Efficient one-sided linearization of spline geometry

Jörg Peters

University of Florida

Abstract. This paper surveys a new, computationally efficient technique for linearizing curved spline geometry, bounding such geometry from one side and constructing curved spline geometry that stays to one side of a barrier or inside a given channel. Combined with a narrow error bound, these reapproximations tightly couple linear and nonlinear representations and allow them to be substituted when reasoning about the other. For example, a subdividable linear efficient variety enclosure (sleve, pronounced like Steve) of a composite spline surface is a pair of matched triangulations that sandwich a surface and may be used for interference checks. The average of the sleve components, the *mid-structure*, is a good max-norm linearization and, similar to a control polytope, has a well-defined, associated curved geometry representation. Finally, the ability to fit paths through given channels or keep surfaces near but outside forbidden regions, allows extending many techniques of linear computational geometry to the curved, nonlinear realm.

1 Introduction

Compared to piecewise linear, say triangle meshes, higher-order representations, such as b-splines, Bézier patches and subdivision surfaces, offer improved compression (if the target shape is sufficiently smooth) through higher approximation order, arbitrary resolution to any prescribed tolerance, and generally a higher level of abstraction. On the other hand, meshes and triangulations are pervasive, as a result of measuring, for end-uses, such as graphics rendering or driving machine tools, and for the majority of finite element codes. No wonder then that b-spline, Bézier and subdivision control nets look attractive as mediator: on one hand, they approximate a nonlinear shape; on the other hand, they finitely, but completely, define the smooth, nonlinear shape. Armed with the convex hull property and the convergence under subdivision, it is therefore tempting to use the control meshes as end-use or computational meshes. However, control meshes have shortcomings. The control net is far from the best possible geometric approximand for the a given budget of linear pieces. It can cross the curve or surface and therefore does not provide a one-sided approximation. Finally, until recently, there was no theory giving easy access to the error (not just the rate of decay under subdivision) of the distance of the control net to the actual curved object. Consequently, despite their geometrically indicative control structure, objects in b-spline, Bézier or generalized subdivision representation pose numerical and implementation challenges, say when measuring distance between objects, re-approximating for format conversion, meshing with tolerance, or detecting the silhouette. It should be emphasized that naive linearization, such as triangulation by sampling, reapproximates without known error and not safely from one side.

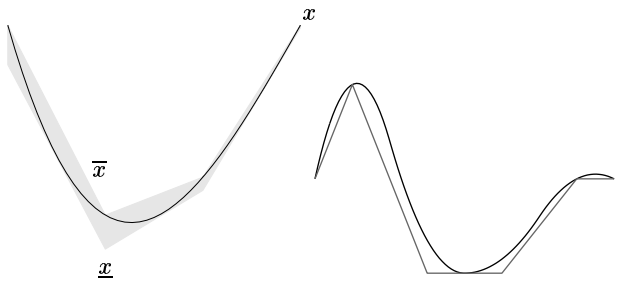


Fig. 1. (*left*) A cubic B´ezier segment x and its optimized, grey 3-piece slefe bounded by $\overline{x}$ and $\underline{x}$. (*right*) Spline curve computed to be above but close to a given piecewise linear barrier.

Subdividable linear effi cient function enclosures [7,8,9], short slefes (pronounced like sleve), by contrast, are a low-cost technique for constructing piecewise linear bounds that sandwich nonlinear functions. The slefe of a function x consists of 2 one-sided bounds $\overline{x}, \underline{x}$ so that $\underline{x} \le x \le \overline{x}$ over the domain of interest. Here $\overline{x}$ and $\underline{x}$ are, for example, piecewise linear functions bounding the grey region in Figure 1, *left*. In practice, slefe bounds are observed to be very tight. Analytically, at present, only the case of cubic functions in one variable is fully understood: the slefe width of a convex x is within 6% of the optimal, narrowest; at an inflection, the error ratio is at most 3:5. The extension of slefes to free-form surfaces requires some care – but the resulting algorithm is still rather simple and sleves inherit many of the properties of slefes such as near-optimality in the L^∞ norm and refi nability for adaptive multiresolution.

The average $\overline{\underline{x}} := (\overline{x} + \underline{x})/2$ is called *mid-structure*. It is well-defi ned also for vector-valued functions $\mathbf{x}$ and its construction does not require the construction of the sleve of $\mathbf{x}$. By making the mid-structure along a boundary depend only on the boundary, e.g. a space curve for a patch in $\mathbb{R}^3$ or the endpoint for a curve, mid-structures join continuously if their patches join continuously. In contrast to approximation theory, which strives to establish optimality and uniqueness over all suffi ciently smooth functions, mid-structures are a concrete, effi ciently computable approximation with a small, quantifi able L^∞ error. (We recall that Chebyshev economization applies to degree reduction by one polynomial degree and only to a single polynomial segment. The problem at hand is to determine a best continuous, piecewise linear, max-norm approximation of a nonlinear curve.)

Since the construction of slefes is simple, inverse problem of one-sided linearization can also be effi ciently addressed: to fi nd a spline (from a fi xed spline space) that stays close to but to one side of a given piecewise linear curve (Figure 1, *right*). In the related CHANNEL problem, a channel is given and a smooth spline is sought that stays inside that channel. Inverse problems address underconstrained design problems where the emphasis is not on optimality or uniqueness of the solution but on feasibility or on pinpointing the cause of infeasibility (which might trigger a refi nement of the spline space). Such tools should improve design and shorten the design cycle and extend techniques of computational geometry to the domain of curved smooth paths and surfaces. For example, solutions to the inverse problem yield a postprocessing scheme that smoothes edges to one side while preserving the essential quality of the series of

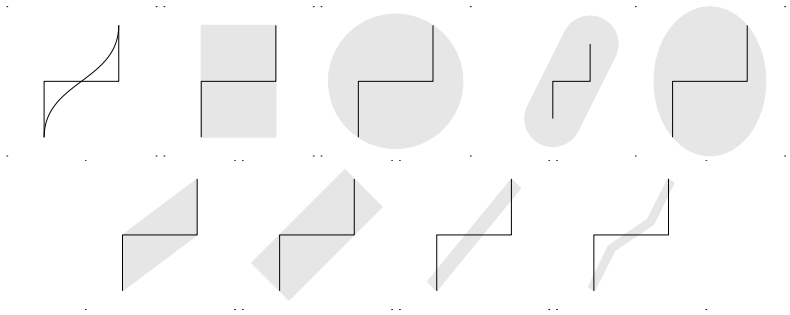


Fig. 2. Enclosures based on control points: cubic curve with B´ezier control polygon, axis-aligned box, bounding circle, Filip et al. bound (scaled by 1/2), bounding ellipse convex hull, oriented bounding box, 'fat arc', 3-piece sleve.

straight edges generated as: upper hulls, edges of a binary space partition, parts of triangulations, road maps or visibility graphs.

Overview: Section 1.1 below contrasts sleves with other commonly used bounding constructs and points to prior work. Section 2, page 4, reviews the slefe construction, for functions in one variable and for tensored multivariate functions. Section 3, p. 10, discusses midstructures in more detail. Curve and surface sleves are explained in Section 4.1, p.12 and a solution strategy for inverse problems, in Section 5, p.16.

1.1 Commonly used bounding constructs

The enclosure of a geometric object is a *bounding construct*, consisting typically of two sheets (or more, say in the case of a space curve), such that each sheet is guaranteed not to intersect the object, i.e. each sheet lies to 'one side' of the object. For example, a surface without boundary can be enclosed by an inner and an outer triangulation.

We distinguish between elementary bounding constructs and hierarchical structures that employ these elementary bounding constructs as their oracles. enclosures fall into the category of elementary bounding constructs. A gallery of elementary bounding constructs is shown in Figure 2. That is, enclosures add to the arsenal of axis-aligned bounding boxes (AABB), oriented bounding boxes (OBB), quantized bounding boxes also called 'k-dops' or discrete orientation polytopes (convex polytopes whose facets are determined by halfspaces whose outward normals come from a small fi xed set of k orientations) [5, 11, 12], fat arcs [28], convex hulls, bounding spheres and minimal enclosing ellipsoids [30] and the Filip et al. bound [9]. The Filip et al. bound is based on the observation that, on $[0..1]$, the difference between a function f and its linear interpolant at 0 and 1 is bounded by $\|f''\|_{[0..1]}/8$. For a polynomial of degree n the latter is bounded by $n(n-1)$ times the maximal second difference of the Bézier control points. Note that all bounds can be improved by subdividing the curve segment as part of a recursive process. Publications [10] and [13] give a good overview of how elementary bounding constructs are used in the context of hierarchical interference detection (for space partitioning methods see e.g. [2]): simpler constructs like AABBs and spheres

provide fast rejection tests in sparse arrangements, while more expensive k-dops and OBBs perform better on complex objects in close proximity; sleves with adaptive resolution fall in the more expensive comparison category and promise to outperform other bounding constructs for curved, non-polyhedral objects in close proximity, due to their basis-specific pre-optimization that is done off-line (and tabulated once and for all) and local refinability.

