ELSEVIER

Mathematics and Computers in Simulation 44 (1997) 441–456

MATHEMATICS AND COMPUTERS IN SIMULATION

A generalized length strategy for direct optimization in planar grid generation

Jose E. Castillo[1,a,*], James S. Otto[2,b]

[a] Department of Mathematics, San Diego State University, San Diego, CA 92812-0314, USA
[b] Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Accepted 14 August 1997

Abstract

We consider an alternative to conventional grid generation strategies based on the length functional. These conventional strategies include using an inverse mapping (the famous Winslow generator), or augmenting length with other functionals that promote unicity. Both strategies have the drawback that the resulting minimization problem becomes complicated and expensive. As an alternative, we propose a generalized strategy for length which does not use inverse mappings or auxiliary functionals. The strategy, based largely on reference mappings, provides flexibility in controlling grid quality, and its minimization problems can be solved using a simple multigrid algorithm, yielding a grid generation scheme with optimal complexity.

1. Introduction

Length is undoubtedly the most important functional in variational grid generation, producing smooth grids and ensuring well-posedness of the minimization problem when combined with more unruly functionals. Unfortunately, in its most primary form, length can produce folded grids when used with nonconvex geometries. Historically, there have been two solutions to this dilemma: using an inverse mapping (the famous Winslow generator), or augmenting length with other functional that promote unicity. Both strategies have the drawback that the resulting minimization problem becomes complicated and expensive: inverse mapping gives rise to a coupled, nonlinear, system of equations, while the functionals with which length is usually augmented correspond to nonelliptic and ill-conditioned systems. Also, in three dimensions, the strategy of using an inverse does not always ensure unicity. As another alternative, we propose a generalized method for length based on the reference grid

* Corresponding author. E-mail: castillo@myth.sdsu.edu
[1]This work was supported in part by a AWU/DOE Fellowship.
[2]E-mail: jotto@cs.unm.edu. This work was supported by Los Alamos National Laboratory and the Department of Energy.

PII S0378-4754(97)00079-7

approach introduced in [5]. This method does not use inverse mappings or auxiliary functionals. It provides flexibility in controlling grid quality, and its minimization problems can be solved using a simple multigrid algorithm, yielding a grid generation scheme with optimal complexity. It also extends trivially to three dimensions. This method can be viewed as solving a discretized form of the *uncoupled* and *linear* system

$$\begin{aligned} -(Ax_{ss} + Bx_{tt}) &= f \\ -(Cy_{ss} + Dy_{tt}) &= g \end{aligned} \tag{1}$$

where the coefficient functions are determined *dynamically*. That is, the proposed method provides strategies for generating these functions automatically, and in such a way that the desired grid properties are promoted in the resulting solution.

A summary of the rest of this paper is as follows. Section 2 is a brief review of variational grid generation which serves as a point of reference for the method we propose. Section 3 is an introduction to the method of direct optimization, which provides a framework for talking about reference grids. The use of reference grids is a key element in the proposed method. In Section 4, the relationship between reference grids, elliptic grid generators, and interpolation is explored. In Section 5, this relationship is exploited to give a preliminary grid generation algorithm. Additional controls are added to this algorithm in Section 6, giving a generalized length strategy with the desired properties. Section 7 provide a summary and acknowledgements.

2. Variational grid generation

Let the *physical domain*, Ω, be a region in the x, y-plane, and let the *logical domain*, Θ, be a rectangular region in the s, t-plane. Grid generators seek a bijective mapping

$$\begin{bmatrix} x(s,t) \\ y(s,t) \end{bmatrix} : \bar{\Theta} \to \bar{\Omega} \tag{2}$$

of the logical to the physical domain.

In practice, the class of variational grid generators proceed by associating portions of the physical boundary with the sides of the logical domain, as in Fig. 1, and then extending this mapping to the

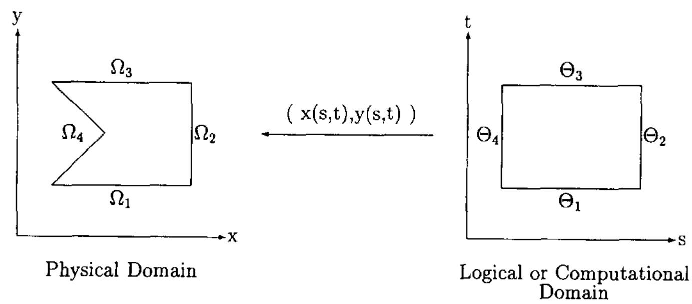

Fig. 1. Physical and logical domains.

interior of the domains by the minimization of a functional constrained by the boundary conditions, [3,10].

