NISTIR 6073

The Application of Numerical Grid Generation to Problems in Computational Fluid Dynamics

Bonita V. Saunders

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Gaithersburg, MD 20899-0001

NIST

NISTIR 6073

The Application of Numerical Grid Generation to Problems in Computational Fluid Dynamics

Bonita V. Saunders

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001

September 1997

U.S. DEPARTMENT OF COMMERCE
William M. Daley, Secretary

TECHNOLOGY ADMINISTRATION
Gary Bachula, Acting Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Robert E. Hebner, Acting Director

The Application of Numerical Grid Generation to Problems in Computational Fluid Dynamics

Bonita V. Saunders
National Institute of Standards and Technology
Gaithersburg, MD 20899
Email: saunders@cam.nist.gov

Abstract

Numerical grid generation, the computation of boundary fitted curvilinear coordinate systems to aid in the numerical solution of partial differential equations, is described. Grid generation plays a crucial role in resolving the problem of handling arbitrarily shaped boundaries when solving physical problems over a field. The driving impetus for the development of grid generation techniques was to solve problems in computational fluid dynamics, but grid generation is applicable to any area where partial differential equations are computed over a field. The use and benefits of grid generation are explained. Common types of grid generation systems are presented and finally, the generation of grids suitable for solving physical problems that arise in solidification theory is examined.

Keywords: numerical grid generation, adaptive grid generation, boundary-fitted grid generation, free/moving boundary problems, solidification modeling

Introduction

In the field of computational fluid dynamics, the complexity of the physical problem often means that any realistic mathematical model must be solved numerically on a computer. One of the most time consuming tasks can be determining and constructing the coordinate system for the computations. A common technique, called numerical grid generation, is to develop a general curvilinear coordinate system that maps the oddly shaped physical domain back to a simpler computational domain such as a square or rectangle. This technique is also known as boundary-fitted or boundary conforming grid generation because the boundary of the mesh, or grid, generated by the coordinate system coincides with the boundary of the physical domain as shown in Figure 1.

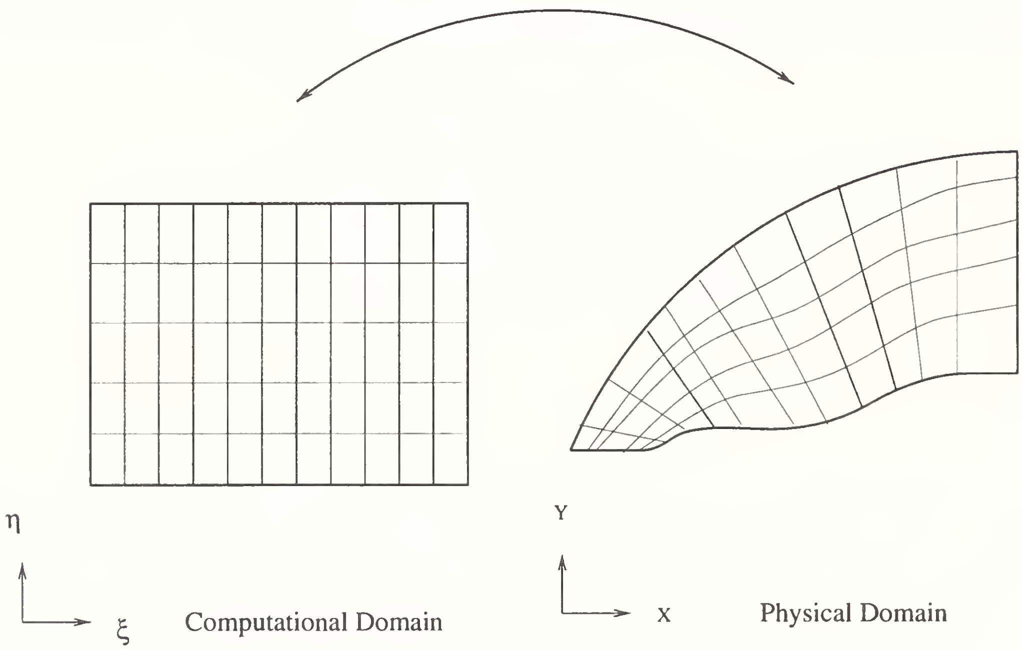


Figure 1: Boundary-fitted grid generation.

