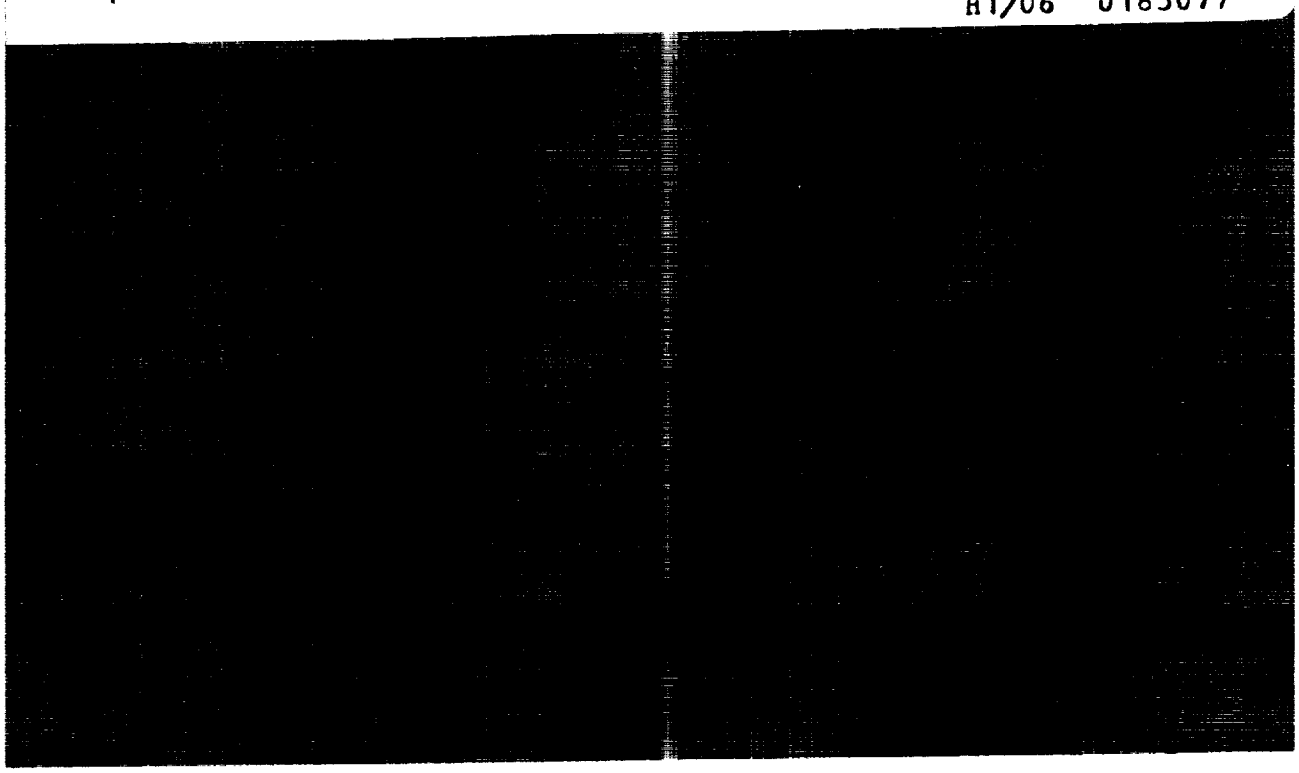
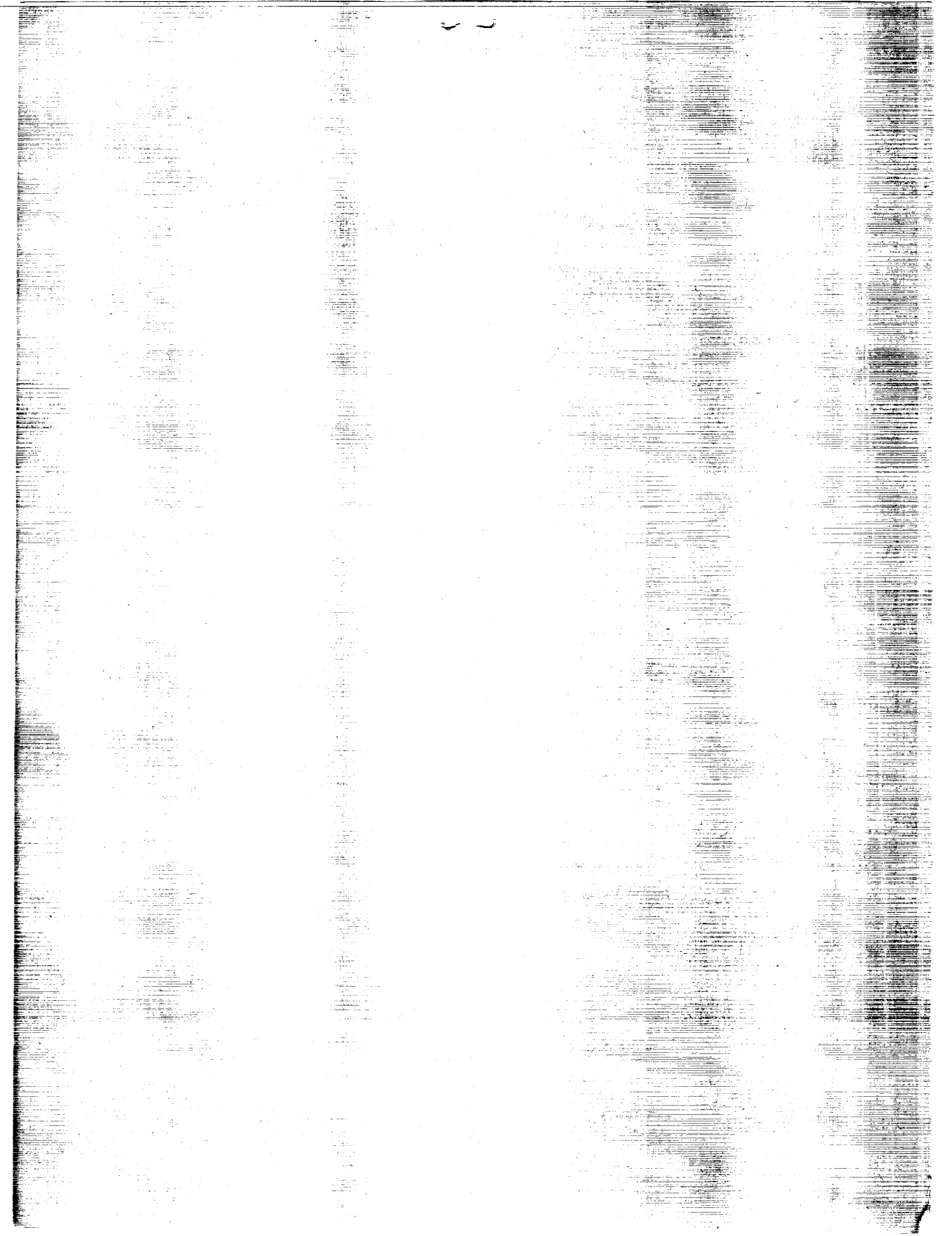
NASA Contractor Report 4194

A Numerical Simulation of the Full Two-Dimensional Electrothermal De-Icer Pad

Konstanty C. Masiulaniec

GRANT NAG3-72
NOVEMBER 1988





NASA Contractor Report 4194

A Numerical Simulation of the Full Two-Dimensional Electrothermal De-Icer Pad

Konstanty C. Masiulaniec
The University of Toledo
Toledo, Ohio

Prepared for
Lewis Research Center
under Grant NAG3-72

National Aeronautics
and Space Administration

**Scientific and Technical
Information Division**

1988

ACKNOWLEDGMENTS

I wish to give special thanks to Doctors T.G. Keith, Jr. and K.J. DeWitt for all of their guidance and assistance with this investigation and for serving as program and research advisors. Many thanks also to Doctors A. Afjeh, I. Stein, Jr. and A. Kumar for serving on the Defense Committee.

Thanks are also due to Dr. J. Shaw of NASA Lewis in Cleveland, Ohio, for his continued financial support through grants funding the de-icing research at The University of Toledo and for his granting to the investigation the use of NASA Lewis' CRAY-XMP.

TABLE OF CONTENTS

NOMENCLATURE

C	Specific heat capacity
h	Convective heat transfer coefficient
H	Enthalpy
i,j	Grid indices
I	Interface location
J	Jacobian of spacial transformation
k	Thermal conductivity
n	Outward normal direction
$\dot{q}$	Rate of internal energy generation
r	Radial Coordinate
R	Radius value
t	Time
T	Temperature
V	Velocity
v	Volume
x,y	Physical plane Coordinates
α, β, γ	Coordinate Transformation parameters
η, ξ	Transformed plane coordinates
ρ	Density
λ	Latent heat of fusion

Subscripts

f	Fusion
i	At inner surface
L	Liquid
M	Melt
O	Outer surface
s	Solid
∞	Ambient

Superscripts

k	Current time step

A Numerical Simulation of the Full

Two-Dimensional Electrothermal

De-Icer Pad

Konstanty C. Masiulaniec

SUMMARY

The ability to predict the time-temperature history of electrothermal de-icer pads is important in the subsequent design of improved and more efficient versions. These de-icer pads are installed near the surface of aircraft components, for the specific purpose of removing any accreted ice. The proposed numerical model can incorporate the full two-dimensional geometry through a section of a region (i.e., section of an airfoil, etc.), that current one-dimensional numerical codes are unable to do. Thus, the effects of irregular layers, curvature, etc., can now be accounted for in the thermal transients. Each layer in the actual geometry is mapped via a body-fitted coordinate transformation into uniform, rectangular computational grids. The relevant heat transfer equations are transformed and discretized. To model the phase change that might occur in any accreted ice, in an enthalpy formulation the phase change equations are likewise transformed and discretized. The code developed was tested against numerous classical and numerical solutions, as well as against experimental de-icing data on a UH1H rotor blade obtained from the NASA Lewis Research Center in Cleveland, Ohio. The excellent comparisons obtained show that this code can be a useful tool in predicting the performance of current de-icer models, as well as in the designing of future models.

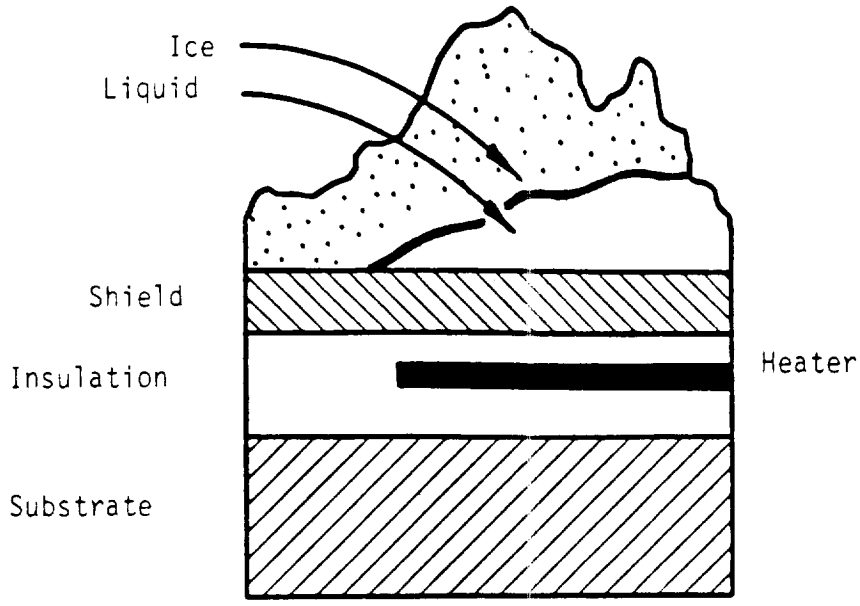
CHAPTER 1 INTRODUCTION

The formation of ice on the exterior surfaces of aircraft has a considerable effect on flight performance, as it increases drag and decreases lift. Thus an aircraft must be designed with the equipment necessary for ice removal or prevention. Basically, aircraft ice protection systems can be classified as either anti-icing or de-icing.

The anti-icing principle involves the prevention of ice formation on the protected area at all times. Typical anti-icing methods make use of chemicals and/or the passage of hot bleed air through channels below the surface on which ice formation is to be prevented. In contrast, de-icing involves the periodic removal of accreted ice by mechanical or thermal means. For ice removal systems, attention must also be given to uniform shedding of the ice. Itagaki [1] elaborates on the dangers of non-uniform shedding. Various de-icing methods that have been investigated include pneumatic boots and thermal techniques. The latter consists of cyclic heating of discrete elements by electrothermal means. The energy requirements are significantly less for de-icing systems than they are for anti-icing systems. From experimental studies, Stallabrass [2] concluded that the electrothermal method has the most advantages as a de-icing mechanism, although it does have some maintainability/ reliability problems. Werner [3] has also reported that the electrothermal de-icing technique is the most commonly used method, and that it has been applied to both fixed and rotary wing aircraft.

The objective of an electrothermal de-icing system is to raise the composite blade surface/ice interface temperature above the melting temperature of ice, resulting in a very thin interfacial layer of liquid which reduces the ice adhesion to the blade surface. Aerodynamic and/or centrifugal forces can then readily sweep the unmelted ice from the surface. A typical electrothermal de-icer pad is essentially a composite body consisting of (1) a metal substrate (the aircraft blade), (2) an inner layer of insulation, (3) a heating element, (4) an outer layer of insulation, and (5) an abrasion shield. Figure 1-1 depicts a two-dimensional cut-away view of the typical construction of an electrothermal de-icer pad, as well as a representative set of materials and thicknesses used for fabrication. The cross-section shown represents a view of the heater pad normal to the run of the heating elements.

The heating element usually employed in an electrothermal de-icer pad consists either of a woven mat of wires and glass fibers or of multiple strips of resistance ribbon. The gaps which exist between the heating elements can reduce the effectiveness of the heating pad de-icing performance, causing non-uniform melting of the ice. The two insulation layers, which usually consist of a resin impregnated glass cloth, serve to provide electrical insulation for the heating element. In order to direct more heat flow toward the ice layer, it is necessary to use a greater thickness for the inner insulation than for the outer insulation. The abrasion shield serves to protect the de-icer pad from rain erosion as well as dust/sand erosion, and to provide more uniform heating, thus minimizing cold spots above the heater gaps.



Layer	Material	Thickness (Hr)	Diffusivity (Ft^2/In)
Substrate	755-T6 Aluminum	0.087	1.65
Inner Insulation	Epoxy/Glass	0.050	0.0087
Heater	Nichrome	0.004	0.138
Outer Insulation	Epoxy/Glass	0.010	0.0087
Abrasion Shield	304 Stainless	0.012	0.15
Ice		0.250	0.0445

Figure 1-1 Typical Materials and Construction of an Electrothermal De-Icer Pad

The ability to predict the performance of an electrothermal de-icer pad is essential to the design and subsequent fabrication of these units. To accomplish this, some method of determining the time-temperature history throughout the pad needs to be developed. Figure 1-2 provides a pictorial representation of an electrothermal heater section that is part of an airfoil, with some indication of the nature of the thermophysics involved. Clearly, the conduction of energy is three-dimensional in nature, and occurs in a curved, composite body. The temperature plot to the right of the figure provides a qualitative representation of a typical temperature distribution. The temperature is highest at the heater center, drops rapidly under the heater (where the insulation is thickest) and less rapidly in the direction of the ice (where the insulation is thinest). Development of an analytical model for such a problem is virtually impossible. A numerical model is more realizable, but even this is somewhat impractical, unless some simplifications are made to the geometry and the thermophysics. Figure 1-3 illustrates three alterations of the full de-icing problem, each having different degrees of problem simplification. The one-dimensional model is the simplest. In this model, all layers are assumed to be planes infinite in extent. The temperature at a given location is assumed to be constant throughout the plane containing that point. It is generally assumed that the layers are in perfect thermal contact and that they have constant material properties.

Stallabrass [2] appears to have been the first to attempt a numerical solution of an electrothermal de-icing problem using a one-dimensional model. His numerical scheme used an explicit finite

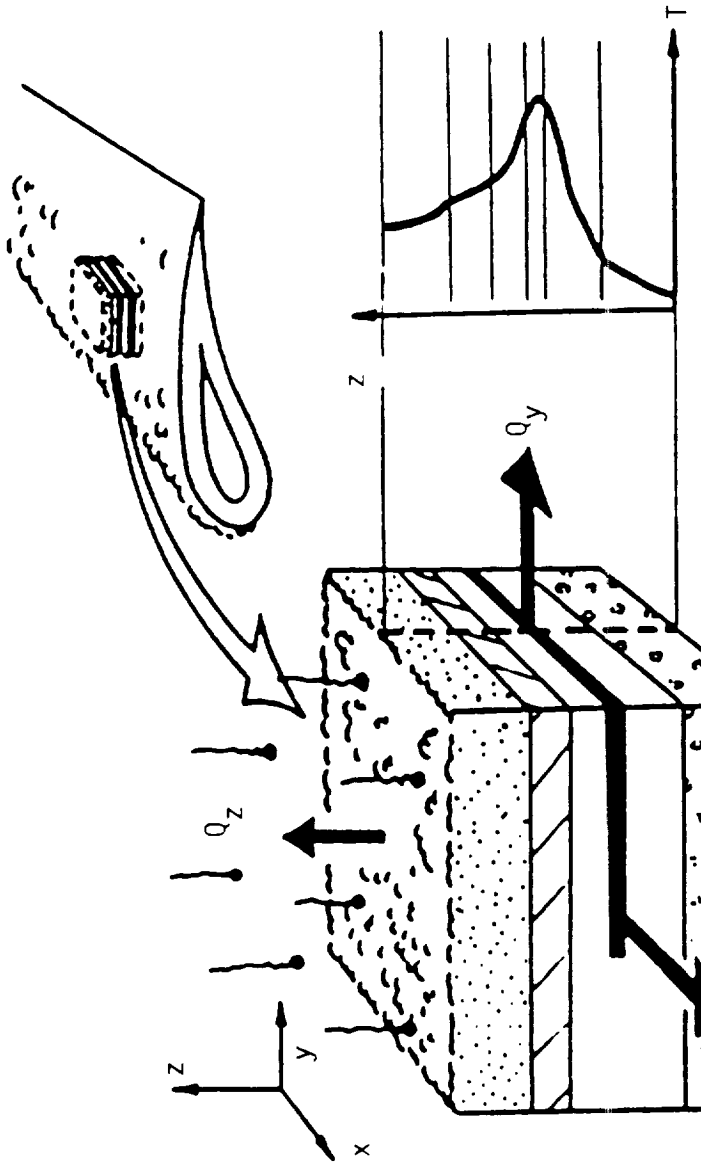
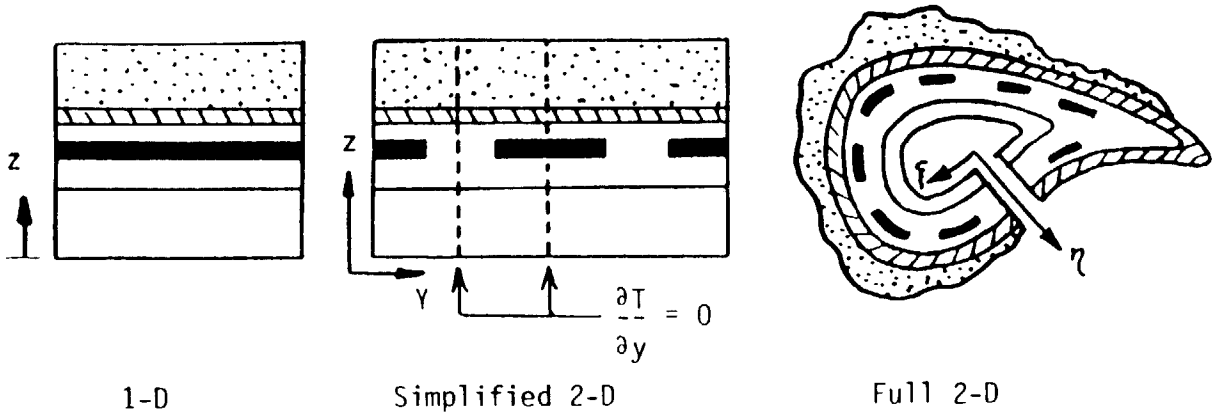


Figure 1-2 Qualitative Representation of the Thermophysics Involved in an Electrothermal De-Icer Pad



1-D Simplified 2-D Full 2-D

Figure 1-3 Degrees of Simplification to the True Thermophysics of the De-Icing Problem.

difference method. Results agreed well with approximate analytical solutions for relatively short real times into the problem. To account for the effect of the phase change on the temperature transients within the composite blade, the node at the ice-abrasion shield interface was held at the freezing temperature until the estimated heat flux into the control volume containing the node was deemed sufficient to cause melting.

Baliga [4] improved the numerical modelling of the same problem by handling the phase change heat transfer via a better approach, making use of the high heat capacity formulation. Marano [5] further improved upon Baliga's numerical formulation by applying the so-called enthalpy method to model the phase change problem. Gent and Cansdale [6], solved the same problem for conduction only (no phase change), and obtained nearly the same results as Marano for conduction only.

The two-dimensional problem, represented by the middle schematic in Fig. 1-3, was solved by Chao [7] and DeWitt, et al. [8]. Chao's work was a direct extension of Marano's one-dimensional numerical formulation and procedures to two dimensions. Of fundamental importance, the effect of the heater gap width on de-icing performance was studied numerically.

Leffel [9] provided detailed experimental results of the thermal transients induced by an electrothermal de-icing unit on a UH1H helicopter rotor blade section. These experimental results were used to validate the codes developed by Chao and Marano. The experimental results revealed that when the layers of a helicopter blade are sufficiently thin, and the curvature sufficiently gradual, Marano's

one-dimensional code yields excellent results over most of the blade. Furthermore, it was found that there are two regions of potentially substantial inaccuracies (depending on heater wattages, material properties, etc.). These are at the immediate edges of the heater banks, and in the region of large curvature at the leading edge of the blade that wraps around the nose block.

Chao's code can model the heater edges, but it can not handle the variable thickness introduced by the nose block, nor the high degree of curvature. Thus, it is necessary to develop a model that can account for these difficulties. This development is pictorially represented by the third schematic of Fig. 1-3. The creation of a numerical code capable of accurately predicting electrothermal de-icer pad thermal transients in more complex regions of the blade is the topic of this work.

Overview of Strategy to Solve Problem

The primary modelling difficulty that must be faced in this study is that due to the irregularity of the blade-layer geometry. There are essentially three approaches that can be taken to account for irregular body curvature. The first is to overlay the irregular geometry with a regular grid. Those points not falling directly on the boundary will require an interpolating scheme that must be incorporated either directly or indirectly into the computational algorithms of the field equations. There will undoubtedly be some degree of inaccuracy introduced into the solution as a consequence. If the solution desired is in a region reasonably far removed from the boundary, this approach will generally yield excellent results. If, however, the solution

desired is located at the boundary, the inaccuracies introduced may be unacceptable. Such is the case in this problem, where the critical temperatures and the initial change of phase occur at the boundary formed by the abrasion shield and the ice layer. Moreover, this approach is computationally quite time consuming.

The second approach is the finite element method. This technique has become fairly common and affords many advantages in problems having irregular geometries. The standard method needed to achieve a solution, however, is by an inversion of the appropriate system matrix. For problems requiring a large number of elements, a large matrix develops. The inversion of this matrix is generally performed by iterative means, thus affording no computational advantage over other techniques. Also, the formuation of the problem required for more complex governing equations will involve the inversion of a multiple number of large matrices for a "matrix statement" equivalent [10], [11] of the field.

In recent years, a finite difference alternative has arisen that accurately models a boundary of irregular shape. By this method the body (or bodies) in the physical domain is numerically transformed into a rectangular region in the computational plane. For a layered body, the transformation produces a rectangle having the same number of layers as the composite in the physical plane. This mapping procedure, known as body fitted coordinate generation, was first developed by Thompson, et al [12], [13], [14], [15], and has become widely used in numerical simulations of field problems [16], [17], [18], [19], [20]. The principal advantage of the procedure is that any set of equations may be numerically solved in a rectangular region on a uniformally spaced grid

system. Thus, numerical interpolation between any irregular boundary and adjacent interior grid points can be avoided. Figure 1-4 depicts the mapping strategy involved for the problem at hand. The numerical/ computational strategy needed essentially reduces to a finite difference solution of a series of stacked, rectangular slabs. The primary disadvantage of this technique is the necessity of spacially transforming those portions of any relevant equations having a spacial dependency. Depending on the equations to be transformed, the resulting set of equations may become much more complex. Obviously, this has the potential of making a numerically stable set of algorithms more difficult to obtain.

This work represents the first known attempt at solving a layered heat transfer problem that includes phase change in an irregular geometry. The following chapter develops the spacial transformation equations, and the operators needed to transform the governing equations. The conduction equations are then transformed in Chapter 3, with the phase change equations being transformed in Chapter 4. These transformed equations are then used to simulate an iced airfoil, and the predictions are compared with experimental test data.

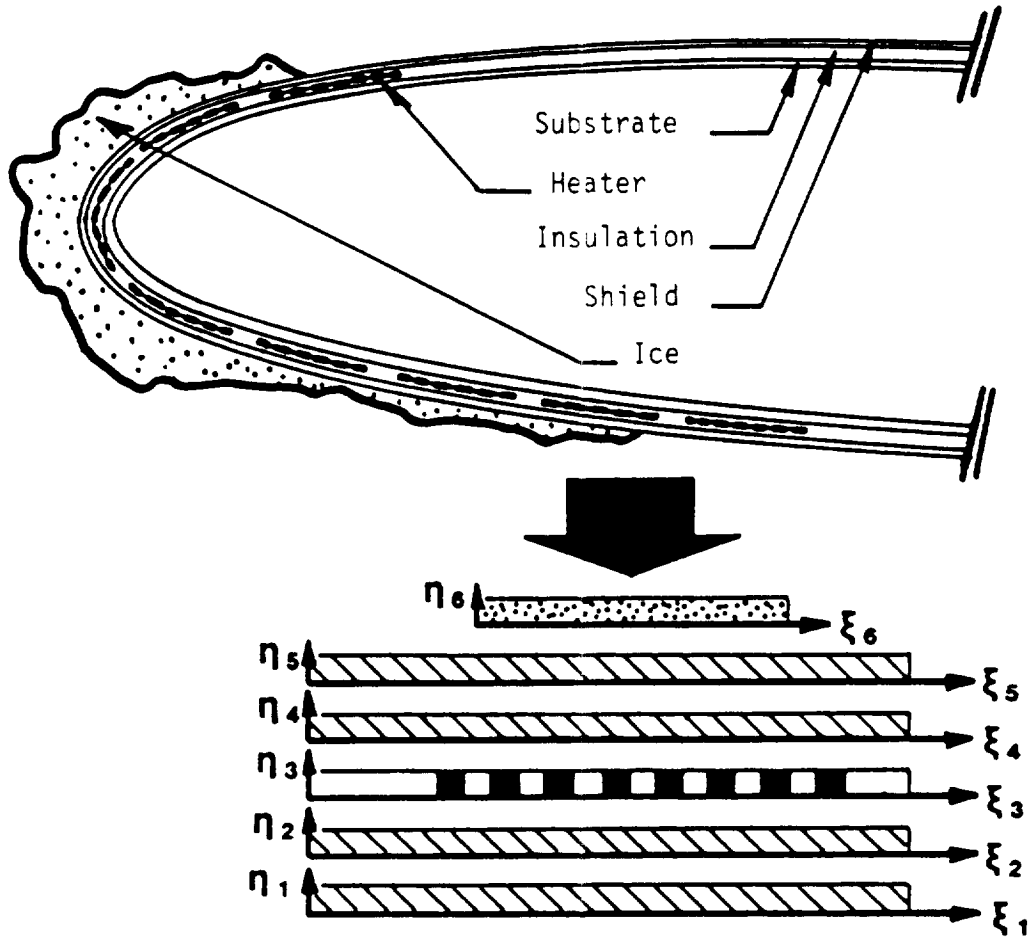


Figure 1-4 Pictorial Representation of how the Composite Blade Geometry is Mapped with Body Fitted Coordinate Transform

CHAPTER 2 - BODY-FITTED TRANSFORMATION

In order to obtain a grid in the transformed plane a method with a system of generating equations needs to be developed. The system of equations that was used to generate the grid was Laplace's equations. Virtually any partial differential equation can be used. Thompson, et al. [15] presents alternatives by adding additional terms to Laplace's equation that skews the grid in a desired direction where exceptionally large gradients of a variable are expected. Laplace's equation was chosen since this equation type is closest to the form of the equations that was solved in the transformed plane. Thus, there are two rectangular coordinate systems that are interrelated through the equations that generate the grid in the transformed plane: x-y in the real plane, and ξ-η in the transformed plane.

It should be noted that for composite bodies, each region must be mapped separately to insure accurate modeling of the true geometry. The initial step of the mapping is to assign boundary points in the real plane, which become boundary conditions in the transformed plane. Thus by mapping each region separately, more accurate geometric modeling is achieved. The assigned boundary points for two regions having a boundary in common must be the same to insure a continuous grid between regions. This makes the subsequent application of boundary condition equations much less complex.

Since the generating equation is initially written in the real plane, an exchange of variables must occur to obtain an equivalent set of equations in the tranformed plane governing the distribution of x and y coordinates in that region. This is the set of equations that is subsequently used to generate a grid. These equations will be developed first, followed by a physical interpretation of the transformation.

Derivation of Body-Fitted Coordinate Transform Equations

The partial differential equation used to generate the grid in the transformed plane, Laplace's equation, is written as

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = 0 \qquad (2\text{-}1)$$

$$\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = 0 \qquad (2\text{-}2)$$

The first step in performing the required exchange of variables is to establish derivatives for the inverse transform. In general notation the inverse transform is represented as

$$x = f(\xi, \eta) \qquad (2\text{-}3)$$

$$y = g(\xi, \eta) \qquad (2\text{-}4)$$

The total derivative of each variable is

$$dx = \frac{\partial f}{\partial \xi} d\xi + \frac{\partial f}{\partial \eta} d\eta \qquad (2\text{-}5)$$

$$dy = \frac{\partial g}{\partial \xi} d\xi + \frac{\partial g}{\partial \eta} d\eta \tag{2-6}$$

Multiplying Eq. (2-5) by $\partial g/\partial \eta$ and Eq. (2-6) by $\partial f/\partial \eta$ and subtracting yields

$$\frac{\partial g}{\partial \eta} dx - \frac{\partial f}{\partial \eta} dy = \left(\frac{\partial f}{\partial \xi} \frac{\partial g}{\partial \eta} - \frac{\partial g}{\partial \xi} \frac{\partial f}{\partial \eta} \right) d\xi \tag{2-7}$$

Similarly, Eqs. (2-5) and (2-6) may be multiplied by $\partial g/\partial \xi$ and $\partial f/\partial \xi$, respectively, and subtracted to give

$$\frac{\partial g}{\partial \xi} dx - \frac{\partial f}{\partial \xi} dy = \left(\frac{\partial g}{\partial \xi} \frac{\partial f}{\partial \eta} - \frac{\partial f}{\partial \xi} \frac{\partial g}{\partial \eta} \right) d\eta \tag{2-8}$$

The Jacobian of the transformation may be defined as

$$J \equiv \begin{vmatrix} \frac{\partial f}{\partial \xi} & \frac{\partial f}{\partial \eta} \\ \frac{\partial g}{\partial \xi} & \frac{\partial g}{\partial \eta} \end{vmatrix} = \frac{\partial f}{\partial \xi} \frac{\partial g}{\partial \eta} - \frac{\partial g}{\partial \xi} \frac{\partial f}{\partial \eta} \tag{2-9}$$

Inserting this into Eqs. (2-7) and (2-8) and rearranging produces

$$d\xi = \frac{1}{J} \frac{\partial g}{\partial \eta} dx - \frac{1}{J} \frac{\partial f}{\partial \eta} dy \tag{2-10}$$

$$d\eta = -\frac{1}{J} \frac{\partial g}{\partial \xi} dx + \frac{1}{J} \frac{\partial f}{\partial \xi} dy \tag{2-11}$$

Since ξ and η are transforms of x and y, the derivatives of ξ and η are written as

$$d\xi = \frac{\partial \xi}{\partial x}\, dx + \frac{\partial \xi}{\partial y}\, dy \tag{2-12}$$

$$d\eta = \frac{\partial \eta}{\partial x}\, dx + \frac{\partial \eta}{\partial y}\, dy \tag{2-13}$$

Comparing like multiples between Eqs. (2-10) and (2-12), and between Eqs. (2-11) and (2-13), yields the following quantities:

$$\frac{\partial \xi}{\partial x} = \frac{1}{J}\frac{\partial g}{\partial \eta} = \frac{1}{J}\frac{\partial y}{\partial \eta} \tag{2-14}$$

$$\frac{\partial \xi}{\partial y} = \frac{1}{J}\frac{\partial f}{\partial \eta} = \frac{1}{J}\frac{\partial x}{\partial \eta} \tag{2-15}$$

$$\frac{\partial \eta}{\partial x} = \frac{1}{J}\frac{\partial g}{\partial \xi} = \frac{1}{J}\frac{\partial y}{\partial \xi} \tag{2-16}$$

$$\frac{\partial \eta}{\partial y} = \frac{1}{J}\frac{\partial f}{\partial \xi} = \frac{1}{J}\frac{\partial x}{\partial \xi} \tag{2-17}$$

Using these indentities, the derivatives can now be rewritten as differential operators, i.e.,

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} = \frac{1}{J}\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi} - \frac{1}{J}\frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta} \tag{2-18}$$

$$\frac{\partial}{\partial y} = \frac{\partial \xi}{\partial y}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial y}\frac{\partial}{\partial \eta} = -\frac{1}{J}\frac{\partial x}{\partial \eta}\frac{\partial}{\partial \xi} + \frac{1}{J}\frac{\partial x}{\partial \xi}\frac{\partial}{\partial \eta} \tag{2-19}$$

$$\frac{\partial^2}{\partial x^2} = \frac{\partial}{\partial x}\,\frac{\partial}{\partial x}$$

$$= \left(\frac{1}{J}\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi} - \frac{1}{J}\frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta}\right)\left(\frac{1}{J}\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi} - \frac{1}{J}\frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta}\right)$$

$$= \frac{1}{J^2}\left[\left(\frac{\partial y}{\partial \eta}\right)^2\frac{\partial^2}{\partial \xi^2} - 2\frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial^2}{\partial \xi \partial \eta} + \left(\frac{\partial y}{\partial \xi}\right)^2\frac{\partial^2}{\partial \eta^2}\right] \qquad (2\text{-}20)$$

Similarly,

$$\frac{\partial^2}{\partial y^2} = \frac{1}{J^2}\left[\left(\frac{\partial x}{\partial \eta}\right)^2\frac{\partial^2}{\partial \xi^2} - 2\frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial^2}{\partial \xi \partial \eta} + \left(\frac{\partial x}{\partial \xi}\right)^2\frac{\partial^2}{\partial \eta^2}\right] \qquad (2\text{-}21)$$

The Laplacian operator can now be written as

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = \frac{1}{J^2}\left\{\left[\left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2\right]\frac{\partial^2}{\partial \xi^2} - 2\left(\frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\right)\frac{\partial^2}{\partial \xi \partial \eta} + \left[\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2\right]\frac{\partial^2}{\partial \eta^2}\right\} \qquad (2\text{-}22)$$

If the following are defined,

$$\alpha = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2 \qquad (2\text{-}23)$$

$$\beta = \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta} \qquad (2\text{-}34)$$

$$\gamma = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2 \qquad (2\text{-}25)$$

Equation (2-22) becomes

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = \frac{1}{J^2}\left(\alpha \frac{\partial^2}{\partial \xi^2} - 2\beta \frac{\partial^2}{\partial \xi \partial \eta} + \gamma \frac{\partial^2}{\partial \eta^2}\right) \tag{2-26}$$

Equation (2-26) can now be used directly to obtain the spacial field equations needed to generate the coordinate lines in the physical plane. Beginning with Laplace's equation in the real plane as the generating equation, these equations are

$$\alpha \frac{\partial^2 x}{\partial \xi^2} - 2\beta \frac{\partial^2 x}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 x}{\partial \eta^2} = 0 \tag{2-27}$$

$$\alpha \frac{\partial^2 y}{\partial \xi^2} - 2\beta \frac{\partial^2 y}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 y}{\partial \eta^2} = 0 \tag{2-28}$$

Because Eqs. (2-27) and (2-28) have the variable coefficients alpha, beta and gamma in common (which are derivative functions of x and y), they are coupled and must therefore be solved simultaneously.

In order to formulate an equality of the normal flux boundary condition in the transformed plane, a vector derivative transformation needs to be developed. The unit normal to any graph f(x,y) = c is given by

$$\vec{n}_f = \frac{\nabla f}{|\nabla f|} \tag{2-29}$$

Consequently the unit normal to η and ξ coordinate lines are

$$\vec{n}_\eta = \frac{\nabla \eta}{|\nabla \eta|} \tag{2-30}$$

$$\vec{n}_\xi = \frac{\nabla \xi}{|\nabla \xi|} \tag{2-31}$$

The solution of Eqs. (2-27) and (2-28) results in a distribution of x and a distribution of y in the solution plane. Thus, in order to represent a unit normal vector in this plane, its 'orientation' in the real plane must be accounted for. The general operator in the real plane for a directional derivative is

$$\nabla = \frac{\partial}{\partial x}\,\vec{i} + \frac{\partial}{\partial y}\,\vec{j} \tag{2-32}$$

Applying Eqs. (2-18) and (2-19) to Eq. (2-32), the general operator in the transformed plane for a directional derivative is

$$\nabla = \frac{1}{J}\left[\left(\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta}\right)\vec{i} + \left(\frac{\partial x}{\partial \xi}\frac{\partial}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial}{\partial \xi}\right)\vec{j}\right] \tag{2-33}$$

It should be noted that the directional derivatives are needed in the real plane, but are being represented by information that is contained in the transformed plane. Thus, the symbols $\vec{i}$ and $\vec{j}$ refer to the real plane and do not need to be transformed.

Using Eq. (2-33), $\nabla\eta$ and $\nabla\xi$ clearly become

$$\nabla\eta = \frac{1}{J}\left(-\frac{\partial y}{\partial \xi}\,\vec{i} + \frac{\partial x}{\partial \xi}\,\vec{j}\right) \tag{2-34}$$

$$\nabla\xi = \frac{1}{J}\left(\frac{\partial y}{\partial \eta}\,\vec{i} - \frac{\partial x}{\partial \eta}\,\vec{j}\right) \tag{2-35}$$

In the next section a physical interpretation of the coefficients alpha, Eq. (2-23), and gamma, Eq. (2-25), and how they are related to arc lengths of cells in the physical plane will be presented. It will be

shown that

$$|\nabla\eta| = \frac{1}{J}\left[\left(\frac{\partial y}{\partial \xi}\right)^2 + \left(\frac{\partial x}{\partial \xi}\right)^2\right] = \frac{\sqrt{\gamma}}{J} \tag{2-36}$$

$$|\nabla\xi| = \frac{1}{J}\left[\left(\frac{\partial y}{\partial \eta}\right)^2 + \left(\frac{\partial x}{\partial \eta}\right)^2\right] = \frac{\sqrt{\alpha}}{J} \tag{2-37}$$

Hence, utilizing Eqs. (2-34) through (2-37), Eqs. (2-30) and (2-31) become

$$\vec{n}_\eta = \frac{1}{\sqrt{\gamma}}\left(-\frac{\partial y}{\partial \xi}\vec{i} + \frac{\partial x}{\partial \xi}\vec{j}\right) \tag{2-44}$$

$$\vec{n}_\xi = \frac{1}{\sqrt{\alpha}}\left(\frac{\partial y}{\partial \eta}\vec{i} - \frac{\partial x}{\partial \eta}\vec{j}\right) \tag{2-45}$$

The temperature gradient at any point in the point plane is defined by

$$\nabla T = \frac{\partial T}{\partial x}\vec{i} + \frac{\partial T}{\partial y}\vec{j} \tag{2-40}$$

Applying Eqs. (2-18) and (2-19) yields the temperature gradient along any point in the transformed plane as

$$\nabla T = \frac{1}{J}\left[\left(\frac{\partial y}{\partial \eta}\frac{\partial T}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial T}{\partial \eta}\right)\vec{i} + \left(\frac{\partial x}{\partial \xi}\frac{\partial T}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial T}{\partial \xi}\right)\vec{j}\right] \tag{2-41}$$

The gradient of temperature normal to a line of constant η becomes

$$\frac{\partial T}{\partial n_\eta} = \vec{n}_\eta \cdot \nabla T$$

$$= \frac{1}{\sqrt{\gamma}} \left(-\frac{\partial y}{\partial \xi} \vec{i} + \frac{\partial x}{\partial \xi} \vec{j} \right) \cdot$$

$$\frac{1}{J} \left[\left(\frac{\partial y}{\partial \eta} \frac{\partial T}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial T}{\partial \eta} \right) \vec{i} + \left(\frac{\partial x}{\partial \xi} \frac{\partial T}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial T}{\partial \xi} \right) \vec{j} \right]$$

$$= \frac{1}{J\sqrt{\gamma}} \left\{ \left[\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2 \right] \frac{\partial T}{\partial \eta} + \right.$$

$$\left. - \left(\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right) \frac{\partial T}{\partial \xi} \right\}$$

$$= \frac{1}{J\sqrt{\gamma}} \left(\gamma \frac{\partial T}{\partial \eta} - \beta \frac{\partial T}{\partial \xi} \right) \qquad (2\text{-}42)$$

The gradient of temperature normal to a line of constant ξ becomes

$$\frac{\partial T}{\partial n_\xi} = \vec{n}_\xi \cdot \nabla T$$

$$= \frac{1}{\sqrt{\alpha}} \left(\frac{\partial y}{\partial \eta} \vec{i} - \frac{\partial x}{\partial \eta} \vec{j} \right) \cdot$$

$$\frac{1}{J} \left[\left(\frac{\partial y}{\partial \eta} \frac{\partial T}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial T}{\partial \eta} \right) \vec{i} + \left(\frac{\partial x}{\partial \xi} \frac{\partial T}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial T}{\partial \xi} \right) \vec{j} \right]$$

$$= \frac{1}{J\sqrt{\alpha}} \left\{ \left[\left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \right] \frac{\partial T}{\partial \xi} + \right.$$

$$\left. - \left(\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \right) \frac{\partial T}{\partial \eta} \right\}$$

$$= \frac{1}{J\sqrt{\alpha}} \left(\alpha \frac{\partial T}{\partial \xi} - \beta \frac{\partial T}{\partial \eta} \right) \qquad (2\text{-}43)$$

Equations (2-42) and (2-43) are those needed in formulating the normal heat flux condition for a boundary common between two regions.

Physical Interpretation and Use of the Body Fitted Coordinate Transformation

Although field equations are written in terms of physical plane coordinates (x,y; r,θ; etc.), their analytical or numerical solution is generally only convenient for a limited number of smooth, regular geometries. If the geometry is irregular the solution in the physical plane can become quite tedious. An irregular body can be mapped into a much simpler shape by the use of a body fitted coordinate transformation, as shown in Figs. 2-1 through 2-3.

If the composite body is simply connected it will have an interior solid which requires a slightly different treatment. This is also illustrated in the bottom two rectangles of Fig. 2-3. Care must be taken to match the edges of the rectangle with corresponding portions of the interface for the layer which surrounds the solid.

The overall result of the transformation is the creation of a series of connected rectangular strips. To generate the coordinate system within these strips, two elliptic partial differential equations, Eqs. (2-1) and (2-2), with Dirichlet boundary conditions are solved. The field equations used to develop the coordinate system are Eqs.(2-27) and (2-28), which are the end result of the exchange of variables needed for the transformed plane. The variable coefficients alpha, beta, and gamma are defined by Eqs. (2-23) through (2-25).

Figures 2-4 through 2-6 conceptually show the procedure needed to generate a grid, and a sample of what the final results might yield.

