Interactive method of variational grid generation

A. ZOCHOWSKI and P. HOLNICKI
Systems Research Institute of the Polish Academy of Sciences, Newelska Str. 6, Warsaw, Poland

Received 25 April 1988
Revised 24 January 1989

Abstract: In the paper the elliptic variational grid generation technique based on methods of optimal control for distributed systems is presented. The interactive nature of the method consists in adaptive changing of the goals of correction steps in the process of grid generation. The corrections concern orthogonality and uniformity of the mesh.

Keywords: Elliptic grid generation, control theory, finite-element method.

1. Introduction

Any grid generation method consists in finding a transformation between physical region $\Omega$ and some topologically simple set $D$ (usually rectangle) which we shall call the reference region. The nodes of a given straight mesh on $D$ correspond to the nodes of the grid in $\Omega$. Such a transformation may be considered a parametrization of $\Omega$.

The curvilinear cells in $\Omega$ are useful for approximation methods if they fulfil a few conditions, such as: smoothness, nonsingularity and orthogonality. Nonsingularity means here that the Jacobian of the transformation should not vanish; it would be the best if all the cells in $\Omega$ had roughly equal areas. Smoothness is related to the regularity of the transformation between $D$ and $\Omega$ and orthogonality is expressed as the desire to keep the angles of cells in $\Omega$ as close to $\frac{1}{2}\pi$ as possible.

Various ways of generating grids address some aspects of these requirements. The conformal method (see, e.g., [2,4]) preserves orthogonality and the same may be said about orthogonal method [3]: the algebraic method uses to this aim free coefficients in polynomials defining the grid lines (see [2,7]).

The most comprehensive approach is connected with elliptic grid generation methods [7]. They are based on the observation, that the transformation $\Phi: D \rightarrow \Omega$ must fulfil boundary conditions such that $\Phi: \partial D \rightarrow \partial \Omega$. This corresponds to the classical Dirichlet problem for partial differential equations.

The elliptic methods contain, as subfamily, the variational techniques. They are based on the observation that orthogonality, nonsingularity as well as smoothness may be expressed as functionals depending on the field $\Phi$. It is therefore possible to construct the global functional
being a linear combination of those three, and then using the Euler–Lagrange technique to
obtain a partial differential equation corresponding to the necessary conditions of minimization
(see the review [5]). To the same group belongs the recently proposed reference grid method [1].

The variational method has some disadvantages. If one tries to minimize the quality functional
exactly, the resulting differential equation is highly nonlinear. Even taking into account that it
must be solved on a simple and regular region $D$, this necessitates using difficult techniques. The
control over the generated grid is also very indirect, via coefficients in a quality functional.

In this paper we propose a version of the variational method which, in our view, alleviates
some of these drawbacks.

2. Grid generation as control problem

Let $\Phi$ be the transformation of $D \subset \mathbb{R}^2$ onto $\Omega \subset \mathbb{R}^2$:

$$\Phi (\xi, \eta) = (X(\xi, \eta), Y(\xi, \eta)).$$

(1)

It must satisfy the relation

$$\Phi|_{\partial D} = (f_1(\xi, \eta), f_2(\xi, \eta)),$$

(2)

where

$$(f_1, f_2): \partial D \to \partial \Omega.$$  

(3)

It is well known (see, e.g., [7]) that the quality of transformation $\Phi$ may be assessed by means
of functionals of:

smoothness

$$I_s = \int_D (\| \nabla X \|^2 + \| \nabla Y \|^2) \, dS,$$ 

(4)

orthogonality

$$I_o = \int_D (X_\xi X_\eta + Y_\xi Y_\eta) \, dS,$$ 

(5)

nonsingularity or uniformity

$$I_v = \int_D [J(X, Y)]^2 w(\xi, \eta) \, dS,$$ 

(6)

where

$$J(X, Y) = \frac{\partial(X, Y)}{\partial(\xi, \eta)}.$$ 

The presence of function $w(\xi, \eta)$ in (6) requires some comment. Since

$$\int_D J(X, Y) \, dS = |\Omega|,$$

the functional $I_v$ is minimal if $wJ = \text{constant}$ on $D$. This implies that $J \approx 1/w$. The sizes of cells
in $\Omega$ are proportional to local value of $J(X, Y)$. Hence the mesh is locally denser where the value
of \( w \) is bigger. In this way the weight function \( w \) allows to control the size of mesh inside \( \Omega \).

