MODELING SCATTERED FUNCTION DATA ON CURVED SURFACES*

Chandrajit L. Bajaj

Department of Computer Science,
Purdue University,
West Lafayette, IN 47907
email: bajaj@cs.purdue.edu

and

Guoliang Xu†

Computing Center,
Chinese Academy of Sciences,
Beijing, 100080
email: xuguo@cs.purdue.edu

Abstract

We present efficient algorithms to model a collection of scattered function data defined on a given smooth domain surface D in three dimensional real space ($\mathbb{R}^3$), by a C^1 cubic or a C^2 quintic piecewise trivariate polynomial approximation F (a mapping from D into $\mathbb{R}^4$). The smooth polynomial pieces or finite elements of F are defined on a three dimensional triangulation called the simplicial hull and defined over the domain surface D. Our smooth polynomial approximations allows one to additionally control the local geometry of the modeled function F. We also present two different techniques for visualizing the graph of the function F.

1 Introduction

In this paper, we consider the following problem: Given an arbitrary collection of points $P = \{(x_i, y_i, z_i, F_i) \epsilon \mathbb{R}^4\}_{i=1}^M$ with $(x_i, y_i, z_i) \epsilon \mathbb{R}^3$ on a given smooth surface D, called the *do-*

*Supported in part by NSF grants CCR 92-22467, DMS 91-01424, AFOSR grants F49620-93-10138, F49620-94-1-0080, ONR grant N00014-94-1-0370 and NASA grant NAG-93-1-1473.

†While visiting the Computer Science Department, Purdue University, West Lafayette, Indiana, USA

main surface, construct a C^1/C^2("/" stands for "or") piecewise smooth function F, known as the *function-on-surface*, such that $F(x_i, y_i, z_i) = F_i, \quad i = 1, \cdots, M$. Also visualize the graph of the function-on-surface F.

The problem of modeling and visualizing functions sampled on physical objects arises in several application areas: characterizing the rain fall on the earth, the pressure on the wing of an airplane and the temperature on a human body. A number of methods have been developed for dealing with this problem (for surveys see [3], [7]). Currently known approaches for approximating function-on-surface data however possess restrictions either on the domain surfaces or the function-on-surfaces. The domain surfaces are usually assumed to be spherical, convex or genus zero. The function-on-surface are not always polynomial [4], [8] or rather higher order polynomial [9] or a large number of pieces [1] compared to the approach of this paper. The method of [1] is a C^1 Clough-Tocher scheme that splits a tetrahedron into 4 subtetrahedra, uses degree 5 polynomials and requires C^2 data on the vertices of each subtetrahedron. Another Clough-Tocher scheme[10] requires only C^1 data at the vertices, for again constructing a C^1 function which is a cubic polynomial over each subtetrahedron, however splits the original tetrahedron into 12 pieces. A C^1 scheme [9] that does not split each tetrahedron uses degree 9 polynomials and requires C^4 data at the vertices. In extending the method of [9] to a C^2 scheme, requires degree 17 polynomials and C^8 data at the vertices of each tetrahedron. Compared to these approaches, our C^1/C^2 construction has no splitting and uses much lower degree polynomials (cubic/quintic) requiring only C^1/C^2 data respectively, at the vertices of each tetrahedron.

Our solution to the modeling problem involves the following steps: **(a)**. Construct a planar triangular approximation T of the domain surface D in the region of the points (x_i, y_i, z_i) on D. **(b)**. Generate C^1/C^2 data at the vertices of the triangulation T for a desired C^1/C^2 smooth approximation, respectively. **(c)**. Construct a simplicial hull (defined below) $\sum$ surrounding the triangulation T. **(d)**. Build the C^1/C^2 function-on-surface F over $\sum$ by locally interpolating the C^1/C^2 data, respectively. **(e)**. Visualize the graph of the function-on-surface F. We shall not address the first two steps **(a)** and **(b)** in this paper. A algorithm for the construction of the triangulation T of the given surface is given in [5]. See also Figure 1.1. However, we require our triangulation to satisfy certain conditions which will be discussed in §3. The problem of estimating the C^1/C^2 data at the vertices of T is studied in a separate paper[2]. In this paper, we detail the steps **(c)**, **(d)** and **(e)** in §3, §4, and §5 respectively, after the notation and preliminary section §2.

2 Notation and Preliminary Details

Bernstein-Bezier (BB) Form: Let $p_1,\ p_2,\ p_3,\ p_4 \in I\!R^3$ be affine independent. Then the tetrahedron with vertices $p_1,\ p_2,\ p_3$, and p_4 is the convex hull defined by $[p_1p_2p_3p_4] = \{p \in I\!R^3 : p = \sum_{i=1}^4 \alpha_i p_i, \alpha_i \geq 0, \sum_{i=1}^4 \alpha_i = 1\}$. For any $p = \sum_{i=1}^4 \alpha_i p_i \in [p_1p_2p_3p_4]$, $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)^T$ denotes the barycentric coordinates of p. Any polynomial $f(p)$ of degree n can be expressed as Bernstein-Bezier(BB) form over $[p_1p_2p_3p_4]$ as $f(p) = \sum_{|\lambda|=n} b_\lambda\ B^n_\lambda(\alpha), \quad \lambda \in \mathcal{Z}^4_+$, where $B^n_\lambda(\alpha) = \frac{n!}{\lambda_1!\lambda_2!\lambda_3!\lambda_4!}\ \alpha_1^{\lambda_1}\alpha_2^{\lambda_2}\alpha_3^{\lambda_3}\alpha_4^{\lambda_4}$ is Bernstein polynomial, $|\lambda| = \sum_{i=1}^4 \lambda_i$ with $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)^T =$

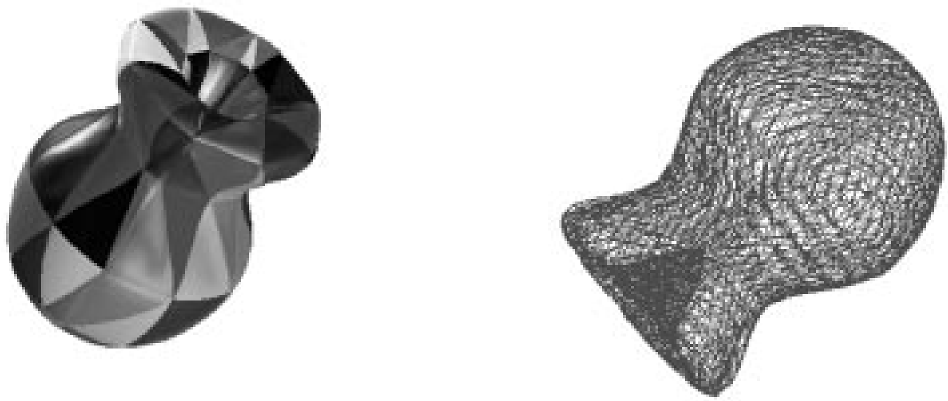

Figure 1.1: A piecewise smooth domain surface D_1 and a triangulation on it.

$\sum_{i=1}^4 \lambda_i e_i$, $b_\lambda = b_{\lambda_1\lambda_2\lambda_3\lambda_4}$(as a subscript, we simply write λ as $\lambda_1\lambda_2\lambda_3\lambda_4$) are called control points or weights, and $\mathcal{Z}_+^4$ stands for the set of all four dimensional vectors with nonnegative integer components. The following basic facts about the BB form will be used in this paper.

