Smooth Blending of Basic Surfaces using Trivariate Box Splines

Jörg Peters *   Michael Wittman

October 23, 1996

Abstract

To blend between basic implicitly defined CSG surfaces we propose to use the zero set of a spline in three variables. The resulting blend surface is generically curvature continuous, of algebraic degree four independent of the number of surfaces joined, and supports both point classification and efficient rendering. A detailed exposition of the 2D analogue blend construction is given.

1 Introduction

Blending surfaces that smoothly join basic primitives such as planar facets, quadrics or the torus has motivated extensive research both on parametrically and implicitly defined surfaces (see e.g. the surveys [21], [12]). The alternative approach explained in this paper may be used to smoothly join an arbitrary number of basic surfaces using a representation of fixed low algebraic degree and a fixed number of polynomial pieces. The representation is based on box splines and has the following properties:

- The blend surface is curvature continuous and joins C^2 with input surfaces that are separate or join C^2 outside the blend volume. The approximation order to general smooth surfaces is $O(h^4)$.
- The algebraic degree of the blend surface is four independent of the number of input surfaces.
- The blend surfaces can be rendered stably and moderately fast.
- The representation is compatible with set-theoretic representations; e.g. point classification (set membership determination) is supported.

*Supported by NSF National Young Investigator grant 9457806-CCR

- The surface representation has volume elements associated with it.
- The blend is volume bounded.

Related piecewise polynomial approaches. Besides the well-known implicit blending constructions of [3], [10], [11], [13], there are currently two main approaches to defining the individual pieces of a function whose zero set represents a surface. The first is to generate an approximate triangulation of the surface and then erect a shell-like structure of trivariate polynomial pieces over this parametrization [20], [5], [9], [4], [1], [14]. The second is to define a function on a regular, global lattice, for example, a piecewise triquadratic, C^1 tensor-product spline [15]. The regular lattice has the advantage that non-rectilinear features of the surface do not require special treatment and that no parametrization other than the lattice structure is imposed. The algorithm defined below is of the second kind. It can be viewed as a systematic way of creating a field in the spirit of 'blobby objects' used in animation [8] or as approximate version of constructive geometry in the sense of Ricci [19].

The paper is structured as follows. After the statement of the algorithm, the third section explains the approach in terms of analogous 2D curve-blending. Section 4 gives the details of the 3D construction. Section 5 discusses the properties of the representation and gives examples. The Appendix reviews box splines to the extent used in this paper.

2 The algorithm

The *input* to the algorithm are

a. the defining polynomials of surfaces of degree $d \leq 4$,

b. a blend volume and its partition,

c. an operation and scaling factors, and

d. a number of averaging steps.

The details of the input will become clear as we work out a detailed example of Section 3. For a quick overview consider the following hints. The defining polynomial of the unit sphere $p_1 \geq 0$ is $p_1(\mathbf{x}) = p_1(x, y, z) = -x^2 - y^2 - z^2 + 1$. The *blend volume* is a typically block-shaped region in 3-space that encloses the blend surface to be constructed. The blend volume need not be aligned with the global xyz coordinate axes but may be the image of a cube under a more general map. An example of an operation is to choose the maximum of a set of coefficients. This will correspond to an approximate union.

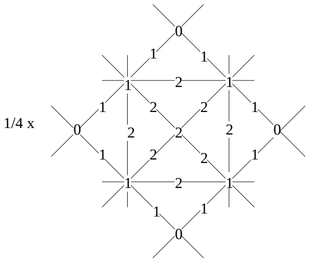


Figure 1: Bernstein coefficients of the Zwart-Powell element M.

The three *steps of the algorithm* are as follows. First, each of the n basic primitives within the blend volume is automatically rerepresented as a trivariate spline. That is, the defining polynomial of each basic primitive, is written as a linear combination of shifts of the spline basis function M. For example, the first defining polynomial is $p_1(\mathbf{x}) = \sum a^1_\alpha M(\mathbf{x} - \alpha)$, where $\alpha \in R^3$ is a point on a lattice that sub-divides the blend volume. Explicit formulas for the a^1_α in terms of the polynomial to be reproduced are given in Sections 3 and 4. The zero set of the spline then defines the surface. Second, the operation specified generates from the n 3D arrays of spline coefficients in each position one entry and hence a new 3D array and associated spline, $p(\mathbf{x}) = \sum a_\alpha M(\mathbf{x} - \alpha)$. Typically, when blending, a_α is defined as $a_\alpha := \max_\ell\{a^\ell_\alpha\}$ except for α in the boundary set. The *boundary set* for blending consists of α within a certain range of a sign change on the boundary entries of one of the arrays where all other arrays have negative entries. That is, where a single input surface meets the blend volume its coefficients are preferred so as to obtain a smooth transition. This preference is also kept during the third step, local averaging which smoothes features. When surfaces are added, blends on blends are avoided by regenerating the blend based on all primitives.

