SMOOTH PATHS IN A POLYGONAL CHANNEL

DAVID LUTTERKORT AND JÖRG PETERS

ABSTRACT. We show how to efficiently smooth a polygon with an approximating spline that stays to one side of the polygon. We also show how to find a smooth spline path between two polygons that form a channel. Problems of this type arise in many physical motion planning tasks where not only forbidden regions have to be avoided but also a smooth traversal of the motion path is required. Both algorithms are based on a new tight and efficiently computable bound on the distance of a spline from its control polygon and employ only standard linear and quadratic programming techniques.

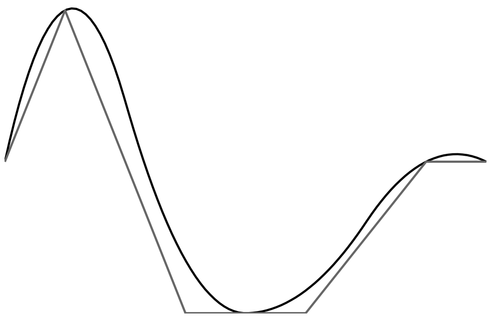

FIGURE 1. The SUPPORT problem: given an input polygon $\underline{c}$ *(grey)* find a spline *(black)* that stays above but close to $\underline{c}$.

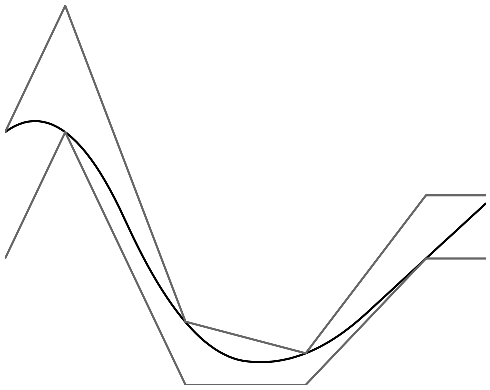

FIGURE 2. The CHANNEL problem: given two non–intersecting input polygons $\underline{c}$ and $\overline{c}$ *(grey)*, find a spline *(black)* that stays between $\underline{c}$ and $\overline{c}$.

1. MOTIVATION

Many industrial applications require the construction of smooth paths that avoid certain forbidden regions. A typical example is the path of a cutting tool which must stay away from the surrounding machine to avoid damage to tool and machine. Another is access planning for adding or modifying a part of a larger assembly. A third is graph layout [DGKN98].

The rich toolkit of computational geometry delivers permissible piecewise linear paths in these situations (see e.g. the references in Chapter 13 of [dBvKOS97]). However, many physical applications demand a smoother path, for example a path that is tangent continuous or even has a continuously varying curvature. The (offset) path of the cutting tool, for example, must be parametrically curvature–continuous to avoid jumps in the acceleration that would strain or damage the driver motors. Creating smooth curves that approximate given data is a standard task of geometric design. The challenge therefore lies in effectively combining the solutions provided by computational geometry and geometric design. Specifically, what is needed and developed in this paper, based on some new insights in spline theory, is a postprocessing scheme that *smoothes edges to one side* distributing change in direction and speed more evenly while preserving the essential quality of the series of straight edges generated by the computational geometry routines as upper hulls, edges of a binary space partition, parts of triangulations, road maps or visibility graphs.

Specifically, we formulate and give a practical algorithm for two basic problems, SUPPORT and CHANNEL. Both problems are stated in terms of spline *functions* rather than planar parametric curves. For physically motivated path problems, the functional notion of continuity is more appropriate than the notion of geometric continuity. A parametric

Date: November 10, 1999.

problem can be reduced to a functional problem by independently constructing functions for each coordinate or by recasting it in polar coordinates if the input polygon is star–shaped. The extension to spline surfaces is more complicated and is not discussed in this paper.

The terms 'close' and 'small number of pieces' in the following statement of the two basic problems are made precise in Section 3.

