Some Topical Variational Geometry Problems in Computer Graphics

G.N. Newsam.

Abstract.
Smooth interpolation of arbitrary curves and surfaces is a major problem in computer graphics. There are very successful variational formulations of similar problems in smooth interpolation of arbitrary functions; these have given rise to the (linear) theory of splines. However, there appears to be as yet no equivalent useful formulation of the general problem, so present computer graphics algorithms for curves and surfaces use somewhat ad hoc extensions of the linear results based on parametric representations of splines and surface patches. The paper briefly describes the present state of affairs in the hope that variational geometers will pick up on some of these unsolved problems and develop a coherent theory of smooth interpolation of arbitrary geometrical objects. If such a theory can be developed, it may possibly revolutionize computer graphics.

1. Introduction.

Computer graphics is one of the many amazingly fast growing technologies spawned in the last two decades (see [4] for a general reference). Apple's Macintosh brought 2D graphical interfaces to the masses, and standard workstations will soon feature interfaces that simulate reality with full 3D graphics. This has been brought to fruition by perhaps the most intensive application of geometry ever, but has been carried out mainly by engineers and computer scientists; surprisingly little has been done by geometers. This paper outlines some pressing problems in the field in the hope of enticing variational geometers to take a more active role in their solution.

The problems described here revolve around construction of smooth curves and surfaces subject to constraints, the most common being that they interpolate a given set of points. A large family of smooth interpolants of functions (the linear splines) are naturally defined through variational forms, yet there seem to be no formulations with equivalent explanatory power for general smooth interpolants. The standard constructions used in computer graphics are therefore based on parametric splines. Linear splines are easy to compute but result in coordinate dependent approximations, while parametric splines depend strongly on the chosen parameterization. In contrast, a general variational formulation holds the promise of a truly coordinate independent construction, but results so far have been disappointing. The hope is that variational geometers have both the insight and the tools to remedy this situation.

The question of what is the "correct" construction is of much more than academic interest. Considerable effort is put into implementing constructions in silicon so that they will run as fast as possible, and efficient constructions can greatly reduce the number of nodes needed to accurately represent a curve or surface. At present graphics chips automatically approximate curves or surfaces with geometric constructions such as lines, arcs of conics, Bezier curves and NURBS (non-uniform rational B-splines). There is as yet no compelling justification, however, for any particular approximation so the door is still open to new constructions of proven merit. Unfortunately, with the increasing trend towards adoption of standards there is a real possibility that existing constructions may soon be chosen as the basis of a future set of graphics approximation standards. This would erect a formidable barrier to further innovation, no matter how beneficial it might be. Thus the whole issue of smooth interpolants needs to be settled quickly. It may be that variational geometers have in fact already done so (I make no claim to be familiar with the literature in this field), in which case there is still a job to do in communicating the results to those who most need them.

The structure of the paper is as follows. Section 2 starts by outlining some of the major results in the smooth interpolation of functions of a single variable by polynomial splines, and their extension to the problem of fitting a smoothing spline through noisy function values. It then notes how these results form the basis of smooth interpolation of arbitrary curves by parametric splines. Finally it summarizes the successes and failings of this approach. Section 3 then looks at the smooth interpolation of an arbitrary 2D curve by minimization of the squared curvature; it sketches the form of the solution and indicates why this has proved unsatisfactory for practical applications. Section 4 looks at the smooth interpolation of surfaces by thin plate splines and surface patches, and notes that no general variational principle seems to exist for this problem. Finally section 5 summarizes the discussions in the previous sections as a list of open problems which may interest someone somewhere sufficiently to solve at least one or two of them.

2. Interpolation of curves by splines.

In order to illustrate the various important issues in smooth interpolation we briefly review the foundations of spline approximation to smooth functions and its achievements in fitting smooth graphs to given data sets. These achievements indicate what we would like of a more general theory for fitting arbitrary smooth curves to given data in two or more dimensions.

2.1 Analogue methods of curve fitting.

The original draughtsman's spline was a flexible piece of wood used in the drawing of smooth curves (especially cross-sections of ship hulls) through predefined points. Weights (known for some reason as ducks) were loaded on the curve at various locations to force it through the points and the shape of the curve was then traced out. Alternatively the points could be marked by pegs and the wood then bent around the pegs. The curve would assume a shape that minimized the bending energy away from the weights or pegs: in a thin rod this energy could be reasonably modelled as the square of the curvature of the curve.

