AN ADAPTIVE DIRECT VARIATIONAL GRID GENERATION METHOD

JOSE E. CASTILLO
Department of Mathematical Sciences, San Diego State University, San Diego, CA 92182, U.S.A.

1. SUMMARY

Variational grid generation techniques are now used to produce grids suitable for solving numerical partial differential equations in irregular geometries. Variational grid generation methods are very robust but slow. The method considered here is a discrete variational method. This method preserves the robustness of the variational method; also, it is very fast so it can be used in time-dependent problems. Here we discuss geometry and solution adaption for the direct discrete grid generation method, and some examples are presented to demonstrate its capabilities.

2. INTRODUCTION

In the variational methods introduced by Steinberg and Roache [13], similar to those introduced by Brackbill and Saltzman [2], three functionals are presented that provide a measure of spacing between the grid lines (smoothness), a measure of the area of the grid cells, and the orthogonality of the grid lines. The minimization problem is usually solved by calculating the Euler-Lagrange (E-L) equations for the variational problem; a grid is created by solving a centered finite difference approximation of these equations. In theory, a straightforward discretization of the integrals should provide a similar solution; instead, there are serious difficulties [3]. In this approach, the derivatives in the integrals are replaced by centered finite differences and the integrals are replaced by summations over the grid points. Only first derivatives of the coordinates appear in the direct minimization problem, so a centered finite difference discretization at a grid point does not involve values at that point. This produces strong decoupling problems (see [11]) for the direct approach that are not shown in the E-L approach.

In the E-L formulation [13], there are certain integral constraints on the solution that are automatically satisfied. In the straightforward direct formulation, the analog of these constraints is not automatically satisfied. The straightforward discretization approach transforms the smoothness integral into a linear minimization problem with a linear constraint, while the area integral is transformed into a nonlinear minimization problem with a nonlinear constraint [3]. Such problems are much harder to solve than the unconstrained problems that occur in the continuous cases. A better formulation, called the "direct" formulation, of the variational grid generation method is obtained when the properties to be controlled are derived directly from the discrete geometry [4,5]. This method is described below. There have been other efforts in generating grids by the optimization of direct properties (see Kennon and Dulikravich [11]); however, the functionals for the method presented here, as well as its properties and the minimization procedure used, differ considerably from theirs.

As in the E-L approach, the direct approach controls three properties of the grid: grid spacing, grid cell areas, and grid orthogonality. The grid spacing and cell area functionals have been studied in [5,6] and the orthogonality functional has been discussed in [8]. Here the effects of a reference grid for adaptivity will be studied. A reference grid is a grid which is simpler than the physical grid but with the same properties. The way our codes work is by specifying the desired properties on the reference grid, and the method produces a grid on the physical region.

Helpful discussions with Professor Stanly Steinberg and Dr. Lars Kai Hansen are greatly appreciated.

Typeset by AASTeX
with the same characteristics as the reference grid [7,13]. To make this paper self-contained, all three functionals will be described below; more detail and analysis can be found in [5,6]. In order to understand the behavior of the direct variational method, it is important to understand the behavior of each functional separately. A good understanding of the solution of each minimization problem will provide information relevant to the more general minimization problem that is being considered, i.e., a weighted combination of the three functionals. First an intuitive description of the functionals will be given, followed by the detailed notation and formulas.

3. DIRECT VARIATIONAL FORMULATION IN 2-D

The simplest length control is given by trying, in a variational sense, to make the grid segments equal. To do this, the sum of the squares of the segment lengths between the grid points should be minimized: Let \( s_{ij} \) be the length between the \((i,j)\) grid point and any neighboring grid point, then

\[
\text{minimize } F_S = \sum s_{ij}^2 \quad (1)
\]

with the constraint, see [13],

\[
C_S = \sum s_{ij} = \text{constant} \quad (2)
\]

For controlling the area of the cells, the sum of the squares of the true discrete area of the quadrilateral cell should be minimized: Let \( A_{ij} \) be the area of the \((i,j)\) grid cell and then

\[
\text{minimize } F_A = \sum A_{ij}^2 \quad (3)
\]

with the constraint

\[
C_A = \sum A_{ij} = \text{constant} \quad (4)
\]

For controlling the angles between grid lines, the following functional should be minimized: Let \( O_{ij} \) be the dot product of two vectors with origin at the \((i,j)\) grid point and then

\[
\text{minimize } F_O = \sum O_{ij}^2 \quad (5)
\]

with no constraint.

