Applied and
Computational
Mathematics
Division

NISTIR 5239

Computing and Applied Mathematics Laboratory

Boundary/Interface Fitted Grid Generation Using Tensor Product B-splines: A Preliminary Study

B.V. Saunders

August 1993

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

NISTIR 5239

Boundary/Interface Fitted Grid Generation Using Tensor Product B-splines: A Preliminary Study

B. V. Saunders

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

August 1993

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

Boundary/Interface Fitted Grid Generation Using Tensor Product B-splines: A Preliminary Study

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

Abstract

Progress in the development of an algebraic grid generation system that tracks 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

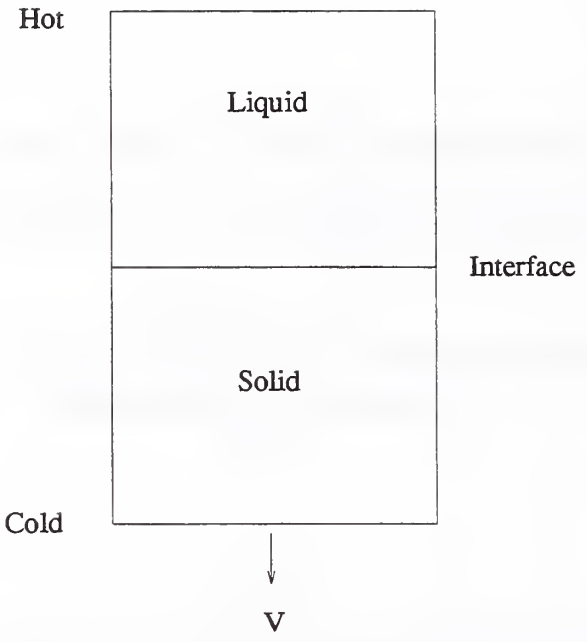

Figure 1: Bridgman growth technique.

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 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 their 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 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 algebraic techniques for generating coordinate systems, focusing on transfinite blending function interpolation, the technique used to obtain an initial approximation for the boundary/interface fitted mesh. 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].

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

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

$$- \sum_{i,j} \Phi_i(\xi)\Psi_j(\eta)\mathbf{f}(\xi_i,\eta_j) \tag{1}$$

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].

Saunders [11] developed a boundary fitted grid generation mapping composed of tensor product B-splines whose coefficients are originally chosen so that the mapping approximates transfinite blending function interpolation. An application of the technique 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. The grid generation system iterates on the coefficients to minimize a functional designed to increase the smoothness and orthogonality of the initial mesh. The grid on the left in Figure 3 shows the mesh obtained using the minimized coefficients. The skewed areas have been eliminated and the overlapping grid lines have been pulled into the interior. The grid on the right, computed with the same coefficients, shows a refined mesh concentrated near the bottom boundary using an exponential function. One dimensional concentrations of this type are easy to obtain by replacing the mapping $\mathbf{T}(\xi,\eta)$ with $\mathbf{T}(\bar{\xi}(\xi),\bar{\eta}(\eta))$ where $\bar{\xi}$ and $\bar{\eta}$ are appropriate spacing control functions in ξ and η [11]. The algorithm discussed in this paper is an extension of the boundary fitted mapping to interpolate 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.

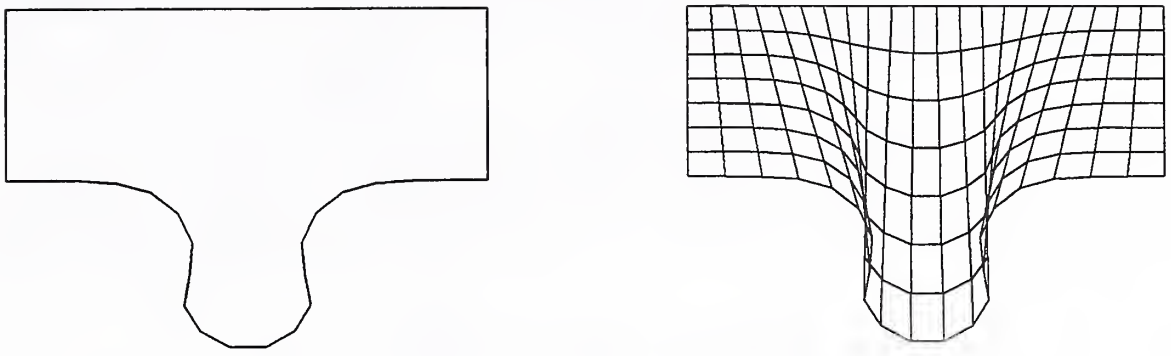

Figure 2: Puzzle shaped domain. Grid on right was produced using an approximation of transfinite blending function interpolation.

