Triangulations from Repeated Bisection

Bernd Hamann and Benjamin W. Jordan

Abstract. We present a method for the iterative refinement of triangulations. Given a coarse triangulation of the compact domain of a bivariate function, we present a refinement strategy based on approximation error. The triangulation is used to compute a best linear spline approximation, using the term best approximation in an integral least squares sense. We improve an approximation by identifying the triangle with largest error and refine the triangulation by bisecting this triangle.

§1. Introduction

In the context of visualizing very large data sets, it is imperative to use hierarchical data representations that allow us to study physical phenomena at multiple levels of detail. General, robust, and efficient methodologies are needed to support hierarchical data representation and visualization.

Triangulations are a natural choice when complicated regions in two – or three – dimensions must be represented. We present a construction of hierarchies of triangulations, constructed as best linear spline approximations. The basic idea is repreated bisection. The coefficients associated with each vertex in a triangulation are computed in a best approximation sense. Whenever one bisects a triangle one needs to compute new spline coefficients for all vertices. One can perform the matrix inversions efficiently due to the fact that the matrices are very sparse. This sparseness allows us to reduce matrix bandwidth significantly. The main principles of our approach become evident from the discussion of the univariate case.

Three main objectives have influenced the design of our construction. *The construction should be simple.* The number of special cases to be considered should be small; a refinement step should cause minimal topological change; and the computation of a best linear spline approximation should be robust. *The construction should be general.* The approach should be applicable to multivariate and multi-valued functions as well; and the approach should handle arbitrary, complex-shaped domains. *The construction should*

Mathematical Methods for Curves and Surfaces II
Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds.), pp. 229–236.

ISBN 0-8265-1315-8.

be efficient. The computation of the triangulation hierarchy should be reasonably efficient. The single steps of our construction are:

1) **Initial approximation.** Define an initial, coarse triangulation of the function's domain and, for all vertices, compute the spline coefficients defining the initial best linear spline approximation.

2) **Error estimation.** Compute appropriate global error and local error (=triangle-specific error) estimates.

3) **Bisection.** Identify the triangle with largest local error and bisect it by splitting its longest edge into two segments – thereby also bisecting all triangles sharing this edge.

4) **Computation of new approximation.** Based on the new vertex set and new triangulation, compute a new best linear spline approximation. Iterate steps 2), 3) and 4) until a termination condition is met.

Remark 1.1. We assume that the function to be approximated is known analytically. Should this not be the case, one needs to first determine such a function, possibly from a finite set of scattered data.

From the set of all best linear spline approximations of F we select a subset consisting of those that we can associate with a particular level in an approximation hierarchy. What does the notion of "level" mean? Given an error tolerance tuple $\mathbf{E} = (\epsilon_0, ..., \epsilon_{m-1})$, $\epsilon_j > \epsilon_{j+1}$, we call an approximation a level-j approximation when its global error (see below) lies in the interval $(\epsilon_{j+1}, \epsilon_j]$. A spline approximation associated with level j is called the representative of level j if its number of knots is minimal among all level-j approximations.

§2. The Univariate Case

Our construction of best approximations requires a few, simple notions from linear algebra and approximation theory. We use an interval-weighted scalar product $\langle f, g \rangle$ for two functions $f(x)$ and $g(x)$, defined over $[a, b]$,

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

and an interval-weighted L^2 norm to measure a function $f(x)$,

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

We include the factor $\frac{1}{b-a}$ to eliminate the influence of interval length when computing interval-specific error estimates for the approximation process.

This ensures "normalization" – we are using *error per unit* to determine which segment to bisect next.

The best approximation of a function F, when approximating it by a linear combination $\sum_{i=0}^{n-1} c_i f_i$ of independent functions $f_0, ..., f_{n-1}$, is defined by the solution of the normal equations

$$\begin{pmatrix} \langle f_0, f_0 \rangle & \cdots & \langle f_{n-1}, f_0 \rangle \\ \vdots & & \vdots \\ \langle f_0, f_{n-1} \rangle & \cdots & \langle f_{n-1}, f_{n-1} \rangle \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} \langle F, f_0 \rangle \\ \vdots \\ \langle F, f_{n-1} \rangle \end{pmatrix}, \tag{3}$$

see, [4]. We will also write this linear system as $M^{[n-1]}\mathbf{c}^{[n-1]} = \mathbf{F}^{[n-1]}$. For an arbitrary set of basis functions f_i one has to investigate means for the efficient solution of this system.

