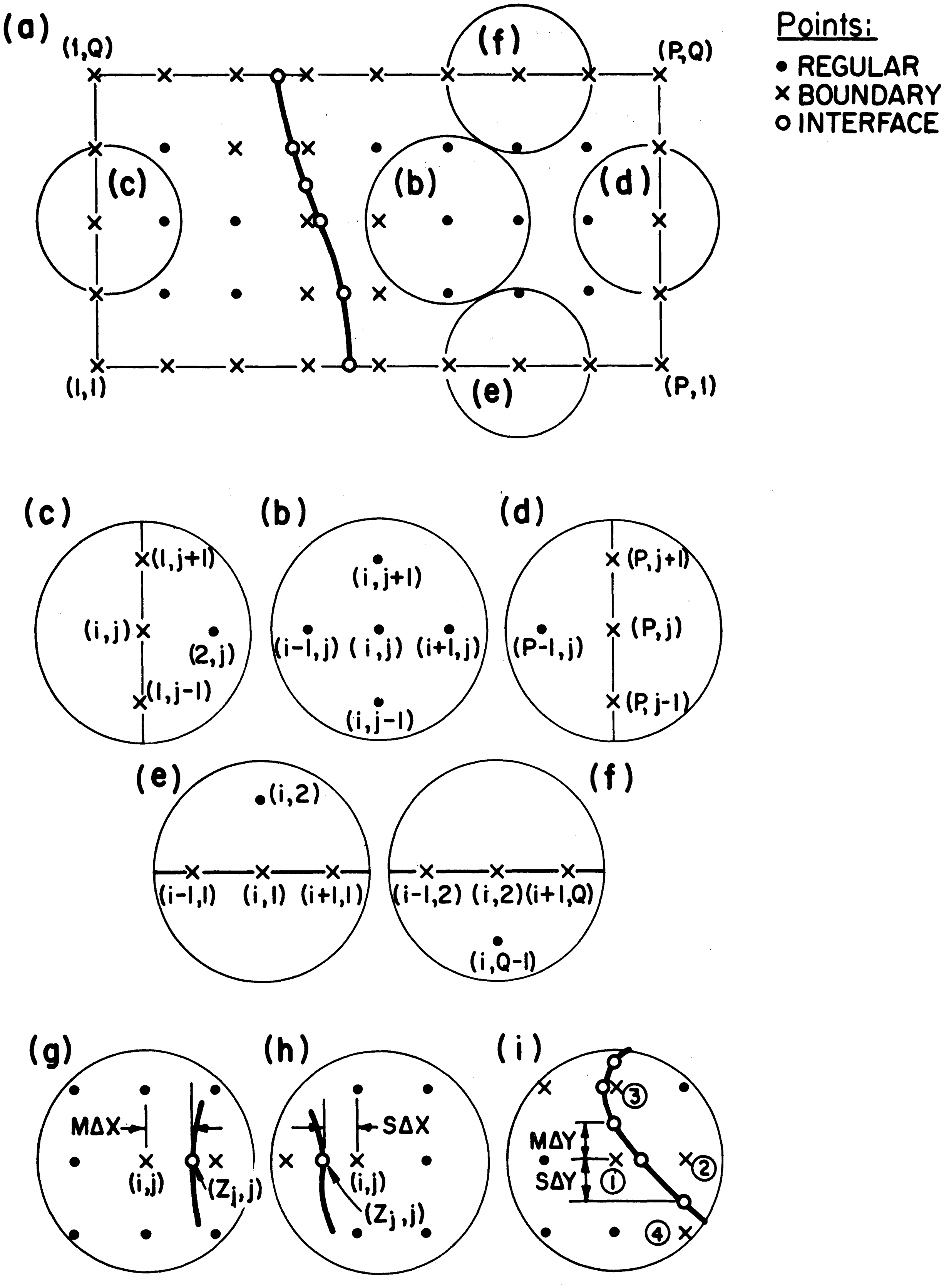

Figure 3.4.

Arrangement of the Grid Points

7. Point (i,j+1) does not belong to the same region as (i,j).

8. Point (i,j-1) does not belong to the same region as (i,j).

These classifications are not exclusive, that is, a given boundary point may belong to more than one group. In the group of problems studied, this statement may be restricted. The first four classifications are exclusive among themselves (but do not exclude a point belonging to any one of the first four groups also belonging to some of the latter four).

Returning to the problem at hand, the method for solving the problem is given by equation (3.5.5) and (3.5.6), the first one of which is used to find T2. The problem is stated by equation (3.7.2) subject to the boundary conditions stated in Section 3.4. Before writing down the equations in detail, let us rearrange the approximation corresponding to (3.5.5) from

$$\frac{T2_{i,j} - T1_{i,j}}{\Delta t/2} = \frac{\alpha_r}{\alpha_l} \Delta_x^2 T1_{i,j} + \frac{\alpha_r}{\alpha_l} \Delta_y^2 T2_{i,j} \qquad (3.9.1)$$

to the following form:

$$(\Delta x)^2 \Delta_y^2 T2_{ij} - 2\cdot \frac{\alpha_l \cdot (\Delta x)^2}{\alpha_r \cdot \Delta t} T2_{ij} = -(\Delta x)^2 \Delta_x^2 T1_{ij} - 2\cdot \frac{\alpha_l \cdot (\Delta x)^2}{\alpha_r \; \Delta t} T1_{ij}. \qquad (3.9.2)$$

Let us introduce the notation

$$L = \frac{\alpha_l (\Delta x)^2}{\alpha_r \Delta t}, \qquad R^2 = \left(\frac{\Delta x}{\Delta y}\right)^2, \qquad BU = -2(R^2+L) \qquad (3.9.3)$$

$$F_1 = (\Delta x)^2 \, \Delta_y^2 \; T2_{ij} - 2L\cdot T2_{ij}, \qquad (3.9.4)$$

$$F_2 = -\,(\Delta x)^2 \, \Delta_x^2 \; T1_{i,j} - 2L\cdot T1_{i,j}. \qquad (3.9.5)$$

Using this notation equation, (3.9.2) becomes $F_1 = F_2$. We will use the approximations developed in the previous section to find $F_1$ and $F_2$ for the various types of grid points.

At an interior point (shown on Figure 3.4.b) equations (3.8.1) and (3.8.2) can be used to obtain:

$$F_1 = R^2\cdot T2_{i,j-1} - BU\cdot T2_{i,j} + R^2\cdot T2_{i,j+1} \qquad (3.9.6)$$

and

$$F_2 = T1_{i-1,j} - 2\cdot T1_{i,j} + T1_{i+1,j} - 2L\cdot T1_{i,j}. \qquad (3.9.7)$$

At the boundary points some modification of either equation (3.9.6) or equation (3.9.7) or both may be necessary. To approximate $F_2$ in equation (3.9.6) equation (3.8.1) was used, but (3.8.1) had to be replaced by some other approximation in computing $F_2$ at boundary points which fall into any of the first four classifications.

When i = 1, as in Figure 3.4.(c) the boundary point has to satisfy boundary condition (3.4.4) in addition to equation (3.7.2). Let us denote the temperature of the surroundings near the boundary point (1,j) as $TII_j$. Using equation (3.8.16) and the derivative given in the boundary condition, we obtain

$$F_2 = (2+2H)TI_{ij} - 2\cdot TI_{i+1,j} - 2H\cdot TII_j - 2L\cdot TI_{ij}, \tag{3.9.8}$$

where H stands for $(\Delta X h_r \ell_l / k_r)$. Similarly, when i = P, equation (3.8.17) is used to approximate $\Delta_x^2 TI_{i,j}$ along with the boundary condition given in equation (3.4.5) to obtain

$$F_2 = -2\cdot TI_{i-1,j} + (2H+2)\ TI_{i,j} - 2H\cdot TIP_j - 2L\cdot TI_{ij}, \tag{3.9.9}$$

where $TIP_j$ is the temperature of the surroundings near the boundary point (P,j) as can be seen on Figure 3.4.(d).

When the boundary point falls into group 3 or 4, equation (3.8.13) is used in setting up $F_2$. In each of these cases the equation uses the temperature at one of the interface points where the temperature is given by the boundary condition (3.4.8), and it is zero. When point (i,j) belongs to group 3, which is illustrated on Figure 3.4.(g),

$$F_2 = -\frac{2}{1+m}\ TI_{i-1,j} + \frac{2}{m}\ TI_{i,j} - 2L\cdot TI_{i,j}, \tag{3.9.10}$$

while if it belongs to group 4, which is illustrated on Figure 3.4.(h),

$$F_2 = \frac{2}{s}\ TI_{ij} - \frac{2}{1+s}\ TI_{i+1,j} - 2L\cdot TI_{i,j}, \tag{3.9.11}$$

where $m = z_j - i$ and $s = i - z_j$.

When a boundary point belongs to any one of the last four classifications, equation (3.8.2) cannot be applied to compute $F_l$. Let us first see what $F_l$ is when the boundary point does not belong to more than one of these last four groups.

When j = l, the arrangement of grid points is shown in Figure 3.4.(e), an approximation similar to the one given by equation (3.8.16) can be used to set up $F_l$. The necessary first derivative is given by the boundary condition (3.4.6). Adopt the notation $G = (\Delta y h_r \ell_l / k_r)$ and $BUU = -2(R^2+GR^2+L)$ to obtain the equation

$$F_l = BUU \cdot T2_{i,j} + 2R^2 \cdot T2_{i,j+l} + 2R^2 G \cdot TJl_i, \qquad (3.9.12)$$

where $TJl_i$ stands for the temperature of the surroundings near point (i,l). The procedure at a point (i,Q) is similar; if the temperature of the surroundings near this point is $TJQ_i$ and the value of the derivative needed in an approximation similar to the one given by (3.8.7) is given in boundary condition (3.4.7), then

$$F_l = 2R^2 \cdot T2_{i,j-l} + BUU \cdot T2_{i,j} + 2R^2 G \cdot TJQ_i. \qquad (3.9.13)$$

The grid points in this case are shown in Figure 3.4.(f).

Some of the grid points which belong to the last two boundary group classifications are shown in Figure 3.4.(i). Denote the coordinates of the interface point (i,w) if w ≥ j, (i,v) if v ≤ j; let m = w-j and s = j-v. When the boundary point, such as point l in the figure, belongs to group 7,

$$F_I = \frac{2R^2}{1+m} T2_{i,j-1} - 2(\frac{R^2}{m} + L) T2_{i,j}, \qquad (3.9.14)$$

while if it belongs to group 8, as point 2 on the figure does,

$$F_I = -2(\frac{R^2}{s} + L) T2_{i,j} + \frac{2R^2}{1+s} T2_{i,j+1}. \qquad (3.9.15)$$

Consider the case of the boundary point belonging to both groups 7 and 8, for example point 3 on the diagram. In this case

$$F_I = -2(\frac{R^2}{ms} + L) T2_{i,j}. \qquad (3.9.16)$$

When the boundary point belongs to both groups 5 and 7,

$$F_I = -2(\frac{R^2}{m} + \frac{R^2G}{m} + L) T2_{i,j} + \frac{2R^2G}{m} TJ1_i, \qquad (3.9.17)$$

while if it belongs to groups 6 and 8,

$$F_I = -2(\frac{R^2}{s} + \frac{R^2G}{s} + L) T2_{i,j} + \frac{2R^2G}{s} TJQ_i. \qquad (3.9.18)$$

This completes the list of forms equation (3.9.2) can take at the various types of points. Before reviewing this list, let us state again the assumptions - some of them implicit until now - used in limiting the type of grid points which can occur. These include:

(a) $P \geqq 4$, $Q \geqq 2$,

(b) the interface intersects each x grid line exactly once, and

(c) $2 \leqq z_j \leqq Q-1$.

Let us write the equation (3.9.2) in the form

$$A\cdot T2_{i,j-1}+B\cdot T2_{i,j}+C\cdot T2_{i,j+1} = F_2+E. \qquad (3.9.19)$$

Table 3.1 summarizes the values of coefficients A, B, C, and E at grid points which do not belong to the 7th or 8th group of boundary points. Table 3.2 lists the values of $F_2$ at any grid point.

Table 3.1.

Values of A, B, C and E in equation (3.9.19)

| j | A | B | C | E |
|---|---|---|---|---|
| Does not satisfy conditions in boundary point class 5, 6, 7 or 8 | $R^2$ | BU | $R^2$ | 0 |
| = 1 | 0 | BUU | $2R^2$ | $-2R^2G\cdot TJ1_i$ |
| = Q | $2R^2$ | BUU | 0 | $-2R^2G\cdot TJQ_i$ |

Table 3.2.

Values of $F_2$

| i | $F_2$ |
|---|---|
| Does not satisfy conditions in Boundary Point Class 1, 2, 3 or 4 | $- T1_{i,j}+2(1-L)T1_{i,j}-T1_{ij}$ |
| = 1 | $-2H\cdot T11_j+2(1+H-L)T1_{ij}-2\cdot T1_{i+1,j}$ |
| = P | $-2\cdot T1_{i-1,j}+2(1+H-L)T1_{ij}-2H\cdot T1P_j$ |
| Satisfies condition for Boundary Point in Class 3 | $-\frac{2}{1+m} T1_{i-1,j} + (\frac{2}{m} - 2L) T1_{i,j}$ |
| Satisfies condition for Boundary Point in Class 4 | $(\frac{2}{s} - 2L) T1_{ij} - \frac{2}{1+s} T1_{i+1,j}$ |

Equation (3.9.19) can be solved for any of the unknown $T2_{i,j}$'s as part of the solution of a system of equations. Since i is the same for each unknown, the system contains at most Q unknowns, and there are at least P sets of equations to be solved. In some cases there may be more.

This is so, because all the equations in a set contain unknowns only at grid points belonging to the same region. If not all the grid points on a y grid line lie in one region, these points will give rise to more than one set of equations. In general, for any given values of i the equations can be set up in order of increasing j, that is j = 1, 2, ..., max. The value of max is the j coordinate of the first boundary point encountered. When max is set, the equations can be solved. If max = Q, all the equations for the given i have been solved. If not, min = max+1 is defined, and a new set of equations are defined for j = min, min+1,..., max where the new value of max is the j coordinate of the next boundary point belonging to group 6 or 7. As i is constant for each set of equations, it can be suppressed once the equations are set up. The coefficients appearing in each equation can be identified by the subscript j as follows:

$$A_j\cdot T2_{j-1}+B_j\cdot T2_j+C_j\cdot T2_{j+1} = D_j. \qquad (3.9.20)$$

The method used for solving these equations is discussed in Section 11.

## 3.10 The Equations Defining $T^{(t+\Delta t)}$

In this chapter equation (3.7.2) will be approximated by equations corresponding to equation (3.5.6), which was used to illustrate the method. The general form of the numerical equations used to approximate equation (3.7.2) is

$$\frac{T3_{i,j} - T2_{i,j}}{\Delta t/2} = \frac{\alpha_r}{\alpha_l} \Delta_x^2 \; T3_{i,j} + \frac{\alpha_r}{\alpha_l} \Delta_y^2 \; T2_{i,j} , \qquad (3.10.1)$$

which can also be written as

$$(\Delta x)^2 \; \Delta_x^2 \; T3_{i,j} - 2L \cdot T3_{i,j} = - (\Delta x)^2 \; \Delta_y^2 \; T2_{i,j} - 2L \cdot T2_{i,j} . \qquad (3.10.2)$$

Let us introduce the following notation:

$$F_3 = (\Delta x)^2 \; \Delta_x^2 \; T3_{i,j} - 2L \cdot T3_{ij} \qquad (3.10.3)$$

and

$$F_4 = - (\Delta x)^2 \; \Delta_y^2 \; T2_{i,j} - 2L \cdot T2_{i,j} . \qquad (3.10.4)$$

It can be seen that $F_l$ is similar to $F_4$ with the exception of a negative sign in front of the term containing the spatial derivatives approximation. Almost the same can be said about $F_2$ and $F_3$, but in this case Tl is replaced by T3 in addition to the above mentioned sign difference. Since the boundary and interior points remained the same, and the derivative to be approximated did not change, the equations to be derived correspond to the ones derived in the previous section with some modifications. It is convenient at this point to introduce

$$Bl = - 2 - 2L \quad \text{and } Bll = - 2 - 2L - 2H . \qquad (3.10.5)$$

This is done to keep the coefficients simple in the expressions containing the unknown variables. Using equations (3.8.1) and (3.8.2) at an interior point

$$F_3 = T3_{i-1,j} + B1\cdot T3_{i,j} + T3_{i+1,j} \qquad (3.10.6)$$

and

$$F_4 = -R^2.T2_{i,j-1} + 2(R^2 - L)\ T2_{i,j} - R^2.T2_{i,j+1} \qquad (3.10.7)$$

is obtained. If the similarity between equations (3.9.6) and (3.10.6) is not immediately apparent, it is because of the difference in the notation used. There is a like similarity between equations (3.9.7) - (3.9.18) and (3.10.7) - (3.10.18).

Let us turn to the boundary points. The arrangement of grid points when i = 1 is shown in Figure 3.4.(c). Using the boundary condition (3.4.4) and equation (3.8.16)

$$F_3 = B11\cdot T3_{i,j} + 2\cdot T3_{i+1,j} + 2H\cdot T11_j \qquad (3.10.8)$$

is obtained. When i = P the grid point arrangement is shown in Figure 3.4.(d). Application of the boundary condition (3.4.5) along with equation (3.8.17) gives

$$F_3 = 2\cdot T3_{i-1,j} + B11\cdot T3_{i,j} + 2H\cdot T11_j. \qquad (3.10.9)$$

When the boundary point falls into group 3, which is illustrated in Figure 3.4.(g), or group 4, which is illustrated in Figure 3.4.(h), equation (3.8.8) is used to obtain the required approximation. Boundary condition (3.4.8) applies in either case. If point (i,j) belongs to group 3,

$$F_3 = \frac{2}{1+m} T3_{i-1,j} - (\frac{2}{m} + 2L) T3_{i,j}, \qquad (3.10.10)$$

while if it belongs to group 4,

$$F_3 = - (\frac{2}{s} + 2L) T3_{i,j} + \frac{2}{1+s} T3_{i+1,j}. \qquad (3.10.11)$$

In the above equations $m = z_j - i$ and $s = i - z_j$.

When j = 1 the arrangement of the grid points is shown in Figure 3.4.(e). In this case boundary condition (3.4.6) is used to derive

$$F_4 = 2(R^2 + R^2G-L) T2_{i,j} - 2R^2 \cdot T2_{i,j+1} - 2R^2G \cdot TJ1_i. \qquad (3.10.12)$$

Figure 3.4.(f) shows the grid points in the neighborhood of point (i,Q). Boundary condition (3.4.7) is used to arrive at

$$F_4 = -2R^2 \cdot T2_{i,j-1} + 2(R^2 + R^2G - L) T2_{i,j} - 2R^2G \cdot TJQ_i. \qquad (3.10.13)$$

The remaining possibilities arise at grid points which belong to the last two boundary group classes. Use the definition adopted in deriving equations (3.9.14) - (3.9.18) for m and s in the following equations.

When point (i,j) belongs to boundary point class 7 but not to 5, 6, or 8,

$$F_4 = - \frac{2R^2}{1+m} \quad T2_{i,j-1} + 2 \, (\frac{R^2}{m} - L) \; T2_{i,j}, \tag{3.10.14}$$

while if it belongs to class 8 but not to 5, 6 or 7,

$$F_4 \; 2 \, (\frac{R^2}{s} - L) \; T2_{i,j} - \frac{2R^2}{1+s} \quad T2_{i,j+1}, \tag{3.10.15}$$

and if it belongs to both class 7 and 8,

$$F_4 = 2 \, (\frac{R^2}{m \cdot s} - L) \; T2_{i,j}. \tag{3.10.16}$$

If the point belongs to both classes, 5 and 7, then both boundary conditions (3.4.6) and (3.4.8) are used in arriving at

$$F_4 = 2 \, (\frac{R^2}{m} + \frac{R^2G}{m} - L) \; T2_{i,j} + \frac{2R^2G}{m} \quad TJI_i. \tag{3.10.17}$$

Boundary condition (3.4.7) along with (3.4.8) is used in the event the boundary point belongs to both classes 6 and 8, and in this case

$$F_4 = 2 \, (\frac{R^2}{s} + \frac{R^2G}{s} - L) \; T2_{i,j} + \frac{2R^2G}{s} \quad TJQ_i. \tag{3.10.18}$$

This completes the list of possible boundary point arrangements. The five expressions given for $F_3$ and the eight for $F_4$ will be used to write an equation corresponding to (3.10.2) at every grid point.

Some of the equations can be summarized in tabular form if equation (3.10.2) is written as

$$A \cdot T3_{i-1,j} + B \cdot T3_{i,j} + C \cdot T3_{i+1,j} = F_4 + E. \qquad (3.10.19)$$

When the point (i,j) is not a boundary point belonging to group 7 or 8, the values of A, B, C, E and $F_4$ can be tabulated and are given in Tables 3.3 and 3.4.

Table 3.3.
Values of A, B, C, and E in equation (3.10.19).

| i | A | B | C | E |
|---|---|---|---|---|
| Does not satisfy conditions for Boundary Point Class 1, 2, 3 or 4 | 1 | BI | 1 | 0 |
| = 1 | 0 | BII | 2 | $-2 \cdot H \cdot TI1_j$ |
| = P | 2 | BII | 0 | $-2 \cdot H \cdot TIP_j$ |
| Satisfies condition for Boundary Point Class 3 | $\frac{2}{1+m}$ | $-\frac{2}{m} - 2\,L$ | 0 | 0 |
| Satisfies condition for Boundary Point Class 4 | 0 | $-\frac{2}{s} - 2\,L$ | $\frac{2}{1+s}$ | 0 |

Table 3.4.
Values of $F_4$.

| j | $F_4$ |
|---|---|
| Does not satisfy conditions for Boundary Point Class 5, 6, 7 or 8 | $- R^2 . T2_{i,j-1} + 2(R^2 - L)\ T2_{i,j} - R^2 \cdot T2_{i,j+1}$ |
| = 1 | $2(R^2 + R^2\ G - L)\ T2_{i,j} - 2\ R^2 \cdot T2_{i,j+1} -$<br>$- 2\ R^2\ G \cdot TJ1_i$ |
| = Q | $-2\ R^2 \cdot T2_{i,j-1} + 2(R^2 + R^2\ G - L)\ T2_{i,j} -$<br>$- 2\ R^2\ G \cdot TJQ_i$ |

For each value of j there are two sets of equations. One contains the equations written at points in region $R_1$, the other those in region $R_2$. The coefficients in the equation obtained at point (i,j) can all be identified by the subscript i; also the subscript j can be suppressed in the equations since it is known to be the same for the whole set. In this notation each equation of each set has the general form

$$A_i \cdot T3_{i-1} + B_i \cdot T3_i + C_i \cdot T3_{i+1} = D_i . \qquad (3.10.20)$$

## 3.11 Solution of the Equations

In the previous sections systems of equations were derived, which, when solved, give the temperature distribution at time t+$\Delta$t from information that was available at time t. The set of equations derived had the form

$$\left.\begin{array}{ll} B_{min}T_{min}+C_{min}T_{min+1} & = D_{min} \\ A_{min+1}T_{min}+B_{min+1}T_{min+1}+C_{min+1}T_{min+2} & = D_{min+1} \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ A_{max-1}T_{max-2}+B_{max-1}T_{max-1}+C_{max-1}T_{max} & = D_{max-1} \\ A_{max}T_{max-1}+B_{max}T_{max} & = D_{max} \end{array}\right\} \quad (3.11.1)$$

In the set there are max-min+1 unknown T's and there are an equal number of equations. The set is tridiagonal, that is, the matrix of the coefficients is tridiagonal. The Gauss-Seidel iteration or the Gauss-Jordan elimination methods are the ones most commonly used to solve a set of linear equations. The Gauss-Seidel iteration method may take many iterations to converge satisfactorily whenever it is convergent. If the usual algorithm is used in the Gauss-Jordan elimination method, it also requires a large amount of computation, and in addition there is the possibility that the round-off error may accumulate considerably.

Peaceman and Rachford proposed a simpler method for solving a tridiagonal set of linear equations. Their method is equivalent to the Gauss-Jordan elimination method, but it takes advantage of the special characteristics of the set of equations: its tridiagonality. Wilkes (100), Carnahan, Luther and Wilkes (17) and others state the algorithm in a slightly modified form, which is used here. Indeed, T satisfies

$$\left.\begin{aligned} &T_{max} = GM_{max} \\ &T_i = GM_i - \frac{C_i T_{i+1}}{BT_i}, \text{ with } i=max-1,\ max-2,\ \ldots,\ min, \end{aligned}\right\} \quad (3.11.2)$$

where

$$\left.\begin{aligned} &BT_{min} = B_{min} \quad \text{and} \quad GM_{min} = \frac{E_{min}}{B_{min}} \\ \text{and} \quad &BT_i = B_i - \frac{A_i C_{i-1}}{BT_{i-1}}, \\ &GM_i = \frac{E_i - A_i \cdot GM_{i-1}}{BT_i}, \text{ with } i = min+1,\ min+2,\ \ldots,\ max. \end{aligned}\right\} \quad (3.11.3)$$

If the equations (3.11.1) are set up in order of increasing indexes, it is not necessary to save all the coefficients until all the equations are set up. BT and GM can be immediately calculated, and it is sufficient to retain all the BT's, GM's and C's, that is, three pieces of information for each equation. Once this information has been collected it is quite simple to calculate all the unknowns in the set of equations using the formulas (3.11.2).

## 3.12 Motion of the Interface

In the previous two sections it was assumed that the location of the boundary is known and is defined by the location of the interface points. As a matter of fact, it was assumed to be known as a function of time. In equations (3.9.10) and (3.9.11) the interface point was the location where $TI_{z_j,j}=0$, that is, the location of the interface was known at time t. In equations (3.9.14) - (3.9.18) and equations (3.10.14) - (3.10.18) the location of the interface points (i,w) and (i,v) are assumed to be known at time t+Δt/2. In equations (3.10.10) and (3.10.11) the interface location was assumed to be known at time t+Δt. In Section 3.5 it was indicated that this information is obtained from the explicit solution of the boundary condition (3.4.9). At that section it was also indicated that the accuracy of the method could be improved by recomputing the location of the interface using the temperature distribution at time t and t+Δt as in a predictor-corrector computation. As a matter of fact, the process of recomputing the temperatures and interface location can be repeated several times until the change in either or both is less than a preselected value. This refinement was not used in the present work.

Let us first rewrite the boundary condition (3.4.9) in the following form:

$$\frac{dX}{dt} = \left[ \frac{k_1(\theta_h - \theta_m)}{\lambda\rho} \frac{\partial T_1}{\partial x} - \frac{k_2(\theta_h - \theta_m)}{\lambda\rho} \frac{\partial T_2}{\partial x} \right] \frac{1}{\alpha_1 \cos^2\gamma} \quad . \qquad (3.12.1)$$

In Section 3.3 the angle $\gamma$ was defined and illustrated in Figure 3.2. It is also possible to define the angle by the following formula:

$$\tan\gamma = \frac{d\bar{X}}{d\bar{y}} = \frac{dX}{dy} = \frac{\Delta x}{\Delta y}\frac{dz}{dj}\,, \qquad (3.12.2)$$

provided that z is a continuous function of j. Since z is known only at the interface points, tan $\gamma$ can be calculated from (3.12.2) using equation (3.8.8) as follows:

$$\tan\gamma = \frac{\Delta x}{\Delta y}\,0.5(z_{j+1} - z_{j-1}) \qquad \text{when } j \neq 1,Q \qquad (3.12.3)$$

or

$$\tan\gamma = \frac{\Delta x}{\Delta y}\,(-1.5z_1+2z_2-0.5z_3), \qquad (3.12.4)$$

and

$$\tan\gamma = \frac{\Delta x}{\Delta y}\,(0.5z_{Q-2}-2z_{Q-1}+1.5z_Q), \qquad (3.12.5)$$

when j is 1 or Q. Equation (3.8.8) is also used to approximate the derivatives appearing on the right hand side of equation (3.12.1). If the boundary point in region $R_1$ is (i,j), and the interface point on the same x grid line is $(z_j,j)$, then

$$\frac{\partial T_1}{\partial x} = \frac{s}{1+s} TI_{i-1,j} - \frac{1+s}{s} TI_{i,j}, \qquad (3.12.6)$$

where $s = z_j - i$. Using a similar procedure in region $R_2$ when the boundary point is $(i,j)$ and the interface point is $(z_j,j)$, the derivative is given by

$$\frac{\partial T_2}{\partial x} = \frac{1+m}{m} TI_{i,j} - \frac{m}{1+m} TI_{i+1,j} \qquad (3.12.7)$$

where $m = i - z_j$. At this point it should be noted that using $\tan \gamma$ in the identity

$$\frac{1}{\cos^2 \gamma} = \tan^2 \gamma + 1 \qquad (3.12.8)$$

along with equations (3.12.6) and (3.12.7) makes the evaluation of dX/dt possible. Once it is computed, the movement of the interface can be calculated from the following equation:

$$z^{(t+\Delta t)} - z^{(t)} = \frac{dX}{dt} \Delta x \, \Delta t. \qquad (3.12.9)$$

The location of the interface at $t+\Delta t/2$ is halfway between the location of the interface at time $t$ and $t+\Delta t$.

At this time it is appropriate to discuss some of the questions that may arise concerning this computation. One of the peculiarities in the above computation is that TI is used in evaluating equation (3.12.1) for the whole time interval. It is possible to imagine the same method with TI being used to evaluate (3.12.1) for the first $\Delta t/2$ time interval, then computing T2. After this has been computed T2 can be used to evaluate (3.12.1) for the second $\Delta t/2$ time interval.

Brian (15) encountered a similar situation when he considered the development of a three-dimensional ADI technique. He pointed out that in his computations - and this applies to the problem under consideration as well as to most other ADI procedures - the results are fundamentally the temperatures at time $t+\Delta t$ obtained from the data at time t. All other temperatures are intermediate values in the computation and if used in any other way the computation may become more or less equivalent to the explicit method. He also points out that in the ADI method the results are computed by solving a set of implicit equations. The results so computed will tend to be smoothed in the direction in which this last set of equations were solved. Since in the problem under consideration the boundary condition (3.4.9) required the evaluation of a derivative in the x direction, the ADI method was set up to obtain the final results by solving the implicit equations in the x direction and obtain the intermediate values by solving the implicit equations in the y direction.

Another advantage of the method is that when the rate of boundary movement is calculated the length of the time interval can be chosen in such a way that it will not lead to a boundary movement greater than some prescribed value. Although the linear problem is convergent for all values of $\Delta t$ and $\Delta x$ and the restriction on nonlinear problems is not too severe in most cases, these increments have to be small before the computational error becomes reasonably small. Usually the acceptable increment size is determined by trial computations and inspection; this is certainly one procedure which must be used to verify each computation.

There is no reason for using the same time interval throughout the computation. In certain parts of a computation the boundary conditions may be almost invariant and the problem can be solved using very large values for $\Delta t$. At other times the boundary conditions may change very rapidly and a very small value is needed for $\Delta t$. There are at least two possible ways to make sure $\Delta t$ is not too large for any given step of the computation: (a) to limit the temperature change at any point in a given time interval, (b) to limit the movement of the boundary in the time interval. Method (b) is better because it is easier to assign the limits and because it signals the need for reducing $\Delta t$ before the computation of temperatures is carried out at every point.

## 3.13 Initial Conditions

The equations needed to compute a solution to the problem described in Sections 3.2 - 3.4 at time $t+\Delta t$ and at later times have been derived assuming that the solution is known at some time $t$. When such solution is available, it can be used to start the computation and there is no special problem. The problem arises when the initial conditions are not available, but have to be computed. Before the computation of the initial conditions can be considered, however, a more precise problem statement is needed. As a matter of fact, we shall consider two possible problems which can be solved by similar methods.

In the first problem we shall set up the initial conditions for a bar which is placed in the furnace and is not moved until equilibrium is reached. The initial condition is the equilibrium temperature distribution in a bar containing two regions, which can be computed either (a) by assuming some temperature distribution which is relatively close to the expected one or (b) by first assuming that the bar is at the phase transformation temperature and it has constant properties for this part of the computation. With assumption (b), it is possible to calculate how the temperature distribution would change. After a few (5, 10, possibly 20) iterations the temperature throughout the bar will start to converge to a temperature distribution which could be observed in a bar containing a single phase with the assumed constant properties. The temperature distribution so computed can be used as assumption (a).

If an assumed temperature distribution is available or has been computed, it is possible to determine the location of the interface, that is, the extent of the individual regions. Using this as an initial condition, the transient problem can be solved with fixed boundary conditions. This computation has to be continued until both the temperature distribution and the location of the interfacial boundary reaches steady state. This result corresponds to the temperature distribution in the bar after it reached equilibrium and before it started to move. It is the initial condition of the remainder of the problem during which the boundary conditions used in the computation describe the temperature of the surroundings of the bar as it moves through the furnace.

A second problem arises when a bar containing only one phase is moved in a furnace with nonuniform temperature distribution; the problem can be solved at this time by techniques used to solve the linear heat conduction problem. Complications do not arise as long as only one phase is present in the bar. Only when the second phase starts to appear at one end of the bar do we need some method which can be used until the method described in the previous sections become applicable. There are several possibilities to consider, but none are very satisfying.

One of the first thoughts one would tend to have is to use an approximation. When the boundary point belongs to both groups 1 and 3 or to 2 and 4 at the same time, an approximation similar to the one used when the boundary point belongs to both groups 5 and 7 or to 6 and 8 might be considered. This approach is similar to the one used by Murray and Landis (76) in one dimension. It is unfortunate that the accumulated error in most cases is too great to allow one to specify the shape or even the location of the boundary with reasonable accuracy.

Another possibility is to allow the conduction problem to be used until some of the material is heated above or cooled below the transformation temperature as if there were no phase transformation to consider. As a matter of fact, this is similar to the physical phenomena of the material supercooling - i.e., cooling below the transformation temperature without the phase transformation taking place. To be sure, the amount of material allowed to "supercool" is small, but finite. At this point it is possible to neglect the latent heat associated with this small region or use a "temperature depression" type of computation to correct some of the error which would be committed by such an assumption. In either case some error will be introduced, more than desirable but not an unreasonable amount. The important thing is that there is a narrow, small region which is in a different region than the rest of the bar.

The rest of the process is the same as that used in one dimension by Churchill and Teller (20) (which was a modification of one of Murray and Landis's (76) methods). It involves the introduction of a smaller $\Delta x$ and possibly smaller $\Delta y$, at least in the region having the smaller size. It may be necessary to reduce the time interval and the spatial intervals in the other region as well. These reduced intervals have to be used until the smaller region becomes sufficiently large as compared to $\Delta x$. When this occurs, the size of $\Delta x$ can be doubled. This process can be repeated as the size of the smaller region increases until the size of the increments reaches the size desired for the computation as a whole. It is important that during this whole process there should be at least two grid points in each region on each x-grid line; that is, the process outlined in the previous sections could be used with only periodic recalculation of some of the constants. Not even this last method was needed for the present investigation.

The above method is based on the principle that one can commit a small error even in certain transient problems if the results are not needed at the time the error is committed or shortly thereafter. In examining the stability characteristics of this type of problem it was pointed out in this paper and in many previous works that the error does not increase, indeed that it decreases as time increases. For this reason a small error committed in the initial conditions will effect the results only for a short period of time, and if the initial error is small enough, its effect will be insignificant. But what happens if the initial error is large? In this case, it takes longer to cause the effect of the error to decay, but at a sufficiently later time its effect will be negligible.

This principle can also be used to set up the initial condition in the second problem. One uses an extrapolation of the boundary conditions for times previous to the desired initial condition. Then one can assume almost any usable initial condition, although the better it approximates the initial condition which would be expected at that earlier time, the better the final results will be. Using this initial condition and the extrapolated boundary conditions at time $t-n\Delta t$, one proceeds to compute the results at time $t$. After one computation with a given $n$ the computation is repeated with larger $n$'s. If the various values for $n$ are suitably chosen, the different computations converge to a solution, which is the true initial condition or the desired result at time $t$.

These considerations allow us to use the procedure outlined for setting up the initial conditions in the first-mentioned problem in the beginning of this section, to be used for setting up the initial conditions and calculating the desired results for the second problem also. It is necessary to have the bar in motion for a sufficiently long period of time before reliable results can be expected for the second problem. Some results confirming these observations will be presented in Chapter V.