SUPPORT**:** Given a piecewise linear function $\underline{c}$ (the *input polygon*) with vertices $(\underline{x}_i, \underline{c}_i)$ and $\underline{x}_0 < \underline{x}_1 < \cdots < \underline{x}_m$, construct a spline function b of a given degree and smoothness that on the interval $[\underline{x}_0, \underline{x}_m]$ stays close to and entirely above or on the input polygon and consists of a small number of pieces.

CHANNEL**:** Given two non–intersecting input polygons $\underline{c} < \overline{c}$ with vertices $(\underline{x}_i, \underline{c}_i)$ and $(\overline{x}_i, \overline{c}_i)$, construct a spline function b that stays between $\underline{c}$ and $\overline{c}$ and consists of a small number of pieces.

The main difficulty in solving SUPPORT and CHANNEL is that they are *continuous problems*: we have constraints of the form $\underline{c}(t) \le b(t)$ respectively $\underline{c}(t) \le b(t) \le \overline{c}(t)$. We avoid solving these continuously constrained feasibility problems by taking advantage of a new, exactly quantifiable and efficiently computable bound on the distance between a spline and its control structure. That is, we use a tight piecewise linear *envelope* for the spline b to simplify the problem and reduce it to finitely many constraints.

The envelope consists of two polygons, one above the spline b and one below it (see Figure 3). To solve SUPPORT

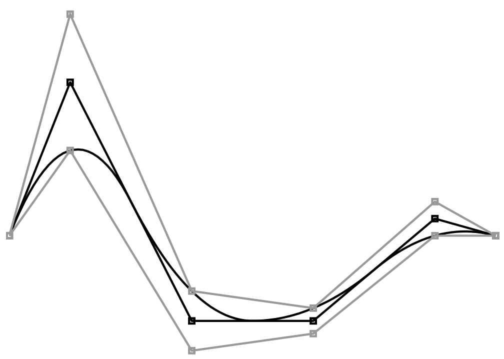

FIGURE 3. A quadratic B–spline and its control polygon *(black)* together with its envelope *(grey)*. The construction of the envelope is explained in Section 2

.

it suffices to find a spline b whose lower envelope lies above the input polygon. The construction of such an envelope which bounds curved, nonlinear geometry in terms of piecewise linear geometry, is presented in Section 2. Once we are able to compute tight envelopes for a spline, we can solve SUPPORT and CHANNEL by linear, respectively quadratic programming depending on what additional properties we would like to enforce. Sections 3.2 and 3.3 describe the algorithms for solving SUPPORT and CHANNEL, respectively.

1.1. **Prior work.** While the motivation for this paper arose from practical considerations, approximation theory has long considered the closely related problems of *one-sided approximation* and *two-sided approximation*. Based on Buck's seminal "Applications of Duality in Approximation Theory" [Buc65], DeVore [DeV68] established the close relation of one-sided approximation to quadrature formulas with nonnegative coefficients and gives a Remez-type algorithm for determining a (unique) solution. This, however, is based on the assumptions that the function to be one-sidedly approximated is differentiable, and that approximands form a strong Chebyshev space. Neither assumption holds for the problems of interest in this paper. Thirty more years have seen a rich body of literature concerning one-sided approximation. The state of the art has been collected in the monograph [Pin89] which also offers algorithms in its last Chapter 7. The reader may check however, that the proposed procedures do not fall into the realm of practical computing in that they rely on a discretization refinement without a bound on the size of the subproblems. Termination is not guaranteed other than that a subsequence must exist that converges. To quote the author [Pin89], page 181, 'Suffice to say that convergence is generally very slow, at least in theory'.

This paper does not solve the hard problems of *one-sided approximation* as formulated by the approximation theorists — we are satisfied with an efficiently computable and tight approximation with a small, quantifiable error, while the approximation theorist looks to establish optimality and uniqueness. We hope to make this difference clear by calling the problems SUPPORT and CHANNEL rather than one-sided approximation and two-sided approximation.

