Hierarchical Spline Approximations

David F. Wiley[1], Martin Bertram[2,1], Benjamin W. Jordan[3], Bernd Hamann[1], Kenneth I. Joy[1], Nelson L. Max[1,4], and Gerik Scheuermann[2]

[1] Center for Image Processing and Integrated Computing (CIPIC), Department of Computer Science, University of California, Davis, CA 95616-8562, U.S.A.; e-mail: {wiley, hamann, joy}@cs.ucdavis.edu
[2] University of Kaiserslautern, Fachbereich Informatik, P.O. Box 3049, D-67653 Kaiserslautern, Germany; e-mail: {bertram, scheuer}@informatik.uni-kl.de
[3] Pixar Feature Division, Pixar Animation Studios, 1001 West Cutting Blvd., Richmond, CA 94804, U.S.A.; e-mail: bjordan@pixar.com
[4] Center for Applied Scientific Computing (CASC), Lawrence Livermore National Laboratory, 7000 East Avenue, L-551, Livermore, CA 94550, U.S.A.; e-mail: max2@llnl.gov

Abstract. We discuss spline refinement methods that approximate multi-valued data defined over one, two, and three dimensions. The input to our method is a coarse decomposition of the compact domain of the function to be approximated consisting of intervals (univariate case), triangles (bivariate case), and tetrahedra (trivariate case). We first describe a best linear spline approximation scheme, understood in a least squares sense, and refine on initial mesh using repeated bisection of simplices (intervals, triangles, or tetrahedra) of maximal error. We discuss three enhancements that improve the performance and quality of our basic bisection approach. The enhancements we discuss are: (i) using a finite element approach that only considers original data sites during subdivision, (ii) including first-derivative information in the error functional and spline-coefficient computations, and (iii) using quadratic (deformed, "curved") rather than linear simplices to better approximate bivariate and trivariate data. We improve efficiency of our refinement algorithms by subdividing multiple simplices simultaneously and by using a sparse-matrix representation and system solver.

1 Introduction

Different methods are known and used for hierarchical representation of very large data sets. Unfortunately, only a small number of these methods are based on well developed mathematical theory. In the context of visualizing very large data sets in two and three dimensions, it is imperative to develop hierarchical data representations that allow us to visualize and analyze data at various levels of detail. General and efficient algorithms are needed to support the generation of hierarchical data representations and their applicability for visualization.

Our discussion deals with the construction of hierarchies of triangulations and spline approximations of functions. The main idea underlying the

construction of our data hierarchy is repeated bisection of simplices (intervals, triangles, or tetrahedra). Bisection is chosen for its simplicity and the possibility to extend it to multiple dimensions easily. We construct an approximation hierarchy by repeatedly subdividing triangulations, "simplicial domain decompositions", and computing a best spline approximations. Our initial approximation is based on a coarse triangulation of the domain of interest, typically defined over the convex hull of all given data points/sites if the function to be approximated is known only at a finite number of locations. We identify regions of large error and subdivide simplices in these regions. These steps are required to perform our algorithm:

1. **Initial approximation.** Define a coarse initial triangulation of the domain of interest and compute the coefficients defining the best spline approximation for this mesh.
2. **Error estimation.** Analyze the error of this approximation by computing appropriate local and global error estimates relative to the input function.
3. **Refinement.** Subdivide the simplex (or set of simplices) with the largest local error estimate.
4. **Computation of best approximation.** Compute a new best linear spline approximation based on the new mesh.
5. **Iteration.** Repeat steps 2, 3, and 4 until a certain approximation error condition is met.

There are many hierarchical methods targeted at approximating large data sets. For example, wavelet methods are described in [2], [13], and [29]. The work described in [29] has the advantage of supporting both lossless and lossy compression. In general, wavelet methods work well for data lying on uniform, rectilinear, and power-of-two grids and provide fast and highly accurate compression.

Simplification methods using data elimination strategies are described in [3], [4], [11], [12], [16], [20], and [21]. These methods are more general than most wavelet methods since arbitrary input meshes can be converted to a form treatable by each method. Refinement methods similar to the ones we discuss here are described in [14], [17], and [27]. Most data-dependent refinement methods can also be adapted to arbitrary meshes.

The method described in [7] performs an iterative "thinning step" based on radial basis functions on scattered points while maintaining a Delaunay triangulation. The results of [7] support the notion that data-dependent triangulations better approximate functions in high-gradient regions.

Comparisons of wavelet, decimation, simplification, and data-dependent methods, including the meshes discussed in [13], [16], [20], and [27], are provided in [19]. This survey discusses the many approaches to surface simplification and also examines the complexity of some of the most commonly used methods.

The methods we discuss here apply to univariate, bivariate, and trivariate data. The underlying principles of our approach become evident in our discussion of the univariate case in Sect. 2. By discussing the univariate case in detail, generalizations to the bivariate and trivariate cases are more easily understood.

The hierarchies of approximations resulting from our methods can be used for visualization. Common visualization methods – including contouring, slicing, and volume visualization (i.e., ray-casting) – that can be applied to our hierarchical approximations are described in [15], [22], [24], [25], and [28].

2 Best Linear Spline Approximation and the Univariate Case

We begin the discussion with best linear spline approximation in the univariate case. Our method requires a few notations from linear algebra and approximation theory, and we discuss these briefly. We use the standard scalar product $\langle f, g \rangle$ of two functions $f(x)$ and $g(x)$, defined over the interval $[a, b]$,

$$\langle f, g \rangle \;=\; \int_a^b f(x)g(x)dx, \tag{1}$$

and the standard L^2 norm to measure a function $f(x)$,

$$\|f\| \;=\; \langle f, f \rangle^{1/2} \;=\; \left(\int_a^b \big(f(x)\big)^2 dx \right)^{1/2}. \tag{2}$$

It is well known from approximation theory, see, for example, [5], that the best approximation $f(x)$ of a given function $F(x)$, when approximating it by a linear combination

$$f(x) = \sum_{i=0}^{n-1} c_i f_i(x) \tag{3}$$

of independent functions $f_0(x), \ldots, f_{n-1}(x)$ (using f_i as abbreviation for $f_i(x)$), is defined by the normal equations

$$\begin{bmatrix} \langle f_0, f_0 \rangle & \cdots & \langle f_{n-1}, f_0 \rangle \\ \vdots & & \vdots \\ \langle f_0, f_{n-1} \rangle & \cdots & \langle f_{n-1}, f_{n-1} \rangle \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} \langle F, f_0 \rangle \\ \vdots \\ \langle F, f_{n-1} \rangle \end{bmatrix}. \tag{4}$$

We can also write this linear system as $\boldsymbol{M}^{[n-1]}\boldsymbol{c}^{[n-1]} = \boldsymbol{F}^{[n-1]}$. This system is easily solved when dealing with a set of mutually orthogonal and normalized basis functions, i.e., in the case when $\langle f_i, f_j \rangle = \delta_{i,j}$ (*Kronecker delta*). In this case, only the diagonal entries of $\boldsymbol{M}^{[n-1]}$ are non-zero and the coefficients c_i

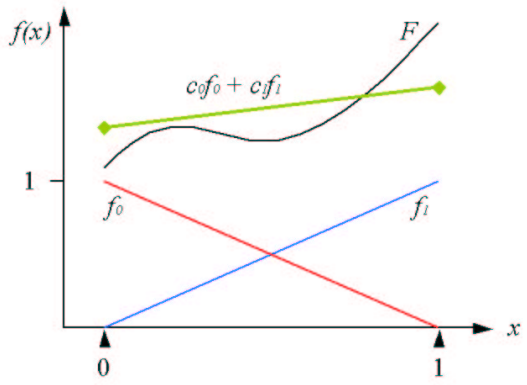

Fig. 1. Basis functions f_i, function F to be approximated, and approximation $f(x) = c_0 f_0 + c_1 f_1$

are given by $c_i = \langle F, f_i \rangle$. For an arbitrary set of basis functions, one must investigate a means for an efficient solution of the linear system define by 4.

Initially we approximate a given univariate function F by a single linear spline segment by computing $\boldsymbol{M}^{[1]}\boldsymbol{c}^{[1]} = \boldsymbol{F}^{[1]}$. We assume, without loss of generality, that F is defined over the interval $[0, 1]$ and that the basis functions are hat functions with the property $f_i(x_j) = \delta_{i,j}$, see Fig. 1.

When using hat functions as basis functions, the only non-zero elements of $\boldsymbol{M}^{[n-1]}$, for a particular row i, are the elements $\langle f_{i-1}, f_i \rangle$, $\langle f_i, f_i \rangle$, and $\langle f_{i+1}, f_i \rangle$. These scalar products are given by

$$\begin{aligned} \langle f_{i-1}, f_i \rangle &= \frac{1}{6}\Delta_{i-1}\,, \\ \langle f_i, f_i \rangle &= \frac{1}{3}\left(\Delta_{i-1} + \Delta_i\right)\,, \quad \text{and} \\ \langle f_{i+1}, f_i \rangle &= \frac{1}{6}\Delta_i\,, \end{aligned} \tag{5}$$

where $\Delta_i = x_{i+1} - x_i$. Thus, $\boldsymbol{M}^{[n-1]}$ is the tridiagonal matrix

$$\boldsymbol{M}^{[n-1]} = \begin{bmatrix} 2\Delta_0 & \Delta_0 & & & & \\ \Delta_0 & 2(\Delta_0 + \Delta_1) & \Delta_1 & & & \\ & \Delta_1 & 2(\Delta_1 + \Delta_2) & \Delta_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & \Delta_{n-2} & \Delta_{n-1} \end{bmatrix}. \tag{6}$$

Since it is necessary to re-compute the coefficients after each refinement due to the global nature of the best approximation problem, one can take advantage of efficient system solvers to solve this system in linear time, see, for example [6].