The theory of slefes has its roots in bounds on the distance of piecewise polynomials to their Bézier or B-spline control net [18, 25]. Compared to these constructions, slefes yield dramatically tighter bounds for the underlying functions since they need not enclose the control polygon. Approximation theory has long recognized the problems of one-sided approximation and two-sided approximation [3]. Algorithmically, though, according to the seminal monograph [24], page 181, the convergence of the proposed Remez-type algorithms is already in one variable 'generally very slow'. The only termination guarantee is that a subsequence must exist that converges. By contrast, the slefes provide a solution with an explicit error very fast and with a guarantee of error reduction under refinement.

Surface simplification for triangulated surfaces has been modified to generate (locally) inner and outer hulls [4, 26]. This requires solving a sequence of linear programs at runtime and applies to already triangulated surfaces. The object oriented bounding boxes for subdivision curves or surfaces in [14] are based on a min–max criterion and require the evaluation of several points and normals on the curve or surface. Thus the dependence on the coefficients is not linear. Linearity of the slefe construction allows us to solve inverse problems, like the CHANNEL problem mentioned earlier. Farin [8] shows that for rational Bézier–curves, the convex hull property can be tightened to the convex hull of the first and the last control point and so-called *weight* points.

Starting with Farouki and Sederberg [27] the use of *interval spline representation* for tolerancing, error maintenance and data fitting has been promoted in a series of papers, collected in a recent book by Patrikalakis and Maekawa [19] (see also [20]). The key ingredient is the solution of nonlinear polynomial systems in the Bernstein basis through rounded interval arithmetic. This, in turn, relies on AABBs based on the positivity and partition of unity property of spline representations. sleves complement this work: while interval spline representations focus on uncertainty of the control points, sleves offer tight two-sided bounds for nonlinear (bounding) curves or surfaces.

2 Subdividable linear efficient function enclosures

2.1 The basic idea

The subdividable linear efficient function enclosure, or slefe of a function x with respect to a domain U is a piecewise linear pair, $\overline{x}, \underline{x}$, of upper and lower bounds that sandwich the function on U: $\overline{x} \geq x \geq \underline{x}$. The goal is to minimize the width,

$$w(x, U) := \overline{x} - \underline{x},$$

in the *recursively applied L^∞ norm*: the width is as small as possible where it is maximal – and, having fixed the breakpoint values where the maximal width is taken on

(zeroth and first breakpoint in Fig. 3), the width at the remaining breakpoints is recursively minimized subject to matching the already fixed break point values.

Slefes are based on the two general lemmas [16, 17] (Section 2.2 gives concrete examples), and the once-and-for-all tabulation of best recursive L^∞ enclosures of a small set of functions, $\mathbf{a}_i$, $i = 1, \ldots, s$ collected into a vector $\mathbf{a}$ below. With $\mathbf{b}$ is a vector of (basis) functions and $\mathtt{x}$ a vector of coefficients, we use the following notation below:

$$x = \mathbf{b} \cdot \mathtt{x} := \sum \mathbf{b}_i \mathtt{x}_i .$$

Lemma 1 (change of basis). *Given two finite-dimensional vector spaces of functions, $\mathcal{B} \neq \mathcal{H}$, $s := \dim \mathcal{B} - \dim(\mathcal{B} \cap \mathcal{H})$, $(\mathbf{b}_i)_{i=1,\ldots,\dim \mathcal{B}}$ a basis of $\mathcal{B}$, $(\mathbf{a}_i)_{i=1,\ldots,s}$ functions in $\mathcal{B}$, and linear maps*

$$L : \mathcal{B} \to \mathcal{H},\ \Delta : \mathcal{B} \to \mathbb{R}^s ,$$

such that (i) $(\Delta_j \mathbf{a}_i)_{i,j}$ is the identity in $\mathbb{R}^{s \times s}$ and (ii) $\ker \Delta = \ker(E - L)$ (where E is the embedding identity) then for any $x := \mathbf{b} \cdot \mathtt{x} \in \mathcal{B}$,

$$(\mathbf{b} - L\mathbf{b}) \cdot \mathtt{x} = (\mathbf{a} - L\mathbf{a}) \cdot (\Delta x).$$

Proof. By (i) $\Delta(I - \mathbf{a}\Delta)\mathbf{x} = 0$ and hence, by (ii), $(E - L)(I - \mathbf{a}\Delta)\mathbf{x} = 0$.

Remarks: We can extend the lemma, say to the bi-infinite spline setting, by defining $s = \infty$ if $\dim \mathcal{B} = \infty$; however, for practical computation, $(\mathbf{a} - L\mathbf{a}) \cdot (\Delta x)$ has to have finitely many terms. In (ii), $\ker \Delta \subset \ker(E - L)$ is needed since for any $x \in \ker \Delta \setminus \ker(E - L)$, $(\mathbf{a} - L\mathbf{a}) \cdot (\Delta x)$ is zero, but not $(\mathbf{b} - L\mathbf{b}) \cdot \mathtt{x}$. Since the width of the enclosure changes under addition of any element in $\ker(E - L) \setminus \ker \Delta$, we also want $\ker(E - L) \subset \ker \Delta$.

Lemma 2 (bounds). *If, with the definitions of Lemma 1, additionally the maps $x \mapsto \underline{x} : \mathcal{B}^s \to \mathcal{H}^s$ and $x \mapsto \overline{x} : \mathcal{B}^s \to \mathcal{H}^s$ satisfy $\underline{\mathbf{a} - L\mathbf{a}} \leq \mathbf{a} - L\mathbf{a} \leq \overline{\mathbf{a} - L\mathbf{a}}$ componentwise on every point of a domain U, and $(\Delta x)_+(i) := \max\{0, \Delta x(i)\}$, and $(\Delta x)_-(i) := \min\{0, \Delta x(i)\}$ then*

$$\underline{x} := Lx + \underline{\mathbf{a} - L\mathbf{a}} \cdot (\Delta x)_+ + \overline{\mathbf{a} - L\mathbf{a}} \cdot (\Delta x)_- ,$$

$$\overline{x} := Lx + \underline{\mathbf{a} - L\mathbf{a}} \cdot (\Delta x)_- + \overline{\mathbf{a} - L\mathbf{a}} \cdot (\Delta x)_+$$

sandwich x on U: $\underline{x} \leq x \leq \overline{x}$.

The general **slefe construction** is as follows where (1),(2),(3),(4) are precomputed, offline and (5) is easy to compute.

1. Choose U, the domain of interest, and the space $\mathcal{H}$ of enclosing functions.
2. Choose a difference operator $\Delta : \mathcal{B} \mapsto \mathbb{R}^s$, with $\ker \Delta = \mathcal{B} \cap \mathcal{H}$.
3. Compute $\mathbf{a} : \mathbb{R}^s \mapsto \mathcal{B}$ so that $\Delta\mathbf{a}$ is the identity on $\mathbb{R}^s$ and each $\mathbf{a}_i$ matches the same $\dim(\mathcal{B} \cap \mathcal{H})$ additional independent constraints.
4. Compute $\underline{\mathbf{a} - L\mathbf{a}}$ and $\overline{\mathbf{a} - L\mathbf{a}} \in \mathcal{H}$.
5. Compute $(\Delta x)_+$ and $(\Delta x)_-$ and assemble $\underline{x}$ and $\overline{x}$ according to Lemma 2.

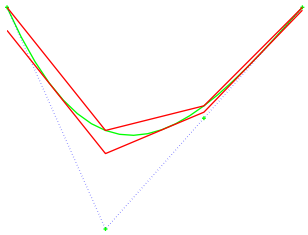


Fig. 3. $a := \mathbf{a}_1^3 := -(2\mathbf{b}_1^3+\mathbf{b}_2^3)/3$ with control polygon and slefe.

2.2 Example in one variable

For a concrete example of the general framework in the previous section, let

- $\mathcal{B}$ be the space of univariate polynomials of degree d, in Bézier form

$$x(u) := \textstyle\sum_{i=0}^{d} x_i \mathbf{b}_i^d(u),\ \mathbf{b}_i^d(u) := \frac{d!}{(d-i)!i!}(1-u)^{d-i}u^i.$$

  Specifically, we choose $d = 3$.
- $\mathcal{H}$ the space of piecewise linear functions $\mathbf{h}_j$ with break points at $j/m, j \in \{0, \dots, m\}$. Specifically, we choose $m = 3$ segments.
- Δx the $d-1$ second differences of the Bézier coefficients

$$\Delta x := \left[\begin{smallmatrix}\Delta_1 x\\ \Delta_2 x\end{smallmatrix}\right] := \left[\begin{smallmatrix}x_0-2x_1+x_2\\ x_1-2x_2+x_3\end{smallmatrix}\right]$$

- $Lx(u) := x_0(1-u) + x_d u$, and therefore $\mathbf{a}_i^d - L\mathbf{a}_i^d = \mathbf{a}_i^d$.
- $U = [0..1]$.