2.2 Polynomial splines and smooth interpolation of functions.

Using pegs to form a shape gives a truly coordinate free curve invariant under rotation of the drawing board, but using weights will really only work well if the curve does not deviate too far from the horizontal. Since gravity imposes a preferred coordinate direction in many design problems, this is not an unreasonable restriction. Moreover it also leads to considerable simplification of the mathematical model. Suppose that the curve S is to pass through the points $\{x_i, y_i\} \ : \ i = 1, \ldots, N$ and that these points sufficiently nicely distributed that S will be a graph $y = f(x)$ w.r.t. to the x-y axes. Then the bending energy of S can be linearized w.r.t. to the coordinate system giving:

$$\int_S \kappa^2(s)\, ds \;=\; \int_{x_1}^{x_N} \frac{|f''(x)|^2}{1 + |f'(x)|^2}\, dx \;\sim\; \int_{x_1}^{x_N} |f''(x)|^2\, dx$$

Consequently S can be approximated as the solution to the constrained optimization problem:

$$\min_f \int_{x_1}^{x_N} |f''(x)|^2\, dx \qquad (1)$$

$$\text{subject to} \quad y_i = f(x_i) \qquad\qquad i = 1, \ldots, N$$

Setting this up as a Lagrangian and using calculus of variations shows that the solution f is given by:

$$\min_f \int_{x_1}^{x_N} |f''(x)|^2\, dx \;+\; \sum_{i=1}^{N} \lambda_i \left[\int_{x_1}^{x_N} f(x)\; \delta(x-x_i)\, dx \;-\; y_i \right]$$

$$\Leftrightarrow \qquad f''''(x) \;=\; -\sum_{i=1}^{N} \lambda_i\, \delta(x-x_i)$$

$$\text{with} \qquad f''(x_1) \;=\; f'''(x_1) \;=\; 0$$

$$\begin{aligned} f'''(x_N) &= f''(x_N) = 0 \\ f(x_i) &= y_i \end{aligned}$$

Thus f must be a cubic within each interval (x_i, x_{i+1}) and must be C^2 on the whole interval $[x_1, x_N]$. More generally, given a set of points x_i $i = 1, \ldots, N$, a polynomial spline $f(x)$ of degree m and smoothness k is a function that is a polynomial of degree m in each subinterval $(x_i x_{i+1})$ and is of order C^k over the whole interval. The points x_i are termed *knot* points. If in addition, $f(x)$ is a spline of order $2m+1$, smoothness m and is chosen to minimize $\int |f^{(m)}(x)|^2\, dx$ subject to $f(x_i) = y_i$ at the knot points, then f is termed a natural spline. [3] is the standard reference on polynomial splines, [11] describes splines in more general settings.

The most convenient formulation for determining f uses a basis of B-splines. A B-spline is a generic polynomial spline with finite support; in particular the cubic B-spline associated with the generic knot points $\{-2, -1, 0, 1, 2\}$ is:

$$B(x) = \begin{cases} (x+2)^3 & -2 \leq x \leq -1 \\ (x+2)^3 - (x+1)^3 & -1 \leq x \leq 0 \\ (2-x)^3 - (1-x)^3 & 0 \leq x \leq 1 \\ (2-x)^3 & 1 \leq x \leq 2 \end{cases}$$

Simple rescaling will map this generic spline to the cubic B-spline $B_i(x)$ on any set of 5 knot points $\{x_{i-2}, x_{i-1}\, x_i, x_{i+1}, x_{i+2}\}$. Now expanding f as:

$$f(x) = \sum_{i=1}^{N} a_i B_i(x)$$

gives a simple linear system $Ba = y$ for the coefficients a_i. The matrix B has entries $B_{ij} = B_i(x_j)$, so its only non-zero elements are on the main diagonal and its two off-diagonals, so the system can be rapidly solved in $O(N)$ operations. Moreover a complete stability analysis can be done that shows that the inversion is in general well-conditioned.