3 The four-direction box spline and curve blending

The general approach, discrete blending in coefficient space, is most easily explained in terms of a lower dimensional (and less smooth and less reproducing) analogue, a C^1 quadratic curve blend in the plane. We base the blend on the centered C^1 piecewise quadratic box spline, the so-called Zwart-Powell element. Figure 1 shows the Bernstein-Bézier coefficients of the quadratic pieces. Example 2 of the Appendix discusses the Zwart-Powell element detail (c.f. Figure 12,lower right). In particular, M has the property that any quadratic $q(x, y)$ can be expressed a linear combination of its integer shifts $\alpha := (\alpha_1, \alpha_2) \in \mathbb{N}^2$; that is, there exist scalars $a(\alpha)$ such that the following (Marsden) identity holds

$$q(x, y) = \sum a(\alpha) M((x, y) - \alpha). \qquad (*)$$

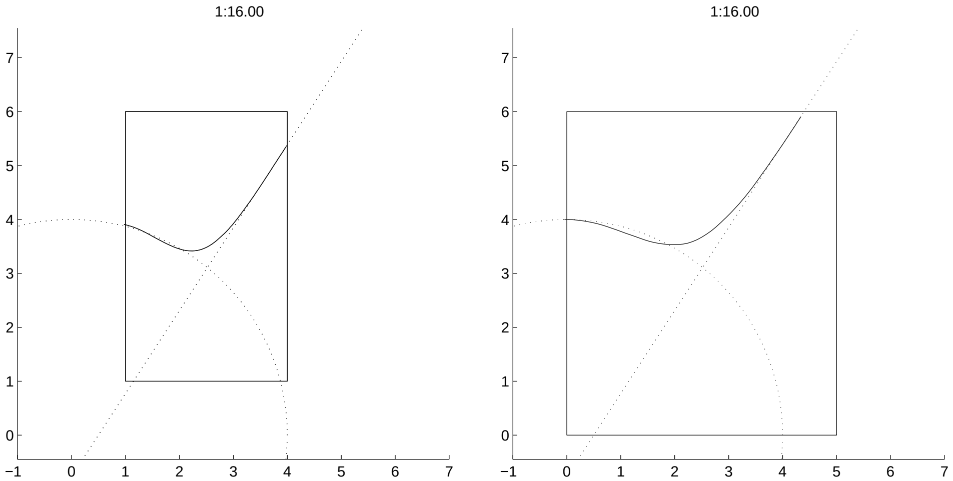


Figure 2: Blends under variation of blend volume size.

To make the example concrete we blend a circular arc with a line segment. Specifically, we choose the input

$p_1(x,y) := -x^2 - y^2 + 16$	defining polynomial for disk of radius 4
$p_2(x,y) := -x + .65y + 3.5$	defining polynomial for half-space
$\lambda := [1, 16]$	scaling of defining polynomials
$\text{bb} := [1,1] \times [4,6],$	lower left and upper right of blend volume
$n := [8,6]$	partition of extended blend volume into 8 × 6 cells
$n_{\text{avg}} := 1$	one averaging step

The output is displayed in the left panel of Figure 2.

For the first step of the algorithm, we need to reexpress the defining polynomials p_i in terms of the integer shifts of M. The scalar weights $a(\alpha)$ depend on q but since $q(x,y) = \sum_{i+j\leq 2} c_{ij} x^i y^j$, it suffices to compute the weights $a_{ij}(\alpha)$ for the six basis functions $x^i y^j$ and set

$$a(\alpha) := \sum_{i+j\leq 2} c_{ij} a_{ij}(\alpha).$$

For given monomials $\alpha^{ij} := \alpha_0^i \alpha_1^j$, the values for the $a_{i,j}$ can be read off from the following matrix equation derived by evaluating $(*)$ at the integers.

$$\begin{pmatrix} a_{20} \\ a_{11} \\ a_{10} \\ a_{02} \\ a_{01} \\ a_{00} \end{pmatrix} (\alpha) = \begin{pmatrix} 1 & & & & & -1/4 \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & -1/4 \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix} \begin{pmatrix} \alpha^{20} \\ \alpha^{11} \\ \alpha^{10} \\ \alpha^{02} \\ \alpha^{01} \\ \alpha^{00} \end{pmatrix}$$

For example, the coefficients corresponding to p_1 and p_2 are

$$a^1(\alpha) * \lambda_1 = -\alpha_0^2 - \alpha_1^2 + 16.5$$
$$a^2(\alpha) * \lambda_2 = 16\alpha_0 - 10.4\alpha_1 - 8$$

To map the integer indices of the array onto the partition of the blend volume, let A be the transformation that maps a box of size $n_1 \times n_2$ at the origin to the blend volume. Then the 2D array V is initialized as

$$V(\beta) := a(\alpha) = a(A(S_M\beta)), \quad \beta \in [0..n_1] \times [0..n_2].$$

S_M is a shift by $1/2$ followed by scaling by $(n_i + 1)/n_i$ in the ith component. This enlargement achieves that evaluation by subdivision converges exactly to the blend volume. Returning to the concrete example, we obtain $\alpha_0 = \frac{3}{8}\beta_0 + 1$, $\alpha_1 = \frac{5}{6}\beta_1 + 1$.

In the second step, the two arrays a^1 and a^2 are merged by choosing in each position the maximum entry, except for the boundary set. Within a index radius of 1 from where the input curve enters the blend volume, the coefficients of that curve are retained provided the other array has negative entries. This boundary set is computed by marking the sign changes of a^1 and a^2 at the array boundary. In the example, there are four such changes but only the upper two are a center of a boundary set.

Figure 3 shows the effect of varying λ. While the zero set that defines the original curves remains unchanged, scaling effects the choice of array entries of the blend. Reducing λ increases the chance that negative coefficients of the corresponding array are selected and decreases the chance of positive coefficients being selected. As Figure 3 demonstrates, this effect can be both desirable and undesirable. By default the slopes in the region of intersection should be of similar order of magnitude. To counteract poor settings of λ a final averaging step is added in which the boundary set is kept fixed and the remaining coefficients are locally averaged by a Gaussian mask to smooth the resulting zero set. This averaging is cheap since there are only $n_1 \times n_2$ interior coefficients.

