The Stefan Problem Solved Via Conjugate Gradient—Like Iterative Methods On a Parallel Vector Machine

Anthony T. Chronopoulos, C.R. Swaminathan and V.R. Voller

International Journal of High Performance Computing Applications 1991; 5; 74
DOI: 10.1177/109434209100500307

THE STEFAN PROBLEM SOLVED VIA CONJUGATE GRADIENT–LIKE ITERATIVE METHODS ON A PARALLEL VECTOR MACHINE

Anthony T. Chronopoulos, C. R. Swaminathan, and V. R. Voller

UNIVERSITY OF MINNESOTA
MINNEAPOLIS, MINNESOTA 55455

Summary

The aim of this paper is to illustrate the validity and efficiency of iterative methods for solving large linear systems arising from the finite element discretizations of the equation governing conduction-controlled solidification processes. Starting from the basic enthalpy equation, two alternative formulations are obtained and fixed-grid finite element discretizations are developed. These discretizations yield a set of nonlinear equations that are linearized using the Newton-Raphson scheme. The linearized equations are used as a basis for evaluating different iterative methods of the conjugate gradient type. Symmetric scaling and incomplete factorization preconditioning of the linear equations are used to improve the convergence properties of the iterative methods. Vectorization and parallelization are also employed to make full use of the CRAY-2 supercomputer. The results indicate that the implementation of currently available iterative solvers leads to efficient solution methodologies for phase change problems.

The International Journal of Supercomputer Applications, Volume 5, No. 3, Fall 1991, pp. 74–91.

Introduction

The effective modeling of phase change phenomena is hampered by the difficulty in handling the associated nonlinearities, and as a result, state-of-the-art computational techniques are required (Rappaz, 1989; Voller, Swaminathan, and Thomas, 1990). One of the major nonlinearities arises from the evolution of the latent heat at the solid-liquid interface as the liquid transforms into solid.

Among the popular numerical approaches to model phase change processes are the so-called fixed-grid techniques (Crank, 1984; Voller, Swaminathan and Thomas, 1990). The essential feature of these techniques is that the latent heat evolution can be accounted for in the governing energy equation by the definition of an apparent heat capacity (Comini et al., 1974), a source term (Voller and Prakash, 1987), or a total enthalpy (White, 1986; Shamsundar and Sparrow, 1975). Consequently, the numerical solution can be carried out on a space grid that remains fixed throughout the calculation. Nonlinearities associated with the phase change, however, remain in the resulting discrete equations.

In the current work, we shall focus on two possible fixed-grid formulations derived from the basic enthalpy equation governing conduction-controlled phase change systems (Crank, 1984). One of the methods is based on the enthalpy equation itself (White, 1986) while the other is based on the recently proposed source-based method (Voller, 1990; Voller and Swaminathan, 1991). By employing the finite element discretization (Zienkiewicz and Taylor, 1989) in space and the fully implicit backward Euler time integration scheme, a set of nonlinear equations is derived for each of the formulations. These equations are linearized using the Newton-Raphson scheme to obtain a system of linear algebraic equations. The essential difference between the two formulations is that the former (White, 1986) gives rise to a nonsymmetric real system while the latter (Voller and Swaminathan, 1991) gives rise to a symmetric positive definite system of linear equations.

These equations are first scaled using symmetric scaling (Jennings and Malik, 1978) so that the resulting coefficient matrix has unit diagonals. Different iterative

methods of the conjugate gradient type that can be used to solve these linear equations are outlined: in particular, the conjugate gradient method (Hestenes and Steifel, 1952) for symmetric positive definite systems and the orthomin(k) method (Vinsome, 1976) and the conjugate gradient squared method (Sonneveld, 1989) for nonsymmetric systems. The convergence properties of these methods and the incomplete factorization preconditioning (Meijerink and Van der Vorst, 1977; Chronopoulos and Ma, 1989a) to improve the convergence are described. Vectorization (Van der Vorst, 1982) and parallelization (Chronopoulos and Ma, 1989a) are employed to make use of the CRAY-2 supercomputing facility.

A distinctive feature of the current work is that the application of the solvers needs to be carried out in a two-step fashion consisting of an outer iteration (Newton linearization) and inner iterations (linear equation solver) within the outer iteration. Most of the previous workers (Shamsundar and Sparrow, 1975; White, 1986; Williams, 1987) have concentrated on modifying iterative methods like the Jacobi, Gauss-Seidel, selective overrelaxation and preconditioned conjugate gradient for solving the nonlinear system.

An isothermal test problem with varying Stefan numbers is solved to illustrate the validity and efficiency of the iterative solvers and compare and contrast the two different formulations.

"A distinctive feature of the current work is that the application of the solvers needs to be carried out in a two-step fashion consisting of an outer iteration and inner iterations. Most of the previous workers have concentrated on modifying iterative methods like the Jacobi, Gauss-Seidel, selective overrelaxation and preconditioned conjugate gradient for solving the nonlinear system."

Results and Discussion

1. THE GOVERNING EQUATION

The basic governing equation for conduction-driven solidification systems can be written as (Crank, 1984)

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

where k is the thermal conductivity, and H is the enthalpy defined as

$$H = \rho c(T - T_{ref}) + g\rho L, \tag{2}$$

where ρ is the density, c is the specific heat capacity, T is the temperature, T_{ref} is an arbitrary reference temperature, g is the volume fraction of the liquid, and L is the

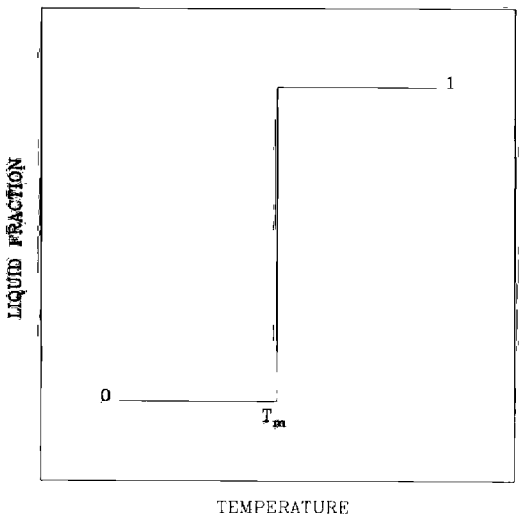

Fig. 1 Liquid fraction temperature relationship for the Stefan problem

latent heat. Note that for the sake of simplicity, we have assumed the thermophysical properties to be a constant. For isothermal solidification problems, the liquid fraction g is related to the temperature T through the step function ($g = F(T)$) as shown in Figure 1. The general governing Eq. (1) is nonlinear in that it is written in terms of two related variables, namely, enthalpy H and temperature T. In a numerical solution, this problem is overcome on appropriate iteration.

