NASA-CR-195849

FINAL
IN-61-CR
0 CIT
8468
30 P

TRANSFORMATION OF TWO AND THREE-DIMENSIONAL REGIONS BY ELLIPTIC SYSTEMS

Final Report

submitted under

NASA Grant NSG 1577
NASA Langley Research Center
Hampton, VA 23665

by

C. Wayne Mastin
NSF Engineering Research Center for
Computational Field Simulation
Mississippi State University
Mississippi State, MS 39762

May 4, 1994

(NASA-CR-195849) TRANSFORMATION OF TWO AND THREE-DIMENSIONAL REGIONS BY ELLIPTIC SYSTEMS Final Report (Mississippi State Univ.) 30 p

N94-33049

Unclas

G3/61 0008468

Summary of Work

Several reports are attached which contain the results of our research at the end of this contract period. Three of the reports deal with our work on generating surface grids. One is a preprint of a paper which will appear in the journal Applied Mathematics and Computation. Another is the abstract from a dissertation which has been prepared by Ahmed Khamayseh, a graduate student who has been supported by this grant for the last two years. A few minor corrections are being made to conform to the Graduate School regulations. The final copy of the dissertation will be submitted by the end of this semester and a copy will be sent to NASA Langley Research Center at that time. The last report on surface grids is the extended abstract of a paper to be presented at the 14th IMACS World Congress in July. This report contains results on conformal mappings of surfaces, which are closely related to elliptic methods for surface grid generation.

A preliminary report is included on new methods for dealing with block interfaces in multiblock grid systems. The development work is complete and the methods will eventually be incorporated into the National Grid Project (NGP) grid generation code. Thus, the attached report contains only a simple grid system which was used to test the algorithms to prove that the concepts are sound. These developments will greatly aid grid control when using elliptic systems and prevent unwanted grid movement.

The last report is a brief summary of some timings that were obtained when the multiblock grid generation code was run on the Intel IPSC/860 hypercube. Since most of the data in a grid code is local to a particular block, only a small fraction of the total data must be passed between processors. The data is also distributed among the processors so that the total size of the grid can be increase along with the number of processors. This work is only in a preliminary stage. However, one of the ERC graduate students has taken an interest in the project and is presently extending these results as a part of his master's thesis.

to appear in Applied Mathematics and Computation

SURFACE GRID GENERATION BASED ON ELLIPTIC PDE MODELS

Ahmed Khamayseh
NSF Engineering Research Center for Computational Field Simulation, Mississippi State University
Mississippi State, Mississippi, 39762, U.S.A.

C. Wayne Mastin
Department of Mathematics and Statistics / NSF Engineering Research Center for Computational Field Simulation
Mississippi State University
Mississippi State, Mississippi, 39762, U.S.A.

Abstract

Elliptic grid generation methods for the construction of surface grids are discussed. An improved elliptic surface grid generation method for parametric surfaces will be presented. Two methods of imposing grid orthogonality have been implemented. Several control function options are available for maintaining a desired distribution of grid points on the surface. The overall effect of these improvements is a reliable and versatile elliptic method for generating and improving surface grids. Examples will be presented to demonstrate the application of the method in generating practical grids.

1. Introduction

Grid generation is a critical area for many numerical simulation problems. Whether using structured or unstructured grids, it is up to the numerical analyst to distribute points throughout an often complex physical region. The distribution of the grid points and the geometric properties of the grid such as skewness, smoothness, and cell aspect ratios have a major impact on the accuracy of the simulation.

Most grid generation proceeds in stages. In two dimensions, the grid is first constructed on the boundary curve or curves and then constructed on the interior of the physical region. In three dimensions, the grid generation process proceeds from curves to surfaces and then to the interior of the physical region. At each stage of the grid generation process, the construction of the desired grid often follows in two steps. First, a grid is constructed by interpolation from the boundary of the region or surface, and then this grid is smoothed, and possibly modified in other ways, by an iterative procedure. The most successful iterative smoothing schemes are based on elliptic systems of partial differential equations that relate the physical and computational variables. The elliptic system may be applied to the boundary grids, the interior grids, or both. The elliptic system may preserve the original distribution of grid points or redistribute points as when constructing an adaptive grid. Orthogonality of the grid may be imposed along certain boundary components of the physical region.

The most difficult and less developed area of elliptic grid generation is that of grid generation on surfaces. This is because the grid must not only be smoothed, but the grid points must also stay on the surface. For surfaces defined by parametric equations, the simplest technique for achieving this goal is to work in parameter space rather than in the physical variables of the surface. There are some disadvantages associated with this approach. The differential equations become more complicated and contain two sets of derivatives, the derivatives of the physical variables with respect to the parametric variables and the derivatives of the parametric variables with respect to the computational variables.

This paper will discuss some of the current problems in elliptic grid generation on surfaces and describe some new features and enhancements that have been recently developed. Grid orthogonality has been one area that has been of particular interest. Although orthogonal grids are generally not necessary for most numerical algorithms, there must be some limit on the skewness of the grid. Boundary orthogonality is also desirable whenever it is necessary to impose Neumann type boundary conditions on one or more of the physical variables in the problem. Also, some of the popular ways of treating Neumann boundary conditions are not accurate when the grid is not orthogonal and others become unstable if the grid is extremely skewed near the boundary.

2. Elliptic surface equations

The elliptic equations for surface grids are a generalization of Laplace equations for harmonic mappings of plane regions. Let S be a simply-connected surface defined by the parametric equations

$$\vec{r} = \vec{r}(u,v), \quad 0 \le u, v \le 1. \tag{2.1}$$

Further, let u and v be functions of the computational variables ξ and η where $0 \le \xi, \eta \le 1$. Now a cartesian coordinate system in the computational rectangle generates a curvilinear coordinate system in the parametric rectangle which maps to a curvilinear coordinate system on the surface. Thus a uniform grid in the computational rectangle generates a curvilinear grid on the surface. The elliptic system of partial differential equations which defines the transformation between computational variables and parametric variables is related to conformal mappings on surfaces which are discussed by Mastin & Thompson [3]. The same equations may also be derived from a differential geometric point of view as in the paper by Warsi [9]. The derivation here is based on conformal mappings since that approach brings out the intrinsic orthogonality and uniformity properties that are inherit in a grid generated by a conformal mapping. A surface grid generated by the conformal mapping of a rectangle onto the surface S is orthogonal and has a constant aspect ratio. These two conditions can be expressed mathematically as the system of equations

$$\vec{r}_\xi \cdot \vec{r}_\eta = 0 \tag{2.2}$$

$$m\,|\vec{r}_\xi| = |\vec{r}_\eta| \tag{2.3}$$

where m is the grid aspect ratio. These equations can be expanded using the chain rule, and it can be shown that the parametric variables satisfy the first order system

$$
\begin{aligned}
\overline{J}\ m\ v_\xi &= -\overline{g}_{12}v_\eta - \overline{g}_{11}u_\eta && (2.4)\\
\overline{J}\ m\ u_\xi &= \overline{g}_{22}v_\eta + \overline{g}_{12}u_\eta && (2.5)
\end{aligned}
$$

where

$$\overline{g}_{11} = \vec{r}_u \cdot \vec{r}_u, \quad \overline{g}_{12} = \vec{r}_u \cdot \vec{r}_v, \quad \overline{g}_{22} = \vec{r}_v \cdot \vec{r}_v, \quad \text{and} \quad \overline{J} = \sqrt{\overline{g}_{11}\overline{g}_{22} - \overline{g}_{12}^2}.$$

If the parametric and computational variables are interchanged so that the parametric variables become the independent variables, the above system can be expressed in the form

$$
\begin{aligned}
\overline{J}\ m\ \eta_v &= \overline{g}_{22}\xi_u - \overline{g}_{12}\xi_v && (2.6)\\
-\overline{J}\ m\ \eta_u &= -\overline{g}_{12}\xi_u + \overline{g}_{11}\xi_v && (2.7)
\end{aligned}
$$

This first order system is analogous to the Cauchy-Riemann equations for the conformal mapping of plane regions. From these equations, it follows that the computational variables ξ and η are solutions of a second order elliptic system. In fact, ξ and η are solutions of the following second order linear elliptic system with $F = G = 0$.

$$
\begin{aligned}
\overline{g}_{22}\xi_{uu} - 2\overline{g}_{12}\xi_{uv} + \overline{g}_{11}\xi_{vv} + (\Delta_2 u)\xi_u + (\Delta_2 v)\xi_v &= F && (2.8)\\
\overline{g}_{22}\eta_{uu} - 2\overline{g}_{12}\eta_{uv} + \overline{g}_{11}\eta_{vv} + (\Delta_2 u)\eta_u + (\Delta_2 v)\eta_v &= G && (2.9)
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta_2 u &= \overline{J}\left[\frac{\partial}{\partial u}\left(\frac{\overline{g}_{22}}{\overline{J}}\right) - \frac{\partial}{\partial v}\left(\frac{\overline{g}_{12}}{\overline{J}}\right)\right], \quad \text{and}\\
\Delta_2 v &= \overline{J}\left[\frac{\partial}{\partial v}\left(\frac{\overline{g}_{11}}{\overline{J}}\right) - \frac{\partial}{\partial u}\left(\frac{\overline{g}_{12}}{\overline{J}}\right)\right].
\end{aligned}
$$