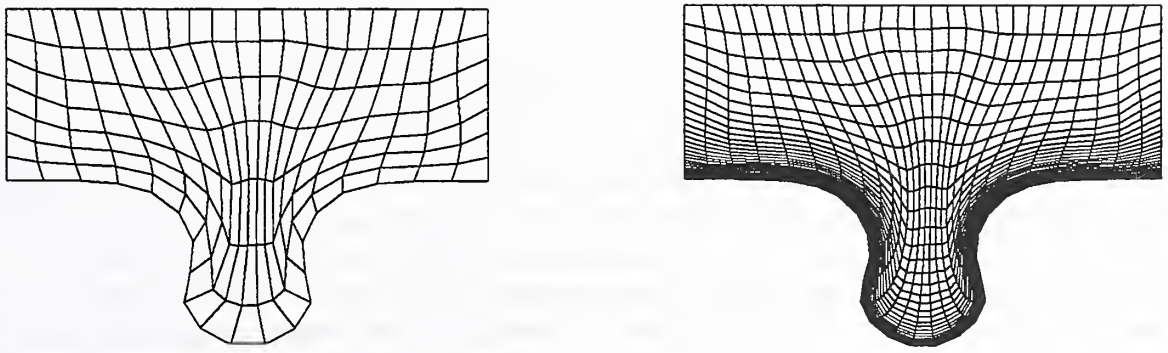

Figure 3: Optimized puzzle grids.

3. 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. The mapping is defined by

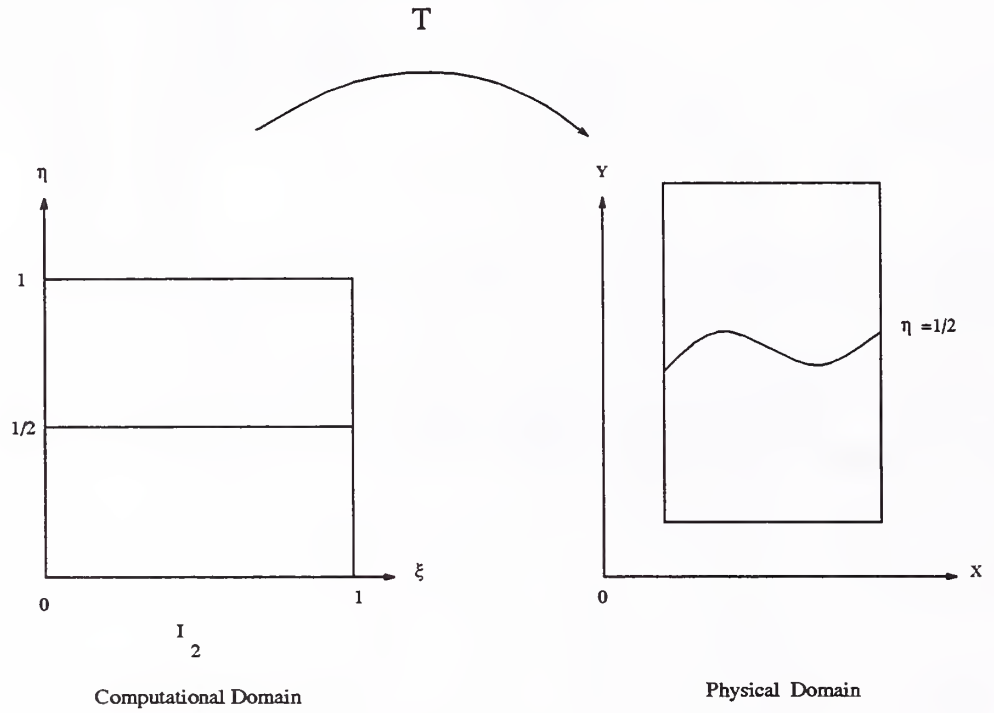

Figure 4: Grid generation mapping.

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

where $0 \le \xi, \eta \le 1$ 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 spline coefficients for $\mathbf{T}$ can be divided into three groups. The boundary coefficients are the coefficients of the 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. To obtain an initial grid, the spline coefficients are chosen to produce a variation diminishing spline approximation to the transfinite blending function interpolant that matches the boundary and interface of the physical domain. In short, this means the coefficients are obtained by evaluating the transfinite mapping at average knot values as discussed in [18]. This shape preserving approximation reproduces straight lines and preserves convexity [11, 18].

If the deformation of the interface is only moderate, then the initial grid may be suitable; but as the interface becomes more distorted, additional smoothing and orthogonality are needed. To increase orthogonality and decrease the overlapping of grid lines and skewness, the interior coefficients of $\mathbf{T}$ are modified to minimize a discrete approximation to the following functional first described in [11]:

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

where $\mathbf{T}$ denotes the grid generation mapping, J is the Jacobian of the mapping, and $w_1(\xi, \eta)$ and $w_2(\xi, \eta)$ are weight functions. 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.