We want to compute a hierarchy of approximations of a function F by refining the initial approximation by adding more basis functions – or, in other words, by inserting more knots. We insert knots repeatedly until we have an approximation whose global error is smaller than some threshold. The error of the first approximation is defined as $E^{[1]} = \|F - (c_0 f_0 + c_1 f_1)\|$. If this value is larger than a specified tolerance, we refine by inserting a knot at $x = \frac{1}{2}$. The addition of this knot changes the basis function sequence and, therefore, we must compute a new best linear spline approximation for the new knot set by solving $\boldsymbol{M}^{[2]}\boldsymbol{c}^{[2]} = \boldsymbol{F}^{[2]}$; the error for this approximation is $E^{[2]} = \|F - \sum_{i=0}^{2} c_i f_i\|$. Should this new error still be too large, we refine the approximation further. We need to define a criterion that determines which segment to bisect. Local error estimates over each segment can be computed easily, so this information can be used to decide which interval to subdivide, in this case, either $[0, \frac{1}{2}]$ or $[\frac{1}{2}, 1]$.

We can define a global error estimate for an approximation based on the knot sequence $0 = x_0 < x_1 < x_2 < \ldots < x_{n-2} < x_{n-1} = 1$. We define the global error as

$$E^{[n-1]} = \|F - \textstyle\sum_{i=0}^{n-1} c_i f_i\|. \tag{7}$$

In order to allow us to decide which segment to subdivide next, we define the local error for an interval $[x_i, x_{i+1}]$ as

$$e_i^{[n-1]} = \left(\int_{x_i}^{x_{i+1}} \left(F - (c_i f_i + c_{i+1} f_{i+1})\right)^2 dx \right)^{1/2}, \; i = 0, \ldots, n-2 \; . \tag{8}$$

We compute local errors for each of the segments and then bisect the segment with the maximum local error estimate at each iteration. If the maximum local error is not unique, it is sufficient to choose one segment randomly for subdivision. (Alternatively, one could subdivide all segments with the same maximum local error, thus leading to a unique solution). To improve efficiency, it is reasonable to select the m segments with the m largest local error estimates for subdivision. Such an approach seems to be more appropriate for very large data sets. An example of a hierarchical approximation of a univariate function is shown in Fig. 2.

The scalar products $\langle F, f_i \rangle$ and the error values $E^{[n-1]}$ and $e_i^{[n-1]}$ are computed by numerical integration. We use *Romberg integration* to perform these steps, see [1] and [17].

3 The Bivariate Case

Given an initial best linear spline approximation based on a small number of triangles, we compute the global error estimate for the approximation and, should this error be too large, bisect the longest edge of the triangle with maximal local error and insert a new knot at the midpoint of the longest

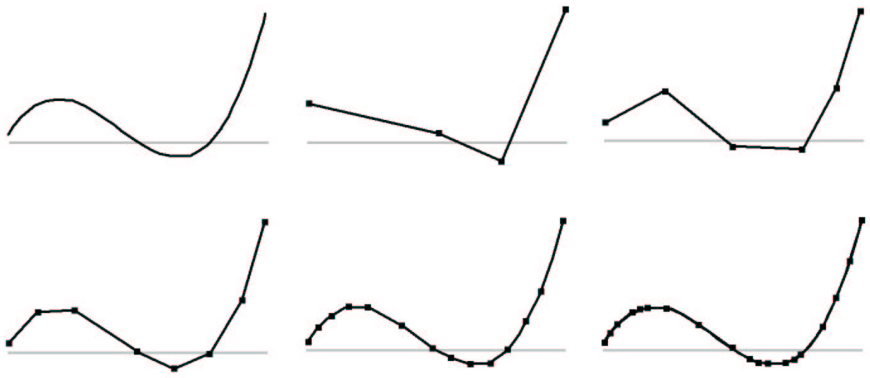

Fig. 2. Five approximations of $F(x) = 10x\left(x - \frac{1}{2}\right)\left(x - \frac{3}{4}\right)$, $x \in [0, 1]$. The original function is shown in the upper-left corner. Number of knots for approximations: 4, 6, 8, 14, and 19

edge of this triangle. If this edge is shared by another triangle, the neighboring triangle is also split.

Refinement leads to insertion of additional knots at each iteration. Again, we must re-compute spline coefficients after each knot insertion step due to the global nature of the best approximation problem.

Concerning the construction of an approximation hierarchy for function $F(x, y)$, we begin with a coarse initial triangulation of the domain. In the case of scattered data, a fitting step must be done to produce a smooth function for which the approximation hierarchy is constructed. Localized versions of *Hardy's multiquadric method*, see [18], or surface reconstruction techniques, similar to the one described in [9], can be used to yield smooth functions interpolating given discrete data. A minimal coarse triangulation of the convex hull of the scattered data sites suffices as an initial triangulation. Earlier work concerning scattered data interpolation and approximation is described in [10] and [23].

The approximation $f(x, y)$ is constructed from the given function $F(x, y)$, the underlying triangulation, which is a set of vertices $\boldsymbol{v_i} = [x_i, y_i]^T$, $i = 0, \ldots, n$, and hat basis functions $f_i = f_i(x, y)$. The basis function f_i – associated with vertex $\boldsymbol{v}_i$ – has a value of one at $\boldsymbol{v}_i$ and varies linearly to zero when going from $\boldsymbol{v}_i$ to all other vertices in the platelet of $\boldsymbol{v}_i$; f_i is zero outside the platelet of $\boldsymbol{v}_i$, see Fig. 3.

Univariate error estimates are easily extended to the bivariate case. Formally, the bivariate global error is the same as in the univariate case (7), and local errors for each triangle T_j are given by

$$e_j^{[n-1]} = \left(\int_{T_j} \left(F - \sum\nolimits_{k=1}^{3} c_{j,k} f_{j,k} \right)^2 dxdy \right)^{\frac{1}{2}},$$

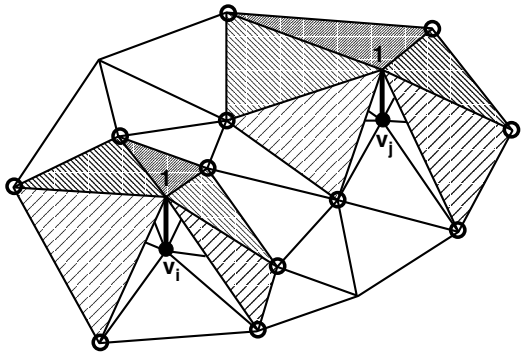

Fig. 3. Platelets of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ and associated basis functions. (Triangles in front have been removed for clarity)

$$j = 0, \ldots, n_T - 1\,, \tag{9}$$

where n_T is the number of triangles in a mesh and $f_{j,k}$ is the basis function associated with the k^{th} vertex of the j^{th} triangle.

The bivariate case requires integration over triangles. For this purpose, we make use of the *change-of-variables theorem*, which allows us to effectively integrate functions over a triangle with vertices $\boldsymbol{v}_0 = [x_0, y_0]^T$, $\boldsymbol{v}_1 = [x_1, y_1]^T$, and $\boldsymbol{v}_2 = [x_2, y_2]^T$:

Change-of-variables theorem

Let R and R^* be regions in the plane and let $M\colon R^* \to R$ be a C^1-continuous, one-to-one mapping such that $M(R^*) = R$. Then, for any bivariate integrable function f, the equation

$$\int_R f(x,y)dxdy = \int_{R^*} f\left(x(u,v), y(u,v)\right) J\; dudv \tag{10}$$

holds, where J is the *Jacobian* of M,

$$J = \det \begin{bmatrix} \frac{\partial}{\partial u}x(u,v) & \frac{\partial}{\partial v}x(u,v) \\ \frac{\partial}{\partial u}y(u,v) & \frac{\partial}{\partial v}y(u,v) \end{bmatrix}. \tag{11}$$

Thus, to effectively compute integrals of functions over triangles we only need to consider the linear transformation

$$\begin{bmatrix} x(u,v) \\ y(u,v) \end{bmatrix} = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}. \tag{12}$$

This transformation maps the *standard triangle* T^* with vertices $\boldsymbol{u}_0 = [0,0]^T$, $\boldsymbol{u}_1 = [1,0]^T$, and $\boldsymbol{u}_2 = [0,1]^T$ in the uv-plane to the arbitrary triangle T with vertices $\boldsymbol{v}_0 = [x_0, y_0]^T$, $\boldsymbol{v}_1 = [x_1, y_1]^T$, and $\boldsymbol{v}_2 = [x_2, y_2]^T$ in the

xy-plane. (Both triangles must be oriented counterclockwise.) For this linear mapping, the change-of-variables theorem yields

$$\int_T f(x,y)dxdy = J\int_{v=0}^{1}\int_{u=0}^{1-v} f\left(x(u,v),y(u,v)\right)dudv, \tag{13}$$

where the Jacobian is given by

$$J = \det\begin{bmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{bmatrix}. \tag{14}$$

The only scalar products of basis function pairs one must consider in the bivariate case are $\langle N_0, N_0\rangle$ and $\langle N_0, N_1\rangle$, where $N_i(u_j, v_j) = \delta_{i,j}$ is a linear spline basis function defined over the standard triangle. (A scalar product $\langle f_i, f_j\rangle$ is only non-zero if the platelets of the vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ intersect.) The values of these two scalar products are

$$\begin{aligned}\langle N_0, N_0\rangle &= \int_{v=0}^{1}\int_{u=0}^{1-v}(1-u-v)^2 dudv = \frac{1}{12} \quad \text{and} \\ \langle N_0, N_1\rangle &= \int_{v=0}^{1}\int_{u=0}^{1-v}(1-u-v)u\ dudv = \frac{1}{24}\,.\end{aligned} \tag{15}$$

Thus, the scalar product $\langle f_i, f_i\rangle$ is given by

$$\langle f_i, f_i\rangle = \textstyle\sum_{j=0}^{n_i-1}\int_{T_j} f_i f_i\ dxdy = \frac{1}{12}\sum_{j=0}^{n_i-1} J_j\ , \tag{16}$$

where n_i is the number of platelet triangles associated with vertex $\boldsymbol{v}_i$ and J_j is the Jacobian associated with the j^{th} platelet triangle. The scalar product $\langle f_i, f_k\rangle$ of two basis functions, whose associated vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_k$ are connected by an edge, is given by

$$\langle f_i, f_k\rangle = \textstyle\sum_{j=0}^{n_{i,k}-1}\int_{T_j} f_i f_k\ dxdy = \frac{1}{24}\sum_{j=0}^{n_{i,k}-1} J_j\ , \tag{17}$$

where $n_{i,k}$ is the number of platelet triangles shared by vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_k$ and J_j is the Jacobian associated with the j^{th} platelet triangle. An example of a hierarchy of bivariate best linear spline approximations is shown in Fig. 4.