It is this system which forms the basis of the elliptic methods for generating surface grids. The source terms (or control functions), F and G, are added to allow control over the distribution of grid points on the surface. In the computation of a surface grid, the points in the computational space are given and the points in the parametric space must be computed. Therefore, in implementation of a numerical grid generation scheme, it is convenient to interchange variables again so that the computational variable ξ and η are the independent variables. Now u and v are solutions of the following quasilinear elliptic system.

$$
\begin{aligned}
g_{22}(u_{\xi\xi} + Pu_\xi) - 2g_{12}u_{\xi\eta} + g_{11}(u_{\eta\eta} + Qu_\eta) &= J^2\Delta_2 u && (2.10)\\
g_{22}(v_{\xi\xi} + Pv_\xi) - 2g_{12}v_{\xi\eta} + g_{11}(v_{\eta\eta} + Qv_\eta) &= J^2\Delta_2 v && (2.11)
\end{aligned}
$$

where

$$\begin{aligned}
g_{11} &= \overline{g}_{11}u_\xi^2 + 2\overline{g}_{12}u_\xi v_\xi + \overline{g}_{22}v_\xi^2, \\
g_{12} &= \overline{g}_{11}u_\xi u_\eta + \overline{g}_{12}(u_\xi v_\eta + u_\eta v_\xi) + \overline{g}_{22}v_\xi v_\eta, \\
g_{22} &= \overline{g}_{11}u_\eta^2 + 2\overline{g}_{12}u_\eta v_\eta + \overline{g}_{22}v_\eta^2, \\
P &= \frac{J}{g_{22}}F, \quad Q = \frac{J}{g_{11}}G, \quad \text{and} \\
J &= u_\xi v_\eta - u_\eta v_\xi.
\end{aligned}$$

3. Boundary orthogonality and control functions

The system of partial differential equations may be solved with either Dirichlet or Neumann type boundary conditions depending on whether the grid points on the boundary of the surface are to remain fixed or are allowed to move during the construction of the grid. Imposing boundary orthogonality is the most common case where grid points are allowed to move along the boundary. However it is orthogonality of the grid on the surface and not in the parametric region that is desired. Thus the appropriate boundary conditions must be imposed on the elliptic system in the parametric region so that the grid is orthogonal on the surface. Two grid lines will be orthogonal on the surface if the inner product $\vec{r}_\xi \cdot \vec{r}_\eta = 0$. This can be expanded using the chain rule to give the following equation

$$\overline{g}_{11}u_\xi u_\eta + \overline{g}_{22}v_\xi v_\eta + \overline{g}_{12}(u_\xi v_\eta + u_\eta v_\xi) = 0. \tag{3.1}$$

The orthogonality condition can be formulated as derivative boundary conditions for the above elliptic system. If the boundary segments $u = 0$ and $u = 1$ are considered, then the orthogonality condition reduces to

$$\overline{g}_{22}v_\xi + \overline{g}_{12}u_\xi = 0. \tag{3.2}$$

Similarly, for the segments $v = 0$ and $v = 1$, orthogonality is imposed by the equation

$$\overline{g}_{11}u_\eta + \overline{g}_{12}v_\eta = 0. \tag{3.3}$$

Thus, the elliptic system together with the given values of u and v on the boundary and the orthogonality boundary conditions form a mixed boundary value problem. Equation (3.2) determines the values of v along the boundary where $u = 0$ or $u = 1$, and equation (3.3) determines the values of u where $v = 0$ or $v = 1$.

The control functions P and Q must be selected so that the grid has the required distribution of grid points in the computational field. Taking $P = Q = 0$ tends to generate a grid with a uniform spacing which is seldom desired. In most cases there is a need to concentrate points in some area of the surface such as along certain boundary contours. There are two methods of computing the control functions. The first is to compute P and Q from some initial grid by interpolating the boundary values of u and v into the interior of the parametric region. All the derivative terms in the elliptic system can be computed from given grid point values leaving two unknowns, P and Q, which can be determined from the two equations (2.10) and (2.11). Now these control function values are smoothed so that the final elliptic grid will be smoother and generally more orthogonal than the initial grid. The control functions can be computed from an initial grid only if the Jacobian of the transformation from computational to parametric variables is nonvanishing. Since this may not always be the case for interpolated grids, there is also the option of computing the control functions from the boundary distribution. The appropriate values for the control functions on the boundary can be derived by assuming the grid lines are orthogonal at the boundary and the spacing normal to the boundary is uniform. The development basically follows that for two dimensional plane regions as found in the book by Thompson et. al. [8]. The formulas for P and Q in the surface equations (2.10) and (2.11) are given below.

$$P = -\frac{s_{\xi\xi}}{s_\xi} + s_\xi \kappa_1 \tag{3.4}$$

$$Q = -\frac{s_{\eta\eta}}{s_\eta} + s_\eta \kappa_2 \tag{3.5}$$

The variable s is the arc length parameter along the boundary of the surface and the variables κ_1 and κ_2 denote the curvature of the boundary curves $\xi = constant$ and $\eta = constant$, respectively. The curvatures are given by

$$\begin{aligned} \kappa_1 &= \sqrt{\frac{g}{g_{11}^3}}\Upsilon_{11}^2, \quad \text{and} \\ \kappa_2 &= \sqrt{\frac{g}{g_{22}^3}}\Upsilon_{22}^1 \end{aligned}$$

where the Christoffel symbols Υ_{ij}^k in the previous formulas are given by

$$\begin{aligned} \Upsilon_{11}^2 &= \frac{1}{2g}\left[g_{11}(2\frac{\partial g_{12}}{\partial \xi} - \frac{\partial g_{11}}{\partial \eta}) - g_{12}\frac{\partial g_{11}}{\partial \xi}\right], \\ \Upsilon_{22}^1 &= \frac{1}{2g}\left[g_{22}(2\frac{\partial g_{12}}{\partial \eta} - \frac{\partial g_{22}}{\partial \xi}) - g_{12}\frac{\partial g_{22}}{\partial \eta}\right], \quad \text{and} \\ g &= g_{11}g_{22} - (g_{12})^2. \end{aligned}$$

Not all of the terms in equations (3.4) and (3.5) for P and Q can be evaluated on the boundary. It is clear that the arc length derivatives for P can be evaluated on the $\eta = 0$ and $\eta = 1$ boundary curves and interpolated in the interior of the computational region, while the arc length derivatives for Q can be evaluated on $\xi = 0$ and $\xi = 1$ and interpolated in the interior. In a similar manner, the curvature in each control function can be computed on opposite boundaries and interpolated. Note that the curvatures cannot be computed solely from information on the boundary curves as the case for regions of the plane. In this manner all of the arc length derivatives and curvatures, and therefore P and Q, can be computed throughout the computational region. This second control option can be used to determine the distribution of grid points on a surface even when an initial grid is of such poor quality in the interior of the surface that it cannot be used itself to generate appropriate control functions.

In the above discussion it was shown how derivative boundary conditions can be used to impose orthogonality. In a numerical algorithm, this requires moving the grid points along the boundary of the surface. Orthogonality can also be imposed by adjusting the control functions near the boundary and keeping the boundary points themselves fixed. Since there are two control functions, not only can the angle at the boundary be chosen, but the distance to the first grid line off of the boundary curve can also be chosen. If it assumed that the grid lines are orthogonal, the values of P and Q can be determined from the elliptic system (2.10) and (2.11) yielding the expressions

$$\begin{aligned} P &= J^2 \frac{\overline{g}_{11}\Delta_2 u \ u_\xi + \overline{g}_{22}\Delta_2 v \ v_\xi + \overline{g}_{12}(\Delta_2 u \ v_\xi + \Delta_2 v \ u_\xi)}{g_{11}g_{22}} \\ &- \frac{\overline{g}_{11}u_{\eta\eta}u_\xi + \overline{g}_{22}v_{\eta\eta}v_\xi + \overline{g}_{12}(u_{\eta\eta}v_\xi + v_{\eta\eta}u_\xi)}{g_{22}} \\ &- \frac{\overline{g}_{11}u_{\xi\xi}u_\xi + \overline{g}_{22}v_{\xi\xi}v_\xi + \overline{g}_{12}(u_{\xi\xi}v_\xi + v_{\xi\xi}u_\xi)}{g_{11}} \end{aligned} \tag{3.6}$$

$$\begin{aligned} Q &= J^2 \frac{\overline{g}_{11}\Delta_2 u \ u_\eta + \overline{g}_{22}\Delta_2 v \ v_\eta + \overline{g}_{12}(\Delta_2 u \ v_\eta + \Delta_2 v \ u_\eta)}{g_{11}g_{22}} \\ &- \frac{\overline{g}_{11}u_{\eta\eta}u_\eta + \overline{g}_{22}v_{\eta\eta}v_\eta + \overline{g}_{12}(u_{\eta\eta}v_\eta + v_{\eta\eta}u_\eta)}{g_{22}} \\ &- \frac{\overline{g}_{11}u_{\xi\xi}u_\eta + \overline{g}_{22}v_{\xi\xi}v_\eta + \overline{g}_{12}(u_{\xi\xi}v_\eta + v_{\xi\xi}u_\eta)}{g_{11}} \end{aligned} \tag{3.7}$$

Along a boundary component where a specified grid spacing is desired, that distance is used for the value of g_{11} or g_{22}. Some of the other derivatives in the equations for P and Q can be computed from the fixed values of u and v on the boundary. Those derivatives that must be computed using interior values are updated during the iterative solution of the elliptic system. At each iteration, these boundary control functions are smoothly blended with the interior control functions to generate a smooth grid that is orthogonal and has a specified spacing along assigned boundary components.

