Splines over iterated Voronoi diagrams
Draft

Gerald Farin
November 23, 2003

Abstract
We present a surface generation method which produces B-spline-like surfaces (or curves) in any dimension. We focus on the 2D quadratic case.

1 Introduction

The concept of B-spline curves and surfaces is central to the field of CAGD, see [8]. While B-spline curves are tremendously successful (and sometimes referred to as NURBS), the surface case is trickier. Tensor product B-spline surfaces go back to de Boor [1]; for a comprehensive treatment see [3] or [4]. The problem with these kinds of surfaces is their rigidity: they are well-suited to handle geometry which has an underlying rectangular structure; arbitrary shapes may only be modeled by pasting together several surfaces and employing the concept of trimmed surfaces.

Several attempts have been made to overcome these restrictions of tensor product surfaces. These include simplex splines [2], triangular patches [6] and spline-like collections thereof [10], B-patches [11], S-patches [5], to cite a few. None can compete with the elegance of the original B-spline approach. This paper is an attempt to develop B-spline-like surfaces that inherit most of the desirable properties of B-spline curves yet are not restricted in their geometry for the surface case.

The de Boor algorithm for B-spline curves\(^1\) is based on repeated piecewise linear interpolation. The domain of a B-spline curve is the real line; its range is 2D or 3D Euclidean space. Linear interpolation is the process of repeating domain relationships for range values. In that sense, each step of the de Boor algorithm may be viewed as lifting a relationship between the domain elements knots and Greville abscissae\(^2\) to corresponding relationships between control points.

\(^1\)also referred to as the de Boor, Cox, Mansfield algorithm.
\(^2\)also referred to as knot averages.
In the context of surfaces, we propose here to replace the concept of piecewise linear interpolation by that of Sibson interpolation, the natural extension of piecewise linear interpolation to higher dimensions. This leads to a new class of curves/surfaces/volumes. This paper concentrates on the “quadratic” surface case.

2 Basics

We briefly introduce the concept of a Voronoi diagram, also known as a Dirichlet tessellation. Consider a collection of sites \( u_i \) in the plane. We associate with each site \( u_i \) a tile \( U_i \) consisting of all points \( u \) that are closer to \( u_i \) than to any other site \( u_k \). We denote the tile vertices by \( u_{ij} \). The collection \( U \) of all tiles is called the Voronoi diagram of the given set of sites. Two sites are called neighbors if their tiles share a common edge. All tiles are convex; if a tile \( U_i \) is infinite, its site is on the boundary of the convex hull of the \( u_i \).

The following recursive construction of the Voronoi diagram is due to R. Sibson [12]: suppose that we already constructed the Voronoi diagram for a set of sites, and we now want to add one more point \( v \). First, we determine which of the previously constructed tiles is occupied by \( v \); referring to Figure 1, let us assume it is \( U_k \). We now draw all perpendicular bisectors between \( v \) and its neighbors, thus forming the tile \( V^1 \) having vertices \( v_{j}^1 \).

Figure 1: Voronoi diagrams: a new point is inserted into an existing tessellation (solid); its subtile is shown with heavy lineweight.
The tile $V$ is formed by cutting out parts of $v$'s neighboring tiles. In Figure 1, let the gray point denote a site $u_i$ and let $\lambda_i(v)$ be the area cut of $V$; let $|V|$ be the area of $V$. Then we can write $v$ as a barycentric combination of its neighbors (note that $\sum \lambda_i = |V|$):

$$v = \sum_i \frac{\lambda_i}{|V|} u_i. \quad (1)$$

This identity is due to R. Sibson [12]; an alternate proof is due to B. Piper [9].

The function $\lambda_i(v)$ is $C^1$ except at the sites $u_k$, and its gradient is given by

$$\nabla \lambda_i(v) = \frac{e_i}{d_i} |m_i - v|$$

where $e_i$ denotes the length of the edge between $V$ and $U_i$, $m_i$ is the midpoint of that edge, and $d_i$ denotes the distance between $v$ and $p_i$. This result is also due to B. Piper [9]. Each $\lambda_i$ is a bivariate rational function with a degree two denominator and a degree four numerator.