The output of the algorithm is a single $(n_1 + 1) \times (n_2 + 1)$ array. The leftmost panel of Figure 6 shows such an array together with the blend volume, the input curve

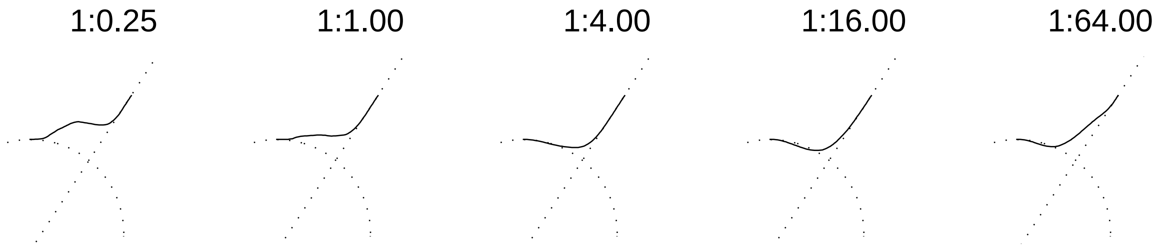


Figure 3: Blends under variation of λ without smoothing.

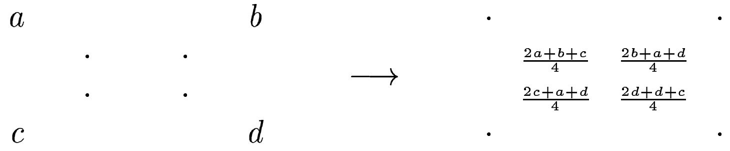


Figure 4: Four-direction box spline subdivision.

outside the blend volume, and an approximation to the zero set of the box spline inside the (extended) blend volume. The array entries are displayed as points shaded proportional to their value, darker for more negative values. To obtain successively better approximations, the array is refined by *subdivision*, essentially quadrupling the number of coefficients at each step. Figure 4 shows the relationship between the entries of an array and the array corresponding to the subdivided function. The refinement rule is derived from the general subdivision algorithm for box splines in the Appendix (see in particular, Figure 11). Figure 5 middle and right show two consecutive steps of subdivision for the circle. Note that the parameter region for α converges towards the blend volume.

We conclude the section with two more examples of blending, Figure 6 and Figure 7. The intersection is achieved by taking the minimum of the coefficients.

4 Blend Surfaces

Extending the blending technique from curves to surfaces requires conceptually nothing new. Instead of the centered 4-direction piecewise quadratic box spline, we choose for M the centered 7-direction quartic box spline in three variables. This implies the following changes. The surfaces will generically be C^2, the subdivision averages 7 directions, S_M shifts by 1.5 in each direction, and hence $n_i \geq 4$ should hold, the boundary set consists of the indices within an index range of 2 from sign changes on the array boundary, and