4. Examples

Five computational examples will be presented to demonstrate the utility of the method in grid generation. In all cases the surface is defined as a Non-Uniform Rational B-Spline (NURBS) surface. The elliptic system is solved using a finite difference discretization and the successive-over-relaxation (SOR) iterative method.

The first three examples (Figures 1,2, and 3) are simple parametric surfaces. Two grids are shown in each figure. The algebraic grid is constructed by transfinite interpolation in the parametric region. The elliptic grid is generated by smoothing the algebraic grid with the elliptic system. The surface grids in Figure are algebraic and elliptic grids for part of an aircraft fuselage. The elliptic grid is generated using zero control functions (*i.e.*, $P = 0$ and $Q = 0$), no orthogonality is applied on the boundaries. The algebraic grid contains cells with near zero areas. The closeup view of the extremely skewed cells shows that these cells have much larger areas and are less skewed in the elliptic grid. The grids in Figure 2 are on a surface with an edge which collapses to a point. The elliptic grid is generated using interpolated control functions given by (3.4) and (3.5) and the boundary points were allowed to move by imposing (3.2) and (3.3) so that the grid is orthogonal on the boundary. A comparison of the two grids in this figure demonstrates how the skewed cells adjacent to a boundary contour can be opened up by the elliptic system while still maintaining the concentration of grid points at the degenerate edge. Figure 3 contains two planer grids. The elliptic grid is generated using initial control functions, and boundary orthogonality has been achieved by iterating the control functions accordings to equations (3.6) and (3.7). The plots of the complete grids shows that the overall distribution of grid points has been maintained by the elliptic method. The closeup views of the algebraic and elliptic grids demonstrates how the control functions in the elliptic system can be used to generate an orthogonal boundary grid with a specified grid spacing at the boundary. This is a typical grid that would be constructed for viscous flow computations.