If $z$-values $s_i$ are assigned at each $u_i$, then a $z$-value corresponding to $v$ may be computed as

$$s(v) = \sum_i \frac{\lambda_i}{|V|} z_i. \quad (2)$$

We denote by $S^1_U$ the space of all Sibson interpolants over a given set $U$.

This is known as nearest neighbor or Sibson interpolation. It has local support: The support region of a site $u_i$ is the union of all circular discs $D_{ijk}$ through three sites $u_i, u_j, u_k$ where $u_j$ and $u_k$ are neighbors of $u_i$. This is illustrated in Figure 2.

The interpolant $s(v)$ is $C^1$ except at the sites $u_i$, where it is $C^0$. At the perimeters of each $D_{ijk}$, there is a $C^2$ discontinuity.

Because of (1), the interpolant has linear precision. In the 1D case (the $u_i$ are elements of the real line), it becomes piecewise linear interpolation. In the 2D case, it is piecewise linear on the boundary of the convex hull of $U^0$.

Sibson’s interpolant (2D) is not idempotent in the following sense: if a point $v$ is inserted into the set of sites, and its function value $s(v)$ is used as its $z$-value, then the resulting interpolant will not be the same as the original one. This follows since the locations of $C^2$ discontinuities (the perimeters of the $D_{ijk}$) are changed.

For more detailed discussion of the interpolant, see [7] and [9].

---

3 these discs are called “covering circles” by L. Traversoni [7].
3 Repeated Sibson interpolation

Let a set $U^0$ of 2D points $u_i$ be given which we will denote as sites. Generate its Voronoi diagram with vertices $U^1_i$ forming a set $U^1$ and continue until we have a set of vertices $U^{n-1}$ for some integer $n$. The integer $n$ is called depth of the scheme, and $U^{n-1}$ is referred to as control grid. Assign a $z-$value to each element of $U^{n-1}$, thus obtaining a 3D control mesh $M^0 = (U^{n-1}, Z^0)$. This mesh will generate a surface in much the same way as a B-spline control polygon generates a curve.

We now describe how to evaluate that surface at a point $v$. We first set $V^0 = U^0 \cup \{v\}$. Generate its Voronoi diagram with vertices forming a set $V^1$ and continue until you have a set of vertices $V^{n-1}$. We will only consider those vertices of $V^{n-1}$ which depend on $v$.

The vertices of $V^{n-1}$ may be expressed in terms of those of $U^{n-1}$:

$$v_i^{n-1} = \sum_j a_{i,j}^{n-1} u_j^{n-1}.$$  

Using the coefficients in that expression, we evaluate the control mesh $M^0$ over the grid $V^{n-1}$.
\[ z_i^1(v) = \sum_j \sigma_{i,j}^0(v) z_j^0. \]

We thus obtain a mesh \( M^1 = (V^{n-1}, Z^1). \)

We know that \( V^{n-2} \) may be expressed in terms of \( V^{n-1} \). Using the coefficients in that expression, we compute \( Z^2 \) from \( Z^1 \). We continue this process, generating meshes \( V^r, Z^{n-r} \) with

\[ z_{i}^{r+1} = \sum_j \sigma_{i,j}^r z_j^r. \]

We continue until we obtain a mesh \( V^0, Z^n \). It contains only one point which depends on \( v \), namely \((v, z^n(v))\). This is the desired point on the surface. The set of all surfaces of the form \( z^n(v) \) is denoted by \( S^n_0 \).

The remainder of this paper explores the case \( n = 2 \), referred to as the “quadratic case” since two levels of Sibson interpolation are used.

### 4 The quadratic curve case

The recursive construction from the previous section allows to construct surfaces defined over point sets in the plane. However, the method is not restricted to a 2D domain; in fact, it is defined for domains of arbitrary dimensions. We now show that the above process yields quadratic B-spline curves for 1D data sets.