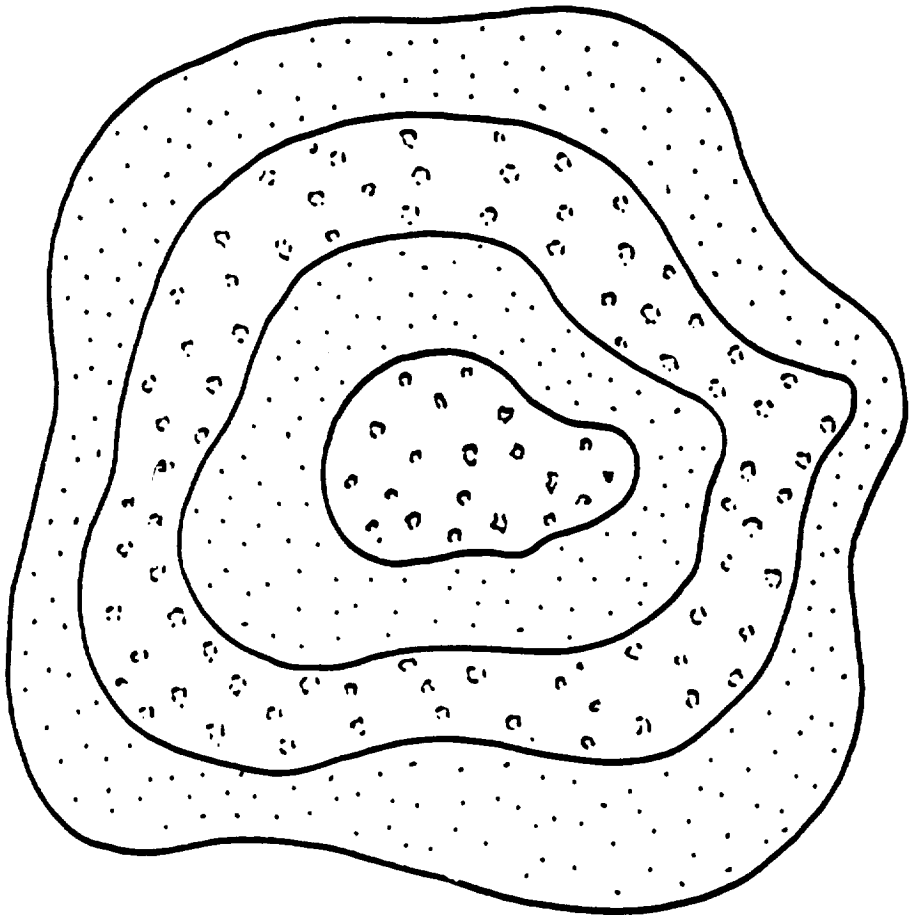


Figure 2-1: A General, Irregularly Shaped Composite Body.

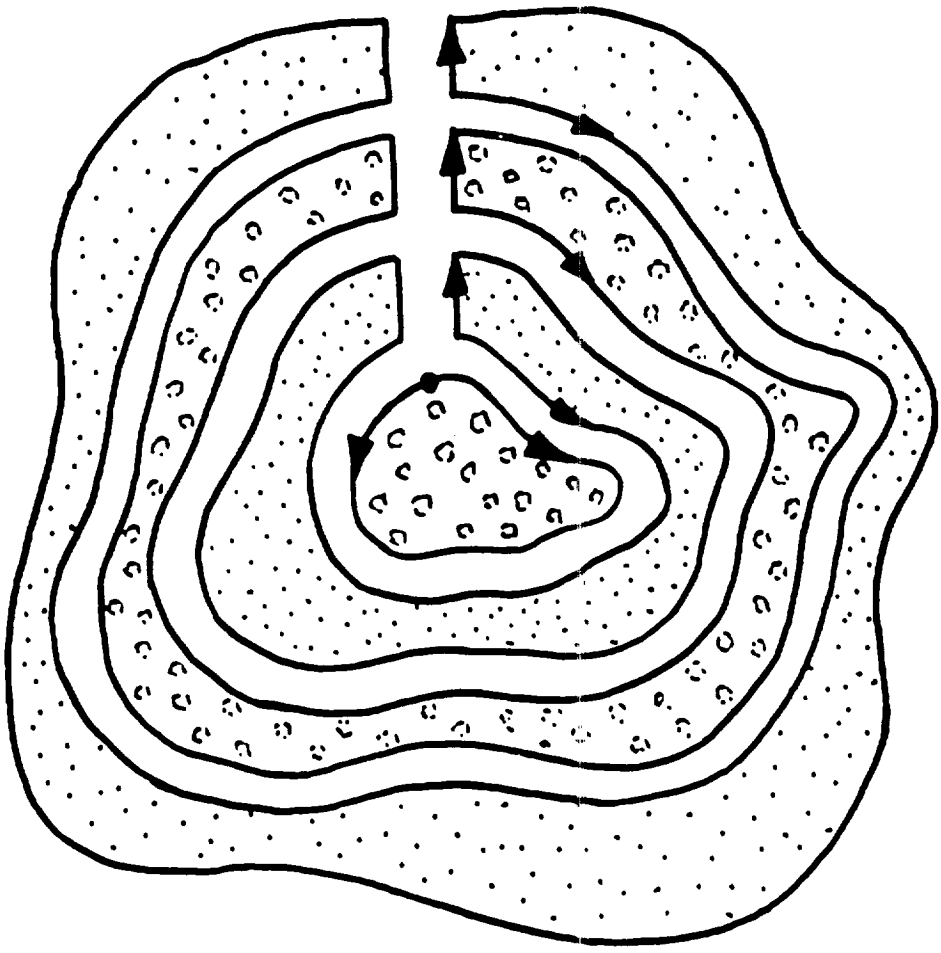


Figure 2-2: Division and Separation of an Irregularly Shaped Composite Body for Subsequent Mapping in the Transformed Plane.

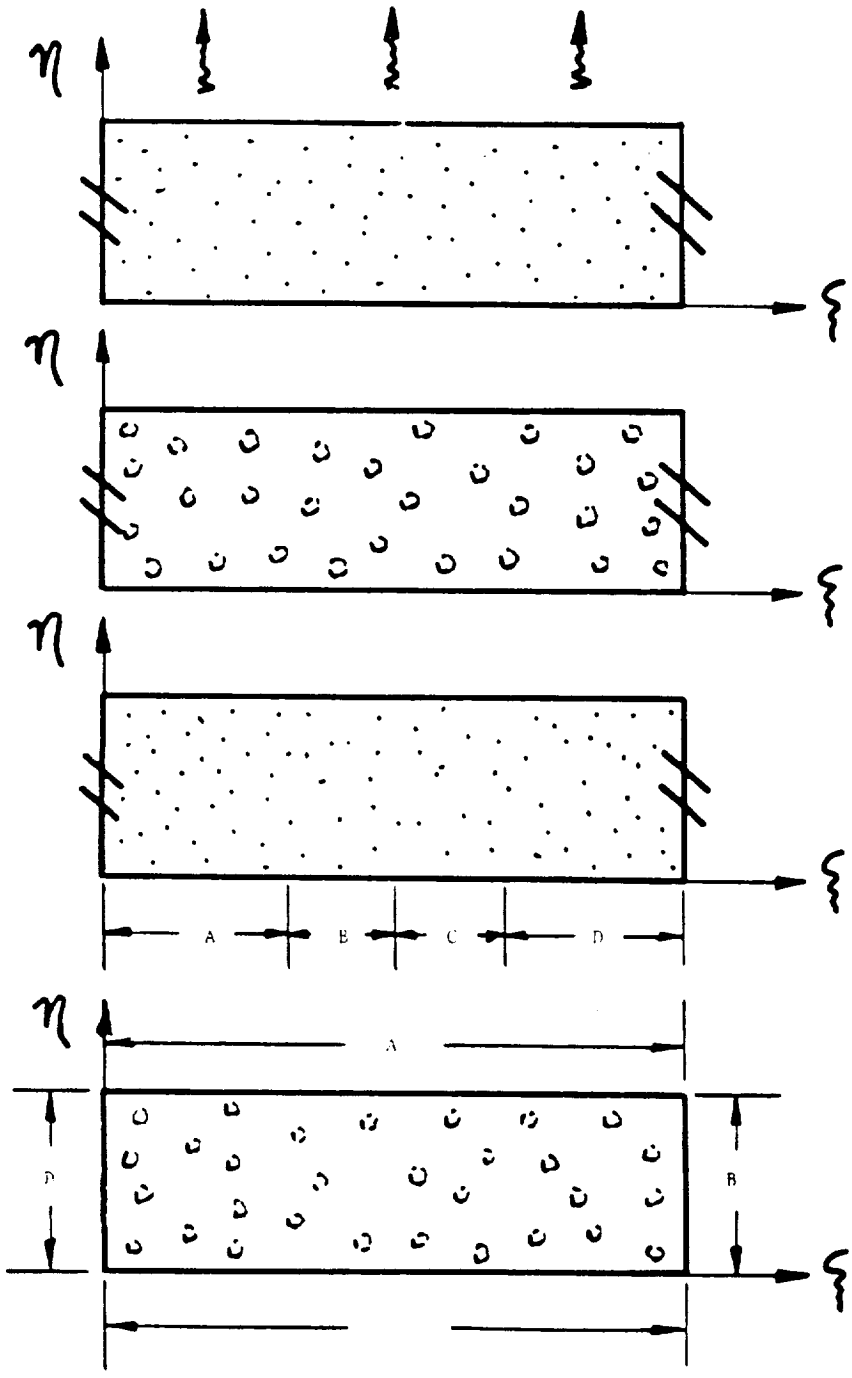


Figure 2-3: Spacial Appearance of an Irregular Composite Body in the Transformed Plane.

A set of x-y coordinate pairs need to be identified for the boundaries of a given region in the physical plane. These values are then assigned to boundary points for a rectangular region in the transformed plane. The distribution of x,y coordinates in the ξ, η plane are tied together by the partial differential equation used to generate the grid. Equations (2-27) and (2-28) are then solved simultaneously on the field, with x and y coordinate values as the unknowns. Upon convergence, the solution yields a distribution of x and y coordinates within the field, thereby accomplishing the goal of generating a grid. If the resulting solution is in turn mapped onto the physical plane, its appearance would take on the shape of a collection of quadrilaterals that approximate the original regions. Thus, it can be seen that an improved approximation is achieved as the number of boundary points chosen is measured, especially on those portions of the boundary that have a high degree of curvature. The numerical solution requires the same number of arbitrary points on the inner boundary as on the outer boundary to maintain a rectangular grid in the transformed plane. Also, composite or layered bodies should have the same boundary points for shared boundaries, as otherwise a temperature solution involving more than one region would be very difficult to numerically implement. Figure 2-6 shows that for a conduction heat transfer problem in the transformed plane there are always three solution fields. The x and y solution fields are determined first (which generates the grid), and then on this grid a solution is obtained for a field equation(s) of interest. Thus, each nodal coordinate for a specified value of ξ and η has at least three

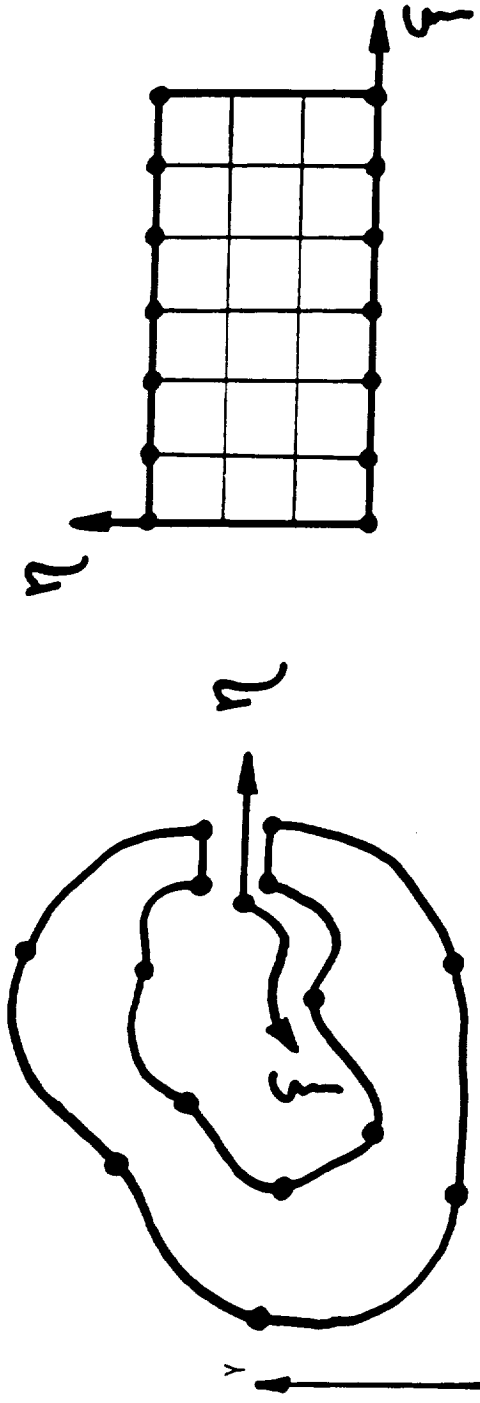


Figure 2-4: Assignment of x-y Coordinate Pairs in the Physical Domain to Edge Boundary Values (Dirichlet Boundary Condition) in the Transformed Plane.

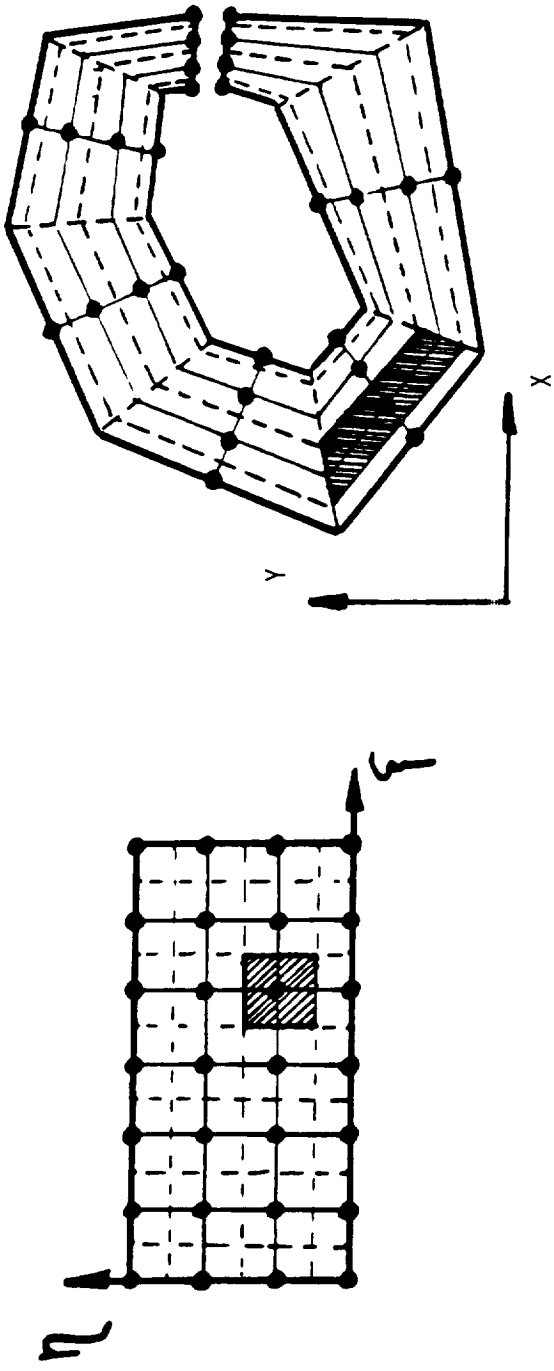


Figure 2-5: Results of a Converged Solution in the Transformed Plane Mapped onto the Physical Plane.

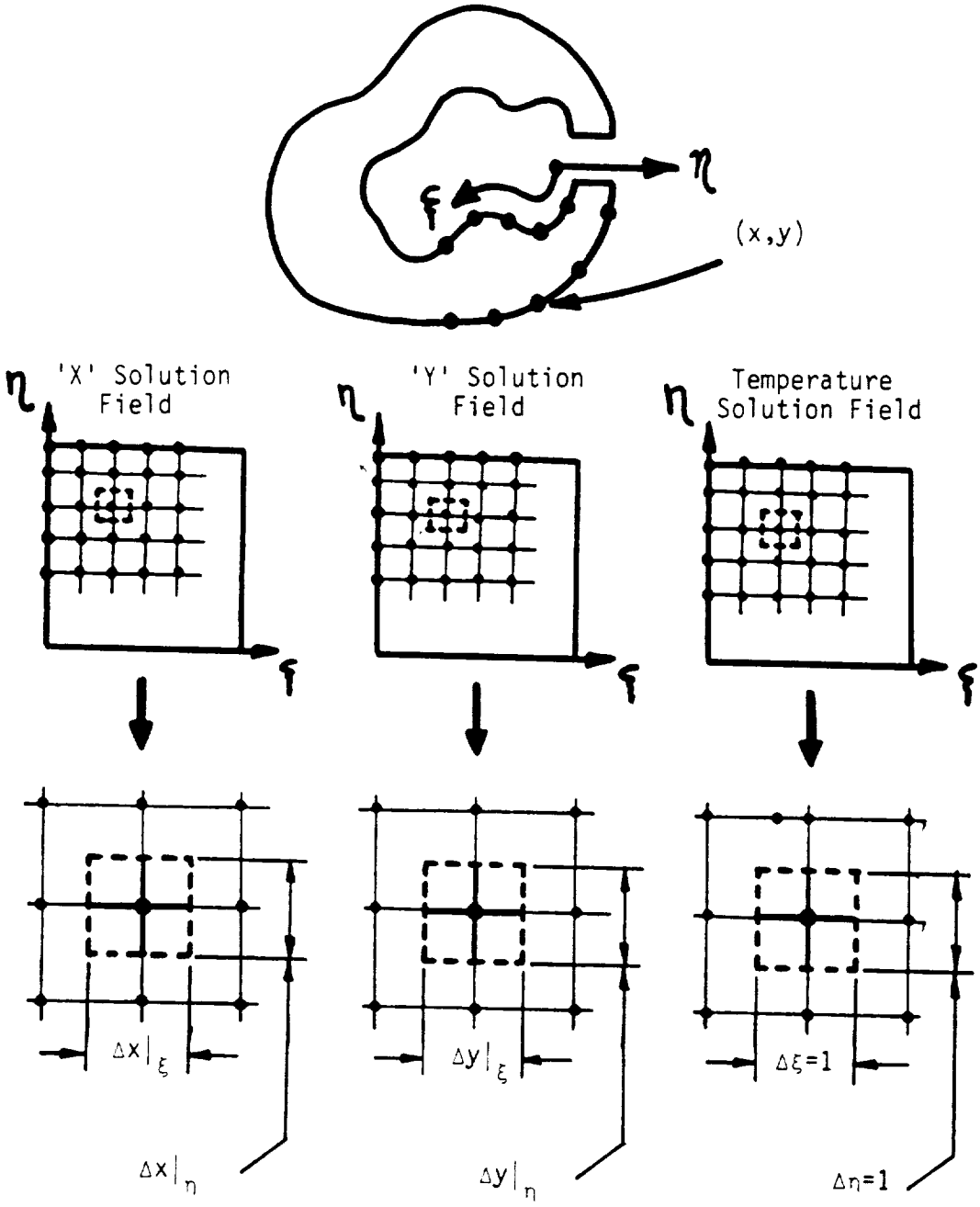


Figure 2-6: The Three Solution Fields Present for each Region.

values associated with it: an x coordinate, a y coordinate, and, in the case shown, a temperature. All of these values at each node must then be used to obtain temperature mappings in the physical plane.

The coefficients alpha, beta and gamma are functions of x and y. In the final solution to the grid generation, there is also a distribution of these three coefficients throughout the field. Thus, these coefficients can change in value from point to point. The physical significance of all the transformation coefficients (including the Jacobian) is illustrated in Figs. 2-7 through 2-10. Figure 2-7 shows an arbitrary quadrilateral drawn onto a square grid in the physical plane. The midpoints of the opposite sides of the edges are connected, which give essentially an average "length" and "width" of the quadrilateral. These dimensions are labled "b" and "a" in Figs. 2-8 and 2-9, respectively. All of the derivatives of x and y with respect to ξ and η are taken as discrete, constant values through a quadrilateral along the lengths a and b. It is inherently assumed that the lengths a and b correspond to lines of constant ξ and η from the transformed plane. The intersection of lines a and b represent, then, the intersection of the lines ξ and η for a given point that was part of the iterated field in the transformed plane.

From reference to Figs. 2-7 through 2-10, it can be seen that the Jacobian of the transformation is the area of the quadrilateral in the physical plane. The square root of alpha is a measure of the average quadrilateral length along the η direction in the physical plane. The square root of gamma is a measure of the average quadrilateral length along the ξ direction in the physical plane. And, finally, beta is a

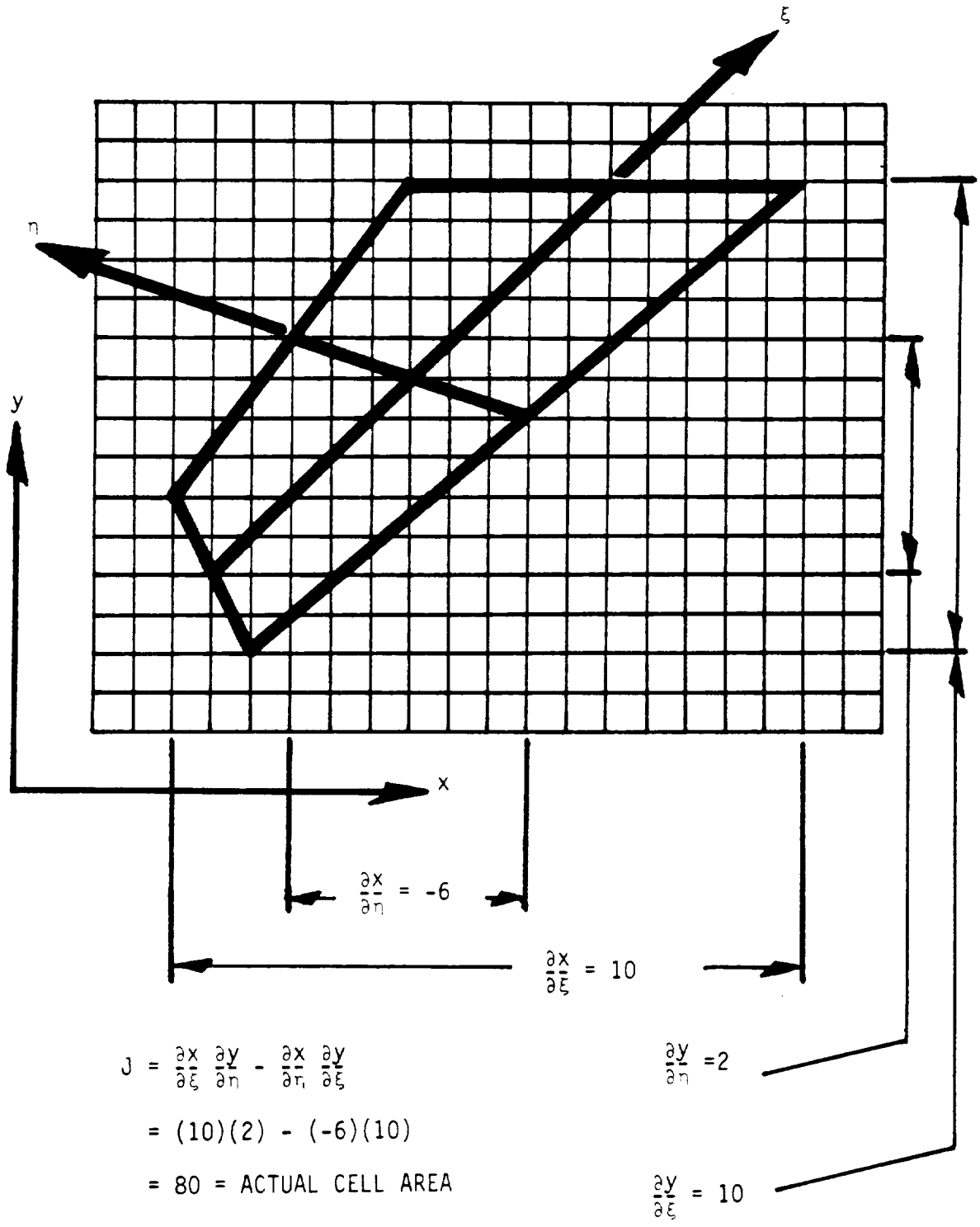
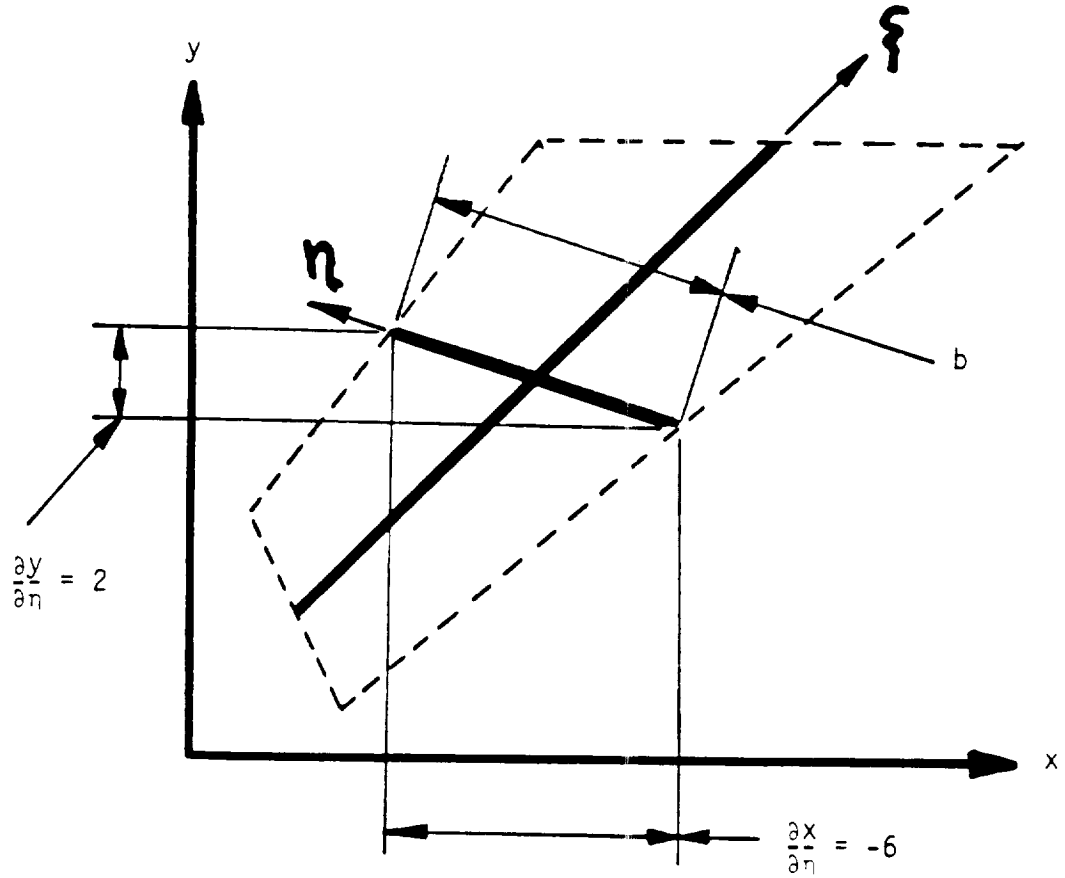


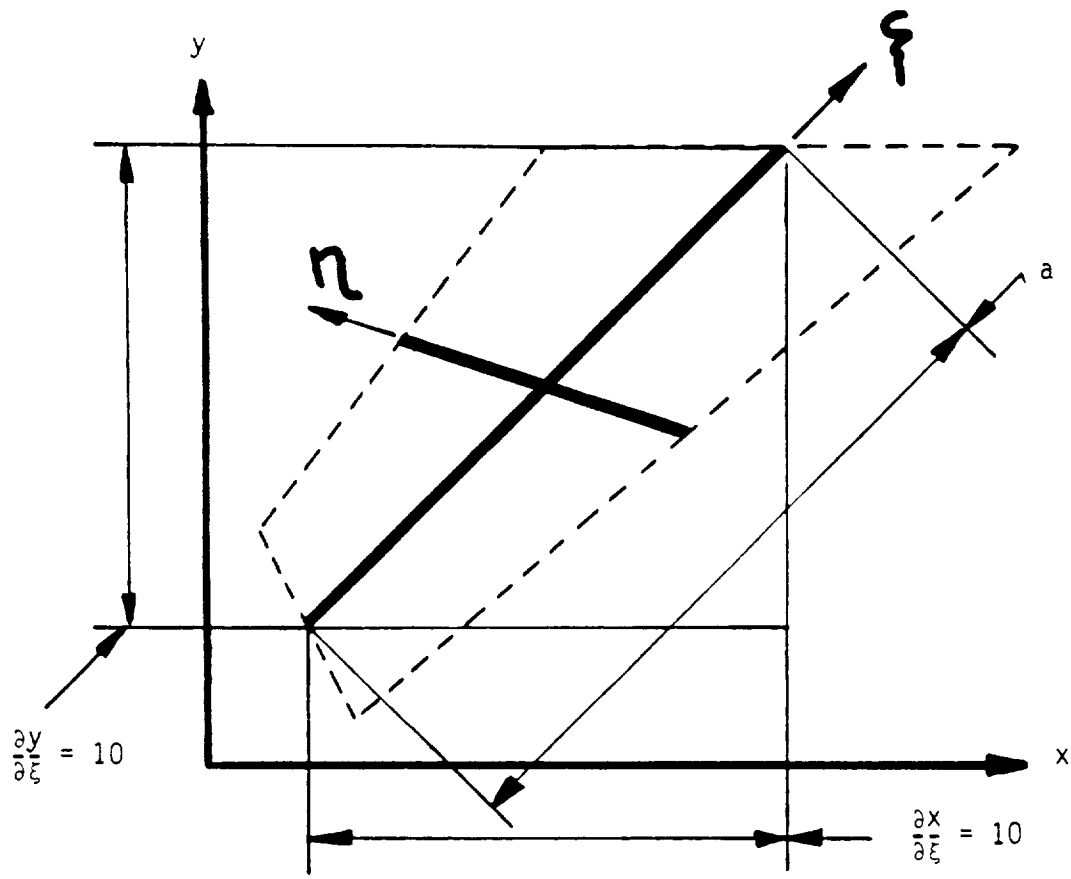
Figure 2-7: Spacial Relationship of a Computational Cell in the Transformed Plane Mapped onto the Real Plane.



$$\alpha = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2 = (-6)^2 + (2)^2 = 40 = b^2$$

$$b = \sqrt{\alpha}$$

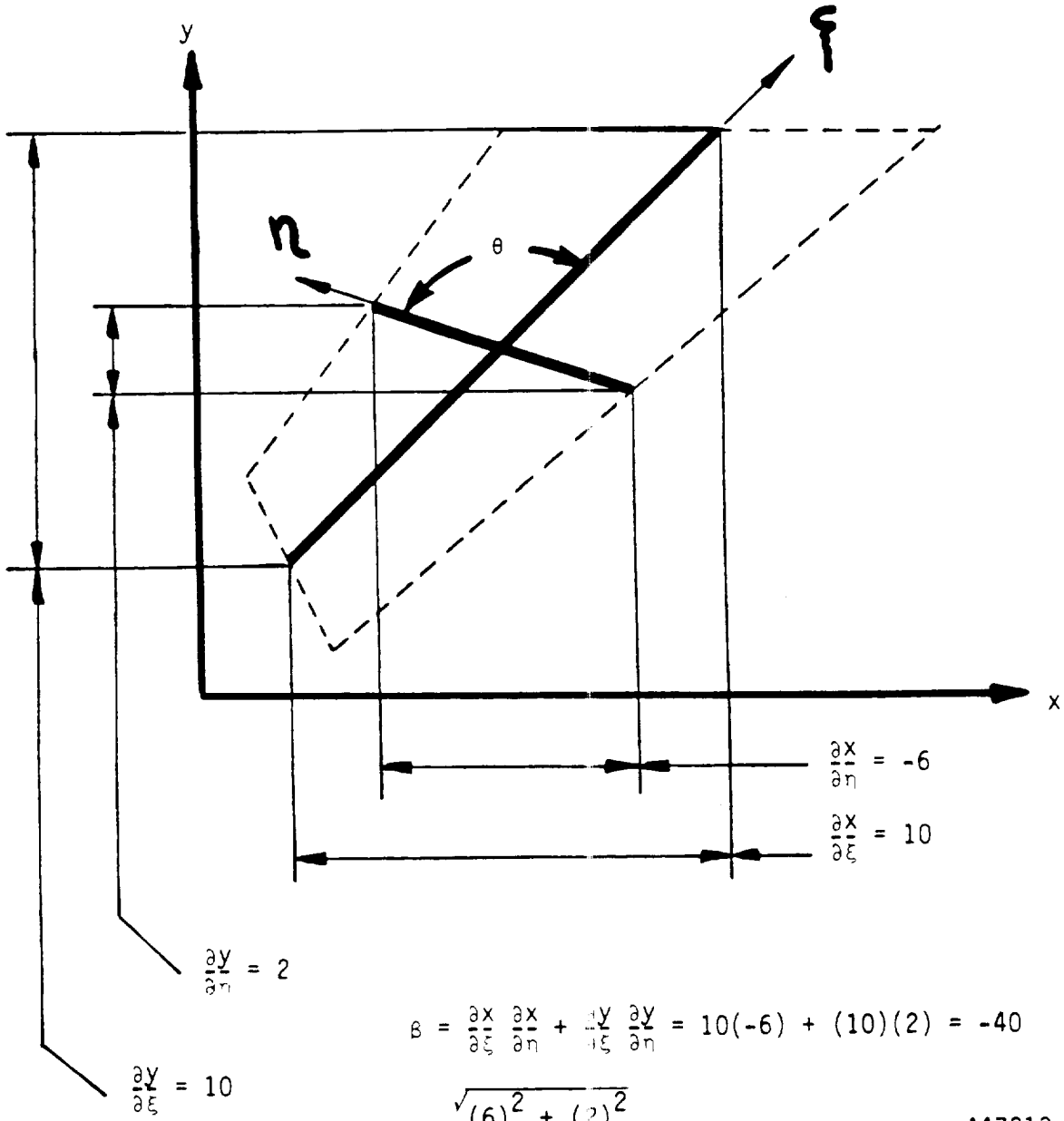
Figure 2-8: Physical Significance of the Transformation Parameter 'Alpha' in the Real Plane.



$$\gamma = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2 = 10^2 + 10^2 = 200 = a^2$$

$$a = \sqrt{\gamma}$$

Figure 2-9: Physical Significance of the Transformation Parameter 'Gamma' in the Real Plane.



$$\beta = \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta} = 10(-6) + (10)(2) = -40$$

$$\cos\theta = \frac{\sqrt{(6)^2 + (2)^2}}{\sqrt{(6)^2 + (2)^2 + (4)^2 + (12)^2}} = -.447213$$

$$\frac{\beta}{\sqrt{\alpha\gamma}} = \frac{-40}{\sqrt{40(200)}} = -.447213 \rightarrow \beta = ab\cos\theta$$

Figure 2-10: Physical Significance of the Transformation Parameter 'Beta' in the Real Plane.

measure of the non-orthogonality of the quadrilateral as measured by the mapping of the ξ and η lines on the corresponding cell/quadrilateral in the physical plane. Thus, the essential dimensions describing the approximate shape of the quadrilaterals in the physical plane (which are the square, uniform computational cells in the transformed plane) are contained within the transformation parameters alpha, beta, gamma and Jacobian.

Numerical Modelling of Body-Fit Transformation Equations

Solving the transformation equations in the computational plane is essentially the same as solving any two field equations that are coupled with Dirichlet boundary conditions. Equations (2-44)* through (2-50)* show how the two field equations have been discretized. All second order derivatives are approximated using three point central differences, and all first order derivatives using two point central differences. Note that the coupling of the equations occurs through the transformation coefficients alpha, beta, and gamma. Equations (2-45) and (2-47) can be solved for x(i,j) and y(i,j) to obtain the required algorithms. Figure 2-11 illustrates a representative portion of the computational grid that Eqs. (2-44) through (2-50) were solved on, explaining the indices i and j.

The iteration strategy is to first sweep through the x field, holding all values of y at the value of the most recent iteration, and then sweep through the y field, holding all values of x fixed from the most recent iteration. The values of the transformation coefficients

---

*Because of their length, these equations are shown on the following page.

$$\alpha \frac{\partial^2 x}{\partial \xi^2} - 2\beta \frac{\partial^2 x}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 x}{\partial \eta^2} = 0 \tag{2-44}$$

$$\alpha(x_{i+1,j} - 2x_{i,j} + x_{i-1,j}) - 2\beta \frac{\frac{x_{i+1,j+1} - x_{i-1,j+1}}{2} - \frac{x_{i+1,j-1} - x_{i-1,j-1}}{2}}{2}$$

$$+ \gamma (x_{i,j+1} - 2x_{i,j} + x_{i,j-1}) = 0 \tag{2-45}$$

$$\alpha \frac{\partial^2 y}{\partial \xi^2} - 2\beta \frac{\partial^2 y}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 y}{\partial \eta^2} = 0 \tag{2-46}$$

$$\alpha(y_{i+1,j} - 2y_{i,j} + y_{i-1,j}) - 2\beta \frac{\frac{y_{i+1,j+1} - y_{i-1,j+1}}{2} - \frac{y_{i+1,j-1} - y_{i-1,j-1}}{2}}{2}$$

$$+ \gamma (y_{i,j+1} - 2y_{i,j} + y_{i,j-1}) = 0 \tag{2-47}$$

$$\alpha = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2 = \left(\frac{x_{i,j+1} - x_{i,j-1}}{2}\right)^2 + \left(\frac{y_{i,j+1} - y_{i,j-1}}{2}\right)^2 \tag{2-48}$$

$$\beta = \frac{\partial y}{\partial \eta}\frac{\partial y}{\partial \xi} + \frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \xi} = \frac{y_{i,j+1} - y_{i,j-1}}{2} \cdot \frac{y_{i+1,j} - y_{i-1,j}}{2}$$

$$+ \frac{x_{i+1,j} - x_{i-1,j}}{2} \cdot \frac{y_{i+1,j} - y_{i-1,j}}{2} \tag{2-49}$$

$$\gamma = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2 = \left(\frac{x_{i+1,j} - x_{i-1,j}}{2}\right)^2 + \left(\frac{y_{i+1,j} - y_{i-1,j}}{2}\right)^2 \tag{2-50}$$

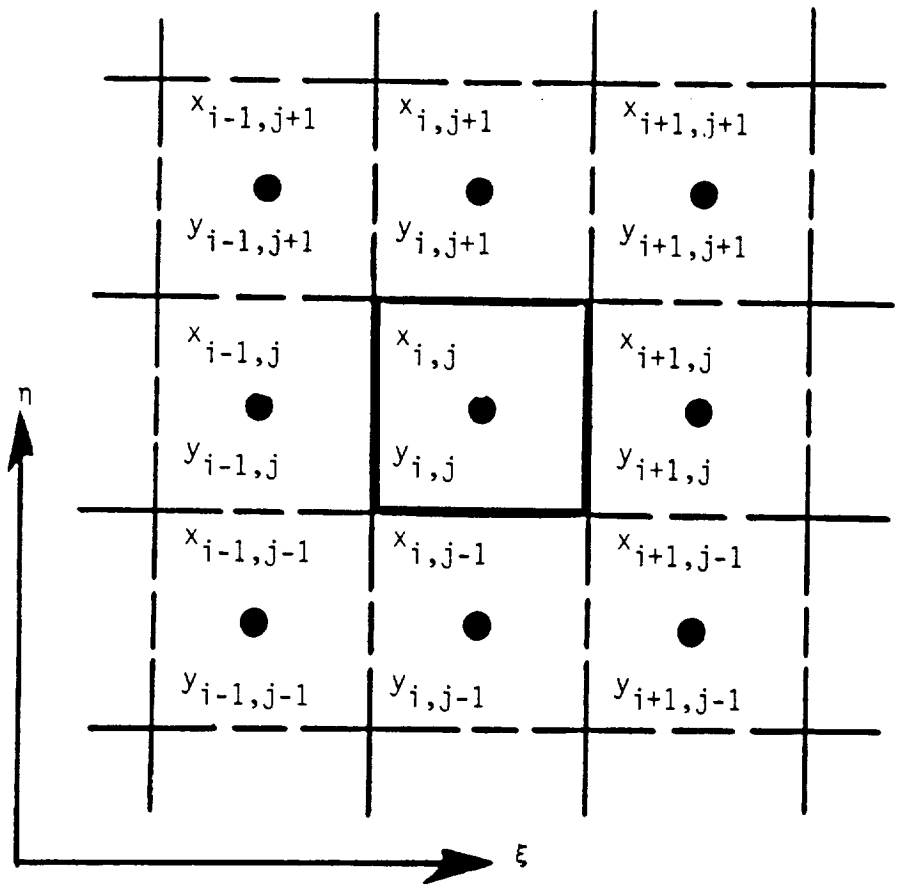


Figure 2-11: Grid Index Notation for Solution of Spacial Transform Algorithms in Transformed Plane

are updated for each point for every iteration to reflect the most recent values of x and y, as the sweep through the field progresses. This procedure continues until all values of x and y are settled i.e., they no longer change to within some specified error.

Prior to the beginning of the iteration, a linear distribution between any two opposing boundary values was assigned to all the field points as an initial guess. This not only reduces the needed CPU time, but also assures that there will be no divide check errors during the first sweep of the field. Also, once convergence has been achieved, the values of the transformation parameters for the boundary points are calculated for use in the boundary condition algorithms. The derivative values needed are calculated using one-sided differencing. Appendix A contains a computer listing of the key subroutine written in Fortran.

Verification of the Transformation Code/Algorithms

In order to verify the accuracy of the coordinate mapping, use will be made of the fact that an analogy exists between the spacial transform solution and a solution of a field equation that has the same form as the mapping equations. To make the comparison, Laplace's equation for steady state conduction was used:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \qquad (2\text{-}51)$$

This equation was solved analytically for the case of two concentric cylinders having fixed temperatures at the inner and outer surfaces.

The solution to the problem is well known, and is given in terms of radial coordinates by

$$T(r) = (T_i - T_o) \frac{\ln \frac{r}{r_o}}{\ln \frac{r_i}{r_o}} + T_o \qquad (2\text{-}52)$$

Using this equation the radial location for a specified temperature may be written

$$r = r_o \left(\frac{r_i}{r_o} \right)^{\frac{T-T_o}{T_i - T_o}} \qquad (2\text{-}53)$$

For an inner and outer radius value of 1 and 9 feet, respectively, and an inner and outer temperature of 400 and 0 degrees Farenheit, respectively, the locations of the isotherms in 40 degree Farenheit increments were calculated, and are listed in Table 2-1. Those radius values for temperature increments of 10 degrees Farenheit are listed in Table 2-2. From a numerical point of view, these two tables represent exact values for a fine and coarse mesh. Listed along side these values are the final results of a spacial transform to map the given circular cylinder. The output for the mapping on a 41 x 41 grid can be found in Appendix B, and the output for the 11 x 11 grid in Appendix C. Note that each coordinate point has an x and y value associated with it. Knowing that the boundary points were assigned with the region centered on the origin of an x-y coordinate system, these values can be easily

converted into an equivalent radial location. This task is much simpler if values are chosen along the x or y axis. The radial locations can then be picked off directly.

Because the generating equations used are Laplacian in form, there is an exact analogy between the resulting x and y distributions and the location of isotherms, where those isotherms are equally spaced (i.e., each grid point outward from the inner boundary represents an equal drop in the number of degrees Farenheit). The percent error between the two solutions presented in Tables 2-1 and 2-2 clearly shows increased accuracy with a finer mesh. The error for an 11 x 11 grid drops from a maximum of 7.13% to a maximum of 0.40% for a 41 x 41 grid. The error, as would be anticipated, increased with increased distance from the boundary where the temperature is known.

TABLE 2-1 ACCURACY OF ITERATED SPACIAL VALUES FOR TWO CONCENTRIC CIRCLES USING A 41 x 41 GRID

Isotherm Value (°F)	Analysis Radius Value	Program Radius Value	% Error
0	9.0000	9.0000	0.0000
10	8.5189	8.5155	0.0399
20	8.0636	8.0573	0.0781
30	7.6326	7.6240	0.1126
40	7.2246	7.2141	0.1453
50	6.8385	6.8263	0.1784
60	6.4730	6.4596	0.2070
70	6.1270	6.1127	0.2333
80	5.7995	5.7845	0.2586
90	5.4895	5.4741	0.2805
100	5.1961	5.1804	0.3021
110	4.9184	4.9026	0.3212
120	4.6555	4.6397	0.3393
130	4.4067	4.3911	0.3540
140	4.1711	4.1558	0.3668
150	3.9482	3.9333	0.3773
160	3.7371	3.7227	0.3853
170	3.5374	3.5235	0.3929
180	3.3483	3.3350	0.3972
190	3.1694	3.1566	0.4038
200	3.0000	2.9878	0.4066
210	2.8396	2.8282	0.4014
220	2.6878	2.6771	0.3980
230	2.5442	2.5342	0.3930
240	2.4082	2.3989	0.3861
250	2.2795	2.2709	0.3772
260	2.1576	2.1497	0.3661
270	2.0423	2.0351	0.3525
280	1.9331	1.9266	0.3362
290	1.8298	1.8240	0.3169
300	1.7320	1.7268	0.3002
310	1.6394	1.6348	0.2805
320	1.5518	1.5478	0.2577
330	1.4689	1.4655	0.2314
340	1.3903	1.3875	0.2013
350	1.3160	1.3137	0.1747
360	1.2457	1.2439	0.1444
370	1.1791	1.1778	0.1102
380	1.1161	1.1153	0.0716
390	1.0564	1.0560	0.0378
400	1.0000	1.0000	0.0000

TABLE 2-2 ACCURACY OF ITERATED SPACIAL VALUES FOR TWO CONCENTRIC CIRCLES USING AN 11 x 11 GRID

Isotherm Value (°F)	Analysis Radius Value	Program Radius Value	% Error
0	9.0000	9.0000	0.0000
40	7.2246	7.0175	2.8665
80	5.7995	5.5137	4.9280
120	4.6555	4.3631	6.2807
160	3.7371	3.4757	6.9947
200	3.0000	2.7860	7.1333
240	2.4082	2.2463	6.7228
280	1.9331	1.8211	5.7938
320	1.5518	1.4841	4.3626
360	1.2457	1.2154	2.4323
400	1.0000	1.0000	0.0000

Since the spacial transform essentially maps an irregular region into a collection of quadrilaterals, the distribution of transformation coefficients can be used as a check on the accuracy of modelling the entire geometry. Table 2-3 shows this comparison, where the overall values of thickness, circumference and area of the annulus are calculated by summing the appropriate coefficient values. The largest error for both grid sizes is in the approximation of the circumference. This is to be expected, since the curvature is being approximated by a series of straight lines. Again, the finer mesh provides a much more accurate modelling of the geometry. The largest error is reduced from 6.46% to 0.419%. The error in radius for a 41 x 41 grid is slightly larger than for an 11 x 11 grid. This is attributable to round-off error, as only single precision was used in the numerical code.