Suppose, for example, that the logical domain is the unit square. Then a canonical variational principle is obtained from the functional,

$$F(x,y) = \frac{1}{2}\int_0^1\int_0^1 (x_s^2 + y_s^2 + x_t^2 + y_t^2)\mathrm{d}s\mathrm{d}t \tag{3}$$

This variational principle has Euler–Lagrange equations,

$$\begin{aligned} &-(x_{ss} + x_{tt} = 0) \\ &-(y_{ss} + y_{tt}) = 0 \end{aligned} \tag{4}$$

To obtain a grid, one discretizes this system over a tensor-product mesh in the logical domain. This elliptic system represents the canonical grid generator; it is intimately connected to the length or smoothness functional. The latter name stems from the above generator's notorious reputation for producing smooth grids. It has wide applicability for convex physical geometries. Unfortunately, it is also notorious for producing folded grids for nonconvex geometries. Historically, there have been two solutions to this dilemma: using an inverse mapping (the famous Winslow generator [11,12]), or augmenting the functional with others that promote unicity. As pointed out earlier, these strategies have some serious drawbacks: inverse mapping replaces Eq. (4) with a coupled, nonlinear, system, while the functionals with which length is usually augmented correspond to nonelliptic and ill-conditioned systems (this latter topic is considered in some detail in [6]). Also, in three dimensions, the strategy of using an inverse does not always ensure unicity. We propose a generalized strategy for length which does not use inverse mappings or auxiliary functionals. The strategy bypasses the variational formulation – it works implicitly with Eq. (4), replacing it with a discretized form of the related system,

$$\begin{aligned} &-(Ax_{ss} + Bx_{tt}) = f \\ &-(Cy_{ss} + Dy_{tt}) = g \end{aligned} \tag{5}$$

In the best of all possible worlds, we would simply be given a system like Eq. (5) to solve, that would contain the "right" auxiliary functions, A,B,. . ., and g (which depend only on s and t) for our problem. The key to the success of the proposed method is in providing a way to determine the appropriate coefficient functions ourselves. In doing this, the role of Castillo's direct optimization formulation of length [4] is key, since it treats the grid generation problem from the beginning as discrete, and provides a convenient framework for the incorporation of reference mappings.

3. The direct optimization approach

Define discrete variables

$$\mathbf{x} = [x_{ij}]^T \quad \text{and} \quad \mathbf{y} = [y_{ij}^T] \quad \text{for} \quad \begin{matrix} 1 \le i \le I \\ 1 \le j \le J \end{matrix} \tag{6}$$

which are in 1-to-1 association with the nodes of an $I \times J$ rectangular grid as in Fig. 2.

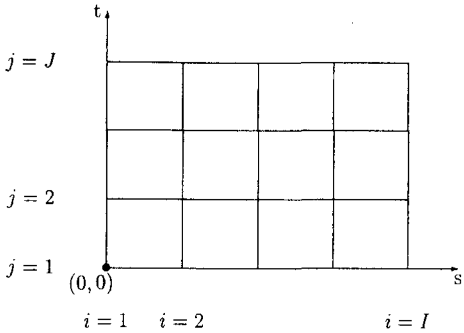

Fig. 2. Logical grid.

Direct optimization begins by assigning boundary values by parameterizing boundary segments. For example, suppose in Fig. 1, the physical boundary Ω_3, is given by

$$\Omega_3(s, t_J) = (x(s, t_J), y(s, t_J)), \quad 0 \leq s \leq b \tag{7}$$

and let

$$(x_{iJ}, y_{iJ}) = \Omega_3(s_i, t_J), \quad 0 = s_1 < \ldots < s_I = b \tag{8}$$

This defines a relation between nodes on the boundaries of the two domains. To complete the relation on the interiors, **x** and **y** are required as before to minimize a functional. The canonical functionals for direct optimization are those which control smoothness, area, and orthogonality. By far, the most important of these is the smoothness or length functional. The latter name comes from the definition of the functional, i.e. as a minimizer of the variation in the lengths of cell edges.

Associate with each "horizontal" edge in the physical domain a length,

$$L_{ij}^h = \left((x_{i+1,j} - x_{ij})^2 + (y_{i+1,j} - y_{ij})^2\right)^{1/2} \tag{9}$$

Do the same for the "vertical" edges,

$$L_{ij}^v = \left((x_{i,j+1} - x_{ij})^2 + (y_{i,j+1} - y_{ij})^2\right)^{1/2} \tag{10}$$

Then the *vanilla length functional* is given by

$$F_L(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{\text{Edges}} (L_{ij}^h)^2 + (L_{ij}^v)^2 \tag{11}$$

Under an appropriate ordering of unknowns (lexicographic in x followed by the same in y), the Hessian, H_L, for length has a 2×2 block structure. The main diagonal blocks are 5-point Laplace operators, while the off-diagonal blocks vanish. These observations indicate that the minimization problem for vanilla length is always well-posed. In addition, the spectral properties of H_L are well known as is the convergence behavior of conjugate gradient and multigrid solution algorithms.

4. Reference grids and elliptic grid generators

Castillo et al. [5] have introduced the concept of a reference grid to provide additional flexibility in constructing useful functionals for variational grid generation.

A rectangular reference grid has the form shown in Fig. 3, where A and B are parameters which are considered to be problem-dependent. This turns out to be a practical example, but there are other useful reference grids. We place no restriction on the form that a reference grid may take; it must merely have the same dimensionality of the logical and physical grids. A reference grid is used to obtain a new functional for direct optimization by modifying the proportionality of the quantities which define a given functional.