Numerical grid generation has been an active area of research for many years, but the bulk of the research has been conducted during the last twenty-five years [1-21]. A great deal of progress has been made, but there is still more work to be done. Although grids can now be made for most boundary configurations, in many cases, especially in three dimensions, the process is neither easy nor automatic. Slight changes in a configuration can cause a lot of additional work. Many engineers complain that in the development of flow solver codes, the generation of the mesh continues to be the most time consuming part of the calculation [3]. Furthermore, the interface that connects the grid generator to the flow solver code is often hard to use and too restrictive [4]. The need for adaptive codes continues to drive a lot of research. In the area of free and moving boundary problems, researchers continue to look for ways of developing grids that easily adapt to rapid and severe changes in a boundary without degrading the accuracy of the computations of the flow solver. These are just a few of the problems facing researchers in grid generation.

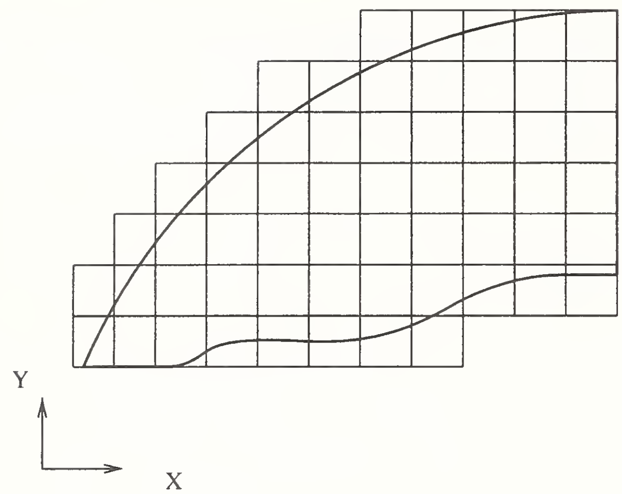
This paper looks at the motivation behind the development of grid generation systems, presents a brief introduction to the field, examines the use of grids, and discusses common types of grid generation systems. It also looks at the author's current interests in the field and in particular, examines the challenge of creating grids that model a special type of free/moving boundary problem that arises in the field of solidification theory.

The Use of Numerical Grid Generation

The mathematical modeling of many physical processes, such as airflow around a wing or fuselage, or fluid flow around a ship, involves the solving of partial differential equations over an oddly shaped field. This makes it difficult to apply numerical solution techniques without introducing undesirable errors into the calculations. Numerical grid generation permits the user to transform the oddly shaped domain to a simpler domain on which it is easier to compute. Numerical grid generation is actually the creation of a curvilinear coordinate system that connects a simpler computational domain, such as a square or rectangle, to the more complicated physical domain as was seen in Figure 1. It is commonly called boundary-fitted or boundary conforming grid generation because the boundary of the mesh, or grid, the system generates on the physical domain matches the physical boundary. Partial differential equations and boundary conditions originally defined on the physical domain are transformed to equations on the simpler domain. The new equations tend to look more complicated, but the natural structure of the computational domain simplifies the coding of finite difference or finite element equations. Boundary conditions are now simple to apply because in the computational domain the boundary points lie on the boundary of a square or rectangle.

The default method of simply placing a rectangular mesh on the physical domain complicates the computation of boundary conditions because the grid may overlap the boundary in some areas as shown in Figure 2. Since no mesh point lies directly on the boundary in such an area, interpolation is typically used to compute the boundary conditions there. This introduces errors into the numerical computations.

Another common technique for dealing with physical boundaries of arbitrary shape is to use triangulation methods. Many find these methods more adept at handling boundaries that have unusual shapes, but such methods generally require more memory because the user must store connectivity information, that is, information about how the mesh points relate to each other and which are neighbors of a given point. Also, even though triangulation methods may appear to fit a boundary well, the user may have trouble making the triangles small enough to get the resolution needed for accurate calculations near the boundaries. This is particularly true for viscous flow simulations needed to study flow around an airplane wing. A fine concentration of grid points primarily in the direction normal to the wing is needed to obtain accurate calculations of the boundary layer near the wing surface [5].



Physical Domain

Figure 2: Cartesian mesh over physical domain.

For all types of grid generation techniques a considerable amount of research continues to be devoted to the development of adaptive techniques in which grid points are either redistributed or added and deleted in response to what happens as the solution evolves [6]. Some methods capture gradient information and redistribute points in areas where the gradient is large. In the case of free and moving boundary problems, grid points are redistributed or added and deleted to follow the motion of the changing boundary.