This comparison and verification demonstrates the accuracy of the spacial transformation subroutine. However, it also points out the possibility of large errors entering the solution of a field equation if the grid that is used is not sufficiently fine to accurately model the region.

TABLE 2-3 ACCURACY OF SPACIAL TRANSFORM BASED ON PHYSICAL SIGNIFICANCE OF TRANSFORMATION COEFFICIENTS COMPARED TO ACTUAL VALUES OF THE REGION IN THE PHYSICAL PLANE

Physical Significance	Parameter		11 x 11 GRID		41 x 41 Grid	
Radius (R_o-R_i)	α	$\Sigma \sqrt{\alpha}$	7.9999	0.001% Error	7.9880	0.025% Error
		Actual Value	8.0000		8.0000	
Circumference ($2\pi R_o$)	γ	$\Sigma \sqrt{\gamma}$	52.9006	6.451% Error	56.3162	0.419% Error
		Actual Value	56.5486		56.5486	
Area ($\pi[R_o^2-R_i^2]$)	J	Σ J	235.1146	6.450% Error	250.2978	0.409% Error
		Actual Value	251.3274		251.3274	

CHAPTER 3 TRANSFORMATION OF CONDUCTION PROBLEM

Transformation of Governing Conduction Equations

Any field equation solved in the transformed plane containing terms having a spacial dependency must have those terms transformed in the same fashion as the spacial coordinates ξ, η were developed. The form of the conduction energy equation used in this study is:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) + q \qquad (3\text{-}1)$$

In the problems considered the thermal properties of the materials are assumed to be constant; hence, Eq. (3-1) can be rewritten as

$$\rho c_p \frac{\partial T}{\partial t} = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) + q \qquad (3\text{-}2)$$

To transform the above equation, the transformation operator developed in the previous chapter, Eq. (2-20), is required, i.e.,

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = \frac{1}{J^2}\left(\alpha \frac{\partial^2}{\partial \xi^2} - 2\beta \frac{\partial^2}{\partial \xi \partial \eta} + \gamma \frac{\partial^2}{\partial \eta^2}\right) \qquad (3\text{-}3)$$

where

$$\alpha = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2 \qquad (3\text{-}4)$$

$$\beta = \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta} \tag{3-5}$$

$$\gamma = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2 \tag{3-6}$$

and

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta} \tag{3-7}$$

Applying Eq. (3-3) to Eq. (3-2) yields:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{k}{J^2}\left(\alpha \frac{\partial^2 T}{\partial \xi^2} - 2\beta \frac{\partial^2 T}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 T}{\partial \eta^2}\right) + q \tag{3-8}$$

This is the transformed energy equation that must be solved in the transformed plane.

Transformation of Boundary Condition Equations

There are two basic methods that can be used for treating an interfacial boundary condition. One is to equate normal heat fluxes and the other is to perform an energy balance on a control volume that includes the boundary, i.e.,

$$\int_v k(\vec{\nabla T}\cdot\vec{n})ds + \int_v \dot{q}\, dv = \int_v \frac{\partial}{\partial t}(\rho c_p T)\, dv \tag{3-9}$$

For a square of rectangular geometry, application of Eq. (3-9) is relatively straightforward. An approximation of Eq. (3-9) can be derived without any special treatment of the interface, and the resulting discretized equation can be cast into simple algebraic form.

To apply the above energy balance to a square cell or control volume in the transformed plane is not as straightforward, however. Equation (3-9) must undergo the same spacial transformation as described above. To apply the resulting transformed equation to a control volume in the computational plane is possible, but the added complexity makes the task quite prohibitive.

Another way of "visualizing" the task is to apply Eq. (3-9) directly to a control volume in the physical plane that has been "unmapped" from the computational plane. Figure 3-1 shows how the resulting cell might appear in the real plane. As can be seen, each cell around the boundary has a different configuration. Computationally, customizing Eq. (3-9) to each cell around the boundary is not easily accomplished, even with a knowledge of the approximate cell geometry and configuration from the transformation parameters at each point. Accuracy needs to be preserved to as great an extent as possible, while maintaining a reasonable degree of computational simplicity. Thus, Eqs. (2-42) and (2-43), which can be used to equate normal heat fluxes through a given boundary point along a line of constant ξ or η, shall be used for the boundary conditions between two adjacent regions.

The boundary conditions for two regions in perfect thermal contact require that there be equality of temperature and normal heat flux:

$$T_1\Big|_I = T_2\Big|_I \tag{3-10}$$

$$k_1 \left.\frac{\partial T}{\partial \eta}_1\right|_I = k_2 \left.\frac{\partial T}{\partial \eta}_2\right|_I \tag{3-11}$$

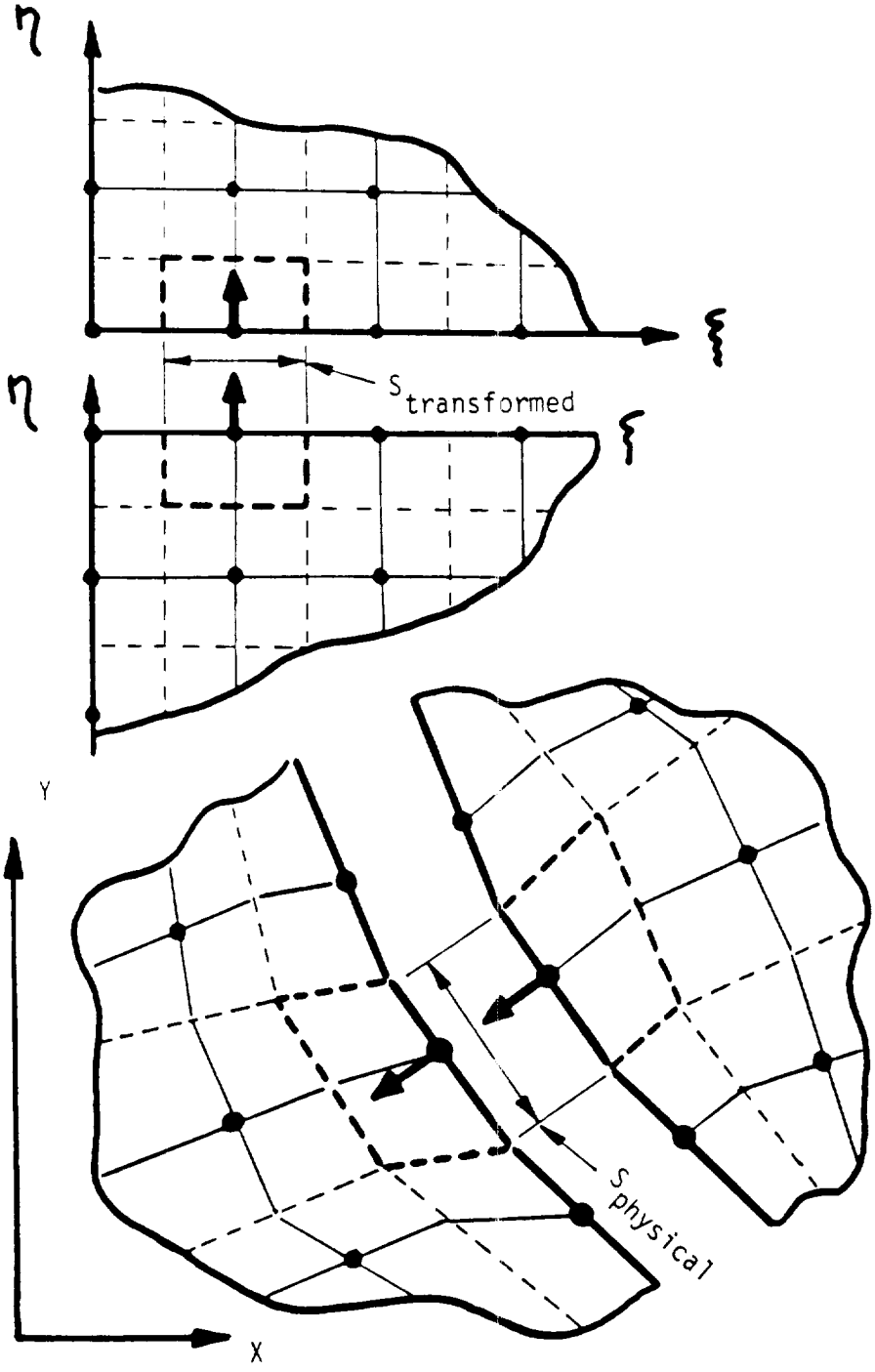


Figure 3-1: A Boundary Contrcl Volume as it Might Appear in both the Physical and Transformed Planes.

On the other hand for an exposed surface, energy conducted to the surface is convected away; therefore,

$$-k \left.\frac{\partial T}{\partial n}\right|_s = h\,(T_s - T_\infty) \tag{3-12}$$

Since the boundary equations contain normal derivatives that are functions of space, a transformation operator is required that will provide an expression for the temperature derivative normal to the surface at a specified point in the real plane. From Chapter 2, these are

$$\left.\frac{\partial}{\partial n}\right|_{\substack{\text{To a line of}\\ \text{constant } \eta}} = \frac{1}{J\sqrt{\gamma}}\left(\gamma \frac{\partial T}{\partial \eta} - \beta \frac{\partial T}{\partial \xi}\right) \tag{3-13}$$

$$\left.\frac{\partial}{\partial n}\right|_{\substack{\text{To a line of}\\ \text{constant } \xi}} = \frac{1}{J\sqrt{\alpha}}\left(\alpha \frac{\partial T}{\partial \xi} - \beta \frac{\partial T}{\partial \eta}\right) \tag{3-14}$$

On the presumption that the boundary is along the upper or lower edge of the transformed plane (a line of constant η), Eqs. (3-11) and (3-12) become, respectively,

$$k_1\left(\frac{\gamma}{J\sqrt{\gamma}}\left.\frac{\partial T}{\partial n}\right|_I - \frac{\beta}{J\sqrt{\gamma}}\left.\frac{\partial T}{\partial \xi}\right|_I\right)_1 = k_2\left(\frac{\gamma}{J\sqrt{\gamma}}\left.\frac{\partial T}{\partial \eta}\right|_I - \frac{\beta}{J\sqrt{\gamma}}\left.\frac{\partial T}{\partial \xi}\right|_I\right)_2 \tag{3-15}$$

$$-k\left(\frac{\gamma}{J\sqrt{\gamma}}\left.\frac{\partial T}{\partial n}\right|_s - \frac{\beta}{J\sqrt{\gamma}}\left.\frac{\partial T}{\partial \xi}\right|_s\right) = h(T_s - T_\infty) \tag{3-16}$$

Equations (3-15) and (3-16) are the transformed layer surface boundary equations that must be solved in the transformed plane.

Numerical Differencing of Conducting Equations

Figure 3-2 shows a section of the grid in the transformed plane. In order to accurately difference the cross-derivative terms in the diffusion equation, all eight nodes surrounding the node being iterated for must be involved in the differencing. Applying second order central differences to all derivatives in Eq. (3-9) yields

$$
\begin{aligned}
J_{i,j}^{2}\,\frac{\rho C_p}{k\Delta t}\left(T_{i,j}^{k+1}-T_{i,j}^{k}\right) = {} & (1-M)\left[\alpha_{i,j}\left(T_{i+i,j}^{k}-2T_{i,j}^{k}+T_{i-1,j}^{k}\right)\right. \\
& -\frac{\beta_{i,j}}{2}\left(T_{i+1,j+1}^{k}-T_{i+1,j-1}^{k}+T_{i-1,j-1}^{k}-T_{i-1,j+1}^{k}\right) \\
& \left.+\gamma_{i,j}\left(T_{i,j+1}^{k}-2T_{i,j}^{k}+T_{i,j-1}^{k}\right)\right] \\
& +M\left[\alpha_{i,j}\left(T_{i+1,j}^{k+1}-2T_{i,j}^{k+1}+T_{i-1,j}^{k+1}\right)\right. \\
& -\frac{\beta_{i,j}}{2}\left(T_{i+1,j+1}^{k+1}-T_{i+1,j-1}^{k+1}+T_{i-1,j-1}^{k+1}-T_{i-1,j+1}^{k+1}\right) \\
& \left.+\gamma_{i,j}\left(T_{i,j+1}^{k+1}-2T_{i,j}^{k+1}+T_{i,j-1}^{k+1}\right)\right]+\frac{J_{i,j}^{2}\,\dot{q}}{k} \qquad (3\text{-}17)
\end{aligned}
$$

It should be noted that a weighting factor M has been used in order that the algorithm may become anything from purely implicit (M equal to zero) to purely explicit (M equal to one). When M is equal to 0.5, the algorithm becomes the Crank-Nicholson scheme. Appendix D contains a

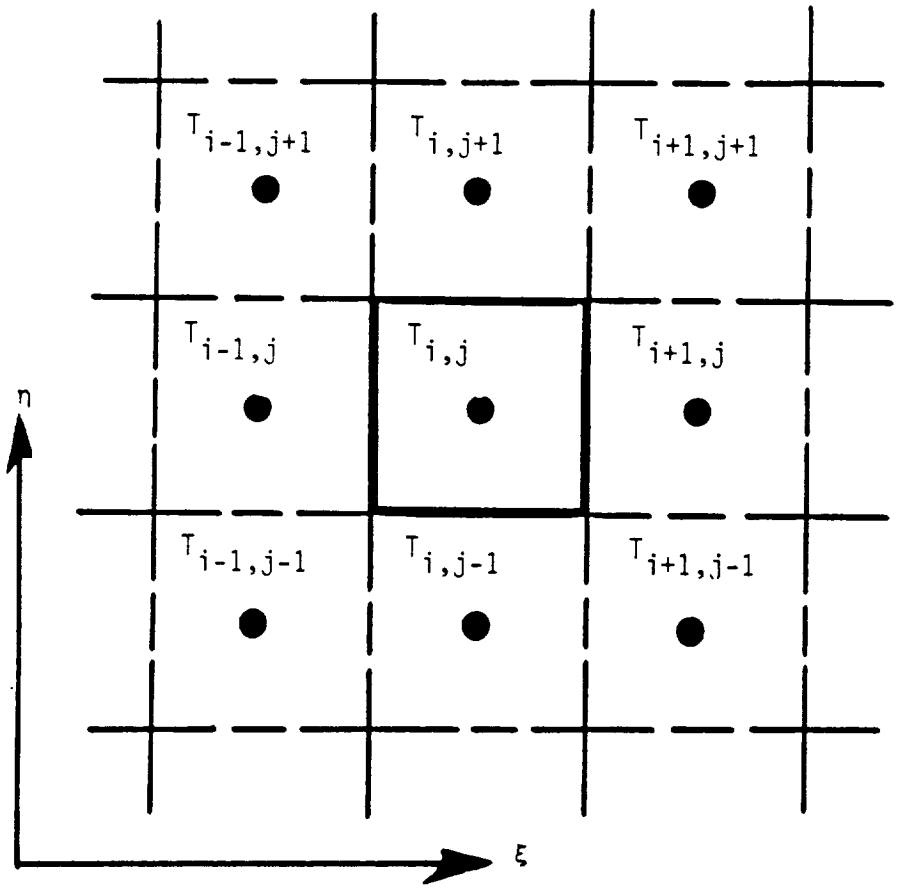


Figure 3-2: Grid Index Notation for Solution of Diffusion Algorithms in Transformed Plane

listing of the subroutine that solves for the conduction problem utilizing Eq. (3-17).

Upon differencing, the boundary condition for equality of normal heat flux at a solid boundary becomes:

$$k_{i,s}\left(\frac{\gamma_{i,s}}{J_{i,1}\sqrt{\gamma_{i,1}}}(T_{i,2}-T_{i,1})-\frac{\beta_{i,1}}{J_{i,1}\sqrt{\gamma_{i,1}}}\frac{T_{i+1,1}-T_{i-1,1}}{2}\right)_{upper}$$

$$=k_{i,jmax}\left(\frac{\gamma_{i,jmax}}{J_{i,jmax}\sqrt{\gamma_{i,jmax}}}(T_{i,jmax}-T_{i,jmax-1})\right.$$

$$\left.-\frac{\beta_{i,jmax}}{J_{i,jmax}\sqrt{\gamma_{i,jmax}}}\frac{T_{i+1,jmax}-T_{i-1,jmax}}{2}\right)_{lower} \qquad (3\text{-}18)$$

The algorithm for a convective boundary condition is similar to Eq. (3-18), except that the left or right hand side is replaced by $h\,(T_{i,1}-T_{\infty})$ or $h\,(T_{i,jmax}-T_{\infty})$, depending on whether the convective boundary is on the upper or lower surface of the grid. Appendix E contains a listing of the subroutine that solves for the normal flux boundary condition using Eq. (3-18).

Verification of Numerical Modelling

A. One Zone, Steady State

The problem used to verify the algorithm for conduction in a single layer under steady state conditions is conduction in offset cylinders. The exact solution to this problem is found in Eckert and Drake [21]. Computed isotherms for a cylinder with an outer radius of 0.3048 meters

(1.0 feet) held at 37.78 degrees C (100 degrees F), an inner radius of 0.0457 meters (0.15 feet) held at 537.78 degrees C (1000 degrees F), and an eccentricity of 22.86 millimeters (0.75) feet are shown in Fig. 3-3 (straight line). For a grid consisting of 17 x 17 nodes, the numerical values are in excellent agreement with the analytical values. The largest error to be found between any two values was less than 0.1%. Thirty-seven seconds of computing time was needed to obtain a converged result on an IBM 4341.

The eccentric cylinder problem was also solved analytically for the case involving uniform internal heat generation by El-Saden [22]. This problem was numerically solved for the case where energy was being uniformly generated at the rate of 20.7 kilojoules per second - cubic meter (2000 BTU per hour - cubic foot). Figure 3-3 compares the exact to predicted temperatures on the lines $\xi = 1$, $\xi = 6$, and $\xi = 8$. These coordinate lines represent cuts in the transformed plane at 0, $\pi/2$, and π respectively. Again there is excellent agreement, with the maximum error being less than 0.5 %.

B. Multiple Zones, Steady State

The steady state temperature distribution in four concentric cylinders was obtained and compared to the elementary analytical solution for this problem. Table 3-1 lists the parameters used for this problem.

Table 3-2 presents the predicted temperatures as compared to the analytical values at three interfaces for various convergence criteria. The surface temperature of the inner cylinder was held at 1000 degrees

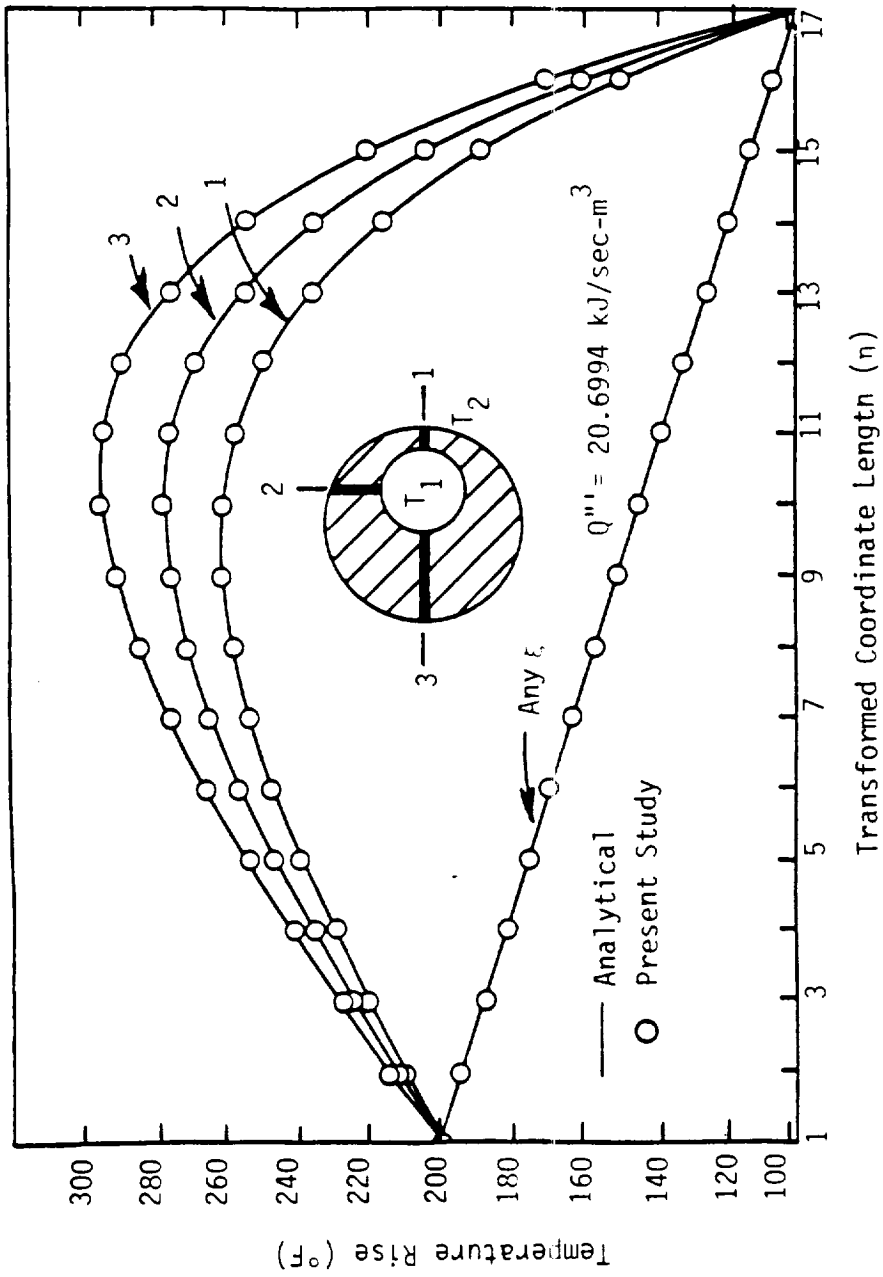


Fig. 3-3: Code Verification Results - Offset Cylinders With and Without Heat Generation

Layer	Inner Radius R_i(mm)	Outer Radius R_o(mm)	Thermal Conductivity k(kJ/hr-m-°C)
1	1000	800	155.77
2	800	700	249.23
3	700	500	18.69
4	500	300	93.46

Table 3-1: Parameters Used in the Multiple Zone Steady State Verification Problem

Table 3-2: Code Verification Results - Interface Temperature Solutions for Concentric Cylinder Problem

Analytical (°C)	Num. 1 (°C)	Num. 2 (°C)	Num. 3 (°C)	Num. 4 (°C)	Num. 5 (°C)
100.00	100.00	100.00	100.00	100.00	100.00
150.69	173.25	158.3	155.6	154.0	153.3
169.65	196.0	178.5	175.4	173.5	172.74
806.60	815.1	809.72	808.9	808.4	808.15
1000.00	1000.00	1000.00	1000.00	1000.00	1000.00
Spatial Convergence	0.0001	0.0001	0.0001	0.0001	0.0001
Temp. Field Convergence	0.1	0.1	0.1	0.1	0.1
Boundary Convergence	0.1	0.1	0.05	0.02	0.01
CPU Time IBM 4341	1:29	3:31	3:52	3:58	4:00

C, while the surface temperature of the outer cylinder was held at 100 degrees C.

The error for the innermost boundary is 2.61 degrees C, for the middle boundary 3.09 degrees C, and for the outermost boundary 1.55 degrees C. The largest error for numerical solution 5 is less than 1.8 percent. The above solutions were obtained on an 11 x 11 grid for each layer. Figure 3-4 displays a graphical representation of the numerical solution. In light of the relatively coarse grid used, the agreement is excellent.

C. One Zone, Time Dependent

Jakob [23] developed exact solutions for three one-dimensional time dependent heat conduction problems: for the temperature transients in an infinitely long square bar; for the temperature transients in an infinite medium with a circular hole; and for the transient temperature response of a hollow cylinder within an infinitely thick wall. All three of these exact solutions were used as a means of establishing the accuracy for the numerical solution of the time dependent conduction equation.

The exact solution developed by Jakob for the infinitely long square bar was for the case where the outer surface was suddenly subjected to a step change in temperature. Table 3-3 compares the numerical results, in dimensionless form, with those presented by Jakob. For the numerical solution, a diffusivity of 0.929 square meters per hour (10 square feet per hour) was used with a 0.2286 meter square bar. The temperature field was initially at a uniform value of -12.222

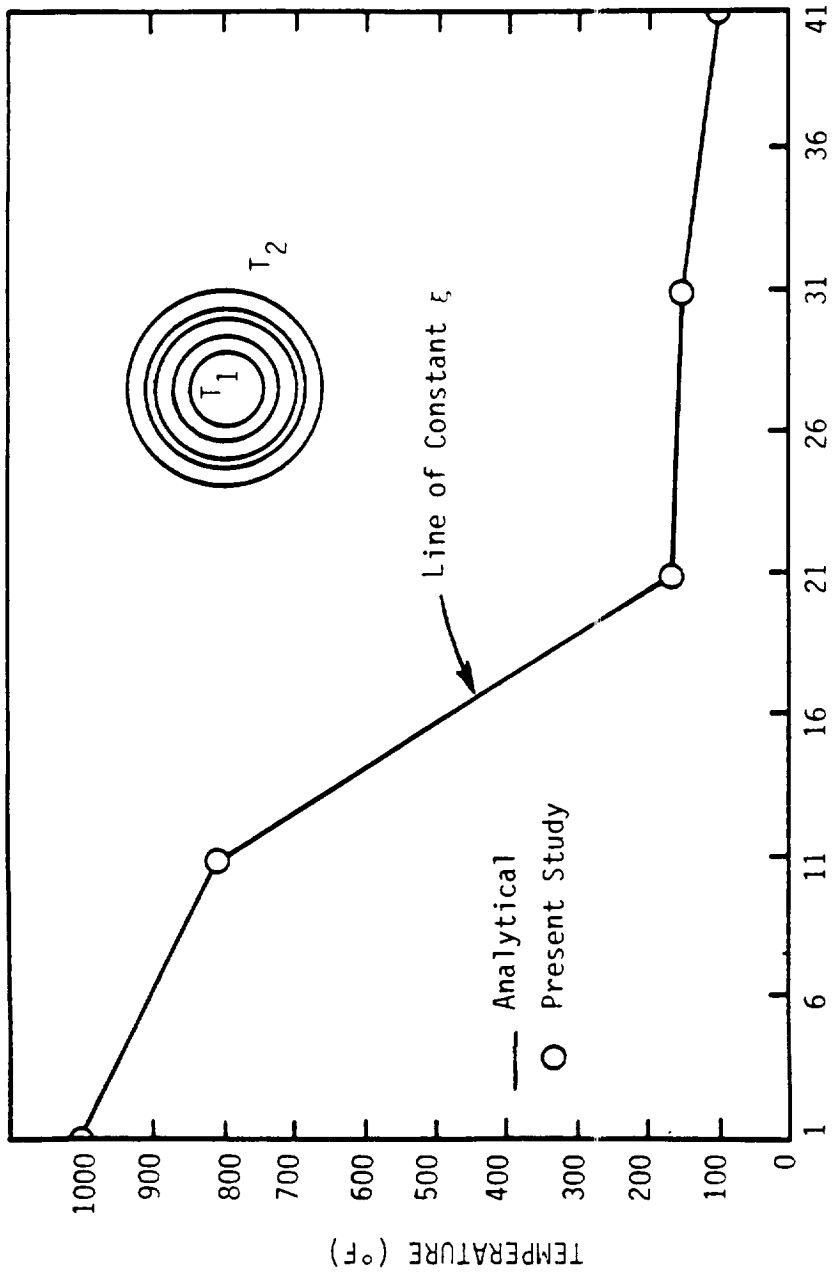


Fig. 3-4: Code Verification Results - Multiple Zones Steady State

Table 3-3: Code Verification Results - Center Temperature Response of an Infinitely Long Square Bar

$\frac{\alpha t}{s^2}$	$\frac{\theta_o}{\theta_i}$ Jakob (23)	$\frac{\theta_o}{\theta_i}$ Numerical	% error
0.032	1.000	0.998	0.126
0.080	0.951	0.935	1.600
0.100	0.901	0.893	0.849
0.160	0.715	0.711	0.530

degrees C (10 degrees F). Three minutes and fifty seconds of CPU time on a NAS 6650 was required to produce the results on a 17 x 17 grid, with convergence on space of 0.00001 and on temperature of 0.001. Convergence was defined as the absolute value of a simple difference of the old and new iterated values.

The exact solution for the case where there was a step change in temperature at the surface of a circular hole in an infinite medium was also solved by Jakob. Obviously, it is not possible for the present computer code to simulate an infinite medium. To approximate this case, concentric cylinders were used with an outer radius of 3.048 meters (10.0 feet) and an inner radius of 0.3048 meters (1.0 foot). The numerical solution will be valid up to the point where the initial thermal wave reaches the outer boundary. For the numerical problem, a diffusivity of 0.9290 square meters per hour (10.0 square feet per hour) was used. The field was initially set at 37.78 degrees C (100 degrees F), and the surface of the hole was suddenly raised and maintained at 537.78 degrees C (1000 degrees F). The results obtained are displayed in Table 3-4. One minute and 57 seconds of CPU time on a NAS 6650 was required using an 11 x 11 grid with tolerance on space of 0.001 and on temperature of 0.01. Convergence was defined as the absolute value of a simple difference of the old and new iterated values. Note that even with the relatively coarse mesh and loose tolerances (as compared to the previous problem) for this single region problem, there is excellent agreement with the analytical results.

The exact solution for the transient temperature response for a hollow cylinder within an infinitely thick wall was solved for the case

Table 3-4: Code Verification Results - Heat Flux at Surface of a Circular Hole in an Infinite Medium

$\frac{\alpha t}{s^2}$	$\frac{q_s}{kL\theta_s}$ Jakob (23)	$\frac{q_s}{kL\theta_s}$ Numerical	% error
0.01	38.51	38.79	-0.72
0.1	14.13	14.53	-2.83
0.6	7.29	7.40	-1.53
1.0	6.18	6.28	-1.61
2.0	5.03	5.13	-2.00
3.0	4.49	4.61	-2.00

with the cylinder having a 1.0 foot radius. To approximate the infinite plate, two concentric cylinders were used with the outer radius set at 10.0 feet. The numerical solution should be unaffected by this false outer boundary up to the point where the initial thermal wave reaches the outer boundary. The thermal diffusivity used was 10.0 square feet per hour. The temperature of the plate was initially set at 100 degrees Farenheit, with the temperature of the inner radius wall suddenly raised to 1000.0 degrees Farenheit at time equal to zero. Figure 3-5 compares the predicted temperature just inside the hole to the exact value presented in Jakob. The computed values are for an 11 x 11 grid with a time step of 0.001 hours. Approximately two minutes of computing time were needed to advance the problem 1000 time steps on an IBM 4341. Considering the coarsness of the mesh used, the results are outstanding.

D. Two Zones, Time Dependent

Jaeger [24] developed an analytical solution for the time dependent temperature distribution in a two-layered circular cylinder. The two concentric layers were in perfect contact and had different thermal properties. Initially the temperature throughout both layers is uniform. At some instance in time, the inner surface of the inner layer is subjected to a step change in temperature. The temperature of the external surface of the outer layer is held at the initial temperature. Jaeger did not consider internal heat generation in the solution.

The closed form solution presented by Jaeger is somewhat tedious to apply. It involves a series solution that necessitates the use of

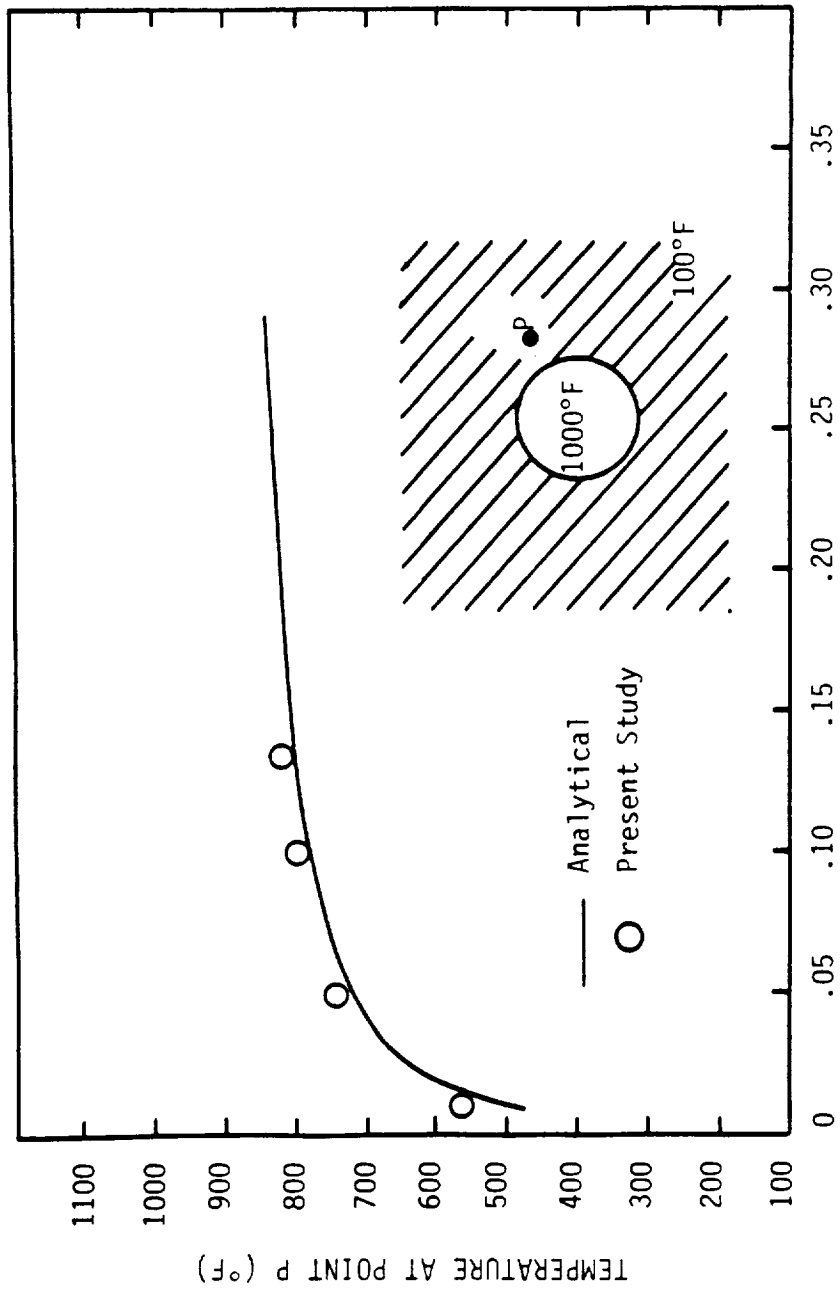


Fig. 3-5: Code Verification Results - Time Dependent Solution for One Zone

eigenvalues. The eigenvalues needed are the roots of a somewhat complex algebraic statement containing Bessel functions. Berger [25] developed a computer program that numerically implements the closed form solution developed by Jaeger. Berger's program was run for a case solved by the transformed time dependent conduction equation. This comparison is thus a check for the time dependent algorithm and boundary condition algorithm for two thermally dissimilar layers.

For the problem solved, the inner circular layer had a conductivity of 0.2014×10^{-3} BTU per inch-second-degree F, a diffusivity of 0.4 square inches per second, an inner radius of 2.0 inches and an outer radius of 5.0 inches. The outer layer had a conductivity of 0.3278×10^{-4} BTU per inch-second-degree F, a diffusivity of 0.2 square inches per second, an inner radius of 5.0 inches and an outer radius of 9.0 inches. The initial temperature was 100 degrees F, with the internal surface of the inner layer being suddenly raised and maintained at 1000 degrees. These particular properties were chosen to duplicate a problem solved by Berger.

Table 3-5 shows a comparison of the temperature rise with time at the interface of the two circular layers. Two comparisons are shown, one with both layers being represented by an 11 x 11 grid, and one with both layers being represented by a 31 x 31 grid. As can be seen, when both layers are represented by the finer grid, the agreement is excellent, with a maximum error of 3.08 percent occurring 3.5 seconds into the problem. The error decreases with additional time into the problem because the rapid temperature increases begin to slow down, as the problem advances toward the steady state solution. It should be

Table 3-5: Comparison Between Exact and Numerical Solutions of the Interface Temperature of a Circular Cylinder with Two Layers

Time (sec)	Interface Temperature (°F)				
	Exact	11 x 11 Grid in each layer	% Error	31 x 31 Grid in each layer	% Error
0.0	100.00	100.00	0.00	100.00	0.00
0.5	100.00	100.11	0.11	100.01	0.01
1.0	100.75	103.06	2.29	101.05	0.30
1.5	105.91	113.22	6.90	107.18	1.20
2.0	117.11	130.11	11.10	119.63	2.15
2.5	133.03	151.27	13.71	136.70	2.76
3.0	151.87	174.57	14.95	156.47	3.03
3.5	172.24	198.62	15.32	177.54	3.08
4.0	193.22	222.62	15.22	199.03	3.01
4.5	214.25	246.11	14.87	220.39	2.87
5.0	234.96	268.83	14.42	241.34	2.72
6.0	274.75	311.61	13.42	281.33	2.39
7.0	311.88	350.72	12.45	318.44	2.10
8.0	346.25	386.41	11.60	352.67	1.85

noted that although a coarse 11 x 11 grid was adequate to determine the transient response of a single layered circular cylinder, the grid is too coarse to provide accurate results with the addition of a second layer; a considerably finer grid is required. This problem can become worse as additional layers are added, requiring finer and finer meshes to provide acceptable accuracies. It should also be noted that the interface in this problem is near the center of the two fixed temperature boundaries. Thus the errors at this location represent the maximum errors in the solution fields at a given instant in time.

E. Multiple Zones, Time Dependent

No closed form solutions have been found in the open literature for treating a time-dependent problem of a body with three or more layers which also contains a heat source. However, numerical solutions to such problems exist and these can be used for code verification.

Stallabrass [2], Baliga [4], Marano [5], and Chao [7], using different numerical approaches, have each solved a one-dimensional time-dependent problem of a six layered slab with one of the layers generating heat. Table 3-6 lists the dimensions, materials and properties of the six layers used to verify the present code. Figure 3-6 presents the results of computations for a time step of 0.001 seconds with a grid having 14 nodes in the 75S-T6 aluminum, 4 nodes in the heater, 7 nodes in the upper insulation, 9 nodes in the stainless steel and 33 nodes in the ice. It can be seen that the predictions are in good agreement with results reported earlier.

It should be noted that to model this series of stacked slabs, the present code used a set of concentric circular layers with the inner

Table 3-6: Materials and Properties Used in The Multiple Zone, Time Dependent, Verification Problem

Layer	Material	Thickness (mm)	Thermal Diffusivity m^2/hr
1	75S-T6 Aluminum	2.210	0.15329
2	Epoxy/glass Insulation	1.270	0.00081
3	Nichrome heater	0.012	0.01282
4	Epoxy/glass Insulation	0.254	0.00081
5	304 Stainless Steel	0.305	0.01394
6	Ice	6.350	0.00413

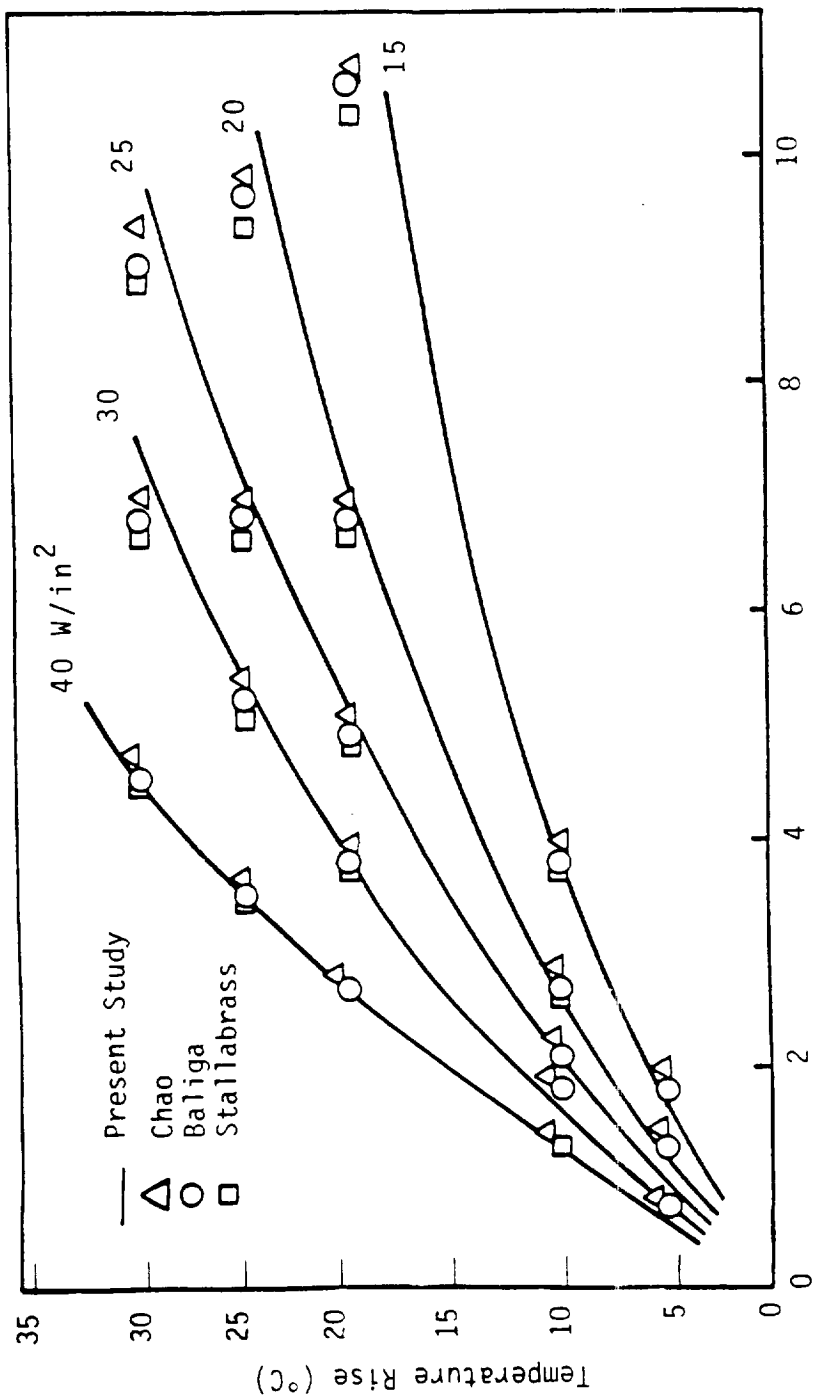


Fig. 3-6: This Study's Predicted Numerical Results for a Six Layered Slab with Internal Heat Generation Compared to Other Numerical Solutions

layer having a large radius. Some of the small discrepancies in the computed results are believed to be due to these geometrical effects. The computational time needed to obtain convergent results on a NAS 6650 varied from two to ten minutes depending on the particular problem considered. To obtain the results shown, minimum tolerances of 0.00001 on space and 0.0001 on temperature were required. Convergence was defined as the absolute value of a simple difference of the old and new iterated values.

CHAPTER 4 TRANSFORMATION OF PHASE CHANGE PROBLEM

Overview of Phase Change Treatments

Solutions to phase change problems have long been a topic of study in the technical community. The characteristic feature of any phase change problem is the coupling of two temperature fields with a moving boundary that not only separates the two fields, but propagates through them. The propagating phase front makes the problem non-linear. Latent heat effects and changes in thermal conductivity between phases increase the non-linearity. Because of the non-linearity problem, only a few exact solutions have been developed, and these are for very restrictive conditions.