This yields

$$\mathbf{a}_1^d := -(2\mathbf{b}_1^d + \mathbf{b}_2^d)/d, \quad \mathbf{a}_2^d := -(\mathbf{b}_1^d + 2\mathbf{b}_2^d)/d,$$

$$x - Lx = \mathbf{a}_1^d \Delta_1 x + \mathbf{a}_2^d \Delta_2 x.$$

By symmetry, it is sufficient to compute the optimal enclosures for $a := \mathbf{a}_1^3$. Due to the convexity of a (see Fig. 3), the piecewise linear interpolant at j/m is an upper bound. We write $\overline{a} := \overline{a}^m$ (where m indicates the number of linear segments of the upper bound) as the vector of its breakpoint values, e.g. the value of $\overline{a}$ at $1/3$ is $-10/27$:

$$27\overline{a} \simeq [0, -10, -8, 0].$$

The lower bound is computed by recursive minimization. The first segment is the dominant segment in the sense that its tightest bound has the largest width among the three segments (see Figure 3 – the genral case is covered in Lemma 5 of [21]). Therefore, we calculate the values of $\underline{a}$ at 0 and $1/3$ by shifting down the first segment of the upper

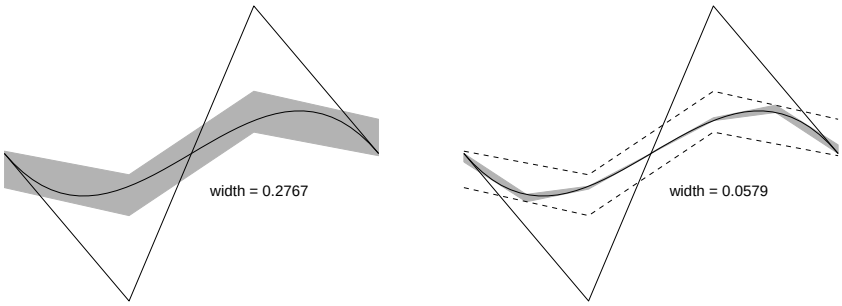


Fig. 4. (*left*) A cubic B´ezier segment with coefficients $0, -1, 1, 0$. The control polygon exaggerates the curve far more than the *grey* 3-piece slefe. (*right*) After one subdivision at the midpoint, the width of the slefe (*grey*) is roughly 1/4th of the width of the unsubdivided slefe (*dashed*).

bound until it is tangent to a. The other two break point values are computed by calculating the tangent line to a, keeping one end fixed. This procedure yields the $m+1$ break point values of the lower bound

$$27\underline{a} \simeq [30, 20, 25 + \frac{\beta_1 - 9}{2}\beta_2, \beta_3] - \frac{38\beta_1}{9},$$

where

$$\beta_1 := \sqrt{57}, \quad \beta_2 := \sqrt{-10 + 2\beta_1},$$
$$\beta_3 := \frac{261}{8} + \frac{\beta_1 - 9}{4}\beta_2 + \frac{3\beta_2 - \beta_1}{8}\sqrt{11 - 12\beta_2 - 2\beta_1 + 2\beta_1\beta_2}.$$

An approximation of the values is $\underline{a}_m \approx [-.0695, -.4399, -.3154, -.0087]$. Evidently, the width

$$w_{\text{slefe}}(x; U) := \max_U \overline{x} - \underline{x} = \max_U \sum_{i=1}^{d-1} (\overline{\mathbf{a}_i - L\mathbf{a}_i}^m - \underline{\mathbf{a}_i - L\mathbf{a}_i}_m)\ |\Delta_i x|$$

is invariant under addition of constant and linear terms to x and one (DeCasteljau) subdivision step at the midpoint, $t = 1/2$ cuts the width to roughly *a quarter* (see Figure 4) since $(\overline{\mathbf{a}_i - L\mathbf{a}_i}^m - \underline{\mathbf{a}_i - L\mathbf{a}_i}_m)$ stays fixed and the maximal $\Delta_i x$ shrinks to $1/4$ its size.

2.3 How good are slefes?

slefe-based bounds are observed to be very tight. Yet, being linear, the slefe construction cannot be expected to provide the best two-sided max-norm approximation, a difficult nonlinear problem. Therefore, it is of interest to see how close to optimal the slefe construction actually is by deriving and comparing it with the narrowest possible enclosure with the same breakpoints. In this section, we determine, for a class of functions, the optimal enclosure width, w_{opt}, and compare it with w_{slefe}. The simplest nontrivial

case is when the function x is a univariate quadratic polynomial; however, in this case, the slefe construction is optimal, because the vector of functions $\mathbf{a} - L\mathbf{a}$ is a singleton and slefes are based on the optimal enclosures of $\underline{\mathbf{a} - L\mathbf{a}}, \overline{\mathbf{a} - L\mathbf{a}}$. Since explicit determination of the least-width, piecewise linear enclosure is a challenge, we consider polynomials x of degree $d = 3$ in Bézier representation on the interval $U = [0..1]$. Generalization of the results to $m \neq 3$ pieces is not difficult; generalization of exact bounds to degree $d > 3$ appears to be non-trivial. Without loss of generality, we assume $\Delta_1 x \geq |\Delta_2 x|$ in the following.

Computing $w_{\mathbf{slefe}}$ Let $w_i := (\overline{a}^m - \underline{a}_m)(i/m)$, $i = 0, 1, \ldots, m$. Then, due to the symmetry $\mathbf{a}_1^3(t) = \mathbf{a}_2^3(1-t)$,

$$w_{\text{slefe}}(x; U) = \max_{i \in \{0,1,2,3\}} \{|\Delta_1 x| w_i + |\Delta_2 x| w_{m-i}\}.$$

Since $w_0 = w_1 > w_2 > w_3$, the term with $i = 1$ is the maximal term, and

$$w_{\text{slefe}}(x; U) = |\Delta_1 x| w_1 + |\Delta_2 x| w_2 \approx 0.0695 |\Delta_1 x| + 0.0191 |\Delta_2 x|.$$

If we set $|\Delta_1 x| := 1 + \epsilon$, $\epsilon \in [0..\infty]$, and $|\Delta_2 x| := 1$ then

$$w_{\text{slefe}}(x; U) = -\frac{1}{243}(270\epsilon + 567 + \frac{9\beta_2}{2}(\beta_1 - 9) - 38\beta_1(\epsilon + 2)).$$

Computing $w_{\mathbf{opt}}$ To determine the width of the *narrowest* possible piecewise linear enclosure for x, $w_{\text{opt}}(x, [0..1])$, with breakpoints at $i/3$, elementary considerations show that it is sufficient to compare the width of functions with first and last coefficient equal zero, that then an increase of the second derivative of x increases w_{opt}; and finally, since $\Delta_1 x > |\Delta_2 x|$, $w_{\text{opt}}(x, [0..1/3]) > w_{\text{opt}}(x, [0..1])$, i.e. the first segment determines w_{opt}.

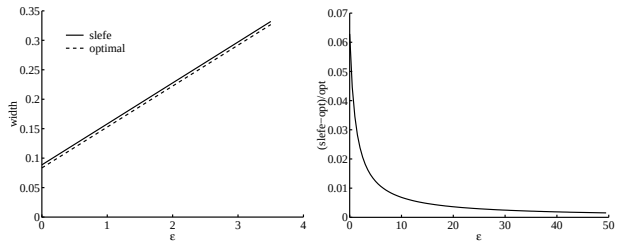


Fig. 5. Convex cubic x. (*left*) Value of $w_{\text{slefe}}(x; U)$ *vs* $w_{\text{opt}}^{\cup}(x; U)$. (*right*) The ratio $\frac{w_{\text{slefe}} - w_{\text{opt}}^{\cup}}{w_{\text{opt}}^{\cup}}(x; U)$.