# CHAPTER IV

# THE COMPUTER PROGRAM

## 4.1 Introduction

The computer program used in this study is discussed in this chapter. The problem to be considered, or rather the family of problems that may be considered, were described in the previous chapter. The complexity and variety of some of the problems were discussed and it was found to be convenient to break the whole problem up and consider only certain aspects of it at any one time. The same idea was carried over to the computer program itself, which is broken up into individual routines, each doing only certain aspects of the computation. The use of a main routine insures that the sequence of computations is correct and corresponds to the problem being solved. There is another routine for taking care of all the computer input-output functions; the reading of data and the printing and/or punching out of computed results. The other routines are used in the computation of the individual parts of the problem itself: one to compute T2, another to calculate T3, one to find the movement of the interface, others to compute the boundary conditions and other needed side-computations. There are several "control" variables that can be used to modify several of the routines.

These variables can be set either by the routines themselves or they can be read in during the computation as data.

The computer programs are written in the MAD (Michigan Algorithm Decoder) compiler language. The routines are listed in Appendix A, together with a brief statement on the role that each variable has in the program. Let us, however, say a few words about the units used in the program and define a few variables here in order to simplify the following discussions.

In Chapter III the problem was first defined in its natural form, then a new set of variables was introduced and the problem restated in dimensionless form. In obtaining the numerical approximation a third form was still needed, which referred to the system of grid points introduced. All three sets could be used for one purpose or another. The user can specify the problem in any consistent set of natural units, leaving the task of conversion to the computer. Some of the control variables, however, refer to grid points and intervals and require an understanding of the program and the function of the variables. The results are printed in their dimensionless finite difference form, not only to save some computation time but because these are easy to visualize and compare. One exception to this is time, which most people can visualize most readily in natural units.

The user specifies in his data the time increment to be used in the computation, DT, and this is not changed by the program at any time. The actual time increment used however is DT/FACTOR, where FACTOR is usually unity, but if necessary it is increased to insure that no x-interface point moves more than MXST intervals in one time increment. Time, T, is calculated by the formula T = TSTART+N·DT, where the initial values of T, TSTART and N are also to be specified by the user. In each time step N is increased by 1/FACTOR.

In the program the subscripts are variables which are placed in parenthesis. For example, in equation (3.9.9) the variable $TIP_j$ was defined, which in the program is written as TIP(J). By suitably defining the subscripted variables CB(J) it is possible to compute a single subscript for a doubly subscripted variable: for example, $TI_{i,j}$ is denoted as TI(I+CB(J)) in the computer program for reasons of efficiency. The more conventional notation of TI(I,J) is retained for doubly subscripted variables in the following discussions.

In the previous chapter the same equations were said to be valid in both subregions of the rectangle, provided the coefficients were properly evaluated. To facilitate the proper evaluation of the coefficients, the variable PHASE is defined, which is 1 in region $R_1$ and 2 in $R_2$. The definition of two sets of coefficients having the same name, but having different subscripts, 1 in Region $R_1$ and 2 in $R_2$ allows the use of the equations without change. In equation (3.9.3) for example, the coefficients L and BU have been defined. In the computer program L(1) and BU(1) are used in region $R_1$, and L(2) and BU(2) in $R_2$. It is evident that L(PHASE) and BU(PHASE) can be used in both regions.

When there are two phases present, on each x-grid line the interface point is bounded by two boundary points: the one belongs to group 3 and has coordinates (AZ1(J),J), the other belongs to group 4 and has coordinates (AZ2(J),J). Usually these are neighboring grid points, but the interface may coincide with the grid point separating them. The variable PMINI will always have the value of the smallest AZI(J) and PMAXI will be equal to the largest AZ2(J). The region where PMINI ≦ I ≦ PMAXI will be referred to as the boundary area.

Other variables will be defined during the discussion. Let us turn to the individual routines, starting the discussion with the subroutines. Once the function and method of calculation in the individual routines is defined, it will not be difficult to describe how the main routine and the program as a whole works.

## 4.2 The Routine MOVEM

This routine is used to compute the location of the interface. It uses the temperature distribution near the boundary at time T, given by TI(I,J) at the grid points, and the location of the interface given by the x-interface points, (Z(J),J). The routine first computes the rate of boundary movement of each x-grid line, MVMTRT(J) which is really $(z_j^{(t+\Delta t)} - z_j^{(t)})/\Delta t$. Following this, the routine checks if the use of the time interval DT during the computation (i.e., FACTOR=1) would result in a greater boundary movement than the allowed MXST; if necessary, FACTOR is increased until the movement on every x-grid line is less than the specified MXST. The routine then computes the coordinates of the x-interface points at time T+DT/FACTOR: (ZPREDI(J),J); and at time T+DT/(2·FACTOR): (ZHFPR(J),J). To define the neighboring boundary points we use AZI, being the largest integer smaller than Z-EPS and AZ2, being the smallest integer larger than ZPREDI+EPS.

It must be pointed out that when the boundary point comes very close to the interface point the accuracy of (3.12.6) and (3.12.7) decreases. To overcome this difficulty AZ1 and AZ2 are not used in deciding which grid point is the boundary point for these equations. If the boundary point is (I,J), I is given in region $R_1$ by the largest integer not exceeding Z-EPS1, while in region $R_2$ it is the smallest integer not smaller than Z+EPS1. EPS and EPS1 are expressed as fractions of the spatial increment Δx and are to be specified by the user.

## 4.3 The Routine T2SOLV.

The routine T2SOLV. is used to find the temperature distribution T2(I,J) at time T+DT/(2·FACTOR), that is, to set up equation (3.9.20) at every grid point and solve the resulting sets of equations. In Section 3.9 it was demonstrated that several factors decide the exact form of the equation. The flow chart illustrating the organization of the computer program is given in Figure 4.1. Since there is at least one set of equations to be solved for each value of I, these sets are established in order of increasing I. Once I has been selected the set of equations will start with J=I=MIN. When I is not in the boundary area, there is only one set of equations to set up. In this case the computer first calculates D (which stands for $F_2$ in Section 3.9), then it directly computes what was referred to in section 3.11 as BT(J), GM(J) and C(J). Since the variable C is used to denote the heat capacity in the program, A(J) is used to denote the coefficient of $T2_{j+1}$, and the program saves

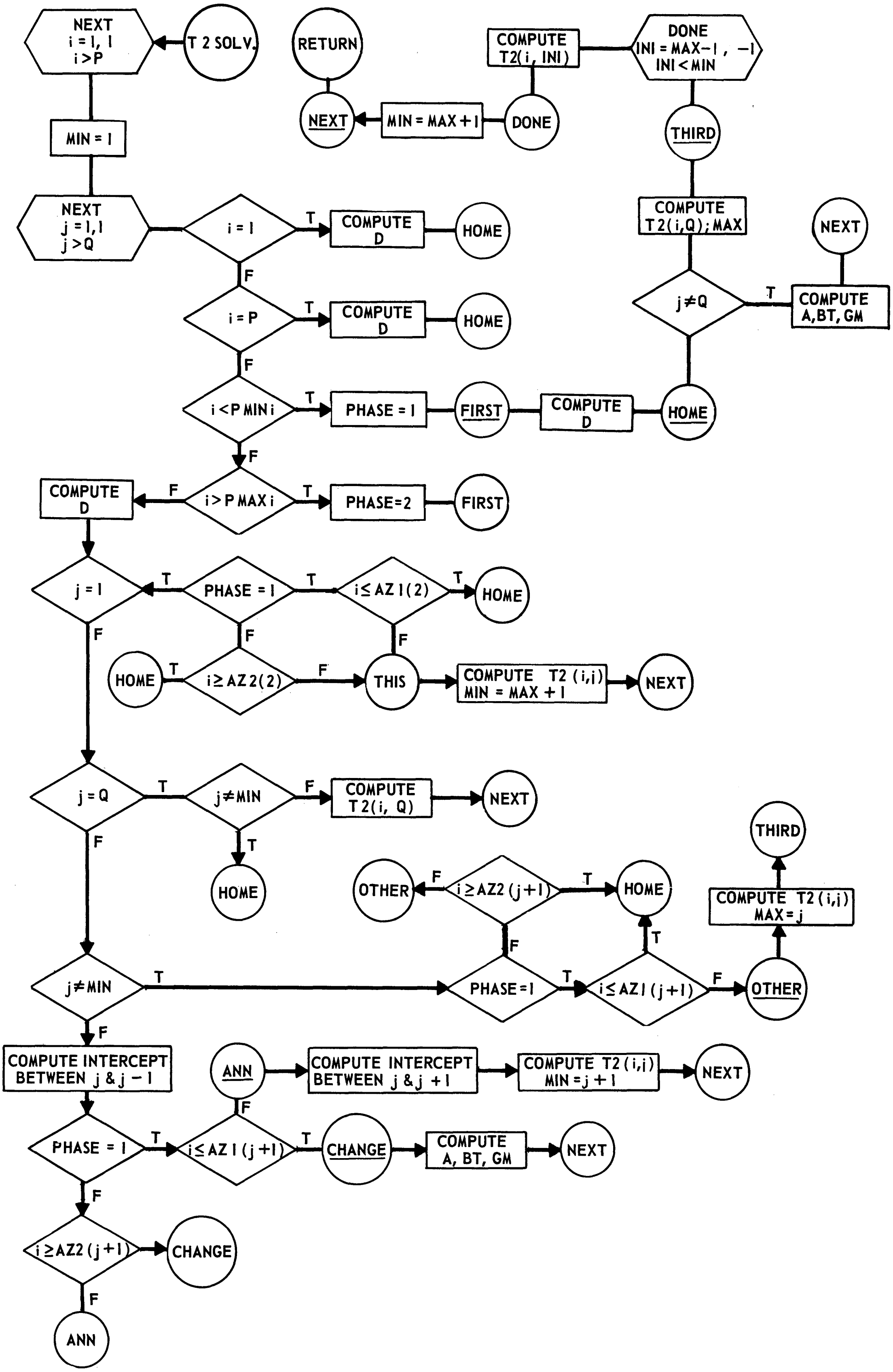

Figure 4.1.

Flow diagram of routine T2SOLV.

the computed BT(J), GM(J) and A(J) for each value of J $\geqq$ MIN. When J = Q is reached, the last equation for a given I is set up, that is, MAX = Q. The set of equations can now be solved using equations (3.11.2) to give T2(I,J) for MIN $\leqq$ J $\leqq$ MAX.

When I is in the boundary area a little more caution is needed. For each value of J it is possible to set the value of PHASE and compute D. Next the routine has to check whether the point (I,J+I) belongs to the same region as point (I,J). If the two points belong to the same region, then there will be more equations in the same set (if J$\neq$Q) and BT, GM and A is computed. If the point does not belong to the same region, then MAX = J and after computing BT and GM the set of equations can be solved. Following this a new set is started with MIN = J+I. Computing BT and GM or the coefficients at a boundary point (J = MIN or MAX) may require the determination of the location of one or two y-interface points, that is, the location where the interface "intercepts" a y-grid line. This is computed with the aid of the function ROOT., which uses the location of three boundary points in a second-order curve-fitting routine. To find the y-interface point between J=I and J=2 for example, one uses the points (I,Z(I)-I), (2,Z(2)-I) and (3,Z(3)-I). The routine is used to find that root of the second-order polynomial through these three points which is closest to I.5. (the root is Z-I=0 or Z=I). The location of the y-interface point is needed to compute s and/or m in (3.9.14) - (3.9.18). Otherwise, the computation is similar to the one described in the previous paragraph.

## 4.4 The Routine T3SOLV.

The routine T3SOLV. is used to compute the temperature distribution T3(I,J) at time T+DT/FACTOR. The equations used by the routine were discussed in Section. 3.10. Although the equations derived seemed similar, the procedure used in solving the set of equations is somewhat different. The difference arises from the difference in the direction: in computing T2 there were Q equations in the set when I was not near the boundary and in the boundary area the number of equations depended on the shape of the interface. In computing T3 the number of equations does not depend on the shape of the interface but only on whether there is an interface or not. In either case, in region $R_1$ there are Q sets of equations each starting with MIN=1 as illustrated in Figure 4.2. If there is no interface, that is, the problem is a simple conduction problem with one phase present, each set contains P equations. If two phases are present, each set in region $R_1$ contains AZ1(J) equations, and there are Q sets of equations in region $R_2$. These start with MIN=AZ2(J) and end with MAX=P. If there are grid points which coincide with the interface, the temperature at these points is zero.

Each set of equations is set up in T3COMP which is shown in Figure 4.2. For each value of I first the J dependent part of D (in Section 3.10 given by $F_4$, equations (3.10.12) - (3.10.18)) is computed. In this computation it is necessary to decide whether the grid point belongs to boundary points group 7 or 8 or both. If it belongs to either or both groups, the y-boundary points have to be located to compute m and s needed in equations (3.10.14) - (3.10.18). Once the location of the y-boundary

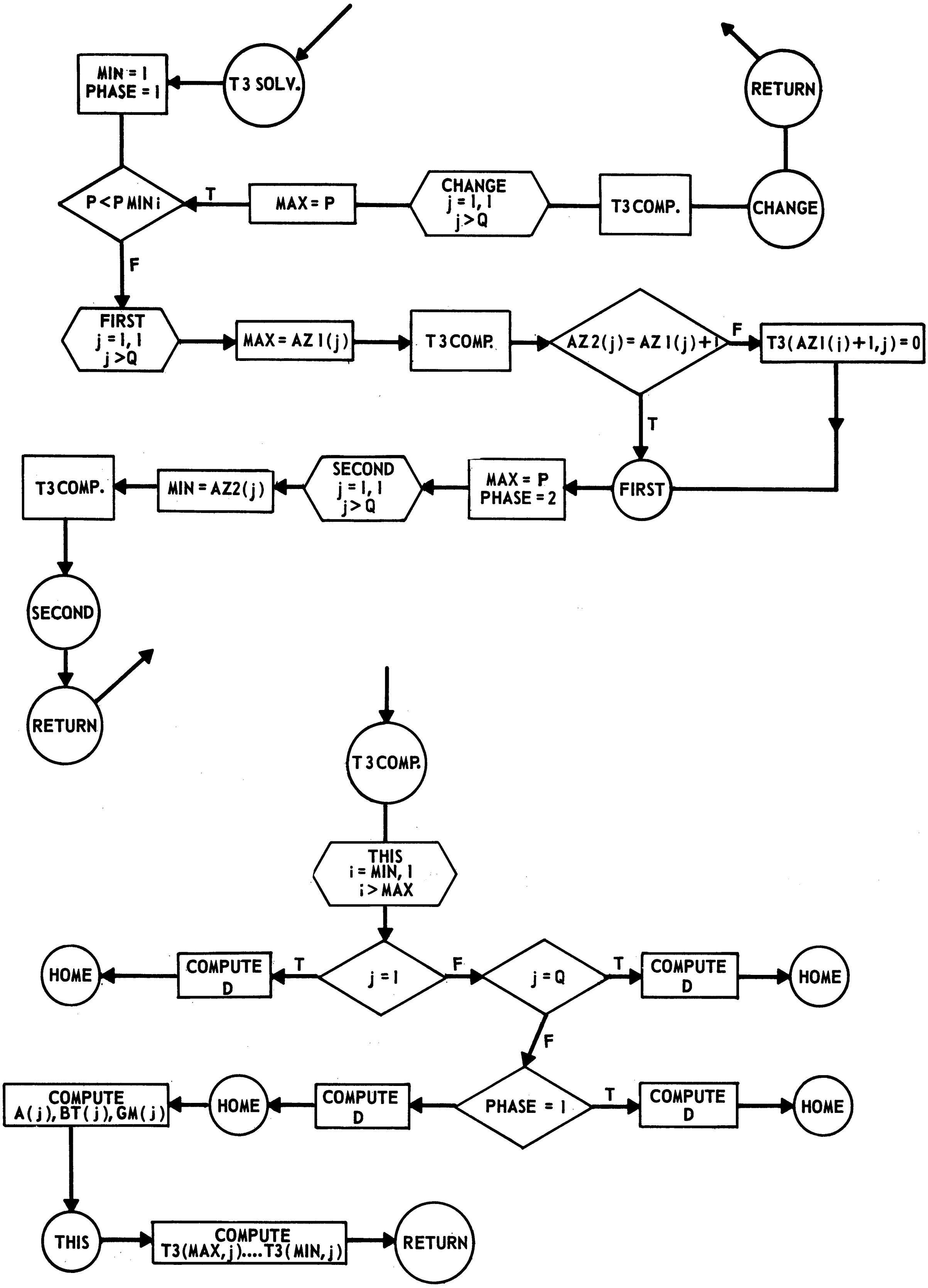

Figure 4.2.

Simplified flow diagram of routine T3SOLV.

points have been determined, it is quite simple to compute the needed A(I), BT(I) and GM(I). When these have been computed for the whole set T3(MAX,J)...T3(MIN,J) can be calculated using the formulas (3.11.2).

## 4.5 The Boundary Temperature Routines

In the previous chapter the temperature of the surroundings near the bar's surface was stated a known function of time and position, but it was left undefined otherwise. Since the temperature distribution in each problem is likely to be different, the function has to be defined and programmed for each problem or group of problems separately. This is a relatively simple procedure and the user may specify the variables needed to define the function. The routine has to be able to compute the temperature distribution of the surroundings near each external boundary point. To demonstrate this idea two functions have been written and are listed. One, referred to as BND, is a modified step-function, while the other, referred to as FIRST is a more complex function. Both are shown in Figure 4.3.

In writing the program certain assumptions were made. At the outset, before the actual computation of the problem is started, the boundary routine is entered as routine BNDSET. At this time the initial temperature distribution is set and the necessary constants are computed. The function is not used again until time T exceeds TMOVE. The routine is entered as routine BOUND. in each time-cycle after this time. In the time cycle it is entered after the routine MOVEM., that is, after the

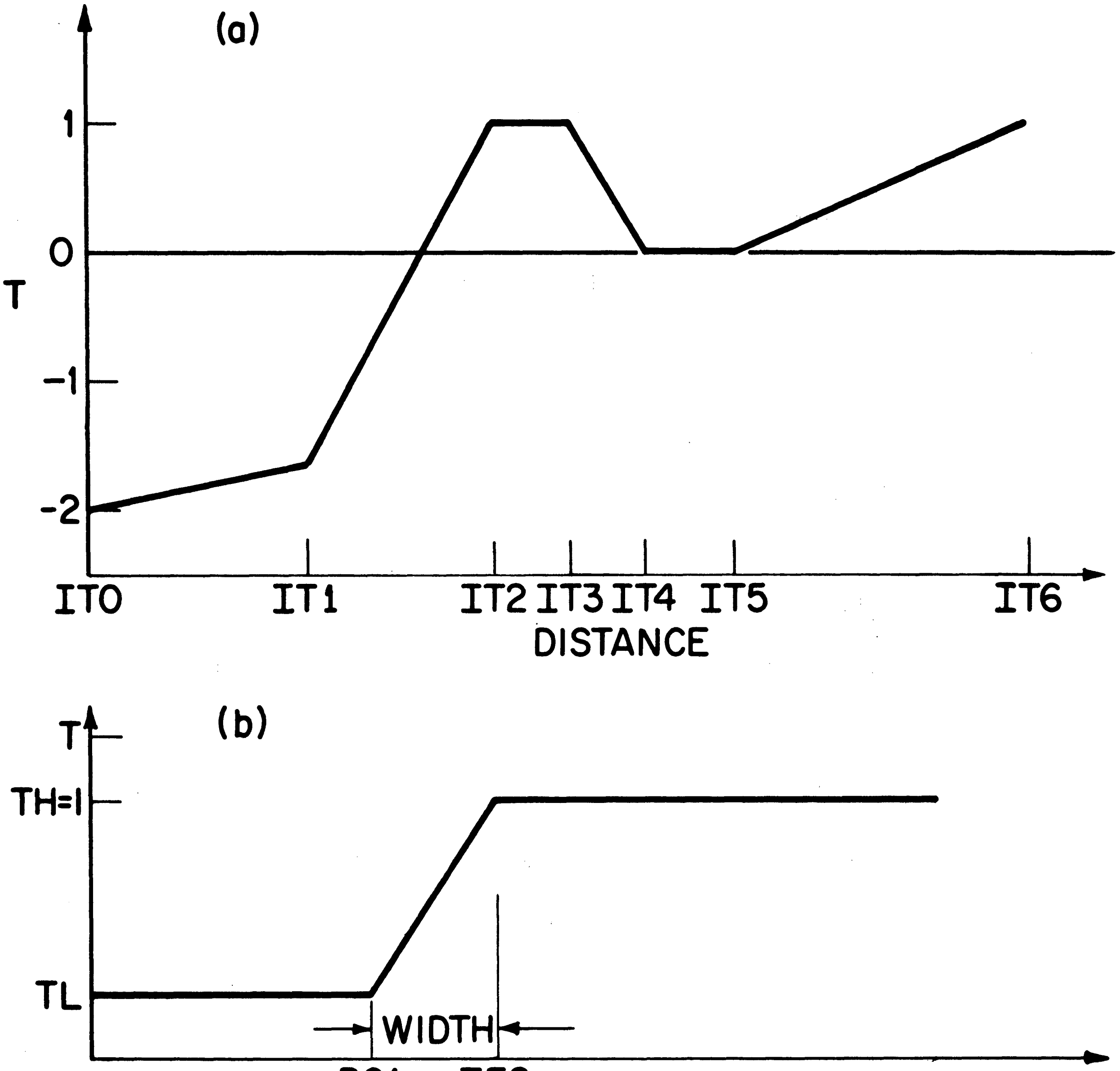

Figure 4.3.

Temperature distribution set up by the boundary routines:
(a) FIRST.; (b) BND.

location of the interface and the time increment to be used have been determined. Another assumption is that the temperature of the surroundings at the ends of the bar is the same near every grid point, that is, $TII_j$ = TII (which can be written TII(0) as well) and $TIP_j$ = TIP (or TIP(0)). These assumptions are not necessary, but to change them would require modification of certain other parts of the problem. The other assumptions, or rather characteristics are part of the function definition, and can be changed simply by defining a different function.

Before discussing the two functions separately let us point out the common characteristics. The temperature distribution on the side where J=I is the same as on the side J=Q, that is, TJI(I) = TJQ(I). The relations TII = TJI(I) = TJQ(I) and TIP = TJI(P) = TJQ(P) also hold. In both routines the temperature distribution is set up relative to a set of points whose location relative to the bar have to be determined. In both routines the location of these points varies when T exceeds TMOVE; the bar and its surroundings move relative to each other. The rate of movement is given by VELOC in Δx/unit time (i.e., intervals/sec) units.

In routine FIRST. the temperature distribution is known relative to points IT0, IT1, ..., IT5, IT6. (See Figure 4.3.) To compute the location of these points another set of points are defined by TE0, TE1, ..., TE6, which are the initial locations of the IT's and are computed relative to THSTRT = TE2 specified by the data. In each entry the relative motion since the time TMOVE is computed and the locations of the IT's are determined. In computing TJI(I) and TJQ(I) the relative location of I to the IT's determines the temperature.

In routine BND. the initial value of TE2 is THSTRT. By specifying WIDTH the location of BOI=TE2-WIDTH can be computed. The location of TE2 and BOI is defined each time the routine is entered. Knowing the position of these points relative to I allows the computation of TJI(I) and TJQ(I).

## 4.6 The Routine LOCATE.

The routine LOCATE. can be used to locate the x-interface points, the neighboring boundary points and related variables. The routine uses the temperature distribution TI(I,J) and routine ROOT. To find the location where TI(Z(J),J) would be zero. Once Z has been determined AZI, AZ2, PMINI and PMAXI are computed as in routine MOVEM. after assuming that ZPREDI=ZHFPR=Z for each J.

## 4.7 The Input - Output Routines.

The actual transmission of information to and from modern high-speed computers is very involved and most users do not actually concern themselves with more than a minimum of technical details. The program and the data is usually typed on a typewriter-like device which converts the information to punched cards or tapes or may even be directly connected with the computer. Before the computer can handle the information, however, it must be converted to the coded form used by the computer.

This process is called reading and it is carried out by a package of programs available on the computer. The reverse process has to be carried out to convert information available in the computer to the form needed by the user. Depending on the end result, this process is called printing or punching. These conversion routines are complex and require a considerable amount of information to be specified either by the programmer or automatically by the compiler. To reduce the required storage area needed for this purpose all input-output functions are carried out by one routine.

In writing a program it is desirable to give the user the opportunity to modify the computation while it is in progress by reading in new data. It is even more desirable to preserve as many of the computed results as possible to prevent the need for repetition of identical computations to find out some detail not recorded at first. It must be pointed out that the present scientific computers handle input-output information very slowly compared to their computing speeds. Handling of large amounts of information just for the record is very expensive and should be minimized. To overcome this difficulty, the program allows the user a wide latitude to specify the input-output information he needs.

Whenever new data is to be read the routine is entered as INPUT. In the data the user may specify the new values of those variables which he wants to change, all the unspecified ones remain unchanged. In each data set the user should specify the variable INTIME. When T exceeds the value of INTIME another set of data will be read in. The data read at any time are immediately printed for the record.

The main output routine is entered through functions OUT. and OUT2. Function OUT. is used when the location of the interface may be unknown. In this case, the printing of the variables related to the interface location is bypassed, while using routine OUT2. these variables may also be printed. The routine is entered at the end of the computational cycle, T already has the value of t+Δt, but T1, T2, T3, Z, ZHFPR and ZPRED1 are as calculated in this last cycle. Whether or not printing and/or punching is done, at the end of the routine T1's take on the value of the T3's, and Z's take the value of the ZPRED1's and a new computational cycle is started.

The function of the control variables INSW and SWOUT are indicated in the listing of the routine. The printing of the results are primarily controlled by the activating switches INSW(10) and INSW(11). (The INSW can have the value of 1B or 0B, which in the MAD language are the true and false logical constants respectively.)

The temperature distribution in the bar as given by the T1's, T2's and T3's may be printed when INSW(10) is 1B. The value of SWOUT(0) determines which of these temperatures is printed. At the same time it is also possible to print the variables used to describe the interface and the boundary conditions. All the Z's are printed if INSW(13) is 1B, all the AZ1's and AZ2's if INSW(17) is 1B, all the TJ1's if INSW(20) is 1B, all the TJQ's if INSW(23) is 1B, and all the ZPRED1's and ZDEVIA's if INSW(15) is 1B. The value of ZDEVIA(J) = ZPRED1(J) - ZPRED1(1) is computed immediately before printing and is a measure of the curvature of the interface.

When INSW(11) is 1B, the temperatures may be printed at selected grid points. The temperature to be printed is controlled by the value of SWOUT(1). The grid points are (DEX(2J-1), DEX(2J)) where J=1, 2, ..., DEXMAX. At the same time it is possible to print the boundary temperatures in a selected interval: the TJ1(J)'s if INSW(19) is 1B, the TJQ(J)'s if INSW(22) is 1B. Here J=TJMIN, TJMIN+1, ..., TJMAX. In addition, it is possible to print the other variables connected with the boundary conditions as indicated in the previous paragraph but using different control variables. In this case the printing of the Z's is controlled by INSW(12), the ZPRED1's and ZDEVIA's by INSW(14), the AZ1's and AZ2's by INSW(16), the TJ1's by INSW(18) and TJQ's by INSW(21).

INSW(10) is turned on by the routine itself whenever T exceeds TDUMP, at this time, TDUMP is also incremented by DTDMP. Similarly INSW(11) is turned on when T > TPOINT, and then TPOINT is incremented by DTPNT. This allows the user to receive one kind of output information at DTDMP time intervals and another kind at DTPNT time intervals. The user may change any information the program has in any computational cycle by reading a new set of data, thus he can change these control variables at his discretion to get the information he deems important.

The temperature distribution and interface boundary information needed to start a computation with the temperature distribution and interface conditions existing at time T may also be punched out for later use. This is accomplished by setting INSW(5) or INSW(24) to 0B. It will be pointed out that INSW(5) is set by the main program in certain instances. When INSW(5)=0B, the computation is terminated with the completion of punching.

## 4.8 The Routines SETCF. and RESTCF.

The routine SETCF. is used to calculate the values of certain constants which depend on the variables used to describe the problem. The routine is used when the problem is initialized or when the user changes a significant variable during the computation.

The routine RESTCF. recalculates those constants which depend on the value of $\Delta t$, and is used whenever the program itself reduces or increases the value of $\Delta t$ to be used for the computation.

## 4.9 The Routine ROOT.

The routine ROOT. is used to compute the coefficients of the function $Y=A\cdot X^2+B\cdot X+C$, and to find one of the roots of this function. The function sought passes through the points (XI,YI), (X2,Y2) and (X3,Y3). The routine finds that root of the equation $Y=0$ which is closer to point X4. If only the coefficients are needed, the routine may be entered as COEF.

## 4.10 The Main Routine

The calculations which are carried out in the various subroutines were described in the previous sections. These routines can solve several problems if the proper constants are set and the routines are called in a certain sequence. The function of the main routine is to keep time and call the subroutines in the order needed to solve the problem the user wants to solve. A simplified flow diagram of the routine is given in Figure 4.4.

The computation starts when the computer's executive program turns the control of the computer over to the user's program for a limited time period. If the computation requires less time than allotted, the program returns the control of the computer to this supervisory program at the end of the computation. If the computation is not finished in the allotted time, the supervisory program interrupts the computation and takes control of the computer. When this happens some information may be lost, therefore, it is desirable to preserve the data necessary to restart the computation at a later time before such interruption occurs. This program is set up to interrupt itself 20 seconds before the supervisory program interrupts, take some action and resume the computation where it was interrupted. The action taken depends on the control variable INSW(4): if it is IB it is set to OB along with INSW(5), if it is already OB, INSW(5) remains unchanged. This allows the user to take advantage of this feature or ignore it: if used the input-output routine will act upon the information when received.

After the interrupt routines and some constants have been set up, the routine calls INPUT. to read in the first set of data. These data usually contain the material constants and those variables necessary to compute some of the constants used in the calculations and needed to set up the initial boundary conditions. Unless the initial temperature distribution is available (INSW(2)=OB) it is assumed to be zero at every

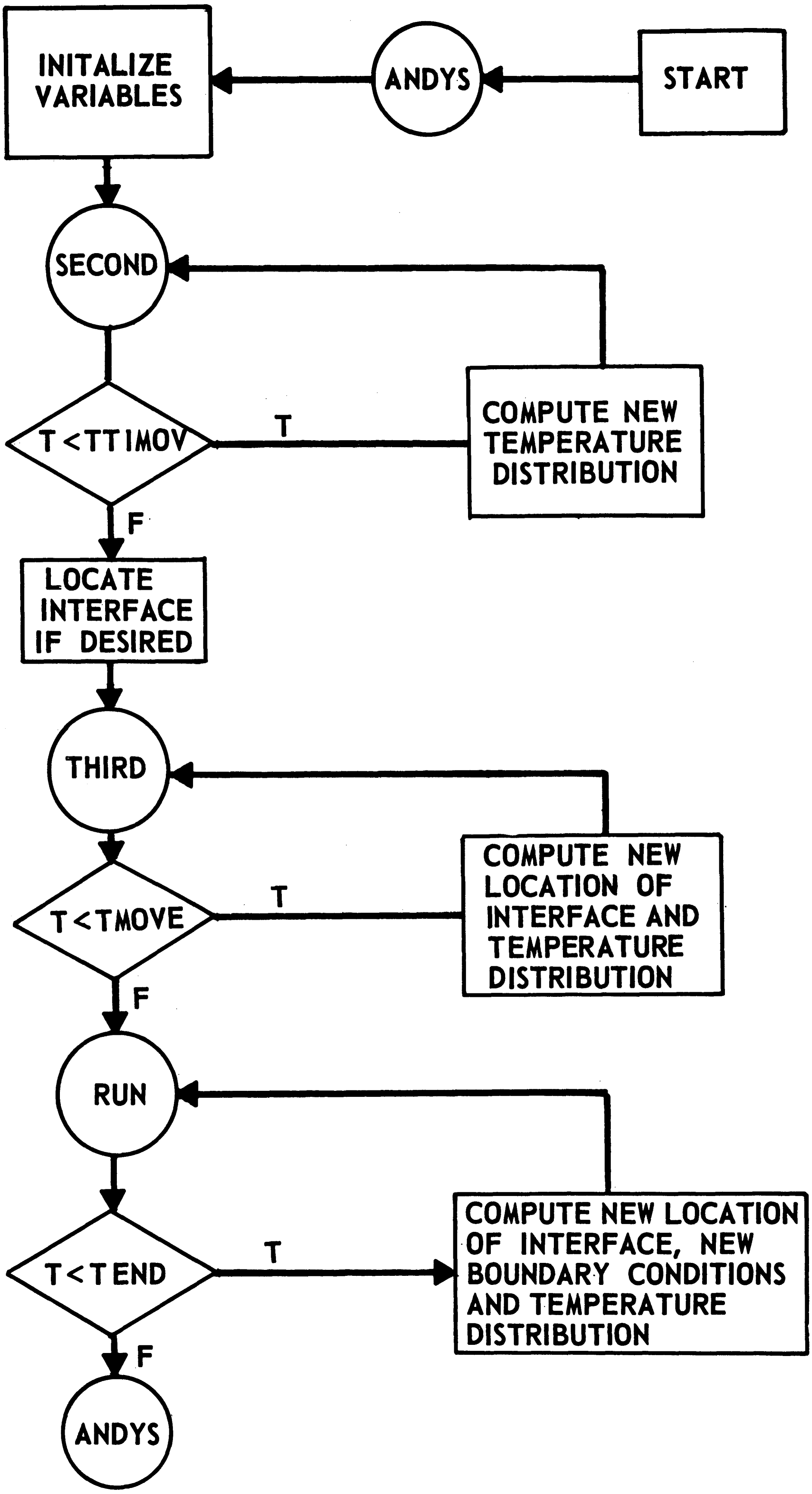
START
ANDYS
INITALIZE VARIABLES
SECOND
T <TTIMOV
T
F
COMPUTE NEW TEMPERATURE DISTRIBUTION
LOCATE INTERFACE IF DESIRED
THIRD
T<TMOVE
T
F
COMPUTE NEW LOCATION OF INTERFACE AND TEMPERATURE DISTRIBUTION
RUN
T<TEND
T
F
COMPUTE NEW LOCATION OF INTERFACE, NEW BOUNDARY CONDITIONS AND TEMPERATURE DISTRIBUTION
ANDYS

Figure 4.4.

Simplified flow diagram of the main routine.

grid point and the region is assumed to contain only phase I. Two typical data sets are given and explained in Appendix B. Once the initial conditions have been set up the main computation can begin.

The computation of the problem is carried out in steps. In each step or cycle of computation the time T is increased by increasing the variable N by 1/FACTOR. If T > INTIME, the user has data to be read at this time and INPUT. is called. If INSW(3) is 1B, some constants may need recalculation by SETCF. and in this case the new constants are also printed by OUT1. The rest of the computation in the step is carried out in one of three modes:

1. This mode is used as long as T < TTIMOV. In this mode, only the temperature distribution is computed. The boundary conditions are not functions of time. This mode can be used to compute pure heat conduction problems. When the condition T ≥ TTIMOV is met, routine LOCATE. is usually used to find the location of the interfacial boundary. This may be bypassed by setting INSW(6)=0B. The computation then continues in the next mode.

2. This mode is used while T < TMOVE. The external boundary conditions are fixed but the interface may move. This mode can be used to compute the steady state solution in a bar containing two phases.

3. The computation in the last mode is carried out until the condition T < TEND is no longer met. In this mode all the boundary

conditions are functions of time. When the computation of the problem is finished, a new problem may be started or the computation may be terminated.

## 4.11 Final Remarks

In writing the program several decisions had to be made. One of the first was whether a self-contained program requiring very little or no external information should be written in preference to a more general one. A self-contained program is restricted by its nature while a general program is restricted to the specific task by the data it receives at the time of execution. In the preceding sections it was shown that the program is fairly versatile and is capable of solving several types of problems. An understanding of the actual operation of the program is required if one is to be able to specify the required data. In Appendix B two sets of data are listed and explained in detail.

A person acquainted with programming and the MAD compiler can discover that there are places where the program could have been written more efficiently: a shorter program is possible, a faster program is possible and a clearer program is also possible. The program is, however, reasonably refined to fulfill its purpose: to demonstrate the method of computation and the principles involved. Its correctness and efficiency will be demonstrated in the next chapter where the results of the sample computations are discussed.

## CHAPTER V

## THE COMPUTED RESULTS

### 5.1 Introduction

In this chapter some of the computed results are presented. To show the correctness and validity of the calculations several problems had to be solved. The use of a very small bar and the modified step-function boundary temperature distribution assisted in developing and evaluating the computer program while using minimum amounts of machine time. When these preliminary computations confirmed the correctness of the program, realistic dimensions and conditions were adopted for those computations which are presented in this chapter.

At this time a word may be said about the computer time required to solve a problem. No statistical analysis was carried out to determine the amount of computation needed for the various parts of the program, but an idea may be gained by observing the time required to solve some of the sample calculations on the IBM 7090 computer. A problem, containing 45x5 grid points, required 280 seconds for 600 iterations when the temperature distribution was printed in every 20 iterations, and it required 480 seconds for 1200 iterations when the temperature distribution was printed in every 100 iterations.