It is worth noting that in this direct variational approach for spacing between the grid lines, the constraint is automatically satisfied since the sum of all the segments is a telescopic sum which depends only on the values on the boundary (see [1,5,6]). For the area sum, the constraint is the sum of the areas of the true quadrilateral cells which is shown to be the total area of the region, and solely depends on the values on the boundary (see [1,5,6]). There is no constraint for the orthogonality functional [13] (and one is not needed).

To control all three properties, a weighted combination of all the sums is to be minimized:

\[
F = aF_S + bF_A + cF_O \quad (6)
\]

where \( a, b, \) and \( c \) are given numbers such that

\[
a + b + c = 1, a \geq 0, b \geq 0, c \geq 0 \quad (7)
\]

4. NOTATION

The following notation is used (see [1,4,5]). A given region \( \Omega \subset \mathbb{R}^2 \) is polygonal if the boundary of \( \Omega \) is the union of simple closed polygons. A grid on a polygonal region \( \Omega \) is a subdivision of \( \Omega \) into quadrilaterals, the vertices of the quadrilaterals are called the points of the grid, and the quadrilaterals are called the cells of the grid. The region will have \( m + 2 \) points in the logical "horizontal" direction \((m \text{ interior points })\) and \( n + 2 \) points in the logical "vertical" direction \((n \text{ interior})\) points; hence, the grid has \( mn \) interior points. Let \( P_{ij} = (x_{ij}, y_{ij})^t, 2 \leq i \leq m, \ldots \)
2 ≤ j ≤ n be the (i, j) point. A column vector of all the 2mn coordinates of the mn interior points is needed, so let z be the column vector formed with the coordinates of the interior points; i.e., if \( P_{r,s} = (x_{r,s}, y_{r,s}) \), then
\[
z^t = (x_{2,2}, y_{2,2}, x_{2,3}, y_{2,3}, \ldots, x_{m+1,n+1}, y_{m+1,n+1})
\]
and \( z \) is of order 2mn. The grid has \( m + 2 \) points on each "horizontal" boundary and \( n + 2 \) points on each "vertical" boundary. The points on the "horizontal" boundaries are
\[
\{P_{1,1}, P_{2,1}, \ldots, P_{m+1,1}, P_{m+2,1}\}
\]
and
\[
\{P_{1,m+2}, P_{2,m+2}, \ldots, P_{m+1,n+2}, P_{m+2,n+2}\}
\]
Similarly, the points on the "vertical" boundary are
\[
\{P_{1,1}, P_{1,2}, \ldots, P_{1,n+1}, P_{1,n+2}\}
\]
and
\[
\{P_{m+2,1}, P_{m+2,2}, \ldots, P_{m+2,n+1}, P_{m+2,n+2}\}
\]
The mn interior points are
\[
\{P_{2,2}, P_{2,3}, \ldots, P_{2,n+1}, \}
\]
\[
P_{3,2}, P_{3,3}, \ldots, P_{3,n+1}, \ldots,
\]
\[
P_{m+1,2}, P_{m+1,3}, \ldots, P_{m+1,n+1}\}
\]
4.1. Grid Spacing Control