Comparison of $w_{\mathbf{opt}}$ and $w_{\mathbf{slefe}}$ We first consider the case of *no inflection*. Without loss of generality, $\Delta_1 x := 1+\epsilon, \epsilon \in [0,\infty]$, $\Delta_2 x := 1$ and with $A := \sqrt{57\epsilon^2 + 135\epsilon + 81}$,

$$w^{\cup}_{\mathrm{opt}}(x;U) := -\frac{1}{243}\frac{(9+9\epsilon-A)(-3A(1+\epsilon)+11\epsilon^2+36\epsilon+27)}{\epsilon^2}.$$

Figure 5, *left* plots w_{slefe} against $w^{\cup}_{\mathrm{opt}}$. The gap between w_{slefe} and $w^{\cup}_{\mathrm{opt}}$ increases with ϵ but is finite at infinity:

$$w_{\mathrm{slefe}}(x,\infty) - w^{\cup}_{\mathrm{opt}}(x,\infty) \approx .0053353794.$$

The relative difference has a maximum of ca. 6% when $\epsilon = 0$ (c.f. Fig. 5,*right*), i.e. when x is of degree 2.

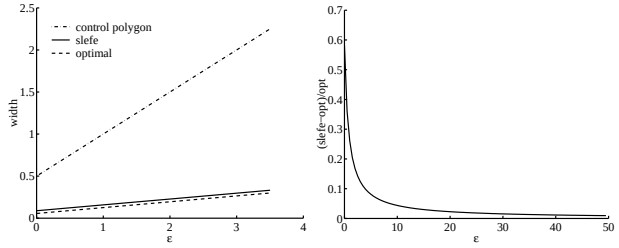


Fig. 6. Cubic x with inflection: (*left*) Value of $w_{\mathrm{slefe}}(x;U)$, $w^{\sim}_{\mathrm{opt}}(x;U)$ and the width based on the convex hull of the control polygon. (*right*) The ratio $\frac{w_{\mathrm{slefe}}-w^{\sim}_{\mathrm{opt}}}{w^{\sim}_{\mathrm{opt}}}(x;U)$

If x has an inflection point, we may assume that $\Delta_1 x := 1+\epsilon$, and $\Delta_2 x := -1$ and get

$$w_{\mathrm{slefe}}(x,\infty) - w^{\sim}_{\mathrm{opt}}(x,\infty) \approx .032775216.$$

The worst ratio $\frac{w_{\mathrm{slefe}}-w^{\sim}_{\mathrm{opt}}}{w^{\sim}_{\mathrm{opt}}}(x;U)$ occurs when x is of the type depicted in Figure 4: if $\Delta_1 x = -\Delta_2 x = 1$ then $w_{\mathrm{opt}}(x,U) = .05593616039$ and $w_{\mathrm{slefe}}(x,U) = .08857673214$. Although the ratio is almost 3:5, the slefe is considerably tighter than the convex hull of the control polygon (c.f. Figure 6, *left*).

2.4 Tensoring slefes

We can bootstrap univariate slefes by tensoring. A tensor-product polynomial $x(s,t)$ of degree d_1, d_2 is in Bézier form if

$$x(s,t) = \sum_{i=0}^{d_1}\sum_{j=0}^{d_2} x_{ij}\mathbf{b}_j^{d_2}(t)\mathbf{b}_i^{d_1}(s).$$

For example, a bi-cubic has 16 coefficients x_{ij}. We enclose $x(s,t)$ by a linear combination of $n \cdot m$-piecewise bilinear hat functions $\mathbf{h}_j^m(t)\mathbf{h}_i^n(s) \in \mathcal{H}^n \times \mathcal{H}^m$, for $(s,t) \in U := [0..1]^2$. Let $x_i(s)$ be the univariate Bézier polynomial with coefficients $x_{i0}, x_{i1}, \ldots, x_{id_2}$. We compute

$$\begin{aligned}\overline{x_i}(t) &:= \sum_{j=0}^{m} \mathtt{u}_{ij}^t \mathbf{h}_j^m(t) := b_{i0}(1-t) + b_{id_2}t \\ &+ \sum_{j=1}^{d_2-1} \underline{\mathbf{a}_j^{d_2}}_m \ \min\{\Delta_j x_i, 0\} + \overline{\mathbf{a}_j^{d_2}}^m \ \max\{\Delta_j x_i, 0\},\end{aligned}$$

and, with $\mathbf{u}_j(s) := \sum_{i=0}^{d_1} \mathtt{u}_{ij}^t \mathbf{b}_i^{d_1}(s)$,

$$\sum_{i=0}^{n} \mathtt{u}_{ij}\mathbf{h}_i^n(s) := \mathtt{u}_{0j}(1-s) + \mathtt{u}_{d_1 j}s + \sum_{i=1}^{d_1-1} \underline{\mathbf{a}_i^{d_1}}_n \ \min\{\Delta_i \mathbf{u}_j, 0\} + \overline{\mathbf{a}_i^{d_1}}^n \ \max\{\Delta_i \mathbf{u}_j, 0\},$$

Similarly, bounding $x(s,t)$ from below, we get

$$\begin{aligned}\underline{x}(s,t) &:= \sum_{j=0}^{m}\sum_{i=0}^{n} \mathtt{l}_{ij}\mathbf{h}_i^n(s)\mathbf{h}_j^m(t) \\ &\le \sum_{j=0}^{m}\big(\sum_{i=0}^{d_1} \mathtt{l}_{ij}^t \mathbf{b}_i^{d_1}(s)\big)\mathbf{h}_j^m(t) = \sum_{i=0}^{d_1}\sum_{j=0}^{m} \mathtt{l}_{ij}^t \mathbf{h}_j^m(t)\mathbf{b}_i^{d_1}(s) \\ \le x(s,t) &= \sum_{i=0}^{d_1}\sum_{j=0}^{d_2} x_{ij}\mathbf{b}_j^{d_2}(t)\mathbf{b}_i^{d_1}(s) \\ &\le \sum_{i=0}^{d_1}\sum_{j=0}^{m} \mathtt{u}_{ij}^t \mathbf{h}_j^m(t)\mathbf{b}_i^{d_1}(s) = \sum_{j=0}^{m}\big(\sum_{i=0}^{d_1} \mathtt{u}_{ij}^t \mathbf{b}_i^{d_1}(s)\big)\mathbf{h}_j^m(t) \\ \le \overline{x}(s,t) &:= \sum_{j=0}^{m}\sum_{i=0}^{n} \mathtt{u}_{ij}\mathbf{h}_i^n(s)\mathbf{h}_j^m(t).\end{aligned}$$

It is not difficult, although the generation of optimal approximation tables for $\underline{\mathbf{a}}$ and $\overline{\mathbf{a}}$ requires care, to extend the slefe construction to box-splines and to rational splines.

3 Mid-structures

The mid-structure

$$\overline{\underline{\mathbf{x}}} := (\overline{\mathbf{x}} + \underline{\mathbf{x}})/2$$

is well-defined for a vector-valued curve or surface $\mathbf{x}$. By making the mid-structure along a boundary depend only on the boundary, e.g. a space curve for a patch in $\mathbb{R}^3$ and the endpoint for a curve, mid-structures join continuously if their patches join continuously. Mid-structures are good L^∞ approximands and may be used, for example, to render more accurately than based on sampling (Figure 7). Certain mid-structures are invertible, e.g. in one variable, if $m = d$, then we can obtain $\mathbf{x}$ from $\underline{\overline{\mathbf{x}}}$.

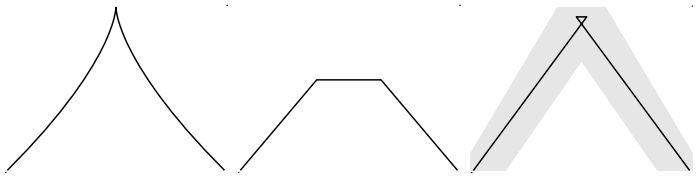


Fig. 7. A cubic B´ezier segment (*left*) finely evaluated, (*middle*) approximated by 4 sample values, (*right*) approximated by a 3-piece mid-path.