Lemma 2.1. *Let $f(p) = F(\alpha) = \sum_{|\lambda|=n} b_\lambda B_\lambda^n(\alpha)$ where α denotes the barycentric coordinates of p. Then for any pair of points $p^{(1)}$ and $p^{(2)}$, with $\alpha^{(1)}$ and $\alpha^{(2)}$ as their barycentric coordinates, we have*

$$\nabla f(p)^T(p^{(1)} - p^{(2)}) = n \sum_{|\lambda|=n-1} b_\lambda^1(\alpha^{(1)} - \alpha^{(2)}) B_\lambda^{n-1}(\alpha)$$

$$(p^{(1)} - p^{(2)})^T \nabla^2 f(p)(p^{(1)} - p^{(2)}) = n(n-1) \sum_{|\lambda|=n-2} b_\lambda^2(\alpha^{(1)} - \alpha^{(2)}) B_\lambda^{n-2}(\alpha)$$

where $\nabla f(p) = [\frac{\partial f(p)}{\partial x} \; \frac{\partial f(p)}{\partial y} \; \frac{\partial f(p)}{\partial z}]^T$, $\quad \nabla^2 f(p) = [\nabla \frac{\partial f(p)}{\partial x}, \; \nabla \frac{\partial f(p)}{\partial y} \; \nabla \frac{\partial f(p)}{\partial z}]$ *and* $b_\lambda^r(\alpha^{(1)} - \alpha^{(2)}) = \sum_{|j|=r} b_{\lambda+j} \; B_j^r(\alpha^{(1)} - \alpha^{(2)})$

See [6] for the two dimensional case of the above lemma. From this lemma we have

Corollary 2.2. *Let $f(p) = \sum_{|\lambda|=n} b_\lambda B_\lambda^n(\alpha)$ be defined on the tetrahedron $[p_1p_2p_3p_4]$, then*

$$b_{(n-1)e_i+e_j} = b_{ne_i} + \frac{1}{n}(p_j - p_i)^T \nabla f(p_i), \; j \neq i \tag{2.1}$$

$$\begin{aligned} b_{(n-2)e_i+e_j+e_k} &= -b_{ne_i} + b_{(n-1)e_i+e_j} + b_{(n-1)e_i+e_k} \\ &+ \tfrac{1}{n(n-1)}(p_j - p_i)^T \nabla^2 f(p_i)(p_k - p_i), \quad j \neq i, k \neq i \end{aligned} \tag{2.2}$$

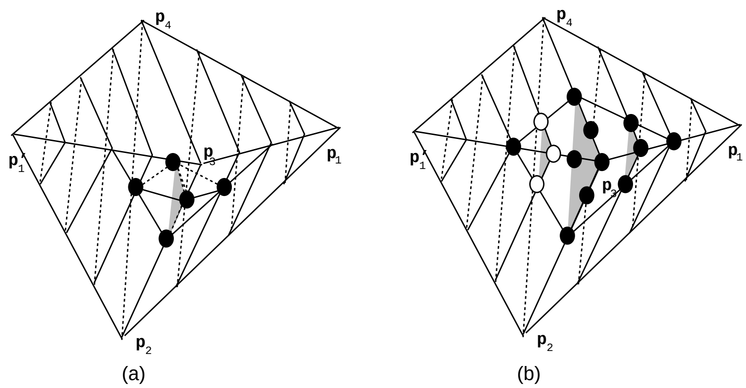

Figure 2.1: The related control points of C^1 (a) and C^2 (b) conditions

The corollary tell us that the weights around a vertex can be computed from the given C^2 data.

Lemma 2.3 ([6]). *Let* $f(p) = \sum_{|\lambda|=n} a_\lambda B^n_\lambda(\alpha)$ *and* $g(p) = \sum_{|\lambda|=n} b_\lambda B^n_\lambda(\alpha)$ *be two polynomials defined on two tetrahedra* $[p_1p_2p_3p_4]$ *and* $[p_1'p_2p_3p_4]$*, respectively. Then*
(i) f *and* g *are* C^0 *continuous at the common face* $[p_2p_3p_4]$ *if and only if*

$$a_\lambda = b_\lambda, \quad for \quad any \quad \lambda = 0\lambda_2\lambda_3\lambda_4, \quad |\lambda| = n \tag{2.3}$$

(ii) f *and* g *are* C^1 *continuous at the common face* $[p_2p_3p_4]$ *if and only if (2.3) holds and*

$$b_{1\lambda_2\lambda_3\lambda_4} = \beta_1 a_{1\lambda_2\lambda_3\lambda_4} + \beta_2 a_{0\lambda_2\lambda_3\lambda_4+0100} + \beta_3 a_{0\lambda_2\lambda_3\lambda_4+0010} + \beta_4 a_{0\lambda_2\lambda_3\lambda_4+0001} \tag{2.4}$$

(iii) f *and* g *are* C^2 *continuous at the common face* $[p_2p_3p_4]$ *if and only if (2.3)-(2.4) holds and*

$$\begin{aligned} b_{2\lambda_2\lambda_3\lambda_4} &= \beta_1^2 a_{2\lambda_2\lambda_3\lambda_4} + 2\beta_1\beta_2 a_{0\lambda_2\lambda_3\lambda_4+1100} + 2\beta_1\beta_3 a_{0\lambda_2\lambda_3\lambda_4+1010} + 2\beta_1\beta_4 a_{0\lambda_2\lambda_3\lambda_4+1001} \\ &+ \beta_2^2 a_{0\lambda_2\lambda_3\lambda_4+0200} + 2\beta_2\beta_3 a_{0\lambda_2\lambda_3\lambda_4+0110} + 2\beta_2\beta_4 a_{0\lambda_2\lambda_3\lambda_4+0101} \\ &+ \beta_3^2 a_{0\lambda_2\lambda_3\lambda_4+0020} + 2\beta_3\beta_4 a_{0\lambda_2\lambda_3\lambda_4+0011} + \beta_4^2 a_{0\lambda_2\lambda_3\lambda_4+0002} \end{aligned} \tag{2.5}$$

where $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)^T$ *are defined by the relation* $p_1' = \beta_1 p_1 + \beta_2 p_2 + \beta_3 p_3 + \beta_4 p_4, \quad |\beta| = 1.$

In Lemma 2.3, if we divide (2.4) and (2.5) by β_4^2, then the C^1 and C^2 conditions become

$$a_{0\lambda_2\lambda_3\lambda_4+0001} = \mu_1 a_{1\lambda_2\lambda_3\lambda_4} + \mu_2 b_{1\lambda_2\lambda_3\lambda_4} + \mu_3 a_{0\lambda_2\lambda_3\lambda_4+0100} + \mu_4 a_{0\lambda_2\lambda_3\lambda_4+0010} \tag{2.6}$$

$$\begin{aligned} &\mu_1(\mu_1 a_{2\lambda_2\lambda_3\lambda_4} + \mu_3 a_{0\lambda_2\lambda_3\lambda_4+1100} + \mu_4 a_{0\lambda_2\lambda_3\lambda_4+1010} - a_{0\lambda_2\lambda_3\lambda_4+1001}) \\ &= \mu_2(\mu_2 b_{2\lambda_2\lambda_3\lambda_4} + \mu_3 b_{0\lambda_2\lambda_3\lambda_4+1100} + \mu_4 b_{0\lambda_2\lambda_3\lambda_4+1010} - b_{0\lambda_2\lambda_3\lambda_4+1001}) \end{aligned} \tag{2.7}$$

respectively, where $\mu_1 = -\frac{\beta_1}{\beta_4}, \mu_2 = \frac{1}{\beta_4}, \mu_3 = -\frac{\beta_2}{\beta_4}, \mu_4 = -\frac{\beta_3}{\beta_4}$, that is $p_4 = \mu_1 p_1 + \mu_2 p_1' + \mu_3 p_2 + \mu_4 p_3$.