In order to control the lengths of the logical "horizontal" segments between the grid points, consider the functional \( S_H \) given by
\[
S_H = \sum_{i=1}^{m+1} \sum_{j=2}^{n+1} \ell_{ij}^2, \quad \ell_{ij} = \|P_{i+1,j} - P_{i,j}\|_2
\]
where \( \ell_{ij} \) are the lengths of the logical "horizontal" segments. Similarly, consider the functional \( S_V \) given by
\[
S_V = \sum_{i=2}^{m+1} \sum_{j=1}^{n+1} \ell_{ij}^2, \quad \ell_{ij} = \|P_{i,j+1} - P_{ij}\|_2
\]
which allows the lengths of the "vertical" segments between the points of the grid to be controlled. Hence, the functional \( F_S \) can be written
\[
F_S(z) = S_H(z) + S_V(z)
\]
4.2. Functional \( F_A \) for Area Control

Let \( F_A \) denote the sum of the squares of the area of the grid cells; i.e.,
\[
F_A = \sum_{i=1}^{m+1} \sum_{j=1}^{n+1} A_{ij}^2
\]
where \( A_{ij} \) is the area of the \((i,j)\)th cell. \( F_A \) will permit control of the area of the cells. In order to be precise, the \((i,j)\)th cell of the grid is the "oriented quadrilateral," \( P_{i,j}, P_{i+1,j}, P_{i+1,j+1}, P_{i,j+1} \). It is important to notice that there are \((m+1)(n+1)\) areas and that there are \( mn \) interior points in the grid; that is, there are \( 2mn \) unknown grid coordinates. Let a be the column vector of order \((m+1)(n+1)\) whose components are the areas of the cells of the grid; i.e.,
\[
a^t = (A_{1,1}, A_{1,2}, \ldots, A_{1,n+1}, A_{2,1}, A_{2,2}, \ldots, A_{2,n+1}, \ldots, A_{m+1,1}, \ldots, A_{m+1,n+1})
\]
where
\[
A_{ij} = (1/2) \text{det}(P_{ij} - P_{i+1,j+1}, P_{i+1,j} - P_{i,j+1})
\]
then, the functional can be written
\[
F_A = \|a\|_2^2
\]
4.3. Functional $F_O$ for Orthogonality Control

There are four angles in each grid cell: upper right, upper left, lower left, and lower right. In order to control the orthogonality of the logical "upper right" angles between the grid lines, consider the functional $O_{UR}$ given by

$$O_{UR} = \sum_i \sum_j O_{ij}^2, \quad O_{ij} = (P_{i+1,j} - P_{ij})(P_{i,j-1} - P_{ij})$$  \hspace{1cm} (13)

where $O_{ij}$ corresponds to the logical "upper right" angles. In order to control the orthogonality of the logical "lower left" angles, consider the functional $O_{LL}$ given by

$$O_{LL} = \sum_i \sum_j \tilde{O}_{ij}, \quad \tilde{O}_{ij} = (P_{i-1,j} - P_{ij})(P_{i,j-1} - P_{ij})$$  \hspace{1cm} (14)

There are similar functionals for the upper left ($O_{UL}$) and lower right angles ($O_{LR}$). The functional $F_O$ can be written

$$F_O = O_{UR} + O_{UL} + O_{LL} + O_{LR}$$  \hspace{1cm} (15)

This is a family of functionals which gives angle control; it can be opted to control the upper, lower interior, and the boundary angles. These functionals allow a great deal of flexibility; it can be decided in advance, based on the physical region, which angles are going to need more control.

5. ADAPTIVITY

The adaptivity of the method is done by means of a reference grid — this can be thought of as weights — to generate the grid points on located places on the physical region and a weight function with information from the governing equations to move the solution at each iteration. Here, for the purpose of demonstrating the adaptivity of the grid, we use the weights from the reference grid. Notice that the weights from the physical solution can also be used for solution adaption; the concept is basically the same except the weights would change at each iteration. The reference grid can be thought of as weights on the length and area functional. This will produce grids with points distribution accordingly to the reference grid.