In 'a path router for graph drawing'[DGKN98], Dobkin, Gansner, Koutsofios and North derive interpolating piecewise cubic curves in Bézier form (Catmull-Rom splines) from the shortest path polyline to trace out a once differentiable path around obstacles. After construction, the path is checked for admissibility (non-intersection with the forbidden region) and recomputed with more curve pieces where it fails. In contrast our approach below uses *a priori* bounds and supports the full gamut of spline representations.

2. A TIGHT ENVELOPE FOR SPLINES

Recall that a spline b in B–spline form is given by

$$b(t) = \sum_{j=0}^{m} b_j N_j^d(t),$$

where the real numbers b_j are called *control points*. The B-spline basis functions N_j^d of degree d are defined by a *knot sequence* (t_k), a non–decreasing sequence of real numbers $t_0 \le t_1 \le \cdots \le t_{m+d+1}$ and well-known recursion formulas (see e.g. [dB78]). The control polygon $\ell(t)$ of the spline is the piecewise linear interpolant to the control points b_j at the *Greville abscissae*

$$t_j^* = \sum_{i=j+1}^{j+d} t_i/d,$$

i.e. $\ell(t_j^*) = b_j$.

At the heart of an envelope for splines is a bound on the distance between the spline b and its control polygon ℓ. The literature contains a number of such bounds. The best known are the min–max bound and the convex hull. The min–max bound states that $\min_j b_j \le b(t) \le \max_j b_j$. Min-max bounds are easily computed but generate unduly large envelopes, and a poor approximation of b. The convex hull yields a tighter bound, stating that the graph of the function lies in the (local) convex hull of the control points (t_j^*, b_j). While this is an improvement over the min–max bound, it still overestimates the extent of b considerably in many practical cases. Other bounds in the literature, for example [Dah86] or [PK94], have been derived in the course of proofs and were only intended to support qualitative arguments. For our purposes, these bounds are either too loose or can not be computed efficiently.

The main argument of this paper relies on a sharp bound recently derived in [LP9x, Corollary 3.2]. This new bound provides narrow envelopes in particular for quadratic, cubic and quartic splines which yield tangent continuity, curvature continuity and continuity of the derivative of the curvature. We restrict the discussion to end point interpolating knot sequences that are typically used in applications. Here the first and last knot have multiplicity $d+1$, viz. $t_0 = t_1 = \cdots = t_d$ and $t_{m+1} = t_{m+2} = \cdots = t_{m+d+1}$.

The *weighted second differences* $\Delta_2 b_j$ of the control points are defined as

$$\Delta_2 b_j = b'_{j+1} - b'_j \qquad \text{where} \quad b'_j = \frac{b_j - b_{j-1}}{t_j^* - t_{j-1}^*}.$$

Over the interval $[t_k^*, t_{k+1}^*]$ the contribution of the ith B–spline to the distance between a spline and its control polygon is captured by the *non-negative and convex* functions

$$\beta_{ki} = \begin{cases} \sum_{j=i}^{\overline{k}} (t_j^* - t_i^*) N_j^d & i > k \\ \sum_{j=\underline{k}}^{i} (t_i^* - t_j^*) N_j^d & i \le k. \end{cases} \tag{2.1}$$

The values $\beta_{ki}(t_k^*)$ will enter as weights into the constraints to be developed. The numbers $\underline{k}$ and $\overline{k}$ are the index of the first and last of at most $d+1$ B–spline basis function N_j^d whose support straddles t_k^*. With these definitions the distance of a spline b to its control polygon ℓ is [LP9x, Lemma 3.1]

$$b - \ell = \sum_{i=\underline{k}}^{\overline{k}} \Delta_2 b_i \, \beta_{ki}.$$

With $\Delta_i^- = \min\{0, \Delta_2 b_i\}$, $\Delta_i^+ = \max\{0, \Delta_2 b_i\}$ this implies upper and lower bounds

$$\ell + \sum \Delta_i^- \beta_{ki} \le b \le \ell + \sum \Delta_i^+ \beta_{ki}.$$