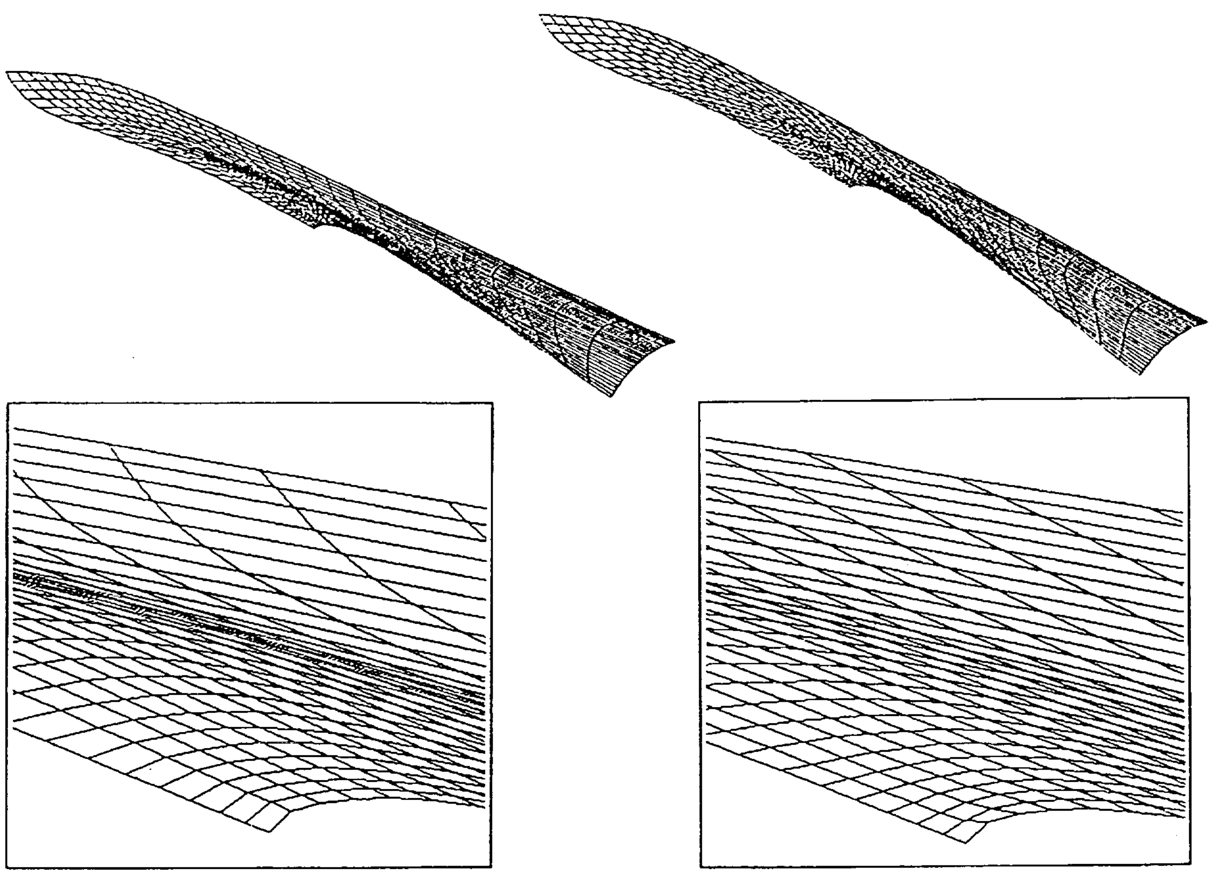

FIGURE 1. *Algebraic Grid (left) and Elliptic Grid (right).*

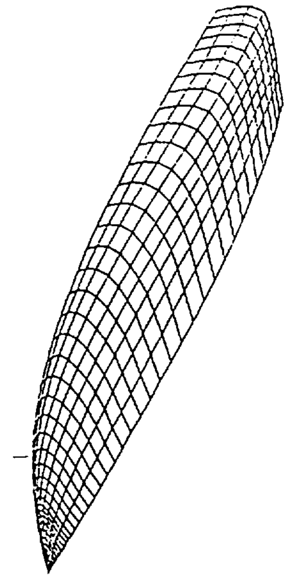

FIGURE 2. *Algebraic Grid (left) and Elliptic Grid (right).*

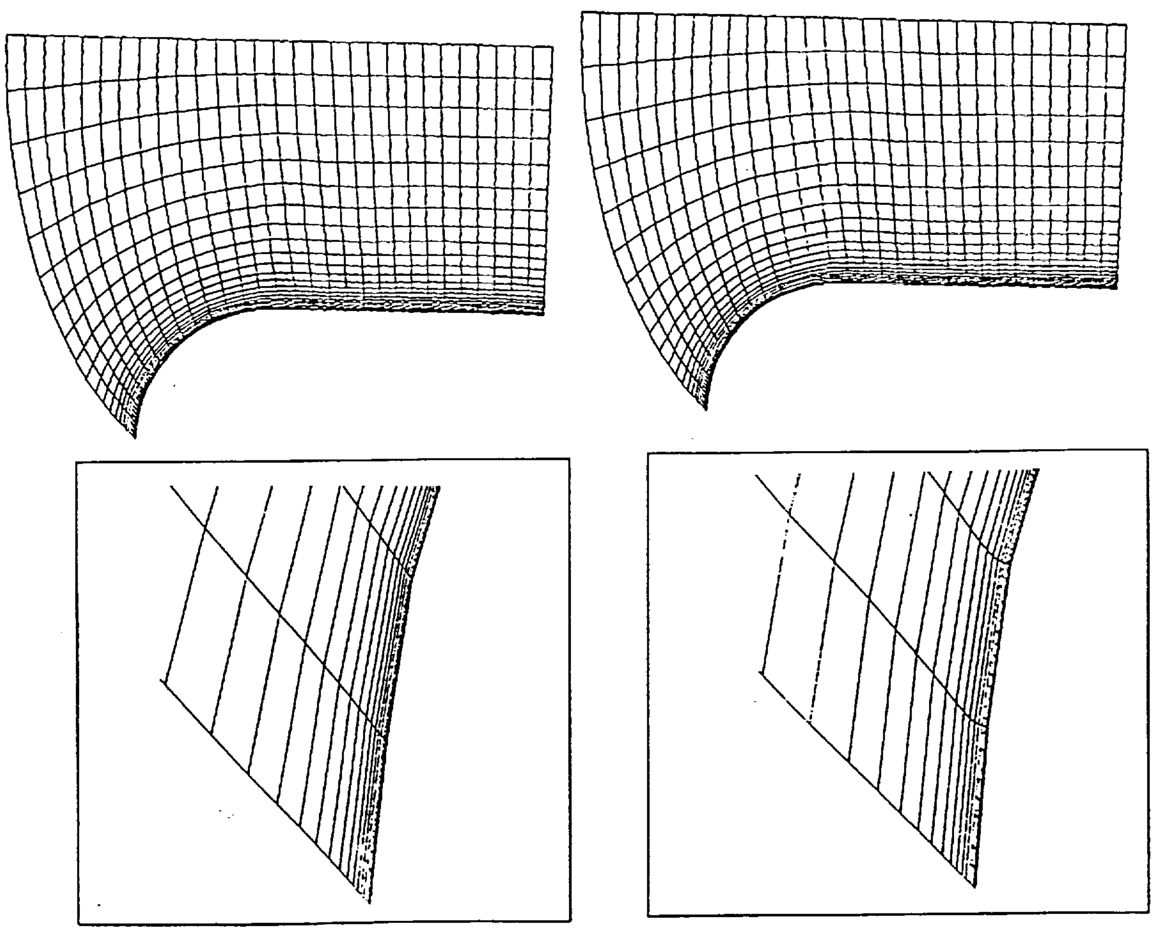

FIGURE 3. *Algebraic Grid (left) and Elliptic Grid (right).*

The next two examples are more realistic configurations. Personal Launch System and F15 geometries are depicted with an interpolated and an elliptically smoothed grid in Figures 4 and 5. Due to the large curvature of the surface, convergence problems were encountered in the numerical solution of the elliptic system. Although there is not a great deal of difference between the the algebraic and the elliptic grids. In the two examples, the elliptic grid does result in a more uniform distribution especially in front of the space shuttle wing where the grid constructed by interpolation has a sparser distribution of points. Also note that some of the kinks in front of the F15 engine inlet were removed by the elliptic system.

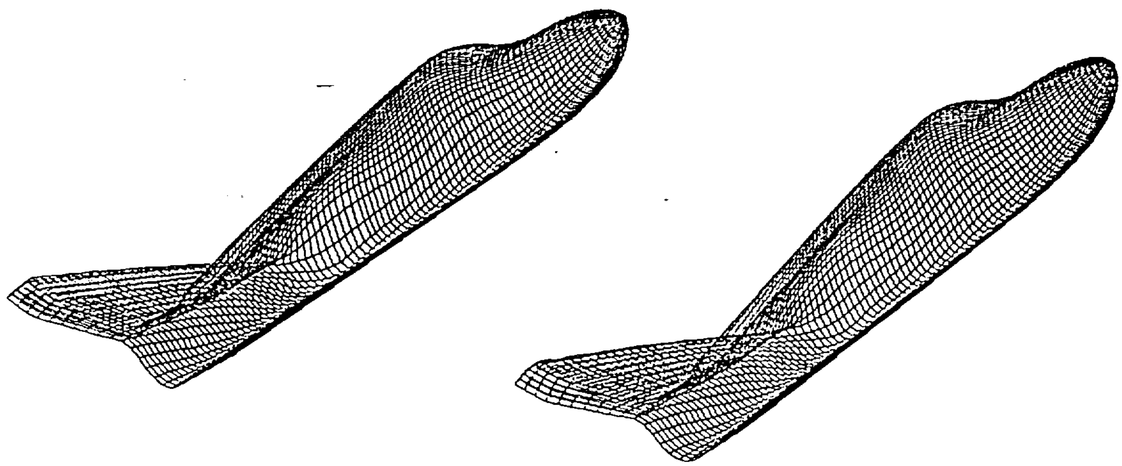

FIGURE 4. *Personal Launch System, Algebraic Grid (left) and Elliptic Grid (right).*

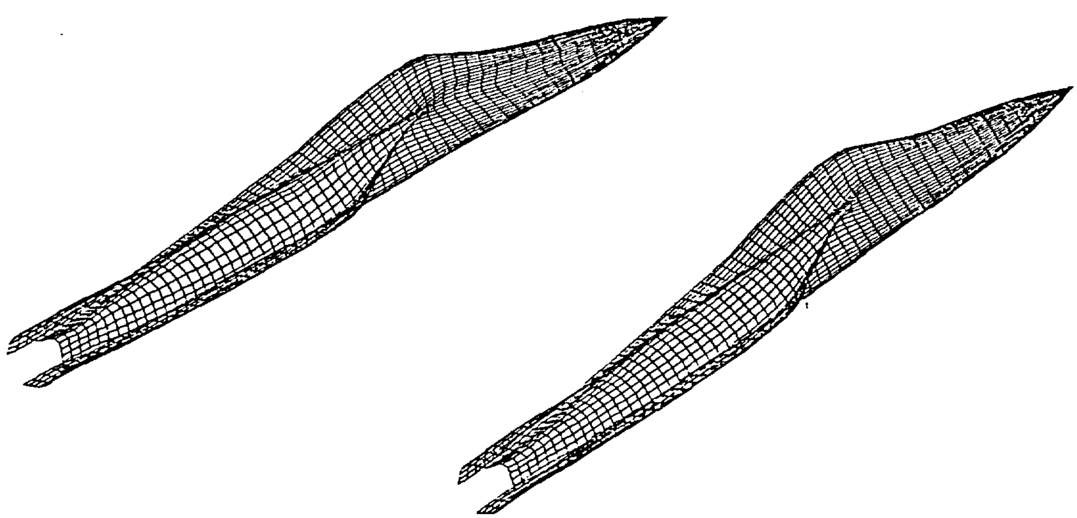

FIGURE 5. *F15 fighter, Algebraic Grid (left) and Elliptic Grid (right).*

5. Conclusion

The elliptic equations for surface grids have been presented along with the control functions and orthogonality techniques which allow a great deal of freedom in designing high quality grids. This capability would be most useful when the parameterization of the surface is such that interpolation of parametric values does not give a satisfactory grid because of a poorly parameterized surface. Since this situation arises frequently when surfaces are defined by CAD packages, the capability to smooth and improve surface grids is essential in any state-of-the-art grid generation code. The method described in this paper works well on surfaces which have a smooth parametric representation with moderate values for the parametric derivatives. If there are large variations in the parametric derivatives, then the elliptic system becomes difficult to solve and the standard iterative methods, such as SOR, will not converge. However, this is precisely the case when the elliptic grid generation methods can be most useful. Thus there is a need for more study on algorithms to solve nonlinear elliptic systems such as those encountered in this paper.

6. Acknowledgments

This research was supported by NASA Langley Research Center under Grant No. NSG 1577.

REFERENCES

[1] P. L. GEORGE, *Automatic Mesh Generation*, Wiley & Sons, New York, NY, 1991.

[2] C. W. MASTIN AND A. KHAMAYSEH, *Elliptic Grid Generation on Surfaces*, Proceedings of the International Conference on Hydro-Science and Engineering, Washington, DC, June 1993.

[3] C. W. MASTIN AND J. F. THOMPSON, *Quasiconformal Mappings and Grid Generation*, SIAM Journal on Scientific and Statistical Computing, Vol. 5, 1984, pp. 305-310.

[4] L. A. PIEGL, *Rational B-spline Curves and Surfaces for CAD and Graphics*, in State of the Art in Computer Graphics, D. F. Rogers and R. A. Earnshaw, eds., Springer-Verlag, New York, NY, 1991, pp. 225–269.

[5] J. C. STRIKWERDA, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole, Pacific Grove, CA, 1989.

[6] T. TAKAGI, K. MIKI, B. C. J. CHEN, AND W. T. SHA, *Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems for Arbitrarily Curved Surfaces*, Journal of Computational Physics, Vol. 58, 1985, pp. 67-79.

[7] P. D. THOMAS, *Construction of Composite Three-Dimensional Grids from Subregion Grids Generated by Elliptic Systems*, AIAA Paper No. 81-0996, 1981.

[8] J. F. THOMPSON, Z. U. A. WARSI AND C. W. MASTIN, *Numerical Grid Generation*, North-Holland, New York, NY, 1985.

[9] Z. U. A. WARSI, *Numerical Grid Generation in Arbitrary Surfaces Through a Second-Order Differential Geometric Model*, Journal of Computational Physics, Vol. 64, 1986, pp. 82-96.

ELLIPTIC SURFACE GRID GENERATION ON ANALYTICALLY DEFINED GEOMETRIES

By

Ahmed Khamayseh

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in the Department of Mathematics and Statistics

Mississippi State, Mississippi

May 1994

Name: Ahmed Khamayseh

Date of Degree: May 14, 1994

Institution: Mississippi State University

Major Field: Mathematical Sciences

Major Professor: Dr. C. Wayne Mastin

Title of Study: ELLIPTIC SURFACE GRID GENERATION ON ANALYTICALLY DEFINED GEOMETRIES

Pages in Study: 102

Candidate for Degree of Doctor of Philosophy

Elliptic grid generation is a grid generation technique where the coordinates of the grid points are computed by solving a system of elliptic partial differential equations. An elliptic generation system for surface grids is developed based on three different approaches: (i) conformal mapping of surfaces onto rectangular regions, (ii) differential geometric models, and (iii) three-dimensional elliptic differential equations. Elliptic generation is fundamentally a smoothing process, and the choice of elliptic equations as the generation system produces a smooth grid. However, in order to exercise control over the grid lines, control functions are included in the elliptic equations. These control functions can be determined by interpolation of spacing and curvature elements from the boundaries, or can be determined and smoothed from the initial algebraic grid. They may also be used to produce orthogonality with specified off-boundary spacing at boundaries through an iterative process.