Types of Grid Generation Systems

The most common types of grid generation systems are partial differential equation generated systems, algebraically generated systems, and systems generated by variational methods. Partial differential equation systems include conformal, elliptic, parabolic and hyperbolic. Such systems tend to produce smooth grids, but they introduce the complication of solving additional partial differential equations besides those governing the physical problem. The use of conformal mapping techniques is probably the oldest method for constructing coordinate systems. In conformal systems the curvilinear coordinates can be generated by solving Laplace's equation with the Cauchy Riemann equations as the boundary conditions. If ξ,η

are the curvilinear coordinates and x,y, the cartesian coordinates, then we have

$$\nabla^2\xi = \frac{\partial^2\xi}{\partial x^2} + \frac{\partial^2\xi}{\partial y^2} = 0 \tag{1}$$

$$\nabla^2\eta = \frac{\partial^2\eta}{\partial x^2} + \frac{\partial^2\eta}{\partial y^2} = 0 \tag{2}$$

with boundary conditions

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

$$\frac{\partial\xi}{\partial y} = -\frac{\partial\eta}{\partial x}. \tag{4}$$

The equations and boundary conditions are transformed to the computational domain and solved. The orthogonality of conformal systems helps minimize truncation error in finite difference calculations, but conformal systems permit little control over grid spacing if grid points need to be concentrated in certain areas to obtain accuracy. A conformal system is actually a special type of elliptic grid generation system.

Elliptic systems go back at least thirty years. Winslow [7] was one of the early users, but elliptic systems became popular in the 1970s and 1980s when they were reintroduced and improved by Joe Thompson and Wayne Mastin *et al* of Mississippi State University [2, 8]. Elliptic systems are generated from either the Laplace equations or the Poisson equations obtained by adding functions P and Q that control the spacing of the coordinate lines:

$$\nabla^2\xi = \frac{\partial^2\xi}{\partial x^2} + \frac{\partial^2\xi}{\partial y^2} = P \tag{5}$$

$$\nabla^2\eta = \frac{\partial^2\eta}{\partial x^2} + \frac{\partial^2\eta}{\partial y^2} = Q. \tag{6}$$

Elliptic systems produce very smooth grids, but choosing the appropriate grid control functions, determining the best way to match a complicated physical boundary with the computational boundary, and solving the elliptic system can be a time consuming and tricky process.

S. Nakamura used parabolic partial differential equations to generate coordinate systems [9] and Steger and colleagues [10] worked on hyperbolic grid generation systems. Such systems are generally faster than elliptic, but they are only applicable to physically unbounded regions [11].

Inherent in any grid generation system is an invertible mapping of the curvilinear coordinates onto the cartesian coordinates. In partial differential equation generated systems the mapping is not directly available, but in algebraic generation systems the mapping is given explicitly. Hence no partial differential equations need to be solved to generate the curvilinear coordinate system. One of the earliest and simplest algebraic generation techniques is transfinite, or blending function, interpolation. In this type of system the grid generation mapping is actually an interpolation function that is described in terms of functions specified on the boundary. More complex systems can be created by also specifying functions on selected interior curves. William Gordon *et al* did a substantial amount of work on these systems at General Motors during the 60s, 70s, and 80s [12, 13]. Others who have worked with algebraic systems are R.E. Smith [14], P.R. Eiseman [15], and B.V. Saunders [16]. Generally algebraic systems are faster grid generators, but the grids tend not to be as smooth. Singularities on the boundary may propagate into the interior of the grid. Some of the problems with algebraic systems can be lessened or eliminated by applying smoothing techniques. In a later section of this paper an algebraic system that generates grids using a mapping composed of tensor product B-splines is described. Variational techniques are used to smooth the grid.

Brackbill and Saltzman [17] and Steinberg and Roache [18] popularized variational grid generation methods. Grids are generated by solving the Euler-Lagrange equations derived by minimizing three integrals that control grid smoothness, orthogonality, and the area of grid cells. J. Castillo [19] developed a discrete variational system that uses sums rather than integrals.

Many grid generation systems consist of a combination of several systems. For example an algebraic system may be used to obtain an initial grid and an elliptic or variational technique used to smooth it. The most useful combination grid system is the multi-block or block-structured system which was introduced in the 1980s. Multi-block systems are formed by dividing the physical domain into several simpler sections or blocks. A grid generation system is then designed for each block. Using multi-block systems, grids have been created for very complicated three dimensional configurations [1].