Error analysis for natural splines is fairly straightforward; the main result is that if $g \in C^{2m+2}$ and f is the natural spline of order $2m+1$ interpolating g at N knots, then:

$$||g^{(k)} - f^{(k)}||_\infty \leq c\, h^{2m+2-k} ||g^{(2m+2)}||_\infty$$

where $h = \max |x_i - x_{i-1}|$, and c is a constant independent of h and g.

Cubic splines have been extremely successful in many applications involving approximation of functions, interpolation of time series and suchlike. In particular, as well as being accurate, easily computed and easily interpreted, they are visually satisfying; it appears that the eye can detect discontinuities in the first or second derivatives of a function of a single variable, but that a C^2 or better function appears smooth. These successes set up the benchmarks that smooth interpolants of general curves are judged against, and have motivated the extensions of cubic splines to curve fitting that will be discussed later.

2.3 Smoothing splines.

In many practical situations the available data is noisy and is best modelled as:

$$y_i = f(x_i) + \varepsilon_i \qquad \varepsilon_i \sim N(0,\sigma^2)$$

where f is the unknown function to be estimated from the data. In such cases it is more appropriate to construct a smoothing spline f_λ that only approximately fits the data by trading off goodness-of-fit against smoothness. In particular, for a given λ, f_λ is defined to be the solution of:

$$\min_f \int_{x_1}^{x_N} /f''(x)/^2\, dx + \lambda \sum_{i=1}^{N} [f(x_i) - y_i]^2$$

Expanding f_λ in B-splines gives:

$$f_\lambda(x) = \sum_{i=1}^{N} a_{\lambda,i} B_i(x)$$

where the vector a_λ is the solution of the system:

$$(A + \lambda B^T B)^{-1} a_\lambda = B^T y \qquad \text{and} \qquad A_{ij} = \int_{-\infty}^{\infty} B_i''(x)\, B_j''(x)\, dx$$

If the variance σ^2 is known a priori, then an optimal choice can be made for λ that minimizes the expected error. However this is unlikely in practice so we need a robust automatic procedure for choosing λ based on the data alone. Fortunately this can be done through *cross-validation.* In this, for a given λ the smooth approximation $f_{\lambda,k}$ is defined to be the solution of the smoothing problem got by dropping the datum (x_k,y_k) from the data set, i.e. to solve:

$$\min_f \int_{x_1}^{x_N} /f''(x)/^2\, dx + \lambda \sum_{i \neq k} [f(x_i) - y_i]^2$$

We now define the cross-validation functional:

$$V(\lambda) \equiv \sum_{k=1}^{N} w_k \left[f_{\lambda,k}(x_k) - y_k\right]^2$$

$V(\lambda)$ essentially measures how good a particular λ is in predicting missing data values. The choice of the optimal λ^* is then the minimizer of $V(\lambda)$. It can be shown that this λ^* is asymptotically optimal in that as $N \uparrow \infty$ it tends to the λ that would be chosen if the variance was known. Moreover under certain reasonable choices for the weights w_k it can be shown ([5], see also [6, 11]) that $V(\lambda)$ need not be calculated by constructing $f_{\lambda,k}(x_k)$ separately for each k, but can be expressed in terms of λ, y and the generalized eigenvalues λ_i of the system $Ax_i = \lambda_i B^T B x_i$.

Again spline smoothing with the use of cross validation for estimation of the smoothing parameter has worked very well in a wide range of practical problems, and so one would hope that it could be included within the framework of a general variational construction for fitting arbitrary smooth curves through noisy data.

2.4 Parametric splines for interpolating curves in two or more dimensions.

The success of splines in interpolating functions has made them the basis for a number of constructions for interpolating arbitrary smooth curves in higher dimensions. These constructions, however, are all based on a parametric representation of the curve followed by a spline interpolation of each coordinate as a function of the parameter. As there appears to be no natural variational form associated with any of these representations, the justification for using any particular representation has to fall back on less intuitively satisfying reasons such as ease of use.

In order to define these splines we use the following notation. $x_i \equiv x(u_i) \equiv \{x_1(u_i), \ldots, x_d(u_i)\}$ will denote points on a curve in d dimensions parameterized by the variable u with knot points at u_i, $i = 1, \ldots, N$. A description of all these splines can be found in [4], but see also [9] for details on NURBS and [1] for details on β-splines.

