JOURNAL OF COMPUTATIONAL PHYSICS 91, 255–277 (1990)

Anomalies in Grid Generation on Curves*

STANLY STEINBERG

*Department of Mathematics and Statistics, University of Mexico,
Albuquerque, New Mexico 87131*

AND

PATRICK J. ROACHE

*Ecodynamics Research Associates, Inc., P.O. Box 8172,
Albuquerque, New Mexico 87198*

Received March 22, 1989; revised May 30, 1989

Attempts to use variational grid-generation methods to generate grids on certain surfaces of modest shape failed. There were sufficient points in the grids to well-resolve the surface, so the failures were not easily explained. Similar difficulties were found for variational grid generation on curves; those problems are caused by multiple solutions of the underlying nonlinear algebraic equations. © 1990 Academic Press, Inc.

1. BACKGROUND

This paper describes several grid-generation anomalies discovered by the authors while using the variational techniques, described in Steinberg and Roache [9], to generate grids on surfaces. These surfaces are of modest shape (they are intended to model a stern wave behind a ship hull) and the grids being generated resolve the surfaces well. The anomalies manifest themselves as convergence problems. Grids are difficult or impossible to generate: for some coarse resolutions (to be expected); for many fine resolutions (not to be expected). For many intermediate resolutions, it is easy to generate excellent grids. Numerical experimentation fails to pinpoint the source of the difficulties. The surface grid-generation equations are a complicated system of two coupled quasi-linear partial differential equations that have a form similar to elliptic equations.

Some numerical experimentation shows that the analogous grid-generation code for curves has the same difficulties as the surface grid generator. This paper presents an analysis of the curve problem because it is simpler than the surface problem and

* This work was partially supported by the Office of Naval Research and the Air Force Weapons Laboratory. This is a modified version of AIAA Paper No. 88-3740, presented at the 1st National Fluid Dynamics Conference, Cincinnati, OH, July, 1988.

0021-9991/90 $3.00

is clearly relevant to it. There are many ways to generate grids on curves which are one-dimensional; there are far fewer methods available for surfaces. Only those curve grid-generation schemes that can be extended to surface schemes will be considered here.

These are reasons to believe that the difficulties encountered in curve grid generation will show up in other grid-generation problems. When the Euler–Lagrange equation for a variational problem are used to generate the grid on a curve, a quasi-linear boundary value problem of the form

$$x''(t)+g(x(t))(x'(t))^2=0, \qquad x(0)=0,\ x(1)=1 \tag{1.1}$$

must be solved. Many of the differential equations used to generate one-dimensional grids have this form, where

$$g(x)=(\ln(f(x)))'=f'(x)/f(x). \tag{1.2}$$

For examples, see [1, Eq. (3); 7, Eq. (4)] or Eq. (2.20) below. When comparing our equations to those used by others, it is helpful to note that

$$\frac{d}{d\xi}\,g(x(\xi))=g'(x)x'. \tag{1.3}$$

The equation for defining a grid equidistributed with respect to a position dependent weight $w(x)$ (i.e., solution-adaptive grid) is given by

$$x'(t)\,w(x(t))=C \tag{1.4}$$

(see [10, p. 371, Eq. (4)]. The derivative of this equation with respect to t gives a second-order differential equation for $x(t)$, which is the same as the equation for the variational problem defining the adaptive grid (see [9]):

$$x''(t)+\frac{w'(x(t))}{w(x(t))}(x'(t))^2=0. \tag{1.5}$$

Since w in (1.5) plays the identical role of f in (1.2), it is clearly possible for the anomalies that cause difficulties in curve grid generation to appear in solution-adaptive grid-generation problems.

A simple quadratic curve

$$y=\alpha x(1-x), \qquad 0\leqslant x\leqslant 1,\ \alpha\geqslant 0, \tag{1.6}$$

is used as an example in this paper; all ideas were also tested on the trigonometric curve

$$y=\varepsilon\sin(\pi x), \qquad 0\leqslant x\leqslant 1,\ \varepsilon\geqslant 0; \tag{1.7}$$

but the results are not presented here because they are similar to those for the quadratic curve. The quadratic is used to simplify calculations in the example, while the trigonometric was chosen to model a wave.

Three types of grids will be discussed here: continuum grids; one-free-point grids; and discrete grids. For each, various solution methods are discussed; including fully-lagged iteration, nominal iteration, and direct minimization. In each case, the problems of existence and uniqueness of the grids are discussed; more importantly, the stability of various solution techniques is analyzed. The stability properties should give insight into which iteration method is best for computing grids.

Because the above-mentioned differential equation can be integrated, the continuum problem can be analyzed in detail (see Section 2). The simplest discrete grid has one free point and two fixed boundary points (see Section 3). In the case of the quadratic curve, all numerical methods can be analyzed completely for this simple problem. When the discrete grid has a modest number of points, numerical codes are used to study the grid (see Section 4). Two numerical codes are used (written by the authors). One is called "gencur," which implements the method described in [9], and another is called "arc2," which is a small test code implementing several solution techniques.

The results in this paper make it clear that the anomalies occur because the discrete grid-generation problem has many solutions. In the one-free-point problem, the grid undergoes a pitchfork bifurcation as the parameters α or ε increase. For grids with a modest number of points on moderate curves, some numerical methods find a multitude of stable solutions. This is illustrated in Fig. 1, where several anomalous grids are shown. In this figure, the curves are the parabolas described above. All grids contain nine points, including the boundary points. The curves illustrate the types of grids that can be obtained from the nominal algorithm that is described below. The first curve illustrates a typical random grid that was used for initial conditions. The grid on the curve of height one-half is not anomalous and is, in fact, an excellent result. If the curve is of height one or higher, then the nominal iteration bifurcates and there are multiple solutions to the grid-generation equation; this is illustrated by the remaining curves. For finer grids see the discussion at the end of Section 4.1. With sufficiently high resolution for a given curve, some of the methods find a unique solution; however, these grids contain many more points than are necessary to resolve the curve. (The continuum grid-generation equation has a unique solution.) More detailed conclusions will be given in the last section of this paper.

