Applied and
Computational
Mathematics
Division

NISTIR 5352

Computing and Applied Mathematics Laboratory

A Boundary Conforming Grid Generation System for Interface Tracking

B.V. Saunders

February 1994

Technology Administration
U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Gaithersburg, MD 20899

NISTIR 5352

A Boundary Conforming Grid Generation System for Interface Tracking

B. V. Saunders

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Computing and Applied Mathematics Laboratory
Applied and Computational Mathematics Division
Gaithersburg, MD 20899

February 1994

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director

A Boundary Conforming Grid Generation System for Interface Tracking

B. V. Saunders
National Institute of Standards and Technology
Gaithersburg, MD 20899
USA

Abstract

The development of an algebraic grid generation system to track a solid-liquid interface during directional solidification of a binary alloy is discussed. A single mapping, constructed with tensor product B-splines, is proposed for calculations of both shallow and deep solidification cells. The initial spline coefficients for the coordinate mapping are modified to minimize a discrete functional that regulates the smoothness and orthogonality of the mesh. The use of transfinite blending function interpolation to obtain an initial grid is examined.

Keywords: boundary fitted grid generation, algebraic grid generation, adaptive, B-splines, transfinite blending functions, directional solidification

1. Introduction

One of the well known techniques used to study the microstructures that develop during solidification of binary alloys is Bridgman growth, a directional solidification technique in which a sample of the alloy is drawn through a constant temperature gradient at a uniform rate of speed, V, as shown in Figure 1. A considerable amount of theoretical work has focused on examining the morphological instabilities of the growing solid-liquid interface [1-6]. Mullins and Sekerka [1] used linear stability theory to predict the critical velocity for the onset of instability for a planar interface. Experimental observations confirm the validity of their results and show that after the onset of instability the structure of the interface can change from planar to cellular to dendritic and back again as the growth velocity is increased [7, 8]. Coriell et al. [2] extended the results of Mullins and Sekerka, including the effects from convection in the liquid.

Although cellular microstructures are much simpler than dendritic, as the control parameters, growth velocity or temperature gradient, are changed, the cells may become very deep and narrow with re-entrant bulb-like shapes. To successfully track the interface, the grid generation mapping must adapt to large deformations of the interface shape while maintaining as much orthogonality and smoothness as possible. Ettouney and Brown [4] successfully modeled slightly nonplanar interfaces by using an algebraic grid generation system where the interface was described in terms of a univariate function. Unfortunately, this representation of the interface, called a Mongé transformation, fails for cells with vertical or re-entrant

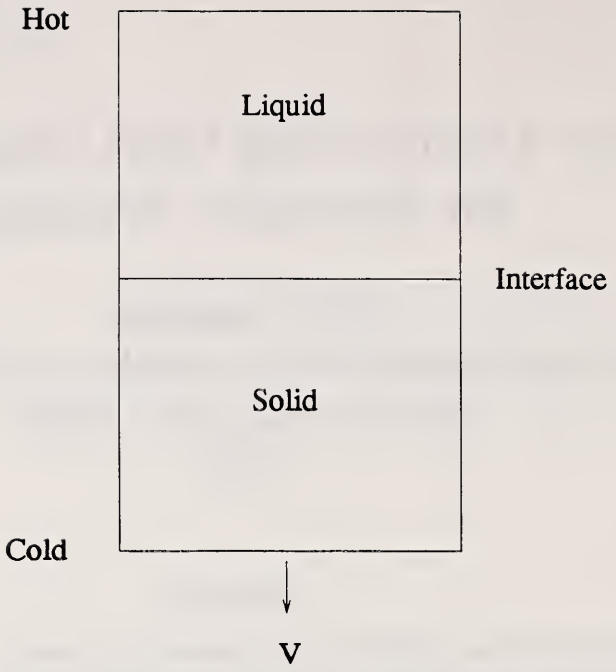


Figure 1: Bridgman growth technique.

walls. To overcome that problem, Ungar and Brown [5] developed a representation defined by the division of the interface into disjoint sections that could be expressed as separate Mongé transformations written in either polar or rectangular coordinates. With this mixed transformation, Ungar and Brown modeled cellular interfaces with grooves as much as 15 times longer than the cell wavelength. Ideally, analysts want a single grid generation technique that can track the interface as it changes from a shallow deformation to a narrow and deeply grooved cell with re-entrant sidewalls. Furthermore, it should capture the change in wavelength if the cell splits.

Tsiveriotis and Brown [6] have made some progress in this area by using a two-step procedure. In one direction the coordinate is defined by a generalized Poisson equation with a scaling condition that forces the grid lines into concave areas of the interface. The other coordinate is obtained by minimizing smoothness and orthogonality functionals similar to those of Brackbill and Saltzman [9]. Tsiveriotis and Brown show impressive examples of grids developed for narrow and re-entrant cells, but it is not clear whether this method is as successful as the previous method of Ungar and Brown because the most narrow cells in the earlier paper are not redone. Also, to obtain better results, they fix a horizontal line near the interface to decouple the far field domain from the domain near the interface. This essentially creates two separate domains whose grid cells are not smoothly connected near the line. This is fine for finite element calculations, but suggests problems for those who are interested in using the technique to create grids suitable for finite difference schemes.

This paper describes progress to date in the development of an algebraic technique for generating a boundary/interface fitted coordinate system that is smooth enough to be used for finite difference calculations. No partial differential equations (pdes) are solved to obtain the coordinate system, and the same system is used for the entire domain. The computation of the interface is not discussed in this paper. It is assumed that the interface is available as a discrete set of points, having been determined by some means such as an evaluation of the Gibbs-Thomson equation which relates the interface mean curvature, concentration of solute in the liquid, and the melting temperature near the interface [8, 10]. The feasibility

of the grid generation technique is evaluated by examining its ability to create grids that conform to interface shapes typical of those seen in experimental observations of directional solidification of binary alloys by the Bridgman growth technique [7, 8]. The grid generation algorithm is an extension of an algebraic technique for generating boundary fitted grids [11]. This paper discusses the modifications needed and addresses the problems in tracking an interface that deforms from a planar shape to a deep, re-entrant cellular microstructure.

Section 2 provides a brief description of two algebraic techniques for generating coordinate systems: transfinite blending function interpolation, the technique used to obtain an initial approximation for the interface fitted mesh, and a boundary fitted grid generation technique that uses a mapping constructed with B-splines. The B-spline mapping is extended to create the mapping for the interface tracking grid generation system . Section 3 explains the construction of the grid generation mapping and Section 4 examines the results to date.

2. Algebraic Grid Generation Techniques

In algebraic grid generation, a direct transformation describes the relationship between the computational and physical domains. The transformation is constructed so that it interpolates the boundary points and/or points in the interior. It may also be constrained to match derivatives. No partial differential equations are solved to obtain the curvilinear coordinates, so algebraic techniques can be easier to construct than pde methods, and give easier control over grid characteristics such as orthogonality and grid point spacing. However, these methods are sometimes criticized for allowing discontinuities on the boundary to propagate into the interior and for not generating grids as smooth as those generated by pde methods. Nevertheless, algebraic techniques have been used successfully to generate grids in both two and three dimensions [11-16].