Neumann's exact solution to this problem, presented in Carslaw and Jaeger [26], was the first known successful solution of a phase change problem. Neumann solved the problem for one dimensional phase change in a semi-infinite region. The material was initially at or above the fusion temperature, and suddenly experiences a step decrease in temperature at the boundary.

Lin [27] has modified Neumann's development to obtain an exact solution for a quasi one-dimensional problem in a region with a varying cross-sectional area. To incorporate varying areas into the analysis, Lin simplified Neumann's approach by assuming that the entire region was at the fusion temperature, and by developing the relevant equations to solve for the interface velocity as a function of time. These

simplifications readily permitted the insertion of a transformed position coordinate functio into the governing equations.

Cho and Sunderland [28] have extended Neumann's exact solution to allow for variable thermal conductivity in both phases. It was assumed that the conductivity varied linearly with temperature. The effect of the phase change speed with conductivity variation was investigated in some detail.

Solution of the classical phase change problem, as described above, involves the calculation of the phase front for specified boundary conditions. This problem may also be solved by prescribing the location of the phase front as a function of time, and then determining the boundary conditions needed as a function of time to produce the specified phase front movement. This is known as the inverse problem. Rubinsky and Shitzer [29] have developed the exact solution to the one-dimensional inverse problem for any arbitrary function describing the movement of the phase front. Previous solutions to the inverse problem were for a specific function.

Gutman [30] has constructed an approximate analytical one-dimensional solution that attempts to account for the effect of superheating or supercooling on the movement of the phase change front. The resulting analytical form was found by matching the inner and outer solutions to the problem. His method provides accurate results for a very limited range of relevant dimensionless parameters.

To date, there have been no exact solutions developed for phase change problems where the phase change front is a function of two space coordinates.

Effort has been devoted recently towards the obtaining of analytical solutions by approximate methods. Ku and Chan [31] have developed an artificial initial condition that accounts for the temperature profile discontinuity at the phase change front. The artificial condition permits solutions to the resulting phase change equations by using inverse Laplace transforms. Depending upon which side of the phase front line the artificial initial condition is applied, a temperature distribution solution can only be obtained for one of the two phases. The technique has not been used to solve a two-dimensional problem as yet, but the analytical results applied to one-dimensional problems compare well with existing solutions.

Zhang, Weinbaum and Jiji [32] have developed an approximate analytical solution to a three-dimensional time dependant phase change problem. The solutions are limited to very small Stefan numbers. Results are presented for a buried pipe example. They combine a quasi-steady approximation with a virtual free surface method to obtain an axisymmetric solution for the region around the pipe wall. Singularities in the differential equations along the pipe would not permit solutions close to the pipe wall.

The analytical and approximate analytical solutions described above result from the solution of partial differential equations. There is another class of approximate techniques that are based on an integral formulation of a heat balance across the melt line. One of the

earliest published works using this approach was by Goodman [33]. Goodman developed a number of approximate solutions for the one-dimensional problem for typical boundary conditions. Results were found to compare well with existing solutions when the assumed temperature profile was a quadratic function.

More recently, Wang and Perry [34] have applied this technique to a one-dimensional problem with initial superheat. This problem is somewhat more involved, as there are two interfaces, one for the phase change front and one for the superheat line. Excellent agreement was obtained between the approximate solutions and a one-dimensional finite element code constructed to duplicate the problem studied.

Virtually all of the available analytical models, whether exact or approximate, are sufficiently complex so that in most cases an alternate procedure must be adopted. This, coupled with the fact that most of the analytical models have demonstrated accuracy only for one-dimensional problems, strongly indicates that any feasible two-dimensional solution must be sought by alternative means.

Because of the nature of the phase change problem, the obvious approach to take is a numerical implementation. By this method, generally a fixed, uniform grid is laid over the problem domain. Three equations are written. Two of these are conduction equations, one for each phase. The third equation is an energy balance, incorporating any latent heat effect, written across the phase change front. This equation ties together the two conduction equations. The advantage to this formulation is that the conduction equations have smooth, continuous derivatives of all primative variables in their respective

domains. This obviously enhances the likelihood of a stable, convergent numerical scheme. The primary disadvantage is in the interpolation required at the phase front. As the problem advances, for each increment of time, the phase front must be relocated. In most instances the front will be between node points. The diffusion algorithms near the phase front then need to incorporate derivatives using the appropriate "shortened" lengths. Because of the numerous interpolation calculations required, the computer codes are somewhat slow. Most of the current work using this approach is in the direction of developing faster diffusion algorithms, or making assumptions about the thermophysics of the problem in order to reduce the number of interpolative calculations. Lazaridis [35], for example, obtained two-dimensional solutions by treating the motion of the fusion front as quasi-one-dimensional. Thus, the temperature gradients need be taken in one direction only. This substantially reduces the number of interpolations needed. What is more, the results show satisfactory agreement with existing solutions.

More recently, finite element methods have been used to solve the moving phase front problem on a fixed grid. O'Neill [36] developed an algorithm that can be used with standard finite element heat conduction codes, using linear interpolation to locate the phase front within elements. Yu and Rubinsky [37] have treated the two relevant conduction equations and the interfacial energy equation as independent governing equations on a finite element mesh. The results of their computer solutions agree well with other closed form solutions.

But even the finite element codes tend to be somewhat punitive with respect to code execution time, because of the interpolation/iterations

needed to track the moving melt front. Thus, numerical techniques have been developed that have attempted to either circumvent, or completely eliminate, the need for the interpolation caused by the phase front.

Numerical researchers have developed a number of ways of circumventing the need for interpolation. These methods essentially involve adapting the grid to the moving melt front. Prusa and Yao [38] have solved a two-dimensional phase change problem of melting around a horizontal cylinder. The problem is solved numerically in cylindrical coordinates, using a radial spacial transform in the governing equations that essentially "stretches" or "contracts" the grid on either side of the phase front. The diffusion equations are then applied to the "stretched" regions. Duda, et al., [39] present a similar technique, calling it boundary immobilization. They also use a stretching transformation, but in their analysis the location of the phase front in transformed coordinates remains at a fixed location. The effect, however, is essentially the same. The regions on either side of the phase front are "stretched" or "contracted" to generate the prescribed regions in transformed coordinates.

Rieger, Projahn and Beer [18] custom fit a grid by use of a body fitted coordinate transformation. As the phase front moves with each time step, a new grid is numerically generated using the phase front as the upper boundary of one region and the lower boundary of the second region. They also solve the problem of melting around a heated horizontal cylinder.

Lynch [40] uses essentially the same strategy as Reiger, Projahn and Beer [18], but applies a finite element mesh generator instead of a body fitted coordinate transform.

Although grid adaption eliminates the need for interpolation, additional calculations are required to generate the new grid (or function for the "stretching" transformation). The latter tends to offset the gains made by the former. Numerical techniques that eliminate the need to solve for the location of the phase front eliminate the need for not only interpolation, but also calculations for adapting the grid, since a rigid, uniform mesh can be used. In order to eliminate the need for any calculations for the location of the phase front, the equations that govern the system need to be reformulated. The reformulation, depending upon the strategy used, may result in the insertion of an additional approximation into the governing equations.

Kikuchi and Ichikawa [41] introduce a special integral transformation that accounts for the discontinuity of the temperature gradient caused by the phase change front, even though the location of the front is unknown. This special integral transformation is known as the freezing index. The equations used for iterating through the temperature fields are developed by applying the freezing index integral to the original set of governing equations. The algebraic set of equations needed are obtained by discretizing the associated variational form of the freezing index. Blanchard and Fremond [42] use a strategy similar to Kikuchi and Ichikawa, except that they use the homographic approximation to model the temperature discontinuity caused by the phase front. In doing this they solve for a variational equality, instead of the variational inequality solved by Kikuchi and Ichikawa. The primary advantage to the homographic insertion is that the originally

non-differentiable change of temperature across the phase front becomes differentiable.

In the heat conduction equation, temperature is usually considered the dependent variable, with time and space coordinates the independent variables. By an appropriate transformation of the conduction equations, however, one of the space coordinates can become the dependent variable, with temperature, time, and any remaining space coordinates the independent variables. By specifying values of time, temperature and one space coordinate (in a two-dimensional problem), the values of the independent space coordinate can be iterated for. By always specifying the same value of temperature at each time step, the movement, or migration, of an isotherm with time can be tracked. Thus, this technique is called the isotherm migration method. Crank and Gupta [43] were the first to apply this technique to a phase change problem in two dimensions. The advantages of this technique, in tracking a moving isotherm at the fusion temperature, are obvious. Results predicted by this approach have been found to be satisfactory, provided two numerical idiosyncracies caused by the technique are circumvented. First, an approximate method is used to calculate the position of the phase front for a short initial time into the problem. After the initial time interval, with an established isotherm, the isotherm migration algorithm can be used. Secondly, depending upon the geometry and/or the thermophysics of the problem involved, the independent space coordinate can become a multi-valued solution to the algorithm containing the dependent variables, where it is inappropriate.

Saitoh [44] extended the two-dimensional isotherm migration method from regular two-dimensional geometries to arbitrarily shaped, doubly connected two-dimensional geometries. This was accomplished by applying the same radial "stretching" transformation as used by Prusa and Yao [38] to the governing equations before exchanging the dependent and independent variables. The numerical results obtained by this method showed excellent agreement with experimental data.

Ozisik [45] discusses the use of a moving heat source in the conduction equations to account for the latent heat effect. He presents a mathematical development that explicitly casts the moving boundary problem into a standard heat conduction problem with a moving heat source. By doing this, the solution can immediatedly be written in terms of Green's function, and numerically implemented.

Another technique for eliminating the necessity of locating the phase front, is use of the so-called high heat capacity method. In this method, the latent heat effect of the phase change is approximated by a large heat capacity over a small temperature range. This technique has been in existence for a number of years, and has recently undergone further development into more sophisticated algorithms and applications. Bonacina, et al. [46] have developed a three-time level implicit scheme using the approach. The temperature dependent properties are evaluated only at the intermediate time step, thus simplifying the solution to the algorithms. The scheme was found to be unconditionally stable and convergent.

Comini, et al. [47] present a similar analysis to the one above, but for a finite element code using triangular elements, so that

irregular geometries could be accomodated. Morgan, Lewis and Zienkiewicz [48] have improved this code by applying the technique with quadratic isoparametric finite elements.

Hsiao [49] modified the manner in which the specific heat was calculated. He accounts for the latent heat, and determined the physical conditions of the node, by using a linear interpolation of the surrounding nodal temperatures. Using this linear interpolation, excellent results are obtained for both one and two-dimensional phase change problems with relatively large time steps and coarse mesh.

Uchikawa and Takeda [50] have applied the high specific heat method to the irregular geometry of a casting mold. In this analysis a transform is used to turn the irregular regions into regular, evenly spaced computational zones through the use of a body fitted coordinate procedure. The governing equations, with the high specific heat method, are then transformed and solved in the transformed plane.

One of the more recent techniques for eliminating the need to calculate the location of the phase front is the enthalpy method. In this method, the specific heat is combined with temperature to form an enthalpy variable in the time dependent term in the diffusion equation. Thus, both temperature and enthalpy are dependent variables. All of the earlier two-dimensional models using this technique assumed that the curve relating enthalpy to temperature had a finite slope. This is only true, however, for materials that undergo a change of phase over a temperature range. Shamsundar and Sparrow [51] have developed a variation to the earlier two-dimensional models. Depending upon both the temperature and the enthalpy of a given node, either temperature or

enthalpy is the dependent variable. When the condition of a node is such that a single phase exists, temperature is the dependent variable. When the enthalpy of a given node is between the values of either phase (a change of phase is occurring), the temperature is known, making enthalpy the dependent variable. This approach easily permits solutions for substances whose change of phase occurs entirely at a single temperature, as well as for those whose change of phase occurs over a range of temperatures. Shamsundar and Sparrow use an implicit finite difference scheme to numerically implement the governing equations. They apply the technique to PCM, a type of wax, that has relatively little change in thermal conductivities between phases.

Crowley [52] applies an explicit finite difference scheme to an enthalpy formulation that is equivalent to that of Shamsundar and Sparrow. Crowley applies his algorithm to the solution of Saitoh's problem [44], with water as the phase change substance. Even with the large difference in thermal conductivities that water exhibits, the numerical results agreed well with published experimental data for the problem.

One of the drawbacks of the enthalpy formulation, particularly for the case where the change of phase occurs at a single temperature, is that a plot of temperature against time for a given node tends to exhibit a "plateauing" effect or tendency. This arises out of the fact that a computational grid models or represents a discrete region in space. Obviously, it requires a finite amount of time to melt a discrete region. As a consequence of a node being held fixed at a single temperature for a discrete amount of time, the effect is also

felt by the surrounding nodes, causing a plateauing of the temperature. Voller and Cross [53] developed a smoothing technique that can be applied to a final set of numerical results. This smoothing technique has the effect of bringing the time-temperature history of a given node into excellent agreement with other published results.

Schneider and Raw [54] have developed a modified enthalpy model that is capable of efficiently solving problems where multiple phase change interfaces exist. For these types of problems their modification reduces computational times by an order of magnitude. Thus, the implicit scheme used here is only applied to one-dimensional problems, but they indicate that it can be easily extended into two dimensions.

Voller [55] provides an alternate method of discretizing the enthalpy formulation. In this approach, the sensible and latent heat terms are discretized separately, thus isolating the non-linearity of the problem as a nodal latent heat source term. His implicit finite-difference scheme yields computational savings of twenty to fifty percent.

Tacke [56] has developed a formulation of the enthalpy method which removes the "plateauing" effect in the time-temperature history of a node. This was accomplished by applying linearized temperature profiles near the phase front for those nodal control volumes containing the front. His numerical solutions substantially reduced the plateauing, while showing excellent agreement with analytical solutions. The linearization does increase the computational time required, but only slightly.

Recently, Schneider [57] has used the enthalpy form of the energy equation, and, in the melted liquid region, coupled it with the momentum equation to account for the effect of free convection effects on the movement of the phase front line. The problem is solved in a rectangular region, with a rise in temperature occurring at only one vertical wall. Boucheron and Smith [58] have solved essentially the same problem, but they couple the momentum equation throughout the field for both phases. They specify for the solid phase a very high value of viscosity in order to insure that the velocities arising from the solution of the momentum equations in this region are negligible or zero. This approach allows a more mathematically straightforward treatment of the phase front.

Because of the relative ease of formulation, simplicity in numerically discretizing the resulting equations, and proven accuracy and stability, the enthalpy formulation was selected to solve the phase change portion of the current problem. The irregular region is mapped into a rectangle using a body fitted coordinate spacial transform. The governing equations must be similarly transformed, and solved on the transformed grid. Writing a general "numerical" energy balance in the transformed plane is virtually impossible. As was discussed in the previous chapter, the square, regular computational cells in the computational or transformed plane unmap into nonuniform, non-orthogonal cells of varying area in the physical plane. Consequently, the transformed partial differential field equations were differenced rather than developed by conservative principles. Because thermal conductivity is not a constant for this problem, there are two possible forms of the

field equation, the conservative form and the non-conservative form, which are mathematically equivalent. When treated numerically in the same fashion, the result will be two algorithms that are not equivalent. The rationale for selecting the equations type is the topic of the next section.

Importance of Equations Type - Neumann Comparison

Since the energy balance at the phase change front is essentially "buried" in the weak formulation of the phase change equations, the discretized form of these equations can become important for an accurate solution. One important distinction in equation form is conservative versus non-conservative. The term "conservative" means that the "purest" form of an equation is preserved or "conserved"; while non-conservative means that some change has been made to the "pure" form (i.e., differential operators have been carried through, etc.). This difference in form for the enthalpy formulation is shown in detail below.

Figure 4-1 depicts the two zones that need to be considered in the traditional formulation of a phase change problem. The equations that apply, if only thermal conduction with no density change is considered, are:

in liquid region

$$\rho_L c_L \frac{\partial T_L}{\partial t} = \frac{\partial}{\partial x}\left(k_L \frac{\partial T_L}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_L \frac{\partial T_L}{\partial y}\right) \qquad (4\text{-}1)$$

in solid region

$$\rho_S c_S \frac{\partial T_S}{\partial t} = \frac{\partial}{\partial x}\left(k_S \frac{\partial T_S}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_S \frac{\partial T_S}{\partial y}\right) \qquad (4\text{-}2)$$

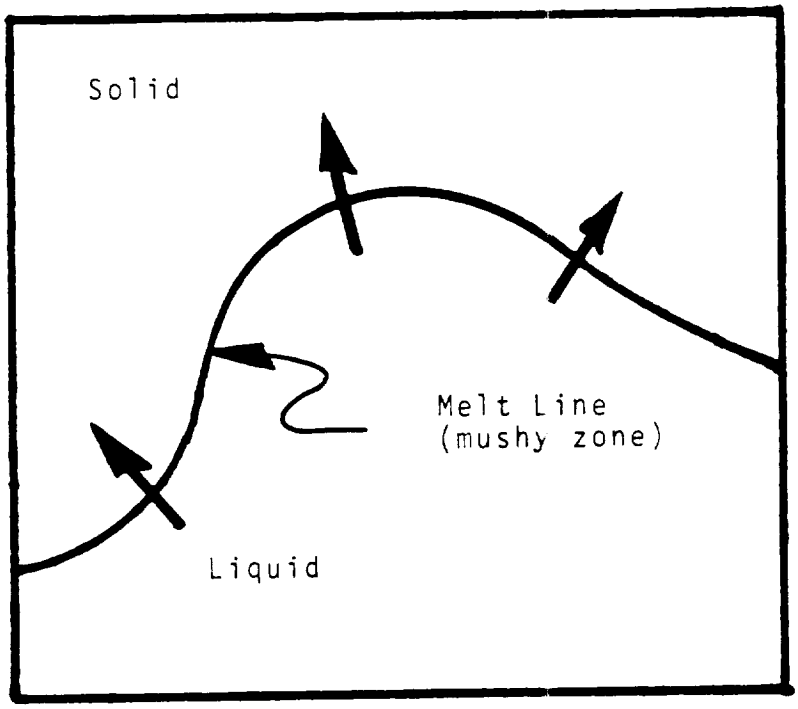


Figure 4-1 The Two Zones needed in the Traditional Formulation of the Phase Change Problem

$$k_S \frac{\partial T_S}{\partial n} - k_L \frac{\partial T_L}{\partial n} = \rho_S v_n \lambda \qquad (4\text{-}3)$$

on $f(x,y,t)=0$

$$T_i(x,y,t) = T_f \qquad i = s \text{ and } L \qquad (4\text{-}4)$$

where n = outward normal to the interface, into the liquid

V_n = velocity of the interface in the normal direction

$f(x,y,t)$ = function describing the interface separating the solid and liquid regions

The above two field equations, coupled by an interfacial energy balance, Eq. (4-3), can be reduced to a single non-linear field equation that eliminates the need for any computation of the location of the phase front. This is known as the weak formulation, and for phase change problems is called the enthalpy formulation. The term "weak formulation" means that less direct information is provided in the solution of the equations (in this case, the location of the phase change line). Thus, the weak formulation for this problem becomes:

$$\rho \frac{\partial H}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) \qquad (4\text{-}5)$$

$$\text{when } H < H_{sm}, \quad T = \frac{H}{\rho C_s} \qquad (4\text{-}6)$$

$$\text{and} \quad k = k_s ;$$

$$\text{when } H_{sm} \leq H \leq H_{Lm}, \quad T = T_m$$

$$\text{and} \quad k = k_s + (k_L - k_s)\frac{H-H_{sm}}{H_{Lm}-H_{sm}} ; \qquad (4\text{-}7)$$

$$\text{and when} \quad H > H_{Lm}\ , \quad T = T_m - \frac{H_{Lm}-H}{\rho C_L} \qquad (4\text{-}8)$$

$$\text{and } k = k_L\ .$$

The above set of equations is much more easily treated computationally than the previous set. The location of the melt line can be determined indirectly from the distribution of enthalpies on the solution domain.

Equation (4-5) is written in the conservative form. If the operators are carried through, Eqs. (4-5) becomes

$$\rho \frac{\partial H}{\partial t} = k \frac{\partial^2 T}{\partial x^2} + k \frac{\partial^2 T}{\partial y^2} + \frac{\partial k}{\partial x}\frac{\partial T}{\partial x} + \frac{\partial k}{\partial y}\frac{\partial T}{\partial y} \qquad (4\text{-}9)$$

and is now cast in a non-conservative form.

Written for the one-dimensional case, Eqs.(4-5) and (4-9) become, respectively

$$\rho \frac{\partial H}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) \qquad (4\text{-}10)$$

$$\rho \frac{\partial H}{\partial t} = k \frac{\partial^2 T}{\partial x^2} + \frac{\partial k}{\partial x}\frac{\partial T}{\partial x} \qquad (4\text{-}11)$$

If a simple explicit differencing procedure is applied per the general one-dimensional grid depicted by Fig. (4-2), Eqs. 4-10 and 4-11 become

$$\rho\left(\frac{H^{k+1} - H^k}{\Delta t}\right) = \frac{1}{(\Delta x)^2}\left(\frac{k_{j+1}^{k+1} + k_j^{k+1}}{2}\left(T_{j+1}^{k+1} - T_j^{k+1}\right) - \frac{k_j^{k+1} + k_{j-1}^{k+1}}{2}\left(T_j^{k+1} - T_{j-1}^{k+1}\right)\right) \qquad (4\text{-}12)$$

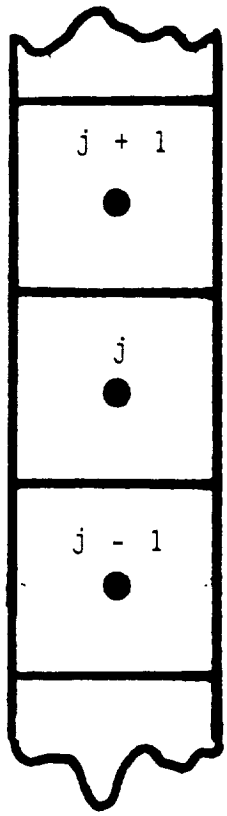


Figure 4-2 General Grid Used for One-Dimensional Explicit Differencing

$$\rho\left(\frac{H^{k+1}-H^{k}}{\Delta t}\right)=\frac{1}{(\Delta x)^{2}}\left(k_{j}^{k+1}\left(T_{j+1}^{k+1}-2T_{j}^{k+1}+T_{j-1}^{k+1}\right)+\frac{\left(k_{j+1}^{k+1}-k_{j-1}^{k+1}\right)\left(T_{j+1}^{k+1}-T_{j-1}^{k+1}\right)}{4}\right) \tag{4-13}$$

where the superscript k denotes the time step and the subscript j the nodal location. The algorithms obtained by using Eqs. (4-12) and (4-13) are clearly not equal to each other since thermal conducting varies as the phase changes. Error is introduced into any mathematical system when that system is numerically discretized. The errors introduced by the algebraic expressions in Eqs. (4-12) and (4-13) are obviously different, with one contributing potentially more error than the other.

As the difference between conductivity increases between phases, so does the potential disparity between Eqs. 4-12 and 4-13, especially for a computation with a node undergoing a change of phase. For this case the melting node would be bounded by at least one all liquid node and at least one all solid node. It should be noted that for constant conductivity between phases, Eqs. (4-12) and (4-13) reduce to identically the same algorithm. The problem arises only with a change in conductivity with phase in the discretized equations.

It should also be mentioned that in the traditional formulation for phase change, this problem cannot occur, regardless of whether the conservative or non-conservative form of the equations are differenced, since conductivity is constant in a given zone. The problem arises in the weak formulation of the equations where the location of the discontinuity is "buried" within the field equation.

Equations (4-12) and 4-13 can be checked for accuracy by comparison with an approximate analytical solution developed by Neumann and presented in Lunardini [59].

Figure 4-3 depicts the problem solved by Neumann. The problem can be formulated as:

$$\frac{\partial^2 T_s}{\partial x^2} = \frac{1}{\alpha_s}\frac{\partial T_s}{\partial t} \tag{4-14}$$

$$\frac{\partial^2 T_L}{\partial x^2} = \frac{1}{\alpha_L}\frac{\partial T_L}{\partial t} \tag{4-15}$$

$$\lim_{x \to \infty} (T_L) = T_\infty \tag{4-16}$$

$$T_s\ (0,t) = T_{surface} \tag{4-17}$$

where $T_{surface} > T_f$

at the interface

$$T_s\ (x,t) = T_L\ (x,t) = T_f \tag{4-18}$$

and

$$k_s \frac{\partial T_s}{\partial x} - k_L \frac{\partial T_L}{\partial x} = \rho\lambda \frac{\partial x}{\partial t} \tag{4-19}$$

In the solution of Eqs. (4-14) through (4-19), if the initial temperature of the liquid is T_f, and the melt distance into the problem is small, the solution can be approximated by:

$$x = \sqrt{\frac{2k_s\ (T_f - T_{surface})t}{\rho\lambda}} \tag{4-20}$$

Tables 4-1 and 4-2 show in detail the numerical computations for solving the one-dimensional phase change problem using the explicit

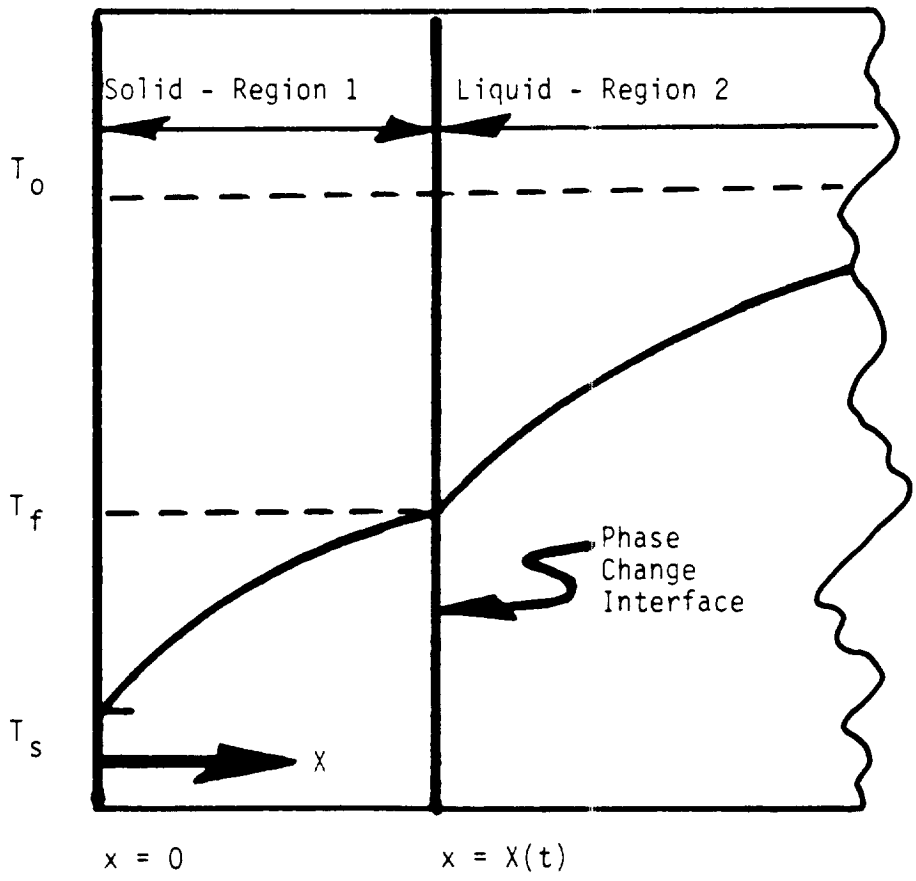


Figure 4-3 Schematic for the Phase Change Problem Solved by Neumann

EXPLICIT ALGORITHM FOR 1-D CONSERVATIVE EQUATION:

$$H^{k+1} = H^{k} + \frac{\Delta t}{\rho(\Delta x)^2}\left[\frac{k^{k}_{j+1}+k^{K}_{j}}{2}\left(T^{k}_{j+1}-T^{k}_{j}\right)\right.$$

$$\left. k^{k}_{j}+k^{k}_{j-1}\left(T^{k}_{j}-T^{k}_{j-1}\right)\right]$$

TIME	H^k	ΔH	H^{k+1}	K^k	K^{k+1}
sec.	BTU/lbm			BUT/hr-ft-°F	
0.0	0.00	0.00	0.00	1.4160	1.4160
0.1	0.00	26.04	26.04	1.4160	1.2164
0.2	26.04	23.05	49.09	1.2164	1.0398
0.3	49.09	20.39	69.48	1.0398	0.8834
0.4	69.48	18.05	87.53	0.8834	0.7450
0.5	87.53	15.98	103.51	0.7450	0.6227
0.6	103.51	14.14	117.65	0.6227	0.5143
0.7	117.65	12.51	130.16	0.5143	0.4184
0.8	130.16	11.08	141.24	0.4184	0.3335
0.9	141.24	9.80	151.04	0.3335	0.3200

Melt Time for this Node Per
the Neumann Solution: 0.7456 sec.

Error (using interpolated time): 7.37%

Table 4-1 Numerical Computation for the Conservatively Differenced Phase Change Equation

EXPLICIT ALGORITHM FOR 1-D NON-CONSERVATIVE EQUATION

$$H^{k+1} = H^k + \frac{\Delta t}{\rho(\Delta x)^2}\left[k_j^k\,(T_{j+1}^k - 2T_j^k + T_{j-1}^k) + \frac{(k_j^k - k_{j-1}^k)(T_{j+1}^k - T_{j-1}^k)}{4}\right]$$

Time	H^k	ΔH	H^{k+1}	K^k	K^{k+1}
(sec)	(BTU/lbm)			(BTU/hr-ft-°F)	
0.0	0.00	0.00	0.00	1.4160	1.4160
0.1	0.00	34.26	34.26	1.4160	1.1534
0.2	34.26	26.38	60.64	1.4160	0.9512
0.3	60.64	20.31	80.95	0.9512	0.7955
0.4	80.95	15.64	96.69	0.7955	0.6567
0.5	96.59	12.05	108.64	0.6567	0.5833
0.6	108.64	9.28	117.92	0.5833	0.5122
0.7	117.92	7.15	125.07	0.5122	0.4574
0.8	125.07	5.50	130.57	0.4574	0.4153
0.9	130.57	4.23	134.80	0.4153	0.3828
1.0	134.80	3.26	138.06	0.3828	0.3578
1.1	138.06	2.52	140.58	0.3578	0.3386
1.2	140.58	1.92	142.50	0.3586	0.3238
1.3	142.50	1.49	143.99	0.3238	0.3200

Melt Time for this Node per the Neumann Solution: 0.7456 sec.

Error (using interpolated time): 74.35%

Figure 4-2 Numerical Computations for the Non-Conservatively Differenced Phase Change Equation

C-2

conservative and non-conservative algorithms. The calculations are performed for water as the phase change substance, and are done to determine the required melt time for the first node only. Figure 4-4 provides a schematic of the problem solved. Tables 4-1 and 4-2 provide further insight into the nature of the numerical error that is introduced into the solution.

For the algorithm based on the conservative form, the heat flux must be written at each edge of the node. To estimate the thermal conductivity at the node edge, the average is taken of the conductivities of the two nodes having a common edge. The conductivity of the node "j" is determined by a linear interpolation between the solid and liquid conductivities, depending on the percentage of the node that has melted. For the algorithm based on the non-conservative form, no averaging is needed since the first order derivatives are evaluated based on values of the two adjacent nodes, which never change until the "j^{th}" node needs to be updated, depending on the percent of the node that has melted.

As the node "warms up" and melts, both solutions track each other with good agreement. The solutions begin to diverge, however, during the melting of the last third of the node. The non-conservative differencing underpredicts the heat flux entering node "j". Clearly, approximating the conductivity derivative, or, the change of conductivity through three nodes, is less accurately modelled in non-conservative differencing. Averaging the conductivities between nodes for the edge, and using these averages in a flux derivative, is a much better numerical approximation. Note that the relative error

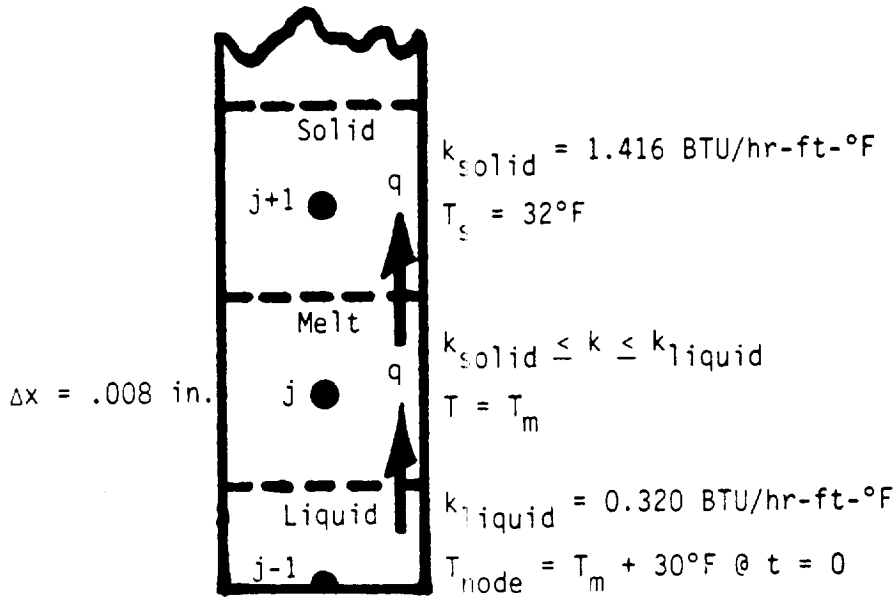


Figure 4-4: Schematic, with Properties, of One-Dimensional Grid Used to Obtain Solutions Presented in Tables 4-1 and 4-2.

between the two methods, just in a first node calculation for this crude discretization, differ by an order of magnitude. Using a smaller mesh or finer time step does not improve the approximation appreciably.

Figures 4-5 and 4-6 show the numerical results of Eqs. (4-10) and (4-11) for a Neumann class of problems. The solutions plot the movement of a melt front with time. At the start of the problem, all the nodes are solid and at the fusion temperature. Suddenly, one edge of the region experiences a step change in temperature, and is held constant at that temperature. The substance under consideration is water, which has a difference in conductivity between the two phases of approximately a factor of four. Solutions are plotted for step changes in the edge temperature of 10, 20, 30, 40 and 50 degrees Farenheit above the fusion temperature.

For this problem, which solves for the melt front passing through a number of nodes into the grid, the Crank-Nicholson implicit finite-difference scheme was employed in the numerical implementation of Eqs. (4-10) and (4-11). This scheme was chosen for the two-dimensional problem because it is unconditionally stable for the conduction equation, [60]. The solutions for the conservative and non-conservative forms with Neumann's solution using Crank-Nicholson differencing provides further justification for choosing the correct form for the two-dimensional problem. Using the Crank-Nicholson differencing scheme, Eqs. (4-10) and (4-11) for the problem solved become:

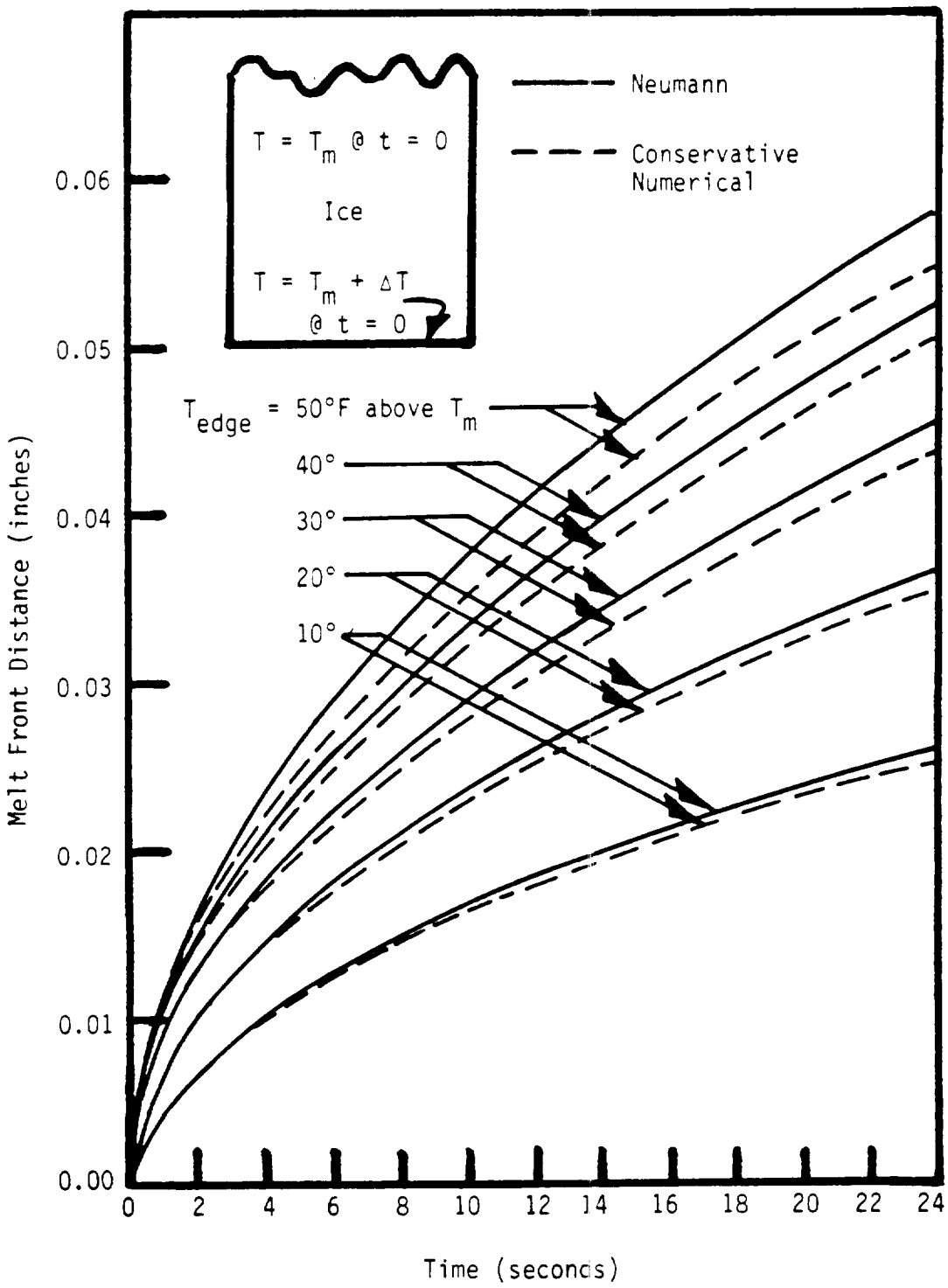


Figure 4-5: Comparison of Neumann's Solution to the Enthalpy Equation Differenced in Conservative Form

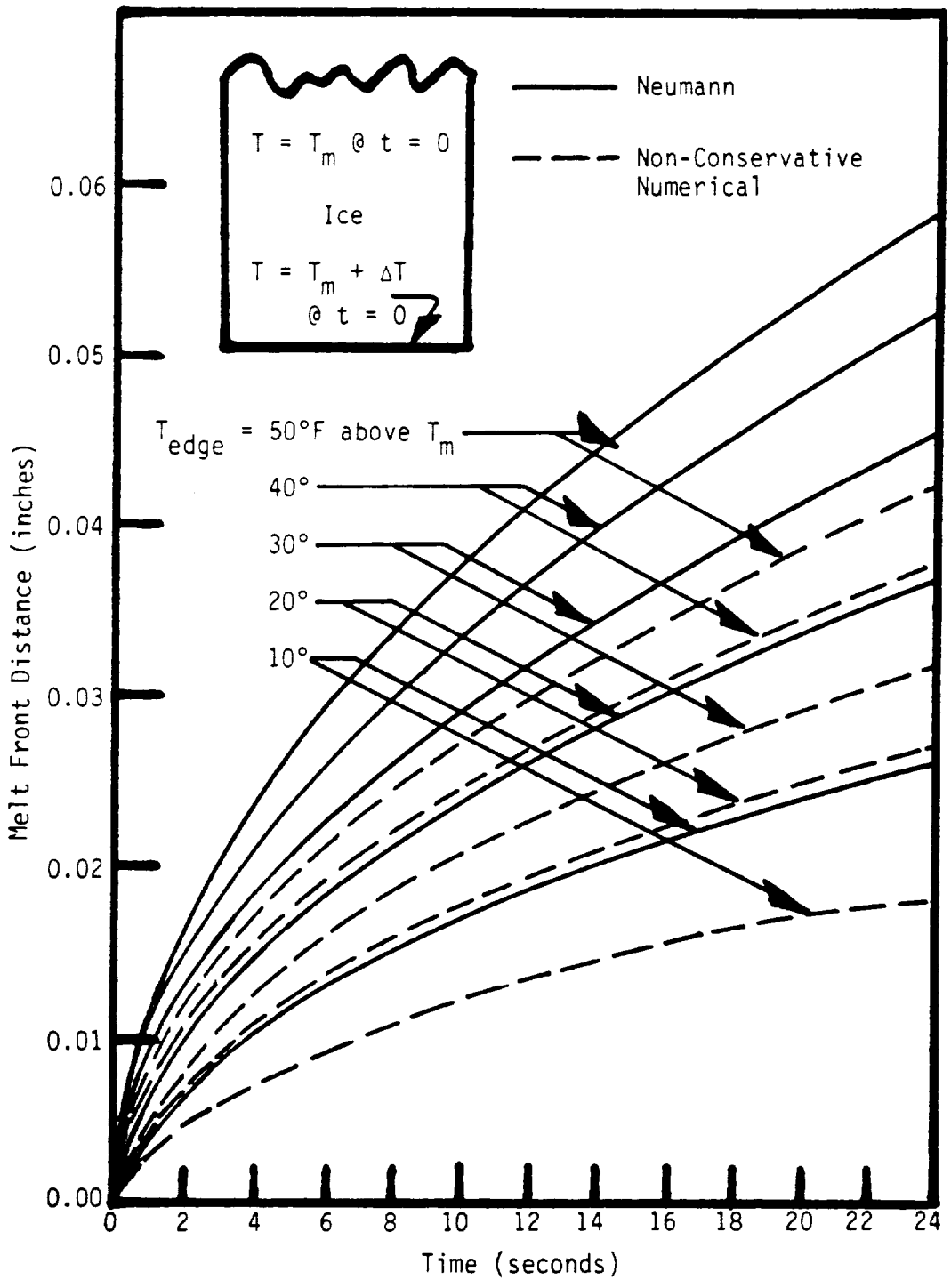


Figure 4-6: Comparison of Neumann's Solution to the Enthalpy Equation Differenced in Non-Conservative Form

$$\rho\left(\frac{H^{k+1}-H^{k}}{\Delta t}\right)=\frac{1}{(\Delta x)^{2}}\left(\frac{k_{j+1}^{k+1}+k_{j}^{k+1}}{2}\left(T_{j+1}^{k+1}-T_{j}^{k+1}\right)-\frac{k_{j}^{k+1}+k_{j-1}^{k+1}}{2}\left(T_{j}^{k+1}-T_{j-1}^{k+1}\right)\right)(.5)$$

$$+\frac{1}{(\Delta x)^{2}}\left(\frac{k_{j+1}^{k}+T_{j}^{k}}{2}\left(T_{j+1}^{k}-T_{j}^{k}\right)-\frac{k_{j}^{k}+k_{j-1}^{k}}{2}\left(T_{j}^{k}-T_{j-1}^{k}\right)\right)(.5) \qquad (4\text{-}21)$$

$$\rho\left(\frac{H^{k+1}-H^{k}}{\Delta t}\right)=\frac{1}{(\Delta x)^{2}}\left(k_{j}^{k+1}\left(T_{j+1}^{k+1}-2T_{j}^{k+1}+T_{j-1}^{k+1}\right)+\frac{\left(k_{j+1}^{k+1}-k_{j-1}^{k+1}\right)\left(T_{j+1}^{k+1}-T_{j-1}^{k+1}\right)}{4}\right)(.5)$$

$$+\frac{1}{(\Delta x)^{2}}\left(k_{j}^{k}\left(T_{j+1}^{k}-2T_{j}^{k}+T_{j-1}^{k}\right)+\frac{\left(k_{j+1}^{k}-k_{j-1}^{k}\right)\left(T_{j+1}^{k}-T_{j-1}^{k}\right)}{4}\right)(.5) \qquad (4\text{-}22)$$

when $H_{sm} \leq H_j^{k+1} \leq H_{Lm}$ $\quad T_j^{k+1} = T_m$

$$\text{and } K_j^{k+1} = k_s + (K_L - k_s)\frac{H_j^{k+1} - H_{sm}}{H_{Lm} - H_{sm}} \qquad (4\text{-}23)$$

when $H_j^{k+1} > H_{Lm}$ $\quad T_j^{k+1} = T_m + \frac{H_{Lm} - H_j^{k+1}}{\rho C_L}$

$$\text{and } k_j^{k+1} = k_L \qquad (4\text{-}24)$$

All second order derivatives are approximated using three point differencing; all first order derivatives using two point differencing. Gauss-Seidel iteration was used so that the temperature profile with the effect of the phase change could be obtained at any time step.

