Computing and Visualization in Science (2005) 8(2): 83–91
(The original publication is available at www.springerlink.com)

Øyvind Hjelle · Morten Dæhlen

Multilevel Least Squares Approximation of Scattered Data over Binary Triangulations

Received: date / Accepted: date

Abstract An adaptive method for approximating huge scattered data sets is presented. The approximation scheme generates multilevel triangulations obtained using a subdivision scheme known as longest edge bisection. Nested function spaces are defined over the multilevel triangulations. The approximation problem is solved by successive refinement of the triangulation while iterative methods are used for solving a system of linear equations at intermediate levels of the multi-level scheme. Regularization terms are coupled with a standard least squares formulation to guarantee uniqueness and control smoothness of the solution.

1 Introduction

The motivation for the approximation scheme proposed in this paper comes from surface modelling in geographic information systems (GIS) and systems for modelling geological structures. We want the method to handle huge scattered data sets with noise, data with uneven distribution, and handle situations where the surface topography varies rapidly over the domain.

Several multilevel methods for surface construction have been studied and presented over the past years, although relatively few deal with approximation of scattered data. In [17] *multilevel B-splines* were used to generate a coarse-to-fine sequence of tensor product B-splines whose sum approaches the final approximation of given scattered data. The method is local in the sense that on the finer tensor product grids each B-spline coefficient is computed from nearby points only. Consequently, the method is also fast. However, if the scattered data are subject to noise or if the data are unevenly distributed over the domain, undesirable behaviour may occur near data locations. Numerical examples and explanations of these phenomena can be found in [17] and [15].

Hierarchical B-splines were introduced in [8], and applied to approximation of data arranged on a regular grid in [9]. Application to scattered data was discussed in [12]. Contrary to multilevel B-splines, the basic idea with hierarchical B-splines is to refine a tensor product grid locally where the approximation error exceeds a specified tolerance. Thereby a global approximation problem is subdivided into a series of local problems. Each local problem gives rise to a (small) linear equation system. This is done recursively until the B-spline surface approximates the scattered data within the given tolerance. The data structure is rather complex and implementation of the method requires considerable more effort than for multilevel B-splines.

A scheme for constructing smooth regular grid functions approximating scattered data was presented in [1]. In a first local step called regularization, a subset of the grid values are determined from nearby points using a classic scattered data interpolation technique like Shepard's method [25] or the radial basis function method [21]. Then, in the next extrapolation step, the grid values found in the first step are extended to the entire grid by solving a biharmonic differential equation. The method can also handle constraints (break lines) imposed on the surface, which makes the method suited for modelling faulted geological structures.

An important part of the multilevel approximation scheme presented in this paper is the construction of a nested sequence of semi-regular triangulations. We want to apply a subdivision scheme that generates an adaptive triangulation where the triangle density reflects the variation in surface topography and distribution of the given data. A scheme known as *longest edge bisection* has become popular for view-dependent visualisation [18,23]. The scheme is also called $4-k$ meshes [26], or restricted quad-tree triangulations [20]. In this paper meshes generated by longest edge bisection are called *binary triangulations* since they can be considered as the result of recursive splitting of one triangle into two new triangles. One important property of the scheme, and variations over it, is that refinement can be done locally without the need to maintain the entire mesh at the same resolution.

Øyvind Hjelle · Morten Dæhlen
Simula Research Laboratory, P.O.Box 134, 1325 Lysaker, Norway
Tel.: +47 67 82 82 75
Fax: +47 67 82 82 01
E-mail: {oyvindhj mortend}@simula.no

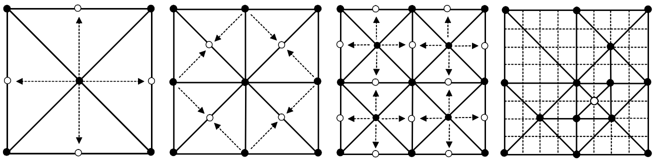

Fig. 1 Recursive longest edge bisection. Dashed arrows indicate parent-child relationships. The rightmost triangulation has only one active vertex from the finest refine level.

Consider the initial triangulation on the left in Fig. 1 with four isosceles triangles. The recursive splitting starts by inserting vertices on the longest edge of each triangle. To maintain a valid triangulation, new edges must be introduced as indicated by dashed arrows. The new triangulation has eight isosceles triangles, whose longest edges are bisected to obtain a new triangulation with 16 triangles (second from right). Each dashed arrow represents a parent-child relationship where the black bullets are *parents* and each circle represents a *child*. We say that a vertex is *active* when it has been inserted into the mesh. Every vertex at a certain refine level need not be active to obtain a valid triangulation. The general rule is that if a vertex is active, then its parents, its grandparents, etc. and all its ancestors at coarser levels must also be active. The four corner vertices are active by default. An example is shown by the rightmost binary triangulation in Fig. 1. For the vertex drawn as a circle to be active, all the vertices drawn as black bullets must also be active. A unique triangulation then results with edges determined by the parent-child relationships involved.

We have implemented an algorithm based on the fact that the triangle vertices at any level of a binary triangulation constitute a subset of a regular grid. Starting from the initial triangulation on the left in Fig. 1 we have a grid Ψ_1 with 2×2 rectangular grid cells. For each grid cell in Ψ_1 a criterion for subdivision is examined. Criteria for subdivision will be explained shortly. If subdivision is required within the grid cell, the vertex in the middle of the cell, say v, belonging to the next finer grid Ψ_2 with 4×4 grid cells is activated. Then the parents of v must also be activated and a unique triangulation results with vertices which are a subset of those in the triangulation shown second from right. The algorithm proceeds on finer and finer grids where each grid Ψ_k is obtained by inserting grid lines halfway between the grid lines of Ψ_{k-1}.