3.1 Univariate and bivariate mid-structures

We define the *mid-path*, $\underline{\overline{x}}$, of x as the m-piece linear function in $\mathcal{H}^n$ with values

$$\underline{\overline{\mathbf{x}}}(\frac{i}{m}) := \begin{cases} \frac{1}{2}(\overline{x}^m + \underline{x}_m)(\frac{i}{m}) & \text{if } 0 < i < n, \\ x_i & \text{if } i = 0 \text{ or } i = m. \end{cases}$$

The choice for $i = 0$ and $i = m$ guarantees that mid-paths of continuously joined Bézier pieces match up at their endpoints. The distance between the polynomial x and $\underline{\overline{x}}$ on the interval $[\frac{i}{m}..\frac{i+1}{m}]$ is bounded by the linear average of the distances at the endpoints; and these distances are evidently bounded by

$$|\mathbf{x} - \underline{\overline{\mathbf{x}}}|(\frac{i}{m}) \le \frac{\epsilon_i}{2}(\overline{x}^m - \underline{x}_m)(\frac{i}{m})$$

where $\epsilon_i = 2$ for $i = 0$ or $i = m$ and $\epsilon_i = 1$ otherwise. We can bound the derivative x' of x in the same manner and with the same number of $m = 3$ linear pieces:

$$(x')^+ := L(x') + \sum_{i=1}^{d-1} \underbrace{\mathbf{a}_i^{d-1}}_{m} \min\{\Delta_i x', 0\} + \overline{\mathbf{a}_i^{d-1}}^m \max\{\Delta_i x', 0\}.$$

For example, if x' is of degree $d = 2$, there is only one function $\mathbf{a}_1^2$ to bound and all we need are the numbers

$$\begin{bmatrix} \overline{\mathbf{a}_1^2}^3 \\ \underbrace{\mathbf{a}_1^2}_3 \end{bmatrix} \simeq \begin{bmatrix} 0.0 & -0.2\overline{2} & -0.2\overline{2} & 0.0 \\ -0.02777780 & -0.25 & -0.25 & -0.02777780 \end{bmatrix}$$

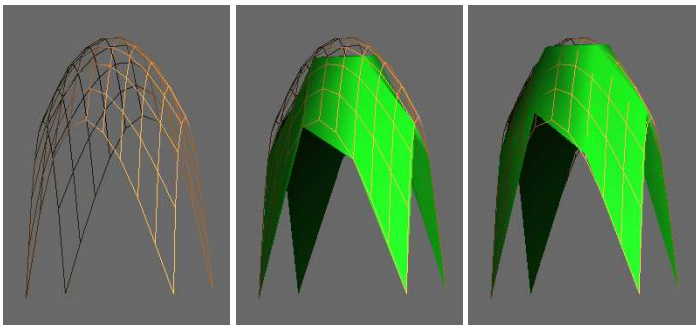


Fig. 8. A bi-quadratic B´ezier patch (*left*) finely evaluated, (*middle*) approximated by sampling, (*right*) approximated by a mid-patch.

to assemble $\overline{\underline{x'}}$. This allows associating a midnormal with every breakpoint of the midpath.

Analogous to midpaths in one variable, the *mid-patch* $\overline{\underline{\mathbf{x}}}$ of $\mathbf{x}$ as the $n \cdot m$-piece bilinear function in $\mathcal{H}^n \times \mathcal{H}^m$. We associate the average of a bivariate enclosure with the interior, the average of a univariate enclosure with the boundaries, and a constant with the vertices of U, so that adjacent midpatches match up continuously along boundaries. Let $x_j(t)$ be the polynomial with coeffi cients x_{0j} and $x_i(s)$ the polynomial with coeffi cients x_{i0} and $U = [0..1] \times [0..1]$. Then

$$\overline{\underline{x}}(\frac{i}{n},\frac{j}{m}) := \begin{cases} \frac{1}{2}(\overline{x}+\underline{x})(\frac{i}{n},\frac{j}{m}) & \text{if } i \notin \{0,n\} \text{ and } j \notin \{0,m\}, \\ \frac{1}{2}(\overline{x_i}+\underline{x_i})(\frac{j}{m}) & \text{if } i \in \{0,n\} \text{ and } j \notin \{0,m\}, \\ \frac{1}{2}(\overline{x_j}+\underline{x_j})(\frac{i}{n}) & \text{if } i \notin \{0,n\} \text{ and } j \in \{0,m\}, \\ x_{ij} & \text{if } i \in \{0,n\} \text{ and } j \in \{0,m\}. \end{cases}$$

4 Extension to Curves and surfaces

Slefes can be leveraged to generate Subdividable Linear Effi cient Variety Enclosures, short sleves, i.e. enclosures of *varieties* such as curves and surfaces in parametric or implicit representation [22].

4.1 Planar curve enclosures

Since both the x and the y component of a planar curve $\mathbf{x}$ provide an upper and a lower bound, we obtain four segments

$$\left[\begin{smallmatrix}\overline{x}\\ \overline{y}\end{smallmatrix}\right], \left[\begin{smallmatrix}\overline{x}\\ \underline{y}\end{smallmatrix}\right], \left[\begin{smallmatrix}\underline{x}\\ \overline{y}\end{smallmatrix}\right], \left[\begin{smallmatrix}\underline{x}\\ \underline{y}\end{smallmatrix}\right]$$

for each interval between breakpoints (see Figure 9, *left*). A certain ‘union’ of these bounds appears to enclose the curve. A simple way to give some structure to this ‘soup’

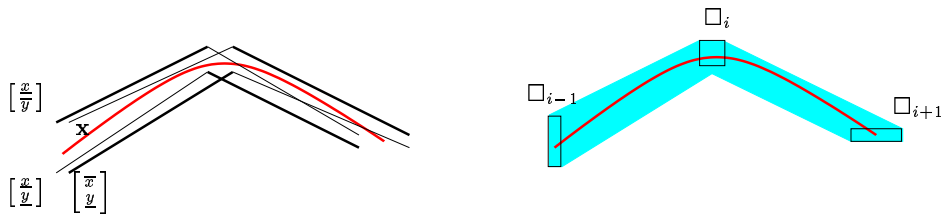


Fig. 9. (*left:*) The curve $\mathbf{x}$ is bounded by a 'union' of four component enclosures. The extreme, outermost components that stay to one side of the curve, are emphasized as fat line segments. Note the gap and the intersection between consecutive extreme segments. (*right:*) The bounding region is equivalently generated as the piecewise linear combination of point enclosures (axis-aligned rectangles) $\square_i$.

of line segments, is to observe that, due to linearity, each piece of the enclosure is a convex combination of consecutive point enclosures $\square_i$, $\square_{i+1}$, where $\square_j$ has the four vertices

$$\square_j \sim \left[\tfrac{\overline{x}}{\overline{y}}\right](\tfrac{j}{n}),\ \left[\tfrac{\overline{x}}{\underline{y}}\right](\tfrac{j}{n}),\ \left[\tfrac{\underline{x}}{\overline{y}}\right](\tfrac{j}{n}),\ \left[\tfrac{\underline{x}}{\underline{y}}\right](\tfrac{j}{n}).$$

That is, a *point enclosure* $\square_j$ is an axis-parallel rectangle or box (Figure 9 right). Differently put, the function enclosures directly yield a *piecewise linear interval enclosure*. Here, linearity is crucial, since the outer curves of general interval Bézier curves (see Farouki and Sederberg [27]) are non-trivial to represent and to work with: only the case where all $\square_j$ are of equal size has to date a short representation [29], p 50. Even in the linear case, deciding which combinations of linear function bounds are outer bounds for the curve, is not immediate. While the case where all $\square_j$ are of equal size is again straightforward, the general characterization requires several (simple) tests since it depends on the relative size and distance of $\square_{i-1}$ and $\square_i$.

Moreover, even linear interval enclosures have two shortcomings: multiplicity, and intersections or gaps. By keeping information on all four components, interference checking between two interval objects would require 16 intersection tests. Moreover, the piecewise linear outer bounds have more pieces or need to be trimmed due to the intersections and gaps between adjacent pieces (fat lines in Figure 9, *left*).

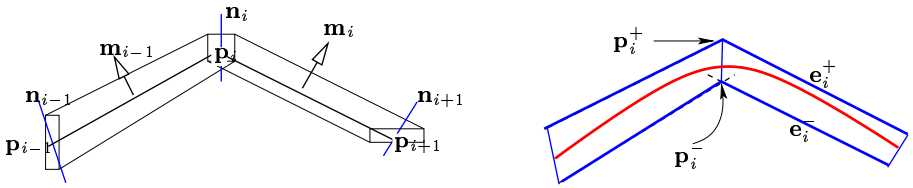


Fig. 10. *left:* Anchor points $\mathbf{p}_j$, normals $\mathbf{n}_j$ and directions $\mathbf{m}_j$ for identifying extreme segments of the enclosure. *right:* Antipodal points $\mathbf{p}_i$ and enclosure pieces $\mathbf{e}_i$.

Gaps and Intersections To address the problem of gaps and intersections, we associate, with each local parameter i/n, a point $\mathbf{p}_i$ that lies in *all* point enclosures associated with that location (there may be two point enclosures of differing size if $i \in \{0, n\}$, say the end of one curve segment and the start of another). We call the point, *anchor point*, because we have in mind to attach a line segment to it with direction $\mathbf{n}_i$, roughly normal to the curve (c.f. Figure 10 left). The two endpoints $\mathbf{p}_i^+$ and $\mathbf{p}_i^-$ will serve as the vertices of the two sheets $\mathbf{e}_i^+$ and $\mathbf{e}_i^-$ of the curve enclosure (c.f. Figure 10 right). If the Bézier pieces join with tangent continuity,

$$\mathbf{p}_i := \left[\begin{array}{c}\overline{x}\\ \underline{y}\end{array}\right](\frac{i}{n}), \text{ and } \mathbf{n}_i := \left[\begin{array}{c}\overline{y'}\\ -\underline{x'}\end{array}\right](\frac{i}{n}) \qquad (3)$$

fit the bill and we can process each piece independent of its neighbor. If the curves meet just with continuity of position then (the normalized) $\mathbf{n}_n$ of the first and (the normalized) $\mathbf{n}_0$ of the second segment need to be averaged.

