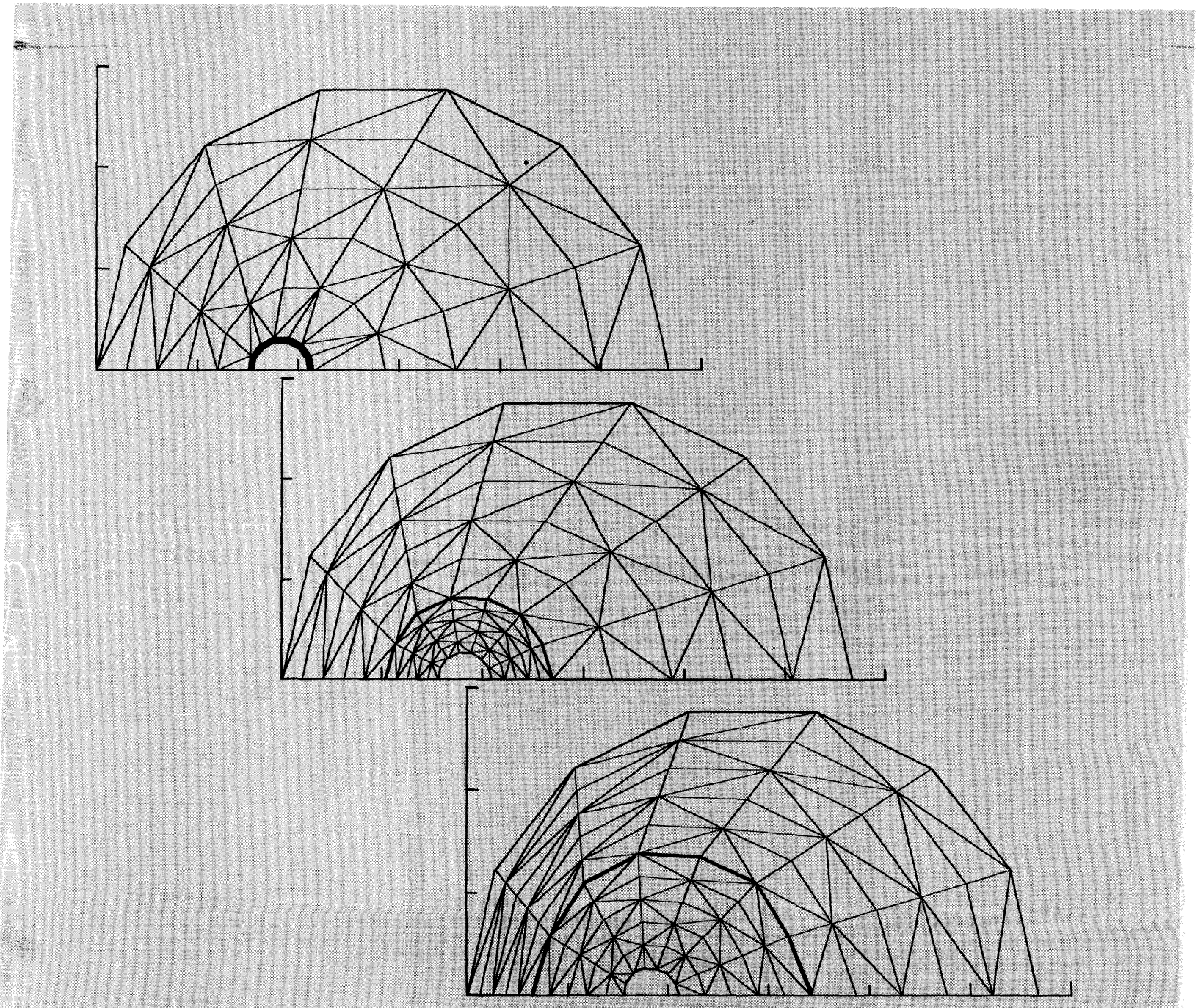
CRREL
REPORT 84-10

US Army Corps of Engineers
Cold Regions Research & Engineering Laboratory

Modeling two-dimensional freezing using transfinite mappings and a moving-mesh finite element technique



CRREL Report 84-10

May 1984

Modeling two-dimensional freezing using transfinite mappings and a moving-mesh finite element technique

Mary Remley Albert

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: CRREL Report 84-10	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): MODELING TWO-DIMENSIONAL FREEZING USING TRANSFINITE MAPPINGS AND A MOVING-MESH FINITE ELEMENT TECHNIQUE	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Mary Remley Albert	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: U.S. Army Cold Regions Research and Engineering Laboratory, Hanover, New Hampshire 03755	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: U.S. Army Cold Regions Research and Engineering Laboratory, Hanover, New Hampshire 03755	12. REPORT DATE: May 1984
	13. NUMBER OF PAGES: 54
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Freezing
Finite element analysis
Mathematical models
Transfinite mappings

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Freezing phase change problems in conduction heat transfer represent a set of moving boundary problems for which much interest currently exists. In the work presented here, two-dimensional freezing is modeled by incorporating the use of transfinite mappings with a moving-mesh finite element technique. The use of transfinite mapping in governing interior mesh motion is shown to provide very acceptable results and is demonstrated to be the most efficient general computational technique used to date. The model developed is capable of using either Cartesian or (r,z) cylindrical coordinates. Both frozen and unfrozen phases may be modeled when conduction governs behavior in both. In the case of freezing of a fluid as it flows through a pipe the usefulness of always having the phase boundary coincident with element boundaries is demonstrated. Results of the model are shown to compare well with analytical and experimental

20. Abstract (cont'd).

results. A von Neumann stability analysis is performed for the numerical solution and tends to support the observation that the occurrence of a high Peclet number in the moving-mesh model of heat conduction may produce distortions of the numerical solution.

PREFACE

This report was prepared by Mary Remley Albert, Mathematician, Applied Research Branch, Experimental Engineering Division, U.S. Army Cold Regions Research and Engineering Laboratory. This work was sponsored by Independent Laboratory In-House Research at CRREL.

The author thanks Dr. Kevin O'Neill for the hours spent in consultation for this work. His advice and guidance are truly appreciated, as is his careful technical review. Also, thanks are extended to Dr. Daniel Lynch of the Thayer School of Engineering at Dartmouth College for his constructive criticism and technical review. The author thanks Dr. George Ashton for advice on the heat transfer coefficients used in the pipe simulation.

CONTENTS

ILLUSTRATIONS

Figure

MODELING TWO-DIMENSIONAL FREEZING USING TRANSFINITE MAPPINGS AND A MOVING-MESH FINITE ELEMENT TECHNIQUE

Mary Remley Albert

INTRODUCTION

Freezing phase change problems in heat transfer represent a set of moving boundary problems for which there is wide application. Ground freezing and freezing in pipes are two of many areas in which two-dimensional freezing may occur; they represent problems for which available analytical solutions are severely limited. Therefore, a numerical solution is sought. In these heat conduction problems the latent heat at the phase interface represents a discontinuity in the governing equations, for which special numerical procedures must be adopted. A variety of numerical techniques have been presented in recent years that attempt to deal with this discontinuity.

One of the most sophisticated techniques involves the use of finite elements where the mesh moves continuously during the solution of the problem, so that the phase interface will always coincide with element boundaries (Lynch and O'Neill 1981, O'Neill and Lynch 1981, Lynch 1982a). This method was chosen as the basis of the model presented here because it provides a smooth and accurate means of tracking the phase front and calculating temperature distributions. In addition, because element and phase boundaries always coincide, the method allows clear specification of conditions at the phase boundary. This is essential in cases where there is flow in the unfrozen region.

An important problem associated with the moving mesh method involves the specification of movement of interior nodes during the simulation. Because the phase boundary may travel large distances and undergo a significant change in shape in the course of the solution, the interior nodes of the mesh must also move to keep the mesh in a reasonable geometrical form. The method to specify the evolution of the interior mesh should be capable of generating interior node locations given a minimum amount of information, it should specify the mesh so that the mesh does not get tangled, and the computations involved should be capable of automation within the usual finite element time stepping procedure (Lynch 1982a). In the work presented here, recent advances in automatic mesh generation are investigated as a new means of specifying the interior mesh for each time step. A two-dimensional finite element program is developed using transfinite mappings in conjunction with a moving mesh. The program uses linear triangular elements and is able to model either Cartesian or (r,z) cylindrical coordinates. Solutions obtained from the program are shown to compare well with analytical solutions.

The method is applied to two problems of practical interest. It is used to model a two-dimensional situation involving freezing, where experimental results are available, and the numerical and

experimental results compare very well. Also, it is used to model two-dimensional freezing in a pipe, where flow through the pipe is driven by a head drop.

While verifying the method it was discovered that numerical distortions of the temperature solution sometimes occur when a region of the mesh containing large elements is moved at a high velocity. This is a new finding with respect to moving mesh methods used for heat conduction, and is due to the occurrence of a large Peclet number in these circumstances. Other researchers have observed that large Peclet numbers yield numerical difficulties in modeling convection on a fixed mesh. This effect is discussed further in the context of a von Neumann stability analysis.

PREVIOUS WORK

The heart of this report lies in the marriage of the transfinite mapping method of automatic mesh generation with the moving-mesh finite element technique. In the past, automatic mesh-generation techniques have been used primarily as a method of initial input for conventional finite element analysis, where the mesh does not move. The moving-mesh finite element technique has been used to model phase change, but never in conjunction with the use of automatic mesh-generation techniques. Thus, the discussion of previous work must be divided into two sections: 1) numerical methods that have been used to model phase change in conduction heat transfer, and 2) automatic mesh-generation techniques.

Numerical methods used to model phase change

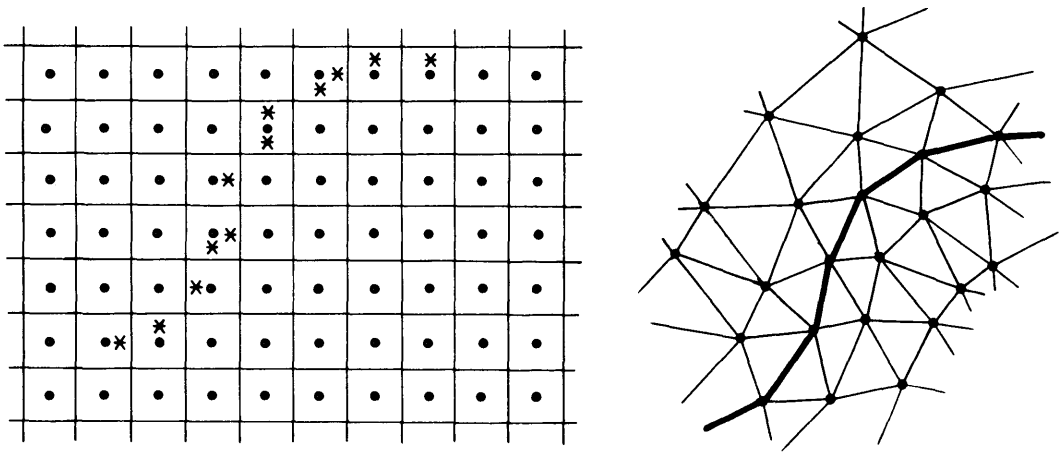
The problem of modeling heat conduction with a freeze-thaw change of phase has been addressed by many analysts; consequently many methods exist for solving the problem. The brief review here will consider only methods that may be used in modeling phase change in two or three dimensions.

There are two general types of approach used for the problem. The first type solves for conduction heat transfer using traditional fixed-mesh finite elements or finite differences where the phase front progresses through the stationary mesh. The discontinuity at the phase boundary is handled by adjusting the heat capacity or enthalpy condition in elements containing the boundary. The second approach involves the use of a mesh in which the phase boundary always lies on particular numerical boundaries, such as element boundaries, and the latent heat condition is imposed on that boundary.

The most basic method using the first approach is the "excess degrees" method (Dusinberre 1949). Here the heat conduction equation is solved as usual, and an enthalpy budget is maintained for nodes along the phase front. The temperature of these nodes is held constant until the heat equivalent to the latent heat has accumulated; the front is then allowed to progress to the next set of nodes. Thus, the front jumps from node to node. The "apparent heat capacity" method (Bonacina and Comini 1973) is similar, except that no enthalpy budget is maintained. Instead, nodes representing the phase front have their heat capacity redefined so that the effect of the latent heat is included. The location of the phase front is not distinct but is usually estimated by interpolating temperatures. This results in a step-like approximation of the front location in time. Although relatively crude, methods where the phase front progresses through a stationary mesh have the advantage that multiple locations of the phase front are possible, and phase fronts may appear and disappear during the course of the solution, making the approach flexible. Figure 1 illustrates the mesh and phase front locations in a fixed mesh approach (apparent heat capacity, for example) and in a moving mesh approach.

The second, more sophisticated approach involves the use of a mesh in which the phase boundary always lies on particular numerical boundaries. Here the location of the phase front progresses smoothly in time. Having the phase front located on element boundaries permits the method to be used in the more general situation where convection may occur in the unfrozen region.

The "isotherm migration" method was developed by Crank and Prahle (1973). It involves a coordinate transformation on the temperature and one of two space variables, so that the location of the phase change isotherm coincides with numerical boundaries. The problem is then solved on a fixed mesh. Due to the limitations of the boundary conditions and the temperature distribution



a. Use of apparent heat capacity method with finite differences. Solid dots represent nodes, and stars represent the interpolated location of the phase front.

b. Use of moving mesh with triangular finite elements. The phase front lies along the highlighted element boundaries.

Figure 1. Two approaches for determining phase front locations.

on the boundary, this method is not completely general. Sparrow et al. (1977) perform a coordinate transformation on spatial variables in the (r,z) cylindrical coordinate system, so that the fixed finite difference mesh has element boundaries that always lie on the phase boundary. Computations are performed on a fixed mesh in the transformed space, and the dependent variable is interpolated in progressing from the mesh for one time step to the mesh in the next time step. Saitoh (1978) expands the one-dimensional spatial coordinate transformation presented by Landau (1950) to two and three dimensions, using an (r,θ) cylindrical coordinate system. Finite differences on a fixed mesh are used, and the phase boundary is fixed with respect to the transformed coordinates. While his method is general in theory, the form of the governing equation becomes very complicated, and special consideration is required in the formulation when boundaries are not smoothly shaped. O'Neill (in press) analyzes phase change using the boundary integral equation method, keeping the phase front location stabilized on a mesh. Since the temperature distribution within each phase has a steady-state profile, this represents a special case.

Keeping the idea of maintaining the phase boundary on particular element boundaries, Bonnerot and Jamet (1977), Lynch and O'Neill (1981), O'Neill and Lynch (1981) and Lynch (1982a) use finite elements on a mesh that moves in time so that the phase front always coincides with certain element boundaries. Numerical computations are performed in the original coordinate system. Bonnerot and Jamet (1977) use finite elements in both space and time, where the effect of finite element formulation in time automatically accounts for the effects of mesh deformation. Lynch and O'Neill (1981) use finite differences in time, and mesh motion effects appear as a velocity term in the governing equation. In Lynch and O'Neill (1981) Hermite basis functions are used to solve one-dimensional problems. In O'Neill and Lynch (1981) these results are compared to results obtained using linear basis functions. Lynch (1982a) uses the method to model phase change in Cartesian coordinates in two dimensions, where interior node movement is specified by the recurrent solution of the equations of elasticity. Lynch shows that the approaches taken by Bonnerot and Jamet, Saitoh, Lynch and O'Neill, and O'Neill and Lynch are mathematically the same, although Saitoh's approach may seem different conceptually.

Automatic mesh-generation techniques

Automatic mesh-generation techniques have been employed to generate initial finite element meshes for use in any kind of finite element model where the mesh is stationary. In this work it has found a new use. At each time step in the solution of the phase change problem, a new location of the phase boundary and possible reassignment of node locations on other boundaries give rise to the need for a new interior mesh to be specified. Automatic mesh-generation techniques

were investigated as a means of specifying new interior node locations because these techniques appear to provide a means of specifying a new mesh in a smooth, unique and simple manner. In the current literature on mesh generation there are three prominent methods: 1) Laplacian schemes, 2) use of isoparametric coordinates, and 3) use of transfinite mappings.

The Laplacian scheme defines the location of interior nodes as

$$p_i = \tfrac{1}{4}(p_{i_1} + p_{i_2} + p_{i_3} + p_{i_4})$$

where p_i represents the coordinate x_i or y_i of the ith node on the interior of the grid, and p_{i_j} represents the coordinate of the jth node directly connected by an element boundary to the ith node. This can be interpreted as the use of the Laplacian finite difference operator on a rectangular grid (Buell and Bush 1973). There are variations of the method. For example, Herrmann (1976) extends the method to more general grid types and combines the use of the Laplacian scheme with isoparametric mappings. Denayer (1978) presents techniques for generating element connectivity and improving the computational efficiency of the method. Laplacian methods yield meshes in which nodes are fairly evenly spaced within the region (not always a desired effect). They may require excessive computation in generating a mesh recurrently during the solution of a problem.

The other two methods use a mapping of the unit square onto the region to be described by a mesh. This unit square may be easily divided by straight horizontal and vertical lines, producing a checkerboard pattern; this system of subdivisions can be mapped back to the original region. A checkerboard requires only a simple programmed algorithm to generate coordinate and incidence lists.

The use of isoparametric coordinates to generate a mesh is an extension of the use of isoparametric coordinates for curvilinear finite element analysis (Zienkiewicz and Phillips 1971). Shape functions are associated with each of the boundary nodes, and the interior coordinates are a linear combination of the product of each boundary node and its shape function. A major restriction with the method is that each side of the zone must be smooth; slope discontinuities may occur only at the corners.

A method similar to the use of isoparametric coordinates, but more general, is the transfinite mapping method discussed by Haber et al. (1981) and Gordon and Hall (1973). This method is so-named because it generates meshes that will match the boundaries of the problem at a non-denumerable number of points. Perhaps its greatest advantage over the use of isoparametric coordinates is that slope discontinuities may occur anywhere along any boundary. This method will be discussed in more detail in a following section.

FINITE ELEMENT FORMULATION

The program developed here is capable of handling two-dimensional Cartesian or radial (r,z) coordinates. The following theory applies to both cases. The equation to be solved in each phase is the heat conduction equation

$$C\frac{\partial T}{\partial t} = \underline{\nabla} \cdot (k \underline{\nabla} T) \tag{1}$$

with the phase interface boundary condition

$$L\frac{d\underline{s}}{dt} = (k\underline{\nabla}T)_f - (k\underline{\nabla}T)_u \, . \tag{2}$$

Here T = temperature
t = time
$\underline{s}$ = location of the phase interface
C = volumetric (sensible) heat capacity
L = volumetric latent heat
k = thermal conductivity.

The subscripts f and u refer to the frozen and unfrozen zones, respectively. The method of solution will use eq 2 to specify the new location of the front at each time step, and eq 1 to solve for the temperature distribution in each phase.

Heat conduction equation on a moving mesh

The finite element formulation is obtained by applying the Galerkin method to eq 1. Multiply each term by the weighting functions ϕ_i and integrate:

$$<\underline{\nabla}\cdot(k\underline{\nabla}T)\phi_i> - <C\frac{\partial T}{\partial t}\phi_i> = 0 \,. \tag{3}$$

Using Green's theorem (or integrating the first term by parts) yields

$$<\underline{\nabla}\cdot(k\,\underline{\nabla}T)\phi_i> = -<k\,\underline{\nabla}T\cdot\underline{\nabla}\phi_i> + \oint_\Gamma k\phi_i\underline{\nabla}T\cdot\mathbf{n}\,d\gamma \tag{4}$$

where $\oint_\Gamma d\gamma$ represents the integral over the boundary, $\mathbf{n}$ represents the unit vector normal to the boundary, and $<>$ represents integration over the area of the region. This result is substituted into eq 3:

$$<k\underline{\nabla}T\cdot\underline{\nabla}\phi_i> + <C\frac{\partial T}{\partial t}\phi_i> = \oint_\Gamma k\phi_i\underline{\nabla}T\cdot\mathbf{n}\,d\gamma \,. \tag{5}$$

Now let the temperature be approximated as follows:

$$T \approx \mathbf{T} = \sum_j^N T_j(t)\phi_j(x,y,t) \quad \text{for the Cartesian case}$$

$$T \approx \mathbf{T} = \sum_j^N T_j(t)\phi_j(r,z,t) \quad \text{for the radial case} \,.$$

Here ϕ_j stands for the basis, or interpolating, functions, which have been chosen to be the same as the weighting functions. Linear triangular elements are used in this model.