It is not difficult to show the following from Corollary 2.2 :

Lemma 2.4. *Let* $f(p)$ *and* $g(p)$ *be defined as Lemma 2.3. If the coefficients of* f *and* g *around the vertices are determined by (2.1)–(2.2), then the* C^1 *and* C^2 *conditions (2.4)–(2.5) related only to these coefficients are satisfied.*

Degree Elevation. The polynomial $f(p) = \sum_{|\lambda|=n} b_\lambda \; B^n_\lambda(\alpha)$ can be written as one of degree $n+1$ (see e.g. [6]). $f(p) = \sum_{|\lambda|=n+1} (Eb)_\lambda \; B^{n+1}_\lambda(\alpha), \quad \lambda \in \mathcal{Z}^4_+$ where $(Eb)_\lambda = \frac{1}{n+1}\sum_{i=1}^4 \lambda_i b_{\lambda-e_i}$. We shall use these formulas in approximating lower degree polynomials, in §4.

3 Simplicial Hull

Given a planar triangular approximation T of D containing (and not necessarily as vertices) the points (x_i, y_i, z_i) on D, a *simplicial hull* of D and T, denoted by $\sum$, is a collection of non-degenerate tetrahedra which satisfies the following:

(1) Each tetrahedron in $\sum$ has either a single edge of T(then it will be called an *edge tetrahedron*) or a single face of T(then it will be called a *face tetrahedron*).

(2) For each face f of T there are at most two face tetrahedra (above and below f) in $\sum$ that share the face f.

(3) Two face tetrahedra that share a common edge do not intersect in any other region. This condition is referred to in this paper as *non-self-intersection.*

(4) For each edge there are two pairs of common face sharing edge tetrahedra in $\sum$, such that each pair blends the two adjacent face tetrahedra on the same side.

(5) The surface D is contained in $\sum$. This condition is referred to in this paper as the *surface containment condition.*

Therefore, a simplicial hull of D and T is in a neighborhood surrounding D. It should be noted that, for the given triangulation T of D, there may exist infinitly many simplicial hulls or perhaps no simplicial hull may exist. However under the following conditions on T, we can always construct a simplicial hull.

Condition 1. *The triangulation T is locally even. That is for every face of T, say $[p_1p_2p_3]$, the angle between the surface normal n_i at the vertex p_i and the normal of the face $[p_1p_2p_3]$ is less than*

$$\tan^{-1}\left(\frac{2s\tan(\frac{1}{2}\min\{\alpha_1,\alpha_2,\alpha_3\})}{\| \, \|p_j - p_i\|(p_k - p_i) + \|p_k - p_i\|(p_j - p_i) \, \|}\right)$$

for $i = 1,2,3$ and distinct $1 \le i,j,k \le 3$. Here s is the area of the face $[p_1p_2p_3]$, and $\alpha_1, \alpha_2, \alpha_3$ are the dihedral angles of the three edges of the face $[p_1p_2p_3]$.

Condition 2. *The surface D is single sheeted on T. That is, for every face of T, say $[p_1p_2p_3]$ let L be a straight line that is perpendicular to the face f and passes through the center c of the inscribed circle of f. Let p_4 and q_4 be the center's nearest points on L off each side of f such that $\|p_4 - c\| = \|q_4 - c\|$ and the three tangent planes at the three vertices are contained in $[p_4p_1p_2p_3q_4]$. Then for any $p \in f$ the broken line $[p_4pq_4]$ intersects the surface D only once.*

Condition 3. *Any two adjacent faces are not coplanar.*

Since the given surface is curved and smooth, by adding additional points on D, we can modify the algorithm of [5] to achieve a T satisfying the above conditions.

For such a T we now show how to construct a simplicial hull $\sum$ in two easy steps.

1. Build Face Tetrahedra. For each face $f = [p_1p_2p_3]$ of T, let L be a straight line that is perpendicular to the face f and passes through the center c of the inscribed circle of f. Let p_4

and q_4 be the center's nearest points on L off each side of f such that $\|p_4 - c\| = \|q_4 - c\|$ and the three tangent planes at the three vertices are contained in $[p_4 p_1 p_2 p_3 q_4]$, then construct two face tetrahedra $[p_1 p_2 p_3 p_4]$ and $[p_1 p_2 p_3 q_4]$.

2. Build Edge Tetrahedra. Let $[p_2 p_3]$ be an edge of T and $[p_1 p_2 p_3]$ and $[p_1' p_2 p_3]$ be the two adjacent faces. Let $[p_1 p_2 p_3 p_4]$ and $[p_1 p_2 p_3 q_4]$, and $[p_1' p_2 p_3 p_4']$ and $[p_1' p_2 p_3 q_4']$ be the face tetrahedra built for the faces $[p_1 p_2 p_3]$ and $[p_1' p_2 p_3]$, respectively. Now two pairs of tetrahedra are constructed. The first pair $[p_1'' p_2 p_3 p_4]$ and $[p_1'' p_2 p_3 p_4']$ is between $[p_1' p_2 p_3 p_4']$ and $[p_1 p_2 p_3 p_4]$. The second pair $[q_1'' p_2 p_3 q_4]$ and $[q_1'' p_2 p_3 q_4']$ is between $[p_1' p_2 p_3 q_4']$ and $[p_1 p_2 p_3 q_4]$. Here $p_1'' \in (p_4 p_4')$ or above (p_4, p_4'), say $p_1'' = \frac{(1-t)}{2}(p_2 + p_3) + \frac{t}{2}(p_4' + p_4)$, $\quad t \geq 1$, so that p_1'' is above $[p_2, p_3]$ and the surface containment condition is satisfied. Similarly, $q_1'' \in (q_4 q_4')$ or below (q_4, q_4'), say $q_1'' = \frac{(1-t)}{2}(p_2 + p_3) + \frac{t}{2}(q_4' + q_4)$, $\quad t \geq 1$, so that q_1'' is below $[p_2, p_3]$ and the surface containment condition is satisfied.

The locally even condition guarantees that the face tetrahedron constructed has height(the distance between the top vertex p_4 or q_4 to the face) at most $r \tan(\frac{1}{2}\min\{\alpha_0, \alpha_1, \alpha_2\})$, where r is the radius of the inscribed circle. Hence the dihedral angles at the bottom edges of the tetrahedron are less than $\frac{1}{2}\min\{\alpha_0, \alpha_1, \alpha_2\}$. Therefore, there is no additional intersection between two adjacent face tetrahedra.