One of the formulations to solve the problem is based directly on Eq. (1) (White, 1986). The second formulation involves lumping the latent heat nonlinearity into a source term (Voller and Swaminathan, 1991). Essentially, on using Eq. (2) we can write

$$\frac{\partial H}{\partial t} = \rho c \frac{\partial T}{\partial t} + \rho L \frac{\partial g}{\partial t}. \tag{3}$$

Substituting Eq. (3) in the basic Eq. (1), we have

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + S, \tag{4}$$

where S is given by

$$S = -\rho L \frac{\partial g}{\partial t}. \tag{5}$$

Equation (4) is commonly referred to as the source-based formulation.

2. DISCRETIZATION

Space discretization of the governing equation subject to appropriate boundary conditions can be achieved by Galerkin's finite element method. The time discretization is achieved using the fully implicit backward Euler time integration scheme. This yields a nonlinear system of equations that must be solved at each time step. A further refinement that is often used in conjunction with the above time integration scheme for solving solidification problems is the so-called lumped capacity formulation (Dalhuijsen and Segal, 1986). The effect of this is to associate with each node a control volume over which the nodal values are representative.

We shall next list the final form of the discretized equations for the two formulations. Note that for the

sake of simplicity, we have neglected contributions from the boundary conditions to the equations.

The total enthalpy formulation is

$$\mathbf{CH} + \Delta t\mathbf{KT} = \mathbf{CH}^{old}, \tag{6}$$

and the source-based formulation is

$$a_p T_p = \sum_{nb} a_{nb} T_{nb} + a_p^o T_p^{old} + \rho L V_p [g_p^{old} - g_p], \tag{7}$$

where the a's and $\mathbf{C}$ and $\mathbf{K}$ are the coefficients of the discretization equation that depend on the thermophysical properties and are evaluated at the current time step and V is the discrete volume associated with a node. The superscript $[]^{old}$ refers to the old time value and the subscripts $[]_p$ and $[]_{nb}$ refer to the node point p and the neighboring nodes, respectively. Note that the equation for the total enthalpy method has been presented in the matrix form while that for the source-based method has been presented in the point form. The primary reason for this choice is to make further discussions on the solution of the nonlinear equations relatively easy to follow. Note that, if required, it is a relatively easy task to write the point form in the matrix form and vice-versa (Voller, Swaminathan, and Thomas, 1990).

"Space discretization of the governing equation can be achieved using the fully implicit backward Euler time integration scheme. This yields a nonlinear system of equations that must be solved at each time step."

3. SOLUTION STRATEGY

3.1 TOTAL ENTHALPY METHOD

The key feature in the enthalpy formulation is to make a substitution of the temperature $\mathbf{T}$ in terms of the total enthalpy $\mathbf{H}$ in the discretized Eq. (6). To this end, it is noted that for isothermal phase change problems the temperature at any node can be written as

$$\mathbf{T} = \beta \mathbf{H} + \gamma, \tag{8}$$

where

$$\beta = \begin{cases} \dfrac{1}{\rho c} & \text{if } \mathbf{T} < T_m \\ 0 & \text{if } \mathbf{T} = T_m \\ \dfrac{1}{\rho c} & \text{if } \mathbf{T} > T_m \end{cases} \tag{9}$$

"The principle behind the source-based method is to recognize that phase change is characterized by the local liquid fraction field. This nodal liquid fraction field keeps track of the phase change as it traverses the domain of interest."

and

$$\gamma = \begin{cases} T_m & \text{if } \mathbf{T} < T_m \\ T_m & \text{if } \mathbf{T} = T_m \\ T_m - \dfrac{L}{C} & \text{if } \mathbf{T} > T_m \end{cases} \tag{10}$$

Note that the reference temperature has been chosen as the melting point T_m. Substitution of this expression for **T** in Eq. (6) will result in a nonlinear equation for the total enthalpy **H.** This nonlinear equation can be linearized using the Newton-Raphson scheme. The final linearized equation has the form

$$\mathbf{J}^m \Delta \mathbf{H}^{m+1} = -\mathbf{R}^m, \tag{11}$$

where the residual **R** is written as

$$\mathbf{R}^m = [\mathbf{C}^m + \Delta t \mathbf{A}^m \mathbf{K}^m]\mathbf{H}^m - \mathbf{C}^m \mathbf{H}^{old} + \Delta t \gamma^m \mathbf{K}^m \tag{12}$$

and the Jacobian **J** is written as

$$\mathbf{J}^m = \frac{d\mathbf{R}^m}{d\mathbf{H}^m} = \mathbf{C}^m + \Delta t \mathbf{A}^m \mathbf{K}^m, \tag{13}$$

where **A** is a diagonal matrix with β as its elements. The superscript $[]^m$ indicates the previous iteration level. The new value of enthalpy is evaluated from

$$\mathbf{H}^{m+1} = \mathbf{H}^m + \Delta \mathbf{H}^{m+1}. \tag{14}$$

Note the Jacobian resulting from the linearization is nonsymmetric and real.

3.2 SOURCE-BASED METHOD

The principle behind the source-based method is to recognize that phase change is characterized by the local liquid fraction field *g*. This nodal liquid fraction field keeps track of the phase change as it traverses the domain of interest.

Following Voller and Swaminathan (1991) an iterative scheme based on Eq. (7) can be written as

$$a_p T_p^{m+1} = \sum_{nb} a_{nb} T_{nb}^{m+1} + a_p^o T_p^{old} + \rho L V_p [g_p^{old} - g_p^{m+1}], \tag{15}$$

where $[]^{m+1}$ indicates the current iteration level. We note that the discretized source term given as

$$S = \rho L V_p [g_p^{old} - g_p^{m+1}] \tag{16}$$

presents, in the general case, a nonlinearity in Eq. (15). On using a truncated Taylor series expansion for g_p^{m+1}, we have

$$g_p^{m+1} = g_p^m + \frac{dF}{dT}[T_p^{m+1} - F^{-1}(g_p^m)], \tag{17}$$

where the slope dF/dT is evaluated at g_p^m. For isothermal problems, dF/dT is set to an arbitrarily large value (e.g., 10^8) and $F^{-1}(g_p^m) = T_m$. Using this linearization, the source term is conveniently expressed as

$$S = S_c + S_p T_p^{m+1}, \tag{18}$$

where

$$S_c = \rho L V_p [g_p^{old} - g_p^m] - S_p F^{-1}(g_p^m) \tag{19}$$

and

$$S_p = -\rho L V_p \frac{dF}{dT}. \tag{20}$$

The iterative scheme now reads

$$[a_p - S_p] T_p^{m+1} = \sum_{nb} a_{nb} T_{nb} + a_p^o T_p^{old} + S_c. \tag{21}$$

Note that the coefficient matrix resulting from this linearization will be symmetric and positive definite. Once Eq. (21) is solved to obtain the temperature field, the liquid fraction is obtained directly from Eq. (17). This iterative process is continued to convergence. For further details on the implementation of the method see Voller (1990) and Voller and Swaminathan (1991).