Now the approximation for T is substituted into eq 5. Note that the substitution into $\partial T/\partial t$ will give two terms:*

$$<T_j k\underline{\nabla}\phi_j\cdot\underline{\nabla}\phi_i> + <C\phi_i T_j\frac{\partial\phi_j}{\partial t}> + <C\phi_i\phi_j\frac{dT_j}{dt}> = \oint_\Gamma k\phi_i\underline{\nabla}T\cdot\mathbf{n}\,d\gamma \,. \tag{6}$$

All of the terms in the above equation are familiar from finite element formulation on a fixed mesh, except for the term containing $\partial\phi_j/\partial t$. Lynch (1982a) shows that

$$\frac{\partial\phi_j}{\partial t} = -\underline{V}\cdot\underline{\nabla}\phi_j \quad \text{where} \quad \underline{V} = \frac{d\underline{x}_i}{dt}\psi_i \,. \tag{7}$$

Here $\underline{x}_i$ represents the coordinate of a node with respect to a fixed reference frame, and ψ_i interpolates between the nodes. $\underline{V}$ represents mesh velocity. For the case of linear triangular finite elements considered here, $\psi_i = \phi_i$. Equation 6 becomes

$$<C\phi_i\phi_j>\frac{\partial T_j}{\partial t} + <k\underline{\nabla}\phi_j\cdot\underline{\nabla}\phi_i>T_j - <C\phi_i\underline{V}\cdot\underline{\nabla}\phi_j>T_j = \oint_\Gamma k\phi_i\underline{\nabla}T\cdot\mathbf{n}\,d\gamma \,. \tag{8}$$

*Note that here and throughout the rest of the text, repeated indices denote summation unless otherwise specified.

This result has been presented in Lynch and O'Neill (1981), O'Neill and Lynch (1981) and Lynch (1982a).

Note that this formulation has added a convection term to the governing differential equation. This apparent convection is due to the movement of the mesh. Accounting for the mesh movement in the governing equation in this manner assures that the temperature associated with each node takes the movement of the node into account mathematically.

Equation 8 is now integrated over time. To cast eq 8 into its matrix formulation, the following matrices are defined:

$$\underline{\underline{M}} = \langle C\phi_i\phi_j \rangle \tag{9}$$

$$\underline{\underline{K}} = \langle k\underline{\nabla}\phi_j \cdot \underline{\nabla}\phi_i \rangle - \langle C\phi_i \underline{V} \cdot \underline{\nabla}\phi_j \rangle \tag{10}$$

$$\underline{r} = \oint_\Gamma k\phi_i \, \underline{\nabla} T \cdot \mathbf{n} \, d\gamma \, . \tag{11}$$

The integral over time is

$$\int_{t^n}^{t^{n+1}} [\underline{\underline{M}}\dot{\underline{T}} + \underline{\underline{K}}\underline{T} = \underline{r}] dt \quad \text{where} \quad \dot{\underline{T}} = \frac{d}{dt}(\underline{T}) \, . \tag{12}$$

Because matrices $\underline{\underline{M}}$ and $\underline{\underline{K}}$ are mesh dependent and the mesh changes with time, the matrices are time dependent. This situation represents a nonlinearity in the problem. For the purpose of integration in time, both are regarded as constant over one time step but are updated each time step. All of the elements of these matrices must pertain to a single mesh; the user is given the option of using the mesh at the beginning of the time step, the mesh after the phase front is moved, or a mesh that is interpolated at any level between those two meshes. The integral over time is evaluated by the use of finite differences, and yields

$$\underline{\underline{M}}(\underline{T}^{n+1} - \underline{T}^n) + \underline{\underline{K}} \, \Delta t[\epsilon \underline{T}^{n+1} + (1-\epsilon)\underline{T}^n] = \int \underline{r} dt$$

$$(\underline{\underline{M}} + \epsilon \Delta t \underline{\underline{K}})\underline{T}^{n+1} = [\underline{\underline{M}} - (1-\epsilon)\Delta t \underline{\underline{K}}] \, \underline{T}^n + \int \underline{r} dt \, . \tag{13}$$

If ϵ=0, the equation is explicit; if ϵ=1, it is fully implicit. The resulting matrices are banded. In the program, only the elements of the band are stored, and the equation is solved in entirety for each time step using a banded matrix solver (Lynch 1982b). The matrices are not stationary.

Several points specific to the program presented here should be addressed now. First, in forming terms like $\langle C\phi_i\phi_j \rangle$ and $\langle k \underline{\nabla}\phi_j \cdot \underline{\nabla}\phi_i \rangle$ on a local level, it is assumed in the program that C and k are constant over any given element. Then, on a local level for Cartesian coordinates,

$$\langle \phi_i\phi_j \rangle = \int_x \int_y \phi_i\phi_j \, dxdy$$

$$\langle \underline{\nabla}\phi_j \cdot \underline{\nabla}\phi_i \rangle = \int_x \int_y \underline{\nabla}\phi_j \cdot \underline{\nabla}\phi_i \, dxdy \, . \tag{14}$$

In the (r,z) cylindrical coordinates the elements represent rings about the z-axis. Here the coordinate r is interpolated over an element by letting $r = r_k\phi_k$:

$$\langle \phi_i\phi_j \rangle = 2\pi \int_r \int_z \phi_i\phi_j \, rdrdz = 2\pi r_k \int_r \int_z \phi_i\phi_j\phi_k \, drdz$$

$$\langle \underline{\nabla}\phi_j \cdot \underline{\nabla}\phi_i \rangle = 2\pi \int_r \int_z \underline{\nabla}\phi_j \cdot \underline{\nabla}\phi_i \, rdrdz = 2\pi r_k \int_r \int_z \underline{\nabla}\phi_j \cdot \underline{\nabla}\phi_i\phi_k \, drdz. \tag{15}$$

All of these integrals are evaluated exactly.

Also, $\underline{V} \cdot \underline{\nabla}\, \phi_j$ may be written as $V_\ell\, \partial\phi_j/\partial x_\ell$, where ℓ is a coordinate direction and is incremented to two (i.e. $x_1 = x$ or r, and $x_2 = y$ or z). For linear triangular elements $\partial\phi_j/\partial x_\ell$ is constant over an element. Then, for an element

$$<C\phi_i \underline{V} \cdot \underline{\nabla}\phi_j> = C \frac{\partial \phi_j}{\partial x_\ell} V_\ell^m \int \phi_i \phi_m dA \tag{16}$$

where m is a node number and V_ℓ^m is calculated from node locations for the two meshes representing the region at the beginning and end of the time step.

Boundary conditions are handled as usual in finite element procedures. For constant temperature nodes the row representing that node in the global matrix is filled with zeros, except "1" is put on the diagonal and the temperature is placed in the vector $\underline{r}$. For nodes on zero flux boundaries, $r_i = 0$. The phase boundary is considered to be a Dirichlet boundary whose constant temperature is equal to the phase change temperature.

Phase boundary movement

The nodes on the phase boundary are moved each time step in accordance with eq 2. Since the numerical solution involves a discretized boundary, the equation cannot be exactly satisfied everywhere; Lynch (1982a) proposes using a weaker, integral form of the equation:

$$\int L\left(\frac{d\underline{s}}{dt}\right)_j \phi_j \cdot \mathbf{n}\, d\gamma = \int k_f \underline{\nabla} T_f \cdot \mathbf{n}\, d\gamma - \int k_u \underline{\nabla} T_u \cdot \mathbf{n}\, d\gamma$$

$$= \int k_f \underline{\nabla} T_f \cdot \mathbf{n} \Sigma\phi_j\, d\gamma - \int k_u \underline{\nabla} T_u \cdot \mathbf{n} \Sigma\phi_j\, d\gamma \tag{17}$$

where $\Sigma\phi_j = 1$. The index j refers to nodes on the phase boundary; the unit vector $\mathbf{n}$ points away from the frozen zone, and the integration is over the phase boundary. Again, subscripts u and f represent unfrozen and frozen regions; repeated f and u subscripts do not indicate summation. Now consider each boundary node j. Equation 17 may be written*

$$\left(\frac{d\underline{s}}{dt}\right)_j \cdot \int \mathbf{n}\phi_j d\gamma = \frac{1}{L} \int \left[\left(k \frac{\partial T}{\partial n}\right)_f - \left(k \frac{\partial T}{\partial n}\right)_u\right] \phi_j\, d\gamma\,. \tag{18}$$

The term $\partial T/\partial n$ is the heat flux normal to the phase boundary, and $(d\underline{s}/dt)_j$ specifies the movement of node j. Let v_j be the magnitude of the vector $(d\underline{s}/dt)_j$, and let $\mathbf{m}_j$, a unit vector, be its direction. For a node on a phase boundary the situation is depicted in Figure 2.

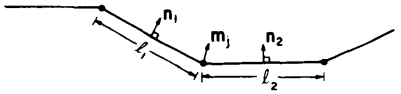


Figure 2. Movement of node j on a phase boundary.

Because ϕ_j has a value of one at node j and zero at other nodes, and is linear,

$$\int \mathbf{n}\phi_j d\gamma = \tfrac{1}{2} \ell_1 \mathbf{n}_1 + \tfrac{1}{2} \ell_2 \mathbf{n}_2 \tag{19}$$

where ℓ denotes an element side length, and the subscripts *1* and *2* refer to adjacent element sides on the phase boundary. Equation 18 will be used to solve for the magnitude of the velocity of the node v_j, but $\mathbf{m}_j$ is not specified inherently in the problem. Here it is assumed that $\mathbf{m}_j$ is a weighted average of the $\mathbf{n}_i$ for nodes where the phase boundary is described by adjacent elements on both sides; that is,

*No summation convention is to be used for any subscript.

$$\mathbf{m}_j = \frac{\underline{m}_j}{|\underline{m}_j|} \quad \text{where } \underline{m}_j = \frac{\ell_1 \mathbf{n}_1 + \ell_2 \mathbf{n}_2}{\ell_1 + \ell_2}. \tag{20}$$

Equation 18 becomes

$$\tfrac{1}{2} v_j (\ell_1 \mathbf{m}_j \cdot \mathbf{n}_1 + \ell_2 \mathbf{m}_j \cdot \mathbf{n}_2) = \frac{1}{L} \left\{ \tfrac{1}{2}\ell_1 \left[\left(k \frac{\partial T}{\partial n}\right)_{f_1} - \left(k \frac{\partial T}{\partial n}\right)_{u_1} \right] + \tfrac{1}{2}\ell_2 \left[\left(k \frac{\partial T}{\partial n}\right)_{f_2} - \left(k \frac{\partial T}{\partial n}\right)_{u_2} \right] \right\} \tag{21}$$

where, for example, $(k\,\partial T/\partial n)_{f_1}$ represents the heat flux across side *1* from the frozen zone. Solving for v_j yields

$$v_j = \frac{1}{L} \frac{\ell_1 \left[\left(k \frac{\partial T}{\partial n}\right)_{f_1} - \left(k \frac{\partial T}{\partial n}\right)_{u_1} \right] + \ell_2 \left[\left(k \frac{\partial T}{\partial n}\right)_{f_2} - \left(k \frac{\partial T}{\partial n}\right)_{u_2} \right]}{\ell_1 \mathbf{m}_j \cdot \mathbf{n}_1 + \ell_2 \mathbf{m}_j \cdot \mathbf{n}_2} . \tag{22}$$

Here, $\partial T/\partial n$ is evaluated as $\partial \mathrm{T}/\partial n$, from the numerical temperature distribution, and node *j* is moved a distance $\Delta s_j = v_j \Delta t$, where Δt is the time step.

For a node on the end of the phase boundary there is only one adjacent element, and $\mathbf{m}_j$ must be specified by the user of the program. Usually its specification will be made obvious by the other boundary conditions of the problem. For example, it may be a node on a line of symmetry, so that $\mathbf{m}_j$ will be directed along the line of symmetry.

The development of the specification of v_j for a node on the end of the phase boundary is analogous to the above discussion, except that all terms with subscript *1* are omitted if the node is on the left side of the boundary and terms with subscript *2* are omitted if it is on the right side.

In conduction problems the steepest temperature gradients are usually present at the beginning of the solution and tend to lessen in time. Since the movement of the phase front is proportional to temperature gradients, using the gradients at the beginning of the time step to project the front will usually result in its being projected too far. The situation may be improved by iterating on the location of the front for each time step. In the model presented here, one iteration is an option. That is, the user is given the choice of no iterations or iterating once. The iteration involves using the average of the temperature gradients at the beginning and end of the time step to reposition the front, rather than merely using the gradients at the beginning of the time step.

TRANSFINITE MAPPINGS

Since the phase boundary may travel large distances and undergo a significant change in shape in the course of the solution, the interior nodes of the mesh must also move in order to keep the mesh in a reasonable geometrical form. The method used in this work to accomplish this task involves the generation of a new mesh each time step, using transfinite mappings. In this section the use of transfinite mappings is investigated.

Haber et al. (1981) and Gordon and Hall (1973) describe the transfinite mapping in terms of projectors. First consider the "lofting" projector. For this projector the boundary curves $\psi_1(u)$ and $\psi_2(u)$ are specified (Fig. 3), and the projector performs a linear interpolation in one direction between the curves. The lofting projector is defined by the equation

$$P(u,v) = (1-v)\psi_1(u) + v\phi_2(u) \tag{23}$$

where u and v are normalized coordinates: $0 \leqslant u \leqslant 1$, $0 \leqslant v \leqslant 1$.

To interpolate in two directions, two lofting projectors may be combined to form a "bilinear" projector. As shown above, each lofting projector forms straight edges in one direction, thus

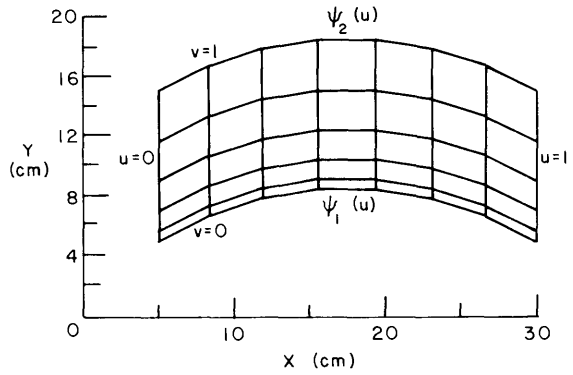


Figure 3. Use of the lofting projector.

introducing some mismatch or distortion of the true region. A Boolean combination of the lofting projectors produces the bilinear projector, which removes distortion. The bilinear projector represents a continuous mapping of the unit square in the transformed space onto the region to be meshed in F-space, where the region has four sides described by the curves $\psi_1(u)$, $\psi_2(u)$, $\xi_1(v)$ and $\xi_2(v)$, and four corners with coordinates $F(u,v)$, where u and v equal zero or one. The bilinear projector is defined as follows:

$$P_B(u,v) = (1-v)\psi_1(u) + v\psi_2(u) + (1-u)\xi_1(v) + u\xi_2(v) - (1-u)(1-v)F(0,0)$$

$$- (1-u)vF(0,1) - uvF(1,1) - u(1-v)F(1,0) \qquad (24)$$

where $0 \leqslant u \leqslant 1$, $0 \leqslant v \leqslant 1$. An example of the use of this projector is shown in Figure 4.

In practice a finite number of nodes is identified on each side; these represent discrete values of ψ and ξ. Thus, ψ and ξ need not be smooth functions or any known functions at all. It matters only that nodal coordinates are known at various places along the boundaries.

Gordon and Hall (1973) and Haber et al. (1981) show illustrations of meshes where excellent results are obtained by this method. Boundaries are accurately described by the mapping, the interior mesh reflects boundary shapes, and elements formed have good aspect ratios. Also, it is reported both in the isoparametric method (Zienkiewicz and Phillips 1971) and in the transfinite mapping method (Gordon and Hall 1973, Haber et al. 1981) that the use of highly distorted boundary curves may result in the generation of nodes that do not lie inside the region ("overspill"). (The recommended path to avoid that problem in two dimensions is to divide the region into one or more simpler regions over which meshes are generated separately.) In general, however, the method of transfinite mappings has been highly successful in creating acceptable meshes.

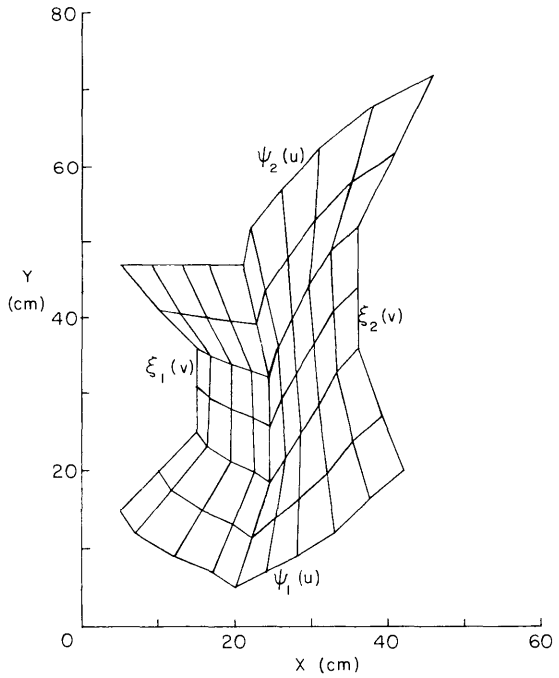


Figure 4. Use of the bilinear projector.

There are a number of subtleties and limitations of the method that are not discussed in the literature. The most straightforward use of transfinite mappings for mesh generation is obtained if interior node locations are assumed to lie at points that correspond to the intersection of lines of constant u and v in the transformed space of the unit square. This requires that opposite sides have the same number of nodes.

Often, more numerical detail will be desired in one section of the mesh than in others, so that the nodes in F-space are generally unequally spaced along the edges of the region. For eq 24 to be used, each boundary node location must be identified with values of u and v in the space of the unit square. Two ways of doing this were investigated: 1) to assign u to be a normalized distance along the edge in F-space and 2) to assign u, for example, as $u = i/N$, where the ith node along the edge is being considered and there are N elements along the edge.

Consider the specification of u and v by the first method, with the additional assumption that the normalization results in a set of nodes with u or v values that are the same on opposite sides. This results in a mapping where the unit square has been "stretched" relatively smoothly in the mapping. In this case, interior mesh lines will reflect boundary shapes (Fig. 5).

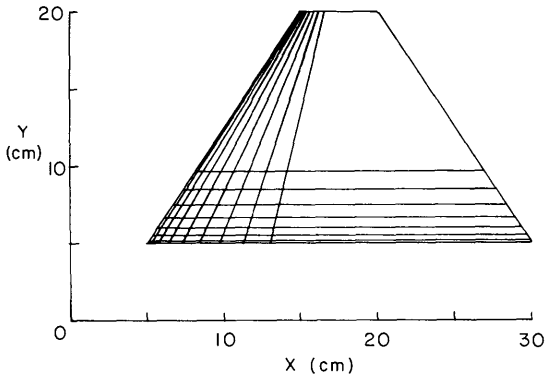


Figure 5. Use of the bilinear projector with method 1.

In general, however, arbitrary subdivision of the boundaries with method 1 will result in normalizations such that nodes on one side have u or v values that differ from the values along the opposite side. In this case, grid lines in F-space would not be identified with lines of constant u and v in the transformed space. Equation 24 could be used in a way that would accommodate this situation. However, while not conceptually difficult, this procedure would involve calculating u and v for each interior node, and would require ψ and ξ values to be specified for any choice of u and v. This would make the mapping less attractive computationally.

Now consider the specification of u and v by the second method; that is, let $u = i/N$. In this case, nodal values of u or v on one side will always be the same as on the opposite side, regardless of node spacing. In general this method results in a more uneven "stretching" of the unit square. The interior mesh will not necessarily reflect boundary shapes. As an illustration, compare Figure 6, which was generated by this method, to Figure 5. Both meshes have the same boundary nodes.