For length, we introduce reference lengths, l_{ij}^h and l_{ij}^v, associated with the reference grid, and define,

$$F_L(\mathbf{x},\mathbf{y}) = \frac{1}{2}\sum_{E_{ij}} \frac{(L_{ij}^h)^2}{(l_{ij}^h)^2} + \frac{(L_{ij}^v)^2}{(l_{ij}^v)^2} \tag{12}$$

One attractive feature of the use of a reference grid is that it allows for standardization of the logical grid: we may always take the logical/computational grid to be a unit grid with uniform mesh spacing in each direction. This, in turn, allows for standardization of the algorithm used for minimization. Now, when a rectangular reference grid with uniform spacing is used with a unit/uniform logical grid, it turns out that the Hessian for the length functional corresponds to a standard discretization for the equivalent problems of Fig. 4. We make this point in order to emphasize the connection between the length functional used in conjunction with a reference grid and various elliptic grid generators.

It turns out that the problem of folding associated with the length functional can in many instances be controlled by use of rectangular reference grid. The following grids were obtained by a method described by the foregoing equivalences and using $A = 1$. For a given geometry (i.e. physical domain) then, grids were computed for various values of the parameter B. The first example is for the **S**-geometry. Suppose we use the choice $B = 1$, which corresponds to the use of Laplace's equation as an elliptic generator. It also corresponds to the use of the trivial reference grid (i.e. a square grid with equal spacing), yielding the vanilla length functional. The resulting physical grid is severely folded, as shown in Fig. 5. Now, the initial motivation behind the use of references grids is to allow one to include information about the physical geometry in the grid functional. Suppose we include the aspect ratio of the sides of the physical domain by setting $B = 14$ (somewhat arbitrarily). Then we get the grid in

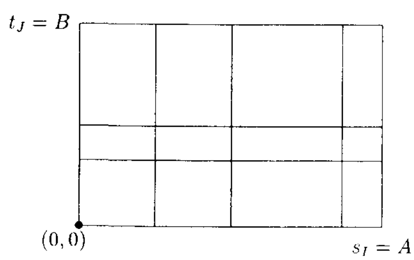

Fig. 3. Rectangular reference grid.

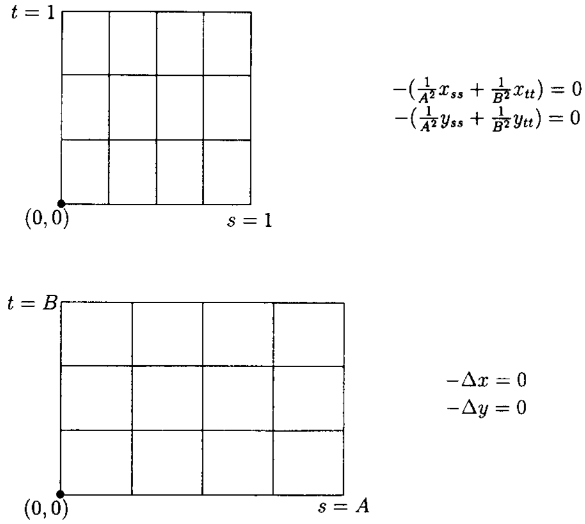

Fig. 4. Equivalent elliptic systems under reference grid.

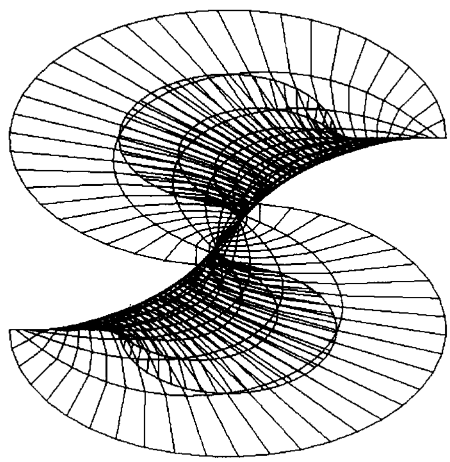

Fig. 5. Vanilla length for the **S**.

Fig. 6. This is a relatively robust procedure in that a suitable parameter may be chosen, by trial and error (we will demonstrate a better way), to produce an unfolded grid in many instances where the vanilla length functional gives a folded grid. In particular, most of the examples of the Rogue's Gallery in [10] can be treated this way. It is not entirely robust, however.

Consider the three-sided valley, or **Fishtail**. This geometry/grid is pictured in Fig. 7, for vanilla length ($B = 1$). One finds that due to the symmetry of the problem no value of B will produce a grid without folding at all of the concave sides. On the other hand, replacing the elliptic system Eq. (4) with

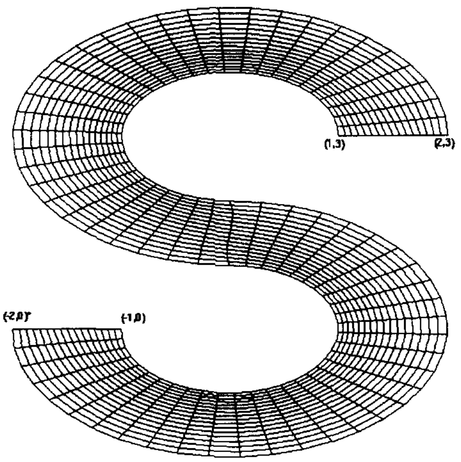

Fig. 6. Length with rectangular reference grid for the **S**.