4. SOLUTION OF LINEAR ALGEBRAIC EQUATIONS

The discretization and linearization processes described so far produced a set of linear algebraic equations, Eq. (11) and Eq. (21), of the form

$$\bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{b}}. \tag{22}$$

Recall that (1) the enthalpy-based formulation yields a nonsymmetric real system of equations, and (2) the source-based formulation yields a symmetric positive definite system of equations. In each case, however, the form of the resulting coefficient matrix (assuming natural ordering of the grid points in two dimensions) is

$$\bar{\mathbf{A}} = [\mathbf{C}\ \mathbf{D}\ \mathbf{E}], \tag{23}$$

where **C, D,** and **E** are block tridiagonal matrices of order N and have the form

$$\begin{aligned} \mathbf{C} &= [\mathbf{C}_{w_i}\ \mathbf{C}_{p_i}\ \mathbf{C}_{e_i}] \quad 1 \le i \le N \\ \mathbf{D} &= [\mathbf{D}_{w_i}\ \mathbf{D}_{p_i}\ \mathbf{D}_{e_i}] \quad 1 \le i \le N \\ \mathbf{E} &= [\mathbf{E}_{w_i}\ \mathbf{E}_{p_i}\ \mathbf{E}_{e_i}] \quad 1 \le i \le N \end{aligned} \tag{24}$$

with $\mathbf{D}_{p_i} > 0$ and $\mathbf{C}_{w_i}$, $\mathbf{C}_{p_i}$, $\mathbf{C}_{e_i}$, $\mathbf{D}_{w_i}$, $\mathbf{D}_{e_i}$, $\mathbf{E}_{w_i}$, $\mathbf{E}_{p_i}$, and $\mathbf{E}_{e_i} \le 0$.

Direct methods (based on Gaussian elimination) could be used to solve this linear system. However, due to nonzero fill-ins, they may become prohibitively expensive for very large and sparse systems. Iterative methods, on the other hand, are particularly suited for such systems. They essentially begin with an initial guess and generate a sequence of approximate solutions that converge to the exact solution. Iterative methods usually involve the coefficient matrix in matrix-vector multiplications, which allows great flexibility in the storage of data.

4.1 GRADIENT ITERATIVE METHODS FOR SYMMETRIC COEFFICIENT MATRICES

The conjugate gradient method (Hestenes and Steifel, 1952) is commonly used for solving a system of equations with a symmetric positive definite coefficient matrix.

Algorithm 1: The Conjugate Gradient Method

Choose $\mathbf{x}_0$
$\mathbf{p}_0 = \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
For $i = 0$ *until convergence* **Do**
$\mathbf{z}_i = \mathbf{M}^{-1}\mathbf{r}^i$
$\alpha_i = \dfrac{(\mathbf{z}_i, \mathbf{r}_i)}{(\mathbf{p}_i, \mathbf{A}\mathbf{p}_i)}$
$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$
$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{p}_i$
$\beta_i = -\dfrac{(\mathbf{z}_{i+1}, \mathbf{r}_{i+1})}{(\mathbf{z}_i, \mathbf{r}_i)}$
$\mathbf{p}_{i+1} = \mathbf{z}_{i+1} + \beta_i \mathbf{p}_i$
Endfor.

The symmetric positive definite matrix **M** is the incomplete factorization preconditioner. We define as vector operations the floating-point operations on vectors of dimension N, ignoring any isolated scaler operations. Vector operations per iteration are $10\,N + 1\,Mv + 1\,W_{prec}$, where Mv is the work for matrix-vector multiplication and W_{prec} is the work for preconditioning. Storage is required for the entire vectors **r, p, x, Ap,** and the matrix **A.** The initial trial vector $\mathbf{x}_0$ is usually taken to be the null vector unless some approximation to the solution is known from the previous time step or iteration.

4.2 GRADIENT ITERATIVE METHODS FOR NONSYMMETRIC COEFFICIENT MATRICES

For nonsymmetric coefficient matrices, the conjugate gradient method breaks down. Extensions of the conjugate gradient method, however, are available for nonsymmetric linear systems (Concus, Golub, and O'Leary, 1976; Eisenstat, Elman, and Schultz, 1983; Young and Lea, 1980; Saad and Schultz, 1986). Two outstanding examples are the orthomin(k) (Vinsome, 1976) and conjugate gradient squared (Sonneveld, 1989). Orthomin(k) applies when the symmetric part of the coefficient matrix is positive definite. However, it requires the storage of an a priori unspecified number of direction vectors that depends on the nonsymmetric extent and the condition number of the matrix. The conjugate gradient squared method is a conjugate gradient–like method that was derived from the biconjugate gradient method (Fletcher, 1976). The method converges faster than the biconjugate gradient method, and also, it does not need the storage of an a priori unspecified number of direction vectors like orthomin(k).

Algorithm 2: The Orthomin(k) Method

Choose x_0
$p_0 = r_0 = b - Ax_0$
For $i = 0$ *until convergence* **Do**

$$\alpha_i = \frac{(r_i, Ap_i)}{(Ap_i, Ap_i)}$$

$$x_{i+1} = x_i + \alpha_i p_i$$
$$r_{i+1} = r_i - \alpha_i Ap_i$$

$$\beta_j^i = -\frac{(AP_r r_{i+1}, Ap_j)}{(Ap_j, Ap_j)} \qquad j \le i$$

$$p_{i+1} = P_r r_{i+1} + \sum_{j=j_i}^{i} \beta_j^i p_j \qquad j_i = \min(0, i - k + 1)$$

$$Ap_{i+1} = AP_r r_{i-1} + \sum_{j=j_i}^{i} \beta_j^i Ap_j \quad j_i = \min(0, i - k + 1)$$

Endfor.

The matrix $\mathbf{P}_r$ is the right preconditioner. Vector operations per iteration are $(6k + 8)\,N + 1\,Mv + 1\,W_{prec}$. Storage is required for the entire vectors **p, r, x, Ap,** the matrix **A,** and k of the previous search directions **p.** The initial trial vector $\mathbf{x}_0$ is usually taken to be the null vector unless some approximation to the solution is known from the previous time step or previous iteration.