As can be seen from Fig. 4-5, Crank-Nicholson differencing of the conservative form of Eq. (4-5) yields results that compare favorably with Neumann's approximate analytical solution. However, the agreement deteriorates with increasing values of the step change temperature. This is to be expected, since if all numerical criteria for a computational zone remain constant, a more rapidly changing solution with real time is apt to be less well approximated by the derivatives.

The large error introduced into the solution by differencing the non-conservative form of Eq. (5-4) is very apparent in Fig. 4-6. The error (defined as the difference between Neumann's value and the numerical value divided by Neumann's value) is fairly consistent, and is equal to approximately 100 percent through the entire solution. The numerical solution shows about twice the amount of time needed to achieve a desired melt distance than that predicted by Neumann's solution.

In the numerical simulation of the weak formulation of the phase change equations, attention needs to be given to the conservative and non-conservative forms of these equations. For the particular problem at hand, clearly the conservative form should be used.

Transformation of Phase Change Equations

Since the phase change equations will be solved in the transformed plane, those portions of the equations that have a spacial dependency

must be transformed. Because it has been shown that the conservative form of the field equation must be used, only the transformation operators for the first order derivatives need to be used. From Chapter 2, the appropriate derivative operators needed to effect the transformation of the field equation are:

$$\frac{\partial}{\partial x} = \frac{1}{J}\left(\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta}\right) \tag{4-25}$$

$$\frac{\partial}{\partial y} = \frac{1}{J}\left(\frac{\partial x}{\partial \xi}\frac{\partial}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial}{\partial \xi}\right) \tag{4-26}$$

Applying these to Eq. (4-5) yields:

$$\rho\frac{\partial H}{\partial t} = \frac{1}{J^2}\left[\left(\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta}\right)\left(k\frac{\partial y}{\partial \eta}\frac{\partial T}{\partial \xi} - k\frac{\partial y}{\partial \xi}\frac{\partial T}{\partial \eta}\right)\right] + \frac{1}{J^2}\left[\left(\frac{\partial x}{\partial \xi}\frac{\partial}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial}{\partial \xi}\right)\left(k\frac{\partial x}{\partial \xi}\frac{\partial T}{\partial \eta} - k\frac{\partial x}{\partial \eta}\frac{\partial T}{\partial \xi}\right)\right] \tag{4-28}$$

Expanding Eq. (4-28) produces the following:

$$\begin{aligned}\rho\frac{\partial H}{\partial t} = \frac{1}{J^2}\Bigg[&\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi}\left(k\frac{\partial y}{\partial \eta}\frac{\partial T}{\partial \xi}\right) - \frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi}\left(k\frac{\partial y}{\partial \xi}\frac{\partial T}{\partial \eta}\right) - \frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta}\left(k\frac{\partial y}{\partial \eta}\frac{\partial T}{\partial \xi}\right)\\ &+ \frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta}\left(k\frac{\partial y}{\partial \xi}\frac{\partial T}{\partial \eta}\right) + \frac{\partial x}{\partial \xi}\frac{\partial}{\partial \eta}\left(k\frac{\partial x}{\partial \xi}\frac{\partial T}{\partial \eta}\right) - \frac{\partial x}{\partial \xi}\frac{\partial}{\partial \eta}\left(k\frac{\partial x}{\partial \eta}\frac{\partial T}{\partial \xi}\right)\\ &- \frac{\partial x}{\partial \eta}\frac{\partial}{\partial \xi}\left(k\frac{\partial x}{\partial \xi}\frac{\partial T}{\partial \eta}\right) + \frac{\partial x}{\partial \eta}\frac{\partial}{\partial \xi}\left(k\frac{\partial x}{\partial \eta}\frac{\partial T}{\partial \xi}\right)\Bigg]\end{aligned} \tag{4-29}$$

$$\rho \frac{\partial H}{\partial t} = \frac{1}{J^2}\left[\frac{\partial y}{\partial \eta}\frac{\partial y}{\partial \eta}\frac{\partial\left(k\frac{\partial T}{\partial \xi}\right)}{\partial \xi} - \frac{\partial y}{\partial \eta}\frac{\partial y}{\partial \xi}\frac{\partial\left(k\frac{\partial T}{\partial \eta}\right)}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial\left(k\frac{\partial T}{\partial \xi}\right)}{\partial \eta}\right.$$

$$+ \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \xi}\frac{\partial\left(k\frac{\partial T}{\partial \eta}\right)}{\partial \eta} + \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \xi}\frac{\partial\left(k\frac{\partial T}{\partial \eta}\right)}{\partial \eta} - \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial\left(k\frac{\partial T}{\partial \xi}\right)}{\partial \eta}$$

$$\left. - \frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \xi}\frac{\partial\left(k\frac{\partial T}{\partial \eta}\right)}{\partial \xi} - \frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \eta}\frac{\partial\left(k\frac{\partial T}{\partial \xi}\right)}{\partial \xi}\right] \qquad (4\text{-}30)$$

$$\rho \frac{\partial H}{\partial t} = \frac{1}{J^2}\left\{\left[\left(\frac{\partial y}{\partial \eta}\right)^2 + \left(\frac{\partial x}{\partial \eta}\right)^2\right]\frac{\partial\left(k\frac{\partial T}{\partial \xi}\right)}{\partial \xi}\right.$$

$$- \left(\frac{\partial y}{\partial \eta}\frac{\partial y}{\partial \xi} + \frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \xi}\right)\frac{\partial\left(k\frac{\partial T}{\partial \eta}\right)}{\partial \xi}$$

$$- \left(\frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}\right)\frac{\partial\left(k\frac{\partial T}{\partial \xi}\right)}{\partial \eta}$$

$$\left. + \left[\left(\frac{\partial y}{\partial \xi}\right)^2 + \left(\frac{\partial x}{\partial \xi}\right)^2\right]\frac{\partial\left(k\frac{\partial T}{\partial \eta}\right)}{\partial \eta}\right\} \qquad (4\text{-}31)$$

But it was shown in Chapter 2 that

$$\alpha = \left(\frac{\partial y}{\partial \eta}\right)^2 + \left(\frac{\partial x}{\partial \eta}\right)^2 \qquad (4\text{-}32)$$

$$\beta = \frac{\partial y}{\partial \eta}\frac{\partial y}{\partial \xi} + \frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \xi} \tag{4-33}$$

$$\gamma = \left(\frac{\partial y}{\partial \xi}\right)^2 + \left(\frac{\partial x}{\partial \xi}\right)^2 \tag{4-34}$$

Thus Eq. (4-31) may be written as

$$\frac{\partial H}{\partial t} = \frac{1}{J^2}\left[\alpha\frac{\partial\left(k\frac{\partial T}{\partial \xi}\right)}{\partial \xi} - \beta\frac{\partial\left(k\frac{\partial T}{\partial \eta}\right)}{\partial \xi} - \beta\frac{\partial\left(k\frac{\partial T}{\partial \xi}\right)}{\partial \eta} + \gamma\frac{\partial\left(k\frac{\partial T}{\partial \eta}\right)}{\partial \eta}\right] \tag{4-35}$$

Equation (4-35) is the transformed enthalpy equation [Eq. (4-27)] that must be solved in the transformed plane.

The boundary condition equations are the same as those for a layered conduction problem, that is, equality of temperature and normal heat flux at the interfaces:

$$T_1\Big|_I = T_2\Big|_I \tag{4-36}$$

$$k_1\left(\frac{\partial T}{\partial n}\right)_1\Big|_I = k_2\left(\frac{\partial T}{\partial n}\right)_2\Big|_I \tag{4-37}$$

For a convection condition:

$$-k\frac{\partial T}{\partial n}\Big|_s = h\,(T_s - T_\infty) \tag{4-38}$$

Transformation operators are needed to provide a value of the temperature derivative normal to a surface at a specified point. Again, from Chapter 2, these are

$$\left.\frac{\partial}{\partial n}\right|_{\substack{\text{To a line of}\\ \text{constant } \eta}} = \frac{1}{J\sqrt{\gamma}}\left(\gamma\frac{\partial T}{\partial \eta} - \beta\frac{\partial T}{\partial \xi}\right) \qquad (4\text{-}39)$$

On the presumption that the boundary is along the upper or lower edge in the transformed plane (a line of constant η), Eqs. (4-37) and (4-38) become

$$k_1\left(\left.\frac{\gamma}{J\sqrt{\gamma}}\frac{\partial T}{\partial \eta}\right|_I - \left.\frac{\beta}{J\sqrt{\gamma}}\frac{\partial T}{\partial \xi}\right|_I\right)_1 = k_2\left(\left.\frac{\gamma}{J\sqrt{\gamma}}\frac{\partial T}{\partial \eta}\right|_I - \left.\frac{\beta}{J\sqrt{\gamma}}\frac{\partial T}{\partial \xi}\right|_I\right)_2$$

$$- k\left(\left.\frac{\gamma}{J\sqrt{\gamma}}\frac{\partial T}{\partial \eta}\right|_s - \left.\frac{\beta}{J\sqrt{\gamma}}\frac{\partial T}{\partial \xi}\right|_s\right) = h(T_s - T_\infty) \qquad (4\text{-}41)\ (4\text{-}42)$$

Equations (4-41) and (4-42) are the transformed normal heat flux equations that must be solved in the transformed plane.

Discretization of the Phase Change Equations

Figure 4-7 depicts a section of the grid in the transformed plane. Because of the fact that the individual elements in the real plane may be irregular and unsymmetric, all eight nodes surrounding the node of concern must be involved in the differencing. It becomes more apparent as to the manner in which Eq. (4-31) needs to be differenced if the equation is "numerically" broken down as follows:

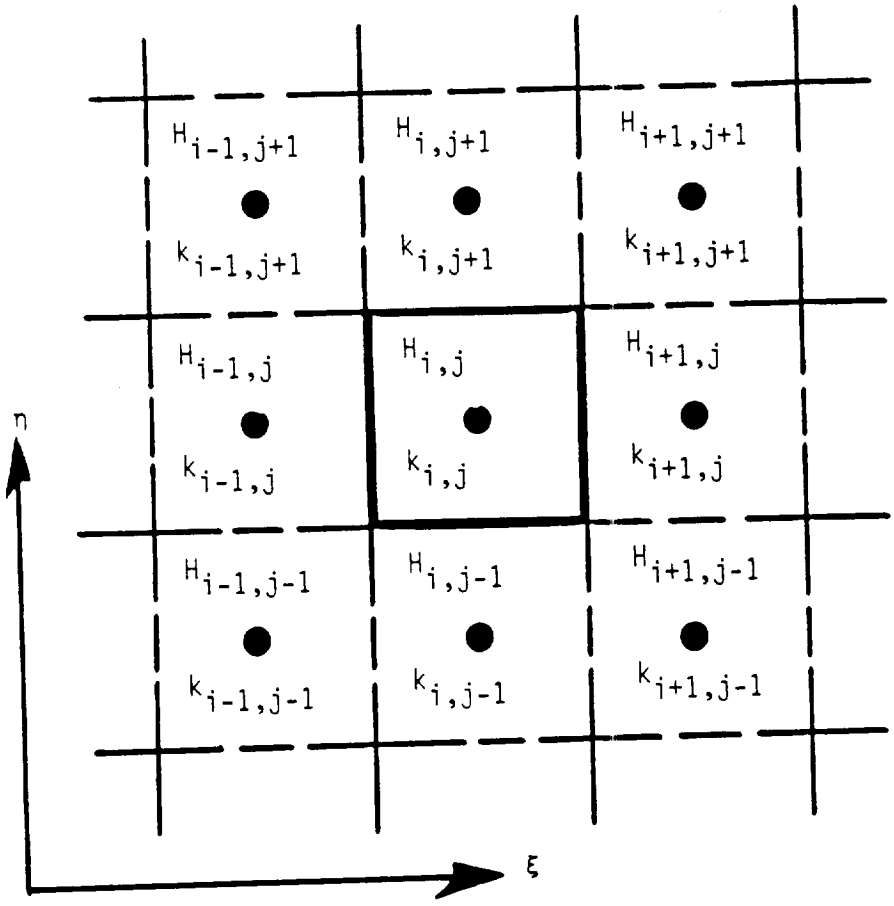


Figure 4-7: Schematic of Grid with a General Node Assignment in the Transformed Plane

$$J^{\prime}\rho \frac{H^{k+1} - H^{k}}{\Delta t} = (1-M)\left[\alpha\left(k\frac{\partial T}{\partial \xi}\Big|^{k}_{i+1/2,j} - k\frac{\partial T}{\partial \xi}\Big|^{k}_{i-1/2,j}\right) - \frac{\beta}{2}\left(k\frac{\partial T}{\partial \eta}\Big|^{k}_{i+1,j} - k\frac{\partial T}{\partial \eta}\Big|^{k}_{i-1,j}\right) - \frac{\beta}{2}\left(k\frac{\partial T}{\partial \xi}\Big|^{k}_{i,j+1} - k\frac{\partial T}{\partial \xi}\Big|^{k}_{i,j-1}\right) + \gamma\left(k\frac{\partial T}{\partial \eta}\Big|^{k}_{i,j+1/2} - k\frac{\partial T}{\partial \eta}\Big|^{k}_{i,j-1/2}\right)\right]$$

$$+ M\left[\alpha\left(k\frac{\partial T}{\partial \xi}\Big|^{k+1}_{i+1/2,j} - k\frac{\partial T}{\partial \xi}\Big|^{k_{+}1}_{i-1/2,j}\right) - \frac{\beta}{2}\left(k\frac{\partial T}{\partial \eta}\Big|^{k+1}_{i+1,j} - k\frac{\partial T}{\partial \eta}\Big|^{k+1}_{i-1,j}\right) - \frac{\beta}{2}\left(k\frac{\partial T}{\partial \xi}\Big|^{k+1}_{i,j+1} - k\frac{\partial T}{\partial \xi}\Big|^{k+1}_{i,j-1}\right) + \gamma\left(k\frac{\partial T}{\partial \eta}\Big|^{k+1}_{i,j+1/2} - k\frac{\partial T}{\partial \eta}\Big|^{k+1}_{i,j-1/2}\right)\right] \tag{4-43}$$

As can be seen, the Crank-Nicholson numerical scheme is being used. Three point differencing is applied to all second order derivatives and two point differencing to all first order derivatives. Note that the locations (I+1/2,J), (I-1/2,J), (I,J+1/2), (I,J-1/2) are at the boundaries of the central node. If the central node should be undergoing a change of phase, it will have at least one solid and at

least one all liquid node adjacent to it. Assigning a correct value of conductivity at the "edge" now becomes a concern. The approach used by Marano [5] and Chao [7] will be used. The conductivity will be estimated by simply averaging the conductivities of the two nodes having this edge in common. The differenced field equation for the phase change region then becomes

$$J_{i,j}^{2}\,\rho\,\frac{H_{i,j}^{k+1}-H_{i,j}^{k}}{\Delta t}=(1-M)\left\{\alpha_{i,j}\,\frac{k_{i+1,j}^{k}+k_{i,j}^{k}}{2}\,(T_{i+1,j}^{k}-T_{i,j}^{k})\right.$$

$$-\alpha_{i,j}\,\frac{k_{i,j}^{k}+k_{i-1,j}^{k}}{2}\,(T_{i,j}^{k}-T_{i-1,j}^{k})$$

$$-\frac{\beta_{i,j}}{2}\left[k_{i+1,j}^{k}\,(T_{i+1,j+1}^{k}-T_{i+1,j-1}^{k})-k_{i-1,j}^{k}\,(T_{i-1,j+1}^{k}-T_{i-1,j-1}^{k})\right]$$

$$-\frac{\beta_{i,j}}{2}\left[k_{i,j+1}^{k}\,(T_{i+1,j+1}^{k}-T_{i-1,j+1}^{k})-k_{i,j-1}^{k}\,(T_{i+1,j-1}^{k}-T_{i-1,j-1}^{k})\right]$$

$$+\gamma_{i,j}\,\frac{k_{i,j+1}^{k}+k_{i,j}^{k}}{2}\,(T_{i,j+1}^{k}-T_{i,j}^{k})$$

$$\left.-\gamma_{i,j}\,\frac{k_{i,j}^{k}+k_{i,j-1}^{k}}{2}\,(T_{i,j}^{k}-T_{i,j-1}^{k})\right\}$$

(Equation continued on next page.)

$$+ (M) \quad \alpha_{i,j} \left\{ \frac{k^{k+1}_{i+1,j} + k^{k+1}_{i,j}}{2} \left(T^{k+1}_{i+1,j} - T^{k+1}_{i,j}\right) \right.$$

$$- \alpha_{i,j} \frac{k^{k+1}_{i,j} + k^{k+1}_{i-1,j}}{2} \left(T^{k+1}_{i,j} - T^{k+1}_{i-1,j}\right)$$

$$- \frac{\beta_{i,j}}{2} \left[k^{k+1}_{i+1,j} \left(T^{k+1}_{i+1,j+1} - T^{k+1}_{i+1,j-1}\right) - k^{k+1}_{i-1,j} \left(T^{k+1}_{i-1,j+1} - T^{k+1}_{i-1,j-1}\right) \right]$$

$$- \frac{\beta i,j}{2} \left[k^{k+1}_{i,j+1} \left(T^{k+1}_{i+1,j+1} - T^{k+1}_{i-1,j+1}\right) - k^{k+1}_{i,j-1} \left(T^{k+1}_{i+1,j-1} - T^{k+1}_{i-1,j-1}\right) \right]$$

$$+ \gamma_{i,j} \frac{k^{k+1}_{i,j+1} + k^{k+1}_{i,j}}{2} \left(T^{k+1}_{i,j+1} - T^{k+1}_{i,j}\right)$$

$$\left. - \gamma_{i,j} \frac{k^{k+1}_{i,j} + k^{k+1}_{i'j-1}}{2} \left(T^{k+1}_{i,j} - T^{k+1}_{i,j-1}\right) \right\} \qquad (4\text{-}44)$$

Note that with the insertion of "M" and "1-M", the algorithm can become anything from purely implicit (value of "M" equal to zero) to purely explicit (value of "M" equal to 1.0). For an "M" of 0.5, the algorithm becomes the Crank-Nicholson scheme.

The boundary condition for equality of normal heat flux at a solid boundary becomes:

$$k_{i,j}\left(\frac{\gamma_{i,1}}{J_{i,1}\sqrt{\gamma_{i,1}}}(T_{i,2}-T_{i,1})-\frac{\beta_{i,1}}{J_{i,1}\sqrt{\gamma_{i,1}}}\frac{T_{i+1,1}-T_{i-1,1}}{2}\right)_{upper}$$

$$=\ k_{i,jmax}\left(\frac{\gamma_{i,jmax}}{J_{i,jmax}\sqrt{\gamma_{i,jmax}}}(T_{i,jmax}-T_{i,jmax-1})\right.$$

$$\left.-\frac{\beta_{i,jmax}}{J_{i,jmax}\sqrt{\gamma_{i,jmax}}}\frac{T_{i+1,jmax}-T_{i-1,jmax}}{2}\right)_{lower} \qquad (4\text{-}45)$$

The algorithm is similar for a convective boundary, except that $h(T(I,1) - T\ infinity)$ or $h(T(I,J_{max}) - T\ infinity)$ replaces either the left or right hand side of Eq.(4-45), depending on whether the convective boundary is on the upper or lower surface of the grid.

It should be noted here that one may wish to account for the latent heat of fusion for the half-cell at the boundary in the phase change region for large mesh sizes. Though the mathematical formulation is correct, it applies only to a point. When numerically modelling a problem, however, the numerical equivalent applies to a discrete region in space. In the problem being modelled, one half of this discrete region at the boundary absorbs heat in the amount of the latent heat of fusion. For a large discrete region (large mesh size) the error introduced in directly applying Eq. (4-45), without modifications to incorporate the latent heat of fusion in the half cell, may be significant.

Logic for the Numerical Solution of Phase Change Field Equation

One approach for numerically implementing Eq. (4-31) under the conditions specified by Eqs. (4-6) through (4-8) in a manner that is computationally convenient, is to write three field equations; one each for the solid, melt, and liquid states. In doing this, only the equation for the melt state needs to be written with an enthalpy term, while the other two equations can be left in terms of the temperature. The nodal enthalpy is then calculated separately from the iterated temperature. The three field equations become

$$\rho c_S \frac{\partial T}{\partial t} = \frac{1}{J^2}\left[\alpha \frac{\partial\left(k \frac{\partial T}{\partial \xi}\right)}{\partial \xi} - \beta \frac{\partial\left(k \frac{\partial T}{\partial \eta}\right)}{\partial \xi} - \beta \frac{\partial\left(k \frac{\partial T}{\partial \xi}\right)}{\partial \eta} + \gamma \frac{\partial\left(k \frac{\partial T}{\partial \eta}\right)}{\partial \eta}\right]$$

where $H = \rho_S c_S T$ and $k = k_S$ at the node (4-46)

$$\rho \frac{\partial H}{\partial t} = \frac{1}{J^2}\left[\alpha \frac{\partial\left(k \frac{\partial T}{\partial \xi}\right)}{\partial \xi} - \beta \frac{\partial\left(k \frac{\partial T}{\partial \eta}\right)}{\partial \xi} - \beta \frac{\partial\left(k \frac{\partial T}{\partial \xi}\right)}{\partial \eta} + \gamma \frac{\partial\left(k \frac{\partial T}{\partial \eta}\right)}{\partial \eta}\right]$$

where $k = k_S + (k_L - k_S)\dfrac{H_{node} - H_{sm}}{H_{Lm} - H_{sm}}$ (4-47)

and $T = T_m$ at the node

$$\rho c_L \frac{\partial T}{\partial t} = \frac{1}{J^2}\left[\alpha \frac{\partial\left(k \frac{\partial T}{\partial \xi}\right)}{\partial \xi} - \beta \frac{\partial\left(k \frac{\partial T}{\partial \eta}\right)}{\partial \xi} - \beta \frac{\partial\left(k \frac{\partial T}{\partial \xi}\right)}{\partial \eta} + \gamma \frac{\partial\left(k \frac{\partial T}{\partial \eta}\right)}{\partial \eta}\right]$$

where $H = H_{Lm} + \rho_L c_L (T - T_m)$

and $k = k_L$ at the node (4-48)

It should be noted that the differencing for the solid and liquid field equations must be almost identical to the melt equation (which resulted in Eq. (4-44)). The only difference is that the specific heat is not combined with temperature in the time derivative to give the enthalpy variable. The reason for this is that even though a node may be all solid or liquid, it may have next to it a node undergoing a change of phase. The heat flux through a cell edge adjacent to a melting node will not be properly accounted for unless there is a proper "averaging" of the conductivities.

Figure 4-8 graphically depicts the iteration logic for a node in a phase change region. A check is first made on the previously calculated enthalpy of a node. Based on this check, either the solid, melt, or liquid algorithm is used. A new value of enthalpy is calculated, and a check is performed to see whether this value is within the range of enthalpies for which the algorithm is valued. If the check is positive, the subroutine moves on to the next node. If it is negative, a new algorithm is chosen, based on the new enthalpy value. In the event that none of the three algorithms calculate an enthalpy value valid within its range, a counter in the subroutine (which increases by a single integer each time an algorithm is used for a given node) will terminate the calculations, indicating that convergence was unachieveable for a node in the melt region.

It should be noted that in the melting subroutine a convergence problem often occurs at those points which represent the onset and conclusion of a phase change. In other words, for example, at the onset of melting, the solid phase algorithm will calculate an enthalpy that is

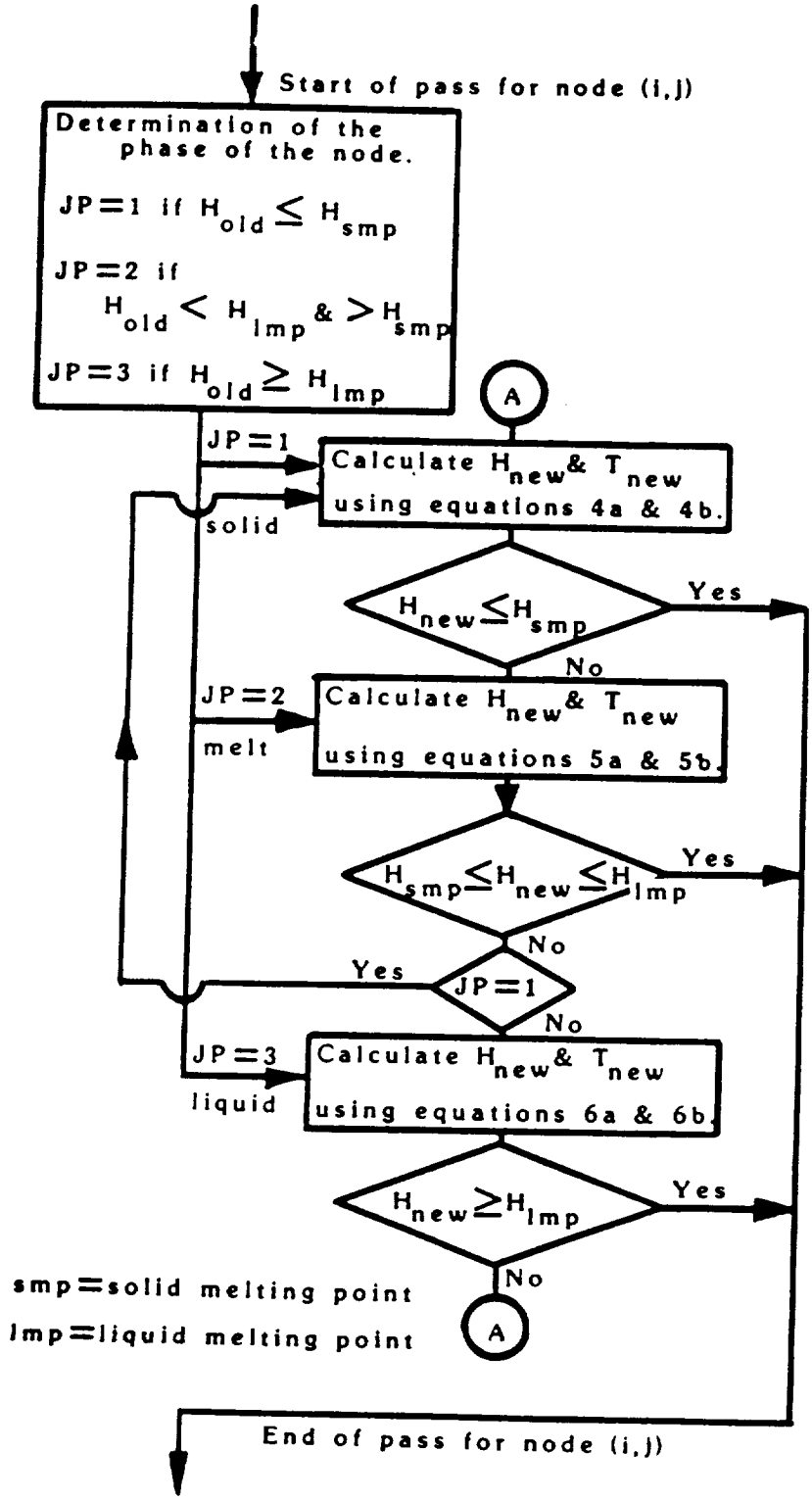


Figure 4-8: Code Logic for the Melting Subroutine

just slightly above its "legal" range, and the melt algorithm will calculate an enthalpy that is just slightly below its "legal" range. This convergence problem is alleviated, with virtually no effect on accuracy, by numerically allowing the "legal" ranges to overlap each other slightly.

Appendix F lists the key subroutine that implements the algorithms for the enthalpy formulation of the phase change equations. Appendix G lists the key subroutine that implements the conductive boundary condition for a phase change region.

Comparison with One-Dimensional Iced Airfoil

Figure 4-9 shows a comparison between results obtained using the computer code developed here and Marano's [5] one-dimensional code developed for a composite body with phase change, which was intended for the purpose of modeling an iced airfoil. Both codes determined the transient response of a "standard" electrothermal deicer, as defined in Marano's work in Table 3-6. The standard deicer was initially at a uniform temperature throughtout. At the initial instant of time, a nichrome heating element was engaged having an equivalent surface heat flux of 25 watts per square inch. Figure 4-9 plots the movement of the melt line in a layer of ice as the composite "warms up", beginning with initial temperatures of 10 and 20 degrees Farenheit below the melt temperature. In the current study, the one-dimensional problem was simulated by using two-dimensional concentric cylinders to identify layer boundaries. This problem, for uniform conditions in the angular direction and a sufficiently large circle radius compared to layer

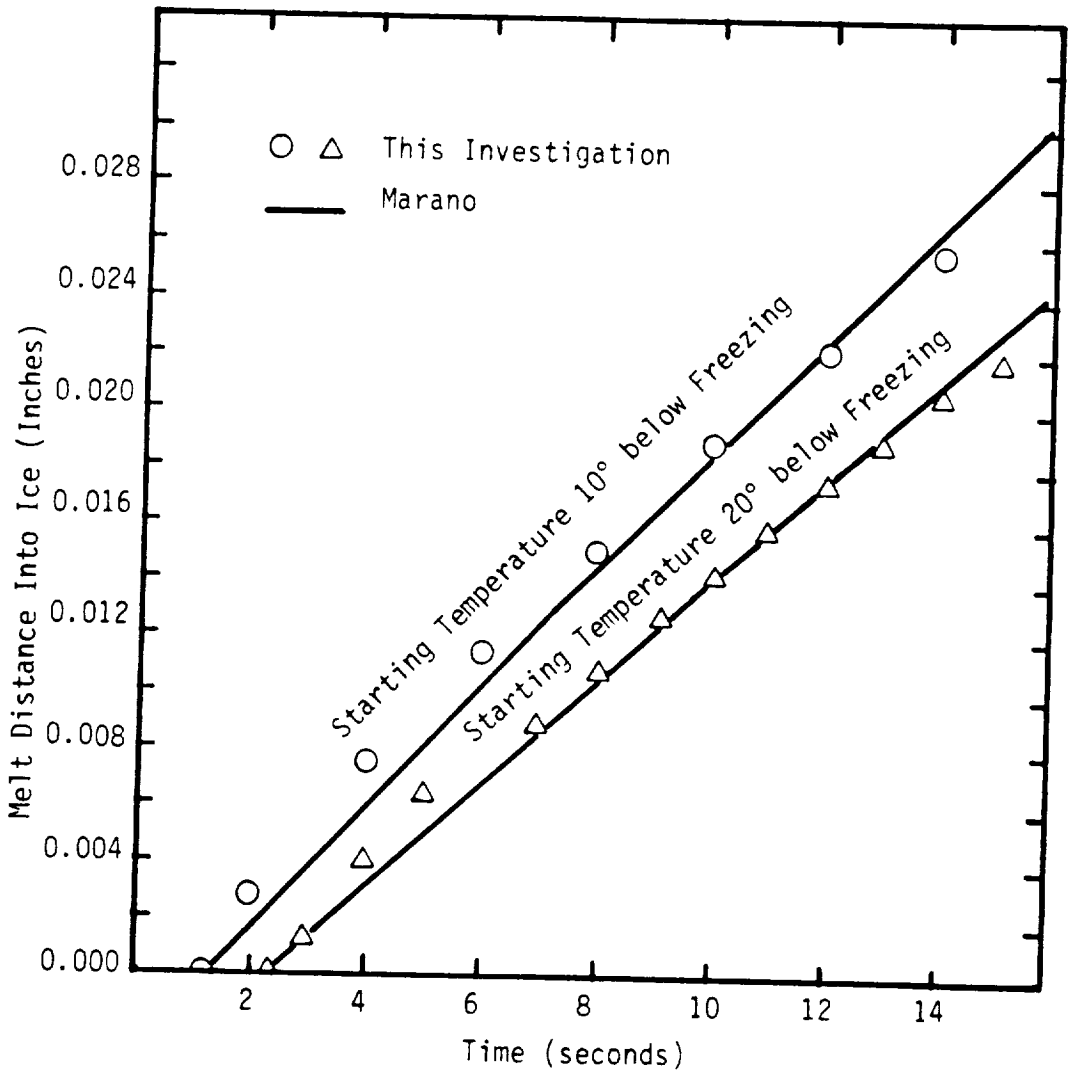


Figure 4-9: Comparison of this Investigation's Phase Front Movement with Marano's Code

thickness, will model

Marano's one-dimensional problem with negligible error due to geometry. The temperatures predicted by both codes compare favorably.

There is initially a slight overprediction in the melt front, with a crossing of lines and a subsequent slight underprediction for later times in the problem. This result would be expected in light of the comparison with the transient response of the ice/abrasion shield interface temperature presented in Chapter 3, Figure 3-6, for the pure conduction problem. As the initial temperature rise at the interface slightly leads the results shown by Baliga [4], Marano [5], and Chao [7], it is to be expected that initially the melt line slightly leads in the phase change problem. Also, as the temperature rise lags slightly later in the problem, one would expect a corresponding lag in the melt line for the phase change problem.

CHAPTER 5 EXPERIMENTAL COMPARISON WITH AN ICED AIRFOIL

Over the years extensive research in the areas of ice accretion, de-icing, and the effects of icing on aero-performance have been investigated in a subsonic icing wind tunnel located at the NASA Lewis Research Facility in Cleveland, Ohio. The IRT (Icing Research Tunnel) is essentially the same as any closed loop subsonic wind tunnel, with two significant differences. First, the tunnel contains a bank of chillers which obtain/maintain tunnel temperatures well below freezing. Secondly, a spray rig exists upstream of the test section to inject a water mist into the airstream. Under proper conditions, ice will accrete on an object in the test section. The intent is to simulate the natural accretion of ice on a body as it would occur in flight under icing conditions.

Recently, a battery of tests was undertaken in the IRT to investigate the performance of an electrothermal de-icer pad installed in a section of a UH1H helicopter rotor blade. A portion of the test results have been reported and analysed by Leffel, et al. [9]. The electrothermal de-icer used in the blade was designed and manufactured by the B. F. Goodrich Co. The testing was conducted in four phases: dry air tests, wet air tests, accretion documentation tests, and, finally, de-icing tests. The computer code developed in the current work was used to simulate the thermal response for a section of the blade for one of the de-icing tests.

Figure 5-1 illustrates the layered construction of the blade section that was fabricated for the test. The assembly is similar to that of a standard de-icer, except that the substrate is composed of three separate layers: an aluminum skin which is wrapped around a doubler that sits on the D-spar. At the leading edge, the doubler thickens and becomes what is known as the noseblock. The noseblock, typically, is made from brass, not aluminum. Each of the layers are bonded together with either an epoxy glue or a film adhesive. Note that with the bonding materials, there are a total of thirteen layers, not including any accreated ice. The heater was divided into eight separate one inch zones along the arc. Each zone car fire independent of the other zones.

As the blade was being fabricated, three layers were heavily instrumented with thermocouples. The thermocouple placements within the layers are graphically depicted in Fig. 5-2. The thermocouples were placed at the inner side of the D-spar, the inner side of the heater mat, and the outer side of the abrasion shield. The rows of thermocouples were placed arc-wise at the heater segment centers, through three "cuts" of the test section. The transient responses of similarly located thermocouples in the three cuts were an indicator of uniformity of de-icing performance in the span-wise direction. Each cut also serves as a backup to insure a reading at a specified location in the event of a thermocouple failure at a similar location.

The material properties of the layers in the test section, along with average values of their thicknesses, are presented in Table 5-1. Modeling a de-icer pad consisting of the thirteen layers (fourteen with a layer of accreted ice) for a reasonable number of nodes along the arc

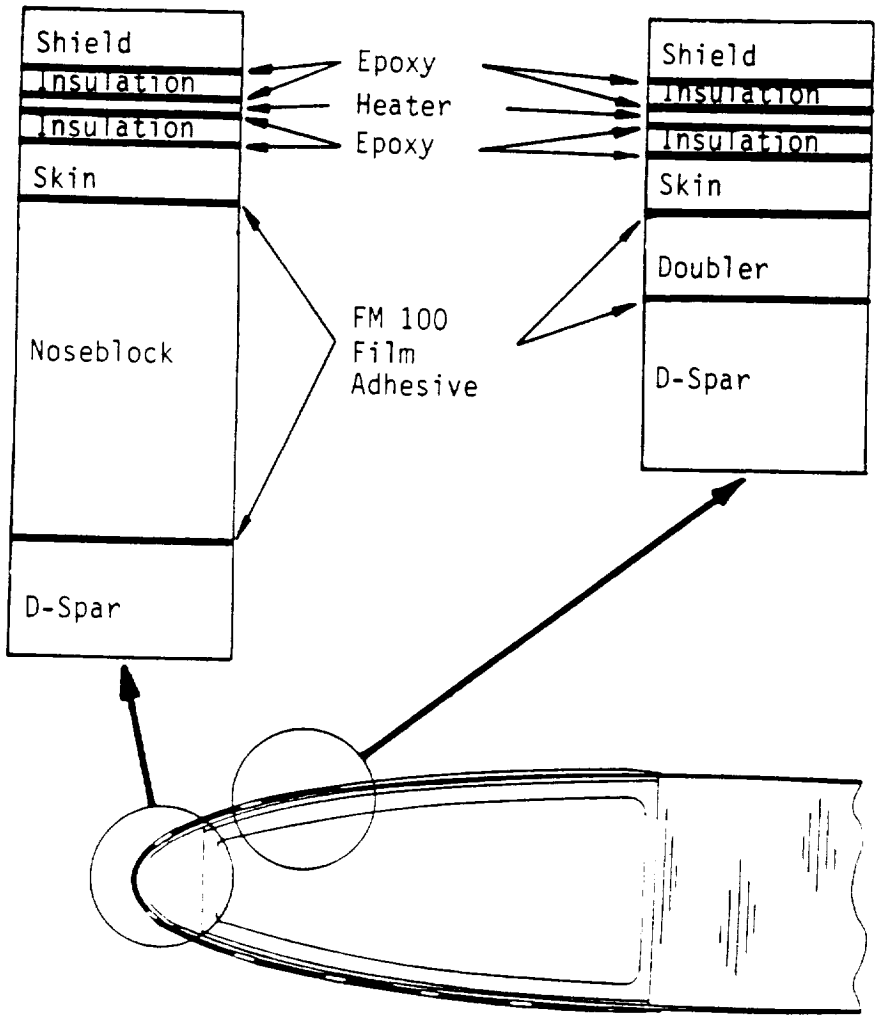


Figure 5-1: Electrothermal De-Icer and Blade Construction of a UH1H Helicopter Rotor Blade

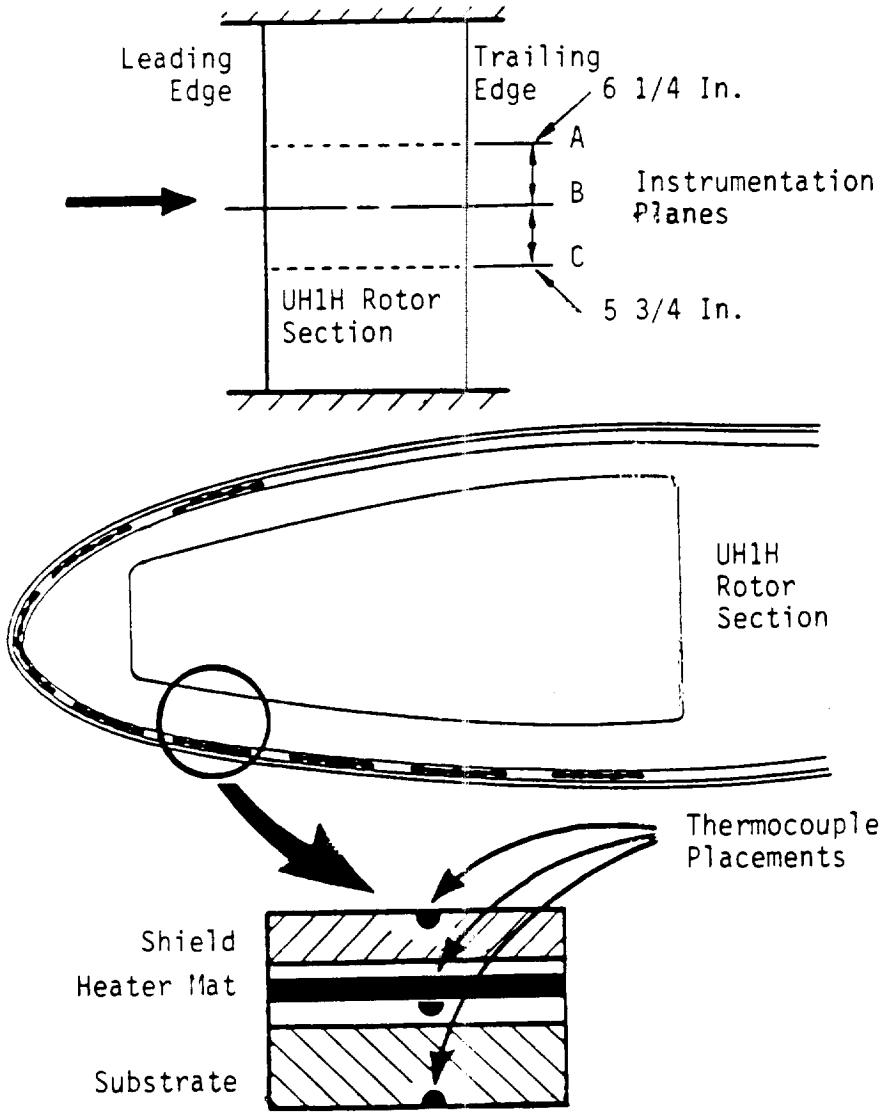


Figure 5-2: Instrumentation Planes and Thermocouple Placements for the UH1H Blade Test Section

ACTUAL BLADE GEOMETRY			NUMERICAL SIMULATION		
Values of k in BTU/hr-ft-°F Values of α in Square ft/hr Values of thickness in inches			Values of k in BTU/hr-ft-°F Values of α in Square ft/hr Values of thickness in inches		
Layer	Properties	Thickness	Thickness	Properties	Layer
Abrasion Shield	k = 8.7 α = 0.15	0.030	0.030	k = 8.7 α = 0.15	Abrasion Shield
Adhesive Epoxy	k = 0.1 α = 0.0058	0.0168			
Insulation	k = 0.22 α = 0.0087	0.0138	0.0388	k = 0.1 α = 0.0058	Insulation
Adhesive Epoxy	k = 0.1 α = 0.0058	0.0082			
Heating Element	k = 60.0 α = 1.15	0.0065	0.0065	k = 60 α = 1.15	Heating Element
Adhesive Epoxy	k = 0.1 α = 0.0058	0.0082			
Insulation	k = 0.22 α = 0.0087	0.138	0.1544	k = 0.1 α = 0.0058	Insulation
Adhesive Epoxy	k = 0.01 α = 0.0058	0.0082			
Blade Skin	k = 8.7 α = 0.15	0.02			
Film Adhesive	k = 0.1 α = 0.0058	0.01			
Aluminum Doubler	k = 102 α = 2.83	0.05	0.265	k = 102 α = 2.83	Aluminum D-Spar
Film Adhesive	k = 0.1 α = 0.0058	0.01			
Aluminum D-Spar	k = 102 α = 2.83	0.175			

Table 5-1 Actual Blade Thicknesses and Material Properties vs. Those Used in Numerical Simulation

would be computationally quite difficult; the estimated CPU time would be extremely large. Thus, a number of layers were "lumped" together in the numerical simulation of the blade. The adhesive epoxy on both sides of the upper and lower insulation layers has been lumped into the insulation. The total thickness assigned is the sum of three individual thicknesses, with the material properties being those of the adhesive. The blade skin, doubler and D-spar, with the two layers of film adhesive, have been lumped into a single "D-spar" layer. Again, the total assigned thickness to this layer is the sum of the five individual thicknesses, with the assigned properties being those of aluminum. Lumping the layers in this fashion for the lower insulation and the D-spar should have very little effect on the thermal transients in the heater and abrasion shield. In the blade construction the lower insulation has 16 times the thickness of the film adhesive, with the properties of the film being on the same order of magnitude as the insulation. Since the lower insulation is 10 times thicker than the upper insulation, most of the energy initially generated in the heaters will be driven toward the abrasion shield. Thus, any lumping below the lower insulation should have a negligible effect on the temperature transients in layers above the heater, especially for short real times into the problem. Lumping together the upper insulation with two epoxy layers, and having material values of the epoxy, may slightly retard the temperature at the abrasion shield. The insulation material values are slightly higher for both conductivity and diffusivity.

The UH1H airfoil is the same as the NACA 0012. Using dimensionless NACA 0012 coordinates provided by Abbot and Doenhoff [61], the coordinates for those portions of the airfoil containing the heaters

were generated. These coordinates were assigned node numbers for subsequent numerical computations, which are displayed in Fig. 5-3. These nodes represent the coordinate locations for the outer edge of the abrasion shield. A coordinate generating subroutine, beginning with these nodes as a starting point, then generated the boundary coordinates for any inner or outer layers. On the outer edge of the abrasion shield, the nodes were spaced at one eighth inch intervals. This made the heater zones one inch wide, with each heater having a node with an adjacent heater. Nodal locations 21, 29, 37, 45, 53, 61, 69 and 77 represent the thermocouple arc locations on the UH1H test section.