Since the β_{ki} are non-negative and convex we can replace the left and the right expressions on the interval $[t_k^*, t_{k+1}^*]$ by linear functions $\underline{e}$ and $\overline{e}$ that interpolate their values at the Greville abscissae

$$\underline{e} = \ell + \mathcal{L}\left(\sum \Delta_i^- \beta_{ki}(t_k^*), \sum \Delta_i^- \beta_{k+1,i}(t_{k+1}^*)\right),$$
$$\overline{e} = \ell + \mathcal{L}\left(\sum \Delta_i^+ \beta_{ki}(t_k^*), \sum \Delta_i^+ \beta_{k+1,i}(t_{k+1}^*)\right).$$

The corresponding envelope is [LP9x, Corollary 3.2]:

$$\underline{e} \le b \le \overline{e}.$$

3. SOLVING SUPPORT AND CHANNEL

With the piecewise linear envelope just derived SUPPORT and CHANNEL can be formulated as feasibility problems with linear constraints. Recall that $m+1$ is the number of spline coefficients. Both m and the knot sequence may be chosen freely by the user (see Algorithm 1 below for a default choice). The constraints for SUPPORT ensure that the lower envelope of the spline stays above the input polygon. The constraints for CHANNEL ensure that the envelope of the spline stays between the input polygons.

3.1. Constraints.

The lower constraints $\underline{C}$ on the $2m$ unknowns Δ_i^- and b_i are as follows.

(1) For $1 \le i \le m-1$ bound the variable Δ_i^- by

$$\Delta_i^- \le 0$$
$$\Delta_i^- \le \Delta_2 b_i.$$

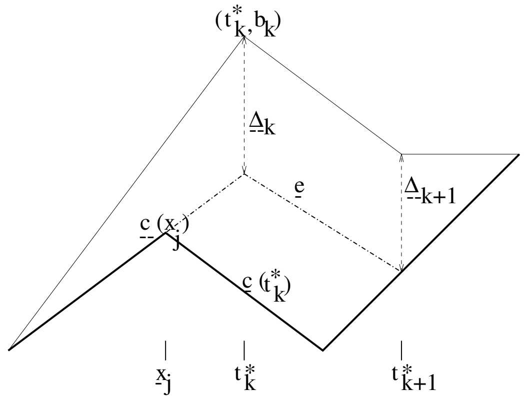

FIGURE 4. Variables of the constraint formulation. Input polygon *(thick)*, lower envelope *(dash-dotted)* and control polygon *(thin)*.

(2) For $0 \le k \le m$ ensure that all breakpoints of the lower envelope $\underline{e}$ of the spline stay above the lower polygon $\underline{c}$,

$$\underline{c}(t_k^*) \le \underline{e}(t_k^*) = b_k + \sum_{\underline{k}<l<\overline{k}} \Delta_l^- \beta_{kl}(t_k^*).$$

(3) For each concave corner $\underline{c}(\underline{x}_j)$ of the lower channel polygon ensure that the corner lies below the linear envelope

$$\underline{c}(\underline{x}_j) \le \mathcal{L}\left(\underline{e}(t_k^*), \underline{e}(t_{k+1}^*)\right)(\underline{x}_j).$$

The constraints $\overline{\mathcal{C}}$ are obtained from $\underline{\mathcal{C}}$ by replacing all inequality signs $\le$ by $\ge$, all underlines by overlines, superscripts $^-$ by $^+$ and concave corners by convex corners in (3).

3.2. **Solving** SUPPORT**.** Given the at most $3m-1+\underline{m}$ constraints $\underline{\mathcal{C}}$, the optimization goal is to keep the control polygon of b as close as possible to $\underline{c}$. The resulting cost function is

$$\sum_{i=0}^{m} (b_i - \underline{c}_i) \to \min.$$

In this formulation, SUPPORT can be solved by a standard solver for linear programs.

Figures 5 and 6 show two examples of SUPPORT and their solutions.