Applications of Numerical Grid Generation

Much of the initial research in numerical grid generation was motivated by a desire to solve problems in computational fluid dynamics. Early systems were used to model aerodynamic and hydrodynamic phenomena such as airflow around an airplane wing or fuselage, airflow around a moving automobile, or fluid flow around a ship or submarine. Over the years the use of grid generation has expanded into nontraditional areas such as the modeling of flow through porous media and the modeling of the solidification of materials, a field which involves the study of fluid flow as well as both heat and mass transfer. Grid generation is also applicable to problems in electromagnetism, structures, and any other area involving physical phenomena that can be modeled by the solution of differential equations over a field. Figure 3 shows a two-dimensional grid around an airfoil, that is, a cross section of a wing.

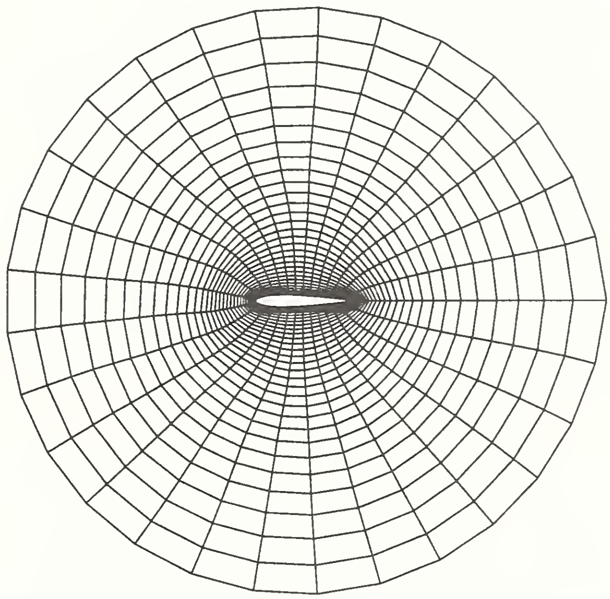


Figure 3: Airfoil grid.

The boundary data was provided by R.E. Smith of NASA Langley Research Center. The grid was generated using an algebraic system developed by the author [16]. Note that the grid is concentrated near the boundary of the airfoil. That is because more points are needed to get an accurate picture of what the flow lines look like in that area. Farther away the air flow is less affected by the body. Hence, the flow tends to be very smooth and uninteresting and fewer grid points are needed for accurate calculations. Also notice that over much of the mesh, the grid lines are close to orthogonal. A large degree of nonorthogonality severely increases the truncation error in finite difference calculations.

Another area where challenging grid generation problems arise is in the study of solidification theory. Understanding the microstructures that develop during the process of solidification can help metallurgists improve the quality of metal products manufactured by casting or welding, or aid solid-state physicists in producing pure semiconductor crystals needed for electronic devices. A common technique used to study solidification is Bridgman growth, a directional solidification method in which a small sample of the metal alloy is drawn through a constant temperature gradient at a uniform rate of speed, V, as shown in Figure 4.

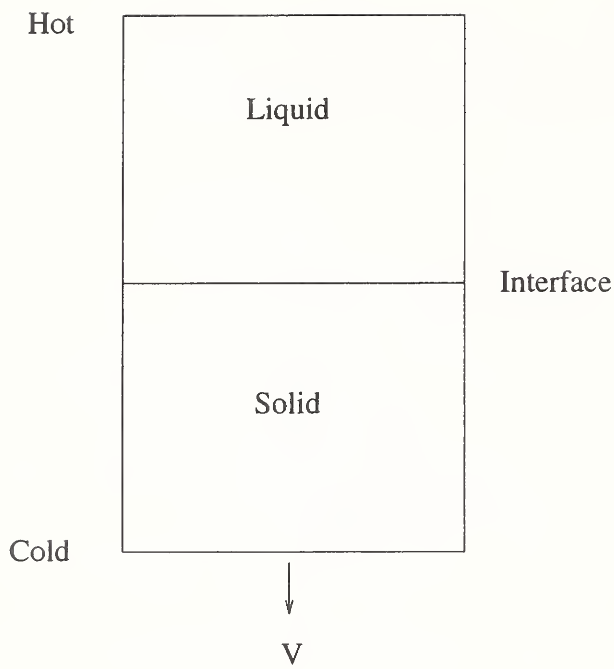


Figure 4: Bridgman growth technique.