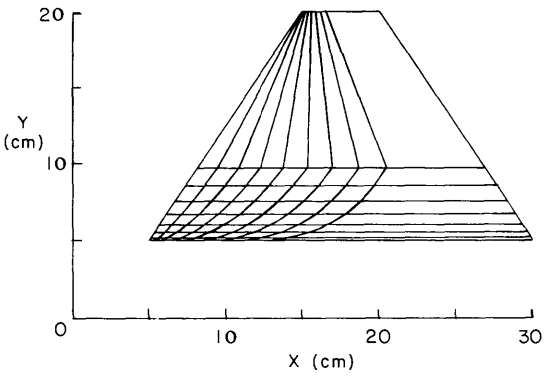


Figure 6. Use of the bilinear projector with method 2.

This use of the mapping allows more flexibility for the analyst to specify boundary node concentrations. The mesh in Figure 7 was generated in this manner; note that boundary nodes are unevenly spaced on opposite sides. If u and v were to be assigned by normalized distances, the mapping would require the additional calculations mentioned above.

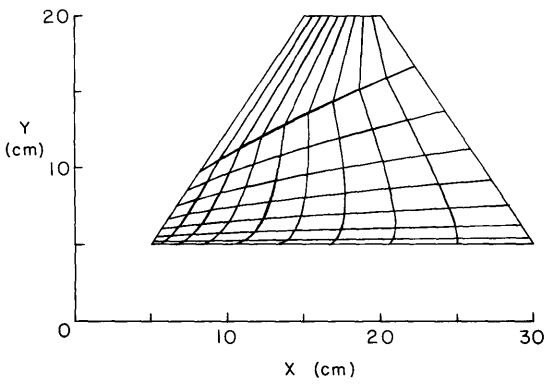


Figure 7. Use of the bilinear projector with method 2, using uneven divisions along the sides.

The use of the "trilinear" projector to generate a mesh in a three-sided region is presented by Haber et al. (1981). It is defined as follows:

$$P_T(u,v) = \tfrac{1}{2}\left[\left(\frac{u}{1-v}\right)\xi(v) + \left(\frac{w}{1-v}\right)\eta(1-v) + \left(\frac{v}{1-w}\right)\eta(w) + \left(\frac{u}{1-w}\right)\psi(1-w)\right.$$
$$\left. + \left(\frac{w}{1-u}\right)\psi(u) + \left(\frac{v}{1-u}\right)\xi(1-u) - w\psi(0) - u\xi(0) - v\eta(0)\right] \qquad (25)$$

where ξ, ψ and η represent the curves describing the three sides; $0 \leqslant u \leqslant 1$, $0 \leqslant v \leqslant 1$, $0 \leqslant w \leqslant 1$; and $u+v+w = 1$. Each of u, v and w increase in a counterclockwise direction along one side.

For this mapping the same two methods of specifying u, v and w on a side will now be investigated such that grid lines and lines of constant u, v and w will coincide. It was found that attempting to take u, v or w as normalized distances along the sides is possible only for the special case where each side is divided into equally spaced segments. Specifying the nodes so that any or all sides are divided into unequally spaced segments will violate the criterion that $u+v+w = 1$.

However, specifying u, v and w boundary values as $u = i/N$, etc., will always satisfy the criterion that $u+v+w = 1$, where this means of specification of u, v and w is used on each of the respective sides. As in the case of the four-sided region this will result in an uneven "stretching" of the triangular region in the transformed space, so that in general the interior mesh may not reflect boundary shapes. Figure 8 depicts a mesh generated in this manner, where the spacing is not the

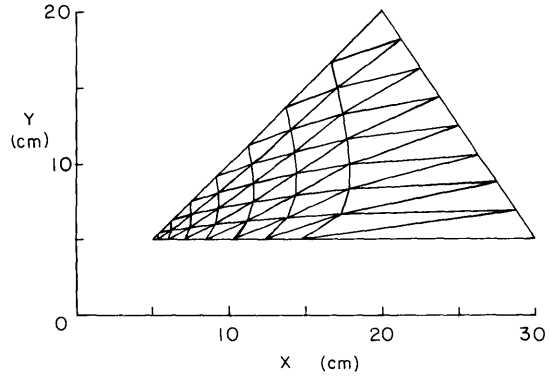


Figure 8. Use of the trilinear projector, using uneven divisions along the sides.

same on the three sides, but an attempt was made to concentrate the nodes at one corner. By using method 2 but choosing node spacing on one side to be proportional to the spacing on another, one may obtain a mesh such as in Figure 9. Nodes are somewhat concentrated at one corner, but now the interior mesh reflects boundary shapes. Now, by trying to concentrate edge nodes severely at one corner but still maintaining side spacings that are proportional to one another, we see that overspill may occur (Fig. 10). Thus, in the use of transfinite mappings, overspill may occur not only as a result of fairly uniform edge node spacings on a highly distorted region, but also for highly concentrated edge node spacings on a regular region.

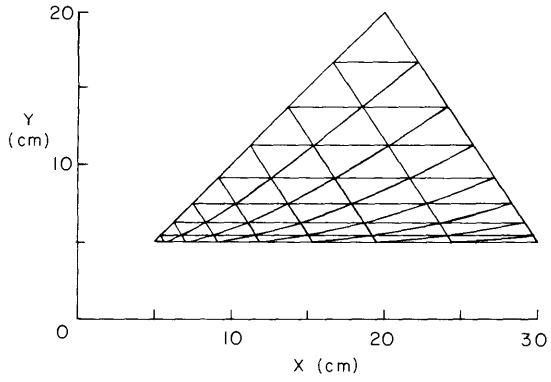


Figure 9. Use of the trilinear projector, using uneven but proportional side divisions.

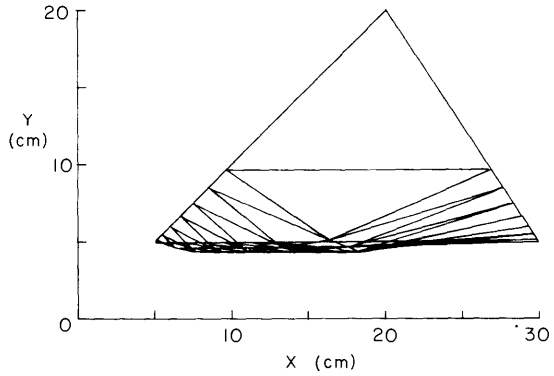


Figure 10. Use of the trilinear projector and the occurrence of overspill.

The trilinear projector results in three-sided elements; the bilinear projector results in four-sided elements. Throughout this work the elements are linear trinagles, so that whenever the bilinear projector was used, a diagonal was constructed through each four-sided element to produce two triangles.

One additional concept worth mentioning in the use of transfinite mappings is that of a "side." Because slope discontinuities may occur anywhere along a side, the word "side" can lose its traditional meaning. For example, Figures 11 and 12 illustrate the use of the bilinear projector to mesh a triangle, and the use of the trilinear projector to create a mesh in a square. Here, the triangle is a four-sided region, and the square is a three-sided region.

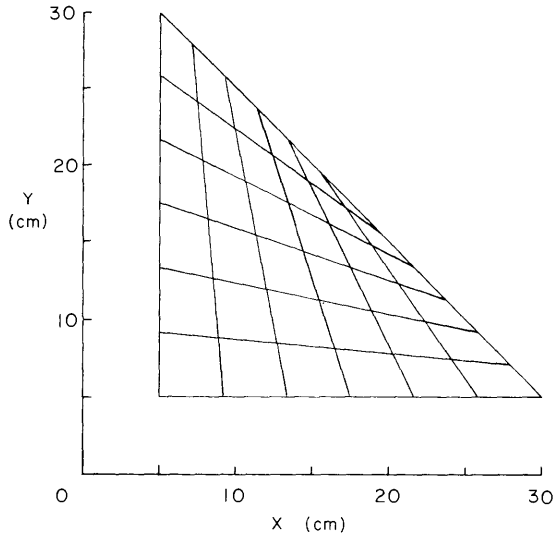


Figure 11. A four-sided region.

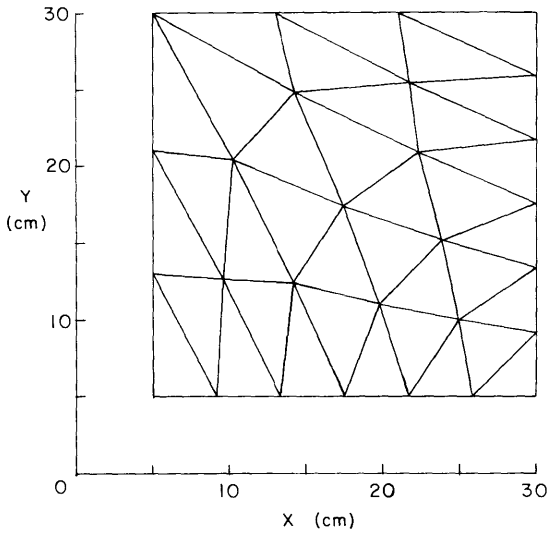


Figure 12. A three-sided region.

FEMOVE

The computer program developed by the author is named Femove. It models two-dimensional phase change in conduction heat transfer by using the moving-mesh finite element technique in conjunction with transfinite mappings as described above. The program interactively allows the selection of the following options:

1) The coordinate system may be either Cartesian or (*r*, *z*) cylindrical coordinates.

2) If desired one iteration may be performed in calculating the location of the phase front at each time step.

3) One or two phases may be modeled.

4) If desired the heat conduction equation may be solved over each phase for several time steps before the front is allowed to move. This is useful if crude temperature distributions are input initially but the user desires a smoother start-up condition for the temperature distribution.

Femove was written in Fortran and is composed of a main program and an assortment of subroutines. A flow chart is shown in Figure 13 for the major processes in the main program and the calls to the subroutines.

Subroutine Ratio

As the phase front moves, the mesh will deform, and the sizes of the frozen and unfrozen zones may expand or contract. Often it is desirable to slide the nodes along the edge of a zone so that they keep their same relative spacing along an edge. For each node on the side of the initial mesh, Ratio determines the distance from the first corner of the side to this node, relative to the length from the first corner along the side to the last corner. Thus, Ratio produces ratios. The edges of the zones may be of an arbitrary overall shape, as long as they are piecewise linear.

Subroutine Move

Move uses the temperature gradients along the phase change front to move the nodes on the front to their location for the new time step.

Subroutine Newmsh

This subroutine administrates the formation of the new mesh in each zone. Information on the location of the edges in each zone must be stored; Newmsh coordinates this information and calls other subroutines to form the new mesh.

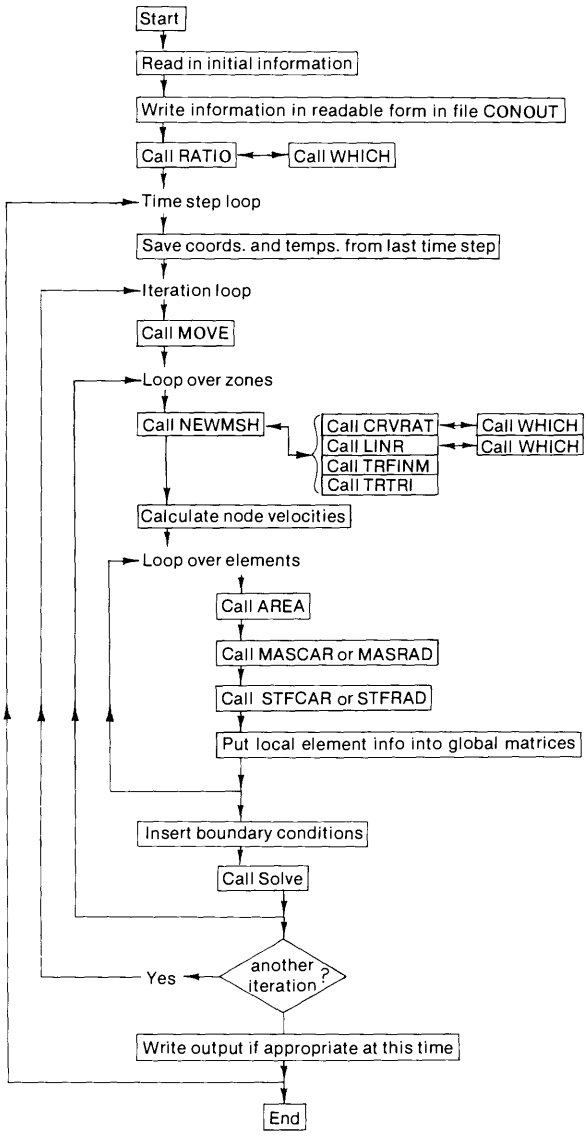


Figure 13. Flow chart for Femove.

Subroutine Crvrat

Crvrat uses the information originally gained in Ratio to slide the nodes tangentially along the new locations of the edges of a zone to position them so that they maintain the same relative positions along the side as they did in the original mesh.

Subroutine Which

Given information on what side of the zone is being examined, Which finds the first and last corners of the side.

Subroutine Linr

If Crvrat is not called, other options are available for specifying node locations along the side of a zone. Linr enables the nodes to be spaced in several ways along a straight line joining the corners. These spacings include: 1) even increments, 2) increment spacing proportional to the square root of the length of the side, or 3) increments spaced in a linearly increasing or decreasing fashion (the second segment is twice the length of the first, the third segment is three times the length of the first, etc.).

Subroutine Trfinm

This subroutine performs a transfinite mapping for a four-sided zone to generate the new interior node locations in the mesh, given the locations of the nodes on the side of the zone.

Subroutine Trtri

Trtri performs a transfinite mapping for a three-sided zone to generate new interior node locations, given the locations of the nodes on the side of a zone.

Subroutine Area

This subroutine calculates the area of a triangular finite element.

Subroutines Mascar and Masrad

For Cartesian coordinates, Mascar forms the local 3x3 matrix, which is denoted by $\underline{\underline{M}}$ in eq 9. Masrad forms the $\underline{\underline{M}}$ matrix using radial coordinates.

Subroutines Stfcar and Stfrad

The matrix labeled $\underline{\underline{K}}$ in eq 10 is formed locally in Cartesian coordinates by Stfcar and in radial coordinates by Stfrad.

Subroutine Solve

This subroutine solves the system of eq 13, which has been stored in banded form (Lynch 1982b).

COMPARISON WITH ANALYTICAL RESULTS

The results of program Femove will be compared to four analytical solutions for heat conduction with phase change: 1) the Neumann solution, 2) outward radial freezing from a cylinder, 3) one-phase freezing in a corner, and 4) two-phase freezing in a corner.

Femove is constructed so that each phase, frozen and unfrozen, is represented by a separate zone in the transfinite mapping. That is, the mesh for each phase is constructed independently, with the restriction that the meshes match to share the nodes along the phase boundary. There are problems in which it will be necessary to model only one of the phases. For example, later in this report the freezing of a pipe containing a flowing liquid will be modeled; the program uses one zone to model the ice formation. The program will also be used to model an experiment involving freezing of saturated sand around a pipe; for this case it is necessary to model both the frozen and unfrozen phases, and two zones are used to accomplish the result. The first three comparisons with analytical solutions model one phase; the fourth models two phases.

The Neumann solution

This solution may be found in Carslaw and Jaeger (1959, p. 286) and is one-dimensional in Cartesian coordinates. It is assumed that the region is initially at its melting point; at time zero a step change in temperature is imposed on the surface so that the material begins to freeze. The case considered is the freezing of water. For this problem Carslaw and Jaeger give the result

$$Y = 2\lambda\sqrt{\alpha t} \tag{26}$$

where λ must be obtained from the expression

$$\lambda \exp(\lambda^2)\, \mathrm{erf}(\lambda) = \frac{C(T_f - T_s)}{L\sqrt{\pi}} \tag{27}$$

where Y = location of the phase front
t = time

α = thermal diffusivity of ice = 0.00134 cm^2/s
L = volumetric latent heat of fusion = 80 cal/cm^3
T_f = phase change temperature = 0°C
T_s = surface temperature = -6°C.

Any density difference between unfrozen and frozen materials is neglected.

The model was assigned an initial frozen layer of 0.188 cm. Forty time steps were taken to model 7200 s (2 hr). Time steps of unequal sizes were taken, so that the steps were linear in the square root of time. (This was done because freezing generally progresses as the square root of time for a constant imposed boundary temperature.) Figure 14 shows the calculated location of the phase front versus the analytical location through time. The results compare well. Figure 15 shows the initial mesh and meshes at intermediate and final times. Nodes in the -6°C surface are

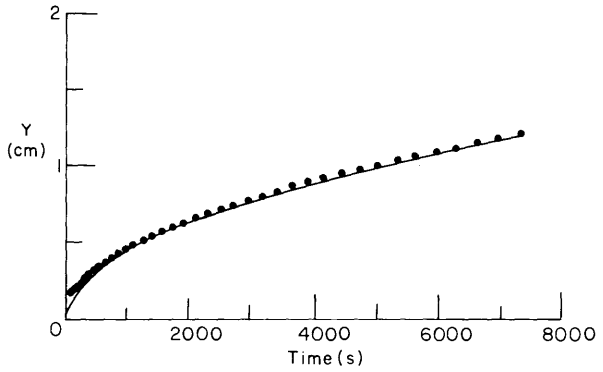


Figure 14. Phase front location comparison with the one-phase Neumann solution. The solid line is the analytical solution; the circles represent the calculated solution.

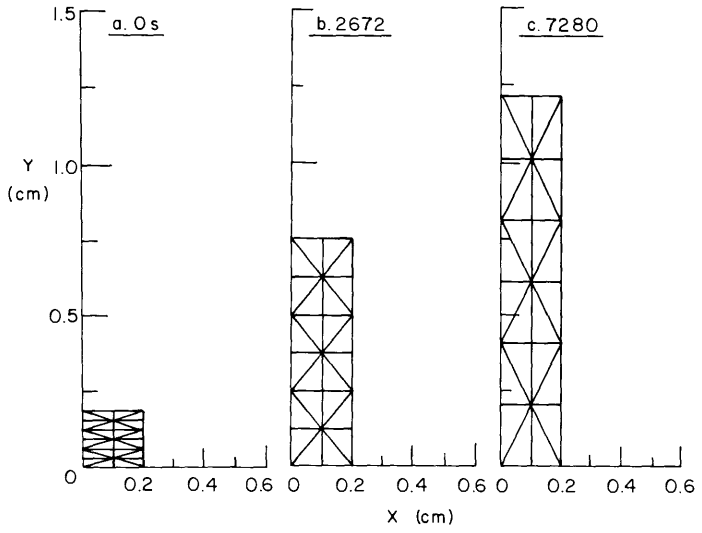


Figure 15. Initial, intermediate and final meshes for the Neumann comparison.

not moved during the run; nodes on the phase interface are moved in accordance with eq 2, and nodes along the zero flux sides are slid tangentially along the sides to maintain an equal spacing between the corners.

This problem was also run using the apparent heat capacity method on a fixed mesh with finite differences (where the alternating-direction implicit procedure was used to solve the resulting set of equations). Seventy-five nodes were used, compared to 21 nodes and 24 elements used

here. Eight hundred time steps of equal size were taken to model two hours of time, and the simulation required 240 cpu seconds on the Prime computer at CRREL. The Neumann simulation using the moving mesh program Femove required 16 cpu seconds. The amount of cpu time required per time step is approximately the same in both cases even though the finite difference simulation used many more nodes. However, because the number of time steps required by the apparent heat capacity method on a fixed mesh is greater, the solution was more costly. Both the finite difference program used (Albert 1983) and the finite element program presented here were developed to model two-dimensional situations. Although there are differences in programming style and in the capability of the programs, so that a strict comparison may not be made, the simulations run for this report generally required much less computer time than similar simulations run using the apparent heat capacity method.