4 The Trivariate Case

The trivariate case is a straightforward generalization of the bivariate case. The only notable difference from the bivariate case is the use of the change-of-variables theorem. We consider the mapping of the *standard tetrahedron* with vertices $\boldsymbol{u}_0 = [0,0,0]^T$, $\boldsymbol{u}_1 = [1,0,0]^T$, $\boldsymbol{u}_2 = [0,1,0]^T$, and $\boldsymbol{u}_3 = [0,0,1]^T$ in uvw-space to the tetrahedron with vertices $\boldsymbol{v}_0 = [x_0, y_0, z_0]^T$,

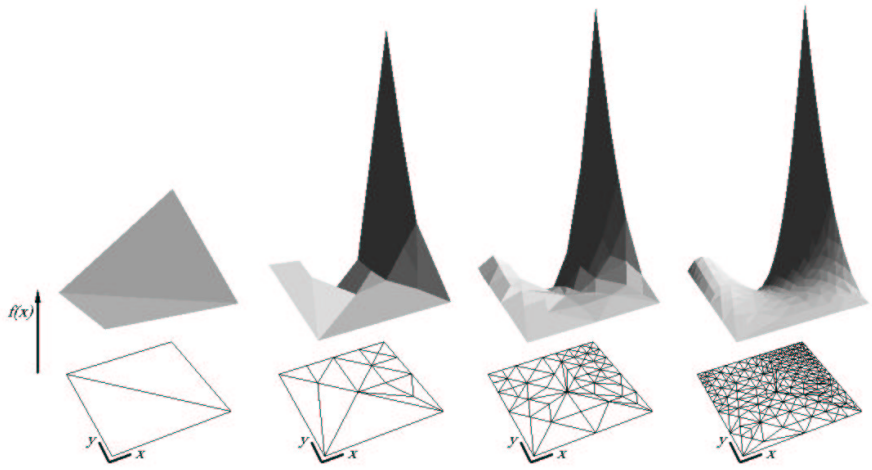

Fig. 4. Four approximations of $F(x,y) = 10x\left(x-\frac{1}{4}\right)\left(x-\frac{3}{4}\right)y^2$, $x,y \in [0,1]$; the origin is indicated by a pair of perpendicular line segments; number of knots for approximations: 4, 15, 58, and 267; corresponding global error estimates: 0.1789, 0.0250, 0.0057, and 0.0011

$\boldsymbol{v}_1 = [x_1, y_1, z_1]^T$, $\boldsymbol{v}_2 = [x_2, y_2, z_2]^T$, and $\boldsymbol{v}_3 = [x_3, y_3, z_3]^T$ in xyz-space. The resulting linear transformation is given by

$$\begin{bmatrix} x(u,v,w) \\ y(u,v,w) \\ z(u,v,w) \end{bmatrix} = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}. \tag{18}$$

In this case, the change-of-variables theorem implies that

$$\int_T f(x,y,z)dxdydz = J\int_{w=0}^{1}\int_{v=0}^{1-w}\int_{u=0}^{1-v-w} f\left(x(u,v,w), y(u,v,w), z(u,v,w)\right) dudvdw, \tag{19}$$

where the Jacobian is given by

$$J = \det \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{bmatrix}. \tag{20}$$

As in the bivariate case, we need to consider only the scalar products of $\langle N_0, N_0\rangle$ and $\langle N_0, N_1\rangle$, where $N_i(u_j, v_j, w_j) = \delta_{i,j}$ is a linear spline basis function over the standard tetrahedron. The values of these two scalar products are

$$\langle N_0, N_0\rangle = \int_{w=0}^{1}\int_{v=0}^{1-w}\int_{u=0}^{1-v-w}(1-u-v-w)^2 dudvdw = \frac{1}{60}$$

and

$$\langle N_0, N_1 \rangle = \int_{w=0}^{1} \int_{v=0}^{1-w} \int_{u=0}^{1-v-w} (1-u-v-w)u \; dudvdw = \frac{1}{120} \; . \quad (21)$$

Thus, the scalar product $\langle f_i, f_i \rangle$ is given by

$$\langle f_i, f_i \rangle = \tfrac{1}{60} \sum_{j=0}^{n_i-1} J_j \; , \quad (22)$$

where n_i is the number of platelet tetrahedra associated with vertex $\boldsymbol{v}_i$ and J_j is the Jacobian associated with the j^{th} platelet tetrahedron. The scalar product $\langle f_i, f_k \rangle$ of two basis functions, whose associated vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_k$ are connected by an edge, is given by

$$\langle f_i, f_k \rangle = \tfrac{1}{120} \sum_{j=0}^{n_{i,k}-1} J_j \; , \quad (23)$$

where $n_{i,k}$ is the number of platelet tetrahedra shared by vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_k$ and J_j is the Jacobian associated with the j^{th} platelet tetrahedron. An example of a hierarchy of trivariate best linear spline approximations is shown in Fig. 5.

5 Finite Element Enhancement

Often, the function being approximated is known only at a finite number of locations. An enhancement to the method we have discussed in previous sections is to take advantage of finitely specified, "discrete" data to save space when storing a hierarchy of approximations. A sequence of knot insertions could potentially reference original data sites rather than explicitly specifying the exact location of each knot when performing edge bisection.

We incorporate the idea of utilizing only originally given data site into the approximation method by modifying the initial input mesh and the bisection step. The vertices in the initial input mesh must be a subset of the original data sites. The convex hull of the data sites suffices for the construction of an initial mesh. Regarding the subdivision steps, in the univariate case, rather than selecting the exact midpoint of an interval subdivision, we select the original data site nearest to the midpoint (while respecting the interval extents) as the new knot. For bivariate and trivariate subdivision, we select the nearest original data site (inside the simplex selected for subdivision) to the midpoint of the longest edge. If the nearest original data site is not unique, we choose one at random. A simplex is not subdivided if there are no original data sites inside it. Care must by taken to ensure a valid mesh since the "snapping" to original data sites can produce inside-out simplices, see Fig. 6.

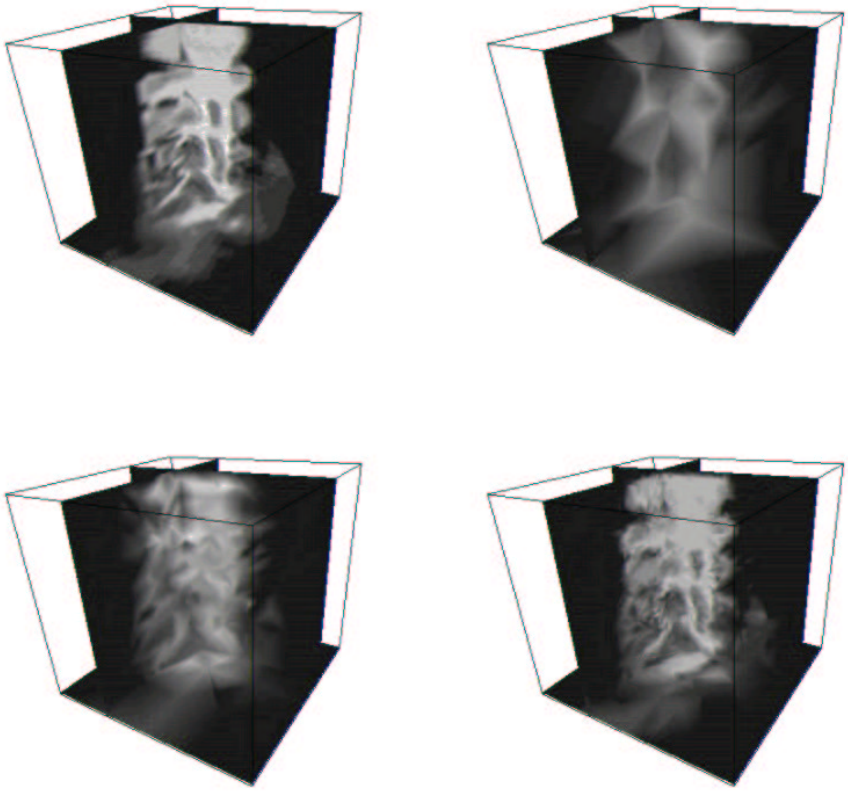

Fig. 5. Three approximations of flame data; original data shown in the upper-left corner (208000 sites). Number of knots for approximations: 68, 514, and 5145; corresponding global error estimates: 0.12, 0.07, and 0.03

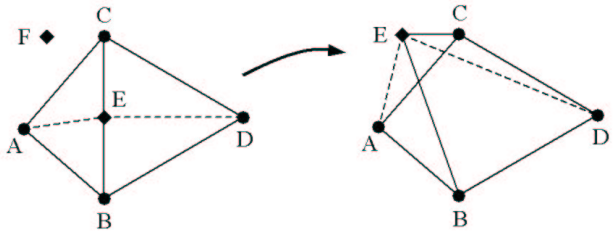

Fig. 6. Inside-out triangles are produced when snapping bisection vertex E to original data site F

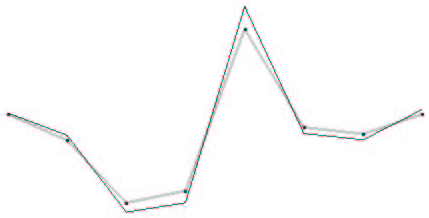

Fig. 7. Demonstration of effect of over- and under-shoots. The lighter polygon shows the original function; the darker polygon shows the approximation

6 Incorporating First-derivative Information

When approximating discontinuous data, we have noticed problems with our generated approximations, namely, "over-" and "under-shoots," see Fig. 7 for an example.
To solve this problem we include first-derivative information for spline construction. In addition, we must then also include first-derivative information in the error computations to help focus the refinement near discontinuous regions and regions with high derivatives/gradients. No algorithmic changes are made otherwise to the original bisection method. The refinement process proceeds by starting with a coarse initial triangulation and iteratively refining the simplices with maximal error.