Algorithm 3: Conjugate Gradient Squared Method

Choose x_0
$r_0 = b - Ax_0$
$q_0 = p_{-1} = 0$
$\rho_{-1} = 1$
For $i = 0$ *until convergence* **Do**

$$\rho_i = \mathbf{r}_0^T \mathbf{r}_i$$

$$\beta_i = \frac{\rho_i}{\rho_{i-1}}$$

$$\mathbf{u}_i = \mathbf{r}_i + \beta_i \mathbf{q}_i$$
$$\mathbf{p}_i = \mathbf{u}_i + \beta_i(\mathbf{q}_i + \beta_i \mathbf{p}_{i-1})$$
$$\mathbf{v}_i = \mathbf{AP}_r \mathbf{p}_i$$
$$\sigma_i = \mathbf{r}_0^T \mathbf{v}_i$$

$$\alpha_i = \frac{\rho_i}{\sigma_i}$$

$$\mathbf{q}_{i+1} = \mathbf{u}_i - \alpha_i \mathbf{v}_i$$
$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{P}_r(\mathbf{u}_i + \mathbf{q}_{i+1})$$
$$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{AP}_r(\mathbf{u}_i + \mathbf{q}_{i+1})$$

Endfor.

The matrix $\mathbf{P}_r$ is the right preconditioner. Vector operations per iteration for the method are $19\,N + 2\,Mv + 2\,W_{prec}$. Storage is required for the entire vectors **p, r, x, q, u, v, s, Ap,** and the matrix **A.** The initial trial vector $\mathbf{x}_0$ is usually taken to be the null vector unless some approximation to the solution is known from the previous time step or the previous iteration.

4.3 CONVERGENCE AND PRECONDITIONING

A major drawback of the conjugate gradient–like methods is that the convergence rate is closely related to the condition number of the matrix. Ill-conditioned problems, having a large condition number, tend to give extremely poor convergence rates, and because of this, in the past elimination methods have been preferred for many problems despite the extra storage requirements caused by fill-in. Mainly in recent years, various methods of preconditioning the equations have been developed, giving markedly better convergence rates.

"Ill-conditioned problems, having a large condition number, tend to give extremely poor convergence rates, and because of this, in the past elimination methods have been preferred for many problems despite the extra storage requirements caused by fill-in."

Preconditioning refers to finding a nonsingular matrix **K,** which can be factored into the product

$$\mathbf{K} = \mathbf{K}_R\mathbf{K}_L, \tag{25}$$

so that

$$\mathbf{K}_L^{-1}\bar{\mathbf{A}}\mathbf{K}_R^{-1} \tag{26}$$

is close to the identity matrix. We suppose that **K** is easily invertible. The system in Eq. (22) is transformed to

$$[\mathbf{K}_L^{-1}\bar{\mathbf{A}}\mathbf{K}_R^{-1}][\mathbf{K}_R\bar{\mathbf{x}}] = \mathbf{K}_L^{-1}\bar{\mathbf{b}}. \tag{27}$$

It is possible to distinguish three different ways of applying the preconditioner (Brussino and Sonnad, 1989); the relation given by Eq. (25) defines the split preconditioning, while the other two ways can be considered as special cases with

$$\begin{aligned} &\textit{Left Preconditioning}: && \mathbf{K}_R = \mathbf{I} \\ &\textit{Right Preconditioning}: && \mathbf{K}_L = \mathbf{I} \end{aligned} \tag{28}$$

4.3.1 SYMMETRIC SCALING PRECONDITIONING

The system of equations (22) is first scaled such that the resulting coefficient matrix has unit diagonal elements. This can be achieved by scaling the rows and columns of the original coefficient matrix individually (Jennings and Malik, 1978). The transformed system can be written as

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{29}$$

where

$$\mathbf{A} = \mathbf{D}\bar{\mathbf{A}}\mathbf{D} \;:\; \mathbf{x} = \mathbf{D}^{-1}\bar{\mathbf{x}} \;:\; \mathbf{b} = \mathbf{D}\bar{\mathbf{b}} \tag{30}$$

and **D** is a diagonal matrix of scaling factors such that the ith element of **D** is

$$d_i = \frac{1}{\sqrt{\bar{a}_{ii}}}, \tag{31}$$

where $\bar{a}_{ii}$ is the ith diagonal element of $\bar{\mathbf{A}}$. It must be noted that the eigenvalues of **A** are grouped more closely together than the eigenvalues of $\bar{\mathbf{A}}$.

4.3.2 INCOMPLETE FACTORIZATION PRECONDITIONING

In incomplete factorization, a lower triangular matrix **L** and an upper triangular matrix **U** are constructed such that they are good approximations of the **L** and **U** factors of **A** and are also sparse. Consider only those locations where **A** has nonzero elements; at these locations, perform operations corresponding to an **LU** factorization using elements only from those locations where **A** has nonzero elements. The sparsity structure of **A** is completely preserved in the resulting factors.

In the present work the incomplete $\mathbf{LDL}^T$ factorization with the two-sided preconditioning is used when the coefficient matrix is symmetric. The incomplete $\mathbf{LDL}^T$ factorization is preferred over the incomplete Cholesky factorization as it avoids computation of any square roots. For the nonsymmetric coefficient matrices, the incomplete **LU** factorization with the right preconditioning is used. The advantage of using the right preconditioning is that it is easy to obtain a residual based on the original system of equations rather than on the preconditioned system, thus making it easier to monitor the convergence.

Let *NZ* **(A)** denote the set of pairs $[i,j]$ for which the entries a_{ij} of the matrix **A** are nonzero, the nonzero pattern of **A.** Then the incomplete factorization algorithm can be written as follows.

Algorithm 4: The $\mathbf{LDL}^T$ *Preconditioning*

For $j = 1$ *to* N **Do**
  For $i = 1$ *to* $j - 1$ **Do**
    $R_i = D_i A_{ij}$
  Endfor

$$D_j = A_{ij} - \sum_{i=1}^{j=1} A_{ij} R_i$$

  For $i = j + 1$ *to* N **Do**

$$\text{if } [i,j] \text{ belongs to } NZ\ (A) \text{ then } L_{ij} = \frac{A_{ij} - \sum_{k=1}^{j-1} L_{ik} R_k}{D_j}$$

  Endfor
Endfor.

Algorithm 5: Incomplete **LU** *Factorization*

For $i = 1$ *to* N **Do**
  For $j = i + 1$ *to* N **Do**
    if $[i,j]$ *belongs to* $NZ\ (A)$ *then*

$$S_{ij} = A_{ij} - \sum_{k=1}^{\min(i,j)-1} L_{ij} U_{ij}$$

$$\text{if } (i > j) \text{ then } L_{ij} = S_{ij}$$

$$\text{if } (i < j) \text{ then } U_{ij} = \frac{S_{ij}}{L_{ij}}$$

    endif
  Endfor
Endfor.

5. IMPLEMENTATION ON CRAY-2: CODE VECTORIZATION AND PARALLELIZATION

The CRAY-2 supercomputer in the Minnesota Supercomputer Institute has four processors, each of which can execute independent tasks concurrently. All processors have equal access to a very large central memory of 512 megawords. Each processor has eight vector registers (each 64 words long) and has data access through a single path between its vector registers and main memory. Each processor has 16 kilowords of local memory with no direct path to central memory but with a separate data path between local memory and vector registers and six parallel pipelines: namely, common memory to vector registers, vector registers to local memory, floating addition and subtraction, floating multiplication and division, integer addition and subtraction, and logical pipelines. Consequently, the implementation of the solution algorithm on CRAY-2 requires proper vectorization and parallelization to take advantage of the machine's full capacity.