Other numerical solution methods can be considered: Newton-type iterations; conjugate gradient; multigrid; and so forth. These methods typically converge faster to one of the solutions, but the solution is dependent on the initial guess for the grid and solution algorithm. The discrete equations have multiple solutions, and there is no way to guarantee that the computed solution will be the desired one; consequently, advanced solution methods are not discussed in this paper.

The analysis given here provides substantial insight into the difficulties of generating grids on curves and surfaces. The authors do not have a solution to the

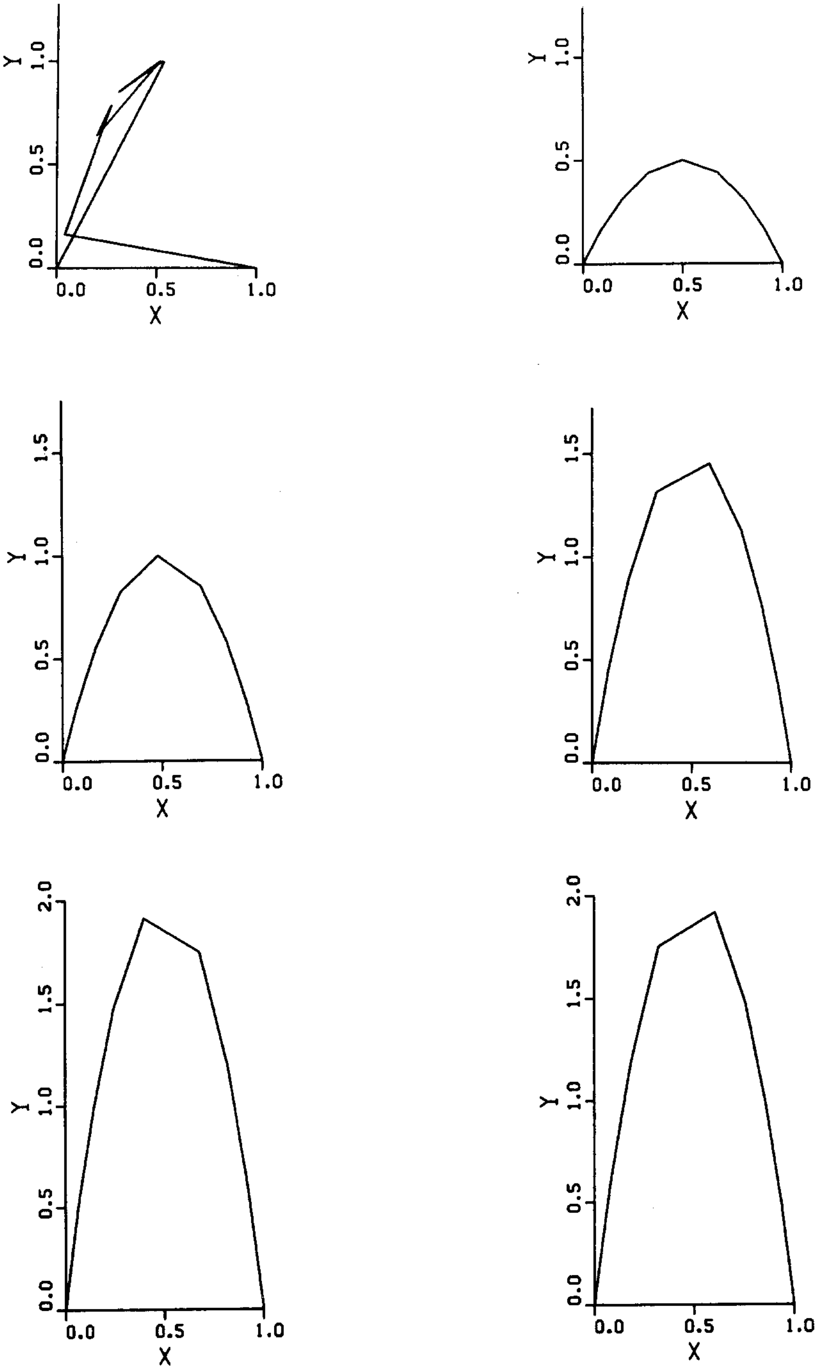

FIG. 1. Anamalous grids.

problem of generating such grids. However, one of their students [6] is studying a wide range of grid-generation algorithms. Most of the algorithms show anomalous behavior, either bifurcations or multiple solutions, but a couple of the algorithms show exceptional promise as robust grid generators.

Many of the algebraic calculations in this paper are done or checked by using the symbol manipulator MACSYMA [8]. Also, the results in Section 2 can be made rigorous by formulating them in a Hilbert or Banach space setting. The rigorous results provide no additional insight, so they were not pursued.

We would like to thank Patrick Knupp for helping to clarify the analysis of the second derivative of the continuum functional.

1.1. *Curves*

The arc length and its differential frequently appear in these calculations. We expected the curvature to play a significant role, but as far as we could see, it does not. Let a curve (in vector form) be given by

$$R(r) = (x(r), y(r), z(r)) \tag{1.8}$$

and its derivative be given by

$$R'(r) = (x'(r), y'(r), z'(r)). \tag{1.9}$$

Then the r derivative of the arc length s is given by

$$\frac{ds}{dr} = \|R'(r)\| = \sqrt{(x'(r))^2 + (y'(r))^2 + (z'(r))^2} = L(r) \tag{1.10}$$

and the length of a segment of the curve between $r = 0$ and $r = r$ is given by

$$s = s(r) = \int_0^r L(\rho)\, d\rho. \tag{1.11}$$

We always assume that

$$L(r) > 0. \tag{1.12}$$

This implies that the arc length is always a strictly increasing function of r. Let

$$S(r) = L^2(r) \tag{1.13}$$

and

$$D(r) = (x''(r))^2 + (y''(r))^2 + (z''(r))^2 \tag{1.14}$$

and note that

$$S'(r) = 2(x'(r)\, x''(r) + y'(r)\, y''(r) + z'(r)\, z''(r)) \tag{1.15}$$

and

$$S''(r) = 2D(r) + 2(x'(r)\,x'''(r) + y'(r)\,y'''(r) + z'(r)\,z'''(r)). \tag{1.16}$$

The curvature K is then given by

$$K^2(r) = \frac{S(r)\,D(r) - (S'(r)/2)^2}{S^3(r)}. \tag{1.17}$$

(Note that the curvature depends only on the first and second derivatives of the coordinates of the curve.)