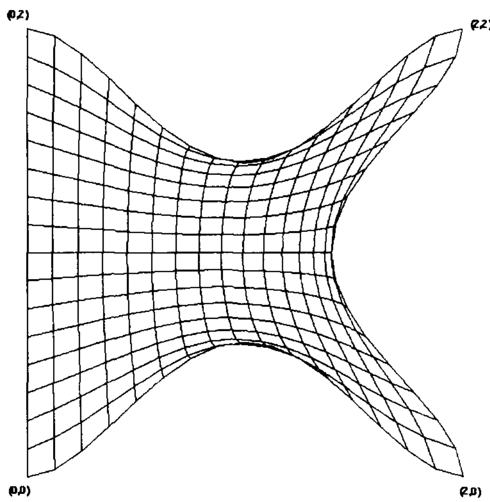

Fig. 7. Vanilla length for the **Fishtail**.

$$-\left(x_{ss}+\frac{1}{B_1^2}x_{tt}\right)=0$$
$$-\left(y_{ss}+\frac{1}{B_2^2}y_{tt}\right)=0 \tag{13}$$

enables the generation of unfolded grids for this example. This system is equivalent to the use of distinct reference grids for each coordinate, and generalizes previous approaches. In particular, the choice $B_1=2$, $B_2=1/2$ yields the grid of Fig. 8. We note that if we allow the reference parameters to converge toward limiting values (i.e. $B_1\rightarrow\infty$ and $B_2\rightarrow 0$) then the associated grids rapidly converge

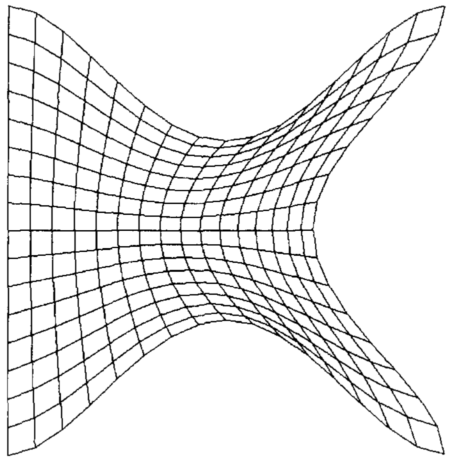

Fig. 8. Distinct reference grids for the **Fishtail**.

to the grid obtained by bilinear interpolation [8] of the boundary values. However, for moderate values of the parameters the resulting grid may have better properties than the interpolation grid. The interpolation grid for the **Fishtail**, for instance, has severe cell compression inside the fins (see Fig. 16). Another example is provided by the **S**. In this case, bilinear interpolation propagates discontinuities at the boundary throughout the interior of the grid. The point to be made here is that the interpolation grid contains information which can be used to control folding.

The use of distinct elliptic PDE's for mappings for x and y corresponds to the use of reference grids in direct optimization. As an example, recall the length functional under a single reference grid, Eq. (12). Its specific form for the case of a rectangular reference grid (and $A = 1$) is

$$F_L(\mathbf{x},\mathbf{y}) = \frac{1}{2}\sum_{E_{ij}} \frac{(x_{i+1,j}-x_{ij})^2 + (y_{i+1,j}-y_{ij})^2}{h_s^2} + \frac{(x_{i,j+1}-x_{ij})^2 + (y_{i,j+1}-y_{ij})^2}{B^2 h_t^2} \tag{14}$$

where $h_s = 1/(I-1)$ and $h_t = 1/(J-1)$. When a pair of rectangular reference grids is used, this becomes

$$F_L(\mathbf{x},\mathbf{y}) = \frac{1}{2}\sum_{E_{ij}} \frac{(x_{i+1,j}-x_{ij})^2 + (y_{i+1,j}-y_{ij})^2}{h_s^2} + \frac{(x_{i,j+1}-x_{ij})^2 + (y_{i,j+1}-y_{ij})^2}{B_1^2 h_t^2} + \frac{(y_{i,j+1}-y_{ij})^2}{B_2^2 h_t^2} \tag{15}$$

A reformulation of the length functional which accounts for distinct reference grids could be given, although the notation becomes rather clumsy, and it does not seem natural. As an alternative, it may be preferable to think of weighting of the length functional as proceeding by various strategies, one of which uses a single reference grid, and another of which uses *aspect weighting* as specified by Eq. (15) or more generally by

$$F_L(\mathbf{x},\mathbf{y}) = \frac{1}{2}\sum_{E_{ij}} \frac{(x_{i+1,j}-x_{ij})^2}{A_1^2 h_s^2} + \frac{(y_{i+1,j}-y_{ij})^2}{A_2^2 h_s^2} + \frac{(x_{i,j+1}-x_{ij})^2}{B_1^2 h_t^2} + \frac{(y_{i,j+1}-y_{ij})^2}{B_2^2 h_t^2} \tag{16}$$

Before closing this section, we note that the strategy of choosing the coefficients in the system

$$-\left(\frac{1}{A^2}x_{ss}+\frac{1}{B^2}x_{tt}\right)=0$$

$$-\left(\frac{1}{A^2}y_{ss}+\frac{1}{B^2}y_{tt}\right)=0 \tag{17}$$

appropriately for nonconvex domains dates back to [2]. However, the use of distinct coefficients for x and y was not considered there.