During the solution of the problem, a large amount of information is obtained. Depending on the user's need most of this may be intermediate results of little or no interest and in such case it is possible to record the final result only. In this investigation the shape and location of the interface as a function of time is of prime interest.

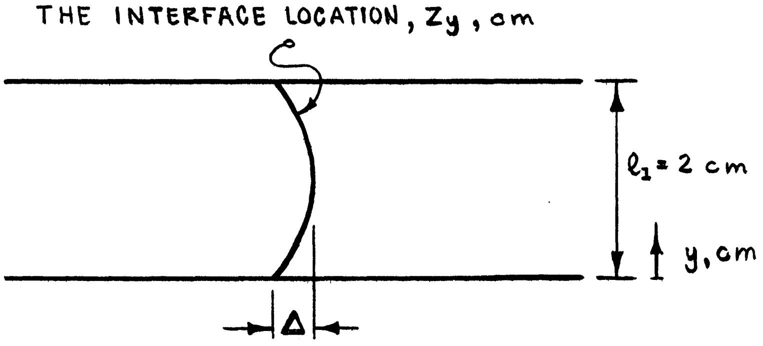

Figure 5.1.

Definition of Δ.

The location of the interface is presented as the distance from the cold end of the bar to the interface at the center of the bar. The shape of the interface is presented with the aid of a new variable, Δ, which is defined in Figure 5.1 as the difference between the X-coordinate of the interface at the center and at the side of the bar. Plotting these two variables as a function of time allows the presentation of a large amount of information in a compact but perhaps abbreviated form.

The problems solved are illustrated in Figures 5.2 and 5.3. A bar, 22cm long and 2 cm wide, is placed in the furnace. The temperature distribution in the furnace is shown in Figure 5.2. Since the temperature distribution in the bar is not known and the location of the interface

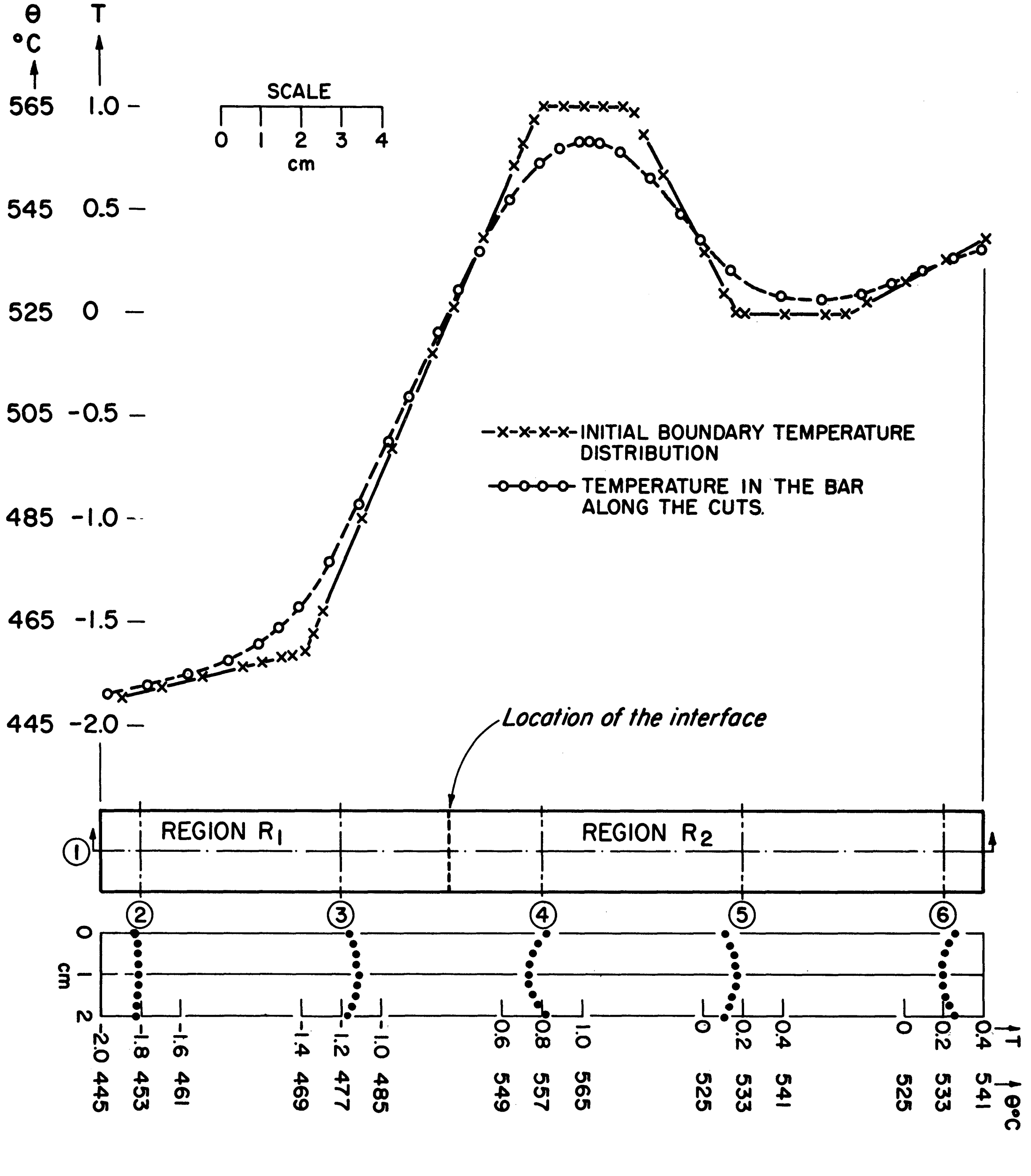

Figure 5.2.

The temperature distribution in the bar (run II, steady state).

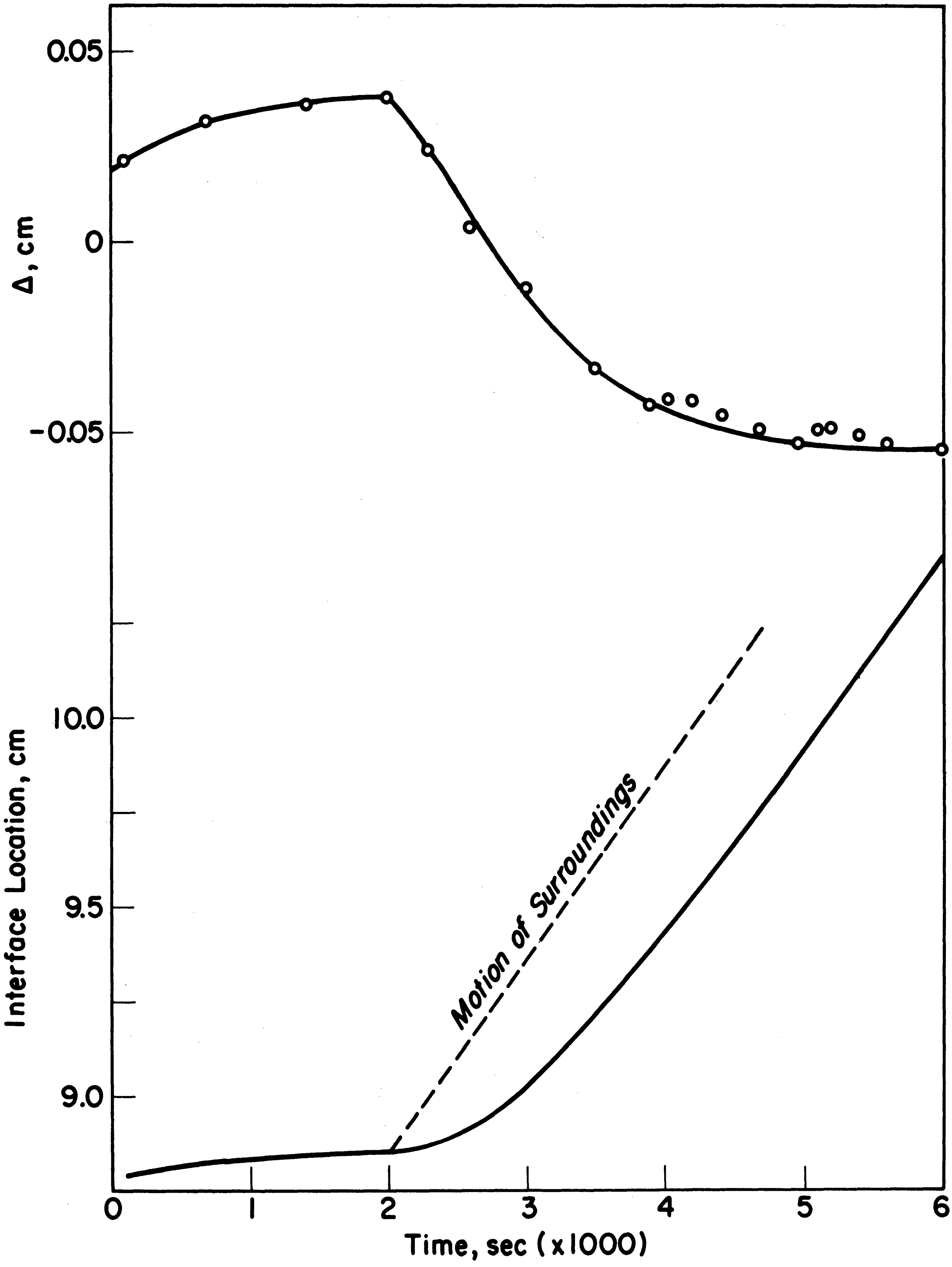

Figure 5.3.

Location of the interface and Δ during the computation (run 1).

is equally undetermined, the temperature of the bar is assumed to be uniform at the transformation temperature. It is further assumed that the properties throughout the bar are the same as in region $R_I$. After calculating the temperature distribution that would exist under these assumptions 50 seconds later the location of the interface is determined. The following computation finds how the temperature distribution, interface location and shape would change in a bar placed in the same furnace if the initial conditions calculated in the previous step were to exist. As the temperature of the bar's surroundings do not change with time during this phase of the computation a steady state is approached. The temperature along selected grid lines of the rectangle is presented in Figure 5.2 at a time when the steady state conditions are almost reached. After steady state, the bar's surroundings begin to move relative to the bar. The position and the shape of the interface are plotted in Figure 5.3, in which the whole computation can be followed. As the bar approaches steady state the interface location and shape does the same; when the bar begins to move, the interface starts to move also. First the motion is slow, but it accelerates until its speed equals that of the surroundings. The shape of the interface starts to change when the bar starts to move, but after an initial change the shape does not seem to vary with time. This can be explained by noting that the shape of the interface is a function of the temperature distribution in the bar near the interface, which in turn is a function only of the external temperature distribution and the velocity of the bar as long as this interface is far from the ends. This is certainly true when the interface is near the center of the bar as in these computations.

In the following section the effect of the various furnace conditions (heat transfer coefficients, velocity) and material properties will be shown. In Section 5.3 the validity of the computations will be discussed.

## 5.2 The Computed Results

The problem previously outlined was solved assuming a number of material properties and external conditions. These are summarized in Table 5.1. The properties of phase 1 and the boundary temperature distribution is the same in all the examples.

TABLE 5.1

SUMMARY OF THE DATA

| Run Number | $\lambda$ cal/gm | $10^4 V$ cm/sec | $k_1/H_1$ cm | $k_2/H_2$ cm | $k_2$ cal/cm·sec.°C | $\alpha_2$ cm$^2$/sec |
|---|---|---|---|---|---|---|
| 1 | 50 | 5.15 | 2.25 | 2.5 | 0.020 | 0.05 |
| 2 | 25 | 5.15 | 2.25 | 2.5 | 0.020 | 0.05 |
| 3 | 100 | 5.15 | 2.25 | 2.5 | 0.020 | 0.05 |
| 4 | 100 | 2.575 | 2.25 | 2.5 | 0.020 | 0.05 |
| 5 | 50 | 2.575 | 2.25 | 2.5 | 0.020 | 0.05 |
| 6 | 50 | 5.15 | 2.25 | 1.905 | 0.020 | 0.05 |
| 7 | 50 | 5.15 | 0.75 | 1.905 | 0.020 | 0.05 |
| 8 | 50 | 5.15 | 0.75 | 0.833 | 0.020 | 0.05 |
| 9 | 50 | 5.15 | 9.0 | 10.0 | 0.020 | 0.05 |
| 10 | 50 | 5.15 | 9.0 | 6.0 | 0.020 | 0.05 |
| 11 | 50 | 5.15 | 2.25 | 1.905 | 0.020 | 0.06 |
| 12 | 50 | 5.15 | 2.25 | 1.905 | 0.025 | 0.05 |
| 13 | 50 | 5.15 | 2.25 | 3.125 | 0.025 | 0.0625 |
| 14 | 50 | 5.15 | 2.25 | 2.19 | 0.023 | 0.0575 |

In all runs $k_1$ = 0.018 cal/cm.sec.°C

$\alpha_1$ = 0.056425 cm$^2$/sec

$\rho$ = 5.8 gm/cc

As the bar moves relative to its surroundings so does the equilibrium position of the interface. The interface will lag behind this equilibrium position, more or less, depending on the magnitude of the velocity and the latent heat. The larger the lag, the more the interface shape will differ from its equilibrium shape. This is shown in Figure 5.4. The pertinent data used for these computations are summarized in Table 5.2. It is interesting to note that the pseudo-steady state interface shape seems to be the same whenever the product λV, the rate of of heat liberation, is the same. This can be seen by comparing the results of runs 1 and 4 or 2 and 5.

TABLE 5.2

DATA: EFFECT OF LATENT HEAT AND VELOCITY

| Run No. | λ, cal/gm<br>Latent Heat | V,cm/sec<br>Velocity | λρV<br>cal/cm$^2$·sec |
|---|---|---|---|
| 1 | 50 | $5.15 \times 10^{-4}$ | $2.575 \times 10^{-2}$ |
| 2 | 25 | $5.15 \times 10^{-4}$ | $1.2875 \times 10^{-2}$ |
| 3 | 100 | $5.15 \times 10^{-4}$ | $5.15 \times 10^{-2}$ |
| 4 | 100 | $2.575 \times 10^{-4}$ | $2.575 \times 10^{-2}$ |
| 5 | 50 | $2.575 \times 10^{-4}$ | $1.2875 \times 10^{-2}$ |

The effect of the heat transfer coefficients was investigated next. The pertinent data are summarized in Table 5.3 and the results are shown in Figure 5.5. In these and subsequent figures only the transient part of the results are shown; the bar starts to move relative to its

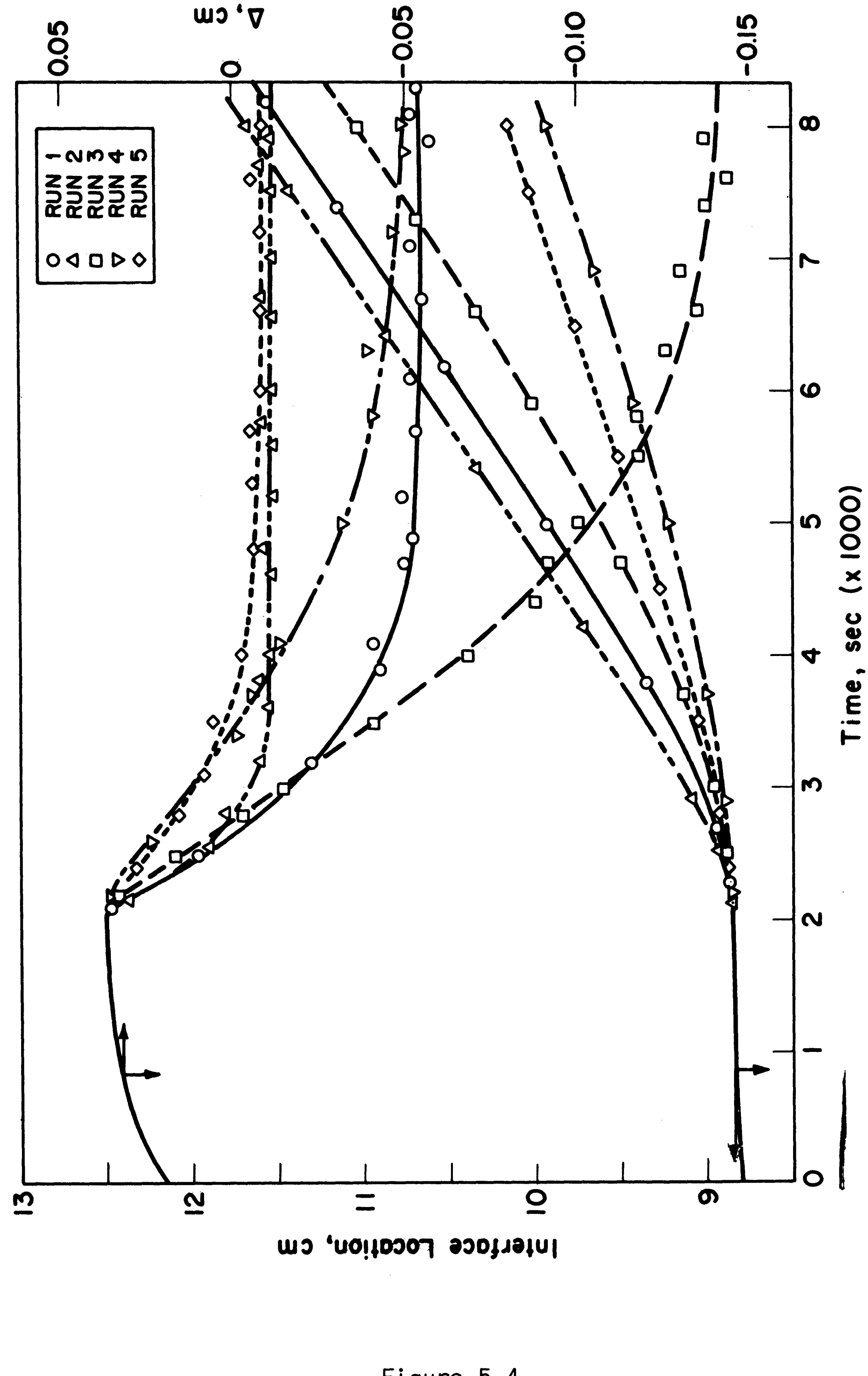

Figure 5.4.

Effect of latent heat and velocity.

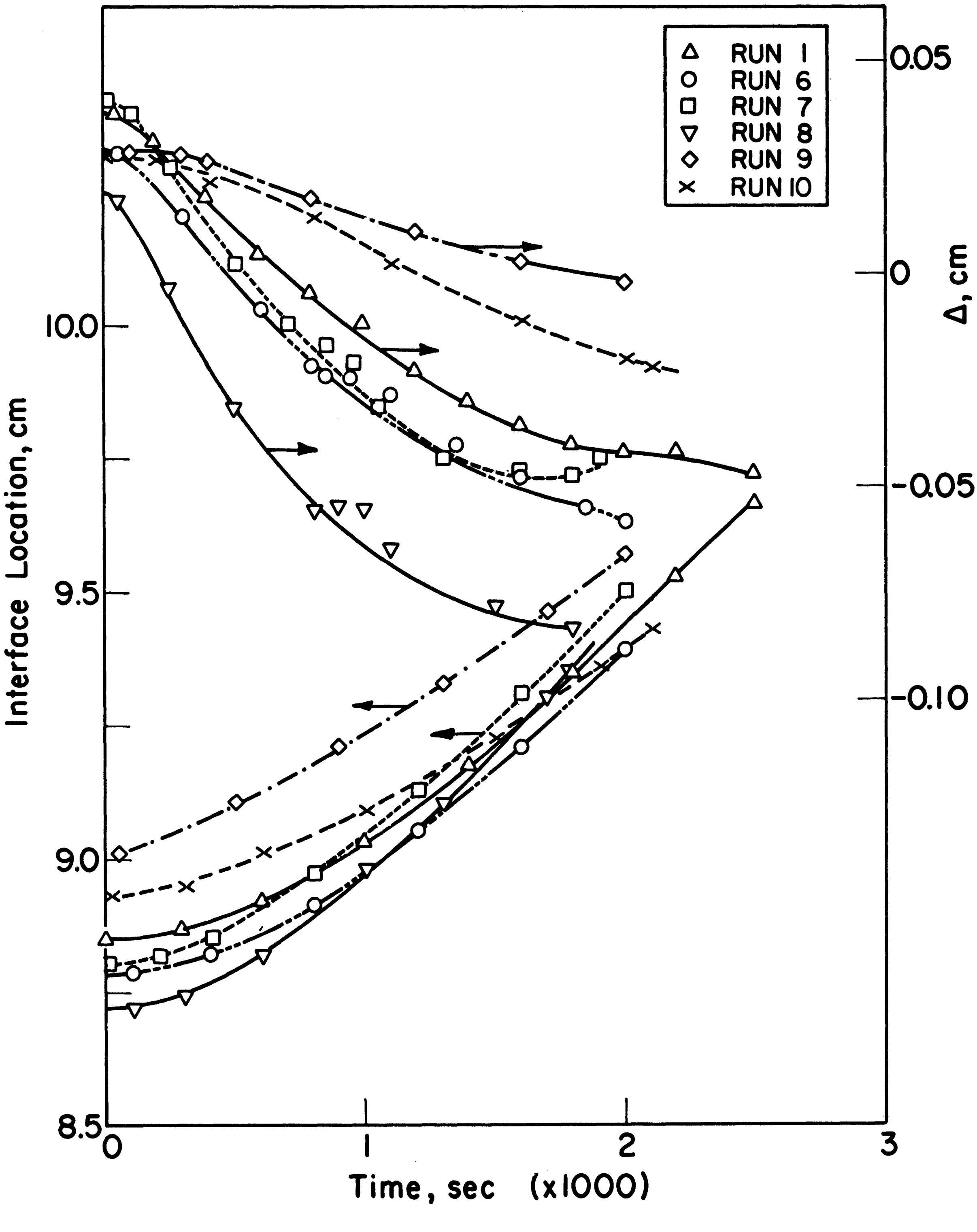

Figure 5.5.

Effect of the heat transfer coefficients.

TABLE 5.3

DATA: EFFECT OF THE HEAT TRANSFER COEFFICIENTS

| Run Number | $k_1/H_1$ cm | $k_2/H_2$ cm | $H_1/H_2$ |
|---|---|---|---|
| 1 | 2.25 | 2.5 | 1.0 |
| 6 | 2.25 | 1.905 | 0.76 |
| 7 | 0.75 | 1.905 | 2.3 |
| 8 | 0.75 | 0.833 | 1.0 |
| 9 | 9.0 | 10.0 | 1.0 |
| 10 | 9.0 | 6.0 | 0.66 |

surroundings at TMOVE = 1000. In some sets of data, particularly in runs 9 and 10, this time was not sufficient for the bar to reach equilibrium. In the figures, time is zero when the bar starts to move.

An increase in $H_1$ tends to move the interface towards the hot end of the bar (i.e., towards $R_2$) and to increase under equilibrium conditions. The reverse is true when $H_2$ is increased: Δ tends to decrease and the interface moves toward the colder end of the bar. The increase response to the change can be noticed for higher values of H in general: runs 8 and 9 best demonstrate this.

The effect of the properties of phase 2 is shown in Figure 5.6 with the pertinent data summarized in Table 5.4. Notice first the practical identity of the results obtained in runs 6 and 11. This tends to indicate that thermal diffusivity, or at least the heat capacity portion of it, does not have an effect on the results. It must be pointed out that this

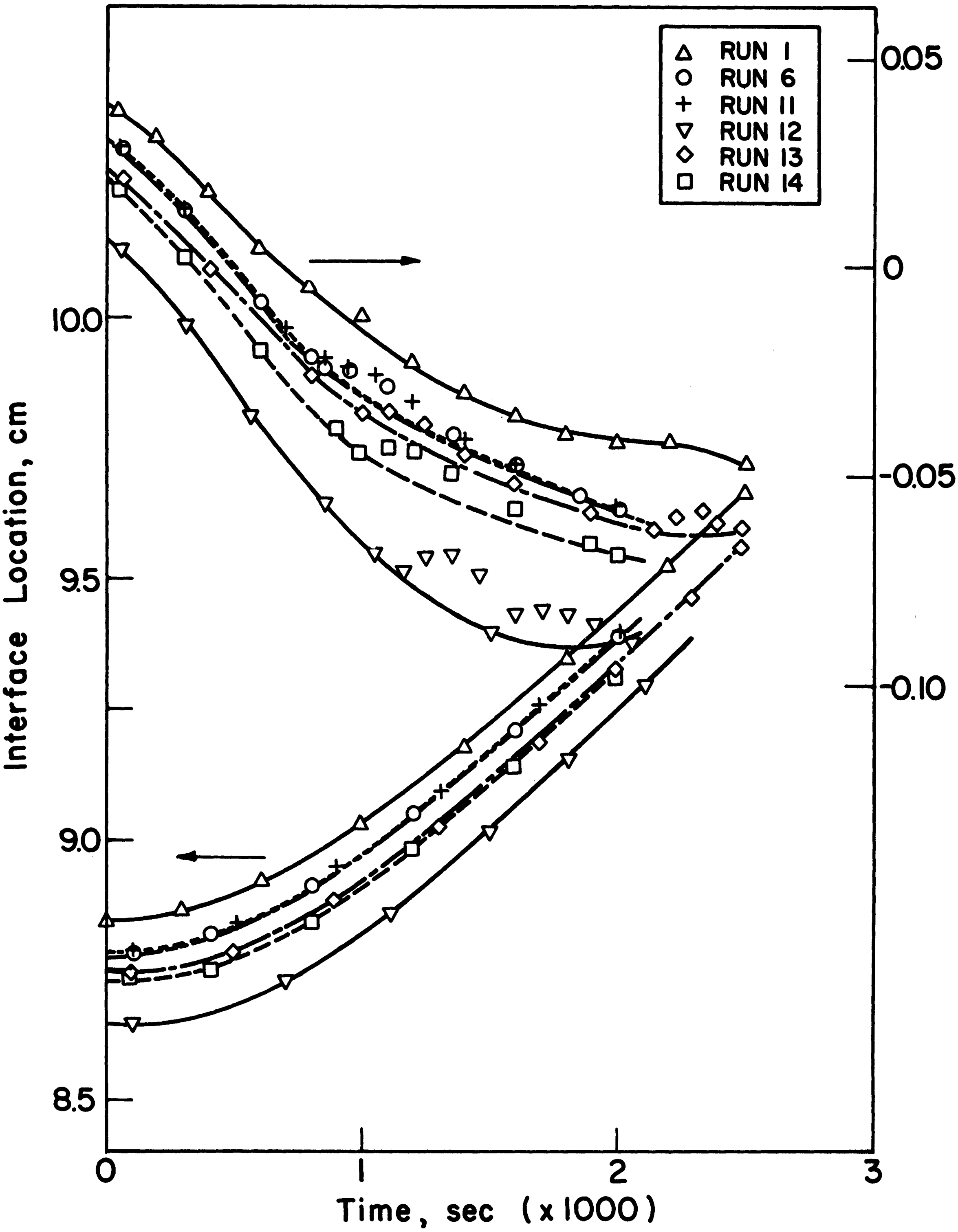

Figure 5.6.

Effect of the material properties.

TABLE 5.4.

DATA: EFFECT OF THE MATERIAL PROPERTIES

| Run Number | $k_2$ cal/cm·sec·°C | $\alpha_2$ cm$^2$/sec | $k_2/H_2$ cm |
|---|---|---|---|
| 1 | 0.020 | 0.050 | 2.50 |
| 6 | 0.020 | 0.050 | 1.905 |
| 11 | 0.020 | 0.060 | 1.905 |
| 12 | 0.025 | 0.050 | 1.905 |
| 13 | 0.025 | 0.0625 | 3.125 |
| 14 | 0.023 | 0.0575 | 2.19 |

conclusion can be drawn only in this particular case; when the variables have different values, this may not be true. The effect of changing the thermal conductivity without changing any other variable can be seen when the results of runs 1 and 13 or 6 and 14 are compared: the interface moves toward the cold end of the bar and $\Delta$ decreases as $k_2$ is increased. This effect is even more pronounced when the other properties of phase 2 increase also, while holding $\alpha_2$ and $k_2/H_2$ constant; this is seen when the results of runs 6 and 12 are compared.

Most of these effects are not new; they only confirm and clearly, quantitatively demonstrate what was known previously from qualitative reasoning. This coincidence of results along with the evidence shown in the next section points toward the usefulness of this type of computation. The limitations and needed improvements of this computation are mentioned in several other places.

## 5.3 The Validity of the Computations

In Chapter III it was pointed out that the convergence of the method remains to be proved in full by computational techniques. First, the effect of the time increment was investigated. The data used in runs 15, 16 and 17 are given in Table 5.5 and the computed results are shown in Figure 5.7. It must be stated that in run 15, the program reduced Δt in a few places and used a time interval of only 25 for 6 iterations. Even this is a large time increment, and the results approximate the results obtained with lower time increments with a surprising degree of accuracy. At lower time increments the results converge: this can be seen by inspecting the results.

TABLE 5.5

DATA: EFFECT OF Δt

| Run Number | Δt sec | $\Delta t/\Delta x^2$ sec/cm$^2$ | Dimensionless $\Delta t/\Delta x^2$ |
|---|---|---|---|
| 15 | 50.0 | 200 | 11.28 |
| 16 | 12.5 | 50 | 2.82 |
| 17 | 5.0 | 20 | 1.128 |

In all of these runs the rest of the data is the same as in Run #1.

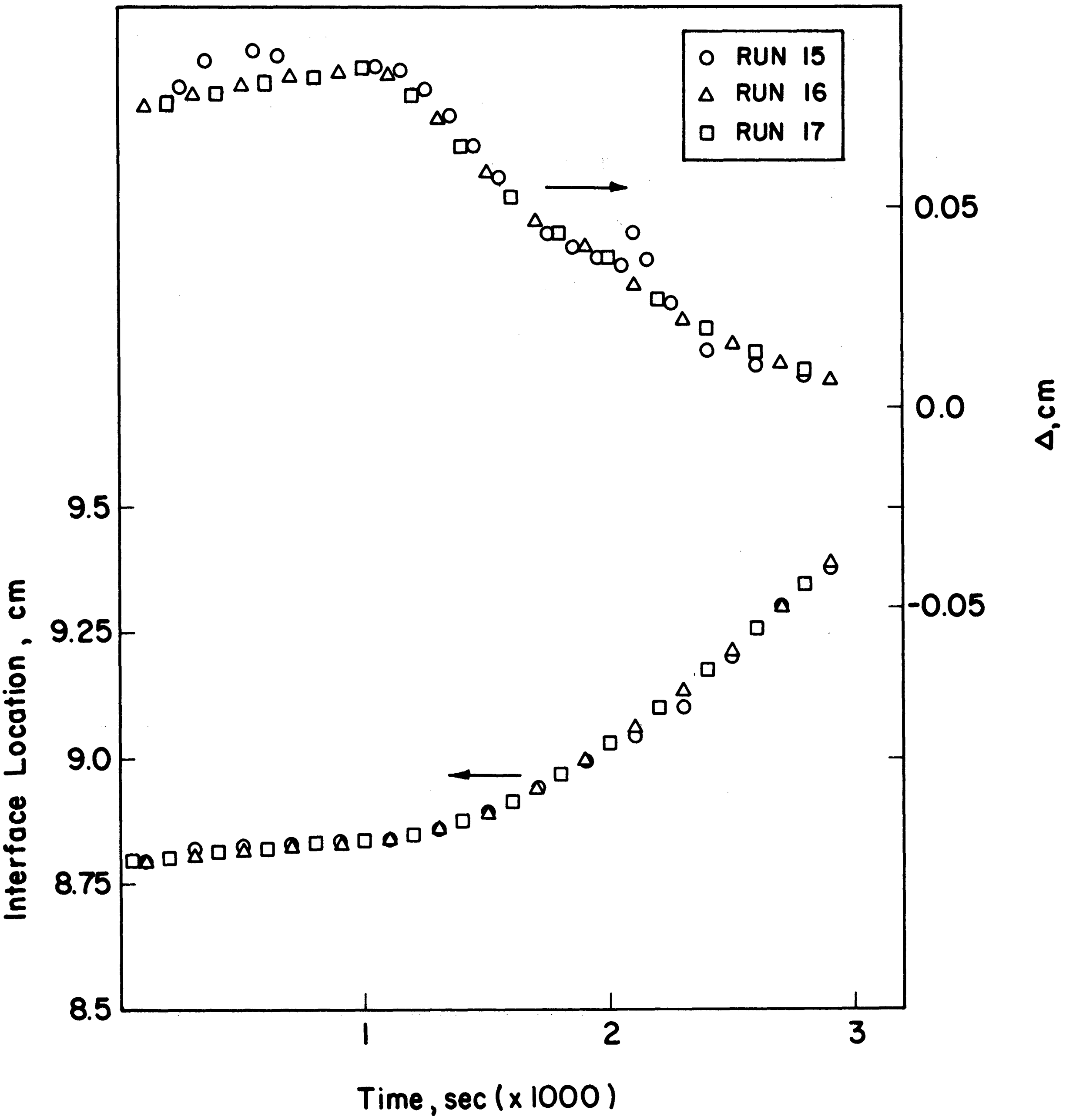

Figure 5.7.

Effect of Δt.

The next question is whether the spatial increments are small enough to represent the results accurately enough. This was tested by making several runs with different values of Δx and Δy; the data are given in Table 5.6 and the results are shown in Figure 5.8. The physical constants used for run 8 were chosen because the interface responded rapidly to changes in the boundary condition; this required a more accurate representation of the results. If the results are accurately represented for this run, as they seem to be, there is every reason to believe that all the runs are accurately represented.

TABLE 5.6

DATA: EFFECT OF Δx AND Δy.

| Run Number | P | Q | Δx cm | Δy cm | Δt sec |
|---|---|---|---|---|---|
| 8 | 45 | 5 | 0.5 | 0.5 | 5.0 |
| 18 | 45 | 9 | 0.5 | 0.25 | 2.5 |
| 19 | 45 | 13 | 0.5 | 0.125 | 2.0 |
| 20 | 89 | 5 | 0.25 | 0.5 | 2.5 |
| 21 | 89 | 9 | 0.25 | 0.25 | 2.5 |

The rest of the data is identical with that of Run #8.

In the previous chapters it was shown that when the grid points are not placed at equal intervals, lower accuracy approximations have to be used. The accuracy of these approximations is even further compromised when the location of the grid points is not known accurately.

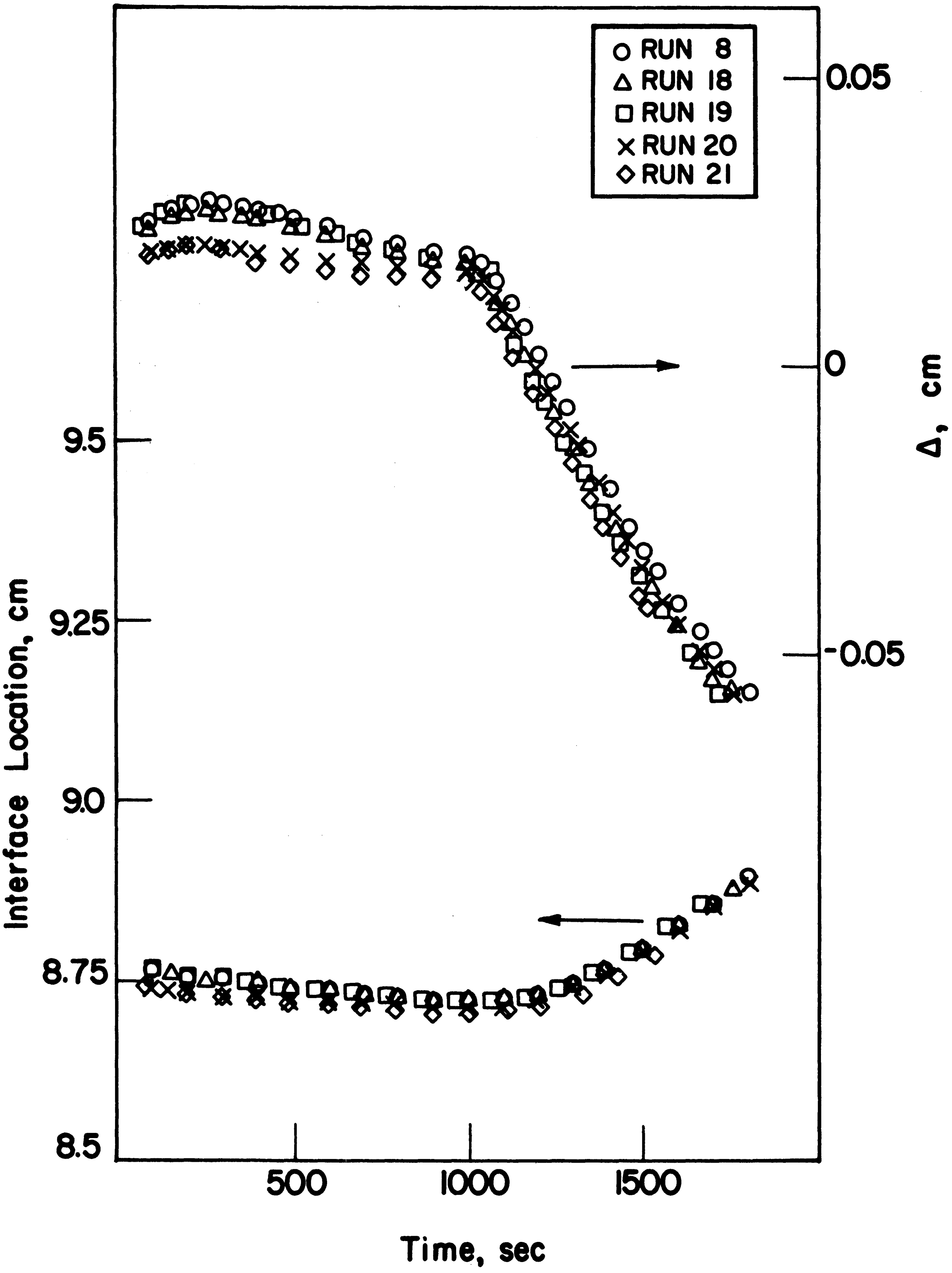

Figure 5.8.