Clearly, the flexibility of the boundary and the interface coefficients is limited. 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 $(0, 1)$, 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 $i = 1, \ldots, m$. Furthermore, 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}, 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 coefficients will be a narrow region along the boundary of the square. Consequently, even if the boundary coefficients remain fixed, the smoothing functional should be able to produce a significant amount of orthogonality and smoothness in the grid. Ideally, of course, the boundary coefficients should also be adjusted by the smoothing functional, but the boundary fittedness requirement severely restricts their movement. For a general boundary of arbitrary shape, small changes in the boundary coefficients can destroy the shape preserving properties of the spline mapping. In such a case, fixing the boundary coefficients is probably the simplest solution. However, if the boundary is rectangular, as it is for Bridgman growth, more flexibility is possible. Using the smoothing functional, the boundary coefficients can be modified so that a reparametrization of the boundary mapping results.

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

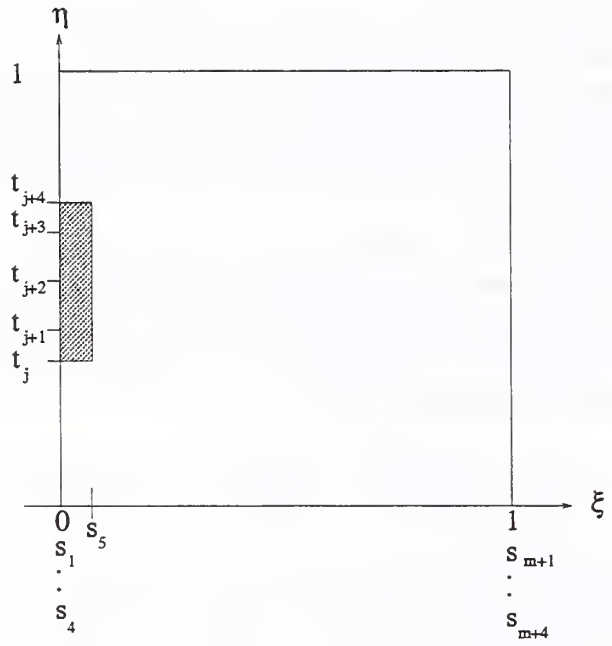

Figure 5: Support of tensor product B-spline B_{1j}.

interface coefficients will be α_{ik}, β_{ik} where $1 \leq i \leq m$ and $l-3 \leq k \leq l$. 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 the associated tensor product B-splines. 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 have an equal effect on the interface mapping. In particular, allowing the coefficients $\alpha_{i,l-3}$ and $\beta_{i,l-3}$ for $1 \leq i \leq m$ to move appears to have very little effect on the accuracy of the interface mapping. Nevertheless, at the outset, the interface coefficients should be chosen to produce as much orthogonality and smoothness as possible. As expected, the results in the next section indicate that the type of blending functions used in defining the initial coefficients is very important.

4. Results

For all the examples shown, constants of values 1 and 10 are used for the Jacobian and orthogonality weights, respectively, in the smoothing functional. As in [11], a simple minimization technique, the cyclic coordinate method, is used to minimize a discrete approximation to the functional. The number of spline coefficients used to define the grid generation mapping is $2 \times N_\xi \times N_\eta$ where N_ξ is the number of B-splines associated with sequence $\{s\}$, or the

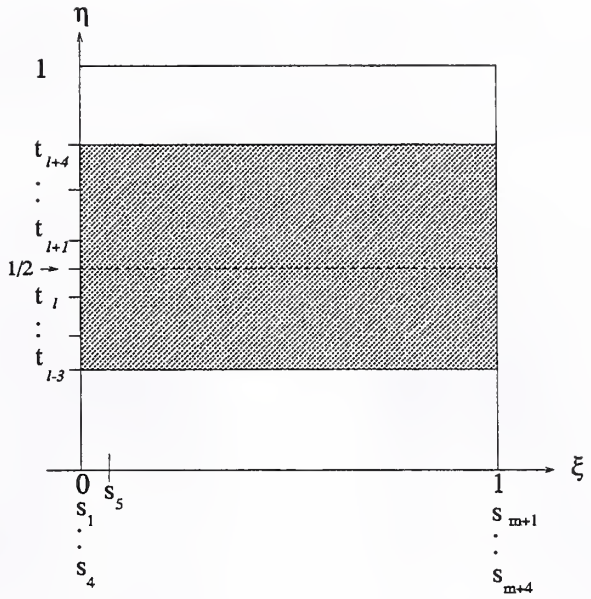

Figure 6: Support of tensor product B-splines nonzero when $\eta = 1/2$.