5. Interpolation and length functional

In this section, we consider two practical remedies to the problem of folding associated with the length functional. Both of these approaches use information supplied by the interpolation solution in order to modify the vanilla length functional. The first uses the notion of a reference grid. The second uses the interpolation solution to generate an appropriate right-hand side which is used in conjunction with the Hessian for the length functional to derive a linear system of equations. Although for clarity of exposition we describe these procedures separately, they can be combined to good advantage, and we will eventually do so, obtaining a generalized strategy.

Motivated by the observation which concludes the previous section, we have found that one way to control folding is to incorporate information about the curvature of the physical domain into the definition of the coefficients which define an elliptic grid generator. Unfortunately, it is a difficult task, in general, to determine a priori the right coefficients to control folding. But we have also seen that an equivalence exists between certain elliptic grid generators and the length functional under a reference grid. Suppose we exploit this connection by using the interpolation grid itself as the reference grid. This provides an automatic way to include information about the geometry, and to generate (discretized) coefficient functions. In the last section, we saw examples of folding problems associated with the vanilla length functional applied to the **S**, and the **Fishtail**. Figs. 9 and 10 present grids for these

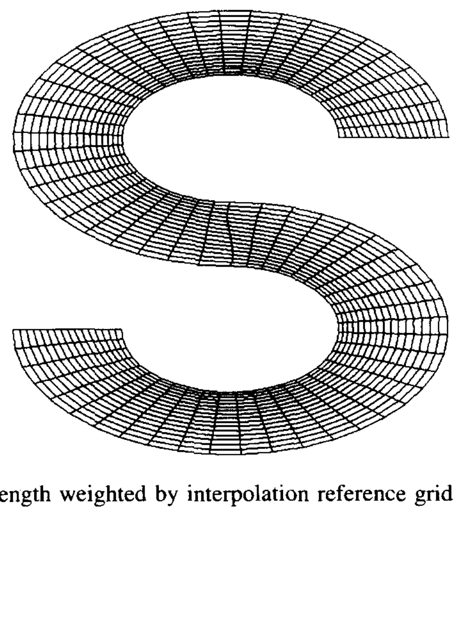

Fig. 9. Length weighted by interpolation reference grid for the S.

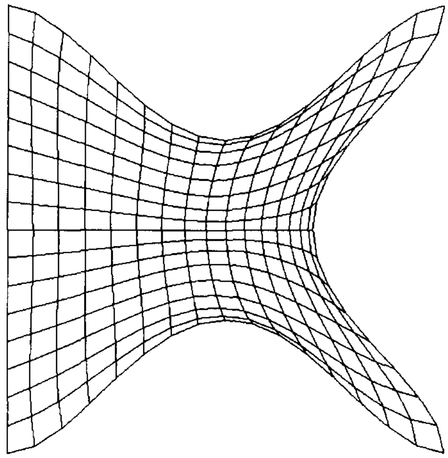

Fig. 10. Length weighted by interpolation reference grid for the **Fishtail**.

geometries generated via Eq. (12), i.e. the weighted length functional, where the weights are determined by using the interpolation grid as a reference grid. Attractive grids are generated in each case. An additional example is provided by the **Swan**, which is considered to be a rather difficult geometry (see, for example, the discussion in [8]). The interpolation grid and the grid generated by length with interpolation weighting are shown in Figs. 11 and 12.

One advantage of this approach is that the procedure is entirely automatic – once a boundary distribution has been determined the user need not supply any information or interact with the solution algorithm. Another advantage (over inverse methods, such as the Winslow generator) is that a system of uncoupled linear equations may be solved to minimize the functional. Furthermore, this system corresponds to an elliptic partial differential system: a straightforward multigrid method may be used to

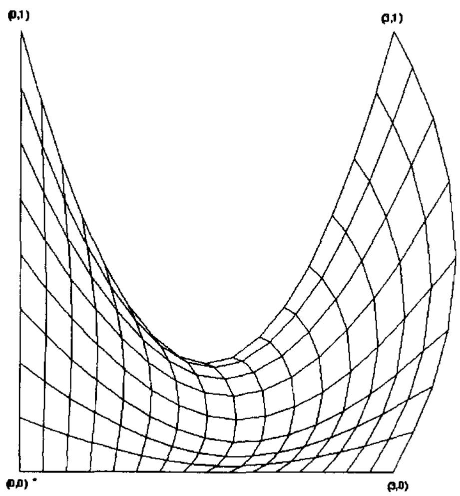

Fig. 11. Interpolation grid for the **Swan**.