Let us introduce the total functional
\[
I_T = \lambda_s I_s + \lambda_o I_o + \lambda_v I_v.
\]

Let \( X - X_0 + \epsilon X', \ Y - Y_0 + \epsilon Y' \), when \( X' - Y' = 0 \) on \( \partial D \) and \( X_0 - f_1, Y_0 - f_2 \) on \( \partial D \). Neglecting the higher order terms in \( \epsilon \) and integrating by parts one obtains the conditions for vanishing of the first variation of functional \( I_T \), or the necessary conditions for optimization:
\[
\begin{align*}
\Delta X_0 &= \frac{\lambda_o}{\lambda_s} \nabla \left[ (\nabla X_0 \cdot \nabla Y_0) \nabla Y_0 \right] + \frac{\lambda_v}{2\lambda_s} \left[ \frac{\partial}{\partial \xi} \left( w \frac{\partial Y_0}{\partial \eta} \right) - \frac{\partial}{\partial \eta} \left( w \frac{\partial Y_0}{\partial \xi} \right) \right], \\
\Delta Y_0 &= \frac{\lambda_o}{\lambda_s} \nabla \left[ (\nabla X_0 \cdot \nabla Y_0) \nabla X_0 \right] + \frac{\lambda_v}{2\lambda_s} \left[ \frac{\partial}{\partial \eta} \left( w \frac{\partial X_0}{\partial \xi} \right) - \frac{\partial}{\partial \xi} \left( w \frac{\partial X_0}{\partial \eta} \right) \right].
\end{align*}
\]

In this way we have obtained the system of coupled 4th-order elliptic equations. Besides being very difficult to solve, they require additional boundary conditions, which must be chosen arbitrarily.

Let us notice that (8), (9) may be rewritten in a form
\[
\Delta X_0 = u(X_0, Y_0), \quad \Delta Y_0 = v(X_0, Y_0).
\]

One of the prices we pay for exact minimization lies in the fact that \( u \) and \( v \), taken as functions of \((\xi, \eta)\), are not independent.

In this paper we propose to relax this requirement without abandoning the minimization of some functional. Namely we assume that smoothness is sufficiently assured by \( X \) and \( Y \) being the solutions of elliptic equations and try to optimize the orthogonality and uniformity of the mesh. The price for decoupling the system (8), (9) is the necessity to use iterative optimization techniques. However, as we shall see in examples, this allows to introduce the interactivity into the process of mesh generation and may be considered an advantage.

The new variational mesh generation problem has the following shape:

minimize \( I(u, v) = \lambda_v I_v + \lambda_o I_o \),

subject to constraints in the form of the state equations
\[
\begin{align*}
\Delta X &= u(\xi, \eta) \text{ in } D, \quad \Delta Y = v(\xi, \eta) \text{ in } D, \\
X &= f_1 \quad \text{on } \partial D, \quad Y = f_2 \quad \text{on } \partial D.
\end{align*}
\]

In order to use iterative minimization techniques, we must compute variations of \( I \) corresponding to variations \( u \to u + u', v \to v + v' \). It may be done in a standard way using the adjoint equations. It is more convenient to express them in the weak form. Let these adjoint variables be denoted \( P \) and \( Q \). They are solutions of the following problem:

\[ \text{find } P, Q \in H_0^1(D) \text{ such that for any } \phi \in H_0^1(D):
\]
\[
\int_D \nabla P \nabla \phi \, dS = \lambda_v \int_D \left( \frac{\partial \phi}{\partial \xi} \frac{\partial Y}{\partial \eta} - \frac{\partial \phi}{\partial \eta} \frac{\partial Y}{\partial \xi} \right) w \, dS
\]
\[
+ 2\lambda_o \int_D (\nabla X \cdot \nabla Y) \left( \frac{\partial \phi}{\partial \xi} \frac{\partial Y}{\partial \xi} + \frac{\partial \phi}{\partial \eta} \frac{\partial Y}{\partial \eta} \right) \, dS.
\]
Solving of (13) and (14) is facilitated by the fact that right-hand sides contain only first-order derivatives of $X$, $Y$ and $\Phi$. It means that in a case of linear finite-element approximation such expressions are constant on triangles constituting the discretization of the rectangle $D$. Let $T$ be a fixed triangle with vertices $T_1, T_2, T_3$. Then any linear function $f$ on $T$ is uniquely defined by a vector of its values in vertices, $\bar{f} = [f(T_1), f(T_2), f(T_3)]^T$. The gradient of $f$ may be obtained from