The weight function used for this example was

$$U(x, y) = \cosh^2(a(x - x_0)^2 + b(y - y_0)^2)^{1/2}$$  \hspace{1cm} (16)

The adaption is also a global adaption in the sense that we do not refine the grid locally but instead redistribute the grid points using the physical weights. When using this approach, one does not have to interpolate on the grid to patch pieces of the solution, which is something one has to do when the grid is refine locally.

6. MINIMIZATION PROCEDURE

The minimization problem associated with the direct variational method can be solved by a nonlinear conjugate gradient method (see [1,6,10,12]); the problem can be posed as a least squares problem. For each functional, or for a combination of them, the number of variables becomes large very rapidly. Consequently, most standard solvers will have difficulty with these problems. In addition, any solver that stores the full gradient or Hessian will have serious storage limitations.

It is important to note here that once the value of either the length, area, or the orthogonality functional has been computed, all of the information needed to compute the gradient and the Hessian has been obtained (see [1,4,5]). In addition, the Hessian is very sparse and the nonzero entries are easily located [4]. These are key factors for a preconditioned conjugate gradient or a truncated Newton implementation. Codes are under development for generating two- and three-dimensional grids that take full advantage of the theory presented in this paper, and it is expected that these codes will be competitive with other grid generation codes.
7. PERFORMANCE

Many examples have been generated using this code (see Figs. 1 through 4). The performance has been satisfactory, giving a substantial check on the method presented in this paper. In addition, the generated grids have been compared to those computed by Steinberg and Roache [13]. An appropriate combination of the two functionals, length and area, produces grids suitable for numerical calculations; however, the orthogonality functional can be used to improve the quality of the grid angles [8]. The capability of controlling interior and/or boundary angles gives the direct method a greater flexibility and generality. The adaptivity in this examples is done by means of a reference grid [7,13].

Fig. 1

Fig. 2

A sample of the test cases is presented in Figures 1 through 4. A particularly difficult geometry is considered. In Figures 1 through 4, the length weight is .087 and the orthogonality weight is .113, the effects of concentration of the points at different parts of the region is achieve by specifying a reference grid, also a weigh function with information obtained from the governing equations can be used at the area functional for solution adaption.
To check how fast our method is for a model problem (see Figures 1 through 4), a version of the method (length and area control only) was tested against an implementation of the homogeneous Thames, Thompson, and Mastin method [14] or Winslow method [15], which solves a coupled system of elliptic equations. In the direct method, the weight used for the length functional was 0.1. The parameters for the minimization part were 1e-5 for the gradient and the line search [12]; the tolerance used for the TTM was 1e-4; also, the use of 1 SOR iteration or 2 SOR iteration per outer (nonlinear) iteration was considered. The number of nonlinear iterations (iter) as well as the time in seconds were presented for three different size grids for the model problem. These tests were done on a Sun-3 workstation. Although none of the methods have been highly tuned, it seems clear that the present method is at least competitive with the most commonly used elliptic grid generation schemes [6]. The cost of adding orthogonality control to the direct method is small; a timing test for the boundary directional angle control was given [8].
8. CONCLUSIONS

The direct variational grid generation method presented here is a robust and efficient method for grid generation. As shown in [6], area control by itself is not adequate; a proper combination of area and length control has the capability of generating grids suitable for numerical calculations. The capability of these methods is greatly enhanced by the orthogonality functional [8]. It is clear that the method is geometry and solution adaptive, as has been demonstrated by the examples.

Work is proceeding on the extension of the method to curves and surfaces in three dimensions; these problems will have a structure analogous to the two-dimensional problem. Thus, it is reasonable to expect that the three-dimensional method will also be competitive with other standard grid generation methods. Also, there is current work on a Conjugate Gradient implementation
that is expected to be faster, a truncated Newton implementation to take advantage of the structure of the Hessian, and a parallel conjugate gradient implementation. These will clearly enhance the capability of the direct grid generation method.

REFERENCES

1. P. Barrera-Sanchez and J.E. Castillo, A large scale optimization problem arising from numerical grid generation, Technical Report, Department of Mathematics and Statistics, University of New Mexico.