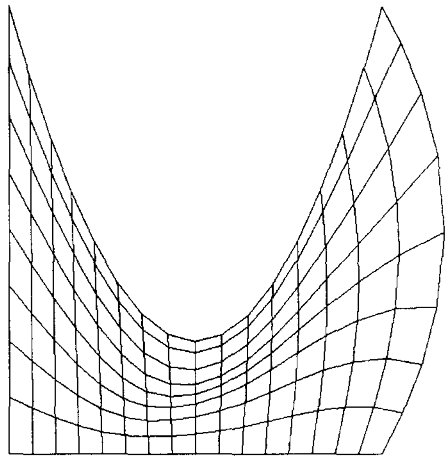

Fig. 12. Length weighted by interpolation reference grid for the **Swan**.

compute a rapid solution. A final advantage of this approach is its robustness: it generates nonfolded grids for a wide variety of nonconvex geometries. Yet, in some instances, the user may wish to exert additional control over grid quality. The second approach we describe has more flexibility to allow this kind of control.

Concatenate the variables **x** and **y** into a single variable **z**. Suppose we write $\nabla F_L(z) = 0$ (for vanilla length) as a linear system and incorporate the boundary conditions in the right-hand side:

$$H_L \mathbf{z}_L = \mathbf{f}_L \tag{18}$$

Let $\mathbf{f}_I = H_L \mathbf{z}_I$, where $\mathbf{z}_I$ is the interpolation solution (restricted to interior nodes). We obtain a new mapping by taking a convex combination of length and interpolation right-hand side.

$$H_L \mathbf{z}_{L\text{-}I} = (1 - \alpha)\mathbf{f}_L + \alpha \mathbf{f}_I, \quad \alpha \in [0, 1] \tag{19}$$

For $\alpha = 0$, this reproduces the vanilla length solution, and for $\alpha = 1$ it reproduces interpolation. Thus, we get a family of grids which, in some sense, lie in between vanilla length and interpolation.

Now, if the grid is ill-behaved in some localized region, it may be more appropriate to weight the right-hand side locally, using the length right-hand side alone where the grid is well-behaved. In order to incorporate local control, we introduce the use of a *mask* for the right-hand side. Choose a set $S = \{(i,j)\}$ of nodes to weight, and use instead,

$$H_L \mathbf{z}_{L\text{-}I} = (1 - \delta_{ij})\mathbf{f}_L + \delta_{ij} \mathbf{f}_I \tag{20}$$

where δ_{ij} is 1 if the node (i,j) is in S, and 0 otherwise. To determine the set S, define a logical distance function,

$$d((m,n),(i,j)) = |m - i| + |n - j| \tag{21}$$

and a *target set* $T = \{(m, n)\}$. This latter set can usually be taken to consist of a small set of nodes, each of which is "centered" in a specific region where improvement of the grid is desired. Then let

$$S = \{(i,j) : d((m,n),(i,j)) \le K \quad \text{for all} \quad (m,n) \in T\} \tag{22}$$

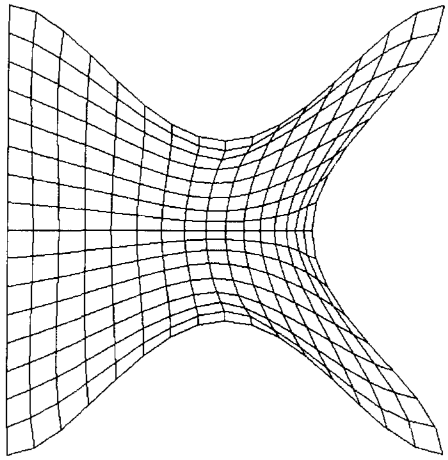

Fig. 13. Length with modified right-hand side for the **Fishtail**.

for some (usually small) integer K. To demonstrate the use of Eq. (20), consider again the **Fishtail**; recall the folding in Fig. 7 associated with vanilla length. We generate a new grid using the target set consisting of the three boundary nodes centered at the concave sides, and $K = 1$ in Eq. (22). The result is shown in Fig. 13.

In certain difficult cases, it turns out that one can gain a good deal of advantage by combining the above strategies of using a reference grid to define a weighted length functional and using an interpolation right-hand side associated with the Hessian for that functional.

Consider the latter strategy of applying a mask to the interpolation right-hand side, applied to the (backwards) **C**. We used a 17×33 grid with a target set of size 11, consisting of points surrounding the **C**'s horizontal slot, and $K = 5$. The result is shown in Fig. 14. Although the grid is much smoother than the interpolation grid, severe folding is evident despite appropriate masking of the interpolation right-

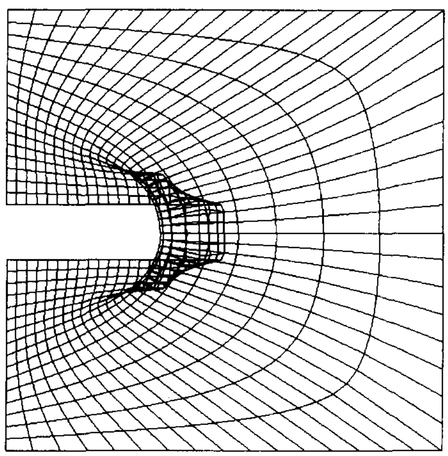

Fig. 14. Length with trivial reference grid and nontrivial target set for the **C**.