We have implemented a simple data structure for the binary triangulation similar to that used in [19] for view-dependent visualisation. A triangulation is represented implicitly only by its (active) vertices and references from each vertex to its two parents. Algorithms operating on the data structure also become simple and efficient, for example algorithms for point localization and algorithms for generating triangle strips for efficient visualisation.

The general scheme for computing surface triangulations over binary triangulations can briefly be outlined as follows. Given scattered data $P = \{(x_i, y_i)\}$ with corresponding real values $\{z_i\}$. Over each triangulation Δ_k in a coarse-to-fine sequence of binary triangulations we construct a least squares surface approximation f_k to the given data. Dependent on the application, different subdivision criteria can be used. For example, a prescribed tolerance may be given to control the maximum deviation between the surface approximation at the finest level and the given data values $\{z_i\}$. Thus, the accuracy of the approximation at one level can be used to decide where to refine, hence to produce the triangulation at the next finer level. For huge data sets it is important to reduce the amount of data or to create a triangulation that reflects the distribution of the scattered data. One way to achieve this, without taking any error measure into account, is to use a counting measure whereby Δ_k is refined as long as there is more than a prescribed number of data locations from P within some portion of the domain, for example inside a grid cell of Ψ_k. Variation over curvature measures could also be used to decide where to refine, for example by using a thin-plate energy measure locally over triangle edges. Since iterative methods are used for computing the least squares approximations, the solution found at one level can be used to compute a starting vector for the next level. More details on the numerical scheme and numerical examples are given in sections 3 and 4, respectively.

2 The Least Squares Approximation

In this paper the least squares approximation is restricted to piecewise linear surfaces. In geographic information systems and geological modelling systems, piecewise linear surfaces are often sufficient and may also be preferred to higher degree surfaces, which may cause undesirable oscillations (e.g. Gibbs phenomena) when modelling terrain with rapidly varying topography. Piecewise linear surfaces are also more efficient to compute and more compliant with other software components in such applications.

Given a set of distinct non-collinear data points $P = (x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m),\ m \geq 3$, in the plane and corresponding real values $z_1, z_2, \ldots, z_m$, we seek a function f over the binary triangulation Δ that approximates the data. We restrict f to the finite dimensional space

$$S_1^0(\Delta) = \left\{ f \in C^0(\Omega) : f|_{t_i} \in \Pi_1 \right\},$$

where Π_1 is the space of bivariate linear polynomials, t_i is a triangle in Δ and Ω is a rectangular domain. As a basis for $S_1^0(\Delta)$ we use standard compactly supported nodal basis functions $N_1(x,y), N_2(x,y), \ldots, N_n(x,y)$ which satisfy $N_i(v_j) = \delta_{ij},\ j = 1, \ldots, n$, where the $v_j = (x_j, y_j)$ are vertices in the underlying triangulation. Thus, a function in $S_1^0(\Delta)$ is written,

$$f(x,y) = \sum_{i=1}^{n} c_i N_i(x,y), \qquad c_i \in R.$$

In the basic least squares problem we seek a coefficient vector $\mathbf{c} = (c_1, \ldots, c_n)^T$ that minimizes

$$\sum_{k=1}^{m} \left(f(x_k, y_k) - z_k \right)^2. \tag{1}$$

This classical least squares fitting problem always has a solution, however the solution is not necessarily unique. Also, the resulting surface $f(x,y)$ may not be sufficiently smooth, especially if the given data are subject to noise. We are particularly interested in different types of measured data, which often contain noise. One may augment (1) with a regularization term, also called *smoothing term* or *penalty term* in [27], to guarantee uniqueness and control smoothness of the solution. Many commonly used smoothing terms can be written on the quadratic form

$$\mathscr{J}(\mathbf{c}) = \mathbf{c}^T\mathbf{E}\mathbf{c}, \tag{2}$$

where $\mathbf{E}$ is a symmetric and positive semidefinite $n \times n$ matrix. When working with spaces of functions of higher degree, the smoothing term usually involves first and second derivatives. For the piecewise linear space $S_1^0(\Delta)$ we may use discrete analogues where $\mathscr{J}(\mathbf{c})$ is the sum of some roughness measure around each vertex or across each interior edge of the triangulation. The functional we want to minimize is now

$$I(\mathbf{c}) = \sum_{k=1}^{m} \left(f(x_k,y_k) - z_k\right)^2 + \lambda\mathbf{c}^T\mathbf{E}\mathbf{c} = \|\mathbf{B}\mathbf{c} - \mathbf{z}\|_2^2 + \lambda\mathbf{c}^T\mathbf{E}\mathbf{c},$$

for some real value $\lambda \geq 0$. Here $\mathbf{z} = (z_1,\ldots,z_m)^T$ and $\mathbf{B}$ is the $m \times n$ matrix

$$\mathbf{B} = \begin{pmatrix} N_1(x_1,y_1) & \cdots & N_n(x_1,y_1) \\ N_1(x_2,y_2) & \cdots & N_n(x_2,y_2) \\ \vdots & & \vdots \\ N_1(x_m,y_m) & \cdots & N_n(x_m,y_m) \end{pmatrix}. \tag{3}$$

Setting the gradient of $I(\mathbf{c})$ equal to zero leads to the normal equations

$$\left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E}\right)\mathbf{c} = \mathbf{B}^T\mathbf{z}. \tag{4}$$

The $n \times n$ matrix $\mathbf{B}^T\mathbf{B}$ is symmetric and positive semidefinite, so the solution to (4) with $\lambda = 0$ is not necessarily unique. If $\mathbf{E}$ is positive definite, then the system matrix $\left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E}\right)$ with $\lambda > 0$ is also positive definite, and thereby nonsingular, which implies that (4) has a unique solution.