Bezier splines

In these the interpolating spline is represented as:

$$x(u) = \sum_{i=1}^{N} a_i B_i^N(u)$$

where $$B_i^N(u) = \binom{n}{i} u^i (1-u)^{n-1}$$

are the Bernstein polynomials. The a_i are termed the control points. Bezier splines are perhaps the oldest of the various splines used to approximate arbitrary curves and are the basis of many smoothing routines in interactive drawing packages such as MacDraw. Note that the curve interpolates the first and last control points, so $a_1 = x_1$ and $a_N = x_N$ but the remainder of the control points must be found by solving a dense system of equations, whose conditioning will depend both on the data x_i and on the distribution of knots u_i. For this reason individual Bezier splines are often restricted to a low degree, and large curves are interpolated by patching together several splines. Continuity of the curve's tangent vector across patches can be enforced by noting that the direction of the tangent at an endpoint (say x_1) lies along the line $x_2 - x_1$. Note also that $x(u)$ always lies in the convex hull of the control points a_i, a property that also holds true for parametric B-splines and NURBS.

Parametric B-splines

In these the interpolating spline is represented as:

$$x(u) \quad = \quad \sum_{i=1}^{N} a_i B_i(u)$$

where the $B_i(u)$ are the B-splines defined above.

Non-Uniform Rational B-Splines (NURBS)

In these the interpolating spline is represented as:

$$x(u) \quad = \quad \frac{\sum_{i=1}^{N} a_i w_i B_i(u)}{\sum_{i=1}^{N} w_i B_i(u)}$$

where the $B_i(u)$ are again B-splines (almost always cubic splines). In addition to the properties noted above NURBS have some additional virtues. First, since conics can be parameterized by rational quadratics, NURBS can represent conic sections exactly. Second NURBS are invariant under affine and projective transformations. In particular, for affine transformations $x \rightarrow Ax+b$:

$$Ax(u) + b \quad = \quad \frac{\sum_{i=1}^{N} (Aa_i + b) w_i B_i(u)}{\sum_{i=1}^{N} w_i B_i(u)}$$

For perspective transformations, let $x \rightarrow \pi(x)$ denote the projection through a viewpoint c on to a plane that passes through the point p and has normal n. Then

$$\pi(x) \quad = \quad (1-\alpha)x + \alpha c \qquad \text{where} \qquad \alpha \quad = \quad \frac{(x-p) \bullet n}{(x-c) \bullet n}$$

and

$$\pi(x(u)) \quad = \quad \frac{\sum_{i=1}^{N} \pi(a_i)\, w_i B_i(u)}{\sum_{i=1}^{N} w_i\, B_i(u)}$$

This result is very useful in graphics displays since any 3D curve must be first projected onto the screen before it can be displayed. Thus a representation which is invariant under such transformations can be used for both 2D and 3D curve fitting without need for re-expression when changing views.

β-splines

β-splines were introduced to take advantage of the fact that geometric smoothness (e.g. continuity of the tangent vector and curvature) of a parameterized curve is a weaker requirement than parametric smoothness (e.g. C^2 continuity of the coordinates w.r.t. to u). In particular, starting with a parameterized cubic B-spline representation, and relaxing the condition of parametric smoothness to require only geometric smoothness frees up two parameters which can then be adjusted to meet other objectives. The resultant curve is written as

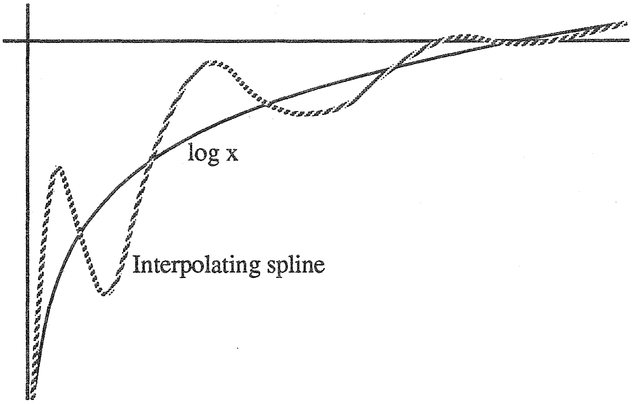
$$x(u) \quad = \quad \sum_{i=1}^{N} a_i\, \beta_i(u : \beta_1, \beta_2)$$

where β_1 controls the bias of the spline towards either endpoint and β_2 controls the tension (i.e. if $\beta_2 = 0$ then the spline is an ordinary B-spline, while if $\beta_2 = \infty$ the spline consists of straight line segments).