For 1D abscissae \( u_i \), Sibson’s interpolant becomes piecewise linear interpolation. Consider a knot sequence

\[ U^0 = \{ \ldots, u_{-1}, u_0, u_1, u_2, \ldots \}. \]

While there may be more elements in the knot sequence, we will concentrate on \( u_{-1}, \ldots, u_2 \). Generate \( U^0 \)'s Voronoi diagram with vertices forming a set \( U^1 \). For the elements \( u_i^1 \) of \( U^1 \), we have

\[ u_i^1 = \frac{1}{2} u_i + \frac{1}{2} u_{i+1}. \]

Assign a \( y \)-value \( d_i^0 \), short: \( d_i \), to each element in \( U^1 \), thus obtaining a polygon with vertices \((u_i^1, d_i)\). This corresponds to the mesh \( M^0 \) in Section 3. The set \( U^1 \) corresponds to the Greville abscissae of quadratic B-splines.

Let \( v \) be a point \( v \in [u_0, u_1] \). Set \( V^0 = \{ u_{-1}, u_0, v, u_1, u_2 \} \). Then

\[ V^1 = \]
\[ [v_{-1}, v_0^1), v_1^1, v_2^1] \]

We note that only \( v_0^1 \) and \( v_1^1 \) depend on \( v \), and we will concentrate on them from now on. Write each element of \( V^1 \) as a linear combination of the elements of \( U^1 \):

\[
\begin{align*}
v_0^1 &= \frac{u_1 - v}{u_1 - u_{-1}} u_{-1} + \frac{v - u_{-1}}{u_1 - u_{-1}} u_0^1, \\
v_1^1 &= \frac{u_2 - v}{u_2 - u_0} u_0^1 + \frac{v - u_0}{u_2 - u_0} u_1^1.
\end{align*}
\]

We have concentrated only on those \( v_i^1 \) which actually depend on \( v \). Using the same linear combinations, assign nearest neighbor \( y \)-values to the \( v_i^1 \), calling them \( d_0^1 \) and \( d_1^1 \):

\[
\begin{align*}
d_0^1 &= \frac{u_1 - v}{u_1 - u_{-1}} d_{-1} + \frac{v - u_{-1}}{u_1 - u_{-1}} d_0, \\
d_1^1 &= \frac{u_2 - v}{u_2 - u_0} d_0 + \frac{v - u_0}{u_2 - u_0} d_2.
\end{align*}
\]

Next, write \( v_0^0 \) as a linear combination of the elements \( v_0^1 \) and \( v_1^1 \) of \( V^1 \):

\[
v_0^0 = \frac{v - u_0}{u_1 - u_0} v_0^1 + \frac{u_1 - v}{u_1 - u_0} v_1^1
\]

and then write \( d_2^0 \) as the same combination in terms of \( d_0^1 \) and \( d_1^1 \):

\[
d_2^0 = \frac{v - u_0}{u_1 - u_0} d_0^1 + \frac{u_1 - v}{u_1 - u_0} d_1^1.
\]

This is the de Boor algorithm for quadratic B-spline curves, see [4] or [8] – thus the proposed scheme produces B-spline curves for the quadratic 1D case. Fig. 3 illustrates.

### 5 The quadratic surface case

Returning to 2D domains, we again consider the special case \( n = 2 \). We are given a set \( U^0 \) of 2D sites \( u_i \) and generate their Voronoi vertices \( u_i^1 \), collected in a set
U^1. We assign a \( z \)-value \( z_i \) to each \( u_1^i \), thus obtaining a mesh \( M^0 = (u_1^i, z_i) \). This forms the control mesh of our surface. We are now going to evaluate it at a point \( v \). We first find the Voronoi vertices \( v_1^i \) of \( V^0 = U^0 \cup \{v\} \). These vertices form the set \( V^1 \). For the purpose of computation, we only consider those elements of \( V^0 \) and \( V^1 \) which actually depend on \( v \) – see Figure 4 for an illustration.

Each \( v_1^i \) may be expressed uniquely (using Sibson’s identity) as a combination of nearby points \( u_1^j \). For example, assume \( v_1^i = \sum \sigma_{i,j} u_1^j \). Then \( z_1^i = \sum \sigma_{i,j} z_j \). In this manner, we compute a value \( z_1^i \) for each \( v_1^i \).

The point \( v \) may be expressed uniquely (again using Sibson’s identity) as a combination of nearby \( v_1^i \), say \( v = \sum \sigma_j v_1^j \). Then \( z^2(v) = \sum \sigma_j d_1^j \) is the desired point on the surface. Fig. 4 illustrates.