Parallelization is automatically achieved by using the autotasking library. Autotasking is most often applied to parallel work found in the independent iteration of DO-loops. If the loop has N iterations, we map the N iterations into P processors so that each processor has the same amount of work to do. To achieve load

balancing, we must consider static and dynamic partitioning. We use static partitioning when the times for each of the loop iterations are equal. An example of contiguous static partitioning with $P = 4$ is as follows (Chronopoulos and Ma, 1989a):

Processor	Assigned Iteration
P1	$I = 1, \ldots, N/4$
P2	$I = (N/4) + 1, \ldots, N/2$
P3	$I = (N/2) + 1, \ldots, 3 * N/4$
P4	$I = (3 * N/4), \ldots, N$

In our case, since the matrix is stored in a vector form as diagonals, the matrix-vector multiplication is vectorized. Also, it can be performed concurrently by distributing the rows of the coefficient matrix and the corresponding elements of the vector among the processors. The inner products and linear combinations are also vector operations. The only area needing further attention is in the preconditioning step (Van der Vorst, 1982). The factorization step by itself is not easily vectorizable or parallelizable in the original form because it is normally done recurrently (row by row). However, it is required only once, before applying the iterative method, and hence the computational cost is usually a small fraction of the total time of the solution. The forward and backward solution steps,

$$\mathbf{Ly} = \mathbf{r} \tag{32}$$

and

$$\mathbf{Uz} = \mathbf{y}, \tag{33}$$

are also recurrent in nature, and hence are not very amenable to vector and parallel processing. Furthermore, these have to be done either once within each iteration with the conjugate gradient and orthomin(k) methods, or twice with the conjugate gradient squared method. These steps can be vectorized using the Neumann series expansion as proposed by Van der Vorst (1982). If we denote

$\mathbf{E}$ = matrix consisting of only the superdiagonal of $\mathbf{L}^T$
$\mathbf{F}$ = matrix consisting of the super-superdiagonals of $\mathbf{L}^T$

and normalize the diagonal entries to 1, then we have $\mathbf{L}^T = \mathbf{I} + \mathbf{E} + \mathbf{F}$. Solving

$$(\mathbf{I} + \mathbf{E} + \mathbf{F})\mathbf{y}_i = \mathbf{r}_i \tag{34}$$

"In our case, since the matrix is stored in a vector form as diagonals, the matrix-vector multiplication is vectorized. Also, it can be performed concurrently by distributing the rows of the coefficient matrix and the corresponding elements of the vector among the processors."

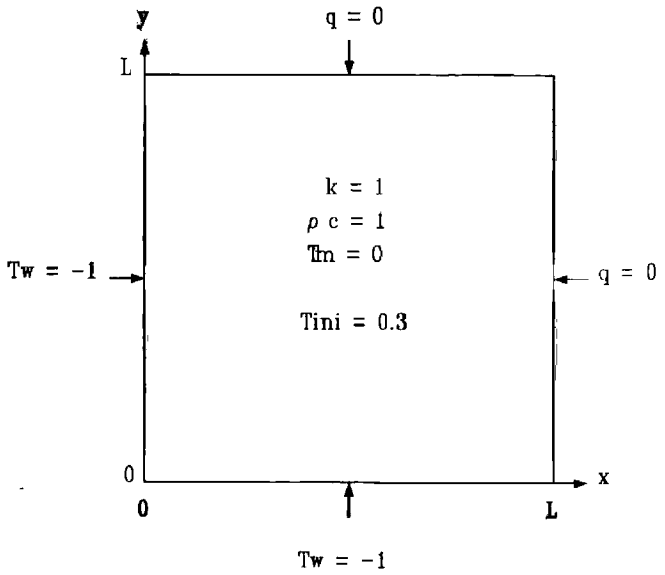

Fig. 2 Solidification in a corner region: problem specification

amounts to

$$(\mathbf{I} + \mathbf{E})\mathbf{y}_i = \mathbf{r}_i - \mathbf{F}\mathbf{y}_{i-1}. \tag{35}$$

Assuming **E** to be small relative to **I** in norm (diagonal dominance), we proceed such that

$$\begin{aligned}\mathbf{y}_i &= (\mathbf{I} + \mathbf{E})^{-1}(\mathbf{r}_i - \mathbf{F}\mathbf{y}_{i-1}) \\ &= (\mathbf{I} - \mathbf{E} + \mathbf{E}^2 - \mathbf{E}^3 + \ldots)(\mathbf{r}_i - \mathbf{F}\mathbf{y}_{i-1})\end{aligned} \tag{36}$$

and truncate the power series at the mth term. We note that the forward solution step has now been written in a fully vectorizable form. We would like to truncate the power series in such a way that the error due to truncation is small compared to the approximation error of **K** with respect to **A.** In the current work truncation after two terms has been used. A similar procedure is used for the back substitution step; see Eq. (33).

6. TEST PROBLEMS

The isothermal solidification of a corner of a uniformly infinite container carrying a liquid initially at a temperature greater than the melting temperature is studied. The conductivity and specific heat are assumed to be constants. The problem is nondimensionalized, as shown in Appendix A. The liquid is initially at a uniform temperature of $\theta = 0.3$. At time $\tau > 0$, the temperature of the surface of the liquid is reduced to -1 and maintained constant. The analysis is done with square elements spaced every 0.0125 (81×81 grid). The model problem with all the properties is shown in Figure 2.

Five problems were designed with the Stefan number, defined by Eq. (A.7) in Appendix A, varying as 0.25, 0.5, 1, 2, and 4, representing varying degrees of nonlinearity. The simulation was carried on till the temperature at the center of the calculation domain reached a value of -0.5. As the Stefan number decreases, the latent heat term prevails over the specific heat term, which results in a slow movement of the phase front and a longer time for the temperature at the center of the domain to reach a value of -0.5. It was decided to perform a minimum of 50 time steps of simulation.