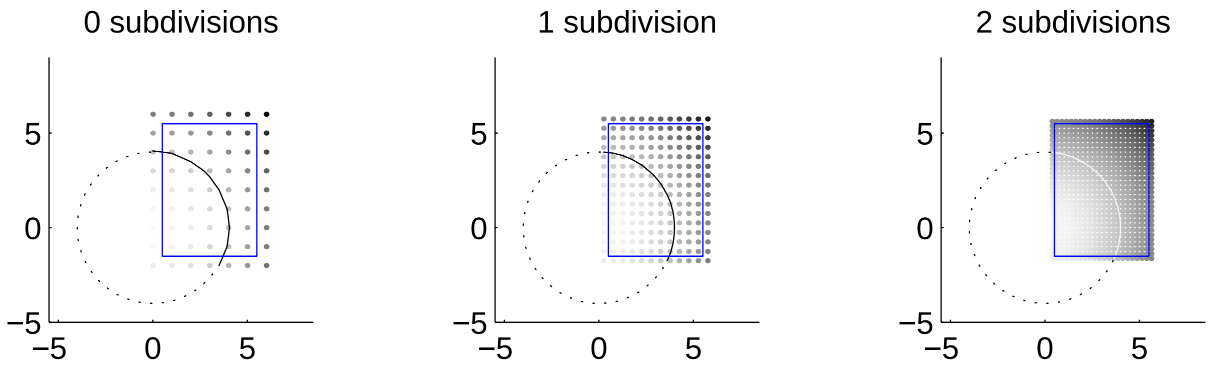


Figure 5: Representation of $-x^2 - y^2 + 16 >= 0$

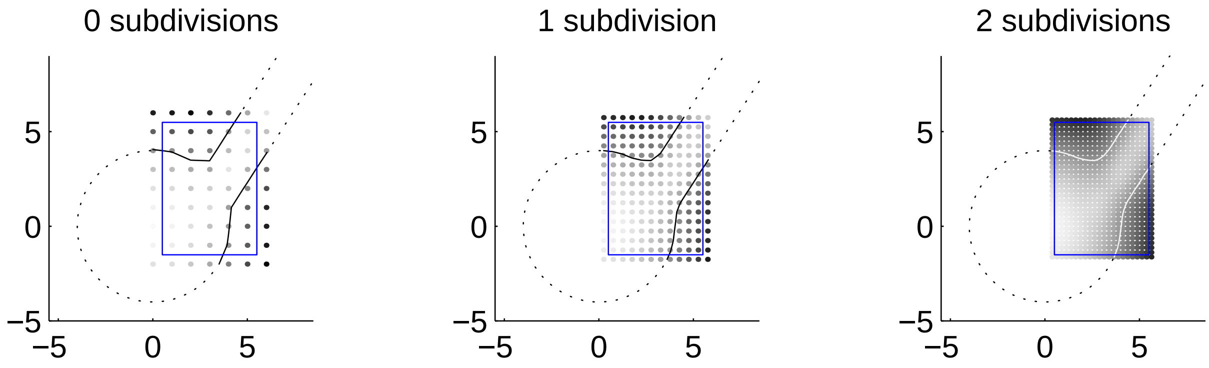


Figure 6: 2D C^1 quadratic blend of $-x^2 - y^2 + 16 >= 0$ union $(-x + .65y + 3.5 >= 0$ intersect $x - .65y - .75 >= 0)$.

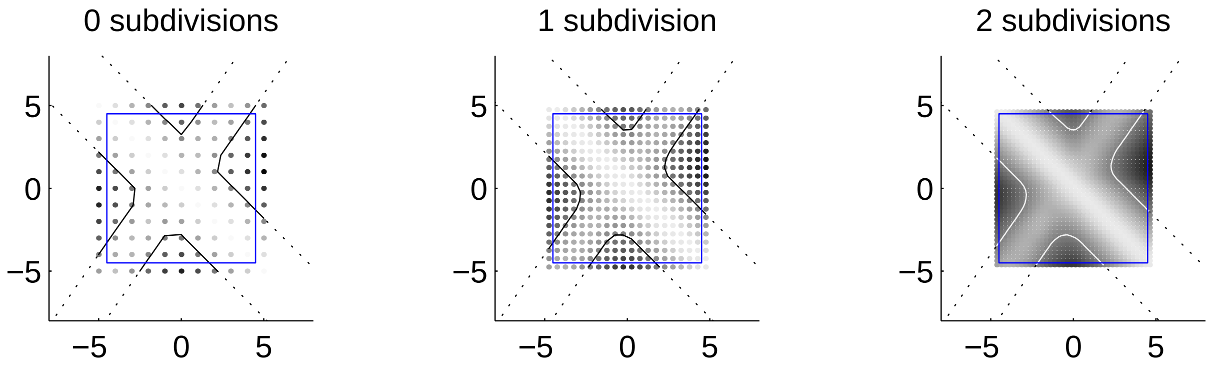


Figure 7: C^1 quadratic blend for $(-x + .7y + 1 >= 0$ intersect $x - .7y + 2.2 >= 0)$ union $(x + y + 2.8 >= 0$ intersect $-x - y + 3.2 >= 0)$.

the 3D array V has the entry $a(\alpha) = a(A(S_M\beta))$ at position $\beta = (\beta_0, \beta_1, \beta_2)$ where $a(\alpha) := \sum_{i+j+k\leq 3} c_{ijk}a_{ijk}(\alpha)$. and

$$\begin{pmatrix} a_{200} \\ a_{110} \\ a_{101} \\ a_{100} \\ a_{020} \\ a_{011} \\ a_{010} \\ a_{002} \\ a_{001} \\ a_{000} \end{pmatrix} (\alpha) = \begin{pmatrix} 1 & & & & & & & & & -5/12 \\ & 1 & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 1 & & & & & & \\ & & & & 1 & & & & & -5/12 \\ & & & & & 1 & & & & \\ & & & & & & 1 & & & \\ & & & & & & & 1 & & -5/12 \\ & & & & & & & & 1 & \\ & & & & & & & & & 1 \end{pmatrix} \begin{pmatrix} \alpha^{200} \\ \alpha^{110} \\ \alpha^{101} \\ \alpha^{100} \\ \alpha^{020} \\ \alpha^{011} \\ \alpha^{010} \\ \alpha^{002} \\ \alpha^{001} \\ \alpha^{000} \end{pmatrix}$$

For example, to represent the cylinder

$$(x-y)^2 + z^2 = 1$$

the array entry $(\beta_1, \beta_2, \beta_3)$ for an integer spaced blend volume at the origin is

$$(\beta_1^2 - \frac{5}{12}) - 2\beta_1\beta_2 + (\beta_2^2 - \frac{5}{12}) + (\beta_3^2 - \frac{5}{12}) - 1.$$

5 Discussion of the blend surfaces

5.1 Continuity

Within the blend volume the blend surface is the zero set of a C^2 function. If this function is regular within the blend volume then, by the implicit function theorem, its zero set is also C^2. It is however, also possible to represent singularities such as the apex of a cone. A C^2 join with the input surface across the boundary of the blend volume is guaranteed if the boundary sets do not overlap since then the blend surface agrees exactly with the input surface.

5.2 Rendering, evaluation and point-classification

To extract a continuous piecewise linear approximation of the zero sheet of the spline, we traverse the coefficient array to detect sign changes. In standard fashion, here additionally motivated by the tetrahedral support of the polynomial pieces, each cube with a sign change is split into tetrahedra according to Figure 10, associating the average of the values at the vertices with the center of each cube and cube face. For each edge whose endpoints have an opposite sign, we mark the midpoint. Each tetrahedron has either zero, three or four marked edge-midpoints. Correspondingly, we add no, one or two (coplanar) triangles connecting the midpoints to a list of triangles. The union of the triangles in the list then form the surface approximation.