6.1 The Univariate Case

To incorporate first-derivative information into the computations, a few changes must be made. We re-define (1), the univariate scalar product $\langle f, g\rangle$ of two functions $f(x)$ and $g(x)$, defined over the interval $[a, b]$, as

$$\langle f, g\rangle \;=\; \int_a^b w_0 f(x)g(x) + w_1 f'(x)g'(x)dx, \tag{24}$$

where the "weights" w_0 and w_1 are non-negative and sum to one. This new scalar product defines a "*Sobolev*-like" L_2 norm – denoted by $\| \|_{Sob}$ – for a function $f(x)$, which replaces (2):

$$\|f\|_{Sob} = \langle f, f\rangle^{1/2} = \left(\int_a^b w_0 \left(f(x)\right)^2 + w_1 \left(f'(x)\right)^2 dx\right)^{1/2}. \tag{25}$$

This norm allows us to measure the quality of an approximation $f(x)$ of a given function $F(x)$ by considering the norm of $D(x) = F(x) - f(x)$:

$$\|D\|_{Sob} = \langle D, D\rangle^{1/2}. \tag{26}$$

We use this new measure to compute interval-specific error estimates. The univariate global error (7) becomes

$$E = \int_a^b w_0 \left(F(x) - f(x)\right)^2 + w_1 \left(F'(x) - f'(x)\right)^2 dx\,, \tag{27}$$

where F is the function to be approximated on the interval $[a, b]$ by function f. We want to minimize E, which is equivalent to minimizing

$$\begin{aligned} E = & \int_a^b w_0 \left(f(x)\right)^2 + w_1 \left(f'(x)\right)^2 \\ & - 2\left(w_0 F(x) f(x) + w_1 F'(x) f'(x)\right) dx \\ & + \int_a^b w_0 \left(F(x)\right)^2 + w_1 \left(F'(x)\right)^2 \ , \end{aligned} \tag{28}$$

which, in matrix notation, reduces to

$$\begin{aligned} E = & \int_a^b \frac{1}{2} \begin{bmatrix} f(x) & f'(x) \end{bmatrix} \begin{bmatrix} 2w_0 & 0 \\ 0 & 2w_1 \end{bmatrix} \begin{bmatrix} f(x) \\ f'(x) \end{bmatrix} \\ & - \begin{bmatrix} f(x) & f'(x) \end{bmatrix} \begin{bmatrix} 2w_0 F(x) \\ 2w_1 F'(x) \end{bmatrix} dx \\ = & \int_a^b \frac{1}{2} \boldsymbol{f}^T Q \boldsymbol{f} - \boldsymbol{f}^T \boldsymbol{l} dx \, . \end{aligned} \tag{29}$$

Substituting (3) into (29) yields

$$\begin{aligned} E = & \frac{1}{2} \begin{bmatrix} c_0 & c_1 & \dots & c_{n-1} \end{bmatrix} \begin{bmatrix} \int q(f_0, f_0) & \cdots & \int q(f_0, f_{n-1}) \\ \vdots & & \vdots \\ \int q(f_{n-1}, f_0) & \cdots & \int q(f_{n-1}, f_{n-1}) \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_{n-1} \end{bmatrix} \\ & - \begin{bmatrix} c_0 & c_1 & \dots & c_{n-1} \end{bmatrix} \begin{bmatrix} \int l(f_0) \\ \vdots \\ \int l(f_{n-1}) \end{bmatrix} dx \\ = & \frac{1}{2} \boldsymbol{c}^T \boldsymbol{A} \boldsymbol{c} - \boldsymbol{c}^T \boldsymbol{l} \, , \end{aligned} \tag{30}$$

where $q(f_i, f_j)$ is quadratic and $l(f_i)$ is linear in f_i, f_j, and their derivatives. The "energy" E is minimal for the set of coefficients c_i resulting from the *Ritz equations*, i.e., the linear system

$$\boldsymbol{A} \boldsymbol{c} = \boldsymbol{l} \ , \tag{31}$$

see [1]. The elements $a_{i,j}$ of the symmetric, positive definite matrix A are given by

$$\begin{aligned} a_{i,j} = w_0 \int_a^b f_i(x) f_j(x) dx + w_1 \int_a^b f_i'(x) f_j'(x) dx \ , \\ i, j = 0, \dots, n-1 \, , \end{aligned} \tag{32}$$

and the elements l_i of the column vector $\boldsymbol{l}$ are given by

$$\begin{aligned} l_i = w_0 \int_a^b F(x) f_i(x) dx + w_1 \int_a^b F'(x) f_i'(x) dx \ , \\ i = 0, \dots, n-1 \, . \end{aligned} \tag{33}$$

Integral values required to compute the matrix elements $a_{i,j}$ are

$$\int_{x_0}^{x_1} \left(f_0(x)\right)^2 dx = \frac{1}{3}\Delta_0\,, \tag{34a}$$

$$\int_{x_{n-1}}^{x_n} \left(f_n(x)\right)^2 dx = \frac{1}{3}\Delta_{n-1}\,, \tag{34b}$$

$$\int_{x_{i-1}}^{x_{i+1}} \left(f_i(x)\right)^2 dx = \frac{1}{3}(\Delta_{i-1}+\Delta_i)\ ,\ i=1,\ldots,n-1\,,\ \text{and} \tag{34c}$$

$$\int_{x_i}^{x_{i+1}} f_i(x) f_{i+1}(x) dx = \frac{1}{6}\Delta_i\ ,\ i=0,\ldots,n-1\,. \tag{34d}$$

Terms involving the first derivative are

$$\int_{x_0}^{x_1} \left(f_0'(x)\right)^2 dx = \frac{1}{\Delta_0}\,, \tag{35a}$$

$$\int_{x_{n-1}}^{x_n} \left(f_n'(x)\right)^2 dx = \frac{1}{\Delta_{n-1}}\,, \tag{35b}$$

$$\int_{x_{i-1}}^{x_{i+1}} \left(f_i'(x)\right)^2 dx = \frac{1}{\Delta_{i-1}}+\frac{1}{\Delta_i}\ ,\ i=1,\ldots,n-1\,,\ \text{and} \tag{35c}$$

$$\int_{x_i}^{x_{i+1}} f_i'(x) f_{i+1}'(x) dx = -\frac{1}{\Delta_i}\ ,\ i=0,\ldots,n-1\,. \tag{35d}$$

Thus, the matrix $\boldsymbol{A}$ is a tridiagonal matrix, given as a "weighted sum" of two tridiagonal matrices, $\boldsymbol{A}_0$ and $\boldsymbol{A}_1$:

$$\boldsymbol{A} = \frac{1}{6}\left(w_0\boldsymbol{A}_0 + w_1\boldsymbol{A}_1\right)\ , \tag{36}$$

where the matrices $\boldsymbol{A}_0$ and $\boldsymbol{A}_1$ are given by

$$\boldsymbol{A}_0 = \begin{bmatrix} 2\Delta_0 & \Delta_0 & & & & \\ \Delta_0 & 2(\Delta_0+\Delta_1) & \Delta_1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \Delta_{n-2} & 2(\Delta_{n-2}+\Delta_{n-1}) & \Delta_{n-1} & \\ & & & \Delta_{n-1} & 2\Delta_{n-1} \end{bmatrix}, \text{ and} \tag{37a}$$

$$\boldsymbol{A}_1 = 6\begin{bmatrix} \frac{1}{\Delta_0} & \frac{1}{\Delta_0} & & & & \\ \frac{-1}{\Delta_0} & \frac{\Delta_0+\Delta_1}{\Delta_0\Delta_1} & \frac{-1}{\Delta_1} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{-1}{\Delta_{n-2}} & \frac{\Delta_{n-2}+\Delta_{n-1}}{\Delta_{n-2}\Delta_{n-1}} & \frac{-1}{\Delta_{n-1}} & \\ & & & \frac{-1}{\Delta_{n-1}} & \frac{1}{\Delta_{n-1}} \end{bmatrix}. \tag{37b}$$

An example of a univariate approximation using first-derivative information is shown in Fig. 8. In this example, the function $F(x) = \sin(4\pi x^2)$ was sampled at eight uniformly spaced locations in the interval $[0, 1]$, defining a

piecewise linear spline $F(x)$, for which the approximations were constructed. It is apparent how the first derivative affects the over- and under-shoots. As more relative weight is assigned to the first derivative, the approximation becomes much better.