2.1. Transfinite Blending Function Interpolation

One of the most common and easiest algebraic methods to implement is transfinite blending function interpolation, where interior regions are represented in terms of boundary functions. This technique, which has been widely used for problems in grid generation [11, 12, 14, 15, 16], was originally developed for problems in computer aided design. Gordon and Hall [14] adapted and applied the technique to grid generation for finite element and finite difference calculations in the late sixties and early seventies. A simple example is illustrated by the mapping $\tilde{\mathbf{T}}$ from the unit square I_2 to the physical domain defined by

$$\begin{aligned} \tilde{\mathbf{T}}(\xi,\eta) &= \sum_i \Phi_i(\xi)\mathbf{f}(\xi_i,\eta) \\ &+ \sum_j \Psi_j(\eta)\mathbf{f}(\xi,\eta_j) \\ &- \sum_{i,j} \Phi_i(\xi)\Psi_j(\eta)\mathbf{f}(\xi_i,\eta_j) \end{aligned} \tag{1}$$

where $0 = \xi_0 < \ldots < \xi_M = 1$ and $0 = \eta_0 < \ldots < \eta_N = 1$. The continuous vector-valued function $\mathbf{f}$ describes the boundary of the physical domain. Φ and Ψ are "blending" or "connecting" functions that satisfy the conditions

$$\Phi_i(\xi_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{2}$$

$$\Psi_k(\eta_l) = \delta_{kl} = \begin{cases} 1, & k = l \\ 0, & k \neq l \end{cases} \tag{3}$$

Therefore, if M=N=1, one might choose Φ and Ψ to be linear Lagrange polynomials so that $\tilde{\mathbf{T}}$ matches $\mathbf{f}$ on the boundary of I_2. If M and N are larger and derivative information is given, the blending functions can be chosen so that interior curves and the derivatives there are matched. Unfortunately, transfinite blending function interpolation may allow boundary singularities to propagate into the interior of the mesh. Furthermore, if the boundary is nonconvex, the grid lines may overlap the boundary. The choice of blending functions and parametrization of the boundary function are very important in alleviating these problems. Transfinite blending function interpolation provides a relatively easy way of obtaining an initial grid that can be refined and smoothed by other techniques, whether algebraic, pde, or variational. However, unfortunately, some techniques cannot use the grid if it contains areas where the grid lines overlap, that is, areas of negative Jacobian [17].

2.2. Boundary Fitted Grid Generation Using Tensor Product B-splines

Saunders [11] developed an algebraic grid generation system that uses a mapping $\mathbf{T}$ from the unit square I_2 to a physical domain of arbitrary shape defined by

$$\mathbf{T}(\xi,\eta) = \begin{pmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_{ij} B_{ij}(\xi,\eta) \\ \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{ij} B_{ij}(\xi,\eta) \end{pmatrix}, \tag{4}$$

where $0 \leq \xi, \eta \leq 1$ and each B_{ij} is the tensor product of cubic B-splines. Hence, $B_{ij}(\xi,\eta) = B_i(\xi)B_j(\eta)$ where B_i and B_j are elements of cubic B-spline sequences associated with finite nondecreasing knot sequences, say, $\{s_i\}_1^{m+4}$ and $\{t_j\}_1^{n+4}$, respectively. As defined by de Boor [18], B-splines are essentially piecewise polynomials with continuity conditions at each breakpoint determined by the repetition of the breakpoint value in the associated knot sequence. For a typical cubic B-spline B_j, the value of the B-spline is determined by the five knots t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4}. Its support is small, that is, B_j can be nonzero only on the interval $[t_j, t_{j+4}]$. Consequently, only four B-splines, B_{j-3}, B_{j-2}, B_{j-1}, B_j can be nonzero on the interval (t_j, t_{j+1}). The initial coefficients are chosen so that the mapping approximates transfinite blending function interpolation. More specifically, the coefficients are selected to produce a variation diminishing spline approximation to the transfinite blending function interpolant $\tilde{\mathbf{T}}$ defined in equation (1). In short, this means the coefficients are obtained by evaluating $\tilde{\mathbf{T}}$ at average knot values as discussed in [18]. This shape preserving approximation reproduces straight lines and preserves convexity [11, 18] . To increase the orthogonality and smoothness of the grid lines the initial coefficients are modified to minimize the functional

$$F = \int_{I_2} w_1 \left\{ \left(\frac{\partial J}{\partial \xi}\right)^2 + \left(\frac{\partial J}{\partial \eta}\right)^2 \right\} + w_2 \left\{ \frac{\partial \mathbf{T}}{\partial \xi} \cdot \frac{\partial \mathbf{T}}{\partial \eta} \right\}^2 dA \tag{5}$$

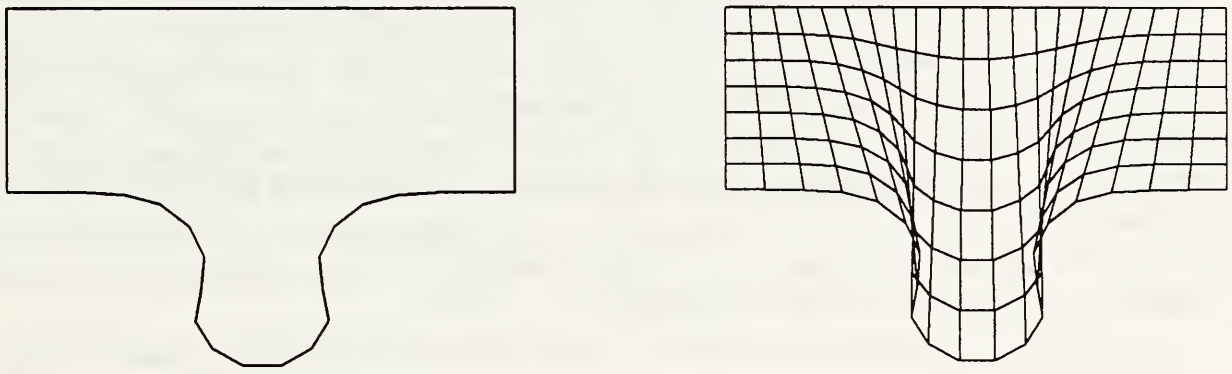
where $\mathbf{T}$ denotes the grid generation mapping, J is the Jacobian of the mapping, and w_1 and w_2 are weight constants. When w_1 is large, the variation of the Jacobian values at nearby points will be small, thereby decreasing skewness. When w_2 is large, the dot product term will be small, causing the grid lines to approach orthogonality. To avoid solving the Euler equations for the variational problem, this functional is approximated in the computer code by the sum

$$G = \sum_{i,j} w_1 \left[\left(\frac{J_{i+1,j} - J_{ij}}{\Delta\xi}\right)^2 + \left(\frac{J_{i,j+1} - J_{ij}}{\Delta\eta}\right)^2 \right] \Delta\xi\Delta\eta$$

$$+\sum_{i,j} w_2 Dot_{ij}^2 \triangle\xi\triangle\eta \tag{6}$$

where J_{ij} is the Jacobian value and Dot_{ij} is the dot product of $\partial\mathbf{T}/\partial\xi$ and $\partial\mathbf{T}/\partial\eta$ at mesh point (ξ_i, η_j) on the unit square. G is actually a fourth degree polynomial in each spline coefficient so the minimum is found by using a cyclic coordinate descent technique which sequentially finds the minimum with respect to each coefficient. This technique allows the minimization routine to take advantage of the small support of B-splines when evaluating the sums that comprise G.

An application of the grid generation algorithm is shown in Figures 2 and 3 for a puzzle shaped domain. The boundary and initial grid are shown in Figure 2. The initial grid was constructed using linear Lagrange polynomials for the blending functions. Note that the grid lines overlap the nonconvex boundary. Figure 3a shows the mesh obtained after the spline coefficients are modified to minimize the smoothing functional. The skewed areas have been eliminated and the overlapping grid lines have been pulled into the interior. The refined mesh in Figure 3b is obtained by evaluating $\mathbf{T}$ at additional points on the square. The concentration near the bottom is achieved by applying an exponential function to the η variable. The algorithm discussed in this paper extends the boundary fitted mapping to fit a curve in the interior of the physical domain. As in the case of the boundary fitted algorithm, it will be shown that the new algorithm is robust enough to smooth and untangle an initial grid containing skewed and overlapping grid lines.



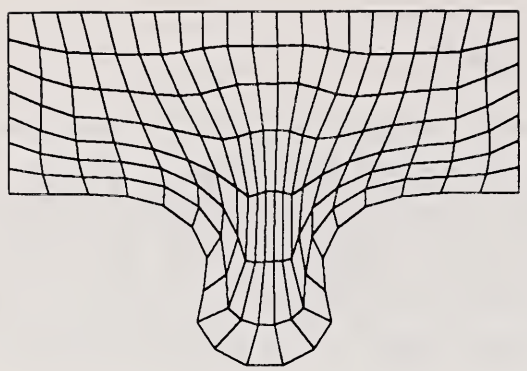
(a) Puzzle shaped boundary.