2.5 Strengths and weaknesses of interpolatory splines.

The main advantage of interpolatory splines for curve fitting is that the underlying theory is linear. The main disadvantage is that the basic theory of natural splines is derived in the context of approximating functions. The advantages of a linear theory are simple algorithms for curve fitting whose performance can be exhaustively analysed. The disadvantage is a setting that is not invariant w.r.t. coordinate rotation, and so many results do not carry through to the fitting of arbitrary curves. We now briefly list some of the consequences of this.

First, a non-invariant setting means that natural polynomial splines do not necessarily preserve many important geometric properties such as convexity. Moreover they will suffer from instability when approximating functions with infinite derivatives even if the function graphs are well behaved when viewed simply as curves in the plane. The following example shows a typical example of these problems: a cubic spline is used to approximate $log\ x$ and the approximation becomes increasingly oscillatory near the origin.



Nevertheless, viewed as a curve in the plane, $log\,x$ is a very well behaved convex curve and one would expect that a "true" interpolatory spline got by minimizing curvature would also be convex, and also be a very accurate approximation.

Second, the lack of an invariant variational formulation means that many useful results on function approximation by splines do not apply when approximating curves by parametric splines. In particular, any result depends crucially on the parameterization, and there seems to be no optimal way to specify this from the data (although an algorithm is given in [9] for determining knot points for NURBS that at least guarantees that the associated matrix B of spline values is totally positive definite and so can be inverted using Gaussian elimination without pivoting). I know of no results, for instance, on approximation of curves by parametric splines that parallel the error bounds on approximation of functions by natural splines. In practice most interpolation by parametric splines is done locally, so that the control points a_i are determined only by a few nearby x_i. This is necessarily done on essentially ad hoc basis. The same comments also apply to smoothing by parametric splines.

3. Interpolation of curves by splines minimizing curvature.

Given the drawbacks of existing methods of curve fitting, we could try and define smooth interpolants by a more exact model of the original draughtsman's spline. In particular, given data $\{x_i \equiv (x_i,y_i)\ :\ i = 1,\ldots,N\}$ we seek to fit a curve S through the data points that minimizes:

$$\int_S \kappa^2(s)\ ds$$

(where s denotes arc length) and to explore the properties of the solution.

3.1 Formal solution to the variational problem.

The minimization can be solved using much the same approach as for the linear case. In particular, we first parameterize the curve $S = \{x(s) : 0 = s_1 \le s \le s_N\}$ by arc length, and switch to intrinsic coordinates $(\varphi(s), s)$ where $\varphi(s)$ denotes the inclination of the curve to the horizontal at the distance s along the curve. Since $\kappa^2(s) = |\varphi'(s)|^2$, we have that the problem can be reformulated as: minimize over knot points s_i and angular functions φ the expression:

$$\int_{s_1}^{s_N} |\varphi'(s)|^2 \, ds$$

subject to: $$\int_{s_i}^{s_{i+1}} x'(s)\, ds = \int_{s_i}^{s_{i+1}} \begin{bmatrix} cos\ \varphi(s) \\ sin\ \varphi(s) \end{bmatrix} ds = x_{i+1} - x_i$$

Forming a Lagrangian as before by introducing the Lagrange multipliers $\lambda_i \equiv \rho_i\,(cos\ \psi_i, sin\ \psi_i)$ and solving the Euler equations shows that φ must satisfy:

$$\varphi''(s) = -\rho_i\ sin\,(\varphi(s) - \psi_i) \qquad s_i < s < s_{i+1}$$

subject to $$\varphi'(s_1) = \varphi'(s_1) = 0$$

and appropriate continuity conditions at the knots. An exact solution can be found for φ in terms of elliptic functions with two free parameters; this then leaves a system of nonlinear equations to be solved for these parameters plus the knots s_i and the multipliers λ_i.