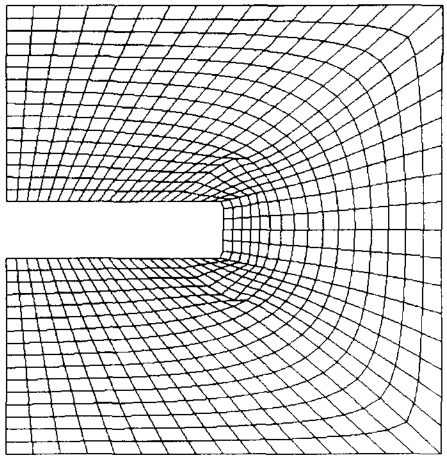

Fig. 15. Generalized length functional for the **C**.

hand side. Fig. 15 shows the result of using a generalized length strategy for this geometry. In particular, the interpolation grid was used as a reference grid in weighting length. Then the Hessian for this functional was applied to interpolation solution, yielding a new interpolation right-hand side. Finally, the same mask as above was applied to this right-hand side and the resulting linear system with Hessian as coefficient matrix was solved. Both smoothness and folding have been controlled by using this combination of strategies.

6. A generalized length strategy

In the previous section, the concept of a generalized length functional was introduced. The advantage of using such a strategy, which incorporates various methods introduced earlier in this paper, was also demonstrated. We note that when this strategy is viewed as functional minimization, the functional is defined only implicitly: it is embodied in a constructive procedure for computing a grid, namely, solving a particular system of linear equations. In this section, we provide a formal definition for what we call a generalized length strategy, and discuss some of its properties. There are three basic components which define such a strategy: a generalized length functional (i.e. a functional derived from vanilla length by weighting) an associated interpolation right-hand side, and a mask for this right-hand side.

A generalized length functional has the form:

$$F_L(\mathbf{x},\mathbf{y}) = \frac{1}{2}\sum_{E_{ij}} \frac{(x_{i+1,j}-x_{ij})^2}{A_{ij}^2} + \frac{(y_{i+1,j}-y_{ij})^2}{B_{ij}^2} + \frac{(x_{i,j+1}-x_{ij})^2}{C_{ij}^2} + \frac{(y_{i,j+1}-y_{ij})^2}{D_{ij}^2} \tag{23}$$

which appears as a weighting of the length functional. In order to determine the weights in an automated way, one may employ a reference grid as in Section 4.

The generalized length functional provides a coefficient matrix for a linear system of equations whose solution is the desired grid. The right-hand side for this system is based on the interpolation right-hand side.

Once again, write $\nabla F_L(\mathbf{z}) = 0$ as a linear system in the matrix H_L and right-hand side $\mathbf{f}_L$. Let $\mathbf{z}_I$ be the grid obtained by interpolating a specified boundary relation on the physical domain (as with transfinite interpolation or a similar procedure). The interpolation right-hand side is obtained by applying the Hessian to the interpolation solution:

$$\mathbf{f}_I = H_L \mathbf{z}_I \tag{24}$$

The last step in constructing a generalized length strategy is generating a mask for the right-hand side. A mask for the right-hand side is obtained by generating a Kronecker-δ function as described in the previous section. The generalized length strategy combines Eqs. (18) and (19) and solves the system of equations

$$H_L \mathbf{z} = (1 - \alpha\delta_{ij})\mathbf{f}_L + \alpha\delta_{ij}\mathbf{f}_L, \quad 0 \le \alpha \le 1 \tag{25}$$

to obtain the grid represented by $\mathbf{z}$.

In practice, we have found that the strategy which uses a pure convex combination of length and interpolation right-hand sides, i.e.

$$H_L \mathbf{z} = (1 - \alpha)\mathbf{f}_L + \alpha\mathbf{f}_I \tag{26}$$

and which weights length by the reference grid obtained by transfinite interpolation gives a very robust approach for handling nonconvex geometries. It avoids the complications associated with determining a mask for the right-hand side, and only depends upon the choice of a single parameter. We have found it especially useful for nonuniform boundary distributions. An example is provided by the **Fishtail** with nodes accumulated near a point on the right side of the south boundary. Fig. 16 shows the interpolation grid for this problem. Fig. 17 shows the result of using Eq. (19) with length weighted by the interpolation grid and $\alpha = 2$.

We close by making three points concerning applicability of the generalized length strategy.

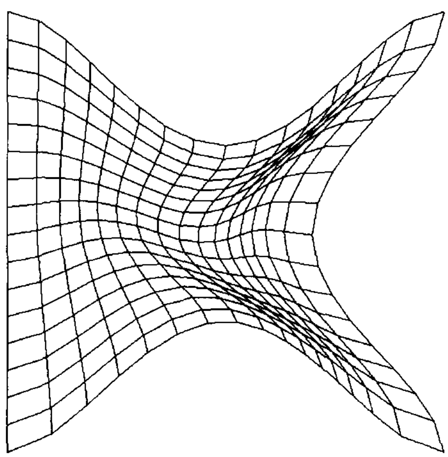

Fig. 16. Transfinite interpolation for the **Fishtail** with nonuniform boundary.