Recently, there has been a move towards Non-Uniform Rational B-Spline (NURBS)-based grid generation systems, where the original geometry is given as analytically defined NURBS surfaces. The NURBS format offers a common mathematical representation for both standard analytic shapes and free-form curves and surfaces. NURBS representations are especially important in implementing elliptic grid generation methods since the derivatives of the physical coordinates with respect to the parametric coordinates can be evaluated directly.

The solution of the elliptic equations is generally done by SOR iteration using a field of optimum acceleration parameters and directed differences based on the sign of the control functions for the first derivatives.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation for the help and guidance of my advisor Dr. Wayne Mastin. He assisted me in preparing and conducting research and in presenting it to the scientific community. He has been a source of inspiration.

I further express my gratitude to Dr. Z. U. A. Warsi who did not only support me during the process of creating this dissertation, but who also taught me very different ways of describing and solving mathematical problems using diverse tools.

I would also like to express special thanks to Dr. Robert Smith, Dr. Bharat Soni, Dr. Ratnasingham Shivaji, and Dr. Jainping Zhu for being on my committee and for their valuable suggestions regarding this work. I am grateful to Dr. Thomas Delillo who read my manuscript carefully and offered many helpful suggestions.

I extend my thanks to all members of the National Grid Project (NGP) research group at Mississippi State University for their assistance and all the benefits I gained in the many discussions I had with them.

This research was supported by NASA Langley Research Center under Grant No. NSG 1577 and by the NSF Engineering Research Center for Computational Field Simulation at Mississippi State University.

TABLE OF CONTENTS

CHAPTER Page

to be presented at 14th IMACS World Congress

Conformal Mapping and Grid Generation on Surfaces*

by

C. Wayne Mastin
Department of Mathematics and Statistics
and
NSF Engineering Research Center for Computational Field Simulation
Mississippi State University
Mississippi State, MS 38762
Phone: 601-325-3414 e-mail: wayne@erc.msstate.edu

1 Introduction

The generation of surface grids is the first major challenge in the construction of grids for computational simulations. Unlike the case for gridding curves, it is not easy to construct a smooth surface grid with a specific distribution of points. This is especially true if the surface is poorly parameterized with little relation between the spacings in the parametric rectangle for the surface and the actual grid point spacings on the surface. In the ideal case, the grid should be dependent only on the geometry of the surface and should be free of the particular parameterization used to define the surface. This is precisely the case with grids generated from conformal mappings of rectangular regions onto surfaces. Although the theory behind conformal mappings of surfaces has been known for some time, there have been few applications. In the field of grid generation, this has been true primarily because there was little interest in gridding analytically defined surfaces such as conic surfaces. However, recent developments in computational geometry have greatly increased the range of analytically defined surfaces in geometric modeling. There has been a trend toward the use of surfaces defined by a standardized set of splines to model arbitrary geometric configurations. Along with a spline definition of a surface comes the ability to evaluate the coordinate values and their derivatives. This is precisely the information that is needed to implement a method for generating a conformal mapping of a rectangular region onto a surface. The mapping then determines a curvilinear coordinate system, and hence a grid, on the surface.

Previous work by Mastin and Thompson [1] developed methods for the construction of quasiconformal mappings of planar regions which are related to the conformal mapping of surfaces. In a recent report, Khamayseh and Mastin [2] have extended these ideas and have examined the use of more general elliptic partial differential equations in the construction of surface grids. In this report, the equations for constructing conformal mappings of surfaces will be developed. Conformal mappings from a rectangular region onto a surface will be constructed and used to generate computational grids for several configurations of practical interest in computational fluid dynamics.

2 Surface Mapping

The derivation of the conformal mapping equations for surfaces will be carried out with an arbitrary simply-connected parametric surface S given by the equation

$$\mathbf{r} = \mathbf{r}(u, v), \; 0 \le u, v \le 1.$$

The variables u and v are the parametric variables for the surface, which are assumed to lie in the unit square, and $\mathbf{r}$ is the vector of surface coordinate values (x, y, z). The surface can be mapped conformally

*This research was supported by NASA Langley Research Center under Grant NSG 1577.

onto a rectangular region R of the plane which will be given as

$$0 \leq \xi \leq M,\ 0 \leq \eta \leq 1.$$

The positive number M is determined once four points on the boundary of the surface are assigned to the four vertices of the rectangular region. The value of M, called the module of the surface, is calculated during the construction of the conformal mapping. For a rectangular region, there is a simple geometric characterization of a conformal mapping onto a surface. Any pair of level curves, ξ = constant and η = constant, will be orthogonal and have the same arc length distribution. These observations follow from the "pure magnification" property of conformal mappings and can be expressed by the equations

$$\mathbf{r}_\xi \cdot \mathbf{r}_\eta = 0, \tag{1}$$

$$|\mathbf{r}_\xi| = |\mathbf{r}_\eta|. \tag{2}$$

In most mapping problems involving surfaces, it usually necessary to work with the parametric variables so that the exact surface is maintained by the mapping algorithm. Following this procedure, the equations (1) and (2) can be expressed in terms of derivatives of parametric variables u and v with respect to ξ and η. The equations are quite length, but can be combined into a simple complex equation

$$|\mathbf{r}_u|^2(u_\xi + iu_\eta)^2 + 2\mathbf{r}_u \cdot \mathbf{r}_v(u_\xi + iu_\eta)(v_\xi + iv_\eta) + |\mathbf{r}_u|^2(v_\xi + iv_\eta)^2 = 0. \tag{3}$$

From this single complex equation, the following first order system can be derived.

$$Ju_\xi = \mathbf{r}_v^2 v_\eta + \mathbf{r}_u \cdot \mathbf{r}_v u_\eta \tag{4}$$

$$-Jv_\xi = \mathbf{r}_u \cdot \mathbf{r}_v v_\eta + \mathbf{r}_v^2 u_\eta \tag{5}$$

The quantity J is the jacobian of the mapping from the rectangular region R to the square region in parameter space and is given by

$$J = u_\xi v_\eta - u_\eta v_\xi.$$

Note that the quadratic equation (3) has two possible solutions, however, only one will give a positive value for J.

3 Examples and Conclusions

The above system (4) and (5) is similar to the Cauchy-Riemann equations in the conformal mapping of planar regions. As such the same numerical methods for constructing conformal mappings of planar regions can be modified to construct conformal mappings of surfaces. In the examples presented here, the mapping and resulting grids were constructed by forming the second order elliptic Beltrami equations for u and v. Derivative boundary conditions were implemented to obtain the correct distribution of grid points on the boundary. The aspect ratio ,M, of the rectangular region R was also approximated during the solution of the elliptic system.

Some of the grids constructed from conformal mappings are given in Figure 1. In all cases the surface was defined as a nonuniform rational B-spline (NURBS). The complexity of these surfaces should indicate that conformal mappings can be used to generated a smooth and relatively uniform distribution of grid points on surfaces of practical interest in the aerospace industry.

References

[1] C. W. Mastin and J. F. Thompson, "Quasiconformal Mappings and Grid Generation", *SIAM J. Sci. Stat. Comput.*, Vol. 5, pp. 305-310, June 1984.

[2] A. Khamayseh and C. W. Mastin, "Surface Grid Generation Based on Elliptic PDE Models", *J. Comp. Appl. Math.*, to appear.