Radial freezing

This solution is one-dimensional in radial (r, z) coordinates. The region outside a cylinder of radius a is assumed to be water initially at its melting point, and the temperature profile is assumed to be steady state, given the location of the phase front at any point in time. This is a reasonable assumption for the high value of latent heat considered here, where the Stefan number is 0.075. At time zero a step change in temperature is imposed on the surface of the cylinder, causing the water to freeze. For this case Carslaw and Jaeger (1959, p. 296) give the analytical result

$$2R^2 \ln\left(\frac{R}{a}\right) - R^2 + a^2 = 4\alpha(T_f - T_s)\, t/L \tag{28}$$

where R = location of the phase front
t = time
a = radius of cylinder = 0.1 cm
α = thermal diffusivity of ice = 0.00134 cm^2/s
L = volumetric latent heat of fusion = 80 cal/cm^3
T_f = phase change temperature = 0°C
T_s = surface temperature = -6°C.

The model was assigned an initial ice layer of 0.1708 cm; this layer corresponds to an initial time of 30 s. Twenty time steps (again, as the square root of time) were taken to model 7200 s (2 hr). The calculated and analytical locations of the phase front are plotted in Figure 16. The results compare well. Initial, intermediate and final meshes are shown in Figure 17.

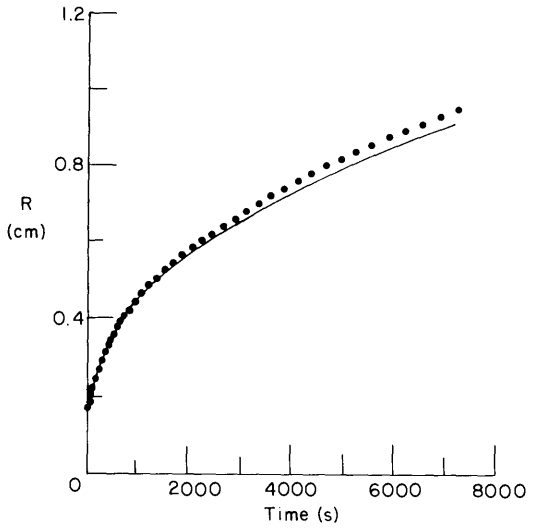


Figure 16. Phase front location comparison with the quasi-steady radial solution. The solid line is the analytical solution; the circles represent the calculated solution.

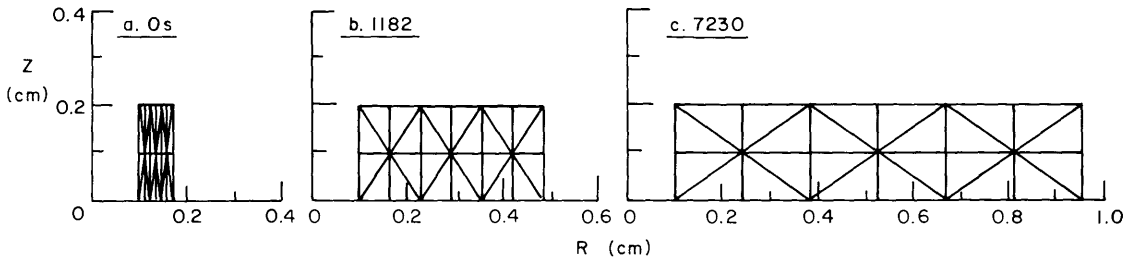


Figure 17. Initial, intermediate and final meshes for the radial comparison.

One-phase corner

A two-dimensional, Cartesian analytical solution to a problem involving freezing is provided by Budhia and Kreith (1973). They arrive at the solutions for both one- and two-phase freezing in a wedge by superposing the results of two auxiliary problems: heat conduction without phase change (with the same initial and boundary conditions as the actual problem), and heat conduction with phase change (with initial and boundary temperatures equal to zero, and the effect of the latent heat at the interface represented as a moving surface source).

In the one-phase corner solution it is assumed that the unfrozen region is composed of a material of uniform temperature equal to the temperature of phase change (0°C), making it necessary to model only the frozen region. The region is modeled by a single four-sided zone; the initial mesh is shown in Figure 18. An initial uniform ice layer 0.5 cm thick was assumed, and the accompanying initial time is 180 s. The temperature of the sides of the corner was maintained constant at -6.0°C. Because of the symmetry of the problem, only half of the corner needs to be modeled; the 45° line of symmetry at the vertex of the corner is assigned the zero-flux boundary condition. The value for the latent heat was 9.3678 cal/cm^3. This is much lower than the value of latent heat for water used in earlier comparisons, so that in this problem the movement of the phase front is much faster than for the earlier comparisons made, and the temperature profile is much less like steady state. This, then, is a true test of the method.

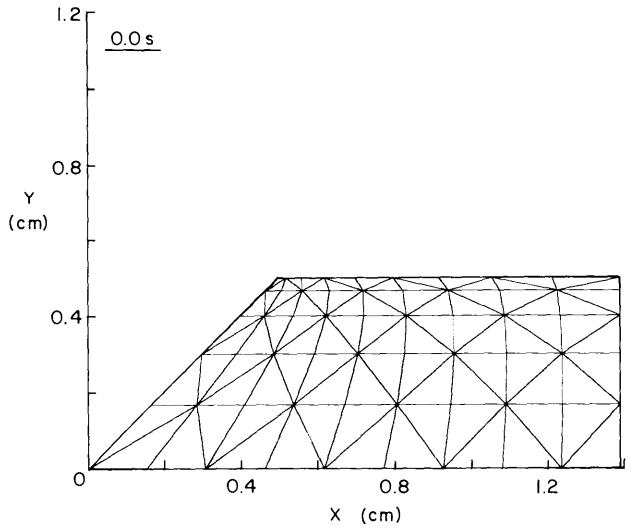


Figure 18. Initial mesh for the one-phase corner problem.

In the analytical solution the sides of the corner extend to infinity. This is modeled by a zero-heat-flux condition at that boundary, and the boundary is moved away from the vertex of the corner at a rate that keeps x^2/t constant (x is the location of the boundary and t is time). The initial location of this boundary was x=1.39 cm; for this location the analytical solution assumes a one-dimensional behavior, so that the assumption of a zero-flux vertical boundary there is valid. Note that a "naive" start was assumed. That is, the initial phase-front profile is a straight line.

The movement of the phase front is in accordance with eq 2; in addition, to prevent the nodes from crossing as the ice fills in the vertex of the corner, the phase boundary nodes are slid tangentially along the phase front, keeping the same relative distance along the front as in the original mesh. Femove was run for 40 time steps (spaced as the square root of time), which model 7440 s (≈ 2 hr)(adjusted time = 7620 s). The run required 67 cpu seconds on the Prime computer at CRREL. The final mesh is shown in Figure 19. For scale comparison, Figure 20 illustrates the initial mesh plotted on the same scale as the final mesh.

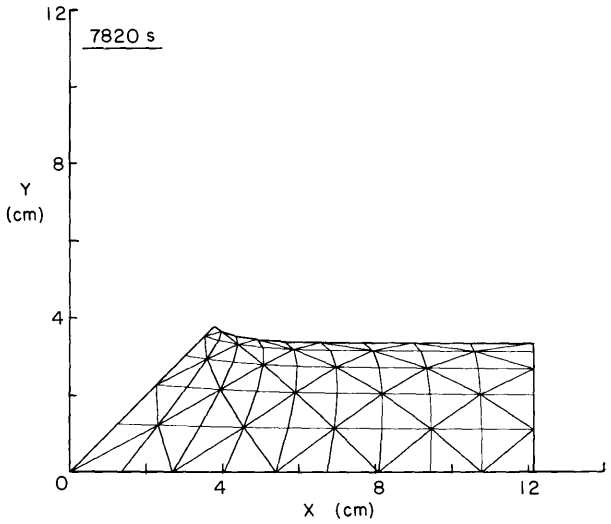


Figure 19. Final mesh for the one-phase corner problem.

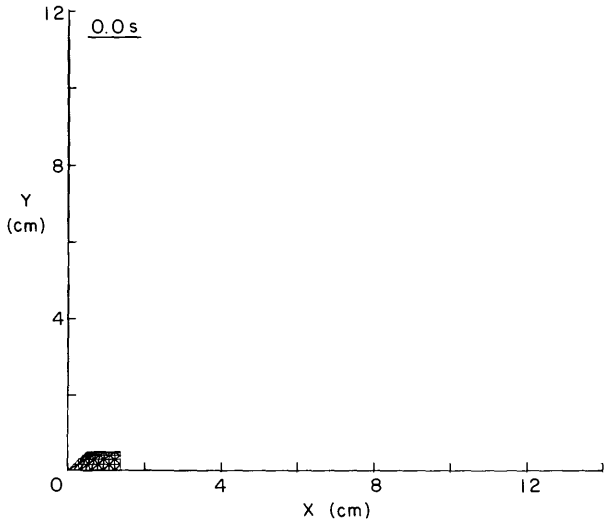


Figure 20. Initial mesh plotted on the same scale as the final mesh.

The results are compared with those of Budhia and Kreith (1973) in Figure 21, where

$$r^* = r/\sqrt{4\alpha t} \tag{29}$$

r = distance from the vertex of the corner
t = time
α = thermal diffusivity in the frozen zone = 0.00134 cm^2/s.

The results compare very well.

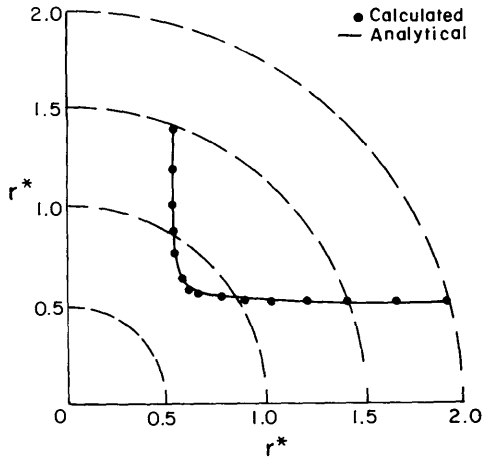


Figure 21. Comparison of the one-phase corner results with the analytical solution.

Two-phase corner

Budhia and Kreith also present analytical results for the case where the region is initially at a uniform temperature above the freezing temperature. Now both frozen and unfrozen zones must be modeled. When two phases are modeled, it is advantageous to iterate once on the location of the phase front for each time step, as described earlier.

The corner has sides held constant at -6.0°C, the initial uniform temperature of the unfrozen material is 1.8°C, the thermal diffusivity is 1.34×10^{-3} cm^2/s, and the latent heat of fusion is again 9.3678 cal/cm^3. As before, the frozen zone is four-sided and has an initial uniform thickness of 0.5 cm. The unfrozen region is modeled by a three-sided zone, as illustrated in Figure 22. The 45° line of symmetry is assigned the zero-flux condition in both phases, as is the vertical boundary farthest from the vertex of the corner. In the unfrozen region the node at the top vertex of the triangle, farthest from the phase front, is assigned a constant 1.8°C temperature. This corner was initially placed at a distance of seven frozen thicknesses above the phase front. Since a vertical zero-flux boundary is maintained below that node, the initial placement of the corner node requires that the vertical zero-flux boundary of the frozen zone has an x-coordinate much greater than is required for satisfaction in assuming the zero-flux condition there. As the simulation progresses, the corner at the top vertex of the triangle is moved to maintain its location seven frozen thicknesses above the phase front, and the vertical boundary is moved in accordance with this node. The parameter for implicitness in time ϵ was set equal to 1.0, and the mesh used each time step was centered in time.

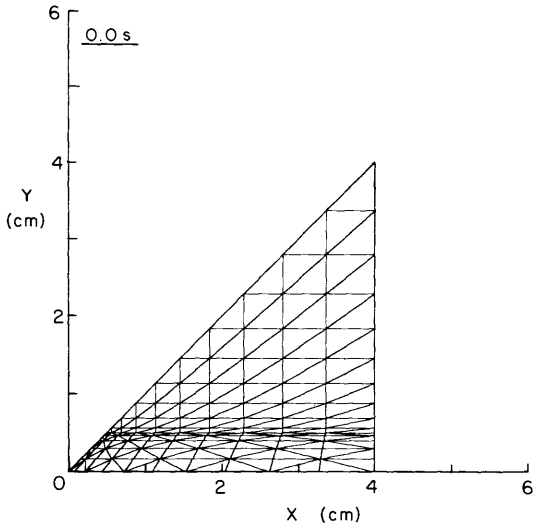


Figure 22. Initial mesh for the two-phase corner comparison.

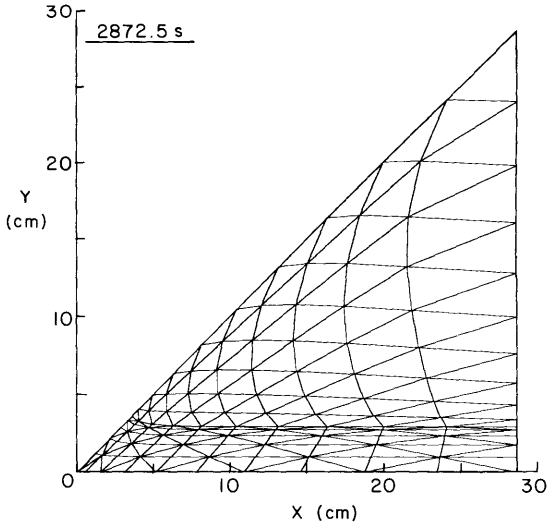


Figure 23. The mesh at an intermediate time step.

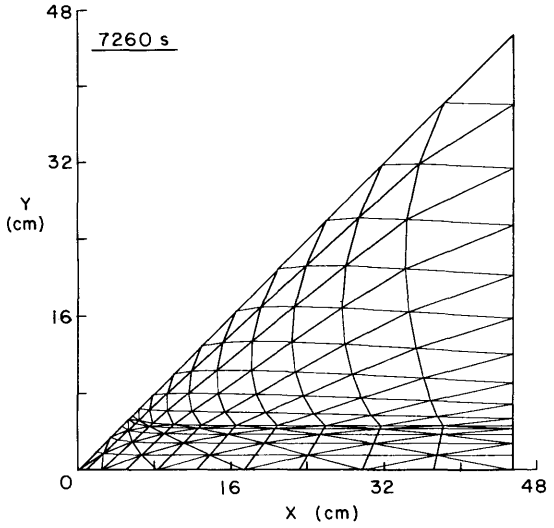


Figure 24. Final mesh for the two-phase corner comparison.

The situation was modeled for 40 time steps, again taken as the square root of time, totaling two hours (the simulation required 232 cpu seconds on the Prime computer at CRREL). An intermediate mesh is shown in Figure 23, and the final mesh is shown in Figure 24. Note that they are plotted on different scales; in Figure 25 the initial mesh is plotted on the same scale as the final mesh. The solution is compared to that of Budhia and Kreith in Figure 26. Very good agreement is found.

During the course of modeling this problem an interesting and disturbing phenomenon was observed. In setting up the original mesh for the three-sided zone the 1.8°C constant-temperature corner represents an infinite boundary approximation, and very small temperature gradients are expected in that region, so an attempt was made to make the elements large in that section of the mesh. An example of this type of mesh is shown in Figure 27. However, when the problem is modeled on that mesh, the temperature in the unfrozen zone grows out of range; temperatures between 0°C and 1.8°C are expected, but temperatures as high as 3°C develop. This occurs even when the time step is reduced by a factor of four.

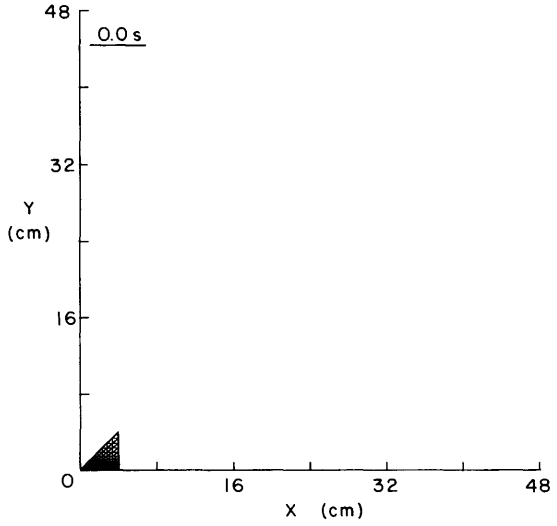


Figure 25. Initial mesh plotted on the same scale as the final mesh.

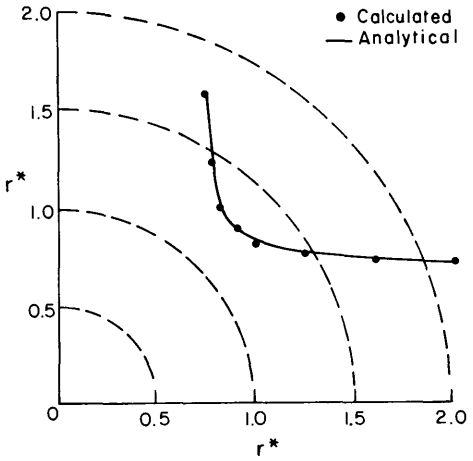


Figure 26. Comparison of the two-phase corner results with the analytical solution.

To illustrate the distortion effect, the temperatures for the y-coordinate of the nodes along the vertical zero-flux boundary in the unfrozen zone are plotted for several cases in Figure 28. The solid circles represent the case where the mesh in Figure 22 is used; the node representing the semi-infinite corner was assigned a constant 1.8°C temperature, and the satisfactory results previously discussed were obtained. For the same mesh the node representing the semi-infinite corner was assigned a zero-flux condition; these results are plotted using an open circle. The results for this mesh differ very little, indicating that the boundary condition at the corner is not at fault for this distortion. When the mesh illustrated in Figure 27 is used, distortion develops. The geometry of the mesh in the unfrozen zone is the only thing that was changed. (The evaluation of the mesh

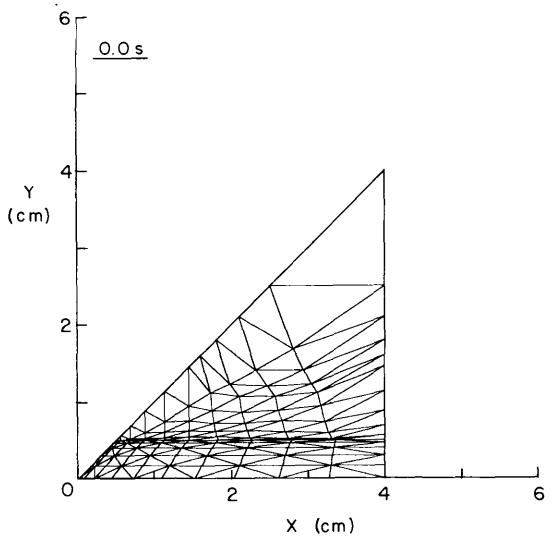


Figure 27. Initial mesh for two-phase corner problem where numerical distortion of the temperature solution occurred.

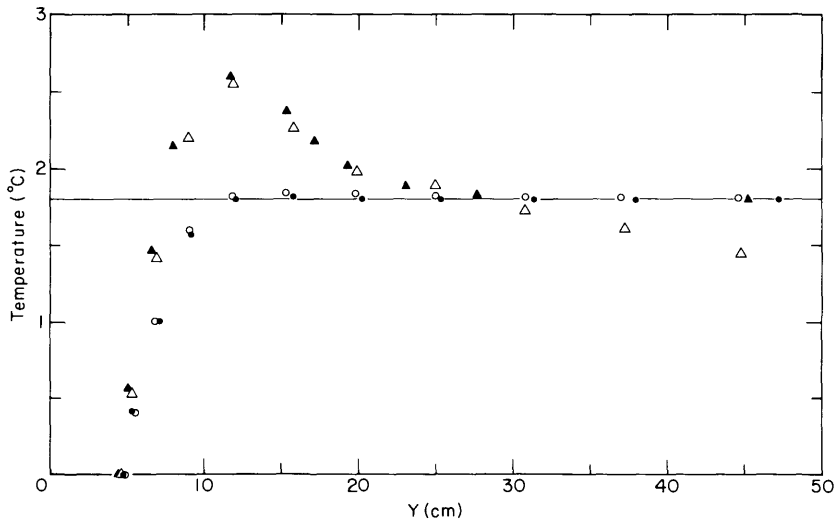


Figure 28. Temperatures at nodes along the vertical zero-flux boundary in the unfrozen zone:

- • *From mesh in Fig. 22 (Pe = 34), constant at semi-infinite corner node*
- ○ *From mesh in Fig. 22 (Pe = 33), zero-flux condition at semi-infinite corner node*
- ▲ *From mesh in Fig. 27 (Pe = 76), constant at semi-infinite corner node*
- Δ *From mesh in Fig. 27 (Pe =78), zero-flux condition at semi-infinite corner node.*

was carefully monitored during the simulation to ensure that no tangling of the mesh occurred.) The solid triangles in Figure 28 represent the case where the node at the semi-infinite corner was assigned a constant 1.8°C temperature, and the open triangles represent the case where the corner was assigned a zero-flux condition. Although the different boundary condition changes the nature of the distortion, the distortion is not eliminated. Note that the temperatures along the vertical boundary were plotted for convenience of illustration, but that the distortion occurred throughout the interior of the unfrozen zone.