The surface approximation is refined by averaging the array entries according to the subdivision rules of the 7-direction box spline (cf. the Appendix). That is, each value is replicated over a cube of half the edge length and then the values on this refined lattice are averaged consecutively in each of the four diagonal directions of the box spline. With each step of the subdivision, the approximation gains two additional digits of accuracy.

To test for intersection, we subdivide depth-first to obtain a nested sequence of bounding boxes. If a single point is to be classified, the check is first against the blend volume then against the subcubes and tetrahedra already generated. If this check fails because the point is very close to the boundary, the spline is exactly evaluated. A stable evaluation algorithm and code for the box-spline basis functions are given in [6].

5.3 Shape and detail control

Since the spline is the limit of the coefficients generated by the subdivision process, and subdivision reduces variation through averaging, its rough shape and hence the shape of its zero set can be inferred from the 3D array of coefficients. The blend can be modified by scaling the array-entries other than the boundary sets. This can be done by local averaging, by changing the scale λ, or by changing the blend volume(cf. Figures 3,2).

5.4 Compatibility with existing shape description techniques.

Combining the blend volume-approach with a set-theoretic modeling environment is straightforward: subtract the volume covered by the blend volume and add the blend volume. The 3D examples consist of clipped parametric or implicit primitives outside the blend volume while the data inside are specified by Boolean expression over the coefficient arrays.

5.5 Examples

Examples of 3D blend surfaces are shown in Figure 8.

Row 1 shows evaluation by subdivision: zero, one and two subdivision applied to a cylinder intersection. Without sparse data structures the unrefined zero set displays in real time; a 3rd level subdivison takes ca 1 minute mostly due to the display of triangles.

Row 2 shows the blending process: two (Inventor) primitives, their union and their smooth blend. The highlighted region is the surface represented by the 7-direction box spline.

Row 3 shows various implicit primitives blended.

Row 4 shows the join of two tori: the torus primitive, the box spline blend and the complete blended object.

6 Conclusion

Smooth surfaces as the zero set of a piecewise polynomial function on a fixed-grid are capable of modeling free-form objects (see e.g. [15], [16]). However, this representation is clearly less efficient for many tasks than parametric spline-based surfacing. Our implementation is therefore specific to isolated blending applications.

The proper intialization of the 3D blend volume is the key to a good blend surface. The main concern with the present implementation is the impact of scaling of the defining polynomials.

References

[1] Bajaj, C., Chen, J., and Xu, G. Free-form surface design with A-patches. In *Graphics Interface '94* (1994), Canadian Man-Computer Communications Society.

[2] Boor, C. D., Höllig, K., and Riemenschneider, S. *Box splines.* Springer Verlag, 1994.

[3] C. Hoffmann, J. H. The potential method for blending surfaces and corners. In *Geometric Modeling, Algorithms and new trends*, G. Farin, Ed. SIAM, 1987, pp. 347–365.

[4] Dahmen, T.-M. T.-S. Cubicoids: modeling and visualization. *Computer Aided Geometric Design10*, 93 (1993), 93–108.

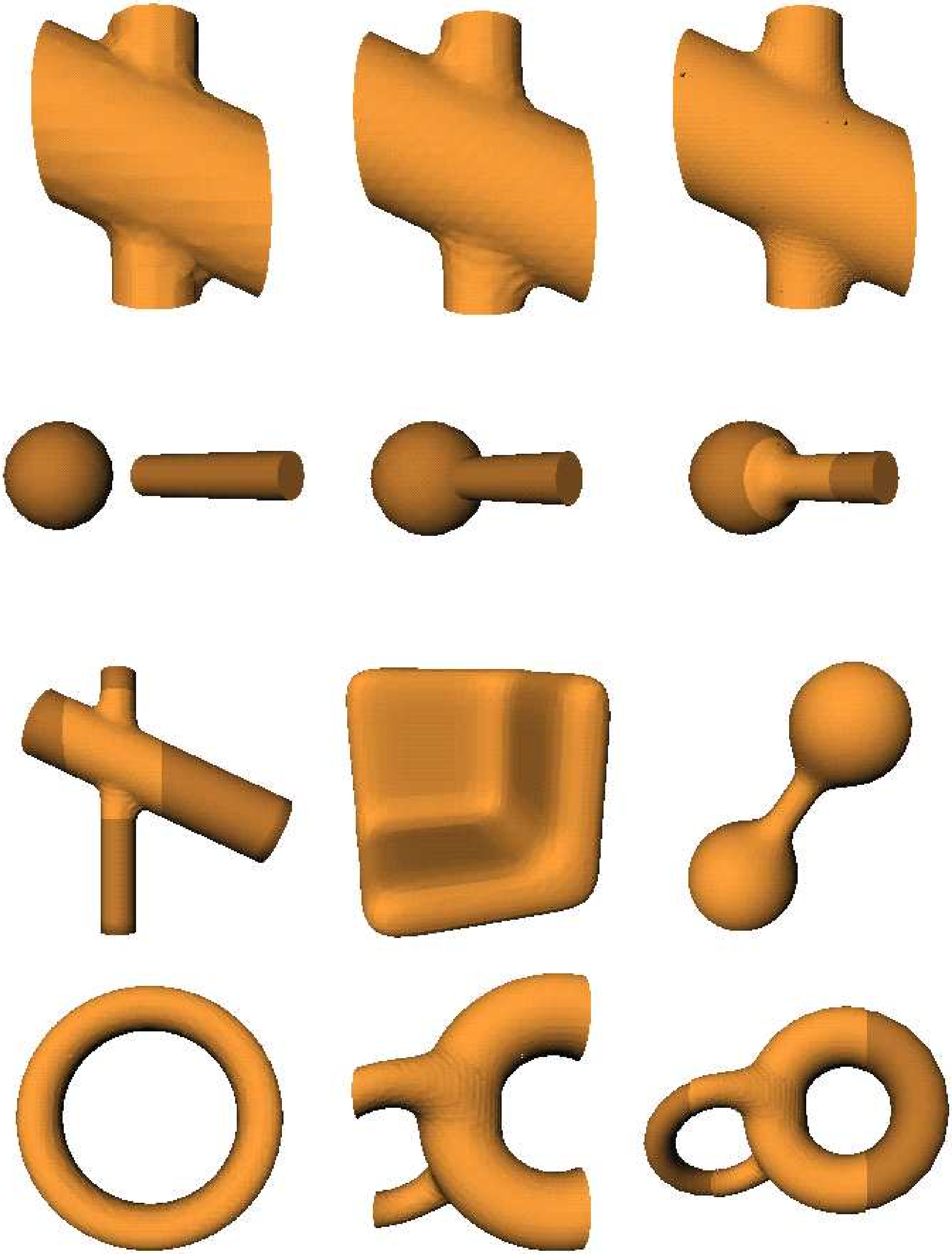
[5] Dahmen, W. Smooth piecewise quadratic surfaces. In *Mathematical Methods in Computer Aided Geometric Design*, T. Lyche and L. Schumaker, Eds. Academic Press, Boston, 1989, pp. 181–193.