2. Continuum Grids

A continuum grid is merely a reparameterization of a curve. Thus, the problem here is essentially to reparameterize a curve using arc length. Such problems have been studied by a number of authors (see Steinberg and Roache [9], or Thompson, Warsi, and Mastin [10]). Explicit expressions for the reparameterization are given which settle the questions of existence and uniqueness of the parameterization. The related variational problem is studied along with several iteration methods for finding the solution.

2.1. *Equidistributed Grids on Curves*

Equidistribution of points on a curve requires that the arc lengths between grid points should be equal. The developments in Steinberg and Roache [9] discuss how to use a variational technique to generate a continuum grid on a curve, so that the grid is distributed according to some given weight. The variational problem for equidistributing the grid is to minimize

$$I(r) = \int_0^1 (L(r)\,r')^2\,d\xi = \int_0^1 S(r)(r')^2\,d\xi \tag{2.18}$$

over all functions $r = r(\xi)$ with $r(0) = 0$ and $r(1) = 1$. The Euler-Lagrange equation for the minimization problem is

$$2S(r)\,r'' + S'(r)(r')^2 = 0, \qquad r(0) = 0,\ r(1) = 1. \tag{2.19}$$

Recall that $S(r) = L^2(r)$, so, in terms of arc length, the previous equation becomes

$$r'' + \frac{L'(r)}{L(r)}(r')^2 = 0, \qquad r(0) = 0,\ r(1) = 1. \tag{2.20}$$

This boundary value problem always possesses a unique solution $r = r(\xi)$. The inverse of the solution $\xi = \xi(r)$ is a normalized arc length, as the following discussion shows.

First note that r equals a constant is a solution of the differential equation (with $r'(\xi)\equiv 0$). To find other solutions, assume that r' is not identically zero. Then, one integration of (2.19) gives

$$S(r)(r')^2=K^2, \tag{2.21}$$

where $K>0$ (note that the left-hand side of the integral is greater than or equal to zero and not identically zero). Note that, because $S(r)>0$ and $K>0$, $r'(\xi)\neq 0$. This equation cannot, in general, be solved for r as an explicit function of ξ. However, the equation can be rewritten as

$$\frac{d\xi}{dr}=\frac{1}{r'}=\frac{\sqrt{S(r)}}{K}=\frac{L(r)}{K}. \tag{2.22}$$

Another integration, and the boundary conditions give

$$\xi=\frac{\int_0^r L(\tau)\,d\tau}{\int_0^1 L(\tau)\,d\tau}. \tag{2.23}$$

This implies that ξ is a normalized arc length $s=K\xi$. Note that (2.22) implies that $d\xi/dr>0$, so the previous equation can always be inverted for $r=r(\xi)$. The inverse function is clearly a non-constant solution of (2.19). Thus, the continuum grid is given by a normalized arc length; this is exactly what is meant by an equidistributed grid.

2.2. *Direct Minimization*

The functional $I(r)$ given in the previous section has a minimum at a point $\tilde{r}$ in the space of smooth functions $r=r(\xi)$, if, for each direction a at the point $\tilde{r}$, the first derivative of the functional is zero and the second derivative is positive. Thus, let $a=a(\xi)$ and $b=b(\xi)$ be smooth and have compact support in the interior of $[0, 1]$. The directional derivative (in the direction a at $\tilde{r}$) is given by

$$D_a I(\tilde{r})=\frac{d}{d\varepsilon}I(\tilde{r}+\varepsilon a)|_{\varepsilon=0}, \tag{2.24}$$

while the mixed second derivative is given by

$$D_b\,D_a I(\tilde{r})=\frac{d}{d\varepsilon}D_a I(\tilde{r}+\varepsilon b)|_{\varepsilon=0}. \tag{2.25}$$

The functional has a minimum at $\tilde{r}$ provided that

$$D_a I(\tilde{r})\equiv 0, \qquad D_a^2 I(\tilde{r})>0, \tag{2.26}$$

for all $a\neq 0$ satisfying the conditions given above.

The computation of $D_a I$ and $D_b D_a I$ is a straightforward application of differentiation and integration by parts:

$$D_a I(r) = \int_0^1 2S(r)\, r'a' + S'(r)(r')^2\, a d\xi,$$

$$= -\int_0^1 (2S(r)\, r'' + S'(r)(r')^2)\, a\, d\xi, \tag{2.27}$$

$$D_b D_a I(r) = \int_0^1 2S(r)\, a'b' + 2S'(r)\, r'a'b + 2S'(r)\, r'ab' + S''(r)(r')^2\, ab\, d\xi. \tag{2.28}$$

The second derivative is then given by

$$D_a^2 I(r) = \int_0^1 2S(r)(a')^2 + 4S'(r)\, r'aa' + S''(r)(r')^2\, a^2\, d\xi. \tag{2.29}$$

The requirement that $D_a I(\tilde{r}) \equiv 0$ for all a implies that

$$2S(\tilde{r})\, \tilde{r}'' + S'(\tilde{r})(\tilde{r}')^2 = 0, \tag{2.30}$$

which is nothing but the Euler–Lagrange equation (2.19) for the minimization of $I(r)$. The discussion of the previous sections shows that the functional has a unique critical point, which is given by the arc length parameterization of the curve. However, this critical point is not necessarily a minimum. Even though, in general, there is no simple formula for the critical point of the minimization problem (i.e., the solution of the Euler–Lagrange equation), this critical point is, in fact, always a minimum. This is seen as follows. Integration of (2.30) (i.e., Eq. (2.21)) along with the chain rule gives

$$S(\tilde{r}) = \frac{K^2}{(\tilde{r}')^2}, \tag{2.31}$$

$$S'(\tilde{r}) = \frac{-2K^2}{(\tilde{r}')^4}, \tag{2.32}$$

$$S''(\tilde{r}) = \frac{8K^2}{(\tilde{r}')^6}, \tag{2.33}$$

where $K > 0$ and $r'(\xi) > 0$. Substituting this into the second derivative of the functional gives

$$D_a^2 I(\tilde{r}) = 2K^2 \int_0^1 \frac{(a'\tilde{r}' - 2a)^2}{(\tilde{r}')^4}\, d\xi. \tag{2.34}$$