Extensive investigations were done to find the source of this problem, including using different mesh configurations with better aspect ratios but all with a large element at the semi-infinite corner of the unfrozen region. If the node at the semi-infinite corner and the vertical boundary are not moved, the solution proceeds as expected. However, by maintaining a large element at the corner and moving it fairly rapidly, distortion does develop. It was finally concluded that the numerical distortion is due to effects of a large Peclet number. The Peclet number is $\ell v/\alpha$, where ℓ is a characteristic length (the element length here), v is the velocity of the convection (the mesh velocity), and α is the thermal diffusivity. Numerical difficulties due to large Peclet numbers are known to occur when modeling convection on a fixed mesh. However, the distortions encountered there are oscillatory and may develop when the Peclet number gets larger than one. In both two-phase corner problems presented above, the maximum Peclet numbers ranged between 30 and 80. The effect of the Peclet number in this problem is discussed further later. It was discovered in this work that the combination of large elements and a rapidly moving mesh can lead to unacceptable distortions in the numerical solution.

APPLICATION

Comparison with experimental results

The finite element program was used to model laboratory data that were obtained at CRREL by O'Neill (in press). The experimental apparatus consisted of a 55-gal. drum filled with saturated, graded Ottawa sand, with an off-center, 2-in. copper pipe passing vertically through the drum. Copper tubing was wound around the outside of the drum, and glycol solution at approximately 5°C was pumped through the tubing in an attempt to maintain the outside of the drum at constant temperature. Cold ethylene glycol was pumped through the internal copper pipe, keeping the surface of the pipe between 4° and -9°C. The subsequent freezing of the sand was monitored via a system of thermocouple strings placed in the sand.

The latent heat of fusion used in the model was 23.5 cal/cm^3, and the other thermal properties were as follows:

Unfrozen	Frozen
$k = 5.6\times10^{-3}$ cal/cm s°C	$k = 9\times10^{-3}$ cal/cm s°C
$C = 0.589$ cal/cm^3 °C	$C = 0.398$ cal/cm^3 °C .

The experiment and determination of material parameters are outlined in O'Neill (in press). Two runs of the experiment were performed; these will be labeled case 1 and 2.

The original finite element mesh (Fig. 29) consists of two (frozen and unfrozen) four-sided zones, each zone 5 elements by 7 elements. Because of the symmetry of the drum, only half of the drum needed to be modeled; the boundary condition along the horizontal line of symmetry was assigned zero flux. The nodes representing the surface of the drum and the surface of the pipe were placed at equal intervals along the corresponding semicircles and were not moved during the simulation. The nodes representing the freezing front were initially equally spaced along a semicircle whose radius was 0.6 cm greater than the pipe. (The location of the phase front was interpolated to be at this location 900 seconds after the start of the run.) The movement of these nodes

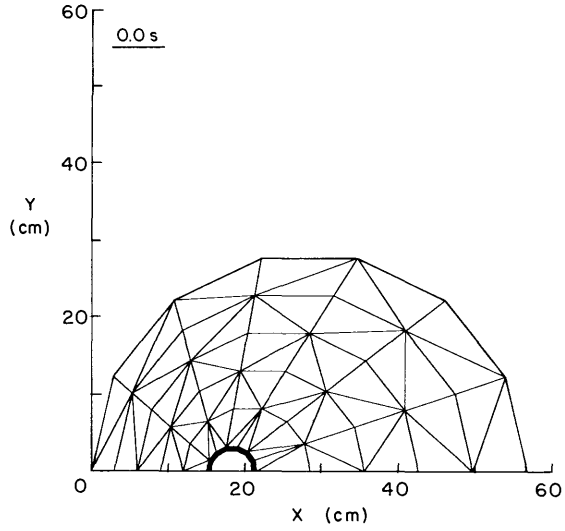


Figure 29. Initial mesh for modeling the drum experiment.

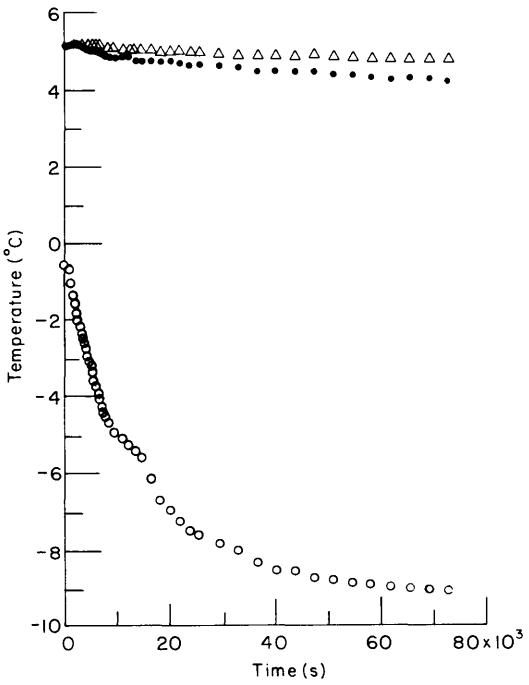


Figure 30. Temperatures used in the simulation: the triangles represent the warm rim, the closed circles represent the cold rim, and the open circles represent the pipe.

was specified by the boundary condition at the interface and were maintained at equal intervals along the interface by subroutine Crvrat.

The temperature of the pipe was monitored by four thermocouples placed on the pipe circumference; the temperature they reported varied by as much as 0.7°C at any given time. For simplicity the model assumed a uniform pipe temperature, which was the average of the thermocouple temperatures. This pipe temperature varied with time.

The outside drum rim temperature was monitored by seven thermocouples; this temperature also varied with space and time. The thermocouples on the rim nearest the pipe exhibited a lower temperature than the rest of the rim throughout the experiment. To model this, the low temperatures were averaged for a given time, and the average temperature was used for the two rim nodes nearest the pipe. The third rim node was assigned a temperature that was an average of the low and high rim temperatures. The rest of the nodes were assigned the high rim temperature, which was the average of the temperatures in that region. All rim temperatures varied with time. The temperatures used to model the pipe, warm rim, and cool rim are plotted against time for case 1 in Figure 30.

The initial temperatures in the sand ranged from 4.9 to 5.3°C, but the model assumed an initial uniform temperature in the unfrozen zone of 5.2°C.

Case 1 was run for 20 hours (72,000 s). This was modeled by Femove using 20 time steps, which were spaced linearly as the square root of time. The resulting temperature distributions along the line of symmetry for a horizontal cross section of the drum are compared in Figure 31 for times of 5 hours (18,000 s) and 20 hours (72,000 s). Both temperature distributions agree very well. The "kink" in the temperature solution at 0°C is a result of the latent heat condition at the phase boundary, where there is a jump in the temperature gradient between the frozen and unfrozen zones. Figure 32 displays the mesh at 5 hours and at the end of the run (20 hours).

The locations of the phase front along the line of symmetry are plotted in Figure 33. The very simplified initial conditions assigned to the model account for the difference between computed and actual values for very early times. The effect disappears, however, and the model predicted the location of the phase front very well at intermediate and later times.

Femove was also used to model case 2, which had the same initial conditions but slightly different boundary conditions at later times. Case 2 was run and modeled for 22.2 hours. The results are very similar to the results of case 1. The temperature distributions are compared in Figure 34 for times of 5.2 hours (18,720 s), and 22.2 hours (79,920 s). Again the results compare very well.

These data have been used in a numerical simulation by O'Neill (in press), where the boundary integral equation method was used to model the phase change. The results obtained here are superior to results obtained by O'Neill. This indicates that the technique used here is much better able to model phase change when the temperature distribution is not quasi-steady, that is, when the latent heat does not have a high value relative to the sensible heat capacity and temperature difference.

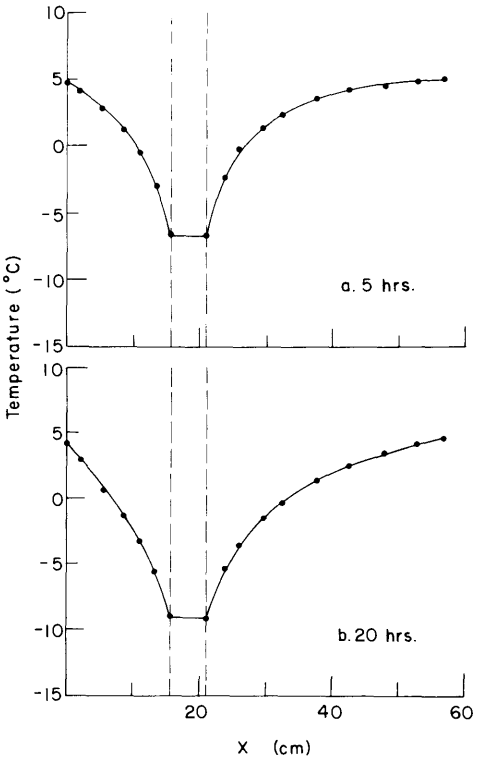


Figure 31. Temperature comparison along horizontal axis of symmetry for case 1. The dots represent data from the experiment. The solid line represents the numerical solution. The dashed lines represent the pipe.

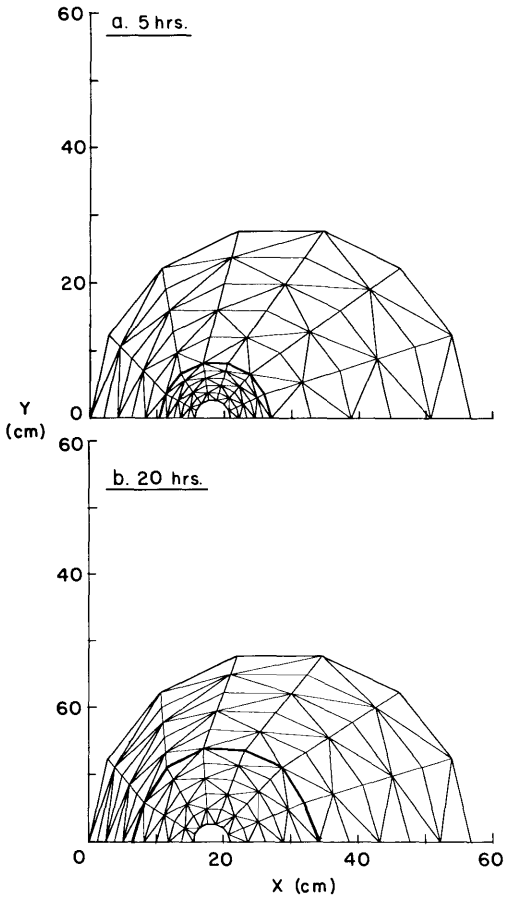


Figure 32. The mesh for the experiment comparison (case 1). The phase front is highlighted for clarity.

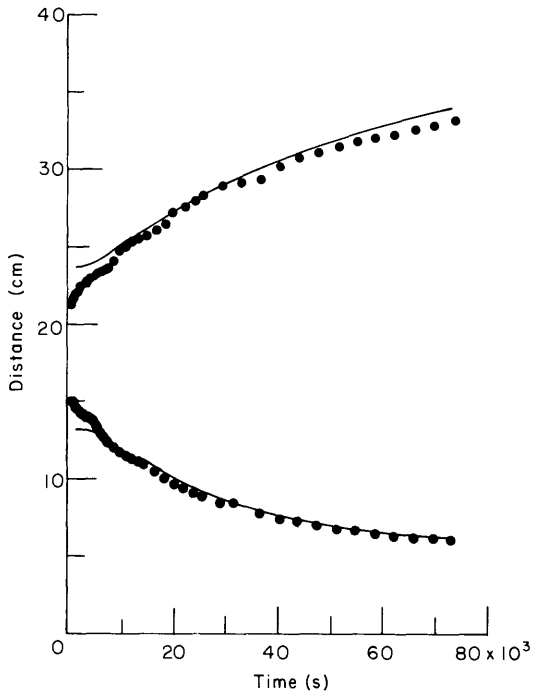


Figure 33. Phase front locations along the horizontal axis of symmetry for case 1. The solid line represents the numerical result, and the dots are locations interpolated from the data.

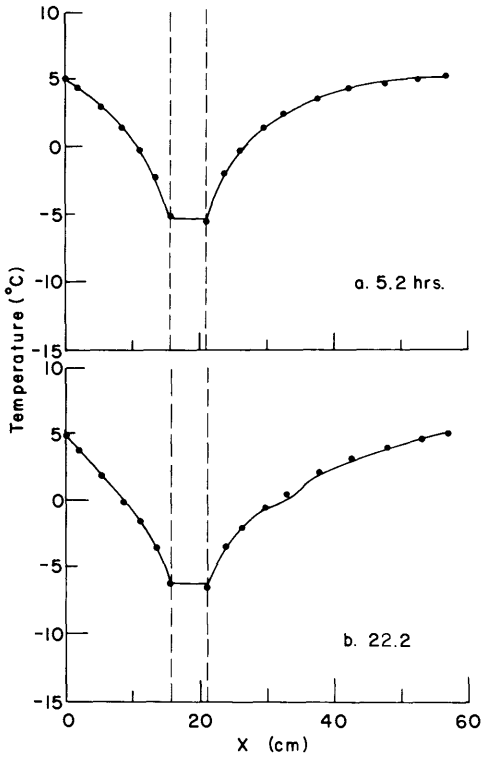


Figure 34. Temperature comparison along the horizontal axis of symmetry for case 2. The dots represent data from the experiment. The solid line represents the numerical solution. The dashed vertical lines represent the pipe.

Freezing of fluid flowing in a pipe

A significant class of problems involving freezing concerns the case where there is flow in the unfrozen region. Numerical models have been developed that involve the use of a mesh that is fixed throughout the simulation (for example, Gartling 1978). A problem with the fixed mesh approach is that the phase boundary rarely coincides with the element boundaries, so that a particular element is often supposed to be composed partly of frozen material and partly of flowing fluid. To deal with this problem, some sort of smearing technique is usually used, in which the location of

the phase front itself is not precisely defined; instead the phase change occurs over a "transition" region, where the fluid is assigned a very high viscosity. These techniques are not prepared to handle a situation where, for example, a boundary condition involving convective heat transfer is associated with the phase boundary. The use of a mesh that is able to move in time overcomes these difficulties. The phase boundary always coincides with element boundaries, making it possible to clearly specify both the thermal and the fluid boundary conditions there.

As an illustration of this technique, consider the case of two-dimensional freezing of fluid flowing through a pipe, where the inlet temperature of the water is above freezing and the pipe walls are maintained at a constant temperature below freezing. Of particular interest is the case where the flow is driven by a head drop across the length of pipe, which is more realistic than the constant-flow assumption used in previous investigations.

For some laminar and turbulent pipe flows Gilpin (1979) and others have observed that the ice/fluid interface assumes a wavelike structure down the length of pipe. By extended extrapolation of Gilpin's work Epstein and Cheung (1982) attempted to identify cases where the ice/fluid interface assumes a wavelike structure and where it assumes a smooth structure. In this report, it will be demonstrated that Femove may be used to model freezing in a pipe when the ice/fluid interface is smooth.

The flow is assumed to be fully developed and turbulent. As the water flows through the cold pipe, its temperature will drop, resulting in the formation of an ice layer that thickens along the length; this, in turn, will tend to slow the flow. In some instances the pipe may freeze shut. Femove will be used to model the frozen region adjacent to the pipe wall; the addition of a convective heat transfer coefficient of the ice/water interface will provide the thermal input from the flowing water. Expressions for the friction factor for the flow and the heat transfer coefficient will be needed to model this phenomenon.

An analytical solution for the problem of turbulent heat transfer in a pipe with uniform flux and cross section was obtained by Petukhov and Popov (1963), who demonstrated that the result agreed well with experimental results. Their formula is

$$Nu = \frac{(f/8)\, Re\, Pr}{2.07 + 12.7\sqrt{f/8}\,(Pr^{2/3} - 1)} \tag{30}$$

where $Nu = hD/k$
h = heat transfer coefficient
D = pipe diameter
k = thermal conductivity
Re = Reynolds number
Pr = Prandtl number

and the friction factor f is to be determined from the Filonenko equation:

$$f = [1.82 \log_{10}(Re) - 1.64]^{-2}. \tag{31}$$

Karlekar and Desmond (1977) may be consulted for more information on this topic.

Strictly speaking, as the freezing progresses, the flow may be slowed enough to become laminar, requiring the use of different expressions for the friction factor and heat transfer coefficient. However, that case was not considered here.

Procedure

A schematic diagram of the pipe and the ice is shown in Figure 35. Because the temperature and Reynolds number may vary over the total length of pipe, the flow is described incrementally, with each increment corresponding to the length of a finite element in the surface of the ice.

In the fluid region each increment is treated as though it were a uniform length of pipe. At each time step, given the initial ice configuration, the fluid velocity may be determined iteratively as follows. First, a guess of an inlet velocity for the pipe v_1 is made. Then the outlet velocity for the first increment, and hence the inlet and outlet velocities for each successive increment, may be determined by conservation of mass in the fluid region:

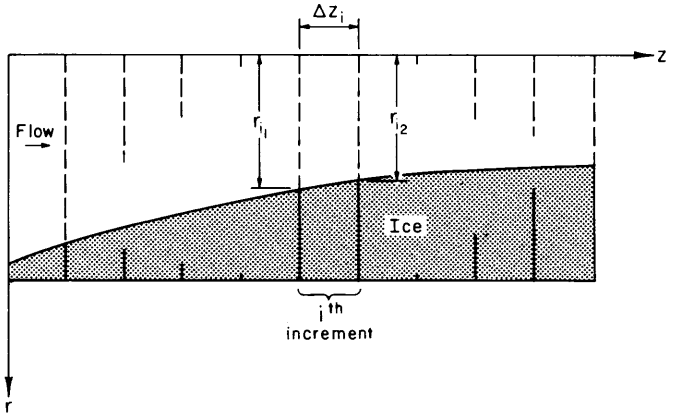


Figure 35. Schematic diagram of pipe with ice interface.

$$v_{i_2} = \left(\frac{A_{i_1}}{A_{i_2}}\right) v_{i_1}$$

where A_{i_1} and A_{i_2} are the cross-sectional areas of the flow at the inlet and outlet section of increment i, respectively. Once all of the v_{i_j} have been determined, the average velocity for each increment is given as:

$$\bar{v}_i = \tfrac{1}{2}(v_{i_1} + v_{i_2}) .$$

The head drop over each increment is calculated from

$$H_i = f_i \frac{\Delta z_i}{\bar{r}_i} \frac{\bar{v}_i^2}{g} \tag{32}$$

where H_i = head drop over an increment
f_i = friction factor for the increment
Δz_i = length of the increment
$\bar{r}_i$ = average radius = $\tfrac{1}{2}(r_{i_1} + r_{i_2})$
g = acceleration due to gravity.

The friction factor for each increment is calculated from eq 31, using the Reynolds number defined as

$$Re = \frac{2\bar{r}_i \bar{v}_i}{\upsilon}$$

where υ is the kinematic fluid viscosity. The head drops of all increments must sum to the head drop over the total length of pipe, which is constant and specified at the beginning of the simulation:

$$\underline{\bar{H}} = \sum_{i}^{N} H_i .$$

If the incremental head drops do not sum to the desired total, a new inlet velocity is tried, and the process is repeated until the head drop criterion is satisfied. Once the fluid velocity in each increment has been determined, the heat transfer coefficient for each increment may be calculated from eq 30.