We are concerned with the construction of a hierarchy of best linear spline approximations to a function F by increasing the number of basis functions – or, in other words, the number of knots. We increase the number of knots, generally one-by-one, until a best approximation is obtained whose associated error is smaller than some threshold.

In general, the computation of a best linear spline approximation to a given function F is an optimization problem, which, when allowing variable knots, is quite involved, see [2]. Our current approach does not permit variable knots, it is based on repeated bisection using the midpoints of intervals. This approach is computationally less expensive than the general optimization problem. In each step, we simply determine the interval with largest error and bisect it. Bisection implies the insertion of the midpoint as a new new knot.

We start by approximating a function F using a single linear segment by computing $M^{[1]}\mathbf{c}^{[1]} = \mathbf{F}^{[1]}$. We assume that F is defined over $[0, 1]$ and that the basis functions f_i are hat functions, *i.e.*, $f_i(x_j) = \delta_{i,j}$ (see Fig. 1). The initial error is $E^{[1]} = ||F - (c_0 f_0 + c_1 f_1)||$, and if this value is larger than the given threshold we insert an additional knot at $x = \frac{1}{2}$. (Inserting this additional knot changes the knot sequence, and, consequently, one obtains three new hat basis functions.) Thus, the next best approximation, using the updated knot sequence, is obtained by solving $M^{[2]}\mathbf{c}^{[2]} = \mathbf{F}^{[2]}$, and the new error value is given by $E^{[2]} = \left|\left|F - \sum_{i=0}^{2} c_i f_i\right|\right|$. Since our method is based on repeated bisection, we compute segment-specific, local errors for the intervals $\left[0, \frac{1}{2}\right]$ and $\left[\frac{1}{2}, 1\right]$ which determine the segment to be bisected next.

Assuming that an intermediate approximation is based on the knot sequence $0 = x_0 < x_1 < x_2 < \cdots < x_{k-2} < x_{k-1} = 1$ and that its coefficient vector is $(c_0, ..., c_{k-1})^T$, we define the global error as

$$E^{[k-1]} = \left|\left| F - \sum_{i=0}^{k-1} c_i f_i \right|\right| \tag{4}$$

and local (segment-specific) errors as

$$e_i^{[k-1]} = \left(\frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} \left(F - (c_i f_i + c_{i+1} f_{i+1})\right)^2 dx \right)^{1/2}, \quad i = 0, ..., k-2. \tag{5}$$

We compute the local error values for each segment and bisect the segment with largest local error value. If there are multiple segments with the same maximal error value, we randomly pick one of them to be bisected. (It is of course possible to bisect all segments with the same maximal error at the same time, thus leading to a unique answer.)

When using hat functions as basis functions, the only non-zero elements of $M^{[k-1]}$, for a particular row i, are the elements $\langle f_{i-1}, f_i \rangle = \triangle_{i-1}/6$, $\langle f_i, f_i \rangle = (\triangle_{i-1} + \triangle_i)/3$, and $\langle f_{i+1}, f_i \rangle = \triangle_i/6$, where $\triangle_j = x_{j+1} - x_j$. Thus, $M^{[k-1]}$ is the tridiagonal matrix

$$M^{[k-1]} = \frac{1}{6} \begin{pmatrix} 2\triangle_0 & \triangle_0 & & & & \\ \triangle_0 & 2(\triangle_0 + \triangle_1) & \triangle_1 & & & \\ & \triangle_1 & 2(\triangle_1 + \triangle_2) & \triangle_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \triangle_{k-2} & 2\triangle_{k-2} & \end{pmatrix}. \quad (6)$$

It is necessary, due to the global nature of the problem, to re-compute all components of the coefficient vectors $(c_0, c_1, c_2, ...)^T$ whenever one inserts a knot.

Remark 2.1. A drawback, when inserting knots one-by-one, is the fact that the method is not local, *i.e.*, inserting a single knot leads to a new system of normal equations. One can significantly enhance the efficiency of the method by bisecting multiple intervals in one step, thereby refining an approximation in different regions in one step.

§3. The Bivariate Case

We proceed similarly in the bivariate case. Given an initial best linear spline approximation based on a small number of triangles, we compute the global approximation error of the associated best linear spline approximation and, should this error be too large, insert the midpoint of the longest edge of the triangle with largest local error as a new knot. Bisection leads to the split of one or two triangles – depending on whether the bisected edge is shared by a second triangle or not.