Mullins and Sekerka discovered that there is a critical velocity at which the solid-liquid interface will become unstable [22]. As the growth speed is increased, the original flat or planar interface deforms to a sinusoidal shape, then to a bulb-like cellular shape, and then to a dendritic shape as shown in Figure 5. To fully understand this process, researchers conduct experiments or model the process numerically. Ideally, to model the process, one should use an adaptive grid with interior grid lines that conform to the shape of the interface as it deforms. However, even tracking the deformation to the cellular shape can be quite difficult because the cells can become very deep and narrow with re-entrant bulb-like shapes as the control parameters, either growth velocity or temperature gradient are modified. The grid must adapt to severe deformations while maintaining as much smoothness and orthogonality as possible. Dendritic shapes are even harder to track. Consequently, in the study of complex dendritic shapes, a considerable amount of research has been devoted to phase field models where the interface is not tracked explicitly [23, 24, 25]. Yet even in such models, grid concentration in the general area of the interface is beneficial.

The next section describes a boundary-fitted grid generation system that can be used in modeling the Bridgman growth of a binary alloy. It is designed to track the interface of the solidifying alloy as it deforms from a planar shape to a deeply grooved cell. Brown and colleagues at MIT have done quite a bit of research in this area [26, 27, 28], but the grids

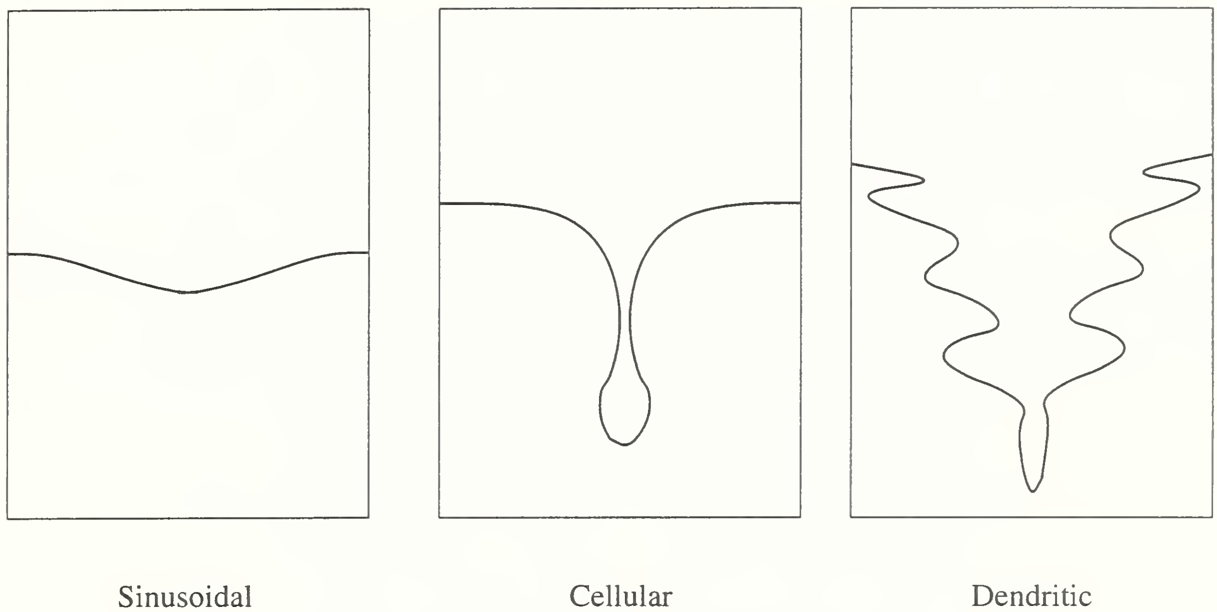


Figure 5: Instability of solid-liquid interface.

they developed for deep cells required that the interface be divided into sections [27], or that two procedures be used, one for each coordinate direction [28]. The system described in this paper requires no division of the interface or domain. Furthermore, it is an algebraic system which means no partial differential equations must be solved to obtain the grid. A more detailed discussion of the mapping is presented in [20].