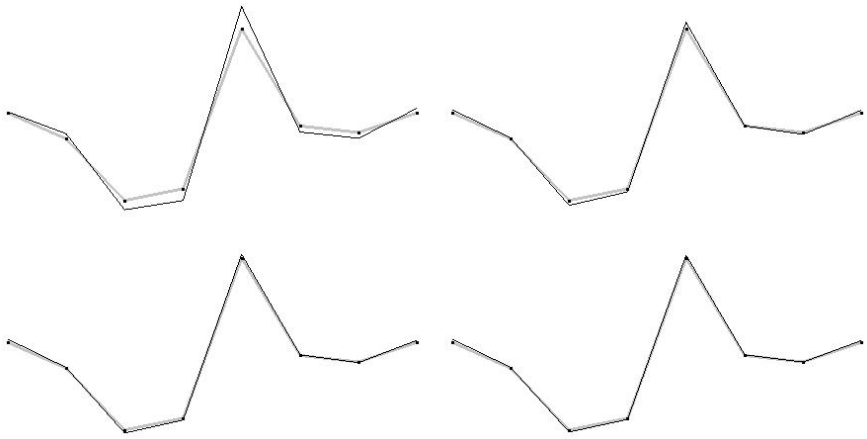

Fig. 8. Four approximations of $F(x) = \sin(4\pi x^2)$ using eight uniformly spaced knots with varying weights (w_0, w_1) given by (1.0, 0.0), (0.99, 0.01), (0.98, 0.02), and (0.97, 0.03) (from upper-left to lower-right corner); the lighter polygon shows the original function (polygon); the darker polygon is the approximation

6.2 The Bivariate Case

Using a generalization of (26), the error functional we want to minimize in the bivariate case is

$$\begin{aligned} E = & \int_T w_{0,0} \left(F(x,y) - f(x,y)\right)^2 + w_{1,0} \left(F_x(x,y) - f_x(x,y)\right)^2 \\ & + w_{0,1} \left(F_y(x,y) - f_y(x,y)\right)^2 dxdy \ , \end{aligned} \tag{38}$$

where F_x denotes the partial derivative of F with respect to x, f_x denotes the partial derivative of f with respect to x, etc. We want to minimize E, which corresponds, in matrix notation, to minimizing

$$\begin{aligned} E = & \int_T \frac{1}{2} \left[f(x,y)\ f_x(x,y)\ f_y(x,y)\right] \begin{bmatrix} 2w_{0,0} & 0 & 0 \\ 0 & 2w_{1,0} & 0 \\ 0 & 0 & 2w_{0,1} \end{bmatrix} \begin{bmatrix} f(x,y) \\ f_x(x,y) \\ f_y(x,y) \end{bmatrix} \\ & - \left[f(x,y)\ f_x(x,y)\ f_y(x,y)\right] \begin{bmatrix} 2w_{0,0}F(x,y) \\ 2w_{1,0}F_x(x,y) \\ 2w_{0,1}F_y(x,y) \end{bmatrix} dxdy \, . \end{aligned} \tag{39}$$

Substituting (3) into (39) yields, formally, the same equation one obtains in the univariate case. Regarding (31), the elements $a_{i,j}$ of the symmetric, positive definite matrix $\boldsymbol{A}$ are given by

$$\begin{aligned} a_{i,j} = w_{0,0} \int_T f_i(x,y) f_j(x,y) dxdy + w_{1,0} \int_T f_{ix}(x,y) f_{jx}(x,y) dxdy \ , \\ + \ w_{0,1} \int_T f_{iy}(x,y) f_{jy}(x) dxdy \ , \quad i,j = 0,\ldots,n-1 \, , \end{aligned} \tag{40}$$

where f_{ix} denotes the partial derivative of basis function f_i with respect to x, etc. The elements l_i of the column vector $\boldsymbol{l}$ are given by

$$\begin{aligned} l_i = w_{0,0} \int_T F(x,y) f_i(x,y) dxdy + w_{1,0} \int_T F_x(x,y) f_{ix}(x,y) dxdy \ , \\ + \ w_{0,1} \int_T F_y(x,y) f_{iy}(x,y) dxdy \ , \quad i = 0,\ldots,n-1 \, . \end{aligned} \tag{41}$$

Integral values required to compute the matrix elements $a_{i,j}$ are

$$\int_{T_i} \left(f_i(x,y)\right)^2 dxdy = \frac{1}{12} \sum_{j=0}^{n_i-1} |J_i| \ , \tag{42}$$

where n_i is the number of platelet triangles, T_i, associated with vertex $\boldsymbol{v}_i$ and J_i is the Jacobian, given by (14), associated with the j^{th} platelet triangle. Two basis functions $f_i(x,y)$ and $f_j(x,y)$ whose associated vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are connected by an edge imply the non-zero integral value

$$\int_{T_{i,j}} f_i(x,y) f_j(x,y) dxdy = \frac{1}{24} \sum_{k=0}^{n_{i,j}-1} |J_k| \ , \tag{43}$$

where $T_{i,j}$ is the set of triangles in common between the platelets of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. The linear polynomial interpolating the values one, zero, and zero at the vertices $[x_0, y_0]^T$, $[x_1, y_1]^T$, and $[x_2, y_2]^T$, respectively, has the partial derivatives

$$f_x(x,y) = -\frac{1}{J} \ \det \begin{bmatrix} 1 & y_1 \\ 1 & y_2 \end{bmatrix} = \frac{y_1 - y_2}{J} \quad \text{and} \tag{44a}$$

$$f_y(x,y) = -\frac{1}{J} \ \det \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \end{bmatrix} = \frac{x_2 - x_1}{J} \, . \tag{44b}$$

Integrals involving these partial derivatives are

$$\int_{T_i} \left(f_{ix}(x,y)\right)^2 dxdy = \frac{1}{2} \sum_{j=0}^{n_i-1} \frac{1}{|J_j|} \ \det^2 \begin{bmatrix} 1 & y_{j,1} \\ 1 & y_{j,2} \end{bmatrix} \quad \text{and} \tag{45a}$$

$$\int_{T_i} \left(f_{iy}(x,y)\right)^2 dxdy = \frac{1}{2} \sum_{j=0}^{n_i-1} \frac{1}{|J_j|} \ \det^2 \begin{bmatrix} x_{j,1} & 1 \\ x_{j,2} & 1 \end{bmatrix} \, , \tag{45b}$$

where $[x_{j,0}, y_{j,0}]^T$, $[x_{j,1}, y_{j,1}]^T$, and $[x_{j,2}, y_{j,2}]^T$ are the counterclockwise-ordered vertices of the n_i platelet triangles associated with vertex $\boldsymbol{v}_i$, see Fig. 9.

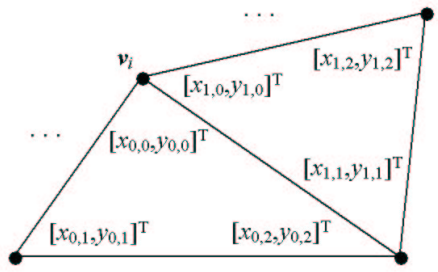

Fig. 9. Indexing scheme for platelet vertices relative to $\boldsymbol{v}_i$ in bivariate case (neighboring triangles oriented counterclockwise)

Other required values are

$$\int_{T_{i,j}} f_{ix}(x,y) f_{jx}(x,y) dxdy \tag{46a}$$

$$= \frac{1}{2} \sum_{k=0}^{n_{i,j}-1} \frac{1}{|J_k|} \det \begin{bmatrix} 1 & y_{k,1} \\ 1 & y_{k,2} \end{bmatrix} \det \begin{bmatrix} 1 & y_{k,0} \\ 1 & y_{k,1} \end{bmatrix} \quad \text{and} \tag{46b}$$

$$\int_{T_{i,j}} f_{iy}(x,y) f_{jy}(x,y) dxdy \tag{46c}$$

$$= \frac{1}{2} \sum_{k=0}^{n_{i,j}-1} \frac{1}{|J_k|} \det \begin{bmatrix} x_{k,1} & 1 \\ x_{k,2} & 1 \end{bmatrix} \det \begin{bmatrix} x_{k,0} & 1 \\ x_{k,1} & 1 \end{bmatrix}, \tag{46d}$$

where $n_{i,j}$ is the number of common platelet triangles and $[x_{k,0}, y_{k,0}]^T$, $[x_{k,1}, y_{k,1}]^T$, and $[x_{k,2}, y_{k,2}]^T$ are the vertices of a common triangle of the platelets of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, see Fig. 10.

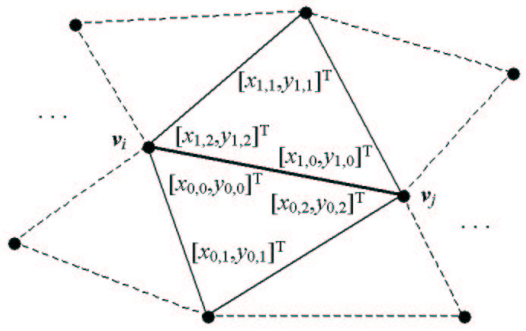

Fig. 10. Indexing scheme for platelet vertices relative to $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ in bivariate case (neighboring triangles oriented counterclockwise)

An example of a bivariate approximation using first-derivative information is shown in Fig. 11. A checkerboard function was digitized to a 100×100 grid to which a linear spline was fit. The approximations were computed for this spline using the first-derivative information and the finite-element approach described in Sect. 5. It is obvious in this example that the first derivative affects the over- and under-shoots significantly. As more weight is added to the first-derivative information, the better the approximation becomes.