This expression is always greater than or equal to zero and can only equal zero if the factor in the numerator is zero. Given $\tilde{r}$, an a which makes the numerator identically zero can always be computed. However, the ordinary differential equation, which a satisfies, implies that if $a(\xi)$ is zero at one point, then $a \equiv 0$. Because a has zero boundary conditions, the second derivative of the functional will always be positive in every allowable direction, and thus, the normalized arc-length parameterization is a minimum. On the other hand, if a is taken to be the solution of the ordinary differential equation except near the boundary points, where a is made to go smoothly to zero, then the integral in (2.34) can be made arbitrarily small and consequently the second derivative cannot be uniformly bounded below. This indicates the possibility of serious computational problems.

2.2.1. Example. A family of planar quadratic curves

$$y = \alpha x(1-x), \qquad 0 \leqslant x \leqslant 1 \tag{2.35}$$

(parameterized by α) is used as an example through out this paper. The parameter α is used to adjust the height ($= \alpha/4$) of the curve. To apply the above results, write the curve in parametric form:

$$x = r, \qquad y = \alpha r(1-r), \qquad z = 0, \qquad 0 \leqslant r \leqslant 1. \tag{2.36}$$

The Euler–Lagrange equation for this problem can be integrated (as was noted above). However, this gives an implicit solution for r as a function of ξ (in fact, an explicit solution for ξ as a function of r) that is transcendental and not explicitly solvable for r as a function of ξ (the solution $\xi(r)$ can easily be computed using MACSYMA). In any case, this solution is a minimum of the functional for any value of α. This is in sharp contrast with what is found for approximations of the Euler–Lagrange equation.

2.3. *The Fully-Lagged Iteration*

The continuum analog of the iteration technique used in the code "gencur" has the worst convergence properties of the techniques studied. However, in multidimensional problems, the analogs of this technique require minimal storage, therefore such techniques are advantageous when this is a consideration. This iteration will be written in the form of an integro-differential equation, so that its convergence properties are clear.

The fully-lagged iteration is based on writing the differential equation so that the linearized version is as simple as possible (i.e., as much of the differential equation as possible is written on the right-hand side of the equation; putting things on the right-hand side of the equation is referred to as lagging). The fully-lagged form of the boundary value problem is

$$r''(\xi) = -\frac{S'(r(\xi))}{2S(r(\xi))}(r'(\xi))^2, \qquad r(0)=0,\ r(1)=1, \tag{2.37}$$

where $r = r(\xi)$ is to be found and $S(r)$ is a given function. The numerical iteration used in "gencur" corresponds to the continuum iteration

$$r''_{n+1}(\xi) = -\frac{S'(r_n(\xi))}{2S(r_n(\xi))}(r'_n(\xi))^2, \qquad r_{n+1}(0) = 0,\ r_{n+1}(1) = 1, \tag{2.38}$$

for $n \geqslant 0$. A typical initial guess is $r_0(\xi) = \xi$. The Green's function for the second derivative with the given boundary conditions is

$$G(\xi, \tau) = \tau(\xi - 1)\, H(\xi - \tau) + \xi(\tau - 1)\, H(\tau - \xi), \tag{2.39}$$

where H is the usual Heavyside function. If

$$F(r)(\xi) = -\int_0^1 G(\xi, \tau)\frac{S'(r(\tau))}{2S(r(\tau))}(r'(\tau))^2\, d\tau, \tag{2.40}$$

then (2.38) is equivalent to

$$r_{n+1} = F(r_n), \qquad n \geqslant 0, \tag{2.41}$$

with r_0 given. Note that it is easy to formulate this operator in a Banach space setting where F will be compact (and nonlinear).

Convergence of the iteration depends on the directional derivative of F at r in the direction c, which was defined as

$$D_c F(r) = \frac{d}{d\varepsilon} F(r + \varepsilon c)|_{\varepsilon = 0}. \tag{2.42}$$

Here $c = c(\xi)$, with $c(0) = c(1) = 0$. If $\tilde{r} = F(\tilde{r})$ and $\|D_c F(\tilde{r})\| < \|c\|$, for all c and some appropriate norm, then the Picard iteration will converge to $\tilde{r}$.

A short computation gives

$$\begin{aligned} D_c F(\tilde{r})(\xi) = &\int_0^1 G(\xi, \tau)\frac{(r'(\tau))^2\, S''(r(\tau))}{2S(r(\tau))}\, c(\tau)\, d\tau \\ &-\int_0^1 G(\xi, \tau)\left(\frac{r'(\tau)\, S'(r(\tau))}{S(r(\tau))}\right)^2 c(\tau)\, d\tau \\ &+\int_0^1 G_\tau(\xi, \tau)\frac{r'(\tau) S'(r(\tau))}{S(r(\tau))}\, c(\tau)\, d\tau. \end{aligned}$$

The critical quantities in $D_c F$ are

$$C_1 = \frac{S'(r)}{S(r)}, \qquad C_2 = \frac{S''(r)}{2S(r)} - \left(\frac{S'(r)}{S(r)}\right)^2. \tag{2.43}$$

The norm of D_cF depends on S' and S'', both of which involve the first, second, and third derivatives of the coordinates of the curve. Recall, that the curvature depends only on the first and second derivatives of the coordinates, so convergence of the iteration depends on more than the curvature.

2.3.1. EXAMPLE. For the quadratic example, the expression for the second critical quantity is complicated; neither the expression for the first or second critical quantity is illuminating. However, the limits of these expression for α large show that there are problems. For the quadratic example,

$$\lim_{\alpha \to \infty} C_1 = \frac{4}{2r-1}, \qquad \lim_{\alpha \to \infty} C_2 = -\frac{12}{(2r-1)^2}. \tag{2.44}$$

Note, that all limits are infinite for $r = \frac{1}{2}$, so convergence troubles are expected for large α.