Next, the heat transfer coefficients are used to determine the temperature of the flowing fluid in each increment. Consider the increment of fluid illustrated in Figure 36. Let T_{i_1} and T_{i_2} be the water temperature at the inlet and outlet of the increment, respectively. Then the heat flows q_{i_j} for the increment are given as:

$$q_{i_1} = \pi r_{i_1}^2 \rho v_{i_1} C_p T_{i_1}$$

$$q_{i_2} = \pi r_{i_2}^2 \rho v_{i_2} C_p T_{i_2} \qquad (33)$$

$$q_{i_w} = h_i S_i (T_{i_w} - T_s)$$

where ρ = density of the flowing fluid
C_p = specific heat of the flowing fluid
h_i = heat transfer coefficient
S_i = surface area of the frustrum formed by the ice boundary around the flowing fluid = $\pi(r_{i_1} + r_{i_2})\sqrt{(r_{i_1} - r_{i_2})^2 + (\Delta z_i)^2}$
T_{i_w} = bulk water temperature for the increment = $(T_{i_1} + T_{i_2})/2$
T_s = temperature of the ice = 0°C.

Energy is conserved:

$$q_{i_1} - q_{i_2} - q_{i_w} = 0 . \qquad (34)$$

Equations 33 may be substituted into eq 34 and solved for T_{i_2}:

$$T_{i_2} = \frac{\pi\rho C_p r_{i_1}^2 v_{i_1} T_{i_1} - \frac{1}{2} h_i S_i (T_{i_1} - 2T_s)}{\pi\rho C_p r_{i_2}^2 v_{i_2} + \frac{1}{2} h_i S_i} . \qquad (35)$$

The inlet water temperature for the first increment is known, and the outlet (and thus inlet) water temperatures for each increment may be calculated from eq 35.

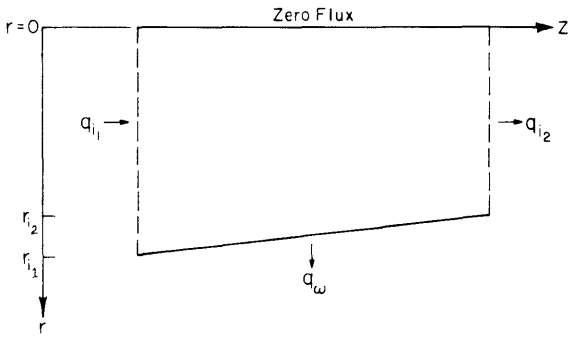


Figure 36. Heat flow into and out of an increment of fluid.

So far for this time step the heat transfer coefficient and bulk water temperature for each increment along the pipe have been determined. Now the program determines the new location of the ice/water interface and calculates the temperature distribution in the ice. For this problem the boundary condition described in eq 2 becomes

$$L \frac{d\underline{s}}{dt} = (k\underline{\nabla T})_f - \underline{q} \qquad (36)$$

where, for each increment along the interface, $q = h_i S_i (T_{i_w} - T_s)$.

Results

The program was used to simulate freezing in two lengths of pipe. For both lengths the pipe radius was 0.6 cm and the pipe temperature was -18°C. For each length, situations where the ice thickness reaches a steady state and where the pipe freezes shut were modeled. The fluid properties were

$k = 1.34 \times 10^{-3}$ cal/cm s °C
$\rho = 1.0$ g/cm^3_o
$C_p = 1.0$ cal/g°C
$\upsilon = 1.79 \times 10^{-2}$ cm^2/s

and the ice properties were

$k = 5.28 \times 10^{-3}$ cal/cm s °C
$C = 0.4459$ cal/cm^3 °C
$L = 80$ cal/cm^3 .

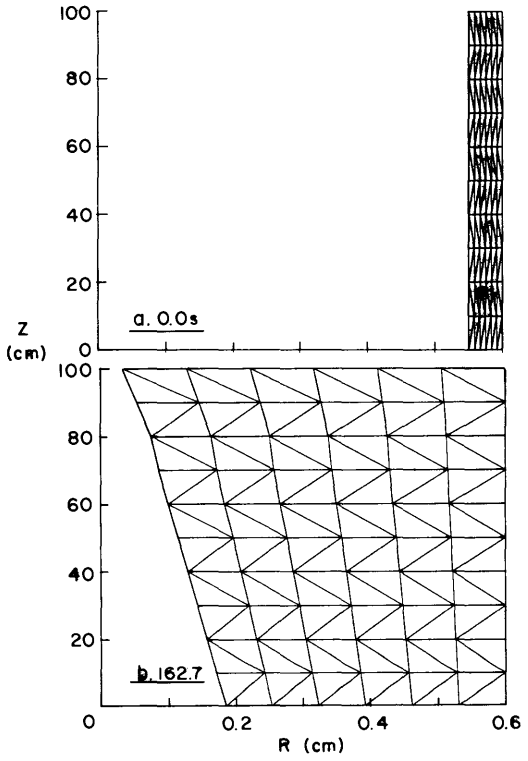


Figure 37. Initial mesh and mesh just before freeze-up for the 1-m pipe.

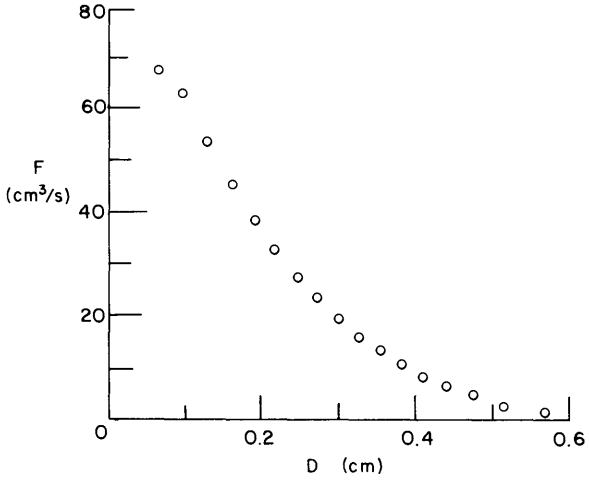


Figure 38. Flow rate F vs ice thickness D at the end of the 1-m pipe, in the case where the pipe freezes up.

The first length of pipe considered was 1.0 m long. An initial ice thickness of 0.05 cm was assumed; the initial mesh is shown in Figure 37a. The total head drop over the length was 30 cm. First an inlet water temperature of 8°C was assigned. Freezing progressed rapidly; Figure 37b shows the ice configuration just before freeze-up. For each time step the flow rate of the outlet water was recorded and is plotted against the ice thickness at the outlet in Figure 38. As the pipe freezes up, the flow rate goes to zero. In Figure 39 the convective heat flux from the water at the end of the pipe and heat flux through the ice there are plotted against the thickness of the ice.

Next, for the same length of pipe and head drop an inlet water temperature of 15°C was assigned. This time the ice formation came to a steady state configuration (Fig. 40). In Figure

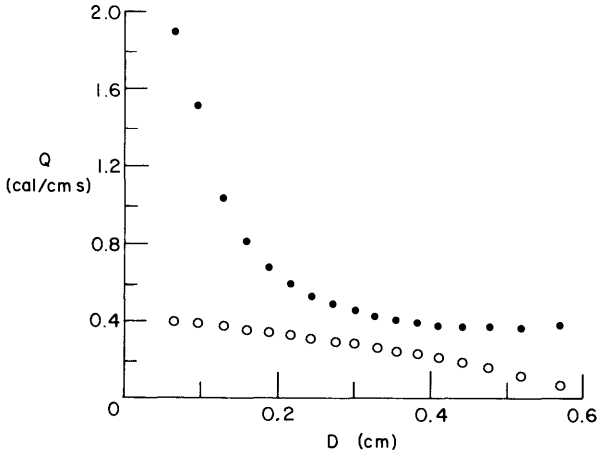


Figure 39. Heat flux Q vs ice thickness D at the end of the 1-m pipe, in the case where the pipe freezes up. The closed circles represent flux from the ice; the open circles represent flux from the water.

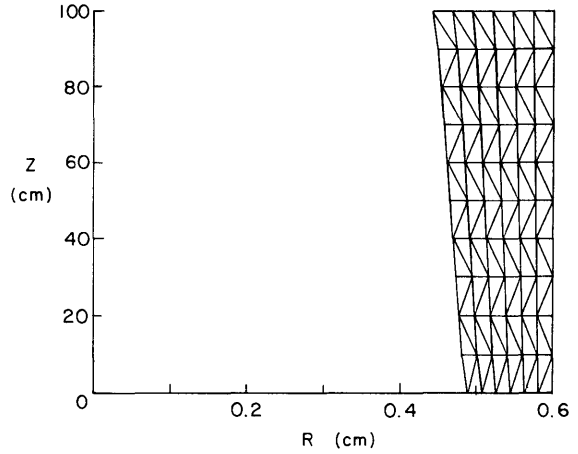


Figure 40. Final mesh, a steady-state ice profile.

41 the flow rate at the outlet of the pipe is plotted against ice thickness; it comes to a steady state value near 41.5 cm^3/s. The convective flux and the heat flux through the ice balance as steady state approaches; this can be seen for the outlet of the pipe in Figure 42.

To obtain a more two-dimensional ice profile, a 5-m-long pipe was also modeled. As before, the initial ice thickness was assumed to be 0.05 cm (Fig. 43a). For this pipe the head drop was first assigned to be 30 cm (the same as for the shorter pipe) and the inlet water temperature, 60°C. Under these conditions the pipe froze shut. The mesh at an intermediate time is shown in Figure 43b and the mesh just before freeze-up is shown in Figure 43c. As can be seen from Figures 44 and 45 the flow rate goes to zero at the outlet as the pipe freezes shut, and the fluxes do not approach a single value.

To demonstrate a situation where the ice formation approaches a steady value for the 5-m pipe, a head drop of 400 cm and an inlet water temperature of 30°C were assigned. The same initial mesh was used. The mesh at steady state is shown in Figure 46. (Note that the ice at the inlet of the pipe section melted from its original location.) Figures 47 and 48 show that the flow rate and heat fluxes come to steady values.

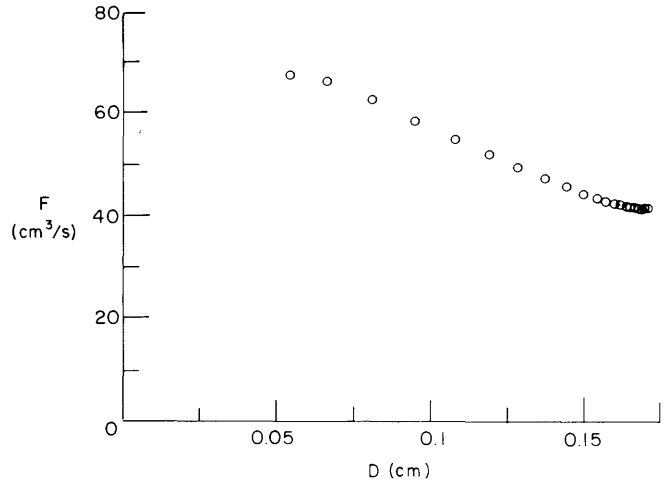


Figure 41. Flow rate F vs ice thickness D at the end of the 1-m pipe, where the ice profile achieved a steady state.

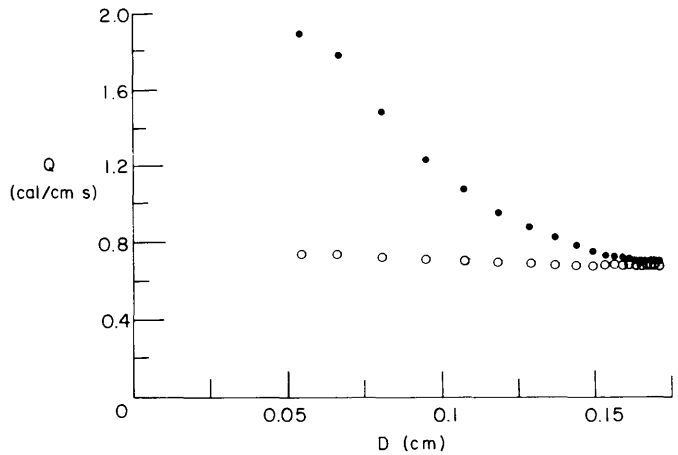


Figure 42. Heat flux Q vs ice thickness D at the end of the 1-m pipe, where the ice profile achieved a steady state. The closed circles represent flux from the ice; the open circles represent flux from the water.

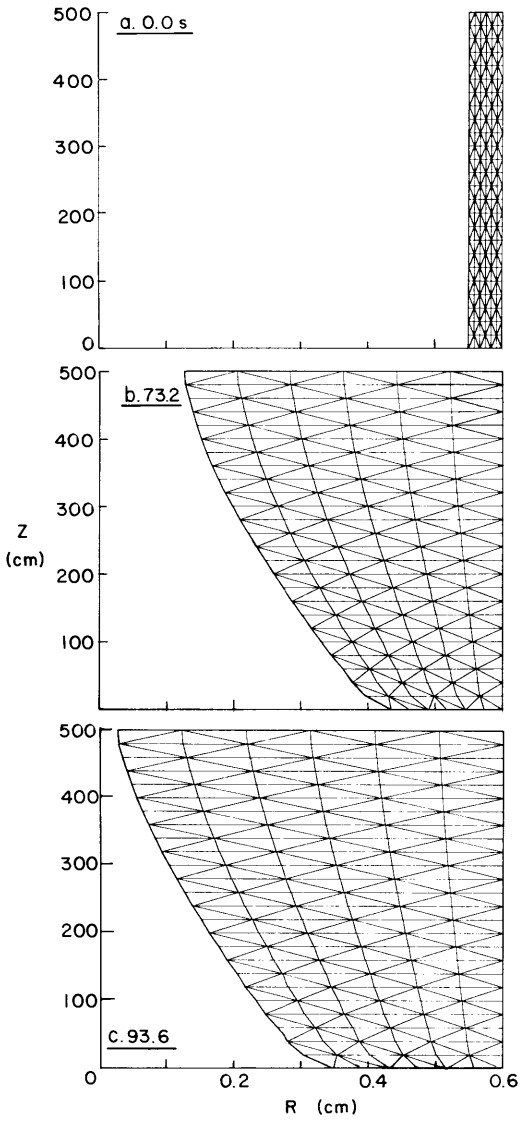


Figure 43. Initial mesh, intermediate mesh, and mesh just before freeze-up for the 5-m pipe.

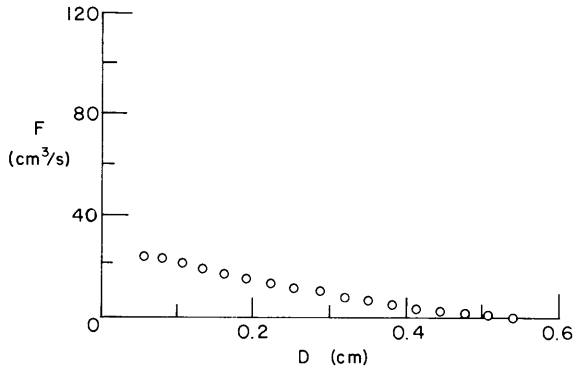


Figure 44. Flow rate F vs ice thickness D at the end of the 5-m pipe, in the case where the pipe freezes up.

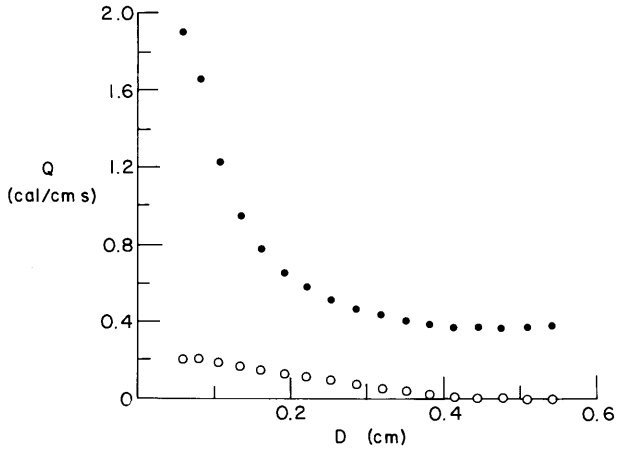


Figure 45. Heat flow Q vs ice thickness D at the end of the 5-m pipe when freeze-up occurs. The closed circles represent flux from the ice; the open circles represent flux from the water.

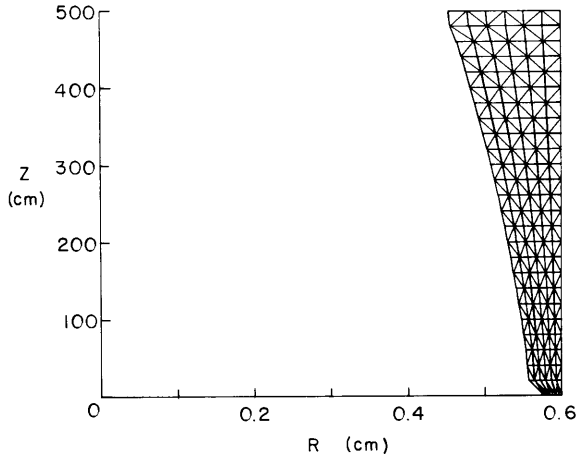


Figure 46. Final mesh for the 5-m pipe, a steady-state ice profile.

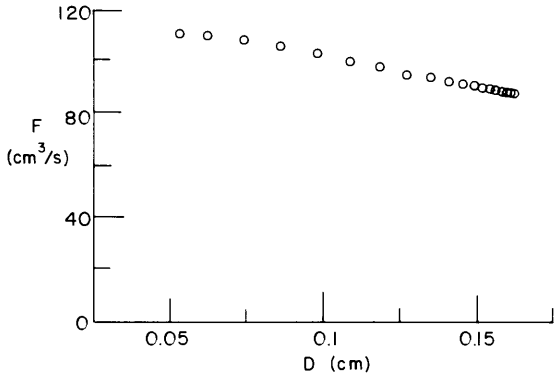


Figure 47. Heat flux F vs ice thickness D for the 5-m pipe where a steady flow is attained.

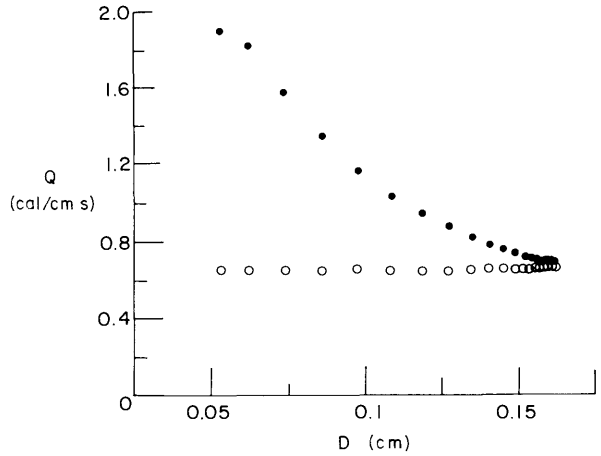


Figure 48. Heat flux Q vs ice thickness D for the 5-m pipe where a steady state flow occurs. The closed circles represent flux from the ice; the open circles represent flux from the water.