"ξ" sequence, and N_η is the number associated with sequence $\{t\}$, the "η" sequence. A concentration of grid points near the interface (coordinate curve $\eta = 1/2$) is obtained by replacing η with the hyperbolic sine function

$$\bar{\eta}(\eta) = \frac{\sinh(2c\eta - c) - \sinh(-c)}{\sinh(c) - \sinh(-c)}$$

where c is a constant that determines the degree of concentration. To increase the accuracy of the spline mapping and decrease the area affected by the interface coefficients, illustrated in Figure 6, the sinh function is also used to concentrate the η knots near the interface. This means that the sequence $\{\bar{\eta}(t)\}$ is used instead of $\{t\}$ in the definition of the mapping. The values of c associated with these two types of concentration will be called the mesh concentration and the knot concentration, respectively. It should also be noted that the term "initial grid" is used to refer to any grid computed using the initial spline coefficients.

The first example illustrates the difficulty in extending the algorithm in [11] to develop a grid that interpolates a solid-liquid interface. 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. $N_\xi = 19$ and $N_\eta = 20$ for this example. To obtain the transfinite approximation, the physical domain is divided into top and bottom halves separated by the interface. In each section linear Lagrange polynomials are used as blending functions to create a mapping that interpolates the boundary and interface. The initial coefficients are chosen to create a variation diminishing spline approximation to the transfinite mapping. A knot concentration of 3 is used to define the spline approximation. In Figure 7 the mesh on the left shows a grid computed with the initial coefficients. As with the original puzzle grid shown in Figure 2, one can see that the grid cells near the re-entrant

curve are skewed and overlap each other. Attempts to improve the grid by minimizing the smoothing functional were unsuccessful. The grid on the right was computed using coefficients obtained after twenty iterations of the cyclic coordinate minimization routine on a 35×60 initial grid with mesh concentration equal 3.

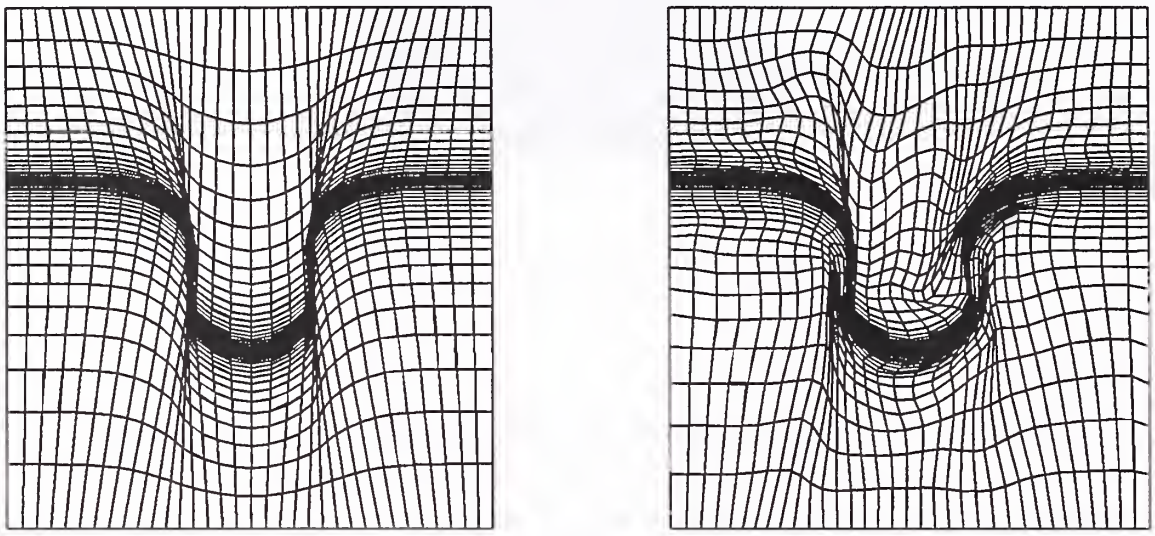

Figure 7: Grids produced using bilinear blending functions. Initial grid is on the left. Grid on right shows smoothing routine unable to improve grid.

The use of higher degree polynomials for the blending functions permits the creation of mappings that not only interpolate the boundary and interface but also match derivatives there. For example, using cubic blending functions in the η coordinate, one can interpolate the top and bottom boundaries, the interface, and the normal vectors at the interface. However, even when orthogonality is obtained at the interface, polynomials of high degree may produce a significant amount of skewness and overlap of grid cells elsewhere in the mesh. Among the polynomials tested: linear, quadratic, cubic, and hermite quintic (fifth degree), the hermite quintic polynomials produced the least amount of skewness and overlap. For all the grids that follow, hermite quintic blending functions were used for the η coordinate and linear Lagrange polynomials for the ξ coordinate. Unlike the previous example, the same transfinite mapping is used over the entire domain. Boundary data input by the user is used to define a continuous vector valued function $\mathbf{f}$ that maps the boundary of the square to the boundary of the physical domain and the line $\eta = 1/2$ to the interface. The transfinite mapping $\tilde{\mathbf{T}}$ is constructed to interpolate $\mathbf{f}$ on the boundary and match $\mathbf{f}$ and $\partial\mathbf{f}/\partial\eta$ at $\eta = 1/2$. It has the form