Effect of Δx and Δy.

This is the reason for the deviations which occur whenever the interfacial boundary is near a y-grid line. This difficulty can be partially overcome by using a different spatial increment, in which case the location of the y-grid lines are different in the two computations. Another possibility is to vary the size of the variable EPSI and not to use those grid points in determining the boundary motion at which the temperature is not known with sufficient degree of accuracy. The corresponding results are shown in Figure 5.9. The data for these runs are given in Table 5.7. The physical constants of run 12 were used for these computations as that run showed the most severe deviation when the boundary was in the neighborhood of a y-grid line.

TABLE 5.7

DATA: EFFECT OF EPSI.

| Run Number | EPSI intervals | P | Δx cm |
|---|---|---|---|
| 12 | 0.1 | 45 | 0.5 |
| 22 | 0.3 | 45 | 0.5 |
| 23 | 0.5 | 45 | 0.5 |
| 24 | 0.05 | 45 | 0.5 |
| 25 | 0.1 | 56 | 0.4 |

The rest of the data is identical with that of Run #12.

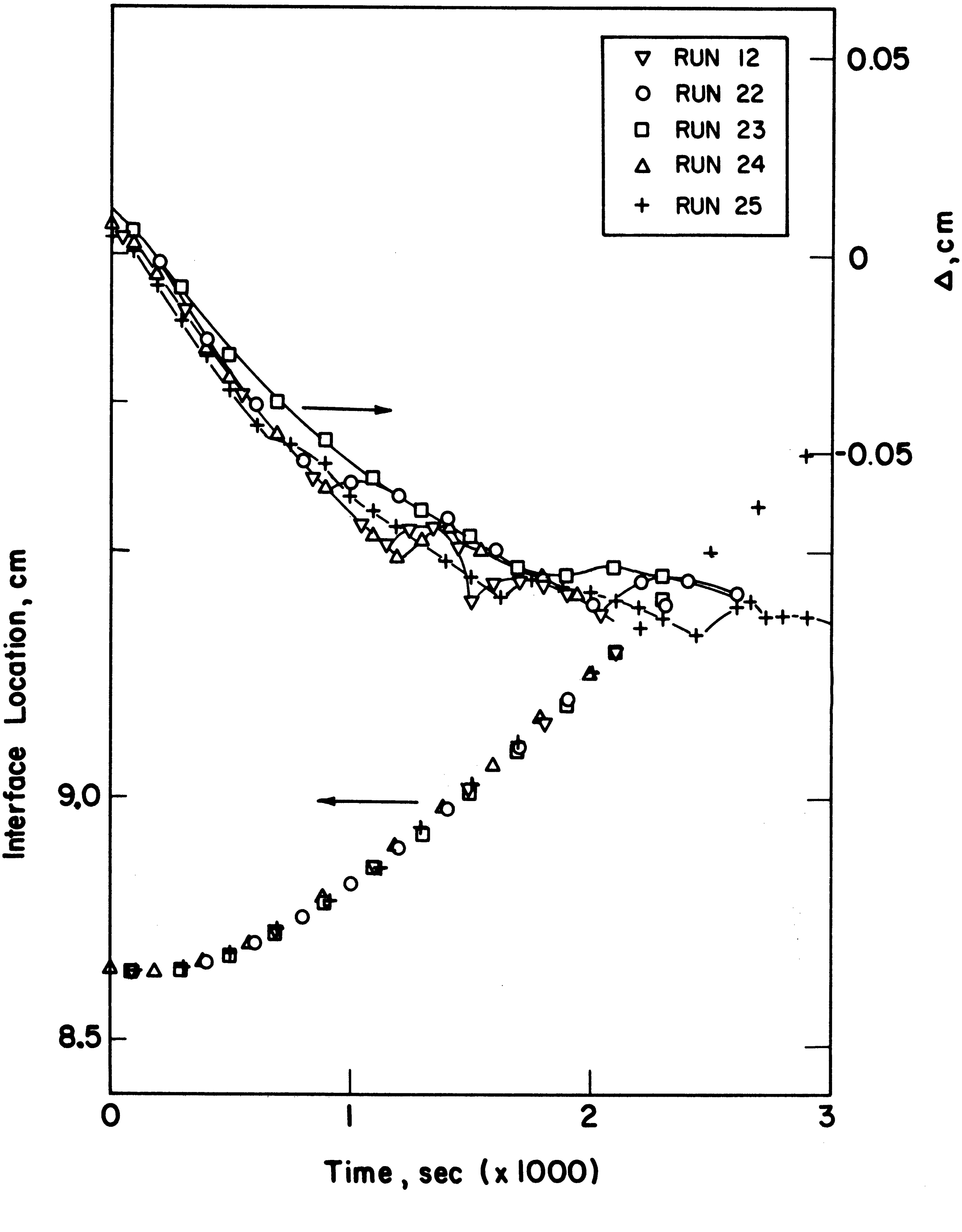

Figure 5.9.

Effect of EPSI.

The evidence presented in this chapter is sufficient to demonstrate that the method is convergent at least for the problems investigated, and that this method can be used to solve the Stefan problem in two dimensions with the accuracy required to predict the shape of the interface.

# CHAPTER VI

## THE EXPERIMENT

The experimental work carried out in this investigation consisted of building and operating a furnace capable of producing decanted solid-liquid interfaces. The furnace is described in Section I while the operating procedure and the results are discussed in Section 2.

### 6.I Description of the Furnace

The furnace was designed to be usable for both zone refining and normal freezing experiments. It is shown on Figure 6.I. The 44-inch long furnace was built inside a 9-inch diameter pipe which rests on 8 ball-bearings and can be rotated around its axis. The furnace is tubular: there is a 4I x 4I mm square opening along the axis. It is made up of three sections: the replaceable center zone and the fixed end zones.

The end zones are I8 inches long. The surface heaters were constructed by winding I8 gauge chromel wire around a rod which had a 4I x 4I mm square cross section. There are three such coils in each end zone: the end coils are 5 inches long while the center one is 8 inches long. The winding is made with a I/4 inch pitch; there are about 26 inches of wire in one inch of coil. These coils are surrounded by

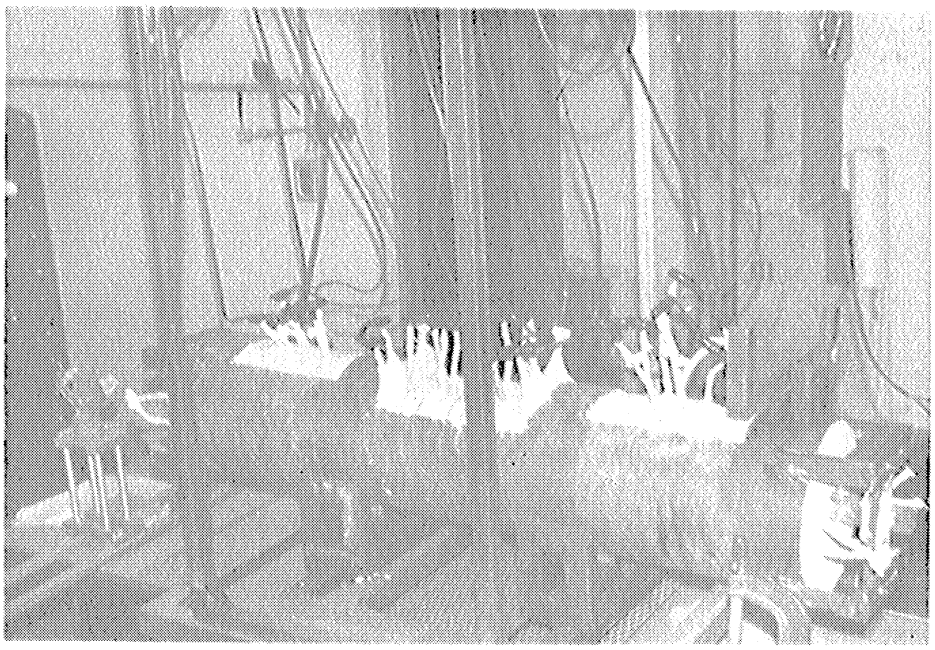

Figure 6.1

Photograph of the experimental furnace.

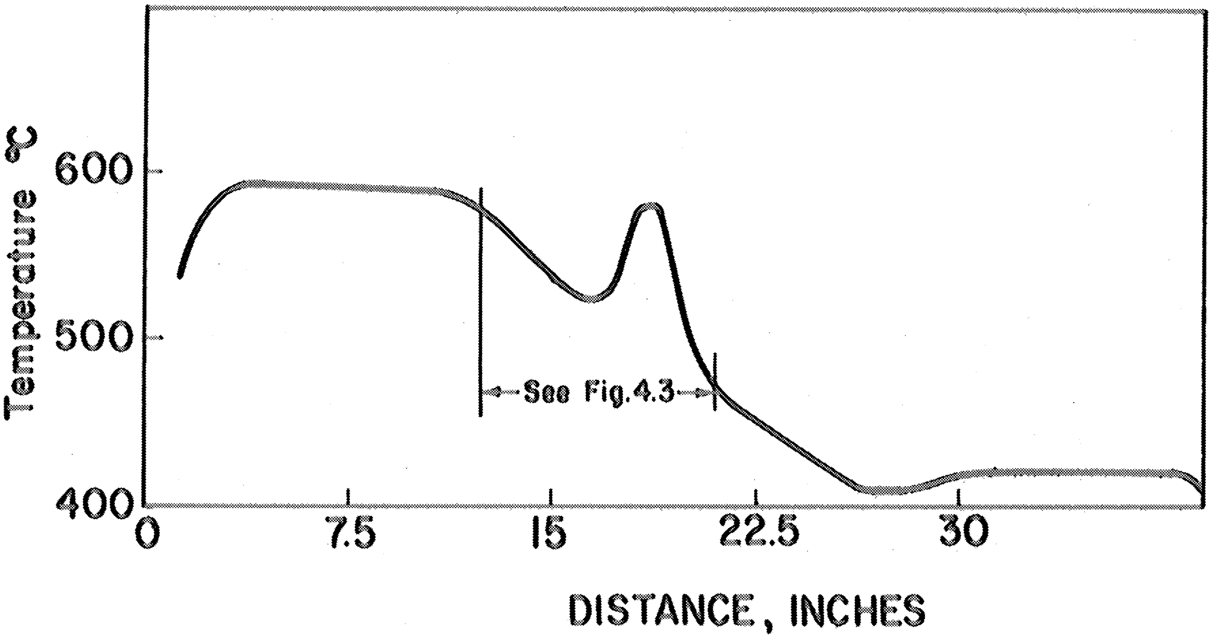

Figure 6.2

Temperature distribution in the experimental furnace.

asbestos insulation until the outside diameter reaches 3.5 inches. It is then surrounded by four 8 inch long "Heavy Duty" half shell heaters. The assembly is placed in position inside the shell and the gap between the heaters and the tube wall is filled with asbestos insulation. This arrangement allows the operator to maintain a fairly constant temperature distribution throughout the end zones.

The center heater is a separate furnace which is to be inserted into the opening left for it between the two end zones. It is 8 inches long and contains 5 equal coils. These coils were made in the same fashion as the coils in the end zones and were surrounded by asbestos insulation until the furnace reached the geometry of the opening. The five individual coils allow the operator to set the temperature distribution according to his needs. If some special temperature distribution pattern is desired which cannot be obtained with the present center zone, a new one can be easily constructed and inserted into the furnace quite readily. The temperature distribution in the furnace is illustrated in Figure 6.2. The portion of the furnace which was of greatest significance during the experiment is indicated in Figure 6.2 and the temperature distribution in this region is shown in Figure 4.3 in a slightly simplified form.

The furnace has a large time constant, and it tends to respond very slowly to changes in power input. Each coil has its own powerstat for power regulation. The smaller coils have a low resistance and the power has to be supplied at low voltage and high current levels. This is accomplished by inserting transformers between the coil and the powerstat. The temperature distribution to be achieved determines the power requirements of the individual heating elements. In order to keep the power input constant throughout the experimental period and avoid power fluctuations due to changes in line voltage,a constant voltage transformer was inserted into the circuit. This arrangement allows the operator to set up and maintain various temperature distributions in the furnace without complex instrumentation.

The furnace was erected next to a moving mechanism which was used to push the samples through the furnace at a constant velocity. In the experiments, 0.73 inches/hr velocity was used, although the mechanism is capable of moving with speeds as low as 0.15 inches/hr or as high as 5 inches/hr.

## 6.2 Procedure and Results

Indium Antimonide (InSb) was the material used for the experiments. It was prepared from its elements by fusion under vacuum to preserve purity. Following the fusion operation the material was rebottled into sample tubes made of General Electric type 204 clear fused quartz with a 20 x 20 mm square bore. A 22 cm long sample tube contained approximately 300 gm InSb.

In the experiments the sample tube containing the InSb was placed in the hot end of the furnace. It was left there for about five hours, during which time the material heated, melted and came to equilibrium. The moving mechanism was started at this point and the material in the sample tube was pushed toward the colder end of the furnace. In the first experiment the sample was pushed until the "expected equilibrium" interface position would have been 9.5 cm away from the cold end of the bar. The sample was then "dumped." In the second experiment the sample was pushed until the "expected equilibrium" interface position would have been 9 cm away from the cold end of the bar. The movement of the bar was stopped at this time, and the bar was left in the furnace for six hours before the sample was "dumped."

The "dumping" procedure was used to preserve the solid-liquid interface for later examination. The procedure consisted of rotating the furnace 180 degrees and pushing the rod rapidly through the colder end of the furnace into the open, where water was sprinkled on the quartz tube. During this process the solidified portion of the bar remained in its original position in the sample tube. The liquid flowed to that part of the tube, which used to be its upper empty portion, where it solidified very fast in response to the rapid cooling. Figure 6.3 shows the interface, still in the sample tube, in this position.

The data used in computer run I were an approximation of the physical constants in the experiments. The position of the interface agreed with the predicted results within the experimental uncertainty. This uncertainty was 1/8 of an inch, therefore, the results show qualitative agreement at most. Indeed, if the bar is cut by a plane which parallels the longitudinal axis of the bar, the following observations can be made:

(I) If the plane is vertical, the interface is not symmetrical around the axis of the bar. This can be attributed to two causes:

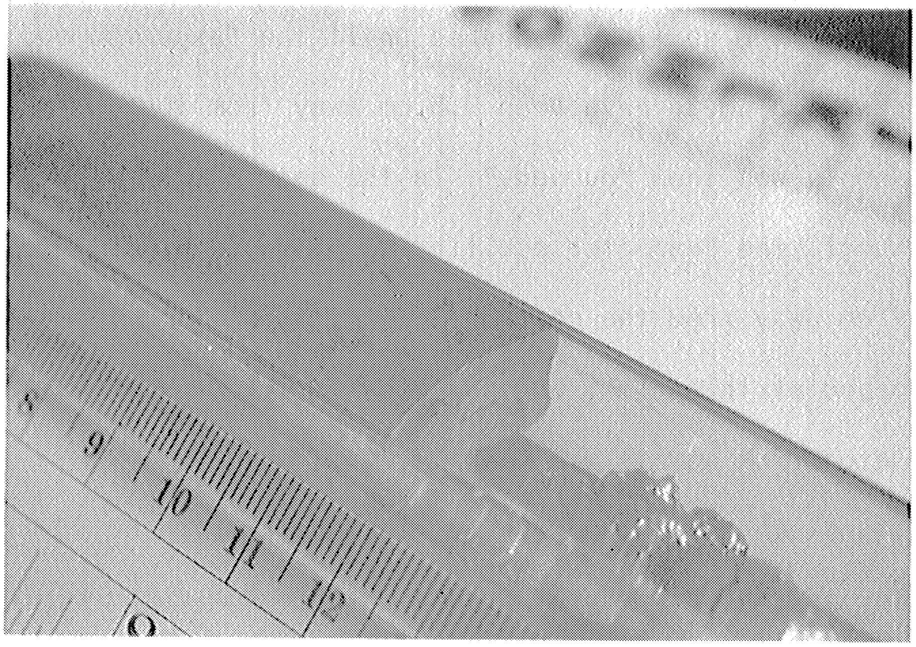

Figure 6.3
Photograph of the interface.

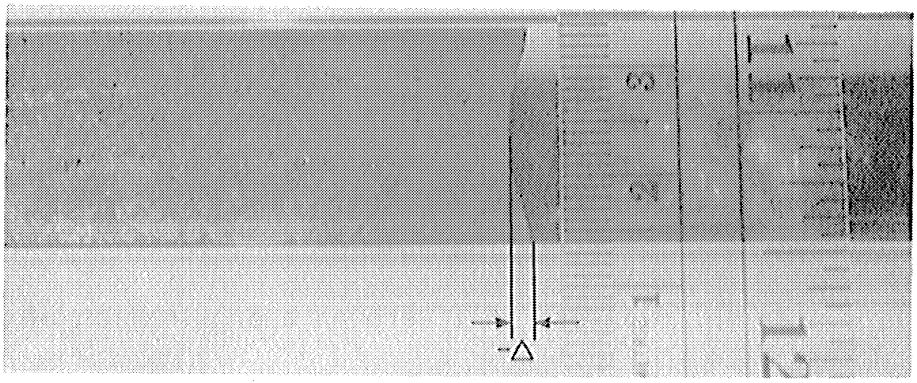

Figure 6.4
Interface shape in the first experiment.

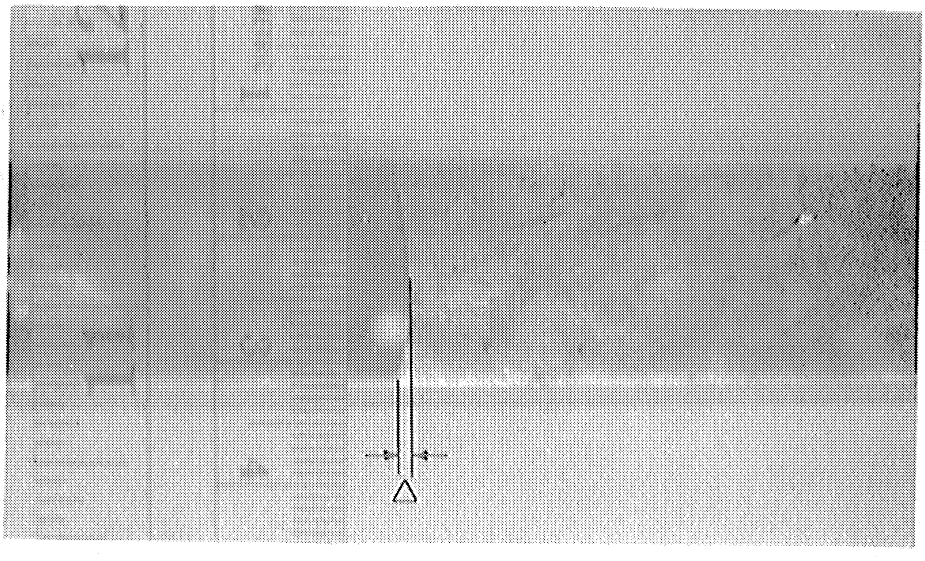

Figure 6.5
Interface shape in the second experiment.

(a) heat is transferred by natural convection in the liquid, i.e., movement of material does occur,

(b) the heat transfer coefficient at the top and at the bottom of the bar are not the same.

(2) If the plane is horizontal, the interface shape is symmetrical around the axis of the bar.

These observations and their explanation imply (a) that the two-dimensional analysis does not strictly apply to any horizontal plane, as heat is being transferred to or from this plane in the vertical direction; (b) that although two-dimensional analysis does apply to a vertical plane, natural convection is not negligible and has to be taken into account. These observations indicate that quantitative agreement cannot be expected, and more refined measurements would only confirm the observed discrepancy.

A closer look at the interface shape is still warranted, even though it only confirms this view. The curvature of the interface does not change appreciably with elevation. A photograph of the interface shape at the bottom of the sample tubes is given in Figures 6.4 and 6.5. The similarity of the interface shapes is immediately apparent. A closer measurement discloses the differences; Δ, which is marked on the photographs is more negative in the first experiment than in the second. The difference between the Δ values appears to agree with the calculated differences. This agreement is in harmony with the conclusions drawn in Section 5.2 from the computed results presented in Figure 5.4.

TABLE 6.1.

COMPARISON OF RESULTS

| | Δ | | Location of the Interface | |
|---|---|---|---|---|
| | Measured | Computed | Measured | Computed |
| First Experiment | - 0.13 cm | + 0.04 cm | 8.9 cm | 9.1 cm |
| Second Experiment | - 0.21 cm | - 0.05 cm | 8.7 cm | 9.0 cm |

TABLE 6.2.

PROPERTIES OF INDIUM ANTIMONIDE

| | |
|---|---|
| Density of solid | 5.8 gm/cc |
| Density of liquid | 6.4 gm/cc |
| Heat capacity of solid | 6.6 cal/°K mole |
| Heat capacity of liquid | 7.4 cal/°K mole |
| Thermal conductivity of solid | 0.018 cal/sec cm°K |
| Melting point | 525°C |
| Latent heat of fusion | 50 cal/gm |

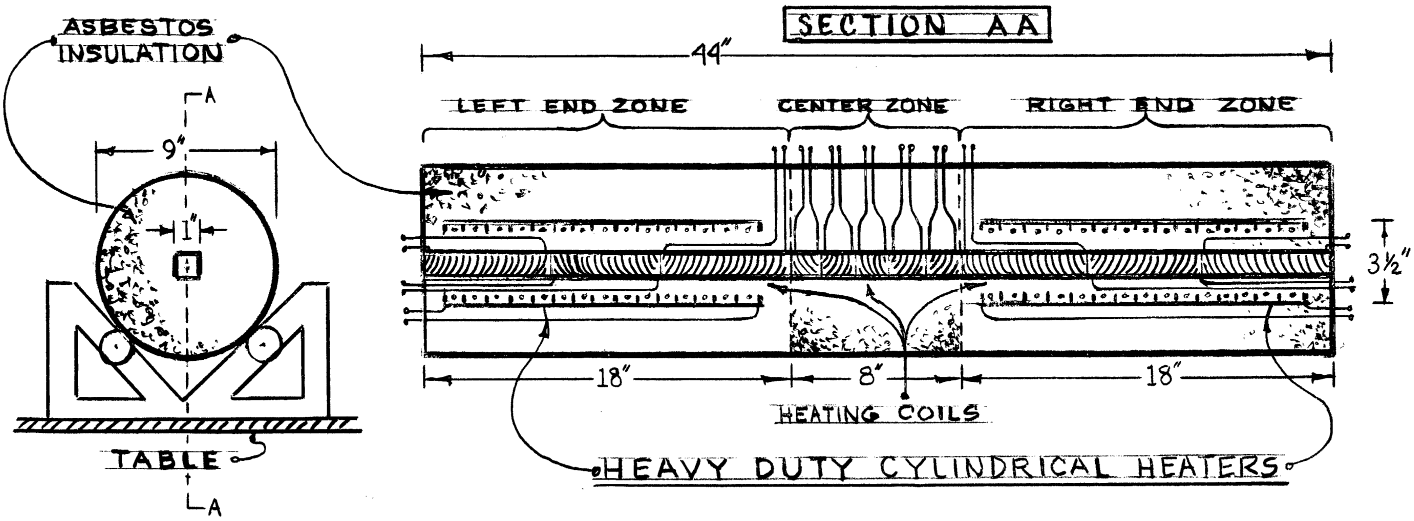

Figure 6.6.

Schematics of the experimental furnace.

# CHAPTER VII

## CONCLUSIONS AND RECOMMENDATIONS

A method for the solution of a two-dimensional Stefan problem has been developed. The computation based on this method is sufficiently accurate to predict not only the temperature distribution and the location of the interface but also the shape of the interface. In addition, a versatile furnace was constructed, which is capable of producing decanted solid-liquid interfaces during normal freezing and zone refining experiments. This feature allows the preservation of these interfaces for later examination.

The results of the experiments and computations cannot be compared directly as the computations are based on a mathematical model containing assumptions which were not met in the experiments. The main discrepancy in the computer program seems to be the assumption that no material movement takes place. This now seems quite unnecessary in view of Wilkes' (100) investigation. In future two-dimensional computations, natural convection should not be neglected. Further improvements in accuracy and generality may be achieved by developing a computer program in which the location of the y-interface points are preserved from iteration to iteration and used to compute the movement of the interface and the location of the new set of interface points. Future developments of this kind should be carried out in cylindrical coordinate systems as experimental verification of the computed results can then be carried out in a simple experiment. In some instances, when natural convection does not take place, the present program is applicable, and it can be extended to be applicable for three-dimensional problems as well. In such an extension some modifications will have to be made. For example, the temperature distribution will have to be computed in three dimensions. Since the ADI method proposed by Peaceman and Rachford (77) cannot be readily extended, it will have to be replaced, possibly by Brian's (15) method.

Regular tubing usually comes with a circular cross-section. If the temperature distribution around such a tube is symmetrical and the tube is held in a vertical position, the cylindrical symmetry is preserved. Normal freezing experiments (such as Bridgeman crystal growing) can be carried out under such conditions. Since such an experiment is truly two-dimensional, it can be used to verify the results of computations. If the physical constants are known and natural convection is taken into account, the results of such an experiment should closely check the results obtained by the appropriate computations.

## THE ROUTINE MOVEM.

```
              EXTERNAL FUNCTION
              REFERENCES ON
              ENTRY TO MOVEM.
   REM    R
   REM    R   SUBROUTINE MOVEM.
   REM    R   COMPUTES THE MOVMENT OF THE INTERFACE AND REDUCES THE
   REM    R   TIME-STEP IF THE PRESET TIME-STEP WOULD RESULT IN TOO
   REM    R   LARGE MOVEMENT OF THE BOUNDARY.
   REM    R
              MXMVRT=0.0
              THROUGH FIRST, FOR J=1,1,J.G.Q
              WHENEVER J.E.1
               RHOL=-1.5*Z(1)+2.0*Z(2)-0.5*Z(3)
              OR WHENEVER J.NE.Q
               RHOL=0.5*(Z(J+1)-Z(J-1))
              OTHERWISE
               RHOL=1.5*Z(Q)-2.0*Z(Q-1)+0.5*Z(Q-2)
              END OF CONDITIONAL
              RHOL=COEF3+COEF4*RHOL*RHOL
   REM    R
   REM    R   COMPUTE THE SLOPE IN THE FIRST REGION (I.L.Z)
   REM    R
              INI=Z(J)-EPS1
              BO1=INI
              INI=INI+CB(J)
              BO1=(Z(J)-BO1+1.0)/(BO1-Z(J))
              SLP1=T1(INI)*BO1-T1(INI-1)/BO1
   REM    R
   REM    R   COMPUTE THE SLOPE IN THE SECOND REGION  (I.G.Z)
   REM    R
              INI=Z(J)+EPS1+1.0
              BO1=INI
              INI=INI+CB(J)
              BO1=(Z(J)-BO1-1.0)/(Z(J)-BO1)
              SLP2=T1(INI)*BO1-T1(INI+1)/BO1
   REM    R
   REM    R   COMPUTE THE VELOCITY OF THE INTERFACE AND CHECK TIME STEP
   REM    R
              MVMTRT(J)=(COEF1*SLP1-COEF2*SLP2)*RHOL
              WHENEVER (.ABS.MVMTRT(J)).G.MXMVRT, MXMVRT=(.ABS.MVMTRT(J))
FIRST
              STEP=MXMVRT*DT
SECOND        THROUGH SECOND, FOR FACTOR=1.0, FACTOR, STEP/FACTOR.L.MXST
              WHENEVER FACTOR.NE.1.0, INSW(1)=1B
              PMAXI=0
              PMINI=P
   REM    R
   REM    R   COMPUTE THE PREDICTED INTERFACE LOCATON
   REM    R
              THROUGH THIRD, FOR J=1,1, J.G.Q
              ZPREDI(J)=Z(J)+MVMTRT(J)*DT/FACTOR
```

```
             ZHFPR(J)=( Z(J)+ZPREDI(J) )/2.0
             AZ1(J)=Z(J)-EPS
             AZ2(J)=ZPREDI(J)+1.0+EPS
             WHENEVER  AZ1(J). L .PMINI,  PMINI=AZ1(J)
             WHENEVER  AZ2(J). G .PMAXI,  PMAXI=AZ2(J)
THIRD
             FUNCTION RETURN
             REFERENCES OFF
             INTEGER AZ1,AZ2,CB,INI,J,PMAXI,PMINI,P,Q
             BOOLEAN INSW
             DIMENSION  MVMTRT(30)
   REM    R  FLOATING POINT BO1,COEF1,COEF2,COEF3,COEF4,
   REM    R                 DT,EPS,EPS1,FACTOR,MVMTRT,MXMVRT,MXST,RHOL,
   REM    R                 SLP1,SLP2,STEP,T1,ZHFPR,ZPREDI,Z
   REM    R  STATEMENT LABEL FIRST,SECOND,THIRD
   REM    R  FUNCTION NAME MOVEM
   REM    R  PROGRAM COMMON AZ1,AZ2,CB,CC1,COEF1,COEF2,COEF3,
   REM    R                 COEF4,DT,EPS,EPS1,FACTOR,INSW,MXST,PMAXI,
   REM    R                 PMINI,P,Q,RHOL,SLP1,SLP2,T1,ZHFPR,ZPREDI,Z
             EQUIVALENCE (CC1,EPS1)
           PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
          1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
          2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
          3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
          4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
          5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
          6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
          7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
           INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
           BOOLEAN INSW
           END OF FUNCTION
```

## THE ROUTINE T2SOLV.

```
              EXTERNAL FUNCTION T2SOLV.
              REFERENCES ON
   REM     R
   REM     R  SUBROUTINE T2SOLV.
   REM     R  FOR ALL VALUES OF I THE FOLLOWING SET OF EQUATIONS ARE
   REM     R  SOLVED IN EACH REGION. THE PARAMETERS ARE DEFINED ELSEWHERE
   REM     R  A*T2(I,J-1)+B*T2(I,J)+C*T2(I,J+1)=D
   REM     R
              THROUGH NEXT, FOR I=1,1, I.G.P
              MIN=1
              THROUGH NEXT, FOR J=1,1, J.G.Q
              JVCT=CB(J)
              WHENEVER I.E.1
               PHASE=1
               INI=I+1
               TSUB=TI1(0)
OVER
               D(J)=2.0*((H(PHASE)+1.0-L(PHASE))*T1(I+JVCT)-H(PHASE)*
           1        TSUB-T1(INI+JVCT))
               TRANSFER TO HOME
              OR WHENEVER I.NE.P
               WHENEVER I.L.PMINI
                PHASE=1
FIRST
                D(J)=(2.0-TWOL(PHASE))*T1(I+JVCT)-T1(I-1+JVCT)-
           1        T1(I+1+JVCT)
HOME
                WHENEVER J.E.1
                 A(1)=2.0*RSQ
                 BT(1)=BUU(PHASE)
                 GM(1)=(D(1)-A(1)*G(PHASE)*TJ1(I))/BT(1)
                 TRANSFER TO NEXT
                OR WHENEVER J.NE.Q
                 A(J)=RSQ
                 BT(J)=BU(PHASE)-RSQ*A(J-1)/BT(J-1)
                 GM(J)= (D(J)-RSQ*GM(J-1))/BT(J)
                 TRANSFER TO NEXT
                OTHERWISE
                 T2(I+JVCT)= (D(Q)-2.0*RSQ*(G(PHASE)*TJQ(I)+GM(Q-1)))/
           1        (BUU(PHASE)-2.0*RSQ*A(J-1)/BT(J-1))
                 MAX=Q
THIRD
                 THROUGH DONE, FOR INI=MAX-1, -1, INI.L.MIN
                 T2(I+CB(INI))=GM(INI)-A(INI)*T2(I+CB(INI+1))/BT(INI)
DONE
                 MIN=MAX+1
                END OF CONDITIONAL
                TRANSFER TO NEXT
              OR WHENEVER I.G.PMAXI
               PHASE=2
               TRANSFER TO FIRST
              OTHERWISE
   REM    R
   REM    R    I IS NOT EQUAL TO 1, P, AND IS NEAR THE BOUNDARY
   REM    R    FIRST THE VALUE OF D IS COMPUTED
   REM    R
               WHENEVER I.E.AZ1(J)
                PHASE=1
                INCR=Z(J)-I
                INI=I-1
RUN
                D(J)=(2.0/INCR-TWOL(PHASE))*T1(I+JVCT)-2.0*T1(INI+JV
          1        /(1.0+INCR)
               OR WHENEVER I.E.AZ2(J)
                PHASE=2
                INCR=I-Z(J)
                INI=I+1
                TRANSFER TO RUN
               OR WHENEVER I.L.AZ1(J)
                PHASE=1
ANDYS           D(J)=(2.0-TWOL(PHASE))*T1(I+JVCT)-T1(I-1+JVCT)
          1        -T1(I+1+JVCT)
               OR WHENEVER I.G.AZ2(J)
                PHASE=2
                TRANSFER TO ANDYS
               OTHERWISE
                T2(I+JVCT)=0.0
                MIN=J+1
                TRANSFER TO NEXT
               END OF CONDITIONAL
               WHENEVER J.E.1
                WHENEVER PHASE.E.1
                 WHENEVER I.LE.AZ1(2), TRANSFER TO HOME
THIS
                 IFL=I
                 ROOT.(1.0,ZHFPR(1)-IFL,2.0,ZHFPR(2)-IFL,3.0,ZHFPR(3
          1        IFL,1.5)
                 INCR=RHOL-1.0
                 TSUB=TJ1(I)
HERE
                 T2(I+JVCT)=(D(J)-2.0*RSQ*G(PHASE)*TSUB/INCR)/(-2.0*
          1        (RSQ/(INCR*INCR)+RSQ*G(PHASE)/INCR+L(PHASE)))
                 MIN=J+1
                 TRANSFER TO NEXT
                OTHERWISE
   REM    R
   REM    R       PHASE == 2
   REM    R
                 WHENEVER I.GE.AZ2(2), TRANSFER TO HOME
                 TRANSFER TO THIS
                END OF CONDITIONAL
               OR WHENEVER J.NE.Q
   REM    R
   REM    R    J =/= 1,Q
   REM    R
                WHENEVER J.NE.MIN
                 WHENEVER PHASE.E.1
                  WHENEVER  I.LE.AZ1(J+1), TRANSFER TO HOME
                  IFL=I
                  JFL=J
                  WHENEVER J.L.(Q-1)
                   W2=JFL+2.0
                   BO2=ZHFPR(J+2)-IFL
                  OTHERWISE
OTHER
                   W2=JFL-1.0
                   BO2=ZHFPR(J-1)-IFL
                  END OF CONDITIONAL
                  MAX=J
                  ROOT.(JFL,ZHFPR(J)-IFL,JFL+1.0,ZHFPR(J+1)-IFL,W2,BO2,
          1        JFL+0.5)
                  INCR=RHOL-JFL
   REM    R
   REM    R       ONLY IN THE FOLLOWING TWO STATEMENTS
   REM    R       A(J)  STANDS FOR THE COEFICIENT 'A' OF THE EQUATION..
   REM    R       OTHERWISE IT STANDS FOR THE COEFFICIENT 'C'........
   REM    R
                  A(J)=2.0*RSQ/(1.0+INCR)
                  T2(I+JVCT)=(D(J)-A(J)*GM(J-1))/(-2.0*RSQ/INCR
          1        -TWOL(PHASE)-A(J)*A(J-1)/BT(J-1))
                  TRANSFER TO THIRD
                 OTHERWISE
   REM    R
   REM    R      PHASE == 2
   REM    R
                  WHENEVER I.GE.AZ2(J+1), TRANSFER TO HOME
                  IFL=I
                  JFL=J
                  TRANSFER TO OTHER
                 END OF CONDITIONAL
                OTHERWISE
   REM    R
   REM    R     J == MIN
   REM    R
                 IFL=I
                 JFL=J
                 WHENEVER J.G.2.AND.PHASE.E.1
                  W2=JFL-2.0
                  BO2=ZHFPR(J-2)-IFL
                 OTHERWISE
                  W2=JFL+1.0
                  BO2=ZHFPR(J+1)-IFL
                 END OF CONDITIONAL
                 W1=JFL-1.0
                 BO1=ZHFPR(J-1)-IFL
                 SL=JFL-0.5
DL1
                 ROOT.(JFL,ZHFPR(J)-IFL,W1,BO1,W2,BO2,SL)
                 WHENEVER SL.E.JFL-0.5
                  INCR=JFL-RHOL
                  WHENEVER PHASE.E.1
                   WHENEVER I.LE.AZ1(J+1)
CHANGE
                    A(J)=2.0*RSQ/(1.0+INCR)
                    BT(J)=-2.0*RSQ/INCR-TWOL(PHASE)
                    GM(J)=D(J)/BT(J)
                    TRANSFER TO NEXT
                   OTHERWISE
                    WHENEVER (J.L.Q-1)
                     W1=JFL+2.0
                     BO1=ZHFPR(J+2)-IFL
                    OTHERWISE
ANN
                     W1=JFL-1.0
                     BO1=ZHFPR(J-1)-IFL
                    END OF CONDITIONAL
                    W2=JFL+1.0
                    BO2=ZHFPR(J+1)-IFL
                    SL=JFL+0.5
                    TRANSFER TO DL1
                   END OF CONDITIONAL
                  OTHERWISE
   REM    R
   REM    R       PHASE == 2
   REM    R
                   WHENEVER I.GE.AZ2(J+1), TRANSFER TO CHANGE
                   TRANSFER TO ANN
                  END OF CONDITIONAL
                 END OF CONDITIONAL
                 T2(I+JVCT)=D(J)/(-2.0*RSQ/(INCR*(RHOL-JFL))
          1        -TWOL(PHASE))
                 MIN=J+1
                 TRANSFER TO NEXT
                END OF CONDITIONAL
               OTHERWISE
   REM    R
   REM    R    J == Q
   REM    R
                WHENEVER J.NE.MIN, TRANSFER TO HOME
                IFL=I
                JFL=J
                ROOT.(JFL,ZHFPR(J)-IFL,JFL-1.0,ZHFPR(J-1)-IFL,JFL-2.0,
          1        ZHFPR(J-2)-IFL,JFL-0.5)
                INCR=JFL-RHOL
                TSUB=TJQ(I)
                TRANSFER TO HERE
               END OF CONDITIONAL
              END OF CONDITIONAL
             OTHERWISE
   REM    R
   REM    R  I == P
   REM    R
              WHENEVER P.G.PMINI, PHASE=2
              INI=I-1
              TSUB=TIP(0)
              TRANSFER TO OVER
             END OF CONDITIONAL
NEXT
             FUNCTION RETURN
             REFERENCES OFF
             INTEGER AZ1,AZ2,CB,I,INI,J,JVCT,MAX,MIN,PHASE,PMAXI,
          1         PMINI,P,Q
   REM    R  STATEMENT LABEL ANDYS,ANN,CHANGE,DL1,DONE,FIRST,HERE,HOME,
   REM    R                 NEXT,OTHER,OVER,RUN,THIS,THIRD
   REM    R  FLOATING POINT A,BO1,BO2,BT,BU,BUU,D,G,GM,H,IFL,INCR,JFL,
   REM    R                L,RHOL,RSQ,SL,T1,T2,TI1,TIP,TJ1,TJQ,TSUB,
   REM    R                TWOL,W1,W2,Z,ZHFPR
   REM    R  FUNCTION NAME ROOT,T2SOLV
             ERASABLE ZZZ(7),A(300),BT(300),D(300),GM(300)
   REM    R  PROGRAM COMMON AZ1,AZ2,BU,BUU,CB,G,H,L,PMAXI,PMINI,P,Q,
   REM    R                RHOL,RSQ,T1,T2,TI1,TIP,TJ1,TJQ,TWOL,Z,ZHFPR
           PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
          1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
          2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
          3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
          4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
          5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
          6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
          7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
           INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
           BOOLEAN INSW
           END OF FUNCTION
```