2.4. *The Nominal Iteration*

The nominal iteration keeps as much information as possible in the linearized equation. The boundary value problem is written

$$r'' + \frac{S'(r)}{2S(r)} (r')^2 = 0, \qquad r(0) = 0,\ r(1) = 1; \tag{2.45}$$

and the iteration scheme is

$$r''_{n+1} + \left(\frac{S'(r_n)}{2S(r_n)} r'_n \right) r'_{n+1} = 0, \qquad r_{n+1}(0) = 0,\ r_{n+1}(1) = 1. \tag{2.46}$$

Set

$$g(r) = \frac{S'(r)}{2S(r)} r', \tag{2.47}$$

so that the iteration becomes

$$r''_{n+1} + g(r_n)\, r'_{n+1} = 0, \qquad r_{n+1}(0) = 0,\ r_{n+1}(1) = 1. \tag{2.48}$$

Note that

$$g(r(\xi)) = \frac{1}{2} \frac{d}{d\xi} (\ln S(r(\xi))). \tag{2.49}$$

The equation for this iteration can be integrated: rewrite the iteration as

$$\frac{r''_{n+1}}{r'_{n+1}} + g(r_n) = 0; \tag{2.50}$$

integrate with respect to ξ to obtain

$$\ln(r'_{n+1}) + \tfrac{1}{2}\ln(S(r_n)) = C. \tag{2.51}$$

This implies that $r'_{n+1}L(r_n) = C$, or

$$r_{n+1}(\xi) = \int_0^\xi \frac{C}{L(r_n(\tau))}\,d\tau, \tag{2.52}$$

where

$$\frac{1}{C} = \int_0^1 \frac{1}{L(r_n(\tau))}\,d\tau. \tag{2.53}$$

Thus the iteration can be written as

$$r_{n+1} = F(r_n),\ F(r) = \int_0^\xi \frac{d\tau}{L(r(\tau))} \bigg/ \int_0^1 \frac{d\tau}{L(r(\tau))}. \tag{2.54}$$

The directional derivative of this functional is (using $S = L^2$)

$$\begin{aligned} D_cF(r) = \bigg(\int_0^\xi \frac{S'(r(\tau))\,c(\tau)\,d\tau}{S^{3/2}(r(\tau))} \int_0^1 \frac{d\tau}{L(r(t))} - \int_0^\xi \frac{d\tau}{L(r(\tau))} \\ \times \int_0^1 \frac{S'(r(\tau))\,c(\tau)\,d\tau}{S^{3/2}(r(\tau))} \bigg) \bigg/ \bigg(\int_0^1 \frac{d\tau}{L(r(\tau))} \bigg)^2. \end{aligned} \tag{2.55}$$

The critical quantity to estimate, to show convergence of this iteration, is $C_f = S'(r)/S^{3/2}(r)$. The square of this quantity is one of the terms in the curvature, but otherwise it is not closely related to it.

2.4.1. Example. For the quadratic example, C_f is complicated; however, C_f is [redacted]

2.5. *A Special Curve*

In many numerical experiments, $S'(r)/S(r)$ is nearly constant. If

$$\frac{S'(r)}{S(r)} = 2C, \tag{2.56}$$

where C is constant, then the boundary value problem for determining the grid becomes

$$r''(\xi) + C(r'(\xi))^2 = 0, \qquad r(0) = 0,\ r(1) = 1. \tag{2.57}$$

This problem is easy to analyze. If it is assumed that a curve is given as $y = f(x)$, $0 \leqslant x \leqslant 1$, then (choosing $x = r$) $S(x) = 1 + (f')^2$, $S'(x) = 2f'f''$, and then $S'/S = 2C$ becomes

$$\frac{f'f''}{1+(f')^2} = C. \tag{2.58}$$

This can be integrated to

$$f' = \pm\sqrt{Ke^{2Cx} - 1}. \tag{2.59}$$

Next, suppose that f satisfies

$$Cf = f' - \arctan(f') + AC, \tag{2.60}$$

where A is some constant; then

$$C = \frac{f'f''}{1+(f')^2}. \tag{2.61}$$

Consequently, if

$$u(x) = \pm\sqrt{Ke^{2Cx} - 1}, \tag{2.62}$$

then

$$f(x) = \frac{u(x) - \arctan(u(x))}{C} + A. \tag{2.63}$$

This is a general solution for the constant S'/S problem.

Curves from this family are chosen for a variety of numerical experiments; the results clearly indicate that the numerical difficulties are not dependent on the size of C alone.

3. One-Point Grids

In this section, grids that contain one free point and two fixed boundary points are analyzed and used to illustrate various ideas discussed in the section on continuum grids. The last subsection of this section is devoted to a discussion of the minimization of a discretized form of the variational integral. One-point grids make a good starting point for the numerical experiments that are reported on in the next section. In fact, the one-point grid is a far better model than the continuum, when computing with highly resolved grids.

3.1. *Multiple Solutions*

The grid is calculated by solving a discretized version of the Euler–Lagrange equation,

$$2S(r)\,r'' + S'(r)(r')^2 = 0, \qquad r(0)=0,\ r(1)=1, \tag{3.64}$$

for $r=r(\xi)$, $0 \leqslant \xi \leqslant 1$. This equation is discretized using centered differences. The one-free-point grid is given by $\xi=0$, $\xi=\frac{1}{2}$, and $\xi=1$; corresponding $r=r(\xi)$ values are $r=0$, $r=r(\frac{1}{2})$, and $r=1$. Consequently,

$$r' \cong \frac{1-0}{1} = 1, \qquad r'' \cong \frac{1-2r+0}{(1/2)^2} = 4(1-2r). \tag{3.65}$$

The discretized Euler-Lagrange equation for the one free point is then

$$8(1-2r)S(r) + S'(r) = 0. \tag{3.66}$$

This equation can be integrated:

$$S(r) = e^{8(r^2-r-A)}, \qquad S(0) = e^{-8A}. \tag{3.67}$$

Now A can be determined from the parameterization of the curve and then (3.67) is a transcendental algebraic equation for r. As this equation is nonlinear, it should typically have multiple solutions; this poses problems for a grid generator.