4 C^1/C^2 Interpolation by Cubic/Quintic

Suppose we have established a simplicial hull $\sum$ for the given triangulation T of D. Now we construct a C^1/C^2 function f over $\sum$ such that f has the given C^1/C^2 data, respectively at each vertex. Let $V_1 = [p_1 p_2 p_3 p_4]$, $\; V_2 = [p_1' p_2 p_3 p_4']$, $\; W_1 = [p_1'' p_2 p_3 p_4]$, $W_2 = [p_1'' p_2 p_3 p_4']$, $\; V_1' = [p_1 p_2 p_3 q_4]$, $\; V_2' = [p_1' p_2 p_3 q_4']$, $\; W_1' = [q_1'' p_2 p_3 q_4]$, $\; W_2' = [q_1'' p_2 p_3 q_4']$ and the cubic/quintic polynomials f_i over V_i, g_i over W_i, f_i' over V_i' and g_i' over W_i' be expressed in Bernstein-Bezier form with coefficients $a_\lambda^{(i)}$, $b_\lambda^{(i)}$, $c_\lambda^{(i)}$ and $d_\lambda^{(i)}$, respectively. Now we shall determine these coefficients step by step. Denote

$$\begin{aligned} p_1'' &= \beta_1^{(1)} p_1 + \beta_2^{(1)} p_2 + \beta_3^{(1)} p_3 + \beta_4^{(1)} p_4, \quad \beta_1^{(1)} + \beta_2^{(1)} + \beta_3^{(1)} + \beta_4^{(1)} = 1 \\ p_1'' &= \beta_1^{(2)} p_1' + \beta_2^{(2)} p_2 + \beta_3^{(2)} p_3 + \beta_4^{(2)} p_4', \quad \beta_1^{(2)} + \beta_2^{(2)} + \beta_3^{(2)} + \beta_4^{(2)} = 1 \\ p_1'' &= \mu_1 p_4 + \mu_2 p_4' + \mu_3 p_2 + \mu_4 p_3, \qquad \mu_1 + \mu_2 + \mu_3 + \mu_4 = 1 \end{aligned} \tag{4.1}$$

C^1 Cubic Scheme

(1) The number 0 weights(see Figure 4.1) are given by the function values at the vertices.

(2) The number 1 weights are determined by formula (2.1) from C^1 data.

(3) The number 2 weights, that is $a_{1110}^{(i)}$, are free.

(4) The number 3 weights are determined by C^1 conditions (2.4) and (2.6). More precisely,

$$a_{0111}^{(i)} = \theta_1^{(i)} a_{1110}^{(1)} + \theta_2^{(i)} a_{0210}^{(i)} + \theta_3^{(i)} a_{0120}^{(i)} + \theta_4^{(i)} a_{1110}^{(2)}, \quad i = 1, 2$$

where

$$\begin{aligned} p_4 &= \theta_1^{(1)} p_1 + \theta_2^{(1)} p_2 + \theta_3^{(1)} p_3 + \theta_4^{(1)} p_1', \qquad \theta_1^{(1)} + \theta_2^{(1)} + \theta_3^{(1)} + \theta_4^{(1)} = 1 \\ p_4' &= \theta_1^{(2)} p_1 + \theta_2^{(2)} p_2 + \theta_3^{(2)} p_3 + \theta_4^{(2)} p_1', \qquad \theta_1^{(2)} + \theta_2^{(2)} + \theta_3^{(2)} + \theta_4^{(2)} = 1 \end{aligned}$$

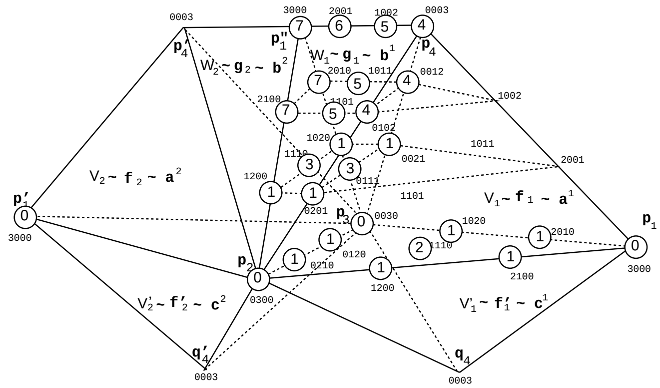

Figure 4.1: Adjacent Tetrahedra, Control Points of Cubic Functions

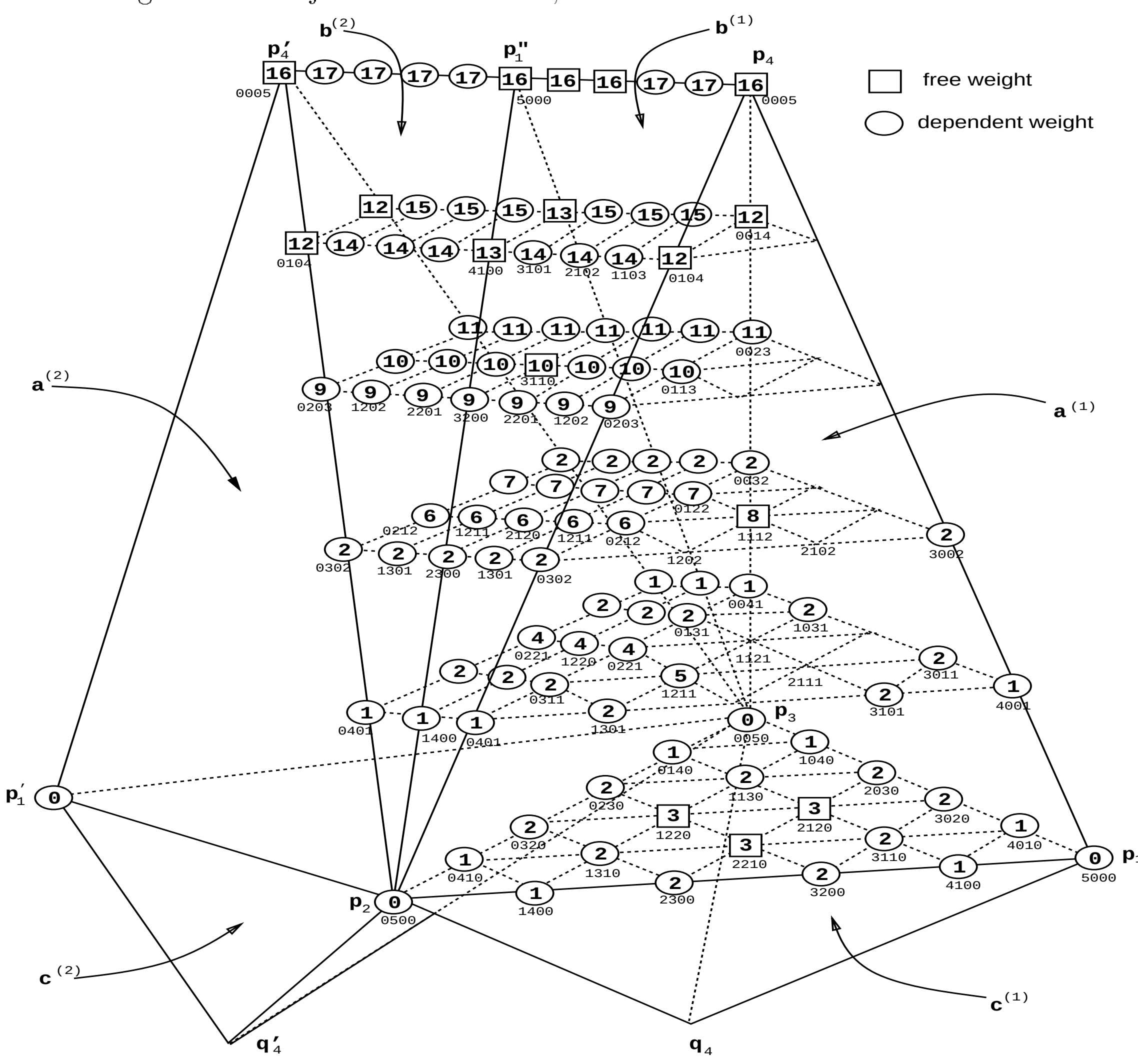

Figure 4.2: Adjacent Tetrahedra, Control Points of Quintic Functions

(5) The number 4 weights are free.
(6) The number 5 weights are determined by C^1 conditions (2.4).
(7) The number 6 weights are free.
(8) The number 7 weights are determined by C^1 conditions (2.6).

The remaining weights with index $\lambda_1\lambda_2\lambda_3\lambda_4$ are determined by C^1 condition (2.4) for $\lambda_4 \le 1$ and freely chosen for $\lambda_4 > 1$.