$$\tilde{\mathbf{T}}(\xi,\eta) \;=\; \sum_{i=0}^{1} \Phi_i(\xi)\mathbf{f}(\xi_i,\eta)$$

$$
\begin{aligned}
&+ \sum_{j=0}^{2} \Psi_j(\eta)\mathbf{f}(\xi,\eta_j) \\
&+ \beta(\eta)\frac{\partial \mathbf{f}}{\partial \eta}(\xi,1/2) \\
&- \sum_{i=0}^{1}\sum_{j=0}^{2} \Phi_i(\xi)\Psi_j(\eta)\mathbf{f}(\xi_i,\eta_j) \\
&- \sum_{i=0}^{1} \Phi_i(\xi)\beta(\eta)\frac{\partial \mathbf{f}}{\partial \eta}(\xi_i,1/2) \qquad (6)
\end{aligned}
$$

where $\xi_0 = 0$, $\xi_1 = 1$ and $\eta_0 = 0$, $\eta_1 = 1/2$, $\eta_2 = 1$. The linear blending functions Φ_i satisfy

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

while the hermite quintic blending functions Ψ_r and β satisfy

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

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 variation diminishing spline approximation to $\tilde{\mathbf{T}}$ produces the mesh shown on the left in Figure 8. As in the first example, the knot concentration equals 3, $N_\xi = 19$ and $N_\eta = 20$. The orthogonality constant K is 3. Although the grid cells are skewed in some areas, the grid is more orthogonal near the interface than the previous example. 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

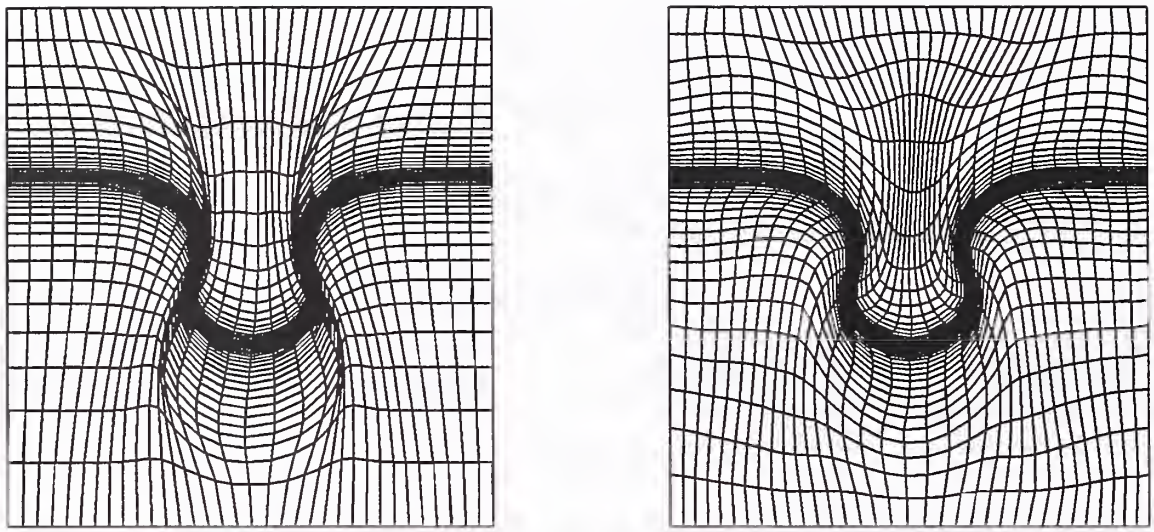

Figure 8: Grids produced using linear and hermite quintic blending functions. Initial grid is on the left. Grid on right was produced after twenty iterations of smoothing routine.

35×60 initial grid with mesh concentration 3, the optimization routine easily improves the smoothness and general orthogonality of the grid cells. This is illustrated in the grid on the right, computed after twenty iterations.

To illustrate the flexibility of the grid generation routine, the next six grids show a typical deformation of an interface from a planar shape to a deep cell. The grids were constructed independently of each other. That is, each grid was created by first starting with boundary data and interface data for the specific interface shape. An initial grid was formed from the spline approximation to the transfinite mapping using a knot concentration of 3. For all the grids, $K = 2$, $N_\xi = 14$ and $N_\eta = 24$. The $2 \times N_\xi \times N_\eta$ spline coefficients were then iterated on to minimize the smoothing functional. Except for the planar grid which required no iterations, the grids shown were produced after twenty iterations using a 30×60 grid with mesh concentration 4. Although all the grids are relatively smooth, one can see that smoothness and orthogonality are harder to maintain as the cell deepens.

The final example shows a grid constructed for a deep re-entrant cell. Figures 12 and 13 show the mesh after 0, 1, 5, and 10 iterations, respectively. The orthogonality constant $K = 2$, $N_\xi = 18$, $N_\eta = 24$ and the knot concentration is 3. The minimization was done on a 60×50 grid with mesh concentration 4. Note that the initial grid overlaps itself several times in the area underneath the bulb-like bottom. The smoothing routine is able to untangle this area after only five iterations.