3.1.1. Example. This is a continuation of the quadratic example. The problem is to choose a point, r, that divides this curve into two pieces of equal length (clearly, the solution should be $r=\frac{1}{2}$). From (1.13) and (1.15),

$$S(r) = 1 + \alpha^2(1-2r)^2, \qquad S'(r) = -4\alpha^2(1-2r), \tag{3.68}$$

and

$$\frac{S'(r)}{S(r)} = -\frac{4(1-2r)}{1/\alpha^2 + (1-2r)^2}. \tag{3.69}$$

The discretized Euler–Lagrange equation is

$$8(1-2r) = \frac{4(1-2r)}{1/\alpha^2 + (1-2r)^2}. \tag{3.70}$$

As desired, $r=\frac{1}{2}$ is a solution of this equation.

If $r \neq \frac{1}{2}$, then the equation becomes

$$(1-2r)^2 = \frac{1}{2} - \frac{1}{\alpha^2}. \tag{3.71}$$

If $\alpha^2 < 2$, this equation has no real solutions; if $\alpha = 2$, then $r = \frac{1}{2}$ is a double root; if $\alpha^2 > 2$, then

$$r_{\pm} = \frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{1}{2} - \frac{1}{\alpha^2}} \tag{3.72}$$

are two real roots of the equation. For α large, these roots are approximately

$$r_{\pm} = \frac{1}{2} \pm \frac{\sqrt{2}}{4}. \tag{3.73}$$

The example shows that the intuitive argument (given in Steinberg and Roache [9]) about dividing curves into pieces of equal arc length does not apply to all solutions of the discretized Euler–Lagrange equation.

3.2. *Iteration*

Both the fully-lagged iteration and the nominal iteration for a one-point grid are the same, because the expression for r' (using centered differences) does not involve the unknown point. The fully-lagged iteration is based on solving the differential equation for the second derivative:

$$r'' = -\frac{S'(r)}{S(r)}(r')^2, \qquad r(0) = 0,\ r(1) = 1. \tag{3.74}$$

For one free point, the discretized equation is

$$4(1 - 2r) = -S'(r)/2S(r). \tag{3.75}$$

The iteration is given by

$$4(1 - 2r_{k+1}) = -S'(r_k)/2S(r_k) \tag{3.76}$$

or

$$r_{k+1} = g(r_k), \qquad g(r) = \frac{1}{2}\left(1 + \frac{S'(r)}{8S(r)}\right). \tag{3.77}$$

If $\tilde{r}$ is a fixed point, $\tilde{r} = g(\tilde{r})$, then the iteration will converge if $|g'(\tilde{r})| < 1$. Now

$$g'(r) = \frac{1}{16}\left\{\frac{S''(r)}{S(r)} - \left(\frac{S'(r)}{S(r)}\right)^2\right\}. \tag{3.78}$$

3.2.1. Example. For the previous quadratic example,

$$g'(1/2) = \alpha^2/2. \tag{3.79}$$

Consequently, the iteration will converge linearly to the solution $r=\frac{1}{2}$ for $\alpha^2<2$, and diverge linearly for $\alpha^2>2$. Also,

$$g'(r_\pm)=\frac{4}{\alpha^2}-1, \tag{3.80}$$

so if $\alpha^2>2$ then $|g'(r_\pm)|<1$. Thus, when $\alpha^2>2$, the root $r=\frac{1}{2}$ is unstable, while the roots $r=r_\pm$ are stable for the iteration. Note that convergence is quadratic when $\alpha=2$, and convergence is slow when α is large.

This example shows that the grid generation equation can have multiple solutions and that the desired solution can be unstable for a reasonable solution method. These results are verified numerically in Section 4.

3.3. *Direct Variational Method*

This section compares the previous results to a direct variational method of the type introduced in Castillo [2–5]. Consider a curve defined as in 1.8, and then let r_i, $1\leqslant i\leqslant n$, be a subdivision of [0, 1], that is, $r_i\leqslant r_{i+1}$, $1\leqslant i\leqslant n-1$, $r_1=0$, $r_n=1$. If

$$\Delta R_i=R(r_i)-R(r_{i-1}) \tag{3.81}$$

and

$$d_i=\|\Delta R_i\|=\sqrt{\Delta x_i^2+\Delta y_i^2+\Delta z_i^2}, \tag{3.82}$$

then a natural analog of the functionals given in Castillo [2–5] would be

$$F=\sum_{i=2}^{n} d_i^2. \tag{3.83}$$

Note that the analog of the implied constraints [9]

$$C=\sum_{i=2}^{n} d_i=\text{const} \tag{3.84}$$

is not satisfied. However, C is bounded below by the square of the distance between the endpoints of the curve.

Consider the simple case where

$$x=r, \qquad y=f(r), \qquad z=0, \qquad f(0)=0, \qquad f(1)=0, \tag{3.85}$$

that is,

$$y=f(x), \qquad 0\leqslant x\leqslant 1. \tag{3.86}$$

If $[0, 1]$ is divided into two intervals, then $r_1 = 0$, $r_2 = r$, $r_3 = 1$, and the functional is

$$F(r) = r^2 + (1 - r)^2 + 2f^2(r). \tag{3.87}$$

To minimize F, set $F' = 0$ and then solve for r.

3.3.1. Example. As before, consider the special case where the curve is quadratic, $f(x) = \alpha x(1 - x)$. Then

$$F'(r) = 4\alpha^2(r - \tfrac{1}{2})(r^2 - r + 1/2\alpha^2). \tag{3.88}$$

Consequently, $r = \frac{1}{2}$ is always a solution. The quadratic has roots

$$r_\pm = \tfrac{1}{2} \pm \tfrac{1}{2}\sqrt{1 - 2/\alpha^2}. \tag{3.89}$$

Note, that $r_\pm$ are symmetric with respect to $r = \frac{1}{2}$. These roots are real for $\alpha^2 \geqslant 2$ and distinct for $\alpha^2 > 2$. Also, $F''(r_\pm) > 0$, for $\alpha^2 > 2$, while $F''(\frac{1}{2}) = (2 - \alpha^2)$ changes sign at $\alpha^2 = 2$. Thus, for $\alpha^2 < 2$ there is one minimum at $r = \frac{1}{2}$; for $\alpha^2 > 2$ there are two minima at $r = r_\pm$ and a unique maximum at $r = \frac{1}{2}$. These results are similar to the results for the Euler–Lagrange method.