C^2 Quintic Scheme

(1) The number 0 weights(see Figure 4.2) are given by the function values at the vertices. For examples, $a^{(1)}_{5e_i} = f(p_i), \; i = 1, 2, 3.$

(2) The number 1 weights are determined by formula (2.1).

(3) The number 2 weights are determined by formula (2.2).

(4) The number 3 weights, that is $a^{(i)}_{1220}, a^{(i)}_{2210}$ and $a^{(i)}_{2120}$, are free.

(5) The number 4 weights are determined by C^1 conditions (2.4), that is

$$a^{(i)}_{0221} = \theta^{(i)}_1 a^{(1)}_{1220} + \theta^{(i)}_2 a^{(i)}_{0320} + \theta^{(i)}_3 a^{(i)}_{0230} + \theta^{(i)}_4 a^{(2)}_{1220}$$

$$b^{(1)}_{1220} = \mu_1 a^{(1)}_{0221} + \mu_2 a^{(2)}_{0221} + \mu_3 a^{(1)}_{0320} + \mu_4 a^{(1)}_{0230}$$

(6) The number 5 and 6 weights have to be determined simultaneously. In determining these weights, we need to consider all the C^1 and C^2 conditions related to the tetrahedra surrounding the vertex p_2. Suppose there are k triangles(hence k edges) around p_2, then by C^1 and C^2 conditions, we have $6k$ equations. That is, crossing each face, we have two equations. The number of related unknowns is also $6k$. That is, k number 5 weights and $5k$ number 6 weights. Now we investigate these equations. It follows from (2.4) and (2.5) that

$$b^{(i)}_{1211} = \beta^{(i)}_1 a^{(i)}_{1211} + \beta^{(i)}_2 a^{(i)}_{0311} + \beta^{(i)}_3 a^{(i)}_{0221} + \beta^{(i)}_4 a^{(i)}_{0212} \tag{4.2}$$

$$\begin{aligned} b^{(i)}_{2210} &= \beta^{(i)}_1\beta^{(i)}_1 a^{(i)}_{2210} + 2\beta^{(i)}_1\beta^{(i)}_2 a^{(i)}_{1310} + 2\beta^{(i)}_1\beta^{(i)}_3 a^{(i)}_{1220} + 2\beta^{(i)}_1\beta^{(i)}_4 a^{(i)}_{1211} + \beta^{(i)}_2\beta^{(i)}_2 a^{(i)}_{0410} \\ &+ 2\beta^{(i)}_2\beta^{(i)}_3 a^{(i)}_{0320} + 2\beta^{(i)}_2\beta^{(i)}_4 a^{(i)}_{0311} + \beta^{(i)}_3\beta^{(i)}_3 a^{(i)}_{0230} + 2\beta^{(i)}_3\beta^{(i)}_4 a^{(i)}_{0221} + \beta^{(i)}_4\beta^{(i)}_4 a^{(i)}_{0212} \end{aligned} \tag{4.3}$$

for $i = 1, 2$. (4.2) and (4.3) can be written briefly as

$$b^{(i)}_{1211} = \beta^{(i)}_1 a^{(i)}_{1211} + \beta^{(i)}_4 a^{(i)}_{0212} i + \gamma^{(i)}_0 \tag{4.4}$$

$$b^{(i)}_{2210} = 2\beta^{(i)}_1\beta^{(i)}_4 a^{(i)}_{1211} + \beta^{(i)}_4\beta^{(i)}_4 a^{(i)}_{0212} + \gamma^{(i)}_1 \tag{4.5}$$

where $\gamma^{(i)}_0$ and $\gamma^{(i)}_1$ are the known terms in (4.2) and (4.3). Since (see (2.6) and (2.7))

$$b^{(1)}_{2210} = \mu_1 b^{(1)}_{1211} + \mu_2 b^{(2)}_{1211} + \gamma_2 \tag{4.6}$$

$$\mu_1^2 b^{(1)}_{0212} - \mu_1 b^{(1)}_{1211} = \mu_2^2 b^{(2)}_{0212} - \mu_2 b^{(2)}_{1211} + \gamma_3 \tag{4.7}$$

where $\gamma_2 = \mu_3 b^{(i)}_{1310} + \mu_4 b^{(i)}_{1220}$ and $\gamma_3 = \mu_2(\mu_3 b^{(2)}_{0311} + \mu_4 b^{(2)}_{0221}) - \mu_1(\mu_3 b^{(1)}_{0311} + \mu_4 b^{(1)}_{0221})$, then by substituting (4.4) into (4.6) and (4.7) and then eliminating $b^{(i)}_{2210}$ from (4.5) and (4.6) we get three equations related to four unknowns which could be written as:

$$\begin{bmatrix} \beta_4^{(1)} - \mu_1 & -\mu_2 \\ -\mu_1 & \beta_4^{(2)} - \mu_2 \end{bmatrix} \begin{bmatrix} \beta_4^{(1)} & 0 \\ 0 & \beta_4^{(2)} \end{bmatrix} \begin{bmatrix} a^{(1)}_{0212} \\ a^{(2)}_{0212} \end{bmatrix}$$

$$= - \begin{bmatrix} 2\beta_4^{(1)} - \mu_1 & -\mu_2 \\ -\mu_1 & 2\beta_4^{(2)} - \mu_2 \end{bmatrix} \begin{bmatrix} \beta_1^{(1)} & 0 \\ 0 & \beta_1^{(2)} \end{bmatrix} \begin{bmatrix} a^{(1)}_{1211} \\ a^{(2)}_{1211} \end{bmatrix} + \begin{bmatrix} \gamma_4^{(1)} \\ \gamma_4^{(2)} \end{bmatrix} \tag{4.8}$$

$$\left[-\mu_1(\beta_4^{(1)} - \mu_1) \;\; \mu_2(\beta_4^{(2)} - \mu_2)\right] \begin{bmatrix} a^{(1)}_{0212} \\ a^{(2)}_{0212} \end{bmatrix} - \left[\mu_1 \beta_1^{(1)}, -\mu_2 \beta_1^{(2)}\right] \begin{bmatrix} a^{(1)}_{1211} \\ a^{(2)}_{1211} \end{bmatrix} = \gamma_5 \tag{4.9}$$

where $\gamma_4^{(1)} = \mu_1\gamma_0^{(1)} + \mu_2\gamma_0^{(2)} + \gamma_2 - \gamma_1^{(1)}$, $\gamma_4^{(2)} = \mu_1\gamma_0^{(1)} + \mu_2\gamma_0^{(2)} + \gamma_2 - \gamma_1^{(2)}$, and $\gamma_5 = \gamma_3 + \mu_1\gamma_0^{(1)} - \mu_2\gamma_0^{(2)}$. Since the coefficient matrix of (4.8) is nonsingular, by solving $[a^{(1)}_{0212}\ a^{(2)}_{0212}]^T$ from (4.8) and then substituting it into (4.9), we get one equation relating to the unknowns $a^{(1)}_{1211}$, $a^{(2)}_{1211}$. Let the equation be in the form

$$\phi_i a^{(1)}_{1211} + \psi_i a^{(2)}_{1211} = \omega_i \tag{4.10}$$

Then, these unknowns form a closed chain around the vertex p_2. The coefficient matrix of all these equations related to the vertex p_2 is in the form of

$$A = \begin{bmatrix} \phi_1 & \psi_1 & & \\ & \phi_2 & \psi_2 & \\ & & \ddots & \\ \psi_k & & & \phi_k \end{bmatrix}$$

The system (4.10) is a solvable in general with one degree of freedom. That is the rank of matrix A is $k-1$. Hence the system can be solved. However, if the surrounding tetrahedra at the same side at p_2 are not closed, the matrix A is in the form of $A = \begin{bmatrix} \phi_1 & \psi_1 & & \\ & \ddots & \ddots & \\ & & \phi_k & \psi_k \end{bmatrix}$ which can be changed to $A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$ if one of the unknowns, say the l-th is chosen to be a free parameter. Hence the system of equations can be decomposed into two sub-systems. Each of the sub-systems can be easily solved.