To justify the use of the smoothing term for controlling the behaviour of the least square fit, we briefly go through a result by Golitschek & Schumaker [27]. Assume for now that $\left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E}\right)$ is nonsingular for $\lambda \geq 0$. Let $f_\lambda(x,y)$ be the penalized least squares fit to the given data with the smoothing parameter λ. Then the mean square error given by

$$T_{\mathbf{z}}(\lambda) = \frac{1}{m}\sum_{k=1}^{m}\left(f_\lambda(x_k,y_k) - z_k\right)^2 \tag{5}$$

is monotone increasing for $\lambda \geq 0$ with derivatives $T_{\mathbf{z}}'(0) = 0$ and $\lim_{\lambda\to\infty} T_{\mathbf{z}}'(\lambda) = 0$. Thus, λ controls the trade-off between smoothness and mean square error. A best fit, best in the sense of minimizing the mean square error, is then obtained by setting the value of the smoothing parameter λ to zero.

It can be shown that the basic least squares problem $\mathbf{B}^T\mathbf{B}\mathbf{c} = \mathbf{B}^T\mathbf{z}$ without a smoothing term has a unique solution if and only if $\mathbf{B}$ has linearly independent columns [11]. If one or more columns of $\mathbf{B}$ are zero vectors, then the columns are not linearly independent. We observe from (3) that each basis function N_j, $j = 1,\ldots,n$ makes a column. A column, say column number l, is a zero vector if $N_l(x_i,y_i) = 0$ for $i = 1,\ldots,m$. This happens if no point of P falls strictly inside the domain Ω_l of N_l, which often occurs for cartographic data and geological data whose distribution may vary rapidly over the domain. For instance, hypsographic data (contour lines) in cartography and seismic data (track data) in geology typically possess such uneven distributions. So in most cases, when constructing surfaces from such data, we need a smoothing parameter λ greater than zero to guarantee uniqueness of (4).

An approximation scheme without a smoothing term, based on refinement of arbitrary triangulations, was proposed by Rippa [22]. Uniqueness then relied on the fact that the vertices of the triangulation were always chosen as a subset of the input data. Consequently, there are n rows of $\mathbf{B}$ that constitute the identity matrix (with a proper ordering of the rows), and the columns are trivially linearly independent.

The system matrix $\mathbf{B}^T\mathbf{B}$ of the basic least squares problem is clearly sparse. Indeed, an element

$$\left(\mathbf{B}^T\mathbf{B}\right)_{ij} = \sum_{k=1}^{m} N_i(x_k,y_k)N_j(x_k,y_k)$$

is non-zero only if $i = j$ or if the domains of the basis functions N_i and N_j overlap so that two triangles share a common edge. Thus, a non-zero off-diagonal element $(\mathbf{B}^T\mathbf{B})_{ij}$ corresponds to an edge in the triangulation connecting the two vertices v_i and v_j. It can be shown that the number of edges in a triangulation with V vertices has an upper bound $3V - 6$. Then the number of non-zero off-diagonal elements in the $n \times n$ matrix $\mathbf{B}^T\mathbf{B}$ has an upper bound $2(3n-6)$, and counting the diagonal which is also non-zero, we find that the average number of non-zeros in each row is approximately 7.

2.1 Smoothing Terms

Smoothing terms on the quadratic form in (2) can be obtained from the *membrane energy*,

$$\int |\nabla g|^2 = \int g_x^2 + g_y^2, \tag{6}$$

and from an approximation of the *thin-plate energy*,

$$\int g_{xx}^2 + 2g_{xy}^2 + g_{yy}^2. \tag{7}$$

Loosely speaking, the membrane energy prefers surfaces with small area, while the thin-plate energy prefers surfaces with

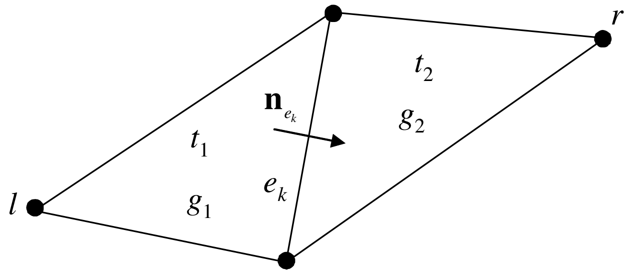

Fig. 2 Stencil for the second order divided difference operator

small overall curvature. Since functions in $S_1^0(\Delta)$ are not twice differentiable we will use an approximation to the thin-plate energy based on a divided difference.

The smoothing term based on the membrane energy can be expressed on the following quadratic form,

$$\begin{aligned}\mathscr{J}_1(\mathbf{c}) &= \int |\nabla f|^2 = \int \left|\sum_{i=1}^{n} c_i \nabla N_i\right|^2 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \int \left(\nabla N_i \cdot \nabla N_j\right) c_i c_j = \mathbf{c}^T\mathbf{E}\mathbf{c}.\end{aligned}$$

Thus, the elements of the smoothing matrix are

$$E_{ij} = \int \nabla N_i \cdot \nabla N_j.$$

Matrix $\mathbf{E}$ is symmetric and sparse with the same sparsity pattern as $\mathbf{B}^T\mathbf{B}$, thus E_{ij} is non-zero when $i = j$ or when (i, j) corresponds to an edge in the triangulation.