3.2 Problems with the formal solution.

There are a number of theoretical problems with this minimization as well as the obvious practical problem of having to solve a fairly large nonlinear system of equations. First, there is no guarantee that a solution even exists. The reason for this is that it is always possible to reduce the value of the functional by appropriately lengthening the curve. To see why this is possible, consider a very large circle of radius R. Then the length of this is $2\pi R$ and its curvature is $1/R$, so the value of the functional is $2\pi/R$ which can always be reduced by increasing R. Thus even if a local minimum of finite length has been found, a lower cost solution can be produced by taking out a segment of this curve between any two knot points and replacing it by a very large loop.

Second, the splines produced by the formal solution are not always intuitively satisfying. For example, Lee and Forsythe [7] show that the local minimizer of square curvature interpolating points on a circle is not necessarily the circle.

3.3 Alternative approaches and unresolved issues.

The difficulties mentioned above mean that curvature minimizing splines are not widely used. A survey of algorithms for their construction are given in [8], while [2] presents an algorithm for constructing approximations based on spiral splines, in which the curvature is constrained to vary only linearly between data points. An even simpler algorithm can be derived based on approximation by arcs of circles; while this cannot be geometrically smooth, it has the advantages of being simple to calculate and of being composed of objects that are standard subroutines in many graphics systems.

The problem of non-existence of global minimizers may be due to an incorrect formulation; perhaps with an alternative choice of curvature functional a global minimum may exist. Likewise it may be possible to prove error bounds for smooth interpolants minimizing certain curvature functionals; or that particular geometric properties (such as convexity or circularity of the point data) are preserved by the interpolant; or that the functional form of the curve is preserved under perspective projection. Finally it would be very useful to find a formulation that allowed the rigourous derivation of smoothing curves for noisy data along the lines of the results for smoothing splines. These sort of results, however, appear to require the skills and insights of a specialist variational geometer. Nevertheless, if a consistent theory could be developed that established at least some of them, there would be considerable interest among numerical analysts and computer graphics specialists in actually constructing the interpolants the theory described.

4. Interpolation of surfaces.

We now look briefly at interpolation of surfaces and higher dimensional objects. Again there is a fairly comprehensive linear theory of interpolating splines for smooth graphs, but in contrast there seems to have been no practical work at all on constructing interpolating surfaces that minimize some general function of curvature. In many ways this is an even more pressing problem than construction of smooth curves, as the increase in full 3D graphics modelling, especially in CAD (computer aided design), has required the construction of very large smooth surfaces and therefore placed a premium on efficient representation. Moreover, as in many CAD models these surfaces are often used to actually define an object, these representations must also be invariant under rotation and suchlike transformations if they are to define the same object under different views.

To bring this point home, let us roughly estimate the number of parameters needed to accurately represent an smooth surface. I am aware of no useful results on the approximation of arbitrary surfaces, but consider instead the simpler problem of approximating a smooth function $f : \Re^d$

$\to \Re$ from an approximation space $S_h \subseteq C^{p-1}$ where h represents some measure of resolution. Then typically S_h will have dimension $N \sim O(h^{-d})$. Moreover the best approximation $\mathcal{P}_h f$ to f in S_h will be in error by $\|f - \mathcal{P}_h f\| \sim O(h^p)$, so that in order to ensure that $\|f - \mathcal{P}_h f\| \leq \varepsilon$ we will have to choose S_h to be sufficiently large that its dimension satisfies:

$$N \quad \sim \quad O((p/h)^d) \quad \sim \quad O((p/\varepsilon^{1/p})^d)$$

Thus by upping the order of the approximation we can substantially reduce the size of the approximation needed to accurately represent the function. This is of special importance when we consider that while display of the approximation takes only $O(N)$ operations, many other important tasks, such as finding a CAD object represented as a surface in a database of such objects may take $O(N^2)$ operations, or even more.