5.1 Support

Clearly our quadratic surfaces have a local support property in the sense that if one of the \( z_i \) has value 1 and all others have value 0, then the surface will only be nonzero near \( u_1^i \). From now on, we will refer to this site as \( u_1^i \). In order to determine the extent of this support region, let \( v \) be a point in it. At the first level in the evaluation process, the Sibson interpolant to the \( u_1^i \) is evaluated at the tile vertices \( v_1^j \) of \( v \). Hence at least one of these vertices must be in the support region of \( u_1^i \). This region is illustrated in Figure 6. Since the \( v_1^j \) are located on control polygon edges containing \( u_1^i \), they must be located on the edges (or edge segments) shown as thick lines in Figure 6.

When we insert \( v \) into \( U \), we form bisectors between \( v \) and neighboring \( u_i \). At least one of these bisectors must intersect one of the bold line segments as shown in Figure 6.
6. Hence \( \mathbf{v} \) must be inside one of the circular discs shown in Figure 4. The light colored discs are those from Figure 6; the dark colored ones are circular discs having as centers endpoints of the bold line segments and passing through neighboring \( \mathbf{u}_i \).

5.2 Domain

A related question is that of a surface’s domain, meaning the set of all points over which the surface may actually be evaluated. Recall that Sibson’s interpolant is only defined inside the convex hull of \( \mathbf{U} \). After we insert \( \mathbf{v} \), the points \( \mathbf{v}_1 \) are obtained by Sibson interpolation to the elements of \( \mathbf{V}^1 \). For this to work, the \( \mathbf{v}_1 \) must be computable, or

\[
\mathbf{V}^1 \subset \text{CH}[\mathbf{U}^1]
\]

where CH denotes the convex hull of a points set. Figure 8 illustrates: the \( \mathbf{u}_i \) are shown as solid circles; the \( \mathbf{u}_1 \) are shown as squares. The possible locations for \( \mathbf{v} \) are restricted to the shaded area.

An example surface is shown in Fig. 9.

We may associate a basis function \( S^2_i(\mathbf{v}) \) with each \( \mathbf{u}_1 \). It is obtained by setting \( z_i = 1 \) and all other \( z_i = 0 \). Hence \( z^2 \) may be expressed as

\[
z^2(\mathbf{v}) = \sum_i z_i S^2_i(\mathbf{v}).
\]
Figure 5: The influence region of a vertex based upon Sibson’s interpolant. The shaded circles pass through the vertex $u_1$ (solid square) and neighboring $u_i$.

## 5.3 Properties

We list several properties our quadratic surfaces:

- Linear precision: if $z_i = l(u_i)$ and $l$ is a bivariate linear function, then $z^2(v) = l(v)$. This follows since Sibson’s interpolant has that property.
- Convex hull property. Again, Sibson’s interpolant has that property.
- Local support (see above).
- Continuity: $z^2(v)$ is $C^2$ except at the $u_i$ where it is $C^1$. (conjectured)
- No knot insertion: if a point $v$ is inserted into $U$ and a refined control mesh is formed, it will not describe the same surface as the original. Reason: Sibson’s interpolant does not have the idempotence property.
- Quadratic inclusion: if all $z_i$ are sampled from a bivariate quadratic, then the resulting surface is a (different) quadratic. Stated differently: if $\mathcal{P}^2$ denotes the set of all bivariate quadratic polynomials, then

\[ \mathcal{P}^2 \subset \mathcal{S}^2. \]
We have experimental evidence for this, but no proof.

6 Conclusions and Future work

We have presented a new class of surfaces which enjoy many of the classic B-spline properties but are not restricted to regular structures. While the proposed method appears promising, many questions are still unresolved:

- Infinite Voronoi vertices: the elements of $U^1$ may be at (or near) infinity. Does this pose a problem?
- Discontinuities: for B-spline curves, continuity may be controlled by the use of multiple knots. Is there a similar concept here?
- Derivatives: we conjecture that the gradient of $z^2(v)$ equals that of $z^1_1(v)$.
Figure 7: The support of a control polygon vertex.

References


Figure 8: The domain (gray) of a quadratic surface.


Figure 9: An example surface.