An approximation to the thin-plate energy functional in (7) can be based on a second order divided difference operator. Fig. 2 shows a stencil with the triangles and vertices involved in the construction of the divided difference. Let $g_i = f|_{t_i}$, $i = 1, 2$, where t_1 and t_2 are the triangles sharing the edge e_k. Further, let $\mathbf{n}_{e_k}$ be a unit vector in the xy-plane orthogonal to the projection of e_k in the xy-plane, and let $|E_I|$ be the number of interior edges in the triangulation. We define the discrete thin-plate energy as the following sum over all interior edges,

$$\mathscr{J}_2(\mathbf{c}) = \sum_{k=1}^{|E_I|} \int_{e_k} \left[\nabla f \cdot \mathbf{n}_{e_k}\right]_J^2$$

where $\left[\nabla f \cdot \mathbf{n}_{e_k}\right]_J = (\nabla g_2 - \nabla g_1) \cdot \mathbf{n}_{e_k}$ is the jump in the derivative of f over the interior edge e_k in the direction of $\mathbf{n}_{e_k}$. This measure was used in [5] for constructing data dependent triangulations, and equivalent measures were used in [13], and in [14] as divided difference operators in subdivision schemes for triangulations. We obtain the following quadratic form,

$$\begin{aligned}\mathscr{J}_2(\mathbf{c}) &= \sum_{k=1}^{|E_I|} \int_{e_k} \left[\sum_{i=1}^{n} c_i (\nabla N_i \cdot \mathbf{n}_{e_k})\right]_J^2 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{|E_I|} \int_{e_k} \left[\nabla N_i \cdot \mathbf{n}_{e_k}\right]_J \left[\nabla N_j \cdot \mathbf{n}_{e_k}\right]_J c_i c_j = \mathbf{c}^T\mathbf{E}\mathbf{c}\end{aligned}$$

where

$$E_{ij} = \sum_{k=1}^{|E_I|} \int_{e_k} \left[\nabla N_i \cdot \mathbf{n}_{e_k}\right]_J \left[\nabla N_j \cdot \mathbf{n}_{e_k}\right]_J.$$

In general, an element E_{ij} is non-zero if $i = j$, or if i and j correspond to vertices of the same stencil. This occurs if the line segment (v_i, v_j) spans an edge, or if v_i is the vertex on the opposite side of an edge from v_j. The latter case corresponds to the vertices with indices l and r in Fig. 2. Compared to the operator for the membrane energy, this operator generates a sparsity pattern with more non-zero entries in the system matrix.

2.2 Uniqueness

It follows from basic linear algebra that since both matrices $\mathbf{B}^T\mathbf{B}$ and $\mathbf{E}$ are positive semidefinite, the system matrix $\left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E}\right)$ for $\lambda > 0$ can also be positive semidefinite. But here we show through geometric analysis that the system matrix is strictly positive definite such that the system of equations in (4) has a unique solution.

For $(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E})$ to be positive semidefinite, we must have

$$\mathbf{c}^T\left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E}\right)\mathbf{c} = \mathbf{c}^T\left(\mathbf{B}^T\mathbf{B}\right)\mathbf{c} + \lambda\mathbf{c}^T\mathbf{E}\mathbf{c} = 0, \tag{8}$$

where both terms $\mathbf{c}^T(\mathbf{B}^T\mathbf{B})\mathbf{c} = 0$ and $\mathbf{c}^T\mathbf{E}\mathbf{c} = 0$ for some vector $\mathbf{c} \neq \mathbf{0}$. We first observe that $\mathbf{c}^T\mathbf{E}\mathbf{c} = \mathscr{J}(\mathbf{c})$, the general energy term.

The membrane energy generates the functional

$$\mathscr{J}_1(\mathbf{c}) = \sum_{k=1}^{|T|} A_k \left|\nabla g_k\right|^2 = \sum_{k=1}^{|T|} A_k \left[(\partial g_k/\partial x)^2 + (\partial g_k/\partial y)^2\right].$$

For this expression to be zero we must have $\partial g_k/\partial x = 0$ and $\partial g_k/\partial y = 0$ for $k = 1, \ldots, |T|$. But then, since the triangulation is connected, all coefficients must be equal. Thus, $\mathbf{c}^T\mathbf{E}\mathbf{c} = 0$ if and only if $c_1 = c_2 = \cdots = c_n$. Furthermore, for the first term in (8) stemming from the basic least squares problem, we must have

$$\mathbf{c}^T(\mathbf{B}^T\mathbf{B})\mathbf{c} = ||\mathbf{B}\mathbf{c}||_2^2 = 0, \tag{9}$$

which implies that $(\mathbf{B}\mathbf{c})_j = \sum_{i=1}^{n} c_i N_i(x_j, y_j) = f(x_j, y_j) = 0$ for all $j = 1, \ldots, m$. But since all coefficients c_k are equal, the function f must then be zero everywhere and all coefficients c_k must be zero. Thus, $\mathbf{c}^T(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E})\mathbf{c} = 0$ implies that $\mathbf{c}$ =0 and we conclude that the system matrix $(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E})$ is indeed positive definite, and therefore nonsingular. Hence, the system of equations (4) always has a unique solution when $\mathbf{E}$ is constructed from the membrane energy.

The discrete thin-plate operator generates the functional

$$\mathscr{I}_2(\mathbf{c}) = \sum_{k=1}^{|E_I|} \int_{e_k} \left[(\nabla g_2 - \nabla g_1)\cdot \mathbf{n}_{e_k}\right]^2 .$$

For this expression to be 0 for $\mathbf{c} \neq 0$, and thus for $\mathbf{E}$ to be positive semidefinite, we must have $(\nabla g_2 - \nabla g_1) = \mathbf{0}$ over all interior edges in the triangulation. Then, since the triangulation is connected, all the gradients of $f|_{t_i}$ over all triangles t_i in Δ are equal and f must be a linear polynomial. But then (9) again implies that $f(x_j, y_j) = 0$ for $j = 1,\dots,m$ under the same assumption that $\mathbf{B}^T\mathbf{B}$ is positive semidefinite. This implies, that f is zero everywhere and that all coefficients in $\mathbf{c}$ must be zero. Thus, we arrive at the same conclusion as above that the system matrix $(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E})$ is positive definite and that (4) always has a unique solution when $\mathbf{E}$ is constructed from the thin-plate energy functional.