## THE ROUTINE T3SOLV.

```
            EXTERNAL FUNCTION T3SOLV.
   REM    R
   REM    R SUBROUTINE T3SOLV.
   REM    R FOR ALL VALUES OF J THE FOLLOWING SET OF EQUTIONS ARE
   REM    R SOLVED IN EACH REGION. THE COEFICIENTS ARE DEFINED ELSEWHERE
   REM    R  A*T3(I-1,J)+B*T3(I,J)+C*T3(I+1,J)=D
   REM    R
             MIN=1
             PHASE=1
            WHENEVER P.L.PMINI
   REM    R
   REM    R   ONLY ONE REGION EXIST.  ITS LIMITS ARE SET FOR EACH J.
   REM    R
             MAX=P
             THROUGH CHANGE, FOR J=1,1,J.G.Q
CHANGE       EXECUTE T3COMP.
             FUNCTION RETURN
            OTHERWISE
   REM    R
   REM    R   TWO REGIONS EXIST. THE LIMITS OF EACH ARE SET SEPARETLY.
   REM    R
             THROUGH FIRST, FOR J=1,1,J.G.Q
             MAX=AZ1(J)
             T3COMP.
             WHENEVER AZ2(J).NE.(AZ1(J)+1)
             T3(AZ1(J)+1+CB(J))=0.0
             END OF CONDITIONAL
FIRST
             MAX=P
             PHASE=2
             THROUGH SECOND, FOR J=1,1, J.G.Q
             MIN=AZ2(J)
             T3COMP.
SECOND
            END OF CONDITIONAL
            FUNCTION RETURN
            INTERNAL FUNCTION T3COMP.
   REM    R
   REM    R SETS UP AND SOLVES THE SYSTEM OF EQUTIONS FOR A GIVEN
   REM    R VALUE OF J INSIDE THE GIVEN REGION.
   REM    R
            JVCT=CB(J)
            THROUGH THIS, FOR I=MIN, 1, I.G.MAX
   REM    R  COMPUTE THE VALUE OF THE J DEPENDENT PART OF D
   REM    R
            WHENEVER J.E.1
             W=RSQ*G(PHASE)
             WHENEVER (I.GE.AZ2(2).AND.PHASE.E.2).OR.(I.LE.AZ1(2).AND.
          1       PHASE.E.1)
              D(I)=2.0*((RSQ+W-L(PHASE))*T2(I+JVCT)-RSQ*T2(I+CB(2))-W*
          1       TJ1(I))
              TRANSFER TO HOME
             OTHERWISE
```

```
                IFL=I
                ROOT.(1.0,ZHFPR(1)-IFL,2.0,ZHFPR(2)-IFL,3.0,
          1         ZHFPR(3)-IFL,1.5)
                INCR=RHOL-1.0
                TSUB=TJ1(I)
DONE
                W=W/INCR
                D(I)=2.0*((RSQ/(INCR*INCR)+W-L(PHASE))*T2(I+JVCT)-W*TSUB
               END OF CONDITIONAL
               TRANSFER TO HOME
              OR WHENEVER J.NE.Q
   REM     R  J =/= 1,Q
   REM     R
               WHENEVER PHASE.E.1
   REM     R   PHASE == 1
   REM     R
                WHENEVER I.LE.AZ1(J-1)
   REM     R    NOT BOUNDED ON  J-1  SIDE
   REM     R
                 WHENEVER I.LE.AZ1(J+1)
   REM     R
   REM     R      NOT BOUNDED ON EITHER SIDE
ANDYS
                  D(I)=2.0*(RSQ-L(PHASE))*T2(I+JVCT)-RSQ*(T2(I+CB(J+1))+
          1         T2(I+CB(J-1)))
                  TRANSFER TO HOME
                 OTHERWISE
   REM     R      BOUNDED ON THE J+1 SIDE ONLY
   REM     R
                  JFL=J
                  IFL=I
                  WHENEVER J.NE.Q
                   BO2 = ZHFPR(J+2)-I
                   W2 = JFL+2.0
                  OTHERWISE
DL1
                   W2=JFL-1.0
                   BO2=ZHFPR(J-1)-I
                  END OF CONDITIONAL
                  SL=JFL+0.5
                  W=JFL+1.0
                  BO1=ZHFPR(J)-IFL
                  BO3=ZHFPR(J+1)-IFL
                  ROOT.(JFL,BO1,W2,BO2,W,BO3,SL)
                  INCR=RHOL-JFL
                  TSUB=T2(I+CB(J-1))
THIRD
                  D(I)=(RSQ/INCR-L(PHASE))*2.0*T2(I+JVCT)-2.0*RSQ*
          1         TSUB/(1.0+INCR)
                  TRANSFER TO HOME
                 END OF CONDITIONAL
                OR WHENEVER I.LE.AZ1(J+1)
   REM     R    BOUNDED  ONLY  ON  J-1  SIDE
   REM     R
                 JFL = J
                 IFL = I
                WHENEVER J.NE.2
                 W2=JFL-2.0
                 BO2=ZHFPR(J-2)-IFL
                OTHERWISE
ANN
                 W2=JFL+1.0
                 BO2=ZHFPR(J+1)-IFL
                END OF CONDITIONAL
                ROOT.( JFL-1.0,ZHFPR(J-1)-IFL,JFL,ZHFPR(J)-IFL,
          1        W2,BO2,JFL-0.5)
                INCR = JFL-RHOL
                TSUB=T2(I+CB(J+1))
                TRANSFER TO THIRD
               OTHERWISE
   REM    R    BOUNDED ON BOTH SIDES
   REM    R
                IFL=I
                JFL=J
                WHENEVER J.E.2
                 W2=JFL+1.0
                 BO2=ZHFPR(J+1)-IFL
                OTHERWISE
                 W2=JFL-2.0
                 BO2=ZHFPR(J-2)-IFL
                END OF CONDITIONAL
                W1=JFL-1.0
                BO1=ZHFPR(J-1)-IFL
                SL=JFL-0.5
HERE
                ROOT.(JFL,ZHFPR(J)-IFL,W1,BO1,W2,BO2,SL)
                WHENEVER SL.E.(JFL-0.5)
                 INCR=JFL-RHOL
                 W1=JFL+1.0
                 BO1=ZHFPR(J+1)-IFL
                 SL=JFL+0.5
                 WHENEVER J.E.Q-1
                  W2=JFL-1.0
                  BO2=ZHFPR(J-1)-IFL
                 OTHERWISE
                  W2=JFL+2.0
                  BO2=ZHFPR(J+2)-IFL
                 END OF CONDITIONAL
                 TRANSFER TO HERE
                END OF CONDITIONAL
RET
                D(I)=(RSQ/((RHOL-JFL)*INCR)-L(PHASE))*2.0*T2(I+JVCT)
                TRANSFER TO HOME
               END OF CONDITIONAL
              OTHERWISE
   REM    R   PHASE == 2
   REM    R
               WHENEVER I.GE.AZ2(J-1)
   REM    R     NOT BOUNDED ON THE J-1 SIDE
   REM    R
                WHENEVER I.GE.AZ2(J+1)
   REM    R      NOT BOUNDED ON EITHER SIDE
   REM    R
                 TRANSFER TO ANDYS
                OTHERWISE
   REM    R      BOUNDED ON THE J+1 SIDE ONLY
   REM    R
                 JFL=J
                 IFL=I
                 TRANSFER TO DL1
                END OF CONDITIONAL
               OR WHENEVER I.GE.AZ2(J+1)
   REM    R     NOT BOUNDED ON THE J-1 SIDE
   REM    R
                JFL=J
                IFL=I
                TRANSFER TO ANN
               OTHERWISE
   REM    R    BOUNDED ON BOTH SIDES
   REM    R
                JFL=J
                IFL=I
                W=JFL-1.0
                W1=JFL+1.0
                BO1=ZHFPR(J-1)-IFL
                BO2=ZHFPR(J)-IFL
                BO3=ZHFPR(J+1)-IFL
                SL=JFL-0.5
OTHER
                ROOT.(W,BO1,JFL,BO2,W1,BO3,SL)
                WHENEVER SL.E.(JFL-0.5)
                 INCR=JFL-RHOL
                 SL=JFL+0.5
                 TRANSFER TO OTHER
                END OF CONDITIONAL
                TRANSFER TO RET
               END OF CONDITIONAL
              END OF CONDITIONAL
             OTHERWISE
   REM    R  J == Q
   REM    R
              W=RSQ*G(PHASE)
              WHENEVER (I.GE.AZ2(J-1).AND.PHASE.E.2).OR.(I.LE.AZ1(J-1)
          1        .AND.PHASE.E.1)
               D(I)=2.0*((RSQ+W-L(PHASE))*T2(I+JVCT)-RSQ*T2(I+CB(Q-1))-
          1        W*TJQ(I))
               TRANSFER TO HOME
              OTHERWISE
               JFL=J
               IFL=I
               ROOT.(JFL,ZHFPR(J)-IFL,JFL-1.0,ZHFPR(J-1)-IFL,JFL-2.0,
          1        ZHFPR(J-2)-IFL,JFL-0.5)
               INCR=JFL-RHOL
               TSUB=TJQ(I)
               TRANSFER TO DONE
              END OF CONDITIONAL
             END OF CONDITIONAL
```

```
HOME
   REM    R  COMPUTE THE COEFFICIENTS, GAMMA AND BETA
   REM    R
             WHENEVER I.E.1
              A(1)=2.0
              BT(1)=BII(PHASE)
              GM(1)=(D(1)-2.0*H(PHASE)*TI1(0))/BT(1)
             OR WHENEVER I.NE.P
              WHENEVER I.E.AZ2(J)
               INCR=I-ZPREDI(J)
               A(I)=2.0/(INCR+1.0)
               BT(I)=-2.0/INCR-TWOL(PHASE)
               GM(I)=D(I)/BT(I)
              OR WHENEVER I.NE.AZ1(J)
               A(I)=1.0
               BT(I)=BI(PHASE)-A(I-1)/BT(I-1)
               GM(I)=(D(I)-GM(I-1))/BT(I)
              OTHERWISE
               INCR=ZPREDI(J)-I
               T3(I+JVCT)=(D(I)-2.0*GM(I-1)/(1.0+INCR))/(-2.0/INCR-
          1         TWOL(PHASE)-2.0*A(I-1)/((1.0+INCR)*BT(I-1)))
              END OF CONDITIONAL
             OTHERWISE
              T3(P+CB(J))=(D(P)-2.0*(H(PHASE)*TIP(0)+GM(P-1)))/
          1         (BII(PHASE)-2.0*A(P-1)/BT(P-1))
             END OF CONDITIONAL
THIS
   REM    R  SUBSTITUTE BACK TO OBTAIN  T3
   REM    R
             THROUGH OVER, FOR I=MAX-1, -1, I.L.MIN
             T3(I+JVCT)=GM(I)-A(I)*T3(I+1+JVCT)/BT(I)
OVER
             FUNCTION RETURN
             END OF FUNCTION
             INTEGER AZ1,AZ2,CB,I,J,JVCT,MAX,MIN,PHASE,PMINI,P,Q
   REM    R  STATEMENT LABEL ANDYS,ANN,CHANGE,DL1,DONE,FIRST,HERE,HOME,
   REM    R                  OTHER,OVER,RET,SECOND,THIRD,THIS
   REM    R  FLOATING POINT A,BI,BII,BO1,BO2,BO3,BT,D,G,GM,H,INCR,JFL,
   REM    R                 IFL,L,RHOL,RSQ,SL,T2,T3,TI1,TIP,TJ1,TJQ,
   REM    R                 TSUB,TWOL,W,W1,W2,ZHFPR,ZPREDI
   REM    R  FUNCTION NAME ROOT,T3COMP,T3SOLV
             ERASABLE ZZZ(7),A(300),BT(300),D(300),GM(300)
   REM    R  PROGRAM COMMON AZ1,AZ2,BI,BII,CB,G,H,L,PMINI,P,RHOL,RSQ,
   REM    R                 T2,T3,TI1,TIP,TJ1,TJQ,TWOL,ZHFPR,ZPREDI
           PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
          1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
          2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
          3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
          4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
          5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
          6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
          7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
           INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
           BOOLEAN INSW
           END OF FUNCTION
```

## THE BOUNDARY TEMPERATURE ROUTINES.

### 1. FIRST.

```
            EXTERNAL FUNCTION BNDSET.
   REM    R
   REM    R SETS UP THE BOUNDARY TEMPERATURE DISTRIBUTION. A CALL ON
   REM    R BOUND. RESETS THE TIME DEPENDENT BOUNDARY VALUES.
   REM    R THIS ROUTINE IS DESIGNATED FIRST.
   REM    R
            REFERENCES ON
            TE2=THSTRT
   REM    R DELTA == THE NUMBER OF INTERVALS/INCH.
            DELTA=2.54*(P-1)/L2
            TE1=TE2-2.3*DELTA
            TE0=TE2-5.1*DELTA
            TE3=TE2+0.9*DELTA
            TE4=TE2+1.9*DELTA
            TE5=TE2+3.0*DELTA
            TE6=TE2+6.7*DELTA
            HOLT=-33.0/20.0
   REM    R
   REM    R COMPUTE THE SLOPE OF THE TEMP-DISTANCE FUNCTION BETWEEN
   REM    R POINTS 0 AND 1, 1 AND 2, 3 AND 4, 5 AND 6.
   REM    R
            SL0=(HOLT-TL)/(TE1-TE0)
            SL1=(1.0-HOLT)/(TE2-TE1)
            SL3=1.0/(TE3-TE4)
            SL5=1.0/(TE6-TE5)
            MOVM=0.0
            TRANSFER TO OVER
            ENTRY TO BOUND.
   REM    R
   REM    R COMPUTE THE DISTANCE MOVED BETWEEN TMOVE AND THE PRESENT
   REM    R TIME.
   REM    R
            WHENEVER T.G.TMOVE
             MOVM=(T-TMOVE)*VELOC
            OTHERWISE
             MOVM=0.0
            END OF CONDITIONAL
OVER
            IT0=MOVM+TE0
            IT1=MOVM+TE1
            IT2=MOVM+TE2
            IT3=MOVM+TE3
            IT4=MOVM+TE4
            IT5=MOVM+TE5
   REM    R
   REM    R COMPUTE THE TEMPERATURE OF THE SURROUNDING.
   REM    R
            THROUGH FIRST, FOR I=1, 1, I.G.P
            IFL=I
            WHENEVER (IFL.GE.IT0).AND.(IFL.LE.IT1)
            TJ1(I)=TL+(IFL-IT0)*SL0
            OR WHENEVER (IFL.GE.IT1).AND.(IFL.LE.IT2)
            TJ1(I)=HOLT+(IFL-IT1)*SL1
            OR WHENEVER (IFL.GE.IT2).AND.(IFL.LE.IT3)
            TJ1(I)=1.0
            OR WHENEVER (IFL.GE.IT3).AND.(IFL.LE.IT4)
            TJ1(I)=1.0+(IFL-IT3)*SL3
            OR WHENEVER (IFL.GE.IT4).AND.(IFL.LE.IT5)
            TJ1(I)=0.0
            OTHERWISE
            TJ1(I)=(IFL-IT5)*SL5
            END OF CONDITIONAL
            TJQ(I)=TJ1(I)
FIRST
            TI1(0)=TJ1(1)
            TIP(0)=TJ1(P)
            FUNCTION RETURN
             REFERENCES OFF
            INTEGER I,P
   REM    R FLOATING POINT DELTA,HOLT,IFL,IT0,IT1,IT2,IT3,IT4,IT5,L2,
   REM    R                MOVM,SL0,SL1,SL3,SL5,T,TE0,TE1,TE2,TE3,TE4,
   REM    R                TE5,TE6,THSTRT,TI1,TIP,TJ1,TJQ,TL,TMOVE,VELOC
   REM    R STATEMENT LABEL FIRST,OVER
   REM    R FUNCTION NAME BNDSET,BOUND
   REM    R PROGRAM COMMON P,THSRT,TI1,TIP,TJ1,TJQ,TL,TMOVE,T,VELOC
           PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
          1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
          2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
          3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
          4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
          5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
          6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
          7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
           INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
           BOOLEAN INSW
           END OF FUNCTION
```

2. <u>BND</u>.

```
             EXTERNAL FUNCTION
             REFERENCES ON
   REM    R  THIS ROUTINE IS DESIGNATED BND.
   REM    R
   REM    R  SUBROUTINE BNDSET.
   REM    R  SETS UP A MODIFIED STEP FUNCTION BOUNDARY TEMPERATURE
   REM    R  DISTRIBUTION.
   REM    R  SUBROUTINE BOUND.
   REM    R  RESETS TIME DEPENDENT BOUNDARY CONDITIONS
   REM    R
             ENTRY TO BNDSET.
             TI1(0)=TL
             TIP(0)=TH
             TE1=(TH-TL)/WIDTH
             TE2=THSTRT
             BO1=TE2-WIDTH
             TRANSFER TO OVER
             ENTRY TO BOUND.
             WHENEVER T.G.TMOVE
              TE2=THSTRT+(T-TMOVE)*VELOC
             OTHERWISE
              TE2=THSTRT
             END OF CONDITIONAL
             BO1=TE2-WIDTH
OVER
   REM    R  THE TEMPERATURE IS TL WHEN I IS LESS THAN BO2, AND IT IS
   REM    R  TH WHEN I IS GREATER THAN TE2.
   REM    R
             THROUGH FIRST, FOR I=1,1, I.G.P
             IFL=I
             WHENEVER IFL.G.TE2
              TJ1(I)=TH
             OR WHENEVER IFL.L.BO1
              TJ1(I)=TL
             OTHERWISE
              TJ1(I)=TL+(IFL-BO1)*TE1
             END OF CONDITIONAL
             TJQ(I)=TJ1(I)
FIRST
             FUNCTION RETURN
             REFERENCES OFF
             INTEGER I,P
   REM    R  FLOATING POINT BO1,IFL,TE1,TE2,TH,THSTRT,TI1,TIP,TJ1,TJQ,
   REM    R                 TL,TMOVE,T,VELOC,WIDTH
   REM    R  STATEMENT LABEL FIRST, OVER
   REM    R  FUNCTION NAME BNDSET,BOUND
   REM    R  PROGRAM COMMON P,TH,THSTRT,TI1,TIP,TJ1OLD,TJ1,TJQOLD,TJQ,
   REM    R                 TL,TMOVE,T,VELOC,WIDTH
           PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
          1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
          2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
          3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
          4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
          5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
          6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
          7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
           INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
           BOOLEAN INSW
           END OF FUNCTION
```

## THE ROUTINE LOCATE.

```
             EXTERNAL FUNCTION
             ENTRY TO LOCATE.
   REM    R  THIS ROUTINE LOCATES THE VALUE OF Z(J) AT WHICH
   REM    R  T1(Z(J),J)=0 FOR ALL VALUES OF J, THE PAIR OF INTEGERS
   REM    R  NEAREST TO Z(J) AND THE EXTREME VALUES OF THESE INTEGERS.
   REM    R
             PMINI=P
             PMAXI=0
             THROUGH THIRD, FOR J=1,1, J.G.Q
             INI=CB(J)
             THROUGH FIRST, FOR I=2,1, I.G.P
             SK= INI+I
   REM    R  FOR EACH J FIND THE INTERVAL I,I-1 WHERE T1 CHANGES SIGN.
             WHENEVER T1(SK)*T1(SK-1).LE.0.0
              BO2=I
              BO1=BO2-1.0
              BO3=BO2+1.0
              INI= SK-1
              W=(BO1+BO2)/2.0
              ROOT.(BO1,T1(INI),BO2,T1(SK),BO3,T1(SK+1),W)
              W1=RHOL
              BO3=BO2-2.0
              ROOT.(BO1,T1(INI),BO2,T1(SK),BO3,T1(SK-1),W)
              Z(J)=(W1+RHOL)/2.0
              ZPREDI(J)=Z(J)
              ZHFPR(J)=Z(J)
              AZ1(J)=Z(J)-EPS
              AZ2(J)=ZPREDI(J)+EPS+1.0
              TRANSFER TO SECOND
             END OF CONDITIONAL
FIRST
             EXECUTE NOTLOC.
SECOND
             WHENEVER AZ1(J).L.PMINI, PMINI=AZ1(J)
             WHENEVER AZ2(J).G.PMAXI, PMAXI=AZ2(J)
THIRD
             EXECUTE FINLOC.
             FUNCTION RETURN
             INTEGER AZ1,AZ2,CB,INI,I,J,PMAXI,PMINI,P,Q,SK
   REM    R  FLOATING POINT BO1,BO2,BO3,EPS,RHOL,T1,W,W1,ZHFPR,ZPREDI,Z
   REM    R  FUNCTION NAME FINLOC,LOCATE,NOTLOC,ROOT
   REM    R  PROGRAM COMMON AZ1,AZ2,CB,EPS,PMAXI,PMINI,P,Q,RHOL,
   REM    R                 ZHFPR,ZPREDI,Z
           PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
          1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
          2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
          3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
          4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
          5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
          6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
          7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
            INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
            BOOLEAN INSW
             END OF FUNCTION
```

## THE INPUT - OUTPUT ROUTINES.

```
              EXTERNAL FUNCTION
              REFERENCES ON
REM     R
REM     R     THIS ROUTINE IS USED FOR ALL INPUT-OUTPUT FUNCTIONS. SINCE
REM     R     THE CONTROLL VARIABLES ARE USED MAINLY TO CONTROL THESE
REM     R     I/O FUNCTIONS, THEIR MEENING IS GIVEN HERE.
REM     R
REM     R     INSW          1B                              0B
REM     R       0     P,Q NEW OR CHANGED          P,Q NOT CHANGED
REM     R       1     FACTOR.NE.1.0, RESTCF.      RESTCF IS NOT EXECUTED
REM     R       2     T1=0.0                      T1 INPUT
REM     R       3     EXEC. SETCF.                DO NOT EXEC. SETCF. AFTER
REM     R             AFTER INPUT                 INPUT
REM     R       4     AFTER TTRAP SET INSW(5)     AFTER TTRAP INSW(5)
REM     R                                         REMAINS UNCHANGED
REM     R       5     NO TTRAP OR INSW(4)=0B      TTRAP OCCURED OR INPUT SET
REM     R             NO PUNCHIG UNLESS BY 24     PUNCH RESULTS
REM     R       6     EXEC. LOCATE                DO NOT EXEC. LOCATE.
REM     R       7     T.L.TMOVE WHEN TESTED       T.G.TMOVE
REM     R       8     S IS GIVEN                  DT IS GIVEN
REM     R       9                                 ONLY S OR DT CHANGED IN INP.
REM     R      10     TDUMP.L.T (SWOUT(0))
REM     R      11     TPOINT.L.T (SWOUT(1))
REM     R      12     PRINT Z WHEN INSW(11)       DO NOT PRINT
REM     R      13     PRINT Z WHEN INSW(10)       DO NOT PRINT
REM     R      14     PRINT ZPREDI IF INSW(11)    DO NOT PRINT
REM     R      15     PRINT ZPREDI IF INSW(10)    DO NOT PRINT
REM     R      16     PRINT AZ1 IF INSW(11)       DO NOT PRINT
REM     R      17     PRINT AZ1 IF INSW(10)       DO NOT PRINT
REM     R      18     PRINT TJ1 IF INSW(11)       DO NOT PRINT
REM     R      19     PRINT TJ1 IF INSW(11)       DO NOT PRINT
REM     R                PARTIALLY
REM     R      20     PRINT TJ1 IF INSW(10)       DO NOT PRINT
REM     R      21     PRINT TJQ IF INSW(11)       DO NOT PRINT
REM     R      22     PRINT TJQ IF INSW(11)       DO NOT PRINT
REM     R                PARTIALLY
REM     R      23     PRINT TJQ IF INSW(10)       DO NOT PRINT
REM     R      24     NO PUNCHIG UNLESS BY 5      PUNCH RESULTS
REM     R      25                                 START AGAIN
REM     R
REM     R
REM     R           SWOUT            PRINTED
REM     R           VALUE        VARIABLE NO(S).
REM     R             1              1+2+3
REM     R             2              1+2
REM     R             3              1+3
REM     R             4              1
REM     R             5              2+3
REM     R             6              2
REM     R             7              3
REM     R             8              NONE
REM     R
```

```
             ENTRY TO INPUT.
   REM    R  SUBRUTINE INPUT HANDLES ALL THE DATA NEEDED.
   REM    R
             WHENEVER T.E.0.0, PRINT COMMENT $1$
             READ AND PRINT DATA
             SWST(0)=SWOUT(0)
             SWST(1)=SWOUT(1)
             FUNCTION RETURN
             ENTRY TO NOTLOC.
   REM    R  IN SUBRUTINE LOCATE THE BOUNDARY WAS NOT LOCATED.
             ERROR RETURN
             ENTRY TO FINLOC.
   REM    R  IN SUBRUTINE LOCATE THE BOUNDARY WAS SUCCESSFULLY LOCATED.
             WHENEVER Q.L.10
              CNT=1
             OTHERWISE
              CNT=2
             END OF CONDITIONAL
             SK=20-CNT
             PRINT FORMAT ZHD, Q, Z(1)...Z(Q)
             SK=10-CNT
             PRINT FORMAT ZPRHD, Q, ZPREDI(1)...ZPREDI(Q)
              I=(Q+1)/2
              THROUGH COMP1, FOR J=2, 1, J.G.I
              ZDEVIA(J)=ZPREDI(J)-ZPREDI(1)
COMP1
              PRINT FORMAT ZDVHD,  I,  ZDEVIA(2)...ZDEVIA(I)
             PRINT FORMAT AZ1HD, Q, Q, AZ1(1)...AZ1(Q),AZ2(1)...AZ2(Q)
             FUNCTION RETURN
   REM    R  OUT. OR OUT2 ARE USED AT THE END OF EACH COMPUTATIONAL
   REM    R  CYCLE, THEY ARE USED TO RESET THE THE T-S AND Z-S.
   REM    R
             ENTRY TO OUT.
             SW=0
             TRANSFER TO STLB(3)
             ENTRY TO OUT2.
             SW=1
STLB(3)
             WHENEVER  TDUMP.L.T
              INSW(10)=1B
              TDUMP=TDUMP+DTDMP
              PRINT FORMAT TFORM, T
             OTHERWISE
              INSW(10)=0B
             END OF CONDITIONAL
             WHENEVER TPOINT.L.T
              INSW(11)=1B
              TPOINT=TPOINT+DTPNT
              WHENEVER .NOT.INSW(10), PRINT FORMAT TFORM, T
             OTHERWISE
              INSW(11)=0B
             END OF CONDITIONAL
             WHENEVER INSW(10)
              INI=P+CB(Q)
              CNT=PQN
              SK=18-CNT
              WHENEVER SWOUT(0).NE.8
              WHENEVER SWOUT(0).L.5
               PRINT FORMAT CT1, INI, T1(0)...T1(INI)
               SWOUT(0)=SWOUT(0)+4
              END OF CONDITIONAL
              WHENEVER SWOUT(0).L.7
               PRINT FORMAT CT2, INI, T2(0)...T2(INI)
               SWOUT(0)=SWOUT(0)+2
              END OF CONDITIONAL
              WHENEVER SWOUT(0).E.7
               PRINT FORMAT CT3, INI, T3(0)...T3(INI)
              END OF CONDITIONAL
              SWOUT(0)=SWST(0)
              TRANSFER TO STLB(SW)
              END OF CONDITIONAL
STLB(1)
              CNT=QCN
              WHENEVER INSW(13)
               SK=20-CNT
              PRINT FORMAT ZHD, Q, Z(1)...Z(Q)
              END OF CONDITIONAL
              WHENEVER INSW(15)
               SK=10-CNT
              PRINT FORMAT ZPRHD, Q, ZPREDI(1)...ZPREDI(Q)
              I=(Q+1)/2
              THROUGH COMP3, FOR J=2, 1, J.G.I
              ZDEVIA(J)=ZPREDI(J)-ZPREDI(1)
COMP3
              PRINT FORMAT ZDVHD,  I,  ZDEVIA(2)...ZDEVIA(I)
              END OF CONDITIONAL
              WHENEVER INSW(17), PRINT FORMAT AZ1HD, Q, Q,
          1        AZ1(1)...AZ1(Q),AZ2(1)...AZ2(Q)
STLB
               CNT=PCN
               SK=16-CNT
               WHENEVER INSW(20)
               PRINT FORMAT TJ1HD, P, TJ1(1)...TJ1(P)
               END OF CONDITIONAL
               WHENEVER INSW(23)
               PRINT FORMAT TJQHD, P, TJQ(1)...TJQ(P)
               END OF CONDITIONAL
             END OF CONDITIONAL
             WHENEVER INSW(11)
              WHENEVER SWOUT(1).NE.8
              WHENEVER SWOUT(1).L.5
               STBOUN(4)=ST1
               PRINT FORMAT STBOUN, (J=1, 1, J.G.DEXMAX,
          1                  T1(DEX(2*J-1)+CB(DEX(2*J))) )
               SWOUT(1)=SWOUT(1)+4
              END OF CONDITIONAL
              WHENEVER SWOUT(1).L.7
               STBOUN(4)=ST2
               PRINT FORMAT STBOUN, (J=1, 1, J.G.DEXMAX,
          1                  T2(DEX(2*J-1)+CB(DEX(2*J))) )
```

```
               SWOUT(1)=SWOUT(1)+2
              END OF CONDITIONAL
              WHENEVER SWOUT(1).E.7
               STBOUN(4)=ST3
               PRINT FORMAT STBOUN, (J=1, 1, J.G.DEXMAX,
          1                 T3(DEX(2*J-1)+CB(DEX(2*J))) )
              END OF CONDITIONAL
              SWOUT(1)=SWST(1)
              END OF CONDITIONAL
              WHENEVER  (SW.E.0), TRANSFER TO STLB(2)
              CNT=QCN
              WHENEVER INSW(12)
               SK=20-CNT
              PRINT FORMAT ZHD, Q, Z(1)...Z(Q)
              END OF CONDITIONAL
              WHENEVER INSW(14)
               SK=10-CNT
              PRINT FORMAT ZPRHD, Q, ZPREDI(1)...ZPREDI(Q)
              I=(Q+1)/2
              THROUGH COMP2, FOR J=2, 1, J.G.I
              ZDEVIA(J)=ZPREDI(J)-ZPREDI(1)
COMP2
              PRINT FORMAT ZDVHD,  I,  ZDEVIA(2)...ZDEVIA(I)
              END OF CONDITIONAL
              WHENEVER INSW(16), PRINT FORMAT AZ1HD, Q, Q,
          1        AZ1(1)...AZ1(Q),AZ2(1)...AZ2(Q)
STLB(2)
               CNT=PCN
               SK=16-CNT
               WHENEVER INSW(18)
                PRINT FORMAT TJ1HD, P, TJ1(1)...TJ1(P)
               END OF CONDITIONAL
               WHENEVER INSW(21)
                PRINT FORMAT TJQHD, P, TJQ(1)...TJQ(P)
               END OF CONDITIONAL
              WHENEVER INSW(22).OR.INSW(19)
               WHENEVER TJMIN.L.10
                CNT=1
               OR WHENEVER TJMIN.L.100
                CNT=2
               OTHERWISE
                CNT=3
               END OF CONDITIONAL
               WHENEVER TJMAX.L.10
                CNT1=1
               OR WHENEVER TJMAX.L.100
                CNT1=2
               OTHERWISE
                CNT1=3
               END OF CONDITIONAL
               SK=17-CNT-CNT1
               WHENEVER INSW(19)
                PRINT FORMAT MTJ1,TJMIN,TJMAX,TJ1(TJMIN)...TJ1(TJMAX)
               END OF CONDITIONAL
               WHENEVER INSW(22)
                PRINT FORMAT MTJQ,TJMIN,TJMAX,TJQ(TJMIN)...TJQ(TJMAX)
               END OF CONDITIONAL
              END OF CONDITIONAL
             END OF CONDITIONAL
   REM    R  REPLACE THE OLD VALUE OF T1 WITH THE NEWLY COMPUTED T3.
             INI=P+CB(Q)
             THROUGH FIRST, FOR J=0, 1, J.G.INI
             T1(J)=T3(J)
             WHENEVER J.LE.Q
              Z(J)=ZPREDI(J)
             END OF CONDITIONAL
FIRST
            WHENEVER (.NOT.INSW(5)).OR.(.NOT.INSW(24))
             INSW(24)=1B
             INI=P+CB(Q)
             CNT=(INI+1)/4
             PUNCH FORMAT PNCH1,T,TMOVE,N,TT1MOV,PMAXI,PMINI,
          1        T1(0)...T1(INI)
             CNT=(Q-3)/4
             SK=Q-3-4*CNT
             CNT1=Q
             PUNCH FORMAT PNCH2,Z(1)...Z(Q),AZ1(1)...AZ1(Q),
          1        AZ2(1)...AZ2(Q)
             WHENEVER .NOT.INSW(5),EXECUTE ERROR.
            END OF CONDITIONAL
             FUNCTION RETURN
             ENTRY TO OUT1.
             PRINT RESULTS COEF1,COEF2,COEF3,COEF4,CON(1),CON(2),RO(1),
          1        RO(2),C(1),C(2),H1,H2,H(1),H(2),G(1),G(2),P,Q,L1,L2,
          2        LATFSN,DT,S,RSQ,L(1),L(2),BI(1),BI(2),BU(1),BU(2),
          3        BII(1),BII(2)
             CNT=QCN
             RHOL=P-1
             RHOL=(CON*DT*RHOL*RHOL)/(RO*C*L2*L2)
             PRINT FORMAT  SEE,RHOL,Q,CB(1)...CB(Q)
             FUNCTION RETURN
             REFERENCES OFF
   REM    R
   REM    R  THESE ARE THE FORMATS USED BY THE ABOVE I/O STATEMENTS.
   REM    R
             VECTOR VALUES TFORM=$1H0/S11,5HTIME=,1PE16.8*$
             VECTOR VALUES CT1=$1H0,S'SK',11HT1(0)...T1(,I'CNT',2H)=,1P6
          1E16.8/(1P8E16.8)*$
             VECTOR VALUES CT2=$1H0,S'SK',11HT2(0)...T2(,I'CNT',2H)=,1P6
          1E16.8/(1P8E16.8)*$
             VECTOR VALUES CT3=$1H0,S'SK',11HT3(0)...T3(,I'CNT',2H)=,1P6
          1E16.8/(1P8E16.8)*$
             VECTOR VALUES ZHD=$1H0,S'SK',9HZ(1)...Z(,I'CNT',2H)=,1P6E16
          1.8/(1P8E16.8)*$
             VECTOR VALUES ZPRHD=$1H0,S'SK',19HZPREDI(1)...ZPREDI(,I'CNT
          1',2H)=,1P6E16.8/(1P8E16.8)*$
             VECTOR VALUES ZDVHD=$1H0,S'SK',19HZDEVIA(2)...ZDEVIA(,I'CNT
          1',2H)=,1P6E16.8/(1P8E16.8)*$
```

```
        VECTOR VALUES TJ1HD=$1H0,S'SK',13HTJ1(1)...TJ1(,I'CNT',2H)=
      1,1P6E16.8/(1P8E16.8)*$
        VECTOR VALUES TJQHD=$1H0,S'SK',13HTJQ(1)...TJQ(,I'CNT',2H)=
      1,1P6E16.8/(1P8E16.8)*$
        VECTOR VALUES STBOUN=$ 1H0, S13,  18HSELECTED T1SOL(J)=,1P6
      1E16.8/(1P8E16.8)*$
        VECTOR VALUES AZ1HD=$14H0AZ1(1)...AZ1(,I'CNT',2H),,
      113HAZ2(1)...AZ2(,I'CNT',2H)=,20(I3,1H,)/30(I3,1H,)*$
        VECTOR VALUES ST1=$ T1(I,$
        VECTOR VALUES ST2=$ T2(I,$
        VECTOR VALUES ST3=$ T3(I,$
        VECTOR VALUES MTJ1=$1H0,S'SK'4HTJ1(,I'CNT'8H)...TJ1(,I'CNT1
      1',2H)=,1P6E16.8/(1P8E16.8)*$
        VECTOR VALUES MTJQ=$1H0,S'SK'4HTJQ(,I'CNT'8H)...TJQ(,I'CNT1
      1',2H)=,1P6E16.8/(1P8E16.8)*$
        VECTOR VALUES PNCH1=$2HT=,1PE15.8,8H, TMOVE=,1PE15.8,1H,/
      12HN=,1PE15.8,8H,TT1MOV=, 1PE15.8,7H,PMAXI=,I3,7H,PMINI=,I3,
      21H,/S11,6HT1(0)=,3(1PE16.8,1H,)/'CNT'(4(1PE16.8,1H,)/)*$
        VECTOR VALUES PNCH2=$S12,5HZ(1)=,3(1PE16.8,1H,)/
      1'CNT'(4(1PE16.8,1H,)/),'SK'(1PE16.8,1H,)/7HAZ1(1)=,'CNT1'(I3,
      21H,)/7HAZ2(1)=,'CNT1'(I3,1H,)*$
        VECTOR VALUES SEE=$1H0/H.0 ONE OF THE DIMENSIONLESS RATIOS
      1 (DELTA T)/(DELTA X)*(DELTA X)  IS .,1PE16.8/ H.0 THE LINEAR
      2SUBSCRIPT OF THE TEMPERATURES ARE COMPUTED BY THE EXPRESSION
      3( CB(J)+I ), WHERE. /H=0 CB(1)...CB(=,I'CNT',2H)=,20(I4,1H,)/
      41H0,25(I4,1H,)*$
        DIMENSION DEX(100),PT(20),SWOUT(1),SWST(1),ZDEVIA(16)
        INTEGER AZ1HD, AZ1,AZ2,CB,CNT1,CNT,CT1,CT2,CT3,DEXMAX,DEX,
      1            INI,I,J,MTJ1,MTJQ,PCN,PMAXI,PMINI,PNCH1,PNCH2,PQN,
      2            P,QCN,Q,SEE,SK,ST1,ST2,ST3,STBOUN,SWOUT,SW,SWST,
      3            TFORM,TJ1HD,TJMAX,TJMIN,TJQHD,ZDVHD,ZHD,ZPRHD
        FORMAT VARIABLE CNT1,CNT,SK
REM   R STATEMENT LABEL COMP1,COMP2,COMP3,STLB,FIRST
REM   R FUNCTION NAME ERROR,FINLOC,INPUT,NOTLOC,OUT,OUT1,OUT2
REM   R FLOATING POINT BI,BII,BU,COEF1,COEF2,COEF3,COEF4,
REM   R                    CON,C,DTDMP,DTPNT,DT,G,H1,H2,H,L1,L2,L,
REM   R                    LATFSN,N,RHOL,RO,RSQ,S,T1,T2,T3,TDUMP,TJ1,
REM   R                    TJQ,TMOVE,TPOINT,T,TT1MOV,ZDEVIA,ZPREDI,Z
        EQUIVALENCE (CC1,EPS1),(TT1MOV,ZZ(1))
       PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
      1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
      2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
      3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
      4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
      5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
      6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
      7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
       INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
       BOOLEAN INSW
       END OF FUNCTION
```