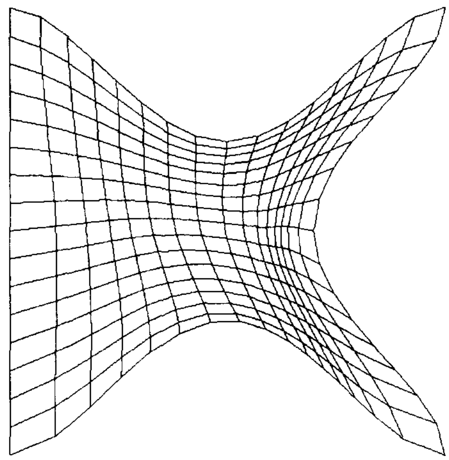

Fig. 17. Generalized length for the **Fishtail** with nonuniform boundary.

Any of the previous strategies based on length (as well as the interpolation grid itself) may be reproduced by choosing parameters appropriately in generalized length. For example, vanilla length is obtained by setting $\delta_{ij} \equiv 0$. The interpolation grid is always obtained by setting $\delta_{ij} \equiv 1$ and $\alpha = 1$ (as long as the generalized length functional yields a nonsingular Hessian).

We can always view Eq. (19) as a discretization of an uncoupled system (for the two coordinates) of elliptic PDEs on the unit square using a rapid solution of Eq. (19). All of the examples in this paper were computed using (our own version of) multigrid based on Gauss–Seidel smoothing. In certain cases, we have added flavors to the solution procedure by considering alternative smoothers, such as variations on red–black and weighted block-Jacobi smoothing. Such alternatives may be useful in certain situations. Consider the case of an aspect-weighted functional with $A_1/B_1 \gg 1$. This corresponds to a highly anisotropic diffusion problem. Therefore, some advantage would be obtained in this case by basing multigrid on appropriate line smoothing. However, a potential user need not be proficient in multigrid to solve the required linear systems; a user-transparent black box multigrid solver [7], which provides some smoothing options, can be used. The conjugate gradient method preconditioned by MILU factorization combined with Schur complement reduction [1] is also a viable and robust method which can be tuned to deal with anisotropy [9].

Finally, the strategy extends formally to three dimensions, by the trivial inclusion of another coordinate.

7. Summary

We combine a number of strategies in grid generation, including direct optimization, reference grids, and interpolation solutions to obtain a generalized length functional for grid generation that does not have the drawbacks associated with standard variational or elliptic methods. In its most general form, this functional provides the practitioner with a wide range of control over grid qualities that effect smoothness and folding of grids. We identify a particular form of the functional that is robust and is easily controlled by a single parameter.

The equations for minimizing the generalized length functional are equivalent to the linear equations obtained by discretizating an elliptic system, and are therefore readily solved by a standard multigrid algorithm. This provides a method with optimal computational complexity.

A detailed overview of direct optimization, including discussions of generalized length, the alignment problem, and dynamic adaptation can be found in [6].

Acknowledgements

We would like to thank Stan Steinberg for his interest, encouragement, and insight.

References

[1] O. Axelsson, I. Gustafsson, On the use of preconditioned conjugate gradient methods for red–black ordered five point difference schemes, J. Comp. Phys. 35 (1980) 284–289.
[2] W.D. Barfield, An optimal mesh generator for Lagrangian hydrodynamic calculations in two space dimensions, J. Comp. Phys. 6 (1970) 417–429.
[3] J.U. Brackbill, J.C. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comp. Phys. 46 (1982) 342–368.
[4] J.E. Castillo, A discrete variational grid generation method, SIAM J. Sci. Stat. Comput. 12 (1991) 454–468.
[5] J.E. Castillo, P.J. Roache, S. Steinberg, Mathematical aspects of variational grid generation II, J. Comp. Appl. Math. 20 (1987) 127–135.
[6] J.E. Castillo, J.S. Otto, A practical guide to direct optimization for planar grid-generation, Comp. Math. Appl., in press.
[7] J.E. Dendy, Jr., Black box multigrid, J. Comp. Phys. 48 (1982) 366–386.
[8] W.J. Gordon, L.C. Thiel, Transfinite mappings and their application to grid generation, J.F. Thompson (Ed.), Numerical Grid Generation, North-Holland, New York, 1982.
[9] K.L. Kipp, J.S. Otto, T.F. Russel, D4Z-a new renumbering for iterative solution of ground-water flow and solute-transport equations, in: Proceedings of the IX International Conference on Computational Methods in Water Resources, Denver, Colorado, June 9–12, 1992.
[10] P.M. Knupp, S. Steinberg, The Fundamentals of Grid Generation, CRC Press, Boca Raton, 1993.
[11] C.W. Mastin, J.F. Thompson, Elliptic systems and numerical transformations, J. Comp. Phys. 62 (1978) 52–62.
[12] A.M. Winslow, Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh, J. Comp. Phys. 2 (1967) 149–172.