3.3.2. Example. Recall, that if the curve is a semicircle given by

$$f(x) = \sqrt{\tfrac{1}{4} - (x - \tfrac{1}{2})^2}, \tag{3.90}$$

then

$$d_2^2 + d_3^2 = 1; \tag{3.91}$$

that is, $F(r)$ is a constant. This is a seriously degenerate problem.

These examples indicate that the direct minimization of a discretized integral has difficulties similar to (perhaps worse than) the Euler–Lagrange methods.

4. Numerical Experiments

In this section, some results from numerical experiments done with the codes “gencur” and “arc2” are given. These codes implement Picard iteration schemes for solving the nonlinear equations; the linear equations are solved using a standard tridiagonal solver. The nonlinear solvers were run without any relaxation; the iteration was stopped when the maximum of the absolute differences of two successive iterates was less than 10^{-5}. This tolerance is a bit difficult to satisfy, i.e., a fairly large number of iterations are required. On the other hand, a fairly tight tolerance is needed to distinguish multiple solutions near the bifurcation point.

The nominal iteration behaves better than the fully-lagged iteration; the results for the fully-lagged iteration are important because this is the type of iteration used in many higher-dimensional problems.

4.1. *The Fully-Lagged Iteration*

The code "gencur" implements the fully-lagged Picard iteration. This code will be compared to "arc2" (which uses the nominal iteration) later in this section. The example discussed in the previous sections is used to illustrate the codes.

4.1.1. *Quadratic Curves.* Table I contains some runs for the curve

$$y = \alpha x(1-x), \qquad 0 \leqslant x \leqslant 1, \tag{4.92}$$

for various values of α. This table illustrates the dependence of the grid on α. These results confirm the results in the section on one-point grids, including the bifurcation of the root at $\alpha = \sqrt{2}$. Recall, that the grid has two boundary points and one free point. The initial grid contains the points $r = 0.0$, 0.01, 1.0. The initial grid is skewed, so that non-symmetric solutions will be found. In Table I, *nliter* is the number of nonlinear iterations, r is the root computed by "gencur" and $\tilde{r}$ is the true root.

Tables II, III, and IV indicate what happens as the number of points in the grid increases. Each table is labelled with a value of α; the values are near $\sqrt{2}$. The column labelled with *points* contains the number of points in the grid. The quantity *dev* measures the smoothness of the grid; which is the standard deviation of the grid lengths divided by the average of the grid lengths. Again, $r(2) = 0.01$, so that the initial grid is skewed.

In Table II, $\alpha > \sqrt{2}$; in Table III, $\alpha = \sqrt{2}$; and in Table IV, $\alpha < \sqrt{2}$. One could conclude from these experiments that all problems disappear as the grid is resolved; this is not true, as shown in the following examples.

As α is increased, grids become increasingly difficult to generate. At some point, the fully-lagged algorithm fails to converge in 200 iterations. This point is called the bifurcation point because of the behavior of the one-point grid. However, it may be better to call this the "divergence" point. The results for the nominal iteration (given in Table V) will clarify this point. Numerical experimentation indicates that grids containing about 20 to 30 points are easy to generate, so some results for grids with 21, 41, and 81 points are presented. The results in Table IV indicate that for $\alpha \cong \sqrt{2}$, grids of most sizes are easy to generate. The results in Table V indicate that the grid containing 21 points bifurcates for α slightly larger than 4. The grid containing 41 points bifurcates for $3.37 < \alpha < 3.38$ while the grid containing 81 points bifurcates for $3.09 < \alpha < 3.10$ (the data for 41 and 81 points looks just like the data for 21 points, and so, it is not presented). The decrease of the bifurcation value with increasing grid size is a very serious problem for the grid generator. Note, that the midpoint of the grid should be $r = \frac{1}{2}$ because the number of grid points is odd and the curve is symmetric about $r = \frac{1}{2}$. The initial grid is evenly distributed in r.

TABLE I

One-Point Grids

α	Nliter	r	$\tilde{r}\alpha$
1.3	46	0.49995	0.5
1.4	200	0.49914	0.5
$\sqrt{2}$	—	—	0.5
1.5	28	0.38221	0.38215
2.0	5	0.25000	0.25000
2.5	11	0.20845	0.20845
3.0	19	0.18820	0.18820
5.0	59	0.16089	0.16088

TABLE II

$\alpha = 1.5$

Points	Nliter	Dev
3	28	0.2
5	8	0.01
9	9	0.005
17	9	0.001
33	7	0.0003

TABLE III

$\alpha = 1.41421356$

Points	Nliter	Dev
3	> 200	0.02
5	5	0.01
9	9	0.004
17	9	0.001
33	7	0.0003

TABLE IV

$\alpha = 1.3$

Points	Nliter	Dev
3	46	0.00007
5	6	0.01
9	8	0.004
17	8	0.0009
33	7	0.0002

TABLE V

Divergence Point

α	Nliter	Center
4.1	13	0.50000
4.200	14	0.49994
4.201	>200	—
4.3	>200	—
5.3	>200	—
5.4	86	0.84667
5.5	42	0.15063

The bifurcation at $\alpha = 4.200$ is very abrupt. For $\alpha \geqslant 5.4$ the interior points for the computed grids all lie on one side of $r = \frac{1}{2}$, and thus, are highly skewed.

These numerical experiments indicate that the fully-lagged iteration has serious difficulties, in particular, the decrease of the bifurcation point with increasing grid size prevents the use of grid refinement to eliminate the convergence difficulties. The results for the nominal iteration (presented below) clarify the situation.

4.2. *The Nominal Iteration*

The code "arc2" implements the nominal iteration. For a one-free-point grid, the fully-lagged and nominal iterations are the same; the codes are different, so there are some minor differences in the runs, but nothing of any significance. The results in the previous subsection for $\alpha = 1.4$, $\sqrt{2}$, and 1.3 are essentially the same for both codes. This is expected, because the values of α are not near a bifurcation point. However, near and above the bifurcation points, the codes behave quite differently. Here, the use of the term "bifurcation" is appropriate because, the midpoint of the grid "falls off" the top of the curve, just as it does in the one-free-point case.