## THE ROUTINES SETCF. AND RESTCF.

```
                EXTERNAL FUNCTION
                REFERENCES ON
                ENTRY TO SETCF.
   REM     R
   REM     R    SUBRUTINE SETCF. COMPUTES THE VALUES OF THE VARIOUS
   REM     R    CONSTANTS
   REM     R    SUBRUTINE RESTCF. RECOMPUTES THE VALUES OF THOSE CONSTANTS
   REM     R    WHICH DEPEND ON FACTOR ( DT ) .
   REM     R
                WHENEVER INSW
                 THROUGH ANN, FOR J=1, 1, J.G.Q
ANN              CB(J)=J*P-P-1
                  WHENEVER P.L.10
                   PCN=1
                  OR WHENEVER P.L.100
                   PCN=2
                  OTHERWISE
                   PCN=3
                  END OF CONDITIONAL
                  WHENEVER (Q.L.10)
                   QCN=1
                  OTHERWISE
                   QCN=2
                  END OF CONDITIONAL
                 INI=P+CB(Q)
                 WHENEVER (INI.L.10)
                  PQN=1
                 OR WHENEVER (INI.L.100)
                  PQN=2
                 OR WHENEVER (INI.L.1000)
                  PQN=3
                 OTHERWISE
                  PQN=4
                 END OF CONDITIONAL
                 COEF3=(P-1)/L2
                 COEF3=COEF3*COEF3
                 COEF4=(Q-1)/L1
                 COEF4=COEF4*COEF4
                 RSQ=COEF4/COEF3
                END OF CONDITIONAL
                X1=L1*L1*RO(1)*C(1)
                WHENEVER INSW(8)
                 DT=X1/(CON(1)*S)
                OTHERWISE
                 S=X1/(CON(1)*DT)
                END OF CONDITIONAL
                WHENEVER INSW(9)
                TEMPDF=THIGH-TMELT
                TL=(TLOW-TMELT)/TEMPDF
                COEF2=TEMPDF/(LATFSN*RO)
                COEF1=CON(1)*COEF2
                COEF2=CON(2)*COEF2
```

```
          W1=H1/CON(1)
          W2=H2/CON(2)
          TE1=L1/(Q-1)
          TE2=L2/(P-1)
          G(1)=W1*TE1
          G(2)=W2*TE1
          H(1)=W1*TE2
          H(2)=W2*TE2
          FACTOR=1.0
          L=L1*L1*COEF3
          BO3 = (RO(1)*C(1)*CON(2))/(RO(2)*C(2)*CON(1))
          BO1=-2.0*RSQ*(1.0+G(1))
          BO2=-2.0*RSQ*(1.0+G(2))
          END OF CONDITIONAL
REM    R                                                    RESTCF.
          ENTRY TO RESTCF.
          WHENEVER FACTOR .E.1.0, INSW(1)=0B
          L(1)=S*FACTOR/L
          L(2)=L(1)/BO3
          TWOL(1)=2.0*L(1)
          TWOL(2)=2.0*L(2)
          BU(1)=-2.0*(RSQ+L(1))
          BU(2)=-2.0*(RSQ+L(2))
          BUU(1) = BO1-2.0*L(1)
          BUU(2) = BO2-2.0*L(2)
          BI(1)=-2.0*(1.0+L(1))
          BI(2)=-2.0*(1.0+L(2))
          BII(1)=BI(1)-2.0*H(1)
          BII(2)=BI(2)-2.0*H(2)
          FUNCTION RETURN
          REFERENCES OFF
          ERASABLE J,X1,X2,X3,X4
          BOOLEAN INSW
          INTEGER CB,INI,J,PCN,PQN,P,QCN,Q
REM    R  STATEMENT LABEL ANN
REM    R  FUNCTION NAME SETCF,RESTCF
REM    R  FLOATING POINT BO1,BO2,BO3,BI,BII,BU,BUU,COEF1,COEF2,
REM    R                 COEF3,COEF4,CON,C,DT,FACTOR,G,H1,H2,H,L1,
REM    R                 L2,LATFSN,L,RO,RSQ,S,TE1,TE2,TEMPDF,THIGH,
REM    R                 TLOW,TL,TMELT,TWOL,W1,W2,X1
REM    R  PROGRAM COMMON CB,COEF1,COEF2,COEF3,COEF4,CON,C,DT,FACTOR,
REM    R                 G,H1,H2,H,INSW,J,L1,L2,LATFSN,L,PCN,PQN,P,
REM    R                 QCN,Q,RFOUR,RO,RSQ,S,TEMPDF,THIGH,TLOW,TL,
REM    R                 TMELT,TWOL
        PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
       1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
       2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
       3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
       4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
       5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
       6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
       7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
        INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
        BOOLEAN INSW
        END OF FUNCTION
```

THE ROUTINE ROOT.

```
            EXTERNAL FUNCTION (XA,YA,XB,YB,XC,YC,X4)
             REFERENCES ON
  REM     R
  REM     R SUBRUTINE COEF FINDS THE COEFFICIENTS TO THE EQUATION
  REM     R Y=AA*X*X+BB*X+CC USING THE THREE PAIR OF COORDINATES GIVEN.
  REM     R
  REM     R SUBRUTINE ROOT FINDS THE ROOT CLOSEST TO X4. ( RHOL=X,Y=0)
  REM     R
            ENTRY TO ROOT.
            SW=1
            TRANSFER TO ANN
            ENTRY TO COEF.
            SW=2
ANN         X1=XA
            X2=XB
            X3=XC
            Y1=YA
            Y2=YB
            Y3=YC
            X12=X1-X2
            X13=X1-X3
            X23=X2-X3
            WHENEVER   (.ABS.(X12)).L.A4.OR.(.ABS.(X23)).L.A4.OR.
          2            (.ABS.(X13)).L.A4.OR.       (.ABS.(((Y1-Y2)*X23
          1   /((Y2-Y3)*X12))-1.0)).L.0.001,      TRANSFER TO ANDYS
            AA=Y1/(X12*X13)-Y2/(X12*X23)+Y3/(X13*X23)
            BB=Y1*(X2+X3)/(-X12*X13)+Y2*(X1+X3)/(X12*X23)-Y3*(X1+X2)/
          1 (X13*X23)
            CC=Y1*X2*X3/(X12*X13)-Y2*X1*X3/(X12*X23)+Y3*X1*X2/(X13*X23
            WHENEVER SW.E.2, FUNCTION RETURN SW
            WHENEVER(.ABS.(AA)).L.(1.0E-10),       TRANSFER TO ANDYS
            X2 = (BB*BB-4.0*AA*CC)
            WHENEVER  X2.L.0.0,  TRANSFER TO ANDYS
            X3=-BB/(2.0*AA)
            X2= (SQRT.( X2 ) )/(2.0*AA)
            WHENEVER (.ABS.(X3+X2-X4 )).L.(.ABS.(X3-X2-X4 ))
            RHOL=X3+X2
            OTHERWISE
            RHOL=X3-X2
            END OF CONDITIONAL
            FUNCTION RETURN SW
ANDYS       WHENEVER (.ABS.X13).G.A4
            BB=(Y1-Y3)/X13
            OTHERWISE
            BB=(Y1-Y2)/X12
            END OF CONDITIONAL
          AA=0.0
          CC=Y1-BB*X1
          WHENEVER SW.E.2
          SW=3
          FUNCTION RETURN SW
          END OF CONDITIONAL
          RHOL=-CC/BB
          SW=4
          FUNCTION RETURN SW
          VECTOR VALUES A4=1.0E-5
           REFERENCES OFF
          INTEGER SW
          ERASABLE  X12,X13,X1,X23,Y1,Y2,Y3
REM     R
REM     R ***  THE SQRT. FUNCTION USES THE FIRST 2 ERASABLES (X12,X13)
REM     R
REM     R FUNCTION NAME COEF,ROOT,SQRT
REM     R STATEMENT LABEL ANDYS,ANN
REM     R FLOATING POINT A4,AA,BB,CC,RHOL,X12,X13,X1,X23,X2,X3,X4,XA,
REM     R                    XB,XC,Y1,Y2,Y3,YA,YB,YC
REM     R PROGRAM COMMON AA,BB,CC,RHOL
REM     R
         PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
        1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
        2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
        3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
        4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
        5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
        6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
        7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
         INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
         BOOLEAN INSW
         END OF FUNCTION
```

## THE MAIN ROUTINE.

```
          REFERENCES ON
  REM   R
  REM   R THIS IS THE MAIN PROGRAM.
  REM   R SET TIME TRAP AT 20 SEC. BEFORE SYSTEM TRAP TO  HERE.
  REM   R
          TIMER=0
  REM   R                                           TIMLFT.
          TIMELF=TIMLFT.(TIMER)-20.0
          TIMER=1
          WHENEVER TIMELF.G.0.0
  REM   R                                           TTRAP.
           EXECUTE TTRAP.(TIMER,TIMELF,HERE)
          END OF CONDITIONAL
  REM   R
  REM   R INITALIZE VARIABLES
  REM   R
          VECTOR VALUES INSW(0)...INSW(49)=1B
ANDYS
          TH=1.0
          TSTART=0.0
          T=0.0
  REM   R                                           INPUT.
          EXECUTE INPUT.
          EXECUTE SETCF.
          EXECUTE OUT1.
  REM   R                                           BNDSET.
          BNDSET.
          WHENEVER INSW(2)
           PMINI=2*P
           PMAXI=0
           THROUGH HOME, FOR J=1, 1, J.G.Q
           AZ1(J)=PMINI
           AZ2(J)=0
           INI=CB(J)
           THROUGH HOME, FOR I=1, 1, I.G.P
           T1(I+INI)=0.0
HOME
          OTHERWISE
  REM   R                                           INPUT.
           EXECUTE INPUT.
           INSW(2)=1B
           WHENEVER .NOT.INSW(5)
            INSW(5)=1B
            WHENEVER INSW(3)
             SETCF.
             OUT1.
             INSW(3)=0B
            END OF CONDITIONAL
            TRANSFER TO SECOND
           END OF CONDITIONAL
          END OF CONDITIONAL
```

```
SECOND
             WHENEVER T.L.TT1MOV
              WHENEVER T.G.INTIME
   REM    R                                                        INPUT.
               EXECUTE INPUT.
               WHENEVER .NOT.INSW(25), TRANSFER TO ANDYS
               WHENEVER INSW(3)
                EXECUTE SETCF.
                EXECUTE OUT1.
                INSW(3)=0B
               END OF CONDITIONAL
              END OF CONDITIONAL
   REM    R                                                        T2SOLV.
              EXECUTE T2SOLV.
   REM    R                                                        T3SOLV.
              EXECUTE T3SOLV.
              N=N+1.0
              T=TSTART+N*DT
               OUT.
              TRANSFER TO SECOND
             END OF CONDITIONAL
             INSW(7)=0B
   REM    R                                                        LOCATE.
             WHENEVER INSW(6), EXECUTE LOCATE.
THIRD
             WHENEVER T.L.TMOVE
              WHENEVER T.G.INTIME
   REM    R                                                        INPUT.
               INPUT.
               WHENEVER .NOT.INSW(25), TRANSFER TO ANDYS
               WHENEVER INSW(3)
                SETCF.
                OUT1.
                INSW(3)=0B
               END OF CONDITIONAL
              END OF CONDITIONAL
   REM    R                                                        MOVEM.
              MOVEM.
   REM    R                                                        RESTCF.
              WHENEVER INSW(1), EXECUTE RESTCF.
              N=N+1.0/FACTOR
              T=TSTART+N*DT
   REM    R                                                        T2SOLV.
              T2SOLV.
   REM    R                                                        T3SOLV.
              T3SOLV.
              OUT2.
              TRANSFER TO THIRD
             END OF CONDITIONAL
RUN
             WHENEVER T.L.TEND
              WHENEVER T.G.INTIME
   REM    R                                                        INPUT.
               EXECUTE INPUT.
               WHENEVER .NOT.INSW(25), TRANSFER TO ANDYS
               WHENEVER INSW(3)
                EXECUTE SETCF.
                EXECUTE OUT1.
                INSW(3)=0B
               END OF CONDITIONAL
              END OF CONDITIONAL
   REM    R                                                     MOVEM.
              EXECUTE MOVEM.
   REM    R                                                     RESTCF.
              WHENEVER INSW(1), EXECUTE RESTCF.
              N=N+1.0/FACTOR
              T=TSTART+N*DT
   REM    R                                                     BOUND.
              EXECUTE BOUND.
   REM    R                                                     T2SOLV.
              EXECUTE T2SOLV.
   REM    R                                                     T3SOLV.
              EXECUTE T3SOLV.
              EXECUTE OUT2.
              TRANSFER TO RUN
             END OF CONDITIONAL
             TRANSFER TO ANDYS
HERE
             TIMELF=0.0
   REM    R                                                     TTRAP.
             EXECUTE TTRAP.(TIMER,TIMELF,RUN)
             WHENEVER INSW(4)
              INSW(4)=0B
              INSW(5)=0B
             END OF CONDITIONAL
             EXECUTE TRTN.(TIMER)
             TRANSFER TO ANDYS
             REFERENCES OFF
             INTEGER AZ1,AZ2,CB,INI,I,J,PMAXI,PMINI,P,Q,TIMER
   REM    R  FUNCTION NAME BNDSET,BOUND,INPUT,LOCATE,MOVEM,OUT1,OUT2,
   REM    R                OUT,RESTCF,SETCF,T2SOLV,T3SOLV,TIMLFT,TRTN,
   REM    R                TTRAP
   REM    R  STATEMENT LABEL ANDYS,HERE,HOME,RUN,SECOND,THIRD
   REM    R  FLOATING POINT DT,FACTOR,INTIME,N,T1,TEND,TH,TIMELF,TMOVE,
   REM    R                 T,TSTART,TT1MOV
   REM    R  PROGRAM COMMON CB,DT,FACTOR,INI,INSW,INTIME,I,J,N,PMINI,P,
   REM    R                 Q,T1,TEND,TH,TIMELF,TMOVE,T,TT1MOV
             EQUIVALENCE (TT1MOV,ZZ(1))
           PROGRAM COMMON AA1,AA,AZ1(30),AZ2(30),BB1,BB,BII(2),BI(2),
          1BU(2),BUU(2),CB(31),CC1,CC,COEF1,COEF2,COEF3,COEF4,CON(2),
          2C(2),DT,EPS,FACTOR,G(2),H1,H2,H(2),INSW(50),INTIME,L1,
          3L2,LATFSN,L(2),MXST,N,PCN,PMAXI,PMINI,PQN,P,QCN,Q,RHOL,
          4RO(2),RSQ,SLP1,SLP2,S,T1(3000),T2(3000),T3(3000),TEMPDF,
          5TEND,THIGH,TH,THSTRT,TI1(0),TIMELF,TIP(0),TJ1(300),
          6TJQ(300),TLOW,TL,TMELT,TMOVE,T,TSTART,TWOL(2),VELOC,
          7WIDTH,ZHFPR(30),ZPREDI(30),Z(30),ZZ(200)
           INTEGER AZ1,AZ2,PCN,PMAXI,PMINI,PQN,P,QCN,Q
           BOOLEAN INSW
           END OF FUNCTION
```

# LIST OF SYMBOLS.

| SYMBOL | BRIEF DESCRIPTION |
|---|---|
| A | VARIABLE USED IN T2SOLV AND T3SOLV. SEE TEXT. |
| A4 | = 1.0E-5, A CONSTANT USED IN ROOT. |
| AA | A COEFFICIENT IN Y=AA*X*X+BB*X+CC CURVE FITTING. |
| AA1 | EXISTS IN PROGRAM COMMON ONLY. |
| ANDYS | A LOCAL STATEMENT LABEL IN SEVERAL ROUTINES. |
| ANN | A LOCAL STATEMENT LABEL IN SEVERAL ROUTINES. |
| AZ1 | THE LARGEST INTEGER SMALLER THEN Z-EPS. |
| AZ1HD | FORMAT USED TO PRINT AZ1 AND AZ2 |
| AZ2 | THE SMALLEST INTEGER LARGER THEN Z+EPS. |
| BB | A COEFFICIENT IN Y=AA*X*X+BB*X+CC CURVE FITTING. |
| BB1 | EXISTS IN PROGRAM COMMON ONLY. |
| BI | = -2-2*L, A CONSTANT (AS LONG AS DT IS CONSTANT). |
| BII | = -2-2*L-2*H, A CONSTANT (IF DT IS CONSTANT). |
| BNDSET | THIS ROUTINE SETS UP THE INITAL BOUNDARY CONDITIONS. |
| BO1 | A LOCAL VARIABLE IN SEVERAL ROUTINES. |
| BO2 | A LOCAL VARIABLE IN SEVERAL ROUTINES. |
| BO3 | A LOCAL VARIABLE IN SEVERAL ROUTINES. |
| BOUND | RESETS THE TIME DEPENDENT BOUNDARY CONDITIONS. |
| BT | VARIABLE USED IN T2SOLV AND T3SOLV. SEE TEXT. |
| BU | = -2*RSQ-2*L, A CONSTANT (IF DT IS CONSTANT). |
| BUU | = -2*RSQ-2*L-2*RSQ*G, A CONSTANT (IF DT IS CONSTANT). |
| C | HEAT CAPACITY (CGS UNITS). |
| CB | IN SUBSCRIPTING INSTEAD OF T1(I,J) USE T1(I+CB(J)). |
| CC | A COEFFICIENT IN Y=AA*X*X+BB*X+CC CURVE FITTING. |
| CC1 | EQUIVALENT TO EPS1. |
| CHANGE | A STATEMENT LABEL USED IN T2SOLV AND T3SOLV. |
| CNT | A FORMAT VARRABCE USED BY THE I/O ROUTINES. |
| CNT1 | A FORMAT VARIABLE USED BY THE I/O ROUTINES. |
| COEF | PART OF ROOT. COMPUTES AA, BB AND CC ONLY. |
| COEF1 | A COEFFICIENT USED IN ROUTINE MOVEM. |
| COEF2 | A COEFFICIENT USED IN ROUTINE MOVEM. |
| COEF3 | A COEFFICIENT USED IN ROUTINE MOVEM. |
| COEF4 | A COEFFICIENT USED IN ROUTINE MOVEM. |
| COMP1 | A STATEMENT LABEL USED IN THE I/O ROUTINES. |
| COMP2 | A STATEMENT LABEL USED IN THE I/O ROUTINES. |
| COMP3 | A STATEMENT LABEL USED IN THE I/O ROUTINES. |
| CON | THERMAL CONDUCTIVITY (CGS UNITS). |
| CT1 | FORMAT USED TO PRINT ALL THE T1(I,J). |
| CT2 | FORMAT USED TO PRINT ALL THE T2(I,J). |
| CT3 | FORMAT USED TO PRINT ALL THE T3(I,J). |
| D | VARIABLE USED IN T2SOLV AND T3SOLV. SEE TEXT. |
| DELTA | THE NUMBER OF INTERVALS/INCH |
| DEX | SUBSCRIPTS OF THE SELECTED TEMP.-S TO BE PRINTED |
| DEXMAX | THE NUMBER OF THE SELECTED TEMP.-S TO BE PRINTED |
| DL1 | A STATEMENT LABEL USED IN T2SOLV AND T3SOLV. |
| DONE | A STATEMENT LABEL USED IN T2SOLV AND T3SOLV. |
| DT | TIME INTERVAL USED TO INCREMENT T. SEE FACTOR. |
| DTDMP | TIME INTERVAL USED TO INCREMENT TDUMP. |
| DTPNT | TIME INTERVAL USED TO INCREMENT TPOINT. |
| EPS | A CONSTANT USED IN COMPUTING AZ1 AND AZ2. |
| EPS1 | A CONSTANT USED IN COMPUTING SLP1 AND SLP2. |

| SYMBOL | BRIEF DESCRIPTION |
|---|---|
| ERROR | A SYSTEM SUBROUTINE USED FOR ERROR RETURNS TO SYSTEM. |
| FACTOR | USED TO REDUCE DT TEMPORARILY IF NECCESSARY. |
| FINLOC | OUTPUT ROUTINE USED IF BOUNDARY IS FOUND |
| FIRST | A STATEMENT LABEL USED IN MANY ROUTINES |
| G | =(HI*L1)/(CON(I)*(Q-1)) |
| GM | VARIABLE USED IN T2SOLV AND T3SOLV. SEE TEXT. |
| H | =(HI*L2)/(CON(I)*(P-1)). |
| H1 | HEAT TRANSFER COEFFICIENT IN REGION 1 (CGS UNITS). |
| H2 | HEAT TRANSFER COEFFICIENT IN REGION 2 (CGS UNITS). |
| HERE | A LOCAL STATEMENT LABEL IN SEVERAL ROUTINES. |
| HOLT | = -33/20 |
| HOME | A LOCAL STATEMENT LABEL IN SEVERAL ROUTINES. |
| I | USUALLY THE GRID INDEX IN THE X DIRECTION. |
| IFL | FLOATING POINT VALUE OF I |
| INCR | A LOCAL VARIABLE USED IN T2SOLV AND T3SOLV. |
| INI | AN INTEGER USED IN MANY ROUTINES. |
| INPUT | ROUTINE USED TO READ DATA. |
| INSW | CONTROL SWITCH. SEE I/O ROUTINE PROGRAM LISTING. |
| INTIME | IF LESS THEN T, DATA IS READ. |
| IT1 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE FIRST. |
| IT2 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE FIRST. |
| IT3 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE FIRST. |
| IT4 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE FIRST. |
| IT5 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE FIRST. |
| IT0 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE FIRST. |
| J | USUALLY THE GRID INDEX IN THE Y DIRECTION. |
| JFL | FLOATING POINT VALUE OF J. |
| JVCT | = CB(J), USED IN T2SOLV AND T3SOLV. |
| L | DIMENSIONLESS DT/(DX*DX) OR DT/(DY*DY). |
| L1 | LENGTH OF BAR, CM. |
| L2 | WIDTH OF BAR, CM. |
| LATFSN | LATENT HEAT GIVEN UP AT THE REGION BOUNDARY, CAL/GM. |
| LOCATE | ROUTINE USED TO FIND THE LOCATION OF THE BOUNDARY. |
| MAX | A VARIABLE USED IN T2SOLV AND T3SOLV. |
| MIN | A VARIABLE USED IN T2SOLV AND T3SOLV. |
| MOVEM | USED TO COMPUTE THE MOVEMENT OF THE BOUNDARY. |
| MOVM | THE DISTANCE MOVED SINCE TIME=TMOVE |
| MTJ1 | FORMAT USED TO PRINT SOME OF THE TJ1(J). |
| MTJQ | FORMAT USED TO PRINT SOME OF THE TJQ(J). |
| MVMTRT | VELOCITY OF THE BOUNDARY AT POINT J. |
| MXMVRT | THE LARGEST MVMTRT. |
| MXST | LARGEST ALLOWED BOUNDARY MOVEMENT/TIME STEP. |
| N | USED TO KEEP TRACK OF TIME (T). SEE TSTART. |
| NEXT | A STATEMENT LABEL USED IN T2SOLV. |
| NOTLOC | USED WHEN LOCATE FAILS TO FIND THE BOUNDARY. |
| OTHER | A STATEMENT LABEL USED IN T2SOLV AND T3SOLV |
| OUT | ROUTINE USED TO PRINT RESULTS |
| OUT1 | ROUTINE USED TO PRINT INITAL VALUES OF CONSTANTS |
| OUT2 | ROUTINE USED TO PRINT RESULTS |
| OVER | A LOCAL STATEMENT LABEL IN SEVERAL ROUTINES. |
| P | THE NUMBER OF GRID POINTS IN THE X DIRECTION. |
| PCN | THE NUMBER OF DIGITS IN P |
| PHASE | USED TO INDICATE REGION NUMBER. |
| PMAXI | EQUAL TO THE LARGEST AZ2(J). |
| PMINI | EQUAL TO THE SMALLEST AZ1(J). |

| SYMBOL | BRIEF DESCRIPTION |
|---|---|
| PNCH1 | FORMAT USED IN PUNCHING RESULTS |
| PNCH2 | FORMAT USED IN PUNCHING Z, AZ1 AND AZ2 |
| PQN | THE NUMBER OF DIGITS IN P+CB(Q) |
| Q | THE NUMBER OF GRID POINTS IN THE Y DIRECTION. |
| QCN | THE NUMBER OF DIGITS IN Q |
| RESTCF | A ROUTINE USED TO RECOMPUTE CONSTANTS IF FACTOR.NE.1 |
| RET | A STATEMENT LABEL USED IN T3SOLV. |
| RHOL | THE ROOT CLOSEST TO X4. |
| RO | DENSITY, GM/CC. |
| ROOT | USED TO FIND A ROOT OF THE QUADRATIC FITTING CURVE. |
| RSQ | = ((L2*(Q-1))/(L1*(P-1))).P.2, OR (DX/DY).P.2. |
| RUN | A STATEMENT LABEL IN T2SOLV AND IN THE MAIN PROGRAM. |
| S | DIMENSIONLESS 1/DT, =(L1*L1*RO*C)/(CON*DT). |
| SECOND | A LOCAL STATEMENT LABEL IN SEVERAL ROUTINES. |
| SEE | A FORMAT USED IN PRINTING. |
| SETCF | ROUTINE USED TO COMPUTE VARIOUS CONSTANTS. |
| SK | A FORMAT VARIABLE IN THE I/O ROUTINES. |
| SL | A LOCAL VARIABLE USED IN T2SOLV AND T3SOLV. |
| SL1 | A VARIABLE USED IN THE BOUNDARY ROUTINE. |
| SL3 | A VARIABLE USED IN THE BOUNDARY ROUTINE. |
| SL5 | A VARIABLE USED IN THE BOUNDARY ROUTINE. |
| SLP1 | SLOPE IN REGION 1, USED TO FIND MVMTRT. |
| SLP2 | SLOPE IN REGION 2, USED TO FIND MVMTRT. |
| SLO | A VARIABLE USED IN THE BOUNDARY ROUTINE. |
| SQRT | THE SQUARE ROOT SYSTEM SUBROUTINE. |
| ST1 | FORMAT USED TO MODIFY STBOUN. |
| ST2 | FORMAT USED TO MODIFY STBOUN. |
| ST3 | FORMAT USED TO MODIFY STBOUN. |
| STBOUN | FORMAT USED TO PRINT SELECTED TEMPERATURES. |
| STEP | USED IN COMPUTING FACTOR IN THE ROUTINE MOVEM. |
| STLB | STATEMENT LABELS USED IN THE I/O ROUTINES. |
| SW | A CONTROL SWITH USED IN THE I/O AND ROOT ROUTINES. |
| SWOUT | USED TO CONTROL WHICH TEMPERATURES ARE TO BE PRINTED. |
| SWST | USED TO PRESERVE SWOUT-S ORIGINAL VALUE. |
| T | TIME, SECONDS. |
| T1 | DIMENSIONLESS TEMPERATURES AT TIME T. |
| T2 | DIMENSIONLESS TEMPERATURES AT TIME T+DT/2. |
| T2SOLV | ROUTINE USED TO COMPUTE T2(I,J). |
| T3 | DIMENSIONLESS TEMPERATURES AT TIME T+DT. |
| T3COMP | AN INTERNAL ROUTINE IN T3SOLV. |
| T3SOLV | ROUTINE USED TO COMPUTE T3(I,J). |
| TDUMP | IF LESS THEN T, INSW(10) CONTROLLED PRINTING STARTS. |
| TE1 | A LOCAL VARIABLE USED MAINLY IN THE BOUNDARY ROUTINE. |
| TE2 | A LOCAL VARIABLE USED MAINLY IN THE BOUNDARY ROUTINE. |
| TE3 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE. |
| TE4 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE. |
| TE5 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE. |
| TE6 | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE. |
| TEMPDF | = THIGH-TMELT |
| TEND | IF LESS THEN T, COMPUTATION IS TERMINATED. |
| TEO | A LOCAL VARIABLE USED IN THE BOUNDARY ROUTINE. |
| TFORM | FORMAT USED IN PRINTING THE TIME. |
| TH | HIGH REFERENCE TEMPERATURE (DIMENSIONLESS), =1. |
| THIGH | HIGH REFERENCE TEMPERATURE. |
| THIRD | A LOCAL STATEMENT LABEL IN SEVERAL ROUTINES. |

| SYMBOL | BRIEF DESCRIPTION |
|---|---|
| THIS | A STATEMENT LABEL USED IN T2SOLV AND T3SOLV. |
| THSTRT | DETERMINES RELATIVE POSITION OF BAR AND SURROUNDING. |
| TI1 | WALL TEMP. (DIM.LES) WHEN I=1 |
| TIMELF | AMOUNT OF USABLE TIME BEFORE TTRAP. TRAPS |
| TIMER | THE CURRENTLY USED CLOCK CELL NO. |
| TIMLFT | GIVES THE AMOUNT OF TIME LEFT BEFORE TRAP WILL OCCUR |
| TIP | WALL TEMP. (DIM.LES) WHEN I=P |
| TJ1 | WALL TEMP. (DIM.LES) WHEN J=1 |
| TJ1HD | FORMAT USED TO PRINT ALL TJ1(J)-S |
| TJMAX | LAST TJ1 AND TJQ TO BE PRINTED IN PARTIAL PRINTING |
| TJMIN | FIRST TJ1 AND TJQ TO BE PRINTED IN PARTIAL PRINTING |
| TJQ | WALL TEMP. (DIM.LES) WHEN J=Q |
| TJQHD | FORMAT USED TO PRINT ALL TJQ(J)-S |
| TL | WALL TEMP. (DIM.LES) AT THE COLD END |
| TLOW | WALL TEMPERATURE AT THE COLD END |
| TMELT | THE TEMPERATURE AT THE INTERFACE. |
| TMOVE | BAR STARTS TO MOVE AT THIS TIME. |
| TPOINT | IF LESS THEN T, INSW(11) CONTROLLED PRINTING STARTS. |
| TRTN | USED TO RETURN TO THE PROGRAM INTERRUPTED BY A TRAP. |
| TSTART | REFERENCE TIME. T=TSTART+N*DT |
| TSUB | TEMPORARY STORAGE IN T2SOLV AND T3SOLV |
| TT1MOV | IF GREATER THEN T, THE BAR CONTAINS ONLY 1 REGION |
| TTRAP | ROUTINE USED FOR TIME TRAPING |
| TWOL | = 2*L, A CONSTANT (AS LONG AS DT IS CONSTANT). |
| VELOC | VELOCITY, INTERVALS/SEC |
| W | A LOCAL VARIABLE IN SEVERAL ROUTINES. |
| W1 | A LOCAL VARIABLE IN SEVERAL ROUTINES. |
| W2 | A LOCAL VARIABLE IN SEVERAL ROUTINES. |
| WIDTH | LENGTH OF THE BOUNDARY TEMPERATURE TRANSITION ZONE. |
| X1 | =XA |
| X12 | =XA-XB |
| X13 | =XA-XC |
| X2 | TEMPORARY STORAGE IN ROOT |
| X23 | =XB-XC |
| X3 | TEMPORARY STORAGE IN ROOT |
| X4 | IF THERE ARE 2 ROOTS, THE ONE CLOSER TO X4 IS RHOL |
| XA | THE X COMPONENT OF THE FIRST POINT USED BY ROOT |
| XB | THE X COMPONENT OF THE SECOND POINT USED BY ROOT |
| XC | THE X COMPONENT OF THE THIRD POINT USED BY ROOT |
| Y1 | =YA |
| Y2 | =YB |
| Y3 | =YC |
| YA | THE Y COMPONENT OF THE FIRST POINT USED BY ROOT |
| YB | THE Y COMPONENT OF THE SECOND POINT USED BY ROOT |
| YC | THE Y COMPONENT OF THE THIRD POINT USED BY ROOT |
| Z | LOCATION OF THE INTERFACE AT TIME T (INTERVAL UNITS) |
| ZDEVIA | = ZPREDI(J)-ZPREDI(1) |
| ZDVHD | FORMAT USED TO PRINT ZDEVIA |
| ZHD | FORMAT USED TO PRINT Z |
| ZHFPR | PREDICTED VALUE OF Z AT TIME T+DT/2 |
| ZPREDI | PREDICTED VALUE OF Z AT TIME T+DT |
| ZPRHD | FORMAT USED TO PRINT ZPREDI |
| ZZ | AVAILABLE FOR MODIFIED ROUTINES THROUGH EQUIVALENCE |
| ZZZ | NOT A VARIABLE, USED TO SAVE 7 ERASABLE LOCATIONS |