6.1 CHOICE OF TIME-STEP SIZE AND CONVERGENCE CRITERIA

After extensive numerical experimentation, we arrived at the following guidelines for the choice of the time-step size and the convergence criteria:

1. Optimal choices of the size of the time step and the inner (linear equation solver) and outer (Newton step) convergence criteria control the accuracy and efficiency of the numerical solution.
2. Most of the CPU time is concentrated in the linear equation solvers since the coefficient matrix and the right-hand vector can be evaluated at every iteration of every time step with a relatively small computational overload. Thus, the choice of the parameters outlined above should be such that the number of sweeps through the inner solver is minimized for a given level of accuracy in the solution.
3. The fraction of liquid solidified at any particular instant of time during the solidification process is a good measure of accuracy. A value within 0.05% of the explicit solution was deemed accurate.
4. The relative norm of the residual is a good measure of the convergence for both the inner and outer solvers. It was decided to restrict the number of outer and inner iterations to 200 and 100, respectively, mainly to prevent the iterative schemes from using an excessive number of iterations on some pathologic problems.
5. The efficiency of the source-based method improves without much loss of accuracy as the number of time steps decreases. However, with the enthalpy formulation, at larger time steps, either the solution is not accurate or the number of outer iterations required for convergence is excessive. Consequently, the choice of the time step had to be optimized.
6. In conventional application of the Newton-Raphson method coupled with linear system solvers, the linear system in the inner loop is solved to the accuracy of the outer iteration in order to preserve efficiency. However, in the source-based formulation, the linear system needs to be solved almost exactly to maintain accuracy in the outer iteration, because of the way in which the liquid fraction field is determined from the temperature field.

These guidelines lead to the two problem formulations outlined in Table 1. The optimal number of previous search directions needed for orthomin(k) is 3 for all the problems. A vector of zeros is a good initial guess for the orthomin(3) and the conjugate gradient squared solvers, while the previous best solution is a good initial guess for the conjugate gradient solver.

Tables 2 through 6 present the results in terms of the size of the time step chosen, given in terms of the R value defined in Eq. (A.10) in Appendix A, level of accuracy, number of inner and outer iterations, and the CPU requirements on a CRAY-2 supercomputer with and without parallelization for each of the iterative solvers as applied to the respective formulations. The results indicate that

1. The source-based method with the conjugate gradient solver is by far the better of the two formulations discussed here, in terms of both CPU and storage requirements. The method is particularly attractive because it permits the use of large time steps to solve the problem without much loss of accuracy.
2. Incomplete factorization preconditioning does not seem to improve the performance of the iterative solvers. Although the number of inner iterations decreased, the CPU time requirement did not decrease. This is because the construction of the incomplete factorization preconditioner is not vectorizable.
3. Parallelization results in a speedup by a factor of 1.5 to 2. It must be mentioned that the machine was not run in a fully dedicated mode because of the load from other users. In a fully dedicated mode, we expect a speedup closer to 4. The speedups on parallelizing the preconditioned versions seem to be lower compared to the direct counterparts, since the incomplete factorization step does not parallelize.

Table 1

Convergence Criteria Used with the Enthalpy-Based and the Source-Based Formulations

Formulation	Solver	Tolerance for Outer Conv.[a]	Tolerance for Inner Conv.[b]
Enthalpy	O P-O CGS P-CGS	0.1	0.01
Source	CG P-CG	0.001	0.000001

[a] Relative norm of the residual less than a tolerance, i.e.,

$$\frac{\|r^m\|}{\|b^0\|} \leq \textit{tolerance},$$

where r is the residue of the nonlinear system of equations and b^0 is the right-hand vector at the beginning of a time step.

[b] Relative norm of the residual less than a tolerance, i.e.,

$$\frac{\|r_k\|}{\|b\|} \leq \textit{tolerance},$$

where r is the residue and b is the right-hand vector at the start of the inner iteration of the scaled linear equations.

Figure 3a shows the progress of the solidification with time as a function of the Stefan number. Figures 3b through 3f depict the temperature history of the center region of the domain for all five problems. The source-based and the enthalpy-based formulations give essentially the same solutions, which compare favorably with the explicit solution.

7. CONCLUSION

Three conjugate gradient–like iterative methods have been tested for the solution of the Stefan problem on a parallel vector machine. Two different formulations of the Stefan problem, namely the enthalpy-based formulation and the source-based formulation, were considered. The results indicate that the combination of some current phase change models and iterative solvers results in an efficient solution methodology for solidification problems.

Further numerical experimentation is necessary to investigate the inefficiency of the preconditioners and to extend these solvers to more difficult problems, for example, those involving a nonlinear relationship between temperature and liquid fraction. This is being done and the results will be reported at a later date. Further work needs to be done in providing definite theories regarding the accuracy and convergence of the iterative solvers, especially for the nonsymmetric systems. Another interesting area for future work is par-

Table 2

Comparison of the Source-Based and the Enthalpy-Based Methods on the Basis of CPU Requirements for Stefan Number 0.25 (% solidified with explicit scheme = 87.55)

Solver	R	No. of Time Steps	% Solid.	No. of Outer Iter.	No. of Inner Iter.	CPU sec 1 Proc.	CPU sec 4 Proc.
Enthalpy-Based Formulation							
O	25.92	200	87.53	1639	19270	54.07	26.13
P-O	25.92	200	87.53	1505	5810	58.21	37.54
CGS	5.76	900	87.54	3906	19559	78.01	37.05
P-CGS	5.76	900	87.54	3796	7169	132.53	80.36
Source-Based Formulation							
CG	103.68	50	87.54	261	14998	23.24	12.31
P-CG	103.68	50	87.52	263	11529	42.01	26.21

Table 3

Comparison of the Source-Based and the Enthalpy-Based Methods on the Basis of CPU Requirements for Stefan Number 0.50 (% solidified with explicit scheme = 88.68)

Solver	R	No. of Time Steps	% Solid.	No. of Outer Iter.	No. of Inner Iter.	CPU sec 1 Proc.	CPU sec 4 Proc.
Enthalpy-Based-Formulation							
O	24.58	125	88.66	1135	13412	38.88	18.45
P-O	24.58	125	88.66	1099	4137	42.11	25.21
CGS	3.20	960	88.68	3864	16993	72.82	35.91
P-CGS	3.20	960	88.68	3853	5826	128.72	82.81
Source-Based-Formulation							
CG	61.44	50	88.66	289	15144	25.61	12.92
P-CG	61.44	50	88.65	288	11592	44.16	29.22

Table 4

Comparison of the Source-Based and the Enthalpy-Based Methods on the Basis of CPU Requirements for Stefan Number 1.00 (% solidified with explicit scheme = 91.61)

Solver	R	No. of Time Steps	% Solid.	No. of Outer Iter.	No. of Inner Iter.	CPU sec 1 Proc.	CPU sec 4 Proc.
Enthalpy-Based-Formulation							
O	21.12	100	91.60	900	10118	28.61	13.93
P-O	21.12	100	91.61	870	3241	32.98	20.62
CGS	2.82	750	91.61	3037	13289	58.65	27.34
P-CGS	2.82	750	91.61	3030	4803	102.81	65.32
Source-Based Formulation							
CG	42.24	50	91.63	305	13513	22.64	11.52
P-CG	42.24	50	91.63	305	10374	39.11	24.21

Table 5

Comparison of the Source-Based and the Enthalpy-Based Methods on the Basis of CPU Requirements for Stefan Number 2.00 (% solidified with explicit scheme = 93.26)