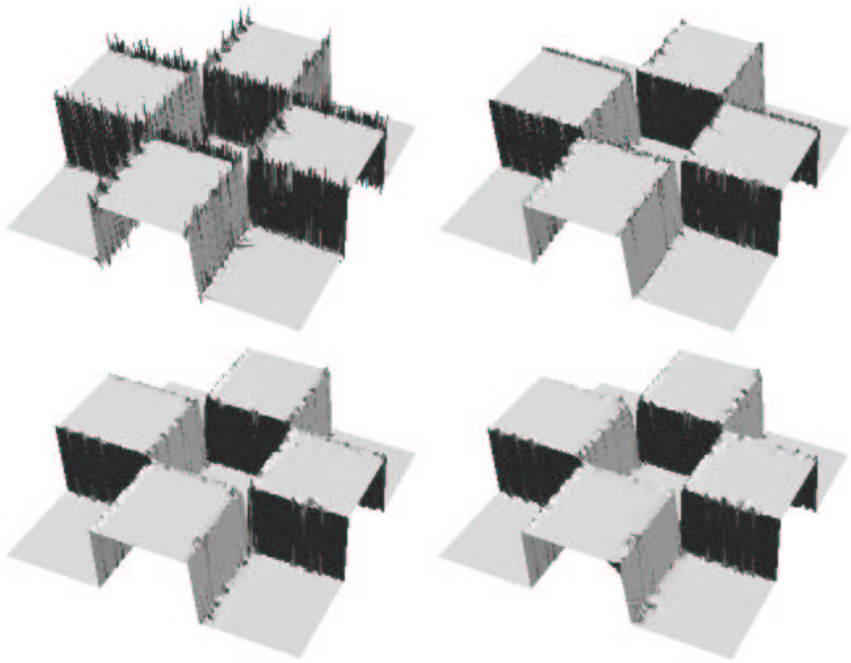

Fig. 11. Four approximations of bivariate checkerboard function with varying weights $(w_{0,0}, w_{1,0}, w_{0,1})$: $(1,0,0)$, $\left(\frac{3}{4}, \frac{1}{8}, \frac{1}{8}\right)$, $\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$, and $\left(\frac{1}{4}, \frac{3}{8}, \frac{3}{8}\right)$, from upper-left to lower-right corner; number of knots varies between 5000 and 6000

6.3 The Trivariate Case

The error functional we minimize in the trivariate setting is given by

$$E = \int_T \sum_{\substack{i,\,j,\,k \geq 0 \\ i+j+k \leq 1}} \left(\frac{\partial^{i+j+k}}{\partial x^i \partial y^j \partial z^k} \left(F(x,y,z) - f(x,y,z) \right) \right)^2 dxdydz \ , \quad (47)$$

which, in matrix notation, corresponds to minimizing

$$
\begin{aligned}
E = \int_T \frac{1}{2} & \left[f(x,y,z)\ f_x(x,y,z)\ f_y(x,y,z)\ f_z(x,y,z) \right] \\
& \begin{bmatrix} 2w_{0,0,0} & 0 & 0 & 0 \\ 0 & 2w_{1,0,0} & 0 & 0 \\ 0 & 0 & 2w_{0,1,0} & 0 \\ 0 & 0 & 0 & 2w_{0,0,1} \end{bmatrix} \begin{bmatrix} f(x,y,z) \\ f_x(x,y,z) \\ f_y(x,y,z) \\ f_z(x,y,z) \end{bmatrix} \\
& - \left[f(x,y,z)\ f_x(x,y,z)\ f_y(x,y,z)\ f_z(x,y,z) \right] \\
& \begin{bmatrix} 2w_{0,0,0}F(x,y,z) \\ 2w_{1,0,0}F_x(x,y,z) \\ 2w_{0,1,0}F_y(x,y,z) \\ 2w_{0,0,1}F_z(x,y,z) \end{bmatrix} dxdydz\,. \qquad (48)
\end{aligned}
$$

Substituting (3) into (48) yields, formally, the same equations one obtains for the univariate and bivariate cases. Regarding (31), the elements $a_{i,j}$ of the symmetric, positive definite matrix $\boldsymbol{A}$ are given by

$$
\begin{aligned}
a_{i,j} = \quad & w_{0,0,0} \int_T f_i(x,y,z) f_j(x,y,z) dxdydz \\
& + w_{1,0,0} \int_T f_{ix}(x,y,z) f_{jx}(x,y,z) dxdydz \\
& + w_{0,1,0} \int_T f_{iy}(x,y,z) f_{jy}(x,y,z) dxdydz \\
& + w_{0,0,1} \int_T f_{iz}(x,y,z) f_{jz}(x,y,z) dxdydz\ , \\
& i,j = 0,\ldots,n-1\ . \qquad (49)
\end{aligned}
$$

The elements l_i of the column vector $\boldsymbol{l}$ are given by

$$
\begin{aligned}
l_i = \quad & w_{0,0,0} \int_T F(x,y,z) f_j(x,y,z) dxdydz \\
& + w_{1,0,0} \int_T F_x(x,y,z) f_{jx}(x,y,z) dxdydz \\
& + w_{0,1,0} \int_T F_y(x,y,z) f_{jy}(x,y,z) dxdydz \\
& + w_{0,0,1} \int_T F_z(x,y,z) f_{jz}(x,y,z) dxdydz\ , \\
& i,j = 0,\ldots,n-1\ . \qquad (50)
\end{aligned}
$$

Integral values required to compute the matrix elements $a_{i,j}$ are

$$
\int_{T_i} \left(f_i(x,y,z) \right)^2 dxdydz = \frac{1}{60} \sum_{j=0}^{n_i-1} |J_i|\ , \qquad (51)
$$

where n_i is the number of platelet tetrahedra, T_i, associated with vertex $\boldsymbol{v}_i$ and J_i is the Jacobian, given by (20), associated with the j^{th} platelet tetrahedron. Two basis functions $f_i(x,y)$ and $f_j(x,y)$ whose associated vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are connected by an edge imply the non-zero integral value

$$\int_{T_{i,j}} f_i(x,y,z) f_j(x,y,z) dxdydz = \frac{1}{120} \sum_{k=0}^{n_{i,j}-1} |J_k| \ , \tag{52}$$

where $T_{i,j}$ is the set of tetrahedra in common between the platelets of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. The linear polynomial interpolating the values one, zero, zero, and zero at the vertices $[x_0, y_0, z_0]^T$, $[x_1, y_1, z_1]^T$, $[x_2, y_2, z_2]^T$, and $[x_3, y_3, z_3]^T$, respectively, has the partial derivatives

$$f_x(x,y,z) = -\frac{1}{J} \det \begin{bmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{bmatrix} \ , \tag{53a}$$

$$f_y(x,y,z) = -\frac{1}{J} \det \begin{bmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{bmatrix} \ , \text{ and} \tag{53b}$$

$$f_z(x,y,z) = -\frac{1}{J} \det \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \ . \tag{53c}$$

Integrals involving these partial derivatives are

$$\int_{T_i} \left(f_{ix}(x,y,z)\right)^2 dxdydz = \frac{1}{6} \sum_{j=0}^{n_i-1} \frac{1}{|J_j|} \det^2 \begin{bmatrix} 1 & y_{j,1} & z_{j,1} \\ 1 & y_{j,2} & z_{j,2} \\ 1 & y_{j,3} & z_{j,3} \end{bmatrix} \ , \tag{54a}$$

$$\int_{T_i} \left(f_{iy}(x,y,z)\right)^2 dxdydz = \frac{1}{6} \sum_{j=0}^{n_i-1} \frac{1}{|J_j|} \det^2 \begin{bmatrix} x_{j,1} & 1 & z_{j,1} \\ x_{j,2} & 1 & z_{j,2} \\ x_{j,3} & 1 & z_{j,3} \end{bmatrix} \ , \text{ and} \tag{54b}$$

$$\int_{T_i} \left(f_{iz}(x,y,z)\right)^2 dxdydz = \frac{1}{6} \sum_{j=0}^{n_i-1} \frac{1}{|J_j|} \det^2 \begin{bmatrix} x_{j,1} & y_{j,1} & 1 \\ x_{j,2} & y_{j,2} & 1 \\ x_{j,3} & y_{j,3} & 1 \end{bmatrix} \ , \tag{54c}$$

where the vertices $[x_{j,1}, y_{j,1}, z_{j,1}]^T$, $[x_{j,2}, y_{j,2}, z_{j,2}]^T$, and $[x_{j,3}, y_{j,3}, z_{j,3}]^T$ denote the boundary vertices of the faces of the platelet tetrahedra associated with vertex $[x_i, y_i, z_i]^T$. Other required values are

$$\begin{aligned} &\int_{T_{i,j}} f_{ix}(x,y,z) f_{jx}(x,y,z) dxdydz \\ &\qquad = \frac{1}{6} \sum_{k=0}^{n_{i,j}-1} \frac{1}{|J_k|} \det \begin{bmatrix} 1 & y_{k,1} & z_{k,1} \\ 1 & y_{k,2} & z_{k,2} \\ 1 & y_{k,3} & z_{k,3} \end{bmatrix} \det \begin{bmatrix} 1 & y_{k,0} & z_{k,0} \\ 1 & y_{k,3} & z_{k,3} \\ 1 & y_{k,2} & z_{k,2} \end{bmatrix} \ , \end{aligned} \tag{55a}$$

$$\int_{T_{i,j}} f_{iy}(x,y,z) f_{jy}(x,y,z) dxdydz$$

$$= \frac{1}{6} \sum_{k=0}^{n_{i,j}-1} \frac{1}{|J_k|} \det \begin{bmatrix} x_{k,1} & 1 & z_{k,1} \\ x_{k,2} & 1 & z_{k,2} \\ x_{k,3} & 1 & z_{k,3} \end{bmatrix} \det \begin{bmatrix} x_{k,0} & 1 & z_{k,0} \\ x_{k,3} & 1 & z_{k,3} \\ x_{k,2} & 1 & z_{k,2} \end{bmatrix}, \text{ and (55b)}$$

$$\int_{T_{i,j}} f_{iz}(x,y,z) f_{jz}(x,y,z) dxdydz$$

$$= \frac{1}{6} \sum_{k=0}^{n_{i,j}-1} \frac{1}{|J_k|} \det \begin{bmatrix} x_{k,1} & y_{k,1} & 1 \\ x_{k,2} & y_{k,2} & 1 \\ x_{k,3} & y_{k,3} & 1 \end{bmatrix} \det \begin{bmatrix} x_{k,0} & y_{k,0} & 1 \\ x_{k,3} & y_{k,3} & 1 \\ x_{k,2} & y_{k,2} & 1 \end{bmatrix}, \qquad (55c)$$

where $n_{i,j}$ is the number of common platelet tetrahedra and $[x_{k,0}, y_{k,0}, z_{k,0}]^T$, $[x_{k,1}, y_{k,1}, z_{k,1}]^T$, $[x_{k,2}, y_{k,2}, z_{k,2}]^T$, and $[x_{k,3}, y_{k,3}, z_{k,3}]^T$ are the vertices of a common tetrahedron of the platelets of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$.