3 Numerical Schemes

At each level k of the multilevel scheme, the system of equations in (4) is relaxed with the Gauss-Seidel method (or alternatively the conjugate gradient method). A nested iteration scheme has been implemented where the resulting surface $f_k \in S_1^0(\Delta_k)$ at one level is used to supply an initial guess for $f_{k+1} \in S_1^0(\Delta_{k+1})$ at the next level. Since all vertices of the triangulation Δ_k are also in Δ_{k+1}, all coefficients from f_k are transferred directly to the next level as starting values for the unknown coefficients of f_{k+1}. Moreover, each coefficient c_{ij}^{k+1} of f_{k+1} corresponding to a vertex (x_i, y_j) in Δ_{k+1} that is not in Δ_k, is simply initialized as $c_{ij}^{k+1} = f_k(x_i, y_j)$. Since the triangulations generated by the binary subdivision scheme are nested, the function spaces defined over them constitute a nested sequence of subspaces, $S_1^0(\Delta_1) \subset S_1^0(\Delta_2) \subset \cdots \subset S_1^0(\Delta_h)$, where Δ_h is the triangulation at the finest level. In addition to providing fast solution of the linear equation system, this coarse-to-fine scheme also generates a sequence of surface approximations to the scattered data at different levels of detail.

In [27] generalized cross validation is proposed for computing the smoothing parameter λ. This method is rather CPU-extensive. In our implementation we use a simpler approach adopted from [7], where the default value is set to

$$\lambda_d = \left\|\mathbf{B}^T\mathbf{B}\right\|_F / \|\mathbf{E}\|_F , \tag{10}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm. The idea is that the contributions from $\mathbf{B}^T\mathbf{B}$ and $\lambda\mathbf{E}$ to the system matrix in (4) should have roughly the same weight. For data without noise this value works well in most cases, but with noise present a much larger λ, up to 1000 times λ_d, is necessary to obtain a sufficiently smooth solution.

At the first levels of the multilevel scheme the system is solved to yield an exact solution. This establishes a good global trend of the surface as a basis for successive improvements when iterating at the finer levels with more unknowns. A combined stopping criterion based on relative improvement of the solution and decrease of the residual, measured by the l_2 norm, was used for the iterative solver. The Gauss-Seidel method performed better than the conjugate gradient method as an iterative solver, even though the conjugate gradient method converges faster when applied on a fixed mesh without a good initial guess. In most cases, between 10 and 20 iterations at each level were sufficient. Any further iterations did not improve the solution significantly when the smoothing parameter was chosen as in (10). But with larger λ the number of iterations at each level was higher, e.g. up to 200 when λ was between 500 and 1000 times λ_d.

A natural improvement of this simple coarse-to-fine ascending scheme is to include recursive coarse grid correction at each level and thus obtain a true *geometric multigrid* solver [4]. To ease implementation we used a standard *algebraic multigrid* solver from the software library ML [16], by which the coarse grid correction step need not be provided explicitly by the user. This library is generic in the sense that its algorithms can operate on any data structure for vectors and matrices if a specified interface is provided by the user. Both full multigrid and repeatedly V-cycles and W-cycles are available in ML. The algorithms, which takes as input a system of equations at the finest level only, run with a fixed number of iterations at each level specified by the user, for example between five and ten Gauss-Seidel iterations as in our examples. For details on the theory of algebraic multigrid, see for example [3,2].

With the default smoothing parameter in (10), the algebraic multigrid schemes and the simple coarse-to-fine scheme outlined above were approximately equally good. But for larger λ the multigrid schemes were significantly faster. An advantage with multigrid, if V-cycles or W-cycles are employed, is that an initial guess for the solution at the finest level can be given when starting the solver. A good approximation of the coefficient in each triangle vertex is easily obtained in most cases by a fast local approximant or interpolant that uses nearby scattered data only, for example Shepard's method [25].

Due to an ill-condition system there is no clear correspondence between the residual and the error, and thus, the number of repeated V-cycles or W-cycles might be difficult to determine. A solution in practical applications is to examine by visual inspection if the resulting surface triangulation is sufficiently smooth and pleasant looking, and then decide if additional multigrid cycles should be performed.

4 Numerical examples

In this section we present numerical examples based on three different data sources: data sampled from the well known Franke's test function [10]; real data from a terrain consist-

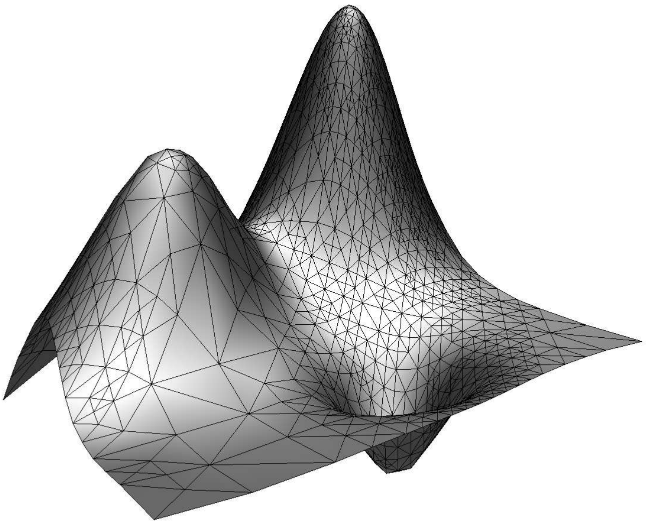

Fig. 3 Approximation to the Franke function, and the resulting binary triangulation imposed on the surface