Solver	R	No. of Time Steps	% Solid.	No. of Outer Iter.	No. of Inner Iter.	CPU sec 1 Proc.	CPU sec 4 Proc.
Enthalpy-Based Formulation							
O	20.48	75	93.28	802	9072	25.21	12.53
P-O	20.48	75	93.29	853	2926	31.92	20.61
CGS	6.40	240	93.27	1287	7506	29.95	15.46
P-CGS	6.40	240	93.27	1176	2607	45.04	27.92
Source-Based Formulation							
CG	30.72	50	93.31	273	9932	16.94	8.67
P-CG	30.72	50	93.31	273	7609	29.03	19.69

Table 6

Comparison of the Source-Based and the Enthalpy-Based Methods on the Basis of CPU Requirements for Stefan Number 4.00 (% solidified with explicit scheme = 97.16)

Solver	R	No. of Time Steps	% Solid.	No. of Outer Iter.	No. of Inner Iter.	CPU sec 1 Proc.	CPU sec 4 Proc.
Enthalpy-Based Formulation							
O	17.92	75	97.19	777	8669	25.05	12.67
P-O	17.92	75	97.19	848	2963	32.01	19.23
CGS	8.96	150	97.20	860	5814	22.53	11.39
P-CGS	8.96	150	97.20	833	2099	34.03	21.02
Source-Based Formulation							
CG	26.88	50	97.19	259	9037	15.06	7.94
P-CG	26.88	50	97.19	259	6939	27.23	17.20

allel preconditioning methods, such as domain decomposition methods, for application on massively parallel machines.

Appendix A: Derivation of Nondimensional Equations

The basic enthalpy equation governing the solidification of a corner region depicted in Figure 2 is

$$\frac{\partial H}{\partial t} = \nabla \cdot (k \nabla T) \tag{A.1}$$

with the initial condition

$$T = T_{ini} \qquad \text{at } t = 0 \tag{A.2}$$

and the boundary condition

$$T = T_w \quad \text{at } x = 0 \quad \text{and} \quad y = 0 \text{ surfaces} \tag{A.3}$$

We define a set of nondimensional variables

$$\begin{aligned} \theta &= \frac{(T - T_m)}{(T_m - T_w)} \\ \zeta &= \frac{H}{\rho c (T_m - T_p)} \\ X &= \frac{x}{L} \; ; \; Y = \frac{y}{L} \\ \tau &= \frac{kt}{\rho c L^2} \; , \end{aligned} \tag{A.4}$$

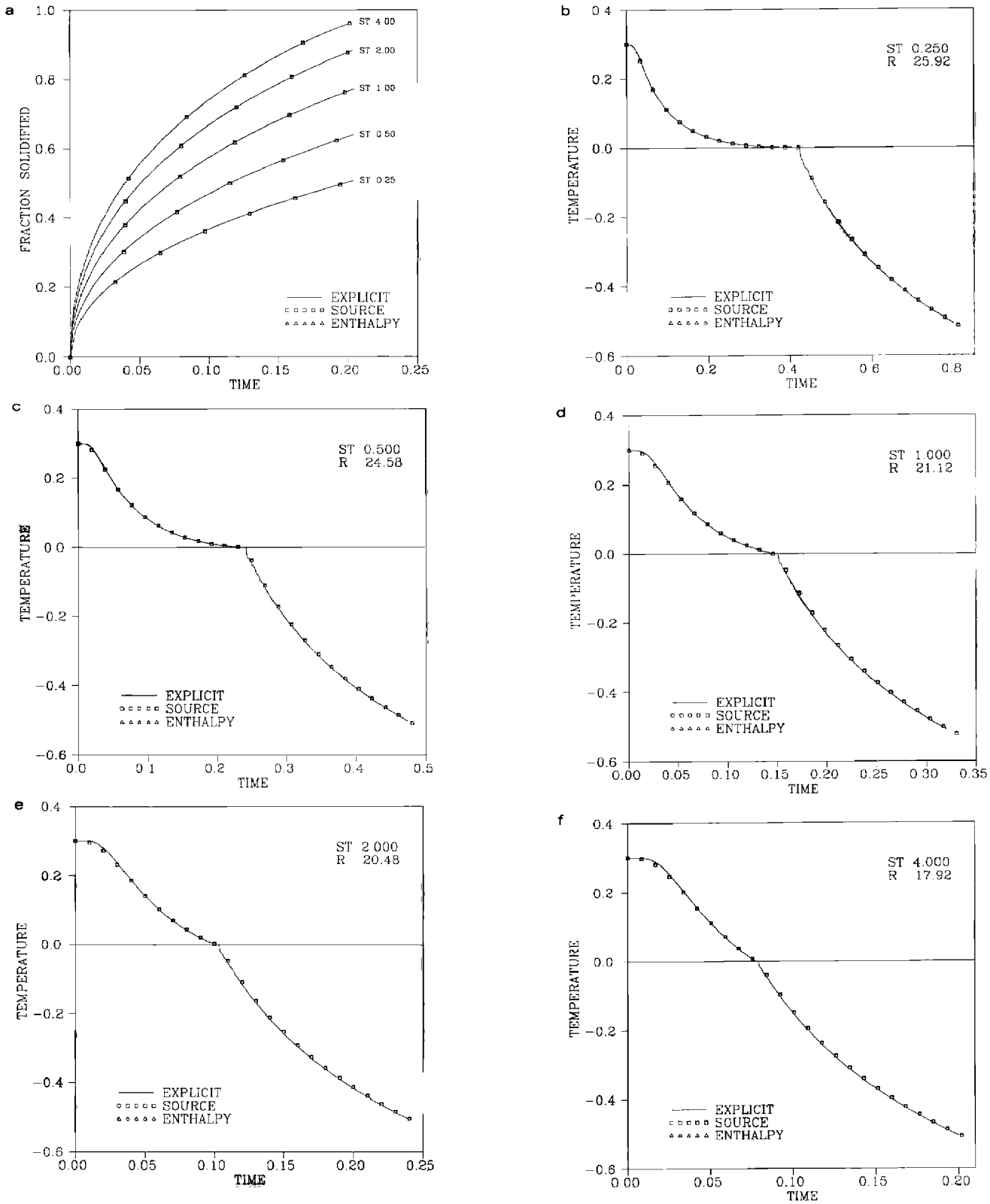

Fig. 3 Solidification in a corner region: progress of the solidification for various Stefan numbers (a) and temperature history at $X = Y = 0.5$ for Stefan number 0.25 (b); Stefan number 0.50 (c); Stefan number 1.00 (d); Stefan number 2.00 (e); and Stefan number 4.00 (f)

where θ, ζ, X, and τ are the nondimensional temperature, enthalpy, length, and time, respectively, L is the length of the domain, and T_m is the melting point.