Multiplicity To address the problem of multiplicity, we observe that, due to linearity, there is always a pair of linear function enclosures, $\left[\begin{array}{c}\overline{x}\\ \underline{y}\end{array}\right]$ and $\left[\begin{array}{c}\underline{x}\\ \overline{y}\end{array}\right]$ for the left segment in Figure 9, whose linear extensions or trims enclose the other two enclosures over the region of interest. The computationally efficient selection of this *extreme pair* of line segments from the four possible choices, as well as the full algorithm is presented in [23] (see also [22]). Given a (per segment or global) tolerance, the algorithm refines (subdivides) the representation locally until a sleve is obtained whose width is below the presecribed tolerance.

4.2 Interval patch enclosures

For x, y, z we each have an upper and a lower bound yielding eight candidates for enclosures (Figure 12) for u, v in the domain square $[i..i+1]/m_1 \times [j..j+1]/m_2$:

$$\left[\begin{array}{c}\overline{x}\\ \overline{y}\\ \overline{z}\end{array}\right], \left[\begin{array}{c}\overline{x}\\ \overline{y}\\ \underline{z}\end{array}\right], \left[\begin{array}{c}\overline{x}\\ \underline{y}\\ \overline{z}\end{array}\right], \left[\begin{array}{c}\overline{x}\\ \underline{y}\\ \underline{z}\end{array}\right], \left[\begin{array}{c}\underline{x}\\ \overline{y}\\ \overline{z}\end{array}\right], \left[\begin{array}{c}\underline{x}\\ \overline{y}\\ \underline{z}\end{array}\right], \left[\begin{array}{c}\underline{x}\\ \underline{y}\\ \overline{z}\end{array}\right], \left[\begin{array}{c}\underline{x}\\ \underline{y}\\ \underline{z}\end{array}\right].$$

All combinations with positive weights summing to 1 of the eight enclosures form a shell that is a 3D enclosure of the surface piece (see the surfaces with parameter grid in Figure 12). The union of the shells of all patches form a sleve.

Since the pieces are bilinear, we can also view the shell as a bilinear combination of the four point enclosures $\square_{i+\alpha,j+\beta}, \alpha, \beta \in \{0, 1\}$ of the corner points $[x(i+\alpha, j+\beta), y(i+\alpha, j+\beta), z(i+\alpha, j+\beta)]^T$. A *point enclosure* $\square_{i,j}$ is now an axis-parallel box whose vertices are the eight combinations of the corner points of the component slefes (the boxes displayed in Figures 4.2, 12 and 12) and slefes directly yield a *bilinear interval Bézier enclosure*.

The bilinear interval enclosures just defined have three shortcomings for efficient use: nonlinearity, multiplicity and gaps or intersections. The bilinearity of the facets implies that intersections between enclosures result in algebraic curves of degree 4 and force iterative techniques for intersections with rays as opposed to short explicit formulas for triangles. Slivers arise when computing the exact union of the shells which

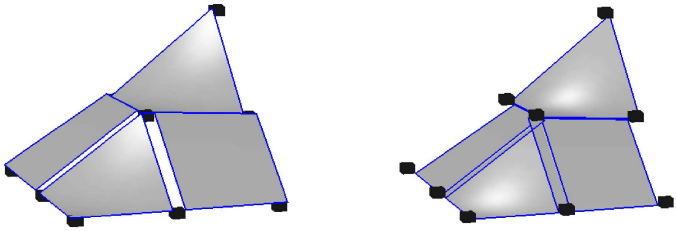


Fig. 11. Four pieces of an upper piecewise bilinear enclosure $\mathbf{x}^+$ and four pieces of a lower bilinear facets $\mathbf{x}^-$. The nine cubes represent point enclosures. Note the gaps and overlaps.

entails intersection of bilinear facets and trimming bilinear patches. Multiplicity, i.e. the choice from eight possible bilinear function enclosures implies up to 64 nonlinear intersection tests when intersecting two patch enclosures. The algorithm in [23, 22] remedies gaps, intersections and slivers just as in the case of one variable using points $\mathbf{p}_{ij}$ and directions $\mathbf{n}_{ij}$ (Figure 12) to construct serve as the vertices points $\mathbf{p}_{ij}^+$ and $\mathbf{p}_{ij}^-$ that support two triangle pairs $\mathbf{e}_{ij}^{+,1}, \mathbf{e}_{ij}^{+,2}$ and $\mathbf{e}_{ij}^{-,1}, \mathbf{e}_{ij}^{-,2}$. The surface enclosure is thus a tent-like construction with support beams in the direction $\mathbf{n}_{ij}$ as shown in Figure 12. Multiplicity is addressed by picking an approximate normal $\mathbf{m}_{ij}$ analoguous to the univariate case and finding an extreme pair of bilinear patches.

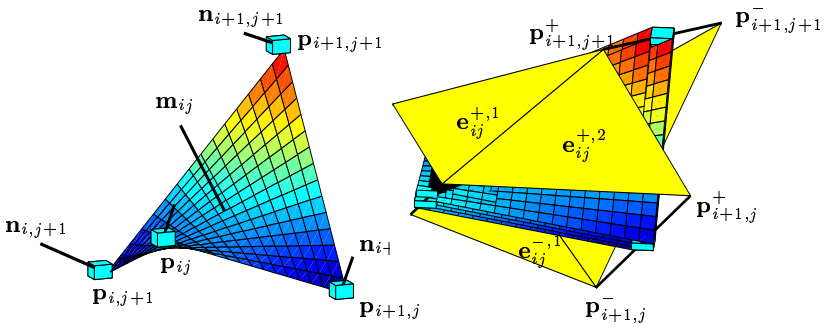


Fig. 12. (*left*) A single bilinear facet of the midpatch with direction $\mathbf{m}_{ij}$, anchor points $\mathbf{p}_{ij}$ and normal direction $\mathbf{n}_{ij}$. (*right*) Antipodal pairs $\mathbf{p}_{ij}^+$ and $\mathbf{p}_{ij}^-$ as interpolation points of the two sheets $\mathbf{e}_{ij}^{+,1}, \mathbf{e}_{ij}^{+,2}$ and $\mathbf{e}_{ij}^{-,1}, \mathbf{e}_{ij}^{-,2}$ of the surface enclosure of eight bilinear facets (with parameter grid) whose extreme pair is $\mathbf{h}_{ij}^+$ and $\mathbf{h}_{ij}^-$. (The black spot is due to Matlab's depth sorting algorithm in the presence of many overlapping surfaces).

Finally, bilinear slefes are replaced by two pairs of triangles per original facet, a subtle operation, since extrapolation of a triangle interpolating three vertices of a

bilinear facet may intersect the extrapolated bilinear facet and it cease to be a one-sided approximation.

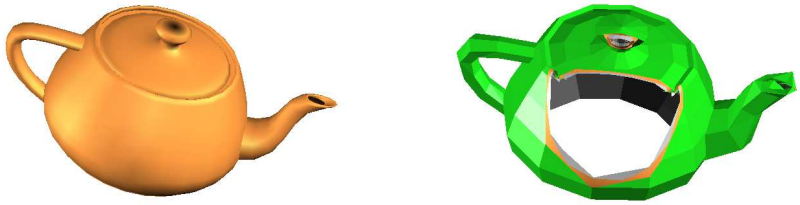


Fig. 13. Teapot inside a sleve.

5 Inverse problems

The simplicity and linearity of the slefe construction allows solving problems like the

CHANNEL Problem: Given a channel or tube, construct a smooth spline that stays within that channel.

(see Figure 14). Solutions can be used to thread pipes past a set of obstacles or determine robot motion paths. The CHANNEL problem is a two-sided version of the

SUPPORT Problem (see Figure 1): given an input polygon, find a spline *(black)* that stays above but close to the polygon.