An example of a trivariate approximation using first-derivative information is shown in Fig. 12. A checkerboard function (the trivariate generalization of the bivariate function) was digitized to a $100 \times 100 \times 100$ grid to which a linear spline was fit. The approximations were computed for this spline. Again, the use of first derivative affects approximation quality substantially. Fig. 12 shows a flat-shaded slice through the approximations. As more weight is added to the first derivative, the discontinuities (abrupt changes from zero to one and vice versa) are captured much better.

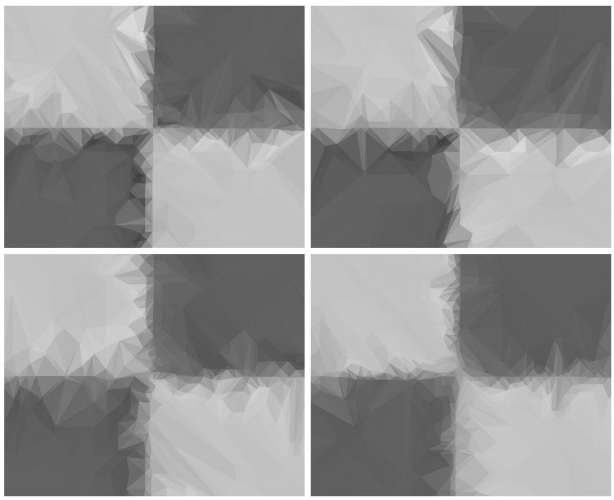

Fig. 12. Four approximations of trivariate checkerboard function with varying weights $(w_{0,0,0}, w_{1,0,0}, w_{0,1,0}, w_{0,0,1})$: $(1,0,0,0)$, $\left(\frac{3}{4}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}\right)$, $\left(\frac{1}{2}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right)$, and $\left(\frac{1}{4}, \frac{3}{12}, \frac{3}{12}, \frac{3}{12}\right)$, from upper-left to lower-right corner; number of knots varies between 1400 and 3000

7 Quadratic Simplices

As an alternative to using linear simplices, we show how to use *curved simplices* to better approximate data. Most scientific data sets contain discontinuities – such as a car body geometry or a pressure field discontinuity – that can often be represented much better with curved elements. Discontinuous data sets can often be approximated better by dividing their domains into several smaller domains with curved boundaries. A combination of geometry/boundary and dependent field variable discontinuities can be treated in an integrated fashion. The discussion in this section only treats the bivariate and trivariate cases.

The algorithms that we have described in the previous sections still apply: we begin with a coarse initial triangulation, which may now contain curved simplices, and repeatedly refine this mesh until a global error tolerance is met.

7.1 Mapping the standard simplex

In the bivariate case, we map the standard triangle, see Sect. 3, to a curved triangular region in physical space by mapping the six knots $\boldsymbol{u}_i = [u_{i,j}, v_{i,j}]^T = \left(\frac{i}{2}, \frac{j}{2}\right),\ i,j \geq 0,\ i+j \leq 2$ (abbreviated in multi-index notation as $|\boldsymbol{i}| = 2$) in parameter space to six corresponding vertices $\boldsymbol{x}_i = [x_{i,j}, y_{i,j}]^T$ in physical space, using a quadratic mapping. The quadratic mapping, using Bernstein-Bézier polynomials $B^2_{\boldsymbol{i}}(\boldsymbol{u})$ as basis functions, see [8] and [26], is given by

$$\boldsymbol{x}(\boldsymbol{u}) = \begin{bmatrix} x(u,v) \\ y(u,v) \end{bmatrix} = \sum_{|\boldsymbol{i}|=2} \boldsymbol{b}_{\boldsymbol{i}} B^2_{\boldsymbol{i}}(\boldsymbol{u}) = \begin{bmatrix} \sum_{|\boldsymbol{i}|=2} c_{i,j} B^2_{i,j}(u,v) \\ \sum_{|\boldsymbol{i}|=2} d_{i,j} B^2_{i,j}(u,v) \end{bmatrix} \tag{56}$$

In the same manner, we define the mapping of the standard tetrahedron, described in Sect. 4, to a curved tetrahedron in physical space, mapping ten knots $\boldsymbol{u}_i = [u_{i,j,k}, v_{i,j,k}, w_{i,j,k}]^T = \left(\frac{i}{2}, \frac{j}{2}, \frac{k}{2}\right),\ |\boldsymbol{i}| = 2$ in parameter space, to ten corresponding vertices $\boldsymbol{x}_i = [x_{i,j,k}, y_{i,j,k}, z_{i,j,k}]^T$ in physical space. The quadratic mapping in the trivariate case is given by

$$\begin{aligned} \boldsymbol{x}(\boldsymbol{u}) &= \begin{bmatrix} x(u,v,w) \\ y(u,v,w) \\ z(u,v,w) \end{bmatrix} = \sum_{|\boldsymbol{i}|=2} \boldsymbol{b}_{\boldsymbol{i}} B^2_{\boldsymbol{i}}(\boldsymbol{u}) \\ &= \begin{bmatrix} \sum_{|\boldsymbol{i}|=2} c_{i,j,k} B^2_{i,j,k}(u,v,w) \\ \sum_{|\boldsymbol{i}|=2} d_{i,j,k} B^2_{i,j,k}(u,v,w) \\ \sum_{|\boldsymbol{i}|=2} e_{i,j,k} B^2_{i,j,k}(u,v,w) \end{bmatrix} . \end{aligned} \tag{57}$$

7.2 Initial Simplicial Decomposition

The original grid, its boundaries, and possibly known locations of field discontinuities (in the dependent field variables) influence how we construct an

initial simplicial decomposition. We consider domain boundaries and known discontinuities of interest to define a set of vertices and a set of edges (possibly curved) connecting these vertices. (We do not discuss in this paper how to obtain these vertices or edges.) With this information, one can define the unique fields and sub-regions in the overall domain that are bounded by these edges. If the edges specified are linear spline curves, a quadratic curve fitting step must take place to approximate the linear segments by quadratic curves. Each of the sub-domains, bounded by curved edges, can then be triangulated to form an initial mesh, see Fig. 13. Approximation is then performed for each sub-domain independently. It is possible that (quadratic) simplices share knot/vertex locations on field boundaries. In this case, since an approximation is computed independently for each sub-domain, there exist two separate coefficients for the same location.

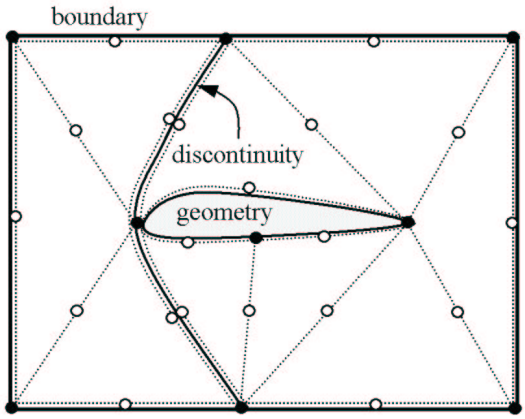

Fig. 13. Decomposition of domain around a wing using bivariate curved simplices. There are two distinct fields defined by the vertices and edges in this example: the field left of the discontinuity and the field right of the discontinuity

Bisection of curved edges of mesh simplices is performed in a manner similar to the linear case: we bisect at the midpoint of the arc and insert a knot at this location. The finite-element approach based on a Sobolev-like norm can be applied, as described in Sect. 5.