Bisection terminates when a certain global error condition is satisfied. If the triangle with largest local error has more than one edge with maximal length, we bisect one of the edges, chosen randomly, with maximal length. (Alternatively, one could split multiple edges simultaneously to guarantee uniqueness.) If multiple triangles share the same maximal local error value, we choose, randomly, one of these triangles to be bisected. (One could choose to bisect larger triangles first – or vice versa – or choose to simultaneously split all triangles sharing the same maximal local error value.)

The union of the triangles in the initial triangulation defines the region to which we apply our refinement scheme to obtain improved approximations of a bivariate function $F(x, y)$. We assume that the boundary of F's domain is approximated well enough by the coarse, initial triangulation.

Some definitions are required for the bivariate setting. Denoting the vertex (knot) set of an intermediate approximation by $\{\mathbf{v}_i = (x_i, y_i)^T | i = 0, ..., k-1\}$ and the coefficient vector associated with the hat basis functions $f_i(x, y)$ by $(c_0, ..., c_{k-1})^T$, the global error is

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

and the triangle-specific, local error for triangle T_j is

$$e_j^{[k-1]} = \left(\frac{1}{\text{area\_of\_}T_j} \int_{T_j} \left(F - \text{spline\_over\_}T_j\right)\right)^2 dxdy \Bigg)^{1/2}, \quad j = 0, ..., n_T - 1, \tag{8}$$

where n_T is the number of triangles.

Considering the m vertices in the initial triangulation, the initial best linear spline approximation is defined by $M^{[m-1]}\mathbf{c}^{[m-1]} = \mathbf{F}^{[m-1]}$. The initial error is $E^{[m-1]} = \left|\left|F - \sum_{i=0}^{m-1} c_i f_i\right|\right|$. If $E^{[m-1]}$ is larger than a prescribed tolerance, we insert the midpoint of the longest edge of the triangle with largest local error as a new knot. We continue to insert knots until an approximation is good enough. In the bivariate setting the local errors are determined by the difference between $F(x, y)$ and an intermediate best linear approximation over each triangle. Fig. 1 illustrates F, hat basis functions, and a best approximation for the univariate and the bivariate cases.

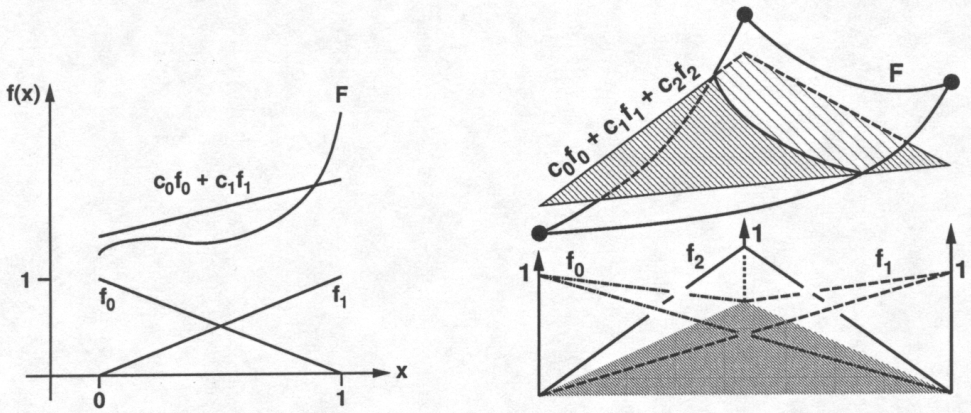

Fig. 1. Basis functions f_i, function F, and F's best approximation.

Considering k knots, the solution of the normal equations requires the inversion of a matrix $M^{[k-1]}$, where the number of non-zero entries per column in this matrix is defined by the valences of the vertices in the triangulation. A vertex $\mathbf{v}_i$ with valence v_i causes $v_i + 1$ non-zero entries in column i. These valence values are not limited, and it is possible to obtain large numbers of non-zero elements in certain columns of $M^{[k-1]}$. Thus, the computation of the coefficients in the bivariate case is much more expensive than in the univariate case.

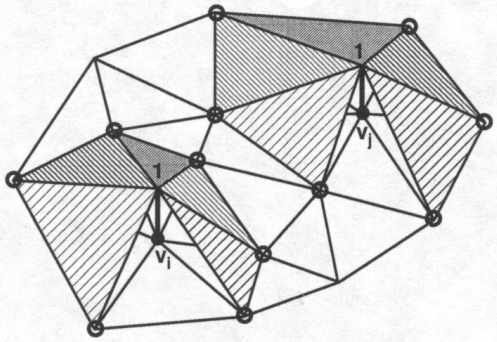

Fig. 2. Platelets of v_i and v_j and associated hat basis functions.