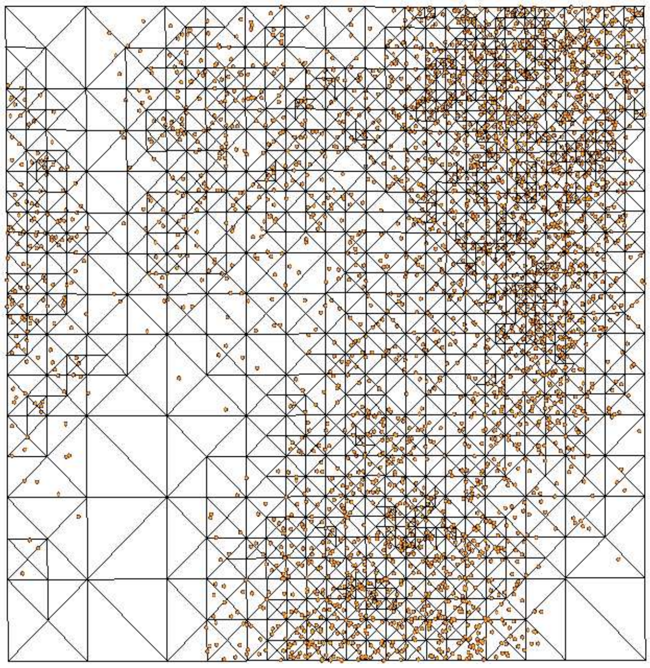

Fig. 4 The same binary triangulation as in Fig. 3 together with input data for numerical examples. The lower left corner corresponds to the nearest corner in Fig. 3.

ing of a combination of hypsographic data (contour data) and scattered measurements from the terrain surface; and the last example uses parametrized 3D scattered data. The regularization term is based on the thin-plate energy in all examples.

Approximation of data sampled from Franke's function. (Consult Figs. 3, 4) The scattered data set consists of 3000 points sampled over the unit square. The points are unevenly distributed in the domain with relatively more data in areas with steep gradient or high curvature. A combined subdivision criterion based on a counting measure and an error measure is used. A grid cell is thus refined if more than two points are inside the grid cell and the error for at least one of the points inside the cell is greater than the prescribed tolerance. The tolerance was 0.25 percent of $|z_{\max} - z_{\min}|$ of the given data, and the default value for λ was used. The coarse-to-fine algorithm terminated after subdivision of the grid Ψ_7 and delivered a triangulation Δ_8 with 1476 vertices and 2866 triangles. The number of Gauss-Seidel iterations at each level was between 9 and 12 to reach the stopping criterion. As expected, there are more triangles in areas with large curvature and high density of data due to the combined error and counting measure used as a subdivision criterion. Also note the nice spatial grading from small triangles to larger triangles in Fig. 4. If the middle vertex of all grid cells were activated when operating on Ψ_7, the resulting triangulation would have 33025 vertices, which also would be the number of unknowns in the equation system at that level. Thus, less than 4.5 percent of the maximum number of available vertices are used in the triangulation.

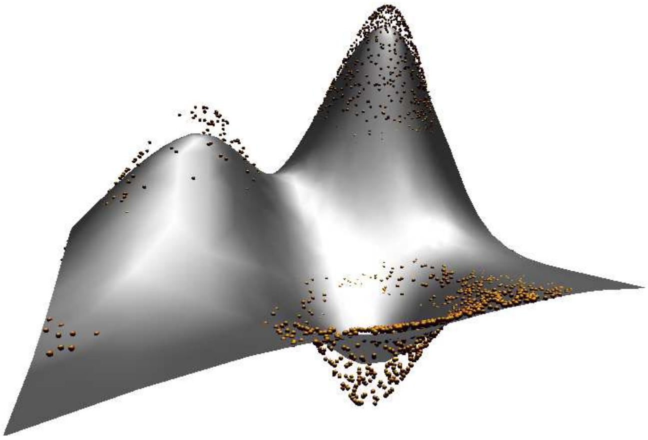

Fig. 5 Approximation with huge smoothing parameter

Fig. 5 demonstrates the effect of choosing a very large λ (10 000 times λ_d), and thus demanding much smoothing. The surface leaves the given data points, and for even larger λ when the smoothing term becomes more dominant, the surface approaches a plane. Also recall that the mean square error given by (5) increases monotonically towards a maximum with increasing λ.

Approximation of noisy data from Franke's function. Normally distributed noise was added to the data set used in the previous example. Subdivision of a grid cell was performed when there was more than two points inside the cell, but no error measure was used. To obtain a smooth pleasant looking surface comparable to the surface produced in the previous example, it was necessary to increase λ to 600 times λ_d. The number of Gauss-Seidel iterations at each level with the coarse-to-fine scheme was between 73 and 207. The algorithm terminated at the same level as in the previous exam-

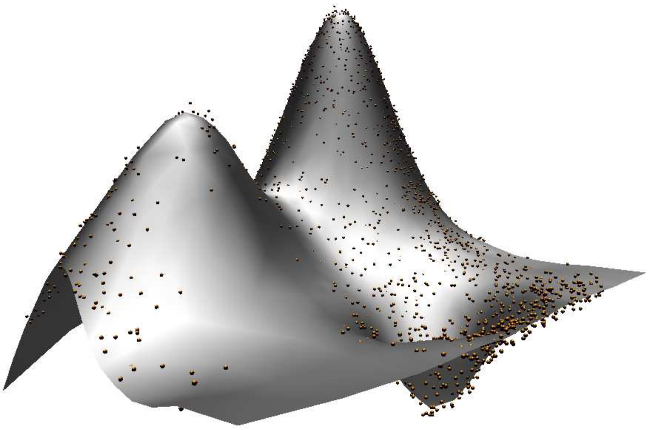

Fig. 6 Approximation to Franke's function from a data set with noise

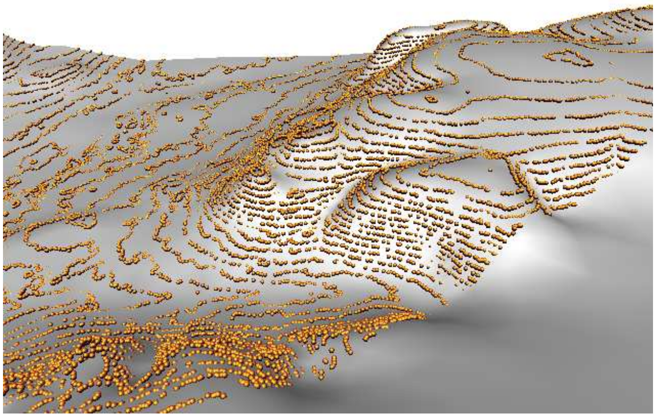

Fig. 7 Approximation of terrain data, and the given data imposed on the surface

ple. Fig. 6 shows the approximation and the given noisy data points.