(7) The number 7 weights are similarly determined as that of number 6.

(8) The number 8 weight $a^{(i)}_{1112}$ are free.

(9) The number 9 weights are determined by the C^1 and C^2 conditions. Both the number of equations and the number of unknowns are $6k$. That is for $i = 1, 2$

$$b^{(i)}_{1202} = \beta_1^{(i)} a^{(i)}_{1202} + \beta_2^{(i)} a^{(i)}_{0302} + \beta_3^{(i)} a^{(i)}_{0212} + \beta_4^{(i)} a^{(i)}_{0203} \tag{4.11}$$

$$\begin{aligned} b^{(i)}_{2201} &= \beta^{(i)}_1\beta^{(i)}_1 a^{(i)}_{2201} + 2\beta^{(i)}_1\beta^{(i)}_2 a^{(i)}_{1301} + 2\beta^{(i)}_1\beta^{(i)}_3 a^{(i)}_{1211} + 2\beta^{(i)}_1\beta^{(i)}_4 a^{(i)}_{1202} + \beta^{(i)}_2\beta^{(i)}_2 a^{(i)}_{0401} \\ &+ 2\beta^{(i)}_2\beta^{(i)}_3 a^{(i)}_{0311} + 2\beta^{(i)}_2\beta^{(i)}_4 a^{(i)}_{0302} + \beta^{(i)}_3\beta^{(i)}_3 a^{(i)}_{0221} + 2\beta^{(i)}_3\beta^{(i)}_4 a^{(i)}_{0212} + \beta^{(i)}_4\beta^{(i)}_4 a^{(i)}_{0203} \end{aligned} \quad (4.12)$$

$$b^{(1)}_{3200} = \mu_1 b^{(1)}_{2201} + \mu_2 b^{(2)}_{2201} + \gamma_6 \quad (4.13)$$

$$\mu_1^2 b^{(1)}_{1202} - \mu_1 b^{(1)}_{2201} = \mu_2^2 b^{(2)}_{1202} - \mu_2 b^{(2)}_{2201} + \gamma_7 \quad (4.14)$$

where $\gamma_6 = \mu_3 b^{(i)}_{2300} + \mu_4 b^{(i)}_{2210}$ and $\gamma_7 = \mu_2(\mu_3 b^{(2)}_{1301} + \mu_4 b^{(2)}_{1211}) - \mu_1(\mu_3 b^{(1)}_{1301} + \mu_4 b^{(1)}_{1211})$. Substitute (4.11) and (4.12) into (4.14), so that we have

$$\mu_1\beta^{(1)}_4(\mu_1 - \beta^{(1)}_4) b^{(1)}_{0203} - \mu_2\beta^{(2)}_4(\mu_2 - \beta^{(2)}_4) b^{(2)}_{0203} = \cdots$$

This is a system that is in the same form as (4.10). The coefficient matrix of this system is nonsingular, in general.

(10) For the number 10 weights, we have six equations parallel to the equations (4.11)–(4.14) with all the indices changed by the rule:

The index of the number 10 weight = The index of the number 9 weight $-e_2 + e_3$

and seven independent weights. By chosing one of them, say $b^{(i)}_{3110}$, to be a free parameter, the system can be solved.

(11) The number 11 weights are determined in the same way as the number 9.

(12) The number 12 and 13 weights are free, while the number 14 are determined by C^1 and C^2 conditions. That is $b^{(i)}_{1103}$ are defined by (2.4). $b^{(i)}_{2102}$ are defined by (2.5). For $b^{(i)}_{3101}$, we have by (2.6) and (2.7) that

$$\mu_1 b^{(1)}_{3101} + \mu_2 b^{(2)}_{3101} = b^{(1)}_{4100} + \gamma_8, \quad -\mu_1 b^{(1)}_{3101} + \mu_2 b^{(2)}_{3101} = \mu_2^2 b^{(2)}_{2102} - \mu_1^2 b^{(2)}_{2102} + \gamma_9$$

where $\gamma_8 = -\mu_3 b^{(i)}_{3200} - \mu_4 b^{(i)}_{3110}$ and $\gamma_9 = \mu_2(\mu_3 b^{(2)}_{2201} + \mu_4 b^{(2)}_{2111}) - \mu_1(\mu_3 b^{(1)}_{2201} + \mu_4 b^{(1)}_{2111})$.

$$b^{(1)}_{3101} = \frac{b^{(1)}_{4100} - \mu_2^2 b^{(2)}_{2102} + \mu_1^2 b^{(2)}_{2102} + \gamma_8 - \gamma_9}{2\mu_1}, \quad b^{(2)}_{3101} = \frac{b^{(1)}_{4100} + \mu_2^2 b^{(2)}_{2102} - \mu_1^2 b^{(2)}_{2102} + \gamma_8 + \gamma_9}{2\mu_1}$$

(13) The number 15 weights are similar to that of number 14, the index being changed by the same rule as above.

(14) The number 16 weights are free, the number 17's are determined by C^1 and C^2 conditions.

(15) The number 0 to number 8 weights of the lower tetrahedra, below faces of T (see Figure 4.2) are determined by C^0, C^1 and C^2 conditions (2.3), (2.4) and (2.5) from weights in the upper tetrahedron.

16 The number 9 to 17 weights of the lower tetrahedra are determined in a fashion similar to the C^0, C^1 and C^2 conditions between the face and edge tetrahedra.

In summary, the construction steps **1–14** and **16** is according to the C^0, C^1 and C^2 conditions across the common faces between face and edge tetrahedra that are both above or both below the original triangulation T. Step **15** is according to the C^0, C^1 and C^2 conditions across the faces of T and between the upper and lower tetrahedra. Therefore, the composite function is global C^2 continuous in $\sum$.

The Use of Free Weights

In both of the C^1 and C^2 schemes described above, there are some free weights which can be freely determined to control the local geometry of F without affecting the continuity. We suggest three approaches or their combinations to achieve this local control. The first is to modify the shape of F by interactively adjusting the free weights. The second is to locally interpolate some of the function-on-surface data earlier approximated by the polynomial in each tetrahedron. The third approach is to least-square approximate some additional lower degree polynomial (acting as a controlling function) by use of the degree elevation formula of §2. For example, in the C^1 scheme, the number 2 weights can be determined by

$$a_{1110}^{(i)} = \frac{1}{4}(a_{1200}^{(i)} + a_{2100}^{(i)} + a_{2010}^{(i)} + a_{1020}^{(i)} + a_{0210}^{(i)} + a_{0120}^{(i)}) - \frac{1}{6}(a_{3000}^{(i)} + a_{0300}^{(i)} + a_{0030}^{(i)})$$

and the number 4 weights are determined by

$$a_{0003}^{(i)} = \frac{1}{3}[2(q_{0101}^{(i)} + q_{1001}^{(i)} + q_{0011}^{(i)}) - (a_{0300}^{(i)} + a_{3000}^{(i)} + a_{0030}^{(i)})]$$