(b) Initial grid.

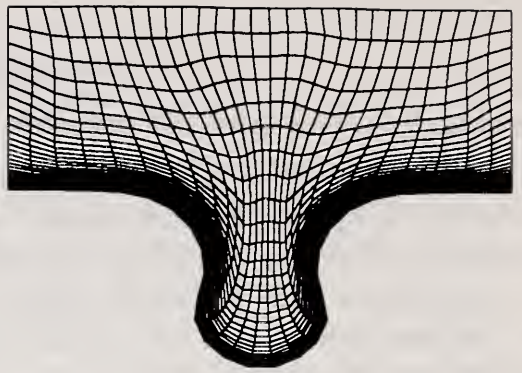
Figure 2: Puzzle shaped domain. Initial grid was produced using an approximation of transfinite blending function interpolation.

3. An Interface Tracking Grid Generation System

The boundary fitted grid generation mapping discussed in the previous section forms the basis for the interface tracking mapping. However, the mapping must now match the interface curve on the interior of the physical domain in addition to fitting the outer physical boundary. Furthermore, the system must be adaptive since the grid lines must change to follow the deforming interface while maintaining as much smoothness and orthogonality as possible.



(a) Optimized grid.



(b) Optimized grid refined.

Figure 3: Optimized puzzle grids. Minimization of smoothing functional successfully pulls grid lines inside the boundary.

3.1. Grid Generation Mapping

The proposed grid generation mapping, $\mathbf{T}$, maps the unit square, I_2, onto the physical domain and is constructed so that the interface is the coordinate curve $\eta = 1/2$ as shown in the figure. As before, the mapping has the form

$$\mathbf{T}(\xi,\eta) = \begin{pmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_{ij} B_{ij}(\xi,\eta) \\ \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{ij} B_{ij}(\xi,\eta) \end{pmatrix}, \tag{7}$$

where $0 \le \xi,\eta \le 1$ and $B_{ij}(\xi,\eta) = B_i(\xi)B_j(\eta)$ where B_i and B_j are elements of cubic B-spline sequences associated with finite nondecreasing knot sequences, $\{s_i\}_1^{m+4}$ and $\{t_j\}_1^{n+4}$, respectively. The spline coefficients for $\mathbf{T}$ can be divided into three groups. The boundary coefficients are the coefficients of those B_{ij} that are nonzero on the boundary of I_2. Since $\mathbf{T}$ is defined so that the interface corresponds to coordinate curve $\eta = 1/2$, the coefficients of the B_{ij} that are nonzero when $\eta = 1/2$ are called the interface coefficients. The remaining coefficients are called the interior coefficients.

Initially, the coefficients are chosen to approximate the following transfinite blending function interpolant that matches the outer boundary and interface of the physical domain:

$$\begin{aligned} \tilde{\mathbf{T}}(\xi,\eta) &= \sum_{i=0}^{1} \Phi_i(\xi)\mathbf{f}(\xi_i,\eta) \\ &+ \sum_{j=0}^{2} \Psi_j(\eta)\mathbf{f}(\xi,\eta_j) \\ &+ \beta(\eta)\frac{\partial \mathbf{f}}{\partial \eta}(\xi,1/2) \\ &- \sum_{i=0}^{1}\sum_{j=0}^{2} \Phi_i(\xi)\Psi_j(\eta)\mathbf{f}(\xi_i,\eta_j) \\ &- \sum_{i=0}^{1} \Phi_i(\xi)\beta(\eta)\frac{\partial \mathbf{f}}{\partial \eta}(\xi_i,1/2) \end{aligned} \tag{8}$$

where $\xi_0 = 0$, $\xi_1 = 1$ and $\eta_0 = 0$, $\eta_1 = 1/2$, $\eta_2 = 1$. The continuous vector valued function $\mathbf{f}$, constructed from boundary data input by the user, maps the boundary of the square to

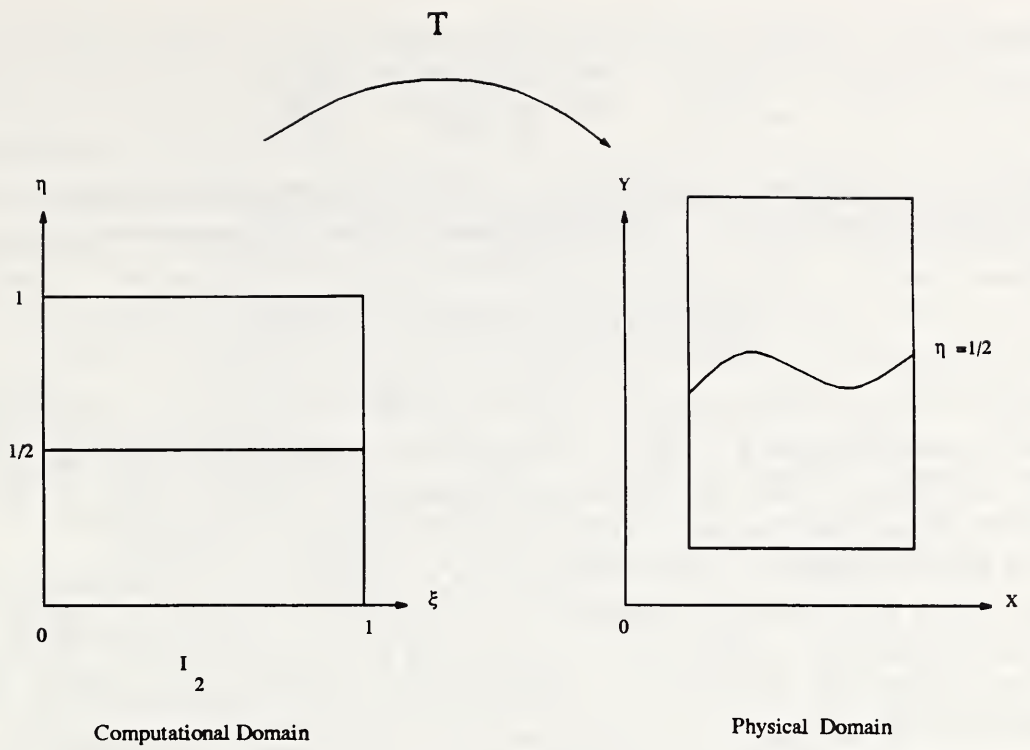


Figure 4: Grid generation mapping.

the boundary of the physical domain and the line $\eta = 1/2$ to the interface. The transfinite mapping $\tilde{\mathbf{T}}$ interpolates $\mathbf{f}$ on the boundary and matches $\mathbf{f}$ and $\partial\mathbf{f}/\partial\eta$ at $\eta = 1/2$. The blending functions Φ_i are linear Lagrange polynomials satisfying

$$\Phi_i(\xi_j) = \delta_{ij} \text{ for } i,j = 0,1$$

while Ψ_r and β are hermite quintic blending functions satisfying

$$\begin{aligned} \Psi_r(\eta_s) &= \delta_{rs} \\ \Psi_r'(\eta_s) &= 0 \text{ for } r,s = 0,1,2 \end{aligned}$$

and

$$\begin{aligned} \beta(\eta_s) &= 0 \\ \beta'(\eta_s) &= \delta_{1s} \text{ for } s = 0,1,2\ . \end{aligned}$$

Although linear, quadratic, and cubic polynomials were also tested as blending functions for the η coordinate, the hermite quintic polynomials produced the least amount of skewness and overlap of grid cells. To force orthogonality at the interface, the components of the derivative

$$\frac{\partial \mathbf{f}}{\partial \eta}(\xi, 1/2)$$

are chosen to be

$$\begin{aligned} \frac{\partial f_1}{\partial \eta} &= -K\frac{\partial f_2}{\partial \xi}/L \\ \frac{\partial f_2}{\partial \eta} &= K\frac{\partial f_1}{\partial \xi}/L \end{aligned}$$

where

$$L = \sqrt{\left(\frac{\partial f_1}{\partial \xi}\right)^2 + \left(\frac{\partial f_2}{\partial \xi}\right)^2}$$

and K is a user defined orthogonality constant that regulates the magnitude of the normal vectors at the interface. The initial coefficients for the grid generation mapping $\mathbf{T}$ are defined by

$$\begin{pmatrix} \alpha_{ij} \\ \beta_{ij} \end{pmatrix} = \tilde{\mathbf{T}}(s_i^*, t_j^*) \text{ for } i = 1, \ldots, m;\ j = 1, \ldots, n \tag{9}$$

where $s_i^* = (s_{i+1} + \ldots + s_{i+3})/3$, $i = 1, \ldots, m$ and $t_j^* = (t_{j+1} + \ldots + t_{j+3})/3$, $j = 1, \ldots, n$. With these coefficients $\mathbf{T}$ produces a variation diminishing spline approximation to $\tilde{\mathbf{T}}$. Variation diminishing spline approximations, discussed in detail in [18], are shape preserving approximations that reproduce straight lines and preserve convexity.