VON NEUMANN ANALYSIS OF THE NUMERICAL METHOD

The von Neumann stability analysis is a technique commonly used for investigating the stability of a numerical method. In performing this analysis one examines the behavior of Fourier series components as they are propagated numerically and analytically. In this section the analysis will be conducted for the numerical solution of the heat conduction equation on a moving mesh. The heat conduction equation, when solved using a moving-mesh technique, acquires a velocity term that gives the equation the same form as the convection-diffusion equation. The seasoned analyst will immediately perceive this in eq 8 presented earlier. Now an alternative method of casting the heat conduction equation into the form of a convection-diffusion equation will be discussed. Lynch (1982a) shows that this leads to the same mathematical formulation as in eq 8. For simplicity, only the one-dimensional form of the equations will be considered:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \; . \tag{37}$$

For the moving-mesh solution, temperature is a function of both time and a moving location in space. Let x be a coordinate with respect to a fixed reference frame, and let x_0 be a coordinate that is attached to a reference point in space, whose motion will be observed:

$$T = T = T[x(x_0,t),t] \; . \tag{38}$$

From the chain rule,

$$\left(\frac{\partial T}{\partial t}\right)_{x_0} = \left(\frac{\partial T}{\partial t}\right)_x + \left(\frac{\partial T}{\partial x}\right)_t \left(\frac{\partial x}{\partial t}\right)_{x_0} \; . \tag{39}$$

Solving for the time derivative of temperature while holding x constant gives

$$\left(\frac{\partial T}{\partial t}\right)_x = \left(\frac{\partial T}{\partial t}\right)_{x_0} - \left(\frac{\partial T}{\partial x}\right)_t \left(\frac{\partial x}{\partial t}\right)_{x_0} \; . \tag{40}$$

Here, $(\partial T/\partial t)_x$ is the time derivative of temperature in the heat conduction equation, so substituting this result into eq 37 yields

$$\left(\frac{\partial T}{\partial t}\right)_{x_0} - \left(\frac{\partial T}{\partial x}\right)_t \left(\frac{\partial x}{\partial t}\right)_{x_0} - \alpha \frac{\partial^2 T}{\partial x^2} = 0 \,. \tag{41}$$

The term $\partial^2 T/\partial x^2$ represents the second derivative of temperature, holding t constant. Note that $(\partial x/\partial t)_{x_0}$ represents the mesh velocity.

This form of the equation is nondimensionalized by defining the following:

$$\xi = \frac{x}{\ell}$$

$$d\tau = \frac{\alpha}{\ell^2}\, dt \tag{42}$$

$$\theta = \frac{T}{\Delta T}$$

where ℓ is a characteristic length (i.e. element length) and ΔT is a characteristic temperature range. Since the element length ℓ is continuously changing in the solution of the problem, there is no fixed proportion between t and τ, but only between dt and $d\tau$. Then the nondimensionalized form of eq 41 is

$$\frac{\partial \theta}{\partial \tau} - Pe \frac{\partial \theta}{\partial \xi} - \frac{\partial^2 \theta}{\partial \xi^2} = 0 \tag{43}$$

where Pe is the Peclet number $[(\ell\, dx/dt)/\alpha]$ and dx/dt represents the mesh velocity. In general the Peclet number is a function of both space and time. However, in some freezing problems the front progresses as the square root of time and the mesh movement may be proportional to the phase front movement, so the element length ℓ will grow as the square root of time ($\ell \propto \sqrt{t}$). The mesh velocity will also be proportional to time ($dx/dt \propto 1\sqrt{t}$). Since the mesh velocity is multiplied by the element length in the definition of Pe, the dependence of Pe on time almost vanishes when the mesh movement is proportional to the phase front movement.

In the numerical solution, time will be divided into $\Delta\tau$, and space will be divided into $\Delta\xi$. It is assumed that the elements are all the same size, although that size changes through time. Then, because ℓ is defined as the element length, $\Delta\xi = \ell \equiv 1$. This means that the system is always analyzed relative to its current scale. The Peclet number is assumed to be locally constant. Note that the von Neumann analysis will pertain to general diffusion problems on a moving mesh and does not depend upon any criterion as to what makes the mesh move.

Pinder and Gray (1977) present a von Neumann stability analysis of the convection-diffusion equation. For the finite element solution using linear elements, they derive the expression for the eigenvalue λ_n' for the numerical solution to the equation. In terms of the nondimensional eq 43, the eigenvalue is

$$\lambda_n' = \frac{1/3[2 + \cos(2\pi/\delta_n)] - (1-\epsilon)\left\{-Pe\ \Delta\tau\ i\ \sin(2\pi/\delta_n) + 2\Delta\tau[1-\cos(2\pi/\delta_n)]\right\}}{1/3[2 + \cos(2\pi/\delta_n)] + \epsilon\left\{-Pe\ \Delta\tau\ i\ \sin(2\pi/\delta_n) + 2\Delta\tau[1-\cos(2\pi/\delta_n)]\right\}} \tag{44}$$

where δ_n is the nondimensional wavelength, which is the wavelength divided by the element length, and ϵ represents the degree of implicitness of the numerical solution. Their analysis also includes expressions for the phase lag and amplitude ratios between the Fourier components of the numerical and analytical solutions at a time when a wave has progressed one wavelength.

To provide guidance in choosing the time step and to give a general indication of the accuracy of the solution, the Peclet number can be associated with the Stefan number, which controls the rate of freezing. To this end, a one-dimensional form of eq 2 is examined, assuming that the unfrozen phase is at the phase change temperature:

$$L \frac{ds}{dt} = k \frac{\partial T}{\partial x} . \tag{45}$$

The nondimensional form of this equation is

$$Pe_f = S_t \frac{\partial \theta}{\partial \xi} \tag{46}$$

where Pe_f represents the Peclet number at the phase front and S_t is the Stefan number ($S_t = C\,\Delta T/L$). The term $\partial\theta/\partial\xi$ represents the gradient near the front. Since ΔT is chosen to be the temperature difference between the phase front and a stationary constant temperature boundary, $\Delta\theta$ over the frozen zone equals one. Since $\Delta\xi \equiv 1$ for one element, $\Delta\xi$ over the frozen zone is equal to M, where M is the number of elements over the frozen zone. Then $\partial\theta/\partial\xi = 1/M$, and eq 46 becomes

$$Pe_f \approx \frac{S_t}{M} . \tag{47}$$

With finite boundaries the fastest mesh motion is located at the front, so the maximum Peclet number will be $Pe \approx S_t/M$. In applications involving ground freezing, for example, a large value for S_t would be on the order of 1. Therefore, $Pe_f < 1$.

In the case of infinite boundaries the largest Pe will typically occur at the boundary that approximates the infinite condition. In a problem where the medium is freezing into an infinite unfrozen region, if the distance to the infinite boundary is always kept at about the same ratio to the frozen thickness, then proportionally greater stretching and greater mesh velocities occur towards infinity. In this case, over the unfrozen zone, dx/dt is increased by some factor f_1, and the element size is increased by some factor f_2 relative to their values at the front. The resulting Peclet number will be increased:

$$Pe_i \approx f_1 f_2 \frac{S_t}{M} . \tag{48}$$

In the cases examined here, $f_1 f_2$ ranged between 1 and 400.

Thus, the Stefan number characteristic of the problem determines an approximate range of Peclet numbers that may be encountered in the numerical solution. The size of the Peclet number will vary over the mesh, but the largest values are anticipated at the front if there are no semi-infinite boundaries or at moving semi-infinite boundaries should they occur.

The choice of time step is also not arbitrary. For $Pe \approx 1$ or larger, convection in the problem is significant. (The word "convection" is used here to designate apparent convection effects in the numerical solution, which result from the inclusion of the mesh velocity term in the governing equation.) Pe represents the nondimensional mesh velocity. To keep the time step from being too large, reasonable bounds are taken as $0 < \Delta\tau < 1/Pe_{max}$. (This prevents the solution from passing over an entire element in one time step.) For $Pe \ll 1$, diffusion dominates the equation, and a conservative estimate for $\Delta\tau$ is 0.1. (This corresponds to the nondimensional time it takes for the effect of a step temperature change at a boundary to be felt across an element.)

The Fourier series components propagate analytically at a wave speed of Pe; however, the numerical method will propagate different components at different speeds. To examine the amplitude ratios and phase shifts after the solution has passed through one element, it is assumed here that the solution is characterized by a nondimensional velocity equal to Pe.

Then the number of time steps required to move the solution through one element is $|1/(Pe\Delta\tau)|$. The amplitude ratio R, which is the ratio of the computed amplitude to the analytical amplitude, becomes

$$R = \left[\frac{\lambda'_n}{\exp\left\{-(2\pi/\delta_n)^2\ \Delta\tau\right\}}\right]^{\left|\frac{1}{Pe\Delta\tau}\right|} \tag{49}$$

and the phase shift Θ between the computed phase and the analytical phase is :

$$\Theta = \left|\frac{1}{Pe\Delta\tau}\right| \theta'_n - 2\pi/\delta_n \ . \tag{50}$$

Several Stefan numbers were chosen as a basis for investigations. Compared to ground freezing or the freezing of water, freezing occurs slowly for $S_t = 1/80$, which represents freezing of water where the temperature difference across the frozen zone is 1°C. Rapid freezing will occur for $S_t = 1$, which would represent freezing in a fairly dry soil with $\Delta T > 1$. From these Stefan numbers eq 48 is used to find the corresponding Peclet number. In all cases it is assumed that there are $M = 5$ elements across the frozen zone. Then, for $S_t = 1/80$, $Pe = 0.0025$ when $f_1 f_2 = 1$, and $Pe = 1$ when $f_1 f_2 = 400$. For $S_t = 1$, $Pe = 0.2$ when $f_1 f_2 = 1$, and $Pe = 80$ when $f_1 f_2 = 400$. For the two extreme cases of $Pe = 0.0025$ and $Pe = 80$ and for the intermediate case when $Pe = 1$, the eigenvalues, amplitude ratios and phase shifts (as defined in eqs 44, 49 and 50, respectively) are examined as a function of wavelength δ (Figs. 49-51).

A system is stable when the eigenvalues defined above do not exceed 1. For all values of Pe and $\Delta\tau$ examined here the system is stable for $0.5 \leqslant \epsilon \leqslant 1.0$, as can be seen in Figures 49-51. In several cases, such as when $Pe = 1$ and $\Delta\tau = 0.1$, the system is stable for all values of ϵ. This indicates that $\Delta\tau$ in those cases can be considered a tiny time step for the Peclet number. Thus, for $0.5 \leqslant \epsilon \leqslant 1.0$ the moving mesh solution is stable numerically.

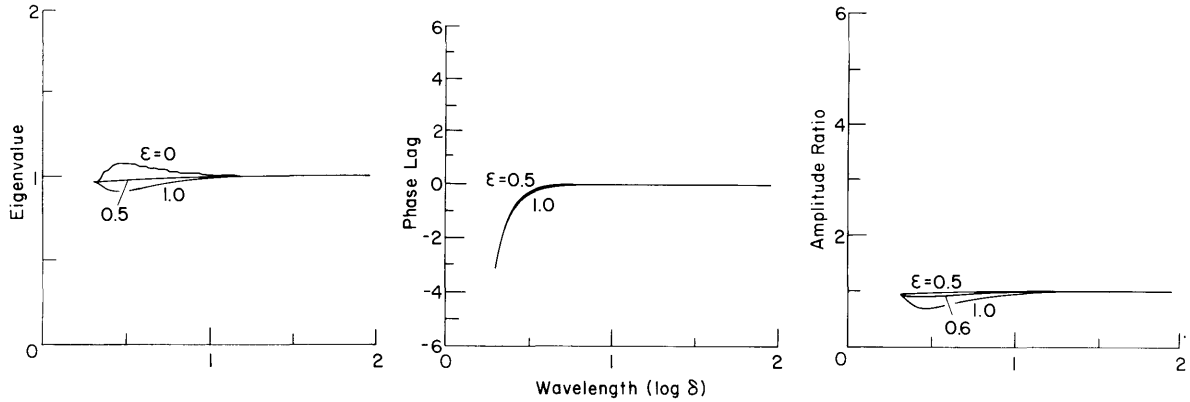
Because the frequency content varies from problem to problem, it is impossible to make specific recommendations about the optimal choice of $\Delta\tau$, for example, but general trends may be observed and used as rule-of-thumb guidelines.

First consider the case where convection is significant ($Pe = 80$). Figure 49 shows the behavior when $\Delta\tau = 1/4Pe$, $\Delta\tau = 1/Pe$ and $\Delta\tau = 4/Pe$. When $\epsilon = 0.5$, the analytical and numerical Fourier components systems tend to converge faster to values of 0 and 1 for shorter wavelengths, in the phase lag and amplitude ratio plots, respectively, than higher values of ϵ. However, the shorter wavelengths, which have significant phase lags, are more damped (have lower amplitude ratios) for $\epsilon = 1.0$ than for lower values of ϵ; thus, the numerical solution may show less distortion if a higher value of ϵ is used. As $\Delta\tau$ is increased from $1/4Pe$, longer wavelengths become affected by the mismatch of the analytical and numerical Fourier components.

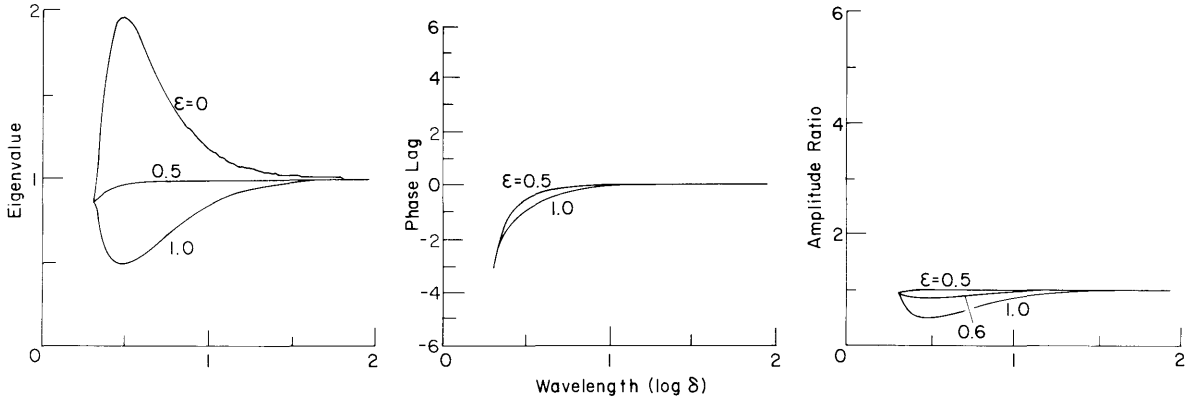
Now consider the intermediate case where $Pe = 1$. Figure 50 illustrates the cases for $\Delta\tau = 1/4Pe$, $\Delta\tau = 1/Pe$, $\Delta\tau = 4/Pe$ and $\Delta\tau = 0.1$. As was the case when $Pe = 80$, increasing $\Delta\tau$ makes the system less accurate for longer wavelengths. However, now the shorter wavelengths with significant phase lags are more damped for $\epsilon = 0.5$ than for $\epsilon = 1.0$.

Figure 51 illustrates the situation when $Pe = 0.0025$ and $\Delta\tau = 0.1$, $\Delta\tau = 1.0$ and $\Delta\tau = 10.0$. As with $Pe = 1$, shorter wavelengths with significant phase shifts are more damped for $\epsilon = 0.5$ than for higher ϵ values. Thus, it appears that when conduction is significant and convection is of equal or less significance, choosing a value of ϵ closer to 0.5 than to 1.0 may yield a better numerical result.

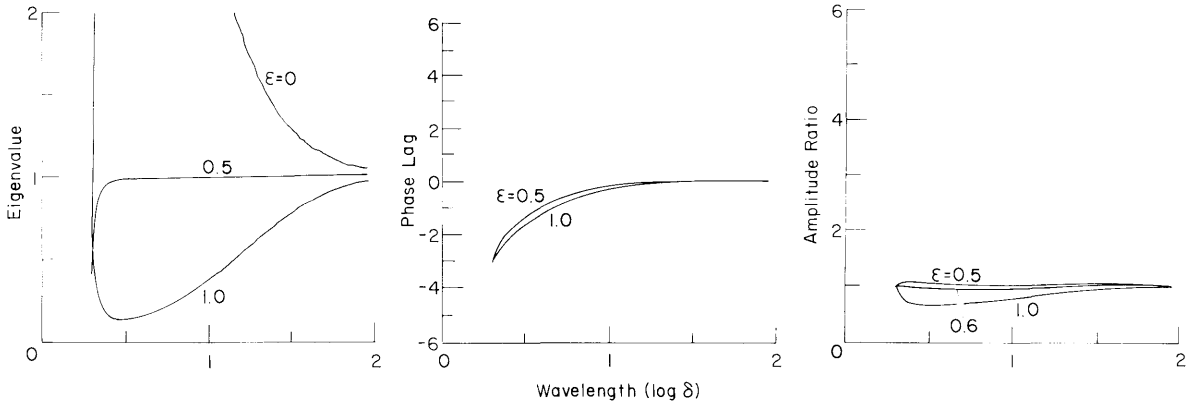
Over the entire range of Peclet numbers examined, the shorter wavelengths with significant phase lags are generally more damped for lower values of Pe, so that the overall solution may be better for lower values of Pe and reasonable values of $\Delta\tau$ than for higher values of Pe. Thus, in freezing problems on a moving mesh the highest value of Pe must be considered in assessing the potential numerical difficulties that may be encountered. For finite boundaries the highest value of Pe will occur at the freezing front. However, where moving infinite boundaries are modeled it is more likely that higher values of Pe may occur there than at the phase front.



a. $\Delta\tau = 1/4Pe = 0.003125.$



b. $\Delta\tau = 1/Pe = 0.01250.$



c. $\Delta\tau = 4/Pe = 0.050.$

Figure 49. Eigenvalue, phase lag and amplitude ratio vs wavelength when Pe = 80. The abscissa is the base ten logarithm of the nondimensional wavelength δ.