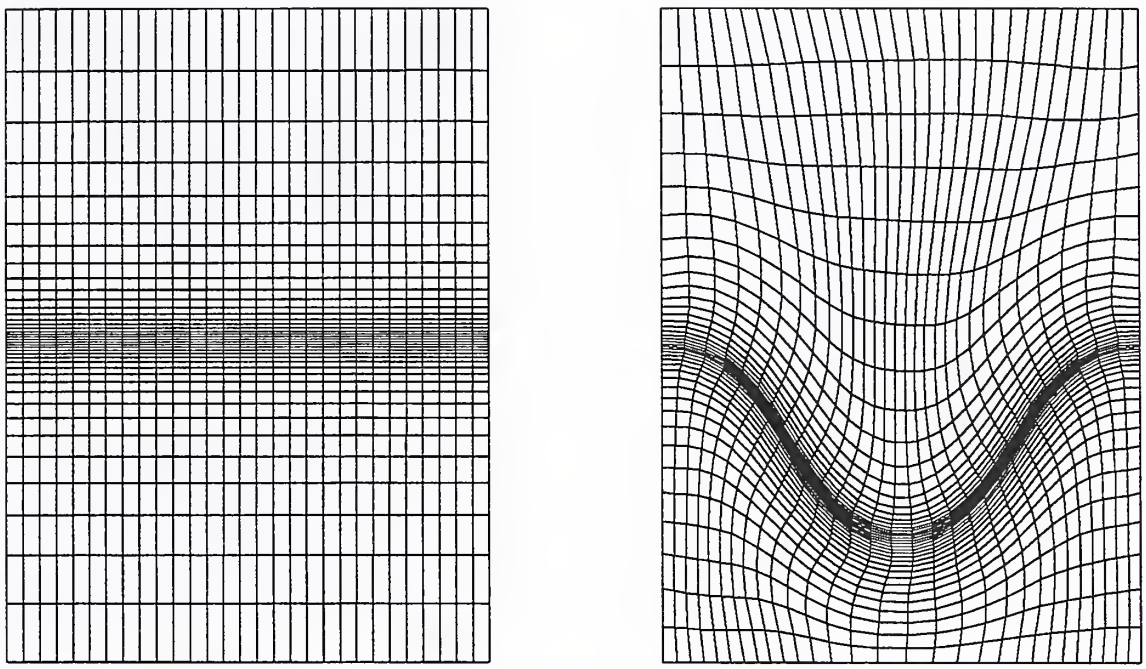

Figure 9: Deformation of a solid-liquid interface. Stages 0 and 1.

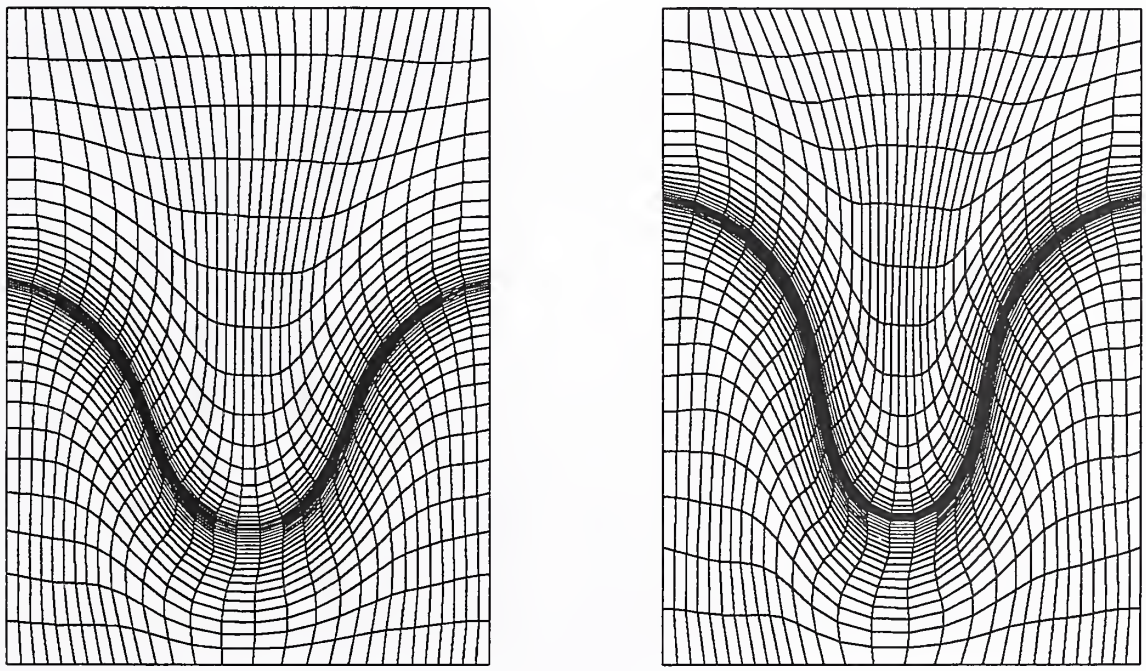

Figure 10: Deformation of a solid-liquid interface. Stages 2 and 3.