As with the boundary fitted system, the smoothness and orthogonality of grid lines is enhanced by modifying the coefficients to minimize the discrete smoothing functional defined in equation (6). However, the flexibility of the boundary and the interface coefficients is limited since even small changes in the coefficients can destroy the shape preserving properties of the spline mapping. Fortunately, the B-spline knot sequences $\{s\}$ and $\{t\}$ can be chosen so that the boundary coefficients affect only the boundary and a narrow area inside the boundary. Using the continuity properties of B-splines with repeated knots [18], one can show that if the first four knots of $\{s_i\}_1^{m+4}$ and $\{t_j\}_1^{n+4}$ are 0, the last four knots are 1, and the rest located in the interval $(0,1)$, then the only boundary coefficients are α_{1j}, β_{1j} and α_{mj}, β_{mj} for $j = 1, \ldots, n$ and α_{i1}, β_{i1} and α_{in}, β_{in} for $i = 1, \ldots, m$. Unless all the "interior" knots are clustered near the center of $(0,1)$, the boundary coefficients will have a minimal effect on the interior of the square. For example, α_{1j} and β_{1j} will only affect points on the support of B_{1j}, that is, the narrow band $[0, s_5] \times [t_j, t_{j+4}]$ illustrated in Figure 5, where s_5 is the first interior knot in $\{s_i\}_1^{m+4}$. In fact the area of influence of all the boundary

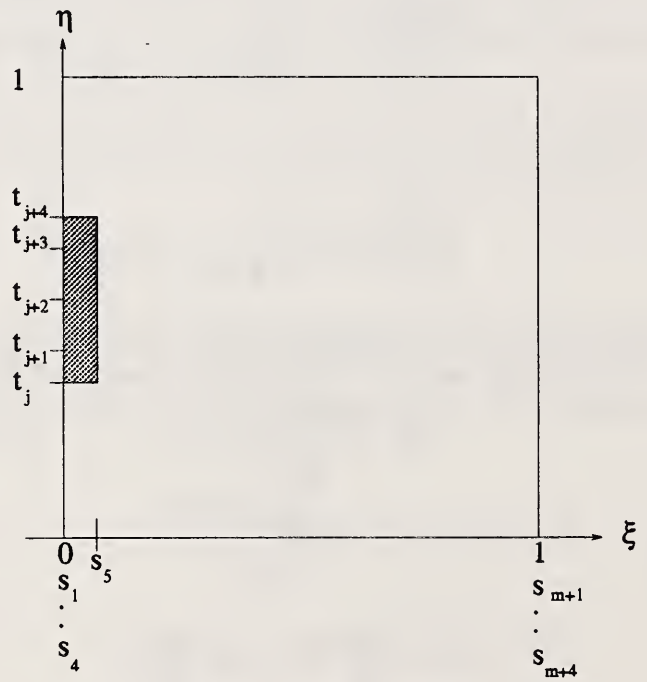


Figure 5: Support of tensor product B-spline B_{1j}.

coefficients will be a narrow region along the boundary of the square. Consequently, even if the boundary coefficients are fixed to guarantee that "boundary fittedness" is maintained, the smoothing functional should be able to produce a significant amount of orthogonality and smoothness in the grid.

Ideally, however, one would like to include the boundary coefficients in the minimization process to ensure that grid lines near the boundary are smooth and orthogonal. When the boundary is rectangular, as it is with Bridgman growth, this is indeed possible. The four sides are vertical or horizontal line segments satisfying either $x(0,\eta) = x_0$, $x(1,\eta) = x_1$, $y(\xi,0) = y_0$, or $y(\xi,1) = y_1$, where x_0, x_1, y_0, and y_1 are constants. By fixing the appropriate α or β coefficients one can guarantee that these boundary equations will remain satisfied. For example, on the left vertical boundary where $x(0,\eta) = x_0$, only $B_{1j}, j = 1,\ldots,n$ are nonzero there so that $x(0,\eta) = \sum_{j=1}^{n} \alpha_{1j} B_{1j}(0,\eta)$. Hence, $x(0,\eta) = x_0$ remains true if α_{1j}, $j = 1,\ldots,n$, that is, the α boundary coefficients for that side, are fixed. The corresponding β coefficients, β_{1j}, $j = 1,\ldots,n$, are free to move. Similarly, for the right side, the α boundary coefficients, α_{mj}, $j = 1,\ldots,n$ are fixed. For the horizontal sides, the β boundary coefficients are fixed. With the boundary coefficients fixed in this manner, minimizing the smoothing functional causes a redistribution of the boundary points along the vertical or horizontal lines. Of course, it is possible that a boundary point could be pushed beyond the endpoint of the line segment, but this produces a negative Jacobian, a problem generally corrected by the smoothing action of the Jacobian terms in the functional.

Since the interface may be quite complex, very little change is permitted in the interface coefficients when the other coefficients are adjusted by the minimization of the smoothing functional. Unfortunately, unlike the boundary coefficients, the interface coefficients affect a significant number of points on the interior besides those mapped onto the interface. To determine the interface coefficients one first finds l such that $1/2 \in [t_l, t_{l+1}]$. Then the interface coefficients will be α_{ik}, β_{ik} where $1 \le i \le m$ and $l-3 \le k \le l$. These are the coefficients of the tensor product B-splines that might be nonzero when $\eta = 1/2$. However, fixing these coefficients affects the mapping $\mathbf{T}$ not only on $[0,1] \times [t_l, t_{l+1}]$, but also on the much larger band, $[0,1] \times [t_{l-3}, t_{l+4}]$ seen in Figure 6, that is, the total support of all the tensor product B-splines that are possibly nonzero on $[0,1] \times [t_l, t_{l+1}]$. This band can be narrowed significantly by using a larger number of knots for the t sequence and concentrating some near $1/2$. However, concentrating the knots too closely will affect the smoothness of the grid lines. On the other hand, since a continuous B-spline is close to zero near the boundary of its support, all interface coefficients do not equally affect the mapping of the interface. In particular, allowing the coefficients $\alpha_{i,l-3}$ and $\beta_{i,l-3}$ for $1 \le i \le m$ to move appears to have very little effect on the accuracy of the interface mapping. Nevertheless, the inflexibility of the interface coefficients suggests that they should be chosen to produce as much orthogonality and smoothness as possible at the outset.

3.2. The Algorithm

Although the calculation of the interface must be addressed in order to integrate the grid generation system with the equations that model the directional solidification problem, the current code assumes that both the boundary and interface data are available as sets of discrete points. The focus here is on demonstrating the ability of the grid generation mapping to adapt as the interface changes.