# APPENDIX B

## SAMPLE DATA SETS

THE FIRST DATA SET (USED IN RUN 21)

```
$DATA
T=0,N=0,TSTART=0,TT1MOV=49.9,TMOVE=999,TEND=9900.1,INTIME=996,
TDUMP=-0.000001, TPOINT= 99.99, DTDMP=100.0, DTPNT=10.0,
INSW(8)=0B,INSW(12)=0B,0B,INSW(16)=0B,0B,0B,0B,1B,0B,0B,0B,
SWOUT(0)=7,8, EPS=0.00001, EPS1=0.1, MXST=0.05,
DT=2.5,P=89,Q=9,L1=2.0,L2=22.0,H1=0.024,H2=0.024,VELOC=0.00206,
RO=5.8,5.8,5.8, CON=0.018,0.018,0.20, C=0.055,0.055,0.0689655,
THIGH=565.0, TMELT=525.0, TLOW=446.0, THSTRT=44.6, LATFSN=50*
INSW(3)=0B, INSW(24)=0B, INTIME=2196.0*
INSW(3)=0B, INSW(24)=0B, INTIME=2201.0*
```

THE SECOND DATA SET (USED IN RUN 4)

```
$DATA
T=0,N=0,TSTART=0,TT1MOV=49.9,TMOVE=1999,TEND=9900.1,INTIME=3994,
TDUMP=1999.999, TPOINT=1999.999, DTDMP=500, DTPNT=10,
INSW(8)=0B,INSW(12)=0B,0B,INSW(16)=0B,0B,0B,0B,1B,0B,0B,0B,
SWOUT(0)=7,8,EPS=0.00001,EPS1=0.1,MXST=0.05,INSW(2)=0B,INSW(6)=0B,
DT=5.0,P=45,Q=5,L1=2.0,L2=22.0,H1=0.008,H2=0.008,VELOC=0.000515,
RO=5.8,5.8,5.8, CON=0.018,0.018,0.20, C=0.055,0.055,0.0689655,
THIGH=565.0, TMELT=525.0, TLOW=446.0, THSTRT=22.8, LATFSN=100*
T=2000.00,  N=400.00,  PMAXI=19,  PMINI=18,
T1(0)= -1.82896416E 00, -1.81319176E 00, -1.79230173E 00,
 -1.76705405E 00, -1.73734686E 00, -1.70254382E 00, -1.66148441E 00,
 -1.61236973E 00, -1.55251122E 00, -1.47772430E 00, -1.38040116E 00,
 -1.24186139E 00, -1.07804120E 00, -8.99005234E-01, -7.10290384E-01,
 -5.15797436E-01, -3.18680227E-01, -1.21689732E-01,  6.49085975E-02,
  2.33676806E-01,  3.90716144E-01,  5.28569895E-01,  6.34510505E-01,
  6.94243526E-01,  7.23632258E-01,  7.27280098E-01,  7.04304820E-01,
  6.45005888E-01,  5.57937008E-01,  4.58545601E-01,  3.57119510E-01,
  2.64022177E-01,  1.95635755E-01,  1.52749200E-01,  1.25266974E-01,
  1.09358713E-01,  1.03432034E-01,  1.07530946E-01,  1.24870247E-01,
  1.49732517E-01,  1.78843664E-01,  2.09878865E-01,  2.40767205E-01,
  2.69240886E-01,  2.91890621E-01, -1.82154840E 00, -1.80572905E 00,
 -1.78489430E 00, -1.75916447E 00, -1.72825140E 00, -1.69137219E 00,
 -1.64716540E 00, -1.59355813E 00, -1.52756757E 00, -1.44504289E 00,
 -1.34051023E 00, -1.20835969E 00, -1.05350426E 00, -8.82315755E-01,
 -7.00115722E-01, -5.11227202E-01, -3.19332936E-01, -1.27838776E-01,
  5.30243492E-02,  2.13283691E-01,  3.60099146E-01,  4.86527032E-01,
  5.84514630E-01,  6.48263043E-01,  6.81228143E-01,  6.86046934E-01,
  6.63327050E-01,  6.12616032E-01,  5.39779848E-01,  4.54736412E-01,
   3.67213506E-01,  2.86675760E-01,  2.22012874E-01,  1.75596799E-01,
   1.44533329E-01,  1.26239622E-01,  1.19105613E-01,  1.22416916E-01,
   1.35867846E-01,  1.56699061E-01,  1.82154287E-01,  2.09791422E-01,
   2.37350777E-01,  2.62583274E-01,  2.83179972E-01, -1.81928734E 00,
  -1.80328184E 00, -1.78238213E 00, -1.75645827E 00, -1.72512230E 00,
  -1.68752839E 00, -1.64224729E 00, -1.58713044E 00, -1.51915881E 00,
  -1.43437050E 00, -1.32823855E 00, -1.19756472E 00, -1.04530296E 00,
  -8.76641262E-01, -6.96634209E-01, -5.09669018E-01, -3.19594061E-01,
  -1.30050312E-01,  4.88591146E-02,  2.06324774E-01,  3.49862427E-01,
   4.72919005E-01,  5.68753839E-01,  6.33062798E-01,  6.66968054E-01,
   6.72350717E-01,  6.50339133E-01,  6.02350360E-01,  5.33829242E-01,
   4.53406143E-01,  3.70321938E-01,  2.93454185E-01,  2.30142948E-01,
   1.83091629E-01,  1.51029789E-01,  1.31960736E-01,  1.24333811E-01,
   1.27163206E-01,  1.39485128E-01,  1.59041564E-01,  1.83282979E-01,
   2.09781757E-01,  2.36261192E-01,  2.60561454E-01,  2.80818069E-01,
  -1.82154840E 00, -1.80572905E 00, -1.78489430E 00, -1.75916447E 00,
  -1.72825140E 00, -1.69137219E 00, -1.64716540E 00, -1.59355813E 00,
  -1.52756758E 00, -1.44504289E 00, -1.34051023E 00, -1.20835969E 00,
  -1.05350426E 00, -8.82315767E-01, -7.00115728E-01, -5.11227202E-01,
  -3.19332933E-01, -1.27838756E-01,  5.30243742E-02,  2.13283700E-01,
   3.60099146E-01,  4.86527032E-01,  5.84514630E-01,  6.48263043E-01,
   6.81228143E-01,  6.86046934E-01,  6.63327056E-01,  6.12616032E-01,
   5.39779854E-01,  4.54736412E-01,  3.67213506E-01,  2.86675760E-01,
   2.22012874E-01,  1.75596799E-01,  1.44533329E-01,  1.26239623E-01,
   1.19105615E-01,  1.22416916E-01,  1.35867846E-01,  1.56699061E-01,
   1.82154287E-01,  2.09791425E-01,  2.37350777E-01,  2.62583274E-01,
   2.83179972E-01, -1.82896416E 00, -1.81319176E 00, -1.79230171E 00,
  -1.76705402E 00, -1.73734687E 00, -1.70254384E 00, -1.66148444E 00,
  -1.61236973E 00, -1.55251122E 00, -1.47772430E 00, -1.38040116E 00,
  -1.24186139E 00, -1.07804118E 00, -8.99005210E-01, -7.10290378E-01,
  -5.15797436E-01, -3.18680221E-01, -1.21689723E-01,  6.49086058E-02,
   2.33676812E-01,  3.90716153E-01,  5.28569907E-01,  6.34510517E-01,
   6.94243526E-01,  7.23632258E-01,  7.27280098E-01,  7.04304820E-01,
   6.45005888E-01,  5.57937008E-01,  4.58545601E-01,  3.57119510E-01,
   2.64022177E-01,  1.95635758E-01,  1.52749200E-01,  1.25266974E-01,
   1.09358713E-01,  1.03432035E-01,  1.07530947E-01,  1.24870247E-01,
   1.49732518E-01,  1.78843664E-01,  2.09878865E-01,  2.40767205E-01,
   2.69240886E-01,  2.91890621E-01,
 Z(1)=  1.86238843E 01,  1.86793494E 01,  1.87000467E 01,
   1.86793491E 01,  1.86238843E 01,
 AZ1(1)=18,18,18,18,18,   AZ2(1)=19,19,19,19,19*
 INSW(3)=0B, INSW(24)=0B, INTIME=5991.0*
 INSW(3)=0B, INSW(24)=0B, INTIME=7991.0*
 INSW(3)=0B, INSW(24)=0B, INTIME=8001.0*
```

## EXPLANATION

### The First Data Set

In the first data set the first seven lines of information are read at the beginning of the computation. The value of the variables read at this time regulate the calculations carried out by the program. The first line starts out with information stating that the computation will start at time T=0 (T=0, N=0, TSTART=0). The first mode will be used until T becomes 50 (TTIMOV=49.9). The second mode will be used until T becomes 1000 (TMOVE=999). The computation will continue in the third mode until T exceeds 9900.1 (=TEND). The program will read in more information when T will exceed INTIME=996. This will occur at T=997.5, and will influence the input - output subroutine at the end of the computational cycle, when T will already have the value 1000.

Information is printed whenever T exceeds TDUMP and INSW(10) activates one set of print statements, or T exceeds TPOINT and INSW(11) activates another set of print statements. T will exceed TDUMP at the end of the first computational cycle, as TDUMP = -0.000001 originally. After INSW(10) is set to activate the print statements, TDUMP is incremented by DTDMP (=100). INSW(10) becomes activated again when T becomes 100. At this time INSW(11) also becomes activated and both TDUMP and TPOINT are incremented, the former by DTDMP, the latter by DTPNT (=10).

Next, some of the other control variables are set. At the beginning of the computation all the INSW's are IB unless otherwise set by the data. The information conveyed in the third line states that in the data DT is available, that Z, AZ1, AZ2, and TJQ should not be printed at any time, that the printing of TJ1 should not be activated by INSW(11). The individual INSW's are defined in the listing of the input - output subroutine, in Appendix A.

In the fourth line, SWOUT(0) is 7, indicating that whenever INSW(10) controls the printing the temperature distribution given by T3 is printed; while SWOUT(1) has the value 8, indicating that whenever INSW(11) controls the printing, the temperature distribution is not printed in any form. Next the radius of an interface point is set: EPS=0.00001 interval units. EPS1 (=0.1 interval) is used in determining which grid points are used to calculate the temperature gradient at the interface. MXST (=0.05 interval) is the maximum movement of any x-interface point in one time interval.

In the fifth line the interval sizes are set: DT=2.5 sec; there are 89 grid points (P) on a 22 cm (L2) long x-grid line and 9 grid points (Q) on a 2 cm (L1) long y-grid line. The heat transfer coefficient at the surface of each subregion is 0.024 (H1,H2) and the relative velocity of the bar and its surroundings is 0.00206 intervals/second (VELOC).

In the sixth line the material properties are set. RO is 5.8 throughout the region, while CON is 0.018 in $R_1$ and 0.020 in $R_2$. The heat capacity C is 0.055 in $R_1$ and 0.0689655 in $R_2$ (this unusual value was used to compare some results with earlier computations where the RO·C product was taken as 0.4).

In the seventh line the latent heat of fusion (LATFSN=50) is given along with the reference temperatures and THSTRT. This completes the data received by the program at the time it starts the computation.

When T is 997.5, the program will attempt to read in more data, and the eighth line of information is now received. This tells the program that SETCF· will not have to be executed, that the program should read in more information when T exceeds 2196 and that a set of cards should be punched. The cards punched contain information needed to restart the computation using the calculated values available at this time as an initial condition. When T is 2197.5, the ninth line is read. This line is similar to the eighth line; it directs the program to read the next data set at T=2002.5. Since no data are available at that time, however, the computation is terminated.

## The Second Data Set

The second data set starts out much like the first one; in the first seven lines there are only two major differences, in the fourth line INSW(2) and INSW(6) are set 0B. This tells the program that an initial solution is available and the interface location will not have to be determined by LOCATE. Later, when this initial condition is read, the next 61 lines of information become available to the program. The information obtained overrides some of the first-hand information, i.e., T=2000, N=400 indicates that the calculation will start not at

T=0, but at T=2000. The rest of the data give the location of the interface (Z) and related variables (AZ1, AZ2, PMIN1, PMAX1) along with the temperature distribution T1. The temperatures are arranged in the following order: (1,1)...(P,1),(1,2)...(P,2),...,(1,Q)...(P,Q). The last three lines of information are data to be read as the computation progresses as in the first data set.

# APPENDIX C

## SAMPLE OUTPUT

A portion of the printed output from run 21 is reproduced on the next few pages as a sample.

```
 3.58544171E 01  3.58335495E 01  3.58123887E 01

          ZDEVIA(2)...ZDEVIA(5)=  2.11601257E-02  4.20279503E-02  5.68408966E-02  6.21366501E-02

            TIME=  9.39999998E 02

          ZPREDI(1)...ZPREDI(9)=  3.58121526E 01  3.58332476E 01  3.58540475E 01  3.58688223E 01  3.58741069E 01  3.58688229E 01
 3.58540481E 01  3.58332479E 01  3.58121526E 01

          ZDEVIA(2)...ZDEVIA(5)=  2.10947990E-02  4.18949127E-02  5.66697121E-02  6.19540215E-02

            TIME=  9.50000000E 02

          ZPREDI(1)...ZPREDI(9)=  3.58119220E 01  3.58329514E 01  3.58536851E 01  3.58684230E 01  3.58736956E 01  3.58684233E 01
 3.58536857E 01  3.58329520E 01  3.58119220E 01

          ZDEVIA(2)...ZDEVIA(5)=  2.10294724E-02  4.17633057E-02  5.65009117E-02  6.17737770E-02

            TIME=  9.59999990E 02

          ZPREDI(1)...ZPREDI(9)=  3.58116961E 01  3.58326602E 01  3.58533296E 01  3.58680308E 01  3.58732924E 01  3.58680314E 01
 3.58533299E 01  3.58326605E 01  3.58116961E 01

          ZDEVIA(2)...ZDEVIA(5)=  2.09641457E-02  4.16336060E-02  5.63349724E-02  6.15963936E-02

            TIME=  9.69999993E 02

          ZPREDI(1)...ZPREDI(9)=  3.58114746E 01  3.58323741E 01  3.58529815E 01  3.58676466E 01  3.58728966E 01  3.58676469E 01
 3.58529818E 01  3.58323744E 01  3.58114746E 01

          ZDEVIA(2)...ZDEVIA(5)=  2.08992958E-02  4.15067673E-02  5.61718941E-02  6.14218712E-02

            TIME=  9.79999995E 02

          ZPREDI(1)...ZPREDI(9)=  3.58112577E 01  3.58320931E 01  3.58526409E 01  3.58672699E 01  3.58725083E 01  3.58672702E 01
 3.58526415E 01  3.58320937E 01  3.58112577E 01

          ZDEVIA(2)...ZDEVIA(5)=  2.08353996E-02  4.13832664E-02  5.60121536E-02  6.12506866E-02

            TIME=  9.89999998E 02

          ZPREDI(1)...ZPREDI(9)=  3.58110455E 01  3.58318171E 01  3.58523062E 01  3.58669007E 01  3.58721280E 01  3.58669013E 01
 3.58523068E 01  3.58318174E 01  3.58110455E 01

          ZDEVIA(2)...ZDEVIA(5)=  2.07715034E-02  4.12607193E-02  5.58552742E-02  6.10823631E-02

            TIME=  1.00000000E 03

                T3(0)...T3(800)= -1.86464913E 00 -1.85712150E 00 -1.84798650E 00 -1.83797231E 00 -1.82734521E 00 -1.81621626E 00
-1.80462764E 00 -1.79258034E 00 -1.78004335E 00 -1.76695542E 00 -1.75322275E 00 -1.73871416E 00 -1.72325337E 00 -1.70660767E 00
-1.68847172E 00 -1.66844280E 00 -1.64597996E 00 -1.62032758E 00 -1.59034862E 00 -1.55408438E 00 -1.50733308E 00 -1.43811627E 00
-1.35402288E 00 -1.26204219E 00 -1.16497543E 00 -1.06429113E 00 -9.60914946E-01 -8.55497003E-01 -7.48523217E-01 -6.43370607E-01
-5.31338960E-01 -4.21669501E-01 -3.11552575E-01 -2.01116130E-01 -9.03571284E-02  1.91356757E-02  1.21117920E-01  2.22687513E-01
```

```
  3.22873726E-01  4.20830011E-01  5.15598434E-01  6.05920970E-01  6.89884579E-01  7.63903624E-01  8.19009554E-01  8.52094519E-01
  8.73526978E-01  8.86911809E-01  8.93751562E-01  8.94637001E-01  8.89543653E-01  8.77744234E-01  8.57163811E-01  8.21835601E-01
  7.66100436E-01  7.00791866E-01  6.30109489E-01  5.56337005E-01  4.81101632E-01  4.05811429E-01  3.31896633E-01  2.61083370E-01
  1.95992929E-01  1.42069788E-01  1.09658267E-01  8.72630954E-02  7.09381229E-02  5.89176917E-02  5.02374077E-02  4.43136531E-02
  4.07941681E-02  3.95061386E-02  4.04625970E-02  4.39571470E-02  5.09283429E-02  6.44207764E-02  8.14683771E-02  1.00604475E-01
  1.21131320E-01  1.42619342E-01  1.64755547E-01  1.87285106E-01  2.09981546E-01  2.32625636E-01  2.54983401E-01  2.76772264E-01
  2.97589293E-01  3.16714734E-01  3.32446891E-01 -1.86118445E 00 -1.85372788E 00 -1.84499191E 00 -1.83531466E 00 -1.82489780E 00
 -1.81384625E 00 -1.80220233E 00 -1.78996287E 00 -1.77708568E 00 -1.76349008E 00 -1.74905355E 00 -1.73360506E 00 -1.71691510E 00
 -1.69868164E 00 -1.67850983E 00 -1.65588322E 00 -1.63011865E 00 -1.60029215E 00 -1.56510766E 00 -1.52266437E 00 -1.47012664E 00
 -1.40395764E 00 -1.32613945E 00 -1.23989892E 00 -1.14754330E 00 -1.05063570E 00 -9.50276625E-01 -8.47277379E-01 -7.42258060E-01
 -6.35705620E-01 -5.28010386E-01 -4.19490159E-01 -3.10404468E-01 -2.00958219E-01 -9.12933850E-02  1.67659204E-02  1.16219376E-01
  2.14724120E-01  3.11368415E-01  4.05169535E-01  4.94976532E-01  5.79331368E-01  6.56255931E-01  7.22944003E-01  7.75723058E-01
  8.13549280E-01  8.39609003E-01  8.56257963E-01  8.64854240E-01  8.66017556E-01  8.59759808E-01  8.45458043E-01  8.21686876E-01
  7.86148852E-01  7.37803912E-01  6.80330527E-01  6.16829139E-01  5.49638867E-01  4.80656409E-01  4.11591589E-01  3.44166845E-01
  2.80326524E-01  2.22514260E-01  1.73934917E-01  1.37547572E-01  1.10406925E-01  9.00672877E-02  7.49229234E-02  6.39303631E-02
  5.64056134E-02  5.19166905E-02  5.02357364E-02  5.13323307E-02  5.54059684E-02  6.29462278E-02  7.46274298E-02  8.95422268E-02
  1.06787601E-01  1.25706868E-01  1.45817369E-01  1.66741432E-01  1.88162670E-01  2.09797227E-01  2.31371123E-01  2.52597085E-01
  2.73143652E-01  2.92585233E-01  3.10318732E-01  3.25472221E-01 -1.85933085E 00 -1.85161361E 00 -1.84293857E 00 -1.83339658E 00
 -1.82308500E 00 -1.81206851E 00 -1.80037253E 00 -1.78798313E 00 -1.77484642E 00 -1.76086572E 00 -1.74589631E 00 -1.72973736E 00
 -1.71212040E 00 -1.69269390E 00 -1.67100279E 00 -1.64646158E 00 -1.61831926E 00 -1.58561461E 00 -1.54712546E 00 -1.50133866E 00
 -1.44655137E 00 -1.38144812E 00 -1.30667824E 00 -1.22387056E 00 -1.13466284E 00 -1.04043138E 00 -9.42278111E-01 -8.41077292E-01
 -7.37525403E-01 -6.32182586E-01 -5.25505757E-01 -4.17874956E-01 -3.09615219E-01 -2.01016754E-01 -9.23608768E-02  1.45271964E-02
  1.12272167E-01  2.08623284E-01  3.02707976E-01  3.93504533E-01  4.79807884E-01  5.60172874E-01  6.32864457E-01  6.95893955E-01
  7.47389841E-01  7.86770862E-01  8.15102053E-01  8.33657038E-01  8.43390059E-01  8.44819331E-01  8.38020098E-01  8.22641325E-01
  7.97976887E-01  7.63269085E-01  7.18635863E-01  6.65897238E-01  6.07237726E-01  5.44732934E-01  4.80293560E-01  4.15731686E-01
  3.52852637E-01  2.93541634E-01  2.39802676E-01  1.93608055E-01  1.56190187E-01  1.26749749E-01  1.04001187E-01  8.67763567E-02
  7.41555119E-02  6.54617500E-02  6.02312475E-02  5.81877863E-02  5.92250216E-02  6.33881694E-02  7.08231688E-02  8.16004860E-02
  9.52854991E-02  1.11296840E-01  1.29091182E-01  1.48201834E-01  1.68230142E-01  1.88826914E-01  2.09673584E-01  2.30464548E-01
  2.50890157E-01  2.70620012E-01  2.89289269E-01  3.06502748E-01  3.21912909E-01 -1.85837395E 00 -1.85045709E 00 -1.84175219E 00
 -1.83224812E 00 -1.82197708E 00 -1.81097022E 00 -1.79923612E 00 -1.78675070E 00 -1.77345107E 00 -1.75922991E 00 -1.74392852E 00
 -1.72732760E 00 -1.70913516E 00 -1.68897069E 00 -1.66634572E 00 -1.64064093E 00 -1.61108209E 00 -1.57672143E 00 -1.53644083E 00
 -1.48901336E 00 -1.43329191E 00 -1.36860499E 00 -1.29525456E 00 -1.21424197E 00 -1.12680581E 00 -1.03414850E 00 -9.37326634E-01
 -8.37227678E-01 -7.34582877E-01 -6.29992592E-01 -5.23953843E-01 -4.16887099E-01 -3.09162700E-01 -2.01130190E-01 -9.31569171E-02
  1.29855871E-02  1.09721847E-01  2.04791301E-01  2.97337613E-01  3.86334294E-01  4.70578843E-01  5.48688811E-01  6.19135845E-01
  6.80378181E-01  7.31172007E-01  7.71042663E-01  8.00371623E-01  8.19878387E-01  8.30229819E-01  8.31849790E-01  8.24860036E-01
  8.09110379E-01  7.84310323E-01  7.50314814E-01  7.07573277E-01  6.57384872E-01  6.01491606E-01  5.41761583E-01  4.80053246E-01
  4.18188959E-01  3.57970405E-01  3.01184702E-01  2.49546772E-01  2.04504442E-01  1.66855370E-01  1.36400700E-01  1.12411357E-01
  9.40258074E-02  8.04535818E-02  7.10546279E-02  6.53587633E-02  6.30591404E-02  6.39918017E-02  6.80985159E-02  7.53577924E-02
  8.56658483E-02  9.87024426E-02  1.14023075E-01  1.31159186E-01  1.49668643E-01  1.69150390E-01  1.89241260E-01  2.09605634E-01
  2.29923323E-01  2.49878982E-01  2.69156978E-01  2.87449065E-01  3.04490077E-01  3.20146731E-01 -1.85807501E 00 -1.85008854E 00
 -1.84136488E 00 -1.83186652E 00 -1.82160497E 00 -1.81059909E 00 -1.79885092E 00 -1.78633238E 00 -1.77297722E 00 -1.75867432E 00
 -1.74326017E 00 -1.72650933E 00 -1.70812190E 00 -1.68770793E 00 -1.66476841E 00 -1.63867426E 00 -1.60864668E 00 -1.57374817E 00
 -1.53290297E 00 -1.48498191E 00 -1.42899778E 00 -1.36442520E 00 -1.29149280E 00 -1.21103665E 00 -1.12416959E 00 -1.03202967E 00
 -9.35651660E-01 -8.35923159E-01 -7.33584940E-01 -6.29249948E-01 -5.23428559E-01 -4.16555166E-01 -3.09016114E-01 -2.01181644E-01
 -9.34472382E-02  1.24412918E-02  1.08841665E-01  2.03485134E-01  2.95519018E-01  3.83917901E-01  4.67485744E-01  5.44868767E-01
  6.14612794E-01  6.75311583E-01  7.25877923E-01  7.65856630E-01  7.95463711E-01  8.15255344E-01  8.25801218E-01  8.27490163E-01
  8.20460021E-01  8.04629993E-01  7.79839295E-01  7.46106625E-01  7.03957635E-01  6.54577416E-01  5.99582297E-01  5.40768594E-01
  4.79968899E-01  4.19000536E-01  3.59655336E-01  3.03680003E-01  2.52695280E-01  2.08007577E-01  1.70326158E-01  1.39586323E-01
  1.15217742E-01  9.64619315E-02  8.25783753E-02  7.29444164E-02  6.70900381E-02  6.46982116E-02  6.55845290E-02  6.96562999E-02
  7.68436396E-02  8.70026708E-02  9.98353481E-02  1.14933835E-01  1.31853843E-01  1.50163160E-01  1.69461511E-01  1.89382102E-01
  2.09584373E-01  2.29744124E-01  2.49545488E-01  2.68679842E-01  2.86859933E-01  3.03861758E-01  3.19606939E-01 -1.85837395E 00
 -1.85045707E 00 -1.84175217E 00 -1.83224809E 00 -1.82197706E 00 -1.81097020E 00 -1.79923609E 00 -1.78675066E 00 -1.77345106E 00
 -1.75922990E 00 -1.74392851E 00 -1.72732760E 00 -1.70913516E 00 -1.68897070E 00 -1.66634573E 00 -1.64064093E 00 -1.61108209E 00
 -1.57672144E 00 -1.53644083E 00 -1.48901336E 00 -1.43329191E 00 -1.36860499E 00 -1.29525456E 00 -1.21424197E 00 -1.12680581E 00
 -1.03414848E 00 -9.37326622E-01 -8.37227666E-01 -7.34582871E-01 -6.29992586E-01 -5.23953843E-01 -4.16887105E-01 -3.09162709E-01
 -2.01130205E-01 -9.31569433E-02  1.29855466E-02  1.09721828E-01  2.04791296E-01  2.97337613E-01  3.86334294E-01  4.70578843E-01
  5.48688811E-01  6.19135845E-01  6.80378181E-01  7.31172013E-01  7.71042681E-01  8.00371623E-01  8.19878387E-01  8.30229819E-01
  8.31849790E-01  8.24860036E-01  8.09110391E-01  7.84310329E-01  7.50314820E-01  7.07573295E-01  6.57384884E-01  6.01491618E-01
  5.41761589E-01  4.80053252E-01  4.18188965E-01  3.57970408E-01  3.01184702E-01  2.49546772E-01  2.04504442E-01  1.66855370E-01
  1.36400700E-01  1.12411359E-01  9.40258074E-02  8.04535806E-02  7.10546279E-02  6.53587639E-02  6.30591416E-02  6.39918023E-02
  6.80985171E-02  7.53577936E-02  8.56658483E-02  9.87024426E-02  1.14023076E-01  1.31159186E-01  1.49668643E-01  1.69150390E-01
  1.89241260E-01  2.09605634E-01  2.29923323E-01  2.49878982E-01  2.69156978E-01  2.87449065E-01  3.04490377E-01  3.20146731E-01
 -1.85933085E 00 -1.85161363E 00 -1.84293860E 00 -1.83339661E 00 -1.82308501E 00 -1.81206851E 00 -1.80037253E 00 -1.78798313E 00
 -1.77484640E 00 -1.76086570E 00 -1.74589628E 00 -1.72973734E 00 -1.71212040E 00 -1.69269390E 00 -1.67100279E 00 -1.64646158E 00
 -1.61831926E 00 -1.58561459E 00 -1.54712546E 00 -1.50133866E 00 -1.44655137E 00 -1.38144809E 00 -1.30667822E 00 -1.22387056E 00
 -1.13466285E 00 -1.04043140E 00 -9.42278123E-01 -8.41077304E-01 -7.37525409E-01 -6.32182598E-01 -5.25505763E-01 -4.17874962E-01
 -3.09615228E-01 -2.01016772E-01 -9.23609066E-02  1.45271526E-02  1.12272139E-01  2.08623263E-01  3.02707964E-01  3.93504530E-01
  4.79807884E-01  5.60172886E-01  6.32864457E-01  6.95893961E-01  7.47389841E-01  7.86770862E-01  8.15102053E-01  8.33657062E-01
  8.43390059E-01  8.44819343E-01  8.38020098E-01  8.22641325E-01  7.97976893E-01  7.63269103E-01  7.18635875E-01  6.65897250E-01
  6.07237732E-01  5.44732934E-01  4.80293566E-01  4.15731686E-01  3.52852643E-01  2.93541640E-01  2.39802682E-01  1.93608056E-01
  1.56190187E-01  1.26749749E-01  1.04001187E-01  8.67763567E-02  7.41555119E-02  6.54617500E-02  6.02312475E-02  5.81877869E-02
  5.92250222E-02  6.33881694E-02  7.08231699E-02  8.16004872E-02  9.52855003E-02  1.11296842E-01  1.29091184E-01  1.48201838E-01
  1.68230146E-01  1.88826916E-01  2.09673584E-01  2.30464548E-01  2.50890160E-01  2.70620012E-01  2.89289269E-01  3.06502748E-01
  3.21912915E-01 -1.86118445E 00 -1.85372788E 00 -1.84499191E 00 -1.83531466E 00 -1.82489780E 00 -1.81384623E 00 -1.80220231E 00
 -1.78996284E 00 -1.77708566E 00 -1.76349007E 00 -1.74905354E 00 -1.73360503E 00 -1.71691510E 00 -1.69868162E 00 -1.67850983E 00
 -1.65588321E 00 -1.63011864E 00 -1.60029213E 00 -1.56510766E 00 -1.52266437E 00 -1.47012664E 00 -1.40395766E 00 -1.32613946E 00
 -1.23989893E 00 -1.14754330E 00 -1.05063570E 00 -9.50276637E-01 -8.47277379E-01 -7.42258072E-01 -6.35705626E-01 -5.28010398E-01
 -4.19490170E-01 -3.10404474E-01 -2.00958237E-01 -9.12934160E-02  1.67658789E-02  1.16219351E-01  2.14724106E-01  3.11368409E-01
  4.05169523E-01  4.94976526E-01  5.79331356E-01  6.56255931E-01  7.22943997E-01  7.75723052E-01  8.13549268E-01  8.39608991E-01
  8.56257963E-01  8.64854228E-01  8.66017556E-01  8.59759796E-01  8.45458031E-01  8.21686876E-01  7.86148858E-01  7.37803912E-01
  6.80330533E-01  6.16829145E-01  5.49638873E-01  4.80656409E-01  4.11591589E-01  3.44166845E-01  2.80326530E-01  2.22514260E-01
  1.73934919E-01  1.37547575E-01  1.10406926E-01  9.00672889E-02  7.49229246E-02  6.39303631E-02  5.64056134E-02  5.19166911E-02
  5.02357370E-02  5.13323307E-02  5.54059690E-02  6.29462278E-02  7.46274298E-02  8.95422280E-02  1.06787604E-01  1.25706869E-01
  1.45817371E-01  1.66741434E-01  1.88162671E-01  2.09797230E-01  2.31371123E-01  2.52597085E-01  2.73143652E-01  2.92585233E-01
  3.10318738E-01  3.25472224E-01 -1.86464912E 00 -1.85712147E 00 -1.84798646E 00 -1.83797228E 00 -1.82734519E 00 -1.81621625E 00
 -1.80462761E 00 -1.79258032E 00 -1.78004336E 00 -1.76695544E 00 -1.75322272E 00 -1.73871413E 00 -1.72325332E 00 -1.70660764E 00
 -1.68847169E 00 -1.66844277E 00 -1.64597993E 00 -1.62032755E 00 -1.59034860E 00 -1.55408436E 00 -1.50733306E 00 -1.43811622E 00
 -1.35402283E 00 -1.26204216E 00 -1.16497542E 00 -1.06429113E 00 -9.60914922E-01 -8.55496979E-01 -7.48523188E-01 -6.40370584E-01
 -5.31338948E-01 -4.21669495E-01 -3.11552575E-01 -2.01116139E-01 -9.03571415E-02  1.91356677E-02  1.21117905E-01  2.22687501E-01
  3.22873721E-01  4.20830017E-01  5.15598446E-01  6.05920976E-01  6.89884585E-01  7.63903624E-01  8.19009554E-01  8.52094519E-01
  8.73526978E-01  8.86911809E-01  8.93751562E-01  8.94637012E-01  8.89543653E-01  8.77744234E-01  8.57163823E-01  8.21835613E-01
  7.66100454E-01  7.00791878E-01  6.30109501E-01  5.56337011E-01  4.81101638E-01  4.05811435E-01  3.31896636E-01  2.61083379E-01
  1.95992933E-01  1.42069790E-01  1.09658268E-01  8.72630966E-02  7.09381229E-02  5.89176911E-02  5.02374065E-02  4.43136531E-02
  4.07941687E-02  3.95061392E-02  4.04625982E-02  4.39571482E-02  5.09283435E-02  6.44207770E-02  8.14683795E-02  1.00604476E-01
  1.21131322E-01  1.42619343E-01  1.64755550E-01  1.87285107E-01  2.09981543E-01  2.32625639E-01  2.54983401E-01  2.76772264E-01
  2.97589293E-01  3.16714734E-01  3.32446891E-01

            ZPREDI(1)...ZPREDI(9)=  3.58108377E 01  3.58315456E 01  3.58519793E 01  3.58665395E 01  3.58717549E 01  3.58665398E 01
  3.58519796E 01  3.58315462E 01  3.58108377E 01

            ZDEVIA(2)...ZDEVIA(5)=  2.07080841E-02  4.11415100E-02  5.57017326E-02  6.09173775E-02

                TJ1(1)...TJ1(89)= -1.88113752E 00 -1.86971316E 00 -1.85828881E 00 -1.84686445E 00 -1.83544010E 00 -1.82401575E 00
 -1.81259139E 00 -1.80116704E 00 -1.78974269E 00 -1.77831833E 00 -1.76689398E 00 -1.75546962E 00 -1.74404527E 00 -1.73262092E 00
 -1.72119656E 00 -1.70977221E 00 -1.69834787E 00 -1.68692350E 00 -1.67549916E 00 -1.66407479E 00 -1.65265045E 00 -1.56290659E 00
 -1.44950365E 00 -1.33610070E 00 -1.22269775E 00 -1.10929482E 00 -9.95891869E-01 -8.82488918E-01 -7.69085979E-01 -6.55683029E-01
 -5.42280090E-01 -4.28877145E-01 -3.15474197E-01 -2.02071249E-01 -8.86683011E-02  2.47346461E-02  1.38137579E-01  2.51540527E-01
  3.64943460E-01  4.78346419E-01  5.91749352E-01  7.05152315E-01  8.18555248E-01  9.31958210E-01  1.00000000E 00  1.00000000E 00
  1.00000000E 00  1.00000000E 00  1.00000000E 00  1.00000000E 00  1.00000000E 00  1.00000000E 00  1.00000000E 00  9.74803090E-01
  8.76377892E-01  7.77952701E-01  6.79527503E-01  5.81102312E-01  4.82677114E-01  3.84251913E-01  2.85826719E-01  1.87401526E-01
  8.89763236E-02   .00000000E 00   .00000000E 00   .00000000E 00   .00000000E 00   .00000000E 00   .00000000E 00   .00000000E 00
   .00000000E 00   .00000000E 00   .00000000E 00   .00000000E 00   .00000000E 00  2.44733199E-02  5.10747248E-02  7.76761293E-02
  1.04277535E-01  1.30878940E-01  1.57480344E-01  1.84081750E-01  2.10683155E-01  2.37284559E-01  2.63885963E-01  2.90487367E-01
  3.17088774E-01  3.43690178E-01  3.70291582E-01

            ZPREDI(1)...ZPREDI(9)=  3.58108377E 01  3.58315456E 01  3.58519793E 01  3.58665395E 01  3.58717549E 01  3.58665398E 01
  3.58519796E 01  3.58315462E 01  3.58108377E 01

            ZDEVIA(2)...ZDEVIA(5)=  2.07080841E-02  4.11415100E-02  5.57017326E-02  6.09173775E-02
```

```
        TIME=  1.00999999E 03

          ZPREDI(1)...ZPREDI(9)=  3.58110112E 01  3.58314073E 01  3.58517253E 01  3.58662200E 01  3.58714160E 01  3.58662203E 01
 3.58517259E 01  3.58314079E 01  3.58110112E 01

          ZDEVIA(2)...ZDEVIA(5)=  2.03962326E-02  4.07142639E-02  5.52086830E-02  6.04047775E-02

        TIME=  1.02000000E 03

          ZPREDI(1)...ZPREDI(9)=  3.58120823E 01  3.58316424E 01  3.58517149E 01  3.58660781E 01  3.58712366E 01  3.58660787E 01
 3.58517155E 01  3.58316430E 01  3.58120823E 01

          ZDEVIA(2)...ZDEVIA(5)=  1.95603371E-02  3.96327972E-02  5.39960861E-02  5.91545105E-02

        TIME=  1.02999999E 03

          ZPREDI(1)...ZPREDI(9)=  3.58140025E 01  3.58323497E 01  3.58520281E 01  3.58661878E 01  3.58712915E 01  3.58661884E 01
 3.58520287E 01  3.58323503E 01  3.58140025E 01

          ZDEVIA(2)...ZDEVIA(5)=  1.83472633E-02  3.80258560E-02  5.21855354E-02  5.72891235E-02

        TIME=  1.03999999E 03

          ZPREDI(1)...ZPREDI(9)=  3.58167166E 01  3.58335155E 01  3.58526707E 01  3.58665776E 01  3.58716106E 01  3.58665776E 01
 3.58526710E 01  3.58335161E 01  3.58167166E 01

          ZDEVIA(2)...ZDEVIA(5)=  1.67989731E-02  3.59539986E-02  4.98609543E-02  5.48939705E-02

        TIME=  1.05000000E 03

          ZPREDI(1)...ZPREDI(9)=  3.58201680E 01  3.58351400E 01  3.58536270E 01  3.58672574E 01  3.58722052E 01  3.58672574E 01
 3.58536276E 01  3.58351406E 01  3.58201680E 01

          ZDEVIA(2)...ZDEVIA(5)=  1.49722099E-02  3.34591866E-02  4.70895767E-02  5.20372391E-02

        TIME=  1.05999999E 03

          ZPREDI(1)...ZPREDI(9)=  3.58243024E 01  3.58372173E 01  3.58549136E 01  3.58682331E 01  3.58730817E 01  3.58682331E 01
 3.58549142E 01  3.58372176E 01  3.58243024E 01

          ZDEVIA(2)...ZDEVIA(5)=  1.29146576E-02  3.06110382E-02  4.39305305E-02  4.87790108E-02

        TIME=  1.06999999E 03

          ZPREDI(1)...ZPREDI(9)=  3.58290708E 01  3.58397374E 01  3.58565721E 01  3.58695063E 01  3.58742413E 01  3.58695063E 01
 3.58565724E 01  3.58397377E 01  3.58290708E 01

          ZDEVIA(2)...ZDEVIA(5)=  1.06663704E-02  2.75011065E-02  4.04353142E-02  4.51703072E-02

        TIME=  1.08000000E 03

          ZPREDI(1)...ZPREDI(9)=  3.58344287E 01  3.58426905E 01  3.58585718E 01  3.58710769E 01  3.58756861E 01  3.58710769E 01
 3.58585724E 01  3.58426914E 01  3.58344287E 01

        ZDEVIA(2)...ZDEVIA(5)=  8.26168060E-03  2.41432190E-02  3.66482735E-02  4.12573814E-02


             TIME=  1.08999999E 03

        ZPREDI(1)...ZPREDI(9)=  3.58403361E 01  3.58460665E 01  3.58609113E 01  3.58729452E 01  3.58774161E 01  3.58729452E 01
 3.58609116E 01  3.58460674E 01  3.58403361E 01

        ZDEVIA(2)...ZDEVIA(5)=  5.73015213E-03  2.05750465E-02  3.26089859E-02  3.70798111E-02


             TIME=  1.09999999E 03

                T3(0)...T3(800)= -1.86693680E 00 -1.85937995E 00 -1.85022463E 00 -1.84019890E 00 -1.82956879E 00 -1.81844544E 00
-1.80687179E 00 -1.79485013E 00 -1.78235152E 00 -1.76931763E 00 -1.75565876E 00 -1.74124928E 00 -1.72592026E 00 -1.70944904E 00
-1.69154403E 00 -1.67182189E 00 -1.64977059E 00 -1.62468170E 00 -1.59550601E 00 -1.56048203E 00 -1.51596035E 00 -1.45188472E 00
-1.36998340E 00 -1.27916524E 00 -1.18277819E 00 -1.08247921E 00 -9.79268897E-01 -8.73826647E-01 -7.66642553E-01 -6.58078521E-01
-5.48398435E-01 -4.37777627E-01 -3.26285332E-01 -2.13800856E-01 -9.97048795E-02  1.42266729E-02  1.11569615E-01  2.10709831E-01
 3.09502023E-01  4.06710649E-01  5.01243931E-01  5.91840333E-01  6.76735795E-01  7.52875662E-01  8.13208854E-01  8.48651123E-01
 8.71507525E-01  8.85956120E-01  8.93723845E-01  8.95531917E-01  8.91481817E-01  8.81051278E-01  8.62672961E-01  8.32072389E-01
 7.78417438E-01  7.14224285E-01  6.44179052E-01  5.70733374E-01  4.95576489E-01  4.20127964E-01  3.45785984E-01  2.74170420E-01
 2.07593694E-01  1.50461324E-01  1.14619078E-01  9.05735230E-02  7.32529795E-02  6.05426204E-02  5.13411146E-02  4.49961549E-02
 4.11119169E-02  3.94804490E-02  4.00767231E-02  4.31341976E-02  4.94496280E-02  6.16058517E-02  7.80306101E-02  9.67980134E-02
 1.17073868E-01  1.38377257E-01  1.60370655E-01  1.82784940E-01  2.05383989E-01  2.27941310E-01  2.50216922E-01  2.71922907E-01
 2.92651206E-01  3.11676964E-01  3.27293923E-01 -1.86344285E 00 -1.85594587E 00 -1.84718299E 00 -1.83749060E 00 -1.82706977E 00
-1.81602556E 00 -1.80440119E 00 -1.79219519E 00 -1.77936804E 00 -1.76584296E 00 -1.75150266E 00 -1.73618335E 00 -1.71966513E 00
-1.70165874E 00 -1.68178679E 00 -1.65955679E 00 -1.63431995E 00 -1.60520396E 00 -1.57099517E 00 -1.52992943E 00 -1.47938187E 00
-1.41595507E 00 -1.34007807E 00 -1.25511366E 00 -1.16357255E 00 -1.06714310E 00 -9.66991544E-01 -8.63957667E-01 -7.58664298E-01
-6.51578909E-01 -5.43048424E-01 -4.33315229E-01 -3.22513336E-01 -2.10636422E-01 -9.74754214E-02  1.42349181E-02  1.09786199E-01
 2.05715650E-01  3.00736386E-01  3.93628711E-01  4.83130461E-01  5.67767578E-01  6.45638114E-01  7.14108545E-01  7.69615537E-01
 8.09523630E-01  8.37151444E-01  8.55077302E-01  8.64821136E-01  8.67127299E-01  8.62129593E-01  8.49357700E-01  8.27591944E-01
 7.94664001E-01  7.48169941E-01  6.91905630E-01  6.29151988E-01  5.62357879E-01  4.93472624E-01  4.24217564E-01  3.56288806E-01
 2.91557077E-01  2.32330513E-01  1.81675959E-01  1.43116379E-01  1.14405349E-01  9.29434025E-02  7.69661403E-02  6.53262275E-02
 5.72721332E-02  5.23234606E-02  5.02122021E-02  5.08679938E-02  5.44424307E-02  6.13634032E-02  7.22646821E-02  8.66020572E-02
 1.03435139E-01  1.22052231E-01  1.41933642E-01  1.62678127E-01  1.83953479E-01  2.05464390E-01  2.26927936E-01  2.48049343E-01
 2.68490380E-01  2.87818894E-01  3.05424857E-01  3.20428178E-01 -1.86156876E 00 -1.85380447E 00 -1.84509785E 00 -1.83553810E 00
-1.82522185E 00 -1.81421399E 00 -1.80254100E 00 -1.79019096E 00 -1.77711312E 00 -1.76321533E 00 -1.74835913E 00 -1.73235194E 00
-1.71493636E 00 -1.69577540E 00 -1.67443322E 00 -1.65034948E 00 -1.62280628E 00 -1.59088558E 00 -1.55341977E 00 -1.50895607E 00
-1.45582552E 00 -1.39260377E 00 -1.31943300E 00 -1.23782912E 00 -1.14945520E 00 -1.05573460E 00 -9.57804441E-01 -8.56555712E-01
-7.52682328E-01 -6.46722358E-01 -5.39089489E-01 -4.30096334E-01 -3.19972888E-01 -2.08887979E-01 -9.70026708E-02  1.29219623E-02
 1.07525456E-01  2.01498145E-01  2.93948072E-01  3.83775792E-01  4.69716918E-01  5.50301147E-01  6.23795581E-01  6.88203704E-01
 7.41514277E-01  7.82617635E-01  8.12462783E-01  8.32364464E-01  8.43356860E-01  8.46044159E-01  8.40587652E-01  8.26717889E-01
 8.03778017E-01  7.70925587E-01  7.27839243E-01  6.76241976E-01  6.18342161E-01  5.56256592E-01  4.91924077E-01  4.27165163E-01
 3.63773862E-01  3.03608179E-01  2.48647490E-01  2.00907081E-01  1.61880058E-01  1.31061032E-01  1.07202004E-01  8.90913987E-02
 7.57537335E-02  6.64617741E-02  6.07086641E-02  5.81805712E-02  5.87369227E-02  6.23919851E-02  6.92708212E-02  7.94600546E-02
 9.26373732E-02  1.08241139E-01  1.25714542E-01  1.44571696E-01  1.64396718E-01  1.84826270E-01  2.05530250E-01  2.26193422E-01
 2.46497750E-01  2.66105121E-01  2.84643161E-01  3.01708850E-01  3.16945896E-01 -1.86059991E 00 -1.85263227E 00 -1.84389305E 00
-1.83436963E 00 -1.82409355E 00 -1.81309615E 00 -1.80138721E 00 -1.78894484E 00 -1.77570960E 00 -1.76157910E 00 -1.74640135E 00
-1.72996610E 00 -1.71199317E 00 -1.69211747E 00 -1.66987027E 00 -1.64465700E 00 -1.61573350E 00 -1.58218606E 00 -1.54292938E 00
-1.49675477E 00 -1.44248885E 00 -1.37934163E 00 -1.30738376E 00 -1.22749358E 00 -1.14087424E 00 -1.04873177E 00 -9.52134919E-01
-8.51976860E-01 -7.48981535E-01 -6.43725979E-01 -5.36666781E-01 -4.28167826E-01 -3.18532503E-01 -2.08049765E-01 -9.70728159E-02
 1.17917369E-02  1.05818805E-01  1.98695199E-01  2.89652908E-01  3.77668533E-01  4.61515594E-01  5.39784133E-01  6.10912150E-01
 6.73291385E-01  7.25530690E-01  7.66907561E-01  7.97678787E-01  8.18542635E-01  8.30198658E-01  8.33125031E-01  8.27499723E-01
 8.13212550E-01  7.89968437E-01  7.57530755E-01  7.16153926E-01  6.67033482E-01  6.11882794E-01  5.52574104E-01  4.90976560E-01
 4.28918439E-01  3.68200928E-01  3.10611501E-01  2.57885194E-01  2.11543357E-01  1.72539183E-01  1.40836212E-01  1.15772413E-01
 9.64887512E-02  8.21681523E-02  7.21346819E-02  6.58817232E-02  6.30688453E-02  6.35031599E-02  6.71050715E-02  7.38457018E-02
 8.36385059E-02  9.62089503E-02  1.11133523E-01  1.27943960E-01  1.46188729E-01  1.65454452E-01  1.85365784E-01  2.05576065E-01
 2.25755316E-01  2.45579401E-01  2.64724457E-01  2.82874221E-01  2.99755245E-01  3.15225396E-01 -1.86029699E 00 -1.85225855E 00
```