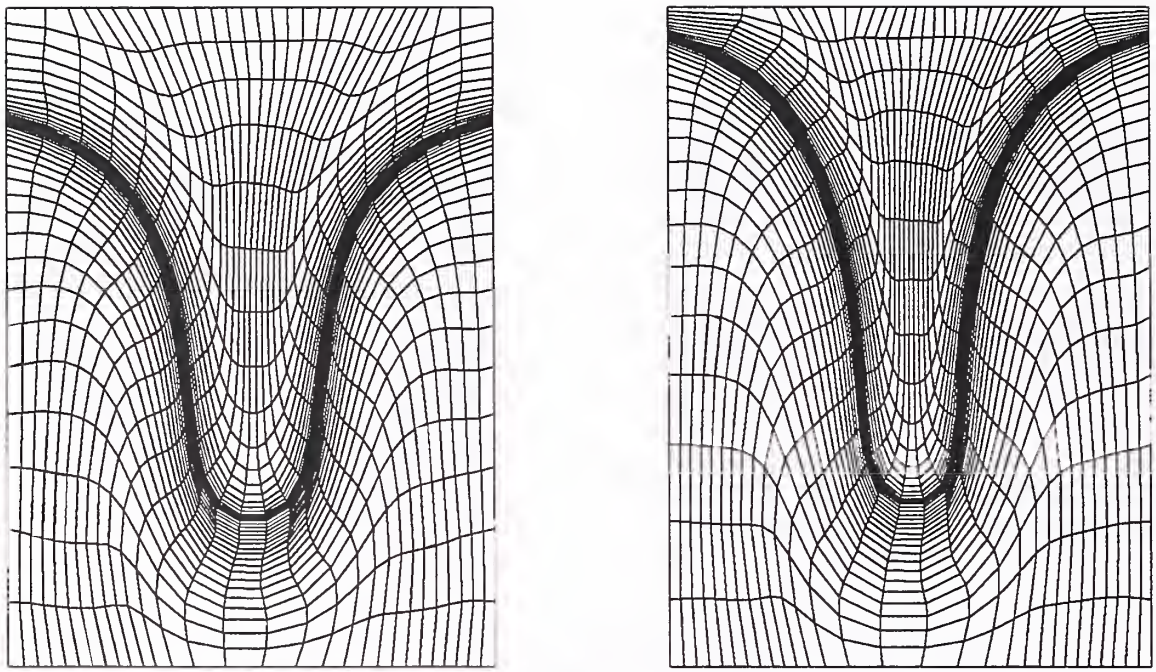

Figure 11: Deformation of a solid-liquid interface. Stages 4 and 5.