Using the boundary data and initial interface data, the code first computes the B-splines needed to define $\mathbf{T}$. If $\{s\}_1^{m+4}$ and $\{t\}_1^{n+4}$ are the sequences chosen to define the B-splines,

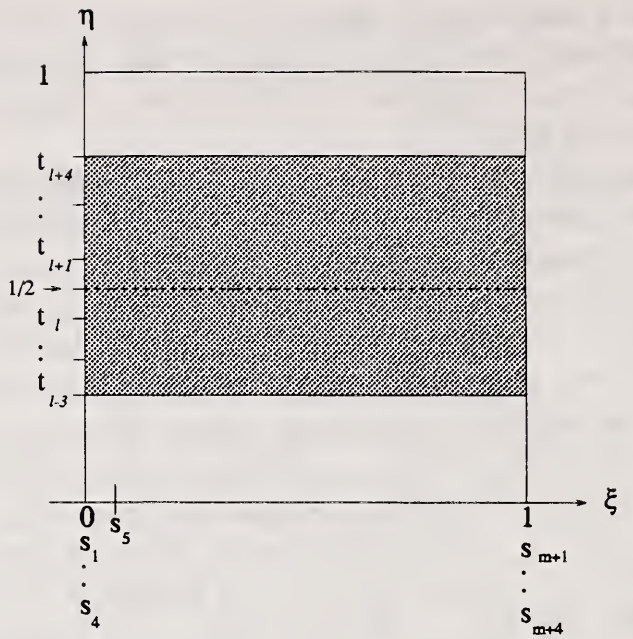


Figure 6: Support of tensor product B-splines nonzero when $\eta = 1/2$.

then the total number of coefficients used in the mapping will be $2 \times m \times n$. Consequently, although the number of knots must be sufficient to accurately capture the shape of a complicated interface, this must be balanced by the fact that increasing the number of knots increases the size of the optimization problem. Fortunately, the accuracy of the mapping is also improved by concentrating the t knots near the interface. This is accomplished by replacing $\{t\}$ with the sequence $\{\gamma(t)\}$ where γ is the hyperbolic sine function

$$\gamma(t) = \frac{\sinh(2ct - c) + \sinh(c)}{2\sinh(c)}$$

where c is a constant that determines the degree of concentration. This has the additional beneficial effect of decreasing the area influenced by the interface coefficients. The mesh over which the smoothing functional is evaluated must have at least $2 \times m \times n$ grid points, and the concentration of points in the η direction should be similar to the concentration of knots for sequence $\{t\}$.

As stated above, the transfinite blending function mapping $\tilde{\mathbf{T}}$ is evaluated at average knot values to obtain coefficients that produce a variation diminishing spline approximation to $\tilde{\mathbf{T}}$. The coefficients are modified to minimize the discrete smoothing functional to obtain more smoothness and orthogonality in the grids. This step is unnecessary if the interface is planar. The actual grid is obtained by evaluating $\mathbf{T}$ at equally spaced values of ξ and η. Replacing η with the sinh mapping $\gamma(\eta)$ concentrates the grid near the interface. To create a grid for the next interface, the code first reads the data for the new interface. The old interface coefficients plus one layer of coefficients above and below them, that is, α_{ik}, β_{ik} where $1 \leq i \leq m$ and $l - 4 \leq k \leq l + 1$, are replaced by coefficients that produce a variation diminishing spline approximation to the transfinite blending function mapping that interpolates the new interface. The smoothing functional is then minimized in order to eliminate any overlap of grid lines and improve smoothness and orthogonality. The process is repeated to obtain a grid for the next interface.

These steps are illustrated in Figure 7 which shows a sequence of grids concentrated near curves that demonstrate the type of sinusoidal deformation that appears soon after the

onset of instability during the directional solidification of a metal alloy. The first grid, representing a planar interface, requires no additional smoothing. The eight other grids consist of pairs representing the initial and smoothed grids, respectively. The minimization routine successfully "untangles" overlapping lines while enhancing the smoothness and orthogonality of the grid lines. A concentration of grid points near the interface was obtained by evaluating $\mathbf{T}(\xi, \gamma(\eta))$ with sinh concentration constant $c = 4$.

3.3. Implementation

The usefulness of this technique for generating grids for solidification problems will largely be determined by how fast and efficiently the grids can be produced. Of course, the primary bottleneck in the grid generation routine is the minimization of the smoothing functional. Depending on the number of knots used to define the grid generation mapping, the optimization problem can be of fairly large scale. For example, if the knot sequences $\{s\}$ and $\{t\}$ both have 22 elements, then the number of spline coefficients, α_{ij} and β_{ij}, will be $2 \times (22-4) \times (22-4)$ or 648 [18]. Hence we have a minimization problem involving 648 variables. An ongoing problem is to determine the best optimization technique to use. Currently, the code uses a modified cyclic coordinate descent algorithm which minimizes a multivariable function by sequentially finding the minimum with respect to each variable. The cyclic coordinate descent method is relatively easy to implement and requires no gradient information to determine the direction of descent, but in general it has a much slower convergence rate than conjugate gradient or quasi-Newton methods [19]. However, the simplicity of the technique makes it easy to exploit the properties of B-splines during the minimization process. For each spline coefficient, the sum in equation (6) need only be computed over the area that the coefficient affects. The small support of B-splines means that this area can be quite small. The example illustrated in Figure 8 shows the support of tensor product B-spline $B_{6,j}$ for specified knot sequences $\{s\}$ and $\{t\}$ and a given distribution of mesh points on the computational domain. If the functional is minimized with respect to coefficients $\alpha_{6,j}$ or $\beta_{6,j}$ then the sums in (6) need only be computed over i, j such that $2 \leq i \leq 5$ and $4 \leq j \leq 7$. The B-spline values and their derivatives are computed using the de Boor routines BVALUE, BSPLVN, BSPLVD, and INTERV [18].

Even with the smaller sums a significant amount of time can be wasted during the execution of the minimization routine if the code can not take advantage of the optimization features of the computer being used. At the present time the code is running on a Cray Y-MP4E/232 computer system having two central processing units (CPUs). The small support of tensor product B-splines facilitates the design of an algorithm that takes advantage of the vectorization features of the computer. One can easily show that the support of a tensor product B-spline B_{ij} will be disjoint, except possibly on the support boundary, from the support of any tensor product B-spline B_{pq} where either $|p-i| \geq 4$ or $|q-j| \geq 4$. This is illustrated in Figure 9 for $B_{6,j}$. This means that the minimization code can be designed so that the smoothing functional is minimized with respect to the coefficients associated with those tensor product B-splines having disjoint support simultaneously. The sums must be restricted so that they are computed only over those points that fall inside the support area, but this does not appear to decrease the effectiveness of the minimization routine. Rewriting the code to minimize a vector of coefficients simultaneously produced a speedup by a factor close to 4.5. By integrating small, frequently called routines into the calling routines (inlining) and eliminating redundant code, a larger speedup was obtained. For example, the original scalar code eliminated the overlap of grid lines, that is, areas with