When algebraic multigrid was used starting from the triangulation produced at the finest level, 5 V-cycles with 6 Gauss-Seidel iterations at each level were necessary to obtain the same residual norm. (The initial guess was simply the mean value of all data values $\{z_i\}$.) When more noise was added to the data and λ was increased, multigrid was even more superior.

Terrain modelling from hypsographic data and scattered data. The terrain model shown in figures 7 and 8 was derived from approximately 50 000 points consisting of both hypsographic data and scattered data points measured from the underlying terrain. An error measure was used as a subdivision criterion although the data contained noise. An acceptable smooth surface was obtained with a smoothing factor 10 times λ_d. The number of Gauss-Seidel iterations at each level was between 41 and 593 by the coarse-to-fine scheme. With an error tolerance of 2.5 percent of $|z_{\max} - z_{\min}|$, the algorithm terminated after subdivision at level 11 and delivered the triangulation Δ_{12} with 31204 vertices. Relatively few subdivisions were done at the last two levels to capture

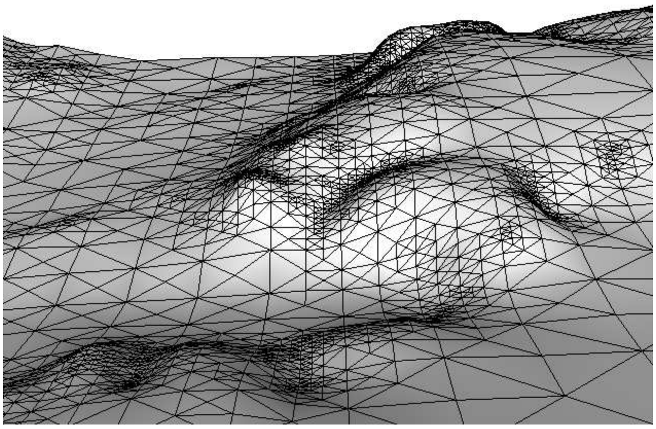

Fig. 8 Approximation of terrain data, and triangulation imposed on the surface

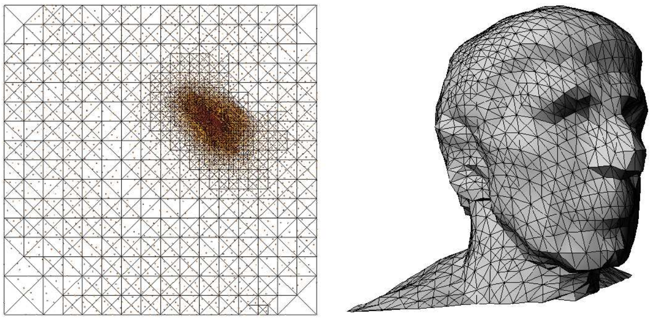

Fig. 9 Approximation of parametrized 3D scattered data

remaining details in the terrain and meet the given tolerance. The number of vertices in Δ_{12} is only 1.5 percent of the maximum number of available vertices at that level. Algebraic multigrid (V-cycles) performed slightly better with $\lambda = 10\lambda_d$, but when λ was increased algebraic multigrid was superior. The triangulation shown in Fig. 8 was produced by a larger tolerance to avoid too many triangles in the presentation. The mesh is finer in areas with rapidly varying topography, and thereby captures the necessary details. We also observe the natural extrapolation of the surface to areas without input data, which is due to the thin-plate energy. Algorithms with good extrapolation properties are important in many applications. For example, in geological modelling faults and horizons must be extended to intersect each other with clean cuts outside their initial domain when creating boundary-based volume models [24].

Approximation of parametrized 3D scattered data. A useful application of the adaptive properties of the multilevel scheme is demonstrated in Fig. 9. The scattered data shown in the parameter space on the left are parametrizations of scattered data points sampled from the 3D object on the right. Thus a one-to-one mapping exists between scattered data points in 3D space and the 2D parametric space. The mapping was computed by a method called "shape-preserving" parametrization by Floater [6]. A characteristic of this method is that nearby points are mapped closer and closer together in the parameter space as the distance between the 3D scattered data and the boundary of the 3D object increases. The samples in the example were relatively uniform distributed on the 3D object, while the parametrized points ends up in a cluster in the parameter space.

The approximation scheme was run by treating each of the space dimensions x, y and z separately at each level. A counting measure was used as a subdivision criterion (based on number of points in the parameter space) to ensure that the same binary triangulation was used for each space dimension at each level. The triangulation obtained at the finest level is shown on the left in Fig. 9, and on the right the mapping of the result back to 3D space is shown. While the size of the triangles in the 2D parameter space vary heavily, we observe that they have approximately the same size when mapped back to 3D space.

5 Concluding Remarks

The novelty of the method presented in this paper lies in the use of binary triangulations as an effective tool for generating a nested sequence of triangulations used in a multilevel scheme for solving the scattered data approximation problem. Binary triangulations give rise to extremely simple data structures. The fact that binary triangulations are standard tools for view-dependent visualisation, also makes the resulting surfaces well suited for fast rendering. In particular the method is efficient when fitting surfaces to huge scattered data sets and data unevenly distributed over the domain. When using subdivision criteria based on error measure or counting measure, the triangle density adapts automatically to the distribution of the data with nice spatial grading as can be seen in Fig. 3 and 8. The triangle density also reflects the variation in surface topography. Another useful feature observed by numerical experiments is the natural extrapolation of the surface to areas without data. We emphasize the practical relevance of these qualities in applications like geological modelling and approximation of cartographic data.