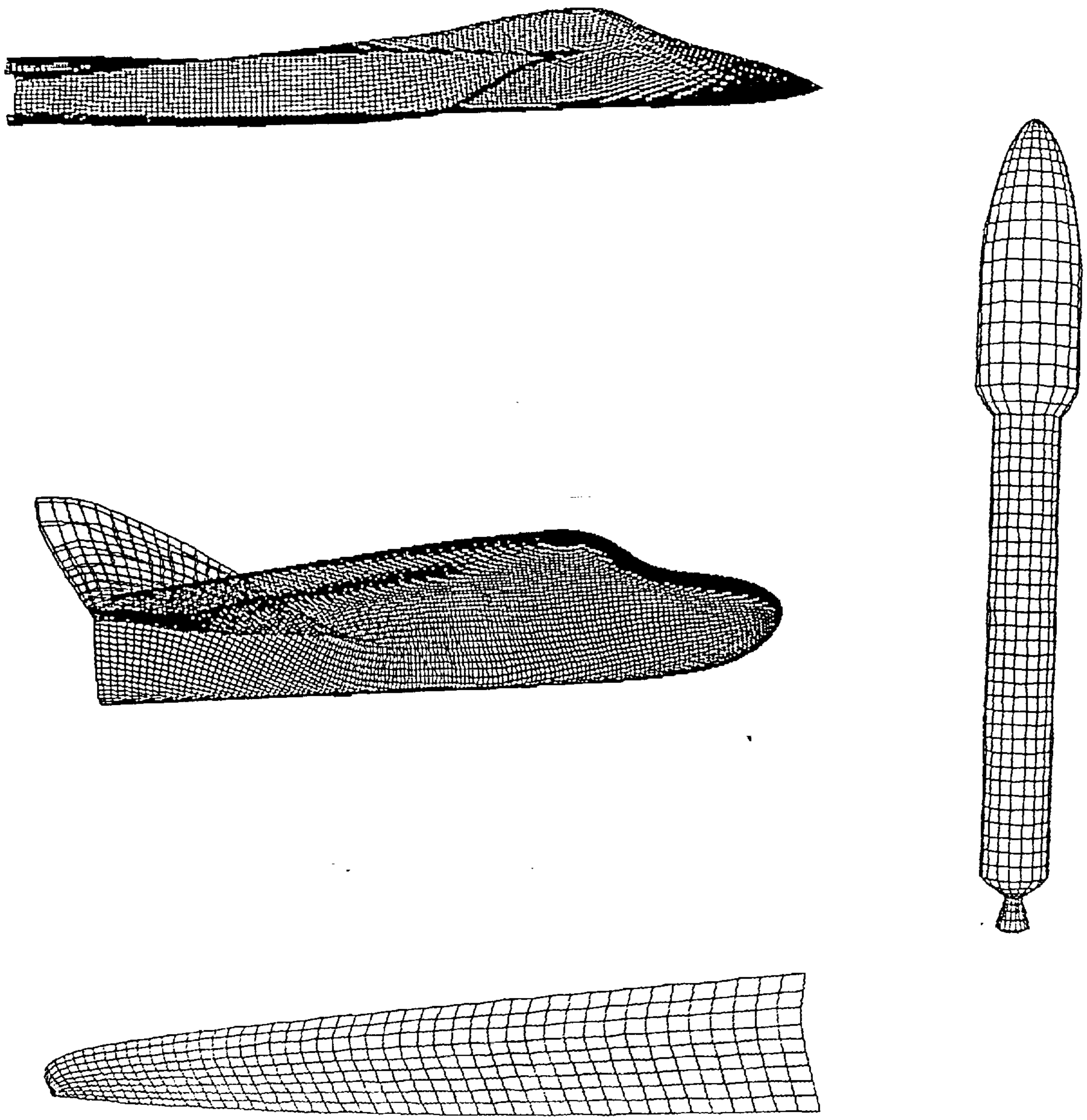

Figure 1: Surface grids generated by conformal mappings.

The Treatment of Interior Block Boundaries in Elliptic Grid Generation*

by

C. Wayne Mastin
Department of Mathematics and Statistics and
NSF Engineering Research Center for Computational Field Simulation
Mississippi State University
Mississippi State, MS 38762

1 Introduction

In the field of elliptic grid generation, there has been a considerable effort devoted to the construction of boundary conditions based on some desired grid property. These conditions may take the form of orthogonality conditions and/or grid spacing conditions. They are typically imposed by either allowing the points to move along the boundary or by modifying the control functions in the elliptic system near the boundary. Implementations of these techniques may be found in the works of Thompson [2] and Sorenson [1]. Neither of these methods are directly applicable to interior block boundary faces. However, there is clearly a need to control the interior block boundaries in certain cases. An early effort in this direction used the biharmonic equation to generate grids with fixed block boundaries. That method did not become very popular because of well-known difficulties in solving the biharmonic equation.

This report will develop methods for treating interior block boundaries using the traditional Lapacian based elliptic grid generating equations. The objective is to generate a smooth grid with the block boundaries remaining fixed of nearly fixed. Two methods will be developed. The first method allows the points to slide along the boundary during the iterative solution of the elliptic system. This method is motivated by the sliding point (or Neumann) boundary condition that has been developed for generating a grid which is orthogonal along some boundary surface. Note that the Neumann boundary conditions could not be directly applied to interior boundaries without resulting in a discontinuity in grid lines. The second method imposes continuity along block boundaries by modifying the control functions during the iteration procedure. This is somewhat similar to the use of control functions to impose orthogonality and spacings along boundary surfaces. That procedure could actually be used to generate grids with fixed interior faces. However, neither the orthogonality condition nor the fixed spacing criteria would necessarily be a desireable feature at interior block boundaries. Therefore, a simpler procedure is derived which only attempts to maintain grid line and slope continuity along a fixed interior face.

*This research was supported by NASA Langley Research Center under Grant NSG 1577.

2 Sliding Point Methods

If a surface is to remain fixed during the solution of the elliptic system, it is likely that the surface has been defined by some analytical method, possibly a spline or a quadratic surface. That case will be considered first. A method will be described which uses the parametric description of the surface so that the grid points will remain precisely on the surface. The implementation of the method is straightforward. Each point on an interior face is updated as usual during the iterative solution of the elliptic difference equations. Then this new point is projected onto the surface. In order to maintain the efficiency of the algorithm, a direct method is used for the projection. Since the projection step is the only new part of the method, it is the only step that will be described here. This step is not derived for any particular elliptic system, but could be used in conjunction with any grid smoothing algorithm.

Consider an interior block boundary which is defined parametrically by the equation

$$r = r(u, v)$$

where $r = (x, y, z)$ and u and v are the parametric coordinates of the points on the boundary surface. Let $\hat{r} = r(u_0, v_0)$ be an arbitrary grid point. During the process of smoothing the multiblock grid, the point is updated using the iteration equations. Suppose the new point, which may not lie on the surface, is denoted by $\tilde{r}$. In order to retain the block boundary, this point will be projected onto the original parametric surface. The general procedure is illustrated in Figure 1. The objective in this case is to determine the parametric coordinates of the projection. The projection, which will be the grid point for the next iteration step, will be simply denoted as $r(u, v)$.

Assume that $r(u, v)$ is close to $r(u_0, v_0)$ and can be approximated by a linear Taylor polynomial.

$$r(u, v) = r(u_0, v_0) + r_u(u_0, v_0)(u - u_0) + r_v(u_0, v_0)(v - v_0) \tag{1}$$

If ξ and η denote the curvilinear coordinate directions on the surface, then the orthogonal projection $r(u, v)$ of the iteration point $\tilde{r}$ onto the surface will satisfy the equations

$$\begin{aligned} (\tilde{r} - r(u, v)) \cdot r_\xi &= 0 \\ (\tilde{r} - r(u, v)) \cdot r_\eta &= 0 \end{aligned} \tag{2}$$

Substituting the above linear approximation for $r(u, v)$ in (1) into the system (2) gives the following linear system for parametric values u and v.

$$\begin{aligned} (r_u \cdot r_\xi)(u - u_0) + (r_v \cdot r_\xi)(v - v_0) &= (\tilde{r} - \hat{r}) \cdot r_\xi \\ (r_u \cdot r_\eta)(u - u_0) + (r_v \cdot r_\eta)(v - v_0) &= (\tilde{r} - \hat{r}) \cdot r_\eta \end{aligned} \tag{3}$$

Here it has been assumed that the surface is defined by a set of parametric equations, and therefore, the derivatives with respect to the parametric variables u and v could be computed directly from the parmetric equations for the surface. However, the derivatives with respect to the curvilinear coordinates on the surface would have to be approximated by finite differences. Two conditions are necessary for the system (3) to be nonsingular and hence to determine unique values for $u - u_0$ and $v - v_0$. On expanding the derivatives r_ξ and r_η using the chain rule, the determinant of the coefficient matrix can be expressed as

$$[(r_u \cdot r_u)(r_v \cdot r_v) - (r_u \cdot r_v)^2][u_\xi v_\eta - u_\eta v_\xi].$$

Thus it is seen that the system will be solvable if both the mapping from curvilinear coordinates to parametric coordinates and the mapping from parametric coordinates to surface coordinates have nonvanishing jacobians.

A similar projection approach can be used for interior block boundaries which are not parametrically defined. The method will exactly keep points on a planar boundary, but points on a curved surface will gradually drift off of the initial surface.

Let $\hat{r}$ denote the current position of the grid point on the boundary and $\tilde{r}$ denote the location after an iteration of the elliptic difference equations. The next step will be to project the point $\tilde{r}$ onto a plane passing through the point $\hat{r}$. The plane is determined by the curvilinear directions of the grid lines on the boundary. To be precise we replace $\tilde{r}$ by a point r of the form

$$r = \hat{r} + \alpha r_\xi + \beta r_\eta, \tag{4}$$

where α and β are arbitrary parameters. If r is the foot of the perpendicular from $\tilde{r}$ to the plane determined by this equation, then

$$\begin{aligned}(\tilde{r} - r) \cdot r_\xi &= 0 \\ (\tilde{r} - r) \cdot r_\eta &= 0\end{aligned} \tag{5}$$

On substituting the above approximation for r given in (4) into the system (5) we derive the following system for determining the coefficients α and β.

$$\begin{aligned}\alpha(r_\xi \cdot r_\xi) + \beta(r_\xi \cdot r_\eta) &= (\tilde{r} - \hat{r}) \cdot r_\xi \\ \alpha(r_\xi \cdot r_\eta) + \beta(r_\eta \cdot r_\eta) &= (\tilde{r} - \hat{r}) \cdot r_\eta\end{aligned}$$

The solvabilty of the system can be easily determined. The coefficient matrix is $(r_\xi \cdot r_\xi)(r_\eta \cdot r_\eta) - (r_\xi \cdot r_\eta)^2$, and this quantity is nonzero provided the mapping from curvilinear coordinates to surface coordintes is nonzero, or equivalently, the curvilinear coordinate lines intersect at a nonzero angle. Of course, a highly skewed grid may result in a poorly conditioned system.