4.2.1. *Quadratic Curves.* As the theory in the section on continuum equations indicates, the bifurcation points for the fully-lagged and nominal iterations are different. For the nominal iteration, the bifurcation is not so abrupt and the bifurcation point is somewhat larger than it is for the fully-lagged iteration. The bifurcation points for the nominal iteration are as follows: for 21 points, $4.1 < \alpha < 4.2$; for 41 points, $5.7 < \alpha < 5.8$; and for 81 points, $7.6 < \alpha < 7.7$. These data indicate that the bifurcation point satisfies

$$\alpha \approx 0.8\sqrt{n}. \tag{4.93}$$

The fact that the bifurcation point grows with the grid resolution is a good feature of the nominal iteration; the nominal iteration limiting behavior for high resolution approaches that of the continuum model. Tables VI and VII illustrate the sensitivity of the grid to the initial conditions as well as the bifurcation of 21 point grids (all

Table VI

Bifurcation Point

α	Nliter	Center
4.1	8	0.50000
4.2	8	0.49994
4.3	8	0.50000
5.3	8	0.50000
5.4	32	0.42247
5.5	8	0.49999

parameters are the same as stated in the previous section). Note that the nominal iteration converges for values of the parameter above and below the bifurcation point.

The behavior presented in Table VI is odd; in particular, the center point returns to a value of 0.5 for large α. In Table VII, the initial grid is skewed to the left; the center point has the value 0.1 (otherwise all parameters are the same as stated above). Now, the bifurcation point can be seen clearly.

4.3. *Comments*

4.3.1. *Multiple Solutions.* When large numbers of numerical experiments are run on the nominal iteration for parameter values above the bifurcation point, a large number of solutions are found, dependent on the initial conditions. No pattern of interest is found in the solutions, therefore, these results are not presented.

4.3.2. *Improved Convergence.* Under-relaxing the nonlinear iteration improves convergence. Some numerical experimentation was used to find optimal relaxation factors. As expected, the factor depends on the shape of the curve and the number of grid points; the optimal factors are between 1 and $\frac{1}{100}$. Because nothing novel comes out of this work (and the additional parameter makes the results more complicated) it is not presented here. It is also possible to use multigrid algorithms to

Table VII

Skewed Initial Grid

α	Nliter	Center
4.1	97	0.50021
4.15	216	0.50094
4.2	71	0.51507
4.3	27	0.52881
4.5	14	0.54408
5.3	18	0.57491
5.5	58	0.57999

compute these grids; however, the bad behavior of the coarse grids presents special difficulties for these methods.

4.3.3. *Alternate Differencing Schemes.* Significant effort was expended on experimentation with alternate differencing formulas: centered, one-sided ("upwind"), and ZIP differencing. The results were uniformly negative. The alternates do not affect the qualitative fundamental difficulty of multiple solutions. Furthermore, the first-order accurate method introduces so much error ("artificial diffusion," in CFD terms) that when solutions are obtained for the easy (low amplitude) problems, the results are entirely unsatisfactory, i.e., the discrete solutions patently do not equidistribute the arc lengths, except for excessively fine resolutions.

5. Conclusions

The fact that the discrete grid-generation equations have multiple solutions is the most important conclusion. Once this is established, changing the method of solving the discrete equations does not help. If the method converges, then the solution found depends on the initial data. What is needed is a new formulation for the grid-generation problem that eliminates the multiple solutions.

The analysis of the continuum grid-generation problem shows that the Euler–Lagrange equation has a unique solution and this solution is a critical point of the length functional. Moreover, all second directional derivatives are positive at the critical point so it is a minimum. In sharp contrast to this, all discrete algorithms studied have a bifurcation after which the solution of the variational problem ceases to be unique.

The one-point grid undergoes a pitchfork bifurcation when the parameter in each of the example curves is increased. It is also possible to show that the minimization of an analogous discretized integral has an even worse bifurcation. The explicit results given in this problem also provide a check for the numerical codes. Note that these results do not distinguish between the fully-lagged and nominal iterations.

The numerical experiments show the true difficulty; the discrete equations have multiple solutions for curves of modest shape. The fully-lagged iteration behaves poorly as the resolution of the grid increases. The nominal iteration behaves better, converging for parameter values well above the bifurcation point. The fully-lagged iteration diverges so quickly as the curve-parameter values increase, that it provides little information about the grid-generation difficulty. The nominal iteration converges so well that it clearly delineates the bifurcation of the grid from a unique solution to multiple solutions.

The analysis done in this paper provides substantial insight into the difficulties (and possible remedies) inherent in the curve and surface grid-generation problem. There is good reason to anticipate analogous problems for solution-adaptive algorithms. Work is now proceeding on developing better algorithms [6]. One possibility is to not equidistribute the grid, but to distribute it so that its spacing is proportional to some function of the curvature.

References

1. D. A. Anderson, *Appl. Math. Comput.* **24**, 211 (1987).
2. J. E. Castillo, *SIAM J. Sci. Stat. Comput.*, to appear.
3. J. E. Castillo, *Appl. Numer. Math.*, to appear.
4. J. E. Castillo, in *Proceedings of 6th IMACS Meeting, Lehigh University, 1987*, edited by R. Vichnevetsky and R. S. Stepleman, (IMACS, New York, 1987), p. 501.
5. J. E. Castillo, Thesis, Dept. Math. Stat., University of New Mexico, 1987 (unpublished).
6. P. Knupp, Thesis, Dept. Math. Stat., University of New Mexico, 1989 (unpublished).
7. K. Matsuno and H. A. Dwyer, *J. Comput. Phys.* **77**, 40 (1988).
8. VAX UNIX MACSYMA Reference Manual, Version 11, Symbolics, Inc., 1985.
9. S. Steinberg and P. J. Roache, *Numer. Methods PDE* **2**, 71 (1986).
10. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North-Holland, New York, 1985).