Using these variables, the basic enthalpy equation can be written as

$$\frac{\partial \zeta}{\partial \tau} = \nabla^2 \theta. \tag{A.5}$$

The source-based formulation becomes

$$\frac{\partial \theta}{\partial \tau} = \nabla^2 \theta - \frac{1}{St} \frac{\partial g}{\partial \tau}, \tag{A.6}$$

where St is the Stefan number given by

$$St = \frac{c(T_m - T_w)}{L}, \tag{A.7}$$

where L is the latent heat associated with the solidification. The initial and boundary conditions become, respectively,

$$\theta = \theta_{ini} \quad \text{at } \tau = 0 \tag{A.8}$$

and

$$\theta = -1 \quad \text{at } X = 0 \quad \text{and} \quad Y = 0 \text{ surfaces} \tag{A.9}$$

A nondimensional number R that controls the stability of the transient algorithm is defined as

$$R = \Delta\tau \left(\frac{1}{\Delta X^2} + \frac{1}{\Delta Y^2} \right). \tag{A.10}$$

For the Euler forward explicit scheme with a finite element discretization in space, the stability limit is given by $R = 0.375$.

ACKNOWLEDGMENT

We would like to thank the referees, whose valuable comments helped enhance the quality of presentation of this paper. The research was supported in part by an Army High Performance Computing Research Center grant and a National Science Foundation grant (CCR-8722260). We would also like to acknowledge the Minnesota Supercomputing Institute (MSI) for providing time on the CRAY-2.

BIOGRAPHIES

Anthony T. Chronopoulos obtained his B.Sc. and M.S. degrees in mathematics from the University of Athens (Greece) and the University of Minnesota in 1979 and 1981, respectively. He received his Ph.D. in computer science from the University of Illinois at Urbana-Champaign in 1986. Since then he has been an assistant professor in the Department of Computer Science at the University of Minnesota. His interests include numerical algorithms and parallel processing.

C. R. Swaminathan received his B.Tech. degree in metallurgical engineering from the Banaras Hindu University (India) in 1988 and his M.S. degree in civil and mineral engineering from the University of Minnesota in 1990. Since then he has been working toward a Ph.D. degree in mechanical engineering at the University of Minnesota. His interests include development of numerical techniques to model solidification processes and phenomena.

V. R. Voller received his B.S. and M.S. degrees in applied mathematics from the University of East Anglia (England) in 1975 and 1977, respectively. He received his Ph.D. in numerical and mathematical modeling from Sunderland Polytechnic (England) in 1980. Since 1985 he has been an associate professor in the Department of Civil and Mineral Engineering at the University of Minnesota. His major interests include development of numerical techniques for thermal-fluid problems, with particular emphasis on problems involving phase change.

SUBJECT AREA EDITOR

William Jalby

REFERENCES

Brussino, G., and Sonnad, V. 1989. A comparison of direct and preconditioned iterative techniques for sparse, unsymmetric systems of linear equations, *Internat. J. Numer. Methods Engrg.* 28:801–815.

Chronopoulos, A. T., and Ma, S. 1989a. Implementation of iterative methods for large sparse nonsymmetric linear systems on a parallel vector machine. *Internat. J. Supercomput. Appl.* 4(4):19–24.

Chronopoulos, A. T., and Ma, S. 1989b. On squaring Krylov subspace iterative methods for nonsymmetric linear systems. UM CSci. TR-89-67. Minneapolis: University of Minnesota.

Comini, G., Del Giudice, S., Lewis, R. W., and Zienkiewicz, O. C. 1974. Finite element solution of nonlinear heat conduction problems with special reference to phase change. *Internat. J. Numer. Methods Engrg.* 8:613–624.

Concus, P., Golub, G. H., and O'Leary, D. P. 1976. A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations. In *Sparse matrix computations,* edited by J. R. Bunch and D. Rose. New York: Academic Press.

Crank, J. 1984. *Free and moving boundary problems.* Oxford: Clarendon Press.

Dalhuijsen, A. J., and Segal, A. 1986. Comparison of finite element techniques for solidification problems. *Internat. J. Numer. Methods Engrg.* 23:1807–1829.

Eisenstat, S. C., Elman, H. C., and Schultz, M. H. 1983. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.* 20:345–357.

Fletcher, R. 1976. Conjugate gradient methods for indefinite systems. *Lecture Notes in Math.* 509:73–89.

Hestenes, M., and Steifel, E. 1952. Method of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards* 49:409–439.

Jennings, A., and Malik, G. M. 1978. The solution of sparse linear equations by the conjugate gradient method. *Internat. J. Numer. Methods Engrg.* 12:141–158.

Meijerink, J. A., and Van der Vorst, H. A. 1977. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.* 31:148–162.

Rappaz, M. 1989. Modeling of microstructure formation in solidification processes. *Internat. Mat. Rev.* 34:93–123.

Saad, Y., and Schultz, M. 1986. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.* 7:856–859.

Shamsundar, N., and Sparrow, E. M. 1975. Analysis of multidimensional conduction phase change via the enthalpy model. *Trans. ASME, J. Heat Trans.* 97:333–340.

Sonneveld, P. 1989. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.* 10:36–52.

Van der Vorst, H. A. 1982. A vectorizable variant of some ICCG methods. *SIAM J. Sci. Statist. Comput.* 3:350–356.

Vinsome, P. 1976. An iterative method for solving sparse sets of simultaneous linear equations. *J. Res. Nat. Bur. Standards* 49:409–439.

Voller, V. R. 1990. A fast implicit finite difference method for the analysis of phase change problems. *Num. Heat Trans.* 17B:115–129.

Voller, V. R., and Prakash, C. 1987. A fixed grid numerical modeling methodology for convection/diffusion mushy region phase change problems. *Internat. J. Heat Mass Trans.* 30:1709–1719.

Voller, V. R., and Swaminathan, C. R. 1991. A general source-based method for solidification phase change. *Num. Heat Trans.* 19B:175–190.

Voller, V. R., Swaminathan, C. R., and Thomas, B. G. 1990. Fixed grid techniques for phase change problems: a review. *Internat. J. Numer. Methods Engrg.* 30:875–898.

White, R. E. 1986. A nonlinear parallel algorithm with application to the Stefan problem. *SIAM J. Numer. Anal.* 23:639–653.

Williams, M. A. 1987. Iterative solution of a nonlinear system arising in phase change problems. Technical report ORNL-6398. Oak Ridge, Tenn.: Oak Ridge National Laboratory.

Young, D. M., and Lea, K. C. 1980. Generalized conjugate gradient accelerations of nonsymmetric iterative methods. *Linear Algebra Appl.* 34:159–194.

Zienkiewicz, O. C., and Taylor, C. 1989. *The finite element method.* New York: McGraw Hill.