3 Control Function Methods

While the above techniques can be applied to any grid smoothing method, the next technique has been developed for the three-dimensional elliptic grid generation equations that have been implemented in many grid codes such as EAGLE, GRIDGEN, and NGP. The grid generation equations are of the form

$$g^{11}(r_{\xi\xi} + P\ r_\xi) + g^{22}(r_{\eta\eta} + Q\ r_\eta) + g^{33}(r_{\zeta\zeta} + R\ r_\zeta) + g^{12} r_{\xi\eta} + g^{13} r_{\xi\zeta} + g^{23} r_{\eta\zeta} = 0 \tag{6}$$

where g^{ij} are the contravariant metric tensor components. The functions P, Q, and R are used to control the distribution of grid points in the three-dimensional region.

Now consider a two block region with discontinuity in grid line slopes and spacings. Suppose that the grid points on the block boundary are to remain fixed, and the control functions are to be updated during the iteration procedure to force continuity in grid line spacings and tangents at the boundary points. A typical situation indicating an initial grid line and the desired grid line at the end of the grid smoothing is illustrated in Figure 2.

In most all grid generation problems, the control functions are chosen to be nonzero functions which reflect some desired grid distribution in the region. The control functions are normally fixed during the iterative solution of the elliptic system. This set of control functions will be denoted as P_0, Q_0, and R_0. In order to generate a smooth grid in the region about the boundary surface, the control functions will be updated at each iteration, and the values at iteration n will be P_n, Q_n, and R_n. Along the boundary surface, a set of control functions is computed by solving the elliptic system (6) for P, Q, and R using the finite difference approximations for the derivatives in the elliptic system. These difference approximations involve the points on the boundary surface and the neighboring points in each block. Consider the case where the boundary surface corresponds to the index $\zeta = \zeta_{max}$ in the one of the blocks. The control functions computed from the evaluation of the difference equations on the surface will be denoted as

$$\tilde{P}(\xi,\eta),\ \tilde{Q}(\xi,\eta),\ \tilde{R}(\xi,\eta).$$

The control functions that will be used at iteration $n+1$ are a blended value of the control functions at iteration n and the values computed on the surface. The blending is accomplished using an exponential decay factor of the form

$$f = exp(\omega(\zeta - \zeta_{max})$$

The control functions on the surface are divided with one-half the computed value used in each of the blocks having the surface as a common boundary. Thus, the final form for the control function values at iteration $n+1$ are

$$\begin{aligned} P_{n+1}(\xi,\eta,\zeta) &= \frac{1}{2} f \tilde{P}(\xi,\eta) + (1-f)P_n(\xi,\eta,\zeta), \\ Q_{n+1}(\xi,\eta,\zeta) &= \frac{1}{2} f \tilde{Q}(\xi,\eta) + (1-f)Q_n(\xi,\eta,\zeta), \\ R_{n+1}(\xi,\eta,\zeta) &= \frac{1}{2} f \tilde{R}(\xi,\eta) + (1-f)R_n(\xi,\eta,\zeta). \end{aligned}$$

Note that this procedure is similar to that used by the GRAPE code of Sorenson [1] and the EAGLE code of Thompson [2] to enforce orthogonality and spacing along a boundary surface. However, the implementation is much simpler since only smoothness of the grid is desired, and no particular grid spacing or angle is imposed at the surface. It this method, the discontinuity is essentially dissipated by spreading its effect into both neighboring regions.

4 Example

The above methods for elliptic grid generation with fixed interior block boundaries are being implemented in the NGP grid code. Since that implementation is not complete at the present time, a simple two-dimensional example will be presented which demonstrates the basic features of the schemes.

The four grids in Figure 3(a-d) were generated on the same two-block region using different methods. The grid in Figure 3(a) was computed using transfinite interpolation. The grid was constructed independently in each block and there is an obvious discontinuity in grid spacing and direction at the boundary between the two blocks. The grid in Figure 3(b) was constructed using the standard multiblock elliptic grid generation method. The discontinuities at the block boundary have been eliminated. However, the boundary curve has moved and resulted in skewed grid lines along part of boundary of the region. Figure 3(c) is a grid which results when the grid points are allowed to slide along the block boundary. The slope discontinuity has been eliminated, but not the spacing discontinuity. The grid in Figure 3(d) has been constructed using the iterative update of the control functions to enforce continuity of grid line spacings and slopes along the interface.

Although the grid in this example is a two-dimensional grid, the grid was generated as a three-dimensional grid with an extra layer of grid points on each side of the planar grid. This technique is being used in NGP for all two-dimensional elliptic grids.

5 Conclusions

For the example presented in this report, and in other examples that were used as test cases, the control function approach appeared to generate the most pleasing grids. However, the method does have some disadvantages. As in the GRAPE procedure, the iterative adjustment of control functions can sometimes cause problems with convergence of the iterative scheme. The quality of the grid is also sensitive to the decay factor ω that is used to blend the block boundary control functions with the existing control functions in the region. The sliding point methods seem to work well at eliminating slope discontinuities, although the remaining spacing discontinuity is a problem. An attempt was made to use an adjustment of one of the control functions along with the sliding point technique to generate a grid with both slope and spacing continuity, but that method proved to be unstable.

References

[1] R. L. Sorenson, Three-Dimensional Zonal Grids About Arbitrary Shapes by Poisson's Equation, in *Numerical Grid Generation in Computational Fluid Mechanics*, S. Sengupta, J. Hauser, P. R. Eisman, and J. F. Thompson, eds., Pineridge Press, Swansea, 1988.

[2] J. F. Thompson, A Composite Grid Generation Code for General 3D Regions – the EAGLE Code, *AIAA Journal*, 1988 (26), p. 271.

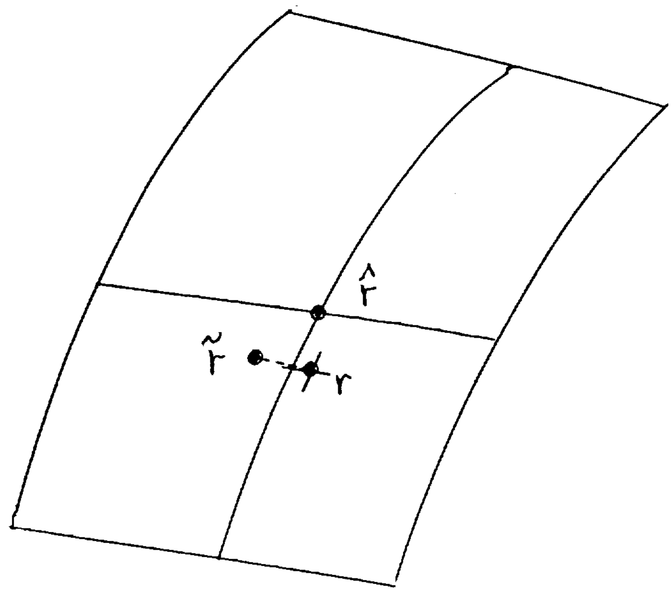

Figure 1: Sliding points on a fixed surface.

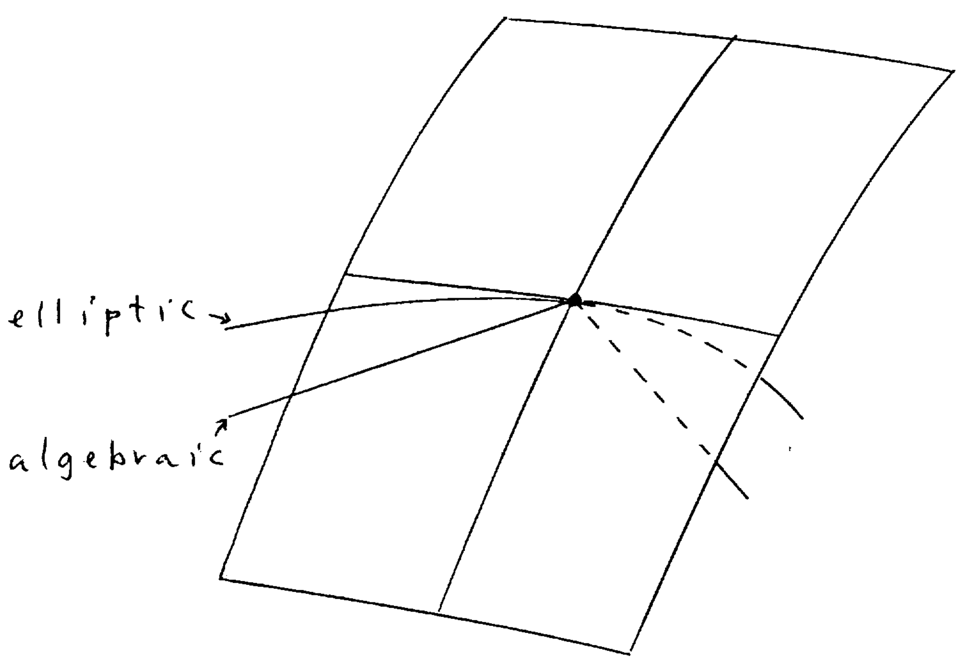

Figure 2: Effect of interior control points for a surface.