$$\frac{\partial f}{\partial \xi} = \bar{G}_\xi \cdot \bar{f}, \quad \frac{\partial f}{\partial \eta} = \bar{G}_\eta \cdot \bar{f},$$

where $\bar{G}_\xi, \bar{G}_\eta$ are constant for a given triangle row vector.

Let in addition

$$G = \begin{bmatrix} \bar{G}_\xi \\ \bar{G}_\eta \end{bmatrix}$$

be a $2 \times 3$ constant matrix.

Given these well-known, in the framework of finite-element method, formulae, one immediately obtains for (13):

$$(\nabla P \cdot \nabla \Phi) = \bar{P}^T G^T G \bar{\Phi}, \quad (\nabla X \cdot \nabla Y) = \bar{X}^T G^T G \bar{Y},$$

$$\frac{\partial \Phi}{\partial \xi} \frac{\partial Y}{\partial \eta} - \frac{\partial \Phi}{\partial \eta} \frac{\partial Y}{\partial \xi} = \bar{Y}^T [\bar{G}_\eta \bar{G}_\xi - \bar{G}_\xi \bar{G}_\eta] \bar{\Phi},$$

$$\frac{\partial \Phi}{\partial \xi} \frac{\partial Y}{\partial \xi} + \frac{\partial \Phi}{\partial \eta} \frac{\partial Y}{\partial \eta} = \bar{Y}^T [\bar{G}_\xi \bar{G}_\xi + \bar{G}_\eta \bar{G}_\eta] \bar{\Phi},$$

and similarly for (14). All these expressions are constant in $T$.

Taking into account that $G^T G$ constitutes a stiffness matrix over $T$, so it must be calculated anyway, it is evident that obtaining right-hand sides for (13) and (14) does not add significantly to the computing effort in comparison to simple equations (12).

Then by standard arguments,

$$\delta I(u, v; \ u', v') = \int_D (u'P + v'Q) \, dS.$$  \hspace{1cm} (15)

Hence, the correction to $(u, v)$ may have the form

$$u' = -\tau P, \quad v' = -\tau Q,$$

where $\tau$ is the step length.

The final iterative procedure is expressed as follows:

(i) set $u = v = 0$,

(ii) solve (12), (13) and (14) for $(X, Y)$ and $(P, Q)$,

(iii) assess the mesh obtained,
(iv) if necessary, make corrections
\[ u := u + u', \quad v := v + v' \]
and return to (ii), else stop.

If the steps (iii) and (iv) were performed automatically, such a procedure would simply constitute a primitive minimization algorithm. However, since the assessment is performed by the human, it may be quite different. Let us notice that from linearity of (13) and (14) follow the relations
\[ P = \lambda_v P_v + \lambda_o P_o, \quad Q = \lambda_v Q_v + \lambda_o Q_o \]
and the corrections may be split in a similar way
\[ u' = - (\tau \lambda_v) P_v - (\tau \lambda_o) P_o, \quad v' = - (\tau \lambda_v) Q_v - (\tau \lambda_o) Q_o. \]

Having examined the mesh carefully in step (iv), the operator may decide which aspect needs improvement, and then set new values of \( \lambda_v \) and \( \lambda_o \) accordingly. After gaining some experience, he should obtain good meshes in one or two iterations.

Summing up, the interactive nature of grid generation procedure proposed in this paper consists in adaptive changing the goals of correction steps.

The method has been implemented on an IBM-PC/AT compatible computer. Despite the complicated appearance of (13) and (14), solving them as well as solving the state equation (12) is very fast due to the fact that they have the same stiffness matrix, constant throughout the iteration process, and all have similar Dirichlet boundary conditions.