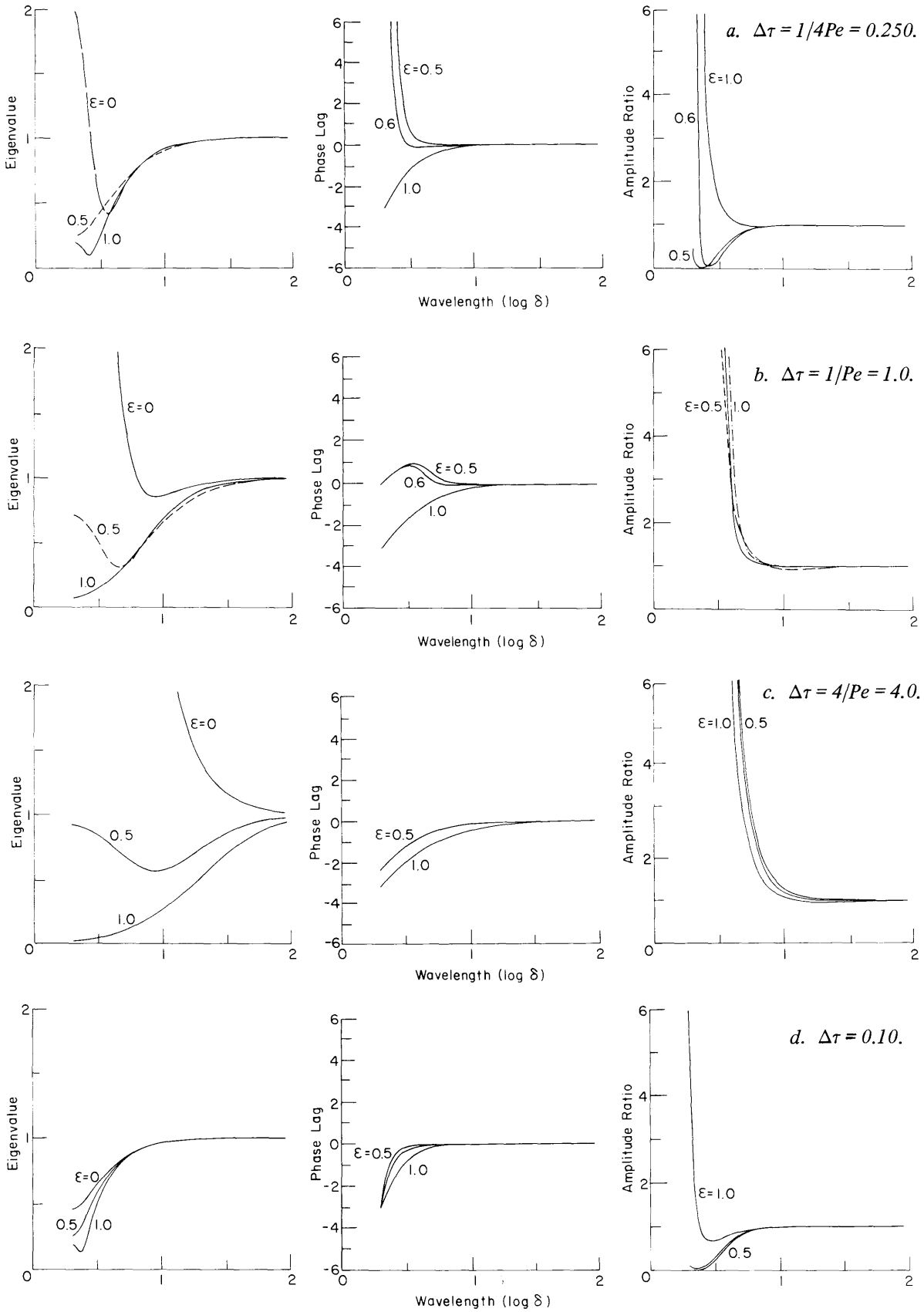
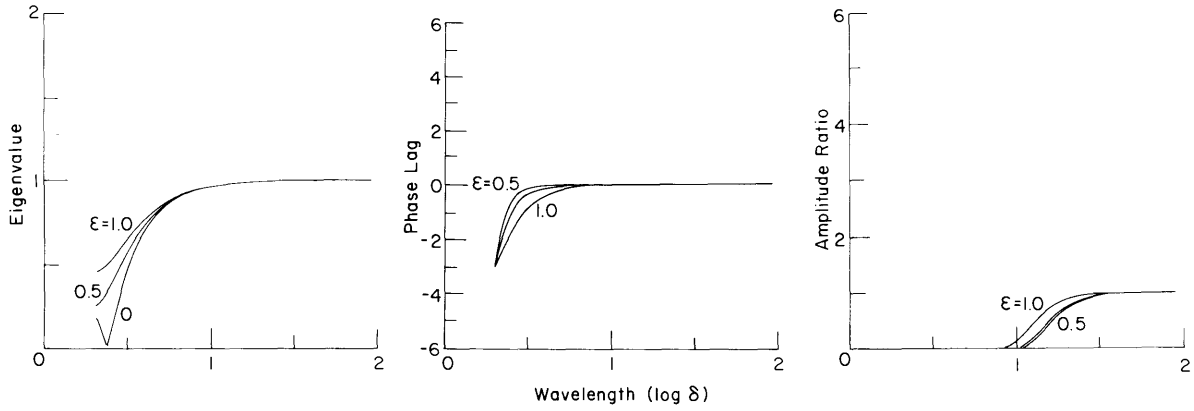
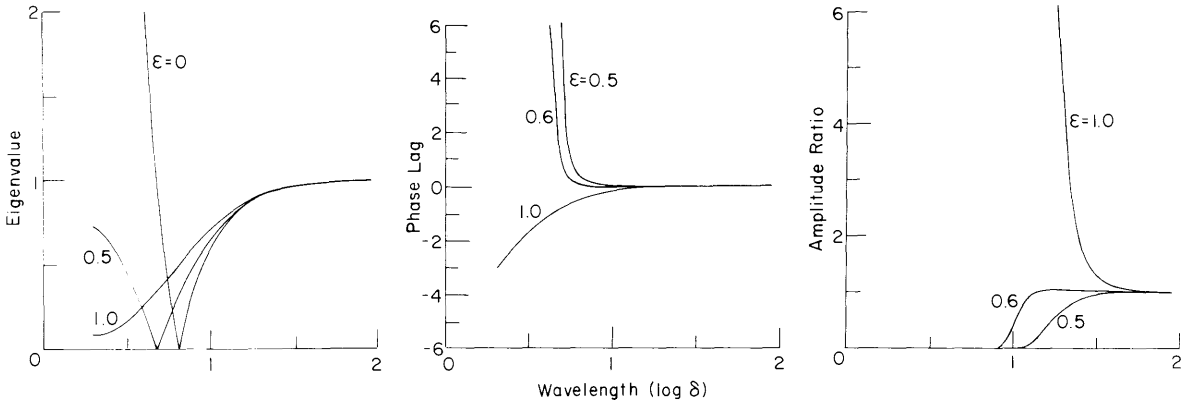


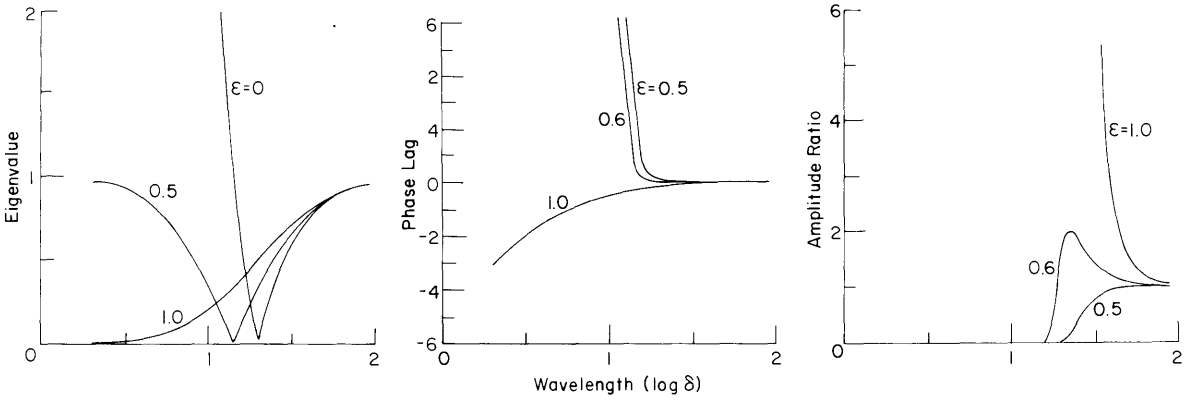
Figure 50. Eigenvalue, phase lag and amplitude ratio vs wavelength when Pe = 1.0. The abscissa is the base ten logarithm of the nondimensional wavelength δ.



a. $\Delta\tau = 0.1$.



b. $\Delta\tau = 1.0$.



c. $\Delta\tau = 10.0$.

Figure 51. Eigenvalue, phase lag and amplitude ratio vs wavelength when Pe = 0.0025. The abscissa is the base ten logarithm of the nondimensional wavelength δ.

Now, for comparison with these results, Peclet numbers present in the numerical solutions run here and elsewhere will be examined. First consider the two-phase corner simulation presented earlier.

In the run using the mesh depicted in Figure 27 unacceptable distortions of the temperature solution occurred in the triangular unfrozen region. In this case the largest element used coincides with the location of the highest mesh velocity. The Peclet number at the corner farthest from the phase front had a value of $Pe_i = 0.41$. Both of these results were calculated from numerical results using the definition of the Peclet number: $Pe = (\ell v/\alpha)$, where ℓ is the element length, v is the mesh velocity, and α is the thermal diffusivity.

These actual values agree fairly well with values that would be predicted from the Stefan number. In this problem, $S_t = C\Delta T/L = 4$. Then, using eq 47, $Pe_f = S_t/M = 0.8$. This is on the same order of magnitude as $Pe_f = 0.41$ actually encountered. Since the element at the semi-infinite corner has an initial length (along the 45° line of symmetry) of 2.9 cm and since the corresponding length for the element at the phase boundary is 0.12 cm, then $f_2 = 24$. The ratio of the velocity of the nodes there remained constant throughout the simulation at a value of $f_1 = 8$. Then the Peclet number predicted by eq 48 is $Pe_i = f_1 f_2 S_t/M = 154$. This is in fair agreement with the Peclet number of 79 present at the end of the run. It was recommended that the maximum time step taken be less than $\Delta\tau_{max} = 1/Pe$; using the predicted value of Pe, this yields $\Delta\tau_{max} = 6.5\times10^{-3}$. From actual values at the semi-infinite corner and the definition of $\Delta\tau$

$$\Delta\tau_{max} = \frac{\alpha}{\ell^2}\,\Delta\tau_{max} = \frac{1.34\times10^{-2}}{24^2}(200) = 4.6\times10^{-4}\ .$$

This illustrates that the largest time step used was still very conservative.

Now consider the two-phase corner simulation that was modeled using the initial mesh illustrated in Figure 22. The largest Peclet number in that situation also occurred at the semi-infinite corner. Now, however, because the element there is smaller, the largest value actually occurring was $Pe = 32$. ($Pe = 64$ would have been predicted.) The maximum time step taken was $\Delta\tau_{max} = 4.75\times10^{-3}$, an order of magnitude smaller than the smallest time step recommended using predicted values. This simulation yielded very acceptable results.

The one-dimensional simulations presented earlier involved very small Peclet numbers, mainly because the largest velocities were associated with the phase front, and no semi-infinite boundaries were present. In the Neumann run the maximum Peclet number was $Pe_f = 0.011$, and $\Delta\tau_{max} = 15$. (The Stefan number was 0.075.) In the radial comparison the Stefan number was also 0.075, the maximum Peclet number was $Pe_f = 0.02$, and $\Delta\tau_{max} = 24$.

Peclet numbers encountered by Lynch and O'Neill (1981) also will be examined for a finite element moving mesh (one-dimensional) used to model heat conduction without phase change and for one-dimensional two-phase heat conduction with phase change. These Peclet numbers are inferred from parameter values and graphs presented in Lynch (1982a). Also, Hermitian elements were used. One of these elements is assumed here to be roughly equivalent to the use of three linear triangular elements. Then, for the heat conduction without phase change the largest Peclet number is estimated to be $Pe = 0.5$, and the maximum time step is $\Delta\tau_{max} = 0.04$. (This is considerably less than $\Delta\tau = 1/Pe$.) For the heat conduction with phase change, at the phase front $Pe_f = 0.02$, and the largest Peclet number appeared at the semi-infinite boundary, where $Pe_i = 0.4$.

The first observation from these results is that the numerical distortions occurred for the largest numerical Peclet value encountered in these simulations, $Pe = 78$. (The two-phase corner problem was also run for a case where $Pe_{max} = 250$, and numerical distortions occurred.) The Peclet numbers for the one-dimensional problems were all on the order of 1 or less, and no distortions occurred. In the two-dimensional case a Peclet value of 32 occurred with a solution where numerical results were acceptable. However, for the simulations where Peclet values of 79 and higher were observed, numerical difficulties were present. The von Neumann analysis supports this observation; phase shift and amplitude ratios generally get worse as the Pe number is increased.

The von Neumann analysis also indicates that the solution will be less accurate as the time steps increase. However, the time steps encountered here were all on the small end of the scale of time steps investigated in the analysis. It was observed in the unacceptable two-phase simulations that decreasing the size of the time steps by a factor of four made almost no difference in the numerical solution.

It has therefore been observed that the occurrence of high Peclet numbers in the finite element moving-mesh solution to the heat conduction equation may be linked with numerical distortion of the temperature distribution. The limits of this occurrence when a moving mesh is used will be the subject of further research.

COMPARISON OF COMPUTATIONAL EFFORT: TRANSFINITE MAPPINGS VS EQUATIONS OF ELASTICITY

A topic of concern in moving-mesh methods is specifying the evolution of the interior mesh as the solution progresses. Not only should the geometry of the mesh be satisfactory, but the method used to specify the geometry should be computationally efficient. In this section the number of operations required to generate a square mesh using transfinite mappings is compared to the number required if the mesh is specified by solution of the equations of elasticity. Lynch (1982a) discusses a two-dimensional moving-mesh solution in which he recurrently solves the equations of elasticity to specify interior node motion, but the number of computations required are not discussed.

The transfinite mappings will be examined first. The nodes in the sides and corners of the mesh have known locations, which were specified either by movement of the phase front or by some other designated positioning along the boundaries. For simplicity, consider a square region where there are N+1 elements along the side; then there are N (non-corner) nodes along each side. Thus, locations of N^2 interior nodes must be specified. From eq 23 and the assumption that u and v are specified by the preferred method discussed earlier [so that the coefficients such as $(1-v)$ are stored], it is evident that the method will require 24 N^2 multiplications and 14 N^2 additions. The number of operations is on the order of N^2 because there are N^2 interior nodes, and the location of each interior node depends only on eight boundary node locations. This number is required regardless of whether large or small boundary excursions have occurred. Storage of the edge nodes and their coefficients requires $8N$+8 locations.

To solve the equations of elasticity the solution of a banded matrix is required. Again with a square grid with N elements along each side the matrix will be $N^2 \times (N+3)$ if square elements are used. Obviously the computational effort required depends on which of the many matrix solution techniques are used.

Bettess (1981) estimates that the number of arithmetic operations necessary for solving a banded matrix is $(N+1)^2(N+3)^2/2$. However, he does not indicate which matrix solution technique this represents. Westlake (1975) lists operation counts for a variety of techniques, all used to solve a full $N^2 \times N^2$ matrix; the number of operations ranges from on the order of N^4 to N^6.

Lynch (1982a) uses subroutine Solve, which was mentioned earlier in the discussion of Femove. Solve solves a banded matrix using the Doolittle method to upper-triangularize the matrix, then modifies the vector on the right side, and performs the back substitution. If the orientation of boundaries where tangential motion is allowed is constant, then the triangularization need only be performed once. This represents the best case for minimum computations for this method. The worst case would be to perform the whole process for each time step. From Solve and the assumption that the computer effort for all arithmetic operations is equal (in fact, the level of effort is machine-dependent), it may be seen that the upper triangularization requires operations on the order of N^3, and that modifying the load vector and the back substitution requires operations on the order of N^2. The storage requirement is $N^2 \times NHB$, where NHB represents the half-bandwidth. If the nodes of the square grid are labeled consecutively row-by-row, $NHB = (N+3)/2$. Thus, the storage requires a number of locations on the order of N^3.

Both the transfinite mapping and solution of the equations of elasticity, then, require on the order of N^2 operations for the best case in solving the equations of elasticity (which is also a special case). However, the elasticity conditions require substantially more storage space than the mappings. In the worst case using subroutine Solve (when the triangularization needs to be performed for each time step) the equations of elasticity require N^3 computations, substantially more than the transfinite mappings. The method of transfinite mappings, then, is at least as efficient computationally as the solution of the equations of elasticity, and in many cases it is far more efficient, depending on the procedure used in matrix solution for the equations of elasticity. The method of transfinite mappings also requires substantially less storage space.

CONCLUSIONS

It has been demonstrated that the use of transfinite mappings in conjunction with a moving-mesh finite element method yields readily acceptable results for modeling freezing phase change in conduction heat transfer. A program was developed that uses this method; it is capable of modeling two-dimensional situations, either in Cartesian or (r, z) cylindrical coordinates. Solutions generated using this method compare well to analytical solutions. Excellent results were obtained when modeling a two-dimensional situation for which experimental results were available. The usefulness of having the phase boundary always coincident with element boundaries was demonstrated by modeling the freezing of flow in a pipe.

Some subtleties and limitations of the use of transfinite mappings in mesh generation were investigated. In general the method seems to provide very good results. In modeling phase change the use of transfinite mappings represents the most efficient general means of specifying interior mesh motion used to date.

When using a moving finite element mesh to model heat conduction, care must be taken to ensure that the Peclet number based on mesh velocity does not become too large anywhere in the mesh, or numerical distortions of the temperature solution may occur. The Peclet number was linked to the Stefan number, and some very general guidance was given so that the degree of numerical difficulty can be estimated before running the program.

LITERATURE CITED

Albert, M.R. (1983) Computer models for two-dimensional transient heat conduction. U.S.A. Cold Regions Research and Engineering Laboratory, CRREL Report 83-12. AD A134893.

Bettess, P. (1981) Operation counts for boundary integral and finite element methods. *International Journal for Numerical Methods in Engineering,* **16**: 519-528.

Bonacina, C. and G. Comini (1973) On the solution of the nonlinear heat conduction equations. *International Journal of Heat and Mass Transfer*, **16**: 581-589.

Bonnerot, R. and P. Jamet (1977) Numerical computation of the free boundary for the two-dimensional Stefan problem by space-time finite elements. *Journal of Computational Physics,* **25**: 163-181.

Budhia, H. and F. Kreith (1973) Heat transfer with melting or freezing in a wedge. *International Journal of Heat and Mass Transfer,* **16**: 195-211.

Buell, W.R. and B.A. Bush (1973) Mesh generation–A survey. *Journal of Engineers in Industry, Transactions of the American Society of Mechanical Engineers,* **95**(B): 1-15.

Carslaw, H.S. and J.C. Jaeger (1959) *Conduction of Heat in Solids.* London: Oxford University Press.

Crank, J. and R.D. Prahle (1973) Melting ice by the isotherm migration method. *Bulletin of the Institute for Mathematical Applications,* **9**: 12.

Denayer, A. (1979) Automatic generation of finite element meshes. *Computer Structures,* **9**: 359-364.

Dusinberre, G.M. (1949) *Numerical Analysis of Heat Flow.* New York: McGraw-Hill.

Epstein, M. and F.B. Cheung (1982) On the prediction of pipe freeze-shut in turbulent flow. *Journal of Heat Transfer,* **104**: 381-384.
Gartling, D.K. Finite element analysis of convective heat transfer problems with change of phase. In *Numerical Methods in Laminar and Turbulent Flows* (C. Taylor, K. Morgan, C. Brebbia, Eds.). London: Pentech Press.
Gilpin, R.R. (1979) The morphology of ice structure in a pipe at or near transition Reynolds numbers. *American Institute of Chemical Engineers, Symposium Series 189,* **75**: 89-94.
Gordon, W.J. and C.A. Hall (1973) Construction of curvilinear coordinate systems and application to mesh generation. *International Journal for Numerical Methods in Engineering,* 7: 461-477.
Haber, R., M.S. Shepard, J.F. Abel, R.H. Gallagher and D.P. Greenberg (1981) A general, two-dimensional, graphical finite element preprocessor utilizing discrete transfinite mappings. *International Journal for Numerical Methods in Engineering,* **17**: 1015-1044.
Herrmann, L.R. (1976) Laplacian-isoparametric grid generation scheme. *Journal of the Engineering Mechanics Division, American Society of Civil Engineers,* **102**: 749-756.
Karlekar, B.B. and R.M. Desmond (1977) *Engineering Heat Transfer.* Boston: West Publishing Company.
Landau, H.G. (1950) Heat conduction in a melting solid. *Quarterly of Applied Mathematics,* 8: 81-84.
Lynch, D.R. (1982a) Unified approach to simulation on deforming elements with application to phase change problems. *Journal of Computational Physics,* **47**: 387-411.
Lynch, D.R. (1982b) Subroutine SOLVE, A banded matrix solver. Unpublished paper, Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire.
Lynch, D.R. and K. O'Neill (1981) Continuously deforming finite elements for the solution of parabolic problems, with and without phase change. *International Journal for Numerical Methods in Engineering,* **17**: 81-96.
O'Neill, K. (in press) Boundary integral equation solutions for moving boundary phase change problems. *International Journal for Numerical Methods in Engineering.*
O'Neill, K. and D.R. Lynch (1981) A finite element solution for freezing problems, using a continuously deforming coordinate system. In *Numerical Methods in Heat Transfer* (R. Lewis, K. Morgan, O.C. Zienkiewicz, Eds.). New York: John Wiley and Sons.
Petukhov, B.S. and V.N. Popov (1963) Theoretical calculation of heat exchange and frictional resistance in turbulent flow in tubes of an incompressible fluid with variable physical properties. *High Temperature,* **1**: 15-19.
Pinder, G.F. and W.G. Gray (1977) *Finite Element Simulation in Surface and Subsurface Hydrology.* New York: Academic Press.
Saitoh, T. (1978) Numerical method for multi-dimensional freezing problems in arbitrary domains. *Journal of Heat Transfer,* **100**: 294-299.
Sparrow, E.M., S.V. Patankar and S. Ramadhyani (1977) Analysis of melting in the presence of natural convection in the melt region. *Journal of Heat Transfer,* **99**: 520-526.
Westlake, J.R. (1975) *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations.* Huntington, New York: Krieger Publishing Company.
Zienkiewicz, O.C. and D.V. Phillips (1971) An automatic mesh generation scheme for plane and curved surfaces by isoparametric coordinates. *International Journal for Numerical Methods in Engineering,* **3**: 519-528.