A Grid Generation Mapping for Solidification Modeling

To facilitate the modeling of Bridgman growth, the boundary fitted grid generation mapping should fit the interface curve on the interior of the physical domain in addition to fitting the rectangular outer boundary. Furthermore, the system should be adaptive since the grid lines must change to follow the deforming interface while maintaining as much smoothness and orthogonality as possible. Therefore, we design a mapping, T, that 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 Figure 6. The mapping has the form

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

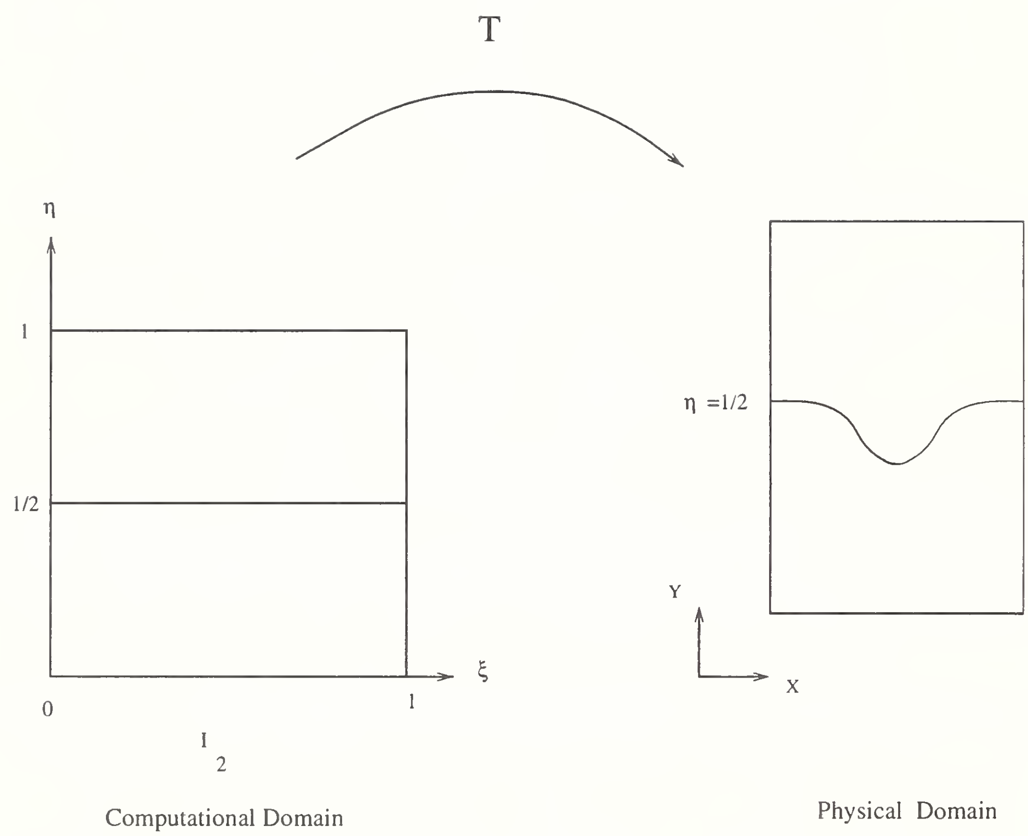


Figure 6: Grid generation mapping.

where $0 \leq \xi, \eta \leq 1$ 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 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. 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 a transfinite blending function interpolant that matches the outer boundary and interface of the physical domain. The smoothness and orthogonality of grid lines are enhanced by modifying the coefficients to minimize the discrete smoothing functional

$$\begin{aligned} G &= \sum_{i,j} w_1 \left[\left(\frac{J_{i+1,j} - J_{ij}}{\triangle \xi} \right)^2 + \left(\frac{J_{i,j+1} - J_{ij}}{\triangle \eta} \right)^2 \right] \triangle \xi \triangle \eta \\ &\quad + \sum_{i,j} w_2 Dot_{ij}^2 \triangle \xi \triangle \eta \end{aligned} \tag{8}$$

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. When w_1 is large, the variation of the Jacobian values at nearby points will be small. This decreases the variation in grid cell areas and thereby enhances grid smoothness. When w_2 is large, the dot product term will be small, causing the grid lines to approach orthogonality. Surprisingly, G is 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. The minimization code takes advantage of the small support of B-splines when evaluating the sums that comprise G and is highly vectorizable.

Figures 7 and 8 show the system's ability to generate grids that conform to extremely deformed cellular shapes that are typical of experimental and numerical results seen to date. The grids in Figure 7 are for a sinusoidal interface. The first grid was obtained by choosing spline coefficients to approximate a transfinite interpolation mapping. The second grid shows the improved grid obtained after the coefficients are modified to minimize the smoothing functional G. The system untangles and smoothly distributes the grid lines underneath the interface. The grids in Figure 8 show the grid system's ability to maintain a significant amount of smoothness and orthogonality, while adapting to a very deep interface.