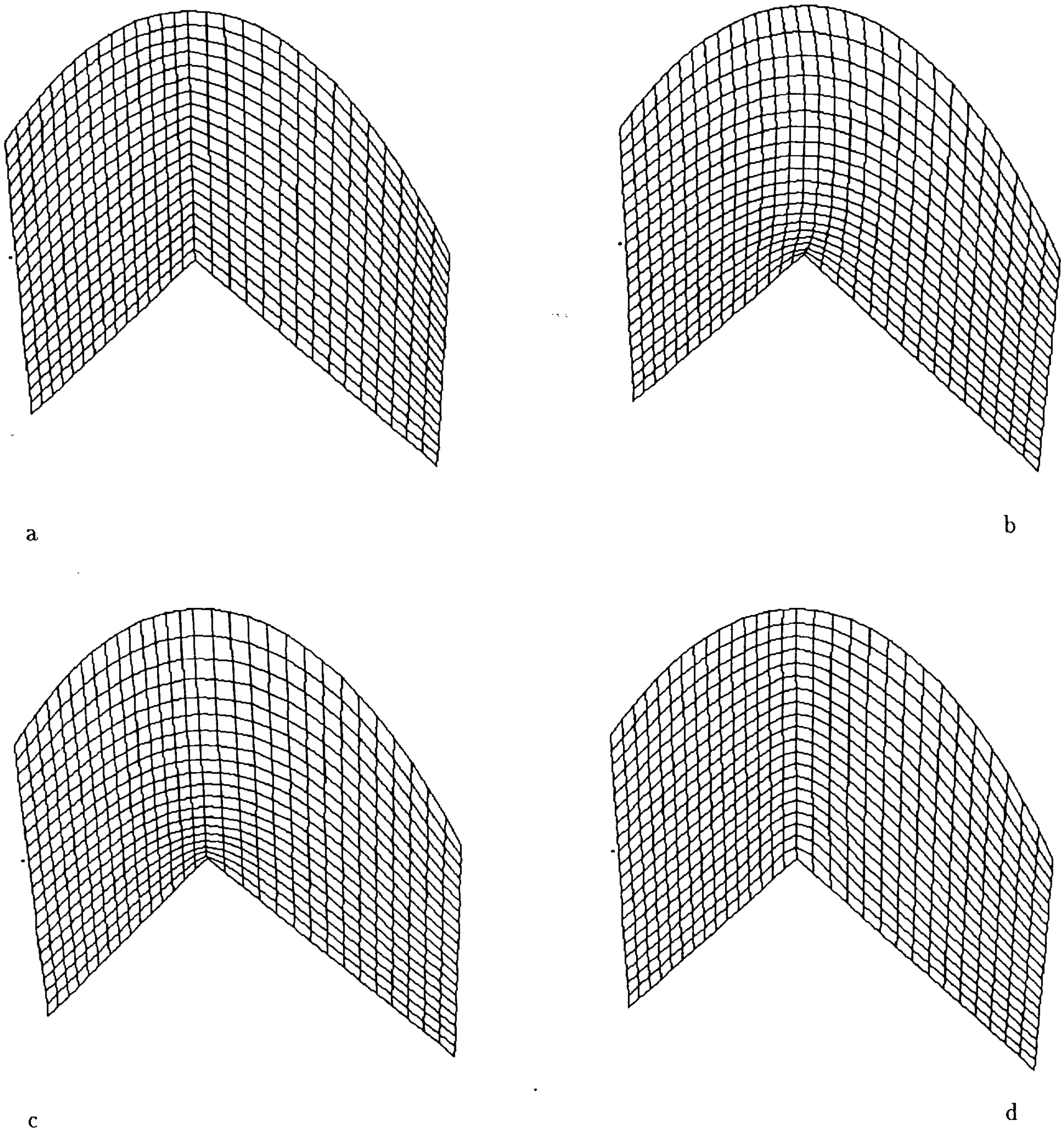

Figure 3: Grid constructed for a two-block two-dimensional region using: (a) interpolation, (b) multiblock elliptic grid generation, (c) sliding points, and (d) control function smoothing.

Three-Dimensional Grid Generation on the Intel IPSC/860 Parallel Computer

by

C. Wayne Mastin and Ken Mancel
Department of Mathematics and Statistics
and
NSF Engineering Research Center for Computational Field Simulation
Mississippi State University
Mississippi State, MS 38762

A version of the multiblock three-dimensional grid generation code that is presently being used in the NGP system has been ported to the Intel IPSC/860 parallel computer. The majority of the data in the code is local to each block and the only data that must be passed between nodes is the grid coordinates. Grids were constructed on a sequence of grids ranging from a single-block $33 \times 20 \times 20$ grid on one processor up to a 32-block $1025 \times 20 \times 20$ grid on 32 processors. These same grids were generated on one of our Silicon Graphics workstations.

As can be seen in Figure 1, there was nearly complete parallelism for this particular example. It should be noted that this problem is not necessarily representative a real world grid generation problem and there are several factors which would act to reduce the efficiency of a parallel computer such as the Intel. All of the grid blocks in this problem were of the same size, and the blocks were all stacked in the first coordinate direction so that each processor only had to communicate with two different nodes. In a realistic problem, the grid blocks would be of varying sizes and it would be desireable to have several of the smaller blocks on a single processor. Also, in a multiblock grid system each block would have up to six neighbors with which it would have to send and receive data.

It is hoped that this preliminary work will motivate further work on the implementation of multiblock grid generation codes on parallel computers. There is the potential for considerable savings in computational time. This would negate one of the main complaints in using elliptic methods, that is, the grid generation takes to long.

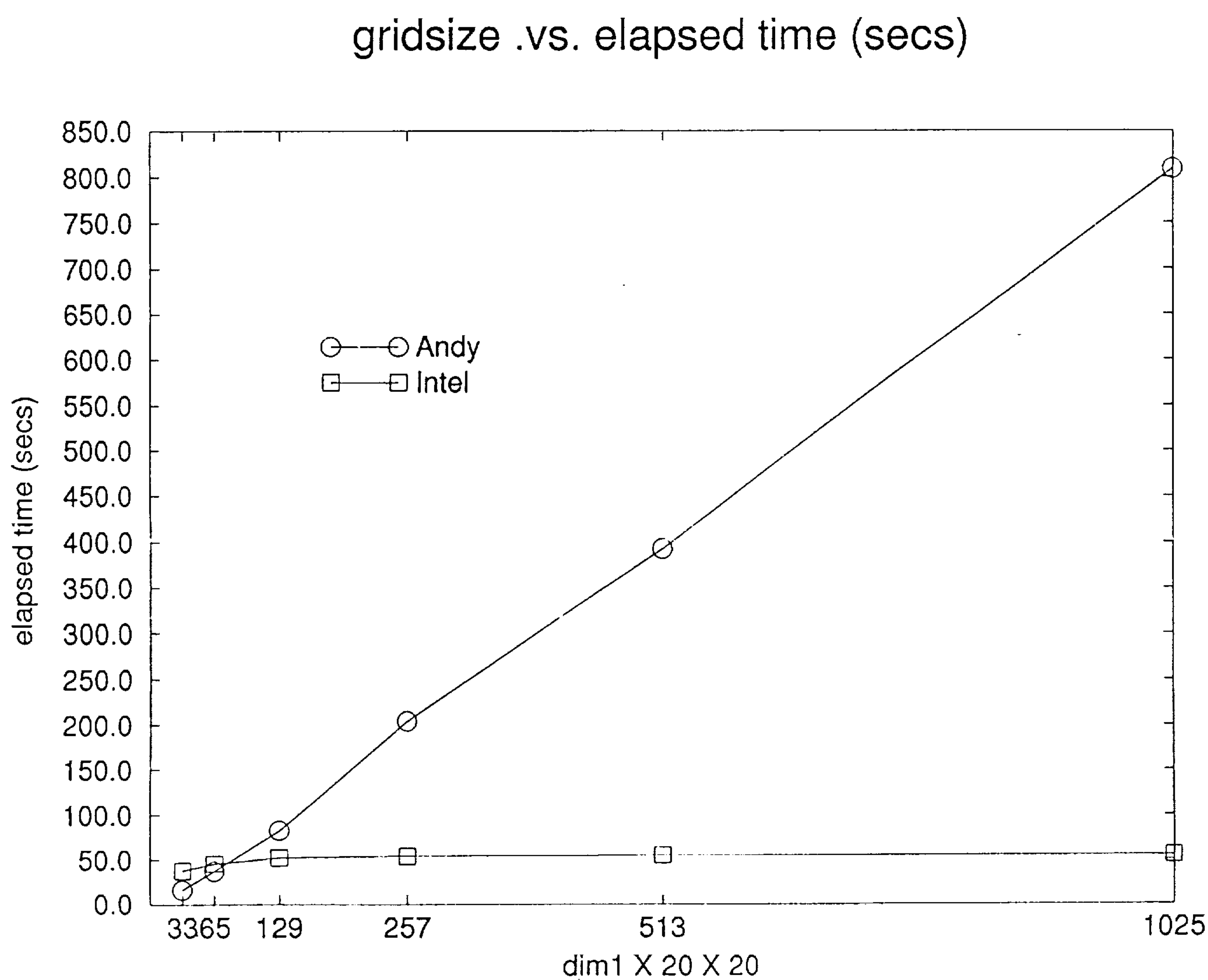

Figure 1: Computational time for 50 iterations on the Intel IPSC/860 and a Silicon Graphics workstation (Andy).