3.3. **Solving** CHANNEL**.** There are at most $6m-2+\underline{m}+\overline{m}$ constraints $\underline{\mathcal{C}}$ and $\overline{\mathcal{C}}$, and we can choose from a variety of optimization goals. To keep b centered between $\underline{c}$ and $\overline{c}$, i.e. $\underline{c}_i - b_i = b_i - \overline{c}_i$ for each i, we use the cost function

$$\sum_{i=0}^{m} \underline{c}_i - 2b_i + \overline{c}_i \to \min.$$

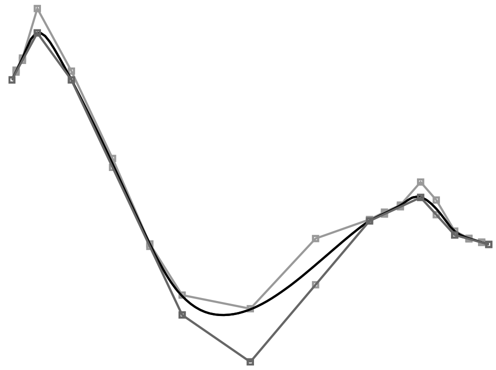

FIGURE 5. A solution to the SUPPORT problem. The constructed quadratic spline *(black)* lies on the input polygon *(dark grey)*. The lower part of the envelope *(light grey)* is not visible since it matches the input polygon exactly.

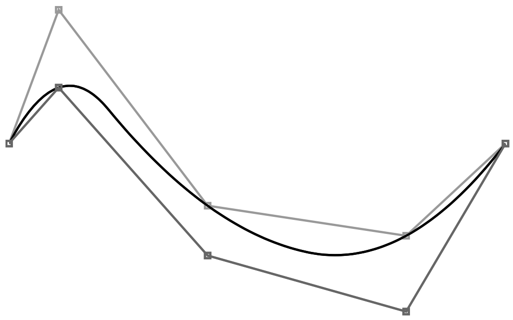

FIGURE 6. A solution to the SUPPORT problem. The constructed cubic spline *(black)* lies on the input polygon *(dark grey)*. The envelope *(light grey)* matches the input polygon in its lower part.

In this formulation, CHANNEL can be solved by a standard solver for linear programs.

To keep b as close as possible to a straight line, for which $\Delta_2 b_j = 0$, we use the cost function

$$\sum_{i=1}^{m-1} (\Delta_2 b_i)^2 \to \min.$$

This results in a quadratic program that again can be solved by a standard solver.

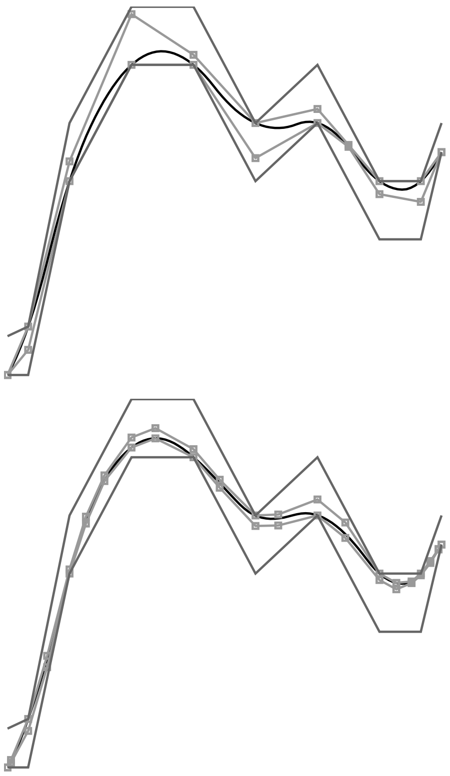

FIGURE 7. A tangent continuous quadratic *(top)* and a curvature continuous cubic *(bottom)* solution to a CHANNEL problem. The input polygons $\underline{c}$ and $\overline{c}$ *(dark grey)* enclose the splines envelope *(light grey)* and hence the spline *(black)*.

3.4. Initial knot sequence. The user may choose the initial non-decreasing knot sequence freely, for example as the abscissae of the corners of $\underline{c}$. Algorithm 1 below generates a short knot sequence (t_i) such that the corresponding sequence of Greville abscissae include the breakpoints x_i of the input polygon(s). The idea is to introduce additional knots (and Greville abscissae) where the input sequence (x_i) contains gaps that are large relative to the surrounding intervals.