Remark 3.1. Arbitrarily skinny triangles can result from repeated bisection. Therefore, one should incorporate a constraint into the algorithm that prohibits the generation of triangles whose shapes are not acceptable.

The change-of-variables theorem allows us to effectively compute scalar products of hat functions in the bivariate case. A hat function $f_i(x, y)$ associated with vertex $\mathbf{v}_i$ is the spline basis function whose function value is one at $\mathbf{v}_i$ and zero at all other vertices. The function f_i varies linearly between zero and one over the triangles defining $\mathbf{v}_i$'s platelet, see Fig. 2. (The platelet of $\mathbf{v}_i$ is the set of triangles sharing $\mathbf{v}_i$ as a common vertex.)

Applying the change-of-variables theorem, the scalar product $\langle f_i, f_i \rangle$ is given by

$$\langle f_i, f_i \rangle \;=\; \sum_{j=0}^{n_i-1} \int\limits_{T_j} f_i \; f_i \; dxdy \;=\; \frac{1}{12} \sum_{j=0}^{n_i-1} J_j, \tag{9}$$

where n_i is the number of platelet triangles associated with vertex $\mathbf{v}_i$, T_j is the j-th platelet triangle, and J_j is the Jacobian of the j-th platelet triangle,

$$J_j \;=\; \det \begin{pmatrix} x_1^j - x_0^j & x_2^j - x_0^j \\ y_1^j - y_0^j & y_2^j - y_0^j \end{pmatrix}. \tag{10}$$

Here, $\left(x_0^j, y_0^j\right)^T$, $\left(x_1^j, y_1^j\right)^T$, and $\left(x_2^j, y_2^j\right)^T$ are the coordinate vectors of T_j's counterclockwise-oriented vertices. (The platelet of $\mathbf{v}_i$ is the set of triangles $\mathcal{T}_i = \{T_j\}_{j=0}^{n_i-1}$.) The scalar product $\langle f_j, f_k \rangle$ of two basis functions whose associated vertices $\mathbf{v}_j$ and $\mathbf{v}_k$ are connected by an edge is

$$\langle f_j, f_k \rangle \;=\; \sum_{i=0}^{n_{j,k}-1} \int\limits_{T_i} f_j \; f_k \; dxdy \;=\; \frac{1}{24} \sum_{i=0}^{n_{j,k}-1} J_i. \tag{11}$$

(The set $\mathcal{T}_{j,k} = \{T_i\}_{i=0}^{n_{j,k}-1}$ is the union of the triangles in $\mathbf{v}_j$'s and $\mathbf{v}_k$'s platelets.)

Remark 3.2. The matrices $M^{[k-1]}$ are sparse, *i.e.*, relatively few matrix entries are different from zero, and we utilize the Cuthill-McKee algorithm to reduce the bandwidths of the matrices before inverting them, see [3]. A lower

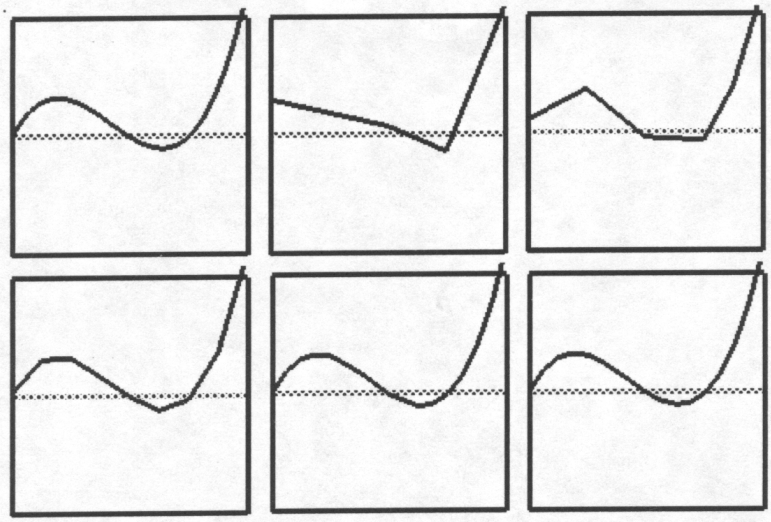

Fig. 3. $F = 10x\left(x - \frac{1}{2}\right)\left(x - \frac{3}{4}\right)$; 4-, 6-, 8-, 14-, and 19-knot approximations.

bound for the bandwidth that one can achieve when applying a bandwidth-reducing strategy is given by the vertex with maximal valence. One might therefore consider imposing a threshold on maximal vertex valence. Efficiency is a concern when performing bisection for a triangulation consisting of a large number of vertices, since each bisection step requires the re-computation of all coefficients. The Cuthill-McKee algorithm, a graph-based vertex indexing scheme, allows one to efficiently insert a new vertex; since the insertion of a vertex is a "local" graph operation one can also efficiently determine the new vertex indices leading to a new small-bandwidth matrix.