7.3 Best Approximation for Quadratic Simplices

We denote basis functions associated with simplex vertex $\boldsymbol{v}_i$ by $f_i(\boldsymbol{x})$. The basis function corresponding to edge e_j of a simplex is denoted by $g_j(\boldsymbol{x})$. The best approximation is denoted by $f(\boldsymbol{x})$, and we write it as a linear combination of the basis functions associated with the simplex corners and edges. Assuming that there are n distinct corner vertices and m distinct edges, we can write the best approximation as

$$f(\boldsymbol{x}) = \sum_{i=0}^{n-1} c_i f_i(\boldsymbol{x}) + \sum_{j=0}^{m-1} d_j g_j(\boldsymbol{x}). \tag{58}$$

In matrix form, the normal equations are

$$\begin{bmatrix} \langle f_0, f_0\rangle & \cdots & \langle f_0, f_{n-1}\rangle & \langle f_0, g_0\rangle & \cdots & \langle f_0, g_{m-1}\rangle \\ \vdots & & \vdots & \vdots & & \vdots \\ \langle f_{n-1}, f_0\rangle & \cdots & \langle f_{n-1}, f_{n-1}\rangle & \langle f_{n-1}, g_0\rangle & \cdots & \langle f_{n-1}, g_{m-1}\rangle \\ \langle g_0, f_0\rangle & \cdots & \langle g_0, f_{n-1}\rangle & \langle g_0, g_0\rangle & \cdots & \langle g_0, g_{m-1}\rangle \\ \vdots & & \vdots & \vdots & & \vdots \\ \langle g_{m-1}, f_0\rangle & \cdots & \langle g_{m-1}, f_{n-1}\rangle & \langle g_{m-1}, g_0\rangle & \cdots & \langle g_{m-1}, g_{m-1}\rangle \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_{n-1} \\ d_0 \\ \vdots \\ d_{m-1} \end{bmatrix}$$
$$= \begin{bmatrix} \langle F, f_0\rangle \\ \vdots \\ \langle F, f_{n-1}\rangle \\ \langle F, g_0\rangle \\ \vdots \\ \langle F, g_{m-1}\rangle \end{bmatrix} . \tag{59}$$

The Bernstein-Bézier quadratic basis functions are defined as

$$B^2_{\boldsymbol{i}}(\boldsymbol{u}) = \frac{2!}{(2-i-j)!i!j!}(1-u-v)^{2-i-j}u^i v^j \quad \text{and} \tag{60a}$$

$$B^2_{\boldsymbol{i}}(\boldsymbol{u}) = \frac{2!}{(2-i-j-k)!i!j!k!}(1-u-v-w)^{2-i-j-k}u^i v^j w^k \tag{60b}$$

in the bivariate and trivariate cases, respectively. Again, we can use of the change-of-variables theorem and implement error estimates as described in Sects. 3 and 4 to compute a best approximation. The needed inner products, defined over the standard simplex in parameter space, are given by

$$\langle B_{i,j}, B_{k,l}\rangle = \frac{1}{180}\begin{bmatrix} 6&3&1&3&1&1 \\ 3&4&3&2&2&1 \\ 1&3&6&1&3&1 \\ 3&2&1&4&2&3 \\ 1&2&3&2&4&3 \\ 1&1&1&3&3&6 \end{bmatrix} \tag{61}$$

in the bivariate case, and

$$\langle B_{i,j,k}, B_{l,m,n}\rangle = \frac{1}{1260}\begin{bmatrix} 6&3&1&3&1&1&3&1&1&1 \\ 3&4&3&2&2&1&2&2&1&1 \\ 1&3&6&1&3&1&1&3&1&1 \\ 3&2&1&4&2&3&2&1&2&1 \\ 1&2&3&2&4&3&1&2&2&1 \\ 1&1&1&3&3&6&1&1&3&1 \\ 3&2&1&2&1&1&4&2&2&3 \\ 1&2&3&1&2&1&2&4&2&3 \\ 1&1&1&2&2&3&2&2&4&3 \\ 1&1&1&1&1&1&3&3&3&6 \end{bmatrix} \tag{62}$$

in the trivariate case.

8 Conclusions and Future Work

We have discussed a best linear spline approximation scheme and several enhancements to improve and generalize. To adapt our method to multi-valued data, one can, at each iteration, approximate each dependent variable separately.

In terms of computational efficiency, large linear systems are produced when dealing with vary large data sets. This is manageable, since the resulting systems are sparse and can easily be treated with sparse-system solvers.

Investigating the effectiveness of curved elements for approximating data is a topic of our current research. With the use of curved elements growing, we believe that using curved elements in approximations, and supporting them directly during rendering, will become an important visualization research area.

9 Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48. This work was also supported by the National Science Foundation under contract ACI 9624034 (CAREER Award), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, and through the National Partnership for Advanced Computational Infrastructure (NPACI); the Office of Naval Research under contract N00014-97-1-0222; the Army Research Office under contract ARO 36598-MA-RIP; the NASA Ames Research Center through an NRA award under contract NAG2-1216; the Lawrence Livermore National Laboratory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159; and the North Atlantic Treaty Organization (NATO) under contract CRG.971628 awarded to the University of California, Davis. We also acknowledge the support of ALSTOM Schilling Robotics and SGI. We thank the members of the Visualization and Graphics Research Group at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California, Davis.

References

1. Boehm, W. and Prautzsch, H. (1993), Numerical Methods, A K Peters, Ltd., Wellesley, MA
2. Bonneau, G. P., Hahmann, S. and Nielson, G. M. (1996), BLAC-wavelets: A multiresolution analysis with non-nested spaces. In: Yagel, R. and Nielson, G. M. (Eds.) Visualization '96, IEEE Computer Society Press, Los Alamitos, CA, 43–48

3. Cignoni, P., De Floriani, L., Montani, C., Puppo, E. and Scopigno, R. (1994), Multiresolution modeling and visualization of volume data based on simplicial complexes. In: Kaufman, A. E. and Krüger, W. (Eds.) 1994 Symposium on Volume Visualization, IEEE Computer Society Press, Los Alamitos, CA, 19–26
4. Cignoni, P., Montani, C., Puppo, E., and Scopigno, R. (1997), Multiresolution representation and visualization of volume data, IEEE Transactions of Visualization and Computer Graphics **3(4)**, 352–369
5. Davis, P. J. (1975) Interpolation and Approximation. New York: Dover
6. Duff, I. S., Erisman, A. M., and Reid, J. K. (1986), Direct Methods for Sparse Matrices, Clarendon Press, Oxford, England, 239–251
7. Dyn, N., Floater, M. S., and Iske, A. (2000), Adaptive thinning for bivariate scattered data, Technische Universität München, Fakultät für Mathematik, München, Germany, Report TUM M0006, 2000.
8. Farin, G. (2001), Curves and Surfaces for Computer Aided Geometric Design, fifth edition, Academic Press, San Diego, CA
9. Floater, M. S. and Reimers, M. (2001), Meshless parameterization and surface reconstruction, Computer Aided Geometric Design **18**, 77–92
10. Franke, R. (1982), Scattered data interpolation: Tests of some methods, Math. Comp. **38**, 181–200
11. Gieng, T. S., Hamann, B., Joy, K. I., Schussman, G. L. and Trotts, I. J. (1997), Smooth hierarchical surface triangulations. In: Yagel, R. and Hagen, H. (Eds.) Visualization '97, IEEE Computer Society Press, Los Alamitos, CA, 379–386
12. Gieng, T. S., Hamann, B., Joy, K. I., Schussman, G. L. and Trotts, I. J. (1998), Constructing hierarchies for triangle meshes, IEEE Transactions on Visualization and Computer Graphics **4(2)**, 145–161
13. Gross, M. H., Gatti, R. and Staadt, O. (1995), Fast multiresolution surface meshing. In: Nielson, G. M. and Silver, D. (Eds.) Visualization '95, IEEE Computer Society Press, Los Alamitos, CA, 135–142
14. Grosso, R., Lürig, C. and Ertl, T. (1997), The multilevel finite element method for adaptive mesh optimization and visualization of volume data. In: Yagel, R. and Hagen, H. (Eds.) Visualization '97, IEEE Computer Society Press, Los Alamitos, CA, 387–394
15. Hagen, H., Müller, H. and Nielson, G. M. (Eds.) (1993), Focus on Scientific Visualization, Springer-Verlag, New York, NY
16. Hamann, B. (1994), A data reduction scheme for triangulated surfaces, Computer Aided Geometric Design **11(2)**, Elsevier, 197–214
17. Hamann, B., Jordan, B. W. and Wiley, D. F. (1999), On a construction of a hierarchy of best linear spline approximations using repeated bisection, IEEE Transactions on Visualization and Computer Graphics **5(1)**, 30–46, and **5(2)**, 190 (errata)
18. Hardy, R. L. (1971), Multiquadric equations of topography and other irregular surfaces, Journal of Geophysical Research **76**, 1905–1915
19. Heckbert, P. S. and Garland, M. (1997), Survey of polygonal surface simplification algorithms, Technical Report, Computer Science Department, Carnegie Mellon University, Pittsburg, Pennsylvania, to appear
20. Hoppe, H. (1996), Progressive meshes. In: Rushmeier, H. (Ed.) Proceedings of SIGGRAPH 1996, ACM Press, New York, NY, 99–108
21. Hoppe, H. (1997), View-dependent refinement of progressive meshes. In: Whitted, T. (Ed.) Proceedings of SIGGRAPH 1997, ACM Press, New York, NY, 189–198

22. Kaufman, A. E. (Ed.) (1991), Volume Visualization, IEEE Computer Society Press, Los Alamitos, CA
23. Nielson, G. M. (1993), Scattered data modeling, IEEE Computer Graphics and Applications **13(1)**, 60–70
24. Nielson, G. M., Müller, H. and Hagen, H. (Eds.) (1997), Scientific Visualization: Overviews, Methodologies, and Techniques, IEEE Computer Society Press, Los Alamitos, CA
25. Nielson, G. M. and Shriver B. D. (Eds.) (1990), Visualization in Scientific Computing, IEEE Computer Society Press, Los Alamitos, CA
26. Piegl, L. A. and Tiller, W. (1996), The NURBS Book, second edition, Springer-Verlag, New York, NY
27. Rippa, S. (1992), Long and thin triangles can be good for linear interpolation, SIAM J. Numer. Anal. **29(1)**, 257–270
28. Rosenblum, L. J, Earnshaw, R. A., Encarnação, J. L., Hagen, H., Kaufman, A. E., Klimenko, S., Nielson, G. M., Post, F. and Thalmann, D. (Eds.) (1994), Scientific Visualization—Advances and Challenges, IEEE Computer Society Press, Los Alamitos, CA
29. Staadt, O. G., Gross, M. H. and Weber, R. (1997), Multiresolution compression and reconstruction. In: Yagel, R. and Hagen, H. (Eds.) Visualization '97, IEEE Computer Society Press, Los Alamitos, CA, 337–346