5.1 The CHANNEL problem for space curves

The emphasis in this problem is neither on the optimality nor the uniqueness of the solution, although we will see that we can easily augment the feasibility problem with a linear (or quadratic) optimization function. Our approach to solving the CHANNEL problem is to construct a 'sleeve' around the candidate space curve; then it is sufficient

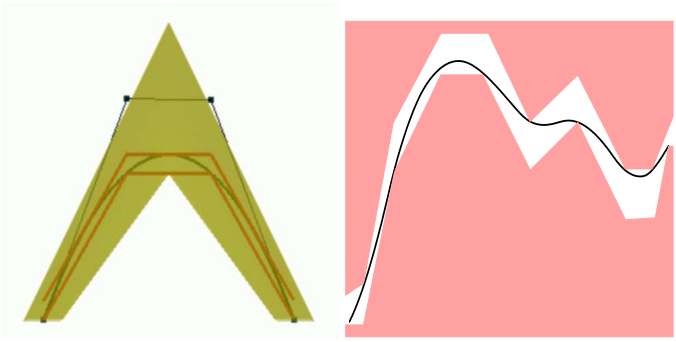


Fig. 14. Simple 2D examples. (*left*) the curve segment is fit into the *shaded* channel by fitting its sleve (piecewise linear) into the channel. Characteristically, the control polygon (*square control points*) does not stay within the channel. (*right*) A more complex, but still function-based channel scenario.

to constrain this sleeve – rather than the original nonlinear curve – to stay within the channel. This approximation reduces the original, complexity-wise intractable, *continuous feasibility problem* to a simple *linear feasibility problem* that is solvable by any Simplex or Linear Program solver!

Three properties are crucial to make this approach work.

(i) The enclosure must depend linearly on the coeffi cients of the spline representation.
(ii) The enclosure should be near-optimal in the max-norm, i.e. as narrow as possible.
(iii) The enclosure should be refi nable to adaptively adjust to where the channel is particularly narrow or tricky to navigate.

Requirement (i) rules out oriented bounding box and convex hull-based approaches (see Section 1.1, page 3) since the coeffi cients of the curve will be variable as well as [14]. Requirement (ii) rules out the use of looser bounding constructs such as bounding spheres and axis-aligned bounding boxes (c.f. Figure 16). The best match of linearity, tightness and refi nability for the CHANNEL problem are therefore slefe-based constructions. For *functions* in one variable, the CHANNEL problem was fi rst formulated in [15]. A closely related set of problems appears in graph layout [7] where, however, the emphasis is on a large number of piecewise linear curves with few pieces. By tightly linking discrete and non-linear techniques, our new approach may be viewed as bridging a gap between established techniques of *computational geometry and geometric design*.

Channel definition We defi ne a channel as a sequence of *cross-sections*. Each cross-section has n_{c_sides} edges. For example, in Figure 15, each cross-section is quadrilateral with the four vertices not necessarily co-planar. Adjacent cross-section pairs constitute a channel segment, or *c-segment*. Two points of one cross-section and the corresponding pair of points from its neighboring cross-section form a face of the channel. A face is not necessarily planar and is split into two triangles. The normals $\mathbf{n}_j^c$ are consistently

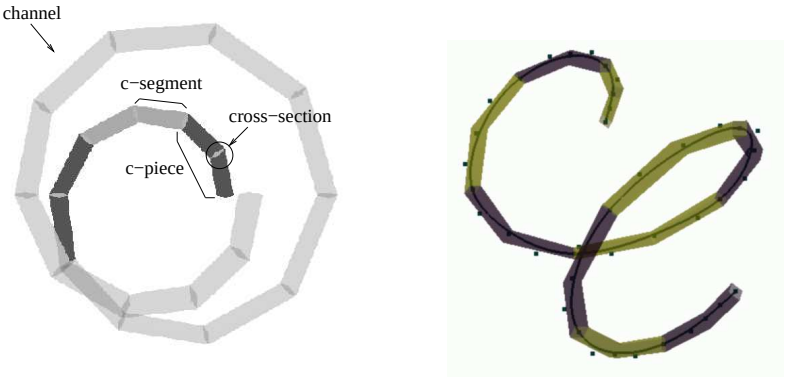


Fig. 15. Composition of a CHANNEL. (*right:*) A solution curve (control points indicated as black squares) fit into the given, transparently rendered channel whose consecutive c-pieces alternate between yellow and grey (transparent).

oriented outwards. We combine n_{c_segs} c-segments to form a *c-piece* (this is analogous to line segments connected to form a control polygon). In Figure 15, each c-piece consists of two adjacent c-segments. The union of n_{pcs} c-pieces forms a channel. In the following, one curve piece in Bézier form is fitted through each c-piece, and C^1 continuity constraints are enforced between adjacent polynomial pieces. The sleves are calculated for each $c \in (1, \ldots, n_{pcs})$, $i = 1 \ldots, d-1$ and $v \in \{x, y, z\}$ as

- $\triangle^c_{i,v} := x^c_{i-1,v} - 2x^c_{i,v} + x^c_{i+1,v}$
- ${\triangle^c_{i,v}}^+ := \max(\triangle^c_{i,v}, 0)$
- ${\triangle^c_{i,v}}^- := \min(\triangle^c_{i,v}, 0)$
- $\overline{e}^c_v := L(\mathbf{x^c}) + \overline{\mathbf{a}^d} \cdot {\triangle^c_v}^- + \underline{\mathbf{a}^d} \cdot {\triangle^c_v}^+$
- $\underline{e}^c_v := L(\mathbf{x^c}) + \underline{\mathbf{a}^d} \cdot {\triangle^c_v}^+ + \overline{\mathbf{a}^d} \cdot {\triangle^c_v}^-$

The constraint system We can now formulate the CHANNEL problem as a feasibility problem for fitting a degree d polynomial piece through each of n_{pcs} c-pieces. For each $c \in (1, \ldots, n_{pcs})$, $i = 1 \ldots, d-1$ and $v \in \{x, y, z\}$, we have following constraints:

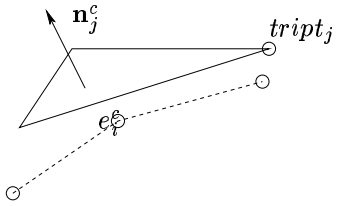
1. ${\triangle^c_{i,v}}^+ \geq \triangle^c_{i,v}$ and ${\triangle^c_{i,v}}^+ \geq 0$
2. ${\triangle^c_{i,v}}^- \leq \triangle^c_{i,v}$ and ${\triangle^c_{i,v}}^- \leq 0$
3. ${\triangle^c_{i,v}}^+ + {\triangle^c_{i,v}}^- = \triangle^c_{i,v}$
4. Set $x^c_{d,v} = x^{c+1}_{0,v}$ equal (e.g. to the center of a cross-section) to ensure C^0 continuity.
5. Set $x^c_{d,v} = \frac{1}{2}\left(x^c_{d-1,v} + x^{c+1}_{0,v}\right)$ to ensure C^1 continuity.

The remaining constraints combine the x, y, and z components to ensure that the curve stays within the channel. Here the linearity of the sleve construction pays off.

1. At each sleve breakpoint, $e_i^c := (e_{i,x}^c, e_{i,y}^c, e_{i,v}^c)$, where $e_{i,v}^c \in \{\underline{e}_{i,v}^c, \overline{e}_{i,v}^c\}$, needs to be within the channel. Hence, for every such e_i^c and each point $tript_j^c$ in the corresponding c-segment, we enforce

$$\mathbf{n}_j^c \cdot (e_i^c - tript_j^c) \leq 0,$$

where $\mathbf{n}_j^c$ is the outward-pointing normal:



2. At each concave (inward-pointing) channel breakpoint $tript_j^c$, and the normals of the triangles incident to $tript_j^c$, constrain any point on the corresponding enclosure segment, $e'^c := ue_{i-1}^c + ve_i^c$, to satisfy

$$\mathbf{n}_j^c \cdot (e'^c - tript_j^c) \leq 0$$

with $u + v = 1$ and $u > 0$ and $v > 0$.

The objective function A typical linear program has the form

$$\min_b f(b), \qquad \text{subject to linear equality and inequality constraints.}$$

The previous section listed the constraints and we may add an objective function f. (If there is no objective function, then the problem is called a linear feasibility problem.) A possible choice for f is to minimize the sum of the absolute second-differences. Since $\triangle_{i,v}^{c\,+} \geq 0$, $\triangle_{i,v}^{c\,-} \leq 0$, and $\triangle_{i,v}^{c\,+} + \triangle_{i,v}^{c\,-} = \triangle_{i,v}^c$, it follows that $|\triangle_{i,v}^c| = \triangle_{i,v}^{c\,+} - \triangle_{i,v}^{c\,-}$. Hence, to minimize kinks, the objective function is:

$$\sum_{c=1}^{n_{pcs}} \left(\sum_{i=1}^{d-1} (\triangle_{i,x}^{c\,+} - \triangle_{i,x}^{c\,-} + \triangle_{i,y}^{c\,+} - \triangle_{i,y}^{c\,-} + \triangle_{i,z}^{c\,+} - \triangle_{i,z}^{c\,-}) \right)$$

Examples Using the open-source *PCx* [6], the channel in Figure 15 to be traversed by a twenty segment, degree 4 spline $n_{c\_segs} = 2$ and one polynomial piece for each pair of c-segments, results in 3560 constraints (567 equations, and 720 variables, generated by a scripting language) and is solved in 5.45 seconds on a generic PC. The example in Figure 16 highlights the crucial role played by the near-optimal width of the bounding construct.