The test case simulated by the numerical code was designated in Leffel's work as reading 234, position 5 (node 69), thermocouples 53 (abrasion shield) and 26 (heater). For this particular test, the wind tunnel test speed was 100 mph, the ambient temperature was 16 °F, the angle of attack was zero degrees, the heater power density was 8 watts per square inch, and the air liquid water content was 2.2 grams per cubic meter with an average droplet diameter of 19.2 microns. The accretion test for this run showed that approximately three eighths of an inch of ice had accreted on the test section near node 69. For the first cycling of heater zone "G" (see Fig. 5-3), the heater was engaged for twenty seconds, and then disengaged for thirty seconds.

A comparison of the thermocouple data generated at position 5 (node 69) with the numerical simulation predictions is presented in Fig. 5-4 for the conditions described above. As can be seen, there is excellent agreement between the numerical and experimental results for that portion of the cycle for which the heater is engaged, and for the first third of

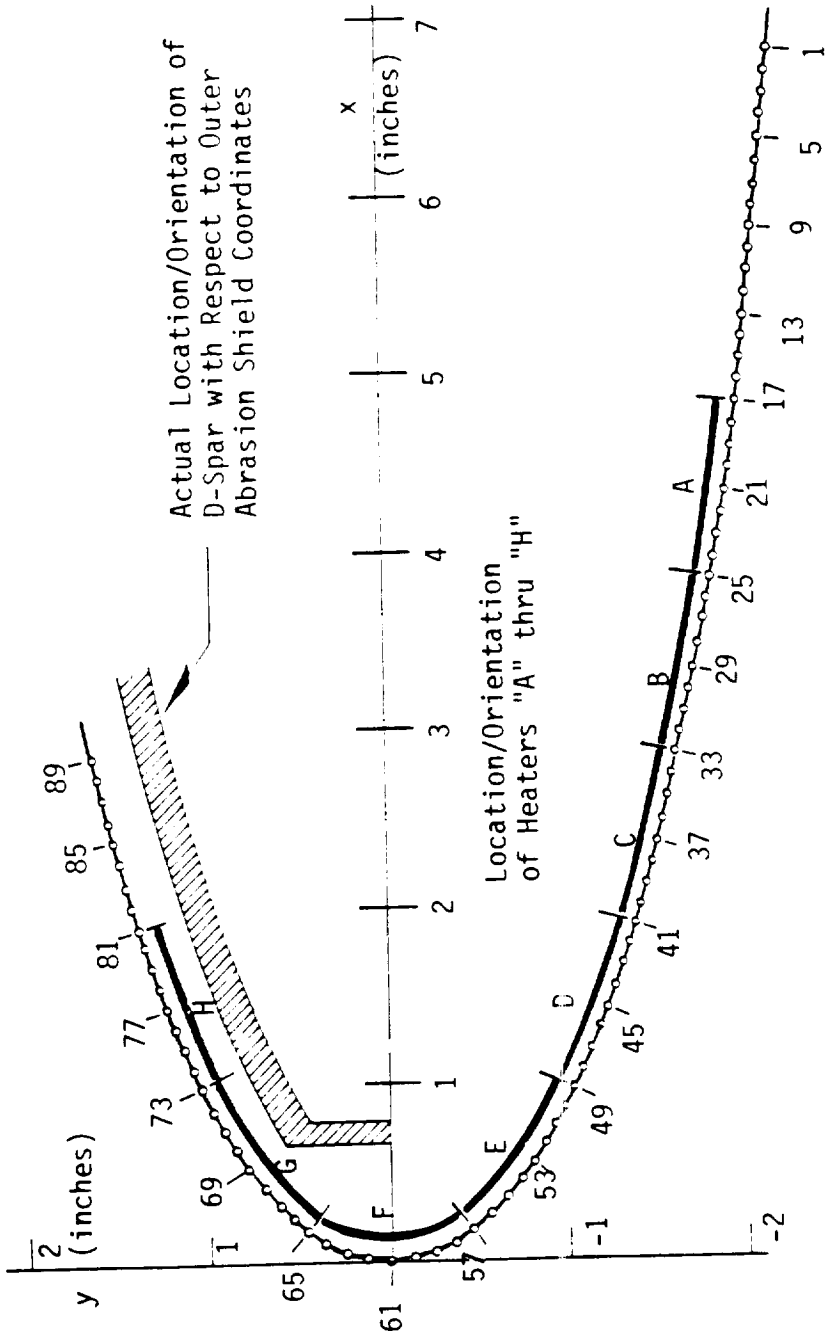


Figure 5-3: Node Numberings and Heater Designations for Numerical Model of the UH1H Rotor Test Section. Nodes Shown on Axes Represent Location of Outer Surface of Abrasion Shield

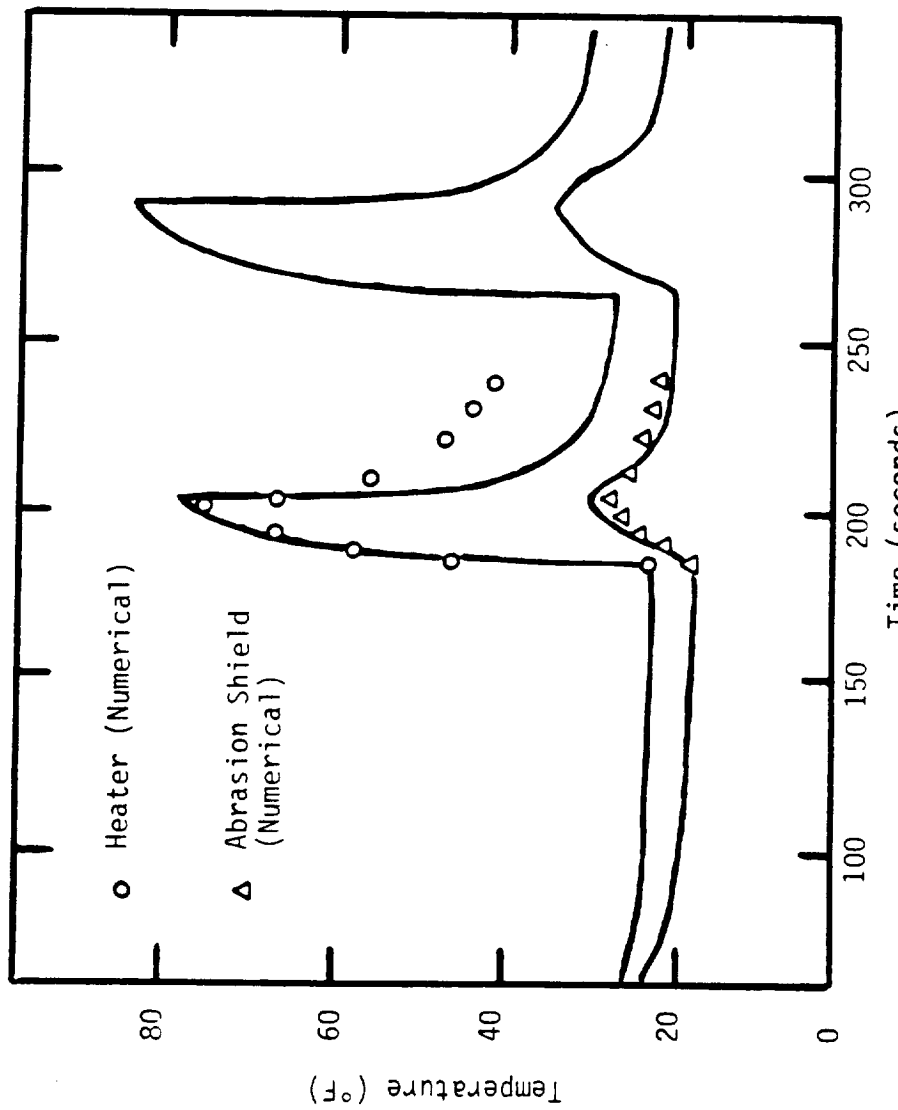


Figure 5-4: Comparison of an Experimental Result (Position 5, Reading 234) with Numerical Prediction

that portion with the heater disengaged. For both the heater and abrasion shield, the numerical model underpredicts the magnitude of the heat dissipation in the test case for the latter portion of the cycle when the heater is disengaged. Since the numerical simulation is clearly modelling the transients accurately for the "warm up" and the first portion of "cool down", there is obviously some physical phenomenon occurring in the test case that is not properly accounted for in the numerical simulation of the latter portion of the "cool down". One obvious possibility is the loss of ice (either through natural shedding or with an "assist" by the initial warming) at this location. With less ice at this location (ice is an excellent insulator) the heat would have dissipated much more quickly in the test case. A second possibility, and one that is much more likely, is that the thermal properties of the materials in the test section are changing with temperature. The numerical code can handle this condition with only slight modification, provided that the material properties are known as a function of temperature. Whether or not material properties for the insulation, epoxy glue, and film adhesive vary as a function of temperature is not currently known.

A test case of the numerical code was run to determine the extent of any geometric effects on temperature transients that current one-and two-dimensional codes cannot model. The same thickness and materials were used as in the comparison above, but this time the heater density was 16 watts per square inch. In this simulation, in which all of the heaters were fired, a case is provided where effects due to geometry only could be investigated. Table 5-2 displays the results of selected

Time	Interface Node Location	Node Number of Grid Along ARC (See Figure 5-3)					
(sec.)		61	63	65	67	69	71
0 sec.	Ice/Shield	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Heater Center	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Insul/Skin	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5 sec.	Ice/Shield	6.3409	6.4442	6.3408	6.4591	6.7132	6.6432
	Heater Center	83.5590	83.4471	83.3544	83.1967	83.0735	83.0294
	Insul/Skin	1.0139	1.0128	1.0136	1.0121	1.0102	1.0107
10 sec.	Ice/Shield	14.2206	14.4677	14.3608	14.6921	15.2466	15.1438
	Heater Center	110.5556	110.4994	110.3956	110.2703	110.1711	110.0925
	Insul/Skin	1.2309	1.2154	1.2216	1.1976	1.1711	1.1762
15 sec.	Ice/Shield	20.5982	20.9592	20.9178	21.4587	22.2432	22.1434
	Heater Center	124.5167	124.4782	124.4024	124.3121	124.2312	124.1223
	Insul/Skin	1.7470	1.7028	1.7078	1.6290	1.5519	1.5619
20 sec.	Ice/Shield	25.5802	26.0332	26.0632	26.7896	27.7632	27.6700
	Heater Center	135.9227	135.8977	135.8489	135.7917	135.7250	135.5794
	Insul/Skin	2.3981	2.3210	2.3143	2.1635	2.0265	2.0370
25 sec.	Ice/Shield	26.1890	26.6542	26.8154	27.6286	28.6001	28.5437
	Heater Center	95.4441	95.4979	95.5753	95.6736	95.6865	95.4023
	Insul/Skin	3.0307	2.9235	2.8993	2.6758	2.4824	2.4891

Table 5-2 Time-Temperature History of Selected Interface Nodes for a Numerical Simulation of a UH1H Rotor Section with a De-Icer Input of 16 W/in^2.

points on layer interfaces along the leading edge arc (where curvature is the highest). As can be seen, the outer layers experience a slight temperature drop, while the inner layers experience a slight temperature rise, for those regions near the leading edge. This is exactly what would be expected. As the thermal wave moves outward from the heater, there is more mass to absorb the generated energy. As the thermal wave moves inward from the heater, there is less mass, as a consequence of curvature, to absorb the generated energy. Consequently, one would expect a slight temperature drop outward from the heater, and a slight temperature rise inward from the heater, in contrast to results predicted using a one-dimensional model. For the particular case run, the temperature rise at the blade's stagnation point is 2.35 degrees Farenheit lower due to curvature/geometry. Out of a total temperature rise of 27.54 degrees expected for the one-dimensional case, this represents an error of 8.5% at this point due to geometric effects 25 seconds into the problem. Obviously, the magnitude of the error will be primarily a function of distance from the source of the thermal disturbance, the strength of the thermal disturbance, the degree of curvature, and the amount of real time into the problem, for this particular simulation.

CHAPTER 6 CONCLUSIONS

In this thesis a computer code has been developed that is capable of handling any number of irregularly shaped layers in a composite body undergoing a transient conduction process. The code is capable of handling thermal generation within any of the layers, as well as phase change in an outer layer. The computer code was verified for a wide variety of test problems.

The computer code was employed to simulate the actual transient thermal response of a UH1H rotor blade equipped with an electrothermal de-icer system. Good agreement over the majority of a heating cycle with test data was obtained. Additional runs with the code on the UH1H blade cross-section clearly showed the effect of geometry/curvature on the thermal transients.

It is believed that this code can be a very useful tool in airfoil ice protection design. It may also be used to determine where and under what conditions one-dimensional codes will yield satisfactory results (with obvious savings in time and money), and where a two-dimensional code is required.

Additional work is needed to apply more numerically sophisticated techniques into the spacial transform, the conduction, and phase change portions of the problem to accelerate convergence. The run made for the experimental comparison required 30 minutes of CPU time on a CRAY-XMP located at the NASA-Lewis Research Facility in Cleveland, Ohio.

Finally, additional IRT tunnel experimentation is needed in order to provide additional insight into the thermc-physics and other related phenomena effecting electrothermal de-icer performance.

REFERENCES

1. Itagaki, K., "Self-Shedding of Accreted Ice from High-Speed Rotors," ASME, 345 E. 47 St., New York, N.Y., 10017, Paper No. 83-WA/HT-68, 1983.

2. Stallabrass, J. R., "Thermal Aspects of De-Icer Design," Presented at the International Helicopter Icing Conference, May 23 to 26, 1972, In Ottawa, Canada.

3. Werner, J. B., "Ice Protection Investigation for Advanced Rotary-Wing Aircraft," Lockheed-California Report IR 25237-10, 1973.

4. Baliga, G., "Numerical Simmulation of One-Dimensional Heat Transfer in Composite Bodies with Phase Change," M. S. Thesis, University of Toledo, Toledo, Ohio, 1980.

5. Marano, J., "Numerical Simulation of an Electrothermal De-Icer Pad," M. S. Thesis, University of Toledo, Toledo, Ohio, 1982.

6. Gent, R. W., and Cansdale, J. T., "One-Dimensional Treatment of Thermal Transients in Electrically Deiced Helicopter Rotor Blades," Royal Aircraft Establishment, Procurement Executive, Ministry of Defense, Farnborough, Hunts, England, Technical Report 80159, 1980.

7. Chao, D. F., "Numerical Simulation of Two-Dimensional Heat Transfer in Composite Bodies with Application to De-Icing of Aircraft Components," Ph.D. Dissertation, University of Toledo, Toledo, Ohio, 1983.

8. DeWitt, K. J., Keith, T. G., Jr., Chao, D. F., and Masiulaniec, K. C., "Numerical Simulation of Electrothermal De-Icing Systems," AIAA, 1290 Avenue of the Americas, New York, N.Y., 10104, Paper No. AIAA-83-0114, 1983.

9. Leffel, K. L., "A Numerical and Experimental Investigation of Electrothermal Aircraft Deicing," NASA Lewis Research Center, Cleveland, Ohio, NASA Contractor Report 175024, January, 1986.

10. Becker, Eric B., Carey, Graham F. and Oden, J. Tinsley, "Finite Elements, an Introduction," Volume 1, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 07632, 1981.

11. Baker, A. J., "Finite Element Computational Fluid Mechanics," Hemisphere Publishing Corporation, McGraw-Hill Book Company, 1983.

12. Thompson, J. F., Thames, F. C. and Mastin, C. W., "Tomcat - A Code for Numerical Generation of Boundary Fitted Curvilinear Coordinate Systems on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, Vol. 24, 1977, pp. 272-302.

13. Thompson, J. F., Thames, F. C. and Mastin, C. W., "Boundary-Fitted Curvilinear Coordinate Systems for Solution of Partial Differential Equations on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies," NASA Contractor Report 2729, 1976.

14. Thompson, J. F., Thames, F. C. and Mastin, C. W., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate Systems for Fields Containing Any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, Vol. 15, 1974, pp. 299-319.

15. Thompson, J. F., "A Survey of Grid Generation Techniques in Computational Fluid Dynamics," AIAA, 1290 Avenue of the Americas, New York, N.Y. 10104, Paper No. AIAA-83-0447, 1983.

16. McWhorter, J. C., III, and Sadd, H. M., "Numerical Anisotropic Heat Conduction Solutions Using Boundary-Fitted Coordinate Systems," ASME Journal of Heat Transfer, Vol. 102, May, 1980, pp. 308-311.

17. Goldman, A. and Kao, Y. C., "Numerical Solution to a Two-Dimensional Conduction Problem Using Rectangular and Cylindrical Body-Fitted Coordinate Systems," ASME Journal of Heat Transfer, Vol. 103, November, 1981, pp. 753-758.

18. Rieger, H., Projahn, U. and Beer, H., "Analysis of the Heat Transport Mechanisms During Melting Around a Horizontal Circular Cylinder," International Journal of Heat and Mass Transfer, Vol. 25, No. 1, 1982, pp. 137-147.

19. Masiulaniec, K. C., Keith, T. G., Jr., and DeWitt, K. J., "Finite Difference Solutions of Heat Conduction Problems in Multi-Layered Bodies with Complex Geometries, ASME, 345 E. 47th St., New York, N.Y. 10017, Paper No. 84-HT-58, 1984.

20. Uchikawa, S. and Takeda, R., "Use of Boundary-Fitted Coordinate Transformation for Unsteady Heat Conduction Problems in Multiconnected Regions with Arbitrarily Shaped Boundaries," ASME Journal of Heat Transfer, Vol. 107, pp. 494-498, 1985.

21. Eckert, E. R. G. and Drake, R. M., Jr., "Analysis of Heat and Mass Transfer," McGraw-Hill Book Company, New York, 1972.

22. El-Saden, M. R., "Heat Conduction in an Eccentrically Hollow, Infinitely Long Cylinder with Internal Heat Generation," ASME Journal of Heat Transfer, Vol. 83, November, 1961, pp. 510-512.

23. Jakob, M., "Heat Transfer," Vol. 1, John Wiley and Sons, 1962.

24. Jaeger, J. C., "Heat Conduction in Composite Circular Cylinders," Philosophical Magazine, Ser. 7, Vol. 32, No. 213, October, 1941, pp. 324-335.

25. Berger, Carl L., "Transient Temperature Distributions in Two-Layer Hollow Cylinders," M. S. Thesis, The University of Toledo, Toledo, Ohio, 1986.

26. Carslaw, H. S. and Jaeger, J. C., "Conduction of Heat in Solids," Oxford University Press, London, England, 1959.

27. Lin, Sui, "One-Dimensional Freezing or Melting Process in a Body With Variable Cross-Sectional Area," International Journal of Heat and Mass Transfer, Vol. 14, 1971, pp. 153-156.

28. Cho, S. H. and Sunderland, J. E., "Phase Change Problems With Temperature-Dependent Thermal Conductivity," ASME Journal of Heat Transfer, Vol. 96, May, 1974, pp. 214-217.

29. Rubinsky, Boris and Shitzer, Abraham, "Analytic Solutions to the Heat Equation Involving a Moving Boundary with Applications to the Change of Phase Problem (The Inverse Stefan Problem)," ASME Journal of Heat Transfer, Vol. 100, May 1978, pp. 300-304.

30. Gutman, L. N., "On the Problem of Heat Transfer in a Phase-Change Slab Initially Not at the Critical Temperature," ASME Journal of Heat Transfer, Vol. 109, February, 1987, pp. 5-9.

31. Ku, J. Y. and Chan, S. H., "A Systematic Approach to the Exact Solutions of Some Phase-Change Problems," ASME, 345 E. 47 St., New York, N.Y., 10017, Paper No., 84-HT-1, 1984.

32. Zhang, G. P., Weinbaum, S. and Jiji, L. M., "An Approximate Three-Dimensional Solution for Melting or Freezing Around a Buried Pipe Beneath a Free Surface," ASME Journal of Heat Transfer, Vol. 108, November, 1986, pp. 900-906.

33. Goodman, T. R., "The Heat-Balance Integral and Its Application to Problems Involving a Change of Phase," Transactions of the ASME, Vol. 80, February, 1958, pp. 335-342.

34. Wang, H. P. and Perry, E. M., "Integral Solutions for Casting Processes with Initial Superheat and Natural Boundary Conditions," ASME, 345 E. 47th St., New York, N.Y., Paper No. 86-WA/HT-96, 1986.

35. Lazaridis, Anastas, "A Numerical Solution of the Multi-Dimensional Solidification (or Melting) Problem," International Journal of Heat and Mass Transfer, Vol. 13, Permagon Press, 1970, pp. 1459-1477.

36. O'Neill, K., "Fixed Mesh Finite Element Solution for Cartesian Two-Dimensional Phase Change," ASME Journal of Energy Resources Technology, Vol. 105, December 1983, pp. 436-441.

37. Yoo, Jaisuk and Rubinsky, Boris, "Numerical Computation Using Finite Elements for the Moving Surface in Heat Transfer Problems with Phase Transformation," Numerical Heat Transfer, Vol. 6, 1983, pp. 209-222.

38. Prusa, J. and Yao, L. S., "Melting Around a Horizontal Cylinder: Part 1 - Perturbation and Numerical Solutions for Constant Heat Flux Boundary Condition," ASME Journal of Heat Transfer, Vol. 106, May, 1984, pp. 376-384.

39. Duda, J. L., Malone, Michael F., Notter, Robert H. and Vrentas, J. S., "Analysis of Two-Dimensional Diffusion-Controller Moving Boundary Problems," International Journal of Heat and Mass Transfer, Vol. 18, 1975, pp. 901-909.

40. Lynch, Daniel R., "Unified Approach to Simulation on Deforming Elements with Application to Phase Change Problems," Journal of Computational Physics, Vol. 47, 1982, pp. 387-411.

41. Kikuchi, Noboru and Ichikawa, Yasuaki, "Numerical Methods for a Two-Phase Stefan Problem by Variational Inequalities," International Journal for Numerical Methods in Engineering, Vol. 14, 1979, pp. 1221-1239.

42. Blanchard, D. and Fremond, M., "The Stefan Problem: Computing Without the Free Boundary," International Journal for Numerical Methods in Engineering, Vol. 20, 1984, pp. 747-771.

43. Crank, J. and Gupta, Radhey S., "Isotherm Migration Method in Two-Dimensions," International Journal of Heat and Mass Transfer, Vol. 18, 1975, pp. 1101-1106.

44. Saitoh, T., "Numerical Method fo Multi-Dimensional Freezing Problems in Arbitrary Domains," Vol. 100, May 1978, pp. 294-299.

45. Ozisik, M. Necati, "A Note on the General Formulation of Phase Change Problems as Heat Conduction Problems with a Moving Heat Source," ASME Journal of Heat Transfer, Vol. 100, May 1978, pp. 370-371.

46. Bonacina, C., Comini, G., Fasano, A. and Primicerio, M., "Numerical Solution of Phase-Change Problems," International Journal of Heat and Mass Transfer, Vol. 16, 1973, pp. 1825-1832.

47. Comini, G., Del Guidice, S., Lewis, R. W. and Zienkiewicz, O. C., "Finite Element Solution of Non-Linear Heat Conduction Problems with Special Reference to Phase Change," International Journal for Numerical Methods in Engineering, Vol. 8, 1974, pp. 613-624.

48. Morgan, K., Lewis, R. W. and Zienkiewicz, O. C., "An Improved Algorithm for Heat Conduction Problems with Phase Change," International Journal for Numerical Methods in Engineering, Vol. 12, 1978, pp. 1191-1195.

49. Hsiao, J. S., "An Efficient Algorithm for Finite-Difference Analysis of Heat Transfer with Melting and Solidification," Numerical Heat Transfer, Vol. 8, 1985, pp. 653-666.

50. Uchikawa, S. and Takeda, R., "Use of Boundary-Fitted Coordinate Transformation for Unsteady Heat Conduction Problems in Multiconnected Regions with Arbitrarily Shaped Boundaries," ASME Journal of Heat Transfer, Vol. 107, pp. 494-498, 1985.

51. Shamsundar, N. and Sparrow, E. M., "Analysis of Multi-Dimensional Conduction Phase Change Via the Enthalpy Model," ASME Journal of Heat Transfer, Vol. 97, August, 1975, pp. 333-340.

52. Crowley, A. B., "Numerical Solution of Stefan Problems," International Journal of Heat and Mass Transfer, Vol. 21, 1978, pp. 215-218.

53. Voller, V. and Cross, M., "Accurate Solutions of Moving Boundary Problems Using the Enthalpy Method," International Journal of Heat and Mass Transfer, Vol. 24, 1981, pp. 545-556.

54. Schneider, G. E. and Raw, M. J., "An Implicit Solution Procedure for Finite Difference Modeling of the Stefan Problem," AIAA Journal, Vol. 22, No. 11, November, 1984, pp. 1685-1690.

55. Voller, V. R., "Implicit Finite-Difference Solutions of the Enthalpy Formulation of Stefan Problems," IMA Journal of Numerical Analysis," Vol. 5, 1985, pp.201-214.

56. Tacke, Karl-Herman, "Discretization of the Explicit Enthalpy Method for Planar Phase Change," International Journal for Numerical Methods in Engineering, Vol. 21, 1985, pp. 543-554.

57. Schneider, G. E., "Computation of Heat Transfer with Solid/Liquid Phase Change Including Free Convection," American Institute of Aeronautics and Astronautics, 1633 Broadway, New York, NY, 10019, Paper No. AIAA-85-0404, 1985.

58. Boucheron, Edward A. and Smith, Richard N., "Application of the Enthalpy Method to Multi-Dimensional Solidification Problems with Convection," ASME, 345 E. 47th St., New York, NY, 10017, Paper No. 86-WA/HT-42, 1986.

59. Lunardini, Virgil J., Heat Transfer in Cold Climates, Van Nostrand Reinhold, New York, NY, 1981, pp. 353-370.

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

61. Abbott, Ira H., and Von Doenhoff, Albert E., Theory of Wing Sections, Dover Publications, Inc., New York, NY, 1959.

APPENDIX A

Computer Subroutine for Spacial Mapping

```
      *....*...1.........2.........3.........4.........5.........6.........7.
      C
      C
      C
 1          SUBROUTINE SPACE (IMAX,JMAX,TX,TY,CA,CB,CG,CJ)
      C     IMPLICIT REAL*8(A-H,O-Z)
 2          DIMENSION TX(70,52),TY(70,52),CA(70,52),CB(70,52),CG(70,52),
           1CJ(70,52)
 3          REAL MAXER
      C
      C
      C    ******************************************************************
      C    * SET UP LINEAR DISTRIBUTION ON X MAP                            *
      C    ******************************************************************
      C
      C
 4          DO 5 I=1,IMAX
 5          XDEL=(TX(I,JMAX)-TX(I,1))/(JMAX-1)
 6          KJMAX=JMAX-1
 7          DO 6 J=2,KJMAX
 8        6 TX(I,J)=TX(I,1)+(J-1)*XDEL
 9        5 CONTINUE
      C
      C
      C    ******************************************************************
      C    * SET UP LINEAR DISTRIBUTION ON Y MAP                            *
      C    ******************************************************************
      C
      C
10          DO 15 I=1,IMAX
11          YDEL=(TY(I,JMAX)-TY(I,1))/(JMAX-1)
12          DO 16 J=2,KJMAX
13       16 TY(I,J)=TY(I,1)+(J-1)*YDEL
14       15 CONTINUE
      C
```

```
      C
      C    ********************************************************************
      C    * COMPUTATION THROUGH FIELD NOT INCLUDING RE-ENTRANT                *
      C    *       BOUNDARY                                                   *
      C    ********************************************************************
      C
      C
   15       50 MAXER=0.0
   16          KIMAX=IMAX-1
   17          DO 20 I=2,KIMAX
   18          DO 21 J=2,KJMAX
   19          XE=(TX(I+1,J)-TX(I-1,J))/2.0
   20          XN=(TX(I,J+1)-TX(I,J-1))/2.0
   21          YE=(TY(I+1,J)-TY(I-1,J))/2.0
   22          YN=(TY(I,J+1)-TY(I,J-1))/2.0
   23          XEN=(TX(I+1,J+1)-TX(I+1,J-1)+TX(I-1,J-1)-TX(I-1,J+1))/4.0
   24          YEN=(TY(I+1,J+1)-TY(I+1,J-1)+TY(I-1,J-1)-TY(I-1,J+1))/4.0
   25          RALPHA=XN*XN+YN*YN
   26          RBETA=YN*YE+XN*XE
3.0 (MAY 1983)            VS FORTRAN           DATE: JAN 15, 1987    TIME: 17:31
      *....*...1.........2.........3.........4.........5.........6.........7.
   27          RGAMMA=XE*XE+YE*YE
   28          RJACOB=XE*YN-XN*YE
   29          XITER=((RALPHA*(TX(I+1,J)+TX(I-1,J)))+(RGAMMA*(TX(I,J+1)+TX(I,J-
               11)))-2.0*RBETA*XEN)/(2.0*(RALPHA+RGAMMA))
   30          YITER=((RALPHA*(TY(I+1,J)+TY(I-1,J)))+(RGAMMA*(TY(I,J+1)+TY(I,J-
               11)))-2.0*RBETA*YEN)/(2.0*(RALPHA+RGAMMA))
      C        XER=ABS((TX(I,J)-XITER)/(XITER+0.00000001))
   31          XER=ABS(TX(I,J)-XITER)
   32          IF(XER.GT.MAXER)MAXER=XER
      C        YER=ABS((TY(I,J)-YITER)/(YITER+0.00000001))
   34          YER=ABS(TY(I,J)-YITER)
   35          IF(YER.GT.MAXER)MAXER=YER
   37          TX(I,J)=XITER
   38          TY(I,J)=YITER
   39          CA(I,J)=RALPHA
   40          CB(I,J)=RBETA
```

```
41          CG(I,J)=RGAMMA
42       21 CJ(I,J)=RJACOB
43       20 CONTINUE
      C
      C
      C    ********************************************************************
      C    * COMPUTATION ALONG RE-ENTRANT BOUNDARY                            *
      C    ********************************************************************
      C
      C
44          DO 30 J=2,KJMAX
45          XE=(TX(2,J)-TX(IMAX-1,J))/2.0
46          XN=(TX(IMAX,J+1)-TX(IMAX,J-1))/2.0
47          YE=(TY(2,J)-TY(IMAX-1,J))/2.0
48          YN=(TY(IMAX,J+1)-TY(IMAX,J-1))/2.0
49          XEN=(TX(2,J+1)-TX(2,J-1)+TX(IMAX-1,J-1)-TX(IMAX-1,J+1))/4.0
50          YEN=(TY(2,J+1)-TY(2,J-1)+TY(IMAX-1,J-1)-TY(IMAX-1,J+1))/4.0
51          RALPHA=XN*XN+YN*YN
52          RBETA=YN*YE+XN*XE
53          RGAMMA=XE*XE+YE*YE
54          RJACOB=XE*YN-XN*YE
55          XITER=((RALPHA*(TX(2,J)+TX(IMAX-1,J)))+(RGAMMA*(TX(IMAX,J+1)+
           1TX(IMAX,J-1)))-2.0*RBETA*XEN)/(2.0*(RALPHA+RGAMMA))
56          YITER=((RALPHA*(TY(2,J)+TY(IMAX-1,J)))+(RGAMMA*(TY(IMAX,J+1)+
           1TY(IMAX,J-1)))-2.0*RBETA*YEN)/(2.0*(RALPHA+RGAMMA))
      C     XER=ABS((TX(IMAX,J)-XITER)/(XITER+0.00000001))
57          XER=ABS(TX(IMAX,J)-XITER)
58          IF(XER.GT.MAXER)MAXER=XER
      C     YER=ABS((TY(IMAX,J)-YITER)/(YITER+0.00000001))
60          YER=ABS(TY(IMAX,J)-YITER)
61          IF(YER.GT.MAXER)MAXER=YER
63          TX(IMAX,J)=XITER
64          TY(IMAX,J)=YITER
65          TX(1,J)=XITER
66          TY(1,J)=YITER
67          CA(1,J)=RALPHA
68          CA(IMAX,J)=RALPHA
```

```
  69          CB(1,J)=RBETA
  70          CB(IMAX,J)=RBETA
  71          CG(1,J)=RGAMMA
3.0 (MAY 1983)            VS FORTRAN           DATE: JAN 15, 1987    TIME: 17:31
      *....*...1.........2.........3.........4.........5.........6.........7.
  72          CG(IMAX,J)=RGAMMA
  73          CJ(1,J)=RJACOB
  74       30 CJ(IMAX,J)=RJACOB
      C
      C
      C    ********************************************************************
      C    * CHECK FOR CONVERGENCE REQUIREMENT                                 *
      C    ********************************************************************
      C
      C
  75          IF(MAXER.GT.0.00001)GOTO 50
      C
      C
      C    ********************************************************************
      C    * SPACE COEFFICIENTS ALONG BOUNDARIES                               *
      C    ********************************************************************
      C
      C
  76          DO 60 K=2,KIMAX
  77          XN=TX(K,2)-TX(K,1)
  78          XE=(TX(K+1,1)-TX(K-1,1))/2.0
  79          YN=TY(K,2)-TY(K,1)
  80          YE=(TY(K+1,1)-TY(K-1,1))/2.0
  81          CA(K,1)=XN*XN+YN*YN
  82          CB(K,1)=YN*YE+XN*XE
  83          CG(K,1)=XE*XE+YE*YE
  84          CJ(K,1)=XE*YN-XN*YE
  85          XN=TX(K,JMAX)-TX(K,JMAX-1)
  86          XE=(TX(K+1,JMAX)-TX(K-1,JMAX))/2.0
  87          YN=TY(K,JMAX)-TY(K,JMAX-1)
  88          YE=(TY(K+1,JMAX)-TY(K-1,JMAX))/2.0
```

```
 89          CA(K,JMAX)=XN*XN+YN*YN
 90          CB(K,JMAX)=YN*YE+XN*XE
 91          CG(K,JMAX)=XE*XE+YE*YE
 92       60 CJ(K,JMAX)=XE*YN-XN*YE
 93          XN=TX(1,JMAX)-TX(1,JMAX-1)
 94          XE=(TX(2,JMAX)-TX(IMAX-1,JMAX))/2.0
 95          YN=TY(1,JMAX)-TY(1,JMAX-1)
 96          YE=(TY(2,JMAX)-TY(IMAX-1,JMAX))/2.0
 97          CA(1,JMAX)=XN*XN+YN*YN
 98          CB(1,JMAX)=YN*YE+XN*XE
 99          CG(1,JMAX)=XE*XE+YE*YE
100          CJ(1,JMAX)=XE*YN-XN*YE
101          CA(IMAX,JMAX)=CA(1,JMAX)
102          CB(IMAX,JMAX)=CB(1,JMAX)
103          CG(IMAX,JMAX)=CG(1,JMAX)
104          CJ(IMAX,JMAX)=CJ(1,JMAX)
105          XN=TX(1,2)-TX(1,1)
106          XE=(TX(2,1)-TX(IMAX-1,1))/2.0
107          YN=TY(1,2)-TY(1,1)
108          YE=(TY(2,1)-TY(IMAX-1,1))/2.0
109          CA(1,1)=XN*XN+YN*YN
110          CB(1,1)=YN*YE+XN*XE
111          CG(1,1)=XE*XE+YE*YE
112          CJ(1,1)=XE*YN-XN*YE
113          CA(IMAX,1)=CA(1,1)
3.0 (MAY 1983)            VS FORTRAN          DATE: JAN 15, 1987    TIME: 17:31
      *....*...1.........2.........3.........4.........5.........6.........7.
114          CB(IMAX,1)=CB(1,1)
115          CG(IMAX,1)=CG(1,1)
116          CJ(IMAX,1)=CJ(1,1)
117          RETURN
118          END
```

APPENDIX B

Computer Output of Spacial Transform Coefficients for a Coarse Mesh

```
    11          11        0.3278E-04        0.2000E+00        0.0000E+00
    11          11        0.2014E-03        0.4000E+00        0.0000E+00
    41           7        0.5093E-05        0.3480E-03        0.0000E+00
    41           4        0.1759E-03        0.5520E-02        0.5330E+01
    41          12        0.5093E-05        0.3480E-03        0.0000E+00
    41          17        0.1542E-02        0.6600E-01        0.0000E+00
  0.1000E+07        0.1929E-05        0.4720E+03        0.4720E+03        0.1000E+00
STARTING TEMPERATURE =   472.00
```

SPACIAL TRANSFORM SOLUTION FOR ZONE

X MAPPING

9.0000	7.2811	2.7811	-2.7812	-7.2812	-9.0000	-7.2811	-2.7811	2.7812	7.2812
9.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
7.0175	5.6773	2.1685	-2.1686	-5.6773	-7.0175	-5.6773	-2.1685	2.1685	5.6773
7.0175	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
5.5137	4.4607	1.7038	-1.7039	-4.4607	-5.5137	-4.4607	-1.7038	1.7038	4.4607
5.5137	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
4.3631	3.5298	1.3483	-1.3483	-3.5299	-4.3631	-3.5298	-1.3483	1.3483	3.5298
4.3631	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
3.4757	2.8119	1.0740	-1.0741	-2.8119	-3.4757	-2.8119	-1.0740	1.0740	2.8119
3.4757	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									

2.7860	2.2539	0.8609	-0.8609	-2.2539	-2.7860	-2.2539	-0.8609	0.8609	2.2539
2.7860	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
2.2463	1.8173	0.6941	-0.6941	-1.8173	-2.2463	-1.8173	-0.6941	0.6941	1.8173
2.2463	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
1.8211	1.4733	0.5627	-0.5628	-1.4733	-1.8211	-1.4733	-0.5627	0.5628	1.4733
1.8211	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
1.4841	1.2006	0.4586	-0.4586	-1.2006	-1.4841	-1.2006	-0.4586	0.4586	1.2006
1.4841	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
1.2154	0.9833	0.3756	-0.3756	-0.9833	-1.2154	-0.9833	-0.3756	0.3756	0.9833
1.2154	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
1.0000	0.8090	0.3090	-0.3090	-0.8090	-1.0000	-0.8090	-0.3090	0.3090	0.8090
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									

Y MAPPING

-0.0000	-5.2901	-8.5595	-8.5595	-5.2900	0.0000	5.2901	8.5595	8.5595	5.2901
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
-0.0000	-4.1248	-6.6741	-6.6740	-4.1248	0.0000	4.1248	6.6740	6.6740	4.1248
-0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
-0.0000	-3.2409	-5.2439	-5.2439	-3.2409	0.0000	3.2409	5.2439	5.2439	3.2409
-0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
-0.0000	-2.5646	-4.1496	-4.1496	-2.5646	0.0000	2.5646	4.1496	4.1496	2.5646
-0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
-0.0000	-2.0430	-3.3055	-3.3055	-2.0429	0.0000	2.0429	3.3055	3.3055	2.0429
-0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									

-0.0000	-1.6376	-2.6496	-2.6496	-1.6375	0.0000	1.6376	2.6496	2.6496	1.6375
-0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
-0.0000	-1.3203	-2.1363	-2.1363	-1.3203	0.0000	1.3203	2.1363	2.1363	1.3203
-0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
-0.0000	-1.0704	-1.7320	-1.7319	-1.0704	0.0000	1.0704	1.7320	1.7320	1.0704
-0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
-0.0000	-0.8723	-1.4114	-1.4114	-0.8723	0.0000	0.8723	1.4114	1.4114	0.8723
-0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
-0.0000	-0.7144	-1.1559	-1.1559	-0.7144	0.0000	0.7144	1.1559	1.1559	0.7144
-0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
-0.0000	-0.5878	-0.9511	-0.9511	-0.5878	0.0000	0.5878	0.9511	0.9511	0.5878
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									

ALPHA VALUES FOR THE LAYER ARE :

3.930280	3.930252	3.930280	3.930289	3.930271
3.038534	3.038523	3.038528	3.038530	3.038529
1.761444	1.761443	1.761440	1.761440	1.761452
1.038439	1.038438	1.038440	1.038438	1.038441
0.621851	0.621850	0.621849	0.621849	0.621850
0.377859	0.377859	0.377859	0.377858	0.377860
0.232761	0.232761	0.232760	0.232761	0.232762
0.145231	0.145231	0.145231	0.145231	0.145231
0.091716	0.091716	0.091716	0.091716	0.091716
0.058583	0.058583	0.058583	0.058583	0.058583
0.046394	0.046395	0.046394	0.046395	0.046394

BETA VALUES FOR THE LAYER ARE :

0.000172	0.000035	-0.000020	-0.000031	-0.000030
-0.000063	0.000028	-0.000018	-0.000010	-0.000018
-0.000022	0.000005	-0.000005	-0.000007	-0.000010
-0.000006	0.000000	0.000001	-0.000007	-0.000002
0.000000	-0.000000	0.000003	-0.000002	0.000000
0.000001	-0.000000	0.000002	0.000001	0.000001
0.000002	0.000001	0.000001	0.000001	0.000002
0.000002	0.000001	0.000001	0.000001	0.000002
0.000002	0.000000	0.000001	0.000001	0.000002
0.000001	0.000000	0.000001	0.000001	0.000001
-0.000003	-0.000000	0.000000	0.000001	0.000000

GAMMA VALUES FOR THE LAYER ARE :

27.985016	27.984741	27.984695	27.984726	27.984741
17.013947	17.013962	17.013870	17.013855	17.013855
10.503365	10.503394	10.503348	10.503335	10.503335
6.577075	6.577109	6.577087	6.577077	6.577081
4.173593	4.173619	4.173608	4.173602	4.173599
2.681605	2.681616	2.681611	2.681610	2.681606
1.743235	1.743243	1.743240	1.743237	1.743236
1.145768	1.145776	1.145772	1.145768	1.145767
0.760941	0.760944	0.760941	0.760939	0.760939
0.510357	0.510356	0.510354	0.510353	0.510353
0.345494	0.345491	0.345490	0.345491	0.345491

JACOBIAN VALUES FOR THE LAYER ARE :

10.487561	10.487476	10.487505	10.487524	10.487501
7.190097	7.190085	7.190073	7.190073	7.190069
4.301290	4.301294	4.301282	4.301279	4.301294
2.613405	2.613412	2.613409	2.613404	2.613409
1.611010	1.611012	1.611010	1.611009	1.611009
1.006612	1.006615	1.006614	1.006613	1.006613
0.636991	0.636993	0.636991	0.636992	0.636993
0.407923	0.407924	0.407923	0.407923	0.407923
0.264179	0.264179	0.264179	0.264179	0.264178
0.172911	0.172911	0.172911	0.172911	0.172910
0.126605	0.126605	0.126605	0.126605	0.126604

APPENDIX C

Computer Output of Spacial Transform Coefficients for a Fine Mesh

```
                    41          41        0.3278E-04        0.2000E+00        0.0000E
+00
                    41          41        0.2014E-03        0.4000E+00        0.0000E
+00
                    41           7        0.5093E-05        0.3480E-03        0.0000E
+00
                    41           4        0.1759E-03        0.5520E-02        0.5330E
+01
                    41          12        0.5093E-05        0.3480E-03        0.0000E
+00
                    41          17        0.1542E-02        0.6600E-01        0.0000E
+00
                  0.1000E+07        0.1929E-05        0.4720E+03        0.4720E+03
     0.1000E+00
                 STARTING TEMPERATURE =  472.00
SPACIAL TRANSFORM SOLUTION FOR ZONE
X MAPPING
  9.0000  8.8892  8.5595  8.0190  7.2811  6.3639  5.2900  4.0859  2.7811  1.4079
 -0.0000 -1.4079 -2.7812 -4.0859 -5.2901 -6.3640 -7.2812 -8.0191 -8.5595 -8.8892
 -9.0000 -8.8892 -8.5595 -8.0191 -7.2811 -6.3640 -5.2901 -4.0859 -2.7812 -1.4079
 -0.0000  1.4079  2.7811  4.0859  5.2901  6.3640  7.2811  8.0190  8.5595  8.8892
  9.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
  0.0000
  8.5155  8.4107  8.0988  7.5874  6.8892  6.0214  5.0053  3.8660  2.6314  1.3321
 -0.0000 -1.3321 -2.6315 -3.8660 -5.0053 -6.0214 -6.8892 -7.5874 -8.0988 -8.4107
 -8.5155 -8.4107 -8.0988 -7.5874 -6.8892 -6.0214 -5.0053 -3.8660 -2.6315 -1.3321
 -0.0000  1.3321  2.6314  3.8660  5.0053  6.0214  6.8892  7.5874  8.0988  8.4107
  8.5155  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
  0.0000
  8.0573  7.9581  7.6630  7.1791  6.5185  5.6974  4.7360  3.6579  2.4898  1.2604
 -0.0000 -1.2605 -2.4899 -3.6580 -4.7360 -5.6974 -6.5185 -7.1792 -7.6630 -7.9581
 -8.0573 -7.9582 -7.6630 -7.1792 -6.5185 -5.6974 -4.7360 -3.6580 -2.4899 -1.2605
 -0.0000  1.2604  2.4899  3.6580  4.7360  5.6974  6.5185  7.1791  7.6630  7.9581
  8.0573  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
  0.0000
```