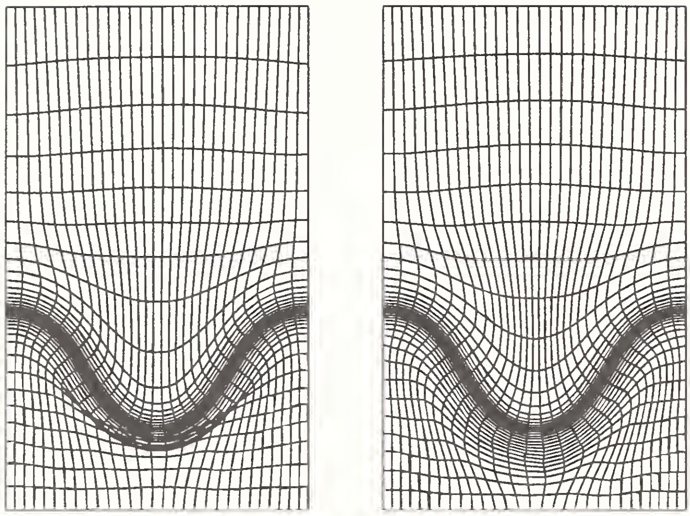


Figure 7: Initial and optimized grids for mildly deformed sinusoidal shaped interface.

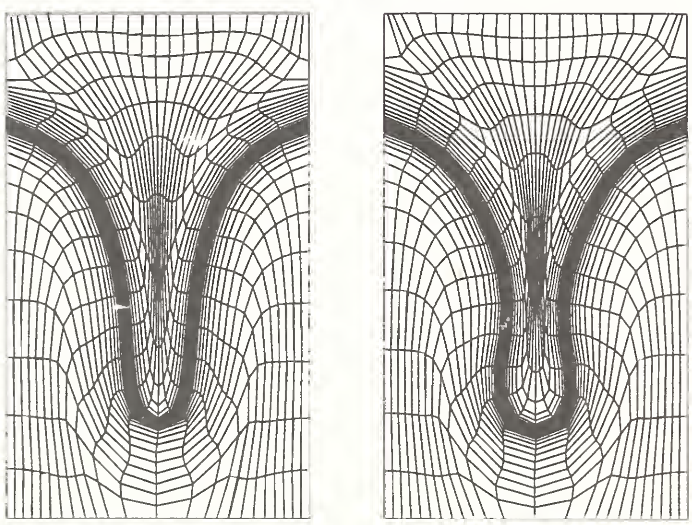


Figure 8: Optimized grids for deep and re-entrant cellular interfaces.

Conclusions and Comments

We have presented an overview of numerical grid generation, briefly described its use, and looked at some areas of application. Numerical grid generation can be a very effective tool in removing the complication of shape from the modeling of physical phenomena over a field. A great deal of progress has been made over the last thirty years, but more work is needed in several areas such as grid adaption, the development of flexible codes, and the development of better interfaces for flow solvers and geometric modelers. Also, the teaching of grid generation techniques needs to be more widespread so that the field is more accessible to potential users rather than grid generation specialists. Furthermore, even though the original motivation for grid generation came from the field of computational fluid dynamics, grid generation is now used by researchers in many fields. Unfortunately, many grid generation conferences continue to be dominated by researchers in the field of aerodynamics. A concerted effort should be made to encourage the attendance of researchers in other fields. At the same time, grid generation specialists should seek out other areas of application and present their work at conferences in other fields.

References

[1] Thompson, J. F. - A Reflection on Grid Generation in the 90s: Trends, Needs, and Influences, Numerical Grid Generation in Computational Field Simulations, ed. by B.K. Soni *et al*, pp. 1029-1110, Mississippi State University, Mississippi, 1996.

[2] Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W. - Boundary-fitted Coordinate Systems for Numerical Solution of Partial Differential Equations: A Review, Journal of Computational Physics, Vol. 47, pp. 1-108, 1982.

[3] Connell, S.D., Sober, J.S. and Lamson, S.H. - Grid Generation and Surface Modeling for CFD, Proceedings of the Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics Workshop, NASA Conference Publication 3291, p. 29, NASA Lewis Research Center, Cleveland, Ohio, 1995.

[4] Cosner, R.R. - Future Requirements in Surface Modeling and Grid Generation, Proceedings of the Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics Workshop, NASA Conference Publication 3291, p. 3, NASA Lewis Research Center, Cleveland, Ohio, 1995.

[5] Kallinderis, Y. - Discretization of Complex 3-D Flow Domains with Adaptive Hybrid Grids, Numerical Grid Generation in Computational Field Simulations, ed. by B.K. Soni *et al*, pp. 505-515, Mississippi State University, Mississippi, 1996.