[6] de Boor, C. On the evaluation of box splines. *Numer.Algorithms 5* (1993), 5–23.

[7] de Boor, C., and DeVore, R. Approximation by smooth multivariate splines. *Trans Amer Math Soc 276* (1983), 775–788.

[8] G. Wyvill, C. Mc Pheeters, B. W. Data structure for soft objrects. *The Visual Computer 2*, 4 (1986), 227–234.

Figure 8: see Section 5.5.



[9] Guo, B. *Modeling arbitrary smooth objects with algebraic surfaces.* PhD thesis, Computer Science, Cornell University, 1991.

[10] Holmström, I. Piecewise quadric blending of implicitly defined surfaces. *Computer Aided Geometric Design 4* (1987), 171–190.

[11] J. Li, J. Hoschek, E. H. G^{n-1}-functional splines for interpolation and approximation of curves, surfaces and solids. *Computer Aided Geometric Design 7* (1990), 209–220.

[12] J. Vida, R.R. Martin, T. V. A survey of blending methods that use parametric surfaces. *Computer Aided Design26*, 5 (1994), 341–365.

[13] Kosters, M. Quadratic blending surfaces for complex corners. *The Visual Computer*, 5 (1989), 134–146.

[14] Middleditch, D. Solid models with piecewise algebraic free -form faces. *CSG* (1994).

[15] Moore, D., and Warren, J. Approximation of dense scattered data using algebraic surfaces. In *Proceedings of the 24th Hawaii Intl. Conference on Computer Systems Sciences, Kaui, Hawaii* (1991), pp. 681–690.

[16] Peters, J. C^2 surfaces built from zero sets of the 7-direction box spline. In *Mathematical Methods in Curve and Surface Design VI* (1996), G. Mullineux, Ed.

[17] Powell, M. Piecewise quadratic surface fitting for contour plotting. In *Software for Numerical Mathematics* (1969), D. Evans, Ed., Academic Press, pp. 253–271.

[18] Powell, M., and Sabin, M. Piecewise quadratic approximation on triangles. *ACM Trans. of Math. Software 3* (1977), 316–325.

[19] Ricci, A. A constructive geometry for computer graphics. *Computer Journal 16*, 2 (1974), 157–160.

[20] Sederberg, T. Techniques for cubic algebraic surfaces, tutorial part ii. *IEEE Computer Graphics and applications 10*, 5 (1990), 12–21.

[21] Woodwark, J. Blends in geometric modeling. In *Mathematics of surfaces II*, R. R. Martin, Ed. Clarendon Press, 1987, pp. 255–297.

[22] Zwart, P. Multivariate splines with non-degenerate partitions. *SIAM J. of Num Analysis 10* (1973), 665–673.

7 Box splines

Box splines represent a generalization of univariate spline theory to several variables. Since box splines were introduced by de Boor and DeVore [7] a rich theory has been developed and collected in the "box spline book" [2] which serves as reference for the following exposition of these piecewise polynomial functions.

The box spline M_Ξ in s variables is defined by the $s \times n$ matrix Ξ (pronounced Xi) with columns in $R^s \backslash 0$. For the purposes of this paper we may assume that the first s columns of Ξ form the identity matrix I. This yields the following inductive definition of the box spline. If $\Xi = I$, then M_Ξ is the function that is 1 on the unit cube and 0 elsewhere:

$$M_I(x) := \begin{cases} 1, & \text{if } x \in [0..1)^s, \\ 0, & \text{else.} \end{cases}$$

This box spline is piecewise constant, has degree zero and is discontinuous. If $\Xi \cup \xi$ is any matrix formed from Ξ by the addition of the column $\xi \in R^s$, then the box spline $M_{\Xi \cup \xi}$ is given by the convolution equation

$$M_{\Xi\cup\xi}(u) = \int_0^1 M_\Xi(u - t\xi).$$

If $s = 1$ then this is exactly the B-spline construction by convolution.