$$a_{0102}^{(i)} = \frac{1}{3}(2q_{0101}^{(i)} + a_{0003}^{(i)}), \quad a_{1002}^{(i)} = \frac{1}{3}(2q_{1001}^{(i)} + a_{0003}^{(i)}), \quad a_{0012}^{(i)} = \frac{1}{3}(2q_{0011}^{(i)} + a_{0003}^{(i)})$$

where

$$q_{0101}^{(i)} = \frac{3}{4}(a_{1101}^{(i)} - a_{1011}^{(i)} + a_{0111}^{(i)} + a_{0201}^{(i)}) - \frac{1}{4}(q_{1100}^{(i)} - q_{1010}^{(i)} + q_{0110}^{(i)} + a_{0300}^{(i)})$$

$$q_{1001}^{(i)} = \frac{3}{4}(a_{1101}^{(i)} + a_{1011}^{(i)} - a_{0111}^{(i)} + a_{2001}^{(i)}) - \frac{1}{4}(q_{1100}^{(i)} + q_{1010}^{(i)} - q_{0110}^{(i)} + a_{3000}^{(i)})$$

$$q_{0011}^{(i)} = \frac{3}{4}(-a_{1101}^{(i)} + a_{1011}^{(i)} + a_{0111}^{(i)} + a_{0021}^{(i)}) - \frac{1}{4}(-q_{1100}^{(i)} + q_{1010}^{(i)} + q_{0110}^{(i)} + a_{0030}^{(i)})$$

$$q_{1100}^{(i)} = \frac{1}{4}(3a_{1200}^{(i)} + 3a_{2100}^{(i)} - a_{0300}^{(i)} - a_{3000}^{(i)})$$

$$q_{1010}^{(i)} = \frac{1}{4}(3a_{2010}^{(i)} + 3a_{1020}^{(i)} - a_{0030}^{(i)} - a_{3000}^{(i)})$$

$$q_{0110}^{(i)} = \frac{1}{4}(3a_{0210}^{(i)} + 3a_{0120}^{(i)} - a_{0300}^{(i)} - a_{0030}^{(i)})$$

5 Visualization and Examples

We can visualize the graph of the constructed function F on the domain surface D either by projecting the iso-contours onto the surface D, or by directly dsiplaying iso-contours or the surface graph of the function F in space.

Displaying Iso-contours of F on D

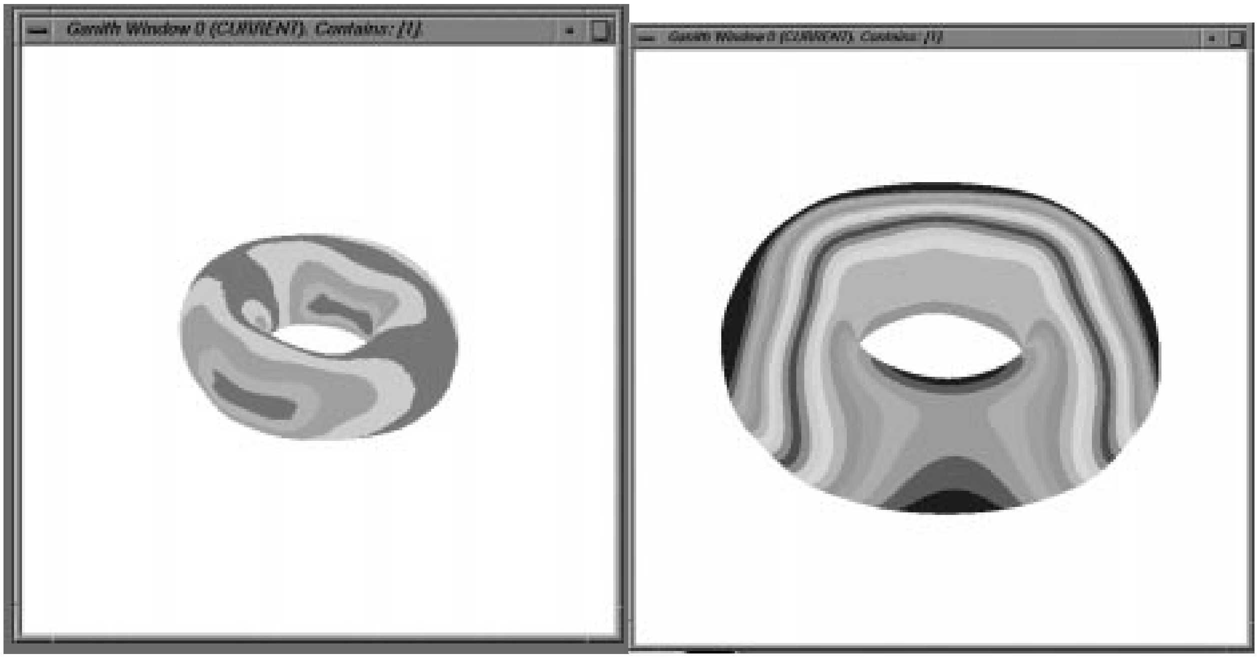

Figure 5.1: Iso-contours of a C^1 approximated function F shown on a domain torus D

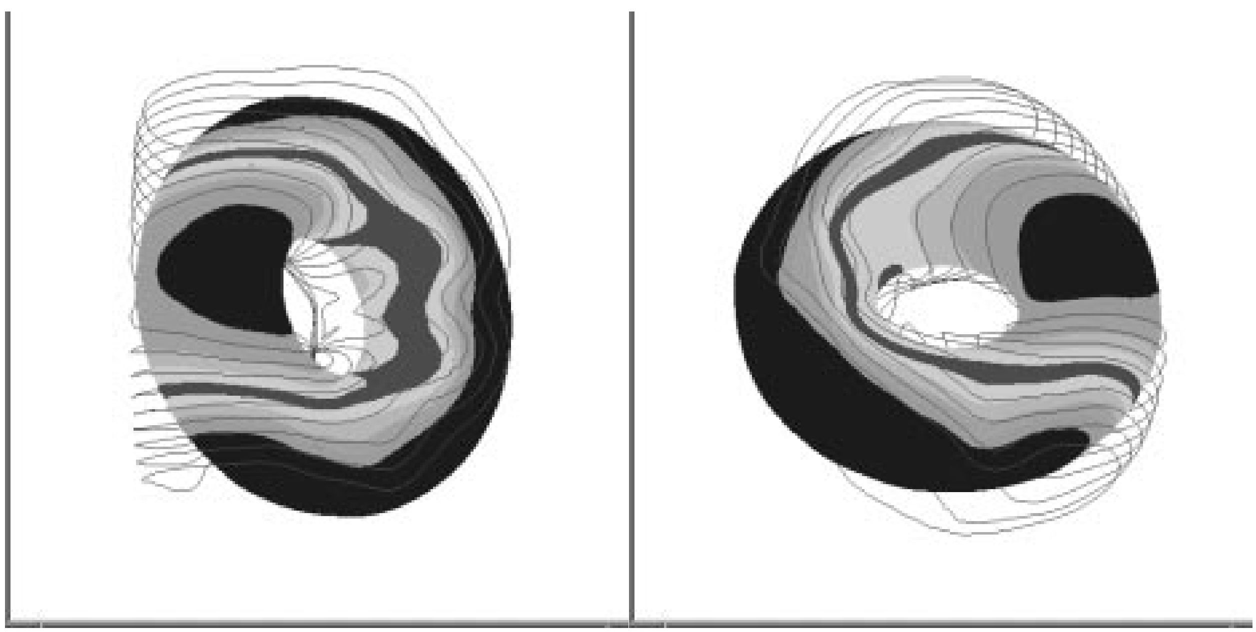

Figure 5.2: Iso-contours of a C^2 approximated function F shown on and surrounding a domain torus D using a normal projection