At the end we shall discuss the form of the weight function \( w(\xi, \eta) \). In the majority of applications its value would be \( w = 1 \). In some cases, when we want to make the grid denser in the neighbourhood of some points \( (\xi_i, \eta_i), \ i = 1, \ldots, n \), we assume the formula
\[ w(\xi, \eta) = \sum_{i=1}^{n} \exp\left[ -r \left( |\xi - \xi_i|^2 + |\eta - \eta_i|^2 \right) \right]. \]

We call these points \( (\xi_i, \eta_i) \in D \) (or respectively \( (X(\xi_i, \eta_i), Y(\xi_i, \eta_i)) \in \Omega \)) the attractors.

3. Implementation and examples

The reference region used in our implementation was rectangular, with sides of \( n \) and \( m \) units, and was divided into \( n \cdot m \) squares. The parameters \( n \) and \( m \) were chosen individually for each physical region, according to its topological properties. The fact that cells in \( D \) had fixed size, independent of \( \Omega \), allowed us to standardize the values of parameters \( \lambda_v \) and \( \lambda_o \), so that in most cases the operators can safely choose parameters in the ranges:
\[ 0 \leq \lambda_v \leq 1, \quad 0 \leq \lambda_o \leq 1. \]

However this is not a rigid rule. Every iteration of the algorithm amounts to one step in the direction of steepest descent for the functional
\[ F = \hat{\lambda}_v I_v + \hat{\lambda}_o I_o \]
with chosen \( \hat{\lambda}_v, \hat{\lambda}_o \). The values \( \hat{\lambda}_v \) and \( \hat{\lambda}_o \) may change from one iteration to another. Their ratio reflects the relative importance of uniformity or orthogonality, while the absolute values
constitute a coefficient by which the gradient of $I$ is multiplied in order to obtain the step-length in the direction of descent. So in choosing these absolute values one can use intuition connected with minimizing single-variable functions. For example, if uniformity is important ($\lambda_o = 0$) and the current mesh is far from uniform, it is probable that the gradient is very big. So the multiplier $\lambda_v$ should be small (even 0.001 or smaller in extreme cases). And vice versa: if the grid is nearly uniform, then the gradient is small and $\lambda_v$ may be bigger, even greater than 1 (but then further improvement is unnecessary anyway). Generally, too big a step may cause overshooting the optimum, or in our case overcorrection.

The same concerns the parameter $r$ in the function $w$ which defines the range of influence of the attractor. In fact it has been fixed in our version of the program.

The action of the algorithm may be illustrated by two examples. In the first, the square $\Omega$ has an attractor in the center. The mesh after one correction ($\lambda_v = 0.5$, $\lambda_o = 0$) is shown in Fig. 1. The second example illustrates the effect of orthogonality correction, also after one step ($\lambda_v = 0$, $\lambda_o = 0.7$), see Fig. 2.

The more complicated meshes may be obtained by the well-known method of glueing part of the reference rectangle $D$. For example by equating the opposite sides one gets the region with the topological characteristics of the ring. Several glueings give multiple-connected regions. Such an example, after one uniformity correction ($\lambda_v = 0.5$) is shown in Fig. 3. It also demonstrates the applicability of the procedure to more realistic problems.

Finally, we give the results of applying the procedure to the test region between two ellipses, as discussed in [1]. Figure 4 shows the mesh before corrections ($u = v = 0$), after one uniformity correction ($\lambda_v = 0.5$) and after subsequent orthogonality correction ($\lambda_o = 0.5$).

In the examples listed above we have used the iteration steps with clearly separated uniformity and orthogonality corrections, so only one parameter of the pair $\lambda_v$, $\lambda_o$ is positive at any time. However, this is not necessary. In fact, it is advantageous to put a small value for the other parameters in order to prevent overshooting the optimum.

---

Fig. 1. The effect of the attractor.

Fig. 2. The effect of orthogonality correction.
parameter (the less important one), because this regularizes the calculation correction, when adding some uniformity component (e.g., $\lambda_O = 0.5$, $\lambda_V = 0.1$) is usually better than $\lambda_O = 0.5$, $\lambda_V = 0$.

4. Conclusions

The procedure described above has been tested on many examples. It is being now implemented for 3D case. There exists also many possibilities of development within the framework of the iterative algorithms described above like, e.g., step by step grid construction, beginning from 1D lines (also falling into this framework) through 2D faces to 3D solids.

References