7.6240	7.5301	7.2508	6.7930	6.1679	5.3910	4.4812	3.4612	2.3559	1.1926
-0.0000	-1.1927	-2.3559	-3.4612	-4.4813	-5.3910	-6.1679	-6.7930	-7.2508	-7.5301
-7.6240	-7.5301	-7.2508	-6.7930	-6.1679	-5.3910	-4.4813	-3.4612	-2.3559	-1.1927
-0.0000	1.1926	2.3559	3.4612	4.4813	5.3910	6.1679	6.7930	7.2508	7.5301
7.6240	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
7.2141	7.1252	6.8610	6.4278	5.8363	5.1011	4.2403	3.2751	2.2292	1.1285
-0.0000	-1.1285	-2.2293	-3.2751	-4.2403	-5.1011	-5.8363	-6.4278	-6.8610	-7.1252
-7.2141	-7.1253	-6.8610	-6.4278	-5.8363	-5.1011	-4.2403	-3.2751	-2.2293	-1.1285
-0.0000	1.1285	2.2293	3.2751	4.2403	5.1011	5.8363	6.4278	6.8610	7.1252
7.2141	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
6.8263	6.7423	6.4922	6.0823	5.5226	4.8270	4.0124	3.0991	2.1094	1.0679
-0.0000	-1.0679	-2.1095	-3.0991	-4.0124	-4.8270	-5.5226	-6.0823	-6.4922	-6.7423
-6.8263	-6.7423	-6.4922	-6.0823	-5.5226	-4.8270	-4.0124	-3.0991	-2.1095	-1.0679
-0.0000	1.0679	2.1094	3.0991	4.0124	4.8270	5.5226	6.0823	6.4922	6.7423
6.8263	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
6.4596	6.3801	6.1434	5.7555	5.2259	4.5676	3.7968	2.9326	1.9961	1.0105
-0.0000	-1.0105	-1.9961	-2.9326	-3.7969	-4.5676	-5.2259	-5.7556	-6.1434	-6.3801
-6.4596	-6.3801	-6.1435	-5.7556	-5.2259	-4.5676	-3.7969	-2.9326	-1.9961	-1.0105
-0.0000	1.0105	1.9961	2.9326	3.7968	4.5676	5.2259	5.7555	6.1434	6.3801
6.4596	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
6.1127	6.0374	5.8135	5.4464	4.9453	4.3223	3.5929	2.7751	1.8889	0.9562
-0.0000	-0.9562	-1.8889	-2.7751	-3.5929	-4.3223	-4.9453	-5.4464	-5.8135	-6.0374
-6.1127	-6.0374	-5.8135	-5.4465	-4.9453	-4.3223	-3.5929	-2.7751	-1.8889	-0.9562
-0.0000	0.9562	1.8889	2.7751	3.5929	4.3223	4.9453	5.4464	5.8135	6.0374
6.1127	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
5.7845	5.7133	5.5014	5.1540	4.6798	4.0903	3.4000	2.6261	1.7875	0.9049
-0.0000	-0.9049	-1.7875	-2.6261	-3.4000	-4.0903	-4.6798	-5.1540	-5.5014	-5.7133
-5.7845	-5.7133	-5.5014	-5.1540	-4.6798	-4.0903	-3.4000	-2.6261	-1.7875	-0.9049
-0.0000	0.9049	1.7875	2.6261	3.4000	4.0903	4.6798	5.1540	5.5014	5.7133
5.7845	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									

5.4741	5.4067	5.2062	4.8774	4.4286	3.8707	3.2176	2.4852	1.6916	0.8563
-0.0000	-0.8563	-1.6916	-2.4852	-3.2176	-3.8707	-4.4286	-4.8774	-5.2061	-5.4067
-5.4741	-5.4067	-5.2062	-4.8774	-4.4286	-3.8708	-3.2176	-2.4852	-1.6916	-0.8563
-0.0000	0.8563	1.6915	2.4851	3.2176	3.8707	4.4286	4.8774	5.2061	5.4067
5.4741	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
5.1804	5.1166	4.9269	4.6158	4.1910	3.6631	3.0449	2.3518	1.6008	0.8104
-0.0000	-0.8104	-1.6008	-2.3518	-3.0449	-3.6631	-4.1910	-4.6158	-4.9268	-5.1166
-5.1804	-5.1166	-4.9269	-4.6158	-4.1910	-3.6631	-3.0450	-2.3518	-1.6008	-0.8104
-0.0000	0.8104	1.6008	2.3518	3.0449	3.6631	4.1910	4.6158	4.9268	5.1166
5.1804	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
4.9026	4.8422	4.6626	4.3682	3.9663	3.4666	2.8816	2.2257	1.5149	0.7669
-0.0000	-0.7669	-1.5150	-2.2257	-2.8816	-3.4666	-3.9663	-4.3682	-4.6626	-4.8422
-4.9026	-4.8422	-4.6626	-4.3682	-3.9663	-3.4666	-2.8816	-2.2257	-1.5150	-0.7669
-0.0000	0.7669	1.5149	2.2257	2.8816	3.4666	3.9662	4.3682	4.6626	4.8422
4.9026	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
4.6397	4.5826	4.4127	4.1340	3.7536	3.2808	2.7271	2.1064	1.4337	0.7258
-0.0000	-0.7258	-1.4337	-2.1064	-2.7271	-3.2808	-3.7536	-4.1340	-4.4126	-4.5826
-4.6397	-4.5826	-4.4127	-4.1340	-3.7536	-3.2808	-2.7272	-2.1064	-1.4337	-0.7258
-0.0000	0.7258	1.4337	2.1064	2.7271	3.2808	3.7536	4.1340	4.4126	4.5826
4.6397	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
4.3911	4.3370	4.1762	3.9125	3.5525	3.1049	2.5810	1.9935	1.3569	0.6869
0.0000	-0.6869	-1.3569	-1.9935	-2.5810	-3.1049	-3.5524	-3.9125	-4.1762	-4.3370
-4.3911	-4.3370	-4.1762	-3.9125	-3.5525	-3.1050	-2.5810	-1.9935	-1.3569	-0.6869
-0.0000	0.6869	1.3569	1.9935	2.5810	3.1049	3.5524	3.9125	4.1761	4.3370
4.3911	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
4.1558	4.1047	3.9524	3.7029	3.3621	2.9386	2.4427	1.8867	1.2842	0.6501
0.0000	-0.6501	-1.2842	-1.8867	-2.4427	-2.9386	-3.3621	-3.7029	-3.9524	-4.1047
-4.1558	-4.1047	-3.9524	-3.7029	-3.3621	-2.9386	-2.4427	-1.8867	-1.2842	-0.6501
-0.0000	0.6501	1.2842	1.8867	2.4427	2.9386	3.3621	3.7029	3.9524	4.1047
4.1558	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									

```
 3.9333  3.8848  3.7408  3.5046  3.1821  2.7812  2.3119  1.7856  1.2154  0.6153
 0.0000 -0.6153 -1.2154 -1.7856 -2.3119 -2.7812 -3.1821 -3.5046 -3.7407 -3.8848
-3.9333 -3.8848 -3.7408 -3.5046 -3.1821 -2.7812 -2.3119 -1.7857 -1.2154 -0.6153
-0.0000  0.6153  1.2154  1.7856  2.3119  2.7812  3.1821  3.5045  3.7407  3.8848
 3.9333  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
 3.7227  3.6769  3.5405  3.3169  3.0117  2.6323  2.1881  1.6901  1.1504  0.5824
 0.0000 -0.5824 -1.1504 -1.6901 -2.1881 -2.6323 -3.0117 -3.3169 -3.5405 -3.6769
-3.7227 -3.6769 -3.5405 -3.3169 -3.0117 -2.6323 -2.1881 -1.6901 -1.1504 -0.5824
-0.0000  0.5823  1.1504  1.6900  2.1881  2.6323  3.0117  3.3169  3.5405  3.6769
 3.7227  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
 3.5235  3.4801  3.3510  3.1394  2.8505  2.4915  2.0710  1.5996  1.0888  0.5512
 0.0000 -0.5512 -1.0888 -1.5996 -2.0710 -2.4915 -2.8505 -3.1394 -3.3510 -3.4801
-3.5235 -3.4801 -3.3510 -3.1394 -2.8506 -2.4915 -2.0710 -1.5996 -1.0888 -0.5512
-0.0000  0.5512  1.0888  1.5996  2.0710  2.4915  2.8505  3.1394  3.3510  3.4801
 3.5235  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
 3.3350  3.2939  3.1718  2.9715  2.6980  2.3582  1.9602  1.5140  1.0306  0.5217
 0.0000 -0.5217 -1.0306 -1.5140 -1.9602 -2.3582 -2.6980 -2.9715 -3.1717 -3.2939
-3.3350 -3.2939 -3.1718 -2.9715 -2.6981 -2.3582 -1.9602 -1.5140 -1.0306 -0.5217
-0.0000  0.5217  1.0305  1.5140  1.9602  2.3582  2.6980  2.9715  3.1717  3.2939
 3.3350  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
 3.1566  3.1178  3.0021  2.8126  2.5538  2.2321  1.8554  1.4331  0.9754  0.4938
 0.0000 -0.4938 -0.9754 -1.4331 -1.8554 -2.2321 -2.5538 -2.8126 -3.0021 -3.1178
-3.1566 -3.1178 -3.0021 -2.8126 -2.5538 -2.2321 -1.8554 -1.4331 -0.9755 -0.4938
-0.0000  0.4938  0.9754  1.4331  1.8554  2.2321  2.5538  2.8126  3.0021  3.1178
 3.1566  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
 2.9879  2.9511  2.8416  2.6622  2.4172  2.1128  1.7562  1.3565  0.9233  0.4674
 0.0000 -0.4674 -0.9233 -1.3565 -1.7562 -2.1127 -2.4172 -2.6622 -2.8416 -2.9511
-2.9879 -2.9511 -2.8416 -2.6622 -2.4173 -2.1128 -1.7562 -1.3565 -0.9233 -0.4674
-0.0000  0.4674  0.9233  1.3565  1.7562  2.1127  2.4172  2.6622  2.8416  2.9511
 2.9879  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

```
 2.8282  2.7934  2.6898  2.5200  2.2881  1.9999  1.6624  1.2840  0.8740  0.4424
 0.0000 -0.4424 -0.8740 -1.2840 -1.6624 -1.9998 -2.2881 -2.5200 -2.6898 -2.7934
-2.8282 -2.7934 -2.6898 -2.5200 -2.2881 -1.9999 -1.6624 -1.2840 -0.8740 -0.4424
-0.0000  0.4424  0.8740  1.2840  1.6624  1.9998  2.2881  2.5200  2.6898  2.7934
 2.8282  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 2.6771  2.6442  2.5461  2.3853  2.1658  1.8930  1.5736  1.2154  0.8273  0.4188
 0.0000 -0.4188 -0.8273 -1.2154 -1.5736 -1.8930 -2.1658 -2.3853 -2.5461 -2.6442
-2.6771 -2.6442 -2.5461 -2.3854 -2.1659 -1.8930 -1.5736 -1.2154 -0.8273 -0.4188
-0.0000  0.4188  0.8273  1.2154  1.5736  1.8930  2.1658  2.3853  2.5461  2.6442
 2.6771  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 2.5342  2.5030  2.4101  2.2580  2.0502  1.7919  1.4895  1.1505  0.7831  0.3964
-0.0000 -0.3964 -0.7831 -1.1505 -1.4895 -1.7919 -2.0502 -2.2580 -2.4101 -2.5030
-2.5342 -2.5030 -2.4101 -2.2580 -2.0502 -1.7919 -1.4896 -1.1505 -0.7831 -0.3964
-0.0000  0.3964  0.7831  1.1505  1.4895  1.7919  2.0502  2.2580  2.4101  2.5030
 2.5342  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 2.3989  2.3694  2.2815  2.1374  1.9407  1.6963  1.4100  1.0891  0.7413  0.3753
-0.0000 -0.3753 -0.7413 -1.0891 -1.4100 -1.6963 -1.9407 -2.1374 -2.2815 -2.3694
-2.3989 -2.3694 -2.2815 -2.1374 -1.9408 -1.6963 -1.4100 -1.0891 -0.7413 -0.3753
-0.0000  0.3753  0.7413  1.0891  1.4100  1.6963  1.9407  2.1374  2.2815  2.3694
 2.3989  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 2.2709  2.2429  2.1597  2.0234  1.8372  1.6058  1.3348  1.0310  0.7017  0.3552
-0.0000 -0.3552 -0.7017 -1.0310 -1.3348 -1.6058 -1.8372 -2.0234 -2.1597 -2.2429
-2.2709 -2.2429 -2.1597 -2.0234 -1.8372 -1.6058 -1.3348 -1.0310 -0.7017 -0.3552
-0.0000  0.3552  0.7017  1.0309  1.3348  1.6058  1.8372  2.0234  2.1597  2.2429
 2.2709  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 2.1497  2.1233  2.0445  1.9154  1.7392  1.5201  1.2636  0.9760  0.6643  0.3363
-0.0000 -0.3363 -0.6643 -0.9760 -1.2636 -1.5201 -1.7392 -1.9154 -2.0445 -2.1233
-2.1498 -2.1233 -2.0445 -1.9154 -1.7392 -1.5201 -1.2636 -0.9760 -0.6643 -0.3363
-0.0000  0.3363  0.6643  0.9760  1.2636  1.5201  1.7392  1.9154  2.0445  2.1233
 2.1497  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

```
 2.0351  2.0101  1.9355  1.8133  1.6464  1.4390  1.1962  0.9239  0.6289  0.3184
-0.0000 -0.3184 -0.6289 -0.9239 -1.1962 -1.4390 -1.6464 -1.8133 -1.9355 -2.0101
-2.0351 -2.0101 -1.9355 -1.8133 -1.6464 -1.4390 -1.1962 -0.9239 -0.6289 -0.3184
-0.0000  0.3184  0.6289  0.9239  1.1962  1.4390  1.6464  1.8133  1.9355  2.0101
 2.0351  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.9266  1.9029  1.8323  1.7166  1.5587  1.3623  1.1324  0.8747  0.5954  0.3014
-0.0000 -0.3014 -0.5954 -0.8747 -1.1324 -1.3623 -1.5587 -1.7166 -1.8323 -1.9029
-1.9266 -1.9029 -1.8323 -1.7166 -1.5587 -1.3623 -1.1324 -0.8747 -0.5954 -0.3014
-0.0000  0.3014  0.5954  0.8747  1.1324  1.3623  1.5587  1.7166  1.8323  1.9029
 1.9266  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.8240  1.8015  1.7347  1.6252  1.4756  1.2897  1.0721  0.8281  0.5636  0.2853
-0.0000 -0.2853 -0.5636 -0.8281 -1.0721 -1.2897 -1.4756 -1.6252 -1.7347 -1.8015
-1.8240 -1.8015 -1.7347 -1.6252 -1.4756 -1.2897 -1.0721 -0.8281 -0.5636 -0.2853
-0.0000  0.2853  0.5636  0.8280  1.0721  1.2897  1.4756  1.6252  1.7347  1.8015
 1.8240  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.7268  1.7055  1.6423  1.5386  1.3970  1.2210  1.0150  0.7839  0.5336  0.2701
-0.0000 -0.2701 -0.5336 -0.7839 -1.0150 -1.2210 -1.3970 -1.5386 -1.6423 -1.7055
-1.7268 -1.7055 -1.6423 -1.5386 -1.3970 -1.2210 -1.0150 -0.7839 -0.5336 -0.2701
-0.0000  0.2701  0.5336  0.7839  1.0150  1.2210  1.3970  1.5386  1.6423  1.7055
 1.7268  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.6348  1.6147  1.5548  1.4567  1.3226  1.1560  0.9609  0.7422  0.5052  0.2557
-0.0000 -0.2557 -0.5052 -0.7422 -0.9609 -1.1560 -1.3226 -1.4567 -1.5548 -1.6147
-1.6348 -1.6147 -1.5548 -1.4567 -1.3226 -1.1560 -0.9609 -0.7422 -0.5052 -0.2557
-0.0000  0.2557  0.5052  0.7422  0.9609  1.1560  1.3226  1.4566  1.5548  1.6147
 1.6348  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.5478  1.5288  1.4721  1.3791  1.2522  1.0945  0.9098  0.7027  0.4783  0.2421
-0.0000 -0.2421 -0.4783 -0.7027 -0.9098 -1.0945 -1.2522 -1.3791 -1.4721 -1.5288
-1.5478 -1.5288 -1.4721 -1.3791 -1.2522 -1.0945 -0.9098 -0.7027 -0.4783 -0.2421
-0.0000  0.2421  0.4783  0.7027  0.9098  1.0945  1.2522  1.3791  1.4721  1.5288
 1.5478  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

```
 1.4655  1.4474  1.3937  1.3057  1.1856  1.0362  0.8614  0.6653  0.4528  0.2292
-0.0000 -0.2293 -0.4529 -0.6653 -0.8614 -1.0362 -1.1856 -1.3057 -1.3937 -1.4474
-1.4655 -1.4474 -1.3937 -1.3057 -1.1856 -1.0362 -0.8614 -0.6653 -0.4529 -0.2293
-0.0000  0.2292  0.4528  0.6653  0.8614  1.0362  1.1856  1.3057  1.3937  1.4474
 1.4655  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.3875  1.3704  1.3196  1.2363  1.1225  0.9811  0.8156  0.6299  0.4288  0.2171
-0.0000 -0.2171 -0.4288 -0.6299 -0.8156 -0.9811 -1.1225 -1.2363 -1.3196 -1.3704
-1.3875 -1.3704 -1.3196 -1.2363 -1.1225 -0.9811 -0.8156 -0.6299 -0.4288 -0.2171
-0.0000  0.2171  0.4288  0.6299  0.8156  0.9811  1.1225  1.2363  1.3196  1.3704
 1.3875  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.3137  1.2976  1.2494  1.1705  1.0628  0.9289  0.7722  0.5964  0.4060  0.2055
-0.0000 -0.2055 -0.4060 -0.5964 -0.7722 -0.9289 -1.0628 -1.1705 -1.2494 -1.2976
-1.3137 -1.2976 -1.2494 -1.1705 -1.0628 -0.9289 -0.7722 -0.5964 -0.4060 -0.2055
-0.0000  0.2055  0.4060  0.5964  0.7722  0.9289  1.0628  1.1705  1.2494  1.2976
 1.3137  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.2439  1.2286  1.1830  1.1083  1.0063  0.8796  0.7311  0.5647  0.3844  0.1946
-0.0000 -0.1946 -0.3844 -0.5647 -0.7311 -0.8796 -1.0063 -1.1083 -1.1830 -1.2286
-1.2439 -1.2286 -1.1830 -1.1083 -1.0063 -0.8796 -0.7311 -0.5647 -0.3844 -0.1946
-0.0000  0.1946  0.3844  0.5647  0.7311  0.8796  1.0063  1.1083  1.1830  1.2286
 1.2439  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.1778  1.1633  1.1202  1.0494  0.9529  0.8328  0.6923  0.5347  0.3640  0.1842
-0.0000 -0.1843 -0.3640 -0.5347 -0.6923 -0.8328 -0.9529 -1.0494 -1.1202 -1.1633
-1.1778 -1.1633 -1.1202 -1.0494 -0.9529 -0.8328 -0.6923 -0.5347 -0.3640 -0.1843
-0.0000  0.1843  0.3640  0.5347  0.6923  0.8328  0.9529  1.0494  1.1202  1.1633
 1.1778  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 1.1153  1.1015  1.0607  0.9937  0.9023  0.7886  0.6555  0.5063  0.3446  0.1745
-0.0000 -0.1745 -0.3446 -0.5063 -0.6555 -0.7886 -0.9023 -0.9937 -1.0607 -1.1015
-1.1153 -1.1015 -1.0607 -0.9937 -0.9023 -0.7886 -0.6555 -0.5063 -0.3446 -0.1745
-0.0000  0.1745  0.3446  0.5063  0.6555  0.7886  0.9023  0.9937  1.0607  1.1015
 1.1153  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

1.0560	1.0430	1.0044	0.9409	0.8544	0.7467	0.6207	0.4794	0.3263	0.1652
-0.0000	-0.1652	-0.3263	-0.4794	-0.6207	-0.7467	-0.8544	-0.9409	-1.0044	-1.0430
-1.0560	-1.0430	-1.0044	-0.9409	-0.8544	-0.7467	-0.6207	-0.4794	-0.3263	-0.1652
-0.0000	0.1652	0.3263	0.4794	0.6207	0.7467	0.8544	0.9409	1.0044	1.0430
1.0560	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
1.0000	0.9877	0.9511	0.8910	0.8090	0.7071	0.5878	0.4540	0.3090	0.1564
-0.0000	-0.1564	-0.3090	-0.4540	-0.5878	-0.7071	-0.8090	-0.8910	-0.9511	-0.9877
-1.0000	-0.9877	-0.9511	-0.8910	-0.8090	-0.7071	-0.5878	-0.4540	-0.3090	-0.1564
-0.0000	0.1564	0.3090	0.4540	0.5878	0.7071	0.8090	0.8910	0.9511	0.9877
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									

Y MAPPING

-0.0000	-1.4080	-2.7812	-4.0860	-5.2901	-6.3640	-7.2812	-8.0191	-8.5595	-8.8892
-9.0000	-8.8892	-8.5595	-8.0190	-7.2811	-6.3639	-5.2900	-4.0859	-2.7811	-1.4079
0.0000	1.4079	2.7812	4.0859	5.2901	6.3640	7.2812	8.0191	8.5595	8.8892
9.0000	8.8892	8.5595	8.0191	7.2812	6.3640	5.2901	4.0859	2.7812	1.4079
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
0.0000	-1.3322	-2.6315	-3.8660	-5.0053	-6.0214	-6.8892	-7.5874	-8.0988	-8.4107
-8.5155	-8.4107	-8.0988	-7.5874	-6.8892	-6.0214	-5.0053	-3.8660	-2.6314	-1.3321
0.0000	1.3321	2.6315	3.8660	5.0053	6.0214	6.8892	7.5874	8.0988	8.4107
8.5155	8.4107	8.0988	7.5874	6.8892	6.0214	5.0053	3.8660	2.6315	1.3321
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
0.0000	-1.2605	-2.4899	-3.6580	-4.7360	-5.6974	-6.5185	-7.1792	-7.6630	-7.9582
-8.0573	-7.9581	-7.6630	-7.1791	-6.5185	-5.6974	-4.7360	-3.6580	-2.4899	-1.2604
0.0000	1.2604	2.4899	3.6580	4.7360	5.6974	6.5185	7.1791	7.6630	7.9581
8.0573	7.9582	7.6630	7.1792	6.5185	5.6974	4.7360	3.6580	2.4899	1.2605
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
0.0000	-1.1927	-2.3560	-3.4612	-4.4813	-5.3910	-6.1679	-6.7930	-7.2508	-7.5301
-7.6240	-7.5301	-7.2508	-6.7930	-6.1679	-5.3910	-4.4813	-3.4612	-2.3559	-1.1926
0.0000	1.1926	2.3559	3.4612	4.4813	5.3910	6.1679	6.7930	7.2508	7.5301
7.6240	7.5301	7.2508	6.7930	6.1679	5.3910	4.4813	3.4612	2.3559	1.1927
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
0.0000	-1.1285	-2.2293	-3.2751	-4.2403	-5.1011	-5.8363	-6.4278	-6.8610	-7.1253
-7.2141	-7.1252	-6.8610	-6.4278	-5.8363	-5.1011	-4.2403	-3.2751	-2.2293	-1.1285
0.0000	1.1285	2.2293	3.2751	4.2403	5.1011	5.8363	6.4278	6.8610	7.1252
7.2141	7.1253	6.8610	6.4278	5.8363	5.1011	4.2403	3.2751	2.2293	1.1285
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									
0.0000	-1.0679	-2.1095	-3.0991	-4.0124	-4.8270	-5.5226	-6.0823	-6.4922	-6.7423
-6.8263	-6.7423	-6.4922	-6.0823	-5.5226	-4.8270	-4.0124	-3.0991	-2.1094	-1.0679
-0.0000	1.0679	2.1094	3.0991	4.0124	4.8270	5.5226	6.0823	6.4922	6.7423
6.8263	6.7423	6.4922	6.0823	5.5226	4.8270	4.0124	3.0991	2.1095	1.0679
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000									

```
 0.0000 -1.0105 -1.9961 -2.9326 -3.7969 -4.5676 -5.2259 -5.7556 -6.1434 -6.3801
-6.4596 -6.3801 -6.1434 -5.7555 -5.2259 -4.5676 -3.7969 -2.9326 -1.9961 -1.0105
-0.0000  1.0105  1.9961  2.9326  3.7969  4.5676  5.2259  5.7555  6.1434  6.3801
 6.4596  6.3801  6.1435  5.7556  5.2259  4.5676  3.7969  2.9326  1.9961  1.0105
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.9562 -1.8889 -2.7751 -3.5929 -4.3223 -4.9453 -5.4464 -5.8135 -6.0374
-6.1127 -6.0374 -5.8135 -5.4464 -4.9453 -4.3223 -3.5929 -2.7751 -1.8889 -0.9562
-0.0000  0.9562  1.8889  2.7751  3.5929  4.3223  4.9453  5.4464  5.8135  6.0374
 6.1127  6.0374  5.8135  5.4465  4.9453  4.3223  3.5929  2.7751  1.8889  0.9562
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.9049 -1.7875 -2.6261 -3.4000 -4.0903 -4.6798 -5.1540 -5.5014 -5.7133
-5.7845 -5.7133 -5.5014 -5.1540 -4.6798 -4.0903 -3.4000 -2.6261 -1.7875 -0.9049
-0.0000  0.9049  1.7875  2.6261  3.4000  4.0903  4.6798  5.1540  5.5014  5.7133
 5.7845  5.7133  5.5014  5.1541  4.6798  4.0903  3.4000  2.6261  1.7875  0.9049
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.8563 -1.6916 -2.4852 -3.2176 -3.8707 -4.4286 -4.8774 -5.2061 -5.4067
-5.4741 -5.4067 -5.2062 -4.8774 -4.4286 -3.8707 -3.2176 -2.4852 -1.6916 -0.8563
-0.0000  0.8563  1.6915  2.4851  3.2176  3.8707  4.4286  4.8774  5.2061  5.4067
 5.4741  5.4067  5.2062  4.8774  4.4286  3.8708  3.2176  2.4852  1.6916  0.8563
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.8104 -1.6008 -2.3518 -3.0449 -3.6631 -4.1910 -4.6158 -4.9268 -5.1166
-5.1804 -5.1166 -4.9269 -4.6158 -4.1910 -3.6631 -3.0449 -2.3518 -1.6008 -0.8104
-0.0000  0.8104  1.6008  2.3518  3.0449  3.6631  4.1910  4.6158  4.9268  5.1166
 5.1804  5.1166  4.9269  4.6158  4.1910  3.6631  3.0449  2.3518  1.6008  0.8104
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.7669 -1.5150 -2.2257 -2.8816 -3.4666 -3.9663 -4.3682 -4.6626 -4.8422
-4.9026 -4.8422 -4.6626 -4.3682 -3.9663 -3.4666 -2.8816 -2.2257 -1.5150 -0.7669
-0.0000  0.7669  1.5149  2.2257  2.8816  3.4666  3.9662  4.3682  4.6626  4.8422
 4.9026  4.8422  4.6626  4.3682  3.9663  3.4666  2.8816  2.2257  1.5150  0.7669
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

```
 0.0000 -0.7258 -1.4337 -2.1064 -2.7271 -3.2808 -3.7536 -4.1340 -4.4126 -4.5826
-4.6397 -4.5826 -4.4127 -4.1340 -3.7536 -3.2808 -2.7272 -2.1064 -1.4337 -0.7258
-0.0000  0.7258  1.4337  2.1064  2.7271  3.2808  3.7536  4.1340  4.4126  4.5826
 4.6397  4.5826  4.4127  4.1340  3.7536  3.2808  2.7272  2.1064  1.4337  0.7258
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.6869 -1.3569 -1.9935 -2.5810 -3.1049 -3.5524 -3.9125 -4.1762 -4.3370
-4.3911 -4.3370 -4.1762 -3.9125 -3.5525 -3.1050 -2.5810 -1.9935 -1.3569 -0.6869
-0.0000  0.6869  1.3569  1.9935  2.5810  3.1049  3.5524  3.9125  4.1762  4.3370
 4.3911  4.3370  4.1762  3.9125  3.5525  3.1050  2.5810  1.9935  1.3569  0.6869
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.6501 -1.2842 -1.8867 -2.4427 -2.9386 -3.3621 -3.7029 -3.9524 -4.1047
-4.1558 -4.1047 -3.9524 -3.7029 -3.3621 -2.9386 -2.4427 -1.8867 -1.2842 -0.6501
-0.0000  0.6501  1.2842  1.8867  2.4427  2.9386  3.3621  3.7029  3.9524  4.1047
 4.1558  4.1047  3.9524  3.7029  3.3621  2.9386  2.4427  1.8867  1.2842  0.6501
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.6153 -1.2154 -1.7856 -2.3119 -2.7812 -3.1821 -3.5046 -3.7407 -3.8848
-3.9333 -3.8848 -3.7408 -3.5046 -3.1821 -2.7812 -2.3119 -1.7856 -1.2154 -0.6153
-0.0000  0.6153  1.2154  1.7856  2.3119  2.7812  3.1821  3.5045  3.7407  3.8848
 3.9333  3.8848  3.7408  3.5046  3.1821  2.7812  2.3119  1.7857  1.2154  0.6153
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.5823 -1.1504 -1.6901 -2.1881 -2.6323 -3.0117 -3.3169 -3.5405 -3.6769
-3.7227 -3.6769 -3.5405 -3.3169 -3.0117 -2.6323 -2.1881 -1.6901 -1.1504 -0.5824
-0.0000  0.5823  1.1504  1.6900  2.1881  2.6323  3.0117  3.3169  3.5405  3.6769
 3.7227  3.6769  3.5405  3.3170  3.0117  2.6323  2.1881  1.6901  1.1504  0.5824
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.5512 -1.0888 -1.5996 -2.0710 -2.4915 -2.8505 -3.1394 -3.3510 -3.4801
-3.5235 -3.4801 -3.3510 -3.1394 -2.8505 -2.4915 -2.0710 -1.5996 -1.0888 -0.5512
-0.0000  0.5512  1.0888  1.5996  2.0710  2.4915  2.8505  3.1394  3.3510  3.4801
 3.5235  3.4801  3.3510  3.1394  2.8505  2.4915  2.0710  1.5996  1.0888  0.5512
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

```
 0.0000 -0.5217 -1.0306 -1.5140 -1.9602 -2.3582 -2.6980 -2.9715 -3.1717 -3.2939
-3.3350 -3.2939 -3.1718 -2.9715 -2.6981 -2.3582 -1.9602 -1.5140 -1.0306 -0.5217
-0.0000  0.5217  1.0306  1.5140  1.9602  2.3582  2.6980  2.9715  3.1717  3.2939
 3.3350  3.2939  3.1718  2.9715  2.6981  2.3582  1.9602  1.5140  1.0306  0.5217
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.4938 -0.9754 -1.4331 -1.8554 -2.2321 -2.5538 -2.8126 -3.0021 -3.1178
-3.1566 -3.1178 -3.0021 -2.8126 -2.5538 -2.2321 -1.8554 -1.4331 -0.9755 -0.4938
-0.0000  0.4938  0.9754  1.4331  1.8554  2.2321  2.5538  2.8126  3.0021  3.1178
 3.1566  3.1178  3.0021  2.8126  2.5538  2.2321  1.8554  1.4331  0.9755  0.4938
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.4674 -0.9233 -1.3565 -1.7562 -2.1127 -2.4172 -2.6622 -2.8416 -2.9511
-2.9879 -2.9511 -2.8416 -2.6622 -2.4172 -2.1128 -1.7562 -1.3565 -0.9233 -0.4674
-0.0000  0.4674  0.9233  1.3565  1.7562  2.1127  2.4172  2.6622  2.8416  2.9511
 2.9879  2.9511  2.8417  2.6622  2.4172  2.1128  1.7562  1.3565  0.9233  0.4674
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.4424 -0.8740 -1.2840 -1.6624 -1.9998 -2.2881 -2.5200 -2.6898 -2.7934
-2.8282 -2.7934 -2.6898 -2.5200 -2.2881 -1.9999 -1.6624 -1.2840 -0.8740 -0.4424
-0.0000  0.4424  0.8740  1.2840  1.6624  1.9998  2.2881  2.5200  2.6898  2.7934
 2.8282  2.7934  2.6898  2.5200  2.2881  1.9999  1.6624  1.2840  0.8740  0.4424
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.4188 -0.8273 -1.2154 -1.5736 -1.8930 -2.1658 -2.3853 -2.5461 -2.6442
-2.6771 -2.6442 -2.5461 -2.3853 -2.1658 -1.8930 -1.5736 -1.2154 -0.8273 -0.4188
-0.0000  0.4188  0.8273  1.2154  1.5736  1.8930  2.1658  2.3853  2.5461  2.6442
 2.6771  2.6442  2.5461  2.3853  2.1658  1.8930  1.5736  1.2154  0.8273  0.4188
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.3964 -0.7831 -1.1505 -1.4895 -1.7919 -2.0502 -2.2580 -2.4101 -2.5030
-2.5342 -2.5030 -2.4101 -2.2580 -2.0502 -1.7919 -1.4895 -1.1505 -0.7831 -0.3964
-0.0000  0.3964  0.7831  1.1505  1.4895  1.7919  2.0502  2.2580  2.4101  2.5030
 2.5342  2.5030  2.4101  2.2580  2.0502  1.7919  1.4896  1.1505  0.7831  0.3964
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

```
 0.0000 -0.3753 -0.7413 -1.0891 -1.4100 -1.6963 -1.9407 -2.1374 -2.2815 -2.3694
-2.3989 -2.3694 -2.2815 -2.1374 -1.9408 -1.6963 -1.4100 -1.0891 -0.7413 -0.3753
-0.0000  0.3753  0.7413  1.0891  1.4100  1.6963  1.9407  2.1374  2.2815  2.3694
 2.3989  2.3694  2.2815  2.1374  1.9408  1.6963  1.4100  1.0891  0.7413  0.3753
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.3552 -0.7017 -1.0310 -1.3348 -1.6058 -1.8372 -2.0234 -2.1597 -2.2429
-2.2709 -2.2429 -2.1597 -2.0234 -1.8372 -1.6058 -1.3348 -1.0310 -0.7017 -0.3552
-0.0000  0.3552  0.7017  1.0310  1.3348  1.6058  1.8372  2.0234  2.1597  2.2429
 2.2709  2.2429  2.1597  2.0234  1.8372  1.6058  1.3348  1.0310  0.7017  0.3552
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.3363 -0.6643 -0.9760 -1.2636 -1.5201 -1.7392 -1.9154 -2.0445 -2.1233
-2.1498 -2.1233 -2.0445 -1.9154 -1.7392 -1.5201 -1.2636 -0.9760 -0.6643 -0.3363
-0.0000  0.3363  0.6643  0.9760  1.2636  1.5201  1.7392  1.9154  2.0445  2.1233
 2.1498  2.1233  2.0445  1.9154  1.7392  1.5201  1.2636  0.9760  0.6643  0.3363
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.3184 -0.6289 -0.9239 -1.1962 -1.4390 -1.6464 -1.8133 -1.9355 -2.0101
-2.0351 -2.0101 -1.9355 -1.8133 -1.6464 -1.4390 -1.1962 -0.9239 -0.6289 -0.3184
-0.0000  0.3184  0.6289  0.9239  1.1962  1.4390  1.6464  1.8133  1.9355  2.0101
 2.0351  2.0101  1.9355  1.8133  1.6464  1.4390  1.1962  0.9239  0.6289  0.3184
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.3014 -0.5954 -0.8747 -1.1324 -1.3623 -1.5587 -1.7166 -1.8323 -1.9029
-1.9266 -1.9029 -1.8323 -1.7166 -1.5587 -1.3623 -1.1324 -0.8747 -0.5954 -0.3014
-0.0000  0.3014  0.5954  0.8747  1.1324  1.3623  1.5587  1.7166  1.8323  1.9029
 1.9266  1.9029  1.8323  1.7166  1.5587  1.3623  1.1324  0.8747  0.5954  0.3014
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.2853 -0.5636 -0.8281 -1.0721 -1.2897 -1.4756 -1.6252 -1.7347 -1.8015
-1.8240 -1.8015 -1.7347 -1.6252 -1.4756 -1.2897 -1.0721 -0.8281 -0.5636 -0.2853
-0.0000  0.2853  0.5636  0.8281  1.0721  1.2897  1.4756  1.6252  1.7347  1.8015
 1.8240  1.8015  1.7347  1.6252  1.4756  1.2897  1.0721  0.8281  0.5636  0.2853
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

```
 0.0000 -0.2701 -0.5336 -0.7839 -1.0150 -1.2210 -1.3970 -1.5386 -1.6423 -1.7055
-1.7268 -1.7055 -1.6423 -1.5386 -1.3970 -1.2210 -1.0150 -0.7839 -0.5336 -0.2701
-0.0000  0.2701  0.5336  0.7839  1.0150  1.2210  1.3970  1.5386  1.6423  1.7055
 1.7268  1.7055  1.6423  1.5386  1.3970  1.2210  1.0150  0.7839  0.5336  0.2701
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.2557 -0.5052 -0.7422 -0.9609 -1.1560 -1.3226 -1.4567 -1.5548 -1.6147
-1.6348 -1.6147 -1.5548 -1.4567 -1.3226 -1.1560 -0.9609 -0.7422 -0.5052 -0.2557
-0.0000  0.2557  0.5052  0.7422  0.9609  1.1560  1.3226  1.4567  1.5548  1.6147
 1.6348  1.6147  1.5548  1.4567  1.3226  1.1560  0.9609  0.7422  0.5052  0.2557
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.2421 -0.4783 -0.7027 -0.9098 -1.0945 -1.2522 -1.3791 -1.4721 -1.5288
-1.5478 -1.5288 -1.4721 -1.3791 -1.2522 -1.0945 -0.9098 -0.7027 -0.4783 -0.2421
-0.0000  0.2421  0.4783  0.7027  0.9098  1.0945  1.2522  1.3791  1.4721  1.5288
 1.5478  1.5288  1.4721  1.3791  1.2522  1.0945  0.9098  0.7027  0.4783  0.2421
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.2293 -0.4529 -0.6653 -0.8614 -1.0362 -1.1856 -1.3057 -1.3937 -1.4474
-1.4655 -1.4474 -1.3937 -1.3057 -1.1856 -1.0362 -0.8614 -0.6653 -0.4529 -0.2292
 0.0000  0.2292  0.4529  0.6653  0.8614  1.0362  1.1856  1.3057  1.3937  1.4474
 1.4655  1.4474  1.3937  1.3057  1.1856  1.0362  0.8614  0.6653  0.4529  0.2293
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.2171 -0.4288 -0.6299 -0.8156 -0.9811 -1.1225 -1.2363 -1.3196 -1.3704
-1.3875 -1.3704 -1.3196 -1.2363 -1.1225 -0.9811 -0.8156 -0.6299 -0.4288 -0.2171
 0.0000  0.2171  0.4288  0.6299  0.8156  0.9811  1.1225  1.2363  1.3196  1.3704
 1.3875  1.3704  1.3196  1.2363  1.1225  0.9811  0.8156  0.6299  0.4288  0.2171
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

-0.0000 -0.2055 -0.4060 -0.5964 -0.7722 -0.9289 -1.0628 -1.1705 -1.2494 -1.2976
-1.3137 -1.2976 -1.2494 -1.1705 -1.0628 -0.9289 -0.7722 -0.5964 -0.4060 -0.2055
 0.0000  0.2055  0.4060  0.5964  0.7722  0.9289  1.0628  1.1705  1.2494  1.2976
 1.3137  1.2976  1.2494  1.1705  1.0628  0.9289  0.7722  0.5964  0.4060  0.2055
-0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

```
-0.0000 -0.1946 -0.3844 -0.5647 -0.7312 -0.8796 -1.0063 -1.1083 -1.1830 -1.2286
-1.2439 -1.2286 -1.1830 -1.1083 -1.0063 -0.8796 -0.7311 -0.5647 -0.3844 -0.1946
 0.0000  0.1946  0.3844  0.5647  0.7311  0.8796  1.0063  1.1083  1.1830  1.2286
 1.2439  1.2286  1.1830  1.1083  1.0063  0.8796  0.7311  0.5647  0.3844  0.1946
-0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.1843 -0.3640 -0.5347 -0.6923 -0.8328 -0.9529 -1.0494 -1.1202 -1.1633
-1.1778 -1.1633 -1.1202 -1.0494 -0.9529 -0.8328 -0.6923 -0.5347 -0.3640 -0.1843
 0.0000  0.1843  0.3640  0.5347  0.6923  0.8328  0.9529  1.0494  1.1202  1.1633
 1.1778  1.1633  1.1202  1.0494  0.9529  0.8328  0.6923  0.5347  0.3640  0.1843
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.1745 -0.3446 -0.5063 -0.6555 -0.7886 -0.9023 -0.9937 -1.0607 -1.1015
-1.1153 -1.1015 -1.0607 -0.9937 -0.9023 -0.7886 -0.6555 -0.5063 -0.3446 -0.1745
 0.0000  0.1745  0.3446  0.5063  0.6555  0.7886  0.9023  0.9937  1.0607  1.1015
 1.1153  1.1015  1.0607  0.9937  0.9023  0.7886  0.6555  0.5063  0.3446  0.1745
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

 0.0000 -0.1652 -0.3263 -0.4794 -0.6207 -0.7467 -0.8544 -0.9409 -1.0044 -1.0430
-1.0560 -1.0430 -1.0044 -0.9409 -0.8544 -0.7467 -0.6207 -0.4794 -0.3263 -0.1652
 0.0000  0.1652  0.3263  0.4794  0.6207  0.7467  0.8544  0.9409  1.0044  1.0430
 1.0560  1.0430  1.0044  0.9409  0.8544  0.7467  0.6207  0.4794  0.3263  0.1652
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000