7.1 Box spline properties

The box spline has the following properties.

(i) M_Ξ is positive and its shifts sum to one: $\sum_{\alpha \in Z^s} M_\Xi(\cdot - \alpha) = 1$.

(ii) The support of M_Ξ is $\Xi[0..1)^s$, i.e. the set sum of the columns contained in Ξ.

(iii) M_Ξ is piecewise polynomial of degree $n - s$. That is, each convolution in another direction ξ increases the degree by one.

(iv) M_Ξ is $\rho - 2$ times continuously differentiable, where ρ is the minimal number of columns that need to be removed from Ξ to obtain a matrix whose columns do not span R^s.

(v) M_Ξ reproduces all polynomials of degree $m := \rho - 1$ and none of degree higher than $n - s$.

(vi) The L^p approximation order of the spline space $S :=$ span $(M(\cdot - \alpha))$ is ρ. That is with the refinement of the lattice $x \to x/h$, $h < 1$, $\operatorname{dist}(f, \sum a(\alpha) M_\Xi((\cdot - \alpha)/h)) = O(h^\rho)$ for all sufficiently smooth f.

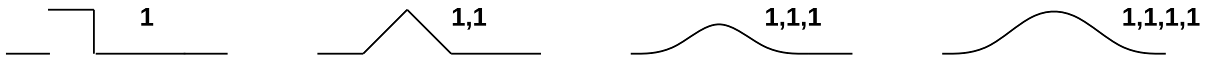


Figure 9: Uniform univariate splines

Thus the n columns of Ξ, which may be interpreted as directions in R^s, determine the support of the piecewise polynomial and its continuity properties. Understanding the number ρ requires an analysis of the independent submatrices of Ξ.

7.2 Box spline examples

We develop the three examples relevant to this paper.

1. The well-known univariate uniform cubic B-spline has the matrix (direction set)

$$\Xi := \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

Figure 9 shows in order the characteristic function of the 1-dimensional cube and its repeated convolution in the direction 1 yielding the linear 'hat' function, the quadratic and finally the cubic B-spline. We determine the characteristic numbers as

$$s = 1, n = 4, \text{ and } \rho = 4$$

since all elements of the set have to be removed to make it nonspanning in R^s. The degree of the B-spline pieces is $n - s = 3$ and the continuity is of order $\rho - 2 = 2$ as expected. The cubic spline formed as a linear combination of B-splines is guaranteed to at least reproduce polynomials of degree $4 - 4 = 0$ and none of degree higher than 3. The approximation order is 4.

2. The bivariate box spline M_Ξ based on the matrix

$$\Xi := \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

is called Zwart-Powell element [22], [17], [18]. It is displayed in Figure 12 (lower right). The characteristic numbers are

$$s = 2, n = 4, \text{ and } \rho = 3$$

and hence the element is of degree 2 and its polynomial pieces are connected C^1. Since $n - \rho = 2 = n - d$ for the ZP-element, its linear combination reproduce exactly all quadratic polynomials; that is any quadratic $q(x, y)$ can be written as

$$q(x, y) = \sum_{\alpha \in Z^2} a(\alpha) M_\Xi((x, y) - \alpha).$$

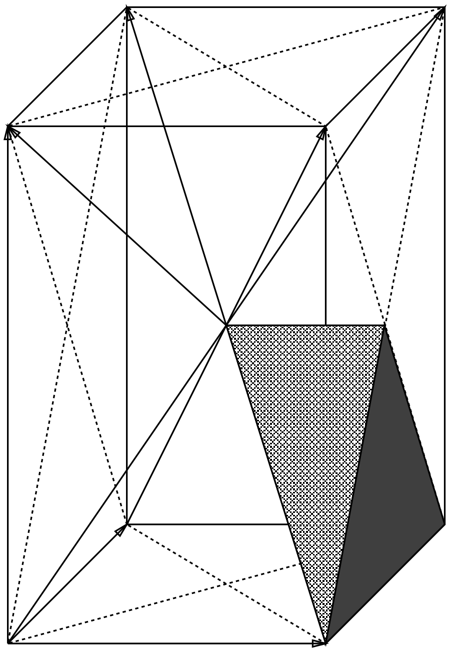


Figure 10: The 7 directions of the box spline and its domain tetrahedra

The Zwart-Powell element stands out among the low-degree box-splines defined over the plane, in that it has maximal smoothness equal to the degree minus one and is piecewise polynomial over a regular triangulation.

3. The 7-direction box spline is a similar serendipity element among the trivariate box splines. It is based on the direction matrix

$$\Xi := \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 1 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

The seven directions defined by the colums of the matrix cut R^3 into a symmetric regular arrangement of tetrahedra. The characteristic numbers of the 7-direction box spline are

$$s = 3, n = 7, \text{ and } \rho = 4.$$

Thus the polynomial piece defined over each tetrahedron is of degree $n-s = 4$ and splines formed as a linear combination of shifts of the box spline are $C^{\rho-2} = C^2$. Elements of the spline space reproduce all cubics in three variables (and some additional polynomials of degree four) and the approximation order is 4.

$$\begin{array}{cc} & \\ 4a & 4b \\ & \\ 4c & 4d \end{array} \quad \rightarrow \quad \begin{array}{cccc} a & a & b & b \\ a & a & b & b \\ c & c & d & d \\ c & c & d & d \end{array} \quad \rightarrow_{1/2} \quad \begin{array}{cccc} . & . & . & . \\ aa & ab & bb & . \\ ac & bc & bd & . \\ cc & cd & dd & . \end{array} \quad \rightarrow_{1/4} \quad \begin{array}{cccc} . & . & . & . \\ . & . & . & . \\ . & aabc & abbd & . \\ . & accd & bcdd & . \end{array}$$

Figure 11: Four-direction box spline subdivision.