3.5. Knot insertion. While we are assured of a solution to SUPPORT, for example a straight line above $\underline{c}$, CHANNEL may not have a solution for the given requirements of degree, smoothness and break points. This difficulty is overcome by inserting knots where the channel constraints are violated. Concretely, we augment the knot sequence (t_k) with additional breaks near the breaks of the input polygons where the envelope returned by the linear or quadratic program violates the CHANNEL conditions. This adaptive strategy is guaranteed to yield a solution since the envelope of b converges to b under refinement of the knot sequence. A rigorous proof of finite termination inserts knots equidistantly so that the simple bound for uniform splines, $|b-\ell| \leq (d+1)/24 \max|\Delta_2 b_i|$, applies which contracts by $1/4$ as the knot distance is halved.

Algorithm 1 Compute an interpolating knot sequence (t_k) for a B–spline basis of degree d such that the Greville abscissae contain the sequence $x_0 < x_1 < \cdots < x_m$.

Require: $x_0 < x_1 < \cdots < x_m$ and $d > 1$
  $t_0 = t_1 = \cdots = t_d \leftarrow x_0$
  $k \leftarrow d$
  for $l \leftarrow 1 \ldots m-1$ **do**
    repeat
      $t_{k+1} \leftarrow d\, x_l - \sum_{j=k-d+2}^{k} t_j$
      if $t_{k+1} > x_{l+1}$ **then** $--$ New knot too big, add intermediate Greville abscissae
        $t_{k+1} \leftarrow x_l - \min\{x_{l+1} - x_l, x_l - t_k\}/2$
      $k \leftarrow k+1$
    until $d\, x_l = \sum_{j=k-d+1}^{k} t_j$
  $t_{k+1}, \ldots, t_{k+d} \leftarrow x_m$
Ensure: (t_k) interpolating knot sequence
Ensure: For any $0 \leq i \leq m$ exists a $k \geq i$ with $x_i = \sum_{j=k+1}^{k+d} t_j$

3.6. Examples. Figure 7 shows two solutions to a CHANNEL problem. Both were computed using the quadratic cost function above and the knot sequence generated by algorithm 1. For the quadratic spline, the control polygon has only one more break than the input polygon, while the cubic spline has twice as many control points as there are points in the input polygon.

REFERENCES

[Buc65] 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.

[Dah86] W. Dahmen. Subdivision algorithms converge quadratically. *JCAM*, 16:145–158, 1986.

[dB78] Carl de Boor. *A practical guide to splines*. Springer-Verlag, New York, 1978.

[dBvKOS97] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational geometry*. Springer-Verlag, Berlin, 1997. Algorithms and applications.

[DeV68] R. DeVore. One-sided approximation of functions. *J of Approximation Theory*, 1:11–25, 1968.

[DGKN98] 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.

[LP9x] David Lutterkort and Jorg Peters. The distance between a (B)-spline and its control polygon. *Numerische Mathematik*, 199x. submitted, see also, TR 98–013, September 1998, Dept CISE, University of Florida.

[Pin89] Allan M. Pinkus. *On L^1-approximation*. Cambridge University Press, Cambridge, 1989.

[PK94] H. Prautzsch and L. Kobbelt. Convergence of subdivision and degree elevation. *Advances in Computational Mathematics*, 2:143–154, 1994.

DAVID LUTTERKORT, DEPT. OF COMPUTER SCIENCE, PURDUE UNIVERSITY, WEST LAFAYETTE, IN 47907–1398

E-mail address: lutter@cs.purdue.edu

URL: http://www.cs.purdue.edu/people/lutterdc

JÖRG PETERS, DEPTARTMENT OF COMPUTER AND INFORMATION SCIENCE, CSE ROOM 326, UNIVERSITY OF FLORIDA, GAINESVILLE, FL 32611–6120

E-mail address: jorg@cise.ufl.edu

URL: http://www.cise.ufl.edu/˜jorg