# NOMENCLATURE

NOTE: Certain quantities, which have only a localized significance, and whose meaning is clearly indicated in the text, will not be defined here.

The following variables, used in the computer program were discussed in Chapter IV and are defined in Appendix A:

AZ1
AZ2
BO1
CB
D
DT
DTDMP
DTPNT
EPS
EPS1
FACTOR
INSW
IT0
IT1
...
IT5
IT6
L
MAX
MIN
MVMTRT
MXST
N
PMAX1
PMIN1
SWOUT
T
TE0
TE1
...
TE5
TE6
TEND
THSTRT
TMOVE
TPOINT
TSTART
TTIMOV
Z
ZDEVIA
ZHFPR
ZPRED1

The following variables are defined here:

| | |
|---|---|
| $c, c_p$ | Specific heat. |
| $h$ | The heat transfer coefficient. |

$i$ Denotes a grid point whose x coordinate is $(i-1)\Delta x$.

$j$ Denotes a grid point whose y coordinate is $(j-1)\Delta y$.

$k$ Thermal conductivity.

$\ell_1, \ell_2$ Dimensions of the rectangle.

$m, s$ Indicates the fraction of the interval an interface point is separated from a grid point.

$t$ Dimensionless time.

$\bar{t}$ Time in natural units.

$x, y$ Dimensionless spatial coordinates.

$\bar{x}, \bar{y}$ Spatial coordinates in natural units.

A, B, C, D Coefficients in equations (3.9.20) and (3.10.20).

*B1 = -2-2L.

B11 = -2-2L-2H.

BT Defined in equation (3.11.3).

BU $= -2(R^2+L)$.

BUU $= -2(R^2+GR^2+L)$.

E Constant in equations (3.9.19) and (3.10.19).

$F_1$ $= (\Delta x)^2 \Delta_y^2 \; T2_{i,j} - 2L\cdot T2_{i,j}$.

*The symbols T1, T2, T3, T11, T1P, TJ1, TJQ, BU, BUU, BT, GM, B1 and B11 were used in the computer program and are adopted for the present discussion. These are distinct symbols, not multiplication of several separate symbols. To eliminate any confusion, whenever such a symbol is multiplied by a constant or variable a "." is used to indicate the multiplication.

| | |
|---|---|
| $F_2$ | $= -(\Delta x)^2 \ \Delta_x^2 \ T1_{i,j} - 2L\cdot T1_{i,j}$. |
| $F_3$ | $= (\Delta x)^2 \ \Delta_x^2 \ T3_{i,j} - 2L\cdot T3_{i,j}$. |
| $F_4$ | $= -(\Delta x)^2 \ \Delta_y^2 \ T2_{i,j} - 2L\cdot T2_{i,j}$. |
| G | $= h_r \ \ell_1 \ \Delta y/k_r$. |
| GM | Defined in equation (3.11.3). |
| H | $= h_r \ \ell_1 \ \Delta x/k_r$. |
| L | $= \dfrac{\alpha_1 (\Delta x)^2}{\alpha_r (\Delta t)}$ . |
| P | The number of equally spaced grid points in the x direction. |
| Q | The number of equally spaced grid points in the y direction. |
| R | Denotes a region or subregion. |
| $R^2$ | $= (\Delta x/\Delta y)^2$ . |
| $S_i$ | The boundary of $R_i$. |
| $S_{ij}$ | Interface between regions $R_i$ and $R_j$. |
| T | Dimensionless temperature. |
| T1 | $= T^{(t)}$ . |
| T2 | $= T^{(t+\Delta t/2)}$. |
| T3 | $= T^{(t+\Delta t)}$. |

| | |
|---|---|
| T*,T**,T*** | Intermediate approximations of T. |
| $TI1_j$ | Temperature of the surroundings near the point (1,j). |
| $TIP_j$ | Temperature of the surroundings near the point (P,j). |
| $TJ1_i$ | Temperature of the surroundings near the point (i,1). |
| $TJQ_i$ | Temperature of the surroundings near the point (i,Q). |
| Z | Location of the interface in interval units. |
| X | Dimensionless location of the interface. |
| $\bar{X}$ | Location of the interface in natural units. |
| $\alpha$ | Thermal diffusivity, = $k/\rho c$, |
| $\gamma$ | An angle, defined in Figure 3.2. |
| $\Delta$ | Indicates an interval of the variable following it, or a variable defined in Figure 5.1. |
| $\Delta_x^2$ | Indicates numerical approximation of $\frac{\partial^2}{\partial x^2}$. |
| $\varepsilon$ | Radius of a grid point's surroundings. |
| $\theta$ | Temperature in natural units. |
| $\theta_m$ | The transformation temperature. |
| $\theta_h$ | A reference temperature. |
| $\lambda$ | Latent heat of fusion. |
| $\rho$ | Density. |

# BIBLIOGRAPHY

1. Allen, D.N. de G. and Severn, R.T., "The Application of Relaxation Methods to the Solution of Non-Elliptic Partial Differential Equations," Quart. J. Mech. Appl. Math., 15, (1962), pp. 53-62.

2. Altman, M., "Some Aspects of the Melting Solution For a Semi-Infinite Slab," Chem. Engr. Progress Symposium Series, 57, No. 32, (1961), p. 16.

3. Batten, G. W., "Second Order Correct Boundary Conditions for the Numerical Solution of the Mixed Boundary Problem for Parabolic Equations," Math. of Comp., 17, (1963), pp. 405-413.

4. Baxter, D. C., "The Fusion Times of Slabs and Cylinders," Trans. ASME-J. Heat Transfer, 84C, (1962), pp. 317-326.

5. Bellman, R., "On the Boundness of Solutions of Non-Linear Differential and Difference Equations," Trans. Amer. Math. Soc., 62, (1947), pp. 357-386.

6. __________, "On the Existence and Boundedness of Solutions of Non-Linear Partial Differential Equations of Parabolic Type," Trans. Amer. Math. Soc., 64, (1948), pp. 21-44.

7. Boley, B. A., "A Method of Heat Conduction Analysis of Melting and Solidification Problems," J. Math. and Phys., 40, (1961), pp. 300-313.

8. __________, "Upper and Lower Bounds for the Solution of a Melting Problem," Quart. Appl. Math., 21, No. 1, (Apr. 1963), pp. 1-11.

9. Boley, B. A. and Weiner, J. H., Theory of Thermal Stresses, John Wiley and Sons, Inc., New York, 1960.

10. Bolling, G. E. and Tiller, W. A., "Growth from Melt. I. Influence of Surface Intersections in Pure Metals," J. Appl. Phys., 31, (1960), pp. 1345-1350.

11. __________, "Growth from Melt. II. Cellular Interface Morphology," J. Appl. Phys., 31, (1960), pp. 2040-2045.

12. __________, "Growth from Melt. III. Dendritic Growth," J. Appl. Phys., 32, (1961), pp. 2587-2605.

13. Bowers, R., Ure, R. W., Jr., Bauerle, J. E. and Cornish, A. J., "InAs and InSb as Thermoelectric Materials," J. Appl. Phys., 30, (1959), p. 930.

14. Braun, J. H. and Pellin, R. A., "The Shape of Melt-Crystal Interfaces During Float Zoning of Silicon," J. Electrochem. Soc., 108, (1961), pp. 969-974.

15. Brian, P. L. T., "A Finite-Difference Method of High-Order Accuracy for the Solution of Three-Dimensional Transient Heat Conduction Problems," AIChE Journal, 7, No. 3, (Sept. 1961), pp. 367-370.

16. Busch, G. and Schneider, M., "Heat Conduction in Semiconductors," Physica, 20, (Nov. 1954), p. 1084.

17. Carnahan, B., Luther, H. A. and Wilkes, J. O., Applied Numerical Methods, John Wiley and Sons, Inc., New York, 1964. (Prelim. Edition).

18. Carslaw, H. S. and Jaeger, J. C., Conduction of Heat in Solids, Clarendon Press, Oxford, 1959.

19. Seider, W. D. and Churchill, S. W., "The Effect of Insulation on Freezing Motion," Chem. Engr. Progress Symposium Series, 61, No. 59, (1965).

20. Teller, A. S. and Churchill, S. W., "Freezing Outside a Sphere," Chem. Engr. Progress Symposium Series, 61, No. 59, (1965).

21. Citron, S. J., "Heat Conduction in a Melting Slab," J. Aero/Space Sci., 27, (March 1960), pp. 219-228.

22. Collatz, L., The Numerical Treatment of Differential Equations, Springer-Verlag, Berlin, 1960.

23. Cook, J. S., Mason, D. R. and Smith, P. H., "Temperature Profiles in a Cylindrical Zone-Refining Ingot," Electrochem. Techn., 1, (1963), p. 300.

24. Crank, J., "Two Methods for the Numerical Solution of Moving-Boundary Problems in Diffusion and Heat Flow," Quart. J. Mech. and Appl. Math., 10, (1957), pp. 210-231.

25. __________, The Mathematics of Diffusion, Clarendon Press, Oxford, 1964.

26. Dewey, C. F., Jr., Schlesinger, S. I. and Sashkin, L., "Temperature Profiles in a Finite Solid with Moving Boundary," J. Aero/Space Sci., 27, (Jan. 1960), pp. 59-64.

27. Donald, D. K., "Thermal Behavior in Vacuum Zone Refining," Rev. Sci. Instr., 32, (1961), p. 811.

28. Douglas, J., Jr., "On the Numerical Integration of $\partial^2U/\partial x^2 + \partial^2U/\partial y^2 = \partial U/\partial t$ by Implicit Methods," J. Soc. Indust. Appl. Math., 3, (1955), pp. 42-65.

29. ___________, "The Solution of the Diffusion Equation by a High Order Correct Difference Equation," J. Math. Physics, 35, (1956), pp. 145-152.

30. ___________, "On the Numerical Integration of Quasi-Linear Parabolic Differential Equations," Pacific J. Math., 6, (1956), pp. 35-42.

31. ___________, "A Uniqueness Theorem for the Solution of the Stefan Problem," Proc. Amer. Math. Soc., 8, (1957), pp. 402-408.

32. ___________, "A Note on Alternating Direction Implicit Method for the Numerical Solution of Heat Flow Problems," Proc. Amer. Math. Soc., 8, (1957), pp. 409-412.

33. ___________, "The Application of Stability Analysis in the Numerical Solution of Quasi-Linear Parabolic Differential Equations," Trans. Amer. Math. Soc., 89, (1958), pp. 484-518.

34. ___________, "A Survey of Numerical Methods for Parabolic Differential Equations," Advances in Computers, 2, (1961), pp. 1-54. (F.L.Alt, Editor, Academic Press, New York)

35. ___________, "Alternating Direction Method for Three Space Variables," Num. Math., 4, (1962), pp. 41-63.

36. Douglas, J., Jr., and Gallie, T. M., Jr., "On the Numerical Integration of a Parabolic Differential Equation Subject to a Moving Boundary Condition," Duke Math. J., 22, (1955), pp. 557-572.

37. Douglas, J., Jr., and Gunn, J. E., "Two High-Order Correct Difference Analogs for the Equation of Multi-Dimensional Heat Flow," Math. Comp., 17, No. 81, (Jan. 1963), pp. 71-80.

38. ___________, "A General Formulation of Alternating Direction Methods," Num. Math., 6, 1964, pp. 428-453.

39. Douglas, J., Jr., and Jones, B. F., "On Predictor-Corrector Methods for Non-Linear Parabolic Differential Equations," J. Soc. Indust. Appl. Math., 11, No. 1, (March 1963), pp. 195-204.

40. Douglas, J., Jr., and Rachford, H. H., Jr., "On the Numerical Solution of Heat Conduction Problems in Two or Three Space Variables," Trans. Amer. Math. Soc., 82, (1956), pp. 421-439.

41. Drabble, J. R. and Goldsmid, J. H., *Thermal Conduction in Semiconductors*, Pergamon Press, 1961.

42. Du Fort, E. C. and Frankel, S. P., "Stability Conditions in the Numerical Treatment of Parabolic Differential Equations," *Math. Tables Aids Comp.*, *7*, (1953), pp. 135-152.

43. Dusinberre, G. M., "A Note on Latent Heat in Digital Computer Calculations," A.S.M.E. Paper No. 58-HT-7. Presented at the ASME-AIChE Joint Heat Transfer Conference, Evanston, Ill., 1958.

44. Ehrlich, L. W., "A Numerical Method of Solving a Heat Flow Problem with Moving Boundary," *J. Assoc. Comput. Mach.*, *5*, (1958), pp. 161-176.

45. Evans, G. W., II, Isaacson, E. and Macdonald, J. K.L., "Stefan-Like Problems," *Quart. Appl. Math.*, *8*, (1950), pp. 312-319.

46. Friedman, A., "Free Boundary Problems for Parabolic Equations I. Melting of Solids," *J. Math. Mech.*, *8*, (1959), pp. 499-517.

47. __________, "Remarks on Stefan-Type Free Boundary Problems for Parabolic Equations," *J. Math. Mech.*, *9*, (1960), pp. 885-903.

48. __________, *Partial Differential Equations of Parabolic Type*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1964.

49. Forsythe, G. E. and Wasow, W. R., *Finite Difference Methods for Partial Differential Equations*, John Wiley and Sons, Inc., New York, 1960.

50. Goodman, T. R. and Shea, J. J., "The Melting of Finite Slabs," *Trans. ASME-J. Appl. Mech.*, *82*, (March 1960), pp. 16-24.

51. Hamill, T. D. and Bankoff, S. G., "Maximum and Minimum Bounds on Freezing-Melting Rates with Time Dependent Boundary Conditions," *A.I.Ch.E. J.*, *9*, No. 6, (Nov. 1963), pp. 741-744.

52. Hamilton, D. R. and Seidensticker, R. G., "Growth Mechanism of Germanium Dendrites, Kinetics and the Nonisothermal Interface," *J. Appl. Phys.*, *34*, (May 1963), pp. 1450-1460.

53. Hashemi, H. T., *Heat Conduction with Change of Phase*, Ph.D. Thesis, The U. of Oklahoma, Norman, Oklahoma, 1965.

54. Hunt, M. D., Spittle, J. A. and Smith, R. W., "The Decanted Interface," *Canad. J. Phys.*, *41*, (Sept. 1963), pp. 1420-1423.

55. Ioffe, A. F. and Regel, A. R., "Non-Crystalline, Amorphous, and Liquid Electronic Semiconductors," *Progress in Semiconductors*, *4*, 1960, pp. 237-291. (A. F. Gibson, Gen. Editor, Heywood and Co., Ltd., London)

56. Jost, W., Diffusion in Solids, Liquids, Gases, Academic Press, Inc., New York, 1960.

57. Kellogg, R. B., "Another Alternative Direction Implicit Method," J. Soc. Indust. Appl. Math., 11, No. 4, (Dec. 1963), p. 976.

58. Kolodner, I. I., "Free Boundary Problem for Heat Equation with Applications to Problems of Change of Phase," Comm. Pure Appl. Math., 9, (1956), pp. 1-31.

59. Kulwicki, B. M., The Phase Equilibria of Some Compound Semiconductors by DTA Calorimetry, Ph.D. Dissertation, The University of Michigan, 1963.

60. Kyner, W. T., "An Existence and Uniqueness Theorem for a Non-Linear Stefan Problem," J. Math. Mech., 8, (1959), pp. 483-498.

61. ___________, "On Free Boundary Value Problem for the Heat Equation," Quart. Appl. Math., 17, (1959), pp. 305-310.

62. Landau, H. G., "Heat Conduction in a Melting Solid," Quart. Appl. Math., 8, (1950), pp. 81-94.

63. Lawson, W. D. and Nielsen, S., Preparation of Single Crystals, Butterworths Publications Ltd., London, 1958.

64. Lax, P. D., "Weak Solutions of Non-Linear Hyperbolic Equations and Their Numerical Computation," Comm. Pure Appl. Math., 7, (1954), pp. 159-193.

65. Lax, P. D. and Richtmyer, R. D., "Survey of the Stability of Linear Finite Difference Equations," Comm. Pure. Appl. Math., 9, (1956), pp. 267-293.

66. Lees, M., "Alternating Direction and Semi-Explicit Methods for Parabolic Partial Differential Equations," Num. Math., 3, (1961), pp. 398-412.

67. Lightfoot, N. M. H., "The Solidification of Molten Steel," Proc. London Math. Soc., 31, No. 2, (1930), pp. 97-116.

68. Longwell, P. A., "A Graphical Method for Solution of Freezing Problems," A.I.Ch.E. J., 4, No. 1, (March 1958), pp. 53-57.

69. Lotkin, M., "The Calculation of Heat Flow in Melting Solids," Quart. Appl. Math., 8, (1960), pp. 79-85.

70. Luke, Y. L., "Numerical Solution of the Heat Conduction Equation," Chem. Eng., 68, (Jan. 9, 1961), pp. 95-102.

71. McMordie, R. K., "Steady-State Conduction with Variable Thermal Conductivity," Trans. ASME-J. Heat Transfer, 84C, (1962), p. 92.

72. Mikeladze, S. E., "O Chislennom Integrirovanii Uravnenij Ellipticheskogo I Parabolicheskogo Tipov," Izv. Akad. Nauk SSSR. Ser. Mat., 5, (1941), pp. 57-74.

73. Miranker, W. L., "A Free Boundary Value Problem for the Heat Equation," Quart. Appl. Math., 16, (1958), pp. 121-130.

74. Miranker, W. L. and Keller, J. B., "The Stefan Problem for a Nonlinear Equation," J. Math. Mech., 9, (1960), pp. 67-70.

75. Mitchell, A. R. and Fairweather, G., "Improved Forms of the Alternating Direction Methods of Douglas, Peaceman and Rachford for Solving Parabolic and Elliptic Equations," Num. Math., 6, (1964), pp. 285-292.

76. Murray, W. D. and Landis, F., "Numerical and Machine Solutions of Transient Heat-Conduction Problems Involving Melting or Freezing," Trans. ASME-J. Heat Transfer, 81C, (1959), p. 106.

77. Peaceman, D. W. and Rachford, H. H., Jr., "The Numerical Solution of Parabolic and Elliptic Differential Equations," J. Soc. Indust. Appl. Math., 3, No. 1, (March 1955), pp. 28-41.

78. Pearcy, C. M., "On Convergence of Alternating Direction Procedures," Num. Math., 4, (1962), pp. 172-176.

79. Pfann, W. G., Zone Melting, John Wiley and Sons, Inc., New York, 1966.

80. Poots, G., "An Approximate Treatment of a Heat Conduction Problem Involving a Two-Dimensional Solidification Front," Int. J. of Heat and Mass Transfer, 5, (May 1962), pp. 339-348.

81. __________, "On the Application of Integral-Methods of the Solution on Problems Involving the Solidification of Liquids Initially at Fusion Temperature," Int. J. of Heat and Mass Transfer, 5, (May 1962), pp. 525-531.

82. Price, P. H. and Slack, M. R., "Stability and Accuracy of Numerical Solutions of the Heat Flow Equation," British J. of Appl. Phys., 3, (1952), pp. 379-384.

83. __________, "The Effect of Latent Heat on Numerical Solutions of the Heat Flow Equation," British J. of Appl. Phys., 5, (1954), pp. 285-287.

84. Richtmyer, R. D., Difference Methods for Initial Value Problems, Interscience Publishers, New York, 1957.

85. Rose, M. E., "A Method for Calculating Solutions of Parabolic Equations with a Free Boundary," Math. Comput., 14, (1960), pp. 249-256.

86. Ruoff, A. L., "An Alternate Solution of Stefan's Problem," Quart. of Appl. Math., 16, (1958), pp. 197-201.

87. Sanders, R. W., "Transient Heat Conduction in a Melting Finite Slab. An Exact Solution," ARS J., (Nov. 1960), pp. 1030-1031.

88. Shortley, G. H. and Weller, R., "The Numerical Solution of Laplace's Equation," J. Appl. Phys., 9, (1938), pp. 334-348.

89. Stefan, J., "Uber die Theorie der Eisbildung, inbesondere uber die Eisbildung im Polarmere," Annalen der Physik und Chemie, 42, (1891), p. 269.

90. Stuckes, A. D., "Thermal Conductivity of Indium Antimonide," Phys. Review, 107, (July 1957), p. 427.

91. Sunderland, J. E. and Grosh, R. J., "Transient Temperature in a Melting Solid," Trans. ASME-J. Heat Transfer, 83C, (1961), pp. 409-414.

92. Thompson, R. J., "Difference Approximations for Inhomogenous and Quasi-Linear Equations," J. Soc. Indust. Appl. Math., 12, No. 1, (March 1964), pp. 189-199.

93. Tiller, W. A., "Effect of Grain Boundaries on Solute Partitioning During Progressive Solidification," J. Appl. Phys., 33, (1962), p. 3106.

94. Todd, J., Survey of Numerical Analysis, McGraw-Hill Book Co., Inc., 1962.

95. Trench, W. F., "On an explicit Method for the Solution of a Stefan Problem," J. Soc. Indust. Appl. Math., 7, No. 2, (June 1959), pp. 184-204.

96. Ure, R. W., Jr., Bowers, R. and Miller, R. C., "Materials for Thermo-Electric Applications," in Properties of Elemental and Compound Semiconductors by H. C. Gatos, Met. Soc. Conf., Vol. 5, Interscience Publishers, New York (1959), pp. 245-259.

97. Varga, R. S., "On Higher Order Stable Implicit Methods for Solving Parabolic Partial Differential Equations," J. Math. Physics, 40, (1961), pp. 220-231.

98. Varga, R. S., Matrix Iterative Analysis, Prentice-Hall, Inc., 1962.

99. Wilcox, W. R. and Duty, R. L., "Macroscopic Interface Shape During Solidification," Report No. ATN-64(9236)-18, Aerospace Corp., 1964.

100. Wilkes, J. O., The Finite Difference Computation of Natural Convection in an Enclosed Rectangular Cavity, Ph.D. Thesis, The University of Michigan, Ann Arbor, Michigan, 1963.

101. Young, D. M. and Frank, T. G., "A Survey of Computer Methods for Solving Elliptic and Parabolic Partial Differential Equations," I.C.C. Bull., 2, (1963), pp. 3-61.

102. Zener, C., "Theory of Growth of Spherical Precipitates from Solid Solution," J. Appl. Phys., 20, (1949), p. 950.