-0.0000 -0.1564 -0.3090 -0.4540 -0.5878 -0.7071 -0.8090 -0.8910 -0.9511 -0.9877
-1.0000 -0.9877 -0.9511 -0.8910 -0.8090 -0.7071 -0.5878 -0.4540 -0.3090 -0.1564
 0.0000  0.1564  0.3090  0.4540  0.5878  0.7071  0.8090  0.8910  0.9511  0.9877
 1.0000  0.9877  0.9511  0.8910  0.8090  0.7071  0.5878  0.4540  0.3090  0.1564
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000
```

ALPHA VALUES FOR THE LAYER ARE :

0.234702	0.234698	0.234697	0.234694	0.234692
0.222154	0.222152	0.222147	0.222145	0.222143
0.198727	0.198724	0.198726	0.198725	0.198723
0.177777	0.177782	0.177784	0.177781	0.177782
0.159050	0.159053	0.159054	0.159053	0.159055
0.142305	0.142304	0.142306	0.142305	0.142309
0.127331	0.127330	0.127330	0.127332	0.127334
0.113938	0.113938	0.113938	0.113941	0.113942
0.101959	0.101958	0.101959	0.101959	0.101961
0.091242	0.091242	0.091242	0.091243	0.091244
0.081655	0.081655	0.081655	0.081656	0.081657
0.073079	0.073079	0.073079	0.073080	0.073079
0.065406	0.065407	0.065406	0.065407	0.065406
0.058543	0.058542	0.058541	0.058543	0.058542
0.052399	0.052399	0.052399	0.052400	0.052399
0.046903	0.046903	0.046903	0.046903	0.046902
0.041984	0.041984	0.041984	0.041984	0.041984
0.037583	0.037583	0.037583	0.037582	0.037582
0.033644	0.033645	0.033644	0.033643	0.033643
0.030120	0.030120	0.030120	0.030119	0.030118

0.026965	0.026966	0.026965	0.026965	0.026964
0.024142	0.024142	0.024142	0.024142	0.024142
0.021615	0.021615	0.021615	0.021616	0.021615
0.019354	0.019354	0.019354	0.019355	0.019355
0.017331	0.017331	0.017331	0.017331	0.017331
0.015519	0.015519	0.015520	0.015520	0.015520
0.013898	0.013898	0.013898	0.013898	0.013898
0.012446	0.012446	0.012447	0.012447	0.012446
0.011147	0.011147	0.011147	0.011147	0.011147
0.009983	0.009983	0.009983	0.009983	0.009983
0.008941	0.008941	0.008941	0.008941	0.008941
0.008009	0.008009	0.008008	0.008008	0.008008
0.007173	0.007173	0.007173	0.007173	0.007173
0.006425	0.006425	0.006425	0.006425	0.006425
0.005756	0.005756	0.005755	0.005755	0.005755
0.005156	0.005156	0.005156	0.005156	0.005155
0.004619	0.004619	0.004618	0.004618	0.004618
0.004138	0.004138	0.004137	0.004137	0.004137
0.003707	0.003707	0.003707	0.003707	0.003707
0.003321	0.003321	0.003321	0.003321	0.003321
0.003141	0.003141	0.003141	0.003141	0.003141

BETA VALUES FOR THE LAYER ARE :

0.000075	0.000019	0.000007	0.000002	0.000006
-0.000007	0.000018	0.000009	0.000005	0.000001
-0.000003	0.000015	0.000012	0.000009	0.000003
0.000001	0.000010	0.000010	0.000010	0.000008
0.000001	0.000008	0.000011	0.000014	0.000011
0.000002	0.000007	0.000011	0.000013	0.000010
0.000003	0.000005	0.000009	0.000009	0.000007
0.000003	0.000004	0.000007	0.000007	0.000006
0.000002	0.000003	0.000005	0.000006	0.000005
0.000002	0.000002	0.000003	0.000004	0.000003
0.000002	0.000000	0.000001	0.000001	0.000002
0.000002	-0.000000	0.000001	0.000000	0.000001
0.000001	-0.000001	-0.000000	-0.000000	0.000001
0.000001	0.000000	-0.000001	-0.000001	0.000000
0.000001	-0.000000	-0.000002	-0.000001	0.000000
0.000001	-0.000000	-0.000002	-0.000002	-0.000001
0.000001	-0.000000	-0.000002	-0.000002	-0.000000
0.000001	-0.000001	-0.000002	-0.000002	-0.000001
0.000001	-0.000001	-0.000001	-0.000002	-0.000001
0.000000	-0.000000	-0.000001	-0.000000	-0.000001

0.000000	-0.000000	0.000000	0.000000	-0.000001
0.000000	0.000000	0.000001	0.000001	0.000000
0.000000	0.000000	0.000001	0.000001	0.000001
0.000000	0.000000	0.000001	0.000000	0.000000
0.000000	-0.000000	-0.000000	-0.000000	0.000000
0.000000	-0.000000	-0.000000	-0.000000	0.000000
0.000000	-0.000000	-0.000000	-0.000000	-0.000000
0.000000	-0.000000	-0.000000	-0.000000	-0.000000
0.000000	-0.000001	-0.000001	-0.000000	-0.000000
0.000000	-0.000001	-0.000001	-0.000000	-0.000000
0.000000	-0.000001	-0.000001	-0.000000	-0.000000
0.000000	-0.000000	-0.000001	-0.000000	-0.000000
0.000000	-0.000000	-0.000001	-0.000001	-0.000000
0.000000	-0.000000	-0.000001	-0.000001	-0.000000
-0.000000	-0.000000	-0.000000	-0.000000	-0.000000
-0.000000	-0.000000	-0.000000	-0.000000	-0.000000
-0.000000	-0.000000	-0.000000	-0.000000	-0.000000
0.000000	-0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	-0.000000	0.000000	0.000000	0.000000
-0.000001	-0.000000	0.000000	-0.000000	-0.000000

GAMMA VALUES FOR THE LAYER ARE :

1.982297	1.982210	1.982210	1.982200	1.982200
1.774610	1.774612	1.774572	1.774553	1.774546
1.588758	1.588765	1.588745	1.588731	1.588715
1.422436	1.422443	1.422441	1.422431	1.422419
1.273581	1.273591	1.273596	1.273587	1.273577
1.140352	1.140362	1.140370	1.140371	1.140366
1.021104	1.021116	1.021130	1.021133	1.021129
0.914365	0.914376	0.914389	0.914397	0.914396
0.818816	0.818828	0.818841	0.818851	0.818850
0.733284	0.733292	0.733307	0.733317	0.733320
0.656713	0.656721	0.656732	0.656744	0.656748
0.588163	0.588168	0.588179	0.588190	0.588195
0.526791	0.526794	0.526802	0.526814	0.526817
0.471841	0.471842	0.471850	0.471860	0.471864
0.422640	0.422640	0.422646	0.422655	0.422658
0.378584	0.378585	0.378588	0.378596	0.378599
0.339135	0.339136	0.339138	0.339143	0.339146
0.303809	0.303810	0.303812	0.303814	0.303817
0.272174	0.272175	0.272176	0.272177	0.272179
0.243843	0.243844	0.243844	0.243845	0.243846

0.218469	0.218470	0.218471	0.218471	0.218472
0.195744	0.195745	0.195745	0.195746	0.195747
0.175389	0.175390	0.175391	0.175392	0.175393
0.157157	0.157158	0.157159	0.157160	0.157161
0.140826	0.140827	0.140828	0.140829	0.140830
0.126197	0.126198	0.126199	0.126200	0.126201
0.113093	0.113093	0.113094	0.113095	0.113096
0.101353	0.101353	0.101354	0.101355	0.101355
0.090836	0.090836	0.090836	0.090837	0.090837
0.081413	0.081413	0.081413	0.081414	0.081414
0.072971	0.072970	0.072970	0.072971	0.072971
0.065407	0.065406	0.065406	0.065406	0.065406
0.058629	0.058629	0.058628	0.058628	0.058628
0.052556	0.052556	0.052555	0.052554	0.052554
0.047114	0.047113	0.047113	0.047112	0.047112
0.042237	0.042237	0.042236	0.042235	0.042235
0.037866	0.037866	0.037865	0.037865	0.037864
0.033949	0.033949	0.033948	0.033948	0.033948
0.030439	0.030439	0.030438	0.030438	0.030438
0.027293	0.027293	0.027292	0.027291	0.027291
0.024473	0.024472	0.024472	0.024472	0.024472

JACOBIAN VALUES FOR THE LAYER ARE :

0.682092	0.682072	0.682070	0.682063	0.682061
0.627883	0.627881	0.627869	0.627861	0.627856
0.561898	0.561895	0.561894	0.561890	0.561884
0.502869	0.502877	0.502879	0.502873	0.502873
0.450071	0.450076	0.450078	0.450075	0.450076
0.402837	0.402837	0.402841	0.402840	0.402846
0.360581	0.360582	0.360583	0.360588	0.360589
0.322771	0.322773	0.322776	0.322780	0.322783
0.288938	0.288940	0.288944	0.288945	0.288948
0.258662	0.258664	0.258667	0.258670	0.258671
0.231568	0.231570	0.231572	0.231575	0.231577
0.207322	0.207323	0.207325	0.207327	0.207327
0.185621	0.185623	0.185624	0.185626	0.185626
0.166201	0.166200	0.166201	0.166204	0.166204
0.148815	0.148815	0.148817	0.148819	0.148818
0.133254	0.133254	0.133254	0.133256	0.133256
0.119324	0.119324	0.119325	0.119325	0.119326
0.106855	0.106856	0.106856	0.106855	0.106856
0.095693	0.095694	0.095693	0.095692	0.095692
0.085700	0.085700	0.085700	0.085699	0.085698

0.076753	0.076754	0.076753	0.076753	0.076752
0.068743	0.068743	0.068743	0.068744	0.068743
0.061572	0.061572	0.061572	0.061573	0.061573
0.055151	0.055151	0.055152	0.055153	0.055153
0.049403	0.049403	0.049403	0.049404	0.049404
0.044255	0.044255	0.044256	0.044256	0.044256
0.039645	0.039646	0.039646	0.039646	0.039646
0.035517	0.035517	0.035518	0.035518	0.035518
0.031820	0.031820	0.031821	0.031820	0.031820
0.028509	0.028509	0.028509	0.028509	0.028509
0.025543	0.025543	0.025543	0.025543	0.025543
0.022887	0.022887	0.022887	0.022887	0.022887
0.020508	0.020508	0.020507	0.020507	0.020507
0.018376	0.018376	0.018376	0.018376	0.018376
0.016467	0.016467	0.016467	0.016466	0.016466
0.014757	0.014757	0.014756	0.014756	0.014756
0.013224	0.013225	0.013224	0.013224	0.013224
0.011852	0.011852	0.011852	0.011851	0.011851
0.010622	0.010622	0.010622	0.010622	0.010622
0.009520	0.009520	0.009520	0.009520	0.009520
0.008767	0.008767	0.008767	0.008767	0.008767

APPENDIX D

Computer Subroutine for Transformed Conduction Equation

```
      *....*...1.........2.........3.........4.........5.........6.........7.*
      C
      C
      C
 1          SUBROUTINE TSOLN (IMAX,JMAX,TS,CA,CB,CG,CJ,GEN,COND)
      C     IMPLICIT REAL*8(A-H,O-Z)
 2          DIMENSION TS(70,52),CA(70,52),CB(70,52),CG(70,52),CJ(70,52)
 3          REAL MAXER
      C
      C
      C    ***************************************************************
      C    * COMPUTATION THROUGH TEMPERATURE FIELD NOT INCLUDING       *
      C    *       RE-ENTRANT BOUNDARY                                 *
      C    ***************************************************************
      C
      C
 4      350 MAXER=0.0
 5          KIMAX=IMAX-1
 6          KJMAX=JMAX-1
 7          DO 340 I=2,KIMAX
 8          DO 341 J=2,KJMAX
 9          TEN=(TS(I+1,J+1)-TS(I+1,J-1)+TS(I-1,J-1)-TS(I-1,J+1))/4.0
10          TITER=((CA(I,J)*(TS(I+1,J)+TS(I-1,J)))+(CG(I,J)*(TS(I,J+1)+TS(I,
           1J-1)))-2.0*CB(I,J)*TEN+(CJ(I,J)*CJ(I,J)*GEN)/COND)
           2/(2.0*(CA(I,J)+CG(I,J)))
11          TER=ABS((TS(I,J)-TITER)/TITER)
12          IF (TER.GT.MAXER) MAXER=TER
14      341 TS(I,J)=TITER
15      340 CONTINUE
```

```
C
C
C   ********************************************************************
C   * COMPUTATION THROUGH TEMPERATURE FIELD ALONG RE-ENTRANT           *
C   *       BOUNDARY                                                   *
C   ********************************************************************
C
C
16          DO 345 J=2,KJMAX
17          TEN=(TS(2,J+1)-TS(2,J-1)+TS(IMAX-1,J-1)-TS(IMAX-1,J+1))/4.0
18          TITER=((CA(IMAX,J)*(TS(2,J)+TS(IMAX-1,J)))+(CG(IMAX,J)*
           1(TS(IMAX,J+1)+TS(IMAX,J-1)))-2.0*CB(IMAX,J)*TEN
           2+(CJ(IMAX,J)*CJ(IMAX,J)*GEN)/COND)/(2.0*(CA(IMAX,J)+CG(IMAX,J)))
19          TER=ABS((TS(IMAX,J)-TITER)/TITER)
20          IF (TER.GT.MAXER) MAXER=TER
22          TS(IMAX,J)=TITER
23      345 TS(1,J)=TITER
24          IF (MAXER.GT.0.0001) GOTO 350
25          RETURN
26          END
```

```
     *....*...1.........2.........3.........4.........5.........6.........7.*
     C
     C
     C
     C
1           SUBROUTINE TIME1(TOLD,TNEW,CA,CB,CG,CJ,IMAX,JMAX,DIFFUS,CONDUC,
           1GENRAT,RATIO,TSTEP,ICOUNT)
     C      IMPLICIT REAL*8(A-H,O-Z)
2           DIMENSION TOLD(70,52),TNEW(70,52),CA(70,52),CB(70,52),CG(70,52),
           1CJ(70,52)
     C
     C
     C    ****************************************************************
     C    * THIS SUBROUTINE CONTAINS A GENERAL ALGORITHM FOR             *
     C    *      FOR THE TIME DEPENDENT PROBLEM THAT COMBINES BOTH       *
     C    *      THE IMPLICIT AND EXPLICIT ALGORITHMS. THE VALUE         *
     C    *      'RATIO' DETERMINES WHAT PERCENTAGE OF EACH SPECIFIC     *
     C    *      ALGORITHM IS CONTAINED IN THE GENERAL ALGORITHM.        *
     C    *      RATIO = 0.0 MAKES THE GENERAL ALGORITHM FULLY           *
     C    *      EXPLICIT, AND RATIO = 1.0 FULLY IMPLICIT. WITH          *
     C    *      RATIO = 0.5, THE GENERAL ALGORITHM REDUCES TO THE       *
     C    *      CRANK-NICHOLSON ITERATIVE SCHEME.                       *
     C    ****************************************************************
     C
     C
3           REAL MAXER
     C
     C
```

```
C    ***********************************************************
C    * COMPUTATION THROUGH TEMPERATURE FIELD NOT INCLUDING     *
C    *      RE-ENTRANT BOUNDARY                                *
C    ***********************************************************
C
C
 4        ICOUNT=0
 5    350 MAXER=0.0
 6        KIMAX=IMAX-1
 7        KJMAX=JMAX-1
 8        DO 340 J=2,KJMAX
 9        DO 341 I=2,KIMAX
10        TEXPL=(1.0-RATIO)*(CA(I,J)*(TOLD(I+1,J)-2.0*TOLD(I,J)+TOLD(I-1,J))
         1-(CB(I,J)/2.0)*(TOLD(I+1,J+1)-TOLD(I+1,J-1)+TOLD(I-1,J-1)-TOLD(I-1
         2,J+1))+CG(I,J)*(TOLD(I,J+1)-2.0*TOLD(I,J)+TOLD(I,J-1)))
11        TIMPL=RATIO*(CA(I,J)*(TNEW(I+1,J)+TNEW(I-1,J))-(CB(I,J)/2.0)*(TNEW
         1(I+1,J+1)-TNEW(I+1,J-1)+TNEW(I-1,J-1)-TNEW(I-1,J+1))+CG(I,J)*(TNEW
         2(I,J+1)+TNEW(I,J-1)))
12        TITER=(TEXPL+TIMPL+CJ(I,J)*CJ(I,J)*GENRAT/CONDUC*(CJ(I,J)*CJ(I,J)*
         1TOLD(I,J))/(DIFFUS*TSTEP))/((CJ(I,J)*CJ(I,J))/(DIFFUS*TSTEP)+2.0*
         2RATIO*CA(I,J)+2.0*RATIO*CG(I,J))
13        TER=ABS(TNEW(I,J)-TITER)
14        IF(TER.GT.MAXER)MAXER=TER
16    341 TNEW(I,J)=TITER
17    340 CONTINUE
C
```

```
C    ***************************************************************
C    * COMPUTATION THROUGH TEMPERATURE FIELD ALONG RE-ENTRANT      *
C    *       BOUNDARY                                              *
C    ***************************************************************
C
C
18          DO 345 J=2,KJMAX
19          TEXPL=(1.0-RATIO)*(CA(IMAX,J)*(TOLD(2,J)-2.0*TOLD(IMAX,J)+TOLD
           1(IMAX-1,J))-(CB(IMAX,J)/2.0)*(TOLD(2,J+1)-TOLD(2,J-1)+TOLD(IMAX-1
           2,J-1)-TOLD(IMAX-1,J+1))+CG(IMAX,J)*(TOLD(IMAX,J+1)-2.0*TOLD(IMAX,
           3J)+TOLD(IMAX,J-1)))
20          TIMPL=RATIO*(CA(IMAX,J)*(TNEW(2,J)+TNEW(IMAX-1,J))-(CB(IMAX,J)/2.0
           1)*(TNEW(2,J+1)-TNEW(2,J-1)+TNEW(IMAX-1,J-1)-TNEW(IMAX-1,J+1))+CG(
           2IMAX,J)*(TNEW(IMAX,J+1)+TNEW(IMAX,J-1)))
21          TITER=(TEXPL+TIMPL+CJ(IMAX,J)*CJ(IMAX,J)*GENRAT/CONDUC+(CJ(IMAX,J)
           1*CJ(IMAX,J)*TOLD(IMAX,J))/(DIFFUS*TSTEP))/((CJ(IMAX,J)*CJ(IMAX,J))
           2/(DIFFUS*TSTEP)+2.0*RATIO*CA(IMAX,J)+2.0*RATIO*CG(IMAX,J))
22          TER=ABS(TNEW(IMAX,J)-TITER)
23          IF(TER.GT.MAXER)MAXER=TER
25          TNEW(IMAX,J)=TITER
26        345 TNEW(1,J)=TITER
C
C
C    ***************************************************************
C    * CHECK FOR CONVERGENCE REQUIREMENT                           *
C    ***************************************************************

27          ICOUNT=ICOUNT+1
28          IF(MAXER.GT.0.001)GOTO 350
29          RETURN
30          END
```

APPENDIX E

Computer Subroutine for Transformed Conductive Boundary Equation

```
     *....*...1.........2.........3.........4.........5.........6.........7.*
     C
     C
     C
1          SUBROUTINE BNDRY1 (TUN,BU,GU,JU,CU,TLN,BL,GL,JL,CL,IMAX,JMAX,
          1ITERAT,TUO,TLO)
     C     IMPLICIT REAL*8(A-H,O-Z)
2          REAL MAXER,JL,JU
3          DIMENSION TUN(70,52),BU(70,52),GU(70,52),JU(70,52),TLN(70,52),
          1BL(70,52),GL(70,52),JL(70,52),TUO(70,52),TLO(70,52)
     C
     C
     C    ***************************************************************
     C    * THIS SUBROUTINE SOLVES FOR THE INTERFACE BOUNDARY           *
     C    *       CONDITION:                                            *
     C    *                   -K1(D(T1)/D(N1)) = -K2(D(T2)/D(N2))       *
     C    *                                                             *
     C    * LATERAL HEAT FLUX AND HEAT GENERATION ARE NOT               *
     C    *       ACCOUNTED FOR IN THE ALGORITHM                        *
     C    *                                                             *
     C    ***************************************************************
     C
     C
4          ITERAT=0
     C
     C
     C    ***************************************************************
     C    * SWEEP THROUGH BOUNDARY POINTS NOT INCLUDING END POINTS      *
     C    ***************************************************************
```

```
      C
      C
    5       30 MAXER=0.0
    6          KIMAX=IMAX-1
    7          DO 40 K=2,KIMAX
    8          B1=(CU*GU(K,1))/(JU(K,1)*SQRT(GU(K,1)))
    9          B6=(CU*BU(K,1)*(TUN(K+1,1)-TUN(K-1,1))/2.0)/(JU(K,1)*SQRT(GU(K,1
           1 )))
   10          B7=(CU*BU(K,1)*(TUO(K+1,1)-TUO(K-1,1))/2.0)/(JU(K,1)*SQRT(GU(K,1
           1 )))
   11          B2=(B7+B6)/2.0
   12          B3=(CL*GL(K,JMAX))/(JL(K,JMAX)*SQRT(GL(K,JMAX)))
   13          B8=(CL*BL(K,JMAX)*(TLN(K+1,JMAX)-TLN(K-1,JMAX))/2.0)/(JL(K,JMAX)*
           1SQRT(GL(K,JMAX)))
   14          B9=(CL*BL(K,JMAX)*(TLO(K+1,JMAX)-TLO(K-1,JMAX))/2.0)/(JL(K,JMAX)*
           1SQRT(GL(K,JMAX)))
   15          B4=(B8+B9)/2.0
   16          TITER=((B1/2.0)*(TUN(K,2)+TUO(K,2)-TUO(K,1))-B2+
           1 (B3/2.0)*(TLN(K,JMAX-1)+TLO(K,JMAX-1)-TLO(K,JMAX))+B4)/
           2 (B1/2.0+B3/2.0)
   17          TER=ABS(TUN(K,1)-TITER)
   18          IF(TER.GT.MAXER)MAXER=TER
   20          TUN(K,1)=TITER
   21       40 TLN(K,JMAX)=TITER
      C
3.0 (MAY 1983)            VS FORTRAN           DATE: JAN 15, 1987    TIME: 17:31:
      *....*...1.........2.........3.........4.........5.........6.........7.*
      C
      C    ***************************************************************
      C    * EVALUATION OF ENDPOINTS                                      *
      C    ***************************************************************
```

```
      C
      C
   22          B1=(CU*GU(IMAX,1))/(JU(IMAX,1)*SQRT(GU(IMAX,1)))
   23          B6=(CU*BU(IMAX,1)*(TUN(2,1)-TUN(IMAX-1,1))/2.0)/(JU(IMAX,1)*
              1SQRT(GU(IMAX,1)))
   24          B7=(CU*BU(IMAX,1)*(TUO(2,1)-TUO(IMAX-1,1))/2.0)/(JU(IMAX,1)*
              1SQRT(GU(IMAX,1)))
   25          B2=(B6+B7)/2.0
   26          B3=(CL*GL(IMAX,JMAX))/(JL(IMAX,JMAX)*SQRT(GL(IMAX,JMAX)))
   27          B8=(CL*BL(IMAX,JMAX)*(TLN(2,JMAX)-TLN(IMAX-1,JMAX))/2.0)/
              1(JL(IMAX,JMAX)*SQRT(GL(IMAX,JMAX)))
   28          B9=(CL*BL(IMAX,JMAX)*(TLO(2,JMAX)-TLO(IMAX-1,JMAX))/2.0)/
              1(JL(IMAX,JMAX)*SQRT(GL(IMAX,JMAX)))
   29          B4=(B8+B9)/2.0
   30          TITER=((B1/2.0)*(TUN(IMAX,2)+TUO(IMAX,2)-TUO(IMAX,1))-B2+
              1 (B3/2.0)*(TLN(IMAX,JMAX-1)+TLO(IMAX,JMAX-1)-TLO(IMAX,JMAX))+B4)/
              2 (B1/2.0+B3/2.0)
   31          TER=ABS(TUN(IMAX,1)-TITER)
   32          IF(TER.GT.MAXER)MAXER=TER
   34          TUN(IMAX,1)=TITER
   35          TLN(IMAX,JMAX)=TITER
   36          TUN(1,1)=TITER
   37          TLN(1,JMAX)=TITER
   38          ITERAT=ITERAT+1
   39          IF(MAXER.GT.0.001)GOTO 30
   40      219 FORMAT(' ',' MAXER PREVIOUS STEP =  ',F10.4)
   41      220 FORMAT(' ',' MAXER FINAL =  ',F10.4)
   42      221 FORMAT(' ',11F10.4)
   43      230 FORMAT(' ',' TU(K,1)=  ',F10.4,'  TITER =  ',F10.4,'  TER =  ',
```

```
     1F10.4,'  MAXER =  ',F10.4)
240 FORMAT(' ',' B1 = ',F10.4)
241 FORMAT(' ',' B2 = ',F10.4)
242 FORMAT(' ',' B3 = ',F10.4)
243 FORMAT(' ',' B4 = ',F10.4)
244 FORMAT(' ',' TITER = ',F10.4)
604 FORMAT(' ','CU = ',F10.4,' GU(K,1) = ',F10.4,' JU(K,1) = ',F10.4)
614 FORMAT(' ','CU = ',F10.4,' BU(K,1) = ',F10.4,' TU(K+1,1) = ',
     1F10.4)
615 FORMAT(' ','TU(K-1,1) = ',F10.4,' JU(K,1) = ',F10.4,' GU(K,1) = ',
     1F10.4)
624 FORMAT(' ','CL = ',F10.4,' GL(K,JMAX) = ',F10.4,' JL(K,JMAX) = ',
     1F10.4)
634 FORMAT(' ','CL = ',F10.4,' BL(K,JMAX) = ',F10.4,' TL(K+1,JMAX) = '
     1,F10.4)
635 FORMAT(' ','TL(K-1,JMAX) = ',F10.4,' JL(K,JMAX) = ',F10.4,
     1' GL(K,JMAX) = ',F10.4)
800 FORMAT(' ','FOR NODE ',I3)
      RETURN
      END
```

APPENDIX F

Computer Subroutine for Transformed Phase Change Equations

```
      SUBROUTINE MELT2(TOLD,TNEW,CA,CB,CG,CJ,IMAX,JMAX,DIFFUS,CONDUC,
     1GENRAT,RATIO,TSTEP,ICOUNT,HOLD,HNEW,COLD,CNEW,HSMELT,HLMELT,
     2SPHTS,SPHTL,CONDS,CONDL,TMELT,DENSTY,OUT,TX,TY,TSTART)
      DIMENSION TOLD(121,30),TNEW(121,30),CA(121,30),CB(121,30),
     1 CG(121,30),CJ(121,30),HOLD(121,30),HNEW(121,30),COLD(121,30),
     2 CNEW(121,30),TX(121,30),TY(121,30)
C
C
C    ****************************************************************
C    * THIS SUBROUTINE CONTAINS ALL THREE ALGORITHMS FOR THE        *
C    *      TIME DEPENDENT MELTING PROBLEM THAT COMBINES BOTH       *
C    *      THE IMPLICIT AND EXPLICIT ALGORITHMS. THE VALUE         *
C    *      'RATIO' DETERMINES WHAT PERCENTAGE OF EACH SPECIFIC     *
C    *      ALGORITHM IS CONTAINED IN THE GENERAL ALGORITHM.        *
C    *      RATIO = 0.0 MAKES THE GENERAL ALGORITHM FULLY           *
C    *      EXPLICIT, AND RATIO = 1.0 FULLY IMPLICIT. WITH          *
C    *      RATIO = 0.5, THE GENERAL ALGORITHM REDUCES TO THE       *
C    *      CRANK-NICHOLSON ITERATIVE SCHEME.                       *
C    * THERE IS A SEPARATE ALGORITHM FOR EACH OF THE THREE          *
C    *      POSSIBLE STATES OF A COMPUTATIONAL CELL: SOLID,         *
C    *      MELT, AND LIQUID.                                       *
C    ****************************************************************
C
C
      REAL MAXER
C
C
C    ****************************************************************
C    * COMPUTATION THROUGH TEMPERATURE FIELD NOT INCLUDING          *
C    *       RE-ENTRANT BOUNDARY                                    *
C    ****************************************************************
C
C
```

```
      ICOUNT=0
C     WRITE(6,999)
C     WRITE(6,904)
C     WRITE(6,999)
C     CALL COEFPR(IMAX,JMAX,CA,CB,CG,CJ)
C     WRITE(6,999)
C     CALL TPRINT(IMAX,JMAX,TOLD,TNEW)
C     WRITE(6,999)
C     CALL HPRINT(IMAX,JMAX,HOLD,HNEW)
C     WRITE(6,999)
C     CALL CPRINT(IMAX,JMAX,COLD,CNEW)
C     WRITE(6,999)
C     CALL BFPRIN(IMAX,JMAX,TX,TY)
C     WRITE(6,999)
C     WRITE(6,912)IMAX,JMAX,DIFFUS,CONDUC
C     WRITE(6,999)
C     WRITE(6,913)GENRAT,RATIO,TSTEP,ICOUNT
C     WRITE(6,999)
C     WRITE(6,914)HSMELT,HLMELT,SPHTS,SPHTL
C     WRITE(6,999)
C     WRITE(6,915)CONDS,CONDL,TMELT,DENSTY
C     WRITE(6,999)
  350 MAXER=0.0
      KIMAX=IMAX-1
      KJMAX=JMAX-1
      HSEMEL=HSMELT-1.0
      HLEMEL=HLMELT-1.0
      DO 340 J=2,KJMAX
      DO 341 I=51,71
      OUT=0
      ITRIP=1
C     WRITE(6,906)I,J
C     WRITE(6,999)
```

```
C      WRITE(6,902)DELKXO,DELKXN,DELTXO,DELTXN
C      WRITE(6,999)
C      WRITE(6,903)DELKYO,DELKYN,DELTYO,DELTYN
C      WRITE(6,999)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       CB1O=((((COLD(I+1,J+1)+COLD(I+1,J-1))/2.0)*((TOLD(I+1,J+1)-
      1 TOLD(I+1,J-1))/2.0)-
      2 ((COLD(I-1,J+1)+COLD(I-1,J-1))/2.0)*((TOLD(I-1,J+1)-
      3 TOLD(I-1,J-1))/2.0))/2.0)*CB(I,J)
       CB2O=((((COLD(I+1,J+1)+COLD(I-1,J+1))/2.0)*((TOLD(I+1,J+1)-
      1 TOLD(I-1,J+1))/2.0)-
      2 ((COLD(I+1,J-1)+COLD(I-1,J-1))/2.0)*((TOLD(I+1,J-1)-
      3 TOLD(I-1,J-1))/2.0))/2.0)*CB(I,J)
       CA1O=((COLD(I+1,J)+COLD(I,J))/2.0)*((TOLD(I+1,J)-
      1 TOLD(I,J))-
      2 (COLD(I,J)+COLD(I-1,J))/2.0)*((TOLD(I,J)-
      3 TOLD(I-1,J)))*CA(I,J)
       CG1O=((COLD(I,J+1)+COLD(I,J))/2.0)*((TOLD(I,J+1)-
      1 TOLD(I,J))-
      2 (COLD(I,J)+COLD(I,J-1))/2.0)*((TOLD(I,J)-
      3 TOLD(I,J-1)))*CG(I,J)
       TEXPL=(CA1O-CB1O-CB2O+CG1O)*(1.0-RATIO)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
       CB1N=((((CNEW(I+1,J+1)+CNEW(I+1,J-1))/2.0)*((TNEW(I+1,J+1)-
      1 TNEW(I+1,J-1))/2.0)-
      2 ((CNEW(I-1,J+1)+CNEW(I-1,J-1))/2.0)*((TNEW(I-1,J+1)-
      3 TNEW(I-1,J-1))/2.0))/2.0)*CB(I,J)
       CB2N=((((CNEW(I+1,J+1)+CNEW(I-1,J+1))/2.0)*((TNEW(I+1,J+1)-
      1 TNEW(I-1,J+1))/2.0)-
      2 ((CNEW(I+1,J-1)+CNEW(I-1,J-1))/2.0)*((TNEW(I+1,J-1)-
      3 TNEW(I-1,J-1))/2.0))/2.0)*CB(I,J)
       CA1N=((CNEW(I+1,J)+CNEW(I,J))/2.0)*((TNEW(I+1,J))-
      1 (CNEW(I,J)+CNEW(I-1,J))/2.0)*((-
      2 TNEW(I-1,J)))*CA(I,J)
```

```
      CG1N=((CNEW(I,J+1)+CNEW(I,J))/2.0)*((TNEW(I,J+1))-
     1 (CNEW(I,J)+CNEW(I,J-1))/2.0)*((-
     2 TNEW(I,J-1)))*CG(I,J)
      TIMPL=(CA1N-CB1N-CB2N+CG1N)*RATIO
      COEFT=(CA(I,J)*((CNEW(I+1,J)+CNEW(I,J))/2.0)
     1 +CA(I,J)*((CNEW(I-1,J)+CNEW(I,J))/2.0)
     2 +CG(I,J)*((CNEW(I,J+1)+CNEW(I,J))/2.0)
     3 +CG(I,J)*((CNEW(I,J-1)+CNEW(I,J))/2.0))*RATIO
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     WRITE(6,905)HOLD(I,J),TEXPL,TIMPL
C     WRITE(6,999)
      IF(HOLD(I,J).LT.HSMELT)GOTO 701
      IF(HOLD(I,J).GT.HSMELT.AND.HOLD(I,J).LT.HLMELT)GOTO 702
      IF(HOLD(I,J).GT.HLMELT)GOTO 703
      OUT=0.0
  701 COEFS=(CJ(I,J)*CJ(I,J)*DENSTY*SPHTS)/TSTEP
      TITER=(TEXPL+TIMPL+COEFS*TOLD(I,J))/(COEFS+COEFT)
      IF(TITER.LT.TSTART)TITER=TSTART
      HITER=SPHTS*(TITER-TMELT)
      CITER=CONDS
C     IF(ITRIP.LT.2)GOTO 621
C     WRITE(6,906)I,J
C     WRITE(6,907)
C     WRITE(6,999)
C     WRITE(6,908)TITER,HITER,CITER
C     WRITE(6,999)
C 621 CONTINUE
      OUT=OUT+1
      IF(OUT.GT.100)GOTO 805
      IF(HITER.LT.HSMELT)GOTO 800
  702 COEFH=(CJ(I,J)*CJ(I,J)*DENSTY)/TSTEP
      HITER=HOLD(I,J)+(TEXPL+TIMPL-COEFT*TNEW(I,J))/COEFH
      TITER=TMELT
      CITER=CONDS-(CONDS-CONDL)*((HITER-HSMELT)/(HLMELT-HSMELT))
```

```
C       ITRIP=3
C       WRITE(6,906)I,J
C       WRITE(6,909)
C       WRITE(6,999)
C       WRITE(6,908)TITER,HITER,CITER
C       WRITE(6,999)
        OUT=OUT+1
        IF(OUT.GT.100)GOTO 805
        IF(HITER.GT.HSEMEL.AND.HITER.LT.HLMELT)GOTO 800
        IF(HITER.LT.HSEMEL)GOTO 701
  703   COEFL=(CJ(I,J)*CJ(I,J)*DENSTY*SPHTL)/TSTEP
        TITER=(TEXPL+TIMPL+COEFL*TOLD(I,J))/(COEFL+COEFT)
        HITER=HLMELT+(TITER-TMELT)*SPHTL
        CITER=CONDL
C       WRITE(6,906)I,J
C       WRITE(6,910)
C       WRITE(6,999)
C       WRITE(6,908)TITER,HITER,CITER
C       WRITE(6,999)
        OUT=OUT+1
        IF(OUT.GT.100)GOTO 805
        IF(HITER.GT.HLEMEL)GOTO 800

        IF(HITER.GT.HLEMEL)GOTO 800
        IF(HITER.LT.HLMELT)GOTO 701
  805   WRITE (6,901)
  901   FORMAT(' ','CONVERGENCE NOT ACHIEVED FOR A NODE IN THE MELT REGION
       1')
        IF(OUT.GT.100)GOTO 810
  800   HER=ABS(HITER-HNEW(I,J))
        HNEW(I,J)=HITER
        TNEW(I,J)=TITER
        CNEW(I,J)=CITER
        IF(HER.GT.MAXER)MAXER=HER
```

```
C      WRITE(6,911)TNEW(I,J),HNEW(I,J),CNEW(I,J),HER
C      WRITE(6,999)
  341 CONTINUE
  340 CONTINUE
C
C
C   ****************************************************************
C   * COMPUTATION THROUGH TEMPERATURE FIELD ALONG RE-ENTRANT       *
C   *       BOUNDARY                                               *
C   ****************************************************************
C
C
C
C   ****************************************************************
C   * CHECK FOR CONVERGENCE REQUIREMENT                            *
C   ****************************************************************
C
C
      ICOUNT=ICOUNT+1
      IF(MAXER.GT.0.001 )GOTO 350
  810 CONTINUE
  902 FORMAT(' ','DELKXO = ',E12.4,' DELKXN = ',E12.4,' DELTXO = ',
     1E12.4,' DELTXN = ',E12.4)
  903 FORMAT(' ','DELKYO = ',E12.4,' DELKYN = ',E12.4,' DELTYO = ',
     1E12.4,' DELTYN = ',E12.4)

  904 FORMAT(' ','A SET OF COMPUTATIONS IN THE MELT SUBROUTINE IS COMMEN
     1CING')
  905 FORMAT(' ','HOLD(I,J) = ',E12.4,' TEXPL = ',E12.4,' TIMPL = ',
     1E12.4)
  906 FORMAT(' ','FOR NODE I = ',I3,', J = ',I3,' THE FOLLOWING COMPUTAT
     1IONS APPLY:')
```

```
907 FORMAT(' ','THE SOLID ALGORITHM IS BEING USED')
908 FORMAT(' ','TITER = ',E12.4,' HITER = ',E12.4,' CITER = ',
   1E12.4)
909 FORMAT(' ','THE MELT ALGORITHM IS BEING USED')
910 FORMAT(' ','THE LIQUID ALGORITHM IS BEING USED')
911 FORMAT(' ','TNEW(I,J) = ',E12.4,' HNEW(I,J) = ',E12.4,
   1' CNEW(I,J) = ',E12.4,' HER = ',E12.4)
912 FORMAT(' ','IMAX = ',I6,'  JMAX = ',I6,'  DIFFUS = ',E12.4,
   1'  CONDUC = ',E12.4)
913 FORMAT(' ','GENRAT = ',E12.4,'  RATIO = ',E12.4,'  TSTEP = ',
   1E12.4,'  ICOUNT = ',I6)
914 FORMAT(' ','HSMELT = ',E12.4,'  HLMELT = ',E12.4,'  SPHTS = ',
   1E12.4,'  SPHTL = ',E12.4)
915 FORMAT(' ','CONDS = ',E12.4,'  CONDL = ',E12.4,'  TMELT = ',
   1E12.4,'  DENSTY = ',E12.4)
999 FORMAT(' ','       ')
    RETURN
    END
```

APPENDIX G

Computer Subroutine for Transformed Conductive Boundary Condition where One Layer's Boundary Undergoes a Change of Phase

```
      SUBROUTINE BNDRY9 (TU,BU,GU,JU,CU,TL,BL,GL,JL,CL,IMAX,JMAX,
     1ITERAT,TMELT,HNEW,CONDS,CONDL,SPHTS,SPHTL,HSMELT,HLMELT)
      REAL MAXER,JL,JU
      DIMENSION TU(121,30),BU(121,30),GU(121,30),JU(121,30),
     1 TL(121,30),BL(121,30),GL(121,30),JL(121,30),CU(121,30),
     2 HNEW(121,30)
C
C
C    ****************************************************************
C    * THIS SUBROUTINE SOLVES FOR THE INTERFACE BOUNDARY            *
C    *        CONDITION:                                            *
C    *                    -K1(D(T1)/D(N1)) = -K2(D(T2)/D(N2))       *
C    *                                                              *
C    * LATERAL HEAT FLUX AND HEAT GENERATION ARE NOT                *
C    *        ACCOUNTED FOR IN THE ALGORITHM                        *
C    *                                                              *
C    ****************************************************************
C
C
      ITERAT=0
C
C
C    ****************************************************************
C    * SWEEP THROUGH BOUNDARY POINTS NOT INCLUDING END POINTS       *
C    ****************************************************************
C
C
   30 MAXER=0.0
      KIMAX=IMAX-1
      DO 404 I=1,IMAX
      IF(TU(I,1).LT.TMELT)GOTO 401
      IF(TU(I,1).EQ.TMELT)GOTO 402
      IF(TU(I,1).GT.TMELT)GOTO 403
```

```
  401 HNEW(I,1)=SPHTS*(TU(I,1)-TMELT)
      CU(I,1)=CONDS
      GOTO 404
  402 HNEW(I,1)=(HSMELT+HLMELT)/2.0
      CU(I,1)=(CONDS+CONDL)/2.0
      GOTO 404
  403 HNEW(I,1)=HLMELT+(TU(I,1)-TMELT)*SPHTL
      CU(I,1)=CONDL
  404 CONTINUE
      DO 40 K=41,81
      B1=(CU(K,1)*GU(K,1))/(JU(K,1)*SQRT(GU(K,1)))
      B2=(CU(K,1)*BU(K,1)*(TU(K+1,1)-TU(K-1,1))/2.0)/(JU(K,1)
     1 *SQRT(GU(K,1)))
      B3=(CL*GL(K,JMAX))/(JL(K,JMAX)*SQRT(GL(K,JMAX)))
      B4=(CL*BL(K,JMAX)*(TL(K+1,JMAX)-TL(K-1,JMAX))/2.0)/(JL(K,JMAX)*
     1SQRT(GL(K,JMAX)))
      TITER=(B1*TU(K,2)-B2+B3*TL(K,JMAX-1)+B4)/(B1+B3)
C     TITER=TU(K,1)+1.5*(TITER-TU(K,1))
      TER=ABS((TU(K,1)-TITER))
      IF(TER.GT.MAXER)MAXER=TER
      TU(K,1)=TITER
   40 TL(K,JMAX)=TITER
C
C
C    ***************************************************************
C    * EVALUATION OF ENDPOINTS                                     *
C    ***************************************************************
C
C
C     B1=(CU(IMAX,1)*GU(IMAX,1))/(JU(IMAX,1)*SQRT(GU(IMAX,1)))
C     B2=(CU(IMAX,1)*BU(IMAX,1)*(TU(2,1)-TU(IMAX-1,1))/2.0)/(JU(IMAX,1)*
C    1SQRT(GU(IMAX,1)))
C     B3=(CL*GL(IMAX,JMAX))/(JL(IMAX,JMAX)*SQRT(GL(IMAX,JMAX)))
C     B4=(CL*BL(IMAX,JMAX)*(TL(2,JMAX)-TL(IMAX-1,JMAX))/2.0)/
C    1(JL(IMAX,JMAX)*SQRT(GL(IMAX,JMAX)))
```

```
C      TITER=(B1*TU(IMAX,2)-B2+B3*TL(IMAX,JMAX-1)+B4)/(B1+B3)
C      TER=ABS((TU(IMAX,1)-TITER))
C      IF(TER.GT.MAXER)MAXER=TER
C      TU(IMAX,1)=TITER
C      TL(IMAX,JMAX)=TITER
C      TU(1,1)=TITER
C      TL(1,JMAX)=TITER
       ITERAT=ITERAT+1
       IF(MAXER.GT.0.001 )GOTO 30
  219 FORMAT(' ',' MAXER PREVIOUS STEP =  ',F10.4)
  220 FORMAT(' ',' MAXER FINAL =  ',F10.4)
  221 FORMAT(' ',11F10.4)
  230 FORMAT(' ',' TU(K,1)=  ',F10.4,'  TITER =  ',F10.4,'  TER =  ',
     1F10.4,'  MAXER =  ',F10.4)
  240 FORMAT(' ','  B1 = ',F10.4)
  241 FORMAT(' ','  B2 = ',F10.4)
  242 FORMAT(' ','  B3 = ',F10.4)
  243 FORMAT(' ','  B4 = ',F10.4)
  244 FORMAT(' ','  TITER = ',F10.4)
  604 FORMAT(' ','CU = ',F10.4,' GU(K,1) = ',F10.4,' JU(K,1) = ',F10.4)
  614 FORMAT(' ','CU = ',F10.4,' BU(K,1) = ',F10.4,' TU(K+1,1) = ',
     1F10.4)
  615 FORMAT(' ','TU(K-1,1) = ',F10.4,' JU(K,1) = ',F10.4,' GU(K,1) = ',
     1F10.4)
  624 FORMAT(' ','CL = ',F10.4,' GL(K,JMAX) = ',F10.4,' JL(K,JMAX) = ',
     1F10.4)
  634 FORMAT(' ','CL = ',F10.4,' BL(K,JMAX) = ',F10.4,' TL(K+1,JMAX) = '
     1,F10.4)
  635 FORMAT(' ','TL(K-1,JMAX) = ',F10.4,' JL(K,JMAX) = ',F10.4,
     1' GL(K,JMAX) = ',F10.4)
  800 FORMAT(' ','FOR NODE ',I3)
       RETURN
       END
```

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-4194		

4. Title and Subtitle	5. Report Date
A Numerical Simulation of the Full Two-Dimensional Electrothermal De-Icer Pad	November 1988
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Konstanty C. Masiulaniec	None (E-4381)
	10. Work Unit No. 505-68-11

9. Performing Organization Name and Address	11. Contract or Grant No.
The University of Toledo Department of Engineering Science Toledo, Ohio 43606	NAG3-72
	13. Type of Report and Period Covered Contractor Report Final

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	

15. Supplementary Notes

Project Manager, Mario Vargas, Propulsion Systems Division, NASA Lewis Research Center. This report was a dissertation submitted in partial fulfillment of the requirements for the Doctor of Philosophy Degree in Engineering Science in June 1987.

16. Abstract

The ability to predict the time-temperature history of electrothermal de-icer pads is important in the subsequent design of improved and more efficient versions. These de-icer pads are installed near the surface of aircraft components, for the specific purpose of removing any accreted ice. The proposed numerical model can incorporate the full two-dimensional geometry through a section of a region (i.e., section of an airfoil, etc.), that current one-dimensional numerical codes are unable to do. Thus, the effects of irregular layers, curvature, etc., can now be accounted for in the thermal transients. Each layer in the actual geometry is mapped via a body-fitted coordinate transformation into uniform, rectangular computational grids. The relevant heat transfer equations are transformed and discretized. To model the phase change that might occur in any accreted ice, in an enthalpy formulation the phase change equations are likewise transformed and discretized. The code developed was tested against numerous classical numerical solutions, as well as against experimental de-icing data on a UH1H rotor blade obtained from the NASA Lewis Research Center in Cleveland, Ohio. The excellent comparisons obtained show that this code can be a useful tool in predicting the performance of current de-icer models, as well as in the designing of future models.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
De-Icer	Unclassified – Unlimited Subject Category 06

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified	208	A10

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1988

National Aeronautics and
Space Administration
Code NIT-4
Washington, D.C.
20546-0001

Official Business
Penalty for Private Use, $300

NASA

NASA

National Aeronautics and
Space Administration

Washington, D.C.
20546

SPECIAL FOURTH CLASS MAIL
BOOK

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

Official Business
Penalty for Private Use $300

L: 001 CR-4194 890111S0905694
NASA
SCIEN & TECH INFO FACILITY
ACCESSIONING DEPT
P O BOX 8757 BWI ARPRT
BALTIMORE MD 21240