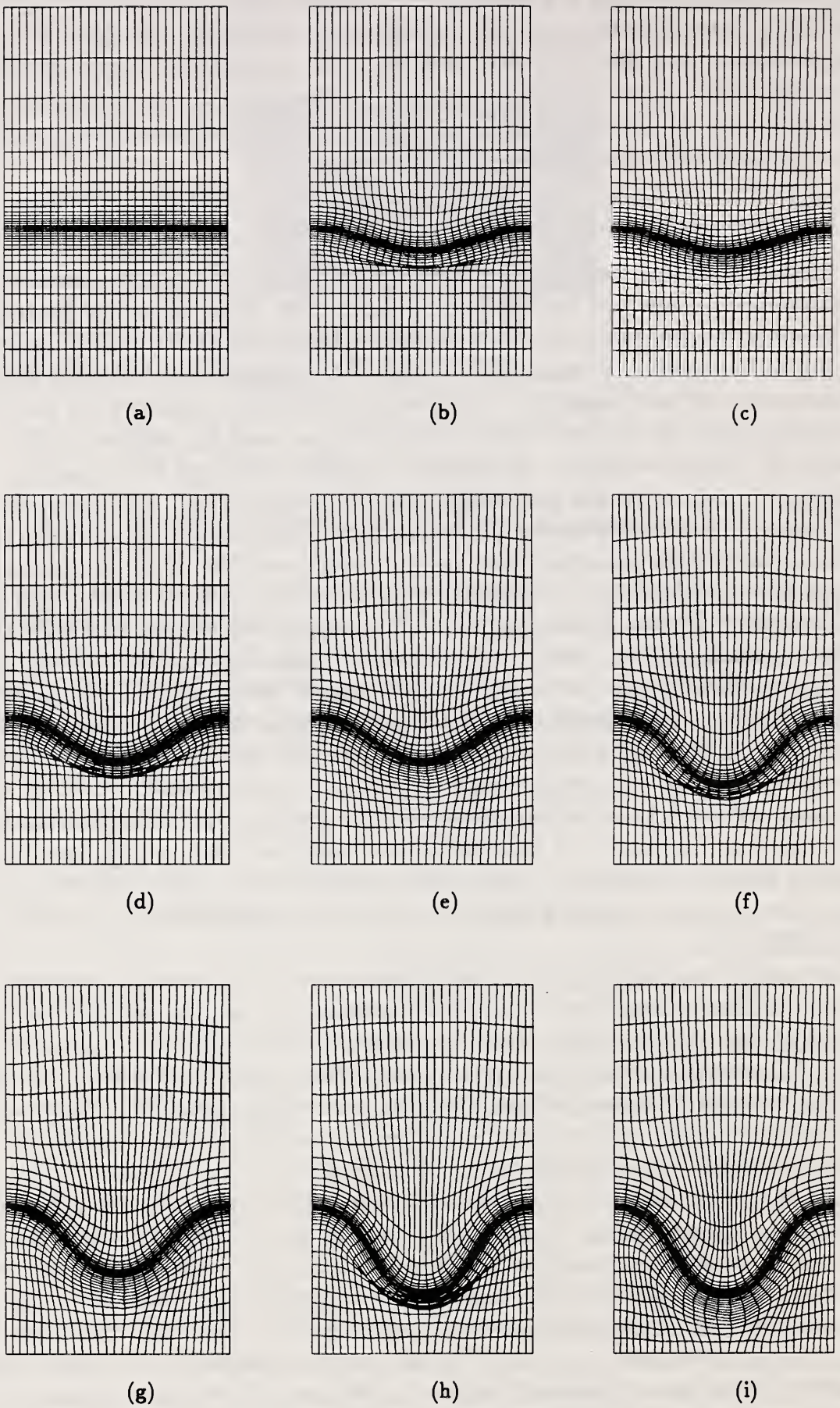


Figure 7: Grids simulating the gradual deformation of an interface.

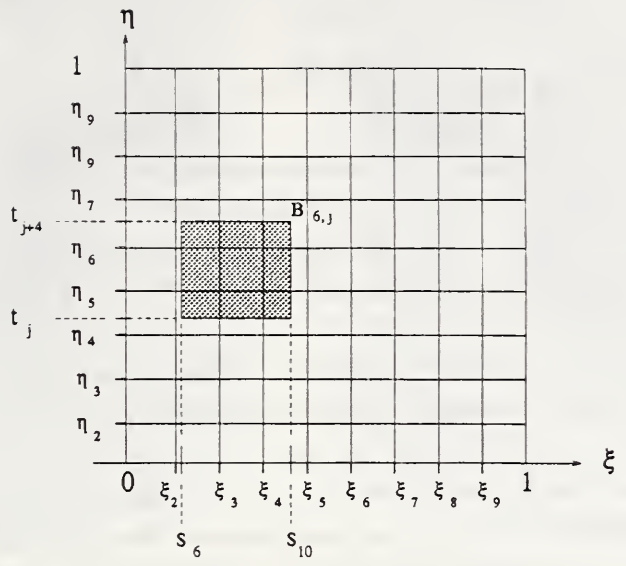


Figure 8: Support of tensor product B-spline $B_{6,j}$.

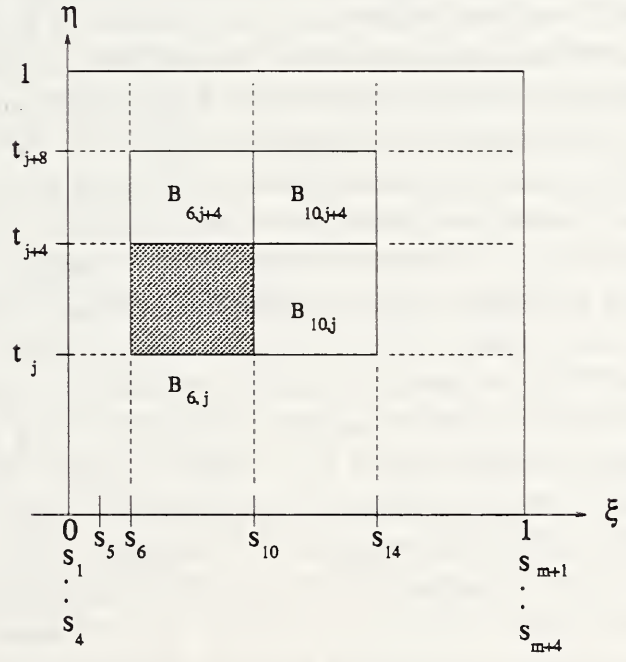


Figure 9: Tensor product B-splines with disjoint support. The shaded area shows the support of $B_{6,j}$. The surrounding boxes show other tensor product B-splines with disjoint support.

Table 1: User CPU Time (seconds)

Iterations	Scalar	Vectorized	Vectorized+
5	216.9	48.6	13.3
10	442.36	97.04	27.05

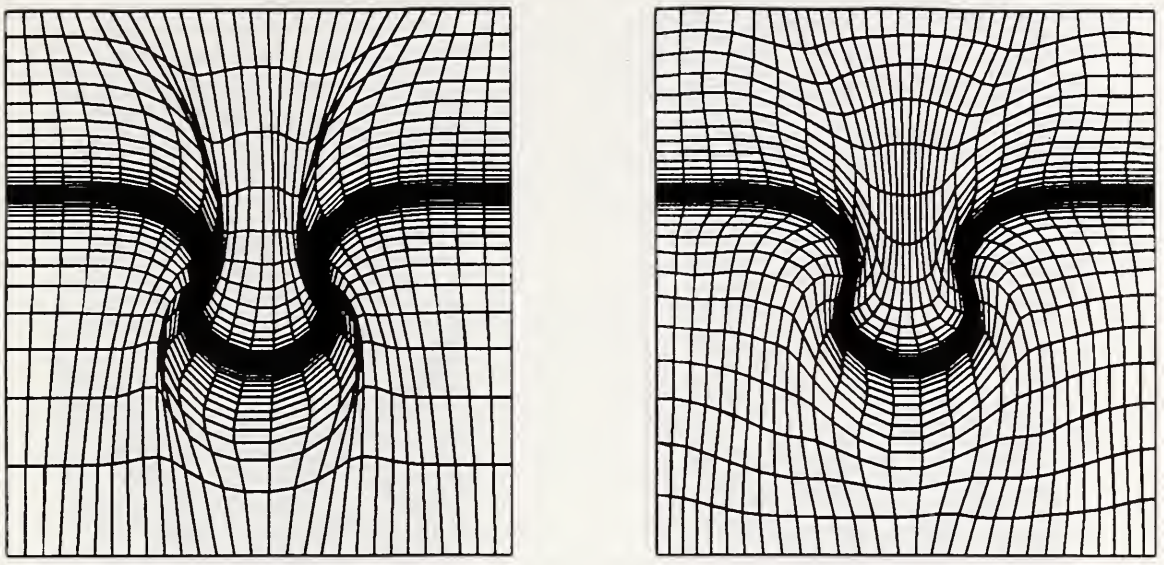
negative Jacobian, by determining the appropriate interval that would guarantee a positive Jacobian for each coefficient. However, this is unnecessary since the Jacobian terms of the smoothing functional tend to perform the same function by untangling grid lines to minimize the difference in Jacobians at nearby grid points.