Even though the simple coarse-to-fine scheme works well for data without noise, convergence is significantly improved by using a standard algebraic multigrid solver when a large smoothing parameter λ is necessary for noisy data. We would probably benefit even more from a geometric multigrid solver based on the triangular grids produced by our coarse-to-fine scheme. Another interesting topic for further research is to study methods for computing a good λ when approximating data with noise.

Acknowledgements The authors thank Kent-Andre Mardal at Simula Research Laboratory for fruitful discussions and advices on applying algebraic multigrid, and an anonymous referee for suggestions on improvements of the paper.

References

1. Arge, E., Dæhlen, M., Tveito, A.: Approximation of scattered data using smooth grid functions. Journal of Computational and Applied Mathematics **59**, 191–205 (1995)
2. Brandt, A.: Algebraic multigrid theory: The symmetric case. Applied Mathematics and Computation **19**, 23–56 (1986)
3. Brandt, A., McCormick, S.F., Ruge, J.: Algebraic multigrid (AMG) for sparse matrix equations. In: D.J. Evans (ed.) Sparsity and its Applications, pp. 257–284. Cambridge University Press (1984)
4. Briggs, W.L., Henson, V.E., McCormick, S.F.: A multigrid tutorial: second edition. Society for Industrial and Applied Mathematics (2000)
5. Dyn, N., Levin, D., Rippa, S.: Data dependent triangulations for piecewice linear interpolation. IMA Journal of Numerical Analysis **10**, 137–154 (1990)
6. Floater, M.S.: Parametrization and smooth approximation of surface triangulations. Computer Aided Geometric Design **14**, 231–250 (1997)
7. Floater, M.S.: How to approximate scattered data by least squares. Tech. Rep. STF42 A98013, SINTEF, Oslo (1998)
8. Forsey, D.R., Bartels, R.H.: Hierarchical B-spline refinement. In: ACM Computer Graphics (SIGGRAPH '88 Proceedings), pp. 205–212 (1988)
9. Forsey, D.R., Bartels, R.H.: Surface fitting with hierarchical splines. ACM Transactions on Graphics **14**, 134–161 (1995)
10. Franke, R.: Scattered data interpolation: Tests of some methods. Math. Comp. **38**, 181–200 (1982)
11. Golub, G.H., Loan, C.F.: Matrix Computations, third edn. John Hopkins University Press, Baltimore and London (1996)
12. Greiner, G., Hormann, K.: Interpolating and approximating scattered 3D data with hierarchical tensor product B-splines. In: A. Méhauté, C. Rabut, L.L. Schumaker (eds.) Surface Fitting and Multiresolution Methods, pp. 163–172 (1997)
13. Guskov, I.: Multivariate subdivision schemes and divided differences. Tech. rep., Department of Mathematics, Princeton University (1998)
14. Guskov, I., Sweldens, W., Schröder, P.: Multiresolution signal processing for meshes. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 325–334. ACM Press/Addison-Wesley Publishing Co. (1999)
15. Hjelle, Ø.: Approximation of scattered data with multilevel B-splines. Tech. Rep. STF42 A01011, SINTEF Applied Mathematics, Oslo (2001)
16. Hu, J., Tong, C., Tuminaro, R.S.: ML 2.0 Smoothed aggregation user's guide. Tech. Rep. SAND2001-8028, Sandia National Laboratories, Albuquerque NM (2000)
17. Lee, S., Wolberg, G., Shin, S.Y.: Scattered data interpolation with multilevel B-splines. IEEE Transactions on Visualization and Computer Graphics **3**(3), 229–244 (1997)
18. Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L.F., Faust, N., Turner, G.A.: Real-time, continuous level of detail rendering of height fields. In: ACM SIGGRAPH 96, pp. 109–118 (1996)
19. Lindstrom, P., Pascucci, V.: Terrain simplification simplified: A general framework for view-dependent out-of-core visualization. IEEE Transactions on Visualization and Computer Graphics **8**(3), 239–254 (2002)
20. Pajarola, R.B.: Large scale terrain visualization using the restricted quadtree triangulation. In: Proceedings of the conference on Visualization '98, pp. 19–26. IEEE Computer Society Press (1998)
21. Powell, M.J.D.: The theory of radial basis function approximation in 1990. In: W. Light (ed.) Advances in Numerical Analysis, Vol II, pp. 105–210. Oxford Science Publications (1992)
22. Rippa, S.: Adaptive approximation by piecewise linear polynomials on triangulations of subsets of scattered data. SIAM Journal on Scientific and Statistical Computing **18**(3), 1123–1141 (1992)
23. Röttger, S., Heidrich, W., Slusallek, P., Seidel, H.P.: Real-time generation of continuous levels of detail for height fields. In:

V. Skala (ed.) Proceedings of 1998 International Conference in Central Europe on Computer Graphics and Visualization, pp. 315–322 (1998)

24. Schneider, S.: Pilotage automatique de la construction de modèles géologiques surfaciques. Ph.D. thesis, Université Jean Monnet et Ecole Nationale Supérieure des Mines de Saint-Etienne (2002)
25. Shepard, D.: A two-dimensional interpolation function for irregularly spaced data. In: Proc. 23rd Nat. Conf. ACM, pp. 517–524 (1968)
26. Velho, L., Gomes, J.: Variable resolution $4-k$ meshes: Concepts and applications. Computer Graphics Forum **19**(4), 195–212 (2000)
27. Von Golitschek, M., Schumaker, L.L.: Data fitting by penalized least squares. In: J.C. Mason, M.G. Cox (eds.) Algorithms for Approximation II, pp. 210–227. Chapman & Hall (1990)