5.2 The SUPPORT problem for surfaces

The support problem is a simpler optimization problem, using only one-sided constraints. Remarkably, if we were to attempt to solve SUPPORT over all nonsmooth functions with arbitrary break points, rather than for splines with fixed break points, the problem would be NP hard [1].

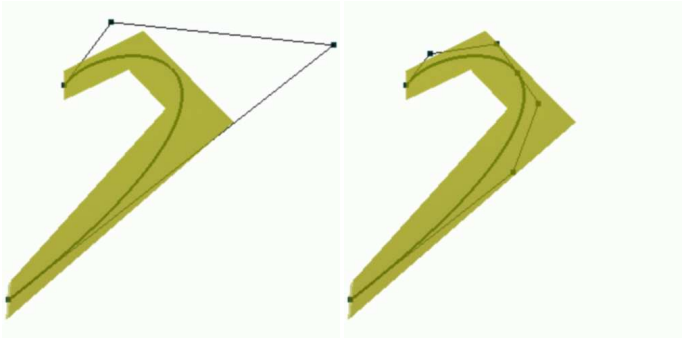


Fig. 16. A sharp 2D channel. (*left*) The control polygon of the solution lies well outside the channel. (*right*) The control polygon of the solution after subdivision still violates the channel boundaries illustrating the need for the tight slefe bounds.

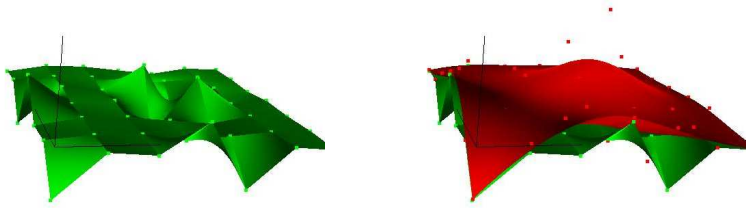


Fig. 17. Bilinear barrier surface (*left*) and a solution to the SUPPORT problem (*right*).

6 Open Problems, Future Work

Much of the usefulness of slefes and sleves relies on the narrowness of slefes. To date, this near-optimality has only been *proven and quantified* for cubic functions in one variable (Section 2.3). Further investigation of polynomials of degree d leads, after removal of symmetries, to 2^{d-2} distinct cases of minimization problems in $d-2$ variables. An essential difficulty is to explicitly determine the narrowest enclosure for a class of functions – if that were easy, we would not need sleves. Numerical analysis for small d leads to the conjecture that the ratio (optimal max-norm enclosure width : the width of the slefe construction) is minimal when all second differences are equal. If true, this would settle the question for univariate polynomials.

The proper use of the observed near-optimality of sleves must be established by looking in detail at applications. Here the balance between the cost of generating the enclosure and the cost of using the enclosure has to be evaluated. For example, while sleves have clear advantages in applications that value tightness and linearity of the bounds, such as the inverse problems sketched in Section 5 and certain applications in marine cartography, the relative merits of computing sleves for intersection testing and root finding *vis a vis* established techniques, say as summarized in [19], has to be characterized for specific classes of problems.

Acknowledgments

This survey has drawn from joint work and work in progress with Xiaobin Wu, David Lutterkort and Ashish Myles. Thanks also to the referees for asking for more details on [9] and pointing out further references on interval spline computations.

References

1. Pankaj K. Agarwal and Subhash Suri. Surface approximation and geometric partitions. *SIAM Journal on Computing*, 27(4):1016–1035, August 1998.
2. Julien Basch. *Kinetic Data Structures*. PhD thesis, Stanford University, 1999.
3. R. C. Buck. Applications of duality in approximation theory. In *Approximation of Functions (Proc. Sympos. General Motors Res. Lab., 1964)*, pages 27–42. Elsevier Publ. Co., Amsterdam, 1965.
4. Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick P. Brooks, Jr., and William Wright. Simplification envelopes. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 119–128. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.
5. A. Crosnier and J. R. Rossignac. Technical section — tribox bounds for three-dimensional objects. *Computers and Graphics*, 23(3):429–437, June 1999.
6. Joe Czyzyk, Sanjay Mehrotra, Michael Wagner, and Stephen Wright. http://www-fp.mcs.anl.gov/otc/Tools/PCx.
7. D. Dobkin, E.R. Gansner, E. Koutsofios, and S.C. North. A path router for graph drawing. In *Proceedings of the 14th annual symposium on Computational Geometry*, pages 415–416. ACM Press, New York, 1998. June 1-10 1998, Minneapolis, MN.

8. Gerald Farin. Tighter convex hulls for rational B´ezier curves. *Comput. Aided Geom. Design*, 10:123–125, 1993.
9. Daniel Filip, Robert Magedson, and Robert Markot. Surface algorithms using bounds on derivatives. *CAGD*, 3(4):295–311, 1986.
10. S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. *Computer Graphics*, 30(Annual Conference Series):171–180, 1996.
11. Timothy L. Kay and James T. Kajiya. Ray tracing complex scenes. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 269–278, August 1986.
12. James T. Klosowski, Joseph S. B. Mitchell, Henry Sowizral, and Karel Zikan. Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36, January 1998.
13. James Thomas Klosowski. *Effi cient collision detection for interactive 3d graphics and virtual environments*. PhD thesis, State Univ. of New York at Stony Brook, May 1998.
14. Leif P. Kobbelt, Katja Daubert, and Hans-Peter Seidel. Ray tracing of subdivision surfaces. In *Rendering Techniques '98 (Proceedings of the Eurographics Workshop)*, pages 69–80, New York, June 1998. Springer-Verlag.
15. D. Lutterkort and J. Peters. Smooth paths in a polygonal channel. In *Proceedings of the 15th annual symposium on Computational Geometry*, pages 316–321, 1999.
16. D. Lutterkort and J. Peters. Optimized refinable enclosures of multivariate polynomial pieces. *Comput. Aided Geom. Design*, 18(9):851–863, 2001.
17. David Lutterkort. *Envelopes for Nonlinear Geometry*. PhD thesis, Purdue University, May 2000.
18. D. Nairn, J. Peters, and D. Lutterkort. Sharp, quantitative bounds on the distance between a polynomial piece and its B´ezier control polygon. *Computer Aided Geometric Design*, 16(7):613–633, Aug 1999.
19. N. M. Patrikalakis and T. Maekawa. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer-Verlag, Berlin, 2002.
20. N.M. Patrikalakis and T. Maekawa. Intersection problems. In *Handbook of Computer Aided Geometric Design*. Springer, 2001.
21. J. Peters and X. Wu. On the optimality of piecewise linear max-norm enclosures based on slefes. In *Proceedings of the 2002 St Malo conference on Curves and Surfaces*, 2003.
22. Jörg Peters and Xiaobin Wu. Optimized refinable surface enclosures. Technical report, University of Florida, 2000.
23. Jörg Peters and Xiaobin Wu. Sanwiching spline surfaces. *Computer-Aided Geometric Design*, 200x.
24. Allan M. Pinkus. *On L^1-approximation*. Cambridge University Press, Cambridge, 1989.
25. U. Reif. Best bounds on the approximation of polynomials and splines by their control structure. *Comput. Aided Geom. Design*, 17(6):579–589, 2000.
26. P.V. Sander, Xianfeng Gu, S.J. Gortler, H. Hoppe, and J. Snyder. Silhouette clipping. *Computer Graphics*, 34(Annual Conference Series):327–334, 2000.
27. T. W. Sederberg and R. T. Farouki. Approximation by interval bezier curves. *IEEE Computer Graphics and Applications*, 12(5):87–95, September 1992.
28. Thomas W. Sederberg, Scott C. White, and Alan K. Zundel. Fat arcs: A bounding region with cubic convergence. *Comput. Aided Geom. Design*, 6:205–218, 1989.
29. G. Shen and N.M. Patrikalakis. Numerical and geometric properties of interval B-splines. *International Journal of Shape Modeling*, 4(1,2):35–62, 1998.
30. E. Welzl. Smallest enclosing disks (balls and ellipsoids). In Hermann Maurer, editor, *Proceedings of New Results and New Trends in Computer Science*, volume 555 of *LNCS*, pages 359–370, Berlin, Germany, June 1991. Springer.