Table 1 shows the user CPU times for a problem of moderate size. For this problem, $m = 18$ and $n = 24$ so that the number of spline coefficients is $2 \times 18 \times 24 = 864$. The minimization was performed on a 60×50 mesh. Since the system contains only two processors, very little effort was focused on parallelizing the code. The automatic parallelization option of the compiler was tested, but it produced only a minor speedup in the code. The table shows the single processor times for five and ten iterations of the minimization routine for three cases: the scalar code, the initial vectorized version where the code was rewritten to minimize with respect to a vector of coefficients (vectorized), and the improved vectorized version in which some routines were inlined and others eliminated (vectorized+). The final vectorized run is over 16 times faster than the scalar run.

4. Results and Discussion

For all the examples shown, the number of spline coefficients used to define the grid generation mapping is $2 \times m \times n$ where m is the number of B-splines associated with knot sequence $\{s\}$, or the "ξ" sequence, and n is the number associated with knot sequence $\{t\}$, the "η" sequence. Recall that a concentration of grid points near the interface (coordinate curve $\eta = 1/2$) is obtained by evaluating $\mathbf{T}$ at $(\xi, \gamma(\eta))$. To increase the accuracy of the spline mapping and decrease the area affected by the interface coefficients, the sinh function is also used to concentrate the η knots near the interface. This means that the sequence $\{\gamma(t)\}$ is used instead of $\{t\}$ in the definition of the mapping. The constant values of c associated with these two types of concentration are called the mesh concentration and the knot concentration, respectively. The orthogonality constant K represents the magnitude of the normal vectors at the interface as defined in section 3.1. It should also be noted that the term "initial grid" is used to refer to any grid computed using the initial spline coefficients.

In the first example the bottom of the puzzle boundary displayed in Figure 2 is now shown as an interface. This is similar to the re-entrant shape that commonly appears in cellular microstructures. For this example $m = 19$ and $n = 20$. The orthogonality constant K is 6 and the knot concentration is 3. The variation diminishing spline approximation to $\tilde{\mathbf{T}}$ produces the mesh shown on the left in Figure 10. Although the grid cells are skewed in some areas, the grid appears orthogonal near the interface. Hence, the initial grid already looks fairly good there. This is important since most of the interface coefficients remain fixed throughout the minimization process. Using a 40×42 initial grid with mesh concentration 3, the optimization routine significantly improves the smoothness and general orthogonality of the grid cells. This is illustrated in the grid on the right, computed after forty iterations. The Jacobian and orthogonality weights used for the smoothing functional, w_1 and w_2, were 0.65 and 10, respectively.



(a) Initial grid. (b) Optimized grid.

Figure 10: Grids produced using linear and hermite quintic blending functions. Grid on right was produced after forty iterations of the smoothing routine.

To illustrate the flexibility of the grid generation routine, the next six grids show grids that fit curves that are typical of the shapes seen during deformation of an interface from a sinusoidal shape to a deep cell. These grids actually continue the sequence that was started in Figure 7. The first grid in Figure 11 is the final grid in Figure 7. The grids show the meshes obtained after the smoothing functional is minimized. The initial grids, most of which contained overlapping grid lines, are not shown. For all of the grids $m = n = 18$ and the knot concentration is 3. The minimization was performed on a 40×40 mesh using a mesh concentration of 3. An orthogonality constant of $K = 4$ was used for the grids in Figure 11 and Figure 12a. This was increased to 4.5 for Figure 12b and to 5 for the grids in Figure 13 to maintain the smoothness and orthogonality near the interface as the interface cell deepened. In every case the smoothing routine successfully removed overlapping grid lines. The number of iterations of the minimization routine executed varied from 5 or 10 for the mildly deformed shapes of Figures 7 and 11 to around 30 for the deeper and re-entrant grooves in Figure 13. The orthogonality weight for the smoothing functional, w_2, was kept at 10 for all grids. The Jacobian weight, w_1, ranged between .01 and .2 for the grids in Figure 7, but was set to .5 for the deeper deformations shown in Figures 11b, 12, and 13. The grids were able to maintain a significant amount of orthogonality and smoothness both within the interior and along the boundary as the grid points redistributed themselves to follow the interface.

5. Conclusions

The development of an algebraic grid generation system to track a solid-liquid interface has been discussed. The proposed grid generation mapping, composed of tensor product B-splines, effectively approximates interface shapes of varying degrees of complexity. For each shape the initial coefficients are chosen to approximate a transfinite blending function

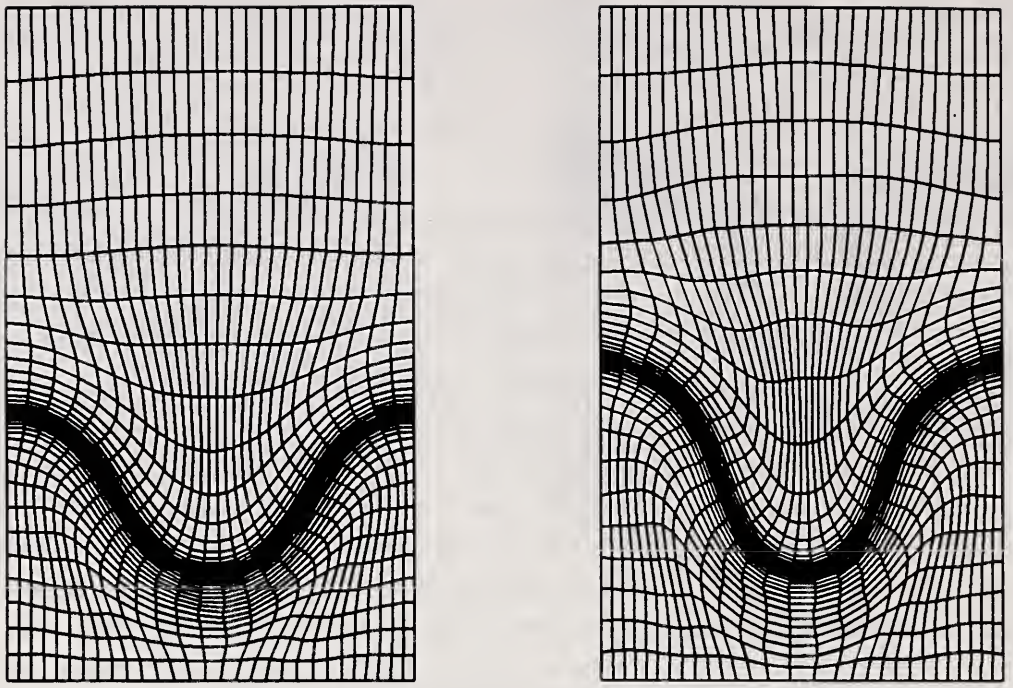


Figure 11: Grids show the mildly deformed sinusoidal shapes commonly observed during the initial stages of the deformation of a solid-liquid interface.

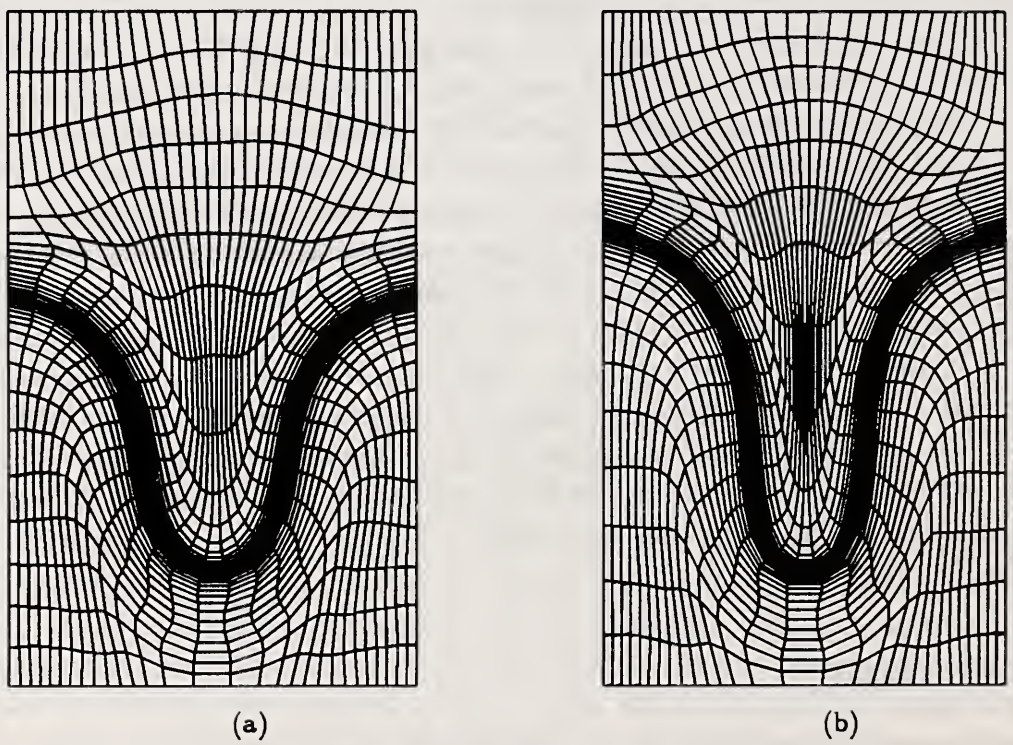


Figure 12: Grids show deeper grooves, but orthogonality is maintained at the interface.