4.1 Thin plate splines.

The natural extension of Eq. (1) for defining a smooth interpolant to surface data is:

$$\min_f \int_{\Re^2} |f_{xx}(x,y)|^2 \; + \; 2\,|f_{xy}(x,y)|^2 \; + \; |f_{yy}(x,y)|^2 \; dx\, dy$$

$$\text{subject to} \quad z_i \; = \; f(x_i,y_i) \qquad\qquad i \; = \; 1, \ldots, N$$

Letting x denote the point (x,y), calculus of variations establishes that:

$$\Delta^2 f(\mathbf{x}) \quad = \quad - \sum_{i=1}^{N} \lambda_i \, \delta(\mathbf{x} - \mathbf{x}_i)$$

$$\text{with} \qquad f(\mathbf{x}_i) \quad = \quad z_i$$

plus the condition that $f(x)$ asymptote to a hyperplane as $|x| \uparrow \infty$. Therefore

$$f(x) \quad = \quad \sum_{i=1}^{N} a_i \, \varphi(|x - x_i|) \; + \; a_0 \; + \; a_{-1}x \; + \; a_{-2}y$$

$$\text{where} \qquad 0 \quad = \quad \sum_{i=1}^{N} a_i$$

$$\text{and} \qquad \varphi(r) \quad = \quad r^2 \ln r$$

The resulting splines are known as *thin plate splines* as the physical analogue of this procedure (corresponding to a draughtsman's use of a spline to interpolate a curve) is to load a thin metal plate with weights at the locations x_i and so force it to take on heights z_i. Thin plate splines have been successfully used in a number of applications [11], and can be generalized to interpolate surfaces in higher dimensions or with smoother functions; the basic spline derived by minimizing derivatives of order m in d dimensions is:

$$\varphi(r) = \begin{cases} r^{2m-d} \ln r & d \text{ even} \\ r^{2m-d} & d \text{ odd} \end{cases}$$

Extensions to the case of smoothing splines are straight-forward; Dr. M. Hutchinson of CRES at ANU has a FORTRAN package that uses cross validation to fit smooth thin plate splines through arbitrary data sets.

4.2 Surface patches.

Unlike B-splines, thin plate splines do not form the basis for any surface constructions in computer graphics. The main reason for this is that they have infinite support and so are not suitable for modelling essentially localized surfaces and cannot be easily modified. Instead most surface modelling in computer graphics is done using quadrilateral patches formed by taking tensor products of the parametric splines described in section 3. A typical patch composed of B-splines would be:

$$x(u,v) = \sum_{i=1}^{M} \sum_{j=1}^{N} a_{ij} B_i(u) \, B_j(v)$$

where the a_{ij} are now control points in $\Re^3$ and the patch is defined over a grid of knots formed by taking the product of the two knot vectors for u and v respectively.

The patches themselves are usually C^2 or better, but it is difficult to ensure better than C^1 continuity when joining two patches. This is actually not a major problem as it when it comes to surfaces, it seems that a surface is visually smooth simply as long as its tangent plane is continuous. Moreover this geometric continuity is even weaker than the C^1 parametric continuity achieved by most patch joins. Patches run into difficulty, however, when fitting data that can not be arranged onto an essentially rectangular grid of knot points; while the odd irregular element within a grid can be coped with, truly irregular grids present very messy problems. More generally, surface fitting by patches is an involved and intricate discipline, albeit one of great importance (see [9, 10] for further details).

4.3 Variational problems in surface fitting.

As already noted there appears to be no available results on characterization or calculation of smooth interpolants that minimize curvature measures. There is some reason to believe that these interpolants may differ significantly from the thin plate splines described in section 4.1; the singularities at the origin (and therefore at data points) exhibited by thin plate splines may not be the same as those of true curvature minimizing interpolants at data points. Nevertheless both the general shape of such interpolants and any properties they may have is still obscure, and equally importantly there is no obvious way of actually calculating them.

There are a number of other important variational problems in surface modelling by patches. In particular a surface defined by a very large and possibly irregular grid of patches often needs to be systematically re-gridded to produce a smaller, more regular but essentially equally accurate representation. This problem arises very commonly in CAD surfaces constructed from measuring real world shapes. Data may be collected, for example, by shining a finely spaced, regular grid of bright lights onto the surface and then measuring the resultant distortion of the grid in an image of the surface: the distortion gives the depth of the surface in the direction of view. This and similar methods have two problems: they tend to heavily over-sample smooth regions of the object; and they produce very irregular grids in areas where the object's local topology is definitely not a graph (e.g. where the handle of a cup joins its body).