7.3 Box spline subdivision

To quickly approximate any box spline we may use subdivision. Since the shifts of the box spline M_Ξ form a nonnegative, local partition of unity, a spline formed as a linear combination of shifts of the box spline is a finite convex combination of its coefficients $a(\alpha)$. To the extent that the local variation of the coefficients is small, the coefficients $a(\alpha)$ approximate the spline well. This is the basis for fast algorithms for graphic display and rendering. The key observation is that the variation of the coefficients is reduced when the spline is expressed in terms of box splines corresponding to the refined lattice $\frac{1}{2}Z^s$:

$$\sum_{j \in Z^s} a(j)M(x-j) = \sum_{k \in \frac{1}{2}Z^s} a_{1/2}(k)M(2(x-k)).$$

The successive computation of a sequence of refined coefficients $a_1, a_{1/2}, \ldots$ is called a **subdivision algorithm:** We compute $a_{h/2}$ from a_h for α on the finer mesh $\frac{h}{2}Z^s$. First set

$$a_{h/2}(\alpha) := \begin{cases} 2^s a_h(\alpha), \text{if } \alpha \in hZ^s \\ 0, \text{else} \end{cases} .$$

Then average in each of the directions in Ξ. That is for each $\xi \in \Xi$ compute, careful not to overwrite still needed values,

$$a_{h/2}(\alpha) \leftarrow (a_{h/2}(\alpha) + a_{h/2}(\alpha - \xi/2))/2.$$

Under mild assumptions on the matrix Ξ that are satisfied by all three box splines defined above, the sequence of control points converges quadratically to the spline [2], (30)Theorem, page 169.

The sequence of array entries for the subdivision of a spline with coefficients a, b, c, d and M the ZP-element is displayed in Figure 11.

To illustrate the effect of subdivision as approximate evaluation, we choose one box-spline coefficient (at the origin) non-zero, and all other coefficients equal zero, i.e.

$$a(\alpha) = \begin{cases} 1, & \text{if } \alpha_1 = \alpha_2 = 0 \\ 0, & \text{else} \end{cases} \qquad \alpha := (\alpha_1, \alpha_2).$$

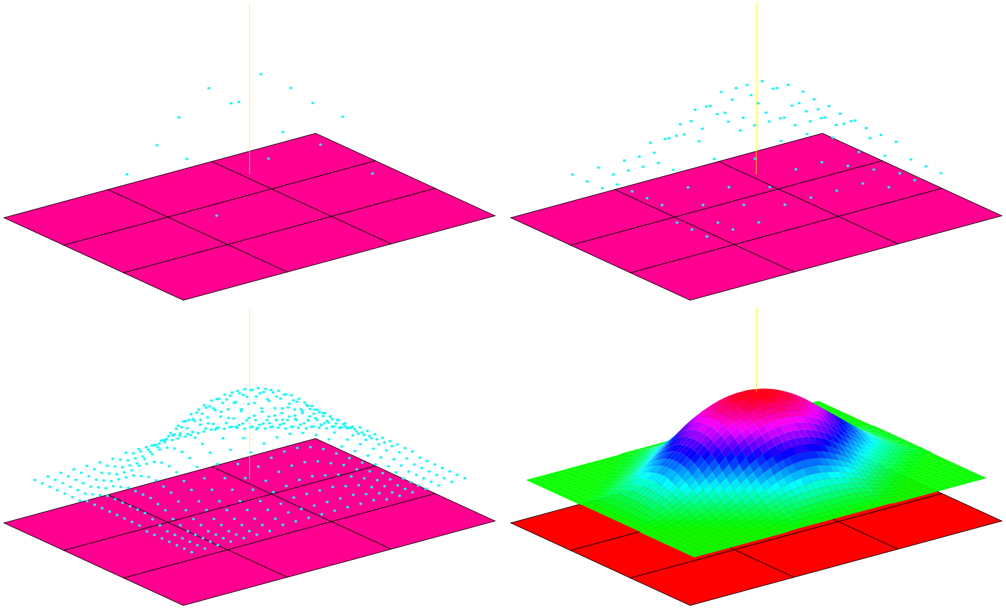


Figure 12: The Zwart-Powell element M_Ξ approximated using 4 steps of subdivision. The point cloud are the coefficients generated by the refinement.

Then

$$\sum_\alpha a(\alpha) M_\Xi((x,y) - \alpha) = M_\Xi(x,y),$$

and the spline represents just a single basis function. Figure 12 below shows four steps of subdivision on the spline coefficients. The central spike is of height 1.

One can convert a piecewise polynomial in box-spline form into any other piecewise polynomial representation such as the power form or the Bernstein form. For example, in the Bernstein-Bézier form, the Zwart element is represented by 28 quadratic pieces with coefficients 1/2, 1/4, 1/8 and 0 (c.f. Figure 12).