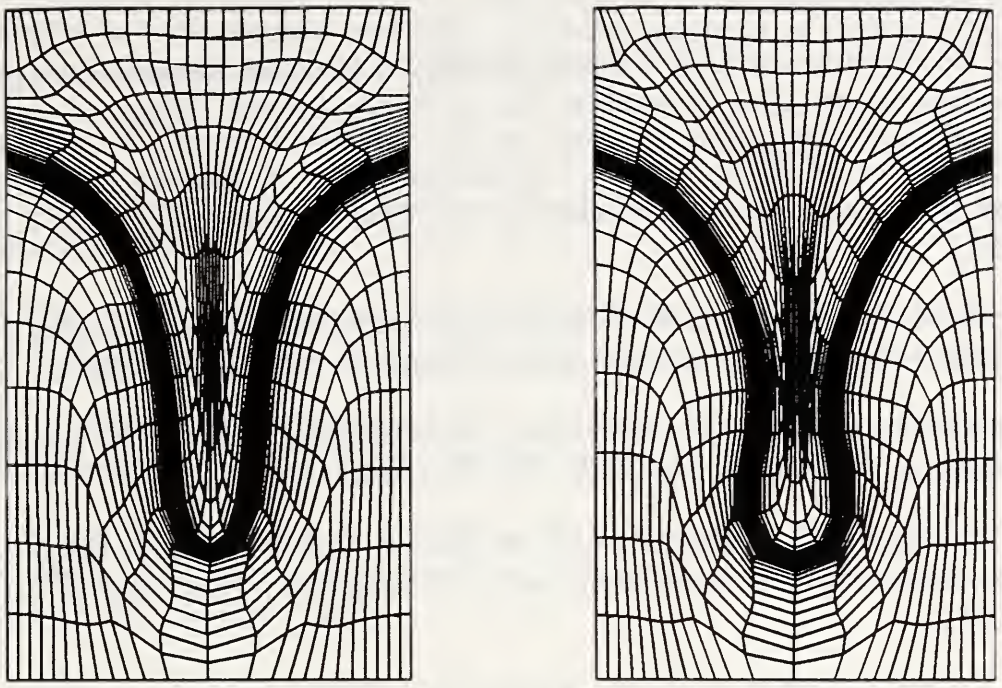


Figure 13: Grid on left shows deep and narrow groove. Grid on right shows deep re-entrant bulb-like cell.

mapping that produces nearly orthogonal grid lines near the interface. Adjusting the coefficients to minimize a discrete smoothing functional smoothes the spacing of the grid lines, untangles overlapping lines, and enhances the orthogonality of the initial grid even as the interface becomes more distorted. Smoothness and orthogonality is improved along the outer boundary by adjusting those boundary coefficients whose movement will not alter the shape of the boundary.

Currently, the grid generation system allows some interaction as the interface changes so that orthogonality and smoothness parameters can be adjusted as the interface cell deepens. Ideally, this should all be done automatically by the code. Additional study will be continued in this area. The next phase, which has already begun, is to couple the grid generation algorithm with equations that determine the interface shape so that the system can be used in the numerical analysis of microstructures that develop during directional solidification. Although the system will be designed to track the deformation of a planar interface into a deep cell, it may also be used to improve calculations in phase field models [20, 21, 22] where the interface is not viewed as a curve or surface with zero thickness. In such models accuracy can be improved by concentrating the grid points in the area near the interface even if the interface is not tracked exactly.

Disclaimer

Identification of commercial products in this paper does not imply recommendation or endorsement by NIST.

References

[1] W. W. Mullins and R. F. Sekerka, Stability of a planar interface during solidification of a dilute binary alloy, *J. Applied Physics* **35**(2), 444-451 (1964).

[2] S. R. Coriell, M. R. Cordes, W. J. Boettinger and R. F. Sekerka, Convective and interfacial instabilities during unidirectional solidification of a binary alloy, *J. Crystal Growth* **49**(1), 13-28 (1980).

[3] G. B. McFadden and S. R. Coriell, Nonplanar interface morphologies during unidirectional solidification of a binary alloy, *Physica D* **12**, 253-261 (1984).

[4] H. M. Ettouney and R. A. Brown, Finite-element methods for steady solidification problems, *J. Comput. Phys.* **49**, 118-150 (1983).

[5] L. H. Ungar, M. J. Bennett and R. A. Brown, Cellular interface morphologies in directional solidification. IV. The formation of deep cells, *Phys. Rev. B* **31**(9), 5931-5940 (1985).

[6] K. Tsiveriotis and R. A. Brown, Boundary-conforming mapping applied to computations of highly deformed solidification interfaces, *Int. J. Numer. Methods Fluids* **14**, 981-1003 (1992).

[7] V. Seetharaman, M. A. Eshelman and R. Trivedi, Cellular spacings-II. Dynamical studies, *Acta Met.* **36**, 1175-1185 (1988).

[8] W. Kurz and D. J. Fisher, *Fundamentals of Solidification* , Trans Tech, Switzerland (1989).

[9] J. U. Brackbill and J. S. Saltzman, Adaptive zoning for singular problems in two dimensions, *J. Comput. Phys.* **46**, 342-368 (1982).

[10] D. P. Woodruff, *The Solid-Liquid Interface*, Cambridge University, London (1973).

[11] B. V. Saunders, Algebraic grid generation using tensor product B-splines, NASA CR-177968 (1985).

[12] R. E. Smith, Two-boundary grid generation for the solution of the three-dimensional compressible navier-stokes equations, NASA TM-83123 (1981).

[13] P. R. Eiseman and R. E. Smith, Mesh generation using algebraic techniques, *Numerical Grid Generation Techniques* , NASA CP-2166 (1980).

[14] W. J. Gordon and C. A. Hall, Construction of curvilinear co-ordinate systems and applications to mesh generation, *Int. J. Numer. Methods Eng.* **7**, 461-477 (1973).

[15] C. R. Forsey, M. G. Edwards and M. P.Carr, An investigation into grid patching techniques, *Numerical Grid Generation Techniques*, NASA CP-2166 (1980).

[16] J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, Boundary-fitted coordinate systems for numerical solution of partial differential equations: a review, *J. of Comput. Phys.* **47**, 1-108 (1982).

[17] D. F. Hawken, J. J. Gottlieb and J. S. Hansen, Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations, *J. of Comput. Phys.* **95**, 254-302 (1991).

[18] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York (1978).

[19] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, Massachusetts (1984).

[20] A. A. Wheeler, W. J. Boettinger, and G. B. McFadden, Phase-field model for isothermal phase transitions in binary alloys, *Physical Review A* **45**(10), 7424-7439 (1992).

[21] A. A. Wheeler, B. T. Murray and R. J. Schaefer, Computation of dendrites using a phase field model, *Physica D* **66**, 243-262 (1993).

[22] S. -L. Wang, R. F. Sekerka, A. A. Wheeler, B. T. Murray, S. R. Coriell, R. J. Braun and G. B. McFadden, Thermodynamically-consistent phase-field models for solidification, *Physica D* **69**, 189-200 (1993).