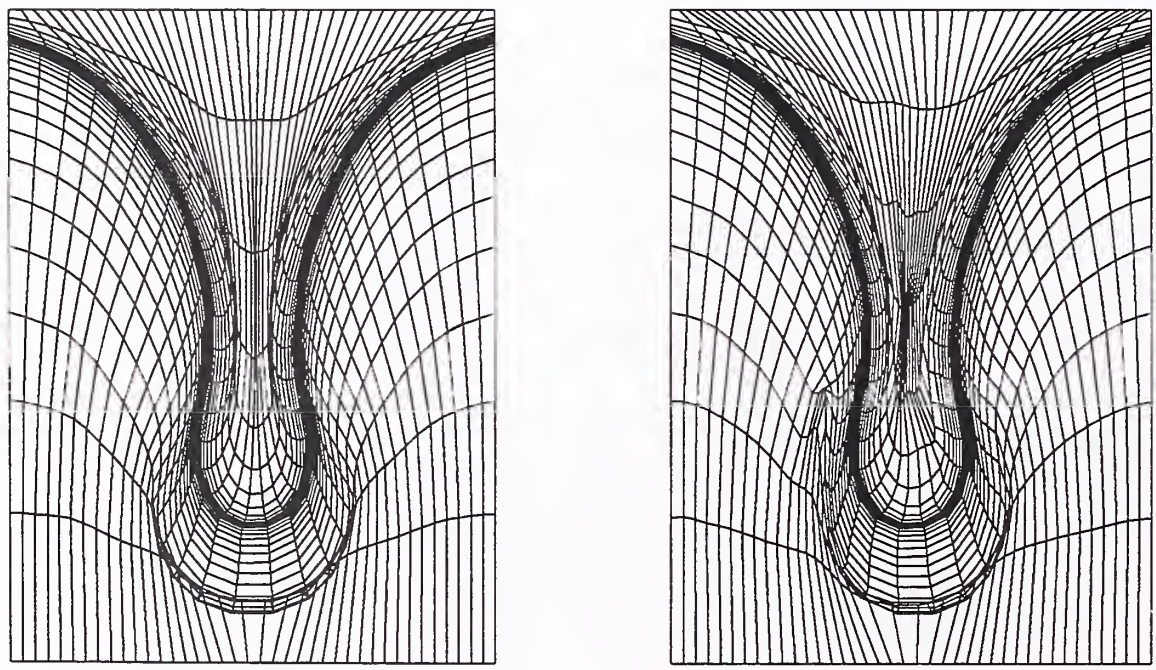

Figure 12: Re-entrant interface shape. Initial grid on left shows skewed and overlapping grid cells. Grid on right produced after 1 iteration of smoothing routine.

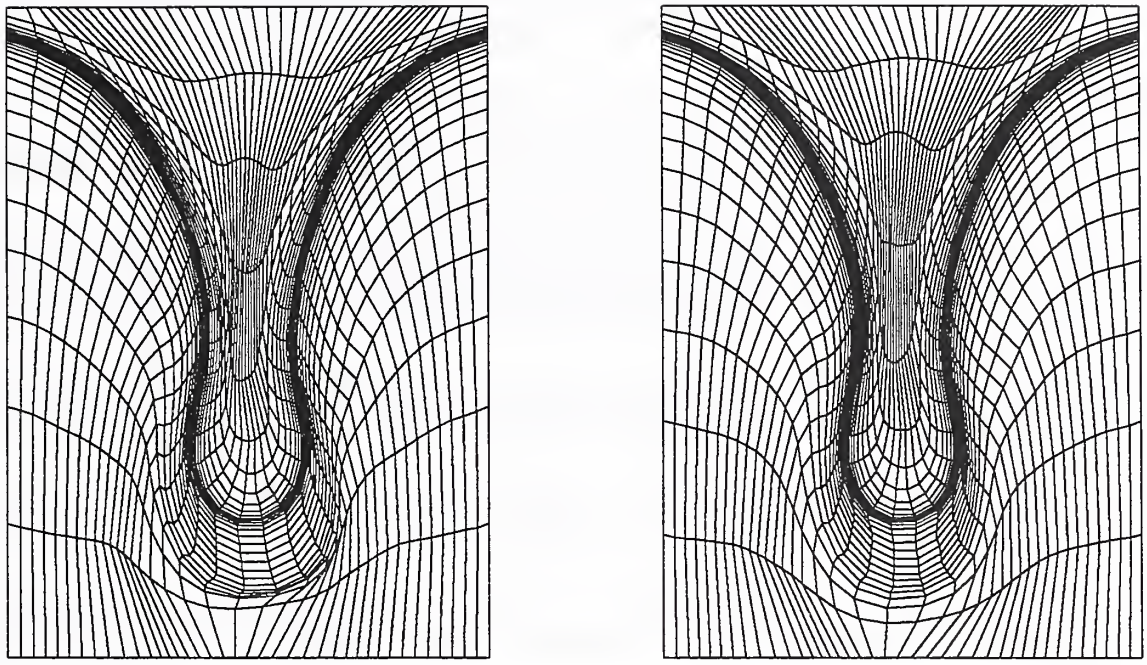

Figure 13: Re-entrant interface shape. Grid on left produced after 5 iterations of smoothing routine. Grid on right produced after 10 iterations.

5. Conclusions

Progress in the development of an algebraic grid generation system that tracks a solid-liquid interface has been discussed. The proposed grid generation mapping effectively interpolates interior curves of varying degrees of complexity from planar to deep re-entrant grooved shapes. Transfinite blending function interpolation appears to be a good method for creating an initial grid if the blending functions are chosen appropriately. Additional study will be continued in this area.

Currently, the grids are developed by starting with the boundary and interface data for a particular shape. Hence, the grid generation problem becomes quite challenging for deeply grooved and narrow re-entrant shapes. In an actual application, the initial grid will be the grid used in the previous iteration. Therefore, the interface change should be more gradual. Hopefully, starting with a previously developed grid will allow the system to generate meshes for interfaces that are more narrow and grooved. This is the current area of study.

The next phase will be 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 where the interface is not viewed as a curve or surface with zero thickness. Accuracy can be improved by concentrating the grid points in

the area near the interface even if the interface is not tracked exactly.

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**, 444 (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**, 13 (1980).

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

[4] H. M. Ettouney and R. A. Brown, "Finite-element methods for steady solidification problems," J. Comput. Phys., **49**, 118 (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**, 5931 (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 (1992).

[7] V. Seetharaman, M. A. Eshelman and R. Trivedi, "Cellular spacings-II. Dynamical studies," Acta Met. **36**, 1175 (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 (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 (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 (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 (1991).

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