[6] Hawken, D.F., Gottlieb, J.J., and Hansen, J.S. - Review of Some Adaptive Node-Movement Techniques in Finite-Element and Finite-Difference Solutions of Partial Differential Equations, Journal of Computational Physics, Vol. 95, pp. 254-302, 1991.

[7] Winslow, A.M. - Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh, Journal of Computational Physics, Vol. 2, pp. 149-172, 1967.

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

[9] Nakamura, S. - Marching Grid Generation Using Parabolic Differential Equations, Numerical Grid Generation, ed. by J.F. Thompson, North-Holland, pp. 775-807, 1982.

[10] Steger, J.L. and Sorenson, R.L. - Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates, Numerical Grid Generation Techniques, ed. by R.E. Smith, pp. 463-478, NASA-CP-2166, 1980.

[11] Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W. - Numerical Grid Generation: Foundations and Applications, North-Holland, 1985.

[12] Gordon, W.J. - Spline-Blended Surface Interpolation Through Curve Networks, Journal of Mathematics and Mechanics, Vol. 18, No.10, pp. 931-952, 1969.

[13] Gordon, W.J. and Hall, C.A. - Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation, International Journal for Numerical Methods in Engineering, Vol. 7, pp.461-477, 1973.

[14] Smith, R.E. - Algebraic Grid Generation, Numerical Grid Generation, ed. by J.F. Thompson, North-Holland, 1982.

[15] Eiseman, P.R. - A Multi-Surface Method of Coordinate Generation, Journal of Computational Physics, Vol. 33, pp. 118-150, 1979.

[16] Saunders, B.V. - Algebraic Grid Generation Using Tensor Product B-splines, NASA CR-177968, 1985.

[17] Brackbill, J.U. and Saltzman, J.S. - Adaptive Zoning in Singular Problems in Two Dimensions, Journal of Computational Physics, Vol. 46, pp. 342-368, 1982.

[18] Steinberg, S. and Roache, P.J. - Variational Grid Generation, Numerical Methods for Partial Differential Equations, Vol. 2, pp. 71-96, 1986.

[19] Castillo, J.E. - Discrete Variational Grid Generation, Mathematical Aspects of Numerical Grid Generation, ed. by J.E. Castillo, pp. 35-48, 1991.

[20] Saunders, B.V. - A Boundary Conforming Grid Generation System for Interface Tracking, Computers and Mathematics with Applications, Vol. 29, No. 10, pp. 1-17, 1995.

[21] Saunders, B.V. - A Boundary-Fitted Grid Generation System for Interface Tracking, Numerical Grid Generation in Computational Field Simulations, ed. by B.K. Soni it et al, pp. 599-608, Mississippi State University, Mississippi, 1996.

[22] Mullins, W.W. and Sekerka, R.F. - Stability of a Planar Interface During Solidification of a Dilute Binary Alloy, Journal of Applied Physics, Vol. 35, No. 2, pp. 444-451, 1964.

[23] Wheeler, A.A., Boettinger, W.J. and McFadden, G.B. - Phase-Field Model for Isothermal Phase Transitions in Binary Alloys, Physical Review A, Vol. 45, No. 10, pp. 7424-7439, 1992.

[24] Wheeler, A.A., Murray, B.T. and Schaefer, R.J. - Computation of Dendrites Using a Phase Field Model, Physica D, Vol. 66, pp. 243-262, 1993.

[25] Wang, S.-L., Sekerka, R.F., Wheeler, A.A., Murray, B.T., Coriell, S.R., Braun, R.J. and McFadden, G.B. - Thermodynamically-Consistent Phase-Field Models for Solidification, Physica D, Vol. 69, pp. 189-200, 1993.

[26] Ettouney, H.M. and Brown, R.A. - Finite-Element Methods for Steady Solidification Problems, Journal of Computational Physics, Vol. 49, pp. 118-150, 1983.

[27] Ungar, L.H., Bennett, M.J. and Brown, R.A. - Cellular Interface Morphologies in Directional Solidification. IV. The Formation of Deep Cells, Phys. Rev. B, Vol. 31, No. 9, pp. 5931-5940, 1985.

[28] Tsiveriotis, K. and Brown, R.A. - Boundary-Conforming Mapping Applied to Computations of Highly Deformed Solidification Interfaces, Int. J. Numer. Methods Fluids, Vol. 14, pp. 981-1003, 1992.