If the surface is topologically equivalent to a disk, one possible solution is to find a conformal mapping $\mathcal{F}$ of the surface onto the unit disk (or square). Next a uniform grid can be placed on the disk and then mapped back to the surface by $\mathcal{F}^{-1}$. The resulting grid on the surface will also be regular and so suitable for patch construction. Moreover the surface grid should be concentrated in regions of high curvature, but spread out in smoother regions. Conformal maps can be constructed as the minima of certain curvature functionals; the question is whether these are the most suitable variational forms for this particular problem.

5. Summary of open questions.

In conclusion, it seems that the following questions are still open. Positive answers to them would be not only of considerable theoretical interest in extending existing results for the linear theory of approximating smooth graphs to the approximation of smooth curves, but also of great practical import for computer graphics. The theoretical questions are:

1. Is there a natural variational formulation for defining a smooth curve / surface to interpolate arbitrary data in two or more dimensions?

2. Can the solutions to such formulations be characterized?

3. Do the solutions preserve general geometric properties such as convexity?

4. Can useful error bounds be derived for the approximation of smooth curves / surfaces?

5. Can the formulations be extended to define smoothing curves / surfaces for noisy data, and do these have same form as the exact interpolants? Can smoothing be done robustly and automatically?

6. Is the general functional form of the interpolants invariant under perspective transformations?

The more practical questions are:

7. What is the "best" set of graphics primitives for generating curves and surfaces in software and hardware?

8. Can the solutions be efficiently computed?

Finally, in regard to the last question, the nonlinear nature of the problems means that iterative methods are likely to be used for their solution. Consider the general problem:

$$\min_{S} \; V(S) \;\equiv\; \int_{S} I(\kappa(s))\, ds$$

subject to various constraints

where $I(\kappa)$ is some function of curvature. Direct application of steepest descent would give (constrained) iterates of the form:

$$S_t - S_{t-1} = -\rho_t \nabla V(S_t)$$

This looks very similar to general surface flow problems of the form:

$$\frac{dS(t)}{dt} = -\nabla V(S(t))$$

that have been widely studied by variational geometers, so that it may be possible to use existing results both to prove existence of solutions and convergence of numerical algorithms to them. Furthermore, higher order iterative algorithms such as Newton's method:

$$S_t - S_{t-1} = -[H_V(S_t)]^{-1} \nabla V(S_t)$$

where H_V is the Hessian, can also be translated into surface flow problems of the form:

$$\frac{dS(t)}{dt} = -[H_V(S(t))]^{-1} \nabla V(S(t))$$

In the light of the many extra stability and convergence properties that Newton's method has in comparison with steepest descent, it is interesting to speculate what extra convergence properties the above surface flow may have.

Acknowledgements.

I would like to thank Ian Coope for some very helpful pointers and references on the problem of finding the interpolant with minimum square curvature, and the Centre and DITAC for the opportunity and support to attend the conference.

References

[1] R. Bartels, J. Beatty and B. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*, Morgan Kaufman, Los Altos, California, 1987.

[2] I. Coope, Curve fitting with nonlinear spiral splines, preprint, 1991.

[3] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.

[4] J.D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, Massachussetts, 1990.

[5] G. Golub, M. Heath and G. Wahba, Generalized cross validation as a method for choosing a good ridge parameter, *Technometrics*, **21** (1979), 215-24.

[6] M. Hutchinson and F. de Hoog, Smoothing noisy data with spline functions, *Numer. Math.*, **47** (1985), 99-106.

[7] E.H. Lee and G.E. Forsythe, Variational study of nonlinear spline curves, *SIAM Rev.*, **15** (1973), 120-33.

[8] M.A. Malcolm, On the computation of nonlinear spline functions, *SIAM J. Numer. Anal.*, **14** (1977), 254-82.

[9] L. Piegl, On NURBS: A summary, *IEEE Comp. Graphics & Applics.*, (1991), 55-71.

[10] R.F. Sarraga, Computer modelling of surfaces with arbitrary shapes, *IEEE Comp. Graphics & Applics.*, (1990), 67-77.

[11] G.Wahba, *Spline Models for Observational Data*, SIAM, Philadelphia, 1990.

Information Technology Division,
Electronics Research Laboratory,
DSTO,
P.O. Box 1600,
Salisbury, S.A. 5108,
Australia.