We display the iso-contours on the domain surface by showing different colors in the region between two iso-contours. In our approach, we achieve this by first generating a planar triangular approximation of the domain surface, and then generating the corresponding four dimensional triangles on F, and finally intersecting these triangles with the iso-values to get the line segments of the iso-contours. Let w be a given iso-value, $[p_1 p_2 p_3]$ be a triangle on D. Without loss of generality, we may assume $F(p_1) \leq F(p_2) \leq F(p_3)$. Then if $w < F(p_1)$ or $w > F(p_3)$, the triangle does not intersect the iso-value. If $w \in [F(p_1), F(p_3)]$, say $w \in [F(p_1), F(p_2)]$, let $t_1 = \frac{w-F(p_1)}{F(p_2)-F(p_1)}$, $t_2 = \frac{w-F(p_1)}{F(p_3)-F(p_1)}$, $q_1 = t_1 p_1 + (1-t_1)p_2$, $q_2 = t_2 p_1 + (1-t_2)p_3$, then $[q_1 q_2]$ is one segment of the contour $F(p) = w$. The collection of all of these line segments form a piecewise approximation to the iso-contours. By increasing the resolution of the triangulation of the domain surface, we can get better approximations of the iso-contours. Figure 5.1 (left and right) shows the iso-contours of a C^1 approximated function F, on a domain torus D. Figure 5.2 (left and right) shows the iso-contours of a C^2 approximated function F, on a domain torus D. The iso-contours of the C^2 approximated function F are also shown surrounding the domain torus using the normal projection scheme given below.

Displaying Iso-contours and the graph of F in $\mathbb{R}^3$

Since the iso-contours may not clearly indicate the geometric shape of the function-on-surface, one often plot the function-on-surface in one way or another. One approach is to use a radial projection from some center of the domain. However, if the domain surface is not convex or has non-zero genus, this projection scheme has difficulties caused by self-intersection. Another more natural way is to use the normal projection, that is, project the point p on the domain surface D to a distance proportional to $F(p)$ in the normal direction of D at p: $G(p) = p + L\frac{\nabla f(p)(F(p)-F_{min})}{\|\nabla f(p)\|(F_{max}-F_{min})}$ where L is a positive scalar, F_{min} and F_{max} are minimum and maximum values of F on D. Here L has to be chosen properly so that the projected surface G does not self-intersect.

Figures 5.3, 5.4, (left and right) shows the iso-contours of a C^2 approximated function F, on a domain D. The iso-contours of the C^2 approximated function F are also shown surrounding the domain using the normal projection scheme.

References

[1] P. Alfeld. A trivariate clough-tocher scheme for tetrahedral data. *Computer Aided Geometric Design*, 1:169–181, 1984.

[2] C. Bajaj and G. Xu. *The Estimation of the Gradients and Hessian for Functions defined on Surfaces.* Computer Science Technical Report, Purdue University, 1994.

[3] R.E. Barnhill and T.A. Foley. Methods for constructing surfaces on surfaces. In G.Farin, editor, *Geometric Modeling: Methods and their Applications*, pages 1–15. Springer, Berlin, 1991.

[4] R.E. Barnhill, K. Opitz, and H. Pottmann. Fat surfaces: a trivariate approach to triangle-based interpolation on surfaces. *Computer Aided Geometric Design*, 9:365–378, 1992.

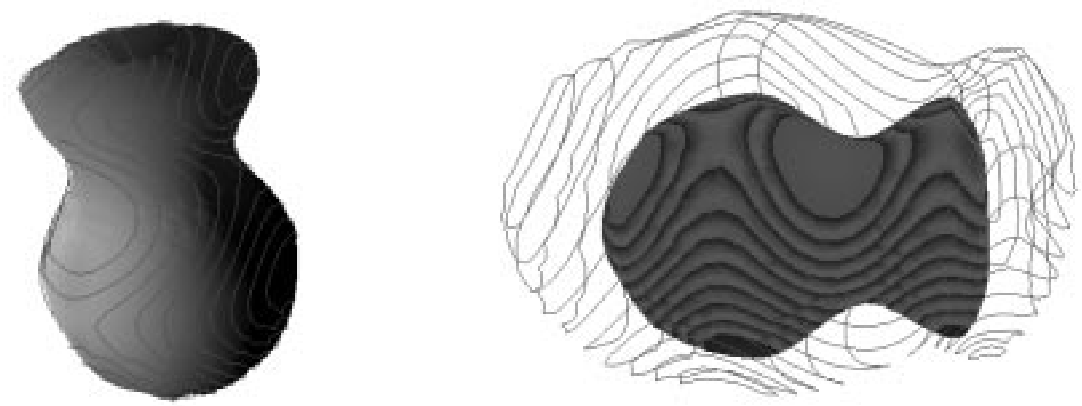

Figure 5.3: Iso-contours of a C^2 approximated function F_1 shown on and surrounding a domain surface D_1.

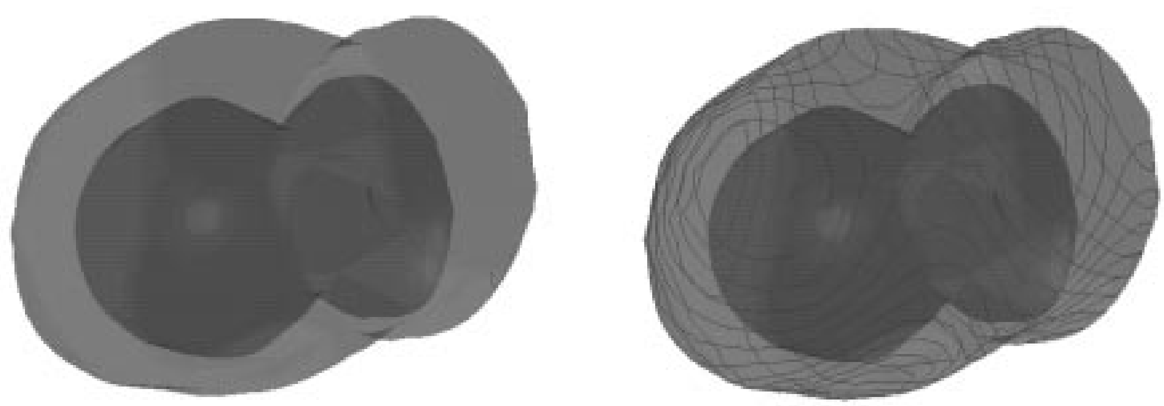

Figure 5.4: The surface and iso-contours in $\mathbb{R}^3$ of the C^2 approximated function F_1 surrounding the domain D_1. These are a normal projection from the domain D_1

[5] L. P. Chew. Guaranteed-quality mesh generation for curved surface. In *9th Annual Computational Geometry*, pages 274–280, CA,USA, May 1993.

[6] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press Inc., 1990.

[7] R. Franke. Recent advances in the approximation of surfaces from scattered data. In C.K.Chui, L.L.Schumarker, and F.I.Utreras, editors, *Multivariate Approximation*, pages 275–335. Academic Press, New York, 1987.

[8] G. Nielson, T. Foley, B. Hamann, and D. Lane. Visualizing and Modeling Scattered Multivariate Data. *IEEE Computer Graphics And Applications*, 11:47–55, 1991.

[9] K.L. Rescorla. C^1 Trivariate Polynomial Interpolation. *Computer Aided Geometric Design*, 4:237–244, 1987.

[10] A. J. Worsey and G. Farin. An n-dimensional clough-tocher interpolant. *Constructive Approximation*, 3:99–110, 1987.