§4. Examples

We have tested our approach for univariate and bivariate test functions, using Romberg integration for the computation of the scalar products $\langle F, f_i \rangle$ and the errors, see [1]. We have inserted knots one-by-one.

Figs. 3 and 4 show a univariate and a bivariate example. The upper-left corner image in both figures is an "exact" rendering of the original function. Univariate functions are rendered as graphs $\big(x, f(x)\big)^T$, where the squares in Fig. 3 indicate the region $[0,1] \times [-1,1]$. Bivariate functions are rendered as graphs $\big(x, y, f(x,y)\big)^T$ using flat-shaded surface triangulations, where the cubes in Fig. 4 indicate the region $[0,1] \times [0,1] \times [0,1]$. (The initial triangulation is defined by splitting the unit square into the two triangles obtained by connecting $(0,0)^T$ and $(1,1)^T$. Global approximation errors are based on equations (4) and (7). Using a vector notation, the global errors of the five approximations shown in Fig. 3 are given by $(0.816, 0.241, 0.073, 0.022, 0.010)$, and the errors for the three approximations shown in Fig. 4 are $(0.062, 0.009, 0.001)$.

§5. Conclusions

We have presented a method for the refinement of best linear spline approximations for univariate and bivariate functions. The spline basis functions are

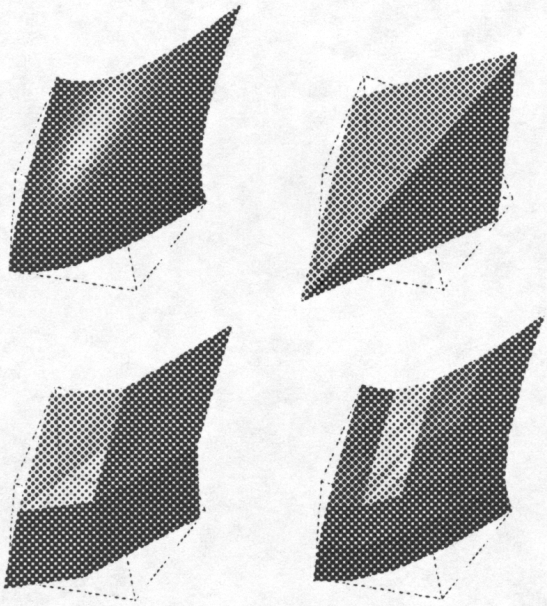

Fig. 4. $F = x^2 + y^2$; 4-, 11-, and 41-knot approximations.

defined over collections of triangles. The approach is sound, robust, general, and fairly efficient. We are extending the scheme to trivariate functions in the context of hierarchical volume visualization. We plan to improve the efficiency of our approach and generalize it to allow more general knot placement. We will extend the approach to multi-valued functions, in particular to vector fields.

Acknowledgments. This work was supported by NSF grants ASC-9210439 (Research Initiation Award) and ASC-9624034 (CAREER Award), by ONR grant N00014-97-1-0222, and by ARO grant ARO-36598-MA-RIP.

References

1. Boehm, W. and H. Prautzsch, *Numerical Methods*, A K Peters, Wellesley, 1993.
2. Cantoni, A., Optimal curve fitting with piecewise linear functions, IEEE Transactions on Computers **20** (1971), 59–67.
3. Cuthill, E. and J. McKee, Reducing the bandwidth of sparse symmetric matrices, in *Proc. ACM National Conference*, Association for Computing Machinery, New York, 1969, 157–172.
4. Davis, P. J., *Interpolation and Approximation*, Dover Publications, New York, 1975.

Bernd Hamann, Co-Director
Center for Image Processing and Integrated Computing (CIPIC)
Department of Computer Science
University of California, Davis
Davis, CA 95616-8562
hamann@cs.ucdavis.edu

Benjamin W. Jordan
Pixar Animation Studios
1001 West Cutting Boulevard
Richmond, CA 94804
bjordan@pixar.com